Vakilinia *Journal of Cloud Computing: Advances, Systems and Applications* (2018) 7:2
DOI 10.1186/s13677-017-0103-2

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Energy efficient temporal load aware resource allocation in cloud computing datacenters

Shahin Vakilinia

Abstract

Cloud computing datacenters consume huge amounts of energy, which has high cost and large environmental impact. There has been significant amount of research on dynamic power management, which shuts down unutilized equipment in a datacenter to reduce energy consumption. The main consumers of power in a datacenter are servers, communications network and the cooling system. Optimization of power in a datacenter is a difficult problem because of server resource constraints, network topology and bandwidth constraints, cost of VM migration, the heterogeneity of workloads and the servers. The arrival of new jobs and departure of completed jobs also create workload heterogeneity in time. As a result, most of the previous research has concentrated on partial optimization of power consumption, which optimizes either server and/or network power consumption through placement of VMs. Temporal load aware optimization, minimization of power consumption as a function of time has vastly been studied. When optimization also included migration, then solution had been divided into two steps, in the first step optimization of server and/or network power consumption is performed and in the second step migration of VMs has been taken care of, which is not an optimal solution. In this work, we develop joint optimization of power consumption of servers, network communications and cost of migration with workload and server heterogeneity subject to resource and bandwidth constraints through VM placement. Optimization results in an integer quadratic program (IQP) with linear/quadratic constraints in number of VMs assigned to a job on a server. IQP can only be solved for very small size systems, however, we have been able to decompose IQP to master and pricing sub-problems which may be solved through column generation technique for systems with larger sizes. Then, we have extended the optimization to manage temporal heterogeneity of the workload. It is assumed that time-axis is slotted and at the end of each slot jobs makes probabilistic complete/partial release of the VMs that they are holding and there will also be new job arrivals according to a Poisson process. The system will perform re-optimization of power consumption at the end of each slot that also includes the cost of VM migration. In the re-optimization, VMs of unfinished jobs may experience migration while new jobs are assigned VMs. We have obtained numerical results for optimal power consumption for the system as well as its power consumption due to two heuristic VM assignment algorithms. The results show optimization achieves significant power savings compared to the heuristic algorithms. We believe that our work advances state-of-the art in dynamic power management of datacenters and the results will be helpful to cloud service providers in achieving energy saving.

Keywords: Cloud computing, Virtual machine placement, Integer linear programming, Integer quadratic programming, Optimization, Resource allocation, Column generation, Datacenter power management

Correspondence: Shahin.Vakilinia@gmail.com
Synchromedia Lab, ETS, 500 Rue Jean D'Estress, Montreal, QC H3C6W1, Canada

Introduction

The datacenters have been growing exponentially and together with that their power consumption. The energy consumption results in high operational cost and large impact on the environment. It is expected that the electricity demand for datacenters to rise more than 66% over the period 2011–2035 [1]. As a result, there has been significant research on how to reduce power consumption of the datacenters. The main consumers of power in a datacenter are servers, communications network and the cooling system. It has been determined that an idle server consumes about 70% of its peak power [2]. Dynamic power management together with server consolidation has been used to reduce power consumption by temporarily shutting down servers when they are not required. Server consolidation refers to migration of VMs to as few servers as possible so as to prevent underutilization of the servers. However, server consolidation is challenging because energy cost of migration and, if not carefully done, network communications cost may rise. Server consolidation may result jobs being assigned VMs from multiple servers, which may increase communication traffic between VMs. Thus it is important that optimization of power consumption includes servers, network communications and cost of migration. It has been determined that network accounts for at least 20% of the energy consumption of a cloud computing center and it may rise upto 50% under light job loading, which is typical of the data centers [3]. Since dynamic power management turns off the idle servers, it also reduces power consumption of the cooling system.

The optimization of power consumption also needs to take into account heterogeneity of the workloads and servers. Cloud workloads often have very large variations in their resource requirements, arrival rates and execution times. Cloud centers also have heterogeneity in their servers. In time, datacenters update the configuration of their resources and upgrade the processing capabilities, memory and storage spaces. They also construct new platforms based on the new high performance servers while the older servers are still operational. The heterogeneity of both servers and workloads increases complexity of the optimization of power consumption.

In this paper, we developed joint optimization of power consumptions of the servers, network communications and cost of VM migration with workload and server heterogeneity subject to server resource and network bandwidth constraints. We assumed a hierarchical two-tier datacenter network, though the work can be easily extended to higher tier networks. Optimization results in an integer quadratic program (IQP) with linear and quadratic constraints in number of VMs assigned to a server. This IQP problem is NP-hard [4] and it can only be solved for very small size systems. Due to similarity between our optimization problem and cutting stock problem, we utilized column generation (CG) technique to solve this optimization problem for larger systems. Then, we have extended our solution to handle temporal heterogeneity of the workload due to arrival and departure of the jobs. We assumed that the time-axis is slotted and at the end of each slot jobs are completed either partially or fully and new jobs arrive to the system according to a Poisson process during each time slot. In the partial completion a job releases each of its VMs according to independent Bernoulli trials, while in full completion each job departs from the system according to independent Bernoulli trials. Thus at the end of each slot, the workload load of the system consists of new arriving jobs during the present slot and unfinished jobs from the previous slots. We determine new VM placement by solving this optimization problem that also allows migration of the VMs of unfinished jobs in the system. VMs migrate if the energy savings outweigh cost of the migration. Management of VM migration requires addition of new constraints to the optimization. The main contributions of our work is as follows,

- It formulates joint optimization of server, network and migration power consumption with bandwidth constraints for a given network topology. It performs power optimization and VM migration simultaneously. The optimization problem is expressed as an IQP with quadratic constraints, which can only be solved for very small size systems.
- We have been able to cast this optimization problem into an integer linear programming (ILP), which may be solved through column generation technique for larger size systems. It appears that this is the first application of the column generation technique to the solution of the optimization of power consumption problem in cloud computing systems.
- The work incorporates temporal variation of the workload to the optimization, which allows general arrival and departure of the jobs as a function of discrete-time. This enables re-optimization of the power consumption at the discrete-time instants.

The remainder of this paper is organized as follows:Related work is presented in section 2 and system model in section 3. Section 4 presents IQP modeling of the optimization problem and the section after CG modeling of the problem. Section 6 develops the probabilistic extension of the model. Section 7 presents the temporal load aware formulation of the optimization problem. Section 8 discusses optimization structure and complexity mitigation and the section following that presents numerical results

regarding the analysis in the paper. Section 9 discuss the details of assumptions made in this research. Finally, section 10 presents conclusions of the paper.

Related work

In this section, we will present a survey of the related work on the dynamic power management in cloud computing centers. The previous work on dynamic power management may be classified into two as with or without power optimization and the first case may be further subdivided into two depending on whether or not optimization is joint over the servers and network power consumption. Classification may also include other parameters such as workload and server heterogeneity awareness and VM migration. Almost all of the previous works present heuristics rather than solving the optimization problem due to its complexity and then, they perform simulation to determine accuracy of the proposed heuristic.

First, we describe the previous work on dynamic power management without power optimization, which simply turns off idle servers to conserve power consumption. In [5], the effectiveness of dynamic power management in data centers had been investigated using *M/M/k* queuing model with matrix analytic technique. In [6], this analysis had been extended to the heterogeneous workload case.

Next we explain the previous work on dynamic power management with optimization of network power consumption, which is also referred to as traffic aware VM placement. In [7], an ad-hoc framework has been proposed which minimizes energy consumption of the datacenter network. The framework consists of two steps, and it assumes that the traffic patterns of the jobs are known. In the first step, VM assignment is done in a manner that the traffic in the network is reduced. In the second step, energy-efficient routing of the traffic is carried out that minimizes the number of active switches. In [8], it has been observed from real datacenter network traces that traffic demands of different flows do not peak at exactly the same time. As a result, [8] proposed monitoring of the traffic flows in the network and their consolidation into a small subset of links and switches periodically and shutting down of unutilized switches for energy saving.

Next, we describe previous work on dynamic power management with joint optimization of server and network power consumption. In [4], VM placement problem taking into account server operation and network communication costs had been studied. There is a tradeoff between physical machine (PM) cost and network cost where the PM cost is minimized if minimum number of servers is active. However, this may result in jobs being assigned VMs from multiple servers, which increases the network cost. The work proposes an algorithm minimizing the network-cost with fixed PM-cost. The proposed algorithm doesnot consider resource constraints of the servers, network topology and bandwidth constraints of the links. In [9] also VM placement minimizing power consumption has been studied. The work considers both server and communications network power consumptions as well as bandwidth constraints of the links. Server power consumption is assumed to be function of CPU operating frequency. Network infrastructure is assumed to have a hierarchical tree topology and following the optimization, idle servers and switches are turned off. They prove that job load of the servers should be balanced to achieve minimum server power consumption. Starting from this result, they propose a heuristic to assign the VMs to servers, which assigns VMs with high communication requirements among them to the same server. The work assumes that servers are homogeneous and doesnot consider resource constraints of the servers. The joint server and network power consumption optimization has also been studied in [10]. They proposed a unified model that combines server and network optimization by converting the VM assignment to a routing problem. However, optimization problem hasnot been solved due to its complexity, and instead the network is divided into clusters, which are optimized in parallel. The assignment of VMs to the servers and flows to the links in clusters are performed using a heuristic. The [11] also studied the joint optimization of server and network power consumption. They formulated the problem as an integer programming problem, proved that it is NP-hard and then proposed two greedy algorithms for VM scheduling.

Next, we explain previous work on heterogeneity aware dynamic power management. As mentioned earlier, due to inevitable platform upgrades or enhanced hardware resources, cloud platforms gradually become heterogeneous over time, which makes the VM placement problem more complex. In [12], the impact of hardware heterogeneity on the performance of public clouds had been investigated. During a two-year period, the activities of datacenters (DCs) are measured to establish some useful performance benchmarks that might affect the dynamic resource allocation in cloud DCs. Then these benchmarks, such as for CPU performance and network communication overhead, are utilized to evaluate the impact of heterogeneity on the performance of cloud computing centers. In [13] also heterogeneity of workloads and PMs have been considered. According to their resource demands and performance requirements jobs have been divided into classes, and similarly servers have been grouped based on their platform ID and capacities for different resources. Then, heterogeneity aware resource monitoring and management system dubbed

"Harmony" was proposed to perform dynamic capacity provisioning that minimizes the total energy consumption and scheduling delay considering heterogeneity as well as reconfiguration costs. The work assumes that a job will always be placed on a single server, as a result the optimization doesnot include network communications cost.

Next we describe the previous work on VM migration aware dynamic power management. In [14], an algorithm named as Peer VM Aggregation (PVA) has been proposed to migrate VMs of a job with high communication demands to the same server in order to reduce network utilization and power consumption. Simulation results show that average network utilization is reduced by %25. In [2], a two stage VM placement algorithm minimizing power consumption with migration has been proposed. In the first stage, VM placement is determined by solving a bin packing problem that minimizes power consumption. In the second stage, VM migration is applied at job departure points from the system that adapts the VM placement according to the released resources. Both stages of the problem are formulated as mixed integer linear programming (MILP) problem. While the work includes resource constraints of the servers in the optimization, but not network communications cost.

The work in this paper combines several of the above optimization problems together, therefore its results are more comprehensive and reliable. Our work jointly optimizes power consumption of servers and communications network and it includes both workload and server heterogeneity, resource constraints of the servers, network topology and link bandwidth constraints. Our work allows optimization to be done at discrete-time instants as the time evolves and some jobs depart and new ones arrive. It also models the VM migration and its cost, which enables adjustment of VM placement that re-optimizes power consumption under the new workload. In the previous work, optimization involving migration was performed in two steps, the first step performing VM placement that minimizes power consumption and the second step performing individual VM migration if it is cost effective. Clearly, this is not optimal because of partitioning of the problem into two separate sub-problems and piecemeal migration. Our work also includes the time-dimension in the optimization, which is absent from the previous work.

System model

In this section, we will present model of the system under consideration for optimization of power consumption through placement of VMs for the jobs. A datacenter consists of servers and communications network that provides connectivity among the servers and they are the main consumers of power in the system. Power consumption of a datacenter depends on its architecture, and in this work, we assume a hierarchical architecture, which is one of the commonly used topologies in the datacenters. It is assumed that the datacenter consists of a collection of Performance Optimized modular Data Centers (PoD). Each PoD consists of a number of racks and each rack contains a collection of servers. In Fig. 1, we show a typical two-tier datacenter network [15, 16], which has servers housed in a rack connected to a Top-Of-Rack (TOR) switch. The TOR switch provides connectivity among the servers of a rack and also connects the rack to the Core Switch (CS) of its host PoD. Core switches depending on the datacenter topology such as clique or fat-tree [15] may have different types of connectivity that provides varying amounts of bandwidths for communications among the PoDs. In this work, we assume that connectivity of core switches has the mesh topology.

The main activities resulting in power consumption are processing of the jobs by the servers and the communications between the servers. A job may be served by multiple VMs, which may be located on different servers. A job will have communications demand, when it is assigned VMs on different servers. The magnitude of this demand between two servers will be assumed to be proportional to the product of the number of VMs assigned to that job on the two servers. We assume that servers will be in one of two states, either *on* or *off* state. A server will be in the *on* state if it has at least one VM assigned to one of the jobs and otherwise it will be in the *off* state. An *on* server will consume constant power and an *off* server zero power.

We include server and workload heterogeneity in the model. We assume that a datacenter has T types of servers, where each server type is determined by the amount of different types of resources that it contains. A server type may have K different types of resources such as bandwidth, storage, CPU and memory. The amount of resources owned by a server of each type are given by a unique resource vector. We let M_t denote number of type t servers in the datacenter, and m'th type t server as $m_t \in \{1_t, \ldots, M_t\}$ with $t \in \{1, \ldots, T\}$. Power consumption of an *on* type t server will be denoted by Q_t. We assume that a server may have R different VM configurations. Each VM configuration is determined by the amount of different types of resources that it is allocated. We let i_r^k denote the type k resource requirement of a type r VM. We assume that there are H types of jobs, where each job type requires a random number of VMs from a group of VM types. Each job type has a different mix of VM types and a geometrically distributed service time in number of slots with a different parameter. We let N_h denote number of type h jobs in the datacenter,

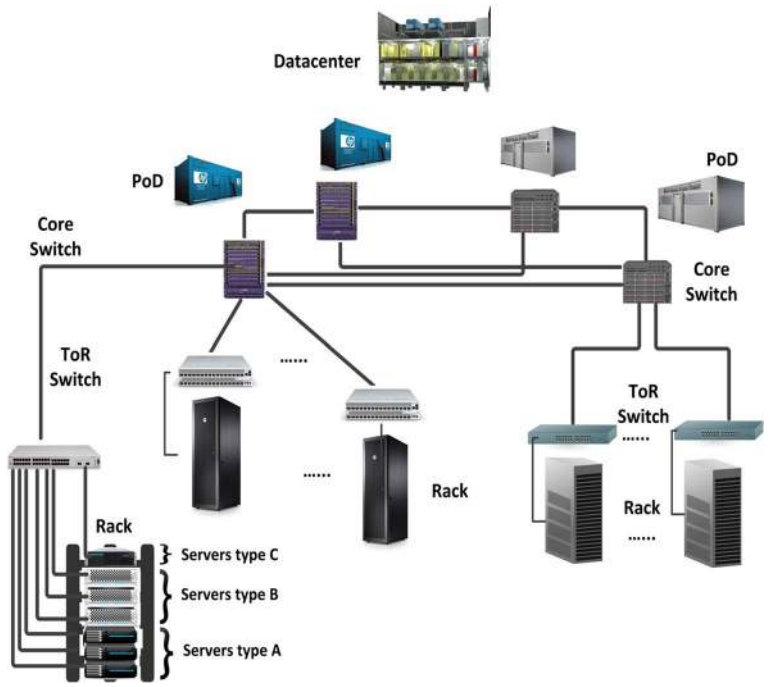

Fig. 1 Hierarchical Datacenter Architecture

$h \in \{1, ..., H\}$, and $v^r_{n_h}$ denote the number of type r VMs that job n_h requires, $n_h \in \{1_h, ..., N_h\}$. Let also N denote total number of jobs in the datacenter, then, $N = \sum_{h=1}^{H} N_h$.

The optimization problem also includes communication network bandwidth constraints to prevent traffic congestion. We assume that, there is no communication congestion between the servers located in the same rack because they are connected to their ToR switch with high capacity links. The communications congestion may occur either in the (TORS-CS) links or in PoD links (CS-CS). We assume that a ToR switch will be turned off if none of the servers in that rack are being utilized. Similarly CS in a PoD will be turned off if all the servers connected to its racks are off. We note that an *on* switch consumes a constant power plus load dependent variable power; the former will be referred to as static and the latter as dynamic power respectively. We will let $PS_{\ell,e^{ToRS}}, PS^{CS}_{\ell}$ denote static power consumption of a ToR switch on the e'th rack of PoD ℓ, and CS switch in PoD ℓ respectively. Similarly, we will let $PD_{\ell,e^{ToRS}}, PD^{CS}_{\ell}$ denote dynamic power consumption of these switches for per bit transmission rate. We also let PW_{NIC} denote the dynamic power consumption at the network interface card (NIC) of a server for per bit transmission rate.

The notation introduced in the above as well as others for this optimization problem has been summarized in Table 1. From this table,

$$M_t = \sum_{\ell=1}^{L} \sum_{e=1}^{d_\ell} M^t_{\ell,e} \tag{1}$$

$$M_{\ell,e} = \sum_{t=1}^{T} M^t_{\ell,e} \tag{2}$$

$$a_{\ell,e} = \{1_{\ell,e}, ..., , m_{\ell,e}, ..M_{\ell,e}\} \tag{3}$$

where $m_{\ell,\ e}$ denotes the m'th server on the e'th rack of PoD ℓ. The total power consumption of the datacenter will be minimized if the job load is served by minimum number of servers and each job is assigned VMs from as few servers as possible. In the next two sections, we will model the optimization problem first using IQP and then CG technique.

Modeling of the optimization problem with Integer Quadratic Programming (IQP)

In this section, we will model optimization problem of the system described in the previous section as an integer quadratic programming (IQP). The power consumption of a datacenter consists of static and dynamic power consumptions of the switches, dynamic power consumption of the interface cards and power consumption of the servers.

We first determine the dynamic power consumption due to communications of two VMs. Let $P^{n_h}_{m_t,m'_{t'}}$ denote total dynamic communication power consumption between two VMs located on servers m_t, $m'_{t'}$ and serving job n_h, then it is given by,

$$P^{n_h}_{m_t,m'_{t'}} = \begin{cases} 0, & if\ m_t = m'_{t'} \\ \vartheta_{n_h}\left(2PW_{NIC} + PD_{\ell,e^{ToRS}}\right), & if\ m_t, m'_{t'} \in a_{\ell,e}, m_t \neq m'_{t'} \\ \vartheta_{n_h}\left(2PW_{NIC} + PD_{\ell,e^{ToRS}} + PD^{CS}_{\ell} + PD^{ToRS}_{\ell,e'}\right), & if\ m_t \in a_{\ell,e}, m'_{t'} \in a_{\ell,e'}, e \neq e' \\ \vartheta_{n_h}\left(2PW_{NIC} + PD_{\ell,e^{ToRS}} + PD^{CS}_{\ell} + PD^{CS}_{\ell'} + PD^{ToRS}_{\ell',e'}\right), & if\ m_t \in a_{\ell,e}, m'_{t'} \in a_{\ell',e'}, e', \ell \neq \ell' \end{cases} \quad (4)$$

In the above, it has been assumed that communication power consumption between two VMs assigned to a job depends on the type of job but not on the types of VMs. As may be seen, this power depends on the location of the servers housing the VMs and on the data rate, which depends on the job type.

As defined in Table 1., the scheduling variable $x^{m_t}_{r,n_h}$ denotes number of type r VMs on the server assigned to serve job n_h and connectivity variable $\tilde{x}^{m_t}_{n_h}$ denotes total number of VMs assigned to job n_h on the m^{th} type t server assigned to serve job n_h, where, $\tilde{x}^{m_t}_{n_h} = \sum_{r=1}^{R} x^{m_t}_{r,n_h}$. We would like to determine optimal values of the scheduling variables $x^{m_t}_{r,n_h}$ that minimizes the datacenter power consumption. Next let us define the binary variables, y_{m_t} to denote *on* or *off* status of m^{th} type t server, $\eta_{\ell,\ e}$ status of the ToR switch serving to rack e on PoD ℓ as active or not, and ξ_ℓ status of the CS serving PoD ℓ as active or not. Then from the notation introduced in Table 1.,

$$y_{m_t} = \begin{cases} 1 & \sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h} \tilde{x}^{m_t}_{n_h} > 0 \\ 0 & \sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h} \tilde{x}^{m_t}_{n_h} = 0 \end{cases} \quad (5)$$

$$\eta_{\ell,e} = \begin{cases} 1 & \sum_{t=1}^{T}\sum_{m_t \in a_{\ell,e}} y_{m_t} > 0 \\ 0 & \sum_{t=1}^{T}\sum_{m_t \in a_{\ell,e}} y_{m_t} = 0 \end{cases} \quad (6)$$

$$\xi_\ell = \begin{cases} 1 & \sum_{e=1}^{d_\ell} \eta_{\ell,e} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \forall e \in \{1,\ldots,d_\ell\}, \forall \ell \in \{1,\ldots,L\}. \quad (7)$$

Then, the optimization problem for minimization of total power consumption is given below,

$$Min\left[\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\sum_{t=1}^{T}\sum_{t'=1}^{T}\sum_{m_t=1_t}^{M_t}\sum_{m_{t'}=1_{t'}}^{M_t} P^{n_h}_{m_t,m_{t'}}\left(\tilde{x}^{m_t}_{n_h}\tilde{x}^{m'_{t'}}_{n_h}\right) + \sum_{\ell=1}^{L}\left(\xi_\ell PS^{\ell}_{CS} + \sum_{e=1}^{d_\ell}\eta_{\ell,e}PS^{\ell,e}_{ToR}\right) + \sum_{t=1}^{T} Q_t \sum_{m_t=1}^{M_t} y_{m_t}\right] \quad (8)$$

ST. (5), (6), (7),

$$\tilde{x}^{m_t}_{n_h} = \sum_{r=1}^{R} x^{m_t}_{r,n_h}\ \forall n_h \in \{1,\ldots,N_h\}, m_t \in \{1_t,\ldots,,M_t\} \quad (9)$$

$$\sum_{t=1}^{T}\sum_{m_t=1_t}^{M_t} x^{m_t}_{r,n_h} \geq v^{r}_{n_h}\ \forall r \in \{1,\ldots,R\}, n_h \in \{1_h,\ldots,,N_h\}, h \in \{1,\ldots,H\} \quad (10)$$

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\sum_{r=1}^{R} x^{m_t}_{r,n_h} i^{k}_{r} \leq c^{k}_{t}\ \forall k \in \{1,\ldots,K\}, m_t \in \{1_t,\ldots,,M_t\} \quad (11)$$

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\vartheta_{n_h}\sum_{m_t \in a_{\ell,e}}\left[\sum_{\ell'=1}^{L}\sum_{e'=1}^{d_{\ell'}}\sum_{m'_{t'} \in a_{\ell',e'}}\left(\tilde{x}^{m_t}_{n_h}\tilde{x}^{m'_{t'}}_{n_h}\right) - \sum_{t',m' \in a_{\ell,e}}\left(\tilde{x}^{m_t}_{n_h}\tilde{x}^{m'_{t'}}_{n_h}\right)\right] \leq S_{\ell,e}\ \forall e \in \{1,\ldots,d_\ell\}, \forall \ell \in \{1\ldots L\} \quad (12)$$

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\vartheta_{n_h}\sum_{e=1}^{d_\ell}\sum_{m_t \in a_{\ell,e}}\sum_{e'=1}^{d_{\ell'}}\sum_{m'_{t'} \in a_{\ell',e'}}\left(\tilde{x}^{m_t}_{n_h}\tilde{x}^{m'_{t'}}_{n_h}\right) \leq CP_{\ell,\ell'}\ \forall \ell, \ell' \in \{1\ldots L\}, \ell \neq \ell' \quad (13)$$

We note that $m_t \in \{1_t, \ldots, M_t\}$ stands for $\forall t \in \{1, \ldots, T\}$.

In the objective function, the first term corresponds to the total dynamic communication power consumption in the datacenter. Second term represents the static part of communication power consumption and finally the last term corresponds to the power consumption of the servers. Constraint group (10) ensures that VM requirements of each job are satisfied and group (11) guarantees that resource demands of jobs scheduled on a server do not exceed that server's resource capacities. The constraints (12) and (13) ensure that bandwidth demands do not violate the capacities of TORs to CS and CS to CS links respectively. In these constraints, as defined in Table 1., $S_{\ell,\ e}$, $CP_{\ell,\ell'}$ denote capacities of TORS to CS and CS to CS links respectively.

From the Eqs. (8–13), the optimization problem is in the form of Integer Quadratic Programming (IQP) in the scheduling variables $x^{m_t}_{r,n_h}$. However, from the definitions of the variables y_{m_t}, $\eta_{\ell,\ e}$, ξ_ℓ given in Eqs. (5–7), the IQP problem has other nonlinear constraints. Next, we would like to convert the nonlinearities due to y_{m_t}, $\eta_{\ell,\ e}$, ξ_ℓ into a form with linear constraints, which will make the problem simpler. This can be achieved by replacing each of the equations in (5–7) by a pair of constraints as follows,

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\tilde{x}^{m_t}_{n_h} - y_{m_t} \geq 0 \quad (14)$$

Table 1 Summary of notation

Parameters	Definition
R	Number of VM types.
N	Number of jobs in the datacenter.
K	Number of resource types.
T	Number of different types of servers.
Q_t	Power usage of type t servers.
$v^r_{n_h}$	Number of type r VMs required by job n_h.
c^k_t	Type k resource capacity of a type t server.
i^k_r	Amount of type k resource required by a type r VM.
L	Number of PoDs in the data center.
d_ℓ	Number of racks in PoD ℓ.
b_ℓ	Set denoting racks in PoD ℓ.
$a_{\ell,\ e}$	Set denoting servers on rack e in PoD ℓ.
ϑ_{n_h}	Data rate of VMs serving job n_h.
$S_{\ell,\ e}$	Capacity of the link connecting rack e to its PoD ℓ CS switch.
$CP_{\ell,\ell'}$	The capacity of the link connecting CS switches of PoDs ℓ and ℓ'.
M_t	Total number of type t servers.
$M^t_{\ell,e}$	Number of type t servers in rack e of PoD ℓ.
$M_{\ell,\ e}$	Total number of servers in rack e of PoD ℓ.
PS^{CS}_ℓ	Static power usage rate of the CS switch in PoD ℓ.
$PS_{\ell,e^{ToRS}}$	Static power usage of the ToR switch on rack e in PoD ℓ.
$PD^{ToRS}_{\ell,e}$	Dynamic communication power usage of e^{th} rack ToR switch of PoD ℓ.
PD^{CS}_ℓ	Dynamic communication power usage of PoD ℓ CS switch.
PW_{NIC}	Dynamic communication power usage of server NIC card switch (for bit per second).
$P^{n_h}_{m_t,m'_{t'}}$	Dynamic communication power usage between two VMs serving job n_h allocated in servers m_t, and $m'_{t'}$.
$P'^{n_h}_{\ell,e;\ell',e'}$	Dynamic communication power usage between two VMs serving job n_h allocated in a server in rack e of PoD ℓ and in a server in rack e' in PoD $\ell'$of.
$PR_{\ell,\ e}$	Power supply of rack e on PoD ℓ.
Variables	Definition
$x^{m_t}_{r,n_h}$	Number of type r VMs in m^{th} type t server assigned to job n_h.
$\widetilde{x}^{mt}_{\ nh}$	Number of VMs in m^{th} type t server assigned to serve job n_h.
y_{m_t}	Binary variable denoting *on* or *off* status of m^{th} type t server.
$\eta_{\ell,\ e}$	Binary variable that assumes value of one if at least one server on rack e in PoD ℓ is active and otherwise zero.
ξ_ℓ	Binary variable that assumes value of one if at least one server in PoD ℓ is active and otherwise zero.
J_t	Total number of configuration patterns of a type t server.
$\widetilde{j}_t$	Configuration pattern $j\sim_t$ introduced by pricing problem t.
x^{jt}_{r,n_h}	Number of type r VMs assigned to job n_h on a type t server by pattern j_t.
$\widetilde{x}^{jt}_{nh}$	Number of VMs assigned to job n_h on a type t server by pattern j_t.
$m^{j_t}_{\ell,e}$	Number of active type t servers with pattern j_t in the e^{th} rack of PoD ℓ.

$$\theta y_{m_t} - \sum_{h=1}^{H} \sum_{n_h=1_h}^{N_h} \widetilde{x}^{m_t}_{n_h} \geq 0 \tag{15}$$

$$\sum_{t=1}^{T} \sum_{m_t \in a_{\ell,e}} y_{m_t} - \eta_{\ell,e} \geq 0 \qquad \forall e \in \{1, \ldots, d_\ell\}, \forall \ell \in \{1 \ldots L\} \tag{16}$$

$$\theta \eta_{\ell,e} - \sum_{t=1}^{T} \sum_{m_t \in a_{\ell,e}} y_{m_t} \geq 0 \qquad \forall e \in \{1, \ldots, d_\ell\}, \forall \ell \in \{1 \ldots L\} \tag{17}$$

$$\sum_{e=1}^{d_\ell} \eta_{\ell,e} - \xi_\ell \geq 0 \qquad \forall \ell \in \{1 \ldots L\} \tag{18}$$

$$\theta \xi_\ell - \sum_{e=1}^{d_\ell} \eta_{\ell,e} \geq 0 \qquad \forall \ell \in \{1 \ldots L\} \tag{19}$$

Thus we replace Eq. (5) with inequalities (14, 15). Definition in (5) implies that $\sum_{h=1}^{H} \sum_{n_h=1_h}^{N_h} \widetilde{x}^{m_t}_{n_h} = 0 \Leftrightarrow y_{m_t} = 0$ and $\sum_{h=1}^{H} \sum_{n_h=1_h}^{N_h} \widetilde{x}^{m_t}_{n_h} > 0 \Leftrightarrow y_{m_t} = 1$. From these observations, the inequalities in (14, 15) follow, where θ denotes a very large integer number. The correspondence between (6) and (16, 17) and between (7) and (18, 19) may be established similarly. In the remainder of the paper, these pairs of constraints will be referred to as "positive integer to binary linear conversion constraints" (IBLC).

As a result, our optimization problem has been converted to IQP with linear and quadratic constraints given by (8, 19). This optimization problem is NP hard and it can only be solved for very small size systems using the branch and bound technique [17].

Modeling of the optimization problem with column generation (CG) method

In this section, we will apply column generation technique to obtain another solution to our problem, which may be used with larger size systems. This technique originally had been applied to cutting-stock problem, which consists of cutting a set of available stock lengths to meet customer orders for items in required lengths and quantities with the objective of minimizing the wasted material [18]. Distinct combination of items in length and quantities cut from a stock length is called a pattern. In column generation approach, the optimization problem is divided into two types of subproblems referred to as restricted master and pricing problems [18]. The Restricted master problem (RMP) determines if the explored patterns satisfy the job demand constraints. The pricing problem finds a new pattern to feed the RMP. The objective function of the

pricing problem is in fact the reduced cost coefficient of the RMP. The RMP and pricing problems collaborate until reduced cost coefficients (objectives) of the pricing problems become negative indicating optimal solution has been reached. In our problem, there will be T pricing problems, one for each server type. RMP is in the form of integer linear program (ILP) and pricing problems are combinatorial optimization problems. RMP is solved using continuous relaxation, which at the end requires integer rounding of the results. There has been some work for application of column generation technique in quadratic programming [19–21].

In relation to the cutting-stock problem, server types and jobs are similar to stock and item lengths respectively, however server types and jobs have multiple resource constraints, while stocks and items have length as the only limit factor. Further, we have a complicated objective function compared to cutting stock problem. Let us define a pattern as a distinct combination of the number of VMs from each type of VMs that a server can accommodate. Let j_t denote such a pattern and J_t total number of patterns available for a type t server, then $j_t \in \{1_t, ..., J_t\}$. At the end of the solution, each active server is assigned one of these patterns. The new introduced notation may also been found in Table 1. Let $x^{j_t}_{r,n_h}$ denote number of type r VMs that has been assigned to job n_h in a server with pattern j_t and similarly, $\tilde{x}^{j_t}_{n_h}$ denote total number of VMs assigned to job n_h at a server with pattern j_t. Then, we have the following equality between the two variables,

$$\tilde{x}^{j_t}_{n_h} = \sum\nolimits_{r=1}^{R} x^{j_t}_{r,n_h}$$

The column vector $\boldsymbol{a_{j_t}} = \left(x^{j_t}_{1,n_1}, .. , x^{j_t}_{1,n_H}; ..; x^{j_t}_{r,n_1}, .., x^{j_t}_{r,n_H} ; ..; x^{j_t}_{R,n_1}, .., x^{j_t}_{R,n_H}\right)^{\boldsymbol{Tr}}$ will denote j'th pattern of type t server.

Let us define binary variable $y^{j_t}_{m_t}$ to denote whether or not a given type t server has pattern j_t, then,

$$y^{j_t}_{m_t} = \begin{cases} 1 & \text{if } m'\text{th server of type } t \text{ is active and has } j_t \text{ pattern.} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Next, let $m^{j_t}_{\ell,e}$ denote number of active type t servers with pattern j_t in the e^{th} rack of PoD ℓ,

$$m^{j_t}_{\ell,e} = \sum\nolimits_{m_t \in a_{\ell,e}} y^{j_t}_{m_t} \tag{21}$$

Finally, the state of rack e on PoD ℓ as active or not is determined as,

$$\eta_{\ell,e} = \begin{cases} 1 & \sum_{t=1}^{T}\sum_{j_t=1_t}^{J_t} m^{j_t}_{\ell,e} > 0 \\ 0 & otherwise \end{cases} \tag{22}$$

We note that total dynamic communication power consumption between two VMs located on servers m_t, $m'_{t'}$ and serving job n_h is still given by Eq. (4). Then, the optimization problem for the RMP is given by,

$$Min \Bigg[\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\sum_{t=1}^{T}\sum_{t'=1}^{T}\sum_{m_t=1_t}^{M_t}\sum_{m_{t'}=1_{t'}}^{M_t}\sum_{j_t=1_t}^{J_t} \sum_{j_{t'}=1_{t'}}^{J_{t'}} P^{n_h}_{m_t,m_{t'}} \left(y^{j_t}_{m_t} \tilde{x}^{j_t}_{n_h} y^{j_{t'}}_{m_{t'}} \tilde{x}^{j'_{t'}}_{n_h} \right) + \sum_{\ell=1}^{L} \left(\xi_\ell PS^{\ell}_{CS} + \sum_{e=1}^{d_\ell} \eta_{\ell,e} PS^{\ell,e}_{ToR} \right) + \sum_{t=1}^{T} Q_t \sum_{m_t=1}^{M_t}\sum_{j_t=1_t}^{J_t} y^{j_t}_{m_t} \Bigg] \tag{23}$$

ST. (18), (19)

$$\tilde{x}^{j_t}_{n_h} = \sum\nolimits_{r=1}^{R} x^{j_t}_{r,n_h}, \quad \forall n_h \in \{1, ..., N_h\}, j_t \in \{1_t, ..., , J_t\}$$

$$m^{j_t}_{\ell,e} = \sum\nolimits_{m_t \in a_{\ell,e}} y^{j_t}_{m_t} \quad \forall j_t \in \{1_t, ..., , J_t\}, t \in \{1, ..., T\}, \forall e \in \{1, ..., d_\ell\}, \forall \ell \in \{1...L\}$$

$$\sum_{t=1}^{T}\sum_{m_t=1_t}^{M_t}\sum_{j_t=1_t}^{J_t} y^{j_t}_{m_t} x^{j_t}_{r,n_h} \geq \nu^{r}_{n_h} \quad \forall r \in \{1, ..., R\}, n_h \in \{1_h, ..., , N_h\}, h \in \{1, ..., H\} \tag{24}$$

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h} \vartheta_{n_h} \sum_{t=1}^{T}\sum_{m_t \in a_{\ell,e}}\sum_{j_t=1_t}^{J_t} \left[\sum_{\ell'=1}^{L}\sum_{e'=1}^{d_{\ell'}}\sum_{t'=1}^{T}\sum_{j_{t'}=1_{t'}}\sum_{m'_{t'} \in a_{\ell',e'}} \left(y^{j_t}_{m_t} x_{\sim^{j_t}_{n_h}} y^{j_{t'}}_{m_{t'}} x_{\sim^{j'_{t'}}_{n_h}} \right) - \sum_{m'_{t'} \in a_{\ell,e}} \left(y^{j_t}_{m_t} x_{\sim^{j_t}_{n_h}} y^{j_{t'}}_{m_{t'}} x_{\sim^{j'_{t'}}_{n_h}} \right) \right] \leq S_{\ell,e} \forall e \in \{1, ..., d_\ell\}, \forall \ell \in \{1...L\} \tag{25}$$

$$\sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h} \vartheta_{n_h} \sum_{e=1}^{d_\ell}\sum_{t=1}^{T}\sum_{m_t \in a_{\ell,e}}\sum_{j_t=1_t}^{J_t}\sum_{e'=1}^{d_{\ell'}} \sum_{t'=1}^{T}\sum_{m'_{t'} \in a_{\ell',e'}}\sum_{j_{t'}=1_{t'}} \left(y^{j_t}_{m_t} \tilde{x}^{j_t}_{n_h} y^{j_{t'}}_{m_{t'}} \tilde{x}^{j'_{t'}}_{n_h} \right) \leq CP_{\ell,\ell'} \forall \ell, \ell' \in \{1...L\}, \ell \neq \ell' \tag{26}$$

$$\sum_{j_t=1_t}^{J_t} m^{j_t}_{\ell,e} \leq M^{t}_{\ell,e} \tag{27}$$

$$\sum_{t=1}^{T}\sum_{j_t=1_t}^{J_t} m^{j_t}_{\ell,e} - \eta_{\ell,e} \geq 0 \qquad \forall e \in \{1, ..., d_\ell\}, \forall \ell \in \{1...L\} \tag{28}$$

$$\theta\eta_{\ell,e} - \sum_{t=1}^{T}\sum_{j_t=1_t}^{J_t} m_{\ell,e}^{j_t} \geq 0 \qquad \forall e \in \{1,\ldots,d_\ell\}, \forall \ell \in \{1\ldots L\} \tag{29}$$

We note that in the above optimization problem, scheduling and connectivity variables $x_{r,n_h}^{j_t}, \tilde{x}_{n_h}^{j_t}$ are treated as constants. In the objective function, (23), the first term corresponds to power consumption of the interface cards and dynamic power consumption of active switches due to communication load, second term to static power consumption of active switches and the third term to power consumption of active servers. Constraint (24) ensures satisfaction of the VM requirements of the jobs. The constraints (25) and (26) ensure that bandwidth demands of the jobs do not violate the capacities of the TORS to CS links and CS to CS links respectively. Constraint (27) ensures that the demand for servers donot exceed the available server capacity. Constraints (28) and (29) are IBLC for the variable $\eta_{\ell,\,e}$ defined in (22). However, the variables $y_{m_t}^{j_t}$ and $y_{m'_{t'}}^{j'_{t'}}$ defined in (20) and their product makes the objective function and the constraints (25, 26) nonlinear. Let us define the following binary variable in order to remove this nonlinearity,

$$\phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} = y_{m_t}^{j_t} y_{m'_{t'}}^{j'_{t'}} \tag{30}$$

Then,

$$\phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} = \begin{cases} 1 & if\ y_{m_t}^{j_t} = y_{m'_{t'}}^{j'_{t'}} = 1 \\ 0 & otherwise \end{cases} \tag{31}$$

The binary multiplication in the above can be linearized through the following constraints,

$$\phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} \geq y_{m_t}^{j_t} + y_{m'_{t'}}^{j'_{t'}} - 1 \tag{32}$$

$$\phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} \leq y_{m_t}^{j_t}, \phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} \leq y_{m'_{t'}}^{j'_{t'}}\ \phi_{m_t,m'_{t'}}^{j_t,j'_{t'}} \geq 0 \tag{33}$$

Thus the constraints (32, 33) need to be added to the above optimization problem given in (23–29).

Next, we present the T pricing problems one for each server type. The pricing problem for server type t attempts to introduce the new pattern $j\sim_t$ to the RMP through solution of the following optimization problem,

$$Max \sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\sum_{r=1}^{R} u_{r,n_h}^{t}\left(x_{r,n_h}^{\tilde{j}_t}\right) \tag{34}$$

$$ST. \sum_{h=1}^{H}\sum_{n_h=1_h}^{N_h}\sum_{r=1}^{R} x_{r,n_h}^{\tilde{j}_t} i_r^k \leq c_t^k \forall k \in \{1,\ldots,K\} \tag{35}$$

where $x_{r,n_h}^{\tilde{j}_t}$ represents number of type r VMs assigned to job n_h by pattern $\tilde{j}_t$. The pricing problem's objective function is the reduced cost function of the RMP with respect to server type t and u_{r,n_h}^{t} coefficients denote the values of the dual variables of the RMP for type t server. Constraint group (35) ensures that resource constraints of the servers are satisfied.

As shown in Fig. 2, in the column generation technique, RMP and pricing problems are solved iteratively. In each iteration, following solution of the pricing problems, the pattern of the server type t with the highest objective function value is introduced to the RMP. The iterations continue, as long as there are reduced cost functions with positive values. The algorithm terminates when all the reduced cost functions become negative and as a result no new pattern is introduced to the RMP. However, the obtained solution corresponds to the continuous relaxation of the problem, and therefore the results need to be rounded into integer values, which will be dealt in a later section.

We note that CG gives us a linear solution of the problem, which reduces solution's complexity but this is achieved at the expense of substantial increase in number of variables and constraints [22].

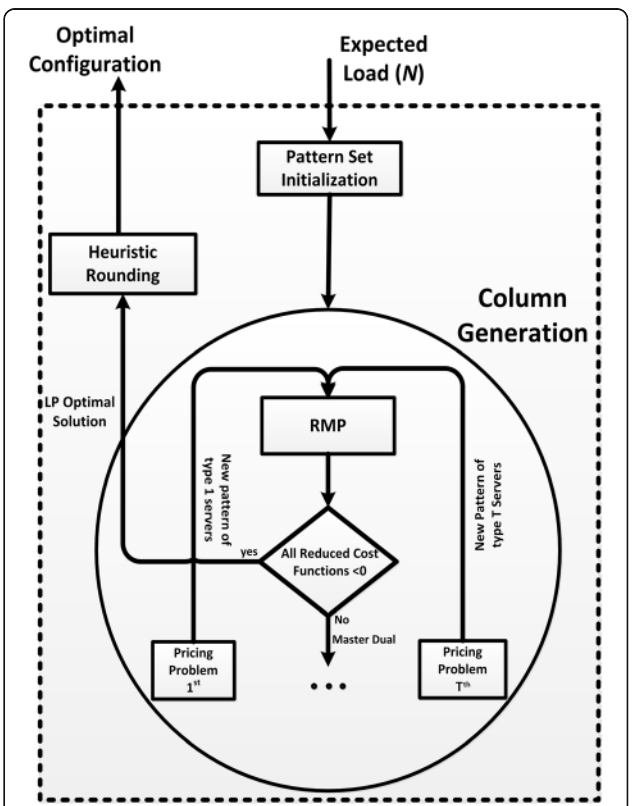

Fig. 2 Column Generation Implementation of the Optimization

Probabilistic model

In the previous sections, we have assumed deterministic traffic rates for communications between VMs and constant power consumption for active servers; however, in practice these quantities are random and vary as functions of time. In this section, we extend the optimization problem of the previous sections to a more realistic model, where VM communication rates and server power consumption are considered as random variables.

First, we assume that the data rate between two VMs serving to a type h job, ϑ_{n_h}, is a random variable. As a result, bandwidth constraints given in (12, 13) for IQP and (25, 26) for CG become probabilistic. For example (12) and (25) may be expressed as,

$$Pr\left(\sum_{h=1}^{H}\vartheta_{n_h}\sum_{n_h=1_h}^{N_h}\Psi_{\ell,e,n_h} > S_{\ell,e}\right)\le p \tag{36}$$

where Ψ_{ℓ,e,n_h} for IQP and CG are given by,

$$\Psi_{\ell,e,n_h} = \sum_{m_t\in a_{\ell,e}}\left[\sum_{\ell'=1}^{L}\sum_{e'=1}^{d_{\ell'}}\sum_{m'_{t'}\in a_{\ell',e'}}\left(\tilde{x}_{n_h}^{m_t}\tilde{x}_{n_h}^{m'_{t'}}\right) - \sum_{m'_{t'}\in a_{\ell,e}}\left(\tilde{x}_{n_h}^{m_t}\tilde{x}_{n_h}^{m'_{t'}}\right)\right], \textit{for IQP} \tag{37}$$

$$\Psi_{\ell,e,n_h} = \sum_{t=1}^{T}\sum_{j_t=1_t}^{J_t}\left\{\left[\tilde{x}_{n_h}^{j_t}\sum_{\ell'}\in\{1,..L\},\ell\neq\ell'\sum_{e'=1,}^{d_{\ell'}}\sum_{t'=1}^{T}\sum_{j'_{t'}=1_{t'}}^{J_{t'}}\left(\sum_{m_t\in a_{\ell,e}}\sum_{m'_{t'}\in a_{\ell',e'}}\phi_{m_t,m'_{t'}}^{j_t j'_{t'}}\right)\tilde{x}_{n_h}^{j'_{t'}}\right] + \left[\tilde{x}_{n_h}^{j_t}\sum_{e'\in b_\ell,e'\neq e}\sum_{t'=1}^{T}\sum_{j'_{t'}=1_{t'}}^{J_{t'}}\left(\sum_{m_t\in a_{\ell,e}}\sum_{m'_{t'}}\in a_{\ell'},e'\phi_{m_t,m'_{t'}}^{j_t j'_{t'}}\right)\tilde{x}_{n_h}^{j'_{t'}}\right]\right\}, \text{for CG} \tag{38}$$

In the above the objective is to keep probability of link congestion below a threshold value of p.

As in [23], we assume that total traffic rate follows a Gaussian distribution, which from the Central Limit Theorem remains an accurate model even if the individual flows are non-Gaussian [24, 25]. Next, we assume that mean and standard deviation of ϑ_{n_h} are given by λ_h and σ_h respectively, then the constraint (36) may be expressed as,

$$\sum_{h=1}^{H}\left(\lambda_h\sum_{n_h=1_h}^{N_h}\Psi_{\ell,e,n_h}\right) + \zeta\sqrt{\sum_{h=1}^{H}{\sigma_h}^2\left(\sum_{n_h=1_h}^{N_h}\Psi_{\ell,e,n_h}\right)^2}\le S_{\ell,e} \tag{39}$$

where $\zeta = \Phi^{-1}(1-p)$ and Φ^{-1} is the inverse function of the normal CDF. From [25], the LHS may be bounded, which reduces the above constraint to,

$$\sum_{h=1}^{H}\left[(\lambda_h+\zeta\sigma_h)\sum_{n_h=1_h}^{N_h}\Psi_{\ell,e,n_h}\right]\le S_{\ell,e} \tag{40}$$

In the previous sections, we assumed that power consumption of an *on* type t server is a constant denoted by Q_t. In fact, power consumption is random and depends on processing utility, I/O, load, memory usage etc. Let q_t denote power consumption of a type t server, then from [23], q_t has a general probability distribution, which varies in the range $[0.5Q_t, Q_t]$ with mean and standard deviation denoted by ω_t, δ_t respectively. It is better to avoid high power consumption at the rack level in order to prevent system failure [26]. As a result, we introduce the following probabilistic constraint,

$$Pr\left(\sum_{t=1}^{T}q_t\sum_{j_t=1_t}^{J_t}m_{\ell,e}^{j_t} > PR_{\ell,e}\right)\le p \tag{41}$$

where $PR_{\ell,\,e}$ denotes the power supply of rack e on PoD ℓ. As before, from the central limit theorem we assume that the total power consumption at the rack level has a Gaussian distribution. Similar to the analysis for Eq. (36), Eq. (41) can be linearized as follows,

$$\sum_{t=1}^{T}\omega_t\sum_{j_t=1_t}^{J_t}m_{\ell,e}^{j_t} + \zeta\sum_{t=1}^{T}\delta_t\sum_{j_t=1_t}^{J_t}m_{\ell,e}^{j_t}\le PR_{\ell,e} \tag{42}$$

This completes the extension to a probabilistic model with random server power consumption and link utilization. Thus new optimization problem also includes constraints (42) on rack's power consumption and on link utilization (25) is replaced by (40).

Temporal load aware VM placement

In this section, we will study VM placement with optimization of power consumption as a function of time, which will also be referred to as dynamic job scheduling. As a result, it will be assumed that time-axis is slotted and VMs are assigned to jobs in units of slot times. We will assume that arrival of jobs to the system is according to a Poisson process, though the analysis is applicable to other arrival processes. The new arriving jobs during the present slot and leftover jobs from the present slot will be scheduled for service in the next slot. We will consider two types of service disciplines, a job either releasing its assigned VMs simultaneously or individually according to Bernoulli trials at the end of each slot. In the former case, a leftover job will require full complement of its VMs and in the latter case a subset of the VMs it's currently holding. At the beginning of the next slot, the system will schedule the new arriving jobs and the leftover unfinished jobs from the previous slot such that power consumption is minimized. For the scheduling of leftover jobs, there are two options depending whether or not VM migration is

allowed. If VM migration is allowed, then leftover jobs are scheduled like the new jobs, on the other hand, if no migration is allowed then the new jobs can only be scheduled to VMs not utilized by the leftover jobs. As a result of migration, the system may end up in a state that consumes less power, however, migration has communication and processing overhead that optimization needs to take into account. Let G_r denote normalized power consumption cost of migration of type r VMs, which from [27] may be determined as follows,

$$G_r = (VM\ Migration\ power\ Consumption(Source + Destination)) \times Migration\ Duration\ Time\ Slot\ Duration$$

Optimization will allow VM migration if power saving due to migration offsets the cost of migration. As a result, the optimization may result in partial VM migration.

Since jobs release their VMs according to the Bernoulli trials, number of leftover jobs to the next slot will be a random variable with Binomial distribution. However, to make the analysis tractable we will assume that number of leftover jobs is a constant given by the mean of the Binomial distribution. Let $N_{h'}$ denote number of the type h leftover jobs from the current slot and N_h total number of jobs to be scheduled in the next slot, which include both leftover as well as new arriving jobs. We note that $N_h \geq N_{h'}$ and $n_h \in (1_h, .., N_{h'}, .., , N_h)$ and the first $N_{h'}$ jobs in the set correspond to the leftover jobs from the current slot. Next, we will develop both dynamic IQP and CG models for re-optimization of power consumption.

Dynamic *IQP* model

Let us consider n_h^{th} job, which is in the system in the current slot and will continue to receive service in the next slot. Let $x'^{m_t}_{r,n_h}$, $x^{m_t}_{r,n_h}$ denote the number of type r VMs assigned to this job over the m^{th} type t server during the current and next slots respectively. Based on the new notation introduced in Table 2, we define the following binary variable,

$$\beta^{m_t}_{r,n_h} = \begin{cases} 1 \text{ if } x^{m_t}_{r,n_h} - x'^{m_t}_{r,n_h} < 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{43}$$

The value of $\beta^{m_t}_{r,n_h}$ shows whether type r VMs required by job n_h have migrated or not. In the case of VM migration from this type of server, then $x^{m_t}_{r,n_h} < x'^{m_t}_{r,n_h}$ and as a result $\beta^{m_t}_{r,n_h}$ will have a nonzero value and in all other cases a zero value. The objective function of this optimization problem is given by,

Table 2 Additional notation

Parameters	Definition
$x'^{m_t}_{r,n_h}$	Number of type r VMs of server m_t assigned to serve job n_h at the current time slot.
$N_{h'}$	Total number of current type h jobs
$v'^{r}_{n_h}$	Number of type r VMs required by job n_h at current time slot left in the system.
$x'^{j_t}_{r,n_h}$	Number of type r VMs serving job n_h on a type t server with pattern j_t at current time slot
$m'^{j_t}_{\ell,e}$	Number of active type t servers with pattern j_t in the rack of pod ℓ at the current time slot.
$\varphi'^{f,j_t}_{\ell,e}$	Binary variable denoting whether type t server on rack e in pod ℓ which has pattern j_t at current time slot is active or not.
G_r	Power consumption related to the migration of type r VMs
$\tilde{x}^{\tau_t}_{r,h}$	Number of type r VMs assigned to type h jobs over the server type t by the initialization pattern τ_t.
R_h	Different VM types demanded by type h jobs.

$$Min \left[(8) + \sum_{h=1}^{H} \sum_{n_h=1_h}^{N_{h'}} \sum_{r=1}^{R} \sum_{\ell=1}^{L} \sum_{e=1}^{d_\ell} \sum_{t=1}^{T} \sum_{m_t \in a_{\ell,e}} G_r \beta^{m_t}_{r,n_h} \left| x^{m_t}_{r,n_h} - x'^{m_t}_{r,n_h} \right| \right] \tag{44}$$

where absolute value of $(x^{m_t}_{r,n_h} - x'^{m_t}_{r,n_h})$ corresponds to number of VM migrations. In the above, migration of a VM will be allowed if it results in power saving larger than cost of migration. Q_t in (8) also is considered as a linear function of total number of $\tilde{x}^{m_t}_{n_h}$s to better approximate the dependence of the utilization of the server in power consumption.

Job scheduling without VM migration can be achieved by assigning to G_r a very large value, which prevents migration as its cost cannot be offset by any amount of power saving. As a result, unfinished jobs will preserve their VM assignments. Finally, we have to add the following constraints into the optimization problem in order to linearize Eq. (43),

$$x^{m_t}_{r,n_h} - x'^{m_t}_{r,n_h} + \theta \beta^{m_t}_{r,n_h} < 1 \tag{45}$$

$$x^{m_t}_{r,n_h} - x'^{m_t}_{r,n_h} + \theta \beta^{m_t}_{r,n_h} \geq 0 \tag{46}$$

where (45), (46) are $\forall r \in \{1, ..., R\}$, $m_t \in \{1, ..., M_t\}$, $t \in \{1, ..., T\}$, $n_h \in \{1_h, ..., N_h\}$, $h \in \{1, ..., H\}$.

Dynamic CG model

Next, we consider the dynamic CG model. Assume that n_h^{th} job is in the system in the current slot and will continue to receive service in the next slot. Let

Table 3 Characteristics of Server types

Index (t)	Model	Num. of Cores $c_{t,1}$	Memory $c_{t,2}$	Num. of PMs M_t	Power Supply Q_t	ω_t, δ_t
1	Dell PE T110	4	16GB	1000	350 W	200 W, 20 W
2	Dell PE T410	8	128GB	300	580 W	400 W, 20 W
3	Dell PE M910	32	512GB	200	2750 W	1500 W, 100 W
4	Dell PE R810	16	512 GB	250	2200 W	1200 W, 100 W
5	Dell PE M915	64	1 TB	100	2750 W	1500 W, 100 W
6	Dell PE R910	40	2 TB	150	3000 W	1500 W, 100 W
7	HP DL320e Gen8	4	32GB	1000	350 W	200 W, 20 W
8	HP DL360e Gen8	8	384GB	500	750 W	400 W, 50 W
9	HP DL380p Gen8	8	768GB	250	1200 W	700 W, 50 W
10	HP DL360 G7	4	768GB	250	1200 W	700 W, 50 W
11	HP DL385p G7	16	768 GB	150	2000 W	1200 W, 100 W
12	HP DL370 G6	16	2 TB	100	2300 W	1150 W, 100 W

$x'^{j_t}_{r,n_h}$, $x^{j_t}_{r,n_h}$ denote number of type r VMs assigned to this job over the j_t'th pattern during the current and next slots respectively. Similarly, $\phi'^{f,j_t}_{\ell,e}$, $\phi^{f,j_t}_{\ell,e}$ are binary variables indicating whether f'th type t server on rack e in PoD ℓ is active and has pattern j_t during the current and next slots respectively. In this model, we define binary variable $\beta^{f,t}_{\ell,e,r,n_h}$ to show whether or not r type VMs required by job n_h have migrated or not from a server as follows,

$$\beta^{f,t}_{\ell,e,r,n_h} = \begin{cases} 1 \ if \ \sum_{j_t=1_t}^{J_t} \left(x^{j_t}_{r,n_h} \phi^{f,j_t}_{\ell,e} - x'^{j_t}_{r,n_h} \phi'^{f,j_t}_{\ell,e} \right) < 0 \\ 0 \qquad otherwise \end{cases} \tag{47}$$

We note that the summation in the above allows the use of a different pattern at the server as long as it preserves the number of VMs assigned by the original pattern to this job. The objective function of this optimization problem is given by,

$$Min \left[(23) + \sum_{h=1}^{H} \sum_{n_h=1_h}^{N_{h'}} \sum_{r=1}^{R} G_r \sum_{\ell=1}^{L} \sum_{e=1}^{d_\ell} \sum_{t=1}^{T} \sum_{f=1}^{M^t_{\ell,e}} \beta^{f,t}_{\ell,e,r,n_h} \sum_{j_t=1_t}^{J_t} \left| x^{j_t}_{r,n_h} \phi^{f,j_t}_{\ell,e} - x'^{j_t}_{r,n_h} \phi'^{f,j_t}_{\ell,e} \right| \right] \tag{48}$$

As in the previous subsection, job scheduling without VM migration can be achieved by setting G_r to a very large value. Finally, similar to the previous subsection, we have to add the following constraints to the problem in order to linearize (47),

Table 4 No. of servers per type per rack of POD ℓ

$M^t_{\ell,e}$ ℓ-> t	ℓ =1	ℓ =2	ℓ =3	ℓ =4
1	12	0	22	6
2	4	0	8	8
3	6	0	8	6
4	8	0	6	6
5	12	0	8	0
6	8	0	4	4
7	0	12	0	28
8	0	10	0	10
9	0	6	0	4
10	0	2	0	8
11	0	10	0	0
12	0	4	0	0
Sum	50	44	56	80

Table 5 No. of servers per type per PoD

PoD No(ℓ) Server type(t)	ℓ =1	ℓ =2	ℓ =3	ℓ =4
1	144	0	264	72
2	48	0	96	96
3	72	0	96	72
4	96	0	72	72
5	144	0	96	0
6	96	0	48	48
7	0	144	0	336
8	0	120	0	120
9	0	72	0	48
10	0	24	0	96
11	0	120	0	0
12	0	48	0	0
Sum	600	528	672	960

Table 6 Characteristics of Typical Switches

Switch Type	Interface data rates	Product name	Power (static) $PS_{\ell}^{ToR/CS}$	Power max	Power (Dynamic) $PD_{\ell,e}^{ToR/CS}$
TOR	10 GbpS int	NEC IP8800	25 W	145 W	65 nano w/bps
	40 GbpS, ext. ($S_{\ell,\ e}$)				
Core Switch	200 Gbps ($CP_{\ell,\ell'}$)	HP A12500	200 W	10,700 W	10 nano w/bps

$$\sum_{j_t=1_t}^{J_t}\left(x_{r,n_h}^{j_t}\phi_{\ell,e}^{f,j_t}-x'^{j_t}_{r,n_h}\phi'^{f,j_t}_{\ell,e}\right)+\theta\beta_{\ell,e,r,n_h}^{f,t}<1 \tag{49}$$

$$\sum_{j_t=1_t}^{J_t}\left(x_{r,n_h}^{j_t}\phi_{\ell,e}^{f,j_t}-x'^{j_t}_{r,n_h}\phi'^{f,j_t}_{\ell,e}\right)+\theta\beta_{\ell,e,r,n_h}^{f,t}\geq 0 \tag{50}$$

where (49), (50) are $\forall r\in\{1,...,R\}, f\in\left\{1,...,M_{\ell,e}^{t}\right\}, t\in\{1,,,T\}, n_h\in\{1_h,...,,N_h\}, h\in\{1,...,H\}$.

Optimization structure and complexity mitigation

In this section, we consider initialization of the optimization and rounding of the solution of relaxed problem to integer values.

CG initialization

We use offline initialization to reduce computation time for the solution of the optimization problem. Without initialization, in the first iterations, the RMP does not contain adequate columns to provide beneficial dual information to pricing sub-problems [28]. An appropriate initialization helps to reduce number of iterations of the solutions of RMP and pricing problems through introduction of optimal patterns, which are patterns that maximize resource utilization of active servers. Using the notation introduced in Table 2, we define the initialization (optimization) problem as follows,

$$Max\sum_{r=1}^{R}x_{r,h}^{\tilde{I}_t}i_r^k \tag{51}$$

$$ST.\ \sum_{\substack{r=1\\ \forall r\in R_h}}^{R}x_{r,h}^{\tilde{I}_t}i_r^{k'}\leq c_t^{k'}\forall k'\in\{1,...,K\} \tag{52}$$

where, R_h denotes the set of VM types available to type h jobs. Solving this problem for each $\{k,t,h\}$ results in the best Υ_t patterns for different types of jobs. Then, for a type t server we will have $\Upsilon_t HK$ initial patterns. To obtain $x_{r,n_h}^{\tilde{I}_t}$s, which are introduced in the previous sections and are related to the initial pattern $I\sim_t$, $x_{r,h}^{I\sim_t}$ is assigned to a type h job while other jobs are set to zero. Hence, for each $x_{r,h}^{I\sim_t}$ variable there would be N_h different patterns. Thus, initial number of patterns for server type t will be equal to $\sum_{h=1}^{H}N_h\Upsilon_t K$. So in the proposed initialization, we may have separate candidate patterns for each job. Then through collaboration of the pricing problems and RMP, new patterns, that consider different jobs in a server will be introduced by pricing problems.

Heuristic rounding termination algorithm

As mentioned earlier, LP problem (solvable in polynomial time) has less complexity compared to ILP problem (NP-hard optimization problem). In the CG solution of our optimization problem, RMP has been formulated as a LP and pricing problems as ILP type. As a result, we need to determine the optimal ILP solution of the RMP after the solution of the relaxed LP. Typically, this is done through the branch and bound algorithm [18], which is time consuming, as a result, we propose a heuristic method that satisfies the scheduling time constraint [28, 29]. The proposed method will round up and down the values of the scheduling variables, $m_{\ell,e}^{j_t}$, of the LP solution [30, 31]. This operation will be carried out after $m_{\ell,e}^{j_t}$ have been sorted according to their priorities. $m_{\ell,e}^{j_t}$s more likely to be rounded down will be given higher priority. Following this operation, it is possible that all the servers of a rack will become inactive in which case TOR switch serving to that rack will be turned off to save power.

Table 7 VM configurations

Index (r)	Model	vCPU (i_r^1)	Mem (GiB) (i_r^2)
1	t2.micro	1	1
2	t2.small	1	2
3	t2.medium	2	4
4	m3.medium	1	3.75
5	m3.large	2	7.5
6	m3.xlarge	4	15
7	c3.large	2	3.75
8	c3.xlarge	4	7.5
9	c3.2xlarge	8	15
10	c3.4xlarge	16	30
11	c3.8xlarge	32	60
12	r3.large	2	15.25
13	r3.xlarge	4	30.5
14	r3.2xlarge	8	61
15	r3.4xlarge	16	122
16	r3.8xlarge	32	244
17	g2.2xlarge	8	15
18	cg1.xlarge	16	22.5

Table 8 Jobs Types and their VM requirements

Index (h)	Job types	VM type	VM types Percentage	C_h	a_h	Traffic rates, ω_h, σ_h (Mbps)
1	Graphical Processing Jobs	g2.2xlarge	%70	50	0.14	3, 0.2
		cg1.xlarge	%30			
2	Scientific Jobs 1	c3.largec3	30%	100	0.14	0.7, 0.05
		c3,xlarge	30%			
		c3.2xlarge	20%			
		c3.4xlarge	10%			
		c3.8xlarge	10%			
3	Scientific Jobs 2	r3.large	30%	100	0.14	0.7, 0.05
		r3,xlarge	30%			
		r3.2xlarge	20%			
		r3.4xlarge	10%			
		r3.8xlarge	10%			
4	Scientific Jobs 3	m3.medium	50%	100	0.14	12, 2
		m3.large	30%			
		m3.xlarge	20%			
5	Data Search	m3.xlarge	30%	100	0.14	1, 0.1
		r3.large	20%			
		r3,xlarge	20%			
		r3.2xlarge	10%			
		r3.4xlarge	10%			
		r3.8xlarge	10%			
6	Enterprise Infrastructure Services	t2.micro	30%	100	0.15	3, 0.5
		t2.small,	20%			
		t2.medium	20%			
		m3.medium	10%			
		m3.large	10%			
		m3.xlarge	10%			
7	Peer 2 Peer Services	c3.large, c3,xlarge	30%, 30%	100	0.15	10, 1
		c3.2xlarge, c3.4xlarge	20%, 10%			
		c3.8xlarge	10%			
		+	+	+		
		r3.large, r3,xlarge	30%, 30%	50		
		r3.2xlarge, r3.4xlarge	20%, 10%			
		r3.8xlarge	10%			

First, let us define $s_{\ell,\ e}$ as the set of scheduling variables for the rack e on PoD ℓ,

$$s_{\ell,e} = \left\{ m_{\ell,e}^{j_t}, j_t \in \{1_t, \ldots, , J_t\}, t \in \{1, \ldots, T\} \right\}$$

and define set S as the set with its elements given by the subsets $s_{\ell,\ e}$ as given below,

$$S = \left\{ s_{\ell,e} | 1_\ell \le e \le d_\ell, 1 \le \ell \le L \right\}$$

Next, we split S into two mutually exclusive subsets,

$$S = \{S_1, , S_2\}$$

where S_1 consists of all $s_{\ell,\ e}$ that have their scheduling variables with values strictly less than one and S_2 otherwise. The elements of S_1denote potentially inactive racks, while the elements of S_2 active racks. From the above definition, elements of $S_1$are given higher priority than S_2 in the rounding operation. Next, we explain the steps of the proposed heuristic, we note that Step i) applies only to S_1, while the remaining steps apply both to S_1 and S_2.

Step i) Sort the elements of set S_1 according to the number of active servers in a rack, $\sum_{t=1}^{T} \sum_{j_t=1_t}^{J_t} m_{\ell,e}^{j_t}$, in

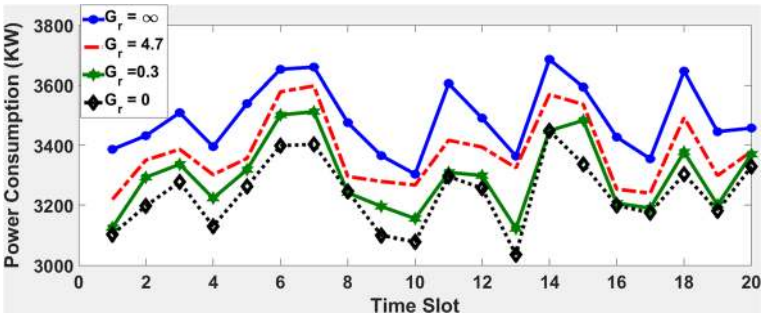

Fig. 3 Optimal power consumption as a function of time with migration cost G_r as a parameter

ascending order with the first element of S_1 having the least number of active servers. We note that the order of the elements of S_2 isnot significant for rounding operation.

Step ii) Sort the scheduling variables in each $s_{\ell,\ e}$ subset according to server type efficiency and resource scarcity. First, we explain how to determine server type efficiency. Depending on the job load some resources become critical and may become performance bottleneck [32, 33]. As a result, we first sort resources according to their criticalities. For a given job load, let L^kdenote the total demand for type k resource,

$$L^k = \sum_{h=1}^{H} \sum_{n_h=1_h}^{N_h} \sum_{r=1}^{R} v_{n_h}^r\, i_r^k \quad \forall k \in \{1, \ldots, K\}$$

Then, the resource types may be ordered according to their criticality using the following formula,

$$max_k \frac{L^k}{\sum_{t=1}^{T} M_t c_t^k} \tag{53}$$

Thus higher is the ratio of total demand to total amount of that resource in the datacenter, then higher will be the criticality of that resource. Next, we define efficiency of a server type with respect to resource type k as the ratio of (c_t^k/Q_t) with higher value indicating higher efficiency. Then, we order server types according to their efficiency for the critical resource. In the case of a tie, server efficiencies with respect to second critical resource will be used to break down the ties and so on and so forth. System will prefer to use the server types with higher efficiencies. The scheduling variables in each $s_{\ell,\ e}$ subset will be sorted in ascending order according to the efficiency of their server types.

Step iii) Sort the scheduling variables with common server type in each $s_{\ell,\ e}$ subset according to pattern efficiency:

The patterns of each server type will be sorted in ascending order according to their resource utilization $\sum_{h=1}^{H} \sum_{n_h=1_h}^{N_h} \sum_{r=1}^{R} x_{r,n_h}^{j_t} i_r^k$.

Step iv) Apply the rounding down operation. Following the completion of sorting, all the $m_{\ell,e}^{j_t}$ s within the set S have been assigned priority with the first element of the set having the highest priority in rounding down operation. Initially, we round up all the $m_{\ell,e}^{j_t}$ variables with non-integer values. Then, rounding down operation is applied from the highest to lowest priority $m_{\ell,e}^{j_t}$s one by one. In this operation, each $m_{\ell,e}^{j_t}$ is decremented by one if the demand constraints are not violated, $\sum_{\ell=1}^{L} \sum_{e=1}^{d_\ell} \sum_{t=1}^{T} \sum_{j_t \in J_t} \left(x_{n,r}^{j_t} \right) m_{\ell,e}^{j_t} < v_n^r$.

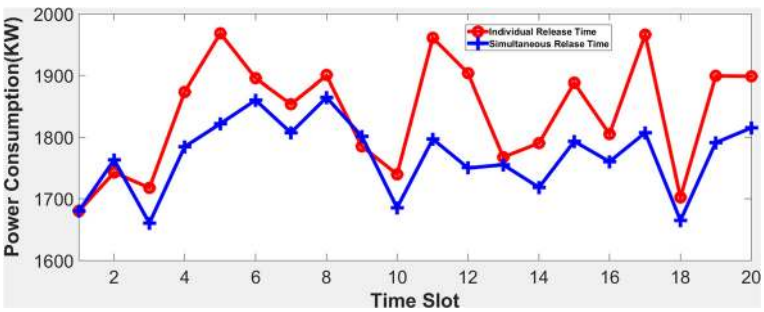

Fig. 4 Optimal power consumption as a function of time for individual and simultaneous VM release with parameter ρ_h=0.3 and migration cost G_r=0.3

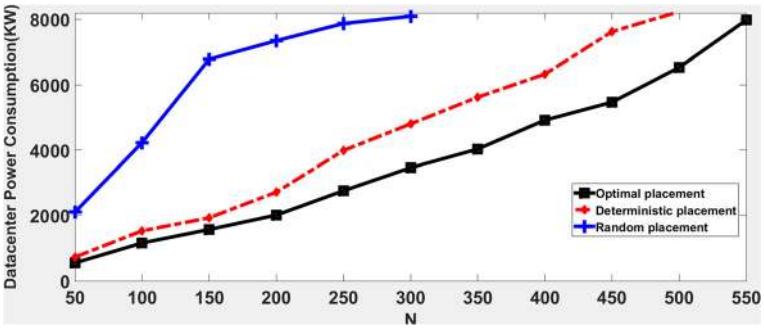

Fig. 5 Optimal, deterministic and random heuristic power consumptions as a function of the number of jobs in the system for constant server power consumption

The complexity order of the proposed algorithm may be approximated as,

$$O\left(\sum_{t=1}^{T} M_t J_t \left(min\left(log\left(\sum_{t=1}^{T} M_t J_t\right), NR\right)\right)\right)$$

where, M_t, N, R and J_t are number of type t servers, number of jobs, number of VM types and number of patterns of type t servers respectively. $\left(\sum_{t=1}^{T} M_t J_t\left(log\left(\sum_{t=1}^{T} M_t J_t\right)\right)\right)$ is due to the sorting part and $NR\sum_{t=1}^{T} M_t J_t$ is due to the demands for constraints part.

Numerical results

In this section, we present some numerical results regarding the analysis in the paper. Numerical results plot a performance metric for assignments of VMs to new arriving jobs either at an empty or non-empty system that optimizes power consumption. In an empty system, all the VMs are available, while in a non-empty system some of the VMs are occupied by the jobs already in the system. In the non-empty case, a performance metric is plotted as a function of discrete-time and new jobs arrive to the datacenter according to a Poisson process with parameter λ and VMs of the jobs in the system are released according to independent Bernoulli trials.

We compare performance of our optimal VM placement algorithm with that of two heuristic VM scheduling algorithms to be referred to as deterministic and random. The deterministic algorithm is similar to the scheduling scheme proposed in [34] that assigns a job to the PoD and rack with the smallest index number, which has enough idle resources to serve the job. In the random algorithm, each VM of a job is placed to a randomly chosen rack of a PoD with enough idle resources given that communication demand does not violate the link capacities; otherwise a new rack is randomly chosen for the placement of VM.

IBM ILOG CPLEX version 12.4 on a machine at 3.4 GHz(core i7) with 32GB RAM is used as a platform to solve the optimization problem. We solve the optimization problem using both IQP and CG techniques. IQP technique provides exact solution but is applicable to only small size systems, while CG is applicable to systems with large sizes but has rounding approximation. As a result, we test the accuracy of the CG technique against the IQP at the end.

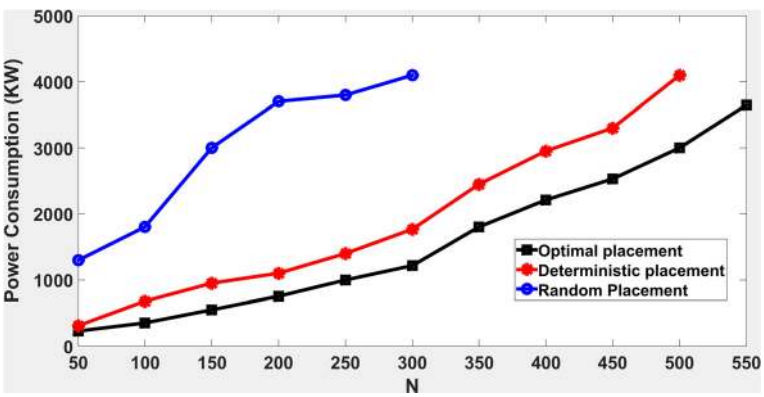

Fig. 6 Optimal, deterministic and random heuristic power consumptions as a function of the number of jobs in the system for random server power consumption

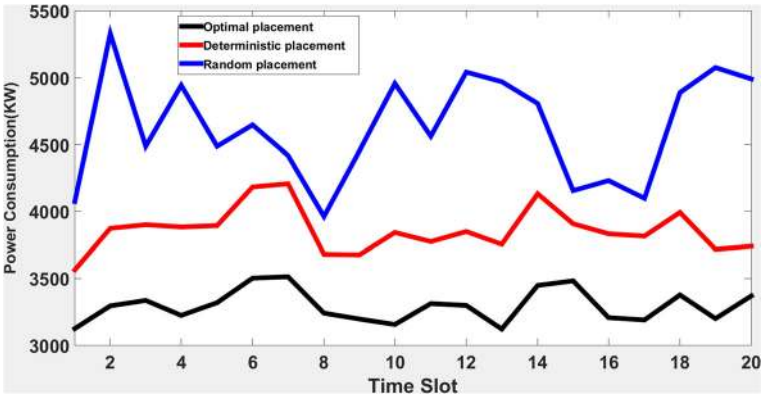

Fig. 7 Optimal power consumption with migration (Gr = 0.3) together with deterministic and random heuristics as a function of time

We assume a datacenter with the hierarchical topology shown in Fig. 1. We presume that the datacenter has 4 PoDs and each PoD having 12 racks. In consonance with [35], we assumed that each rack contains 40 to 80 servers and all the racks of each PoD have the same server composition. Next, we present the parameters of the system used in the generation of numerical results.

i) ***Servers and Server Types***
Considering Amazon instances and Google clusters, we assume $T = 12$ server types with two types of resources, CPU cores and memory. Table 3 presents the amount of resources and power consumption of each server type. Note that Q_t values are for maximum utilization cases. Table 4 presents number of servers per server type per rack at each PoD. Table 5 shows number of servers per server type per PoD, which is obtained by multiplication of each entry of Table 4 by 12.

ii) ***Communication Network Parameters***
Table 6 presents the performance characteristics of the chosen switches for the communications network. Power consumption parameter values of the switches, $PD_{\ell,\,e}$ and $PS_{\ell,\,e}$, are the same as given in [36–38]. We also assume that dynamic power consumption of a NIC is given by $PW_{NIC} = 0.6\ micro W$. ToR switches offer a combination of internal (int) and external (ext) interfaces. The internal interfaces connect to NIC of the blade-servers while the external interfaces connect to Core switches. It is assumed that internal and external interfaces support up to 10 Gbps and 40 Gbps transmission rates respectively.

iii) ***Parameters of VM Types***
We presume that number of VM types is $R = 18$ with their resource requirements given in Table 7. Resources of VMs consist of number of CPU cores and amount of memory. It is assumed that each physical core of a CPU is utilized as a virtual CPU (vCPU). In order to balance CPU, memory and network resources, Amazon t2 and m3 series are

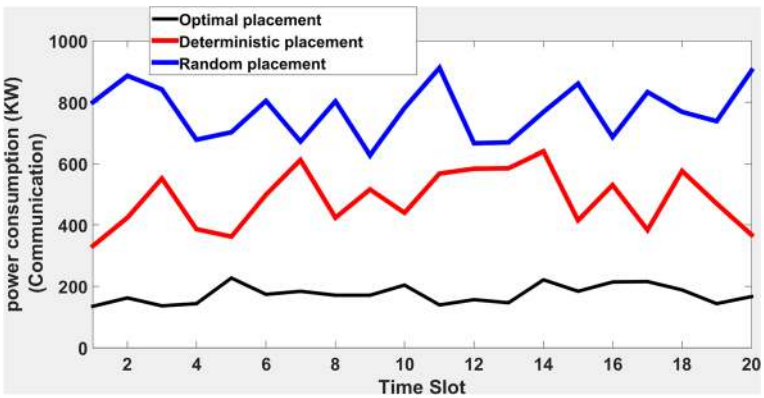

Fig. 8 Communication power consumption for the optimal placement with migration (Gr = 0.3) together with deterministic and random heuristics as a function of time

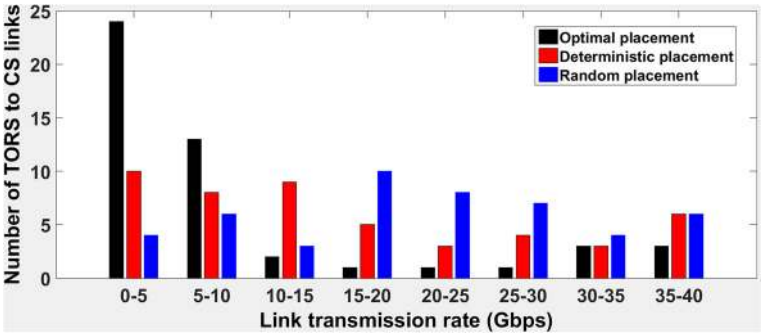

Fig. 9 Histogram of the number of TORS to CS links for optimal, deterministic and random placement of the jobs as a function of the link transmission rate for constant VM traffic rate

appropriate for many applications and servers, Microsoft SharePoint, and enterprise applications. c3 series with higher ratio of vCPU to memory represent compute-optimized Amazon instances which are appropriate for high-traffic web sites, on-demand batch processing, distributed analytics, web servers, and high performance science and engineering applications. r3 series represent memory optimized amazon instances and are recommended for memory bound applications such as high performance databases and distributed cache, in-memory analytics, genome assembly, and larger deployments of SAP. cg1 and g2 are also considered for game streaming, video encoding, 3D application streaming and other server-side graphic workloads.

iv) ***Parameters of job types***

We assume that the number of job types equals to $H = 7$ *with* $h = 1..H$. Table 8 presents requirements and appropriate applications for each job type. The type of each job is determined probabilistically through the values given in the column for parameter α_h. The number of VMs required by a type h job is determined by the constant C_h. From Amazon recommendations in [39, 40], the table present the mixture of VM types required by a job of each type. In each job type, the VMs are chosen probabilistically from the allowed VM types according to the percentages given in the table. Thus, first the type of a job and the number of VMs it requires are determined and then the types of each of its VM.

We assume that the traffic rate between two VMs of a job is either a random variable or a constant. In the former case, the mean and standard deviation of the traffic rate for each job type is given in the last column of the Table 8. In the latter case, the traffic rate for each job type is a constant that equals to the mean value of the variable traffic rate (ω_h). We considered both individual and simultaneous release of VMs of jobs at the end of a slot according to Bernoulli trials. In either case, the success probability in a Bernoulli trial is assumed to be ρ_h for a type h job. For this example, we assumed homogeneous Bernoulli trials with $\rho_h = 0.3\ \forall h \in \{1, \ldots, H\}$. Finally for the power constraint in the probabilistic model, we assume that power supply of a rack is $PR_{\ell,\ e} = 25kW$ [26] and maximum power overloading probability of the racks is set to $p = 0.02$. In the following results,

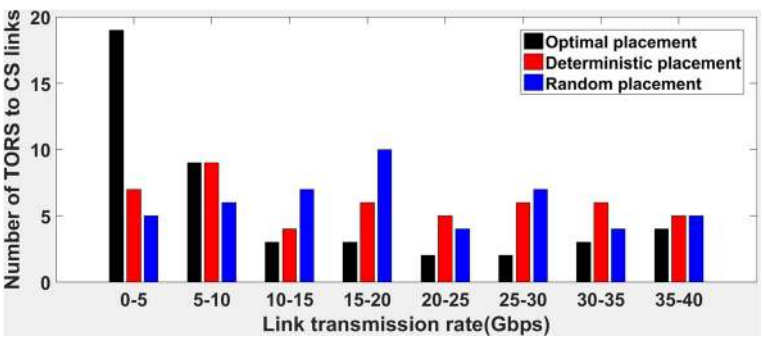

Fig. 10 Histogram of the number of TORS to CS links for optimal, deterministic and random placement of the jobs as a function of the link transmission rate with variable VM traffic rate

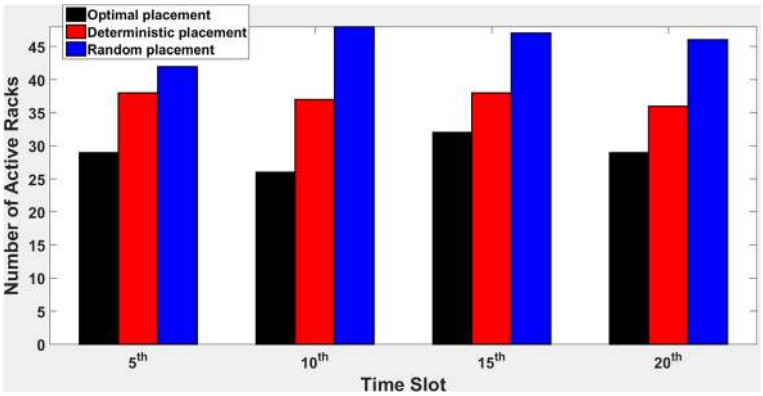

Fig. 11 The number of active racks in the system at slots 5, 10, 15 and 20 for optimal with migration cost (Gr = 0.3), deterministic and random placement of jobs

unless otherwise stated, the arrival of new jobs will be according to a Poisson process with parameter $\lambda = 200$ jobs/slot with constant server power consumptions and VM traffic rates.

The Fig. 3 presents optimal power consumption of the system as a function of the number of time slots with VM migration cost G_r as a parameter and with individual VM release. It may be seen that optimal power consumption increases with the rising cost of VM migration cost. The zero cost migration ($G_r = 0$) and no migration ($G_r = \infty$) provide lower and upperbound for power consumption with about 8% difference between them.

Figure 4 presents optimal power consumption of the system as a function of the number of time slots for both individual and simultaneous release of the VMs assigned to a job and migration cost of G_r=0.3**.** As may be seen, simultaneous release results in lower power consumption compared to individual release.

Figures 5 and 6 show power consumption as a function of the number of jobs in the system for optimal placement of VMs as well as according to the deterministic and random heuristics for constant and random server power consumptions respectively. As may be seen, in both cases, optimal placement of the jobs results in the lowest power consumption, and next to it is deterministic placement. It is also seen that the random server power consumption results in lower system power consumption compared to constant server power consumption.

Figure 7 presents optimal power consumption with migration cost G_r=0.3 as a function of the number of time slots. Also shown in the figure are the power consumptions of deterministic and random heuristics. It may be seen that optimal placement results in about 15% lower power consumption than deterministic heuristic and lower by a bigger amount than random heuristic. For the system of Figs. 7 and 8 shows communication component of the power consumption. As may be seen again, optimal placement results in lower communication power consumption than the two heuristics even by larger margins than the total power consumption.

Figures 9 and 10 show histogram of the number of TORS to CS links as a function of the link transmission rates for optimal, deterministic and random placement

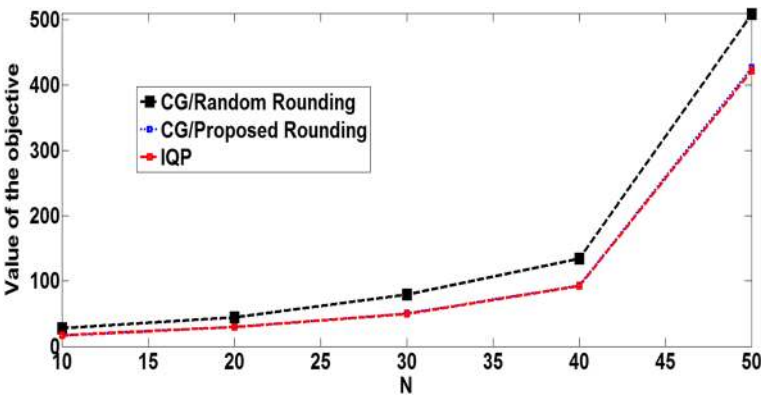

Fig. 12 Comparison of values of the objective functions among IQP, CG/Proposed Rounding and CG/Random Rounding

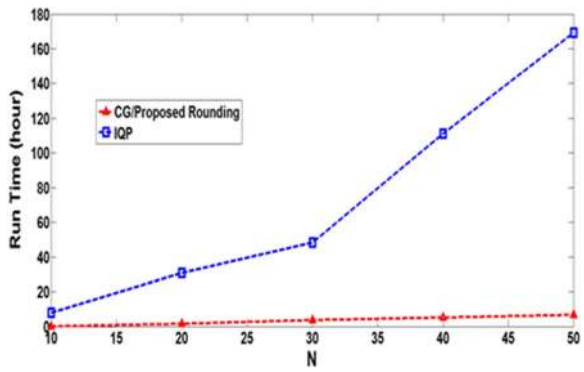

Fig. 13 Comparison of the run time between IQP and CG/proposed rounding

of jobs for constant and variable VM traffic rates respectively. As may be seen in both cases optimal placement results in lower communication traffic compared to the two heuristics.

Figure 11 shows the number of active racks for optimal with migration cost of G_r=0.3, deterministic and random placement of jobs at certain time slots. We note that the total number of the racks in the system is 48. As may be seen, optimal placement results in lower number of active racks compared to deterministic and random heuristics in any time slot (Time slot duration is considered 5 min).

We were able to solve IQP exactly for small size systems, which enabled us to examine quality of the solutions obtained through the CG technique. In Fig. 12, we plotted power consumption of CG/proposed rounding and IQP as a function of the number of jobs. The figure also plots results of random rounding of the relaxed CG solution, which provides an upper-bound for the performance of our optimization model. It can be seen that the optimality gap between the exact IQP results and upper-bound is up to 6%, while between the CG/proposed rounding and exact results is less than 1% for $N <$ 50. Thus, the CG technique with the proposed rounding algorithm results in quite accurate solutions.

Next, we look at the run time of the optimization models in Fig. 13. It can be noticed that as the workload (number of jobs) in the datacenter increases, the runtime of both IQP and CG increase. However, the runtime of the IQP grows exponentially while that of CG almost linearly due to the fact that CG is able to determine the solution by scanning far fewer number of configurations. Please note that the runtime of CG was on the Intel Core i3-2467 M @ 1.60GHz and by application of the parallelism on 12 icores (Since $\boldsymbol{T = 12}$), the run time can be reduced to few seconds. Moreover, the application CG allows scalability of the proposed platform for very large scales.

Table 9 List of Assumptions and alternative possibilities: Possible Topologies of Datacenters

Topology Type	*Tree Family (Fat-Tree, Clos-Network)*	Cube Family (Bcube, MDCube, CamCube)	DCell	Ficonn	Scafida	Jelly Fish
References	[44–48]	[49–51]	[52]	[53]	[54]	[55]

Table 10 List of Assumptions and alternative possibilities: Possible Communication Models among job components

Communication Layer	VM-to-VM (IP-level)		Transport Layer Flow	Application Layer (App. to App.)
Model	*Full Mesh*	Graph	Graph	Graph
References	[56–60]	[61]	[62, 63]	[64–67]

Discussion on assumptions

In this section, assumptions made in this work are listed and evaluated. Proposed assumptions along with alternative possibilities are shown in in Tables 9, 10, 11 and 12 (assumptions of this paper are in italics).

The first Assumption made in this paper is related to Topology. We assumed Fat-Tree Topology. However, the analysis in this work, without loss of generality, can be extended to other tree cloud topologies types. The main reason behind this selection is that more than 70% of the cloud datacenters have tree-based architecture and to make the analysis realistic it is better to consider the most common scenarios. For more information, please refer to [41–43]. It is worth mentioning that this research focused on the infrastructure of large-scale hosted datacenters which is responsible for the management and maintenance of the data and processing jobs of many different companies. Thus, this research is better suited for public cloud scenarios.

Different communication demand models are considered in the literature. Communication demand models of cloud jobs are investigated in different layers as represented in Fig. 14. For instance, communication model can be defined for a cloud-based web application or a map-reduce processing job according to a graph at the application layer. Communication demand among cloud components also can be defined at transportation layer according to the socket (port and IP). In this paper, the prevalent definition of communication model is defined according to the IP addresses at network layer. Please note that, if at least one application resides in a VM communicate with another application on another VM, there is a communication link between two

Table 11 List of Assumptions and alternative possibilities: Common assumption on distribution of data center traffic

Traffic Distribution	*Gaussian Process*	Poisson Process	Other Types (Self-Similar, Weighbull)
References	[68–71]	[72–76]	[77–80]

Table 12 List of Assumptions and alternative possibilities: Power consumption Models of servers as a function of workload

Power consumption model	ON/OFF			DVFS
	Linear	Non-Linear	Multi-State	
References	[81–84]	[85–88]	[89]	[90–92]

VMs. Moreover, in many cloud management platform like OpenStack, there is always some communication overheads among the components (ComputeNodes). Thus, in this paper, we assumed that all the VMs of a job communicate with each other at least once and the graph model among the VMs can be approximated by a Full-mesh model. Consequently, the magnitude of demand between two servers will be assumed to be proportional to the product of the number of VMs assigned to that job on the two servers. For instance, for a job presented in Fig. 14, there is only two communication demand (2×1) exist among the VMs of a job. In Table 10 other works in the literature that assume the mesh model are listed.

A workload of the cloud computing datacenters can be approximated by different processes. Poisson and Gaussian processes are widespread in this approximation. However, for high scale scenarios, due to the central limit theorem, the Gaussian process is more realistic. Many works, as listed in Table 11, applied Gaussian process regression to approximate the Datacenter traffics. Thus, as the size of the public clouds increases, the analysis of this research is more reliable and trustworthy. Please also note that it is too complicated to schedule the unpredicted workload in real time. The power consumption models of cloud computing servers are also listed in Table 12. Many works have found a strong linear relationship between the workload and total power consumption by a server so that the power consumption by a server increases linearly with the growth of server workload from the value of the power consumption in the idle state up to the power consumed when the server is fully utilized. As it explained earlier, it is assumed that the incoming workload (traffic and process) follows a Gaussian distribution. The Linear combination (summation) of Gaussian processes also follows a Gaussian distribution. As the best of our knowledge, the linearity assumption between the power consumption and the workload (Traffic and Process) has been controversial. So, one of the limitations of this paper is that it is constrained to this linear relationship. However, to avoid inaccuracy, the Gaussian assumption is made at the Rack level.

Conclusion

In this paper, we have studied optimization of power consumption in cloud computing centers through VM placement. We have developed joint optimization of power consumption of servers, network communications, and cost of migration with workload and server heterogeneity subject to resource and bandwidth constraints for a cloud computing center with hierarchical network topology. Optimization results in an IQP that can only be solved for systems with small sizes, then we show application of the CG technique that enables solution of systems with larger sizes. CG technique has an approximation as it solves continuous relaxation of the problem, which requires rounding of the solution to integer values. Comparison of the results of CG with IQP shows the accuracy of CG resulting in

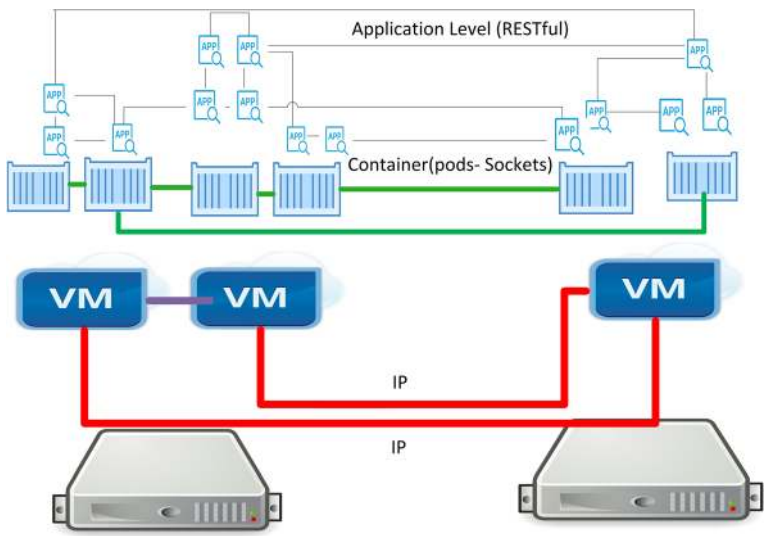

Fig. 14 Communication demand models in different layers

optimality gap less than 2%. Then, the optimization has been extended to manage temporal variation of the workload, which allows arrival and departure of jobs at the discrete-time instants. The system performs re-optimization of the power consumption under the new workload that also includes cost of migration. The numerical results show that optimization achieves power savings compared to the heuristic VM placement algorithms. In general, the field is short of work that solves optimization of power consumption problem and we hope that our work will help to bridge this gap. As far as we know this is the first work that applies CG technique to solve this problem. Results of this work may also be used to test accuracy of future heuristics for VM placement in cloud computing centers. The presented optimization method could also be used for the systems based on containers instead of VMs. We believe that the proposed optimization will be helpful to cloud service providers in realization of energy saving.

Abbreviations

CG: Column generation; CPU: Central processing unit; CS: Core switch; DVFS: Dynamic voltage frequency scaling; ILP: Integer linear programming; IQP: Integer quadratic programming; NIC: Network interface card; PoD: Performance Optimized modular Data Centers; PVA: Peer VM Aggregation; RMP: Restricted master problem; ToRs: Top of Rack switch; VM: Virtual machine

Acknowledgements

I would like to thank Professor Mehmet Ali for his help and guidance in this project.

Funding

There is no external funding and all the publication expense is paid by the author.

Availability of data and materials

All the IBM ILOG CPLEX code for IQP-ILP and CG will be available on http://www.synchromedia.ca/user/637.

Authors' contributions

There is only one Author who did everything.

Authors' information

Shahin Vakilinia (S'07) received the B.Sc. degree from University of Tabriz, Tabriz, Iran and the M.Sc. degree from Sharif University of Technology, Tehran, Iran, both in electrical engineering in 2008 and 2010 respectively. He has got his Ph.D in the Department of Electrical and Computer Engineering at Concordia University, Montreal, QC, Canada in 2015. His current research interests are in the area of Wireless Networks, C-RAN, Cloud Computing, Network Virtualization, Data Center Networks Design, and Optimization. He has published more than 30 conference and journal papers. He is currently involved with Ericsson Research Team.

Competing interests

The author declares that he has no competing interests.

Publisher's Note

Received: 2 August 2017 Accepted: 21 December 2017
Published online: 08 January 2018

References

1. Varasthe A, Goudarzi M (2015) "Server consolidation techniques in virtualized data centers: a survey", accepted to publication. IEEE Syst J
2. Ghribi C, Hadji M, Zeghlache D (2013) Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms. In: The proceedings of 13th IEEE/ACM international symposium on cluster, cloud, and grid computing, pp 671–678
3. Abts D, Marty MR, Wells PM, Klausler P, Liu H (2010) Energy proportional datacenter networks. In: The Proceedigs of ISCA, pp 338–347
4. Li X, Wu J, Tang S, Lu S (2014) Let's stay together: towards traffic aware virtual machine placement in data centers. In: The proceeding of the 33rd IEEE international conference on computer communications, INFOCOM
5. Gandhi A, Harchol-Balter M (2011) How data center size impacts the effectiveness of dynamic power management? In: The proceedings of 49th annual Allerton conference on communication, control, and computing, USA, Allerton, pp 1164–1169
6. Gandhi A, Harchol-Balter M, Raghunathan R, Kozuch MA (2012) Autoscale: Dynamic, robust capacity management for multi-tier data centers. ACM Trans Comput Syst 30(4):14–26
7. Wang L, Zhang F, Aroca J, Vasilakos A, Zheng K, Hou C, Li D, Liu Z (2014) Green DCN: a general framework for Acheving energy efficiency in data center Newtworks. IEEE J Sel Areas Commun 32(1):4–15
8. Wang X, Yao Y, Wang X, Lu K, Cao Q (2012) CARPO: correlation-aware power optimization in data center networks. In: The proceedings of IEEE INFOCOM conference, pp 1125–1133
9. Li D, Wu J (2014) Joint power optimization through VM placement and flow scheduling in data centers. In: The proceedings of IEEE international performance computing and communications conference (IPCCC), pp 1–8
10. Jin H, Cheocherngarn T, Levy D, Smith A, Pang D, Liu J, Pissinou N (2013) Joint host-network optimization for energy-efficient data center networking, IEEE 27th international symposium on parallel & distributed processing, pp 623–634
11. Dai X, Wang JM, Bensau B (2016) Energy efficient virtual machines scheduling in multi-tenant data centers. IEEE Trans Cloud Comput 4(2): 210–221
12. Ou Z, Zhuang H, Lukyanenko A, Nurminen J, Hui P, Mazalov V, Yla-Jaaski A (2013) Is the same instance type created equal? Exploiting heterogeneity of public clouds. IEEE Trans Cloud Comput 1(1):201–214
13. Zhang Q, Boutaba R, Hellerstein L et al (2014) Dynamic heterogeneity-aware resource provisioning in the cloud. IEEE Trans Cloud Comput 2(1):14–28
14. Takouna I, Rojas-Cessa R, Meinel C (2013) Communication aware and energy efficient Sceduling for parallel applications in virtualized data centers. In: The proceedings of IEEE/ACM 6th international conference on utility and cloud computing, pp 251–255
15. Cedric F, Liu H, Koley B, Zhao X, Kamalov V, Gill V (2010) Fiber optic communication technologies: What's needed for datacenter network operations. IEEE Commun Mag 48(7):32–39
16. Ballani H, Costa P, Karagiannis T, Rowstron A (2011) Towards predictable datacenter networks. In: ACM SIGCOMM computer communication review, vol. 41, no. 4, pp 242–253
17. Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley, New York
18. Chvatal V (1983) Linear programming. Macmillan. W. H. Freeman and Company, New York - San Francisco
19. Lübbecke A, Marco E, Desrosiers J (2005) Selected topics in column generation. Oper Res 53(6):1007–1023
20. de Panne V, Cornelis C, Whinston A (1964) The simplex and the dual method for quadratic programming. Oper Res Q 15(4):355–388
21. Beer K, Käschel J (1979) Column generation in quadratic programming. Math Operations Stat, Series Optimization 10(2):179–184
22. Zhang J, Huang H, Wang X (2016) Resource provision, on algorithms in cloud computing: a survey. J Netw Comput Appl 64:23–42
23. Xu D, Liu X, Niu Z (2014) Joint resource provisioning for internet datacenters with diverse and dynamic traffic. IEEE Trans Cloud Comput 7161(99):1–14
24. Xu H, Li B (2012) Cost efficient datacenter selection for cloud services. In: The proceedings of IEEE 1st international conference on Communications in China (ICCC), pp 51–56

25. D. Niu, C. Feng, B. Li, "Pricing cloud bandwidth reservations under demand uncertainty", in ACM SIGMETRICS performance evaluation review, vol. 40, no. 1, pp. 151-162, 2012
26. Zhang X, Wang H, Xu Z, Wang X (2014) Power attack: an increasing threat to data centers. In: The proceedings of the network and distributed system security symposium, NDSS, pp 132–147
27. Huang Q, Gao F, Wang R, Qi Z (2011) Power consumption of virtual machine live migration in clouds. In: The proceedings of the third international conference on communications and mobile computing (CMC), pp 122–125
28. Hinxman AI (1980) The trim-loss and assortment problems: a survey. Eur J Oper Res 5(1):8–18
29. Wäscher G, Gau T (1996) Heuristics for the integer one-dimensional cutting stock problem: a computational study. Oper Res Spectrum 18(3):131–144
30. Poldi KC, Nereu Arenales M (2009) Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. Comput Oper Res 36(6):2074–2081
31. de Carvalho JV (2002) LP models for bin packing and cutting stock problems. Eur J Oper Res 141(2):253–273
32. S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing", in The proceedings of the power aware computing and systems conference, Berkeley, CA, USA, pp. 1-15, 2009
33. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst 28(5):755–768
34. Vakilinia S, Mehmet-Ali M, Qiu D (2015) Modeling of the resource allocation in cloud computing centers. Comput Netw 91(14):453–470
35. Barroso B, André L, Dean J, Holzle U (2003) Web search for a planet: the Google cluster architecture. IEEE Micro 23(2):22–28
36. Aleksic S (2009) "Analysis of power consumption in future high-capacity network nodes", IEEE/OSA. J Opt Commun Netw 1(3):245–258
37. Pries P, Rastin M, Jarschel M, Schlosser D, Klopf M, Tran-Gia P (2012) Power consumption analysis of data center architectures. In: Green communications and networking. Springer, Berlin Heidelberg, pp 114–124
38. Kant K (2009) Power control of high speed network interconnects in data centers. In: The proceedings of IEEE INFOCOM workshops, pp 1–6
39. Mills K, Filliben J, Dabrowski C (2011) Comparing vm-placement algorithms for on-demand clouds. In: The proceedings of the cloud computing technology and science (CloudCom) conference, pp 91–98
40. Juve G, Deelman E, Karan Vahi V, Gaurang M, Berriman B, Berman P, Maechling P (2009) Scientific workflow applications on Amazon EC2. In: The proceedings of 5th IEEE international conference on in E-science workshops, pp 59–66
41. Sankaran GC, Sivalingam KM (2017) A survey of hybrid optical data center network architectures. Photon Netw Commun 33(2):87–101
42. Suryavanshi MM (2017) Comparative analysis of switch based data center network architectures. J Multidiscip Eng Sci Technol (JMEST) 4(9):2458–9403
43. Bari MF, Boutaba R, Esteves R, Granville LZ, Podlesny M, Rabbani MG (2013) Data center network virtualization: a survey. IEEE Commun Surv Tutorials 15(2):909–928
44. Al-Fares M, Radhakrishnan S, Raghavan B, Huang N, Vahdat A (2010) Hedera: dynamic flow scheduling for data center networks. In: NSDI, vol 10, pp 19–19
45. Yoon MS, Kamal AE, Zhu Z (2017) Adaptive data center activation with user request prediction. Comput Netw 122:191–204
46. Al-Fares M, Loukissas A, Vahdat A (2008) "A scalable, commodity data center network architecture." ACM SIGCOMM Computer Communication Review, 38(4):63–74
47. Greenberg A, Hamilton JR, Jain N, Kandula S, Kim C, Lahiri P, Maltz DA, Patel P, Sengupta S (2009) Vl2: a scalable and flexible data center network. SIGCOMM Comput Commun Rev 39(4):51–62
48. Kant K (2009) Data center evolution: a tutorial on state of the art, issues, and challenges. Comput Netw 53(17):2939–2965
49. Wu H, Lu G, Li D, Guo C, Zhang Y (2009) "MDCube: a high performance network structure for modular data center interconnection", Proceedings of the 5th international ACM conference on Emerging networking experiments and technologies (CoNEXT). Rome, pp. 25–39
50. Guo C, Lu G, Li D, Wu H, Zhang X, Shi Y, Tian C, Zhang Y, Lu S (2009) Bcube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM Comput Commun Rev 39(4):63–74
51. Costa, P., et al. CamCube: a key-based data center. Technical report MSR TR-2010-74, Microsoft Research, 2010
52. Guo C, Wu H, Tan K, Shi L, Zhang Y, Lu S (2008) DCell: a scalable and fault-tolerant network structure for data centers. ACM SIGCOMM Comput Commun Rev 38(4):75–86
53. Li, Dan, et al. "FiConn: using backup port for server interconnection in data centers", In Proceeding of IEEE INFOCOM, pp. 2276-2285, 2009
54. Gyarmati L, Trinh TA (2010) Scafida: a scale-free network inspired data center architecture. ACM SIGCOMM Comput Commun Rev 40(5):4–12
55. Singla, Ankit, et al. "Jellyfish: networking data centers randomly" 9th USENIX symposium on networked systems design and implementation (NSDI), vol. 12, pp. 17–17, 2012
56. Benson T, Anand A, Akella A, Zhang M (2010) Understanding data center traffic characteristics. ACM SIGCOMM Comput Commun Rev 40(1):92–99
57. Guo J, Liu F, Huang X, Lui J, Hu M et al (2014) On efficient bandwidth allocation for traffic variability in datacenters. In: Proceeding of IEEE INFOCOM, pp 1572–1580
58. Meng X, Pappas V, Zhang L (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proceeding of IEEE INFOCOM, pp 1–9
59. Vakilinia S, Cheriet M, Rajkumar J (2016) Dynamic resource allocation of smart home workloads in the cloud. In: Proceeding of 12th IEEE international conference on network and service management (CNSM), pp 367–370
60. Fang W, Liang X, Li S, Chiaraviglio L, Xiong N (2013) VMPlanner: optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. Elsevier Comput Netw 57(1):179–196
61. Ataie E, Entezari-Maleki R, Rashidi L, Trivedi KS, Ardagna D, Movaghar A (2017) Hierarchical stochastic models for performance, availability, and power consumption analysis of IaaS clouds. IEEE Trans Cloud Comput 6(1):12–26
62. Benson T, Anand A, Akella A, Zhang M (2011) MicroTE: fine grained traffic engineering for data centers. In: Proceedings of the seventh ACM conference on emerging networking experiments and technologies, p 8
63. Kliazovich D, Pecero JE, Tchernykh A, Bouvry P, Khan SU, Zomaya AY (2016) CA-DAG: modeling communication-aware applications for scheduling in cloud computing. J Grid Comput 14(1):23–39
64. Kliazovich D, Pecero JE, Tchernykh A, Bouvry P, Khan SU, Zomaya AY (2013) CA-DAG: communication-aware directed acyclic graphs for modeling cloud computing applications. In: Proceeding of IEEE sixth international conference on CLOUD computing (CLOUD), pp 277–284
65. Redekopp M, Simmhan Y, Prasanna VK (2013) Optimizations and analysis of bsp graph processing models on public clouds. In: Proceeding of 27th international IEEE symposium on parallel & distributed processing (IPDPS), pp 203–214
66. Vakilinia S, Zhang X, Qiu D (2016) Analysis and optimization of big-data stream processing. In: Proceeding of IEEE global communications conference (GLOBECOM), pp 1–6
67. Phuong PT, Durillo JJ, Fahringer T (2017) Predicting workflow task execution time in the cloud using a two-stage machine learning approach. IEEE Trans Cloud Comput 6(4):121–134
68. Malboubi M et al (2016) Decentralizing network inference problems with multiple-description fusion estimation (mdfe). IEEE/ACM Trans Networking 24(4):2539–2552
69. Bayati A, Asghari V, Nguyen K, Cheriet M (2016) Gaussian process regression based traffic modeling and prediction in high-speed networks. In: Proceeding of IEEE global communications conference (GLOBECOM), pp 1–7
70. Gomez-Miguelez I, Marojevic V, Gelonch A (2013) Deployment and management of SDR cloud computing resources: problem definition and fundamental limits. EURASIP J Wirel Commun Netw 1(1):59–72
71. Lor SS, Vaquero LM, Murray P (2012) In-netdc: the cloud in core networks. IEEE Commun Lett 16(10):1703–1706
72. Dalmazo BL, Vilela JP, Curado M (2014) Onlinetraffic prediction in the cloud: a dynamic window approach, proceeding on IEEE cloud and green computing (CGC), pp 9–14
73. Dalmazo BL, Vilela JP, Curado M (2013) Predicting traffic in the cloud: a statistical approach, proceeding on IEEE cloud and green computing (CGC), pp 121–126
74. Y. Min Sang, A. E. Kamal, and Z. Zhu. "Requests Prediction in Cloud with a Cyclic Window Learning Algorithm" Globecom Workshops (GC Wkshps), 2016 IEEE. IEEE, 2016
75. Min Sang Y, Kamal AE, Zhu Z (2017) Adaptive data center activation with user request prediction. Comput Netw 1(22):191–204
76. Yin J, Lu X, Zhao X, Chen H, Liu X (2015) Burse: a bursty and self-similar workload generator for cloud computing. IEEE Trans Parallel Distrib Syst 26(3):668–680

77. Kandula S, Sengupta S, Greenberg A, Patel P, Chaiken R (2009) The nature of data center traffic: measurements & analysis. In: Proceedings of the 9th ACM SIGCOMM conference on internet measurement conference, pp 202–208
78. Benson T, Akella A, Maltz DA (2010) Network traffic characteristics of data centers in the wild. In: Proceedings of the 10th ACM SIGCOMM conference on internet measurement, pp 267–280
79. Zhang L, Li Z, Wu C, Chen M (2014) Online algorithms for uploading deferrable big data to the cloud. In: Proceeding of IEEE INFOCOM, pp 2022–2030
80. Vakilinia S, Heidarpour B, Cheriet M (2016) Energy efficient resource allocation in cloud computing environments. IEEE Access 4:8544–8557
81. Venkatachalam V, Franz M (2005) Power reduction techniques for microprocessor systems. ACM Comp Surv J CSUR 37(3):195–237
82. Minas L, Ellison B (2009) Energy efficiency for information technology: how to reduce power consumption in servers and data centers. Intel Press, USA
83. Fan X, Weber WD, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: Proceedings of the 34th annual international symposium on computer architecture (ISCA), pp 13–23
84. Paul D, Zhong WD, Bose SK (2017) Demand response in data centers through energy-efficient scheduling and simple incentivization. IEEE Syst J 11(2):613–624
85. Aikebaier A, Enokido T, Takizawa M (2009) Energy-efficient computation models for distributed systems. In: Proc. of the 12th international conference on network-based information systems (NBiS), pp 424–431
86. Enokido T, Suzuki K, Aikebaier A, Takizawa M (2010) Process allocation algorithm for improving the energy efficiency in distributed systems. In: Proc. of IEEE the 24th international conference on advanced information networking and applications (AINA), pp 142–149
87. Enokido T, Aikebaier A, Takizawa M (2011) Process allocation algorithms for saving power consumption in peer-to-peer systems. IEEE Trans Ind Electron 58(6):2097–2105
88. Enokido T, Takizawa M (2012) An extended power consumption model for distributed applications. In: Proceeding of 26th IEEE international conference on advanced information networking and applications (AINA), pp 912–919
89. Inoue T, Aikebaier A, Enokido T, Takizawa M (2013) Power consumption and processing models of servers in computation and storage based applications. Math Comput Model 58(5):1475–1488
90. Wu CM, Chang RS, Chan HY (2014) A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. Futur Gener Comput Syst 37:141–147
91. Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2008) Power and performance management of virtualized computing environments via lookahead control. In: Autonomic computing, (ICAC), international conference on, pp 3–12
92. Mobius C, Dargie W, Schill A (2014) Power consumption estimation models for processors, virtual machines, and servers. IEEE Trans Parallel Distrib Syst 25(6):1600–1614