To appear in *IEEE Trans. on Computer Aided Design*, Vol. 31, No. 10, Oct. 2012.

Energy-Efficient Datacenters

Massoud Pedram, Fellow, IEEE

Abstract—Pervasive use of cloud computing and the resulting rise in the number of datacenters and hosting centers (which provide platform or software services to clients who do not have the means to set up and operate their own compute facilities) have brought forth many concerns including the electrical energy cost, peak power dissipation, cooling, carbon emission, etc. With power consumption becoming an increasingly important issue for the operation and maintenance of the hosting centers, corporate and business owners are becoming increasingly concerned. Furthermore, provisioning resources in a cost-optimal manner so as to meet different performance criteria such as throughput or response time has become a critical challenge. The goal of this paper is to provide an introduction to resource provisioning and power/thermal management problems in datacenters and to review strategies that maximize the datacenter energy efficiency subject to peak/total power consumption and thermal constraints while at the same time meeting stipulated service level agreements in terms of task throughput and/or response time.

Index Terms— Datacenter, Enterprise Computing, Energy Efficient Design, Green Computing, Resource Management, Dynamic Power and Thermal Management.

I. INTRODUCTION

Demand for computing power has been increasing due to the penetration of information technologies in our daily interactions with the world both at personal and communal levels, encompassing business, commerce, education, manufacturing, and communication services. At the personal level, the wide scale presence of online banking, e-commerce, social networking, etc. produce workloads of great diversity and enormous scale. At the same time, computing and information processing requirements of various public organizations and private corporations have been increasing rapidly. Examples include digital services and functions required by various industries, ranging from manufacturing to housing, and from transportation to banking. Such a dramatic increase in the computing resources requires a scalable and dependable *information technology* (IT) infrastructure comprising of servers, storage, network bandwidth, physical infrastructure, Electrical Grid, personnel and billions of dollars in capital expenditure and operational cost to name a few.

Manuscript received December 14, 2011. This work was supported in part by the National Science Foundation.

Massoud Pedram is with the Department of Electrical Engineering at the University of Southern California. Correspondences may be sent to pedram@usc.edu.

Datacenters are the backbone of today's IT infrastructure. The reach of datacenters spans a broad range of application areas from energy production and distribution, complex weather modeling and prediction, to manufacturing, transportation, entertainment, and even social networking. There is a critical need to continue to improve efficiency in all these sectors by accelerated use of computing technologies, which inevitably requires increasing the size and scale of datacenters.

A. The Case for Energy-Efficient Datacenters

By some estimates, datacenter energy consumption has nearly quadrupled in the past decade [1], as more and increasingly powerful servers are brought online to answer search queries, stream audio and video content, complete online transactions, and perform analysis and forecasting in almost every sector of society and the economy. The increased use of "cloud" computing and SaaS (Software-as-a-Service) has greatly accelerated the trend and given rise to anxiety about an impending datacenter energy crisis.

Datacenters are faced with a major impediment of power consumption. To put their energy consumption in perspective, consider a U.S. Environmental Protection Agency (EPA) report to Congress [2], in which it is reported that U.S. datacenters consumed 61 billion kilowatt-hours of power in 2006, which constitutes 1.5% of all power consumed in the U.S. and represents a cost of $4.5 billion.

Electricity consumption of datacenters is among the fastest growing sources of global electricity demand. In the U.S., which hosts approximately 40% of the world's datacenter servers, datacenter electricity consumption increased by nearly 40% even during the economic downturn of 2007-2010 [3]. It was reported in [4] that the datacenter power consumption from servers, storage, communications, cooling, and power distribution equipment accounts for between 1.7% and 2.2% of total electricity use in the U.S. in 2010. This is up from 0.8%of total U.S. power consumption in 2000 and 1.5% in 2005 [5]. Electricity used in U.S. datacenters in 2010 was, however, lower than predicted by the EPA's 2007 report to Congress on datacenters—The 93% growth (Best Guess in 2007) is revised down to 53% growth. Interestingly, the lower range of the EPA's projected power consumption in 2007 assumed that increased virtualization and increased use of technology to cut server power consumption would be responsible for reduced energy consumption in U.S. datacenters. In reality, however, the reduced electricity growth rate compared to earlier estimates was driven mainly by a lower installed server base than was

earlier predicted rather than the efficiency improvements anticipated in the EPA report to Congress [4][5].

As the U.S. economy starts to grow again, the installed server base growth and the corresponding electricity growth rate for datacenters are expected to rise more rapidly (in spite of widespread adoption of virtualization technology which is expected to cut the need for more physical servers).

As an example of datacenter growth, consider Google, which is not only the world's largest server user, but also one that assembles its own servers featuring their own proprietary power saving technology. New York Times reports that Google was running about 1 million volume servers in 2010, up from 25,000 in 2000 and 350,000 in 2005 [5]. Google recently revealed that its datacenters continuously drew almost 260 million watts - about a quarter of the output of a nuclear power plant - to run Google searches, YouTube views, Gmail messaging and display ads on all those services [6]. This is about 0.01% of total worldwide electricity use and less than 1% of global *datacenter electricity use.*

Apart from the total energy consumption, another critical component is the peak power dissipation—in 2006, the peak load on the power grid from datacenters was estimated to be approximately 7 Gigawatts (GW), equivalent to the output of about 15 base-load power plants [2]. This load is increasing as shipments of high-end servers used in datacenters (e.g., blade servers) are increasing at a 20-30% compound annual growth rate. Yet another key factor in building and operating datacenters is the need to cool the IT equipment in the datacenter. To take advantage of the cold climate to cool tens of thousands of servers, and thereby, reduce the electrical energy cost of cooling and air-conditioning equipment, many companies are moving their latest and largest datacenters to cold places. For example, it has recently been reported that Facebook will be building its newest datacenter near the Arctic Circle in Sweden [7].

The environmental impact of datacenters was estimated to be 116.2 million metric tons of CO2 in 2006 [2]. Google alone used about 2.26 million megawatt hours of electricity to run its datacenters in 2010, generating a carbon footprint of 1.46 million metric tons of carbon dioxide [8]. Realizing the perils of spiraling carbon emissions growth, the Intergovernmental Panel on Climate Change (IPCC) has called for an overall greenhouse gas emissions reduction of 60-80%—below levels from 2000—by 2050 to avoid significant environmental damage.

Currently, little of the world's power is from renewable energy sources such as wind and solar. Many datacenter owners and cloud providers are working on changing this by buying electricity directly from wind/solar farms near their datacenters and by establishing an internal "price for carbon emission" (e.g., Google [9], Microsoft [10]). For example, Google has purchased 200 megawatts worth of wind power from local utility grids as a way of lowering the carbon footprint of its operations. Apple's North Carolina datacenter will be powered partly by a giant 20-megawatt solar array and nearly five megawatts of biogas-powered fuel cells. This trend will likely grow, making the datacenters and the Grid more eco-friendly. An important consideration in datacenters is to remove the large chillers that can account for up to one third of a datacenter's power consumption. For its facility in Prineville, Ore., Facebook has designed a structure to maintain evaporative cooling, which keeps the datacenter cool by spraying water into incoming air. Facebook says it has designed its servers to be able to work in that hotter and more humid environment [11].

B. Sources of Energy Inefficiencies in Datacenters

Energy non-proportional servers: A datacenter comprises of thousands to tens of thousands of server machines, working in tandem to provide services to the clients, see for example [12] and [13]. Ideally systems would exhibit energy proportionality, wherein servers consume power in proportion to their load. Current servers are far from energy proportionality. Indeed, servers consume 80% of the peak power even at 20% utilization [14]. The energy non-proportional server hardware is a key contributor to energy inefficiency in a datacenter. Facts are that servers are often utilized with between 10 to 50% of their peak load and that servers experience frequent idle times of rather short duration [15] amplify this issue in the datacenters. This means that servers are not working near their optimal power-performance tradeoff points most of the time, and that idle times in servers consume a big portion of the peak power. Server idle states are a problem because they are not deep enough to provide the minimum energy.

Over-provisioned server and power infrastructure: Current datacenters are way oversized. A typical datacenter is provisioned to handle the peak workload, which occurs fairly infrequently, rather than the average workload. This practice results in underutilized server hardware, which is the most significant factor contributing to excessive energy consumption in datacenters. Indeed, a large fraction of the datacenter energy consumption is due to resource over-provisioning. Note that over-provisioning would not be a problem if each server was completely energy proportional. Additionally, provisioning for the peak power consumption (which requires all servers to simultaneously draw their maximum power) has proven to be very expensive. Indeed, provisioning based on the nominal ratings of servers greatly under-utilizes the power infrastructure [16][17].

Energy-inefficient legacy server hardware: Yet another key factor in creating energy inefficiencies in datacenters is the fact that they are populated by old and energy-inefficient server hardware.

Improving energy efficiency in datacenters requires starting from scratch and replacing the obsolete equipment and facilities with a state-of-the-art datacenter. However, most datacenter owners would do anything to avoid building a new facility, and thus end up squeezing more efficiency out of their inefficient legacy facilities.

Due to strides in low-level power management and improvements in the underlying CMOS devices, VLSI circuits and process architectures, today's blade servers are much more energy efficient than the ones that were designed and deployed in datacenters only a few years ago [18].

Poor power management: Another key contributor to energy inefficiencies in datacenters is the fact the active servers in a datacenter may provide higher performance than is required by the workload. Unfortunately, performance (say low latency or high throughput) is achieved at super-linear cost in energy consumption. So servers working harder than they should can contribute to energy inefficiency in a datacenter. Note that again poor power management is an important factor because of energy non-proportionality of the current servers. In addition, the allocation of server resources to clients or the assignment of tasks to servers may result in server utilization inefficiencies.

Multiple power conversions and low UPS efficiency: Yet another reason for energy inefficient datacenters is the need for multiple power conversions in a datacenter's power distribution system. In particular, the main AC feed coming from the Grid is first connected to DC so that it can be used to charge the battery backup system. The output of this electrical energy storage system then goes through an inverter to produce AC power, which is then distributed throughout the datacenter. Finally the AC power is converted to various DC levels to support various subsystems of a blade server. These conversions are required due to the (oversized and highly redundant) Uninterruptible Power Supply (UPS) modules, which are deployed in the datacenters for voltage regulation (i.e., to remove power spikes) and power backup (e.g., in case of a power failure on the primary AC feed, switch to a secondary feed coming from the battery bank or a set of locally operated diesel generators). Note that most UPS modules in a datacenter operate at 10-40% of their full load capacity only [19] . Unfortunately, at these low load levels, the UPS conversion efficiency (AC-SC-AC) is quite low.

Energy cost of cooling and air conditioning units: Energy has to be devoted to cooling the operating environment, especially given the amount of waste heat generated by today's high-performance processors. Accounting for about 30% of the total energy cost of a datacenter (another 10-15% is due to power distribution and conversion losses in the datacenter), the cooling cost is one of the major contributors of the total electricity bill of large datacenters [20]. These values are shrinking by introducing new cooling techniques and new server and rack configurations for datacenters. They are also smaller for datacenters located in good geographical locations so that they can benefit from ambient cooling. Yet, cooling-related energy consumption is still a significant contributor to the energy inefficiencies in datacenters since it is not energy that is used for performing client-related services.

According to some reports, the physical infrastructure (e.g., the power backup and distribution system and the cooling and air conditioning systems) tends to account for 40-50% of the total datacenter power dissipation [21][24].

C. Power Usage Effectiveness Metric

The *Power Usage Effectiveness* (PUE) rating, which reveals how much power is lost in power distribution and conversion as well as in cooling and air conditioning in a datacenter, is calculated as the ratio of the total power consumption in a datacenter to the total IT equipment power consumption [25]. Note that some prefer to use the reciprocal of PUE, which is known as the *Datacenter infrastructure Efficiency,* or DCiE for short.

The PUE metric has been steadily coming down over the last decade. In 2003, the PUE metric for a typical datacenter was estimated to be about 2.6 [26]. In 2010 Koomey estimated that the average PUE was between 1.83 and 1.92 [4]. Some of the recent datacenters built by Google, Facebook, and Microsoft have pushed PUEs under 1.2 or even 1.1 [27][28]. The U.S. Environmental Protection Agency has chosen to launch its Energy Star for Datacenters ratings using the PUE metric as the basis for its rankings [29]. In particular, the EPA's Energy Star rating will be based on the average PUE ratios for a datacenter, calculated from 12 months of actual measured data.

A more accurate datacenter efficiency metric should focus on estimating the actual amount of power used by the IT equipment to do useful work. The *Datacenter Energy Efficiency* (DCEE) metric may thus be defined as follows:

$$DCEE = ITU \times ITE/PUE$$

where the *IT Utilization* (ITU) denotes the ratio of average IT use over the peak IT capacity in the datacenter, and the *IT Efficiency* (ITE) is the amount of useful IT work done per joules of energy). Note that ITU varies greatly in a 24-hour period whereas ITE is a function of the number and types of active servers and their utilization levels. The reader may refer to [30] for a metric that accounts for dirty vs. green sources of power.

D. Overcoming Energy Inefficiencies in Datacenters

Perfect provisioning of the server infrastructure: The IT infrastructure provided by the datacenter owners/operators must meet various *service agreements* established with the clients. The service agreements include compute power, storage space, network bandwidth, availability and security, etc. Today's datacenters tend to be provisioned for near-peak performance since typical service agreements between clients and hosting datacenters discourage the development of significant performance bottlenecks during peak utilization periods. Such overprovisioning may increase the cost incurred on the datacenters in terms of the electrical energy bill.

Optimal resource provisioning, which allows datacenter operators to use only the minimum resources needed to perform the incoming tasks, in a datacenter is an arduous undertaking. Optimal provisioning is complicated by the fact that over time, datacenter resources become heterogeneous

even if a datacenter is initially provisioned with homogeneous resources. For instance, replacing non-operational servers or adding a new rack of servers to accommodate demand typically leads to installing new resource types that reflect the advances in current state-of-the-art server design.

In general, datacenters serve different (sometimes independent) tasks or serve the same task for different clients. If a physical server is dedicated to each task, the number of tasks that datacenter can support will be limited to the number of physical servers in the datacenter. Also, allocating one task to each server can be energy inefficient because of the energy usage pattern of the tasks. In this context, judicious allocation of resources to clients, mapping of tasks to servers, and server consolidation strategies are some of the most promising methods.

Achieving energy-proportionality at the cluster level: It is desirable to achieve energy proportionality at the server pool or datacenter levels by dynamically moving tasks among servers and doing server consolidations so that the specific shape of the power dissipation versus utilization curve at the server level becomes less important, while the shape of the power-utilization curve at the datacenter level becomes a line that goes through the origin [31]. In addition, it has shown that energy-proportional operation can be achieved for lightly utilized servers with full-system, coordinated idle low-power modes [32]. Such a technique works well for workloads with low average utilization and a narrow dynamic range, a common characteristic of many server workloads.

Effects of using energy-proportional servers in datacenters are studied in [16]. The authors report 50% energy consumption reduction by using energy-proportional servers with idle power of 10% of peak power instead of typical servers with 50% idle power consumption. The authors show that increasing the energy efficiency of the Disk, memory, network cards and CPU helps in creating energy-proportional servers. Furthermore, Dynamic Power Management (DPM) techniques such as dynamic voltage scaling (without latency penalty) and sleep mode for Disk and CPU components (with latency penalty) improve the energy-proportionality of the servers.

Utilization of "wimpy" servers along with high-performance servers: Multicore architectures are superb for throughput-oriented computing in datacenters because they provide ample parallelism for search or analysis over very large data sets. One may classify multicore systems as *brawny-core systems*, whose single-core performance is fairly high, and *wimpy-core systems* (typically composed of ARM or Atom-based processors), whose single-core performance is low. The wimpy-core systems are significantly more energy efficient than the brawny cores.

A growing number of recent studies have focused on redesigning datacenter server clusters using wimpy nodes [33][34]. However, low-end nodes lag far behind traditional nodes in performance. Therefore, a small cluster of traditional nodes must be replaced by a larger cluster of low-end nodes [35]. One should however exercise caution when following this trend. As the number of parallel threads increases, communication overheads increase. In the limit, the amount of inherently serial work performed on behalf of a user request by slow single-threaded cores will dominate the overall execution time. According to U. Holzle of Google, once a chip's single-core performance lags by more than a factor of two or so behind the higher end of current-generation commodity processors, making a business case for switching to the wimpy-core system becomes increasingly difficult because application programmers will see it as a significant performance regression: their single-threaded request handlers are no longer fast enough to meet latency targets [36]. Other studies have shown that the parallel scaleup characteristics of the query workload and the software system largely determines the feasibility of wimpy node configuration for building clusters for such data processing workloads in datacenters [37].

Upgrading the IT infrastructure: High energy efficiency in a datacenter may be achieved by replacing legacy datacenter equipment with more-powerful and energy-efficient state-of-the-art servers. Newer servers use more advanced internal cooling systems, that is, they are engineered to optimize airflow to cool internal components with less power consumed by their fans. These servers use front-to-back cooling with straight-through airflow, and their fan speeds are modulated by measured internal temperatures. This is important because internal server power consumption reductions are typically amplified by savings in the rack and datacenter power distribution and cooling infrastructures.

To increase the storage power efficiency in a datacenter, one can deploy hybrid storage systems with solid-state drives mixed with serial-attached SCSI (Small Computer System Interface) or serial ATA drives. This setup provides performance similar to previous-generation high-end systems, but using only one-half of the energy consumption. Efficiency can also be increased by choosing solid-state drives that use less power because they have no moving parts.

IT equipment upgrades, however, come at a significant capital expenditure (CapEx). A datacenter owner typically weighs CapEx vs. operational expenditure (OpEx) in deciding how much of an equipment upgrade is economically beneficial. Indeed, a typical datacenter houses 3-4 generations of servers (ranging from newer 64-bit, multi-core AMD Opteron and Intel Xeon processors to older 32-bit, single core processors), with the energy-per-instruction cost of the least efficient ones being typically 2-3 times higher than that of the most efficient one. A smart datacenter operator should utilize the most energy efficient servers to run its typical average load and bring the older, less energy efficient servers online only to meet the peak demand.

Deploying heterogeneous multi-core processors: Computer architects and processor chip manufacturers have begun designing heterogeneous chip multi-processors (also

known as asymmetric multi-cores) that consist of at least one large, high-performance core and several small, low-performance cores, all of which expose the same instruction-set-architecture [38][39]. This trend can help improve the performance and energy efficiency of the server hardware under various workload conditions (requiring high performance for both single-thread execution and multi-threaded applications) by making it possible for different applications within a diverse mix of workloads to be run on the "most appropriate" cores.

Recent studies have shown that, compared to homogeneous processor configurations, heterogeneous core architectures can provide significant performance enhancements while also lowering the energy consumption for many applications. These studies also demonstrate that potential savings are strongly influenced by the 'uncore' (components like the last level cache and integrated memory controllers) contribution to overall power consumption, which motivates the need for 'uncore'-awareness in managing heterogeneous multicore platforms and architecting a scalable 'uncore' design to fully realize the intended gains [40].

Optimal power management: System-wide power management is a key technique for improving energy efficiency in datacenters. The power management issue, however, has a few different aspects. First, there is the question of total cost of ownership for the datacenters, which includes the electrical energy cost of operating a datacenter. To minimize this cost, the datacenter's total power dissipation (including the power dissipation in the IT equipment and the physical infrastructure) must be decreased. A trade-off exists between power consumption and performance of the system, and the datacenter's power manager should consider this trade-off carefully when issuing commands. Second, there is the question of the peak capacity of the power source(s) for a datacenter and electrical current limitations of the power delivery network in the datacenter, which in turn set a limit on the peak power draw at the server, chassis, rack, and datacenter levels. This is known as the *power capping* problem. A key challenge in the power capping arises from the distributed nature of power consumption in the datacenter. For example, if there is a power budget for a rack in the datacenter, the problem is how to allocate this budget to different servers and how to control this budget in a distributed fashion.

Improving efficiency of the datacenter power distribution solution: The IT infrastructure provided by large datacenter owners/operators is often *geographically distributed*. This helps with reducing the peak power demand of the datacenters on the local power grid, allowing for more fault tolerance and reliable operation of the IT infrastructure, and even reduced cost of ownership.

As stated earlier, much of a datacenter's power goes to converting AC to DC voltage in the power supplies and regulating voltage on the motherboard (not to mention protecting its reliability with backup schemes), but those components are almost always designed for price, not efficiency. There are, therefore, no easy or quick fixes–One technique is to correctly manage the UPS load capacity and to use modular UPS that maximizes the load capacity of UPS; the other is to use DC power distribution throughout the server room and the datacenter [19]. There are numerous points such as power supplies and UPS, where the incoming AC power can be converted into DC, giving rise to different tradeoffs. See [41] for a carefully crafted paper highlighting the benefits and drawbacks of AC and DC power without fully endorsing either approach.

Maximizing cooling efficiency: One way to lower the energy cost of cooling a datacenter is to deploy *computer room air conditioning* (CRAC) units and air handling units with demand-driven, *variable frequency drive* (VFD) fans within heat exchangers so as to match variable heat loads with variable airflow rates. This solution better matches actual cooling operations to cooling needs as systems go through different use cycles from fully idle to fully used. VFDs are also available in chilled water pumps, chillers, and cooling tower fans. Other techniques include hot aisle containment (to avoid mixing hot and cold air and feeding hotter exhaust air to the cooling units), directed spot cooling for racks with the highest heat loads, and new rack designs that include a passive rear door heat exchanger to provide localized cooling [42].

Orthogonal to this approach is to develop effective (closed-loop) thermal management at the datacenter level. Hot spots can be avoided and power can be saved by raising the outlet temperature of the CRAC units or reducing the speed of the fans. This is because the datacenter facility can be kept at room temperature without requiring the cooling system to work very hard. An example of a dynamic thermal management decision is to migrate all running tasks from a hot server and, subsequently, turning it off.

Bonus–Cloud computing to replace corporate datacenters: A positive development is the hastening use of cloud systems to replace private datacenters. Cloud systems are less expensive to operate, consume less energy, and have higher utilization rates than traditional datacenters, which lead to the belief that much of the work done in internal (corporate) datacenters today will be pushed to the cloud by the end of the decade. Software-as-a-Service, Infrastructure-as-a-Service and Platform-as-a-Service are inherently more efficient than conventional alternatives, and their adoption will be one of the largest contributing factors to the greening of enterprise IT. Continued adoption of cloud computing (with a predicted compound annual growth rate of 29%) will have major implications on datacenter energy consumption. Ultimately, cloud computing has the potential to chop datacenter energy consumption by 31% from 2010 to 2020 [21].

According to James Hamilton, an Amazon VP of cloud computing services, large-scale datacenter practices provide the following key benefits [22]. i) Server, networking and administration costs for a cloud provider are five to seven

times lower than those for an average private provider; ii) The actual cost of power consumed by the servers plus the cost of cooling the servers is 34% of the total cost of ownership of a datacenter, whereas the amortized server costs in a 10-year lifetime of a datacenter is 54% of the total cost. This means that focusing on getting better value from servers and reducing the datacenter power consumption and cooling costs produce the biggest savings for a datacenter operator; iii) Turning off a server is not as economically prudent as using the server fully at all times. This may be exploited, e.g., by using a *spot pricing model*.

It is also important to note the potential of cloud systems to provide economies of scale. They achieve these economies through not only size but also focus. Cloud providers have a key ability to deliver IT services not only at lower cost but also faster, easier, and more flexibly. Their ability to focus and spend resources to achieve economies of scale is something that private datacenters cannot compete with. With a market approaching $200 billion in overall size and perhaps trillions in IT-related expenditures [23], ensuring the energy efficiency of cloud systems is one of the most important challenges for enterprises, vendors, and service providers alike.

E. Paper Overview and Outline

This paper provides a review of and a perspective on important issues related to the design and management of energy efficient datacenters. In addition, it introduces a general framework for resource management problem formulations, accounting for power dissipation and thermal issues as well as performance constraints. Finally, the paper summarizes a sample of some important approaches for addressing the aforesaid problems in the context of the aforesaid problem formulation framework. The review is by no means comprehensive, but aims to present the key problems along with some representative approaches.

This paper is organized as follows. The datacenter organization and some key issues related to datacenter energy efficiency are given in section II. Datacenter management architectures are discussed in section III. The resource arbiter, power manager, and thermal manager agents in a datacenter are detailed in sections IV to VI. The review paper is concluded in section VII.

II. DATACENTERS: HARDWARE AND SOFTWARE

A. Cyber-Physical Organization of Datacenters

As shown in Figure 1, a datacenter is a large cyber-physical system comprising of mechanical, electrical, and IT systems running a variety of services on a multitude of server pools and storage devices connected with a multi-tier hierarchy of aggregators, switches and routers.

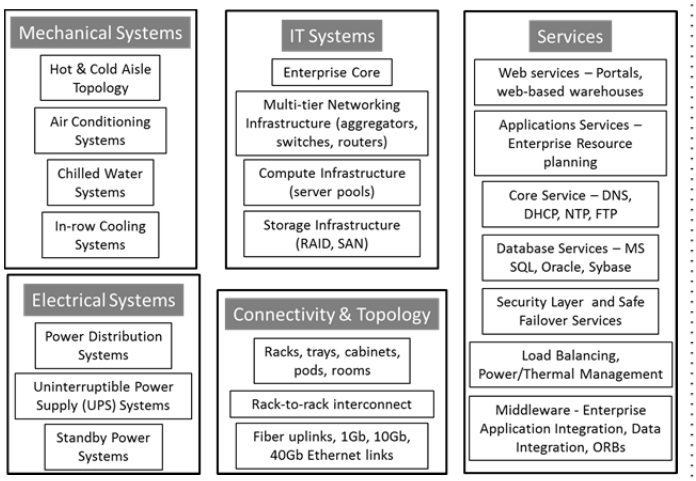

Figure 1. A high-level view of the key cyber-physical components comprising a datacenter.

Many of the public or private hosting centers, e.g., Amazon's and Google's, have been known to use containerized datacenter architecture, which can accommodate 1000-1500 servers per container. Each container or leased space (room) may contain multiple server pools, a *storage area network* (SAN), and CRAC units. All servers within each server pool are identical in terms of their processor configuration, amount of memory and hard drive capacity, and network interface card. Server pools are arranged in rows where each row contains a number of racks. In some cases *global distributed file system* (GFS) is used. GFS manages the disk drives that are connected to each individual server directly. This approach tends to be more complicated but it can provide better performance in some distributed searches such as Google web search [26]. A datacenter facility is comprised of IT rooms (containers), each room comprised of IT pods, each pod comprised of IT rack cabinets, each rack cabinet comprised of IT chassis, each chassis comprised of IT devices (servers, storage node). An IT pod is most commonly defined as a group of IT cabinets deployed together, typically in a row or pair of rows, usually sharing some common infrastructure element like an air handler, Power Distribution Unit (PDU), set of vented floor tiles, patch panel, router, etc. Based on the design of the datacenter, different network fabrics may be used. Intra-rack network fabrics have more bandwidth than inter-rack network fabrics because of the cost. For example each server can have a 1-Gbps link for inter-rack communication but a rack with 40 servers may have only 8 1-Gbps links for communication with servers in other racks.

Apart from the IT infrastructure, the datacenter also contains cooling infrastructure such as CRACs and In-row Cooling Systems. Typically CRACs depend on a Chilled Water System, generally located outside the datacenter facility, in order to cool the returned hot air. The cold air is passed through raised floor plenums or through ceiling attached ones. In order to reduce the hot air recirculation, the cooling infrastructure may impose hot aisle or cold aisle

containment, which efficiently isolates the hot air from cold air and is known to improve the cooling efficiency. Installed on the floor or suspended from overhead, in-row cooling units provide local, focused cooling at the rows of server cabinets that fill the datacenter. In-row cooling offers capacity and efficiency gains by moving the air conditioner from the perimeter of the room closer to the actual load.

In containerized (modular) datacenters [43], heat exchange and power distribution networks are integrated into a standard shipping container that contains servers. Chilled water is used to remove heat from flowing air in the datacenters similar to CRAC units in rack-based datacenters. The container-based datacenters show higher energy efficiency (less power delivery loss and less cooling cost) compared to traditional datacenters.

In conventional power delivery architecture for datacenters, AC power is converted to 12V DC using a bulk power factor correction supply in the rack and distributed throughout the rack to the various blade servers. On the blades the 12V power is regulated to miscellaneous rail voltages for the microprocessor, memory, network adapters, etc. by using conventional multi-phase buck regulators. More recently, researchers have been exploring other power delivery architectures, e.g., one in which the AC power is converted to a *power factor corrected* (PFC) 380V and distributed throughout the rack to the various blades. On the blades, the 380V is converted to 48V and then regulated to 1.xV for microprocessors and memory. Reference [44] shows data pointing to higher power delivery efficiency, more cost savings, and smaller power footprint of the latter datacenter power delivery architecture.

B. Operating System Software for a Datacenter

Similar to a computer, a datacenter needs operating system software to manage the resources and provide service to clients. By datacenter OS, we mean a software stack providing functionality for the overall datacenter that is analogous to what a traditional OS provides on one server. Software platforms such as the Hadoop stack, Amazon Web Services, Windows Azure, and Google's GFS / BigTable / MapReduce stack form today's de facto datacenter OS. To ease the development of software for datacenter applications and hide the complexity of a large computing system, programming frameworks like MapReduce [45], Dryad [46], and Pregel [47] are used. These frameworks automatically handle the data partitioning, distribution, and fault tolerance. The authors of [15] identify key functions of a datacenter OS as i) resource management, ii) hardware abstraction, iii) deployment and maintenance, and iv) software programming framework as detailed next.

The complexity of resource management and providing service to datacenter clients is much higher than the complexity of resource management in a desktop computer because the datacenter is composed of thousands of computers, networking and storage devices. The resource manager in the datacenter OS maps the user tasks to hardware and provides task management services. This resource manager can provide service-agreement-aware resource management or power-aware resource management if needed based on the specified clients' needs in terms of processing, memory capacity and network bandwidth. Hardware abstraction role of datacenter OS provides basic services for tasks like message passing, data storage and synchronization at the cluster computing level. Software image distribution and configuration management, monitoring service performance and quality, and triaging alarms for operators in emergency situations are examples of tasks done in the deployment and maintenance part of the datacenter OS. In addition to datacenter-level software, platform-level and application-level software are employed in a datacenter. Platform-level software provides service to the tasks assigned to servers. These services include providing common kernels, libraries and OS expected to be present in each server. Application-level software implements a specific service in a datacenter.

Similarly, the authors of [48] identify the following traits of traditional operating systems: resource sharing, data sharing between programs, programming abstractions for software development, and debugging and monitoring. They argue that these same functionalities should be provided to datacenter applications as a common layer so that these applications can dynamically share resources and easily exchange data. A key question, however, is where the division between the datacenter OS and the datacenter application will be.

C. Clients, Applications, Tasks, and Workloads

It is important to define our terminology with respect to datacenter users, software applications, tasks and types of workload. Many datacenter organizations include servers (or a server pool) that are dedicated to hosting single applications. This approach makes sense if resource utilization on these servers remains high. Otherwise, consolidation can be beneficial.

Users (clients) start sessions with a hosting datacenter in which they run their applications, e.g., an Office 2007 application or a voice recognition application. Each client's application will potentially generate many requests per active session, each request must typically be serviced within a given response time constraint. So we may have 100 simultaneous user sessions, 40% of which are running the Office app, while 60% running the voice recognition app. The duration of these sessions and the task generation rate per session may be different. So the datacenter management software may for example allocate *m* servers of type I (optimized for Office Workload) to the first group of users and *n* servers of type II (optimized for the speech recognition workload) to the second group of users.

D. Virtualization and Server Consolidation

Virtualization in datacenters refers to the process of replicating a physical server as two or more *virtual machines*

(VMs) and allocating exactly one VM to each task [49]. In this way, each task running on the physical server has the illusion that it has full control of the physical server (although in fact multiple virtual machines or tasks share the physical server). From a different – but equivalent – viewpoint, a VM may be defined as a task that runs on some physical server while being isolated from other tasks that may be running on that same server. Clearly, this technology minimizes dependency of the task on the physical hardware. A *hypervisor* is a program that allows multiple virtual machines to share a single physical machine.

Virtualization provides a new way to improve the power efficiency of the datacenters: *consolidation*. Consolidation means assigning more than one VM to a physical server. As a result, some of the servers can be turned off and power consumption of the computing system decreases. This is because servers consume more than 60% of their peak power in the idle state and turning off a server improves the power efficiency in the system. Again the technique involves performance-power tradeoff. More precisely, if workloads are consolidated on servers, performance of the consolidated VMs may decrease because of the reduction in the available physical resources (CPU, memory, I/O bandwidth) but the power efficiency will improve because fewer servers will be used to service the VMs.

A key benefit of virtualization technology is the ability to contain and consolidate the number of servers in a datacenter. This allows businesses to run multiple applications and OS workloads on the same server. Indeed ten server workloads running on a single physical server is typical, but some companies are consolidating as many as 30 or 40 workloads onto one server [50]. As a result, server utilization increases and the datacenter energy and cooling costs are lowered.

Consolidation is not without performance penalty. Therefore, the datacenter management software must be careful in how much consolidation is performed and at what performance penalty. A recent study by HP shows that the number of active servers serving an Office workload may be reduced by an average of 25% at a performance penalty of 16% [51]. Virtualization makes sense in cases where we have a number of underutilized virtual servers and can gain higher efficiency by combining them and raising server utilization. In a situation like Google or Facebook where we have lots of servers, which are already close to 100% utilization, doing the same thing, virtualization makes less sense and in fact, can add overhead due to the CPU resource used up by a hypervisor. In addition, consolidation is not recommended when data is distributed across the local disk spaces of some servers (for parallel access by running threads), and therefore, no servers may be shut down so as not to lose access to the shared data.

E. Datacenter Design Goals

Design of a modern datacenter is driven by three goals: *service level agreements* (SLAs), *total cost of ownership* (TCO), and sustainability as detailed below.

1) Service Level Agreements

An SLA sets the expectations between the consumer and provider. It is the foundation of how the service provider sets and maintains commitments to the service consumer. A typical SLA includes a set of constraints related to 24-7 availability, accounting, performance, and security at each layer of a datacenter, e.g., services, application infrastructure, compute and storage resources, network infrastructure, and physical facilities.

A good SLA addresses five key aspects: (i) What the provider is promising; (ii) How the provider will deliver on those promises; (iii) Who will measure delivery and how; (iv) What happens if the provider fails to deliver as promised; and (v) How the SLA will change over time. An example template for specifying an SLA is given in [52].

SLAs are the key to making profit in hosting datacenters. In fact there is a direct relationship between a datacenter's total profit and the level of SLA satisfaction by its clients. There are different types of SLAs and different ways to specify SLAs in a hosting datacenter. Different SLA contracts impose different conditions on datacenters, especially in terms of performance constraint to meet and amount of compute, storage, and network bandwidth resources to reserve for each client. Examples of performance constraints are response time constraints for time-sensitive services and throughput constraints for data-driven applications. Additionally, constraints may be deterministic (hard) or probabilistic (soft). A key issue is how much of a penalty a hosting datacenter pays each client if she fails to meet the agreed-upon minimum performance targets. We provide two simple examples next.

Throughput Constraints

Throughput-constrained SLA is one form of SLA, where a client pays a fixed price for meeting its task-level throughput requirement. Since the price paid is fixed, the hosting center's profit is purely a function of its energy consumption. Hence, the objective of profit maximization translates into energy minimization. The throughput-constrained SLA may be formulated as follows. Throughput requirement of client j with the request generation rate λ_j is stipulated to be exactly that. In other words, if the client's requests are assigned to a group of servers, I, and if the throughput provided by server i for client j is μ_{ij}, then we must have: $\lambda_j \leq \sum_{i \in I} \mu_{ij}$.

Average Response Time Constraint

Average response time (latency) constraint SLA stipulates that the average response time per request, $\tau_{j,avg}$, for requests of client j under a given arrival rate λ_j shall never exceed $\tau_{j,max}$. The client pays a fixed price for meeting the average response time constraint. Here the objective of profit maximization translates into energy minimization while still honoring the response time requirement. The response time

is a function of system utilization and, based on queuing theory, as utilization reduces response time decreases. However, operating servers at lower utilization has two undesirable effects. The first negative effect is higher number of active servers. The second adverse effect is the increased electrical energy cost due to energy non-proportional servers.

The datacenter resource arbiter may end up overprovisioning the datacenter resources in order to meet the worst-case task arrival rate and service times, and thereby, avoid paying penalty to clients. This approach increases the operational cost of the datacenter (since more resources will have to be kept active, resulting in an increase in the datacenter's electrical energy bill) and decreases the profitability of the datacenter (because fewer clients/requests may be serviced under fixed datacenter resources).

2) Total Cost of Ownership

Predicting and measuring TCO for the physical infrastructure of datacenters is required for *return-on-investment* (ROI) analysis. TCO is the most critical financial driver for datacenter operation and expansion. Most attempts to quantify TCO end up expressing TCO per datacenter, per square foot of datacenter, or per kW of power consumed in datacenter. TCO can be influenced by energy prices, use of renewable energy, IT trends, impact of efficiency gains in all layers of the datacenter, and even cost of compliance with government regulations. The reader may refer to [53] for a method to calculate the IT, networking, and facilities CapEx and OpEx in datacenters.

According to [54], economics of operating a datacenter are comprised of many factors that contribute to TCO, e.g.:

- Resiliency: The cost is derived from the level of redundant infrastructure built into a datacenter.
- Downtime: The cost of downtime is drastically different among different types of businesses and facility design considerations should reflect this.
- Financial considerations: These factors include financial aspects of site selection, cost segregation, capital recovery factor, staffing costs, and internal rate of return.
- Vertical Scalability: This means cloud computing-like elasticity capabilities incorporated into datacenter infrastructure and available floor space, i.e., increasing power and cooling densities without disrupting the datacenter operation.

Datacenter owners and operators also face regulatory pressure to reduce TCO since service costs are mostly higher than the market can bear, especially in emerging economies.

3) Sustainability

The sustainability goal is to lower the environmental footprint of a datacenter to such an extent that the services it provides are more environmentally friendly than conventional services offered within physical infrastructures. Thus, a sustainable datacenter would have a net positive effect on the environment. The authors of [55] identify five principles for achieving this vision: datacenter scale lifecycle design, flexible and configurable building blocks, pervasive sensing, knowledge discovery, and visualization, and autonomous control. Generally speaking, one must take a comprehensive view of sustainability that goes well past a localized focus on datacenter energy efficiency. The research must include a rigorous *life cycle assessment* (LCA) that accounts for the total carbon/water/air pollutant footprint from manufacturing to service to end-of-life recycling/disposal of all equipment and infrastructure components that go into a datacenter. This is a difficult task with significant implications in terms of the research and development effort spent on improving the datacenter energy efficiency. Fortunately, a number of studies including [56], show that although environmental impacts from manufacturing and end-of-life disposal are important, the lion's share of the carbon emission in today's datacenters is due to their operational (service related) energy use.

III. Datacenter Management Architecture

Datacenter management system determines the admission policy of the tasks at different times; affects the energy consumption of datacenter; sets the revenue of the datacenter in case of hosting datacenters; determines the performance for the served tasks; affects the reliability of the datacenter; and determines the life time of the devices used in the datacenter.

The datacenter manager uses runtime information about the incoming task arrival rate and type, expected workload level, power-performance state of various servers, current thermal map of the datacenter facility, the SLAs for different clients, and so on to make a series of decisions at discrete set of time instances (known as decision epochs). These decisions include:

- Admission decisions for incoming clients and hosted applications,
- Allocating server resources to clients in order to meet the SLAs (or minimize penalty for violating them),
- Task-to-server or cluster assignments and task migration decisions for every task,
- Issuing commands to turn on/off servers or chassis or perform *dynamic voltage and frequency scaling* (DVFS) for each server,
- Power provisioning at the datacenter, cluster, rack, chassis, or server levels,
- Varying cooling and air conditioning parameters.

To perform the above set of decisions, the datacenter manager needs to model and/or predict the task rate for each client and the workload per generated task. This workload prediction ability is a key enabler for a datacenter resource manager. There are many techniques to do datacenter workload modeling and prediction. See reference [57] for a representative work on characterizing datacenter workload demand patterns. See reference [58] for a qualitative comparison of work on datacenter workload modeling based

on the representativeness, accuracy and completeness of these designs.

In addition to large input size (thousands of servers and clients, millions of tasks, large facility size with many thermal zones, etc.), high variability in the datacenter workload makes it impossible to decide about every optimization parameter in the system all at once. More precisely, the datacenter manager cannot issue decisions too frequently (due to overhead of high-frequency decision making). Furthermore, it does not have detailed information about the server states and the workload characteristics. These factors point to an approach in which the datacenter resource management is performed in a hierarchical and distributed manner as detailed below.

A. Model Resource Management Architecture

In this paper, we focus on an exemplary architecture for datacenter resource management, comprised of three parallel agents: i) a *resource arbiter*, ii) a *power manager*, and iii) a *thermal manager*. The architecture of the datacenter resource management system with emphasis on the resource arbiter is depicted in Figure 2. In this architecture, the resource arbiter (allocation/assignment agent) allocates resources to clients and maps a client's tasks to the allocated servers, whereas the power manager looks after power distribution and dissipation issues and performs power provisioning and power management. The thermal manager in turn looks after thermal issues in the datacenter and controls the CRAC units and may initiate actions such as task migration and server shut down in response to thermal emergencies.

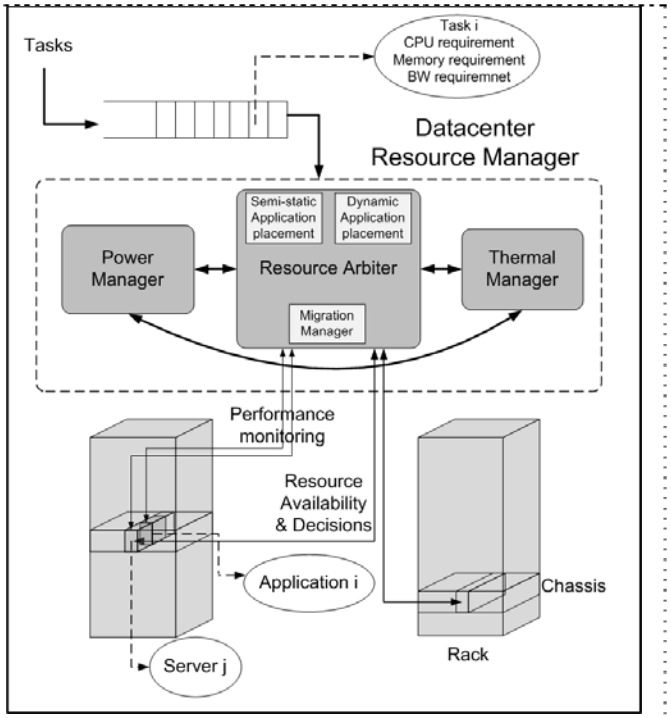

Figure 2. An example datacenter resource management architecture.

Based on the workload characteristics of each client and the client SLAs, the resource requirements for each client are determined (estimated) and allocated before the task-to-server assignment step. To guarantee a required SLA under the worst-case conditions, servers are typically allocated to clients so as to handle their peak workload condition.

Due to the impact of one agent's actions on the global state of the system, each agent must have some information about the other agents' concerns. For example, the resource arbiter, which dictates the resource assignment, may consider the IT and cooling-related power consumptions to decrease the operational cost of the datacenter.

B. Overview of Resource Management Architectures

In this section, a brief overview of the datacenter power (and performance) management architectures proposed in previous works is given.

The exemplary resource management architecture presented above is similar to the *hierarchical* and *coordinated power management architecture* proposed by Raghavendra et al. in [59]. Except that, in addition to the power manager, a VM controller is designed to decrease the average power consumption by grouping the VMs and assigning them to as few servers as possible (thereby turning off any unused servers). This controller uses server utilizations as input to assign VMs to servers.

Control theory has been applied to manage power and performance in datacenters. Chen et al. [60] proposed an autonomic management system for task placement and energy management in hosting datacenters. Two different techniques for power management are used: (i) turning off inactive servers, and (ii) dynamic voltage scaling. Energy consumption of the servers and wear-and-tear cost (cost of turning on/off a server) are considered. Two different approaches (based on queuing theory models and feedback control theory) are presented to minimize the power consumption of the datacenter while satisfying the SLAs.

Wang et al. [61] proposed a coordinated architecture that includes a cluster-level power control loop and a performance control loop for every VM. These control loops are configured to achieve desired power and performance objectives. Cluster-level power control monitors the server power consumptions and sets the DVFS state of the servers to reach a desired consumption level. The VM performance control loops dynamically control each VM's performance by changing the resource allocation policy. In addition, a cluster-level resource coordinator is designed to migrate the VMs in case of performance violation. Reference [62] presented a resource management architecture comprising a dispatcher and local and global managers. Local managers migrate the VMs in case of SLA violations, low utilization of server resources, high temperature, or high amount of communication with another VM that resides in a different server. Global managers receive information from local managers and issue commands for turning on/off servers, applying DVFS or resizing VMs.

Placing multiple copies of a VM on different servers and distributing the incoming requests among these VM copies can reduce the resource requirement for each VM copy and help the cloud provider utilize the servers more efficiently. In [63] the problem of energy-efficient VM placement in a cloud computing system is solved. Precisely, the authors present an approach that first creates multiple copies of VMs and then uses dynamic programming and local search to place these copies on the physical servers. To coordinate various power-performance tradeoff knobs in commodity software that runs on multiple platforms, and in emerging cloud hosted applications that operate on platforms outside developers' control, reference [64] proposes an approach for coordination where power-performance management can be performed within each software module without semantic knowledge regarding other modules.

Researchers have suggested the use of *VM migration* for energy saving. In theory, the VM migration technique promises high energy saving since it enables server consolidation, but it is difficult to apply the technique to servers in a datacenter because of the high overhead of the VM migration technique, e.g., the large system boot time, network traffic caused by the need to transfer the running task and its local context to a new server, and so on. For example, Liu et al. [65] proposed an architecture comprising of a migration manager and monitoring services. The physical machine's cost, VM status, and VM migration cost are used as inputs. The management policy searches among different VM placements to minimize the total cost and execute the live migration moves that the system needs.

There are a number of recent efforts to custom design and build application software, servers, and datacenters from the ground up with the goal of building an energy efficient computing infrastructure at the lowest possible cost. Notable among these efforts is Facebook's open compute [66].

IV. Resource Arbiter

Resource assignment decisions can be made at different timing granularities (seconds or minutes). Solutions with coarser granularity can take a global view of the datacenter state and assign resources by using more sophisticated algorithms. Conversely, smaller time granularity solutions can be used to modify an existing assignment to avoid hot spots or power budget violations. We call the solution with a long decision epoch a *semi-static solution* and the solution with a short epoch a *dynamic solution*.

The resource arbiter is responsible for assigning tasks to servers, migrating tasks if and when needed, and determining the on/off states of various servers. In each decision epoch, tasks in the datacenter can be divided into two groups: (i) new tasks that are not served yet, and (ii) continuing tasks whose service started in the past and need to be serviced for some time into the future. In the case of a dynamic solution, a relatively small number of tasks are considered and the resource arbiter must find a good assignment for these tasks in order to avoid an SLA violation. In the case of a semi-static solution, the resource arbiter, whose objectives are to modify the existing resource assignment solution for continuing tasks and find a good assignment solution for new tasks, considers all tasks simultaneously.

A. Resource Assignment

Tasks require CPU cycles, memory, secondary storage, and networking bandwidth to run on physical machines. These resources are limited in servers and the datacenter, and must often be shared among tasks. Resource requirements are typically dependent on the workload intensity and performance targets. This means that for a given performance target, a fixed resource allocation for tasks that have variable workload is either inadequate or inefficient. It is inadequate if the allocated resources for a task are less than the peak resource requirements. It is inefficient if the task is allocated the maximum resources that it needs, but typically has much lower resource needs than its peak demand. The same situations may arise even if a task's workload is unchanged, but its performance targets are changed.

To allocate the right amount of resources to each task, one can start with an initial resource allocation to the task and change the allocation based on runtime information about the task's workload intensity and achieved level of performance. In this way, it is possible to define a feedback control mechanism that would adjust resource allocations for the tasks to meet the performance targets. Another approach is to use workload prediction and do model-based resource allocation. Here the task behavior is predicted beforehand. These predictions can result in resource requirement evaluation using a model for the target performance metric.

Resource requirement estimation is different for different datacenters. For example, in private (corporate) datacenters, tasks usually have known resource requirements. In contrast, hosting datacenters admit clients with different types of tasks. In the case of Platform as a Service (PaaS), a hosting datacenter provides a pre-determined amount of processing power, memory, hard disc, and/or networking bandwidth to each client (to be used to service all tasks emanating from that client), whereas in the case of Software as a Service (SaaS), the hosting datacenter estimates the resource requirements of each task and assigns resources to satisfy the performance constraints for the tasks per SLA contract. The assumption is that if a client is given the resources that it needs, there will be no SLA violations for that client.

The resource management problem can be formulated using different figures of merit, including maximizing the number of tasks that are serviced in the datacenter (task level throughput), minimizing the total electrical energy cost of serving all the tasks in the datacenter, or maximizing the total profit of the datacenter operator. Without loss of generality, in the following, we focus on the profit maximization version of the resource assignment problem. Furthermore, we assume that the resource requirements of tasks are provided as the input to the problem.

The *profit maximization problem in a hosting datacenter* (PMHD) may be formulated as follows:

$$Max \sum_i \sum_j Price_i(x_{ij}) - \sum_i \sum_j MigrCost_i z_{ij} \\ -S_e T_d (1 + 1/COP) \sum_j (y_j P_j^0 + P_j^p \sum_i c_i^p x_{ij}/C_j^p) \quad (1)$$

subject to:

$$y_j \geq \sum_i x_{ij} \quad (2)$$

$$\sum_j x_{ij} \leq 1 \qquad x_{ij} \in \{0,1\} \quad (3)$$

$$z_{ij} \geq 0 \qquad z_{ij} \geq x_{ij} - x_{ij}^0 \quad (4)$$

$$z_{ij} \in \{0,1\} \quad (5)$$

$$\sum_i x_{ij} c_i^p \leq C_j^p \qquad \sum_i x_{ij} c_i^m \leq C_j^m \quad (6)$$

$$y_j P_j^0 + P_j^p \sum_i c_i^p x_{ij}/C_j^p \leq P_j^{max} \quad (7)$$

$$\theta_j \leq \theta^c \quad (8)$$

Here, x_{ij} denotes the assignment variable for task i and server j, whereas T_d and S_e denote the duration of decision epoch in seconds and the dollar cost per unit of electrical energy consumed, respectively. Moreover, c_i^p, c_i^m, C_j^p, and C_j^m denote the processing and memory requirements of task i, and the processing and memory capacities of server j, respectively. Temperature-related parameters in this formulation, θ_j, θ^c, and COP denote the expected temperature of server j, maximum allowable temperature for the server, and *coefficient of performance* of the CRAC units in datacenter (which is in turn a function of temperature distribution in the datacenter), respectively. The pseudo-Boolean status variable, y_j, denotes the ON (or active) or OFF (or sleep) state of server j. From (2), $y_j = 1$ whenever server j must service some task i.

The first term in the objective function represents the price paid by the clients. The price function is zero if $x_{ij} = 0$; it is equal to a fixed, non-zero value if $x_{ij} = 1$. The second term represents the migration cost, which is the summation of migration costs of all migrating tasks ($MigrCost_i$). A pseudo-Boolean variable, z_{ij}, determines whether or not task i is being moved to server j. From (4) it can be seen that if the previous task-to-server assignment variable (x_{ij}^0) is zero and the current task-to-server assignment variable (x_{ij}) is one, z_{ij} will be set to one, and the migration cost will be subtracted from the total profit. This migration cost can represent the electrical energy cost of migrating tasks or may be set to be equal to the value of profit loss due to client performance degradation during the task migration time.

The third term denotes the power consumption of CRAC and servers based on the set of tasks assigned to the servers. Here, we have assumed that the power consumption of an active server is composed of its idle power consumption (P_j^0) plus a power consumption term related to the server utilization with proportionality coefficient P_j^p. In addition, there is a 1/COP term, which accounts for the cooling cost in the datacenter. The rationale is that the power consumption of the CRAC unit is directly proportional to the total power consumption of the servers and inversely related to the COP of the CRAC units. The COP is a quadratic function of the outlet temperature of the CRAC units.

Constraint (2) determines the number of ON servers based on the allocated resources. Constraint (3) forces the clients to select only one server. Constraint (4) generates the helping variable for calculating the migration cost based on previous and current assignment variables. Constraint (6) shows the resource capacity limitation in each server while optional constraint (7) captures the peak power limitation (P_j^{max}) for the servers. Optional constraint (8) captures the maximum temperature constraint for each server. Note that the spatial temperature distribution in the datacenter is affected by the power dissipation pattern of all servers in the datacenter (which is in turn a consequence of the resource assignment solution). The outlet temperature of the CRAC units is set as a function of this temperature distribution, which subsequently affects the COP value of the CRAC units. Details about how to calculate the cooling cost in terms of datacenter power consumption is provided in Section VI.

The above formulation limits the tasks to be single-tier tasks that are serviced by a single server. In case of multi-tier tasks or single-tier tasks that need more than one server for their execution, the model may be extended by decomposing a given task into a set of related subtasks (resulting in a subtask flow graph to replace each task).

Based on the solution of the PMHD problem, servers are turned on or off and resource utilizations of servers are determined. These resource utilizations affect the decisions of the power and thermal managers about power provisioning and cooling issues. This is the reason that much of the prior work considers resource assignment along with power and thermal management. It is easy to show that a candidate solution to the PMHD problem can be verified in polynomial time and that the multi-dimensional bin-packing problem can be reduced to this problem, which proves that the PMHD problem is NP-complete. In the following we review various heuristic solutions to subsets or variants of the PMHD problem.

The problem of resource allocation is more challenging in the case of hosting centers because clients often have SLA contracts with the datacenter owner who would like to maximize its profit by reducing the SLA violations, decreasing the operational cost, and increasing customers without having to increase the physical assets (resource overbooking) [67]. In this subsection, we provide a general description of the SLA-based resource assignment problem in a hosting datacenter.

To avoid overprovisioning, prediction of client behaviors and resource needs may be used to determine the optimal resource allocation parameters. This means that prediction of clients' request arrival rate and estimation of expected service times for these requests can be used to determine the

"optimal" resource allocation for each client based on target performance metrics per SLA that has been negotiated with the client. To accomplish this, a model to estimate some measures of the quality of service or performance delivered to each client is needed. Queuing theory models are usually used for this estimation.

B. Overview of Related Work

Several versions of the PMHD problem have been investigated in the literature. Some of the prior work focuses on maximizing the number of served tasks in a datacenter (thus, total revenue for the datacenter operator) without considering the energy consumption and electrical energy cost. Example references are [68] and [69], where the authors present heuristic semi-static solutions based on network flow optimization to find a revenue maximizing solution for a scenario in which the total resource requirement of tasks is more than the total resource capacity in the datacenter. The resource assignment problem for tasks with fixed memory, disc, and processing requirements is tackled in [70], where the authors describe an approximation algorithm for solving the problem of maximizing the number of tasks that are serviced in the datacenter.

Another version of the PMHD problem only keeps the third term in the objective function, and thus ends up minimizing the total electrical energy cost. The constraints are to service all incoming tasks and satisfy the specified performance guarantees for each task. A classic example of this approach is the work of Chase et al. in [71], which uses an economics-based approach to manage the resource allocation in a system with shared resources in which clients can bid for resources as a function of the delivered performance. Yet another version of the PMHD problem considers the server and cooling power consumptions during the resource assignment problem. A representative work is reference [72], in which Pakbaznia et al. present a semi-static solution for concurrent task assignment and server consolidation. More precisely, considering the current datacenter temperature map and using an analytical model to predict the future temperature map as a function of the server power dissipations and incoming task rates, locations of the ON servers for the next decision epoch are determined and tasks are assigned to the ON servers so that total power consumption is minimized.

Considering the effect of consolidation on the performance and power consumption of servers is an important consideration in reducing the datacenter power consumption. For example, Srikantaiah et al. [73] describe energy-aware resource assignment based on an experimental study of the performance, energy use, and server utilization levels. Two dimensions for server resources are considered in this work: disk and CPU. The authors recommend using consolidation judiciously so as not to over-utilize servers in any resource dimension. The problem of task placement into a minimum number of ON servers is also discussed and a greedy algorithm for solving it is described.

A good example of considering server power consumption and migration cost in the resource assignment problem is reference [74], which presents power and migration cost aware task placement in a virtualized datacenter. Precisely, the authors present a power-aware task placement controller in a system with heterogeneous server clusters and virtual machines. For this problem, all VMs can be served in the datacenter and each VM has fixed and known resource requirements based on the specified SLA. So the price and cooling cost terms are removed from the objective function of the PMHD problem. An architecture called pMapper and a placement algorithm to solve the assignment problem are presented. There are three types of actions of the pMapper: (i) soft actions like VM re-sizing, (ii) hard actions such as DVFS, and (iii) server consolidation actions. There is a resource arbiter, which has a global view of the applications and their SLAs and issues soft action commands. A power manager issues hard actions whereas a migration manager triggers consolidation decisions in coordination with a virtualization manager. These managers communicate with the arbiter to set the VM sizes and find a good VM placement based on inputs from different managers. The authors use, as the migration cost, the SLA revenue loss due to performance degradation as a result of the VM migration. To optimally place VMs on the servers, the authors rely on some power efficiency metric to rank the servers. A heuristic based on a first-fit decreasing bin-packing algorithm is presented to place the tasks on servers starting with the most power-efficient server.

Many researchers in different fields have addressed the problem of SLA-driven resource assignment. Some of the previous works consider probabilistic SLA constraints with violation penalty, e.g., [75]-[77]. Other works consider utility function-based SLA [78][79]. In [80], the authors adopt a SLA with a soft constraint on the average response time and solve resource assignment problem for multi-tier applications. Other approaches such as reinforcement learning [81] and look-ahead control theory [82] have also been proposed to solve the resource assignment problem considering the SLA constraints. In addition to these semi-static SLA-based resource assignment solutions, some dynamic resource assignment solutions have been proposed in the literature [83][84].

Modeling the performance and energy cost is vital for solving the resource assignment problem under SLA constraints. Bennani et al. [85] present an analytical performance modeling based on queuing theory to calculate the response time of the clients based on CPU and I/O service times. Urgaonkar et al. [86] present an analytical model for multi-tier internet applications based on the mean-value analysis. An example of experimental modeling of power and performance in servers is presented in [87].

V. POWER MANAGEMENT

The process of power management in datacenters includes three steps: (i) estimating or measuring the time-varying

server power consumptions, (ii) scheduling the jobs or placing the VMs on the servers, and (iii) meeting upper bounds on the datacenter power consumptions at different granularity levels [16]. Distributed management architecture is thus composed of *rack-level power provisioners* (RPPs) and a single *datacenter-level power provisioner* (DPP).

A. Hierarchical Power Provisioning

The power provisioners in a datacenter consider the peak power limitation of the hardware (PDU) and the feed (the AC power grid or datacenter's internal power generators). The power limitations are related to the architecture and components of the PDU inside the datacenter.

The RPP divides its power budget (maximum allowable peak power consumption) among all chassis and servers in the target rack based on some policy. This policy can be based on the task assignment solution in each decision epoch, the power consumption histories of servers and chassis, or a simple fair-share allocation policy. The rack-level power budget itself is specified by the DPP based on the amount and type of tasks that have been dispatched to the servers in the target rack. This budget may also be set by the datacenter thermal manager based on the spatial temperature profile of racks in the datacenter to reduce the cooling power cost or avoid thermal emergencies. The RPP uses the minimum of the two limiting values specified by the DPP or the thermal manager as its power budget.

The DPP acts as the datacenter-wide power provisioner. The differences between DPP and RPPs are two-fold: (i) the decision epoch lengths for RPPs are shorter compared to that for the DPP—the rationale is that the RPP lies closer to the tasks running on servers, and hence must be able to quickly re-divide the total rack-level power budget among chassis and servers based on the runtime characteristics of the tasks running on individual servers within the rack; (ii) the RPPs cannot initiate a power shut-down of the rack, i.e., this decision is reserved for the DPP—the rationale is that the DPP is in a better position to predict the total workload coming into a datacenter and hence avoid greedy enclosure shutdowns without a good prediction of the future workload.

B. Server Power Management

The server power management (SPM), which is responsible for the power management of the server itself, receives inputs from the resource arbiter and the RPP. More precisely, it receives performance targets for the tasks that have newly been assigned to the server. It also has information about the tasks that are already running on the server. It receives the peak power consumption limit for the server from the RPP. This limit is computed dependent on feedbacks received from the datacenter thermal manager (based on dynamic temperature profile of the racks and servers), the DPP (based on the limited capacity of the power feed into the datacenter and/or the limitations of the power distribution network in the datacenter), or the SPM (based on the power thermal budget for the processor chips within the server).

The SPM tries to minimize the average power consumption of the server while satisfying the per-task performance requirements and the peak power consumption constraint. The SPM uses two techniques to perform its job: (i) changing the power/performance mode (P-state) of the processor, and (ii) changing the resource utilization and sharing policy among tasks that have been assigned to it. The SPM must also do some level of workload prediction in order to make good decisions and utilize the aforesaid optimization knobs effectively.

SPM techniques focus on putting the power consuming components to idle mode as frequently as possible to maximize the power saving. Studies on different datacenter workloads (cf. [14], [16] and [88]) show frequent short idle times in the workload. Because of the small width of these idle times, processors cannot be switched to deep sleep modes (with approximately zero power consumption) considering the performance penalty of frequent go-to-sleep and wakeup commands. On the other hand, drowsy (or shallow sleep) power modes for usual servers have relatively high power consumption with respect to the sleep mode power consumption. Processor consolidation is the solution. Here the idea is to assign a group of tasks with "complementary idle times" to the same processor so that context switches between consecutive tasks can result in maximum processor utilization (subject to an appropriate setting of the voltage and frequency level for the processor) without the need to transition the processor into or out of sleep states.

An alternative approach is to utilize energy-proportional server architectures. Indeed, a number of architectures have recently been proposed for a processor with very low (approximately zero) idle power consumption to reduce the average power consumption in the case of short idle times [16] and [89].

C. Overview of Related Work

1) Power Provisioning

An early work that addresses the power capping problem sets the power budget at the ensemble level (e.g., a blade server chassis with multiple server slots) to avoid the overprovisioning inefficiencies [90]. Individual bursty workloads are handled within this overall power budget by dynamically redistributing the power budget to the server servicing the workload, from other servers not currently requiring as much power. In cases when this is not possible, performance throttling is used to reduce power to avoid temperature increase beyond a critical threshold. A similar approach in [91], called RackPacker, tracks the power consumption behavior of the servers over time and suggests optimal ways to combine them in racks to maximize rack power utilization. Other approaches include demand-shaping to control the rate of workload execution and dynamic

migration of load to regions of the datacenter with higher power headroom [59][92][93].

From another perspective, the power capping problem may be seen as determining how many servers can be safely powered up under a given power budget. In the following paragraphs, reference [16] is reviewed as the typical work in this area. This paper presents the aggregate power usage characteristics of a large datacenter for different applications. The data can be used to maximize the use of deployed power capacity of datacenters and reduce the risk of power budget or performance constraint violation. The results show a big difference between theoretical and actual power consumptions of server clusters. For example, it is reported that considering a Google datacenter, the ratio of theoretical peak power consumption to actual maximum power consumption is 1.05, 1.28 and 1.39 for rack, PDU, and cluster levels, respectively. Based on the provided measurements and results, the authors outline a dynamic power provisioning policy in datacenters to increase the utilization of available power while protecting the power distribution hierarchy against overdraw. The authors mention that over-subscribing power in the racks is not safe but in PDU and cluster (between 7 to 16% more), over-subscribing power can be quite safe and efficient. Also it is desirable to mix the applications to increase the gap between theoretical and practical peak power to be able to increase the over-subscription of power.

Govindan et al. [94] present a new solution for handling power emergencies in datacenters that leverages existing UPS batteries to temporarily augment the utility supply during power emergencies. This method reduces the frequency and/or number of fuses/circuit-breakers giving way during episodes of power surge in a datacenter, which would disrupt the operation of some hosted applications.

2) Server Power Management

SPM is perhaps the most researched power management problem in the literature. Various Dynamic Power Management (DPM) techniques, which solve different variants of this problem have been presented by researchers. They can be broadly classified into three categories: ad hoc, stochastic, and learning-based methods. Ad hoc policies are based on the idea of predicting whether or not the next idle period length is greater than a specific value (the break-even time T_{be}). A decision to sleep will be made if the prediction indicates an idle period longer than T_{be}. Among these methods Srivastava et al. [95] use a regression function to predict the idle period length while Hwang et al. [96] propose an exponential-weighing average function for predicting the idle period length. Ad hoc methods are easy to implement, but perform well only when the requests are highly correlated; they typically do not take performance constraints into account.

By modeling the request arrival times (rates) and device service times (rates) as stationary stochastic processes, stochastic policies can take into account both power consumption and performance. Stochastic DPM techniques have a number of key advantages over ad hoc techniques. First, they capture a global view of the system, thus allowing the designer to search for a global optimum which can exploit multiple inactive states of multiple interacting resources. Second, they compute the exact solution (in polynomial time) for the performance-constrained power optimization problem. Third, they exploit the vigor and robustness of randomized policies. On the flip side, the performance and power obtained by a stochastic policy are expected values, and there is no guarantee that the results will be optimum for a specific instance of the corresponding stochastic process. Second, policy optimization requires *a priori* Markov models of the service provider and requester. Third, policy implementation tends to be more involved.

In [97], Benini et al. model a power-managed system as a *controllable discrete-time Markov decision process* (MDP) by assuming the non-deterministic service time of a request follows a geometric distribution. Qiu et al. in [98] model a similar system by using a controllable *continuous-time MDP* with Poisson distribution for the request arrival times and exponentially distributed request service times. This in turn enables the power manager (PM) to work in an event-driven manner, and thus reduce the decision making overhead. Other enhancements include time-indexed semi-MDP of Simunic et al. [99]. To cope with uncertainties in the underlying hardware state, DPM policies based on partially observable Markov decision process (POMDP) have been proposed in [100][101]. This stochastic approach is extended in [102] to include a request dispatcher (which performs job assignment in a multi-server system) using a generalized Petri net model. Reference [103] develops stochastic power control (which the authors call "autonomic power management") schemes that capture the power-performance tradeoffs in both single internet servers and datacenters.

Several recent works use machine learning techniques for adaptive policy optimization. Compared to simple ad hoc policies, machine learning-based approaches can simultaneously consider power and performance penalty, and perform well under various workload conditions. In [104], an online policy selection algorithm is proposed, which generates offline and stores a set of DPM policies (referred to as "experts") to choose from. The controller evaluates performance of the experts at the end of each idle period and, based on that, decides which expert should be activated next.

Tan et al. in [105] propose to use an enhanced Q-learning algorithm for system-level DPM. This is a model-free *reinforcement learning* (RL) approach since the PM does not require prior knowledge of the state transition probability function. However, the knowledge of the state and action spaces and also the reward function is required. The Q-learning based DPM learns a policy online by trying to learn which action is best for a certain system state, based on the reward or penalty received. Wang et al. in [106] present an approach for RL-based DPM in a partially observable

environment. The proposed approach can perform learning and power management in a continuous-time and event-driven manner. In addition, it uses enhanced TD(λ) learning algorithm for semi-MDP to accelerate convergence and alleviate the reliance on Markovian property. Finally, a Bayesian classifier-based workload prediction engine is incorporated to provide partial information about the service request (SR) state for the RL algorithm.

Feedback control theory is a powerful tool for dealing with variability in engineered systems. A *proportional-integral* (PI) controller is used in [107] to control the voltage dynamically, while a user-specified system latency in stream processing is used as the set-point for the controller. Alimonda et al. [108] propose a control-theoretic approach for *dynamic voltage scaling* (DVS) in multi-processor system on chip (MPSoC) pipelined architectures. The approach aims to control the inter-processor queue occupancy levels. Wu et al. [109] present an analytical approach to DVS for multiple clock domain processors. It is based on a dynamic stochastic queuing model and a PI controller with queue occupancy being the controlled variable. In [110], the authors consider independent scaling of the voltage/frequency of each core of a chip multiprocessor to enforce a chip-level power budget. Power mode assignments are re-evaluated periodically by a global power manager, based on the performance and average power consumption observed in the most recent period.

Modern processor chips are multi-core chips [111]-[113]. One can thus envision another core-level power manager, which takes care of internal voltage and frequency scaling of the cores within a processor chip and/or performs core consolidation. The solution approaches are similar to those proposed for the SPM and range from open-loop ad hoc optimizations to closed-loop feedback control and implemented at different levels of the hardware/software stack from operating systems, to firmware and hypervisor, to hardware, see for example, [114]-[118].

VI. Thermal Management

The goal of a datacenter's cooling system is to ensure that the server temperatures do not go higher than a redline temperature and the ambient temperature stays at a desired level (25°C). This is because it is believed that if servers work a long time in an environment with temperature higher than the safe operating temperature, their average failure probability will go higher and they will age faster.

A. General Formulation of the Problem

We present a model for heat transfer and its effect on the power consumption of the typical rack-based datacenter with raised-floor architecture using a hot-aisle/cold-aisle cooling system from reference [119].

To model the heat transfer in the datacenter, one must calculate the power consumption of servers inside the datacenter. Utilization levels of the processors, changes in the workload characteristics, and the operation mode of a server affect the power consumption of that server (P_j^s). The server power consumption comprises a constant term plus a linear term related to the server utilization. Total power consumption of a chassis (P_i^{ch}) is calculated by summing the power consumptions of servers inside chassis plus a base power level (P_i^{ch-b}) to account for the fan power and the switching losses in the DC-DC power converters:

$$P_i^{ch} = P_i^{ch-b} + \sum_j P_j^s \tag{9}$$

Each chassis draws cold air to remove the heat from its servers. The hot air then exits the chassis from the rear side. The temperature of the cold air drawn into the i^{th} chassis is called the *inlet temperature* of that chassis and is denoted by T_{in}^i. Similarly, the outlet temperature of the i^{th} chassis, T_{out}^i, is defined as temperature of the hot air that exits the chassis. Consider the i^{th} chassis with a power dissipation of P_i^{ch}, inlet and outlet temperatures of T_{in}^i and T_{out}^i, and an air flow rate of f_i. From the Fourier's Law of Heat Conduction and the Energy Conservation Law:

$$Q_{in} + P_i = Q_{out} \Rightarrow P_i = \rho f_i c_p (T_{out}^i - T_{in}^i) \tag{10}$$

where Q_{in} and Q_{out} denote the input and output heat (flow) rates, respectively. The heat rate is defined as the amount of heat or thermal energy generated or transferred in a unit of time (in this system with air flow). Parameters ρ and c_p denote the air density in kg/m^3, and specific heat of air in J/kg-K. f_i denotes the local air flow rate for the i^{th} chassis in units of m^3/s whereas T_*^i denotes air temperature in K.

The inlet temperature of a chassis depends on the supplied cold air from the CRAC unit and the hot air that is re-circulated from the outlet of other chassis in the datacenter. The outlet temperature of a chassis in turn depends on the inlet temperature and the power consumption of that chassis. A compact heat model for datacenters is presented in [120], where the authors show that the recirculation of heat in a datacenter can be described by a cross-interference matrix. The cross-interference matrix is represented by $\Phi = \{\phi_{ij}\}$ where ϕ_{ij} shows the contribution of the outlet heat rate of the i^{th} chassis in the inlet heat rate of the j^{th} one.

The efficiency of the cooling process depends on many factors such as the substance used in the chiller, the speed of the air exiting the CRAC unit, etc. *Coefficient of performance* (COP), which is a term used to measure the efficiency of a CRAC unit, is defined as the ratio of the amount of heat that is removed by the CRAC unit (Q) to the total amount of energy that is consumed in the CRAC unit to chill the air (E), i.e., [121]:

$$COP = Q/E \tag{11}$$

The COP of a CRAC unit is not constant and varies by the temperature of the cold air that it supplies to the room, T_s. In particular, the higher the supplied air temperature is, the better cooling efficiency will be. The following COP model has been reported for the CRAC unit in an industrial-scale (production) datacenter [121],

$$COP(T_s) = (0.0068\, T_s^2 + 0.0008\, T_s + 0.458) \tag{12}$$

The CRAC power consumption is related to the power consumption in datacenter and $1/COP$. More precisely, increasing the supply cold air temperature decreases the CRAC power consumption. So an important goal is to increase the supply cold air temperature as much as possible.

In any case, the total power dissipation of a datacenter comprised of N chassis is given by:

$$P_{DC} = \left(1 + \frac{1}{COP(T_s)}\right)\sum_{i=1}^{N} P_i^{ch} \quad (13)$$

Suppose that the i^{th} chassis contains M_i servers and K+1 (voltage and frequency) *v-f* levels are available to each server (including *v-f* = 0 corresponding to a fully power-gated server). Let w_{ij} denote the number of servers in the i^{th} chassis which are running at the j^{th} *v-f* setting. Evidently, $u_i \equiv \sum_{j=1}^{K} w_{ij} \leq M_i$ is the number of ON servers in the i^{th} chassis.

The goal is then to minimize (13) by (i) determining the optimum value of T_s, (ii) turning various servers and chassis ON/OFF, and (iii) for ON chassis, determining the number of the ON servers and their corresponding cores' *v-f* levels. The following constraints must be met:

$$\mathbf{T}_{\text{in}} \leq \mathbf{T}_{\text{critical}} \quad (14)$$

$$0 \leq u_i \leq \text{M}_\text{i} \ \forall i \quad (15)$$

$$u_i = \sum_{j=1}^{K} w_{ij} \ \forall i \quad (16)$$

$$\sum_{i=1}^{N} w_{ij} = S_j \ \forall j = 1 \ldots K \quad (17)$$

where $\mathbf{T}_{\text{in}}$ denotes the inlet temperature vector of the chassis, and $\mathbf{T}_{\text{critical}}$ is a vector of size N with all entries equal to the critical inlet temperature, $T_{critical}$ (The inlet temperature of all chassis must be less than this value in order to ensure that the corresponding servers will not overheat and eventually fail). A typical value for $T_{critical}$ is 25°C. S_j is the total number of required servers with the j^{th} *v-f* setting. Let the required number of servers to serve a given set of tasks be S_{tot}. Clearly, $S_{tot} = \sum_{j=1}^{\text{K}} S_j$.

Notice that the power distribution of different chassis in the datacenter directly affects the temperature distribution in the room. The temperature distribution in the room affects the required supply cold air temperature which affects the *COP* and CRAC power consumption. Task placement is a key contributor to determining the power consumption distribution inside datacenter. The other important factor is the power manager's decisions. This underlines the close interactions between the resource arbiter and power manager on one hand and the thermal manager on the other hand.

B. Overview of Related Work

Prior work has outlined the foundation for creation of a "smart" datacenter through the use of flexible cooling resources and a distributed sensing system that can provision the cooling resources based on the need.

Sharma et al. [122] propose a power provisioning scheme (0 to 100% of peak power) to reduce the datacenter cooling power consumption. In this approach, the power provisioned for each server is inversely related to the measured temperature of that server.

To decrease the maximum temperature in the datacenter, and increase the supplied cold air temperature for better energy efficiency in the cooling system, Moore et al. [121] present a temperature-aware workload placement. The proposed temperature-aware workload placement is in fact a power provisioning policy based on the temperature (status) measurements in the system. This means that a portion of the total power requirement of the workloads in the system is assigned to each server based on the server temperature in the previous measurement. Assigning power levels to servers based on the measured temperature can minimize the maximum temperature in the system, and then the cooling system can provide the cool air with higher temperature that means higher energy efficiency. A discrete version of power provisioning policy first given in [122] is introduced in this work to consider discrete power modes in the servers.

Heat recirculation, which means using hot air instead of cold air for cooling the servers, can occur because the cold air is not supplied to the system or the separation between cold aisle and hot aisle is not perfect in some points. Moore et al. also present a method to minimize the maximum temperature in the datacenter by minimizing the heat recirculation effect. The method includes a calibration phase to find the datacenter-related values of heat recirculation for different parts of the datacenter and then use it along with online measurements to do power provisioning. Minimizing heat recirculation using temperature-aware task scheduling is discussed in [123]. The task scheduling policy in this work focuses on making the inlet temperatures of all active servers as even as possible to decrease the cooling system's outlet power consumption. A recent work [124] proposes using thermoelectric coolers as a power management mechanism inside the servers to allow the datacenter cooling system to increase the supply cold air temperature to minimize the required outlet temperature of the CRAC unit, hence minimize the cooling system power consumption.

A 3D computational fluid dynamic (CFD)-based tool for thermal modeling of rack-mounted datacenters is presented in [125]. This tool can be used in the CRAC design process but because the tool has a long execution time, it is not possible to use it in online decision making.

Reference [126] presents a holistic management system that can sense and control a complex heat transfer stack utilizing a thermodynamics-based evaluation model. In particular, it shows a common thermodynamic platform which serves as an evaluation and basis for a policy-based control engine for such a "smart" datacenter with much broader reach - from chip core to the cooling tower.

VII. CONCLUSION

The goal of this paper was to provide an introduction to resource provisioning and power/thermal management problems in datacenters, and summarize key techniques and policies that maximize the datacenter energy efficiency

subject to peak/total power consumption and thermal constraints while meeting given SLAs. In the process, we identified sources of energy inefficiency in datacenters and presented some high level solutions to these problems. There are, however, plenty of opportunities to improve the state of the art. *Our community can contribute a lot* to advancing the design and adaptive control of datacenters with energy efficiency, SLAs, and TCO as the primary considerations as detailed below.

The first step is to develop a *theory for understanding the energy-complexity of computational tasks*. Today, energy-efficiency is benchmarked relative to last year's product; any efficiency gain is touted as success. Instead, we wish to ask what level of efficiency is *possible* and measure solutions relative to this limit. One must thus develop key scientific principles to measure the energy complexity of applications. By combining energy complexity with time complexity of applications, we can then perform fundamental energy-performance tradeoffs at application programming level.

Informed by this new theory, one can then reconsider the design of the hardware platforms that comprise the energy-efficient datacenters. Key sources of inefficiency are lack of energy proportional hardware and the over-provisioning of these servers to meet SLAs given the time-varying application resource demands. An energy-efficient datacenter exploits *hardware heterogeneity* and employs *dynamic adaptation*. Heterogeneity allows energy-optimized components to be brought to bear as application characteristics change. Dynamic adaptation allows the datacenter to adapt and provision hardware components to meet varying workload and performance requirements, which in turn eliminates over-provisioning. Computing, storage and networking subsystems of current datacenters exhibit dismal energy-proportionality. One must attempt to redesign server architectures and network protocols with energy-efficiency and energy-proportionality as the driving design constraint. On the storage front, we must construct hybrid storage systems that assign data to devices based on a fundamental understanding of access patterns and capacity-performance-efficiency trade-offs.

To go beyond the incremental energy efficiency gains possible from component-wise optimization, one must consider the *coordination and control* of storage, networking, memory, compute, and physical infrastructure. By tackling the optimization problem for the datacenter as a whole, one can develop solutions at one layer that will be exploited at other layers. By using the mathematical underpinnings of control theory and stochastic modeling, these approaches enable reasoning about worst-case and average-case behavior of multi-loop compositions of control approaches. One can then develop algorithms to globally manage compute, storage, and cyber-physical resources with the objective of minimizing the total energy dissipation while meeting SLAs.

Finally, to evaluate datacenter designs, one must develop *new methodologies and simulation infrastructure* to quantify the impact and prototype research ideas. Because of the complexity and scale of datacenter applications, conventional evaluation approaches cannot evaluate new innovations with reasonable turn-around time. Hence, we must design *hierarchical models*, which integrate performance and energy estimates across detail and time granularities, and *parallel* cluster-on-a-cluster simulation techniques, that together allow us to quantitatively evaluate systems at an entirely new scale.

Acknowledgement—I would like to thank my Ph.D. student, Hadi Goudarzi, who contributed heavily to the extensive prior work review provided in this paper, and anonymous reviewers for their valuable suggestions.

REFERENCES

[1] Reducing Data Center Power Consumption – KM World. 2010. Available at: http://www.kmworld.com/Articles/Editorial/Feature/Reducing-data-center-power-consumption-68133.aspx .

[2] Report to Congress on Server and Datacenter Energy Efficiency Public Law - U.S. Environmental Protection Agency. 2006. Available at: http://hightech.lbl.gov/documents/data_centers/epa-datacenters.pdf.

[3] Greenpeace 'likes' Facebook's new datacenter, but wants a greener friendship – Greenpeace. 2011. Available at: http://www.greenpeace.org/international/en/press/releases/Greenpeace-likes-Facebooks-new-datacentre-but-wants-a-greener-friendship/ .

[4] Growth in data center electricity use: 2005 to 2010 – J. Koomey. Available at: http://www.analyticspress.com/datacenters.html .

[5] Data Center Power Consumption Grows Less Than Expected: Report – J. Kovar. Available at: http://www.crn.com/news/data-center/231400014/data-center-power-consumption-grows-less-than-expected-report.htm?pgno=2 .

[6] Google Details, and Defends, Its Use of Electricity– J. Glanz, New York Times. 2011. Available at: http://www.nytimes.com/2011/09/09/technology/google-details-and-defends-its-use-of-electricity.html .

[7] Facebook to build massive server farm near the Arctic Circle. Oct. 2011. Available at: http://news.yahoo.com/facebook-build-massive-server-farm-near-arctic-circle-141408285.html .

[8] Google's Energy Story: High Efficiency, Huge Scale – R. Miller. Sept. 2011. Available at: http://www.datacenterknowledge.com/archives/2011/09/08/googles-energy-story-high-efficiency-huge-scale/ .

[9] Data centers that save energy, Available at: http://www.google.com/green/bigpicture/#beyondzero-datacenters.

[10] Microsoft goes green: data centers, offices to be carbon neutral come July, Available at: http://arstechnica.com/information-technology/2012/05/microsoft-goes-green-data-centers-offices-to-be-carbon-neutral-come-july/.

[11] Facebook Shows Why Green Data Centers Matter, Available at: http://gigaom.com/cleantech/facebooks-move-indicates-green-data-centers-now-a-must-have/.

[12] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia. "A view of cloud computing," *Communications of the ACM* 53(4), pp. 50-58., 2010.

[13] R. Buyya. "Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility," *Proc. of Int'l Symp. on Cluster Computing and the Grid*, 2009.

[14] L.A. Barroso and U. Hölzle. "The Case for Energy-Proportional Computing," *IEEE Computer*, vol. 40, 2007.

[15] L.A. Barroso and U. Holzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool Publishers, 2009.

[16] X. Fan, W. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *Proc. of Int'l Symp. on Computer Architecture*, June 2007.

[17] C. Lefurgy, X. Wang, and M. Ware. "Server-Level Power Control," *Proc. Of Int'l Conf. on Autonomic Computing*, 2007.

[18] Power Efficiency Comparison of Enterprise-Class Blade Servers and Enclosures, Dell Inc., 2010. Available at: www.dell.com/.../BladePowerStudyWhitePaper_08112010_final.pdf .

[19] Data Center Energy Efficiency Training, UD Sept. of Energy, Available at: http://hightech.lbl.gov/training/modules/10-electrical-systems.pdf.

[20] N. Rasmussen. "Calculating Total Cooling Requirements for Datacenters," *American Power Conversion*, white paper number 25. 2007.

[21] Cloud Services Axe Data Center Energy Consumption, Power Costs – A.R. Hickey, CRN. Available at: http://www.crn.com/news/cloud/231601736/cloud-services-axe-data-center-energy-consumption-power-costs.htm .

[22] James Hamilton's Blog. Available at: http://perspectives.mvdirona.com/ .

[23] The Business Landscape of Cloud Computing – Daryl Plummer, Financial Times. 2012. Available at: http://www.ft.com/cms/5e231aca-a42b-11e1-a701-00144feabdc0.pdf

[24] EPA Conf. on "Enterprise Servers and Datacenters: Opportunities for Energy Efficiency," *Lawrence Berkeley National Laboratory*. 2006. Available at: http://hightech.lbl.gov/DCTraining/presentations.html.

[25] C. Belady, et al., "Green Grid Datacenter Power Efficiency Metrics: PUE and DCIE", Available at http://www.thegreengrid.org/gg_content/TGG_Data_Center_Power_Efficiency_Metrics_PUE_and_DCiE.pdf .

[26] S. Ghemawat, H. Gobioff, and S-T. Leung. The Google file system. *Proc. of ACM Symp. on Operating Systems Principles*, 2003.

[27] Whose data centers are more efficient? Facebook's or Google's? Available at: http://gigaom.com/cloud/whose-data-centers-are-more-efficient-facebooks-or-googles/.

[28] Microsoft Builds New Data Center in Dublin, Available at: http://facilitygateway.com/news/?p=1937.

[29] EPA to use PUE in Data Center Energy Star, April 2009. Available at: http://www.datacenterknowledge.com/archives/2009/04/22/epa-to-use-pue-in-data-center-energy-star/ .

[30] Harmonizing Global Metrics for Data Center Energy Efficiency 2011. Available at: http://www.greenit-pc.jp/e/topics/release/110228_e.html .

[31] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu, "Delivering energy proportionality with non energy-proportional systems – optimizing the ensemble," *Proc. of HotPower*, 2008.

[32] D, Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: Eliminating server idle power," *Proc. of the 14th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, 2009.

[33] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. "FAWN: A Fast Array of Wimpy Nodes," *Proc. of SOSP*, 2009.

[34] V. J. Reddi, B. Lee, T. Chilimbi, and K. Vaid. "Web Search Using Small Cores: Quantifying the Price of Efficiency," Technical Report MSR-TR-2009-105, *Microsoft Research,* 2009.

[35] R. Merritt, "Facebook likes wimpy cores, CPU subscriptions," Available at: http://www.eetimes.com/electronics-news/4375880/Facebook-likes-wimpy-cores--CPU-subscriptions, 2012.

[36] U. Holzle, "Brawny cores still beat wimpy cores, most of the time," *IEEE Micro* 2010.

[37] W. Lang, J.M. Patel, and S. Shankar, "Wimpy Node Clusters: What About Non-Wimpy Workloads?" *Proc. of the Sixth Int'l Workshop on Data Management on New Hardware*, June 7, 2010.

[38] M. Aater Suleman, O. Mutlu, M.K. Qureshi, and Y.N. Patt, "Accelerating Critical Section Execution with Asymmetric Multicore Architectures," *IEEE Micro*, Vol.30, No.1, pp.60-70, Jan.-Feb. 2010.

[39] R.Kumar, D. M. Tullsen, N. P. Jouppi, P. Ranganathan, "Heterogeneous Chip Multiprocessors," *IEEE Computer*, 38(11):32–38, 2005.

[40] V. Gupta, P. Brett, D. Koufaty, D. Reddy, S. Hahn†. K. Schwan, and G. Srinivasa, "The forgotten 'Uncore': On the energy-efficiency of heterogeneous cores." *USENIX Annual Technical Conf.*, 2012.

[41] Qualitative Analysis of Power Distribution Configurations for Data Centers, Green Grid Consortium, 2007, Available at: http://www.thegreengrid.org/~/media/WhitePapers/TGG_Qualitative_Analysis.pdf?lang=en .

[42] Strategies for Solving the Datacenter Space, Power, and Cooling Crunch, Oracle White Paper, 2010, Available at:http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/sun-datacenter-space-power-wp-075961.pdf .

[43] Cisco Containerized Data Center – Cisco. Available at: http://www.cisco.com/en/US/netsol/ns1121/index.html .

[44] Datacenter Power Delivery Architectures: Efficiency and Annual Operating Costs, 2007. Available at: http://cdn.vicorpower.com/documents/whitepapers/server_efficiency_vichip.pdf.

[45] J. Dean and S. Ghemawat. "MapReduce: Simplified data processing on large clusters." *Commun, of ACM*, 51(1):107–113, 2008.

[46] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. "Dryad: Distributed data-parallel programs from sequential building blocks," *Proc. of Eurosys Conf.,* 2007.

[47] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. "Pregel: a system for large-scale graph processing," *Proc. of the ACM Int'l Conf. on Management of Data,* 2010.

[48] M. Zaharia, B. Hindman, A. Konwinski, A. Ghodsi, A. D. Joesph, R. Katz, S. Shenker, and I. Stoica. "The datacenter needs an operating system," *Proc. of USENIX Conf. on Hot Topics in Cloud Computing*, 2011.

[49] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield. "Xen and the art of virtualization," *Proc. of ACM Symp. on Operating Systems Principles*, 2003.

[50] Energy Efficiency White Paper – Vmware. Available at: www.vmware.com/files/pdf/green_wp.pdf.

[51] Reducing the power consumption of HP ProLiant servers in the data center - HP. Available at:

http://h20195.www2.hp.com/V2/GetDocument.aspx?docname=4AA0-4879ENW&cc=us&lc=en.

[52] Service Level Agreement in the Solaris OE Data Center – Sun Microsystems, Available at: http://www.informit.com/articles/article.aspx?p=26936.

[53] J. Koomey, K. Brill, P. Turner, J. Stanley, and B. Taylor, "A Simple Model for Determining True Total Cost of Ownership for Data Centers," Uptime Institute, 2008, Available at: http://www.privatecloudrentals.com/EMSSamples/TrueCost.pdf and http://www.uptimeinstitute.org/TrueTCO .

[54] Data Center Strategies: Translating Teamwork into TCO, Available at: http://www.datacenterknowledge.com/archives/2011/07/28/data-center-strategies-creating-value-through-internal-perspectives/.

[55] B. J. Watson, A. J. Shah, and M. Marwah, "Integrated Design and Management of a Sustainable Data Center," *ASME InterPACK Conf.*, 2009.

[56] J. Chang, J. Meza, P. Ranganathan, A. Shah, R. Shih, and C. Bash, "Totally green: evaluating and designing servers for lifecycle environmental impact," *Proc. of Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, 2012. pp. 25-36.

[57] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. "Workload Analysis and Demand Prediction of Enterprise Data Center Applications," *Proc. of Int'l Symp. on Workload Characterization*, 2007.

[58] C. Delimitrou and C. Kozyrakis. "Cross-Examination of Datacenter Workload Modeling Techniques," *Proc. of Int'l Conf. on Distributed Computing Systems Workshops*, 2011.

[59] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu. "No power struggles: Coordinated multi-level power management for the datacenter," *Proc. of Architectural Support for Programming Languages and Operating Systems*, 2008.

[60] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang and N. Gautam. "Managing server energy and operational costs in hosting centers," *Proc. of SIGMETRICS,* 2005,

[61] X. Wang and Y. Wang. Co-con: Coordinated control of power and application performance for virtualized server clusters. *Proc. of Int'l workshop on Quality of Service*, 2009.

[62] A. Beloglazov and R. Buyya. Energy efficient resource management in virtualized cloud datacenters. *Proc. of Int'l Conf. on Cluster, Cloud and Grid Computing*, 2010.

[63] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," *Proc. of IEEE Cloud*, pp. 750-757, Jun. 2012.

[64] A. Kansal, J. Liu, A. Singh, R. Nathuji, and T. Abdelzaher. "Semantic-less coordination of power management and application performance," *SIGOPS Oper. Syst. Rev.* 44, 1 (March 2010), pp. 66-70.

[65] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang and Y. Chen. "Greencloud: A new architecture for green datacenter," *Proc. of Int'l Conf. Industry Session on Autonomic Computing and Communications, ICAC-INDST'09*, June 2009.

[66] Hacking Conventional Computing Infrastructure, Available at: http://opencompute.org/

[67] B. Urgaonkar, P. Shenoy, and T Roscoe. "Resource Overbooking and Application Profiling in Shared Hosting Platforms," *Proc. of Symp. on Operating Systems Design and Implementation*, 2002.

[68] A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko and A. Tantawi. "Dynamic placement for clustered web applications," *Proc. of Int'l Conf. on World Wide Web*, May 2006.

[69] C. Tang, M. Steinder, M. Spreitzer and G. Pacifici. "A scalable application placement controller for enterprise datacenters," *Proc. of 16th Int'l World Wide Web Conf.*, May 2007.

[70] F. Chang, J. Ren and R. Viswanathan. "Optimal resource allocation in clouds," *Proc. of Int'l Conf. on Cloud Computing*, July 2010.

[71] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat and R. P. Doyle. "Managing energy and server resources in hosting centers," *Proc. of ACM Symp. on Operating Systems Principles*, October 2001.

[72] E. Pakbaznia, M. GhasemAzar, and M. Pedram. "Minimizing datacenter cooling and server power costs," *Proc. of Design Automation and Test in Europe.* 2010.

[73] S. Srikantaiah, A. Kansal, and F. Zhao. "Energy aware consolidation for cloud computing," *Proc. of Conf. on Power aware Computing and Systems*, 2008.

[74] A. Verrna, P. Ahuja and A. Neogi. "pMapper: Power and migration cost aware application placement in virtualized systems," *Proc. of Int'l Middleware Conf.*. 2008.

[75] Z. Liu, M. S. Squillante and J. L. Wolf. "On maximizing service-level-agreement profits," *Proc. of ACM Conf. on Electronic Commerce*. 2001.

[76] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir and M. Martonosi. "Capping the brown energy consumption of internet services at low cost," *Proc. of Int'l Conf. on Green Computing*, 2010.

[77] H. Goudarzi, M. Ghasemazar, and M. Pedram. "SLA-based optimization of power and migration cost in cloud computing," *Proc. of Int'l Symposium on Cluster, Cloud and Grid Computing*, pp. 172-179, May 2012.

[78] L. Zhang and D. Ardagna. "SLA based profit optimization in autonomic computing systems," *Proc. of Int'l Conf. on Service Oriented Computing*, Nov. 2004.

[79] D. Ardagna, B. Panicucci, M. Trubian, L. Zhang, "Energy-Aware Autonomic Resource Allocation in Multi-Tier Virtualized Environments," *IEEE Trans. on Services Computing*, vol. 99, 2010.

[80] H. Goudarzi and M. Pedram. "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," *Proc. of the IEEE Cloud*, Jun. 2011.

[81] G. Tesauro, N. K. Jong, R. Das and M. N. Bennani. "A hybrid reinforcement learning approach to autonomic resource allocation," *Proc. of Int'l Conf. on Autonomic Computing*, 2006.

[82] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy and G. Jiang. "Power and performance management of virtualized computing environments via lookahead control," *Proc. of Int'l Conf. on Autonomic Computing*, 2008.

[83] A. Chandra, W. Gongt and P. Shenoy. "Dynamic resource allocation for shared datacenters using online measurements," *Proc. of Int'l Conf. on Measurement and Modeling of Computer Systems*, 2003.

[84] N. Bobroff, A. Kochut, and K Beaty. "Dynamic Placement of Virtual Machines for Managing SLA Violations," *Proc. of Int'l Symp. on Integrated Management*, 2007.

[85] M. N. Bennani and D. A. Menasce. "Resource allocation for autonomic datacenters using analytic performance models," *Proc. of Int'l Conf. on Autonomic Computing*. 2005.

[86] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer and A. Tantawi. "An analytical model for multi-tier internet services and its applications," *Proc. of Int'l Conf. on Measurement and Modeling of Computer Systems*, June 2005.

[87] M. Pedram and I. Hwang. "Power and performance modeling in a virtualized server system," *Proc. of Int'l Conf. on Parallel Processing workshops*, 2010.

[88] D. Meisner, C. Sadler, L. Barroso, W. Weber, and T. Wenisch. "Power Management of Online Data-Intensive Services," *Proc. of Int'l Symp. on Computer Architecture*, June 2011.

[89] D. Meisner, B. Gold, and T. Wenisch, "PowerNap: eliminating server idle power," *Proc. of Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, March 2009.

[90] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level Power Management for Dense Blade Servers," *Proc. of Int'l Symp. on Computer Architecture*, pp. 66-77, 2006.

[91] L. Ganesh, J. Liu, S. Nath, and F. Zhao, "Unleash Stranded Power in Data Centers with RackPacker," *Proc. of Workshop on Energy-Efficient Design*, in conjunction with ISCA, 2009.

[92] L. Ramos and R. Bianchini. "C-Oracle: Predictive thermal management for data centers." *Proc. of Int'l Symp. On High-Performance Computer Architecture*, 2008.

[93] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood. "Power Routing: Dynamic Power Provisioning in the Datacenter," *Proc. of Architectural Support for Programming Languages and Operating Systems*, 2010.

[94] S. Govindan, D. Wang, A. Sivasubramaniam, and B. Urgaonkar. "Leveraging Stored Energy for Handling Power Emergencies in Aggressively Provisioned Datacenters," *Proc. of Architectural Support for Programming Languages and Operating Systems*, 2012.

[95] M. Srivastava, A. Chandrakasan and R. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," *IEEE Trans. on VLSI Systems*, 1996.

[96] C. H. Hwang and A. C. Wu, "A predictive system shutdown method for energy saving of event-driven computation," *Proc. of Int'l Conf. on CAD,* 1997.

[97] G. A. Paleologo, L. Benini, et.al, "Policy Optimization for Dynamic Power Management", *Proc. of Design Automation Conf.*, pp.182-187, Jun. 1998.

[98] Q. Qiu and M. Pedram, "Dynamic Power Management Based on Continuous-Time Markov Decision Processes," *Proc. of Design Automation Conf., 1999*.

[99] T. Simunic, L. Benini, P. Glynn and G. De Micheli, "Event-driven power management," *IEEE Trans. on CAD*, 2001.

[100] H. Jung and M. Pedram, "Dynamic power management under uncertain information," *Proc. of Design, Automation and Test in Europe,* pp. 1060-1065, Apr. 2007.

[101] Q. Qiu, Y. Tan and Q. Wu, "Stochastic Modeling and Optimization for Robust Power Management in a Partially Observable System," *Proc. of Design, Automation and Test in Europe*, pp. 779-784, Apr. 2007.

[102] Q. Wu, Q. Qiu and M. Pedram, "Dynamic power management of complex systems using generalized stochastic Petri nets," Proc. of *Design Automation Conf., Jun. 2000*, pp. 352-356.

[103] L. Mastroleon, N. Bambos, C. Kozyrakis, and D. Economou, "Autonomic Power Management Schemes for Internet Servers and Data Centers," *Proc. of the IEEE Global Telecommunications Conf.*, Nov. 2005.

[104] G. Dhiman and T. Simunic Rosing, "Dynamic power management using machine learning," *Proc. of Int'l Conf. on CAD*, pp. 747-754, Nov. 2006.

[105] Y. Tan, W. Liu and Q. Qiu, "Adaptive power management using reinforcement learning," *Proc. of Int'l Conf. on CAD*, pp. 461-467, Nov. 2009.

[106] Y. Wang, Q. Xie, A. Ammari, and M. Pedram, "Deriving a near-optimal power management policy using model-free reinforcement learning and Bayesian classification," *Proc. of Design Automation Conf.,* Jun. 2011.

[107] Z. Lu, J. Hein, M. Humphrey, M. Stan, J. Lach, and K. Skadron. "Control-theoretic dynamic frequency and voltage scaling for multimedia workloads." *Proc. of Int'l Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, Oct. 2002, pp. 156-163.

[108] A. Alimonda, A. Acquaviva, S. Carta, and A. Pisano: "A Control Theoretic Approach to Run-Time Energy Optimization of Pipelined Processing in MPSoCs," *Proc. of Design, Automation and Test in Europe*, 2006, pp. 876-877.

[109] Wu, Q., Juang, P., Martonosi, M., and Clark, D.W.: "Formal Online Methods for Voltage/Frequency Control in Multiple Clock Domain Microprocessors," *Proc. of ASPLOS-XI*, Oct. 2004, pp. 248-259.

[110] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," *IEEE Micro*, Dec. 2006.

[111] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way Multithreaded SPARC Processor," *IEEE Micro*, 2005.

[112] R.Kumar, D. M. Tullsen, N. P. Jouppi, P. Ranganathan, "Heterogeneous Chip Multiprocessors," *IEEE Computer*, 38(11):32–38, 2005

[113] M. Aater Suleman, O. Mutlu, M.K. Qureshi, and Y.N. Patt, "Accelerating Critical Section Execution with Asymmetric Multicore Architectures," *IEEE Micro*, Vol.30, No.1, pp.60-70, Jan.-Feb. 2010.

[114] J. Sharkey, A. Buyuktosunoglu, and P. Bose, "Evaluating Design Tradeoffs in On-Chip Power Management for CMPs," *Proc. of Int'l Symp. on Low Power Electronics and Design*, 2007.

[115] S. Herbert, D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," *Proc. of Int'l Symp. on Low Power Electronics and Design*, 2007.

[116] C. Isci, A. Buyuktosunoglu, C-Y, Cher, Pradip Bose , and M. Martonosi, "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," *Proc. of Int'l Symp. on Microarchitecture,* 2006.

[117] W. Kim, M. Gupta, G. Y. Wei, D. Brook, "System level analysis of fast, per-core DVFS using on-chip switching regulators," *Proc. of High-Performance Computer Architecture*, 2008.

[118] M. GhasemAzar, E. Pakbaznia, and M. Pedram, "Minimizing energy consumption of a chip multiprocessor system through simultaneous core consolidation and dynamic voltage/frequency scaling," *Proc. of Int'l Symp. on Circuits and Systems*, May 2010, pp.49-52..

[119] E. Pakbaznia, M. Pedram. "Minimizing datacenter cooling and server power costs," *Proc. of Int'l Symp. on Low Power Electronics and Design*. 2009.

[120] Q. Tang, S. Gupta, G. Varsamopoulos. "Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Datacenters: A Cyber-Physical Approach," *IEEE Transactions on Parallel and Distributed Systems*. 2008.

[121] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. "Making scheduling "cool": temperature-aware workload placement in datacenters," *Proc. of Conf. on U.S.ENIX Annual Technical Conf*. 2005.

[122] R. Sharma, C. Bash, C. Patel, R. Friedrich, J. Chase. "Balance of power: dynamic thermal management for Internet datacenters," *IEEE Internet Computing*. 2005.

[123] Q. Tang, S. Gupta, and G. Varsamopoulos. "Thermal-Aware Task Scheduling for Datacenters through Minimizing Heat Recirculation," *Proc. of IEEE Cluster*, Sept. 2007.

[124] S. Biswas, M. Tiwari, T. Sherwood, L. Theogarajan, and F. T. Chong. "Fighting fire with fire: modeling the datacenter-scale effects of targeted superlattice thermal management," *Proc. of Int'l Symp. on Computer Architecture*, June 2011.

[125] J. Choi, Y. Kim, A. Sivasubramaniam, J. Srebric, Q. Wang, J. Lee. "Modeling and Managing Thermal Profiles of Rack-mounted Servers with ThermoStat," *Proc. of Int'l Symp. on High Performance Computer Architecture*. 2007.

[126] C.D. Patel, R.K. Sharma, C.E. Bash, C.E., and M. Beitelmal, "Energy Flow in the Information Technology Stack: Introducing the Coefficient of Performance of the Ensemble," *Proc. of Int'l Mechanical Engineering Congress & Exposition*, Nov. 2006.

Massoud Pedram received a B.S. degree in Electrical Engineering from the California Institute of Technology in 1986 and M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1989 and 1991, respectively. He then joined the department of Electrical Engineering - Systems at the University of Southern California where he is currently a professor and Chair of the Computer Engineering.

Dr. Pedram has served on the technical program committee of a number of conferences, including the Design automation Conference, Design and Test in Europe Conference, and International Conference on Computer Aided Design. He co-founded and served as the Technical Co-chair and General Co-chair of the International Symposium on Low Power Electronics and Design in 1996 and 1997, respectively. He was the Technical Program Chair and the General Chair of the 2002 and 2003 International Symposium on Physical Design. Dr. Pedram has published four books and more than 400 journal and conference papers. His research has received a number of Best Paper awards including two from IEEE Transactions on Computer Aided Design and on VLSI Systems. He is a recipient of the NSF's Young Investigator Award (1994) and the Presidential Faculty Fellows Award (a.k.a. PECASE Award) (1996).

Dr. Pedram is a Fellow of the IEEE and an ACM Distinguished Scientist. He currently serves as the Editor-in-Chief of the ACM Transactions on Design Automation of Electronic Systems and the IEEE Journal on Emerging and Selected Topics in Circuits and Systems. His current research focuses on energy-efficient computing, energy storage, low power electronics and design, computer aided design of VLSI circuits, and quantum computing.