UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

SCHEDULING WORKFLOWS ON A CLUSTER OF MEMORY MANAGED MULTICORE MACHINES

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By
HIRA KAJI SHRESTHA
Norman, Oklahoma
2009

SCHEDULING WORKFLOWS ON A CLUSTER OF MEMORY MANAGED MULTICORE MACHINES

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY

Dr. John K. Antonio - Chair

Dr. Sridhar Radhakrishnan

Dr. Amy McGovern

Acknowledgments

I would like to thank my advisor, Dr. John K. Antonio, for his proper guidance and support throughout my graduate studies. Dr. Antonio not only gave me interesting research ideas and feedback, but also enriched my knowledge with valuable suggestions and keen supervision. Dr. Antonio has always been a highly motivating person to me ever since I have joined this research team.

I would also like to thank the other members of our research team: Tom Stockdale, Jeff Muehring, Nicolas Grounds, Jay Sachs, Jason Madden, Matthew Martin, Carlos Sanchez, Josh Zuech, Kelly Crawford, Brett Marcott, Swathikah Thangaraja and Zhenbo Xing for their contribution to this thesis. I heartily appreciate all the discussions, meetings and presentations which are proved to be greatly helpful in this research work.

I would like to thank my thesis committee members Dr. Sridhar Radhakrishnan and Dr. Amy McGovern. I appreciate their questions, comments and advice.

I am very grateful to my parents Thulo Babu Shrestha and Hari Maya Shrestha, my brother Ram Kumar Shrestha, my uncle Tej B. Shrestha and my sister Santoshi Maya Shrestha for their unconditional love and support.

I am very thankful to all my friends for their support and well wishes. Special thanks goes to Pushma Bajracharya for her help and support.

I would like to thank faculty members, staff members and students of the computer science department. Special thanks goes to Barbara Bledsoe, Chyrl Yerdon and Jim Summers for their help and support during my graduate studies.

Contents

List of Tables

List of Figures

Abstract

Workflows are modeled with directed acyclic graphs in which vertices represent computational tasks, referred to as requests, and edges represent precedent constraints among requests. Associated with each workflow is a deadline that defines the time by which all computations of a workflow should be complete. Workflows are submitted by numerous clients to a scheduler that assigns workflow requests to a cluster of memory managed multi-core machines for execution. The objective of the scheduler is to minimize missed workflow deadlines. The characteristics of workflows are assumed to vary along several dimensions, including: arrival rate, periodicity, degree of parallelism, and number of requests. For the purpose of this thesis, an overall scheduling policy is defined by two underlying components: (1) a request selection policy and (2) a machine selection policy. A total of forty-two scheduling policies are evaluated, which are defined according to the cross-product of seven underlying request selection policies and six machine selection policies. The performance of each policy is determined through extensive simulation studies. All of the machine selection policies rely on a specified loading threshold. The simulation studies conducted reveal the existence of an optimal threshold value for each machine selection policy that results in relatively comparable performance across all machine selection policies, for the scenarios considered. Of the seven request selection policies evaluated, three are newly introduced, and the remaining are derived from previously known policies from the literature. From the studies conducted, it is determined that one of the newly introduced request selection policies outperforms the others evaluated in

terms of minimizing a measure of normalized tardiness.

Chapter 1

Introduction and Motivation

1.1 Introduction

The demand of computing power is increasing day by day due to the widespread use and innovative applications of computer technology. With the increasing demand of processing capacity, hardware manufacturers are trying to keep pace with Moores' law [1, 2] by delivering faster processors. To fulfill computational demands, multiple processors can be integrated into one system. Computer systems having multiple processors are called parallel systems [3]. Fig. 1.1 illustrates an example of parallel processing. In the figure there are seven mathematical calculation blocks that are represented in a graph structure. The edges in the graph define the data dependency among the blocks. After the execution of the first block, the following five calculations can be executed in parallel. Fig. 1.1 shows the execution of those parallel computations using a quad core machine.

Parallel systems can be linked together to provide massive computational resources called distributed systems [3]. When users send processing requests to these systems, the allocation of resources to requests becomes very important. The decision making process that assigns requests to the available resources over time is referred to as scheduling. When the allocation of requests to resources is determined in advance

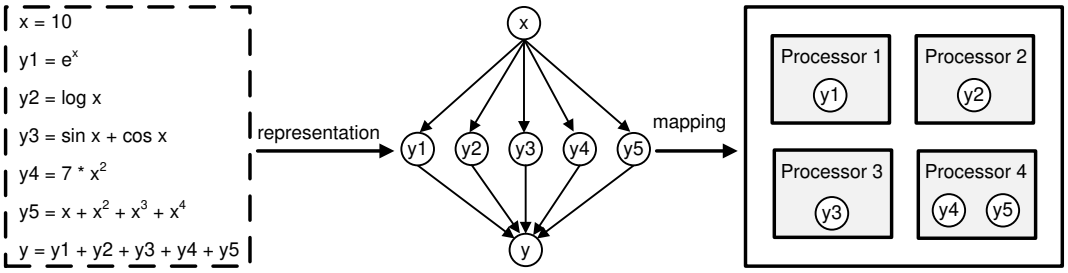

Figure 1.1: Example of parallel processing.

then it is called static scheduling. If the allocation is determined upon the request's arrival to the system, it is known as dynamic scheduling [3]. While scheduling, various parameters need to be considered, and these can be divided into two categories: request parameters and resource parameters. Request parameters include the characteristics of a request, such as CPU requirement, memory requirement, data dependency and execution time. On the other hand, resource parameters include CPU capacity, memory capacity and information about currently running requests. Having such information is crucial for proper assignment of requests to resources. Finding the best association between requests and resources is a very complex problem. Generally, the scheduling problem under such a scenario is known to be NP-complete [3].

This thesis deals with scheduling in a service oriented architecture (SOA). A SOA includes a collection of computational services that can be used to support requirements for complex and evolving applications. In general, services that compose an SOA may call one or more other services to perform part of the calling service's required computation. An SOA provides a platform for distributed computing where a service provider and service consumer interact with each other. Fig. 1.2 shows a block diagram where service consumers interact with service providers by sending service requests and in return they receive responses from those service providers. In an SOA, services can exist over heterogeneous platforms. For instance, services hosted on both Unix and Windows platforms can communicate by using SOA technology.

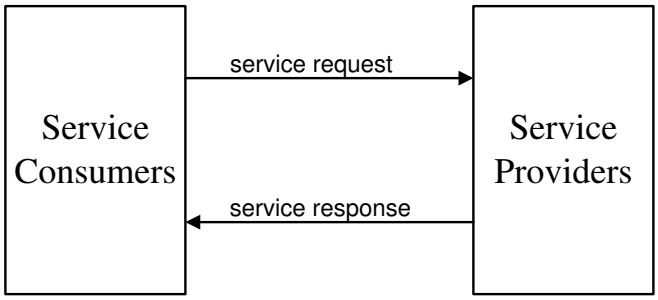

Figure 1.2: Interaction between service consumers and service providers.

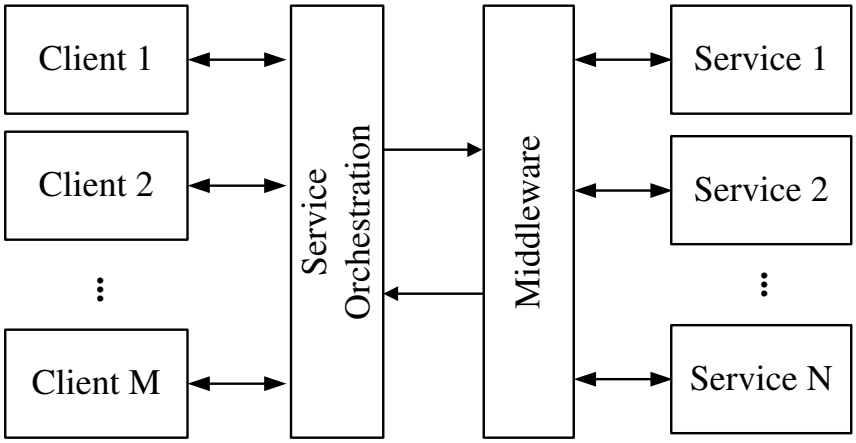

Figure 1.3: Orchestration in SOA.

Service orchestration is the process of integration, coordination, automation and management of services in a composite service environment. It plays a vital role in the automation of web services execution by combining multiple services. The orchestration executes services in a particular order, according to the business rules, data dependency and other constraints. Fig. 1.3 shows a block diagram of an orchestration setup where orchestration of services is done with the help of middleware technologies e.g., JMS, COBRA, MQ, AMQP [4, 5]. When clients send a service request, the service orchestration module designs an execution plan. The orchestration module then routes the service requests with the help of middleware brokers to the machines running requested services.

The functionality of services in an SOA can be strongly influenced by business

requirements and domain-specific terminology and definitions. As a result, new developments and revisions of applications regularly occur at the service abstraction level. The SOA approach provides a foundation upon which enterprise-level applications can be constructed. This thesis considers an SOA that is supported by an underlying cluster of memory managed multicore machines.

The SOA architecture described in the this section has some performance challenges associated with using middleware technology in passing messages between service consumers and service providers. The performance of the system can be enhanced by adding an orchestration layer on top of the middleware technology. The purpose of adding this layer on top of the existing framework is to provide quality of service (QoS) by implementing service level agreements (SLA). The focus of this thesis is on a component of orchestration dealing with the scheduling of service requests to system resources.

In the following section, further motivation behind this research as well as an overview of the proposed scheduling framework is discussed.

1.2 Motivation

There are various factors that need to be considered when improving the computational capabilities and infrastructure of an organization. When serving its clients, an organization should seek to find an optimal balance between satisfying clients' requirements and determining the appropriate quantity and complexity of resources required to do so. This section provides an organizational perspective to analyze these challenges. To enhance overall performance, an organization has to deal with questions that compete for limited available investment:

- How much should be invested in hardware?
- How much should be invested in optimizing system software?

- How much should be paid in penalties to clients due to violation of service agreements?

It is generally difficult to properly balance investment in the aforementioned questions. The following section analyzes these issues in terms of cost, value and risk of each approach.

1.2.1 Service Level Agreement Penalties

An SLA [6, 7, 8] is a formal negotiated agreement between a service provider and a service consumer that specifies the level of service. The SLA specifies performance metrics (like response times, deadlines), and payment penalties for failing to meet the terms of the agreement. The individual performance metrics are called service level objectives (SLOs) [7]. For example, an SLO can be expressed in terms of performance goals like average response time for a set of operation, or can also be expressed in terms of a deadline by which the operation should be completed.

SLA monitoring issues are described in [6, 7, 8]. In [6], a powerful theory regarding the SLA penalties is provided that is based in terms of utility, probability, and cost. The utility not only covers the monetary cost but also the goodwill and other parameters like renewal of service. Therefore the SLA cost can be non-linear relative to the monetary values that has to be paid through penalties.

SLA implementation is very important in an SOA to ensure quality of service (QoS). IBM's Web Service Level Agreement framework [7] describes the specification and monitoring of SLAs for Web Services. It enables providers and consumers to define a wide variety of SLAs and the way they are measured. Similarly, IBM's SAM [8] describes the business oriented service level management (SLM). It has an e-business SLA execution manager which implements an SLA.

An SLA penalty [6] is the cost associated with delayed processing of requests.

The delayed processing of request may be either due to inadequate resources or malfunctioning of resources. In both cases, there exists three courses of action that the service provider may pursue:

- Bear the cost penalty associated with SLA violation;
- Add hardware;
- Invest in system software optimizations.

Generally, an organization pursues all courses of action at some level. Sometimes it may be better to choose only the first action, whereas, most of the time, it is more natural to consider the other two actions. The scenario where the first option might look favorable is described as follows. Consider a case where the cost due to SLA violation is very low compared to the other options. In this scenario if the goal of an organization is to minimize the monetary cost, then bearing cost associated with SLA violation could be a viable option. However, if there are more frequent and high volumes of SLA violations then the goodwill cost associated with SLA violation is not linearly related to the monetary amount that is paid. If the company's goodwill goes down, then old clients may not continue to buy the services and new clients may not join, which would ultimately affect the health of the organization. The cost associated with the SLA violation is illustrated graphically in Fig. 1.4. The dash line represents the net cost associated with SLA violation, whereas, the solid line represents the actual value as a result of cumulative effect of cost due to SLA violation, renewal rate, and goodwill.

1.2.2 Investment in Hardware

In 1965, Gordon Moore estimated the pace of technology in what is now known as Moore's law. According to this law, the processing power of hardware doubles every

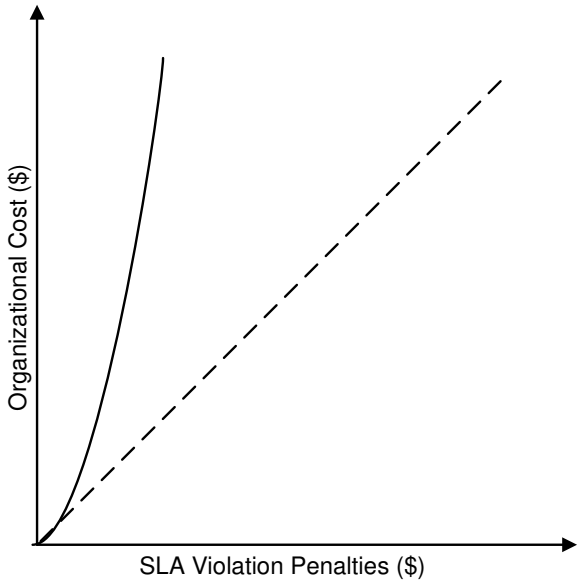

Figure 1.4: Organizational costs associated with SLA violation penalties. Dashed line represents short-term monetary costs; solid line represents long term cost.

18 to 24 months [1, 2]. Fig. 1.5 shows that the prediction by Moore has been fairly accurate.

To increase the performance of a system, adding hardware is often the first action considered by an organization. The reason for choosing hardware enhancement is that it can be faster and easier than attempting to optimize software. Having higher processor speeds results in faster processing of CPU-bound requests. This option seems to be very reasonable to improve throughput of the system dealing with CPU bound requests. In terms of cost, getting the latest hardware may not justify the performance gains. For example, the cost of a quad core machine with 16GB of RAM is much less than eight-times lower the cost of a 32 core machine with 128GB of RAM. There are some instances in which the scaling hardware yields no performance gain. System level issues, network delay and process scheduling may not be addressed by simply adding hardware.

Fig. 1.6 shows the cost associated with adding hardware. The dashed line shows the cost in an ideal scenario (following Moore's law) but the actual cost can increase

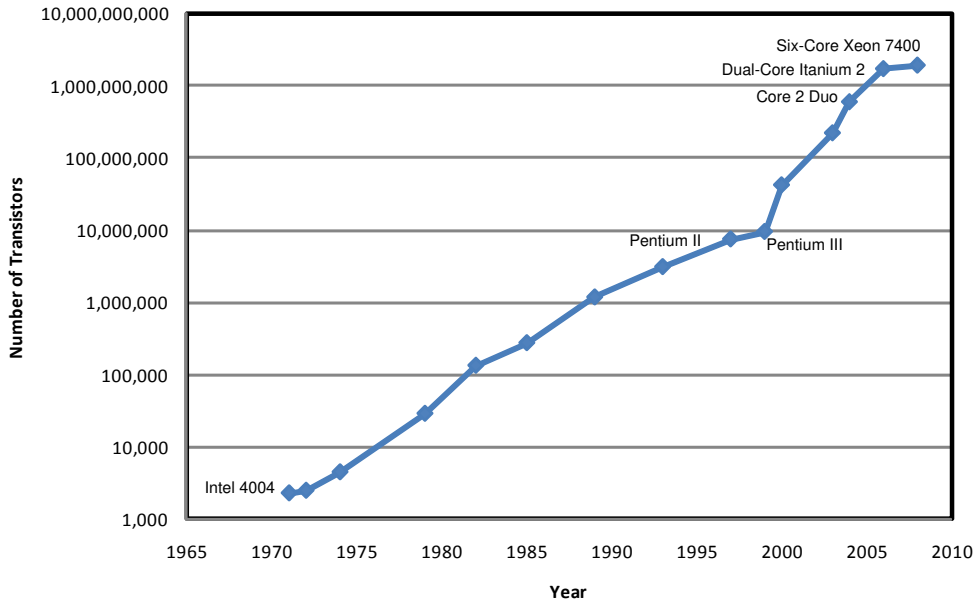

Figure 1.5: Moore's Law from 1970 - 2008 [9, 1, 2].

non-linearly due to the overhead of maintenance/support and diminishing return on performance due to scalability issues.

1.2.3 Investment in Software Optimizations

Software optimization can be challenging and time consuming task. Perhaps due to these reasons, the rules for when to optimize [10] are set forth as: "Don't do it"; and "Don't do it, yet". These statements are made in the context of performing code level optimization. Because changing code for small performance improvement is not worthy due to risk of missing or changing the functionality of the software. As stated previously, adding more hardware is not always a viable action because until and unless resources are being efficiently utilized, the resource inefficiencies remain unresolved. In order to enable scalable platform, software optimization is often necessary. Fig. 1.7 shows that investing in software optimization can have a high initial cost, in long run it can actually benefit an organization because of its expected performance gain and high return on investment (ROI).

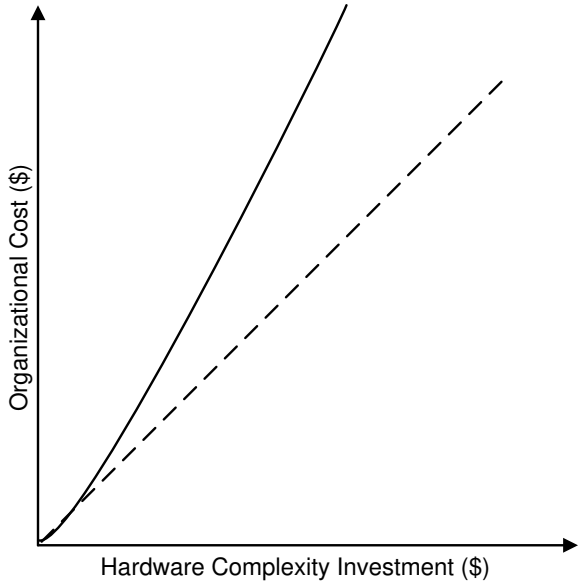

Figure 1.6: Organizational cost associated with Hardware complexity. Dashed line represents short-term monetary costs; solid line represents long term cost.

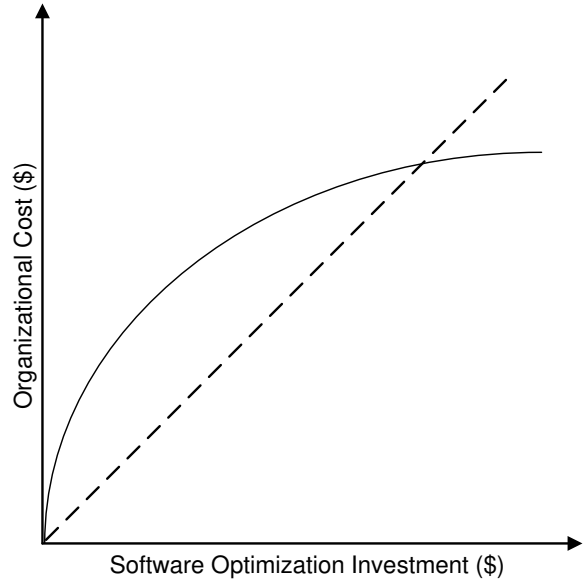

Figure 1.7: Organizational cost associated with investing in software optimization. Dashed line represents short-term monetary costs; solid line represents long term cost.

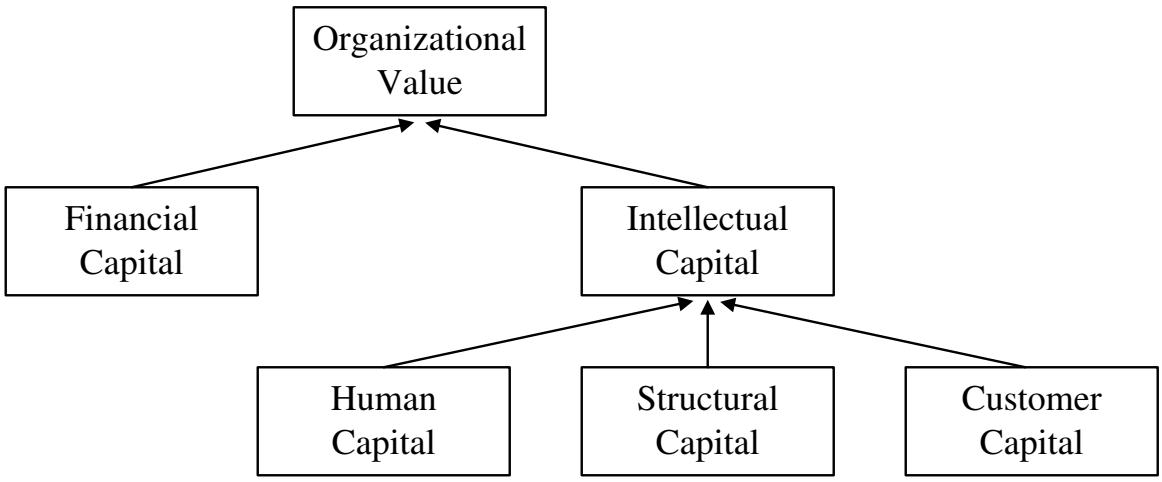

Figure 1.8: Organizational value in terms of financial capital and intellectual capital.

Investing in software optimizations results in investing in more people which not only has direct benefits to an organization but also the indirect benefits in the form of intangible assets of an organization. In broader view, organizational assets is the collective sum of human centered assets, intellectual capital assets, infrastructure assets, and market assets. Fig 1.8 illustrates the organizational value in terms of tangible and intangible assets [11, 12].

The work in this thesis is motivated by the benefit associated with an organization choosing to investing in developing system software optimizations as a viable action in minimizing organizational costs. The particular optimization approach studied is in the scheduling of computational tasks for a collection of computing resources.

1.3 Overview of the Thesis

An application supported by an SOA is modeled in this thesis as a workflow graph (WFG), which is a directed and acyclic graph that defines precedence constraints among service requests required by the application. WFGs can vary greatly in size and structure. For example, a small WFG may contain just a few requests (i.e., vertices) while a large WFG may contain thousands of requests. Regarding structure,

at one extreme a WFG may represent a single chain of requests in which no two requests may be executed in parallel. At another extreme, the structure of a WFG may contain numerous independent chains of requests in which requests belonging to distinct chains may be executed in parallel.

In the framework considered here, WFGs are assumed to be submitted by multiple sources (i.e., clients) to a scheduler. Associated with each submitted WFG is a deadline that defines the time by which all requests of the WFG should complete execution. The main objective of the scheduler is to assign requests of submitted WFGs to machines of the cluster so as to reduce missed deadlines of all WFGs.

The performance of different scheduling polices are evaluated using a simulation environment that provides the scheduler with state information related to the requests of the submitted WFGs and performance/loading information related to machines in the cluster. Four possible states of a request are: blocked; ready; executing; and completed. When all requests of a WFG reach the completed state, the entire WFG is defined as completed. The simulation environment models each machine in the cluster of memory-managed multicore machines. CPU and memory resources are the two primary factors used to characterize machines.

The scheduler is assumed to have estimates for the CPU and memory requirements of each request in a WFG. As request-to-machine assignments are made by the scheduler, the simulation environment updates and tracks the aggregate CPU and memory loading of each machine based on the CPU and memory requirements of all requests currently executing on the machine. The instantaneous relative performance (i.e., efficiency) of a machine is defined according to a function that maps aggregate CPU and memory loading to an efficiency value for that machine. Assigning a request to a machine increases the aggregate CPU and/or memory loading on the machine, which generally decreases the performance of the machine. Likewise, after a request completes execution, the performance (efficiency) of the machine on which

the request was executing generally increases because the aggregate loading on the machine decreases upon a request's completion.

In addition to specifying the machine on which each request is assigned for execution, the scheduler defines when each assigned request begins executing. To illustrate, consider a situation in which all machines of the cluster are currently loaded to near their capacities. In such a case, an intelligent scheduler may not choose to immediately assign newly ready requests to any machine because doing so would likely cause machine overloading thus, extending the completion time of requests currently executing due to a decrease in machine efficiencies.

Based on the assumed machine model, overloading a machine can cause dramatic performance degradation. This aspect of the machine model enables the evaluation of a scheduling policy's ability to determine the appropriate trade-off between starting a request on a machine that is relatively loaded versus holding a request until such a time that the loading of a machine decreases, due to other requests completing execution. Starting a request on a relatively loaded machine could delay the finish time of all requests assigned to that machine to unacceptable levels. On the other hand, delaying the start of requests' execution too long increases the time required for requests to finish, which can also produce undesirable results in terms of meeting overall WFG deadlines.

Three types of WFGs are characterized: Batch, Webservice, and Interactive. The arrival times of Batch WFGs generally have daily periodicity, which distinguishes them from the other WFG types. Furthermore, Batch WFGs generally have a larger number of requests compared to the other two WFG types. The Webservice WFGs generally have more requests than Interactive WFGs. In addition to differences in arrival processes and number of requests, the different WFG types have differences related to their structure and their deadline characteristics.

The approach to scheduling and executing WFGs proposed in this thesis is unified

in the sense that one scheduling mechanism is employed to assign requests from multiple WFG types to machines of a single cluster. A unified approach for scheduling and executing all types of WFGs has advantages over possible alternatives in which separate dedicated scheduling mechanisms and/or clusters of machines are used to support each type of WFG. Having multiple dedicated schedulers/clusters is more difficult to maintain (compared to a unified approach) because over time each scheduler/cluster would likely evolve unique features and support requirements, thus complicating the support and operation of the overall infrastructure. Furthermore, as new types of WFGs (i.e., new applications, products, and services) are developed, previously specialized schedulers/clusters may become obsolete or less efficient.

The remainder of the thesis is organized in the following manner. Chapter 2 includes an overview of related work. Chapter 3 describes the simulation environment developed to evaluate different scheduling policies. Chapter 4 describes specific scheduling policies considered in this thesis. Chapter 5 provides the results of simulation studies, followed by concluding remarks in Chapter 6.

Chapter 2

Related Work

Previous related work is reviewed in four broad areas: (1) machine modeling and simulation environments; (2) automatic memory management; (3) scheduling and load balancing; (4) workflow structures and patterns; (5) middleware technologies; and (6) SOA technologies.

2.1 Machine Modeling and Simulation Environments

Considerable work has been published related to modeling of machines in distributed environments. Much of the past research in this area has focused on modeling and predicting CPU performance, e.g., [13, 14]. The machine model described in the present thesis (refer to Chapter 3) relies on assumed knowledge of the characteristics of the requests (i.e., computational tasks); it is similar in a sense to the static approach proposed in [13]. The basic premise of the work in [13] is to predict the availability, i.e., performance, of a time-shared system under different loading scenarios. Two CPU availability prediction models are proposed: (1) Static Process Assignment Prediction (SPAP), which predicts availability based on prior knowledge of task characteristics

and the state of system; and (2) Dynamic Process Assignment Prediction (DYPAP), which predicts availability using a proposed monitoring tool. The DYPAP monitoring tool does not rely on knowledge of task characteristics, but instead uses real-time measurements in predicting CPU availability.

Methods for predicting running time of tasks executing in a shared grid environment are described in [14]. The approaches described in [14] rely on real-time dynamic measurements; they do not assume prior knowledge of computational workload characteristics. One basic approach proposed in [14] is to employ polynomial fitting techniques to a machine's recent past performance values as a means for predicting future performance. Another approach proposed is based on a technique that compares a machine's recent sequence of performance measures to stored past performance time series to discover similar patterns, which are then used to predict future performance.

In contrast to [14], the distributed system modeled in the present thesis is a dedicated system. Although the WFGs are submitted from a number of different users (i.e., clients), the execution of the WFGs underlying requests are scheduled by one scheduler. Furthermore, the WFGs themselves are implemented on top of a common SOA. In this type of environment, meaningful historical computational characteristics can be compiled by logging past execution performance of WFGs submitted by each client.

2.2 Automatic Memory Management

In memory managed systems, the effect of long and/or frequent garbage collections can lead to undesirable – and difficult to predict – degradations in system performance. Garbage collection tuning, and predicting the impact of garbage collections on system performance, are important and growing areas of research, e.g.,

[15, 16, 17, 18, 19]. To estimate the overhead associated with various garbage collectors, experiments were designed and conducted in [16, 17] to compare the performance associated with executing an application assuming automatic memory management versus explicit memory management. The machine model proposed in the present thesis accounts for the overhead associated with automatic memory management. In [15], it is empirically shown that the negative effects of garbage collection can be avoided if the available heap memory is at least seven times the total amount of required memory, defined as the maximum reachable heap space throughout the lifetime of a program execution. While this result is interesting and insightful, it is generally not practical to assume/require an operational system be endowed with nearly an order of magnitude more memory than is actually required.

Similar to the results reported in [15], it is shown in [16, 17] that automatic memory management performs comparable to, and sometimes better than, explicit memory management provided that the heap size is no less than five times the required memory. Furthermore, [16, 17] includes detailed evaluations and measurements that quantify the degree of performance degradation observed when the heap size is less than five times the memory required. It is shown that as extra heap space of a machine is reduced below a critical value, the machine's performance can degrade dramatically, i.e., execution times may increase by an order of magnitude or more if there is little or no extra heap space. The machine model used in the present thesis defines the performance of a memory managed machine based on a curve derived from empirical results reported in [16, 17]. This curve is used to map a measure of memory loading to a corresponding measure of machine performance.

2.3 Scheduling and Load Balancing

Formulations of realistic scheduling problems are typically found to be NP-complete, hence heuristic scheduling policies are generally employed to provide acceptable scheduling solutions, e.g., refer to [19, 20, 21, 22, 23, 24, 25]. The scheduling evaluations conducted in the present thesis account for the impact that garbage collection has on a machine's performance. Examples of other memory-aware scheduling approaches are described in [19, 26].

Load balancing involves techniques for allocating workload to available machine(s) in a distributed system as a means of improving overall system performance. Examples of both centralized and distributed load balancing approaches are described in [20]. The scheduling framework developed in the present thesis incorporates aspects of load balancing in the sense that the scheduler only assigns requests to machines that have loading factors (for CPU and/or memory) that are below defined thresholds. In typical load balancing schemes, the load assigned to a given machine is compared with the overall average load on the entire system. If this comparison reveals that the given machine is assigned more load than the system average, then excess load on the machine is assigned to more lightly loaded machines; otherwise the machine is declared to be available to take more load.

2.4 Workflow Structures and Patterns

Workflow technology is being widely used to model the business process [27, 28, 29, 30, 31, 32]. In recent days, workflows have evolved to model the business requirements thereby providing some opportunity for reengineering, optimization and automation [27, 28]. It models the business processes and requirements as workflow specifications [27] which can be defined in terms of some structure and pattern [29].

Workflow specifications can be analyzed in terms of control flow, data representation, and resource usage [28]. The control flow describes sequence of activities where activities refer to some volume of work or task. The control flow uses the data for processing the task defined in workflow specification. The resource is also a part of workflow specification which include the infrastructure required to process the business requirements.

Workflows functionality can be described in terms of task sequencing, parallelism, synchronization and iteration. These functionalities are some of the proven business process automation building blocks [28]. Based on the business requirements, the structure and pattern of workflows are defined. There exists multiple categories of workflow patterns, some of them are: workflow control flow patterns [29]; workflow data patterns [30]; and workflow resource patterns [31]. The workflow control patterns define the control flow that might be encountered during modeling and analyzing workflow. Twenty such patterns are defined in [29]. The workflow data patterns in provide various ways in which data is represented and utilized in workflows. In [30], thirty-nine data patterns are described. The workflow resource patterns define various ways in which resources are represented and utilized in workflows. In [31], forty-three workflow resource patterns are identified, which describe the manner in which work items are distributed and executed by resources in workflow systems.

Various classifications of scientific workflows for grid system are developed in [32]. In the paper, workflows are defined in terms directed acyclic graphs having multiple scientific tasks with some dependencies. The paper provides a taxonomy for grid workflow system which focuses on workflow design, scheduling, fault management and data movement.

Workflows represented in this thesis are defined as directed acyclic graphs similar to [32] where the directed edge in the graph represents the precedance constraint that exist between nodes of the graph. A node in the graph which represents a service

request in a Service Oriented Architecture (SOA) [33, 34], posses some amount of computational work to be done similar to [28]. Three types of workflows are defined, namely, Interactive, Batch and Webservice. The graph structures for each type of workflows are different. The degree of parallelism [28] for Batch is higher than Webservice workflows, whereas, Interactive generally have sequential processing [28]. this same basic workflow model/taxonomy is adopted in the present thesis.

2.5 Middleware technologies

Distributed applications can be spread across various networks and even across multiple geographic locations. Middleware technologies are being used to make them work together. The middleware serves as a software layer between those different applications, thereby providing a common platform for heterogeneous computing environments. Message Oriented Middleware (MOM) is being used to pass messages across different applications. Some of the MOM technologies are: Java Message Service (JMS) and Advanced Message Queuing Protocol (AMQP) [4, 5, 35].

JMS defines the standard for messaging in enterprise applications. It provides an interface for communication among enterprise applications through its application programming interface (API). JMS supports both synchronous and asynchronous communication between sender and receiver applications [4]. It defines two communication patterns, Point to Point and Publish/Subscribe. A sender application using point to point messaging uses a queue as a destination where multiple senders can send messages. Receiver applications can extract messages from the queue. A single message can only be consumed by a single receiver application. An application using publish/subscribe communication uses a topic as a destination. Publishers can publish messages to a topic that get delivered to all the subscribers of that topic.

There are various JMS implementations by different vendors for example: OpenJMS, ActiveMQ, WebsphereMQ, EBA Weblogic, JBossMQ, FioranoMQ, Tibco, SonicMQ and SwiftMQ [5, 36]. An analytical model for describing the message throughput is presented in [36], where message throughput for FioranoMQ, SunMQ and WebsphereMQ are evaluated. The JMS performance modeling and benchmarking is done in [37]. It provides detailed performance analysis of EBA Weblogic servers. The performance improvements of JMS based applications are presented in [38].

AMQP is one of the currently evolving MOM technologies that defines standards for message publishing, queuing, and routing [35]. The main purpose of AMQP is to provide interoperability between multiple middleware vendors and implementations. It defines two protocols: a low level wire protocol and an application level protocol which enables enterprise application to pass messages across different platforms [35]. The middleware broker has exchanges and message queues. The sender application sends messages to the exchange which then routes message to the message queues and finally the consumer application consumes message from those queues [35].

JMS and AMQP are being widely used in a number of areas, for example: financial applications, health care applications and manufacturing application [5, 39]. AMQP addresses the interoperability issue of different JMS implementations. The operation of these middleware guarentees the first in first out (FIFO) message passing [4, 40].

2.6 SOA Technologies

SOAs [33, 34] are widely being used by enterprise business applications. SOA use services as building blocks for applications where services interact with each other by sending and receiving messages. Extended Service Oriented Architecture (ESOA) [33] was introduced to provide advanced functionality for e-business applications which provides service composition and coordination for grid services. In SOA, service

providers publish services on the internet that is can be accessed by service consumer anytime and anywhere in the world.

A number of standards and products are available that facilitate service development and interoperability of an SOA, including WSDL, UDDI and SOAP [33]. WSDL is a standard XML format for describing web services that contain a set of definitions as follows: types (defines data types); messages (defines messages); portType (defines operations supported); and binding (defines communication protocals).

UDDI uses WSDL for defining interfaces to web services. The UDDI specification provides a mechanism for providing and accessing services over the Internet.For example a new web service can be added to UDDI registry, which can be accessed programmatically. XML is used to define specification of all APIs in UDDI which is then wrapped in SOAP envelope and tranported over HTTP.

Middleware technologies are also being used by the SOA approach for the interoperability of services running in various platforms [41]. Services can use middleware message passing interfaces, for example JMS and AMQP, to send and receive messages. These Message Oriented Middleware (MOM) technologies provide the standards for storing, queuing and routing various types of messages.

SOA technologies are critical for implementing actual production systems. The focus of the present thesis, however, is on determining good scheduling approaches (which would eventually be implemented in a production system). The next chapter describes the simulator environment that has been developed to aid in the evaluation of scheduling policies.

Chapter 3

Simulation Environment

3.1 Overview

The block diagram of Fig. 3.1 illustrates the three major components of the simulation environment, which is referred to as scheduling framework. The framework presented in the figure models a real world environment where clients send their requests in the form of workflow graphs (WFGs) to the system. The system then executes those WFGs and sends the results back to clients. The WFG Generator in the figure models the client's requests in the form of three categories of WFGs, i.e., Interactive, Batch and Webservice. WFG Generator generates these three categories of WFGs and sends them to the Scheduler. The Scheduler maintains those WFGs in the Scheduling Pool. The Scheduler module selects from two types of scheduling policies: Request Selection Policy (RSP) and Machine Selection Policy (MSP). The Assigner module of the Scheduler uses a RSP to select the next request from the Scheduling Pool to be considered for execution. Examples of RSPs are First Come First Serve (FCFS), Earliest Deadline First (EDF) and Shortest Workflow First (SWF). Similarly, the Assigner selects machines from the Cluster of Machines for the purpose of assignment of requests associated with WFGs using an MSP. Examples of MSPs are Best Pre

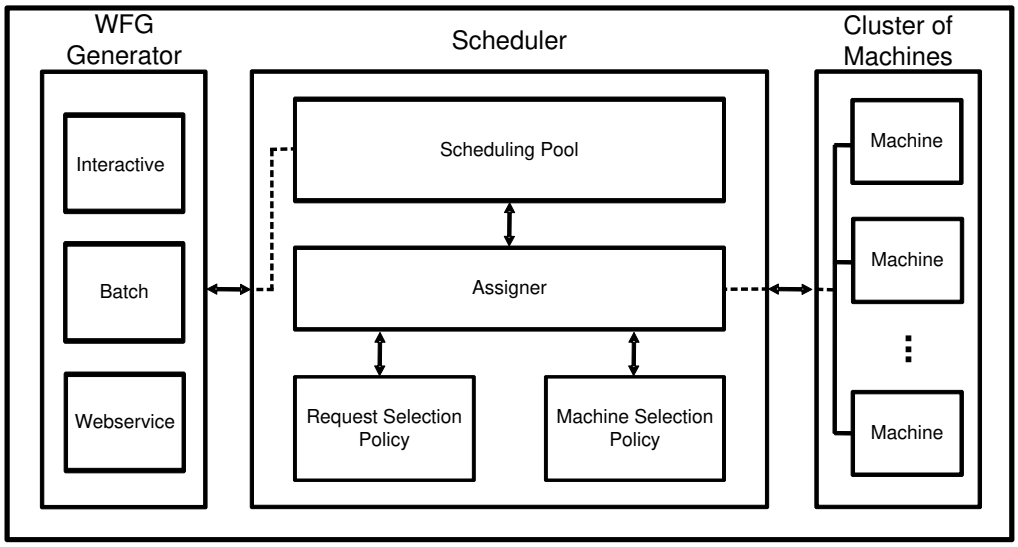

Figure 3.1: Major components of the scheduling framework.

Mapping (BPRM), Best Post Mapping (BPOM) and Least Work Remaining (LWR). Each of these main components of the scheduling framework are described in detail in Sections 3.2 through 3.4.

3.2 WFG Generator

A WFG is a directed acyclic graph that is composed of parallel and sequential combinations of request chains (RCs). An example WFG is shown in Fig. 3.2. The vertices of the graph represent requests and the directed arcs denote precedence constraints that exist between requests. This WFG in the figure contains five RCs. A dashed encircling of one RC is indicated in Fig. 3.2.

A WFG is a hierarchical structure that can be defined in a recursive manner by introducing the concept of a compound node. A compound node represents parallel instances of a common RC. The parallel RCs associated with a compound node represent instances of the same chain of requests that are to be executed with different input data sets. The right side of Fig. 3.3 represents the WFG using three compound

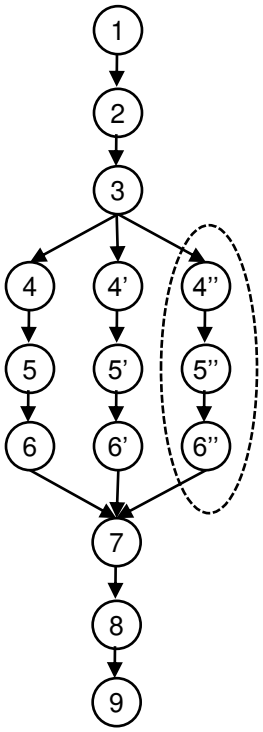

Figure 3.2: Sample WFG with one of five RCs encircled.

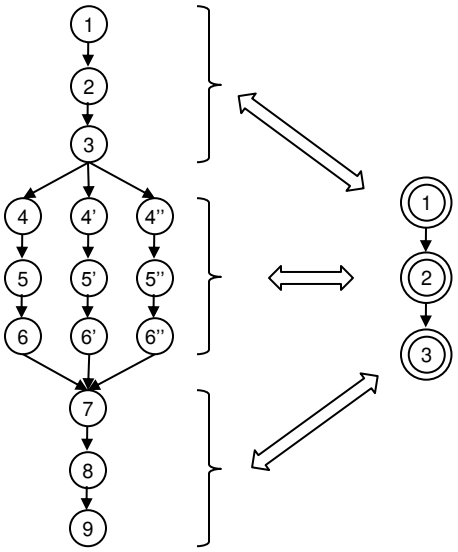

Figure 3.3: Sample WFG (left) and its representation as a sequence of compound nodes (right).

Table 3.1: Definitions of CPU and heap memory requirements for request r.

$C_r > 0$	C_r is the total number of CPU cycles required to execute r on the fastest unloaded machine.
$D_r \geq C_r$	D_r is the ideal execution time duration of r on the fastest unloaded machine.
$U_r = C_r/D_r$	U_r is the CPU utilization factor of r.
$M_r > 0$	H_r is the maximum reachable heap memory requirement of r.
$A_r \geq H_r$	A_r is the total heap space allocated by r.
$G_r = A_r/H_r$	G_r is the garbage generation factor of r.

nodes. The parallel RCs of a compound node could themselves be sequences of compound nodes, thus enabling the representation of WFGs of greater depth than the example shown in the figure.

The primary function of the component labeled WFG Generator in Fig. 3.1 is to provide synthetically generated WFGs to the Scheduler for the purpose of evaluating scheduling policies. The WFG generation process used in this thesis is probabilistic. Parameters for WFG generation rates, WFG structure, and CPU and memory requirements of requests that compose a WFG are defined for each WFG type generated. Table 3.1 summarizes the notation and definitions of basic computational and memory requirements for request r.

The first two parameters, C_r and D_r, represent the CPU cycle requirement and duration requirement of r. The CPU utilization factor of r, $U_r = C_r/D_r$, can be no greater than unity and no less than zero. A request having a CPU utilization factor of unity is typically referred to as a CPU-bound request, e.g., refer to [13].

The parameter H_r and A_r are related to heap memory usage of r. The garbage generation factor of r, $G_r = A_r/H_r$, is a relative measure of how much garbage is generated by request r. The smallest possible value of G_r is unity, which corresponds to the extreme case in which a request does not generate garbage. Large values of G_r correspond to requests that generate garbage at a relatively high rate. This parameter

is defined as the "total allocation to maximum reachable ratio" in [16, 17].

The ideal time required to execute a sequence of requests of an RC is the sum of the individual requests' ideal durations (D_r's) in the RC defined by Eq. 1. To estimate the time required to execute a compound node containing two or more parallel RCs, an assumption must be made regarding the degree of parallelism that will be exploited. The degree of parallelism, also known as parallelization factor pf, for a compound node indicates how fast the compound node need to be completed. If there is no possibility of parallelization then sequential duration is considered. Since a compound node can have multiple RCs, the expected duration of a compound node D_{CN} is calculated using Eq. 2. The term "available" in Eq. 2 refers to the number of parallel RCs in a compound node, whereas, the term "extra" refers to the remainder RCs that are not included in the parallelization. Finally, all the compound node duration are summed to get the expected durations of a WFG D_w.

$$D_{RC} = \sum_{r \in RC} D_r. \tag{1}$$

$$D_{CN} = \frac{(available - extra)D_{RC}}{pf} + extra > 0?D_{RC} : 0. \tag{2}$$

where $extra = available\%pf$.

$$D_w = \sum_{CN \in w} D_{CN}. \tag{3}$$

Each WFG has a deadline that defines the time by which all computations of a WFG should be complete. An estimate of a WFG deadline is calculated by multiplying the expected duration D_w by a deadline factor $df \geq 1$. The lower value of deadline factor indicates the tight deadline and higher being flexible deadline. The deadline of a WFG, denoted as d_w, is defined in Eq. 4.

$$d_w = df D_w. \tag{4}$$

The WFG Generator models the client's requests in the form of Interactive, Batch and Webservice WFGs. It then sends them to the Scheduling Pool, which is maintained by the Scheduler. An Interactive WFG is a type of WFG that has a relatively small number of requests, with each one having a short duration. Because clients demand an immediate response to their interactive requests, the deadlines associated with interactive WFGs are very tight. A Webservice WFG has relatively more requests with longer durations and a looser deadlines. The arrival times of Batch WFGs generally have daily periodicity, which distinguishes them from the other WFG types. Furthermore, Batch WFGs generally have the largest number of requests compared to the other two WFG types. The requests associated with Batch WFGs have longer durations and loosest deadlines compared to the other two types of WFGs.

In addition to differences in arrival processes, number of requests and deadline characteristics, the different WFG types have differences in their structure as well. Interactive WFGs have less opportunity for parallelization compared to Webservice and Batch WFGs. Batch WFGs usually have rich WFG structure with higher chances of parallelization than the other two WFGs.

3.3 Scheduler

The Scheduler component of Fig. 3.1 takes WFGs as input and assigns requests of the WFGs to machines of the cluster for execution. The Scheduler maintains a scheduling pool to hold all the incoming WFGs. The Assigner module of the Scheduler is responsible for selecting requests to be considered for execution. It not only selects the next request to be considered for execution, but also determines the best machine for execution. Each of these components are described in detail in the following

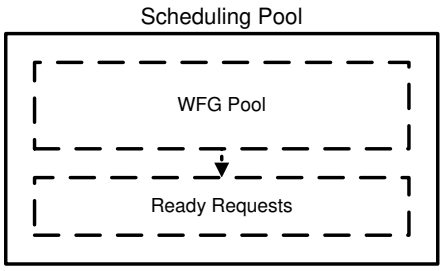

Figure 3.4: Components of Scheduling Pool.

subsections.

3.3.1 Scheduling Pool

The structure of Scheduling Pool has two components: WFG Pool and Ready Requests as shown in Fig. 3.4. The WFG Pool holds all WFGs that have not yet completed execution. Whereas the Ready Requests contains all the requests of the WFGs that can be considered for execution.

3.3.2 Assigner

The Assigner component of the Scheduler in Fig. 3.1 selects ready requests of the WFGs and assigns them to machines of the cluster for execution. In making assignment decisions, the Assigner can make use of computational and memory requirements assumed to be known and available for each request. Having access to such information is realistic in the assumed dedicated environment in which off-line profiling and/or historical logging can be performed to collect and estimate such data. Associated with each WFG is a single timing deadline, and the Assigner can also utilize this information when making request scheduling decisions.

The compound nodes of each WFG are considered in order and are expanded by the Assigner to expose one or more parallel RCs. At any point during the execution of a WFG, the requests associated with one or more RCs are considered by the

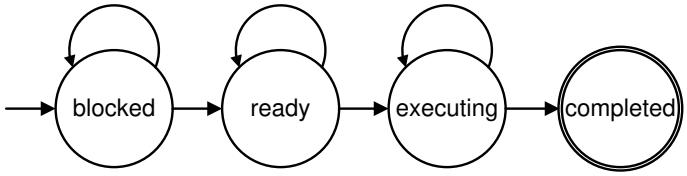

Figure 3.5: State diagram of a request.

Assigner for assignment to machines. The Assigner tracks the status of each request (associated with RCs currently under consideration) according to one of the following state values: blocked, ready, executing, or completed. Whenever a WFG arrives into the scheduling pool, the first compound node gets exposed. The first request of each RC in that compound node become available for assignment, changing to a ready state. The other requests following the ready request in RC are in blocked state. Once the ready request gets assigned to a machine, its state changes to executing. When the assigned machine finishes the execution of request, it's state changes to completed. Fig. 3.5 shows the state transition diagram of a request.

Fig. 3.6 shows an example of a snapshot of the Scheduling Pool at time instant t_i. In the figure, five WFGs are shown which are at various stages of their processing. WFG 1 has two compound nodes whose first compound node contains only one request, whereas, its second compound node contains three RCs each having six requests. In WFG 1, there are three RCs under consideration by the Assigner where one RC has an executing request and the other two have ready requests. Those two ready requests of WFG 1 can be considered by the Assigner for the purpose of mapping on to the machines of the cluster. Similarly, WFG 3 has only one compound node with one RC. It also has one ready request. In case of WFG 5, it has three compound nodes with the second one having three parallel RCs.

Consider that there are three available machines in the cluster with each one capable of running only one request at a time. There are five ready requests in Fig. 3.6. Because there are three machines with each one capable of running only one

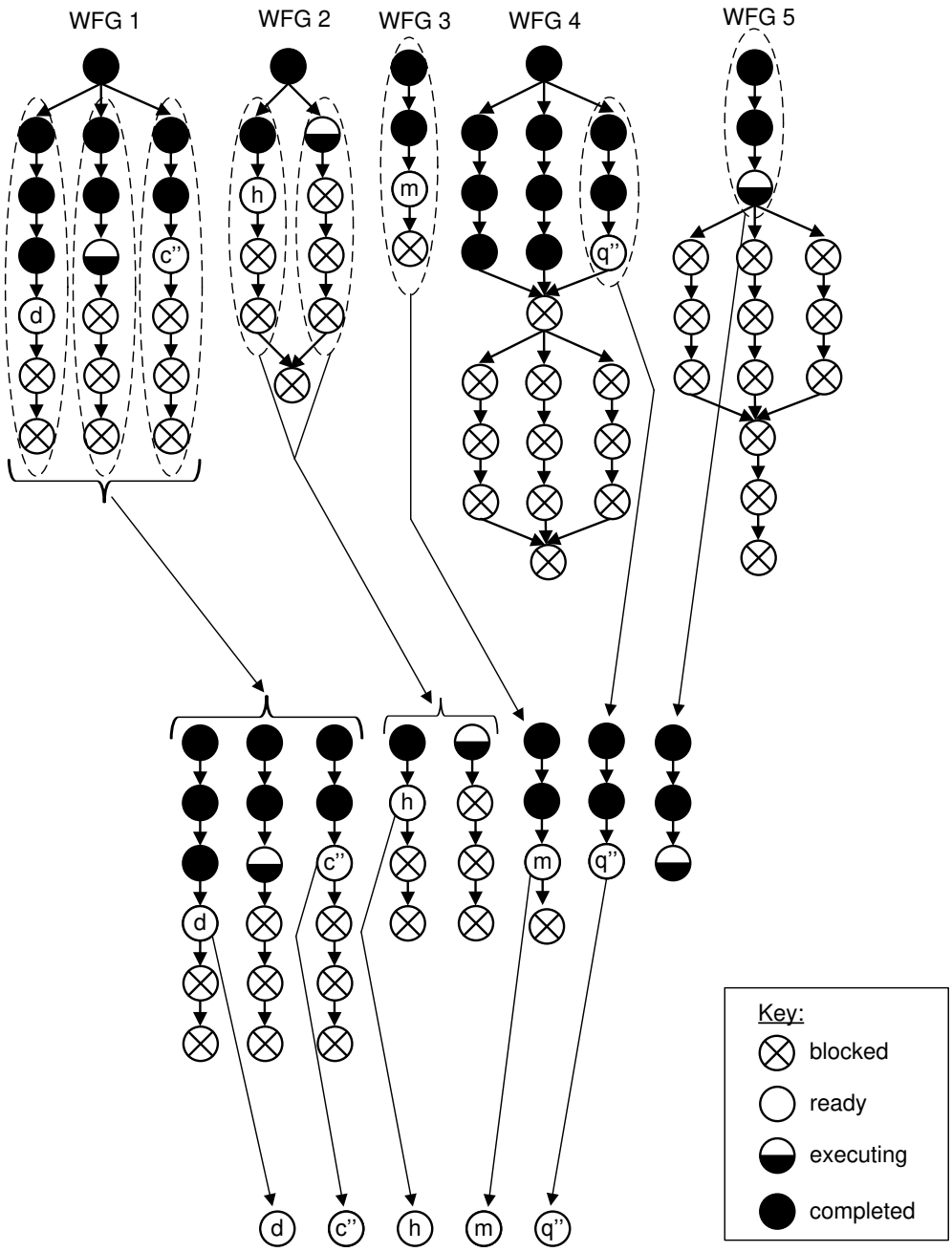

Figure 3.6: Snapshot of Scheduling Pool at the beginning of scheduling instant t_i where the upper part represents WFGs residing in the WFG Pool, the middle part represents RCs under consideration, and the bottom part represents ready requests.

request, only three requests can be assigned to those machines. Fig. 3.7 shows the snapshot after the assignment where ready requests from WFG 1, WFG 3, and WFG 4 get assigned. Note that in the figure, the status of those requests change from ready to executing.

Whenever a request finishes execution, the state of that request changes from executing to completed. Upon completing execution, the request that is the immediate successor of the completed request changes state from blocked to ready. This is illustrated in Fig. 3.7 and Fig. 3.8. Note that in Fig. 3.7, there are six executing requests. In scheduling instant t_{i+1}, three requests get completed and the corresponding changes are shown in Fig. 3.8. The Assigner also detects when all RCs associated with a common compound node complete execution, which triggers the Assigner to expand the successor compound node in the WFG as shown in Fig. 3.8. When WFG 5 completes the final request of its first compound node, the Assigner expands the second compound node.

The time instant that the state of a request r transitions from blocked to ready is defined as the request's birth time and is denoted by b_r. The Assigner uses a Request Selection Policy (RSP) to select the next request to be assigned to a Machine whereas a Machine Selection Policy (MSP) is used to select the best available machine for the assignment. When a request is assigned to a machine, the state of the request transitions from ready to executing. The time instant that the state of a request transitions from ready to executing is denoted as the request's start time and is denoted by s_r. The function of the Assigner is to define the machine assignment and start time (s_r) for each request r. The machine assignment of request r is denoted by M_r, and its value is equal to the identification number of one of the machines in the cluster.

Fig. 3.9 describes the pseudocode for the Assigner. For scheduling instant t_i, the Assigner uses an RSP to order the collection of ready requests, which is denoted

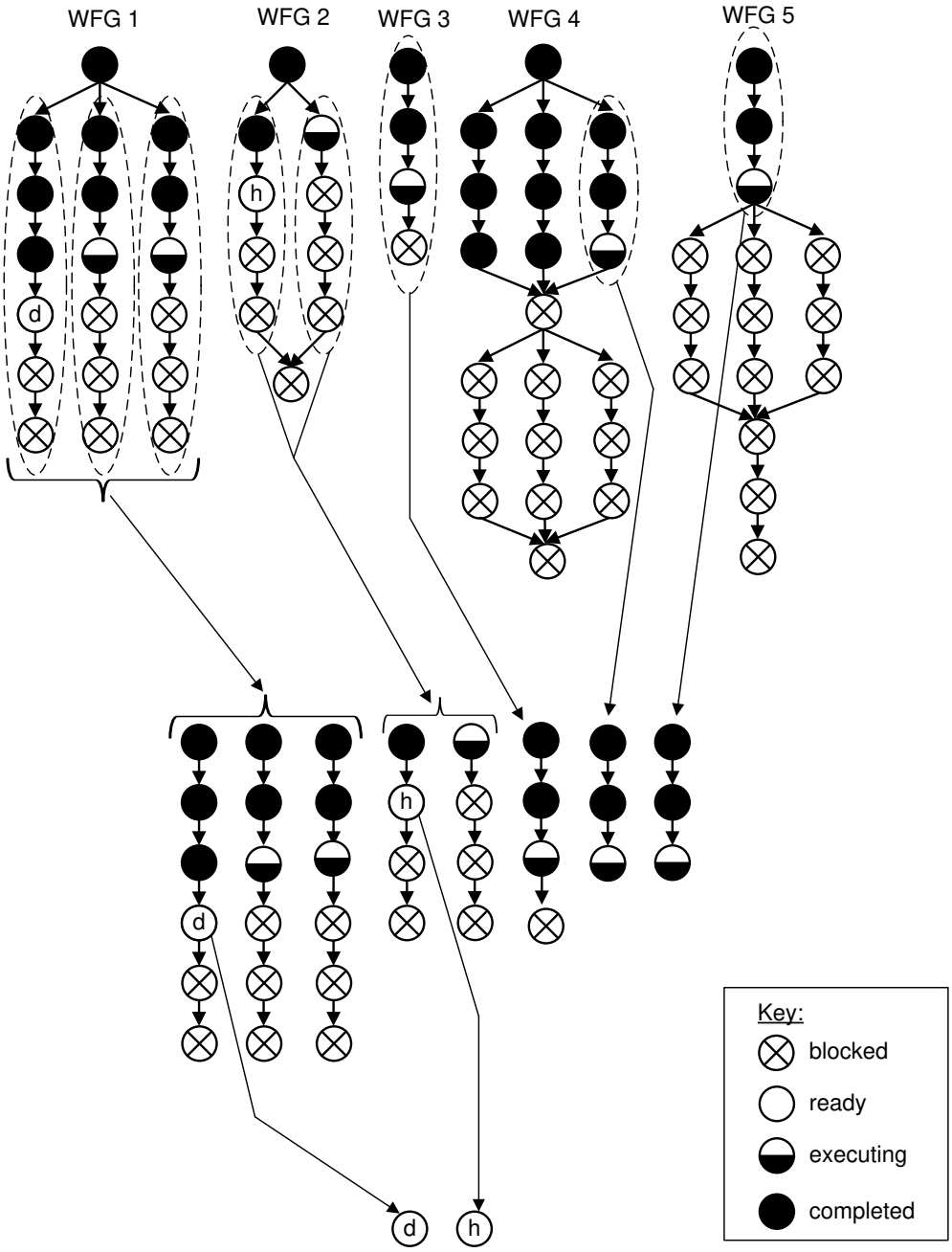

Figure 3.7: Snapshot of Scheduling Pool after assignment is made at scheduling instant t_i.

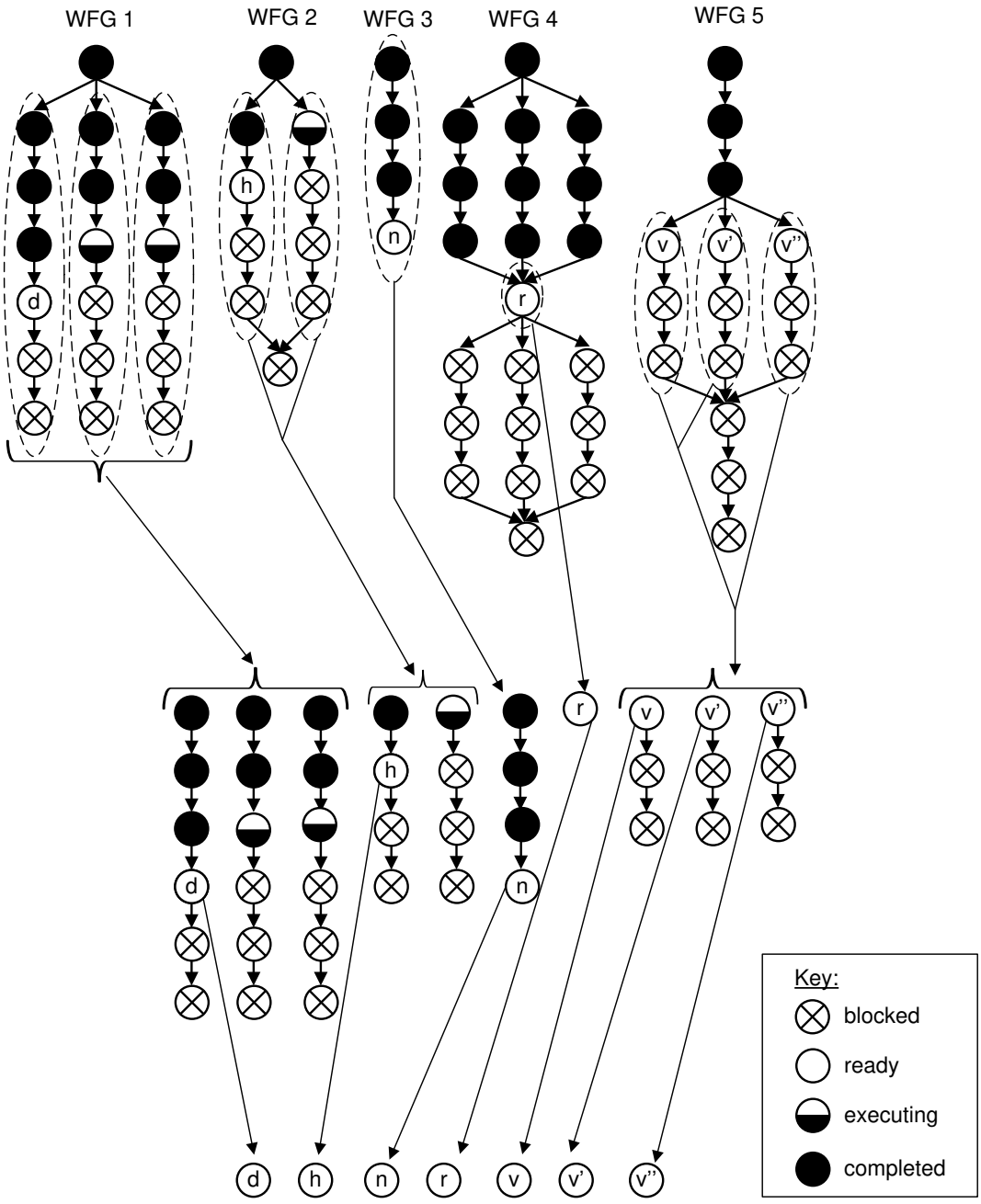

Figure 3.8: Snapshot of Scheduling Pool at scheduling instant t_{i+1}.

as $R(t_i)$. In this thesis, seven different RSPs have been implemented. RSP returns ordered ready requests, which is assigned to r_{list} (line 2). For each request r in r_{list}, the Assigner uses an MSP to find the best available machine to run each request on the r_{list}. The Assigner has the knowledge of machines in the cluster. Symbol $\mathcal{M}$ is used to represent the collection of machines in the cluster. A ready request can only be assigned to a machine that is declared to be available. The availability of a machine is determined using a defined threshold value associated with the machine's current efficiency. Symbol $\hat{e}$ is used to represent the machine threshold efficiency. Specifically, a machine is declared to be available only if it's CPU and memory efficiency is above the defined threshold value. Depending on the resource requirements and threshold values, the MSP may or may not return any available machines. If it does not return any machines then the Assigner excludes that request for this scheduling instant. It further considers the requests from r_{list} that may be appropriate for execution.

Assigner($R(t_i), RSP, MSP$**)**
1 **for scheduling instant** t_i
2 $r_{list} \leftarrow RSP(R(t_i))$
3 **for each** $r \in r_{list}$ **do**
4 $M \leftarrow MSP(r, \mathcal{M}, \hat{e})$
5 **if** $M \neq NULL$ **then**
6 $assign(r, M)$
7 **end if**
8 **end for**

Figure 3.9: Pseudocode for Assigner.

The complexity of the Assigner can be defined in terms of the complexities of a RSP and a MSP being used. The complexities of RSPs and MSPs are presented in Chapter 4. Depending upon the type of RSP and MSP, the complexity varies. Thus the complexity of the Assigner can be expressed as $Complexity(RSP(R(t_i)) + |R(t_i)|Complexity(MSP(R(t_i))$. An in depth analysis of these RSPs and MSPs is performed in Chapter 4.

3.4 Cluster of Machines

The Cluster of Machines in Fig. 3.1 is modeled as a group of machines each of which follows an efficiency-based machine model. The following section describes the machine model in detail.

3.4.1 Machine Model

The machine model takes as input the request-to-machine assignments and associated start times provided by the Assigner. The machine model tracks and updates an efficiency-based model for each machine in the cluster. The efficiency value for a machine depends on the aggregate CPU and memory loading due to all requests executing on the machine. At each scheduling instant, the machine model provides the Assigner with updated efficiency values for all machines and also notifies the Scheduler of any requests that have completed execution.

3.4.1.1 Machine Efficiency

The CPU and memory loading of a given machine changes with time as new requests are assigned to begin executing on the machine and existing requests complete execution on the machine. As a result, the instantaneous efficiency of a machine varies with time. Generally, the efficiency value of a machine decreases when new requests begin executing on the machine, and it increases when request(s) complete execution on that machine.

The efficiency of machine M at time instant t_i, denoted by $e(M, t_i)$, has a value between zero and unity. The number of CPU cycles remaining to complete execution of request r at time instant t_i is denoted by $c_r(t_i)$. The value of $c_r(t_{i+1})$ is calculated based on $c_r(t_i)$ according to the following equation:

$$c_r(t_{i+1}) = \begin{cases} C_r, & t_{i+1} < s_r \\ \max\left\{0, c_r(t_1) - (t_{i+1} - t_i)e(M_r, t)U_r\right\}, & t_{i+1} \geq s_r \end{cases} \quad (5)$$

For time instants less than r's start time, the value of $c_r(t)$ remains constant at C_r (refer to Table 3.1 for definition of C_r) because the request has not yet started executing. For time instants greater than the request's start time, the value of $c_r(t)$ decreases according to the difference equation defined by the second case of Eq. 5. The value deducted from the CPU cycles remaining to complete execution of request r is proportional to the product of the efficiency of the machine on which the request is assigned and that request's CPU utilization factor. Thus, the maximum possible deduction is $t_{i+1} - t_i$, which corresponds to a situation in which the request is executing on a machine with an efficiency of unity and the request has a CPU utilization factor of unity, meaning the request is totally CPU-bound. The application of the max function in the equation ensures that the number of CPU cycles remaining to complete execution of request r is non-negative.

Fig. 3.10 illustrates how changes in a machine's efficiency value affects the time required to execute a request on that machine. From the figure, notice that request r starts executing on the assigned machine at $t = s_r$. Near the beginning of the request's execution, note that the efficiency of the machine is relatively high, and the slope of the curve for $c_r(t)$ is corresponding steep. This indicates that CPU cycles are being deducted from the required amount at a relatively high rate, refer to the term involving $e(M_r, t_i)$ in Eq. 5. Throughout the execution of request r, other requests start executing on the machine (corresponding to decreases in the machine's efficiency value) and complete execution on the machine (corresponding to increases in the machine's efficiency value). The completion time of r is defined at the point in time when $c_r(t) = 0$.

The following discussion describes how the value of a machine's efficiency ($e(M_r, t)$

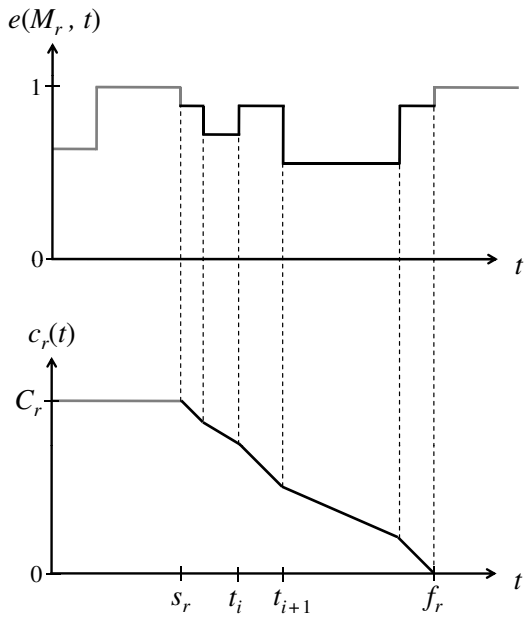

Figure 3.10: Illustration of how a machine's efficiency value affects the time required to execute a request on the machine.

in Eq. 5) is modeled in the simulation environment. Throughout this discussion, it is understood that the efficiency value is related to a particular machine for a particular time instant. Thus, the value of efficiency is often referred to as simply e, instead of $e(m, t)$, to ease notational burden.

CPU and memory resources are the two primary factors used to characterize machines. In the machine model, the overall efficiency is defined by the product of two terms:

$$e = e_c e_h. \tag{6}$$

The terms on the right hand side of Eq. 6 are defined as the CPU efficiency and memory efficiency, respectively. The values of e_c and e_h represent the relative impact on a machine's overall efficiency due to loading of the machine's CPU and heap resources, respectively.

The CPU resource of a machine is defined according to two parameters: (1) speed of the machine's cores and (2) number of cores in the machine. The maximum possible

value of e_c for a particular machine is normalized relative to the speed of the machine in the cluster having the fastest cores. For example, in a cluster where a machine with the fastest cores are twice the speed of those belonging to the machine with the slowest cores, then the former machine will have a maximum CPU efficiency of unity, while the latter machine will have a maximum CPU efficiency of one-half. Thus, in a cluster of identical machines the maximum value for e_c is unity for all machines. Without loss of generality, the maximum value of e_c is assumed to be unity for the rest of this section.

The loading of a machine's CPU resources is defined by the summation of the CPU utilization factors of all requests executing on the machine. Thus, if a machine is currently executing three requests having utilization factors of 1.0, 0.6, and 0.7, the total CPU loading on the machine would be defined as 2.3. The particular mathematical or empirical model used to define the value of e_c as a function of CPU loading is not the primary focus of the present thesis. In general, such models represent e_c as a non-increasing function of total CPU loading. The precise features and parameters of any such model depends on many factors including assumed operating system, virtual machine implementation, CPU architecture, among others.

An idealized function for e_c has a value of unity for all CPU loadings less than the number of cores present in the machine. For CPU loading values greater than the number of cores, the idealized function for e_c decreases according to the ratio of the number of cores to the total CPU loading. In reality, overheads associated with context switching, caching effects, and other complexities that are difficult to model, would prevent the idealized efficiency curve from being realized in practice. Fig. 3.11 illustrates idealized and typical curves for e_c assuming a quad-core machine. The specific function for e_c assumed in the present thesis is given by Eq. 7.

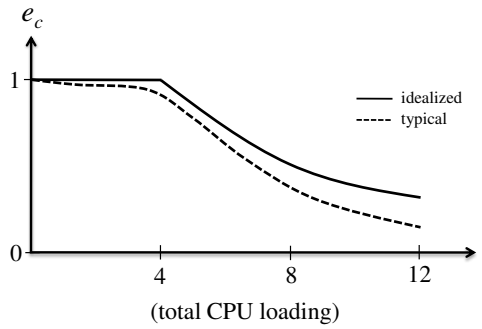

Figure 3.11: Ideal and typical curves for e_c for quad-core machine.

$$e_c = \begin{cases} 1, & \ell_c < 4 \\ (4/\ell_c), & \ell_c \geq 4 \end{cases} \tag{7}$$

The CPU efficiency function of Eq. 7 models a quad-core machine with a CPU loading factor of $\ell_c \geq 0$. The value of ℓ_c is assumed to equal the sum of the U_r's (CPU utilization factors) of all requests executing on the machine.

The memory resource of a machine is defined by two parameters: (1) total heap memory capacity and (2) average rate at which the machine's automatic memory management system can reclaim un-referenced heap space (i.e., garbage). For simplicity of discussion, the second parameter is assumed to be the same for all machines in the cluster. This assumption approximates a cluster configuration in which all machines implement the same virtual machine and have identical garbage collection configuration settings.

In the seminal work of Hertz [16, 17], extensive empirical studies were conducted to measure how the execution time of an application is affected by the relative size of the heap memory. The general conclusion drawn from this work is that execution time is relatively constant provided that available heap space is sufficiently large. As the relative heap space is reduced, then execution time begins to increase. When the heap space is critically small, the execution time of an application can increase significantly.

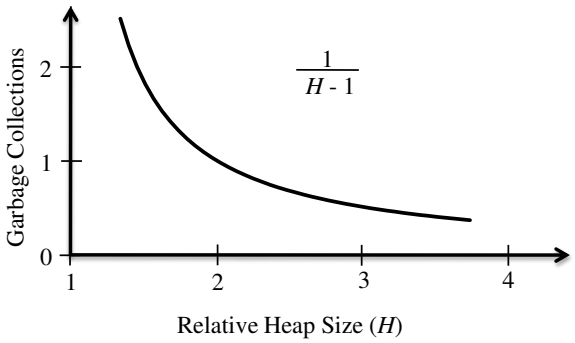

Figure 3.12: Number of garbage collection as a function of relative heap size for a copying garbage collector, derived from [15].

In [15], a mathematical analysis is derived for the classic copying garbage collector. The basic result of the analysis is that the overhead for this garbage collector grows according to $\frac{1}{H-1}$, where H is the size of the heap normalized by the maximum reachable heap memory requirement (defined as H_r in the present thesis). This expression relates to the number of garbage collections required, which clearly increases rapidly as H approaches unity. The shape of this curve associated with this function is fundamentally the same as the ones determined through extensive empirical studies in [16, 17] refer to Fig. 3.12.

To estimate how garbage collections impact the overall execution time, T, of an application, the following expression is proposed:

$$T = K + \frac{1}{H-1}. \tag{8}$$

The value of the parameter K represents the execution time when no garbage collections are performed, i.e., when the heap H is sufficiently large. The value of K is normalized in terms of the time required to perform a garbage collection. Thus, $K = 1$ models the situation in which the time taken to perform a garbage collection is the same as the time required to execute the application. A large value of K corresponds to the case where the time required to perform a garbage collection is

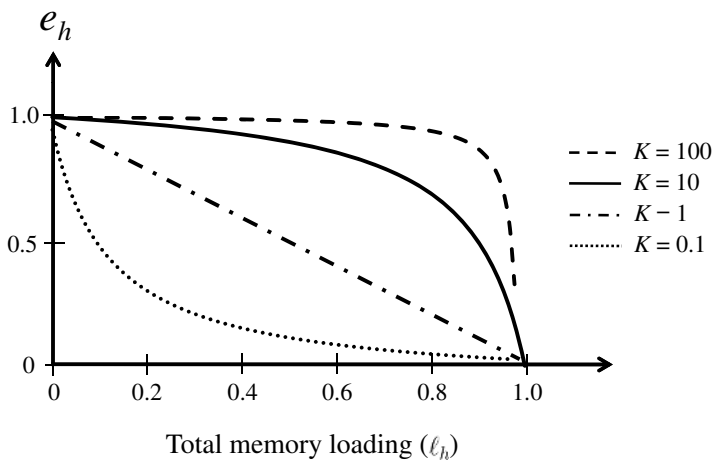

Figure 3.13: Typical curves for e_h associate with Eq. 9.

relatively small compared to the time required to execute the application.

By dividing the ideal execution time of the application, i.e., K, by the overall execution time represented by Eq. 8, an expression for e_h is derived:

$$e_h = \frac{K}{K + \frac{1}{H-1}}. \tag{9}$$

It is convenient to express e_h in terms of the memory loading on the machine, which is defined as the reciprocal of H.

$$e_h = \frac{K}{K + \frac{1}{\frac{1}{\ell_h}-1}}. \tag{10}$$

For example, a memory loading of $\ell_h = 0.5$ is equivalent to the system having a heap size that is twice as large as required by the applications(s), i.e., $\ell_h = 0.5$ in Eq. 10 is equivalent to $H = 2$ in Eq. 9. Fig. 3.13 illustrates e_h as a function of total memory loading for several values of K.

Fig. 3.14 shows a two-dimensional surface plot of $e = e_c e_h$ derived from the idealized curve for e_c depicted in Fig. 3.12 and the curve for e_h shown in Fig. 3.13 (and Eq. 10) for $K = 10$. This is the efficiency function assumed for all the machines in the cluster for the simulations conducted in Chapter 5. From the figure, observe that if

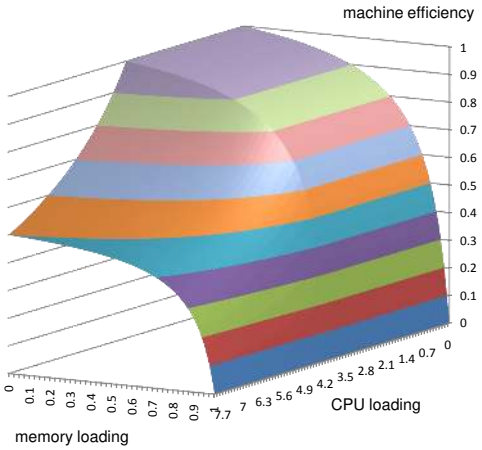

Figure 3.14: Derived machine efficiency surface based on the idealized curve for e_c in Fig. 3.12 and the e_h curve for $K = 10$ in Fig. 3.13.

total CPU loading of a machine exceeds its number of cores, then the efficiency of the machine decreases. Likewise, as the total memory loading of a machine increases, the efficiency of the machine decreases due to overhead associated with increased activity of the virtual machine's automatic memory management system. From the figure, observe that if a machine's CPU and memory resources are both lightly loaded, then the efficiency of the machine will be at or near its maximum value.

3.4.1.2 Request Executor

Once a request is assigned to a machine, the Request Executor module handles processing of the Request. Because multiple requests can be assigned to the same machine, the processing of these requests is done by following the machine efficiency model discussed in the previous section. The assigned requests can be either executing or completed.

As a machine finishes processing an executing request, then the request is said to be completed. At this time, the machine removes the completed request and notifies the Assigner about its completion. It also updates the machine's CPU and memory loading and hence the corresponding machine efficiency.

3.5 Concluding Remarks

In this chapter, simulation environment for scheduling in distributed system is presented. The architecture of the presented system has three major components: WFG Generator, Scheduler and Cluster of Machines. WFG is modeled as the directed acyclic graph composed of parallel and sequential combinations of RCs. Information regarding the duration, deadline and other parameters of WFGs are assumed to be known. Three types of WFGs, Interactive, Webservice and Batch, are generated by WFG Generator which are then sent to the Scheduler. The Scheduler contains the Scheduling Pool, Assigner, RSP and MSP. The Scheduling Pool holds all the executing WFGs and maintains a ready requests. The Assigner uses a RSP to select next request to be considered for execution from the ready requests and a MSP to select the best available machine to execute the selected request. The machine model presented comprehends the effect of CPU and heap memory loading due to requests running on it. The next chapter presents the detail analysis of RSPs and MSPs considered for evaluation.

Chapter 4

Scheduling Policies

4.1 Overview

The block diagram of the scheduling framework (Fig. 3.1) in Chapter 3 illustrates two types of policies used for scheduling: the Request Selection Policy (RSP) and the Machine Selection Policy (MSP). The RSP selects the next request to be considered to begin execution, whereas the MSP selects the framework machine to run.Multiple instances of each of these components is described in detail in Subsections 4.2 and 4.3.

4.2 Request Selection Policies

As described in Chapter 3, at each scheduling instant, the Assigner must decide which, if any, of the ready requests (associated with RCs currently under consideration in the Scheduling Pool) should be assigned to a machine and begin execution. The request selection policies considered in this thesis define a prioritization by specifying the order in which ready requests are considered for assignment and execution on a limited number of available machines. The seven RSPs considered are: First Come First Serve (FCFS), First Come Last Serve (FCLS), Earliest Deadline First (EDF),

Least Laxity First (LLF), Proportional Least Laxity First (PLLF), Shortest Workflow First (SWF), and Dynamic Shortest Workflow First (DSWF).

4.2.1 First Come First Serve

The FCFS policy uses the value of the time instant that a WFG arrives at the scheduler to define the relative priority for all requests associated with the WFG. This policy assigns the highest priority to the ready requests associated with the WFG that arrived furthest in the past and lowest priority to the ready requests of the most recently arrived WFGs. To illustrate, assume that during time instant t_i, five WFGs are in the Scheduling Pool at different stages of processing as shown in Fig. 4.1. Table 4.1 shows the parameters associated with those WFGs. In this case, the FCFS policy assigns priority for the ready requests based on the arrival time of their associated WFGs in the following order: WFG 1, WFG 2, WFG 3, WFG 4 and WFG 5. Because WFG 1 arrives earlier than the other WFGs, FCFS assigns higher priority to all the requests associated with WFG 1. Similarly, all the requests of WFG 2 will have higher priority than WFG 2, WFG 3, WFG 4 and WFG 5. Consider that the cluster can take two requests during this time instant t_i. The FCFS selects the ready request associated with WFG 1 which is illustrated in Fig. 4.2. Further assume that two executing requests associated with WFG 1 and WFG 5 finishes execution in time instant t_{i+1} as shown in Fig. 4.3. Note that in case of WFG 5, three parallel RCs are exposed resulting in three ready requests. Due to the completion of these two requests, two more request can be mapped on to the cluster. Fig. 4.4 shows the mapping effect due to FCFS. Because there is only one ready request in WFG 1, the Assigner selects the ready request of WFG 2 in addition to WFG 1 for the purpose of mapping.

Although not illustrated in this example, it is possible for a lower priority request to be assigned before a higher priority request. This can happen when the resource

Table 4.1: WFG Parameters.

WFG	Type	Arrival Time	Duration	Deadline
1	Batch	5	1400	2000
2	Webservice	10	300	500
3	Interactive	15	15	20
4	Batch	20	900	1100
5	Webservice	25	225	325

requirements of the high priority request are not available in the cluster, but the resources needed for the lower priority request are available. Thus, high priority requests are given first opportunity for assignment, but lower priority requests can be mapped ahead of higher priority requests if their requirements are better matched to available resources.

The pseudocode of the FCFS implementation is presented in Fig. 4.5. During scheduling instant t_i, the FCFS sorts ready requests $R(t_i)$ according to their parent WFG born time b_w and assigns them to a list denoted as r_{list} (line 2). The sorted list r_{list} (line 3) is then returned to the Assigner module of the Scheduler discussed in Chapter 3.The Assigner module then maps those requests to the machines of cluster depending upon their availability. The complexity of the FCFS is $O(|R(t_i)|\log|R(t_i)|)$. The FCFS [42, 43] policy is the simplest of the policies considered; it can make poor decisions because it does not consider the deadline of the WFGs in assigning request priorities. The FCFS policy is included here primarily to serve as a baseline upon which the other more sophisticated policies are compared.

4.2.2 First Come Last Serve

The FCLS also uses the value of the time instant that a WFG arrives at the scheduler to define the priority for all requests associated with the WFG. In contrast to the FCFS, the FCLS policy assigns a higher priority to all requests in WFGs that are born recently. For the example considered in Fig. 4.1, this policy prioritizes in the

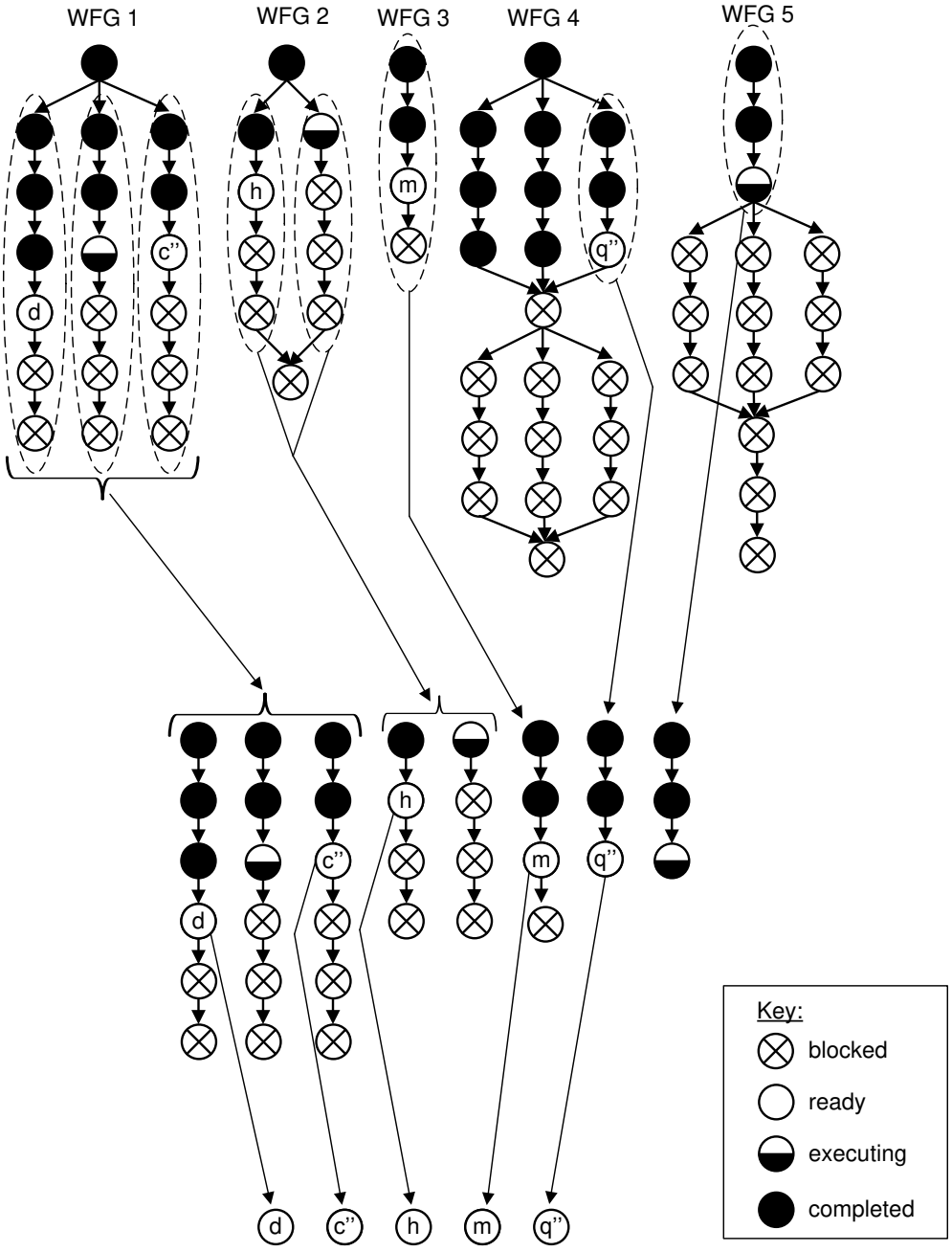

Figure 4.1: Snapshot of the Scheduling Pool at scheduling instant t_i before assignment.

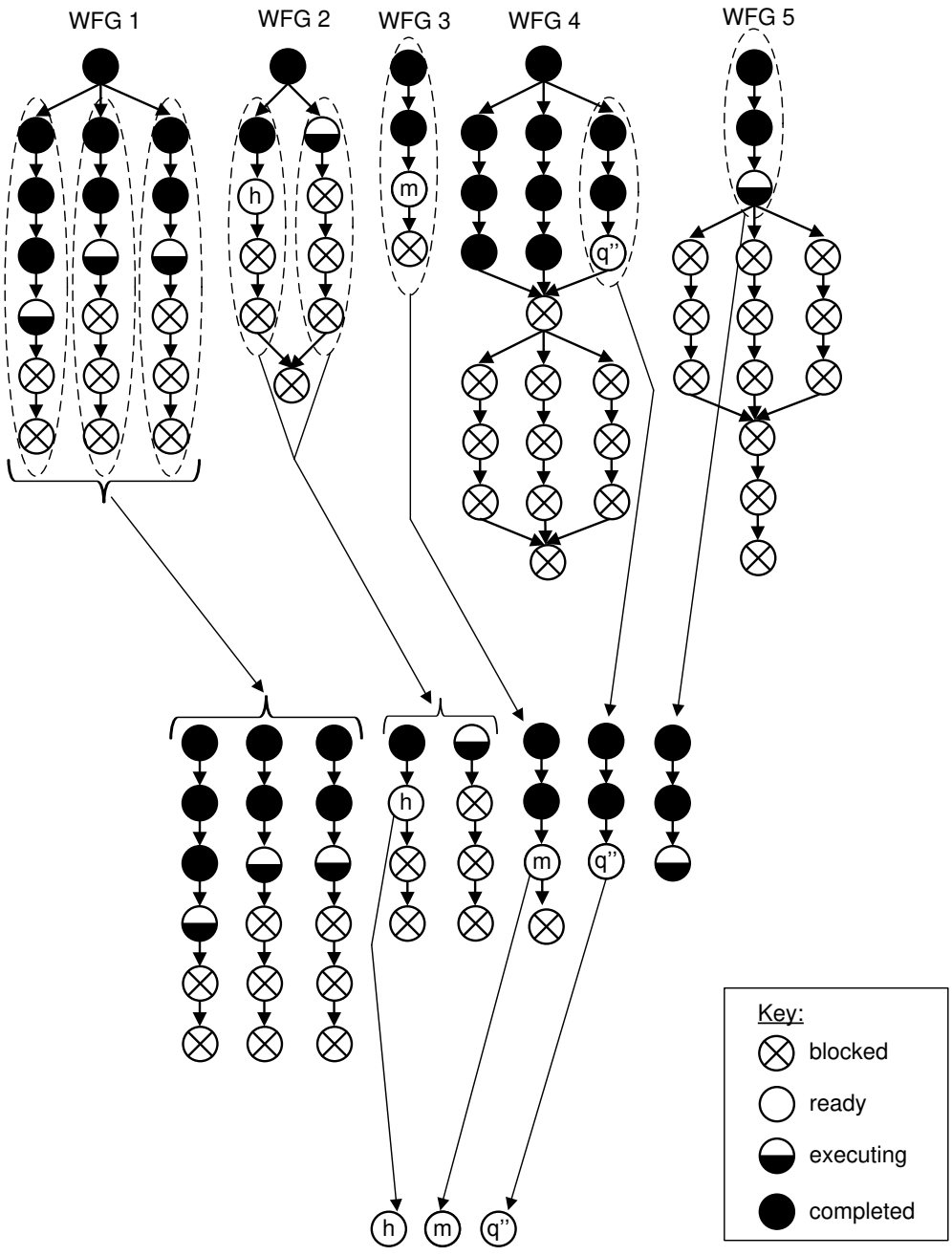

Figure 4.2: Snapshot of the Scheduling Pool at scheduling instant t_i after assignment.

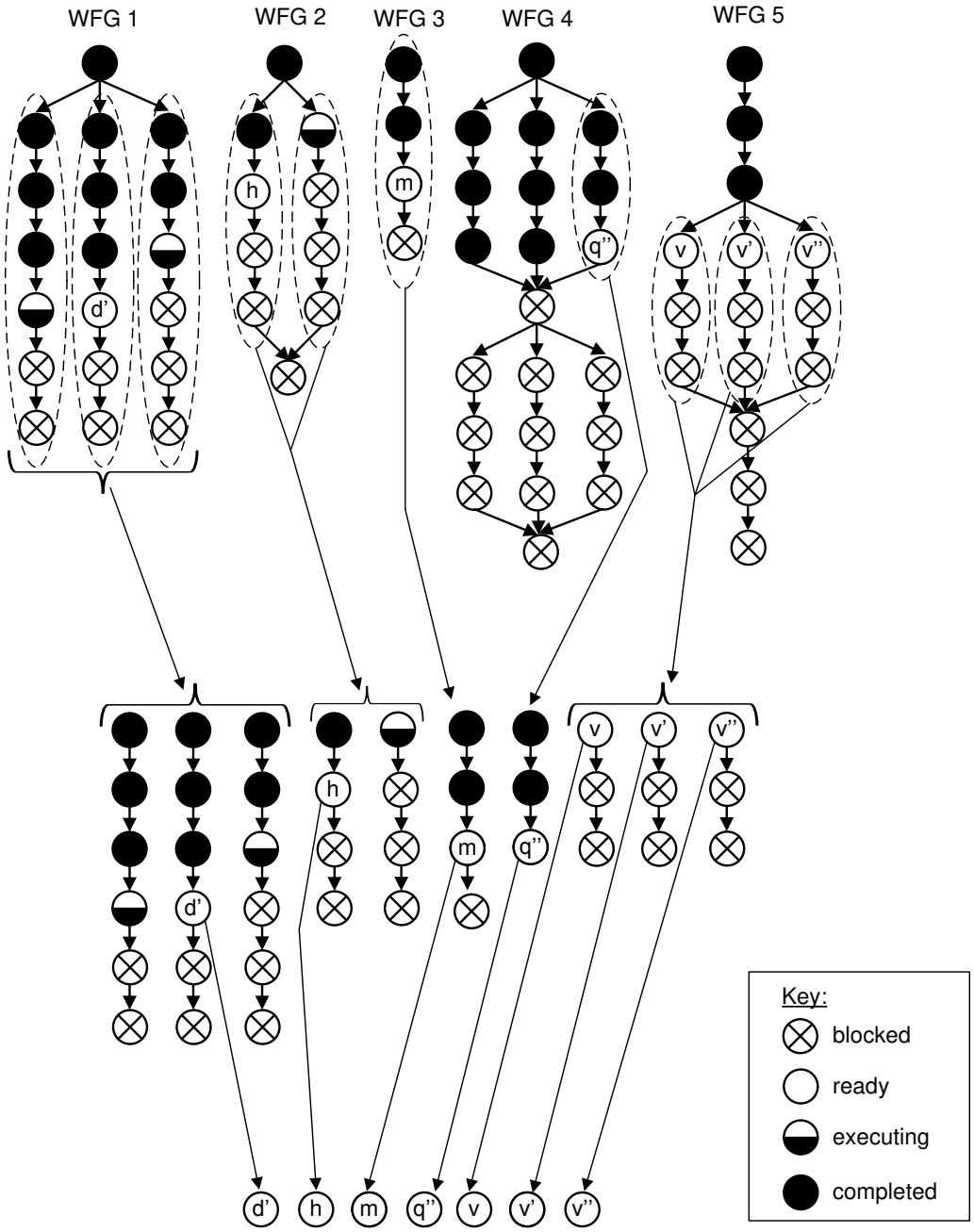

Figure 4.3: Snapshot of the Scheduling Pool at scheduling instant t_{i+1} before assignment.

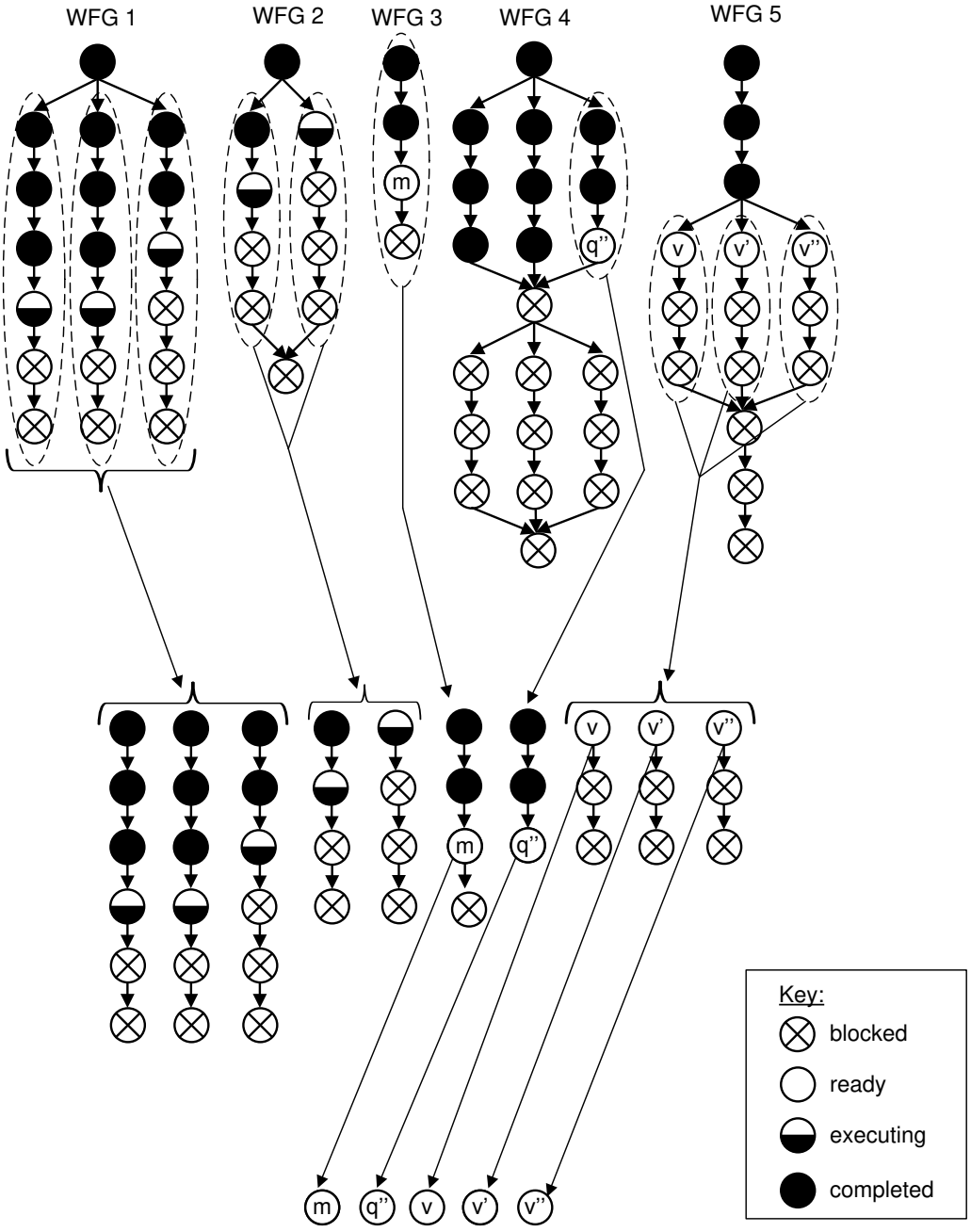

Figure 4.4: Snapshot of the Scheduling Pool at scheduling instant t_{i+1} after assignment.

FCFS($R(t_i)$)
1 **for scheduling instant** t_i
2 $r_{list} \leftarrow sort(R(t_i), b_{w(r)})$
3 **return** r_{list}

Figure 4.5: Pseudocode for the FCFS.

following order: WFG 5, WFG 4, WFG 3, WFG 2 and WFG 1, which is exactly opposite to the FCFS prioritization.

FCLS($R(t_i)$)
1 **for scheduling instant** t_i
2 $r_{list} \leftarrow sort(R(t_i), b_{w(r)})$
3 **return** r_{list}

Figure 4.6: Pseudocode for FCLS.

The pseudocode of FCLS implementation is presented in Fig. 4.6. For scheduling instant t_i, FCLS sorts ready requests $R(t_i)$ in descending order according to their parent WFG born time b_w and assigns them to r_{list} (line 2). The sorted list r_{list} (line 3) is then returned to the Assigner module of the Scheduler. The complexity of FCLS is $O(|R(t_i)|\log|R(t_i)|)$. Similar to FCFS, this policy is likely to make poor decision as it does not consider the deadlines of the WFGs in assigning priorities.

4.2.3 Earliest Deadline First

The EDF policy [24] prioritizes all the ready requests of a WFG using the deadline associated with the WFG. Recall from Chapter 3 that a deadline is associated with each WFG, and this information is assumed to be known by the scheduler. This policy prioritizes all the ready requests associated with WFGs based on the nearest deadline. For the scenario described in the Fig. 4.1 (refer to Table 4.1 for deadlines), the EDF policy prioritizes the execution the ready requests of the associated WFGs in the following order: WFG 3, WFG 5, WFG 2, WFG 4 and WFG 1. As in FCFS, assume that only two requests can be mapped during scheduling instant t_i. This

policy selects ready request associated with WFG 3 and WFG 2 as shown in Fig. 4.7. Even though WFG 5 has higher priority than WFG 2, no requests can be selected because none of them are in a ready state. During the next scheduling instant, two requests complete execution as shown in Fig. 4.8. During this time instant, EDF selects two requests associated with WFG 5 as illustrated in Fig. 4.9.

The pseudocode of the EDF implementation is presented in Fig. 4.10. Because the EDF is a dynamic scheduling policy, the priority of the ready requests associated with the previously running WFGs changes as new WFGs with nearer deadlines arrive. This is one of the policies considered when implementing SLA based scheduling [44, 45, 46] where the terms and conditions of an SLA are modeled in terms of a deadline for each WFG. Since EDF uses a static deadline parameter, there is no additional computation compared to FCFS or FCLS. Hence the complexity of EDF is also the same as FCFS and FCLS, i.e., $O(|R(t_i)|\log|R(t_i)|)$.

4.2.4 Least Laxity First

The LLF policy [24] prioritizes ready requests of a WFG according to their laxity, which is defined as the difference between the deadline of the WFG and the estimated finish time of the WFG. The rationale for giving priority to the requests with smaller values of laxity over larger values is because smaller values of laxity correspond to WFGs that are more likely to miss their deadlines. Laxity values can be negative which have priority over positive laxity values because negative laxity is an indication that the WFG deadline will likely be missed. Unlike the FCFS, FCLS, and EDF scheduling policies, which assign a static priority value to a WFG, the priority values assigned by LLF generally vary with time. To estimate the laxity of a WFG at time instant t_i, denoted as $\tilde{\mathcal{L}}_w(t_i)$, the estimated finish time of the WFG is subtracted from the WFG's deadline as in Eq. 11.

At each scheduling instant, an estimate of each WFG's finish time, denoted as

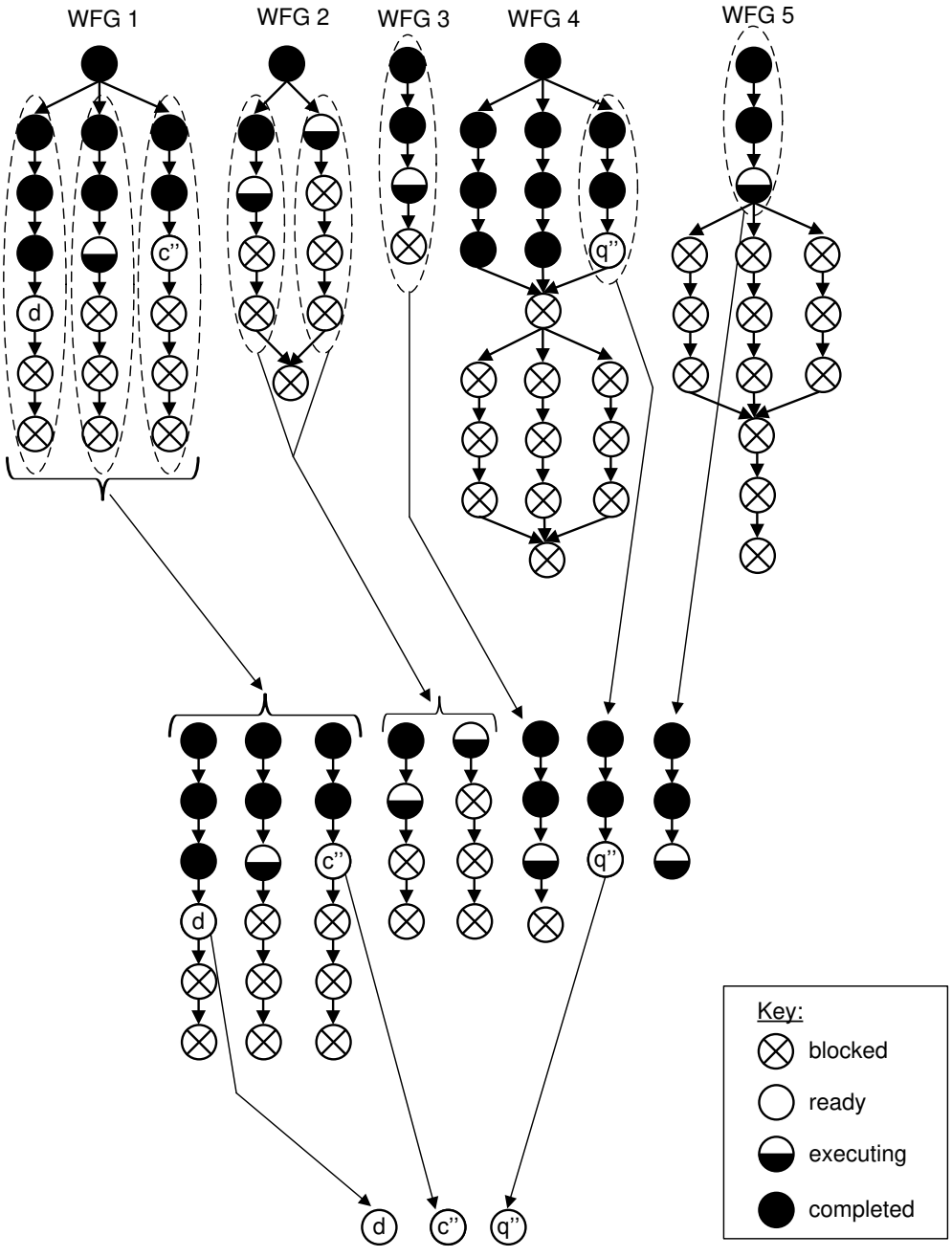

Figure 4.7: Snapshot of the Scheduling Pool at scheduling instant t_i after assignment.

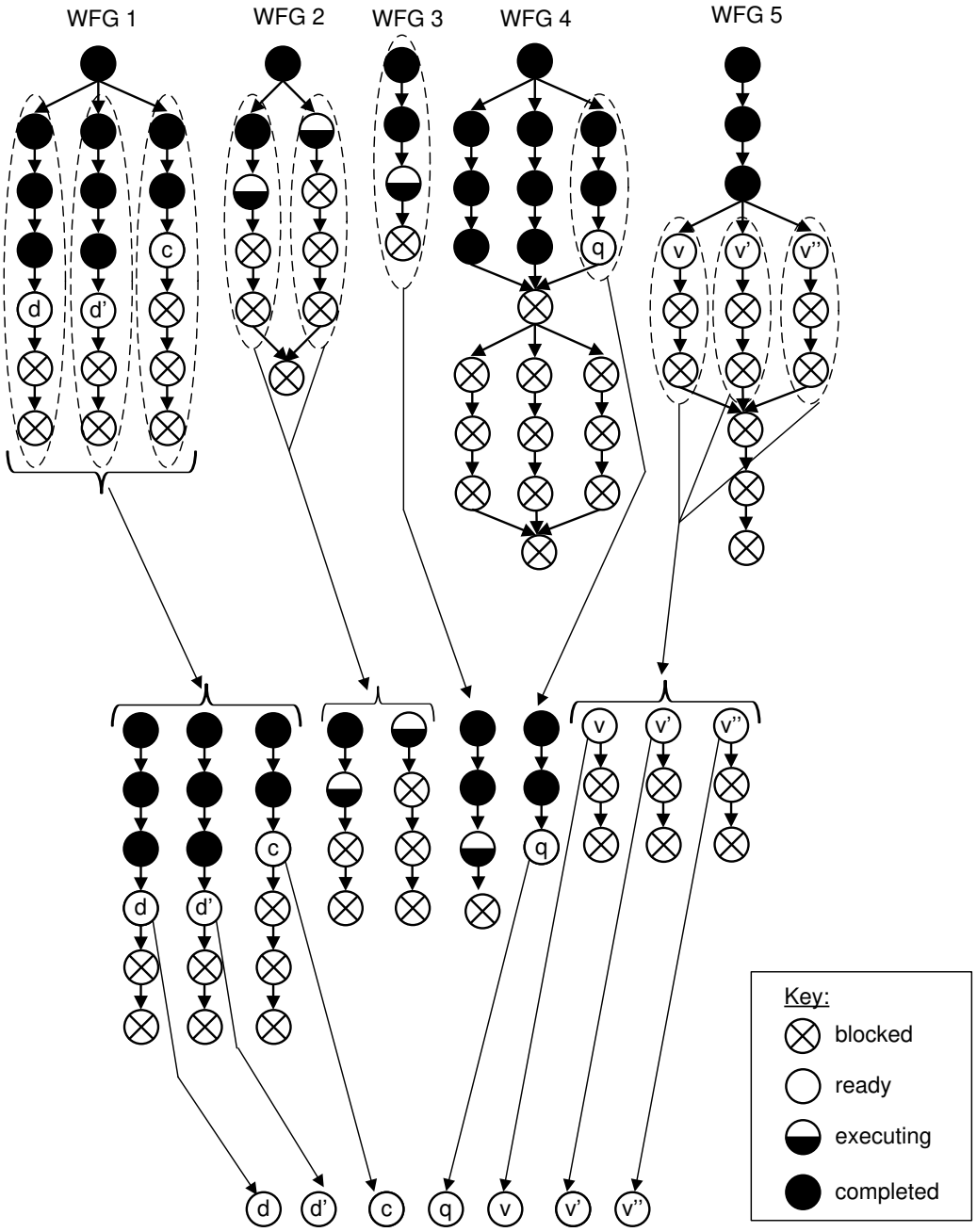

Figure 4.8: Snapshot of the Scheduling Pool at scheduling instant t_{i+1} before assignment.

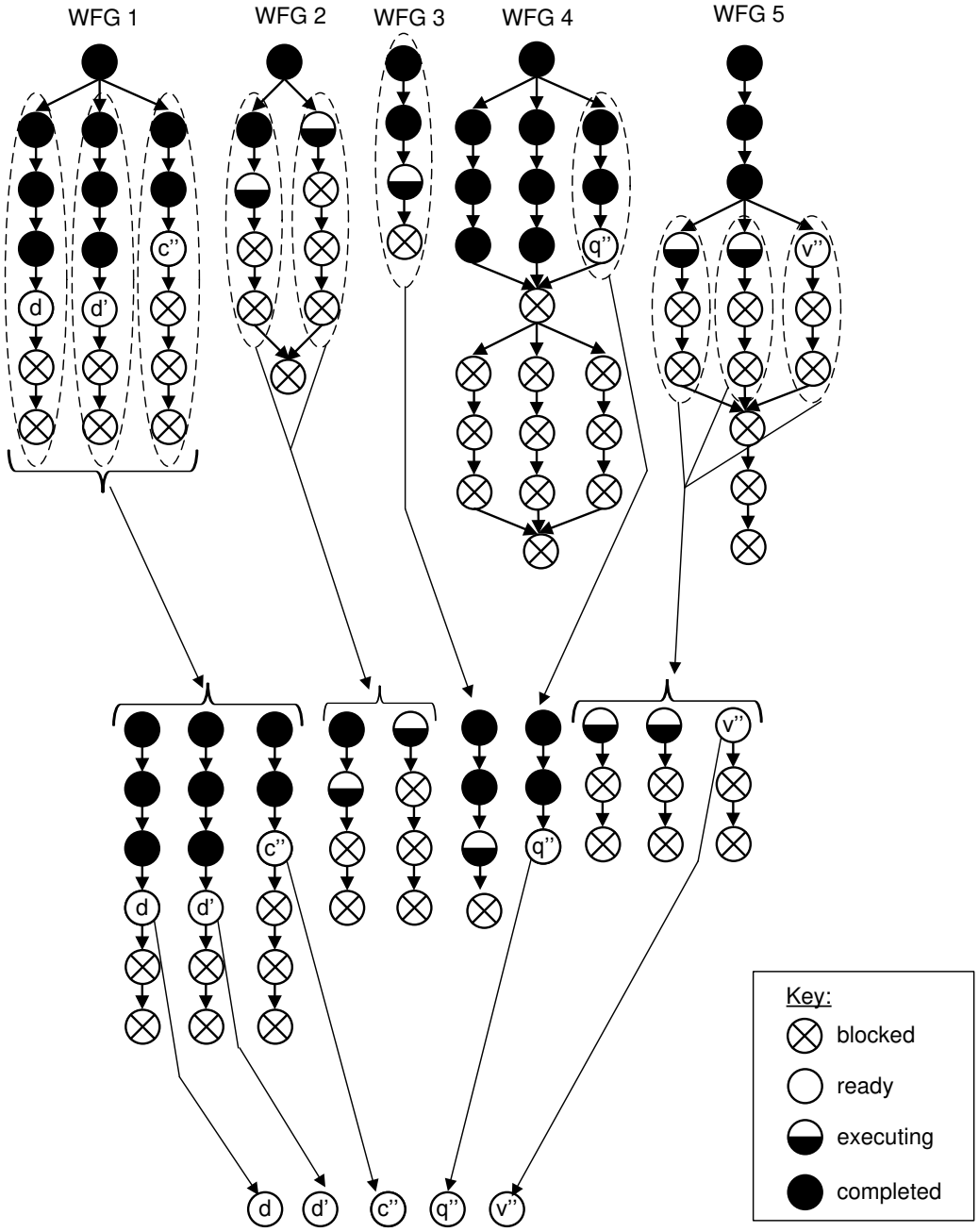

Figure 4.9: Snapshot of the Scheduling Pool at scheduling instant t_{i+1} after assignment.

EDF$(R(t_i))$
1 **for scheduling instant** t_i
2 $r_{list} \leftarrow sort(R(t_i), d_{w(r)})$
3 **return** r_{list}

Figure 4.10: Pseudocode for the EDF.

$\tilde{f}_w(t_i)$, is calculated using Eq. 12. The estimate of the finish time of a WFG w is based on the rate at which work is being done. Here, the numerator represents the product of the WFG execution duration $(t_i - s_w)$ and the amount of work remaining $\left(\sum_{r \in w} C_r - \sum_{\substack{r \in w \\ \tilde{f}_r < t_r}} C_r\right)$, whereas, the term in the denominator $\left(\sum_{\substack{r \in w \\ f_r < t_i}} C_r\right)$ represents the amound of work completed for a WFG w. Note that in the equation, if a WFG w has not started then $s_w = 0$ and the denominator $\left(\sum_{\substack{r \in w \\ f_r < t_i}} C_r\right)$ is also zero. In this case, the estimated finish time is obtained by evaluating the expected ideal duration of the WFG as described in Chapter 3.

$$\tilde{\mathcal{L}}_w(t_i) = d_w - \tilde{f}_w(t_i). \tag{11}$$

$$\tilde{f}_w(t_i) = t_i + (t_i - s_w) \left(\sum_{r \in w} C_r - \sum_{\substack{r \in w \\ \tilde{f}_r < t_i}} C_r\right) \Bigg/ \left(\sum_{\substack{r \in w \\ f_r < t_i}} C_r\right). \tag{12}$$

The pseudocode of LLF implementation is presented in Fig. 4.11. Here, for scheduling instant t_i, the laxity value is calculated for all the ready requests' associated WFGs (line 2). Based on the laxity value calculated, all the ready requests are sorted in ascending order according to their laxity value (line 5). The sorted list r_{list} is then returned to the Assigner module of the Scheduler (line 6). The complexity of LLF is $O(|R(t_i)||R_w(t_i)|)+O(|R(t_i)|\log|R(t_i)|)$, where $|R_w(t_i)|$ represents all the requests of a WFG w. The increase in complexity is due to the additional computation associated with estimating the finish time of WFGs under consideration.

LLF$(R(t_i))$
1 **for scheduling instant** t_i
2 **for each** $r \in R(t_i)$ **do**
3 $\quad \tilde{\mathcal{L}}_{w(r)} \leftarrow d_{w(r)} - \tilde{f}_{w(r)}$
4 **end for**
5 $r_{list} \leftarrow sort(R(t_i), \tilde{\mathcal{L}}_{w(r)})$
6 **return** r_{list}

Figure 4.11: Pseudocode for LLF.

LLF is a dynamic scheduling approach which also considers the deadline information of the WFGs like EDF. However, the LLF policy is even more dynamic than EDF in terms of assigning priorities to requests, because although the deadline value does not changes over time, but the laxity value changes as time passes.

4.2.5 Proportional Least Laxity First

The PLLF policy is an enhancement of the LLF policy that uses a proportional laxity value to prioritize ready requests of a WFG. Because the LLF policy is insensitive to the size of WFGs, this policy can be made to overcome the size problem by implementing the proportional laxity. The proportional laxity value of a WFG, denoted as $\widetilde{\mathcal{PL}}_w$, is defined as the laxity value $d_w - \tilde{f}_w$ divided by the ideal execution time $d_w - b_w$ of the WFG as follow:

$$\widetilde{\mathcal{PL}}_w = \frac{\tilde{\mathcal{L}}_w(t_i)}{d_w - b_w} = \frac{d_w - \tilde{f}_w}{d_w - b_w}. \tag{13}$$

Consider the following example to illustrate the underlying rationals for defining PLLF. When WFGs deadlines are being missed, PLLF assigns higher priority to the smaller WFGs because the proportional laxity of small WFGs tend to be less compared to large WFGs. Table 4.2 shows that even though the laxity of WFG P is less than the laxity of WFG Q, the proportional laxity of WFG Q is less than WFG P's. Thus PLLF assigns priority to WFG Q having the smaller proportional laxity

value. In cases where no deadlines are being missed, PLLF assigns higher priority to the larger WFGs than the smaller WFGs. Table 4.3 shows that PLLF assigns higher priority to WFG R even though its laxity is greater than the laxity of WFG S.

Table 4.2: Example 1 for PLLF.

WFG	b_w	$\tilde{f}_w$	d_w	$\tilde{\mathcal{L}}_w$	$\widetilde{\mathcal{PL}}_w$
P	0	110	100	-10	-1/10
Q	80	105	100	-5	-1/4

Table 4.3: Example 2 for PLLF.

WFG	b_w	$\tilde{f}_w$	d_w	$\tilde{\mathcal{L}}_w$	$\widetilde{\mathcal{PL}}_w$
P	0	110	120	10	1/12
Q	80	105	110	5	1/6

The pseudocode of PLLF is presented in Fig. 4.12. For each scheduling instant t_i, the proportional laxity of each of the ready requests' $R(t_i)$ associated WFG, denoted as $\widetilde{\mathcal{PL}}_{w(r)}$, are calculated (line 3). Then the ready requests are sorted based on the proportional laxity (line 5) and assigned to a list r_{list}. The sorted list is then returned to the Assigner module of the Scheduler. The complexity of PLLF is similar to LLF that is $O(|R(t_i)||R_w(t_i)|) + O(|R(t_i)|\log|R(t_i)|)$.

PLLF($R(t_i)$)
1 **for scheduling instant** t_i
2 **for each** $r \in R(t_i)$ **do**
3 $\quad \widetilde{\mathcal{PL}}_{w(r)} \leftarrow \tilde{\mathcal{L}}_w(t_i)/(d_{w(r)} - b_{w(r)})$
4 **end for**
5 $r_{list} \leftarrow sort(R(t_i), \widetilde{\mathcal{PL}}_{w(r)})$
6 **return** r_{list}

Figure 4.12: Pseudocode for PLLF.

4.2.6 Shortest Workflow First

The SWF policy prioritizes ready requests according to the least processing duration $\mathcal{D}_w$ of associated WFGs. This policy attempts to maximize throughput and minimize average wait time for each request. If small WFGs continue to arrive, the large WFGs will ultimately suffer from process starvation. Due to the potential for process starvation, this policy may not be suitable for a realistic environment where different types and sizes of WFGs need to be handled. Fig. 4.13 shows a snapshot of the Scheduling Pool where each WFG contains only one RC and assumes that each request has the same duration. Because SWF orders according to the duration in ascending order, the scheduling order would be: D, C, E, B, F, A and G. Note that even though WFG A has only one request remaining, SWF assigns the same priority to A and G. Because WFGs B, C, D, E and F are shorter than WFGs A and G, SWF prioritizes them resulting in process starvation for WFGs A and G. If there are machines that can take two requests then this policy assigns requests associated with WFG D and either C or E as shown in Fig. 4.14.

The pseudocode of the SWF implementation is presented in Fig. 4.15. For scheduling instant t_i, SWF sorts ready requests $R(t_i)$ according to their parent WFG's ideal duration $\mathcal{D}_w$ (line 2). The complexity of SWF is $O(|R(t_i)|\log|R(t_i)|)$.

4.2.7 Dynamic Shortest Workflow First

The DSWF policy prioritizes the ready requests according to the least estimated finish time of their parent WFGs. Note that DSWF is a dynamic version of SWF where the shortest WFG is determined by least estimated time to finish. The DSWF policy performs a similar function to SWF but it attempts to minimize process starvation by prioritizing WFGs which have the least remaining work. Consider the example in Fig. 4.13, DSWF avoids process starvation by ordering WFGs in the following order: A, D, C, B, E, F and G. Even though WFG A has longer duration than WFG D,

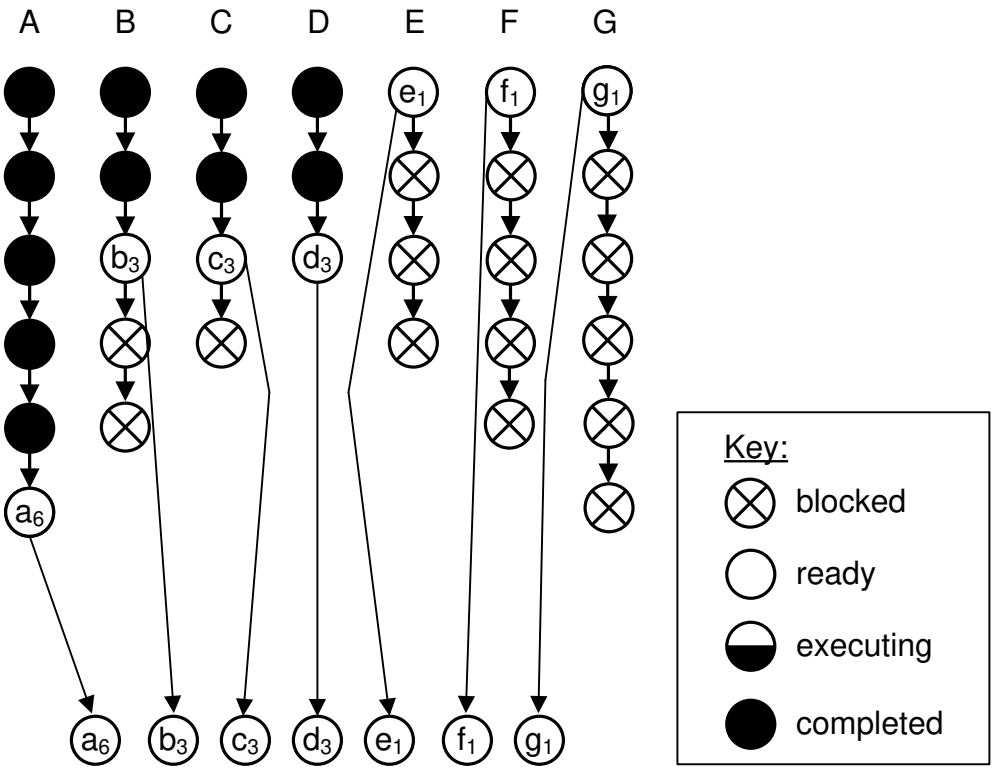

Figure 4.13: Snapshot of the Scheduling Pool at time instant t_i before assignment.

their effective work remaining is the same in both cases. Thus WFG A and D have the same priority. If there are machines that can take two requests, then this policy assigns the ready request associated with WFG A and WFG D as shown in Fig. 4.16. DSWF is an opportunistic approach similar to SRPT [47], trying to minimize the number of outstanding WFGs. This approach also attempts to minimize the mean response time, validating Little's law [48].

The pseudocode of DSWF implementation is presented in Fig. 4.17. For the scheduling instant t_i, DSWF sorts ready requests $R(t_i)$ by giving priority to smaller estimated finish times of the associated WFGs $\tilde{f}_w$ (line 5). The complexity of DSWF is $O(|R(t_i)||R_w(t_i)|) + O(|R(t_i)|\log|R(t_i)|)$ which is slightly more than SWF due to the additional computation of $\tilde{f}_w$.

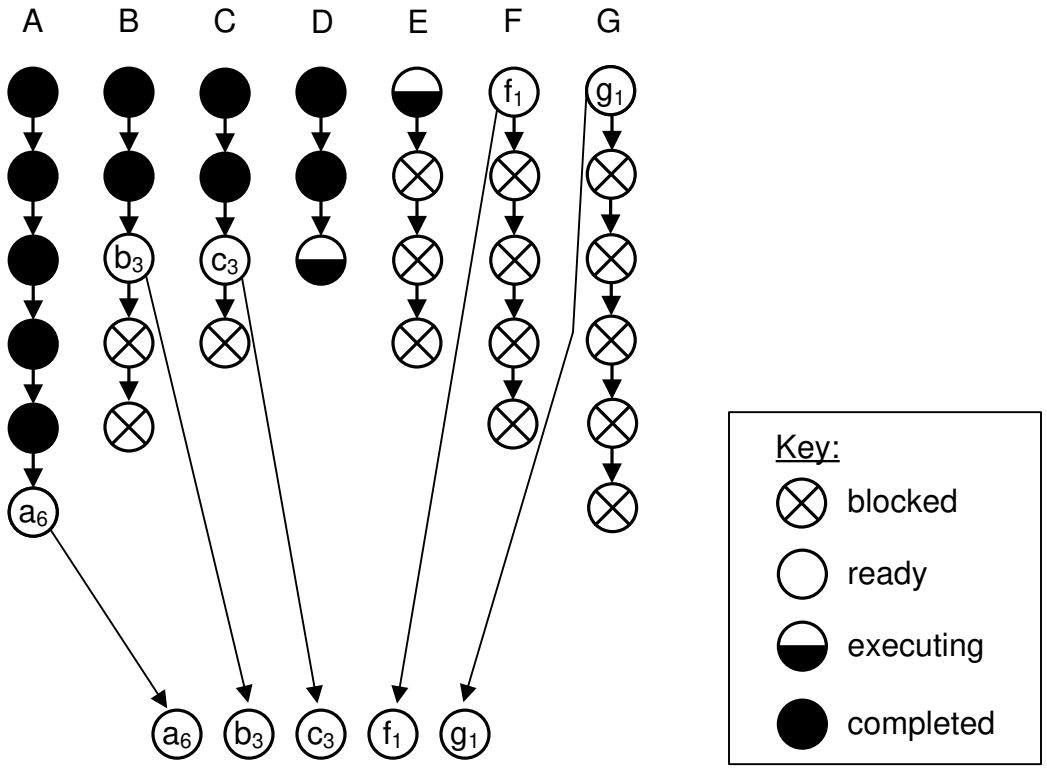

Figure 4.14: Snapshot of the Scheduling Pool at time instant t_i after assignment.

SWF$(R(t_i))$
1 **for scheduling instant** t_i
2 $r_{list} \leftarrow sort(R(t_i), \mathcal{D}_{w(r)})$
3 **return** r_{list}

Figure 4.15: Pseudocode for SWF.

4.3 Machine Selection Policies

In the scheduling framework presented in Chapter 3, a major consideration is the machine selection policy. Here, the machine selection policy determines which machine is appropriate for processing a request. Similar to the request selection policies described in the previous section, the machine selection policy is also important for the efficient utilization of resources. There are various machine selection strategies that can be implemented. In this section, six machine selection heuristics are discussed: Best Pre-Mapping (BPRM), Best Post-Mapping (BPOM), Best CPU (BC), Best Memory

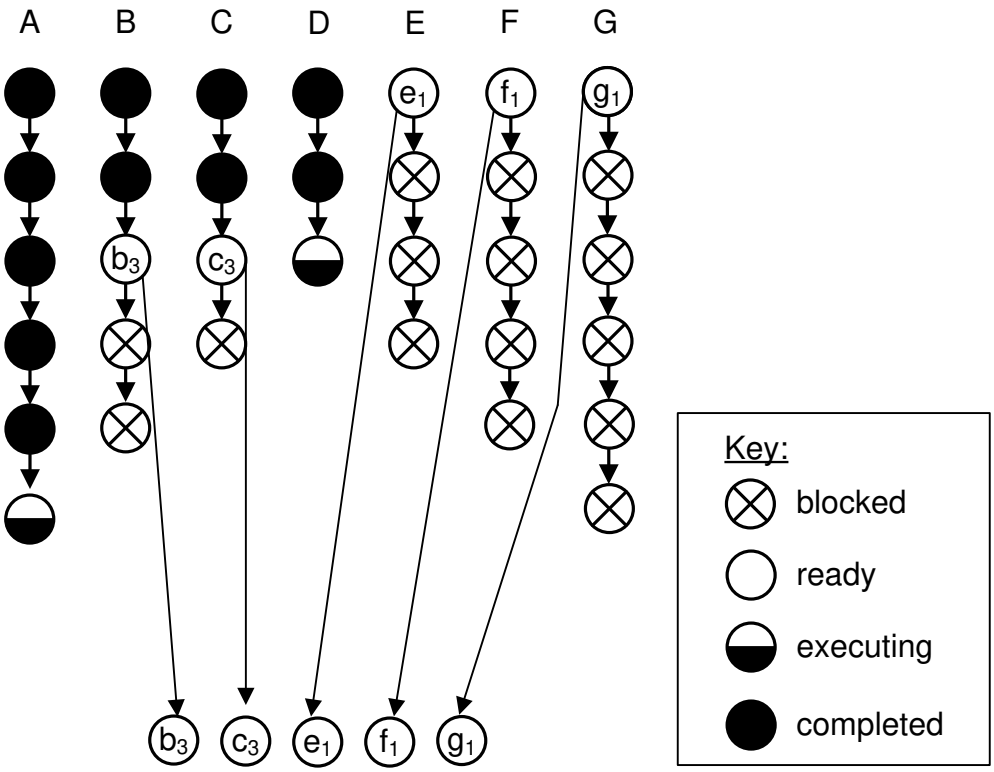

Figure 4.16: Snapshot of the Scheduling Pool at time instant t_i after assignment.

(BM), Least Deadline Missed (LDM) and Least Work Remaining (LWR).

For all policies considered in this thesis, a ready request is only assigned to a machine that is declared to be available. The availability of a machine is determined using defined threshold values associated with the machine's current efficiency due to CPU and memory loadings. Specifically, a machine is declared to be available only if its efficiency due to CPU and/or memory loadings is above the specified threshold value.

4.3.1 Best Pre-Mapping

The BPRM policy selects a machine that has the highest pre-mapping efficiency. Here, pre-mapping means the current efficiency of a machine that does not include the effects of the requests that are going to be mapped. Whereas, the post mapping

DSWF$(R(t_i))$

1 **for scheduling instant** t_i
2 **for each** $r \in R(t_i)$ **do**
3 $compute\tilde{f}_w(r)$
4 **end for**
5 $r_{list} \leftarrow sort(R(t_i), \tilde{f}_{w(r)})$
6 **return** r_{list}

Figure 4.17: Pseudocode for DSWF.

refers to the pseudo mapping for the purpose of calculating efficiency value. Fig. 4.18 illustrates the effect of pre and post mapping a request r on to two quad core machines. In the figure D_r, U_r and H_r represent the duration, CPU utilization and heap memory requirement of r respectively. The value of ℓ_c represents the CPU loading factor, which is the sum of the U_r's of all requests executing on the machine. Similarly, the value of ℓ_h represents the normalized heap memory loading factor which is also sum of the H_r's of all requests executing on that machine, $0 < \ell_h < 1$. The efficiency of a machine e is defined by the CPU and heap memory loading on that machine using Eq. 2. Before mapping request r on to any of the machines, the efficiency of Machine 2 is greater than Machine 1. The BPRM selects Machine 2 over Machine 1 because of its higher pre-mapping efficiency. In this case, assume the threshold value used is 0.70. Fig. 4.19 shows that the BPRM policy selects the Machine 2 because of its high pre-mapping efficiency.

The pseudocode for the BPRM implementation is presented in Fig. 4.20. In the figure, r represents the request that is to be mapped on a machine M of cluster represented as $\mathcal{M}$. As discussed in the examples, each machine in the cluster can only be considered for mapping which has efficiency higher than the defined threshold value represented by $\hat{e}$ (line5). Here, the threshold value is evaluated before the mapping of request r so the efficiency of machine can go below the defined threshold value (refer to of Fig. 4.19) depending upon the the request's characteristics. Due to this reason, this policy does not guarantee the machine efficiency values of $e \geq \hat{e}$ after the

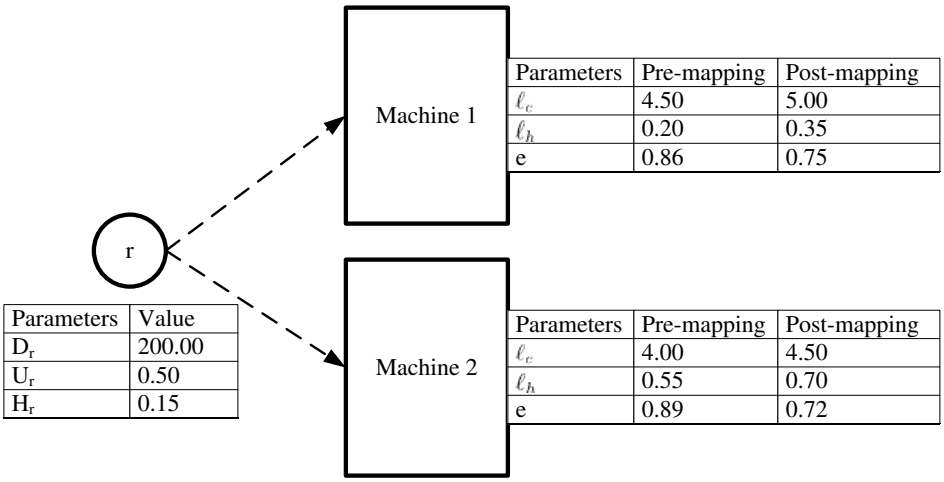

Figure 4.18: Pre and Post mapping of a request r on to two quad-core machines.

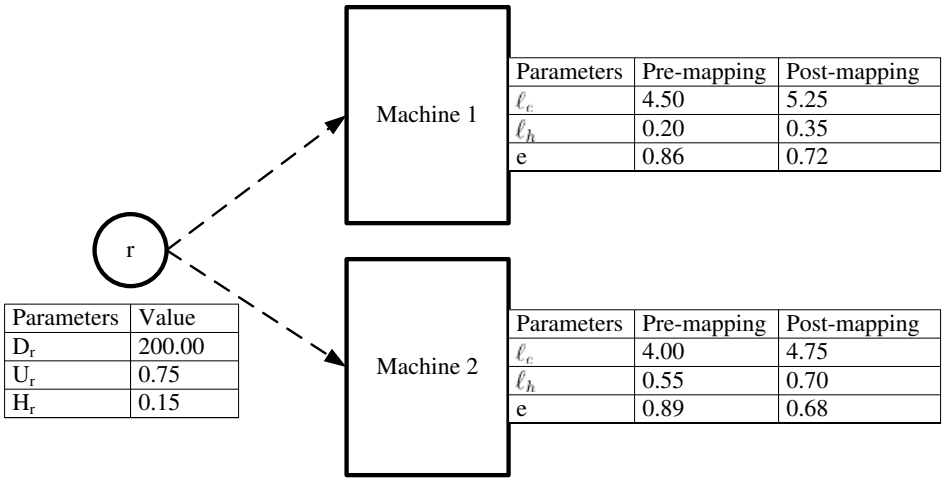

Figure 4.19: Pre and Post mapping of a request r on to two quad-core machines.

mapping is performed. Once the threshold filter is valid, the next step is to select the machine which has the highest pre-mapping efficiency (line 6 to line 9). Finally, the best machine in *result* is returned to the Assigner (line 12) module of the Scheduler for the assignment. The computational complexity of BPRM is $O(|\mathcal{M}|)$.

4.3.2 Best Post-Mapping

The BPOM is a post mapping machine selection policy that selects a machine based on post mapping analysis. The post mapping analysis refers to the estimation of

```
BPRM(r, M, ê)
1  result ← null
2  max ← −∞
3  for each M ∈ M do
4     e ← e(R_M(t_i))
5     if e ≥ ê then
6        if e > max then
7           result ← M
8           max ← e
9        end if
10    end if
11 end for
12 return result
```

Figure 4.20: Pseudocode for BPRM.

the machine's efficiency due to pseudo (i.e., "what if" mapping of request r on to machine M. Because this policy requires the request's CPU and memory utilization, it can only be applied when the CPU and memory characteristics of requests are known. For the example in Fig. 4.18, the BPOM selects Machine 1 because it has a higher post mapping efficiency. In case of the example in Fig. 4.19, this policy filters Machine 2 because it falls below the threshold value and hence chooses the Machine 1 by default. Due to the post mapping effect, the machine efficiency is guaranteed not go below the defined threshold values.

The pseudocode of BPOM implementation is presented in Fig. 4.21. For a give request r, this policy selects the best available machine from the cluster of machines $\mathcal{M}$. This policy considers threshold as the post mapping efficiency (line 4 and line 5). If a machine passes a threshold value then it's efficiency is compared with already known best machine (line 6) and the machine with the highest efficiency is selected (line 7). In the post mapping there is additional computation associated with computing the pseudo mapping efficiency that is $e(R_M(t_i) \cup \{r\})$. Thus the computational complexity of BPOM is $O(|\mathcal{M}||e(R_M(t_i) \cup \{r\})|)$.

```
BPOM(r, M, ê)
1  result ← null
2  max ← −∞
3  for each M ∈ M do
4     e ← e(R_M(t_i) ∪ {r})
5     if e ≥ ê then
6        if e > max then
7           result ← M
8           max ← e
9        end if
10    end if
11 end for
12 return result
```

Figure 4.21: Pseudocode for BPOM.

4.3.3 Best CPU

The BC policy is a type of post mapping that selects a machine with the highest CPU efficiency value. In cases where requests are CPU bound it would be better to choose a machine which has a high CPU efficiency. Examples of CPU bound tasks are encoding, compression, sorting and searching. Fig. 4.22 illustrates the effect of mapping a request r on to two quad core machines using BC. Note that in the figure, both machines have the same pre-mapping CPU efficiency, but Machine 2 has a higher post mapping CPU efficiency. Since BC selects the machine with the highest CPU efficiency, Machine 2 is selected. If the memory intensive requests are being executed on Machine 2, selecting the machine results in poor performance even though it has high CPU efficiency. In Fig. 4.23 Machine 2 has CPU intensive requests being executed and it's overall efficiency is lower than Machine 1. Hence, in this case BC does not produces a good result.

The pseudocode for the BC implementation is presented in Fig. 4.24. For a give request r, this policy searches all the machines $\mathcal{M}$ in cluster for the purpose of mapping (line3). The post mapping threshold is analyzed (line 4 and line 4) and depending on the availability of machines, the machine with the highest CPU

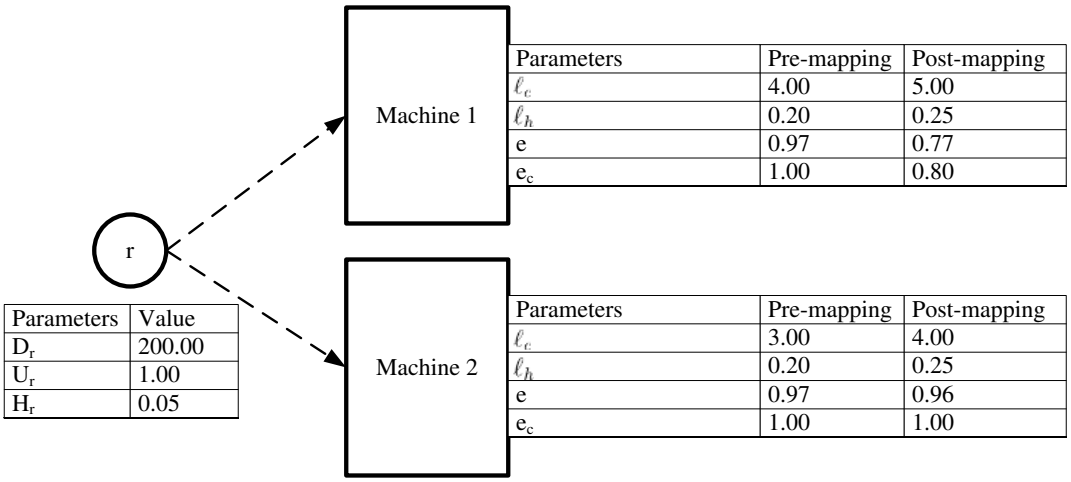

Figure 4.22: Mapping of a request r on to two quad-core machines using BC.

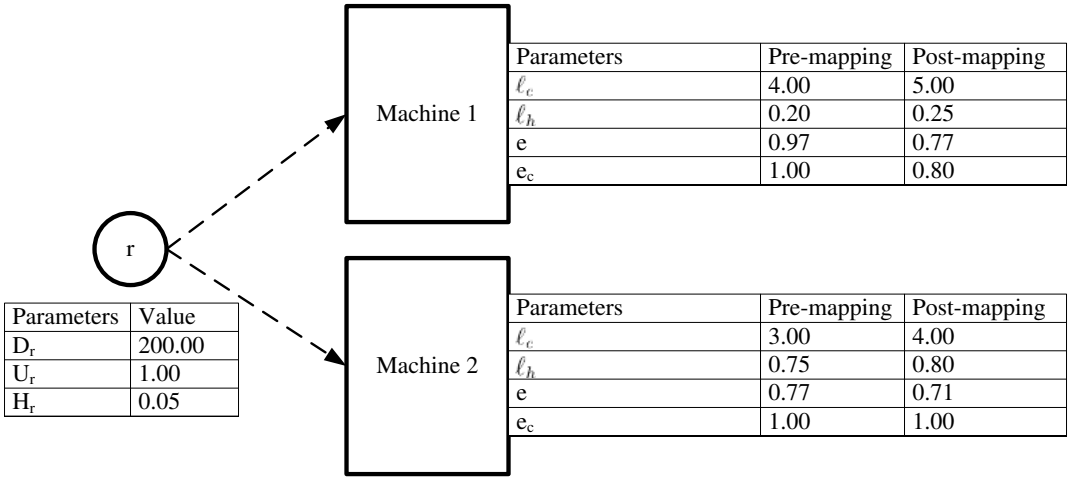

Figure 4.23: Mapping of a request r on to two quad-core machines using BC.

efficiency is selected (line 5 to 7). The computational complexity of BC is equivalent to the complexity of the post mapping policy that is $O(|\mathcal{M}||e(R_M(t_i) \cup \{r\})|)$.

4.3.4 Best Memory

Some applications need large a amount of memory. For example, huge scientific simulations, graphics application, database application and so forth require a large amount of memory. When clients send their processing requests for these applications, the amount of memory required depends on the request's memory characteristics. In

```
BC(r, M, ê)
1  result ← null
2  max ← −∞
3  for each M ∈ M do
4     e ← e(R_M(t_i) ∪ {r})
5     if e ≥ ê then
6        e_c ← e_c(R_M(t_i) ∪ {r})
7        if e_c > max then
8           result ← M
9           max ← e_c
10       end if
11    end if
12 end for
13 return result
```

Figure 4.24: Pseudocode for BC.

such cases, it would be better to choose a machine which has the highest free memory. The BM policy is also a type of BPOM which performs the post mapping analysis of the heap memory and selects a machine which has the highest memory efficiency. Fig. 4.25 illustrates the effect of mapping a request r on two quad core machines using BM. In the figure both pre and post mapping memory efficiency of Machine 1 is higher than Machine 2. Therefore this policy selects Machine 1 even though it's pre mapping overall efficiency is lower than the overall efficiency of Machine 2.

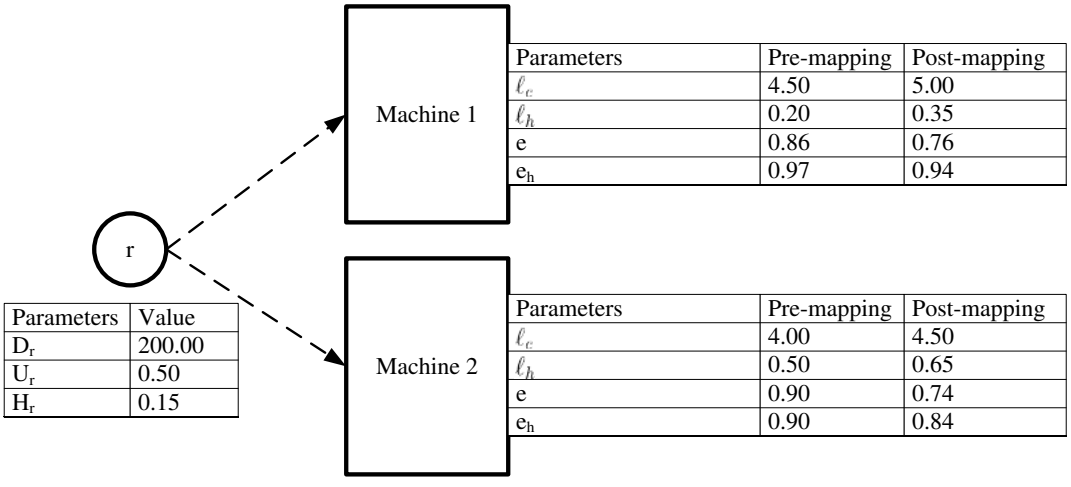

Figure 4.25: Mapping of a request r on to two quad-core machines using BM.

The pseudocode for the BM implementation is presented in Fig. 4.26. For a give request r, this policy searches all the machines $\mathcal{M}$ in the cluster for the purpose of mapping (line3). The post mapping threshold is analyzed (line 4 and line 5) and depending on the availability of machines, the machine with the highest memory efficiency is selected (line 5 to 7).The computational complexity of BM is also equivalent to the complexity of the post mapping policy that is $O(|\mathcal{M}||e(R_M(t_i) \cup \{r\})|)$.

```
   BM(r, M, ê)
 1  result ← null
 2  max ← −∞
 3  for each M ∈ M do
 4     e ← e(R_M(t_i) ∪ {r})
 5     if e ≥ ê then
 6        e_h ← e_h(R_M(t_i) ∪ {r})
 7        if e_h > max then
 8           result ← M
 9           max ← e_h
10        end if
11     end if
12  end for
13  return result
```

Figure 4.26: Pseudocode for BM.

4.3.5 Least Deadline Missed

When mapping a request to a machine, it is not only important to meet the deadline of the request that is going to be mapped but also the deadline of all executing requests on that machine. The LDM policy performs post mapping analysis of the deadlines associated with each request and selects a machine that has the least average tardiness considering only tardy requests defined by Eq. 14. Because there is a deadline associated with each WFG, the deadline of each request is obtained by distributing the WFG deadline proportionally to the request level.

$$\bar{\tau}_{R_M(t_i)} = \frac{\sum\limits_{\substack{r \in R_M(t_i) \\ \tilde{f}_r > d_r}} \left(\tilde{f}_r - d_r \right)}{|R_M^{tardy}(t_i)|}. \tag{14}$$

where $R_M^{tardy}(t_i) = \left\{ r | r \in R_M(t_i) \& \tilde{f}_r > d_r \right\}$

$$\tilde{f}_r(t_i) = t_i + \left\{ \frac{C_r}{U_r} \frac{1}{\bar{e}} \right\}. \tag{15}$$

where, $\bar{e}$ is the cluster efficiency calculated as follow:

$$\bar{e} = \frac{\sum\limits_{M \in \mathcal{M}} e}{|\mathcal{M}|}. \tag{16}$$

Fig. 4.27 illustrates the effect of mapping a request r on to two quad core machines using LDM. Though Machine 1 and Machine 2 in the figure have equal pre and post mapping efficiencies, the least average tardiness considering only tardy requests in Machine 2 is less than in Machine 1. Hence this policy selects Machine 2.

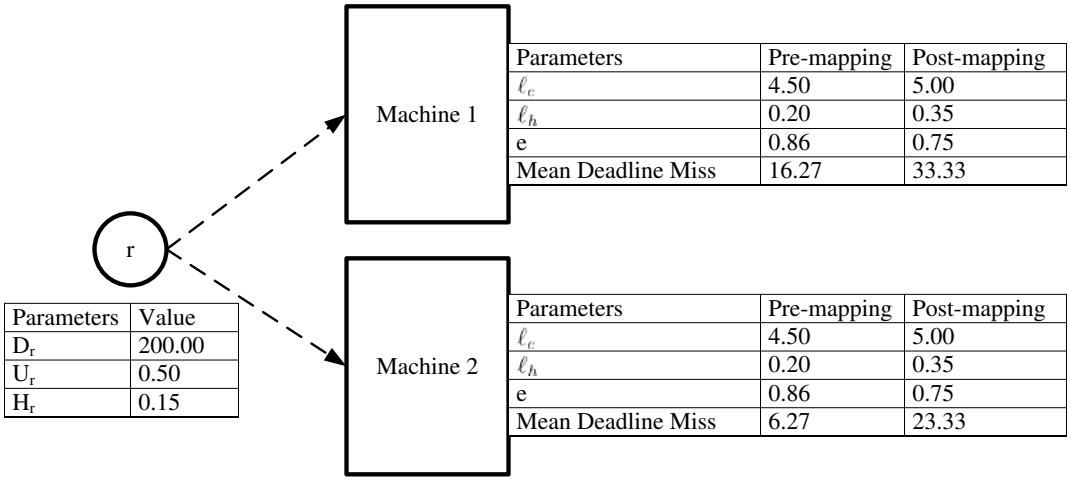

Figure 4.27: Mapping of a request r on to two quad-core machines using LDM.

The pseudocode for the LWR implementation is presented in Fig. 4.28. This policy iterates through all the machines in cluster $\mathcal{M}$ (line 2) and considers only those machines that pass the defined threshold value (line 3 and line 4). In the figure,

$R_M(t_i)$ represents all the requests running on the machine M at time instant t_i. The average tardiness considering only tardy requests is calculated (line 7 to line 16) and then the machine with the least average tardiness is selected (line 17 to 20). Finally the machine is returned to the Assigner for the purpose of mapping. As there are $|R_M(t_i)|$ running requests in the machine M, the computational complexity of LDM is $O(|\mathcal{M}||R_M(t_i)|)$.

LDM$(r, \mathcal{M}, \hat{e})$

```
 1  result ← NULL, min ← −∞
 2  for each M ∈ M do
 3     e ← e(R_M(t_i) ∪ {r})
 4     if e ≥ ê then
 5        meanDeadlineMiss ← 0.0
 6        missedDeadlineCount ← 0
 7        for each r ∈ R_M(t_i) do
 8           τ_r ← f̃_r − d_r
 9           if τ_w(r) > 0 then
10              meanDeadlineMiss+ ← τ_r
11              missedDeadlineCount+ ← 1
12           end if
13        end for
14        if missedDeadlineCount > 0 then
15           meanDeadlineMiss ← meanDeadlineMiss/missedDeadlineCount
16        end if
17        if meanDeadlineMiss < min then
18           min ← meanDeadlineMiss
19           result ← M
20        end if
21     end if
22  end for
23  return result
```

Figure 4.28: Pseudocode for LDM.

4.3.6 Least Work Remaining

In some situations, the loading or efficiency information of the machines is not sufficient for selecting the best available machines. The LWR policy maps a request to a machine which has the least average work remaining. The average work remaining

can be defined by Eq. 17 where, $R_M(t_i)$ represents all the requests running on the machine M at time instant t_i and $\tilde{f}_r(t_i)$ represents the estimated finish time of request r on that machine which is calculated by using Eq. 12. The right hand side of Eq. 17 provides the average estimated finish time of all executing requests for machine M, which corresponds to the average work remaining on that machine.

$$\bar{\tilde{f}}_{R_M(t_i)} = \frac{\sum_{r \in R_M(t_i)} \tilde{f}_r(t_i)}{|R_M(t_i)|}. \tag{17}$$

Consider a scenario as in Fig. 4.29 where a request r needs to be mapped on any one of the two quad core machines. Machine 1 has long requests running which is shown by average work remaining of 1030.92 and 1105.26 for pre and post mapping respectively. In contrast, Machine 2 has short requests running and the average work remaining for pre and post mapping are 51.54 and 82.47 respectively. In the case that the request to be mapped has a longer duration than those executing, whenever those executing requests finish, the efficiency of the machine would increase, thereby executing the mapped request more efficiently. Hence this policy maps the request r to Machine 2 in Fig. 4.29. Fig. 4.30 shows that even though Machine 2 has lower efficiency than Machine 1, LWR policy maps the request r to Machine 2.

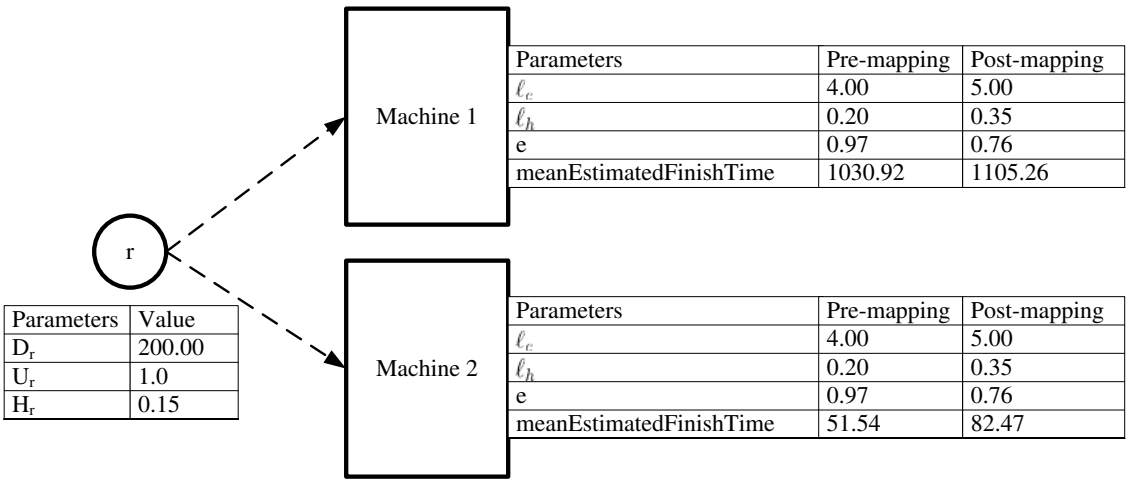

Figure 4.29: Mapping of a request r on to two quad-core machines using LWR.

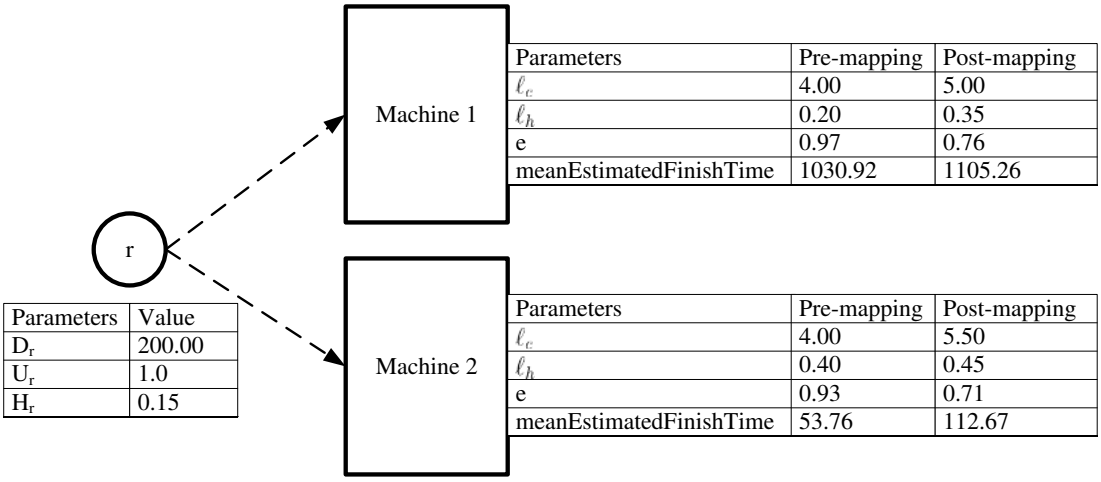

Figure 4.30: Mapping of a request r on to two quad-core machines using LWR.

The pseudocode for the LWR implementation is presented in Fig. 4.31. Similar to LDM policy, this policy also iterates through all the machines in cluster $\mathcal{M}$ (line 2) and considers only those machines that has post mapping efficiency $e(R_M(t_i) \cup \{r\})$ greater than the defined threshold value $\hat{e}$ (line 3 and line 4). In the figure, $R_M(t_i)$ represents all the requests running on the machine M at time instant t_i. The average estimated finish time of all the requests running on machine M is calculated (line 6 to line 9) and then the machine with the least average estimated finish time is selected (line 10 to 11). Finally the machine is returned to the Assigner for the purpose of mapping.As there are $|R_M(t_i)|$ running requests in the machine M, the computational complexity of LWR is $O(|\mathcal{M}||R_M(t_i)|)$.

```
LWR(r, M, ê)
 1  result ← NULL, min ← ∞
 2  for each M ∈ M do
 3     e ← e(R_M(t_i) ∪ {r})
 4     if e ≥ ê then
 5        estimatedFinishTime ← 0
 6        for each r ∈ R_M(t_i) do
 7           estimatedFinishTime+ ← f̃_r(t_i)
 8        end for
 9        meanEstimatedFinishTime ← estimatedFinishTime/|R_M(t_i)|
10        if meanEstimatedFinishTime < min then
11           min ← meanEstimatedFinishTime
12           result ← M
13        end if
14     end if
15  end for
16  return result
```

Figure 4.31: Pseudocode for LWR.

4.4 Concluding Remarks

In this chapter presented and analyzed in depth the seven RSPs and six MSPs. It started with defining each policies with example followed by complexity analysis. Out of seven RSPs, three new RSP are introduced, namely, PLLF, SWF and DSWF. Six new machine selection heuristics are also presented. The performance analysis of each of these policies are performed in Chapter 5.

Chapter 5

Simulation Studies

In this chapter, the performance of the forty-two combinations of Request Selection Policies (RSPs) and Machine Selection Policies (MSPs) are evaluated. The following sections describe the experimental setup followed by the results.

5.1 Experimental Setup

In this thesis, four WFG generation scenarios are considered, each representing a one day period. The experimental setup for these four different cases are described in following sections.

5.1.1 Case Study One

In Case Study One, a single day period is divided into three consecutive epochs. These three epochs are associated with WFG generation characteristics for a typical operational business day. The first epoch is from time = 0 to time = 11 hours; the second epoch is from time = 11 to time = 12 hours; and the third epoch is from time = 12 to time = 24 hours (refer to Fig. 5.1). During the first and third epochs, only Interactive and Webservice WFGs are generated. During the second epoch, all three

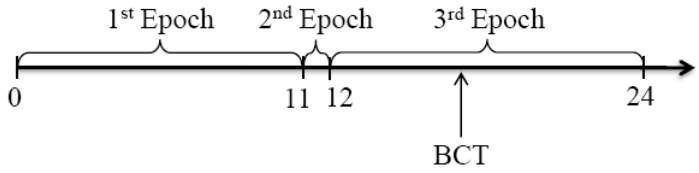

Figure 5.1: Time-line illustrating the three epochs.

types of WFGs are generated. The first and third epochs represent periods of time before and after a relatively short epoch in which Batch WFGs arrive. The start and end times of the second epoch are defined by terms of service-level agreements (SLAs) [12] related to timing of Batch WFG submission and execution. Typical terms of SLAs specify that daily Batch WFGs submitted within a specified time period will be completed by an agreed upon deadline.

Although Batch WFGs only arrive during the second epoch (between hour 11 and hour 12), a typical Batch WFG can take several hours to complete execution. Thus, an important milestone that is measured in the simulation studies is the point in time at which all Batch WFGs have completed execution, which is defined as the Batch Completion Time (BCT). Fig. 5.1 illustrates the three time epochs and the BCT on a time-line.

The parameter value ranges and distributions associated with the simulation studies are summarized in Table 5.1. The table defines parameters related to the structural characteristics for each type of WFG, which are all assumed to be two-levels deep. Also provided in the table are CPU and memory characteristics of the requests associated with each WFG type. The parallelization factor is needed in determining a base deadline for each generated WFG; it defines the degree of parallelism assumed for executing parallel RCs associated with a common compound node. Once a base deadline is determined for a WFG, it is multiplied by the Deadline Factor (last row in the table) to define the actual deadline for the WFG.

Table 5.1: Parameter values for Case Study One.

Parameter	**Interactive WFG**	**Webservice WFG**	**Batch WFG**
*Avg. Inter-Arrival Time (secs)	60	120	60
+Compound Nodes	[1, 1]	[1, 3]	[3, 5]
+Parallel RCs	[1, 2]	[2, 3]	[5, 20]
+Requests in RCs	[5, 8]	[5, 8]	[3, 8]
+Request Ideal Duration (secs), D_r	[1.0, 2.5]	[10.0, 50.0]	[50.0, 175.0]
+Request CPU Utilization, U_r	[0.5, 1.0]	[0.5, 1.0]	[0.5, 1.0]
+Request Heap Memory, H_r	[0.05, 0.15]	[0.05, 0.10]	[0.05, 0.10]
Parallelization Factor	2	2	2
+WFG Deadline Factor	[1.1, 1.2]	[1.3, 1.5]	[1.3, 1.5]

*Poisson process. +Uniform distribution [Min, Max].

Interarrival times of the Interactive and Webservice WFGs in Case Study One are 60 seconds and 120 seconds respectively. The interarrival time of the Batch WFGs is assumed to be 60 seconds during the second time epoch from hour 11 to hour 12; Batch WFGs do not arrive outside this one-hour interval. An example arrival count realization of the arrivals of WFGs into the system are shown in Fig 5.2.

5.1.2 Case Study Two

The request arrival rate in Case Study Two is faster than in Case Study One. Here, the Interactive and Webservice requests arrival rate are six time faster than in Case Study One. The parameter value ranges and distributions associated with the simulation studies are summarized in Table 5.2.

Interarrival times of the Interactive and Webservice WFGs in Case Study Two are 10 seconds and 20 seconds respectively. The interarrival time of the Batch WFGs is the same as in Case Study One. An example arrival count realization of the arrivals of WFGs into the system is shown in the Fig 5.3.

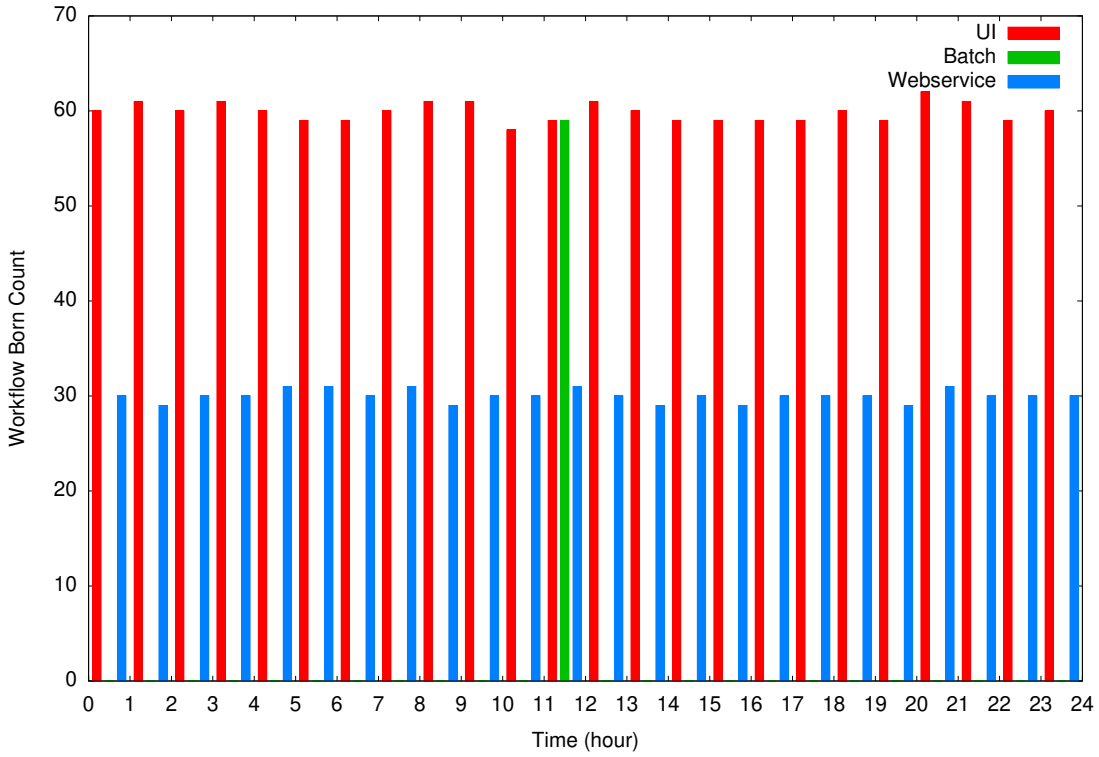

Figure 5.2: Arrival count realization for Case Study One.

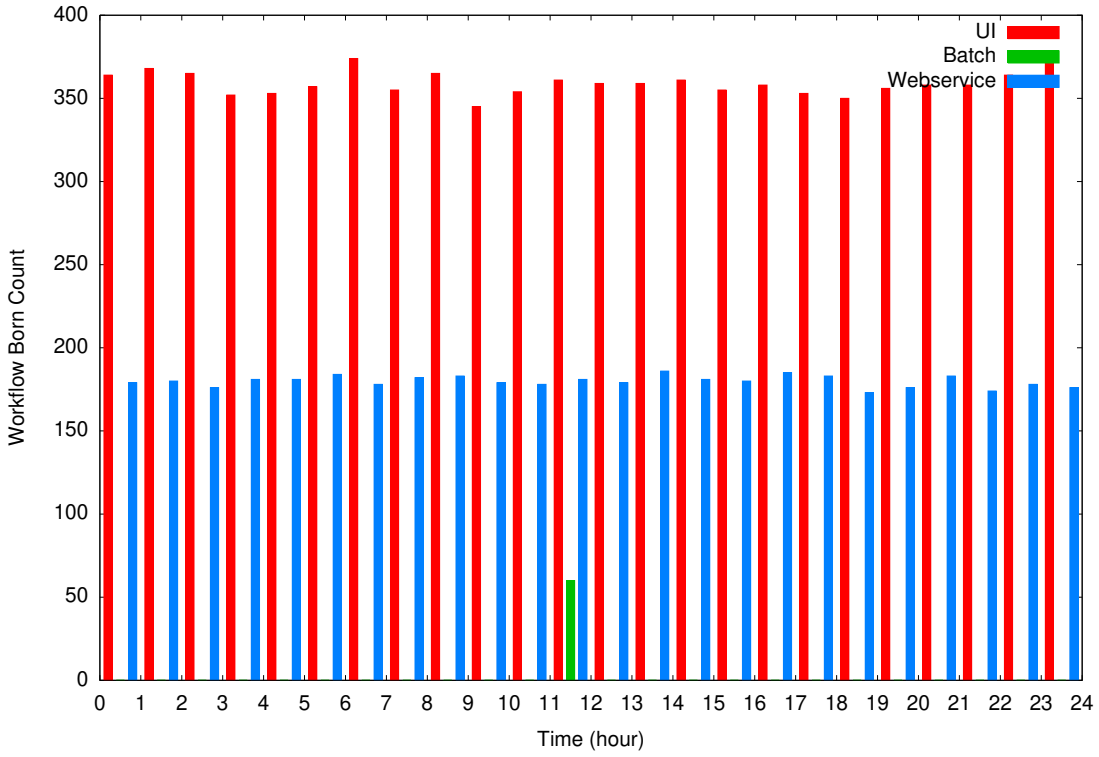

Figure 5.3: Arrival count realization for Case Study Two.

Table 5.2: Parameter values for Case Study Two.

Parameter	**Interactive WFG**	**Webservice WFG**	**Batch WFG**
*Avg. Inter-Arrival Time (secs)	10	20	60
+Compound Nodes	[1, 1]	[1, 3]	[3, 5]
+Parallel RCs	[1, 2]	[2, 3]	[5, 20]
+Requests in RCs	[5, 8]	[5, 8]	[3, 8]
+Request Ideal Duration (secs), D_r	[1.0, 2.5]	[10.0, 50.0]	[50.0, 175.0]
+Request CPU Utilization, U_r	[0.5, 1.0]	[0.5, 1.0]	[0.5, 1.0]
+Request Heap Memory, H_r	[0.05, 0.15]	[0.05, 0.10]	[0.05, 0.10]
Parallelization Factor	2	2	2
+WFG Deadline Factor	[1.1, 1.2]	[1.3, 1.5]	[1.3, 1.5]

*Poisson process. +Uniform distribution [Min, Max].

5.1.3 Case Study Three

The request arrival pattern in Case Study Three is similar to Case Study Two but the parallelization factor is increased for Batch WFGs and decreased for Interactive WFGs. The parameter value ranges and distributions associated with the simulation studies are summarized in Table 5.3.

Table 5.3: Parameter values for Case Study Three.

Parameter	**Interactive WFG**	**Webservice WFG**	**Batch WFG**
*Avg. Inter-Arrival Time (secs)	10	20	60
+Compound Nodes	[1, 1]	[1, 3]	[3, 5]
+Parallel RCs	[1, 2]	[2, 3]	[5, 20]
+Requests in RCs	[5, 8]	[5, 8]	[3, 8]
+Request Ideal Duration (secs), D_r	[1.0, 2.5]	[10.0, 50.0]	[50.0, 175.0]
+Request CPU Utilization, U_r	[0.5, 1.0]	[0.5, 1.0]	[0.5, 1.0]
+Request Heap Memory, H_r	[0.05, 0.15]	[0.05, 0.10]	[0.05, 0.10]
Parallelization Factor	1	2	5
+WFG Deadline Factor	[1.1, 1.2]	[1.3, 1.5]	[1.3, 1.5]

*Poisson process. +Uniform distribution [Min, Max].

5.1.4 Case Study Four

In Case Study Four, the arrival pattern is more realistic where Interactive and Webservice WFGs have variable arrival rates, whereas Batch WFGs has an exponentially decaying arrival rate. An example arrival count realization of the arrivals of WFGs into the system is shown in the Fig 5.4. The parameter value ranges and distributions associated with Case Study Four are summarized in Table 5.4.

Table 5.4: Parameter values for Case Study Four.

Parameter	**Interactive WFG**	**Webservice WFG**	**Batch WFG**
$^{+}$Compound Nodes	[1, 1]	[1, 3]	[3, 5]
$^{+}$Parallel RCs	[1, 2]	[2, 3]	[5, 20]
$^{+}$Requests in RCs	[5, 8]	[5, 8]	[3, 8]
$^{+}$Request Ideal Duration (secs), D_r	[1.0, 2.5]	[10.0, 50.0]	[50.0, 175.0]
$^{+}$Request CPU Utilization, U_r	[0.5, 1.0]	[0.5, 1.0]	[0.5, 1.0]
$^{+}$Request Heap Memory, H_r	[0.05, 0.15]	[0.05, 0.10]	[0.05, 0.10]
Parallelization Factor	1	2	5
$^{+}$WFG Deadline Factor	[1.1, 1.2]	[1.3, 1.5]	[1.3, 1.5]

*Poisson process. $^{+}$Uniform distribution [Min, Max].

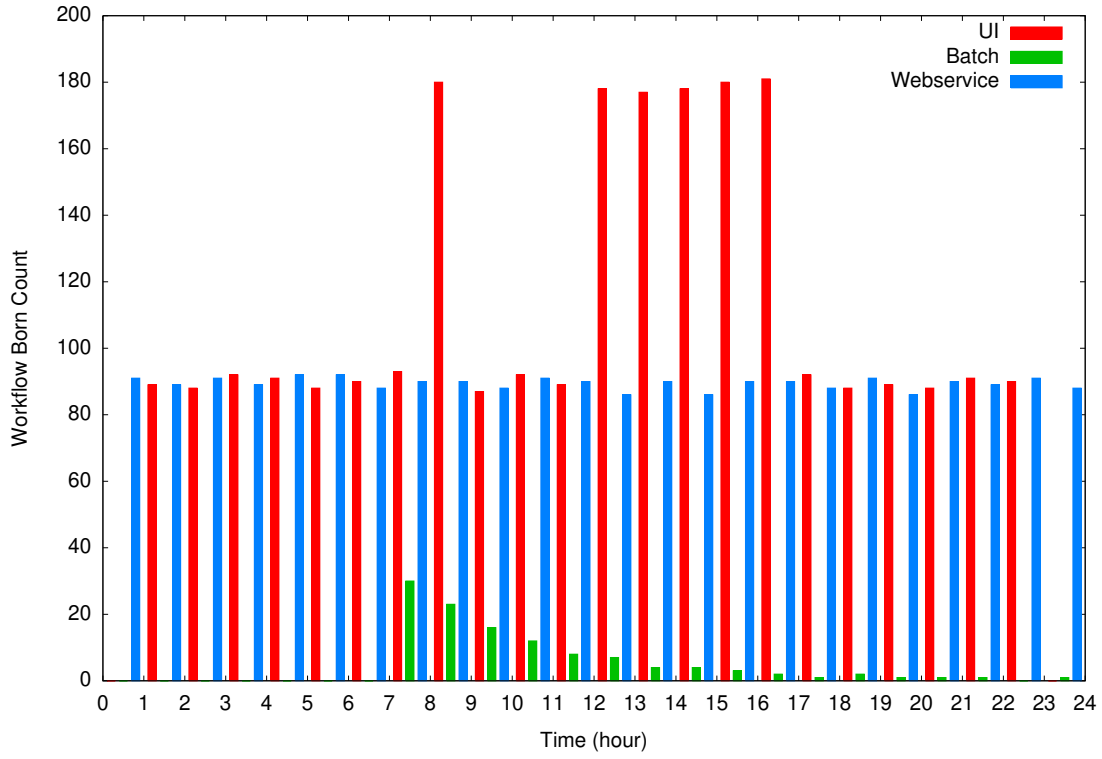

Figure 5.4: Arrival count realization for Case Study Four.

5.2 Results

A total of forty-two combinations of scheduling approaches are considered for the four case studies described in the previous section. For the purpose of evaluating scheduling policies, two cost functions are used, namely, sigmoid cost function and quadratic. Fig. 5.5 and Fig. 5.6 show the reference sigmoid and quadratic cost functions. Let $\mathcal{W}$ be the set of all WFGs and the cost for each WFG w is based on a normalized measure of tardiness τ_w to a cost value. The total cost of the system, denoted by $\mathcal{F}_w(\tau_w)$, is defined by summing the costs of all WFGs in Eq. 18.

$$\mathcal{F}(\tau) = \sum_{w \in \mathcal{W}} \mathcal{F}_w(\tau_w). \tag{18}$$

where $\tau = [\tau_w]_{w \in \mathcal{W}}$.

The normalized tardiness of a WFG w is defined by the following equation:

$$\tau_w = \frac{f_w - d_w}{d_w - b_w}. \tag{19}$$

The numerator of the expression $f_w - d_w$ represents the actual tardiness of w. The denominator of the expression, $d_w - b_w$, represents the maximum desired amount of time allocated for executing w, and is by definition positive. The numerator can be either positive or negative. Thus, $t_w \leq 0$ indicates that w is not tardy and $t_w > 0$ indicates w is tardy.

Because τ_w is normalized, it is straightforward to compare the relative tardiness values of WFGs of different sizes and/or expected durations. For instance, an actual tardiness of $f_w - d_w = 10$ seconds is relatively insignificant if the overall allocated duration is $d_w - b_w = 1$ hour, i.e., $\tau_w = \frac{10}{3600} = 0.0028$. However, a tardiness of 10 seconds could be quite significant if the overall allocated duration is defined to be 40 seconds, i.e., $\tau_w = \frac{10}{40} = 0.25$.

Fig. 5.5 shows the sigmoid function used in this thesis. In the figure, for the normalized tardiness in the range of $-1 \leq \tau_w < 0$, there is relatively low cost that is shown in Fig. 5.5 as ϵ. The cost in the range of $0 < \tau_w \leq \alpha$ and $\alpha < \tau_w \leq \beta$ increase according to the quadratic function and after this point the cost increases very slowly up to the value of γ.

In case of the quadratic cost function, the cost increases as the normalized tardiness increases. The maximum quadratic cost is not bounded as shown in Fig. 5.6.

The following subsection provides the results of the threshold analysis, scheduling policies' performances and resource analysis.

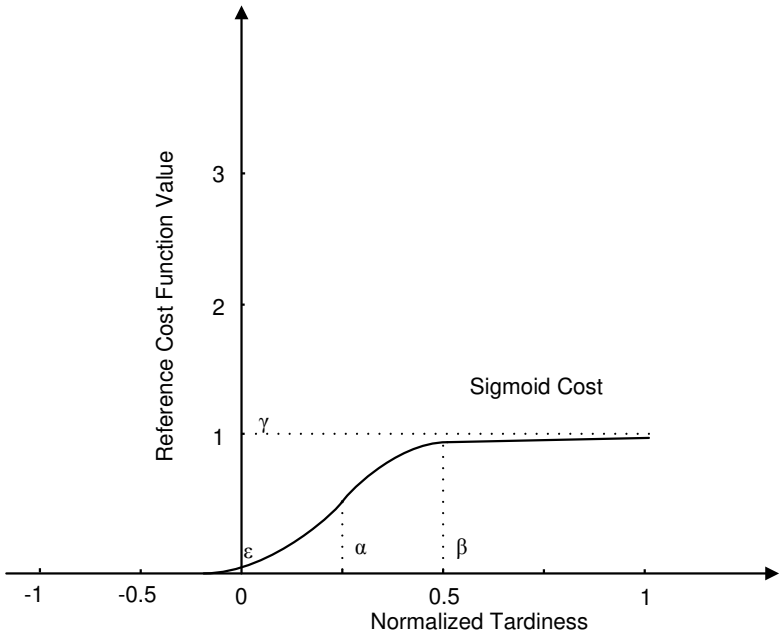

Figure 5.5: The sigmoid cost function.

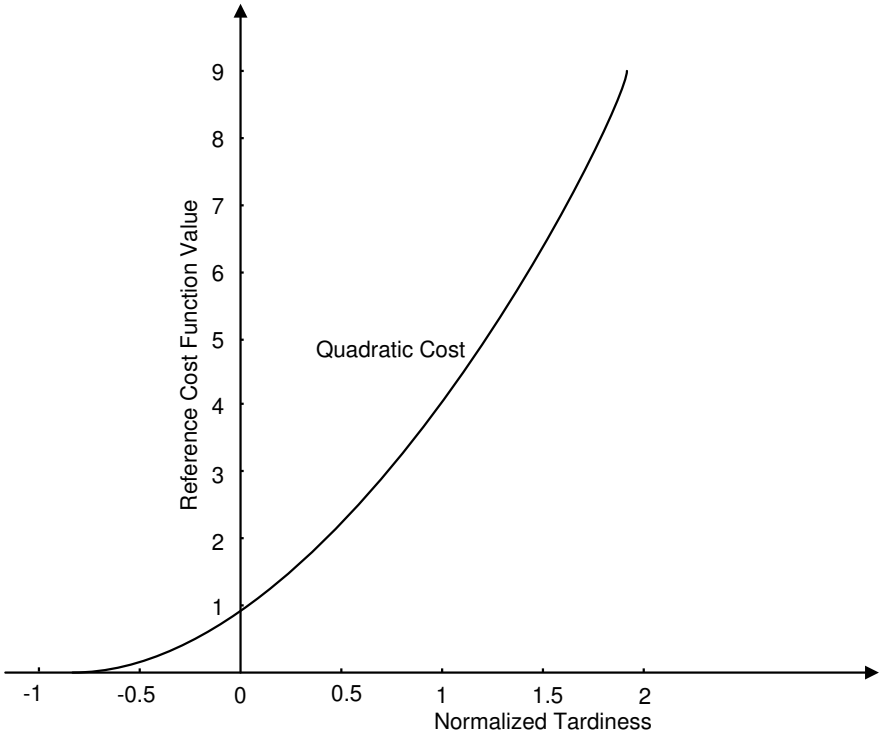

Figure 5.6: The quadratic cost function.

5.2.1 Threshold Analysis

Every machine in the assumed cluster has four cores and automatic memory management. Each machine can process multiple requests at a time but as the loading of a machine increases, efficiency of machine decreases. Given a scenario with request characteristics, there exists an optimal machine efficiency referred to as machine threshold and denoted as $\hat{e}$. Since the machine can process requests according to its capacity, the concept of thresholding a machine is very important. Depending upon the defined threshold values, a machine is available if the machine efficiency is greater than or equal to $\hat{e}$. Two possible ways of thresholding a machine are: pre-mapping threshold and post-mapping threshold. In the case of a pre-mapping threshold the machines are considered to be available if the efficiency of machine is greater than or equal to the $\hat{e}$ before mapping. In contrast, in the case of a post-mapping threshold, the machines are considered to be available if the efficiency of machine is greater than or equal to the $\hat{e}$ after ("what if") mapping.

Defining a good threshold value is dependent on various parameters, including: scheduling policy, requests arrival pattern, and requests characteristics. Due to these reasons, threshold analysis of each combinations are performed. An iterative approach is used to define a good threshold value for each policies. For each policy, experiments are performed for an efficiency threshold value of 0.05 to 0.95 and an optimal efficiency value is identified. Fig. 5.7 shows the cumulative sigmoid cost for efficiency values of 0.30 to 0.95. It shows that the efficiency value of 0.90 results in the lowest cost compared with other efficiency values for PLLF in Case Study One. The corresponding percentage of tardy workflows based on the normalized tardiness is shown in Fig. 5.8. The percentage of tardy workflows is minimized for the efficiency threshold of 0.90. Similarly, Fig. 5.9 and Fig. 5.10 shows the cumulative sigmoid cost and percentage of tardy workflows for PLLF in Case Study Four. Table 5.5 shows the optimal threshold results of all the combinations of RSPs and MSPs.

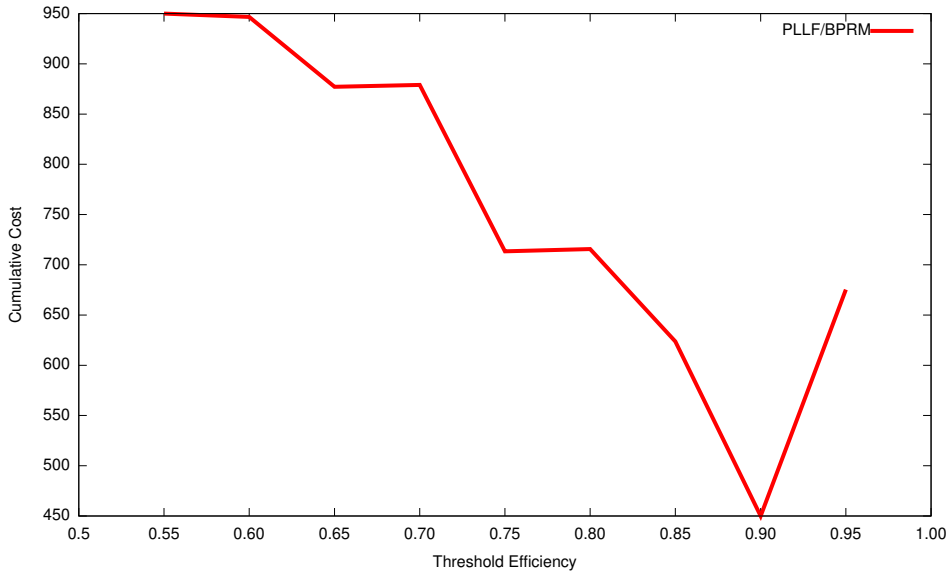

Figure 5.7: The cumulative cost of all WFGs by efficiency threshold for PLLF assuming sigmoid cost in Case Study One.

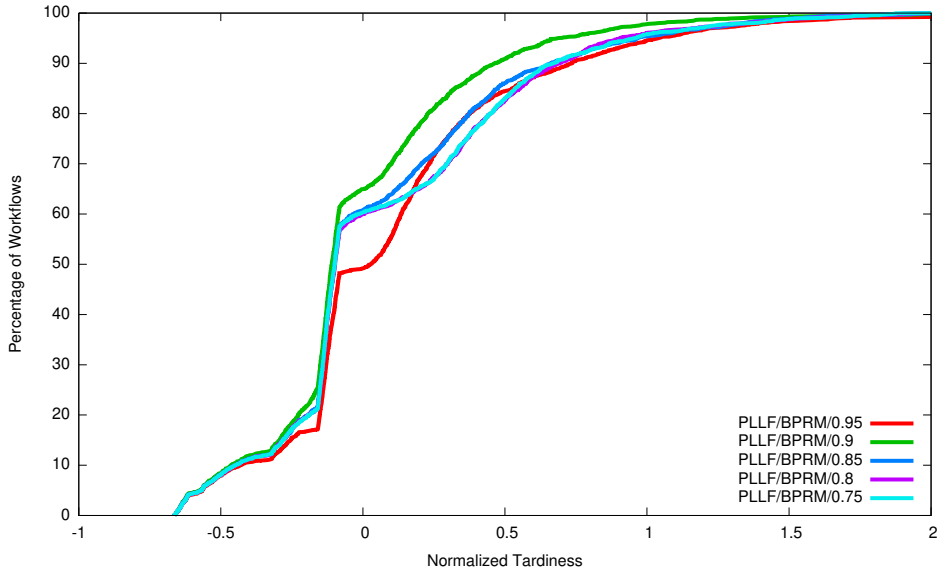

Figure 5.8: Percentage of workflows as a function of normalized tardiness for PLLF in Case Study One.

5.2.2 Scheduling Policies Analysis

In this section, performance analysis of all the RSPs and MSPs are performed based on the threshold values obtained from the previous subsection. There are various measures that can be used for the analysis of each scheduling policy. Performance

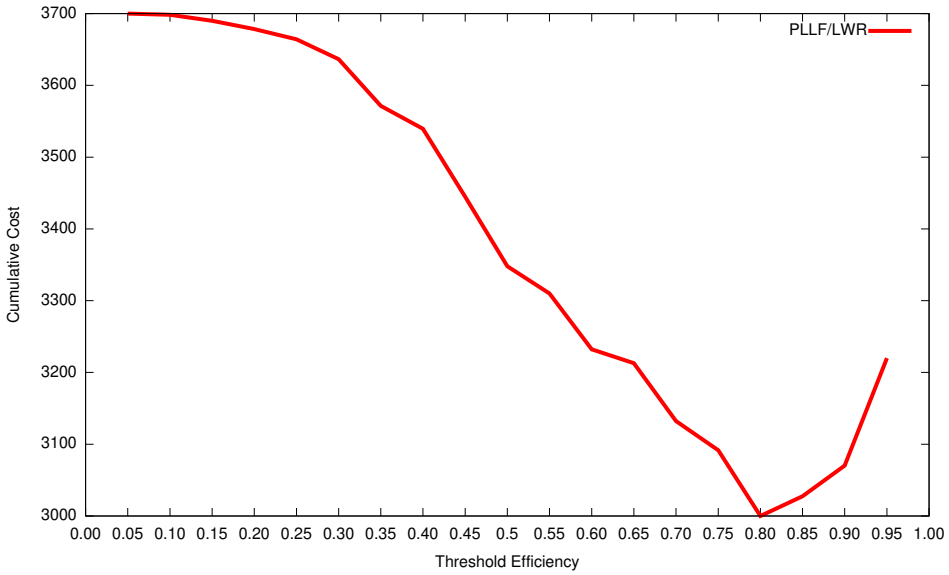

Figure 5.9: The cumulative cost of all WFGs by efficiency threshold for PLLF assuming sigmoid cost in Case Study Four.

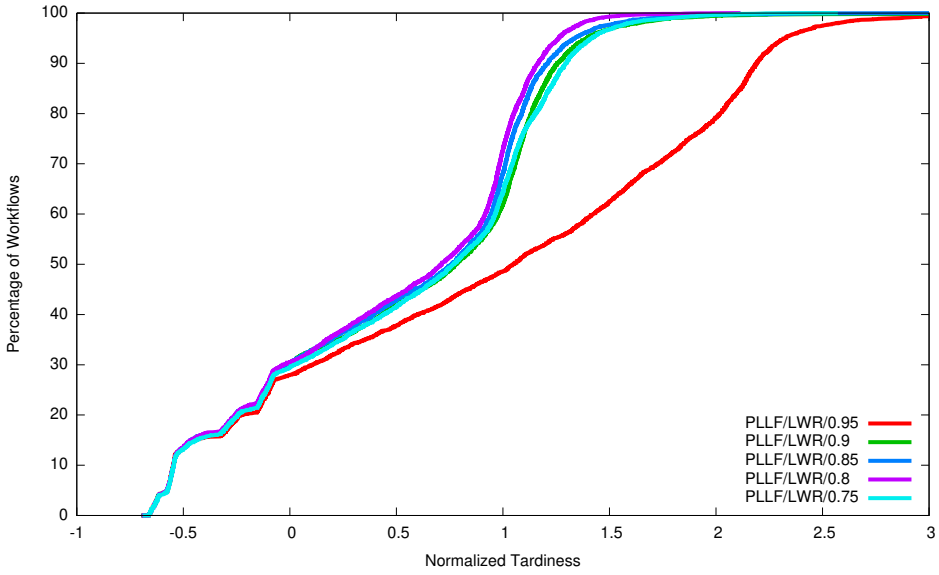

Figure 5.10: Percentage of workflows as a function of normalized tardiness for PLLF in Case Study Four .

analysis of these scheduling policies is based on Batch Completion Time (BCT), sigmoid and quadratic cumulative cost, normalized tardiness, and percentage of late workflows. Note that the results obtained in the following section are the average of multiple runs with different seed values.

Table 5.5: Optimal Threshold Table.

Cases	Policies	BPRM	BPOM	BC	BM	LDM	LWR
Case 1	FCFS	0.9	0.75	0.75	0.75	0.8	0.75
	FCLS	0.95	0.9	0.9	0.9	0.9	0.9
	EDF	0.95	0.9	0.9	0.9	0.9	0.9
	LLF	0.9	0.9	0.9	0.9	0.9	0.9
	PLLF	0.9	0.9	0.9	0.9	0.9	0.9
	SWF	0.95	0.9	0.9	0.9	0.9	0.9
	DSWF	0.95	0.9	0.9	0.9	0.9	0.9
Case 2	FCFS	0.65	0.8	0.8	0.8	0.8	0.8
	FCLS	0.95	0.9	0.9	0.9	0.9	0.9
	EDF	0.9	0.9	0.9	0.9	0.9	0.9
	LLF	0.9	0.8	0.8	0.8	0.8	0.8
	PLLF	0.9	0.85	0.8	0.85	0.85	0.85
	SWF	0.95	0.95	0.95	0.95	0.95	0.95
	DSWF	0.95	0.95	0.95	0.95	0.95	0.95
Case 3	FCFS	0.6	0.8	0.8	0.8	0.8	0.8
	FCLS	0.95	0.9	0.9	0.9	0.9	0.9
	EDF	0.9	0.8	0.8	0.8	0.8	0.8
	LLF	0.9	0.75	0.8	0.75	0.75	0.75
	PLLF	0.85	0.75	0.75	0.8	0.8	0.75
	SWF	0.95	0.95	0.95	0.95	0.95	0.9
	DSWF	0.95	0.95	0.95	0.95	0.95	0.95
Case 4	FCFS	0.9	0.75	0.8	0.75	0.75	0.75
	FCLS	0.95	0.8	0.8	0.8	0.8	0.8
	EDF	0.9	0.8	0.8	0.8	0.8	0.8
	LLF	0.9	0.8	0.8	0.8	0.9	0.8
	PLLF	0.9	0.8	0.8	0.8	0.8	0.8
	SWF	0.95	0.8	0.8	0.8	0.8	0.8
	DSWF	0.95	0.8	0.8	0.8	0.8	0.8

5.2.2.1 Case Study One

The results for the simulation study of Case Study One are summarized in Table 5.6. The table shows the cross product of RSPs and MSPs. For each combination, the optimum threshold value is used from the threshold analysis (refer to Table 5.5). BCT is measured in hours. Cumulative and Max normalized tardiness values are listed in the Table 5.6 along with 50th percentile, 95th percentile and 99th percentile normalized tardiness. Positive values of the tardiness correspond to deadlines being missed by that amount; negative values represent deadlines being met. The Cumulative normalized tardiness in the table is the sum of tardiness for both tardy and non tardy WFGs. Based on the normalized tardiness values, two cost analysis approaches are considered: sigmoid cost and quadratic cost.

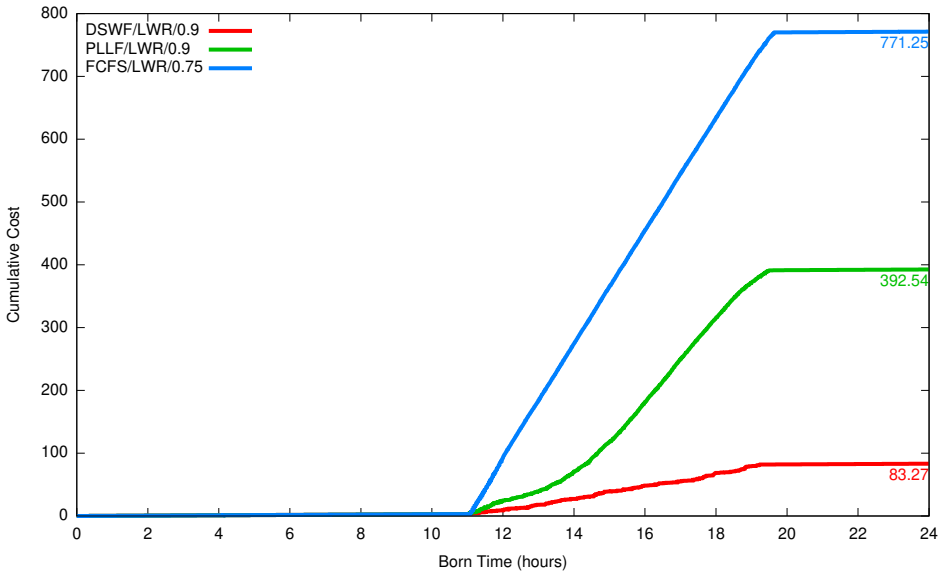

Figure 5.11: The cumulative running cost of all WFGs by born time for DSWF, PLLF and FCFS assuming sigmoid cost in Case Study One.

In Case Study One the performance of DSWF, EDF and SWF are very close to each other. They are approximately six times better than FCFS and three times better than LLF and PLLF. Fig. 5.11 shows that the DSWF significantly minimizes the sigmoid cost compared to PLLF and FCFS. The cost associated with DSWF is approximately six times less than FCFS and three times less than PLLF. The corresponding tardiness graph is shown in Fig. 5.12. Note that in the figure, PLLF minimizes the maximum tardiness.

From Table 5.6, it is evident that post-mapping policies (BPOM, BC, BM, LDM, and LWR) perform better than pre-mapping policies for FCLS, EDF, PLLF, SWF and DSWF. Whereas, the performance of FCFS remains the same for all MSPs. In contrast to all the other RSPs, LLF has smaller sigmoid cost for the pre-mapping policy BPRM, but the percent of late workflow increases compared to the other MSPs. The performance of MSP algorithms depend on the corresponding RSP as well as the threshold. Since Case Study One does not have high traffic, the following section analyzes the high load scenarios.

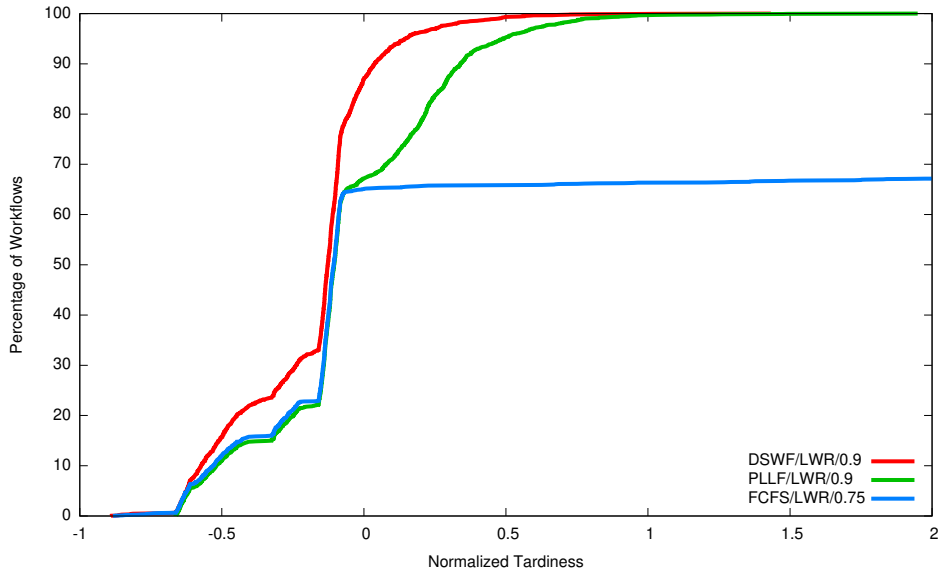

Figure 5.12: Percentage of workflows as a function of normalized tardiness for DSWF, PLLF and FCFS in Case Study One.

5.2.2.2 Case Study Two

The results for the simulation study of Case Study Two are summarized in Table 5.7. Case Study Two is more loaded than Case Study One where Interactive and Webservice WFGs have approximately six times faster arrival rates.

Fig. 5.13 shows that the DSWF significantly minimizes the Sigmoid cost compared to PLLF and FCFS. The cost associated with DSWF is approximately nine times less than FCFS and PLLF. The corresponding tardiness graph is shown in Fig. 5.18. Note that in Fig. 5.18, PLLF performs better than the other policies in terms of minimizing the maximum tardiness, but it makes lots of WFGs tardy there by resulting in high cumulative sigmoid cost.

In Case Study Two, the performance of post-mapping MSPs are always better than pre-mapping MSPs for all RSPs. In case of SWF and DSWF, post mapping MSPs are approximately two times better than pre-mapping policy BPRM. Where as, in case of FCFS, EDF, LLF and PLLF the post mapping policies are only slightly better than pre-mapping policy. In case of FCLS, post mapping policies are about

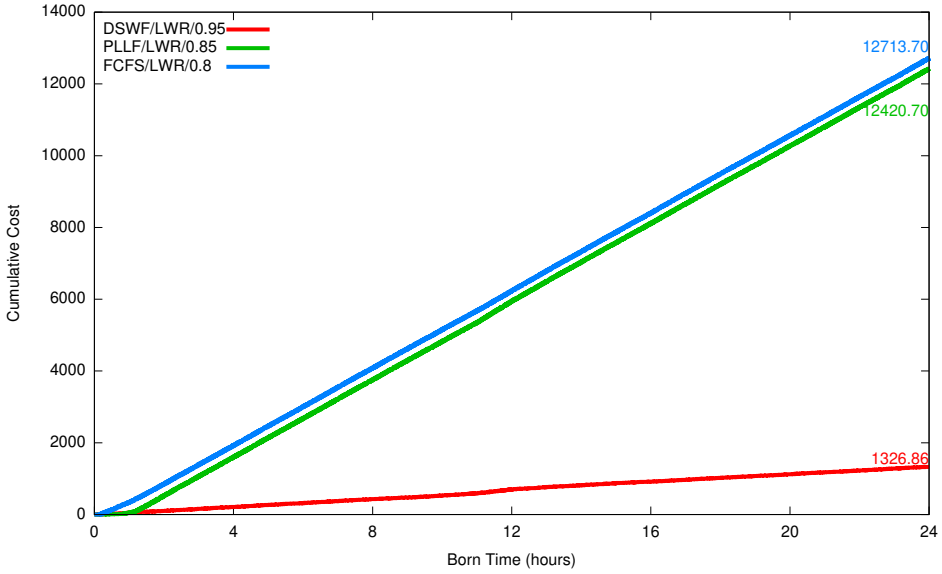

Figure 5.13: The cumulative running cost of all WFGs by born time for DSWF, PLLF and FCFS assuming sigmoid cost in Case Study Two.

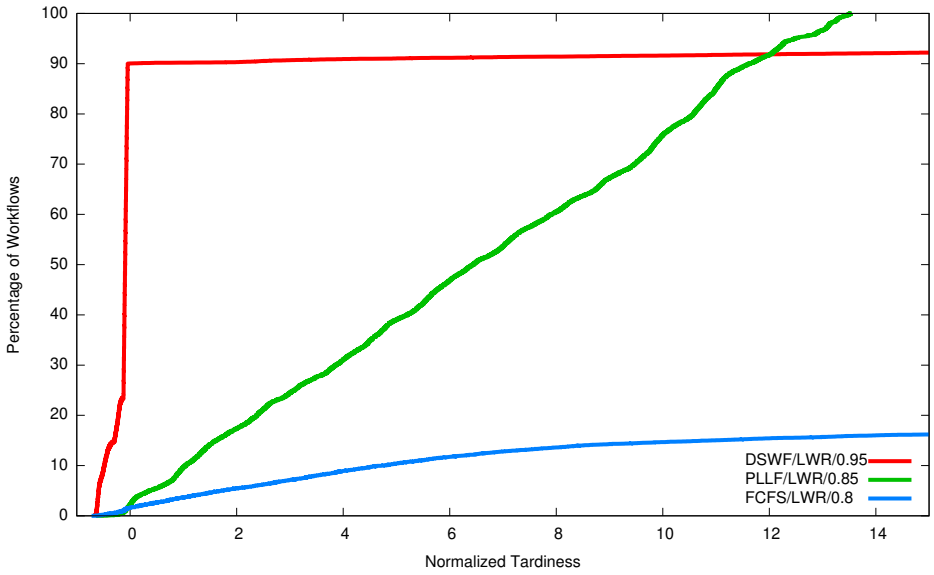

Figure 5.14: Percentage of workflows as a function of normalized tardiness for DSWF, PLLF and FCFS in Case Study Two.

Table 5.6: Statistics for simulation of Case Study One.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.90	20.71	902	9.54E+08	5.94E+05	2187.3	1443.8	1876.0	41	42	16	41
FCFS	BPOM	0.75	20.67	902	9.72E+08	5.93E+05	2183.5	1442.3	1875.4	41	42	16	41
FCFS	BC	0.75	20.70	905	9.68E+08	5.94E+05	2184.1	1443.4	1877.7	46	50	16	41
FCFS	BM	0.75	20.70	901	9.67E+08	5.93E+05	2180.3	1438.9	1877.9	41	42	16	41
FCFS	LDM	0.80	20.68	906	9.68E+08	5.95E+05	2175.2	1444.6	1874.7	47	51	15	41
FCFS	LWR	0.75	20.67	902	9.56E+08	5.93E+05	2176.5	1441.4	1874.5	41	42	16	41
FCLS	BPRM	0.95	23.10	234	6.10E+04	5.14E+02	108.3	0.5	8.7	23	33	36	4
FCLS	BPOM	0.90	22.32	212	4.13E+04	3.80E+02	97.1	0.5	5.7	20	28	31	4
FCLS	BC	0.90	22.36	204	4.11E+04	4.08E+02	97.5	0.4	6.2	21	29	32	4
FCLS	BM	0.90	22.32	207	4.05E+04	3.54E+02	96.8	0.5	5.9	21	29	32	4
FCLS	LDM	0.90	22.33	213	4.09E+04	3.78E+02	98.5	0.5	5.8	21	29	31	4
FCLS	LWR	0.90	22.31	210	3.70E+04	3.19E+02	96.9	0.4	4.6	21	29	31	4
EDF	BPRM	0.95	23.29	184	1.58E+04	8.27E+01	17.3	0.5	8.4	21	31	6	2
EDF	BPOM	0.90	22.44	135	1.70E+03	-3.82E+02	1.7	0.3	0.6	18	27	0	0
EDF	BC	0.90	22.53	141	1.78E+03	-3.40E+02	2.0	0.3	0.6	17	27	0	0
EDF	BM	0.90	22.42	134	1.68E+03	-3.84E+02	1.9	0.3	0.6	17	26	0	0
EDF	LDM	0.90	22.50	135	1.74E+03	-3.66E+02	1.5	0.3	0.6	18	27	0	0
EDF	LWR	0.90	22.46	141	1.72E+03	-3.73E+02	1.7	0.3	0.6	18	27	0	0
LLF	BPRM	0.90	21.52	473	4.37E+06	2.45E+04	149.1	92.2	132.5	32	41	50	15
LLF	BPOM	0.90	22.17	509	6.80E+06	3.50E+04	197.7	128.9	176.7	31	36	58	18
LLF	BC	0.90	22.14	516	6.99E+06	3.61E+04	201.8	131.5	180.1	31	36	58	19
LLF	BM	0.90	22.14	509	6.86E+06	3.51E+04	198.0	129.4	177.6	31	36	58	18
LLF	LDM	0.90	22.13	508	6.91E+06	3.54E+04	198.9	129.8	178.1	31	36	58	18
LLF	LWR	0.90	22.10	512	6.87E+06	3.55E+04	199.2	129.7	177.6	31	36	58	18
PLLF	BPRM	0.90	21.54	473	2.44E+03	-4.90E+01	2.8	0.6	1.0	36	41	56	23
PLLF	BPOM	0.90	22.10	442	2.48E+03	-4.50E+01	3.1	0.6	1.0	37	40	59	27
PLLF	BC	0.90	22.15	436	2.51E+03	8.28E+00	4.4	0.6	1.2	36	40	60	26
PLLF	BM	0.90	22.15	429	2.57E+03	-4.63E+01	3.7	0.6	1.1	36	40	60	26
PLLF	LDM	0.90	22.13	433	2.52E+03	-2.48E+01	3.5	0.6	1.1	37	41	60	27
PLLF	LWR	0.90	22.16	432	2.47E+03	-3.31E+01	4.6	0.6	1.1	36	40	60	27
SWF	BPRM	0.95	23.20	152	1.74E+03	-3.60E+02	1.5	0.3	0.6	20	31	9	0
SWF	BPOM	0.90	22.37	134	1.70E+03	-3.81E+02	2.0	0.3	0.6	17	26	8	0
SWF	BC	0.90	22.43	126	1.75E+03	-3.47E+02	1.6	0.2	0.6	18	27	8	0
SWF	BM	0.90	22.37	137	1.70E+03	-3.79E+02	2.0	0.3	0.6	18	27	8	0
SWF	LDM	0.90	22.41	139	1.73E+03	-3.63E+02	1.6	0.3	0.6	18	27	8	0
SWF	LWR	0.90	22.38	133	1.71E+03	-3.78E+02	1.9	0.3	0.6	18	27	8	0
DSWF	BPRM	0.95	23.23	148	1.74E+03	-3.57E+02	1.7	0.3	0.6	20	31	11	0
DSWF	BPOM	0.90	22.44	136	1.71E+03	-3.75E+02	1.6	0.2	0.6	18	27	7	0
DSWF	BC	0.90	22.35	134	1.77E+03	-3.37E+02	1.9	0.3	0.6	18	27	7	0
DSWF	BM	0.90	22.43	135	1.71E+03	-3.75E+02	1.6	0.2	0.6	18	27	9	0
DSWF	LDM	0.90	22.31	133	1.73E+03	-3.61E+02	1.3	0.3	0.6	18	27	4	0
DSWF	LWR	0.90	22.37	135	1.71E+03	-3.73E+02	1.6	0.3	0.6	18	27	6	0

one and half times better than BPRM in terms of sigmoid cost.

Table 5.7: Statistics for simulation of Case Study Two.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.65	24.50	8440	2.58E+10	1.01E+07	3317.9	2747.4	3106.4	66	67	33	65
FCFS	BPOM	0.80	23.31	8375	2.27E+10	9.11E+06	3083.8	2511.7	2865.7	65	66	26	64
FCFS	BC	0.80	23.34	8438	2.27E+10	9.10E+06	3082.4	2511.0	2864.4	75	81	26	64
FCFS	BM	0.80	23.31	8376	2.27E+10	9.11E+06	3084.6	2512.0	2865.3	65	66	26	64
FCFS	LDM	0.80	23.30	8420	2.27E+10	9.10E+06	3084.4	2510.6	2864.7	70	74	26	64
FCFS	LWR	0.80	23.31	8393	2.27E+10	9.11E+06	3080.0	2512.0	2865.6	71	75	26	64
FCLS	BPRM	0.95	34.67	2745	1.17E+08	7.52E+04	3364.5	19.5	65.0	43	55	80	19
FCLS	BPOM	0.90	33.30	1766	2.88E+07	4.50E+04	1766.4	19.3	56.0	30	35	75	18
FCLS	BC	0.90	33.35	1810	3.27E+07	4.72E+04	1921.8	19.4	56.3	31	38	75	18
FCLS	BM	0.90	33.29	1751	2.21E+07	4.40E+04	1560.0	19.4	55.9	30	36	75	18
FCLS	LDM	0.90	33.30	1761	1.40E+07	4.17E+04	1566.6	19.3	55.8	31	38	75	18
FCLS	LWR	0.90	33.28	1801	2.14E+07	4.43E+04	1429.2	19.3	55.8	31	38	75	18
EDF	BPRM	0.90	32.29	4705	3.06E+09	2.25E+06	1292.3	826.4	1091.2	54	64	60	34
EDF	BPOM	0.90	33.58	4630	3.78E+09	2.57E+06	1462.1	922.8	1237.8	43	47	72	36
EDF	BC	0.90	33.57	4635	3.76E+09	2.55E+06	1454.0	919.2	1231.9	44	49	72	36
EDF	BM	0.90	33.51	4629	3.77E+09	2.56E+06	1458.4	921.0	1235.4	43	47	72	36
EDF	LDM	0.90	33.55	4616	3.76E+09	2.55E+06	1453.0	918.7	1232.4	44	48	72	36
EDF	LWR	0.90	33.52	4628	3.75E+09	2.55E+06	1452.1	917.9	1229.0	44	48	72	36
LLF	BPRM	0.90	31.86	7342	9.55E+09	4.70E+06	2134.7	1593.6	1942.8	63	66	94	56
LLF	BPOM	0.80	31.90	7184	1.04E+10	4.77E+06	2145.3	1602.1	1952.0	61	64	90	55
LLF	BC	0.80	31.94	7190	1.04E+10	4.77E+06	2142.4	1600.7	1950.8	71	79	90	55
LLF	BM	0.80	31.89	7184	1.04E+10	4.77E+06	2146.2	1602.9	1953.4	62	65	91	55
LLF	LDM	0.80	31.96	7206	1.04E+10	4.77E+06	2146.6	1603.1	1953.5	67	73	90	55
LLF	LWR	0.80	31.89	7204	1.04E+10	4.77E+06	2146.7	1604.1	1955.0	67	73	91	55
PLLF	BPRM	0.90	32.62	7351	2.26E+05	2.28E+04	4.4	3.8	4.0	65	67	94	62
PLLF	BPOM	0.85	32.87	7011	2.52E+05	2.37E+04	4.7	4.0	4.2	65	67	94	61
PLLF	BC	0.80	32.68	7170	2.52E+05	2.41E+04	4.7	4.0	4.3	74	81	90	61
PLLF	BM	0.85	32.88	7110	2.51E+05	2.37E+04	4.6	4.0	4.3	65	67	94	62
PLLF	LDM	0.85	32.92	7111	2.52E+05	2.38E+04	4.7	4.0	4.3	68	70	94	62
PLLF	LWR	0.85	32.88	7142	2.52E+05	2.39E+04	4.6	4.0	4.3	67	70	94	62
SWF	BPRM	0.95	35.78	1502	1.63E+05	3.20E+03	21.1	2.6	9.6	34	48	80	6
SWF	BPOM	0.95	43.17	730	7.77E+05	1.44E+04	49.5	9.6	27.4	13	13	93	13
SWF	BC	0.95	43.18	735	7.78E+05	1.44E+04	49.5	9.7	27.4	14	13	93	13
SWF	BM	0.95	43.17	730	7.77E+05	1.44E+04	49.5	9.6	27.4	13	13	93	13
SWF	LDM	0.95	43.15	736	7.77E+05	1.44E+04	49.5	9.7	27.4	14	14	93	13
SWF	LWR	0.95	43.15	729	7.77E+05	1.44E+04	49.5	9.7	27.4	13	13	93	13
DSWF	BPRM	0.95	35.57	1551	2.77E+05	4.44E+03	37.9	1.1	12.5	34	48	87	5
DSWF	BPOM	0.95	42.96	719	1.51E+06	2.08E+04	88.2	14.4	39.4	13	13	92	12
DSWF	BC	0.95	42.92	728	1.65E+06	2.25E+04	97.3	15.8	40.5	13	14	93	12
DSWF	BM	0.95	42.96	719	1.51E+06	2.08E+04	88.2	14.4	39.4	13	13	92	12
DSWF	LDM	0.95	42.95	715	1.78E+06	2.31E+04	113.7	17.1	41.9	13	13	93	12
DSWF	LWR	0.95	42.86	716	1.54E+06	2.10E+04	111.0	14.7	39.5	13	13	91	12

5.2.2.3 Case Study Three

The results for the simulation study of Case Study Three are summarized in Table 5.8. In this case, the expected parallelization factor for Batch WFGs is higher than in Case Study Two resulting in much tighter deadlines. Whereas, the parallelization factor for Interactive WFGs is lower than in Case Study Two resulting in loose deadline.

Here, FCFS and FCLS perform slightly better than in Case Study Two. Since FCFS and FCLS are static scheduling approaches, they still schedule WFGs in the same order as in Case Study Two. As there are more Interactive WFGs than Batch WFGs, the loose deadline for Interactive WFGs gives a performance gain, while Batch WFGs suffer from the tight deadline.

Since EDF, LLF and PLLF are dynamic policies, changing the deadline characteristics directly influences the performance of these policies. These policies favor Batch WFGs over Interactive and Webservice because Batch WFGs' deadlines are made tighter. As the Batch WFG's deadlines are unrealistic, it causes Interactive and Webservice to suffer. Hence, there is degradation in performance of those policies.

DSWF and SWF perform slightly better than in Case Study Two because these two policies schedule WFGs based on their duration rather than deadline. Hence, these policies make scheduling decisions similar to Case Study Two. Since the deadline of Interactive WFGs are made looser, they would not be tardy as in Case Study Two. As there are a fewer number of Batch WFGs compared to Interactive WFGs, the missed deadline cost in Batch WFGs is relatively less than the rewards obtained from Interactive WFGs.

Fig. 5.15 shows that the DSWF significantly minimizes the Sigmoid cost compared to PLLF and FCFS. The cost associated with DSWF is approximately nine times less than FCFS and PLLF. The corresponding tardiness graph is shown in Fig. 5.16. It also shows that PLLF minimizes the maximum tardiness but makes lots of WFGs tardy thereby resulting in high cumulative sigmoid cost.

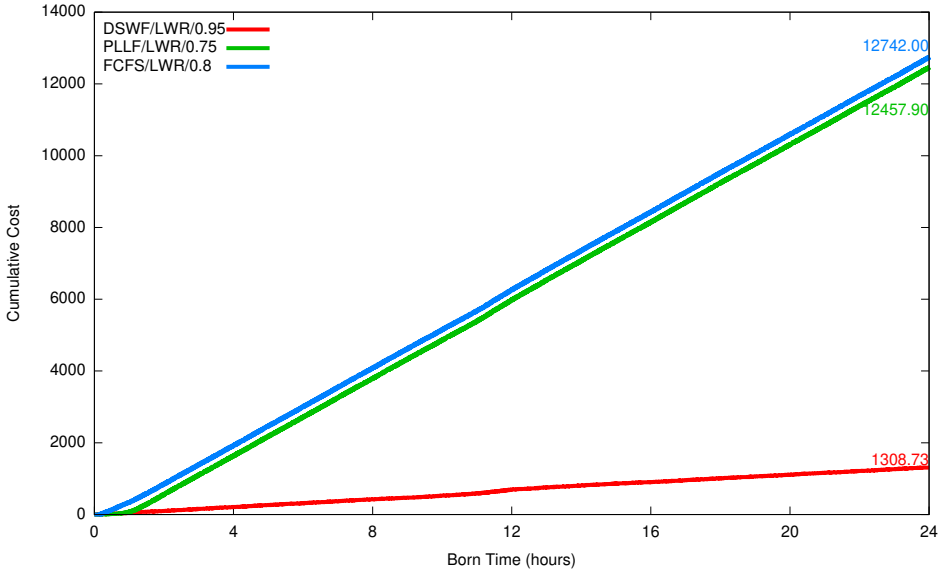

Figure 5.15: The cumulative running cost of all WFGs by born time for DSWF, PLLF and FCFS assuming sigmoid cost in Case Study Three.

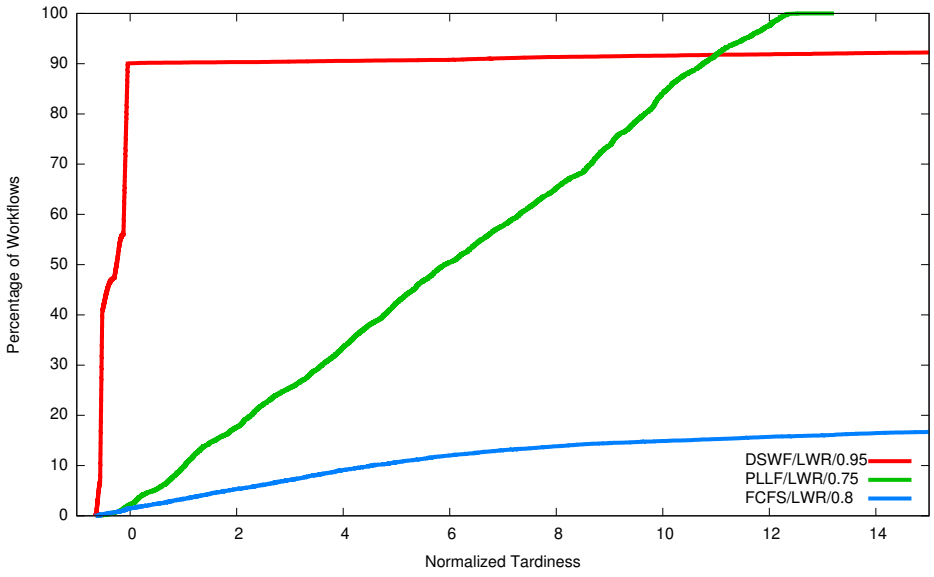

Figure 5.16: Percentage of workflows as a function of normalized tardiness for DSWF, PLLF and FCFS in Case Study Three.

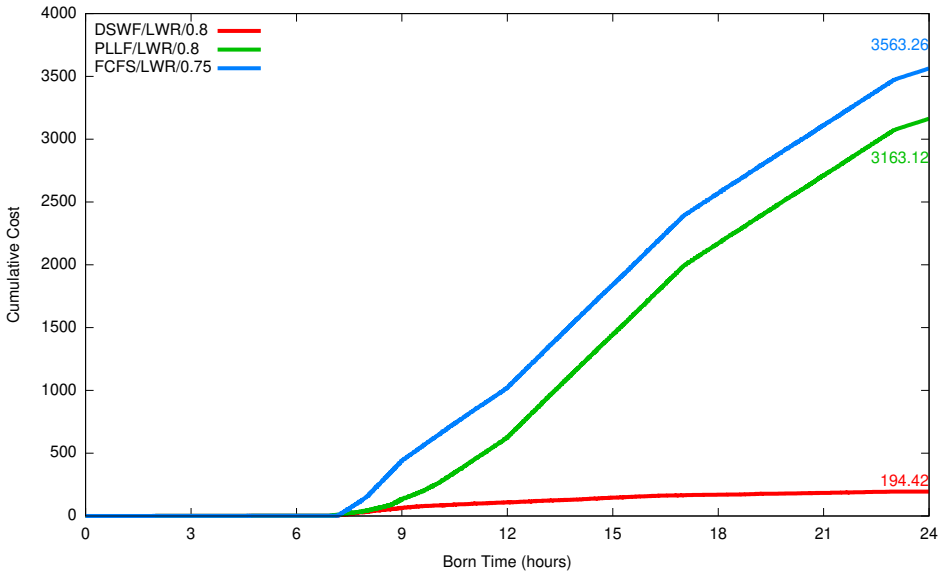

Figure 5.17: The cumulative running cost of all WFGs by born time for DSWF, PLLF and FCFS assuming sigmoid cost in Case Study Four.

As in the previous case studies, post mapping policies perform better than pre-mapping policies which is shown in Table 5.8.

5.2.2.4 Case Study Four

The results for the simulation study of Case Study Four are summarized in Table 5.9. This case study is different than the previous three case studies in terms of arrival rate of WFGs.

Fig.5.13 shows that the DSWF significantly minimizes the Sigmoid cost compared to PLLF and FCFS. The cost associated with DSWF is approximately eighteen times less than FCFS and sixteen times less than PLLF. The corresponding tardiness graph is shown in Fig. 5.18 where PLLF minimizes the maximum tardiness but it makes most of WFGs tardy there by resulting in high cumulative sigmoid cost.

Table 5.9 provides performance of all MSPs which shows that post mapping policies perform better than pre-mapping policy. The performance of MSPs in this case study is consistent to MSPs in the previous three cases.

Table 5.8: Statistics for simulation of Case Study Three.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.60	25.07	8446	1.85E+10	8.08E+06	3472.4	2539.1	3132.3	66	66	75	65
FCFS	BPOM	0.80	23.31	8385	1.43E+10	6.87E+06	3074.2	2204.5	2743.7	65	65	66	64
FCFS	BC	0.80	23.34	8412	1.43E+10	6.87E+06	3070.2	2203.5	2743.0	68	71	65	64
FCFS	BM	0.80	23.31	8386	1.43E+10	6.87E+06	3073.1	2205.0	2745.5	65	65	66	64
FCFS	LDM	0.80	23.31	8405	1.43E+10	6.87E+06	3068.0	2203.8	2743.0	67	68	65	64
FCFS	LWR	0.80	23.31	8391	1.43E+10	6.87E+06	3070.4	2204.3	2743.0	66	68	65	64
FCLS	BPRM	0.95	34.67	1916	6.06E+07	6.22E+04	3154.0	19.4	65.0	24	26	93	19
FCLS	BPOM	0.90	33.30	1289	1.41E+07	4.07E+04	1336.5	19.3	56.3	17	16	93	18
FCLS	BC	0.90	33.35	1315	1.64E+07	4.25E+04	1579.5	19.3	56.6	18	18	93	18
FCLS	BM	0.90	33.29	1283	1.41E+07	4.05E+04	1560.0	19.3	56.2	18	17	93	18
FCLS	LDM	0.90	33.28	1285	1.22E+07	3.95E+04	1569.2	19.2	56.2	18	17	93	18
FCLS	LWR	0.90	33.28	1306	1.07E+07	4.03E+04	1084.5	19.2	56.0	18	17	93	18
EDF	BPRM	0.90	29.11	6765	6.85E+09	3.82E+06	2506.4	1540.4	2166.7	56	58	87	53
EDF	BPOM	0.80	29.03	6671	7.38E+09	3.87E+06	2515.1	1546.4	2174.4	55	56	87	52
EDF	BC	0.80	29.07	6684	7.38E+09	3.87E+06	2514.2	1547.1	2173.7	58	61	87	52
EDF	BM	0.80	29.07	6673	7.38E+09	3.87E+06	2511.9	1546.7	2174.8	55	56	86	52
EDF	LDM	0.80	29.08	6678	7.38E+09	3.87E+06	2509.6	1546.1	2173.4	56	58	87	52
EDF	LWR	0.80	29.05	6676	7.38E+09	3.87E+06	2515.7	1547.3	2174.6	56	58	86	52
LLF	BPRM	0.90	28.14	7972	1.03E+10	5.47E+06	2699.6	1931.2	2479.3	63	63	100	62
LLF	BPOM	0.75	27.91	7928	1.01E+10	5.33E+06	2639.2	1891.7	2423.5	63	63	100	61
LLF	BC	0.80	28.09	7903	1.09E+10	5.48E+06	2699.7	1929.8	2475.7	66	68	100	61
LLF	BM	0.75	27.94	7927	1.01E+10	5.33E+06	2637.5	1890.5	2423.4	63	63	100	61
LLF	LDM	0.75	27.90	7946	1.02E+10	5.34E+06	2641.7	1894.0	2425.2	67	69	100	61
LLF	LWR	0.75	27.91	7937	1.01E+10	5.34E+06	2639.6	1891.2	2424.0	65	67	100	61
PLLF	BPRM	0.85	32.53	7934	2.47E+05	2.64E+04	5.6	4.9	5.2	64	64	100	63
PLLF	BPOM	0.75	32.46	7870	2.45E+05	2.62E+04	5.5	4.8	5.2	64	64	100	63
PLLF	BC	0.75	32.57	7849	2.47E+05	2.65E+04	5.7	4.8	5.2	67	69	100	63
PLLF	BM	0.80	32.68	7843	2.74E+05	2.75E+04	5.9	5.0	5.4	64	64	100	63
PLLF	LDM	0.80	32.75	7804	2.75E+05	2.77E+04	5.8	5.0	5.4	65	66	100	63
PLLF	LWR	0.75	32.48	7871	2.46E+05	2.63E+04	5.6	4.8	5.2	66	68	100	63
SWF	BPRM	0.95	35.78	993	1.61E+05	1.09E+03	22.8	2.5	9.6	17	22	93	6
SWF	BPOM	0.95	43.17	652	7.76E+05	1.25E+04	51.9	9.6	27.4	8	5	96	13
SWF	BC	0.95	43.18	651	7.76E+05	1.26E+04	51.9	9.6	27.4	8	5	96	13
SWF	BM	0.95	43.17	652	7.76E+05	1.25E+04	51.9	9.6	27.4	8	5	96	13
SWF	LDM	0.95	43.15	652	7.76E+05	1.26E+04	51.9	9.6	27.4	8	5	96	13
SWF	LWR	0.90	34.32	449	1.64E+05	7.31E+02	22.1	2.6	9.6	9	10	90	5
DSWF	BPRM	0.95	35.64	972	2.58E+05	2.01E+03	37.7	1.5	12.2	16	21	94	5
DSWF	BPOM	0.95	43.01	616	1.35E+06	1.74E+04	84.1	11.5	37.4	7	4	97	12
DSWF	BC	0.95	42.95	620	1.40E+06	1.81E+04	85.7	11.5	38.5	7	4	96	12
DSWF	BM	0.95	43.01	616	1.35E+06	1.74E+04	84.1	11.5	37.4	7	4	97	12
DSWF	LDM	0.95	42.88	620	1.40E+06	1.81E+04	103.0	12.3	37.7	7	4	96	12
DSWF	LWR	0.95	42.91	617	1.34E+06	1.73E+04	99.3	11.1	37.4	7	4	96	12

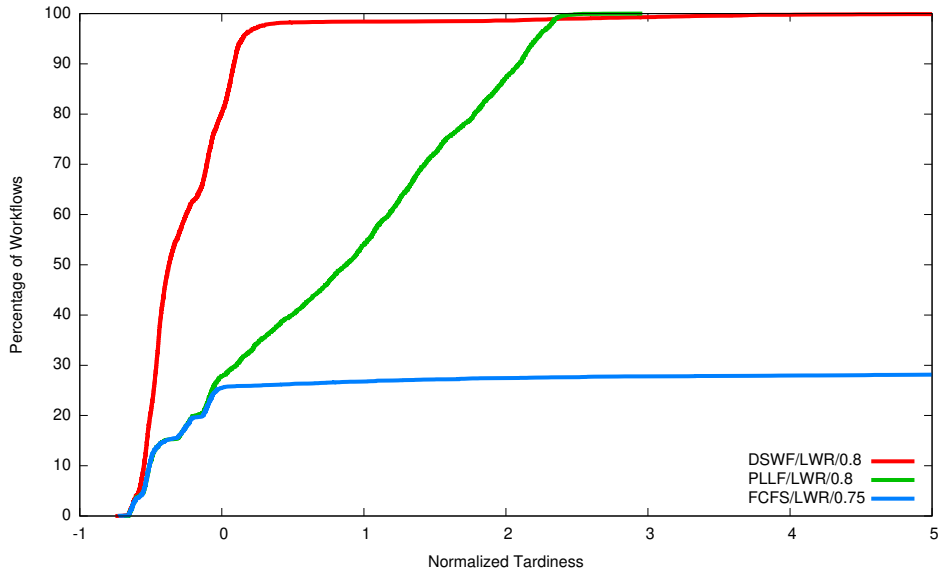

Figure 5.18: Percentage of workflows as a function of normalized tardiness for DSWF, PLLF and FCFS in Case Study Four.

5.2.3 Resource Requirement Analysis

This section analyzes the performance of various policies in terms of number of machines required for the four cases considered to perform with ideal performance. The required number of machines is identified by using a binary search method, where the cluster size of 16 machines is assumed, and more machines (multiple of 16) are added to find the approximate number of the required machines for each policy.

Fig. 5.19 shows the effect of adding machines for FCFS operating on Case Study One. In the figure, it shows that the cumulative cost for FCFS decreases gradually. For example, the percentage of reduction in cost due to adding 16, 32, 48, 64 machines on the cluster of size 16 are 58.22%, 76.26%, 84.74% and 96.57% respectively. Fig. 5.19 shows the corresponding percentage of workflows whose normalized tardiness is at or below the given value of normalized tardiness.

Fig. 5.21 shows the effect of adding machines for PLLF operating on Case Study One. In the figure, it shows that the cumulative cost for PLLF decreases sharply. For example, the percentage of reduction in cost due to adding 16, 32 machines on

Table 5.9: Statistics for simulation of Case Study Four.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.90	31.26	3553	6.71E+09	3.19E+06	3397.0	2608.9	3204.1	74	78	76	70
FCFS	BPOM	0.75	31.28	3554	6.73E+09	3.20E+06	3405.7	2613.5	3210.8	74	78	76	70
FCFS	BC	0.80	31.38	3557	6.73E+09	3.19E+06	3398.7	2611.1	3207.9	76	82	76	70
FCFS	BM	0.75	31.28	3553	6.74E+09	3.20E+06	3407.0	2615.5	3208.6	74	78	76	70
FCFS	LDM	0.75	31.35	3554	6.75E+09	3.20E+06	3413.0	2617.2	3215.8	75	79	76	70
FCFS	LWR	0.75	31.30	3554	6.75E+09	3.20E+06	3413.0	2618.6	3213.1	74	78	76	70
FCLS	BPRM	0.95	34.18	494	6.60E+05	6.87E+03	179.6	2.0	53.5	19	27	81	6
FCLS	BPOM	0.80	30.98	436	3.51E+05	4.14E+03	125.7	0.7	41.6	24	37	78	5
FCLS	BC	0.80	31.11	444	3.45E+05	4.15E+03	112.5	0.7	41.9	26	41	78	5
FCLS	BM	0.80	31.00	442	3.46E+05	4.11E+03	123.3	0.8	41.8	24	37	78	5
FCLS	LDM	0.80	31.10	447	3.54E+05	4.22E+03	112.4	0.8	41.9	25	39	78	5
FCLS	LWR	0.80	31.01	446	3.41E+05	4.10E+03	111.3	0.8	41.8	24	37	78	5
EDF	BPRM	0.90	31.44	2676	2.02E+09	1.37E+06	2391.6	1538.0	2090.8	60	66	80	51
EDF	BPOM	0.80	31.42	2643	2.03E+09	1.37E+06	2398.7	1540.7	2092.0	60	66	80	51
EDF	BC	0.80	31.66	2650	2.03E+09	1.37E+06	2393.9	1539.3	2090.2	62	70	79	51
EDF	BM	0.80	31.42	2639	2.03E+09	1.37E+06	2397.4	1538.4	2093.6	60	66	80	51
EDF	LDM	0.80	31.51	2648	2.03E+09	1.37E+06	2392.7	1537.6	2092.1	61	68	79	51
EDF	LWR	0.80	31.44	2644	2.03E+09	1.37E+06	2397.2	1538.8	2091.5	60	66	80	51
LLF	BPRM	0.90	31.54	3283	4.21E+09	2.32E+06	2865.0	2144.5	2694.6	70	74	99	64
LLF	BPOM	0.80	31.48	3276	4.20E+09	2.31E+06	2858.9	2140.1	2687.7	70	74	98	64
LLF	BC	0.80	31.72	3279	4.20E+09	2.31E+06	2859.4	2141.6	2691.1	72	78	99	64
LLF	BM	0.80	31.48	3281	4.20E+09	2.31E+06	2857.9	2140.6	2688.7	70	74	98	64
LLF	LDM	0.90	32.86	3288	5.49E+09	2.65E+06	3241.5	2436.1	3055.0	70	73	99	64
LLF	LWR	0.80	31.51	3266	4.20E+09	2.31E+06	2859.5	2138.7	2689.6	70	73	98	64
PLLF	BPRM	0.90	31.38	3214	2.25E+04	4.29E+03	3.2	2.3	2.5	72	76	98	66
PLLF	BPOM	0.80	31.30	3169	2.19E+04	4.16E+03	3.0	2.3	2.5	72	75	98	66
PLLF	BC	0.80	31.54	3165	2.20E+04	4.22E+03	3.1	2.3	2.5	74	79	98	66
PLLF	BM	0.80	31.26	3179	2.20E+04	4.18E+03	3.0	2.3	2.5	72	75	98	66
PLLF	LDM	0.80	31.36	3186	2.21E+04	4.21E+03	3.1	2.3	2.5	73	78	98	66
PLLF	LWR	0.80	31.25	3171	2.19E+04	4.18E+03	3.0	2.3	2.5	72	75	98	66
SWF	BPRM	0.95	34.39	247	3.63E+03	-1.29E+03	5.7	0.2	2.0	14	23	69	0
SWF	BPOM	0.80	31.23	211	3.38E+03	-1.27E+03	4.8	0.2	1.5	20	35	61	0
SWF	BC	0.80	31.46	221	3.49E+03	-1.20E+03	4.8	0.2	1.6	22	40	62	0
SWF	BM	0.80	31.22	210	3.38E+03	-1.27E+03	4.8	0.2	1.5	20	35	62	0
SWF	LDM	0.80	31.32	217	3.44E+03	-1.23E+03	4.8	0.2	1.6	22	38	62	0
SWF	LWR	0.80	31.25	224	3.42E+03	-1.25E+03	4.8	0.2	1.6	20	36	62	0
DSWF	BPRM	0.95	34.48	222	4.59E+03	-1.22E+03	13.8	0.2	3.0	13	23	61	0
DSWF	BPOM	0.80	31.19	194	4.10E+03	-1.21E+03	14.0	0.2	2.4	20	36	55	0
DSWF	BC	0.80	31.40	200	4.31E+03	-1.12E+03	13.0	0.2	2.5	22	39	56	0
DSWF	BM	0.80	31.12	197	4.19E+03	-1.19E+03	12.5	0.2	2.5	20	36	55	0
DSWF	LDM	0.80	31.22	205	4.64E+03	-1.15E+03	11.9	0.2	2.4	21	38	54	0
DSWF	LWR	0.80	31.09	197	4.30E+03	-1.19E+03	18.1	0.2	2.3	20	36	54	0

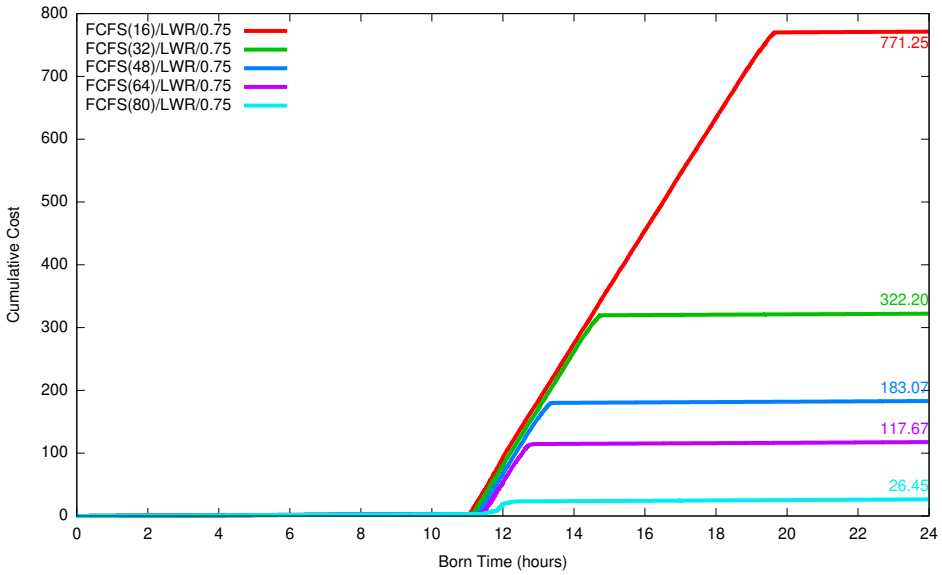

Figure 5.19: The cumulative running cost of all WFGs by born time for FCFS with multiple cluster size operating on the Case Study One.

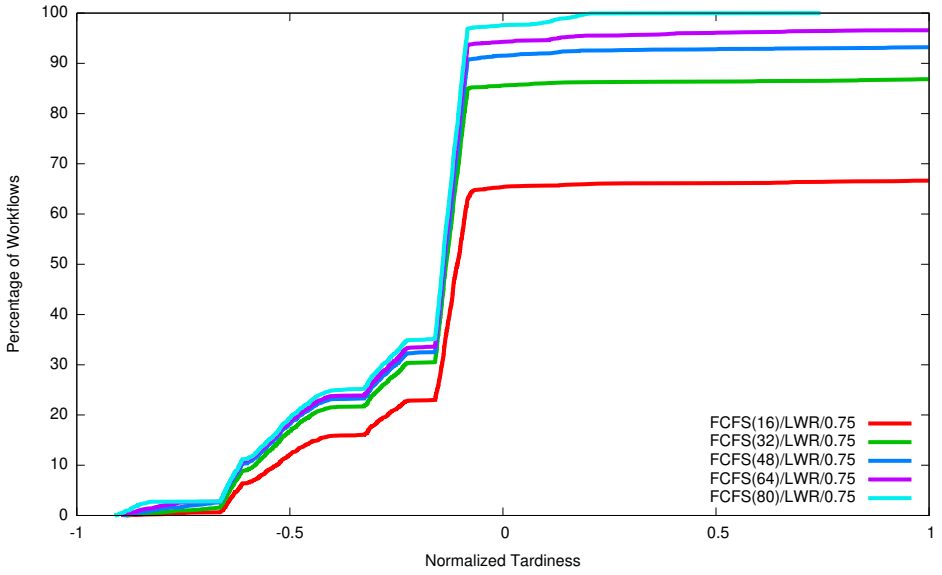

Figure 5.20: Percentage of workflows as a function of normalized tardiness for FCFS with multiple cluster size operating on the Case Study One.

the cluster of size 16 are 91.95%, and 96.68% respectively. Fig. 5.21 shows the corresponding percentage of workflows whose normalized tardiness is at or below the given value of normalized tardiness.

Fig. 5.23 shows the effect of adding machines for DSWF operating on Case Study

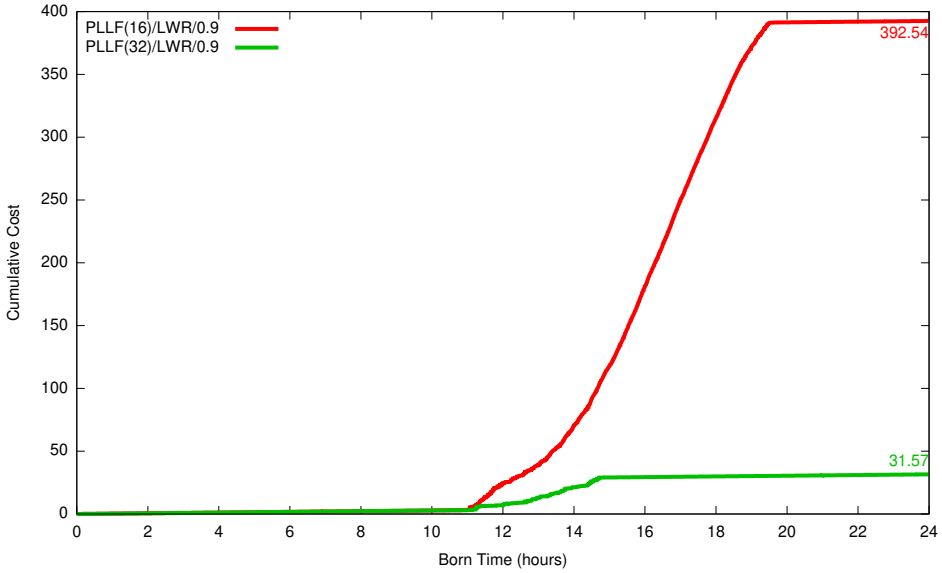

Figure 5.21: The cumulative running cost of all WFGs by born time for PLLF with multiple cluster size operating on the Case Study One.

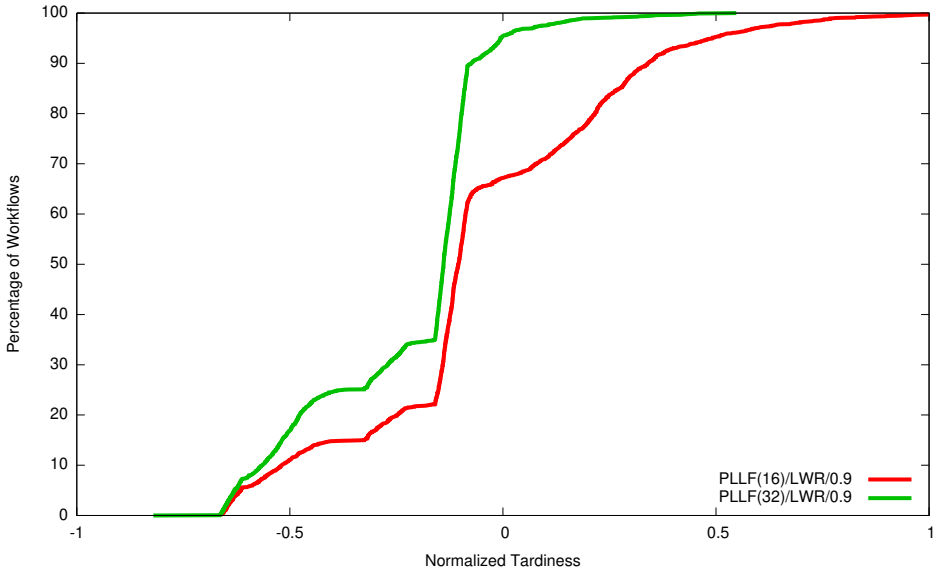

Figure 5.22: Percentage of workflows as a function of normalized tardiness for PLLF with multiple cluster size operating on the Case Study One.

One. In the figure, it shows that the cumulative cost for DSWF decreases as the machines are added to the cluster. For example, the percentage of reduction in cost due to adding 16, 32 machines on the cluster of size 16 are 68.80%, and 76.64% respectively. Fig. 5.21 shows the corresponding percentage of workflows whose normalized

tardiness is at or below the given value of normalized tardiness.

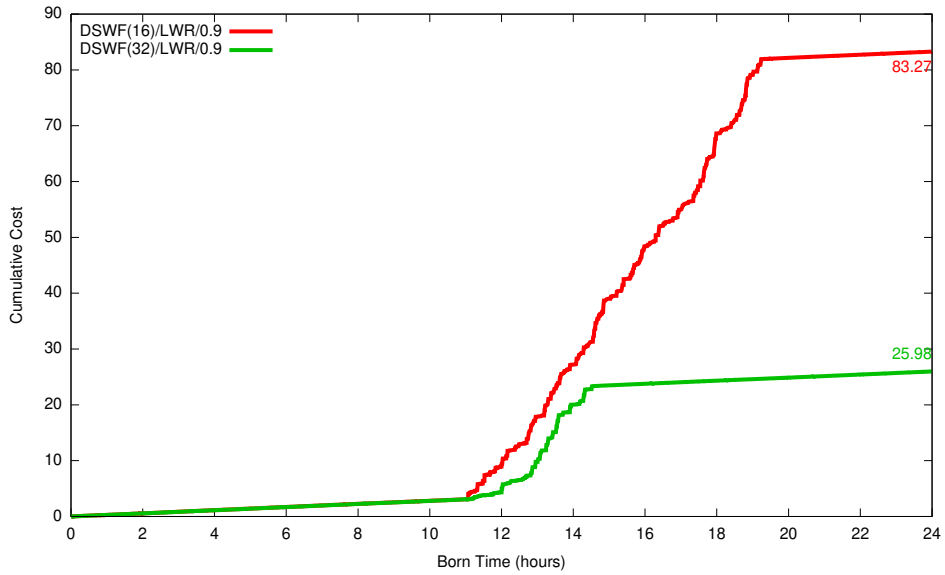

Figure 5.23: The cumulative running cost of all WFGs by born time for DSWF with multiple cluster size operating on the Case Study One.

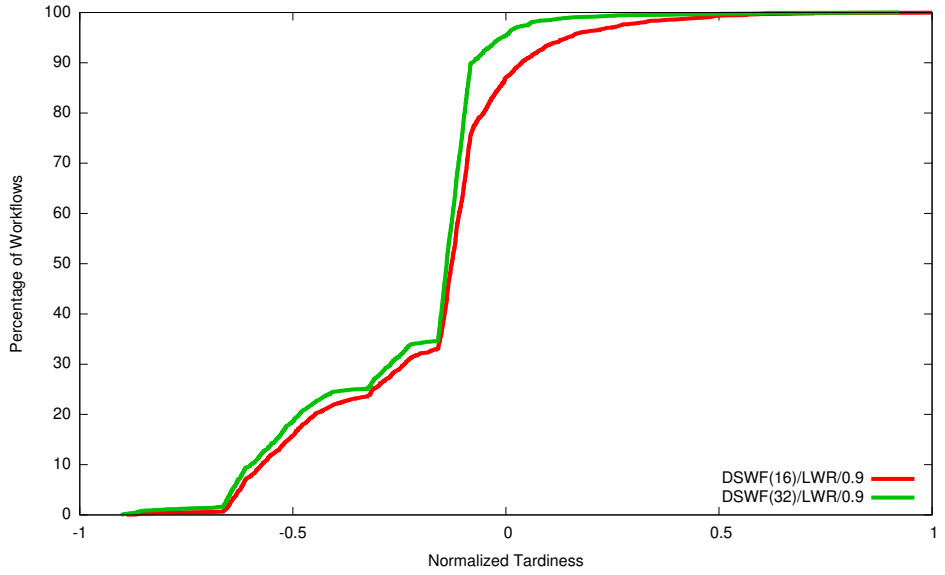

Figure 5.24: Percentage of workflows as a function of normalized tardiness for DSWF with multiple cluster size operating on the Case Study One.

For the purpose of comparing the performances of different policies' resource requirements, three policies are compared for all the cases (refer to Appendix A for all policies). Fig. 5.25 through Fig. 5.28 illustrate how the cumulative cost decreases as

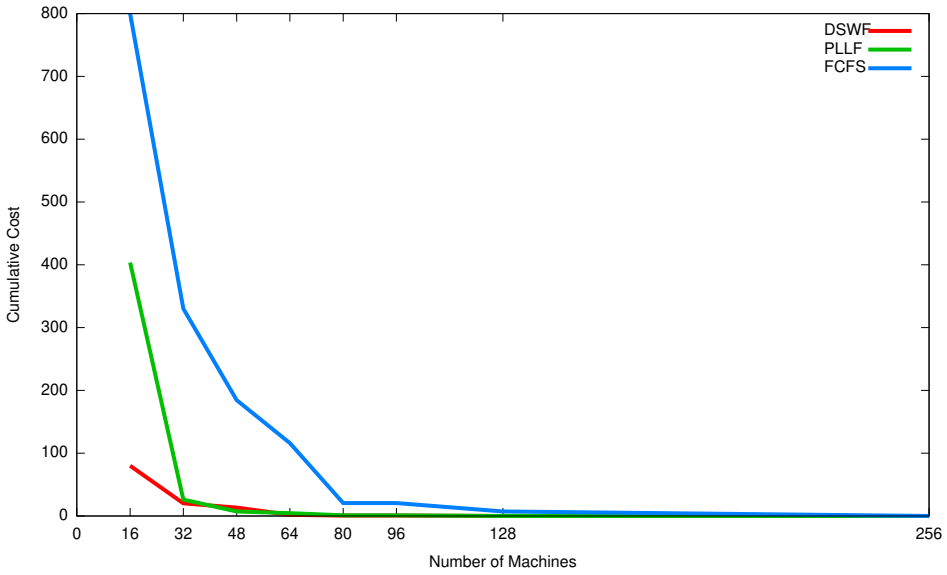

Figure 5.25: Cumulative cost as a function of number of quad-core machines assumed in the cluster for the Case Study One.

more machines are added to the cluster for all the four cases considered before. In Fig. 5.25, the number of machines required for PLLF is about 32, whereas, FCFS requires approximately three to four times more machines to get the similar performance. DSWF requires 32 machines for Case Study One, Case Study Two and Case Study Three which are shown in Fig. 5.25, Fig. 5.26 and Fig. 5.27 respectively. Whereas, PLLF requires 64 machines for Case Study Two and Case Study Three which is double the size than Case Study One. In case of FCFS, more than 128 machines are needed to achieve the similar performance as DSWF and PLLF policies. Fig. 5.28 shows the similar trend in the Case Study Four.

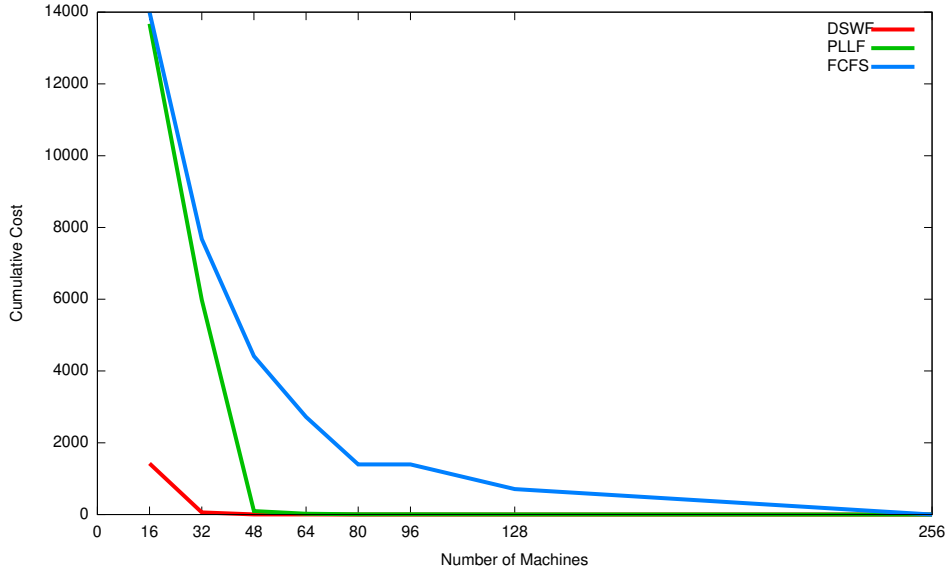

Figure 5.26: Cumulative cost as a function of number of quad-core machines assumed in the cluster for the Case Study Two.

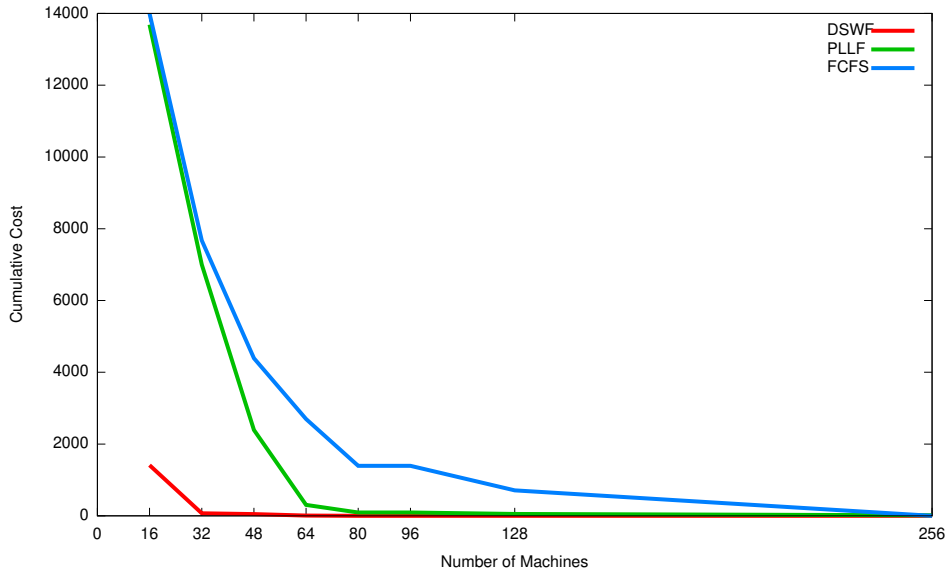

Figure 5.27: Cumulative cost as a function of number of quad-core machines assumed in the cluster for the Case Study Three.

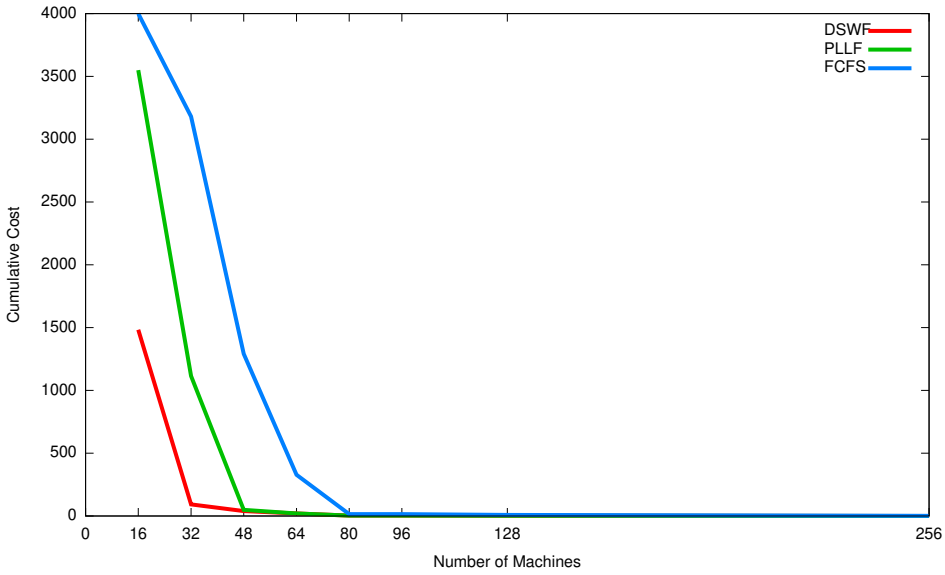

Figure 5.28: Cumulative cost as a function of number of quad-core machines assumed in the cluster for the Case Study Four.

5.3 Concluding Remarks

Simulation studies are conducted for evaluating performances of seven RSPs and six MSPs. Four case studies are analyzed that varied in terms of arrival pattern and request characteristics. Two cost functions, sigmoid and quadratic, are used for comparing the performance of these policies. First, extensive threshold analysis is performed to obtain a good threshold values for each policies in different case studies. Based on the threshold values obtained, performance of RSPs and MSPs are analyzed. It is shown that the performance of DSWF is superior to all the other policies considered in terms of minimizing sigmoid cost. On the other hand, PLLF policy is better in terms of minimizing the maximum tardiness. The resource requirements of DSWF is three to four times less than the FCFS policy considered for four case studies.

Chapter 6

Conclusions

A new simulation environment is introduced for modeling the execution of workflows (WFGs) on a cluster of memory-managed multicore machines. There are three major components of the considered environment, referred to as a scheduling framework: WFG Generator, Scheduler and Cluster of Machines. The WFG Generator models the client's WFGs generation in terms of Interactive, Webservice, and Batch. The Scheduler component of the framework is responsible for scheduling clients' WFGs to the Cluster of Machines using two types of heuristics: Request Selection Policy (RSP) and Machine Selection Policy (MSP). The RSP is responsible for selecting ready requests of WFGs to be considered for execution; the MSP is used to select the best available machine for executing those requests. A cluster of Machines is modeled, each of them following an efficiency based machine model. The machine model proposed comprehends the reality that the efficiencies of memory-managed machines are impacted by the loading of their heap memories as well as the loading of their CPU resources.

Seven RSPs and six MSPs are implemented for the purpose of evaluating these scheduling policies. Forty-two combinations of these RSPs and MSPs are evaluated.

The results of the simulation studies indicate that the newly proposed DSWF scheduling policy outperforms the other evaluated in terms of minimizing sigmoid cost and normalized tardiness. Simulation studies showed that DSWF has high throughput and hence a lower sigmoid cost than all the other policies considered. Another proposed scheduling policy PLLF performed better than LLF and FCFS in minimizing sigmoid cost. In terms of minimizing the normalized tardiness, the PLLF performed better than all the other policies. Two major categories of MSPs are evaluated, namely, pre-mapping and post mapping policies. A new concept of thresholding a machine is introduced. Each of the MSPs considered follow the thresholding of the machines in a cluster. Simulation studies reveal that there exists an optimal threshold value for each MSP. In almost all the cases considered in this thesis, the post mapping policies performed better than the pre-mapping policy.

In the different case studies conducted, it is shown that using intelligent scheduling policies makes a lot of difference in terms of performance and resource requirements. This study showed that the policy DSWF significantly reduces the number of machines required for a given environment. The number of machines required by the DSWF is three to four times less than the FCFS policy. This raises a serious concern over the scheduling policies that are working underneath the system as a black box. This research work suggests that using a good scheduling policy for a given scenario can not only improves the overall system performance, but can also minimize resource requirements. This provides avenues for an organization to move into the direction of better performance as well as lower infrastructure cost.

Chapter 7

Future Work

The following are some of the plans for future work: operational implementation of the scheduling framework; scheduling using planning; robust scheduling; and autonomic scheduling.

To make the scheduling framework operate in a real operational setting, it needs to provide robustness, security and management of client requests. To cope with these necessities and challenges, there can be multiple actions that can be taken. One such action is to provide additional request states. When a request is executing on a machine of the cluster, there can be four new states in addition to the request's state discussed in Chapter 3: migrated, paused, failed, or cancelled. The migrated state refers to the effect of load balancing operations [25] at the machine level. Whenever a machine has a higher load than the other machines in the cluster, then some requests running on that machine could be migrated to the other machines with a lower load. Similarly, pausing an executing request is another important feature which could be useful for debugging purposes. The state associated with pausing an executing request is referred as paused. Sometimes during execution, some requests may fail due to system or application failure. This can be modeled as failed state. The cancelled state refers to the intentional cancellation of an executing request. The

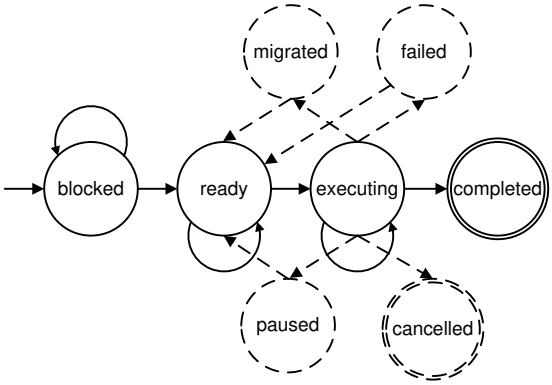

Figure 7.1: Possible states of a Request assigned to a Machine.

dotted lines in Fig 3.15 represent those states that are not yet implemented and are considered for future work.

If the information about the scheduling domain (request arrival rate and duration) are known in advance, planning can help in making scheduling decisions. Generally, it is known that Batch WFGs have daily periodicity. Using this information helps schedule those jobs without interrupting other jobs, like Webservice and Interactive. If there is no information known in advance, then dealing with such cases refers to robust scheduling. In some real systems the information about the arrival and duration is unknown. This is a very challenging area of scheduling research which deals with uncertainty.

In recent days, there has been some advancement toward making a self-aware software system, referred to as autonomic computing [49, 50]. Such a system can adjust itself to handle unpredictable situations. It is very complex and challenging to design a system that can dynamically manage changes in the system. The Autonomic computing can be applied to the scheduling field by making a system have an autonomous scheduler. The term autonomous scheduler is a dynamic scheduler that keeps track of the user's requirements as well as the knowledge of the system, and performs to maximize the overall gain of the system. Machine learning technologies can be used

to address such tasks. Preliminary studies were conducted for the Self Managed Systems (SMS) using reinforcement learning. In the study conducted, a reinforcement learning agent (RLA) was developed which selects a request selection policy among the FCFS, EDF and LLF. A fixed post mapping machine selection policy is used for this purpose. Because the performance of scheduling policies depend on the scenario under consideration and changing the scheduling policies would change the overall performance of the system. The RLA assigns a reward to the scheduling policy being used based on the resulting tardiness of the scheduled WFGs. The preliminarily results showed that the performance of reinforcement learning based scheduling approach was promising in terms of minimizing the tardiness compared to the FCFS policy.

Bibliography

[1] G. E. Moore. Cramming more components onto integrated circuits. *Electonics*, 38(8):114–17, 1965.

[2] S. Hamilton. Taking moore's law into the next century. *Computer*, 32(1):43–48, January 1999.

[3] O. Sinnen. Task scheduling for parallel systems. 2007.

[4] R. Sharma J. Fialli M. Hapner, R. Burridge and K. Hasse. Java message service api tutorial and reference: Messaging for the j2ee platform. 2002.

[5] G. Hohpe and B. Woolf. Enterprise integration patterns: Designing, building, and deploying messaging solutions. 2004.

[6] C. Overton. On the theory and practice of internet slas. *Computer Measurement Group, Journal of Computer Resource Measurement*, 106:32–45, April 2002.

[7] A. Keller and H. Ludwig. The wsla framework - specifying and monitoring service level agreements for web services. *Journal of Network and System Management*, 11(1), 2003.

[8] B. M. et al. Managing of ebusiness on demand sla contracts in business terms using the cross-sla execution manager sam. *IBM, IEEE ISADS*, 2003.

[9] R. R. Schaller. Moores law: Past, present, and future. *IEEE Spectrum*, 34:52–59, 1997.

[10] J. Bentley. *Programming Pearls*. ACM Press & Addison Wesley, New York, NY, 2000.

[11] N. Bontis. Intellectual capital: an exploratory study that develops measures and models. *Management Decision*, pages 63–76, 1998.

[12] N. Bontis. Assessing knowledge assets: a review of the models used to measure intellectual capital. *International Journal of Management Reviews*, 3:41–60, 2001.

[13] M. Beltrán, A. Guzmán, and J. L. Bosque. A new cpu availability prediction model for time-shared systems. *IEEE Transactions on Computers*, 57(7):865–875, July 2008.

[14] Y. Zhang, W. Sun, and Y. Inoguchi. Predicting running time of grid tasks on cpu load predictions. *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing*, pages 286–292, September 2006.

[15] A. W. Appel. Garbage collection can be faster than stack allocation. *Information Processing Letters*, 25(4):275–279, June 1987.

[16] M. Hertz. *Quantifying and Improving the Performance of Garbage Collection*. Ph.D. Dissertation, University of Massachusetts, Amherst, 2006.

[17] M. Hertz and E. D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. *Proceedings of the Object-Oriented Programming Systems, Languages and Applications (OOPSLA 2005)*, October 2005.

[18] R. Jones and R. Lins. *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*. John Wiley & Sons, New York, NY, 1996.

[19] H. Koide and Y. Oie. A new task scheduling method for distributed programs that require memory management. *Concurrency and Computation: Practice and Experience*, 18:941–945, 2006.

[20] S. Dhakal, M. M. Hayat, J. E. Pezoa, C. Yang, and D. A. Bader. Dynamic load balancing in distributed systems in the presence of delays: A regeneration-theory approach. *IEEE Transactions on Parallel & Distributed Systems*, 18(4):485–497, April 2007.

[21] D. Dyachuk and R. Deters. Using sla context to ensure quality of service for composite services. *IEEE Transactions on Computers*, 57(7):865–875, July 2008.

[22] J. K. Kim, S. Shivle, H. J. Siegel, A. A. Maciejewski, T. Braun, M. Schneider, S. Tideman, R. Chitta, R. B. Dilmaghani, R. Joshi, A. Kaul, A. Sharma, S. Sripada, P. Vangari, and S. Sankar Yellampalli. Dynamic mapping in a heterogeneous environment with tasks having priorities and multiple deadlines. *12th Heterogeneous Computing Workshop (HCW 2003), in Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, April 2003.

[23] S. H. Oh and S. M. Yang. A modified least-laxity-first scheduling algorithm for real-time tasks. *Proceedings of the 5th International Workshop on Real-Time Computing Systems and Applications (RTCSA '98)*, pages 31–36, October 1998.

[24] V. Salmani, M. Naghibzadeh, A. Habibi, and H. Deldari. Quantitative comparison of job-level dynamic scheduling policies in parallel real-time systems. *Proceedings TENCON, 2006 IEEE Region 10 Conference*, November 2006.

[25] C. Kim H. Kameda, J. Li and Y. Zhang. Optimal load balancing in distributed computer systems. 1997.

[26] Y. Feizabadi and G. Back. Garbage collection-aware utility accrual scheduling. *Real-Time Systems*, 36(1-2):3–22, July 2007.

[27] M. Hornick D. Georgakopoulos and A. Sheth. An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3:119153, 1995.

[28] B. Kiepuszewski W.M.P. van der Aalst, A.H.M. ter Hofstede and A.P. Barros. Advanced workflow patterns. *7th International Conference on Cooperative Information Systems (CoopIS 2000), O. Etzion and P. Scheuermann (Eds.)*, 1901, 2000.

[29] B. Kiepuszewski A.P. Barros W.M.P. van der Aalst, A.H.M. ter Hofstede. Workflow patterns. *Distributed Parallel Databases*, 14(1):5–51, 2003.

[30] D. Edmond N. Russell, A.H.M. ter Hofstede and W.M.P. van der Aalst. Workflow data patterns. *Technical Report QUT Technical Report FIT-TR-2004-01, Queensland University of Technology, Brisbane*, 2004.

[31] D. Edmond N. Russell, A.H.M. ter Hofstede and W.M.P. van der Aalst. Workflow resource patterns. *BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven*, 2004.

[32] J. Yu and R. Buyya. A taxonomy of scientific workflow systems for grid computing. *SIGMOD Record*, 34(3), 2005.

[33] P. M. Papazoglou. Service -oriented computing: Concepts, characteristics and directions. *International Conference on Web Information Systems Engineering (WISE'03)*, 4, 2003.

[34] R. Khalaf W. Nagy N. Mukhi F. Curbera, M. Duftler and S. Weerawarana. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing*, 6(2):86–93, March 2002.

[35] J. OHara. Toward a commodity enterprise middleware. *Queue*, 5(4):4855, 2007.

[36] M. Menth et al. Throughput performance of popular jms servers. *Joint international conference on Measurement and modeling of computer systems*, pages 367–368, 2006.

[37] J. Bacon K. Sachs, S. Kounev and A. Buchmann. Performance evaluation of message-oriented middleware using the specjms2007 benchmark. *Performance Evaluation*, 66(8):410–434, 2009.

[38] C. Taton and N. De Palma. Improving the performances of jms-based applications. *Autonomic Computing*, 1(1):81 – 102, 2009.

[39] S. Narravula P. Lai H. Subramoni, G. Marsh and D. Panda. Design and evaluation of benchmarks for financial applications using advanced message queuing protocol (amqp) over infiniband. *WHPCF*, page 18, Nov. 2008.

[40] A. Kupsys and R. Ekwall. Architectural issues of jms compliant group communication. *IEEE NCA*, 7:139–148, 2005.

[41] M. Stal. Using architectural patterns and blueprints for service-oriented architecture. *IEEE Software*, 23(2), Mar./Apr. 2006.

[42] T. I. Seidman. First come first serve can be unstable. *IEEE Transactions on Automattic Control*, 39:2166–2171, 1994.

[43] M. Bramson. Instability of fifo queueing networks. *Annals of Applied Probability*, 4:414–431, 1994.

[44] V. Yarmolenko and R. Sakellariou. An evaluation of heuristics for sla based parallel job scheduling. *High Performance Grid Computing Workshop (in conjunction with IPDPS 2006), IEEE Computer Society*, 2006.

[45] R. Sakellariou and V. Yarmolenko. Job scheduling on the grid: Towards sla-based scheduling. *High Performance Computing and Grids in Action*, page 207222, 2008.

[46] K.T. Krishnakumar J. Garibaldi J. MacLaren, R. Sakellariou and D. Ouelhadj. Towards service level agreement based scheduling on the grid. *Workshop on Planning and Scheduling for Web and Grid Services (in conjunction with ICAPS-04)*, 2004.

[47] M. Agrawal M. H Balter, B. Schroeder and N. Bansal. Size-based scheduling to improve web performance. *ACM Transaction on Computer System*, 21, 2003.

[48] J. Little. A proof of the theorem l =lambda w. *Operations Research*, 9, 1961.

[49] I. Whalley D. Chess S. White, J. Hanson and J. Kephart. An architectural approach to autonomic computing. *ICAC*, pages 2–9, 2004.

[50] J. O. Kephart. Research challenges of autonomic computing. *Intl. Conf. Software Eng., ACM*, 27:15–22, 2005.

Appendix A

Resource Analysis Table

Table A.1: Statistics for simulation of Case Study One with 16 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	18.73	774.1	2.68E+08	3.22E+05	1493.2	910.5	1304.8	35.1	35.8	19.4	35.0
FCFS	**BPOM**	0.75	18.70	771.2	2.65E+08	3.19E+05	1484.8	908.3	1309.2	34.9	35.7	19.4	34.7
FCFS	**BC**	0.75	18.73	782.7	2.65E+08	3.20E+05	1492.8	909.3	1310.5	50.2	59.2	17.7	34.9
FCFS	**BM**	0.75	18.76	771.4	2.64E+08	3.19E+05	1493.9	903.7	1305.4	34.9	35.7	19.4	34.7
FCFS	**LDM**	0.80	18.71	786.2	2.62E+08	3.18E+05	1484.6	902.0	1301.2	57.0	69.8	16.1	34.7
FCFS	**LWR**	0.75	18.74	771.3	2.64E+08	3.19E+05	1484.4	907.4	1296.9	34.9	35.7	19.4	34.7
FCLS	**BPRM**	0.95	21.92	199.1	1.50E+04	2.03E+02	51.7	0.4	10.9	20.9	27.9	46.8	4.4
FCLS	**BPOM**	0.90	20.02	138.8	1.02E+04	3.61E+01	41.7	0.3	8.8	15.6	20.3	35.5	4.4
FCLS	**BC**	0.90	20.02	132.9	1.00E+04	5.72E+01	41.6	0.3	8.8	16.2	21.2	35.5	4.4
FCLS	**BM**	0.90	19.98	139.8	9.47E+03	9.21E+00	41.7	0.3	7.0	15.8	20.6	35.5	4.3
FCLS	**LDM**	0.90	20.03	140.4	9.47E+03	2.11E+01	41.7	0.3	7.0	15.7	20.6	35.5	4.3
FCLS	**LWR**	0.90	20.01	143.5	9.47E+03	1.32E+01	41.6	0.3	7.0	16.6	21.9	35.5	4.3
EDF	**BPRM**	0.95	22.31	257.6	5.76E+04	1.38E+03	63.8	1.0	31.5	21.7	27.9	25.8	8.9
EDF	**BPOM**	0.90	20.06	81.4	1.59E+03	-4.29E+02	0.8	0.2	0.4	13.7	21.0	0.0	0.0
EDF	**BC**	0.90	20.15	84.9	1.64E+03	-3.95E+02	0.8	0.2	0.4	13.6	21.0	0.0	0.0
EDF	**BM**	0.90	20.01	79.1	1.58E+03	-4.32E+02	0.9	0.1	0.4	13.1	20.2	0.0	0.0
EDF	**LDM**	0.90	20.11	66.7	1.60E+03	-4.20E+02	0.9	0.1	0.4	13.4	20.7	0.0	0.0
EDF	**LWR**	0.90	20.06	79.9	1.59E+03	-4.28E+02	0.8	0.1	0.4	13.5	20.8	0.0	0.0
LLF	**BPRM**	0.90	19.99	623.4	1.06E+07	5.48E+04	300.9	198.2	273.9	33.7	35.3	88.7	25.6
LLF	**BPOM**	0.90	19.96	544.3	9.19E+06	4.99E+04	282.7	185.3	257.4	29.5	30.1	88.7	23.2
LLF	**BC**	0.90	20.02	562.5	9.84E+06	5.24E+04	291.5	191.8	264.5	30.4	30.8	88.7	24.6
LLF	**BM**	0.90	19.96	546.3	9.42E+06	5.07E+04	288.8	188.2	260.0	29.7	30.2	88.7	23.7
LLF	**LDM**	0.90	19.99	560.0	9.74E+06	5.18E+04	287.3	191.6	263.3	30.2	30.7	88.7	24.1
LLF	**LWR**	0.90	19.98	553.4	9.54E+06	5.13E+04	287.4	189.2	259.8	30.2	30.7	88.7	24.2
PLLF	**BPRM**	0.90	20.12	543.9	2.46E+03	-3.34E+01	2.1	0.6	0.9	34.6	35.2	91.9	28.4
PLLF	**BPOM**	0.90	20.08	390.0	2.23E+03	-1.23E+02	2.0	0.5	0.8	32.7	32.8	88.7	27.7
PLLF	**BC**	0.90	20.13	406.8	2.34E+03	-7.24E+01	1.7	0.6	1.0	32.5	32.6	87.1	27.5
PLLF	**BM**	0.90	20.09	405.8	2.25E+03	-1.14E+02	2.0	0.5	0.9	33.1	33.4	88.7	27.8
PLLF	**LDM**	0.90	20.10	385.8	2.23E+03	-1.12E+02	1.3	0.5	0.9	32.8	33.0	88.7	27.5
PLLF	**LWR**	0.90	20.09	392.5	2.22E+03	-1.23E+02	1.9	0.5	0.8	32.8	32.9	88.7	27.9
SWF	**BPRM**	0.95	22.32	123.0	1.68E+03	-3.87E+02	1.2	0.2	0.5	18.2	26.9	27.4	0.0
SWF	**BPOM**	0.90	20.30	79.8	1.59E+03	-4.28E+02	1.4	0.1	0.5	13.3	19.9	14.5	0.0
SWF	**BC**	0.90	20.37	71.6	1.64E+03	-3.97E+02	1.2	0.1	0.4	14.0	21.0	14.5	0.0
SWF	**BM**	0.90	20.25	83.4	1.59E+03	-4.26E+02	1.1	0.2	0.4	14.2	21.3	14.5	0.0
SWF	**LDM**	0.90	20.34	74.2	1.61E+03	-4.15E+02	0.9	0.1	0.4	13.2	19.8	14.5	0.0
SWF	**LWR**	0.90	20.31	73.8	1.59E+03	-4.30E+02	1.5	0.1	0.4	12.8	19.2	14.5	0.0
DSWF	**BPRM**	0.95	22.17	105.2	1.67E+03	-3.91E+02	1.6	0.2	0.5	17.2	25.2	25.8	0.3
DSWF	**BPOM**	0.90	20.16	80.4	1.60E+03	-4.22E+02	1.2	0.1	0.4	13.6	20.4	11.3	0.1
DSWF	**BC**	0.90	20.06	82.9	1.66E+03	-3.83E+02	1.5	0.2	0.4	14.2	21.4	11.3	0.1
DSWF	**BM**	0.90	20.09	75.8	1.60E+03	-4.21E+02	1.5	0.1	0.4	14.1	20.8	21.0	0.1
DSWF	**LDM**	0.90	20.18	77.6	1.62E+03	-4.07E+02	1.1	0.2	0.4	13.3	20.3	4.8	0.0
DSWF	**LWR**	0.90	20.04	83.3	1.61E+03	-4.19E+02	1.4	0.1	0.5	13.0	19.5	11.3	0.0

Table A.2: Statistics for simulation of Case Study Two with 16 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.65	30.09	12717.5	6.43E+10	1.82E+07	5356.3	4621.4	5022.8	98.6	99.4	68.3	97.5
FCFS	**BPOM**	0.80	29.47	12713.2	6.23E+10	1.79E+07	5252.0	4549.1	4936.8	98.4	99.4	60.0	97.0
FCFS	**BC**	0.80	29.53	12714.3	6.23E+10	1.79E+07	5251.9	4549.0	4937.1	98.4	99.4	60.0	97.0
FCFS	**BM**	0.80	29.44	12713.4	6.23E+10	1.79E+07	5252.0	4548.6	4936.7	98.4	99.4	60.0	97.0
FCFS	**LDM**	0.80	29.46	12714.4	6.23E+10	1.79E+07	5251.8	4549.0	4937.3	98.4	99.4	60.0	97.0
FCFS	**LWR**	0.80	29.42	12713.7	6.23E+10	1.79E+07	5251.7	4548.8	4937.1	98.4	99.4	60.0	97.0
FCLS	**BPRM**	0.95	41.52	7531.3	2.72E+08	1.90E+05	5546.6	77.0	137.3	84.5	99.4	100.0	54.5
FCLS	**BPOM**	0.90	41.59	5083.1	1.12E+08	1.52E+05	6358.4	76.9	136.0	60.1	62.5	100.0	54.7
FCLS	**BC**	0.90	41.60	5092.6	1.22E+08	1.57E+05	4987.3	76.9	136.0	60.6	63.3	100.0	54.7
FCLS	**BM**	0.90	41.60	4959.3	8.50E+07	1.47E+05	5496.9	76.9	135.4	60.5	63.0	100.0	54.9
FCLS	**LDM**	0.90	41.59	4937.2	5.26E+07	1.39E+05	5535.4	76.6	135.2	60.4	63.0	100.0	54.7
FCLS	**LWR**	0.90	41.54	5087.3	8.19E+07	1.48E+05	4995.0	76.6	135.7	60.6	63.3	100.0	54.6
EDF	**BPRM**	0.90	43.52	12345.6	9.65E+09	6.41E+06	3648.7	2148.5	2959.7	98.1	99.4	100.0	95.5
EDF	**BPOM**	0.90	44.54	12201.9	1.13E+10	7.05E+06	3896.3	2302.9	3172.3	95.9	96.1	100.0	95.4
EDF	**BC**	0.90	44.57	12219.8	1.13E+10	7.06E+06	3893.8	2303.7	3174.2	95.8	96.0	100.0	95.4
EDF	**BM**	0.90	44.49	12205.1	1.13E+10	7.05E+06	3893.9	2304.0	3171.8	95.8	95.9	100.0	95.4
EDF	**LDM**	0.90	44.57	12221.6	1.13E+10	7.06E+06	3892.5	2302.8	3174.5	95.9	96.1	100.0	95.4
EDF	**LWR**	0.90	44.65	12225.9	1.13E+10	7.05E+06	3893.7	2302.2	3168.1	95.8	96.0	100.0	95.4
LLF	**BPRM**	0.90	43.31	12553.9	3.11E+10	1.16E+07	4867.7	3655.8	4442.6	98.5	99.4	100.0	96.8
LLF	**BPOM**	0.80	44.30	12473.1	3.46E+10	1.23E+07	5117.8	3848.8	4671.0	97.4	97.6	100.0	96.9
LLF	**BC**	0.80	44.31	12479.8	3.46E+10	1.23E+07	5116.8	3846.8	4671.1	97.6	97.8	100.0	97.0
LLF	**BM**	0.80	44.27	12467.2	3.46E+10	1.23E+07	5119.5	3847.5	4671.3	97.3	97.4	100.0	96.9
LLF	**LDM**	0.80	44.31	12485.1	3.46E+10	1.23E+07	5117.8	3848.2	4672.4	97.4	97.6	100.0	96.9
LLF	**LWR**	0.80	44.30	12461.2	3.46E+10	1.23E+07	5117.6	3845.4	4671.5	97.3	97.5	100.0	97.0
PLLF	**BPRM**	0.90	43.59	12552.4	8.20E+05	7.91E+04	12.6	11.6	12.2	98.6	99.4	100.0	97.2
PLLF	**BPOM**	0.85	44.53	12418.4	9.27E+05	8.43E+04	13.5	12.6	13.3	97.7	98.0	100.0	97.0
PLLF	**BC**	0.80	44.56	12432.3	9.28E+05	8.43E+04	13.5	12.5	13.3	97.8	98.1	100.0	97.0
PLLF	**BM**	0.85	44.52	12421.6	9.26E+05	8.42E+04	13.5	12.5	13.3	97.7	98.0	100.0	97.1
PLLF	**LDM**	0.85	44.55	12451.8	9.27E+05	8.43E+04	13.5	12.5	13.3	97.7	98.0	100.0	97.0
PLLF	**LWR**	0.85	44.61	12420.7	9.27E+05	8.43E+04	13.5	12.5	13.3	97.8	98.2	100.0	97.0
SWF	**BPRM**	0.95	44.51	4785.0	6.23E+05	1.89E+04	71.8	10.0	37.3	72.6	99.4	100.0	18.8
SWF	**BPOM**	0.95	57.17	1360.6	1.90E+06	3.96E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**BC**	0.95	57.20	1360.6	1.90E+06	3.96E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**BM**	0.95	57.17	1360.6	1.90E+06	3.96E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**LDM**	0.95	57.17	1361.1	1.90E+06	3.96E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**LWR**	0.95	57.17	1361.6	1.90E+06	3.96E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
DSWF	**BPRM**	0.95	44.45	4925.6	1.06E+06	2.31E+04	108.7	4.1	48.9	72.0	99.4	100.0	16.9
DSWF	**BPOM**	0.95	56.87	1329.4	2.88E+06	4.79E+04	134.4	36.5	73.9	10.0	0.0	100.0	28.7
DSWF	**BC**	0.95	56.88	1341.9	3.46E+06	5.47E+04	150.7	41.9	78.9	10.1	0.0	98.3	29.1
DSWF	**BM**	0.95	56.87	1329.4	2.88E+06	4.79E+04	134.4	36.5	73.9	10.0	0.0	100.0	28.7
DSWF	**LDM**	0.95	56.89	1347.2	3.25E+06	5.23E+04	196.5	40.4	76.6	10.1	0.0	100.0	29.1
DSWF	**LWR**	0.95	56.89	1326.9	2.90E+06	4.81E+04	187.8	37.6	73.2	10.0	0.0	100.0	28.6

Table A.3: Statistics for simulation of Case Study Three with 16 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.60	31.53	12727.6	4.84E+10	1.51E+07	5846.0	4513.0	5370.8	98.4	98.7	96.7	97.9
FCFS	**BPOM**	0.80	29.47	12741.7	3.94E+10	1.36E+07	5218.9	4056.7	4825.3	98.5	99.4	90.0	97.0
FCFS	**BC**	0.80	29.53	12742.7	3.94E+10	1.36E+07	5219.6	4056.5	4825.4	98.5	99.4	90.0	97.0
FCFS	**BM**	0.80	29.44	12741.7	3.94E+10	1.35E+07	5218.9	4056.4	4825.2	98.5	99.4	90.0	97.0
FCFS	**LDM**	0.80	29.46	12742.2	3.94E+10	1.36E+07	5219.5	4056.7	4825.6	98.5	99.4	90.0	97.0
FCFS	**LWR**	0.80	29.42	12742.0	3.94E+10	1.35E+07	5218.9	4057.1	4825.5	98.5	99.4	90.0	97.0
FCLS	**BPRM**	0.95	41.52	5455.3	1.34E+08	1.66E+05	5546.6	77.0	137.3	53.5	52.7	100.0	54.5
FCLS	**BPOM**	0.90	41.59	3740.9	5.33E+07	1.41E+05	4639.1	76.9	136.0	39.4	31.4	100.0	54.7
FCLS	**BC**	0.90	41.60	3774.0	5.71E+07	1.45E+05	3618.3	76.9	136.0	39.8	31.9	100.0	54.7
FCLS	**BM**	0.90	41.60	3671.2	5.31E+07	1.39E+05	5496.9	76.9	135.4	39.8	31.8	100.0	54.9
FCLS	**LDM**	0.90	41.59	3668.4	4.52E+07	1.35E+05	5535.4	76.6	135.2	39.3	31.2	100.0	54.7
FCLS	**LWR**	0.90	41.54	3766.2	3.94E+07	1.38E+05	3615.9	76.6	135.7	39.8	32.0	100.0	54.6
EDF	**BPRM**	0.90	37.23	12128.0	2.14E+10	8.77E+06	4821.0	3280.2	4445.7	95.6	95.7	100.0	95.4
EDF	**BPOM**	0.80	37.77	12157.5	2.38E+10	9.35E+06	5076.1	3447.1	4669.6	94.2	93.6	100.0	95.4
EDF	**BC**	0.80	37.78	12167.0	2.38E+10	9.36E+06	5076.2	3449.2	4670.4	94.3	93.8	100.0	95.4
EDF	**BM**	0.80	37.82	12160.0	2.38E+10	9.35E+06	5076.2	3448.3	4670.4	94.2	93.6	100.0	95.4
EDF	**LDM**	0.80	37.80	12173.2	2.38E+10	9.35E+06	5076.6	3448.7	4670.2	94.3	93.7	100.0	95.4
EDF	**LWR**	0.80	37.80	12162.1	2.38E+10	9.35E+06	5076.3	3448.7	4670.2	94.3	93.6	100.0	95.4
LLF	**BPRM**	0.90	36.37	12460.3	2.94E+10	1.11E+07	4906.0	3667.3	4552.0	97.1	97.3	100.0	96.7
LLF	**BPOM**	0.75	36.37	12461.0	2.94E+10	1.11E+07	4906.1	3668.3	4551.1	97.2	97.3	100.0	96.8
LLF	**BC**	0.80	37.14	12451.0	3.24E+10	1.17E+07	5155.5	3838.5	4772.8	96.3	96.0	100.0	97.0
LLF	**BM**	0.75	36.30	12460.1	2.94E+10	1.11E+07	4905.2	3667.2	4553.1	97.2	97.3	100.0	96.8
LLF	**LDM**	0.75	36.35	12484.7	2.95E+10	1.11E+07	4909.1	3669.6	4555.5	97.4	97.6	100.0	96.9
LLF	**LWR**	0.75	36.31	12469.3	2.94E+10	1.11E+07	4906.0	3667.0	4551.6	97.3	97.5	100.0	96.8
PLLF	**BPRM**	0.85	43.26	12432.0	7.98E+05	7.76E+04	12.6	11.5	12.2	97.4	97.5	100.0	97.2
PLLF	**BPOM**	0.75	43.39	12431.0	7.99E+05	7.77E+04	12.6	11.5	12.2	97.4	97.5	100.0	97.1
PLLF	**BC**	0.75	43.56	12452.9	8.05E+05	7.81E+04	13.3	11.6	12.2	97.8	98.1	100.0	97.3
PLLF	**BM**	0.80	44.39	12412.6	9.20E+05	8.39E+04	13.6	12.5	13.3	97.0	97.0	100.0	97.0
PLLF	**LDM**	0.80	44.47	12401.7	9.21E+05	8.40E+04	13.6	12.5	13.3	97.0	96.9	100.0	97.0
PLLF	**LWR**	0.75	43.40	12457.9	7.99E+05	7.77E+04	13.2	11.5	12.2	97.7	98.0	100.0	97.2
SWF	**BPRM**	0.95	44.51	3233.0	6.20E+05	1.65E+04	71.8	10.0	37.3	40.3	50.6	100.0	18.8
SWF	**BPOM**	0.95	57.17	1347.3	1.90E+06	3.80E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**BC**	0.95	57.20	1347.3	1.90E+06	3.80E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**BM**	0.95	57.17	1347.3	1.90E+06	3.80E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**LDM**	0.95	57.17	1347.8	1.90E+06	3.80E+04	93.4	31.0	56.6	10.3	0.0	100.0	29.7
SWF	**LWR**	0.90	44.60	1390.2	6.33E+05	1.55E+04	72.1	10.4	37.8	20.2	20.3	100.0	18.8
DSWF	**BPRM**	0.95	44.82	3179.2	1.00E+06	1.99E+04	109.3	5.6	47.9	39.6	50.6	100.0	16.7
DSWF	**BPOM**	0.95	56.85	1316.8	2.90E+06	4.66E+04	134.5	37.0	74.0	10.0	0.0	100.0	28.6
DSWF	**BC**	0.95	56.91	1331.7	3.15E+06	4.97E+04	144.7	39.0	76.0	10.1	0.0	100.0	28.9
DSWF	**BM**	0.95	56.85	1316.8	2.90E+06	4.66E+04	134.5	37.0	74.0	10.0	0.0	100.0	28.6
DSWF	**LDM**	0.95	56.89	1337.7	3.07E+06	4.87E+04	192.9	38.8	75.0	10.2	0.0	100.0	29.2
DSWF	**LWR**	0.95	56.90	1308.7	2.87E+06	4.60E+04	186.8	36.9	73.1	9.9	0.0	100.0	28.5

Table A.4: Statistics for simulation of Case Study Four with 16 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	38.15	3562.3	1.20E+10	4.47E+06	4572.1	3621.3	4389.3	74.2	78.1	75.4	69.6
FCFS	**BPOM**	0.75	38.22	3562.5	1.21E+10	4.48E+06	4593.5	3634.0	4397.4	74.2	78.1	75.4	69.6
FCFS	**BC**	0.80	38.51	3575.5	1.21E+10	4.48E+06	4569.6	3628.8	4394.2	79.6	88.4	75.4	69.6
FCFS	**BM**	0.75	38.24	3562.9	1.21E+10	4.49E+06	4601.4	3639.0	4404.1	74.2	78.1	75.4	69.6
FCFS	**LDM**	0.75	38.27	3566.0	1.22E+10	4.49E+06	4605.5	3642.3	4410.9	75.9	81.4	75.4	69.6
FCFS	**LWR**	0.75	38.25	3563.3	1.22E+10	4.49E+06	4604.7	3645.8	4405.4	74.5	78.7	75.4	69.6
FCLS	**BPRM**	0.95	39.76	608.3	8.02E+05	9.09E+03	219.5	2.0	66.7	20.9	29.0	86.4	7.8
FCLS	**BPOM**	0.80	37.05	495.3	5.92E+05	7.30E+03	192.7	1.2	58.3	24.4	35.9	84.8	7.7
FCLS	**BC**	0.80	37.21	506.1	5.57E+05	7.20E+03	182.7	1.2	59.4	29.9	46.6	84.8	7.5
FCLS	**BM**	0.80	37.11	485.5	5.89E+05	7.45E+03	182.9	1.4	59.7	24.5	36.0	84.8	7.8
FCLS	**LDM**	0.80	37.29	503.0	5.67E+05	7.24E+03	182.5	1.3	58.3	28.8	44.5	84.8	7.4
FCLS	**LWR**	0.80	37.06	498.7	5.53E+05	7.08E+03	182.5	1.2	58.5	24.6	36.5	84.8	7.5
EDF	**BPRM**	0.90	38.17	2478.8	2.62E+09	1.49E+06	3601.0	1977.3	2969.8	56.6	63.5	83.1	47.1
EDF	**BPOM**	0.80	38.15	2427.0	2.62E+09	1.49E+06	3595.0	1976.2	2972.0	55.8	61.8	83.1	47.2
EDF	**BC**	0.80	38.60	2447.5	2.63E+09	1.50E+06	3601.9	1981.5	2973.5	61.3	72.4	83.1	47.2
EDF	**BM**	0.80	38.20	2419.0	2.64E+09	1.50E+06	3605.6	1979.6	2981.1	55.8	61.9	83.9	47.2
EDF	**LDM**	0.80	38.30	2438.2	2.63E+09	1.50E+06	3599.9	1977.3	2978.3	60.3	70.6	83.1	47.1
EDF	**LWR**	0.80	38.22	2423.9	2.63E+09	1.50E+06	3602.4	1979.0	2972.2	55.8	61.8	83.9	47.2
LLF	**BPRM**	0.90	38.47	3246.7	7.10E+09	3.11E+06	4133.8	2981.4	3824.3	69.5	73.3	99.2	63.5
LLF	**BPOM**	0.80	38.43	3260.3	7.07E+09	3.11E+06	4126.3	2974.0	3819.4	69.8	73.3	99.2	64.1
LLF	**BC**	0.80	38.89	3235.5	7.08E+09	3.11E+06	4126.1	2972.0	3819.9	74.7	83.2	99.2	63.5
LLF	**BM**	0.80	38.43	3255.6	7.07E+09	3.11E+06	4128.2	2969.7	3814.9	69.8	73.3	99.2	64.1
LLF	**LDM**	0.90	39.72	3247.1	8.33E+09	3.37E+06	4483.4	3199.8	4161.7	68.4	70.9	99.2	63.8
LLF	**LWR**	0.80	38.42	3216.4	7.07E+09	3.11E+06	4123.8	2967.4	3822.4	69.1	72.8	99.2	63.2
PLLF	**BPRM**	0.90	38.25	3231.0	2.04E+04	4.00E+03	3.2	2.2	2.4	72.7	76.4	99.2	66.9
PLLF	**BPOM**	0.80	38.28	3183.6	1.99E+04	3.89E+03	3.0	2.2	2.4	72.5	75.8	99.2	67.1
PLLF	**BC**	0.80	38.77	3159.4	1.96E+04	3.88E+03	2.7	2.2	2.3	77.3	85.7	99.2	66.4
PLLF	**BM**	0.80	38.17	3199.0	1.99E+04	3.89E+03	3.0	2.2	2.4	72.4	76.0	99.2	66.9
PLLF	**LDM**	0.80	38.16	3215.2	2.01E+04	3.97E+03	2.6	2.2	2.3	76.9	84.4	99.2	66.9
PLLF	**LWR**	0.80	38.17	3163.1	1.98E+04	3.88E+03	3.0	2.2	2.3	72.2	75.7	99.2	66.5
SWF	**BPRM**	0.95	40.14	311.1	3.63E+03	-1.26E+03	5.7	0.3	2.1	15.6	25.9	77.1	0.1
SWF	**BPOM**	0.80	37.82	198.9	3.48E+03	-1.22E+03	5.4	0.2	1.9	19.6	33.6	72.9	0.4
SWF	**BC**	0.80	38.16	220.1	3.62E+03	-1.13E+03	5.4	0.2	1.9	25.2	44.1	72.9	0.4
SWF	**BM**	0.80	37.75	186.0	3.45E+03	-1.24E+03	5.4	0.1	1.9	19.6	33.4	72.9	0.5
SWF	**LDM**	0.80	37.99	200.2	3.57E+03	-1.16E+03	5.4	0.1	1.9	24.0	41.8	72.9	0.5
SWF	**LWR**	0.80	37.82	202.4	3.51E+03	-1.20E+03	5.4	0.1	1.9	20.1	34.5	72.9	0.4
DSWF	**BPRM**	0.95	40.20	276.2	4.71E+03	-1.19E+03	18.8	0.2	2.9	15.5	25.9	69.5	0.3
DSWF	**BPOM**	0.80	37.53	186.8	4.58E+03	-1.14E+03	26.3	0.1	2.4	19.5	33.6	66.1	0.6
DSWF	**BC**	0.80	38.12	207.6	4.81E+03	-1.03E+03	20.6	0.1	2.6	25.3	44.4	69.5	0.5
DSWF	**BM**	0.80	37.38	182.1	4.95E+03	-1.12E+03	25.7	0.1	2.4	19.6	33.6	67.8	0.7
DSWF	**LDM**	0.80	37.52	201.4	7.00E+03	-9.84E+02	29.1	0.1	2.5	24.0	42.0	66.1	0.7
DSWF	**LWR**	0.80	37.51	194.4	5.10E+03	-1.09E+03	32.3	0.1	2.5	19.9	34.4	66.1	0.4

Table A.5: Statistics for simulation of Case Study One with 32 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	15.20	323.6	1.70E+07	5.21E+04	583.2	243.3	443.0	14.5	15.4	0.0	14.1
FCFS	**BPOM**	0.75	15.20	321.0	1.68E+07	5.16E+04	572.3	241.1	444.8	14.3	15.1	0.0	13.8
FCFS	**BC**	0.75	15.32	335.0	1.70E+07	5.22E+04	570.3	243.2	438.6	35.4	47.5	0.0	14.3
FCFS	**BM**	0.75	15.19	321.3	1.69E+07	5.18E+04	577.6	239.2	444.8	14.3	15.1	0.0	13.8
FCFS	**LDM**	0.80	15.24	340.2	1.72E+07	5.23E+04	590.1	245.4	440.0	43.2	59.6	0.0	14.0
FCFS	**LWR**	0.75	15.21	322.2	1.68E+07	5.17E+04	573.4	239.5	445.6	14.8	15.9	0.0	13.8
FCLS	**BPRM**	0.95	16.15	74.3	3.31E+03	-3.23E+02	21.8	0.1	2.0	7.4	9.5	6.5	3.5
FCLS	**BPOM**	0.90	15.26	52.5	2.07E+03	-4.18E+02	10.0	0.0	1.0	6.0	7.7	3.2	2.6
FCLS	**BC**	0.90	15.32	50.7	2.14E+03	-3.75E+02	10.4	0.0	0.9	5.5	7.0	3.2	2.6
FCLS	**BM**	0.90	15.28	54.0	2.08E+03	-4.19E+02	10.6	0.0	0.9	5.6	7.2	3.2	2.6
FCLS	**LDM**	0.90	15.32	50.6	2.10E+03	-3.97E+02	10.2	0.0	0.8	5.7	7.2	3.2	2.8
FCLS	**LWR**	0.90	15.25	53.2	2.06E+03	-4.18E+02	10.0	0.0	1.0	5.9	7.6	3.2	2.6
EDF	**BPRM**	0.95	16.13	38.6	1.44E+03	-5.10E+02	1.0	0.0	0.3	6.3	9.7	0.0	0.0
EDF	**BPOM**	0.90	15.39	28.2	1.42E+03	-5.23E+02	0.8	0.0	0.2	4.5	6.9	0.0	0.0
EDF	**BC**	0.90	15.45	21.5	1.49E+03	-4.79E+02	0.6	0.0	0.2	4.5	6.9	0.0	0.0
EDF	**BM**	0.90	15.39	33.5	1.43E+03	-5.20E+02	0.9	0.0	0.3	4.9	7.5	0.0	0.0
EDF	**LDM**	0.90	15.47	21.9	1.45E+03	-5.04E+02	0.4	0.0	0.2	3.9	6.1	0.0	0.0
EDF	**LWR**	0.90	15.40	31.1	1.42E+03	-5.21E+02	0.7	0.0	0.2	4.5	7.0	0.0	0.0
LLF	**BPRM**	0.90	15.28	71.8	1.51E+03	-4.72E+02	0.6	0.2	0.3	9.9	14.9	0.0	0.7
LLF	**BPOM**	0.90	15.21	33.4	1.43E+03	-5.11E+02	0.8	0.0	0.3	4.8	7.4	0.0	0.0
LLF	**BC**	0.90	15.29	34.0	1.51E+03	-4.64E+02	0.8	0.0	0.3	4.7	7.2	0.0	0.0
LLF	**BM**	0.90	15.26	31.8	1.43E+03	-5.11E+02	0.9	0.0	0.2	5.5	8.4	0.0	0.0
LLF	**LDM**	0.90	15.29	25.4	1.46E+03	-4.92E+02	0.9	0.0	0.2	4.4	6.8	0.0	0.0
LLF	**LWR**	0.90	15.25	39.8	1.45E+03	-5.06E+02	1.1	0.0	0.3	4.8	7.4	0.0	0.0
PLLF	**BPRM**	0.90	15.20	76.9	1.52E+03	-4.62E+02	0.5	0.2	0.3	10.1	15.2	0.0	0.8
PLLF	**BPOM**	0.90	15.11	30.8	1.44E+03	-5.04E+02	0.8	0.0	0.2	4.9	7.5	0.0	0.0
PLLF	**BC**	0.90	15.16	32.6	1.52E+03	-4.56E+02	0.9	0.0	0.2	4.9	7.6	0.0	0.0
PLLF	**BM**	0.90	15.15	26.5	1.43E+03	-5.06E+02	0.9	0.0	0.2	4.9	7.6	0.0	0.0
PLLF	**LDM**	0.90	15.16	30.5	1.48E+03	-4.79E+02	1.3	0.0	0.2	5.0	7.7	0.0	0.0
PLLF	**LWR**	0.90	15.14	31.6	1.44E+03	-5.03E+02	0.5	0.0	0.2	4.6	7.1	0.0	0.0
SWF	**BPRM**	0.95	16.39	35.7	1.44E+03	-5.12E+02	0.8	0.0	0.3	6.0	9.2	0.0	0.0
SWF	**BPOM**	0.90	15.59	24.8	1.41E+03	-5.25E+02	0.6	0.0	0.2	4.0	6.2	0.0	0.0
SWF	**BC**	0.90	15.67	28.5	1.50E+03	-4.74E+02	1.0	0.0	0.2	4.5	7.0	0.0	0.0
SWF	**BM**	0.90	15.59	32.4	1.43E+03	-5.19E+02	0.9	0.0	0.3	4.7	7.2	0.0	0.0
SWF	**LDM**	0.90	15.67	28.3	1.46E+03	-4.97E+02	0.7	0.0	0.2	4.5	6.9	0.0	0.0
SWF	**LWR**	0.90	15.58	28.2	1.42E+03	-5.20E+02	0.9	0.0	0.2	4.5	7.0	0.0	0.0
DSWF	**BPRM**	0.95	16.33	41.2	1.45E+03	-5.07E+02	0.7	0.0	0.3	5.6	8.6	0.0	0.1
DSWF	**BPOM**	0.90	15.60	20.6	1.41E+03	-5.24E+02	0.7	0.0	0.2	4.2	6.5	0.0	0.0
DSWF	**BC**	0.90	15.58	17.2	1.48E+03	-4.78E+02	0.6	0.0	0.1	4.2	6.5	0.0	0.0
DSWF	**BM**	0.90	15.38	21.8	1.42E+03	-5.23E+02	1.3	0.0	0.1	4.1	6.2	0.0	0.1
DSWF	**LDM**	0.90	15.64	19.9	1.45E+03	-5.01E+02	0.8	0.0	0.1	4.7	7.2	0.0	0.0
DSWF	**LWR**	0.90	15.28	26.0	1.43E+03	-5.19E+02	1.2	0.0	0.2	4.6	7.1	0.0	0.0

Table A.6: Statistics for simulation of Case Study Two with 32 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.65	19.56	6967.1	3.45E+09	3.78E+06	1561.5	1145.4	1439.4	53.6	54.1	1.7	53.5
FCFS	**BPOM**	0.80	19.01	6988.9	3.43E+09	3.76E+06	1595.3	1141.3	1449.5	53.7	54.1	1.7	53.6
FCFS	**BC**	0.80	18.99	6989.7	3.43E+09	3.76E+06	1589.7	1141.0	1454.2	53.7	54.2	1.7	53.6
FCFS	**BM**	0.80	18.97	6989.0	3.43E+09	3.76E+06	1594.3	1141.4	1451.7	53.7	54.1	1.7	53.6
FCFS	**LDM**	0.80	19.00	6989.5	3.43E+09	3.76E+06	1594.0	1141.1	1452.7	53.8	54.3	1.7	53.6
FCFS	**LWR**	0.80	18.95	6990.0	3.43E+09	3.76E+06	1597.0	1140.8	1451.9	54.1	54.7	1.7	53.6
FCLS	**BPRM**	0.95	26.55	2211.2	1.57E+05	2.60E+03	114.2	0.4	7.7	37.3	54.1	51.7	3.5
FCLS	**BPOM**	0.90	26.58	537.3	1.68E+05	1.68E+03	114.4	0.2	7.9	15.6	21.4	51.7	3.5
FCLS	**BC**	0.90	26.62	569.3	1.68E+05	1.78E+03	114.4	0.2	8.0	16.5	22.7	51.7	3.5
FCLS	**BM**	0.90	26.58	549.8	1.68E+05	1.69E+03	114.4	0.1	7.9	15.9	21.8	51.7	3.5
FCLS	**LDM**	0.90	26.58	536.1	1.73E+05	1.79E+03	114.4	0.1	8.0	16.1	22.2	51.7	3.5
FCLS	**LWR**	0.90	26.58	582.6	1.68E+05	1.76E+03	114.3	0.2	8.0	16.3	22.5	51.7	3.5
EDF	**BPRM**	0.90	26.17	1223.8	1.14E+04	-1.31E+03	0.7	0.2	0.4	36.0	54.1	18.3	0.1
EDF	**BPOM**	0.90	26.66	132.0	9.07E+03	-2.50E+03	0.5	0.0	0.2	9.5	14.2	23.3	0.0
EDF	**BC**	0.90	26.78	133.4	9.20E+03	-2.42E+03	0.5	0.0	0.2	9.5	14.2	23.3	0.0
EDF	**BM**	0.90	26.71	123.8	9.05E+03	-2.51E+03	0.4	0.0	0.1	9.0	13.4	23.3	0.0
EDF	**LDM**	0.90	26.83	157.6	9.17E+03	-2.44E+03	0.9	0.0	0.2	9.9	14.7	21.7	0.0
EDF	**LWR**	0.90	26.73	118.6	9.14E+03	-2.46E+03	0.4	0.0	0.1	9.5	14.2	23.3	0.0
LLF	**BPRM**	0.90	26.18	5904.9	2.42E+08	8.52E+05	435.5	335.1	403.7	51.3	54.1	96.7	45.2
LLF	**BPOM**	0.80	26.54	5596.8	3.29E+08	9.87E+05	506.9	392.3	470.7	46.5	47.2	96.7	44.5
LLF	**BC**	0.80	26.53	5629.6	3.35E+08	1.00E+06	508.0	394.5	473.8	46.8	47.6	96.7	44.5
LLF	**BM**	0.80	26.48	5459.9	3.34E+08	9.93E+05	509.7	394.9	473.7	45.3	46.2	96.7	42.9
LLF	**LDM**	0.80	26.45	5714.3	3.29E+08	9.91E+05	504.6	392.3	470.7	47.6	48.4	96.7	45.3
LLF	**LWR**	0.80	26.46	5850.6	3.36E+08	1.01E+06	510.1	394.3	473.0	48.5	49.2	96.7	46.3
PLLF	**BPRM**	0.90	26.36	5588.3	1.91E+04	1.86E+03	1.1	0.7	0.9	52.4	54.1	96.7	48.5
PLLF	**BPOM**	0.85	26.80	4948.6	1.70E+04	1.15E+03	1.4	0.6	0.7	50.3	51.0	96.7	48.4
PLLF	**BC**	0.80	26.90	5251.1	1.74E+04	1.35E+03	1.4	0.5	0.7	50.8	51.3	96.7	49.0
PLLF	**BM**	0.85	26.85	5096.1	1.70E+04	1.17E+03	1.3	0.6	0.7	50.6	51.3	96.7	48.7
PLLF	**LDM**	0.85	26.84	5164.6	1.73E+04	1.27E+03	1.4	0.6	0.8	50.7	51.3	96.7	48.7
PLLF	**LWR**	0.85	26.19	5455.5	1.78E+04	1.47E+03	1.2	0.6	0.8	50.9	51.4	96.7	49.2
SWF	**BPRM**	0.95	26.79	1703.2	1.14E+04	-1.41E+03	0.7	0.3	0.4	36.0	54.1	28.3	0.0
SWF	**BPOM**	0.95	31.16	77.6	8.58E+03	-2.76E+03	1.2	-0.1	0.0	0.4	0.0	78.3	0.1
SWF	**BC**	0.95	31.18	78.0	8.70E+03	-2.69E+03	1.2	0.0	0.0	0.4	0.0	78.3	0.1
SWF	**BM**	0.95	31.16	77.6	8.58E+03	-2.76E+03	1.2	-0.1	0.0	0.4	0.0	78.3	0.1
SWF	**LDM**	0.95	31.22	77.9	8.68E+03	-2.70E+03	1.2	-0.1	0.0	0.4	0.0	78.3	0.1
SWF	**LWR**	0.95	31.22	77.9	8.67E+03	-2.71E+03	1.2	-0.1	0.0	0.4	0.0	78.3	0.1
DSWF	**BPRM**	0.95	26.73	1847.8	1.16E+04	-1.35E+03	0.7	0.3	0.4	36.1	54.1	35.0	0.0
DSWF	**BPOM**	0.95	30.80	90.9	8.68E+03	-2.72E+03	4.5	-0.1	0.0	0.5	0.0	91.7	0.1
DSWF	**BC**	0.95	30.91	92.9	8.80E+03	-2.65E+03	3.9	0.0	0.0	0.5	0.0	88.3	0.2
DSWF	**BM**	0.95	30.80	90.9	8.68E+03	-2.72E+03	4.5	-0.1	0.0	0.5	0.0	91.7	0.1
DSWF	**LDM**	0.95	30.93	92.5	8.79E+03	-2.66E+03	3.8	0.0	0.0	0.5	0.0	85.0	0.2
DSWF	**LWR**	0.95	30.80	92.3	8.74E+03	-2.68E+03	2.9	-0.1	0.0	0.5	0.0	93.3	0.1

Table A.7: Statistics for simulation of Case Study Three with 32 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.60	20.14	6965.4	2.65E+09	3.19E+06	1636.7	1072.6	1426.9	53.7	53.8	61.7	53.5
FCFS	BPOM	0.80	19.01	6995.9	2.18E+09	2.85E+06	1595.3	989.8	1343.9	53.9	54.1	40.0	53.6
FCFS	BC	0.80	18.99	6997.7	2.18E+09	2.85E+06	1589.7	990.0	1340.3	53.9	54.1	40.0	53.6
FCFS	BM	0.80	18.97	6996.2	2.18E+09	2.85E+06	1594.3	990.4	1343.7	53.9	54.1	40.0	53.6
FCFS	LDM	0.80	19.00	6997.8	2.18E+09	2.85E+06	1589.5	990.4	1337.1	53.9	54.1	40.0	53.6
FCFS	LWR	0.80	18.95	6996.6	2.18E+09	2.85E+06	1597.0	990.2	1344.1	53.9	54.2	40.0	53.6
FCLS	BPRM	0.95	26.55	1513.9	1.54E+05	4.27E+02	114.2	0.4	7.7	20.1	28.0	76.7	3.5
FCLS	BPOM	0.90	26.58	374.8	1.65E+05	-3.02E+02	114.4	0.1	7.9	7.8	9.4	76.7	3.5
FCLS	BC	0.90	26.62	385.6	1.65E+05	-2.22E+02	114.4	0.1	8.0	8.4	10.4	76.7	3.5
FCLS	BM	0.90	26.58	379.3	1.65E+05	-2.96E+02	114.4	0.1	7.9	8.0	9.8	76.7	3.5
FCLS	LDM	0.90	26.59	390.0	1.65E+05	-2.63E+02	114.6	0.1	7.9	8.1	9.8	76.7	3.5
FCLS	LWR	0.90	26.58	385.7	1.65E+05	-2.49E+02	114.3	0.1	8.0	8.2	10.0	76.7	3.5
EDF	BPRM	0.90	23.36	4756.5	3.78E+08	9.38E+05	759.2	448.5	651.5	41.9	44.6	73.3	36.1
EDF	BPOM	0.80	23.35	4608.6	4.88E+08	1.08E+06	852.9	509.0	731.7	36.3	36.2	73.3	36.1
EDF	BC	0.80	23.34	4621.8	4.88E+08	1.08E+06	852.4	508.7	731.4	36.4	36.3	73.3	36.2
EDF	BM	0.80	23.35	4607.0	4.88E+08	1.08E+06	852.6	508.5	733.0	36.4	36.3	73.3	36.1
EDF	LDM	0.80	23.34	4601.1	4.88E+08	1.08E+06	851.5	508.2	731.5	36.3	36.2	73.3	36.1
EDF	LWR	0.80	23.35	4587.9	4.89E+08	1.08E+06	853.5	509.2	731.7	36.2	36.1	73.3	36.0
LLF	BPRM	0.90	22.30	6588.7	9.89E+08	1.82E+06	1051.4	684.2	944.7	51.9	52.0	100.0	51.0
LLF	BPOM	0.75	22.29	6604.7	9.89E+08	1.82E+06	1049.9	685.5	946.4	52.0	52.0	100.0	51.2
LLF	BC	0.80	22.57	6579.1	1.15E+09	1.98E+06	1112.0	733.9	1009.5	51.1	50.7	100.0	51.1
LLF	BM	0.75	22.36	6591.5	9.89E+08	1.82E+06	1052.1	684.5	944.9	51.9	52.0	100.0	51.0
LLF	LDM	0.75	22.32	6662.2	1.00E+09	1.83E+06	1055.1	688.3	949.5	60.2	64.5	100.0	51.0
LLF	LWR	0.75	22.31	6629.0	9.93E+08	1.82E+06	1055.5	686.1	947.2	58.9	62.4	100.0	51.2
PLLF	BPRM	0.85	26.61	6345.1	3.56E+04	5.47E+03	2.6	1.8	1.9	52.4	52.4	100.0	51.7
PLLF	BPOM	0.75	26.54	6352.8	3.53E+04	5.42E+03	2.4	1.7	1.9	52.4	52.4	100.0	51.6
PLLF	BC	0.75	26.80	6526.9	3.68E+04	6.30E+03	2.7	1.8	1.9	67.0	74.5	100.0	51.7
PLLF	BM	0.80	26.92	6246.6	3.46E+04	5.23E+03	2.6	1.7	1.9	52.0	51.8	100.0	51.7
PLLF	LDM	0.80	26.88	6282.8	3.47E+04	5.27E+03	2.5	1.7	1.9	51.9	51.6	100.0	51.9
PLLF	LWR	0.75	26.58	6385.6	3.60E+04	5.80E+03	2.5	1.7	1.9	59.3	62.8	100.0	51.8
SWF	BPRM	0.95	26.79	1140.8	8.08E+03	-3.57E+03	2.3	0.3	0.4	18.7	27.6	73.3	0.0
SWF	BPOM	0.95	31.16	83.1	6.37E+03	-4.59E+03	3.3	-0.1	0.0	0.5	0.0	90.0	0.1
SWF	BC	0.95	31.18	83.3	6.46E+03	-4.53E+03	3.3	-0.1	0.0	0.5	0.0	90.0	0.1
SWF	BM	0.95	31.16	83.1	6.37E+03	-4.59E+03	3.3	-0.1	0.0	0.5	0.0	90.0	0.1
SWF	LDM	0.95	31.17	83.5	6.44E+03	-4.55E+03	3.3	-0.1	0.0	0.5	0.0	90.0	0.1
SWF	LWR	0.90	26.85	99.9	6.44E+03	-4.41E+03	2.4	0.0	0.1	3.9	5.4	73.3	0.0
DSWF	BPRM	0.95	26.68	1157.2	8.14E+03	-3.53E+03	1.9	0.3	0.4	18.7	27.6	85.0	0.0
DSWF	BPOM	0.95	30.96	90.5	6.54E+03	-4.54E+03	3.9	-0.1	0.0	0.5	0.0	93.3	0.2
DSWF	BC	0.95	30.79	93.4	6.72E+03	-4.47E+03	4.3	-0.1	0.0	0.5	0.0	93.3	0.2
DSWF	BM	0.95	30.96	90.5	6.54E+03	-4.54E+03	3.9	-0.1	0.0	0.5	0.0	93.3	0.2
DSWF	LDM	0.95	30.77	92.4	6.58E+03	-4.50E+03	5.0	-0.1	0.0	0.5	0.0	95.0	0.1
DSWF	LWR	0.95	30.74	89.9	6.52E+03	-4.52E+03	3.4	-0.1	0.0	0.5	0.0	95.0	0.1

Table A.8: Statistics for simulation of Case Study Four with 32 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.90	25.35	2812.7	4.52E+08	7.72E+05	1138.1	752.9	1016.0	59.0	67.6	14.4	51.3
FCFS	BPOM	0.75	25.36	2823.0	4.55E+08	7.76E+05	1147.1	754.5	1016.1	59.2	67.8	14.4	51.5
FCFS	BC	0.80	25.80	2835.9	4.62E+08	7.82E+05	1161.0	758.8	1020.1	66.0	80.7	15.3	51.5
FCFS	BM	0.75	25.36	2828.3	4.57E+08	7.78E+05	1147.0	756.6	1013.0	59.3	67.9	15.3	51.7
FCFS	LDM	0.75	25.83	2834.6	4.63E+08	7.84E+05	1141.8	760.2	1021.7	60.5	70.1	15.3	51.9
FCFS	LWR	0.75	25.36	2835.3	4.60E+08	7.82E+05	1139.6	758.9	1017.1	60.2	69.4	16.1	51.9
FCLS	BPRM	0.95	25.35	230.3	3.42E+03	-1.44E+03	22.1	0.2	0.9	12.5	21.2	53.4	0.1
FCLS	BPOM	0.80	25.36	218.7	2.87E+03	-1.41E+03	3.4	0.2	0.7	17.1	30.2	50.9	0.1
FCLS	BC	0.80	25.79	229.2	3.22E+03	-1.25E+03	10.5	0.2	0.7	24.4	44.1	50.0	0.1
FCLS	BM	0.80	25.36	216.6	3.01E+03	-1.40E+03	10.5	0.2	0.7	17.4	30.7	50.0	0.2
FCLS	LDM	0.80	25.77	228.4	3.03E+03	-1.30E+03	3.5	0.2	0.7	22.6	40.6	50.0	0.0
FCLS	LWR	0.80	25.36	223.1	3.05E+03	-1.37E+03	10.4	0.2	0.7	18.6	32.7	50.9	0.2
EDF	BPRM	0.90	25.35	234.8	2.73E+03	-1.43E+03	1.3	0.2	0.4	17.4	32.5	11.0	0.0
EDF	BPOM	0.80	25.36	76.3	2.49E+03	-1.55E+03	0.5	0.1	0.2	15.0	28.0	11.0	0.0
EDF	BC	0.80	25.80	97.2	2.73E+03	-1.39E+03	0.4	0.1	0.2	23.1	43.4	11.0	0.0
EDF	BM	0.80	25.36	76.1	2.49E+03	-1.55E+03	1.0	0.1	0.2	15.4	28.9	11.0	0.0
EDF	LDM	0.80	25.73	95.8	2.66E+03	-1.43E+03	0.7	0.1	0.2	20.9	39.2	11.9	0.0
EDF	LWR	0.80	25.36	83.6	2.54E+03	-1.51E+03	0.6	0.1	0.2	16.4	30.6	11.9	0.0
LLF	BPRM	0.90	25.36	1943.8	2.65E+07	1.52E+05	309.1	200.6	284.6	46.1	56.5	77.1	32.2
LLF	BPOM	0.80	25.36	1882.9	2.62E+07	1.51E+05	316.4	198.2	281.3	45.4	55.1	76.3	32.3
LLF	BC	0.80	25.80	1918.0	2.76E+07	1.56E+05	316.6	204.3	288.9	52.6	68.6	78.0	32.5
LLF	BM	0.80	25.36	1882.4	2.60E+07	1.50E+05	306.5	198.2	282.5	45.3	55.1	75.4	32.2
LLF	LDM	0.90	25.48	2141.2	5.87E+07	2.39E+05	436.7	291.7	406.5	48.3	55.0	82.2	38.7
LLF	LWR	0.80	25.36	1897.0	2.68E+07	1.53E+05	311.9	200.9	283.8	46.4	56.9	77.1	32.4
PLLF	BPRM	0.90	25.36	1112.7	4.91E+03	-2.88E+02	1.6	0.5	0.7	47.8	57.5	78.0	34.9
PLLF	BPOM	0.80	25.36	949.0	4.60E+03	-4.01E+02	0.9	0.4	0.6	46.9	55.9	78.0	34.7
PLLF	BC	0.80	25.81	1019.1	4.91E+03	-2.15E+02	1.0	0.4	0.6	54.9	70.6	79.7	35.3
PLLF	BM	0.80	25.36	960.6	4.61E+03	-3.96E+02	1.0	0.4	0.6	47.5	56.5	78.8	35.2
PLLF	LDM	0.80	25.74	1019.2	4.82E+03	-2.71E+02	0.9	0.4	0.6	52.4	66.1	79.7	34.8
PLLF	LWR	0.80	25.35	1000.3	4.70E+03	-3.46E+02	1.1	0.4	0.6	48.4	58.1	78.0	35.5
SWF	BPRM	0.95	25.36	120.9	2.40E+03	-1.64E+03	1.2	0.1	0.4	9.3	16.4	28.8	0.0
SWF	BPOM	0.80	25.36	101.5	2.51E+03	-1.55E+03	0.9	0.1	0.3	15.8	28.9	21.2	0.1
SWF	BC	0.80	25.79	123.9	2.75E+03	-1.39E+03	0.9	0.1	0.3	22.8	42.3	22.0	0.0
SWF	BM	0.80	25.36	96.9	2.50E+03	-1.56E+03	0.9	0.1	0.2	15.3	28.1	21.2	0.0
SWF	LDM	0.80	25.74	120.7	2.68E+03	-1.44E+03	0.9	0.1	0.3	21.2	39.2	22.0	0.0
SWF	LWR	0.80	25.36	107.3	2.56E+03	-1.51E+03	0.9	0.1	0.3	16.6	30.5	22.0	0.1
DSWF	BPRM	0.95	25.36	134.1	2.47E+03	-1.62E+03	1.9	0.1	0.5	9.8	17.0	34.8	0.0
DSWF	BPOM	0.80	25.36	112.3	2.60E+03	-1.53E+03	4.4	0.1	0.3	15.4	28.0	28.8	0.1
DSWF	BC	0.80	25.81	139.0	2.81E+03	-1.36E+03	1.8	0.1	0.3	23.4	43.1	30.5	0.0
DSWF	BM	0.80	25.36	116.3	2.57E+03	-1.53E+03	1.4	0.1	0.3	15.9	28.9	27.1	0.1
DSWF	LDM	0.80	25.73	119.9	2.72E+03	-1.42E+03	1.4	0.1	0.3	20.9	38.4	28.0	0.0
DSWF	LWR	0.80	25.36	118.3	2.82E+03	-1.47E+03	9.9	0.1	0.3	17.0	30.8	30.5	0.1

Table A.9: Statistics for simulation of Case Study One with 64 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.90	13.47	113.2	3.76E+05	4.24E+03	147.6	0.2	80.9	5.3	5.8	0.0	4.7
FCFS	BPOM	0.75	13.46	113.9	3.97E+05	4.44E+03	133.8	0.2	82.0	5.2	5.6	0.0	4.7
FCFS	BC	0.75	13.68	132.5	4.80E+05	5.19E+03	153.3	0.2	91.1	29.2	42.8	0.0	4.6
FCFS	BM	0.75	13.46	114.9	4.11E+05	4.54E+03	134.3	0.2	84.6	5.2	5.7	0.0	4.7
FCFS	LDM	0.80	13.53	141.9	5.00E+05	5.45E+03	154.2	0.4	87.6	37.6	55.5	0.0	5.0
FCFS	LWR	0.75	13.46	117.7	4.22E+05	4.61E+03	139.3	0.2	85.8	6.2	7.1	0.0	4.8
FCLS	BPRM	0.95	13.76	20.2	1.39E+03	-5.41E+02	1.7	-0.1	0.1	2.1	3.0	1.6	0.4
FCLS	BPOM	0.90	13.44	13.0	1.37E+03	-5.52E+02	0.8	-0.1	0.0	1.5	2.3	0.0	0.0
FCLS	BC	0.90	13.47	12.6	1.45E+03	-4.99E+02	1.0	0.0	0.1	1.7	2.6	0.0	0.0
FCLS	BM	0.90	13.44	13.4	1.37E+03	-5.52E+02	0.9	-0.1	0.0	1.3	2.0	0.0	0.0
FCLS	LDM	0.90	13.47	12.4	1.41E+03	-5.26E+02	0.4	-0.1	0.0	1.1	1.7	0.0	0.0
FCLS	LWR	0.90	13.43	14.3	1.37E+03	-5.50E+02	0.7	-0.1	0.0	1.5	2.3	0.0	0.0
EDF	BPRM	0.95	13.84	19.3	1.38E+03	-5.50E+02	0.7	-0.1	0.2	2.3	3.6	0.0	0.0
EDF	BPOM	0.90	13.43	10.2	1.36E+03	-5.59E+02	0.4	-0.1	0.0	0.8	1.2	0.0	0.0
EDF	BC	0.90	13.49	7.8	1.44E+03	-5.07E+02	0.2	0.0	0.0	0.9	1.3	0.0	0.0
EDF	BM	0.90	13.43	10.5	1.36E+03	-5.58E+02	0.6	-0.1	0.0	1.0	1.5	0.0	0.0
EDF	LDM	0.90	13.50	10.2	1.40E+03	-5.31E+02	0.6	-0.1	0.0	1.1	1.7	0.0	0.0
EDF	LWR	0.90	13.46	8.8	1.36E+03	-5.58E+02	0.4	-0.1	0.0	1.0	1.5	0.0	0.0
LLF	BPRM	0.90	13.44	24.2	1.39E+03	-5.40E+02	0.4	-0.1	0.2	3.7	5.5	0.0	0.4
LLF	BPOM	0.90	13.42	11.1	1.36E+03	-5.57E+02	0.5	-0.1	0.0	0.9	1.4	0.0	0.0
LLF	BC	0.90	13.52	8.7	1.44E+03	-5.05E+02	0.2	0.0	0.0	1.3	1.9	0.0	0.0
LLF	BM	0.90	13.43	11.9	1.36E+03	-5.57E+02	0.6	-0.1	0.0	1.1	1.7	0.0	0.0
LLF	LDM	0.90	13.50	9.6	1.40E+03	-5.31E+02	0.4	-0.1	0.0	1.2	1.8	0.0	0.0
LLF	LWR	0.90	13.45	8.6	1.36E+03	-5.57E+02	0.3	-0.1	0.0	1.1	1.7	0.0	0.0
PLLF	BPRM	0.90	13.32	26.1	1.40E+03	-5.36E+02	0.3	-0.1	0.2	3.9	5.9	0.0	0.3
PLLF	BPOM	0.90	13.35	10.6	1.36E+03	-5.54E+02	0.4	-0.1	0.0	1.2	1.9	0.0	0.0
PLLF	BC	0.90	13.40	8.4	1.44E+03	-5.03E+02	0.2	0.0	0.0	1.3	2.0	0.0	0.0
PLLF	BM	0.90	13.40	12.0	1.36E+03	-5.54E+02	0.6	-0.1	0.0	1.3	2.1	0.0	0.0
PLLF	LDM	0.90	13.44	10.7	1.40E+03	-5.28E+02	0.6	-0.1	0.0	1.3	1.9	0.0	0.0
PLLF	LWR	0.90	13.33	11.0	1.37E+03	-5.53E+02	0.4	-0.1	0.0	1.3	1.9	0.0	0.0
SWF	BPRM	0.95	13.93	19.3	1.38E+03	-5.50E+02	0.5	-0.1	0.1	2.0	3.1	0.0	0.0
SWF	BPOM	0.90	13.45	9.9	1.36E+03	-5.59E+02	0.4	-0.1	0.0	0.8	1.2	0.0	0.0
SWF	BC	0.90	13.51	7.8	1.44E+03	-5.06E+02	0.1	0.0	0.0	1.1	1.7	0.0	0.0
SWF	BM	0.90	13.45	10.6	1.36E+03	-5.58E+02	0.6	-0.1	0.0	1.3	1.9	0.0	0.0
SWF	LDM	0.90	13.51	8.5	1.40E+03	-5.33E+02	0.2	-0.1	0.0	1.2	1.8	0.0	0.0
SWF	LWR	0.90	13.45	8.3	1.36E+03	-5.57E+02	0.2	-0.1	0.0	0.9	1.5	0.0	0.0
DSWF	BPRM	0.95	13.80	14.3	1.37E+03	-5.52E+02	0.6	-0.1	0.1	1.7	2.6	0.0	0.1
DSWF	BPOM	0.90	13.40	9.8	1.36E+03	-5.58E+02	0.4	-0.1	0.0	1.2	1.8	0.0	0.0
DSWF	BC	0.90	13.51	8.0	1.44E+03	-5.06E+02	0.2	0.0	0.0	1.0	1.5	0.0	0.0
DSWF	BM	0.90	13.41	11.8	1.36E+03	-5.56E+02	0.5	-0.1	0.0	1.1	1.7	0.0	0.0
DSWF	LDM	0.90	13.49	7.8	1.40E+03	-5.32E+02	0.2	-0.1	0.0	1.0	1.5	0.0	0.0
DSWF	LWR	0.90	13.33	8.5	1.36E+03	-5.57E+02	0.4	-0.1	0.0	1.1	1.7	0.0	0.0

Table A.10: Statistics for simulation of Case Study Two with 64 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.65	16.01	2458.4	1.24E+08	3.91E+05	556.5	286.8	430.9	19.2	19.8	0.0	18.3
FCFS	**BPOM**	0.80	15.51	2489.1	1.46E+08	4.30E+05	587.9	308.4	472.5	19.1	19.5	0.0	18.6
FCFS	**BC**	0.80	15.56	2492.1	1.46E+08	4.30E+05	589.8	308.4	470.3	19.2	19.7	0.0	18.6
FCFS	**BM**	0.80	15.51	2488.6	1.46E+08	4.30E+05	592.7	307.6	471.9	19.1	19.5	0.0	18.5
FCFS	**LDM**	0.80	15.52	2490.1	1.47E+08	4.31E+05	593.2	308.4	473.3	19.1	19.6	0.0	18.6
FCFS	**LWR**	0.80	15.52	2502.2	1.45E+08	4.29E+05	590.8	307.4	471.4	25.1	28.5	0.0	18.5
FCLS	**BPRM**	0.95	16.68	644.5	1.05E+04	-2.34E+03	9.1	0.2	0.4	12.1	17.6	0.0	1.1
FCLS	**BPOM**	0.90	16.68	155.1	9.80E+03	-2.68E+03	9.1	0.0	0.2	4.2	5.7	0.0	1.1
FCLS	**BC**	0.90	16.67	162.5	1.03E+04	-2.40E+03	9.2	0.0	0.2	4.3	5.9	0.0	1.1
FCLS	**BM**	0.90	16.67	152.0	9.80E+03	-2.68E+03	9.1	0.0	0.2	4.3	5.9	0.0	1.1
FCLS	**LDM**	0.90	16.70	163.4	9.95E+03	-2.59E+03	9.1	0.0	0.2	4.2	5.8	0.0	1.1
FCLS	**LWR**	0.90	16.67	153.3	1.01E+04	-2.56E+03	9.2	0.0	0.2	3.9	5.3	0.0	1.1
EDF	**BPRM**	0.90	16.32	331.4	8.99E+03	-2.58E+03	0.7	0.1	0.2	11.6	17.5	0.0	0.0
EDF	**BPOM**	0.90	16.67	57.8	8.30E+03	-2.94E+03	0.3	0.0	0.0	2.4	3.6	0.0	0.0
EDF	**BC**	0.90	16.70	61.6	8.73E+03	-2.68E+03	0.4	0.0	0.1	2.3	3.5	0.0	0.0
EDF	**BM**	0.90	16.64	58.4	8.31E+03	-2.94E+03	0.3	0.0	0.1	2.5	3.8	0.0	0.0
EDF	**LDM**	0.90	16.69	57.8	8.43E+03	-2.86E+03	0.3	0.0	0.1	2.3	3.5	0.0	0.0
EDF	**LWR**	0.90	16.69	59.8	8.50E+03	-2.83E+03	0.4	0.0	0.1	2.3	3.5	0.0	0.0
LLF	**BPRM**	0.90	16.21	396.2	9.09E+03	-2.53E+03	0.5	0.2	0.3	12.3	18.6	0.0	0.0
LLF	**BPOM**	0.80	16.51	53.8	8.32E+03	-2.93E+03	0.3	0.0	0.0	2.8	4.2	0.0	0.0
LLF	**BC**	0.80	16.52	59.0	8.76E+03	-2.66E+03	0.3	0.0	0.1	3.1	4.6	0.0	0.0
LLF	**BM**	0.80	16.51	56.6	8.33E+03	-2.93E+03	0.3	0.0	0.1	3.0	4.5	0.0	0.0
LLF	**LDM**	0.80	16.50	55.8	8.45E+03	-2.84E+03	0.3	0.0	0.1	2.9	4.4	0.0	0.0
LLF	**LWR**	0.80	16.52	68.1	8.93E+03	-2.60E+03	0.3	0.0	0.1	8.8	13.2	0.0	0.0
PLLF	**BPRM**	0.90	16.11	408.9	9.11E+03	-2.52E+03	0.6	0.2	0.3	12.6	18.9	0.0	0.0
PLLF	**BPOM**	0.85	16.40	56.6	8.34E+03	-2.91E+03	0.3	0.0	0.1	3.2	4.8	0.0	0.0
PLLF	**BC**	0.80	16.40	57.8	8.78E+03	-2.64E+03	0.3	0.0	0.1	3.2	4.8	0.0	0.0
PLLF	**BM**	0.85	16.32	55.8	8.34E+03	-2.91E+03	0.4	0.0	0.1	3.0	4.5	0.0	0.0
PLLF	**LDM**	0.85	16.42	60.4	8.48E+03	-2.82E+03	0.4	0.0	0.1	3.2	4.8	0.0	0.0
PLLF	**LWR**	0.85	16.29	58.4	8.54E+03	-2.80E+03	0.4	0.0	0.1	3.2	4.8	0.0	0.0
SWF	**BPRM**	0.95	16.82	476.0	8.98E+03	-2.63E+03	0.5	0.2	0.3	11.1	16.7	0.0	0.0
SWF	**BPOM**	0.95	19.16	41.0	8.20E+03	-2.99E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**BC**	0.95	19.18	42.9	8.57E+03	-2.77E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**BM**	0.95	19.16	41.0	8.20E+03	-2.99E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**LDM**	0.95	19.21	42.2	8.45E+03	-2.85E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**LWR**	0.95	19.21	41.9	8.38E+03	-2.89E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BPRM**	0.95	16.56	526.1	9.06E+03	-2.59E+03	0.6	0.2	0.3	11.8	17.9	0.0	0.0
DSWF	**BPOM**	0.95	19.07	41.0	8.21E+03	-2.98E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BC**	0.95	18.95	42.9	8.58E+03	-2.76E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BM**	0.95	19.07	41.0	8.21E+03	-2.98E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**LDM**	0.95	19.12	42.3	8.45E+03	-2.84E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**LWR**	0.95	19.07	41.9	8.39E+03	-2.88E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0

Table A.11: Statistics for simulation of Case Study Three with 64 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.60	16.39	2562.6	8.38E+07	3.14E+05	508.3	203.8	386.9	20.0	20.4	3.3	19.3
FCFS	**BPOM**	0.80	15.51	2476.6	9.34E+07	3.25E+05	584.4	206.3	413.4	19.1	19.5	1.7	18.6
FCFS	**BC**	0.80	15.56	2479.1	9.33E+07	3.25E+05	589.8	206.3	412.5	19.1	19.5	1.7	18.6
FCFS	**BM**	0.80	15.51	2476.4	9.36E+07	3.25E+05	592.7	206.2	412.5	19.1	19.5	1.7	18.5
FCFS	**LDM**	0.80	15.51	2477.7	9.36E+07	3.25E+05	588.6	205.8	415.4	19.1	19.5	1.7	18.6
FCFS	**LWR**	0.80	15.52	2478.8	9.30E+07	3.24E+05	580.3	206.1	411.1	19.7	20.4	1.7	18.5
FCLS	**BPRM**	0.95	16.68	446.2	7.74E+03	-4.30E+03	9.1	0.2	0.4	6.5	9.1	23.3	1.1
FCLS	**BPOM**	0.90	16.68	107.0	7.20E+03	-4.57E+03	9.1	-0.1	0.1	2.0	2.3	23.3	1.1
FCLS	**BC**	0.90	16.67	117.5	7.58E+03	-4.35E+03	9.2	0.0	0.1	2.3	2.7	23.3	1.1
FCLS	**BM**	0.90	16.67	108.5	7.21E+03	-4.57E+03	9.1	-0.1	0.1	2.1	2.5	23.3	1.1
FCLS	**LDM**	0.90	16.69	109.2	7.33E+03	-4.50E+03	8.9	-0.1	0.1	2.2	2.7	23.3	1.1
FCLS	**LWR**	0.90	16.67	111.9	7.39E+03	-4.49E+03	9.2	-0.1	0.1	2.0	2.2	23.3	1.1
EDF	**BPRM**	0.90	16.50	151.2	6.21E+03	-4.54E+03	0.5	0.1	0.2	6.1	9.2	0.0	0.0
EDF	**BPOM**	0.80	16.69	31.1	5.71E+03	-4.84E+03	0.2	-0.1	0.0	0.6	0.9	0.0	0.0
EDF	**BC**	0.80	16.84	32.9	6.02E+03	-4.62E+03	0.3	-0.1	0.0	0.6	0.9	0.0	0.0
EDF	**BM**	0.80	16.77	32.1	5.72E+03	-4.84E+03	0.3	-0.1	0.0	0.7	1.0	0.0	0.0
EDF	**LDM**	0.80	16.80	30.9	5.79E+03	-4.77E+03	0.2	-0.1	0.0	0.6	0.8	0.0	0.0
EDF	**LWR**	0.80	16.69	32.9	5.99E+03	-4.63E+03	0.2	0.0	0.0	1.1	1.6	0.0	0.0
LLF	**BPRM**	0.90	16.24	185.9	6.30E+03	-4.48E+03	0.7	0.1	0.2	6.5	9.8	0.0	0.0
LLF	**BPOM**	0.75	16.20	175.3	6.28E+03	-4.49E+03	0.4	0.1	0.2	6.4	9.7	0.0	0.0
LLF	**BC**	0.80	16.47	33.9	6.06E+03	-4.58E+03	0.3	-0.1	0.0	0.8	1.2	0.0	0.0
LLF	**BM**	0.75	16.16	174.1	6.28E+03	-4.49E+03	0.4	0.1	0.2	6.4	9.7	0.0	0.0
LLF	**LDM**	0.75	16.35	202.8	6.75E+03	-4.13E+03	0.5	0.1	0.2	8.9	13.4	0.0	0.0
LLF	**LWR**	0.75	16.25	188.3	6.71E+03	-4.18E+03	0.4	0.1	0.2	8.1	12.2	0.0	0.0
PLLF	**BPRM**	0.85	16.15	305.8	6.94E+03	-4.07E+03	0.9	0.1	0.3	8.0	10.2	0.0	3.6
PLLF	**BPOM**	0.75	16.08	308.2	6.94E+03	-4.07E+03	0.6	0.1	0.3	8.0	10.2	0.0	3.6
PLLF	**BC**	0.75	16.24	606.6	9.39E+03	-2.48E+03	0.7	0.2	0.3	34.2	49.7	0.0	3.7
PLLF	**BM**	0.80	16.35	77.8	6.65E+03	-4.20E+03	0.5	0.0	0.1	4.7	6.5	0.0	1.2
PLLF	**LDM**	0.80	16.31	102.8	6.78E+03	-4.11E+03	0.7	0.0	0.1	5.1	7.1	0.0	1.3
PLLF	**LWR**	0.75	16.11	303.3	7.36E+03	-3.77E+03	0.6	0.1	0.2	9.4	12.4	0.0	3.6
SWF	**BPRM**	0.95	16.82	322.8	6.22E+03	-4.58E+03	0.5	0.1	0.3	5.8	8.6	6.7	0.0
SWF	**BPOM**	0.95	19.16	34.7	5.70E+03	-4.85E+03	0.6	-0.1	-0.1	0.1	0.0	28.3	0.0
SWF	**BC**	0.95	19.18	36.0	5.95E+03	-4.68E+03	0.6	-0.1	0.0	0.1	0.0	26.7	0.0
SWF	**BM**	0.95	19.16	34.7	5.70E+03	-4.85E+03	0.6	-0.1	-0.1	0.1	0.0	28.3	0.0
SWF	**LDM**	0.95	19.17	35.5	5.85E+03	-4.75E+03	0.6	-0.1	0.0	0.1	0.0	26.7	0.0
SWF	**LWR**	0.90	16.79	33.4	5.84E+03	-4.75E+03	0.3	-0.1	0.0	0.8	1.2	6.7	0.0
DSWF	**BPRM**	0.95	16.66	332.5	6.26E+03	-4.55E+03	0.6	0.1	0.3	6.0	9.1	0.0	0.0
DSWF	**BPOM**	0.95	19.03	34.5	5.71E+03	-4.84E+03	0.5	-0.1	-0.1	0.2	0.0	41.7	0.0
DSWF	**BC**	0.95	18.97	35.0	5.96E+03	-4.67E+03	0.4	-0.1	0.0	0.2	0.0	35.0	0.0
DSWF	**BM**	0.95	19.03	34.5	5.71E+03	-4.84E+03	0.5	-0.1	-0.1	0.2	0.0	41.7	0.0
DSWF	**LDM**	0.95	19.05	37.2	5.86E+03	-4.73E+03	0.4	-0.1	0.0	0.2	0.0	48.3	0.0
DSWF	**LWR**	0.95	19.14	35.4	5.81E+03	-4.77E+03	0.4	-0.1	-0.1	0.2	0.0	36.7	0.0

Table A.12: Statistics for simulation of Case Study Four with 64 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	25.34	270.5	2.29E+05	3.69E+03	76.5	0.6	39.5	6.3	9.3	0.0	3.2
FCFS	**BPOM**	0.75	25.34	267.0	2.48E+05	3.92E+03	78.9	0.4	42.4	6.2	8.8	0.0	3.4
FCFS	**BC**	0.80	25.79	359.3	5.80E+05	8.32E+03	111.2	3.1	63.9	22.8	39.0	0.0	5.2
FCFS	**BM**	0.75	25.34	278.1	2.46E+05	3.89E+03	84.9	0.7	41.3	6.4	9.4	0.0	3.4
FCFS	**LDM**	0.75	25.79	341.9	5.05E+05	7.45E+03	101.0	2.5	61.2	12.0	18.7	0.0	4.9
FCFS	**LWR**	0.75	25.34	301.4	3.20E+05	5.07E+03	87.6	0.8	46.7	13.0	21.3	0.0	3.9
FCLS	**BPRM**	0.95	25.34	28.4	1.98E+03	-1.91E+03	1.1	-0.1	0.1	1.7	3.1	2.5	0.0
FCLS	**BPOM**	0.80	25.34	27.9	1.99E+03	-1.90E+03	0.7	-0.1	0.1	2.8	5.3	0.0	0.0
FCLS	**BC**	0.80	25.79	59.8	2.53E+03	-1.52E+03	0.8	0.1	0.1	17.9	34.1	0.0	0.0
FCLS	**BM**	0.80	25.34	33.0	2.00E+03	-1.89E+03	1.0	-0.1	0.1	2.9	5.5	0.0	0.0
FCLS	**LDM**	0.80	25.75	47.2	2.29E+03	-1.69E+03	0.6	0.1	0.1	11.0	21.0	0.0	0.0
FCLS	**LWR**	0.80	25.34	34.9	2.14E+03	-1.79E+03	0.8	0.0	0.1	5.5	10.4	0.0	0.0
EDF	**BPRM**	0.90	25.34	29.4	1.98E+03	-1.90E+03	0.5	-0.1	0.1	2.5	4.7	0.0	0.0
EDF	**BPOM**	0.80	25.34	15.1	1.96E+03	-1.92E+03	0.2	-0.1	0.1	2.1	3.9	0.0	0.0
EDF	**BC**	0.80	25.80	44.9	2.50E+03	-1.54E+03	0.3	0.1	0.1	17.4	33.1	0.0	0.0
EDF	**BM**	0.80	25.34	17.2	1.97E+03	-1.91E+03	0.3	-0.1	0.1	2.3	4.3	0.0	0.0
EDF	**LDM**	0.80	25.73	36.5	2.26E+03	-1.71E+03	0.5	0.1	0.1	10.3	19.5	0.0	0.0
EDF	**LWR**	0.80	25.34	22.9	2.12E+03	-1.81E+03	0.3	0.0	0.1	5.6	10.6	0.0	0.0
LLF	**BPRM**	0.90	25.34	35.6	2.00E+03	-1.89E+03	0.5	-0.1	0.2	2.8	5.3	0.0	0.0
LLF	**BPOM**	0.80	25.34	22.1	1.98E+03	-1.91E+03	0.6	-0.1	0.1	2.4	4.5	0.0	0.0
LLF	**BC**	0.80	25.80	47.0	2.50E+03	-1.54E+03	0.3	0.1	0.1	17.0	32.3	0.0	0.0
LLF	**BM**	0.80	25.34	21.3	1.98E+03	-1.91E+03	0.3	-0.1	0.1	2.5	4.7	0.0	0.0
LLF	**LDM**	0.90	25.48	15.1	2.00E+03	-1.88E+03	0.4	-0.1	0.0	1.6	3.0	0.0	0.0
LLF	**LWR**	0.80	25.34	25.0	2.12E+03	-1.80E+03	0.3	0.0	0.1	5.8	11.0	0.0	0.0
PLLF	**BPRM**	0.90	25.34	36.5	2.00E+03	-1.89E+03	0.4	-0.1	0.2	2.8	5.4	0.0	0.0
PLLF	**BPOM**	0.80	25.34	19.3	1.98E+03	-1.90E+03	0.4	-0.1	0.1	2.4	4.6	0.0	0.0
PLLF	**BC**	0.80	25.81	49.5	2.52E+03	-1.53E+03	0.3	0.1	0.1	17.8	33.9	0.0	0.0
PLLF	**BM**	0.80	25.34	20.6	1.98E+03	-1.90E+03	0.4	-0.1	0.1	2.5	4.8	0.0	0.0
PLLF	**LDM**	0.80	25.79	38.4	2.27E+03	-1.69E+03	0.4	0.1	0.1	10.7	20.4	0.0	0.0
PLLF	**LWR**	0.80	25.34	27.3	2.13E+03	-1.80E+03	0.5	0.0	0.1	5.8	11.0	0.0	0.0
SWF	**BPRM**	0.95	25.34	14.4	1.94E+03	-1.93E+03	0.7	-0.1	0.0	1.1	2.1	0.0	0.0
SWF	**BPOM**	0.80	25.34	18.6	1.97E+03	-1.91E+03	0.5	-0.1	0.1	2.1	4.0	0.0	0.0
SWF	**BC**	0.80	25.79	46.9	2.50E+03	-1.54E+03	0.5	0.1	0.1	17.0	32.4	0.0	0.0
SWF	**BM**	0.80	25.34	19.2	1.97E+03	-1.91E+03	0.3	-0.1	0.1	2.4	4.6	0.0	0.0
SWF	**LDM**	0.80	25.73	34.9	2.26E+03	-1.70E+03	0.3	0.1	0.1	10.7	20.3	0.0	0.0
SWF	**LWR**	0.80	25.34	21.8	2.11E+03	-1.81E+03	0.6	0.0	0.1	5.2	10.0	0.0	0.0
DSWF	**BPRM**	0.95	25.34	15.2	1.94E+03	-1.93E+03	0.3	-0.1	0.0	1.1	2.1	0.0	0.0
DSWF	**BPOM**	0.80	25.34	16.2	1.97E+03	-1.91E+03	0.5	-0.1	0.1	2.2	4.2	0.0	0.0
DSWF	**BC**	0.80	25.79	45.4	2.51E+03	-1.54E+03	0.6	0.1	0.1	17.9	34.0	0.0	0.0
DSWF	**BM**	0.80	25.34	17.8	1.97E+03	-1.91E+03	0.5	-0.1	0.1	2.1	4.0	0.0	0.0
DSWF	**LDM**	0.80	25.73	36.3	2.26E+03	-1.70E+03	0.3	0.1	0.1	10.7	20.3	0.0	0.0
DSWF	**LWR**	0.80	25.34	22.2	2.12E+03	-1.80E+03	0.3	0.0	0.1	5.6	10.7	0.0	0.0

Table A.13: Statistics for simulation of Case Study One with 128 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	12.86	6.7	1.34E+03	-5.70E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**BPOM**	0.75	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**BC**	0.75	13.08	28.8	1.72E+03	-3.43E+02	0.2	0.1	0.1	26.6	41.0	0.0	0.0
FCFS	**BM**	0.75	12.86	6.7	1.34E+03	-5.70E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**LDM**	0.80	12.93	27.8	1.65E+03	-4.06E+02	0.1	0.1	0.1	33.2	51.1	0.0	0.0
FCFS	**LWR**	0.75	12.86	13.5	1.36E+03	-5.59E+02	0.2	-0.1	0.1	2.0	3.1	0.0	0.0
FCLS	**BPRM**	0.95	12.86	7.5	1.34E+03	-5.69E+02	0.2	-0.1	-0.1	0.3	0.5	0.0	0.0
FCLS	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCLS	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
FCLS	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCLS	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
FCLS	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
EDF	**BPRM**	0.95	12.86	6.8	1.34E+03	-5.70E+02	0.1	-0.1	-0.1	0.1	0.2	0.0	0.0
EDF	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
EDF	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
EDF	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
EDF	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
EDF	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
LLF	**BPRM**	0.90	12.86	6.7	1.34E+03	-5.70E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
LLF	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
LLF	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
PLLF	**BPRM**	0.90	12.86	6.7	1.34E+03	-5.70E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
PLLF	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
PLLF	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**BPRM**	0.95	12.86	6.8	1.34E+03	-5.70E+02	0.1	-0.1	-0.1	0.2	0.3	0.0	0.0
SWF	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BPRM**	0.95	12.86	6.9	1.34E+03	-5.70E+02	0.1	-0.1	-0.1	0.2	0.3	0.0	0.0
DSWF	**BPOM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**BC**	0.90	12.92	7.2	1.43E+03	-5.14E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BM**	0.90	12.86	6.7	1.34E+03	-5.71E+02	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**LDM**	0.90	12.91	6.9	1.39E+03	-5.42E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**LWR**	0.90	12.86	6.7	1.34E+03	-5.68E+02	0.0	-0.1	0.0	0.0	0.0	0.0	0.0

Table A.14: Statistics for simulation of Case Study Two with 128 quad-core machines.

Policies		Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Normalized Tardiness				% of Late Workflows			
RSP	MSP					Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	BPRM	0.65	13.82	476.1	4.17E+05	6.22E+03	74.3	0.0	36.6	4.8	6.0	0.0	2.5
FCFS	BPOM	0.80	13.92	680.4	2.47E+06	2.70E+04	140.1	0.2	83.2	5.3	5.8	0.0	4.2
FCFS	BC	0.80	13.95	689.9	2.54E+06	2.80E+04	140.4	0.2	84.7	5.4	6.0	0.0	4.2
FCFS	BM	0.80	13.92	678.4	2.45E+06	2.68E+04	140.6	0.2	83.5	5.2	5.8	0.0	4.2
FCFS	LDM	0.80	13.96	684.9	2.51E+06	2.74E+04	143.6	0.2	84.2	5.3	5.8	0.0	4.2
FCFS	LWR	0.80	13.91	683.4	2.47E+06	2.71E+04	141.0	0.2	83.8	5.4	6.0	0.0	4.2
FCLS	BPRM	0.95	13.82	193.3	8.25E+03	-3.01E+03	0.7	-0.1	0.2	3.9	5.9	0.0	0.0
FCLS	BPOM	0.90	13.83	55.4	8.02E+03	-3.11E+03	0.7	-0.1	0.0	0.9	1.4	0.0	0.0
FCLS	BC	0.90	13.90	57.6	8.64E+03	-2.73E+03	0.5	0.0	0.0	1.1	1.6	0.0	0.0
FCLS	BM	0.90	13.83	55.5	8.02E+03	-3.11E+03	0.7	-0.1	0.0	0.9	1.4	0.0	0.0
FCLS	LDM	0.90	13.90	57.2	8.28E+03	-2.94E+03	0.6	-0.1	0.0	1.1	1.6	0.0	0.0
FCLS	LWR	0.90	13.83	53.9	8.03E+03	-3.11E+03	0.6	-0.1	0.0	1.0	1.5	0.0	0.0
EDF	BPRM	0.90	13.87	100.9	8.17E+03	-3.03E+03	0.3	-0.1	0.1	3.3	5.0	0.0	0.0
EDF	BPOM	0.90	14.04	42.4	7.98E+03	-3.13E+03	0.3	-0.1	0.0	0.5	0.8	0.0	0.0
EDF	BC	0.90	14.06	46.4	8.60E+03	-2.75E+03	0.2	-0.1	0.0	0.5	0.8	0.0	0.0
EDF	BM	0.90	14.04	42.4	7.98E+03	-3.13E+03	0.3	-0.1	0.0	0.5	0.8	0.0	0.0
EDF	LDM	0.90	14.05	43.6	8.24E+03	-2.97E+03	0.2	-0.1	0.0	0.5	0.7	0.0	0.0
EDF	LWR	0.90	14.02	41.8	8.00E+03	-3.12E+03	0.2	-0.1	0.0	0.4	0.6	0.0	0.0
LLF	BPRM	0.90	13.77	107.3	8.18E+03	-3.02E+03	0.3	-0.1	0.1	3.6	5.4	0.0	0.0
LLF	BPOM	0.80	13.96	41.4	7.98E+03	-3.13E+03	0.2	-0.1	0.0	0.5	0.7	0.0	0.0
LLF	BC	0.80	14.02	46.7	8.62E+03	-2.74E+03	0.3	0.0	0.0	0.8	1.2	0.0	0.0
LLF	BM	0.80	13.96	41.4	7.98E+03	-3.13E+03	0.2	-0.1	0.0	0.5	0.7	0.0	0.0
LLF	LDM	0.80	13.97	43.6	8.25E+03	-2.96E+03	0.2	-0.1	0.0	0.6	0.9	0.0	0.0
LLF	LWR	0.80	13.86	41.7	8.00E+03	-3.12E+03	0.2	-0.1	0.0	0.6	0.9	0.0	0.0
PLLF	BPRM	0.90	13.66	118.0	8.21E+03	-3.01E+03	0.3	-0.1	0.2	3.9	5.8	0.0	0.0
PLLF	BPOM	0.85	13.66	41.3	7.99E+03	-3.13E+03	0.1	-0.1	0.0	0.6	0.9	0.0	0.0
PLLF	BC	0.80	13.74	45.5	8.62E+03	-2.74E+03	0.2	0.0	0.0	0.7	1.1	0.0	0.0
PLLF	BM	0.85	13.69	41.3	7.99E+03	-3.13E+03	0.1	-0.1	0.0	0.6	0.9	0.0	0.0
PLLF	LDM	0.85	13.73	43.6	8.26E+03	-2.95E+03	0.2	-0.1	0.0	0.7	1.1	0.0	0.0
PLLF	LWR	0.85	13.70	41.4	8.00E+03	-3.12E+03	0.1	-0.1	0.0	0.5	0.8	0.0	0.0
SWF	BPRM	0.95	14.23	150.0	8.17E+03	-3.04E+03	0.5	-0.1	0.2	3.1	4.6	0.0	0.0
SWF	BPOM	0.95	14.88	39.8	7.96E+03	-3.14E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	BC	0.95	14.92	42.8	8.56E+03	-2.79E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	BM	0.95	14.88	39.8	7.96E+03	-3.14E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	LDM	0.95	14.86	42.0	8.40E+03	-2.88E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	LWR	0.95	14.81	39.9	7.98E+03	-3.13E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	BPRM	0.95	14.14	145.8	8.15E+03	-3.05E+03	0.5	-0.1	0.2	2.8	4.1	0.0	0.0
DSWF	BPOM	0.95	14.73	39.8	7.97E+03	-3.14E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	BC	0.95	14.94	42.8	8.56E+03	-2.79E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	BM	0.95	14.73	39.8	7.97E+03	-3.14E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	LDM	0.95	14.97	42.0	8.40E+03	-2.88E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	LWR	0.95	14.70	39.9	7.98E+03	-3.13E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0

Table A.15: Statistics for simulation of Case Study Three with 128 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.60	13.84	216.2	5.97E+03	-4.75E+03	0.7	-0.1	0.4	2.4	3.2	0.0	1.0
FCFS	**BPOM**	0.80	13.92	666.5	1.63E+06	1.82E+04	140.1	0.2	66.5	5.3	5.8	0.0	4.2
FCFS	**BC**	0.80	13.95	675.3	1.67E+06	1.89E+04	140.4	0.2	67.8	5.3	5.9	0.0	4.2
FCFS	**BM**	0.80	13.92	664.1	1.61E+06	1.80E+04	140.6	0.2	66.5	5.2	5.8	0.0	4.2
FCFS	**LDM**	0.80	13.95	671.2	1.66E+06	1.86E+04	143.6	0.2	67.4	5.3	5.8	0.0	4.2
FCFS	**LWR**	0.80	13.91	670.5	1.62E+06	1.82E+04	141.0	0.2	67.3	5.3	5.8	0.0	4.2
FCLS	**BPRM**	0.95	13.82	126.0	5.70E+03	-4.89E+03	0.7	-0.1	0.2	2.0	3.0	0.0	0.0
FCLS	**BPOM**	0.90	13.83	33.7	5.54E+03	-4.97E+03	0.5	-0.1	-0.1	0.4	0.6	0.0	0.0
FCLS	**BC**	0.90	13.90	34.5	5.96E+03	-4.66E+03	0.4	-0.1	0.0	0.4	0.7	0.0	0.0
FCLS	**BM**	0.90	13.83	33.3	5.54E+03	-4.97E+03	0.5	-0.1	-0.1	0.4	0.5	0.0	0.0
FCLS	**LDM**	0.90	13.90	34.3	5.71E+03	-4.84E+03	0.4	-0.1	0.0	0.4	0.7	0.0	0.0
FCLS	**LWR**	0.90	13.83	31.7	5.54E+03	-4.97E+03	0.4	-0.1	0.0	0.4	0.6	0.0	0.0
EDF	**BPRM**	0.90	13.85	55.2	5.64E+03	-4.91E+03	0.2	-0.1	0.1	1.8	2.7	0.0	0.0
EDF	**BPOM**	0.80	14.00	28.6	5.51E+03	-4.99E+03	0.3	-0.1	-0.1	0.1	0.2	0.0	0.0
EDF	**BC**	0.80	14.04	29.8	5.94E+03	-4.68E+03	0.1	-0.1	0.0	0.1	0.2	0.0	0.0
EDF	**BM**	0.80	14.00	27.9	5.51E+03	-4.99E+03	0.1	-0.1	-0.1	0.1	0.2	0.0	0.0
EDF	**LDM**	0.80	14.04	28.5	5.68E+03	-4.86E+03	0.1	-0.1	-0.1	0.1	0.2	0.0	0.0
EDF	**LWR**	0.80	14.00	27.8	5.52E+03	-4.98E+03	0.1	-0.1	-0.1	0.1	0.2	0.0	0.0
LLF	**BPRM**	0.90	13.67	60.8	5.66E+03	-4.89E+03	0.3	-0.1	0.1	1.9	2.9	0.0	0.0
LLF	**BPOM**	0.75	13.72	60.1	5.65E+03	-4.90E+03	0.3	-0.1	0.1	1.9	2.8	0.0	0.0
LLF	**BC**	0.80	13.89	30.8	5.95E+03	-4.67E+03	0.2	-0.1	0.0	0.2	0.4	0.0	0.0
LLF	**BM**	0.75	13.70	60.6	5.66E+03	-4.89E+03	0.3	-0.1	0.1	1.9	2.8	0.0	0.0
LLF	**LDM**	0.75	14.04	85.4	6.28E+03	-4.42E+03	0.3	0.0	0.1	5.0	7.5	0.0	0.0
LLF	**LWR**	0.75	13.81	72.5	5.77E+03	-4.82E+03	0.3	-0.1	0.1	3.2	4.8	0.0	0.0
PLLF	**BPRM**	0.85	13.67	63.2	5.67E+03	-4.89E+03	0.3	-0.1	0.1	2.0	3.0	0.0	0.0
PLLF	**BPOM**	0.75	13.67	62.7	5.67E+03	-4.89E+03	0.5	-0.1	0.1	1.9	2.9	0.0	0.0
PLLF	**BC**	0.75	13.86	408.3	8.56E+03	-3.00E+03	0.3	0.1	0.2	32.5	49.0	0.0	0.0
PLLF	**BM**	0.80	13.64	29.0	5.53E+03	-4.98E+03	0.3	-0.1	-0.1	0.2	0.3	0.0	0.0
PLLF	**LDM**	0.80	13.67	29.1	5.70E+03	-4.84E+03	0.1	-0.1	-0.1	0.2	0.3	0.0	0.0
PLLF	**LWR**	0.75	13.69	74.8	5.78E+03	-4.82E+03	0.3	-0.1	0.1	3.1	4.7	0.0	0.0
SWF	**BPRM**	0.95	14.23	100.0	5.65E+03	-4.92E+03	0.5	-0.1	0.2	1.6	2.4	0.0	0.0
SWF	**BPOM**	0.95	14.88	27.5	5.51E+03	-4.99E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**BC**	0.95	14.92	29.5	5.91E+03	-4.71E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**BM**	0.95	14.88	27.5	5.51E+03	-4.99E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**LDM**	0.95	14.88	28.9	5.78E+03	-4.80E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
SWF	**LWR**	0.90	14.25	27.8	5.52E+03	-4.98E+03	0.1	-0.1	-0.1	0.1	0.2	0.0	0.0
DSWF	**BPRM**	0.95	14.16	92.8	5.64E+03	-4.92E+03	0.5	-0.1	0.2	1.5	2.3	0.0	0.0
DSWF	**BPOM**	0.95	14.77	27.5	5.51E+03	-4.99E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**BC**	0.95	14.77	29.5	5.91E+03	-4.71E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**BM**	0.95	14.77	27.5	5.51E+03	-4.99E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**LDM**	0.95	14.83	28.9	5.78E+03	-4.79E+03	0.0	-0.1	0.0	0.0	0.0	0.0	0.0
DSWF	**LWR**	0.95	14.79	27.6	5.52E+03	-4.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0

Table A.16: Statistics for simulation of Case Study Four with 128 quad-core machines.

Policies						Normalized Tardiness				% of Late Workflows			
RSP	MSP	Threshold	BCT	Cumulative Cost (Sigmoid)	Cumulative Cost (Quadratic)	Cumulative	Max	95th %	99th %	Total	UI	Batch	WS
FCFS	**BPRM**	0.90	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**BPOM**	0.75	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**BC**	0.80	25.80	39.7	2.48E+03	-1.55E+03	0.1	0.1	0.1	17.0	32.4	0.0	0.0
FCFS	**BM**	0.75	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCFS	**LDM**	0.75	25.79	24.5	2.18E+03	-1.74E+03	0.2	0.0	0.1	5.1	9.7	0.0	0.0
FCFS	**LWR**	0.75	25.34	17.2	2.01E+03	-1.88E+03	0.2	0.0	0.1	3.0	5.7	0.0	0.0
FCLS	**BPRM**	0.95	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCLS	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCLS	**BC**	0.80	25.81	39.3	2.48E+03	-1.55E+03	0.1	0.1	0.1	17.2	32.6	0.0	0.0
FCLS	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
FCLS	**LDM**	0.80	25.74	28.4	2.20E+03	-1.74E+03	0.1	0.1	0.1	8.7	16.6	0.0	0.0
FCLS	**LWR**	0.80	25.34	11.5	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.8	3.3	0.0	0.0
EDF	**BPRM**	0.90	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
EDF	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
EDF	**BC**	0.80	25.79	39.5	2.48E+03	-1.55E+03	0.1	0.1	0.1	17.0	32.3	0.0	0.0
EDF	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
EDF	**LDM**	0.80	25.73	28.2	2.20E+03	-1.74E+03	0.1	0.1	0.1	8.5	16.1	0.0	0.0
EDF	**LWR**	0.80	25.34	11.4	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.7	3.3	0.0	0.0
LLF	**BPRM**	0.90	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**BC**	0.80	25.78	39.9	2.48E+03	-1.55E+03	0.1	0.1	0.1	16.8	32.0	0.0	0.0
LLF	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
LLF	**LDM**	0.90	25.50	9.8	1.96E+03	-1.91E+03	0.0	-0.1	0.0	0.2	0.3	0.0	0.0
LLF	**LWR**	0.80	25.34	11.4	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.7	3.3	0.0	0.0
PLLF	**BPRM**	0.90	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**BC**	0.80	25.81	39.8	2.48E+03	-1.56E+03	0.1	0.1	0.1	17.1	32.5	0.0	0.0
PLLF	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
PLLF	**LDM**	0.80	25.77	28.1	2.20E+03	-1.74E+03	0.1	0.1	0.1	8.7	16.5	0.0	0.0
PLLF	**LWR**	0.80	25.34	11.4	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.7	3.3	0.0	0.0
SWF	**BPRM**	0.95	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**BC**	0.80	25.81	40.6	2.49E+03	-1.55E+03	0.1	0.1	0.1	17.5	33.3	0.0	0.0
SWF	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
SWF	**LDM**	0.80	25.72	26.6	2.20E+03	-1.74E+03	0.1	0.1	0.1	8.4	15.9	0.0	0.0
SWF	**LWR**	0.80	25.34	11.3	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.7	3.2	0.0	0.0
DSWF	**BPRM**	0.95	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**BPOM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**BC**	0.80	25.81	40.3	2.49E+03	-1.55E+03	0.1	0.1	0.1	17.4	33.1	0.0	0.0
DSWF	**BM**	0.80	25.34	9.3	1.87E+03	-1.98E+03	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0
DSWF	**LDM**	0.80	25.72	26.6	2.20E+03	-1.75E+03	0.1	0.1	0.1	8.2	15.5	0.0	0.0
DSWF	**LWR**	0.80	25.34	11.3	1.97E+03	-1.90E+03	0.1	-0.1	0.0	1.7	3.2	0.0	0.0

Acronyms

RSP	Request Selection Policy
MSP	Machine Selection Policy
FCFS	First Come First Serve
FCLS	First Come Last Serve
LLF	Least Laxity First
PLLF	Proportional Least Laxity First
EDF	Earliest Deadline First
SWF	Shortest Workflow First
DSWF	Dynamic Shortest Workflow First
BPRM	Best Pre-Mapping
BPOM	Best Post Mapping
BC	Best CPU
BM	Best Memory
LDM	Least Deadline Missed
LWR	Least Work Remaining
BCT	Batch Completion Time
SLA	Service Level Agreements
JMS	Java Message Service