Technical Report Draft Number: OSU-CISRC-8/09-TR41

Green Computing: Modeling the Correlation between Incoming Requests and Power Consumption

Ralston Da Silva, Mike Green, Rajesh Nandagiri,
Rajiv Ramnath and Jay Ramanathan
Collaborative for Enterprise Transformation and Innovation
The Ohio State University

08/26/2009

1 Overview

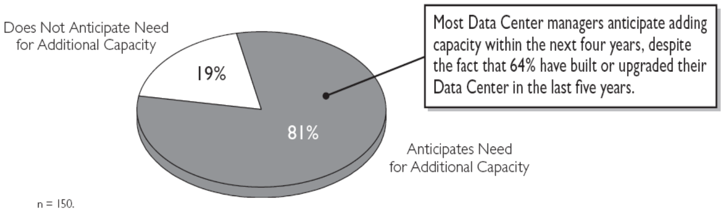
Enterprise data centers require integrated methodologies to manage the rapidly increasing demand for power. Previous approaches, however, have been largely ineffective because they offer no coordinated or comprehensive method to manage power in the data center (DC). In particular, they lack a method to model power utilization or trace its use in the data center, and also lack a comprehensive architecture for power management. This research specifically addresses those deficiencies. We describe (1) an effective method to model power utilization, which is also easy to implement; (2) a method for tracing the use of power in the data center; and (3) a comprehensive data center power management architecture based on these power modeling and tracing capabilities.

2 Business Problem

2.1 Capacity Management

In the past, data centers (DCs) were designed and managed with availability, reliability, and contribution to business value as the principle goals. Capacity limitations were not generally significant, at least with respect to power; i.e., power did not constitute a constraint on the operation of the DC in practice. If power was considered at all in the design or management of the data center, it was only with respect to achieving a data center lifetime which was considered reasonable, typically in the range of 15 - 20 years [1]. As long as this expectation was met, and the enterprise derived sufficient value from the power being used, power was considered no further. Until approximately the last five to seven years, the enterprise suffered no ill effects from this lack of concern with power. More recently, however, several important developments have resulted in power capacity becoming a significant constraint in virtually all data centers.

One of these developments is that the steadily increasing use of information technology (IT) in today's enterprise has led to a rapidly increasing demand for power in data centers.[1] In contrast to the expectation of a 15 - 20 year life cycle of the past; Figure 1 dramatically illustrates the change. Even data centers which have significantly expanded capacity within the last five years are still facing the prospect of insufficient capacity in the next four years.



Most Data Center managers anticipate adding capacity within the next four years, despite the fact that 64% have built or upgraded their Data Center in the last five years.

Figure 1: Percentage of Data Centers that anticipate the need for additional capacity. Source: Hardwiring Green Into Infrastructure and Data

Another significant development has been what may be called a lack of traceability with respect to power, by which we mean the inability to

[1] [1] notes that processing requirements in data centers, and the attendant need for power, grew more than 15 times between 1990 and 2005.

measure how much power is consumed in servicing a particular service request, not just as a whole, but for each component of the transaction. This type of measurement has never been done as well, and it has become increasingly difficult over time. This is because today's enterprise is a collection of a complex set of systems. These complex systems address different operational needs, but work together to address larger business goals. Though these systems are interdependent, they typically have been developed over time and are very different from each other. The wide variety of systems, as well as their complex interactions, makes traceability of power with respect to the operation of these systems more difficult to achieve.

Further, successful enterprises involve constant change due to innovation or attempts to increase efficiency. Indeed, it is often this continuous change that has helped the organization to survive. Constant changes to the enterprise mean constant changes to the different systems that form the enterprise. These changes are reflected in terms of infrastructure changes due to additions to the existing architecture by extending it in lieu of redesigning it, and other implementation changes.

It is difficult to predict the effects of all these changes on the complex enterprise, and each change introduces new challenges in traceability. The people who make decisions that impact the evolution of the enterprise - the path of change taken – need to have good metrics that help them make the best decisions. Due to the complexity of the system, these metrics are hidden in a series of complex parameters, which are dependent on each other. The relationships between these parameters and the business value they represent are not known - which means the people in charge of infrastructure cannot justify their investments to the business stakeholders. The different IT systems implementing the complex enterprise system consist of a huge collection of servers. These servers live in data centers, which house the various applications required by the enterprise. Changes to these applications are thus reflected in the data center. As the enterprise grows, along with becoming more complex, these systems also grow in size and require the addition of new physical infrastructure. This causes the datacenter to grow in size. Typically, data centers are built with the possibility of expansion in mind, but it is not only additional space, but also electrical capacity, that is needed. The ability to expand is always limited, and thus, at some point, the enterprise must build a new data center once the capacity of its existing data centers is reached.

2.2 Increase ROI on power

The complex nature of the enterprise, which is reflected in the data center, as well as changes made to the data center over time, in an ad hoc manner, or sometimes in a more systematic manner - lead to the typical inability to measure and manage power in a comprehensive way. The result is that the business can neither track nor optimize ROI (Return On Investments) or TCO(Total Cost of Ownership) for investments in data center infrastructure (hardware, software, facilities, human resources, etc.), The business has no metrics to determine power costs associated with a particular infrastructure element. This gap in traceability also means that the business cannot make good decisions with respect to infrastructure investment or management.

The enterprise is unable to understand the dynamics of the trade-offs between performance and power consumption, and therefore cannot make sound choices about power expenditures versus value added for the enterprise. On the capacity management front, this amounts to an inability to make informed choices regarding DC lifetime, because the enterprise cannot quantify the tradeoff between DC lifetime reduction and the benefit to the enterprise of making a particular change in DC management or operation which adds some amount of value for the enterprise, but also moves the DC closer to "power out," i.e., a state where the power infrastructure capacity of the data center has been reached.

In order to address the rapidly increasing power requirements, many enterprises have adopted various best practices in recent years, including facilities improvements, virtualization, and measures to dynamically control power usage. Despite these efforts, however, data center power demands have continued to grow at a faster rate than expected [2]. Figure 2 shows that the growth of power demand in the recent past has continued unabated. Ultimately, the only solution is building new DCs. Virtualization, consolidation and using blade servers has slowed this trend somewhat, but data centers continue to run out of power capacity. Building new DCs is costly, and must be justified in terms of ROI. Further, if the underlying issues are not effectively addressed, any newly built data center would be expected to suffer a similar fate to the one(s) it replaces; i.e., it will run out of capacity sooner than expected.

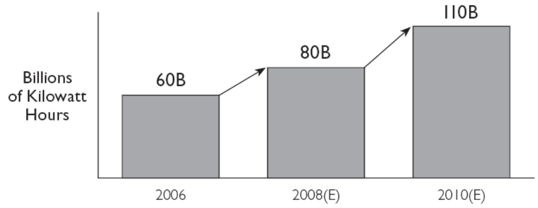


Figure 2: Power Consumption of Data Centers Source: Hardwiring Green Into Infrastructure and Data Center Investments - Data Center Operations Council

Moreover, even if adoption of best practices had been enough to reduce or reverse the growth in the need for power, coordination of the various approaches must also be considered, so that they do not conflict, work at cross-purposes, or otherwise result in significantly sub-optimal data center operation.[2] The literature reveals, however, that few attempts have been made at developing a comprehensive architecture which can be applied to power management, regardless of the particular technologies used in the DC.[3]

Further, we have found no research which addresses the inability to model power, or the lack of traceability of power in the modern data center in a comprehensive way. Specifically, the correlation between a capacity increase and the power consumption increase, as well as the corresponding change in the operating cost for the power consumed, has not been studied.

There is therefore a need to justify these costs from a business perspective. What would be useful is a model that describes the relationship between service level agreements (SLAs), operating level agreements (OLAs), and the power consumed. Such a model would allow the enterprise to understand the relationship between IT performance and power consumed, and therefore to make more informed business decisions about how power is used.

2.3 Carbon Footprint

Over the years, as data centers consume more and more power, their carbon footprint increases. Figure 3 shows a graph from McKinsey's report that places data centers' carbon emissions close to the emissions of the airline industry and steel plants. Data centers need to be more energy efficient and environment friendly.

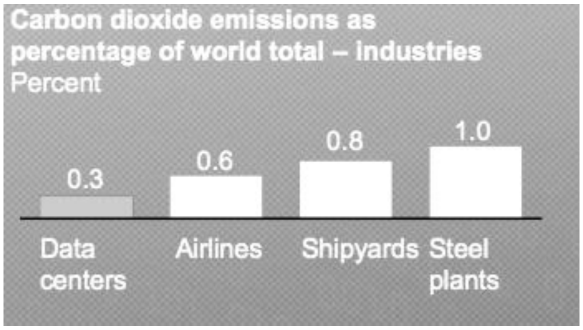


Figure 3: Carbon dioxide emissions of data centers. Source: 2006 McKinsey Report – Uptime Institute

Based on the above, clearly there is a need for the enterprise to reduce the growth in data center power demand, which requires a more

[2] For a discussion of some of the challenges and benefits of such coordination, see, for example, [2] and [3].

[3] One example of work which attempts to address this gap is [3].

comprehensive and effective approach to power management than the piecemeal approach of employing various best practices, with little or no attention to coordination. Such a comprehensive management methodology would benefit the enterprise by: (1) increasing the life of the data center; (2) increasing profitability for the enterprise by maximizing ROI for the data center; (3) reducing costs for power and underutilized capacity; (4) increasing revenue by allowing the enterprise to better leverage the capacity of its data center resources; and (5) preventing "free riding" within the enterprise by allowing development of an improved chargeback model, which more accurately charges business units for the capacity that they use; (6) promoting a greener data center, which may have public relations and perceived corporate environmental responsibility benefits. We explain how our approach offers all these benefits in section 6. Next, however, we characterize the factors in data centers which have led to the current crisis in capacity management and power management

3 Problem Analysis

This section looks at the problem in more detail and describes the problem and the questions the solution is expected to answer.

3.1 Data Center Scenario

Figure 4 attempts to describe the problem at the data center. The data center consists of a huge collection of servers, or physical machines, which have to be provided with power and cooling. The outermost circle represents the data center. The three circles inside the data center represent three physical machines.

Now, if we consider the data center environment to be consolidated and virtualized, then every physical machine will contain one or more virtual machines. (If there is no virtualization, then we can treat the machine as if it is housing a single virtual machine)

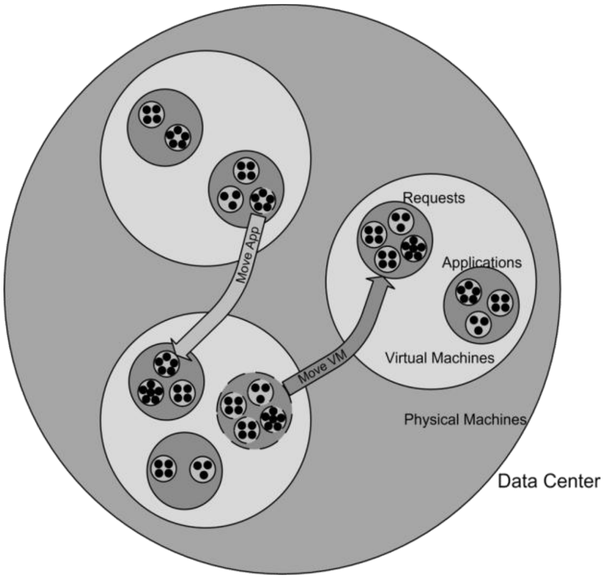


Figure 4: The data center scenario - Physical Machines, Virtual Machines, Applications and Requests

Every virtual machine runs at least one application, and these applications service transactions, which are incoming requests. The requests are shown as black dots in the figure.

The questions the data center management team seek to answer are questions like "Which applications should we club together and put on a single virtual machine?" and "Which virtual machines should we run on a single physical machine? Which virtual machines should we group together to run on the same physical machine?"

Other advanced questions could include "How to dynamically move virtual machines and applications around to make sure we always have an optimal mix on the machines, and at the same time ensure that the SLAs are met?"

The incoming requests, which are represented as dots in Figure 4, have business value associated with them. We want to be able to trace this business value down to the level of resource allocation and power consumption and make more informed decisions when allocating virtual machines to physical ones, or deciding which applications go together on a single physical server.

3.2 Problems with Existing Approaches

As discussed above, the problem in enterprise data center power management is that there is neither an effective approach to modeling power consumption, nor is there traceability of power use in the data center. Beyond these two specific problems, there is no comprehensive architecture for managing power dynamically in the data center that could make use of power modeling and power traceability data to allow optimum tradeoffs between value to the enterprise and power consumption.

3.2.1 Lack of Power Modeling

Power modeling, or the ability to predict power consumption by hardware, and power requirements of applications, is necessary to evaluate the current operation of the data center, and also to predict the effects of changes which are being considered. Any approach which cannot predict power with reasonable accuracy must be based to a large extent on guesswork.

In most data centers, there is relatively little information available on the power being used. Even when data is collected, it is typically not granular enough to be useful in understanding how power is being used, or to be useful in validating attempts to reduce power consumption.

There are two general approaches to the measurement of power. The first approach is direct measurement of the power consumed using an instrument such as a wattmeter. While the most direct, this approach has several limitations: (1) The additional hardware cost; (2) The fact that very few DCs have been built with such direct measurement capability, and therefore it would have to be retro-fit in virtually all data centers. A much more significant problem is that it would be extremely difficult to achieve power measurement that would be granular enough with this kind of approach. Metering individual server racks, or perhaps even individual servers, would be feasible, but metering individual subsystems in each server would be extremely difficult. As we argue below, however, power data on each subsystem is required to enable tracing of power and dynamic power management in the data center.

Since direct measurement of power is problematic, various approaches to measuring power indirectly, or we could say, *modeling power*, have been used. A common approach in the literature has been to model system power use as the aggregate of the power usage of each subsystem, including CPU, memory, disk I/O, network, etc. Both [4] and [5] take such an approach, and demonstrate the general validity and accuracy of modeling power in this way.

Significant advantages of this approach include the fact that virtually all data centers already collect data on subsystem resource utilization, so that if this data can be used to accurately model power consumption of server systems and software applications, the capability to predict dynamic power consumption would be within the reach of many, if not all, enterprise DCs. There are several commercial tools available, for example, Hewlett-Packard’s SiteScope,[4] which collect data on resources such as CPU, memory, disk I/O, and network usage of servers in the data center. Most data centers collect this information and use it for capacity management. Our proposal is to leverage this data, which is already being collected, to model power use of servers and applications in the data center. We

[4] For a description of the tool see: https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-15-25^849_4000_100__

build models of hardware power use, as well as application power profiles, based on the resource utilization data collected. This work is described below in section 6.2.

We point out here that this approach supports other objectives which are also necessary for improved power management. One of them is the ability to model power utilization. Since the use of subsystem resources is what actually consumes power in any computer system, collecting data on the correlation between resource utilization and power consumption allows us to build a model which can predict power consumption.

3.2.2 Inability to trace the use of power

Another difficult challenge is tracing the use of power. By "tracing," we mean the ability to identify how power is used to do IT work, or as we describe it, execute transactions, in the data center. Since a given transaction is typically executed by various physical, and perhaps virtual, servers, in a distributed environment, accurate modeling and sufficiently granular measurements are required to enable this type of tracing. Tracing the use of power is important not so much for the management of power *per se*, but rather for the business purpose of being able to determine how much power is being used by a transaction relative to the value that it generates for the enterprise. This information is useful in various ways, including in making improvements in business processes, allocating scarce resources in the way that will maximize value to the enterprise, and in improved chargeback, as discussed further in section 6.4 below.

Once the subsystem resource use of a given service component running on a particular piece of hardware can be modeled, tracing the power consumed in processing service requests - using the power model described above - also becomes possible. This approach also supports another necessary objective, namely, the ability to dynamically manage power in the data center. For example, as the data center is operating, the power use of numerous different alternative configurations of the data center can be modeled easily, and then the configuration which is best from a power perspective can be chosen.

3.2.3 Inability to curtail rapid growth in power demand

The inability to measure, model, or trace power, results in data centers being unable to get control over the rising demand for it. As new hardware and applications are added to the data center, management is unable to determine precisely how capacity will be affected. Although various best practices have been followed in most data centers, these are to some extent "one size fits all" solutions which may not work well, or at least, may not be optimum, in a given data center. This observation is borne out by the fact that, as discussed above in 2.1, best practices have not succeeded in significantly reducing the rapidly increasing demand for power. We claim that this is because the inability to measure, model, and trace power prevents the enterprise from determining if a given best practice will result in a net benefit or detriment with regard to the demand for power in its data center. Accordingly, our hypothesis is that implementation of the methods which we propose for measuring, modeling, and tracing power will significantly improve the ability to manage power demand in data centers.

3.2.4 Lack of a comprehensive architecture for dynamic power management

We offer a comprehensive data center architecture which is general enough to be implemented in virtually any data center. Further, our method of hardware and application profiling, for tracing power use in the data center, are also relatively easy to implement, by using tools which typical data centers already have available. As we detail below, the architecture does not have to be implemented all at once. It can easily be implemented in stages that allow for continuous improvement of power management in the data center.

The final factor in the failure to effectively manage power is the lack of a comprehensive architecture for dynamic power management. As pointed out above, few attempts have been made to develop this type of architecture for data centers. Such an architecture would serve the role of coordinating the various methods that are used in the data center for power management, including power modeling and tracing, as described above. The management architecture would also allow dynamic changes to be made to the configuration of the data center as IT and power demand changes. Without such a comprehensive architecture, capable of dynamically responding to changing DC conditions, there can be no assurance that significant amounts of power are being wasted at any given point in time. Our architecture prevents this by monitoring IT load and power conditions, and determining if some alternate configuration of the DC would use less power while doing the same IT work and meeting all relevant Service Level Agreements (SLAs). While doing this, however, we also need to quantify the cost of dynamic load balancing, which includes the cost of monitoring and the cost of actually implementing the dynamic changes.

Thus, our research addresses the types of problems which have led to the current challenges which enterprise DCs face in managing power. In particular, our research addresses the gaps in modeling power and in tracing the use of power, as well as the need for a more comprehensive approach to management of power in the modern enterprise data center, by developing a reference architecture within the Adaptive Complex Enterprise (ACE) framework. The ACE framework provides a conceptualization of the modern service enterprise, as well as abstractions for modeling transactions in the enterprise. These abstractions provide mechanisms for realizing traceability of power, and also for managing power in the data center, both locally and globally. Our model enables the enterprise DC to function in a sense and respond (SaR) manner, by taking account of performance per unit of value generated for the enterprise, as well as power requirements. Such SaR operation enables the enterprise DC to manage power while responding dynamically to ever-changing service requirements, by effectively coordinating the diverse technologies within it.

3.2.5 Virtual Machines: Flexibility versus Overhead

We have also developed a general methodology for allocating applications to virtual machines. A key issue is that, for a given number of applications, a larger number of VMs gives more flexibility, but also involves more overhead in running the VMs. For example, if we have ten applications, and we put them on ten separate VMs, then we have the maximum flexibility with respect to how these ten virtualized applications can be matched to physical servers, but we also have ten different VMs, which means we have roughly ten times the overhead incurred by one VM. On the other hand, if we put two of the applications on a single VM, we have five VMs total, which means we have fewer choices in how to match the applications to physical servers, but we only have half the overhead. The tradeoff between flexibility and overhead has to be managed well, to obtain as much benefit as possible from the flexibility without incurring excessive overhead costs.

3.2.6 Application to Hardware Matching

The aim behind application to hardware matching in the data center is to optimally match applications to hardware with respect to the amount of resources consumed by the application. One problem for application to hardware matching is that the types of hardware running in the data center can be heterogeneous. In this sense, this type of matching is similar to what Nathuji et al. call *platform heterogeneity* [12]. Since applications vary in their use of different system resources, a corresponding variation in power use on different hardware systems is to be expected. Application to hardware matching attempts to take advantage of this variation by optimally matching the resource utilization profile of the application to the power characteristics of the physical server system on which it is deployed. Although this idea appears simple enough in principle, we have found no previous research which provides a methodology for doing matching along these lines. We suspect that this is due to the fact that good power models for applications running in DCs have not been generally available previously. Without a power model that correlates application resource utilization with application power consumption, and also hardware power consumption, the only way to do application to hardware matching is to actually run each application on each of the different server systems running in the data center, and study it's impact on power. This appears impractical, and the benefits may not outweigh the costs. By modeling power in terms of resource utilization, we can overcome this problem, because we can accurately estimate the power use of the application running on a particular server system without actually running it.

3.3 Traceability Limitations

As explained in 3.2.2 above, another gap in previous work is the ability to trace the use of power in the data center. While it is critically important for data center managers to know how much power is being used, which is addressed by power modeling, it is arguably just as important to know how the power is being used; in short, how much power is too much (or too little) depends on its use. If the use is important enough, a great deal of power may not be too much. There is no way, however, for the enterprise to make these judgments without detailed information about how power is being used in the data center.

3.4 Need for a Comprehensive Architecture

We now address the need for a comprehensive architecture for dynamic power management. The only attempt at something approaching a general data center architecture in the literature is [16]. This work generally incorporates the advantages of virtualization, heterogeneity awareness, and abstractions for power management in virtualized systems, all of which have been identified above. This paper also provides for coordinated management of IT power and cooling power. Therefore, we generally adopt the high-level features of the architecture for data centers that is presented in [16]. We extend it and generalize it to address certain significant limitations..

In particular, the framework appears to be limited to data centers which have room-based, as opposed to hot and cold aisle, or some other cooling infrastructure configuration. The framework also appears to be based on Computer Room Air Conditioners (CRACs), as opposed to Computer Room Air Handling Units (CRAHUs), which operate on a different principle. We, however, wish to develop an architectural model which is general in nature, and which could be applied to any data center. The details of our cooling power management module are discussed in the next section.

The model in [16], although it provides for communication between the cooling power management module and the IT power management module, does not appear to provide for information to be fed back to the IT power module to assist it in reaching globally optimum decisions on IT power management, i.e., decisions which minimize the net power consumed for both IT and cooling. We add a feedback mechanism to overcome this limitation.

In addition, the model of Nathuji et al. is based on VMs running on the Xen hypervisor, although the model does not appear to depend critically on this fact. We generalize the model so that it can be used in a DC with VMs running on any hypervisor, or even a mixture of different hypervisors throughout the DC.

A fourth limitation of the management architecture in [16] is that it has no clear component for modeling or predicting the power use of particular applications, or of specific hardware. This limitation also results in a lack of traceability. Nathuji et al's model appears to be aimed primarily at reducing power consumption, rather than on identifying the correlation between servicing requests in the data center and the power used. We address this limitation by developing a model of power use that relates to resource utilization, and can therefore capture this correlation. This model is explained in sections 5 and 6.

3.5 Need for a combined approach

The two key factors listed below, make the use of static measures alone insufficient for managing power in the data center. Accordingly, we propose that a Sense and Respond (SaR) architecture is necessary for optimizing power use.

First, power consumption in modern data centers varies widely, even over short periods of time. Two major consumers of power in data centers are IT and cooling [20]. Without even considering the power used by the cooling infrastructure, the dynamic variation due to IT power fluctuations is significant. [21] reports a range of variation between 45% and 106% for typical enterprise class servers.

Second, although current servers use relatively less power at idle than previous generations of hardware, every server consumes some power at idle, while at the same time it does no useful IT work. As can be seen from the range of power variation cited above, even modern hardware typically consumes at least 50% of its peak power consumption even at idle, as can be seen in Figure 19. For this reason, significantly reducing server idle time, in order to increase server utilization as much as possible, has become critical to minimizing power use while still maintaining required performance levels.

Because the number and nature of incoming requests, and the corresponding power use in the data center vary significantly over time, no single static configuration of the data center will be optimum at all times. Rather, the power management mechanism for the DC must be able to respond dynamically to changing, and to some extent unpredictable, conditions. In this sense, the DC management mechanism cannot simply attempt to service the routine requests which it receives while minimizing cost per request. Rather, the DC must be able to sense ever-changing IT and power requirements, in order to dynamically respond to the requests in a way that optimizes the tradeoff between resource utilization and value generation. Thus, we propose that a Sense-and-Respond architecture, which is capable of this type of dynamic adaptation, is required.

3.6 Benefits of Traceability

Traceability gives the organization many benefits. The unit of execution that can be traced is called a transaction. Tracing the workload and treating it as transactions allows us to do the following:

1. **Business Value of a Transaction**
   Traceability allows us to track the business value generated when a transaction is executed. It helps us to provide the business stakeholders with fine grained information. For example, we can now keep track of the revenue associated with a particular transaction.
2. **Resource consumption of a transaction**
   With the fine grained data that we have, we can also calculate the amount of resources needed to execute a particular transaction. For instance, we could predict the amount of

CPU, Memory and I/O needed for a particular request. This information helps us to allocate resources efficiently and to increase the utilization of the resources. Higher resource utilization means lower idle time, which means that the resources are being utilized properly.

3. **Power consumption of a transaction**
   The resource utilization data collected can be used as a proxy to determine the amount of power consumed. Quantifying the amount of power consumed by a transaction allows us to calculate the cost associated with servicing the requests. If the fixed and variable costs of the resources are taken into consideration, we could have a means of quantifying if the energy spent is worth the business value.
4. **Carbon footprint of a transaction**
   Once we quantify the power consumption of a transaction, we can convert this into carbon equivalent values, and hence we can calculate the carbon footprint of transactions. This information could be used to provide new services – like inform the customer of the carbon footprint of the transaction, etc, and can be also used to make the data center more competitive, by having them inform the user of how power-efficient they are.

4 Green Practices

Existing research in green computing has led to several recommendations for making the data center green. These green practices have made their way to being best practices for any organization. This section lists these green practices and describes each of them.

4.1 Server Consolidation

Most of the servers running in a data center are running close to idle. The servers are not very efficient when run at idle, and thus consume a lot of power. However, at higher utilization levels, the servers are more power efficient. Thus, if we consolidate many applications onto a fewer number of servers, the servers can be run at higher utilizations and would be more power efficient.

4.2 Virtualization

From a power conservation perspective, by virtualizing and consolidating a number of applications on one server, all of which were previously being run on multiple physical servers, two beneficial changes occur [6]. First, all of the other servers can now be shut down, and therefore, the power they were consuming previously is saved. Second, the physical server on which the applications are consolidated will spend much less time at idle, and therefore, its otherwise wasted idle power will be significantly reduced. Virtualization and consolidation can be done statically, i.e., a number of applications being run on multiple servers can be virtualized and consolidated permanently on a single physical server, but consolidation of multiple virtual servers on a single physical machine can also be done dynamically, which could provide even greater flexibility and power savings, discussed below in 4.6.

4.3 Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) is another method for reducing power, and can reduce both unnecessary server idling, which consumes power with no benefit, and also over performance of server systems, where the server executes the application at a higher performance level than is required by the relevant SLA, and therefore also uses power unnecessarily. This method is highly dependent on hardware characteristics, and perhaps on the ways in which those hardware characteristics can be manipulated. Some server systems allow CPU voltage and frequency scaling (VFS) to be set in the system BIOS; others allow VFS to be changed dynamically by the operating system; some systems also allow VFS to be done by applications. W. Bircher and L. John [13] investigate dynamic frequency scaling, specifically in multi-core processors, but also illustrates the very significant power savings that can be achieved with little or no loss of performance. [14] shows a useful approach to control voltage and frequency scaling dynamically using a feedback control loop, and accompanying significant reductions in power consumption.

4.4 Smarter Cooling Solutions

Two papers make differing, but persuasive, arguments that data center power consumption cannot be optimally controlled without managing both IT power and cooling power in a coordinated fashion. Nathuji et al. argue that managing IT power and cooling power independently may lead to less than optimal results [16]. For example, a particular workload allocation to some set of VMs on certain physical servers may result in minimum power consumption on the IT side while executing that workload, but if this workload allocation results in hot spots in the data center, so much additional power might be required on the cooling side to remove the excess heat from the hot spots, that any power savings on the IT side will be negated, or even exceeded by, the additional power that is needed for cooling. Further, while a different allocation of the IT load may be less than optimal with respect to IT power consumed or with respect cooling power consumed, it may still be the best option from the point of view of combined IT and cooling power. Clearly, since both IT and cooling consume significant amounts of power in the data center, an approach must be adopted which minimizes the net power consumed, rather than minimizing power on either side independently, which will often result in a less than optimum solution with regard to the net power consumed.

Niles makes the somewhat different argument in [17] that, not only must IT power and cooling power be considered and managed in a coordinated fashion, but beyond this, that cooling infrastructure in the data center must provide for row-based cooling, i.e., more granular control of cooling. This is necessary, Niles argues, because virtualization, along with high-density servers, makes dealing with the presence of hot spots in the data center a constant challenge. We can observe that data centers that do not have granular control of cooling pay a heavy price for hot spots, because the entire room-based cooling system must be run at a higher level in order to remove the heat that is created in the hot spot areas. This presents the real possibility that any power savings from dynamic migration and consolidation of workloads running on virtual servers will be more than negated by hot spots that are created by running such loads on high-density servers. The very disconcerting result is that one of the principal advantages of virtualization for reducing power usage in data centers is lost. Nathuji et al., however, propose an approach for addressing this challenge which does not appear to require row-based cooling [16]. The key idea is to make use of fine-grained intelligence gathered regarding the ability of the cooling system to dissipate heat, and regarding temperature effects in the data center, to avoid the occurrence of hot spots in the first place.

4.5 Continuous Monitoring and Redeployment (Dynamic Migration)

S. Niles, in an APC white paper, discusses benefits of virtualization, including the fact that it enables dynamic migration and consolidation of workloads based on IT resource demands [7]. When a physical server is being utilized at a low level, its virtual servers can be migrated to another physical server. This provides the opportunity for physical servers to be dynamically powered up or shut down in response to changing loads, which reduces total data center power consumption [8]. Such an ability to migrate and consolidate workloads supports a dynamic architecture for data center power management, because it enables dynamic allocation of data center resources based on the current demand, which is subject to significant fluctuation.

Recent research has focused on models and mechanisms for managing virtual machines (VMs) in the data center, in order to manage dynamic migration, and to take maximum advantage of the power savings that virtualization and consolidation offer [2, 9, 10, 11].

4.6 Heterogeneity Awareness

R. Nathuji et al. discuss yet another advantage of virtualization [12]; namely, it provides a looser coupling between the IT load and the underlying physical platform. This loose coupling can be leveraged by seeking optimal matches between workload characteristics and the hardware on which it is run. In typical data centers, the heterogeneous nature of the physical platforms present in the data center presents the opportunity to save significant power by matching load characteristics and hardware. Nathuji et al. report an average reduction in power use of 20% for one such approach [12]. Our approach to power modeling also supports such matching, by allowing characterization of the load in terms of resource requirements, and by facilitating the identification of hardware, through examination of its power consumption in terms of resource utilization, which is optimal for the load.

4.7 Cloud Computing

Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. A number of advantages of cloud computing for the enterprise have been recognized. Among these are improved TCO, reduced infrastructure costs, improved business agility, converting fixed costs to variable costs, and of course, with respect to power, acquiring IT services which can be scaled as needed, even on a temporary basis to deal with peaks in IT requirements, and which do not impose power capacity constraints [18]. Although enterprises have been hesitant thus far to adopt cloud computing broadly as a source of IT services, there are cases where cloud computing might be advantageous, without raising the typical concerns about security, reliability, meeting SLAs, and other issues raised by cloud computing which have not yet been fully resolved. [18] suggests that there are four cases where the cloud should be considered by CIOs, namely, for: (1) new initiatives where budgets are very constricted; (2) business processes which have widely varying or unpredictable load patterns; (3) services provided by non-core systems which are commoditized; and (4) systems where infrastructure management and operations costs are high.[5]

Our own view is that, although cloud computing can be part of an overall capacity management strategy at the present time, it will probably continue to play a somewhat limited role for most enterprises, until the issues which currently limit its use can be satisfactorily resolved. There can be micro consolidation, where we consolidate virtual machines onto physical servers within the data center, and also macro consolidation, where we consolidate virtual machines over geographically separated areas – allowing us to reduce cooling costs by taking advantage of climatic and weather differences of these geographically separated areas. For instance, we can run a high compute load in America during the night, and then during the day, we can run it in a data center in China, thus saving on cooling costs by always having the application run at night, when the atmosphere has significantly lower temperatures.

4.8 Strategic Replacement of Hardware

Strategic hardware replacement attempts to replace older, less efficient hardware with newer systems which have been designed with features which support power conservation. Once power usage can be accurately modeled, we can determine if replacement of certain hardware would result in a large enough power reduction, and if it will offset the total cost of the hardware and the person hours required to replace it. In particular, it is anticipated that certain server systems in typical DCs may be more heavily utilized for certain commonly used applications, and thus may be using a significant fraction of the power in the DC. If these systems were replaced with more modern energy efficient hardware, the total power savings could be considerable. We point out that such a strategy can only be undertaken, though, once power usage can be reliably modeled at the level of server system resource utilization. Several major hardware manufacturers are offering server systems that are significantly more energy efficient than in the past [15]; again, though, the more information DC managers have about the resource utilization of their applications, the better the choices they can make with respect to the optimum characteristics of hardware they choose to replace less efficient systems.

5 Our Approach

The approach we take is to have a model of the datacenter, where we can trace the use of power down to the level of individual transactions performed on the data center. Doing this helps us to quantify the value generated by the power spent, and hence provide the business stakeholders with information they can use to make decisions concerning the data center.

Our approach uses resource utilization as a proxy for power. We convert incoming requests into their corresponding resource utilization, and then use a model for power to convert the resource utilization into power. This section gives an overall view of how this process is carried out.

5.1 Traceability in the data center from requests to power

The incoming requests are monitored and a trace is maintained of all the applications that are used to execute the request. Figure 5 describes how the execution of every incoming request can be split up into executions of transactions serviced by various applications.

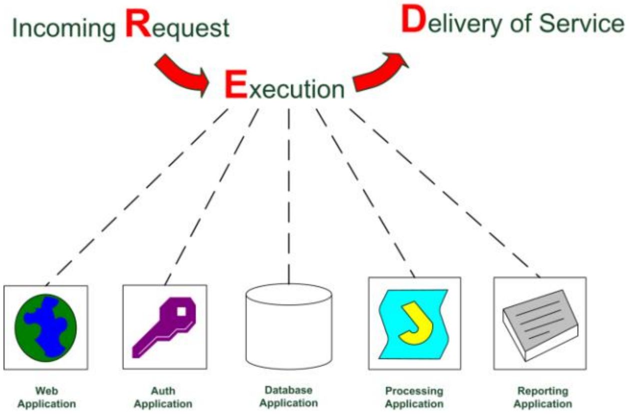


Figure 5: Tracking the application usage of each incoming request

Thus, we can see that every incoming request can be split up into multiple transaction requests for various applications. Now, each application ideally runs on at least one server, and in turn requires resources to execute the transaction. Figure 6 describes how the application requires multiple resources to service a transaction. The usage of these resources can be programmatically collected from the operating system, or can be approximated by knowing the transaction type.

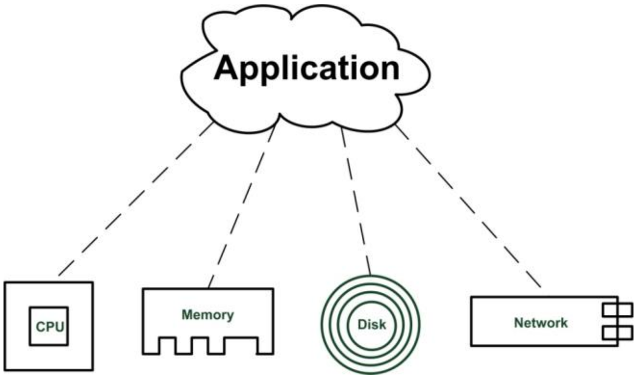


Figure 6: Tracking the resource usage of each application

Every resource on a server is an actual hardware device (Except in a virtual machine, where there is another level of abstraction) and these pieces of hardware consume power. If we study the correlation of resource utilization and power consumption we can have an accurate model of the power consumption of a particular server. Once we learn this correlation, we can predict the power consumption of a resource. This is illustrated in Figure 7.

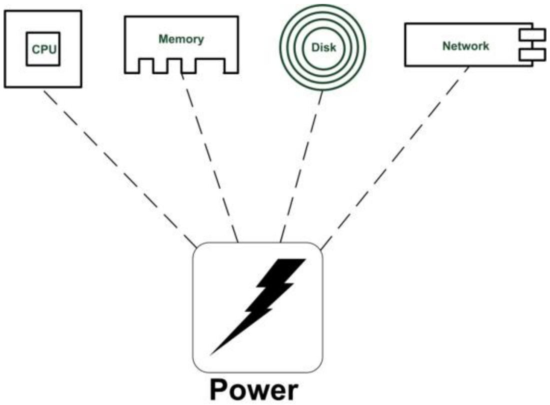


Figure 7: Converting the resource utilization into power

Now that we have studied how the incoming requests can be traced to their usage of different applications, how the different applications utilize different resources and how the different resources consume power to service the requests, we are at a point where we can connect the dots and be able to predict the power consumption of individual requests.

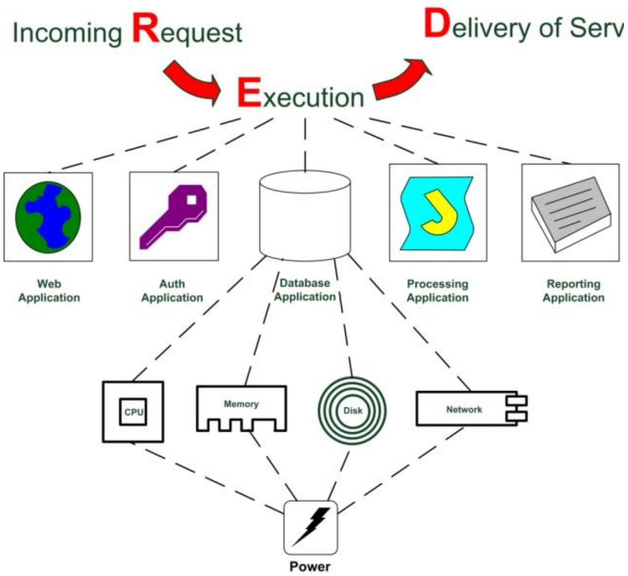


Figure 8: Putting it all together - the correlation of incoming requests to power consumption

Figure 8 combines Figure 5, Figure 6 and Figure 7 and provides the big picture of how the power consumption can be traced to each incoming request.

5.2 Using traceability and profiling to promote an optimized datacenter

What we need now have is a method to convert incoming requests into power. This profiling provides the business with information on how the power is used in an organization. This traceability can be used to track requests types and find out which requests consume the most power. It also enables the organization to have a fine grained information on their power consumption and plug leaks in the system.

In this section we describe how we can use profiling of applications and machines to understand the patterns of application usage and power consumption. We can use this traceability to optimize the operations of the data center.

5.2.1 Machines running at higher optimizations - close to the SLA

We study the SLA of applications and determine the resources needed to maintain this SLA. Usually, in today's enterprise the applications are given more resources than required so that they can maintain the SLA. When a single application resides on a machine that is designed for peak load, we are in a situation where we are actually doing better than the SLA. This might not add any business value.

With consolidation and virtualization, we can now run applications close to their SLA by having multiple apps on the same physical machine. One of the ways to allocate applications/VMs to physical machines can be to allocate high business value apps first, and then proceed to add the lower business value apps, while trying to minimize the total number of servers used. In other words, we are using the business value to allocate resources to transactions that are executed by the apps.

We could make the enterprise even more adaptive if we allocated resources based on the incoming transaction request. We can collect all the resources in a priority queue, and execute the higher business value transactions first, while making sure the SLAs for all the applications are met. We can thus provide higher reliability and a better service, for the transactions with a higher business value.

5.2.2 Align expected SLA with business value of services (Business value Profiling for SLA)

Another important thing to do is to align SLA to better reflect the business value. Many times, the enterprise assigns a random SLA to the application. The SLA must be dependent on the business value and the organization must be able to quantify and justify the SLA requirements of an application.

5.2.3 Correlate SLA to resource utilization (Application profiling)

Our method for resource profiling of applications involves collecting a large sample of subsystem resource utilization data for the application (typically one week or more), and then calculating several key values for the application. These values include the average utilization for each type of resource (CPU, memory reads/writes, disk I/O, network), and also the standard deviation for utilization of each resource. Temporal patterns or cycles can be obtained from the resource utilization data; for example, some applications may be more heavily used on certain days of the week, or at certain times of the day. All of this information is useful for matching the application to the hardware on which it runs (described in 5.2.4, below), and if the application is virtualized, deciding which applications to combine on a single VM.

5.2.4 Convert resource Utilization to power consuption (Hardware Power Profiling)

We have developed a methodology for characterizing the power profile of a computer system. More detailed descriptions of the tools used can be found in section 6; here we give a high-level description of the methodology. The first tool is a synthetic workload generator, which is designed to run the subsystems of the machine at a wide range of loads, and in various combinations. For example, the CPU is run at a range of values from 0% to 100%. Along with the workload generator that is run on the hardware, concurrent system power measurements are taken using a wattmeter. Subsystem resource utilization can be fetched from the operating system, and these values are collected at regular time intervals, as the workload generator is running. Finally, a model (a hardware power profile) is constructed which correlates system power usage with resource utilization. The model can be constructed using a neural network, or using regression.

6 Our Contributions

Section 5 summarized our approach to solving the data center power management problem. We described how we introduce traceability in the data center, and introduced the terms “business value profiling”, “application profiling”, and “hardware profiling”. This section goes one step further and describes how we implemented each of these profiling techniques.

6.1 Power Management

Here we describe, at a high level, how the management mechanism which will monitor conditions in the data center, will be capable of making dynamic changes to reduce power consumption or meet service requirements.

The data center consists of a large number of physical servers. In a non-virtualized environment, each physical server may have one or more applications running on it. In a virtualized environment, a hypervisor will run on the physical hardware, and there may be one or more virtual servers / virtual machines (VMs) running on the hypervisor. In turn, there may be one or more applications running on each virtual server. Power usage, load, and business value information will have to be transmitted across the various levels in this hierarchy: physical machine – hypervisor – VM(s) – application(s).

The data center houses many physical servers. In a virtualized data center, more than one virtual machine runs on a single physical server. Also, each of these virtual machines can run multiple applications. The process of running multiple applications/virtual machines on a single server is called consolidation. As described in section 4.1. Consolidation increases resource utilization and hence increases efficiency.

The following diagram is a model of a data center which uses Virtualization and Consolidation.

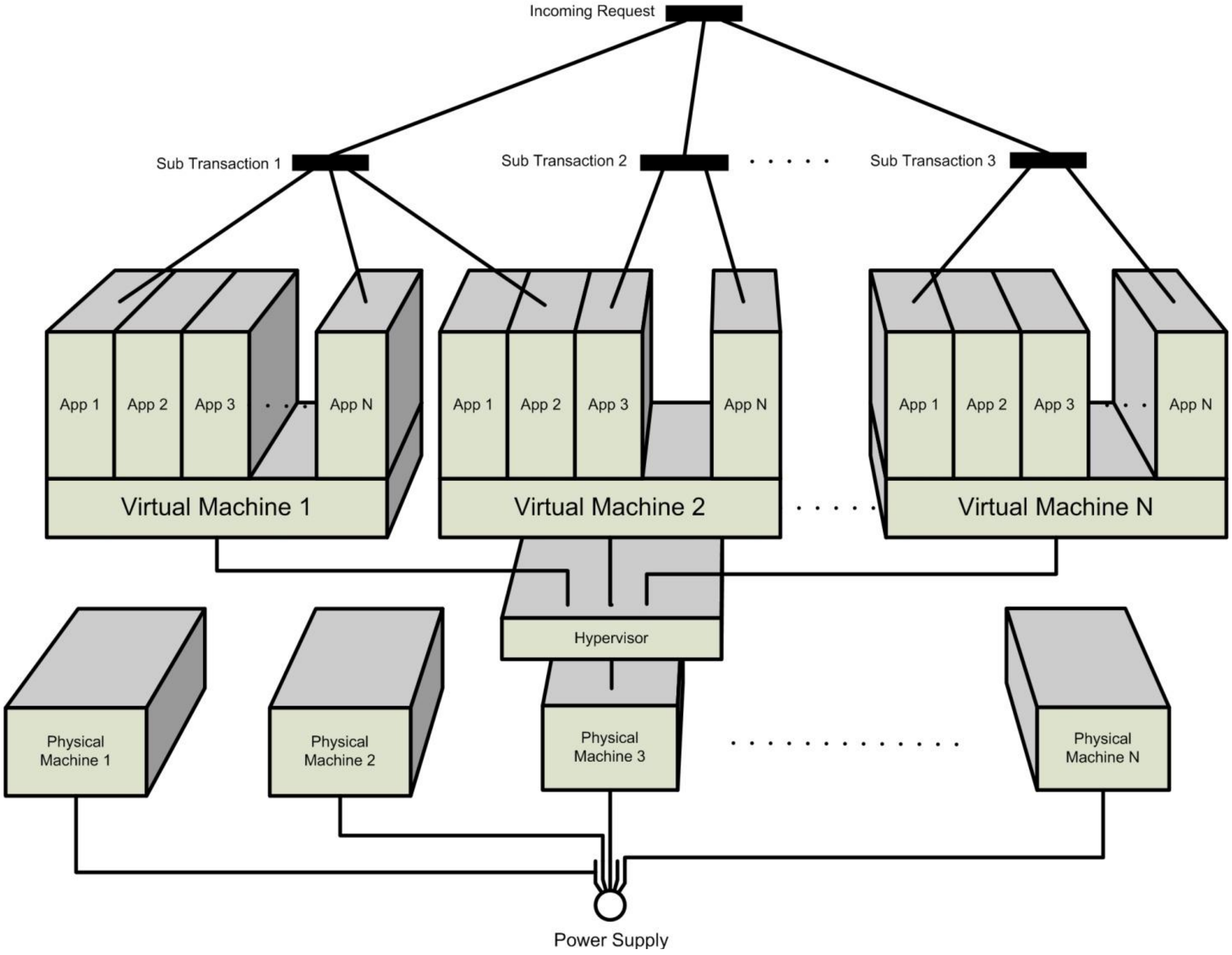


Figure 9: Model of a data center that uses virtualization and consolidation

The data center represented by Figure 9: Model of a data center that uses virtualization and consolidation, illustrates how incoming requests can be broken down into transactions and sub-transactions and how these sub-transactions are serviced by different applications residing on different machines.

Traditionally each application ran on a single dedicated server. This guaranteed availability and reliability, but proved to be very inefficient. The inefficiency was due to underutilized servers that were employed with over-provisioning to guarantee availability. In this model, the physical servers are virtualized and applications run on these virtual machines. A single physical machine can be used to run multiple virtual machines. Each physical machine has a hypervisor installed on it over which all these virtual machines run.

Running many virtual machines on a single physical machine ensures that the physical machine is fully utilized, or at least that the utilization is at a power efficient level. Running servers at low utilization levels is inefficient [23]. As most servers are designed to run at a certain utilization level.

Running multiple virtual machines on a single physical server, however, presents new architectural challenges. We need to decide which application/virtual machine combination is good and which is not. For example, running two computation intensive applications on the same virtual machine probably would not provide the benefits that could be gained by running a combination of computation intensive and I/O intensive applications on a single virtual machine.

Further, we can divide our decisions into two broad categories: Static and Dynamic.

Static Decisions: The architectural decisions that comprise deciding which applications run on which virtual machines. This requires a study of the application profile, and understanding what kinds of resources the application requires.

Dynamic Decisions: consist of deciding on which physical machine a particular virtual machine runs. This is a decision that has to be made in real-time, after studying the current workloads of the virtual machines.

To make these decisions, the infrastructure business unit needs to have granular data on the current load, the current power consumption etc, and we hence need a system that allows for traceability. We need to quantify business value of applications. This information can be gained by tracking the incoming requests and quantifying their value across the different levels of the data center. Having such traceability will enable us to make better decisions when allocating virtual machines to physical machines. For example, when a physical machine is overloaded, we can decide to move out a virtual machine. The traceability will help us to move out the virtual machine with the lowest business value, thus reducing the impact on the machines running high business value applications.

All this points out the need to monitor business value and cost at different levels. We need to make some decisions locally and some decisions globally. We propose a global power manager to make dynamic decisions and decide which virtual machine runs on which physical machine. In addition to the global power manager, we propose local power managers running on each physical machine, virtual power managers running on each virtual machine, and some form of application power manager running along with each application. Figure 10: Hierarchy of Power Managers, shows the power managers at different levels of the system. Business value information is propagated to the global power manager. The upward arrows indicate the power/resource utilization data moving to the layers above, and the downward arrows indicate the business value and SLA information that is sent to the global power manager from the incoming request. The global power manager uses this information to make its decisions.

The global power manager and cooling power manager can also communicate, so that IT power and cooling power can be managed in a coordinated manner.

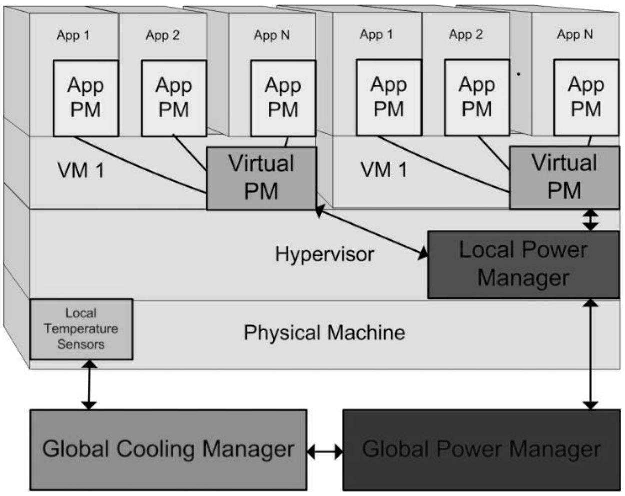


Figure 11: Hierarchy of Power Managers

This approach to solving the problem is a control systems approach. The global power manager tries to optimize the allocation of virtual machines to the physical machines by taking into account the business value and power consumption. The details of this approach are explained in the next section.

6.2 Power profiling of hardware

Section 6.1 suggested a control systems approach to managing the data center. The control system needs a feedback loop where current power consumption information is collected and sent to the global power manager. Implementing such a system has drawbacks

1. It is expensive and requires a complex set of power information collection devices. E.g. Watt meters spread across the entire data center.
2. It cannot provide the data quickly – as we need to have real time information to make the decisions – however there is a time lag introduced when external systems have to interface with the global power manager.
3. It cannot predict the power consumption- this system can only report historical data – thus we cannot have what-if situations where we predict the impact of architectural decisions on power consumption.

All this leads us to having a system where we use a proxy for power and measure this proxy programmatically using the current IT systems which are already in place. This section describes how we use resource utilization as a proxy for power.

6.2.1 Workload Generator

One of the steps in the proposed methodology is building a hardware profile of the physical machines. The detailed method is described in section 6.2.2 which learns the profile of a machine. To ensure that the whole profile of the machine is learned, the physical machine must be loaded at different levels. For example, if we only run the hardware at 20% load, and then run it at 80% load, the whole profile of the machine will not be learned. Further, a large percentage of the time, servers run at 0-5% CPU Utilization. Thus, if we try to learn the profile of a machine, and just run the machine for some time, we will only learn the profile for resource utilization around 5%. If the power consumption curve is not linear, we might end up using this information to predict power consumption at higher utilization levels, which would be a wrong indication of the power consumed.

To learn the profile of the hardware at varying loads, we use a workload generator. The workload generator generates load for the physical machine. In Figure 12: Workload Generator, shows how the workload generator generates the amount of resource utilization specified by the Resource Utilization Specification. The resource utilization could be varied to make the workload generator generate various amounts of load.

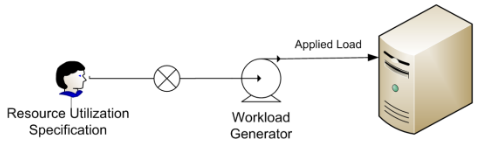


Figure 12: Workload Generator

This method of generating load would suffice, if there were no external load acting on the system. However, in reality, there is an operating system running on the physical machine, and this operating system generates some load that has to be serviced by the machine. If there are other programs installed on the machine, they also generate varying amounts of load. To produce a specified amount of load, then is not a straightforward task, and needs a system that increases or decreases the load produced so that the net effect is that of the desired load. Figure 13 depicts the workload generator implemented with a feedback loop. In this system, the feedback loop constantly measures the load on the system and the either increases or decreases the amount of load generated in an attempt to generate the specified load.

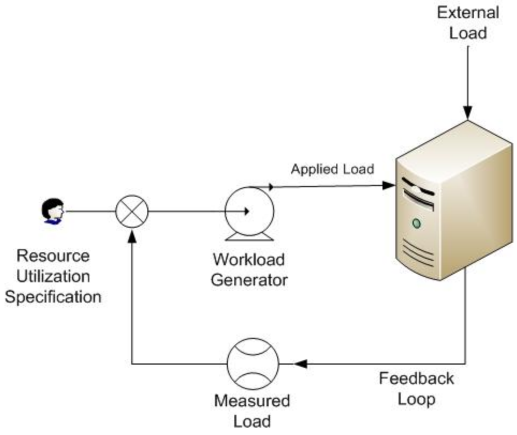


Figure 13: Workload Generator with feedback loop

The workload generator will be implemented to generate CPU, Memory, I/O and Network Load. As of now, it generates only CPU load. The CPU is the main consumer of power, and so we decided to generate only CPU load initially, to test the general validity of the approach.

If the time interval of feedback is small, there will be a lot of variation in the load, and it will be difficult for the workload generator to match that load. We conducted an experiment where we collected CPU resource utilization data of a machine without generating any artificial load. We used CPU because the CPU is the major power consuming resource, and also the resource that has the most variation in load. The profile can be seen in the Figure 14.

Here are the specifications of the machine.

OS	Microsoft Windows XP Pro
System	Dell OptiPlex GX270
Processor	Pentium® 4 2.60 GHz
RAM	1,024 MB

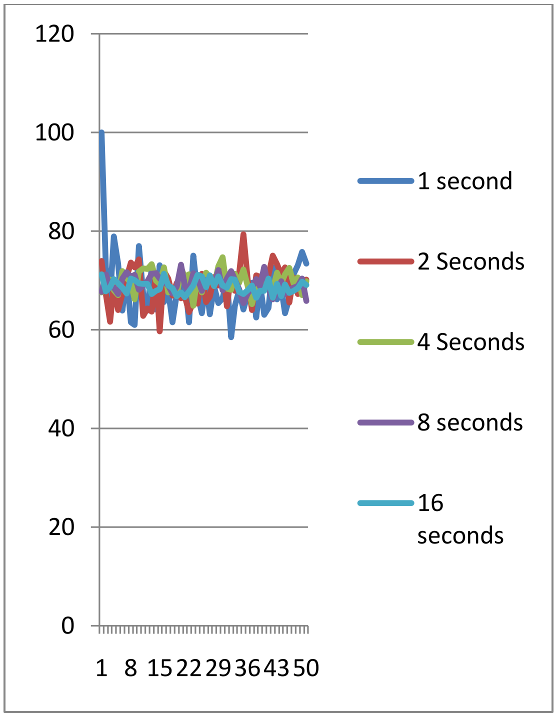


Figure 14: CPU utilization of idle machine

As we can see from Figure 14, the CPU Utilization of an idle machine is not constant, but varies a lot. The figure shows the consumption when power is sampled at 1, 2 or 4 seconds. However, as we increase the interval beyond 4 seconds, the utilization tends to stabilize. It is easier for the workload generator to match a stable utilization. Figure 15 is a magnified view of the idle CPU. From this graph we can see that at 8 seconds, the CPU utilization is relatively stable. In our workload generator, therefore, we measure CPU utilization at 8 second intervals, as we generate CPU load.

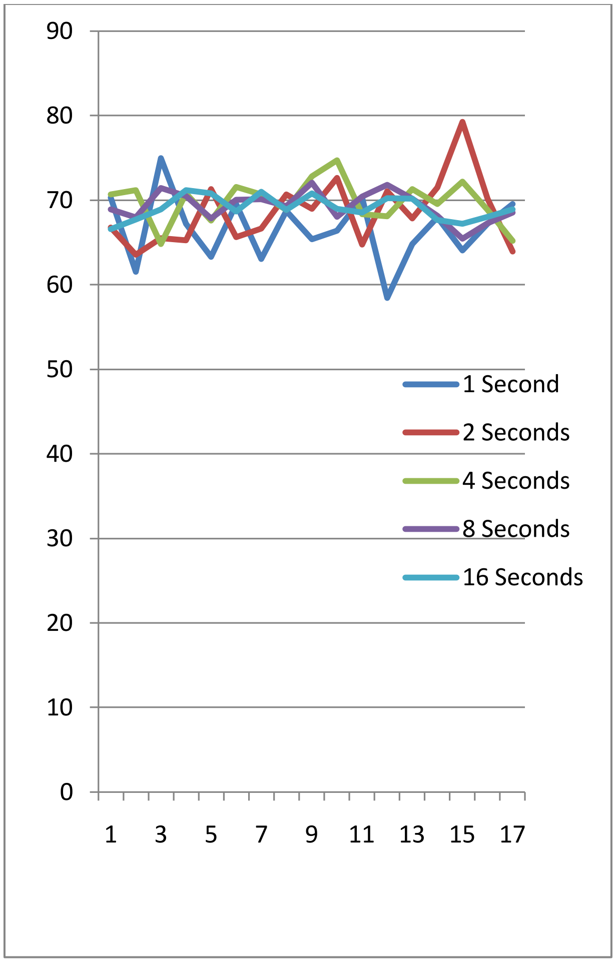


Figure 15 : More granular view of Utilization of an idle CPU

The workload generator is implemented in C++ and is written in a way where a few classes can be rewritten and the code could be ported to any operating system. Figure 16 describes the class diagram of the synthetic workload generator. WindowsOS, WindowsStatus and WindowsThread are classes that are specific to the operating system Windows. The CPULoadThread class generates CPU load. It uses different worker threads to generate load for each processor on the system. The Synthetic workload generator class is the main class and it accepts the required amount of load and then sends that number to the CPULoadThread.

The WindowsStatus class reads the status of resource utilization from the Operating System and presents it to the SyntheticWorkloadGenerator class. The WindowsOS class is used to do things like write an output to the console, and can accept inputs from the users.

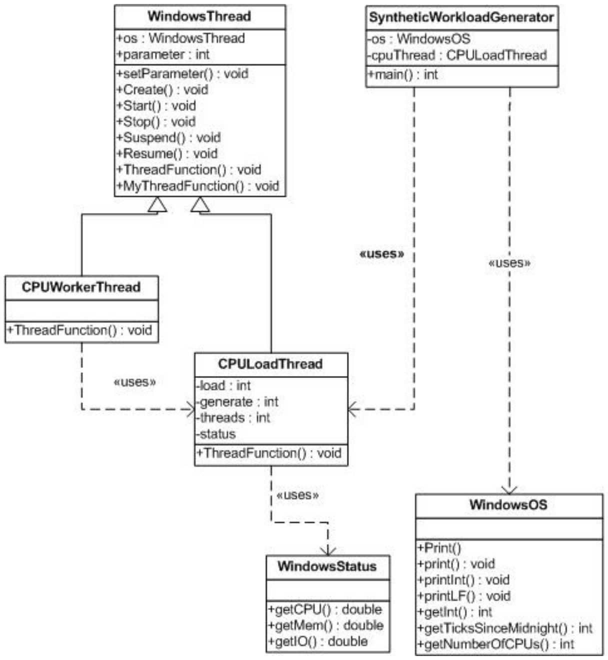


Figure 16 : Class Diagram of Synthetic Workload Generator

To extend this program, classes like MemoryLoadThread, IOLoadThread, and NetworkLoadThread are also derived from WindowsThread. They are not included in the class diagram in figure 16. The workload generator would then generate loads for different resources on the system.

6.2.2 Hardware Profiling

The methodology proposed in this report requires hardware profiling. Hardware profiling involves building a profile of the power consumption of the machine. The power consumed by the machine is the sum of the power consumed by all the subsystems the machine consists of. If we measure the resource utilization of the machine, we can predict the power based on this resource utilization. The hardware profiling is a means of predicting the power the machine will consume under certain resource utilization levels. Having such information is important because it helps us play “what if” games and use this information to do a better allocation of resources, by keeping the power consumption in mind.

Different applications have different Service Level Agreements (SLAs) to meet. A particular request can be assigned different amounts of resources and it will be able to execute in different amounts of time. Thus by varying the allocation of resources to a request, we can meet, exceed or fail to meet the SLAs. Application profiles and hardware profiles together will help us have a model in which we can see the relationship between SLAs and power consumption. Figure 17 shows the relationship of SLAs to Resource Utilization and the relationship of resource utilization to power consumption.

We can collect power consumption at various resource utilization levels, and correlate this utilization to the power consumption. Finding this relationship involves learning the power characteristics of the machine, and we call this learning the hardware power profile of the machine. The learning can be carried out by a neural network, or we can use regression techniques to learn the correlation.

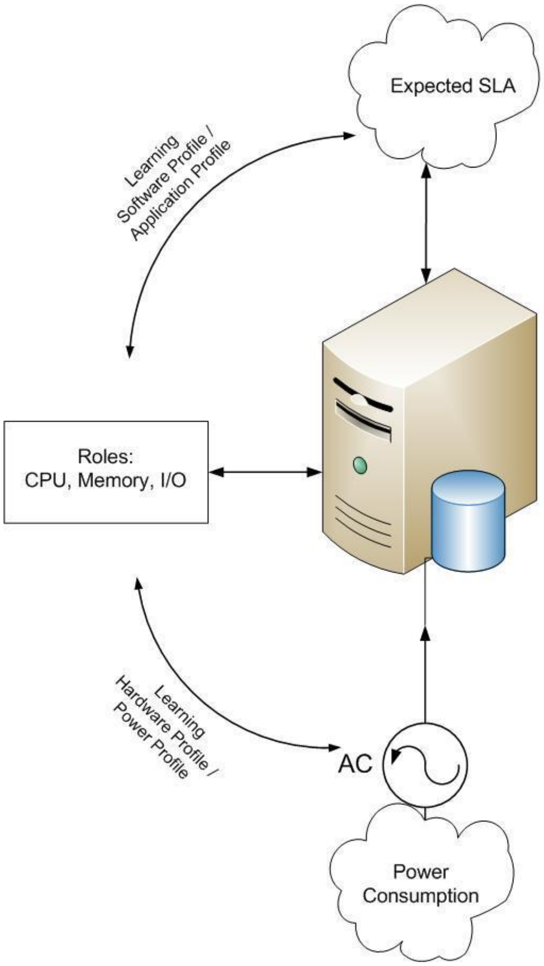


Figure 17 : Correlation between SLA and power consumption

In either case, to make sure we have a good data set, we need to ensure that the machine is loaded at different levels and this utilization spans across the entire spectrum of combinations of utilization of resources. Also we need to load the resources at different levels and learn how much power is consumed at each level. The synthetic workload generator described in section 6.2.1 can be used to generate different amounts of load, and load the machine at different levels. Figure 18 depicts this process where a learning model takes the resource utilization and the power consumption as inputs and learns the profile of the machine while the synthetic workload generator is generating various loads.

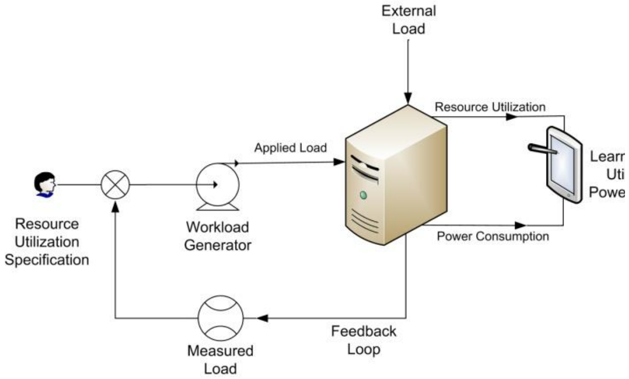


Figure 18 : Synthetic workload generator generating load and a learning model learning the correlation between resource utilization and power consumption

The correlation between CPU and power seems to be linear, as least for the machine used here, as can be seen in Figure 19.

The neural network produced a fairly accurate model of the system, but the regression model had a lower root mean square error, so it turns out to be a better model. We thus decided to go with regression instead of neural networks.

6.2.3 Power Predictor

The power predictor is a program that takes resource utilization as input and gives the predicted power consumption as output.

The implementation of the power predictor depends on the method used for learning the power profile.

When we used the neural network toolbox from Matlab to learn the power profile, the simulate function of the neural network could be used to predict the power, or the simulate function could be implemented as part of a program. But we chose regression which produces coefficients to multiply with each resource utilization value and it gives the power consumption as an output. This is a simpler implementation of the power predictor.

The power predictor was implemented and run in real time. The expected values closely matched the actual measures taken by a wattmeter connected to the machine. The Figure 19 below, shows the power consumption of a machine by varying the CPU utilization between 0% an 100%.

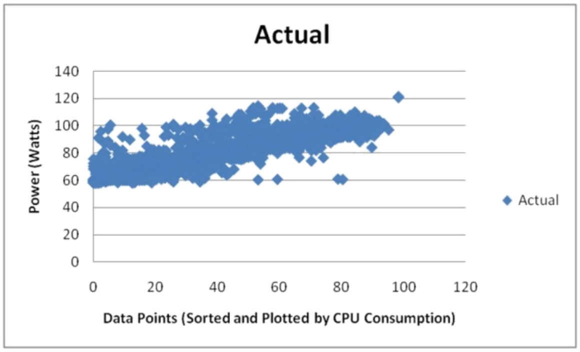


Figure 19: Power consumption at different CPU utilization levels measured at 1 second intervals

If we use only the CPU utilization values and perform regression on the data, we can come up with a linear equation that shows the relationship between CPU utilization and power. This is shown in Figure 20.

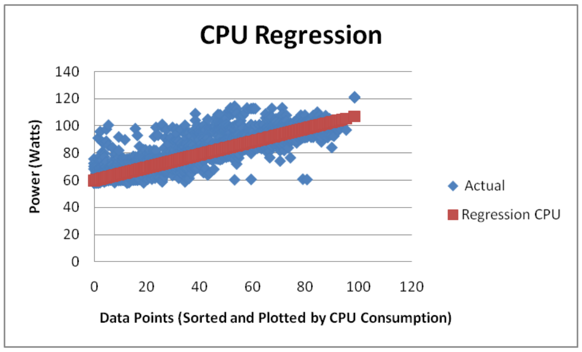


Figure 20: Result of regression on the data using power consumption statistics

Adding other metrics like Memory utilization and I/O utilization, the regression can give an even better prediction of the power. This is shown in Figure 21.

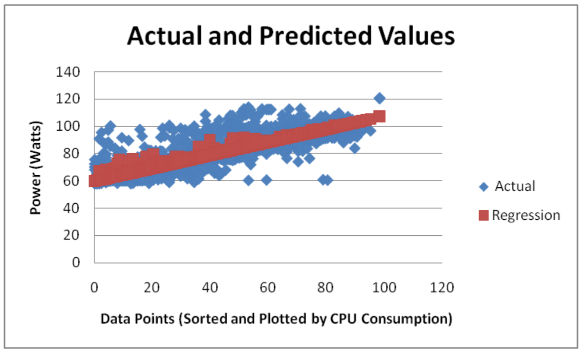


Figure 21: Result of regression using CPU, Memory and I/O data

6.3 Resource Profiling of Applications

There are various tools that can be used to collect resource utilization data. Windows has performance monitor counters that can be registered with the OS, and logs could be collected. HP has a tool called SiteScope, which collects resource utilization data from the machines in the data center.

This information can be studied to find patterns in the usage of power. Once the patterns are learned, we can use the information to make better allocation of resources.

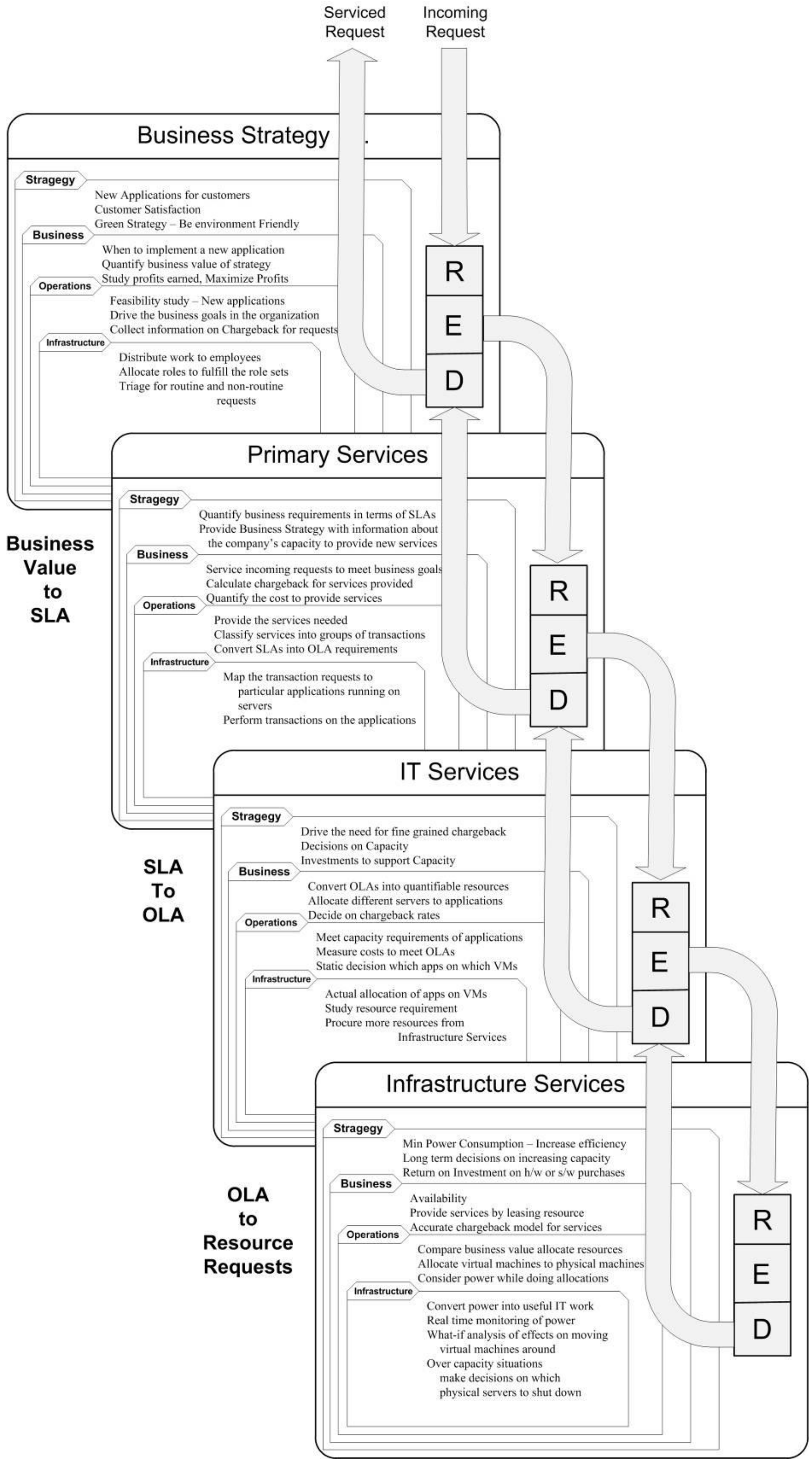
6.4 Studying incoming requests and their use of resources

By using RED (Request-Execution-Delivery) transactions, as posited by the ACE model [19], we also have a way to correlate business value generated by a transaction with the SLAs for the applications that perform the transaction, as well as with the power needed to perform the transaction. Performance of a RED transaction may involve various other RED transactions as sub-transactions. This view of how requests are serviced in the enterprise allows a much more precise characterization of the component services that combine to generate value for the enterprise, and how they combine. Since our hardware and application power profiles allow resource utilization to be correlated with power, we can also determine how much power was used to service a request, so that the enterprise can examine how much value was generated for the power expended. Since applications that service requests also have associated SLAs, the tradeoff between value generation, SLA and power can be assessed, and modified if necessary. For example, managers may decide that a particular application’s SLAs should be lowered, because the transactions it performs are not generating enough value to justify the cost of higher performance.

The data center has to be modeled as a whole, if we want to be able to optimize it. This section describes the data center where the business value of the incoming requests is traced to the applications servicing these requests. The business value is traced further down the chain to the physical machines and the consumption of electricity and cooling power. This tracing thus allows the enterprise to determine how the business value generated by servicing of the request is correlated with the power resources needed to service the request.

Data centers house many servers which generate revenue by servicing incoming requests. These requests can be accesses to a website, a request for a quote, or any other type of computing service. There is value generated when these requests are serviced, and this value generates revenue for the company. The enterprise architecture model described in this section aims to trace the value of these transactions down to their power consumption. This is an attempt to have better traceability of the costs associated with the application, and find out the relationship between business value, the service level agreements, the operating level agreements and the actual costs for the physical machines, the power consumption, and other utilities such as cooling.

We use the adaptive complex enterprise (ACE) model, where every transaction is broken down into Requests, Execution and Delivery (RED) Transactions. The organization is interested in servicing customer requests. Business value is derived by servicing requests from the customer. The global power manager needs to have information on which requests are more important than others, and when it makes a tradeoff between performance and efficiency, it needs to have this information available in a form that it can use. The business value must be quantified and converted into something that can be compared with power. We can begin this process by quantifying the business value for different kinds of transactions.



Serviced Request
Incoming Request
Business Strategy
Stragegy
New Applications for customers
Customer Satisfaction
Green Strategy – Be environment Friendly
Business
When to implement a new application
Quantify business value of strategy
Study profits earned, Maximize Profits
Operations
Feasibility study – New applications
Drive the business goals in the organization
Collect information on Chargeback for requests
Infrastructure
Distribute work to employees
Allocate roles to fulfill the role sets
Triage for routine and non-routine requests
R
E
D
Business Value to SLA
Primary Services
Stragegy
Quantify business requirements in terms of SLAs
Provide Business Strategy with information about the company's capacity to provide new services
Business
Service incoming requests to meet business goals
Calculate chargeback for services provided
Quantify the cost to provide services
Operations
Provide the services needed
Classify services into groups of transactions
Convert SLAs into OLA requirements
Infrastructure
Map the transaction requests to particular applications running on servers
Perform transactions on the applications
R
E
D
SLA To OLA
IT Services
Stragegy
Drive the need for fine grained chargeback
Decisions on Capacity
Investments to support Capacity
Business
Convert OLAs into quantifiable resources
Allocate different servers to applications
Decide on chargeback rates
Operations
Meet capacity requirements of applications
Measure costs to meet OLAs
Static decision which apps on which VMs
Infrastructure
Actual allocation of apps on VMs
Study resource requirement
Procure more resources from Infrastructure Services
R
E
D
OLA to Resource Requests
Infrastructure Services
Stragegy
Min Power Consumption – Increase efficiency
Long term decisions on increasing capacity
Return on Investment on h/w or s/w purchases
Business
Availability
Provide services by leasing resource
Accurate chargeback model for services
Operations
Compare business value allocate resources
Allocate virtual machines to physical machines
Consider power while doing allocations
Infrastructure
Convert power into useful IT work
Real time monitoring of power
What-if analysis of effects on moving virtual machines around
Over capacity situations make decisions on which physical servers to shut down
R
E
D

Figure 22: Treating each level as a business unit

Today's dynamic enterprise operates in a competitive environment, and provides customized solutions to its customers. This involves servicing various non-routine requests. The non-routine requests cannot have a fixed cost associated with them, due to their non-routine nature. To understand the cost of a non-routine request, we need to have granular measurements of the resources needed to fulfill such requests.

Non-Routine requests require a better model that tracks the actual cost of servicing the request. In addition to having continuous improvement, we need to locate areas for improvement and work towards increasing operational efficiency and decreasing costs. Currently businesses do not treat IT as a separate business unit, and even if they do, a fixed price is allocated for IT services. Dividing the business into different levels can help increase efficiency, and provide a better model that reflects the real costs associated with IT services.

We need to treat each level as a business and each level has to track its costs independently. This will help us to have a more granular view of the associated costs, and also will create an atmosphere where each unit can optimize its operations and decrease cost. Figure 22: Treating each level as a business unit, divides the organization into four levels. The incoming request passes through each level, and at each level it requests services from other levels. The incoming request to each level is treated as a RED transaction (which is explained in section 6.4.2), and is broken down into sub-transactions which are serviced by the lower layers. This allows us to keep track of the cost of each sub-transaction and also quantify the business value of each transaction.

Treating each level as a business unit provides many benefits. We can allocate capacity according to the business value and justify the cost for adding new capacity. The cost for providing the service can be accurately tracked and we can identify areas for improvement, and then do a cost-benefit analysis before proceeding to make any changes to the process or organization. Each business unit will have to manage its own assets and costs, and is forced to create profits. This encourages continuous improvement and makes the business units more efficient. Each level can divide the capacity into Request-servicing capacity, Infrastructure service capacity, Operational interaction, and Operational capacity.

The chargeback model has other benefits that include:

- Matching of services to business need
- Forcing the organization to control demand – control expenses and decrease costs
- Highlight areas of service provision that are not cost-effective
- Disciplining business units by providing a fair consumption-based chargeback model - nothing is free, or has a fixed charge
- Ensuring better alignment with enterprise goals
- Pinpointing areas of innovation – by finding areas that are inefficient

This process is described in detail in this section. The description covers how Business value is converted into SLAs and how SLAs are converted into OLAs. OLAs in turn are converted into resource requests. Resource requests can then provide an accurate estimate of the costs. Examples of services that can be quantified are power, cooling, hardware, software and maintenance.

6.4.1 Business IT Alignment

Now that we have divided the organization into various businesses, we have a better chargeback model, and we can quantify the cost of providing services. Quantifying costs and having a traceable workflow is the first step to having a process that incorporates continuous improvement.

At each stage of improvement, we identify areas of opportunity, where we can improve the process and reduce operational costs. The Cost-Benefit Analysis Method (CBAM) specifies how to do this [22]. The different stakeholders in each sub-business consider the performance and availability of the system to identify inefficient points, and increase the efficiency. While doing this, the security of the system needs to be taken into consideration. Also, the system should be built in a way that is easy to modify. CBAM helps stakeholders reflect upon and choose among the potential architectural alternatives.

Organizations need to know how to invest their resources to maximize their gains, meet their schedules and minimize their risks. The Architecture Tradeoff Analysis Method provides a method for understanding the technical tradeoffs that must be made as design and maintenance decisions are made.

Another important component is the Return On Investment (ROI); every architectural decision must take long-term costs into account. Quantifying the ROI can help businesses make informed choices that would have an impact on hardware buying decisions, and other decisions that can influence data center operations. These architectural decisions need to be linked to the business goals and quality attributes.

6.4.2 RED Transaction Model

We propose a QoS (Quality of Service) predictor which uses complex parameters to predict the QoS. The aim is to be able to reduce operating costs while maintaining the minimum QoS. It is another way to make sure the SLA is maintained.

The Adaptive Complex Enterprise (ACE) Framework is used to model the system, and describe the correlation between the operating level metrics and the BioS Goals. BioS stands for Business Infrastructure Operations Strategy, and represent the different stakeholders in the enterprise. BioS goals are the combined view of the goals of the different stakeholders. The customer requests are serviced while fulfilling these BioS Goals.

In our case, the data center IT power consumption can be divided into the power consumption of the various applications housed in the data center. This in turn, can be divided into the power consumption of the servers on which the application runs.

The power consumption of each server can be divided into the power consumption of various sub-systems that make up the server. Thus we can think of the power consumption of the server as the sum of the power consumption of the various resources it uses. In other words, there is a correlation between the resource utilization and power consumption. The resource utilization data in an enterprise is usually collected for capacity management reasons, and most enterprises have a record of this data.

Using the ACE Framework, we break up transactions/interactions of the system into subparts, called RED Transactions. RED stands for Request, Execution and Delivery.

Figure 23 describes an interaction in ACE, and shows how the incoming request is executed and the service delivered. The BioS Goals have to be met as the RED Transaction takes place. BioS Goals are:

B – Lower power consumption per transaction, i.e., decrease operating cost
I – Choose the right servers or resources
O – Migrate virtual servers for load balancing, or in the case of physical servers, find the right mix of applications to run on them such that the cost is minimized
S – Promote customer satisfaction by meeting SLAs, promoting a greener tomorrow.

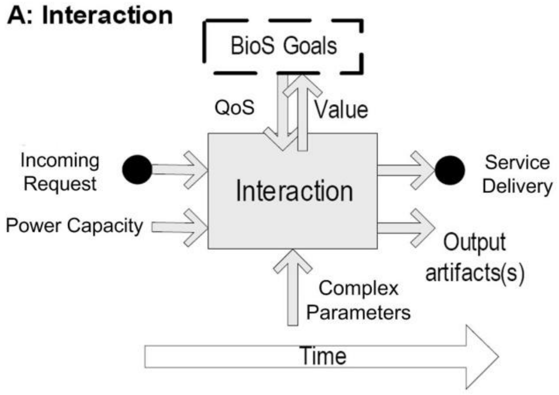


Figure 23: Interaction

The overall goal is to transform the input request into the desired output while maintaining the required QoS. The QoS is a metric to account for fulfillment of the business goals.

We can think of the interaction as a user visiting a website and performing a transaction, such as buying a book from an online store.

The next diagram, Figure 24, depicts how a customer request event moves from one interaction to another. Modeling the system in this way allows us to delay the assignment of roles until the interaction is itself ready to execute. This is called delayed binding. This allows interactions to execute while allowing the virtual machines to be dynamically moved across physical servers.

By delaying allocation of roles, we can make the system respond to non–routine requests more efficiently, by allocating the most efficient combination at that instant.

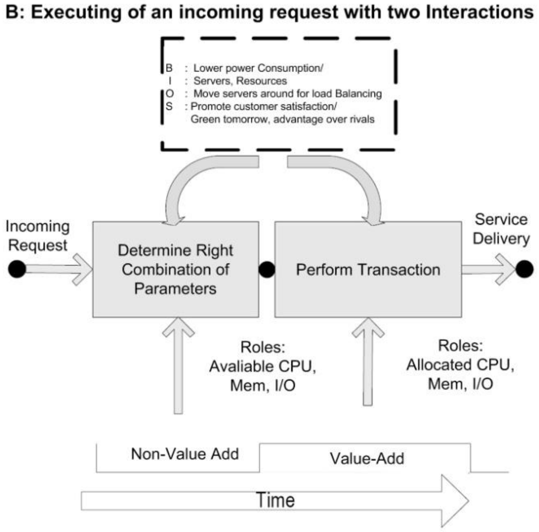

B: Executing of an incoming request with two Interactions
B : Lower power Consumption/
I : Servers, Resources
O : Move servers around for load Balancing
S : Promote customer satisfaction/
Green tomorrow, advantage over rivals
Incoming Request
Determine Right Combination of Parameters
Perform Transaction
Service Delivery
Roles: Avaliable CPU, Mem, I/O
Roles: Allocated CPU, Mem, I/O
Non-Value Add
Value-Add
Time

Figure 24: Executing an incoming request with two interactions

Another benefit of modeling the system in this manner is that it enables us to identify the Value-Add and Non-Value Add time from the customer or request originator’s perspective. The customer simply wants the outcome of the transaction (the service requested), and is generally not concerned with the Non-Value Add tasks performed, like the website keeping track of its inventory, its budget, the shipping options, the price fluctuation of its suppliers, etc. The user also does not want to see delays due to the sub transactions such as the database query, perhaps a configurator running in the background, or a system that is taking time to calculate the power consumption and deciding on the most power efficient way of servicing the request.

Figure 25 shows how we can overlap the execution of interactions to reduce the Non-Value Add time. Agility is achieved here through improved communication and coordination between the different roles.

Customers initiate an Interaction by a Request event as illustrated by the ‘black dot’ in Figure 23, Figure 24 and Figure 25

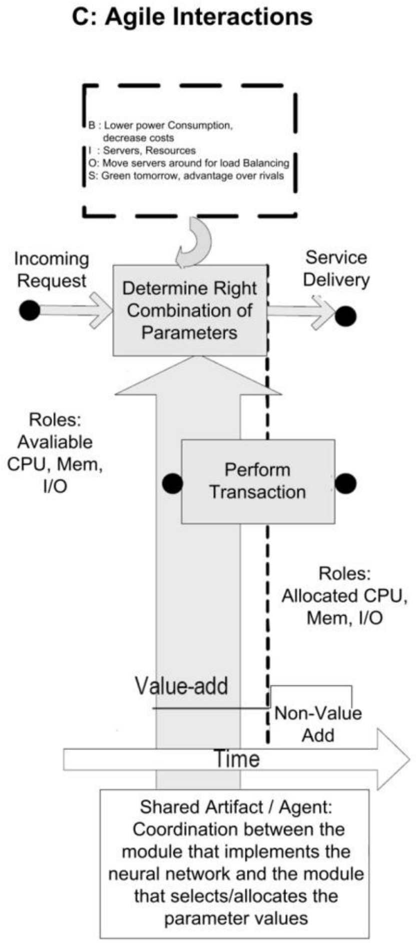

C: Agile Interactions
B : Lower power Consumption,
decrease costs
I : Servers, Resources
O: Move servers around for load Balancing
S: Green tomorrow, advantage over rivals
Incoming Request
Determine Right Combination of Parameters
Service Delivery
Roles: Avaliable CPU, Mem, I/O
Perform Transaction
Roles: Allocated CPU, Mem, I/O
Value-add
Non-Value Add
Time
Shared Artifact / Agent: Coordination between the module that implements the neural network and the module that selects/allocates the parameter values

Figure 25: Agile Interactions

The request then moves through the Requirements, Execution and Delivery stages – reaching the Agreement, Customer Business, and Provider Business milestones respectively.

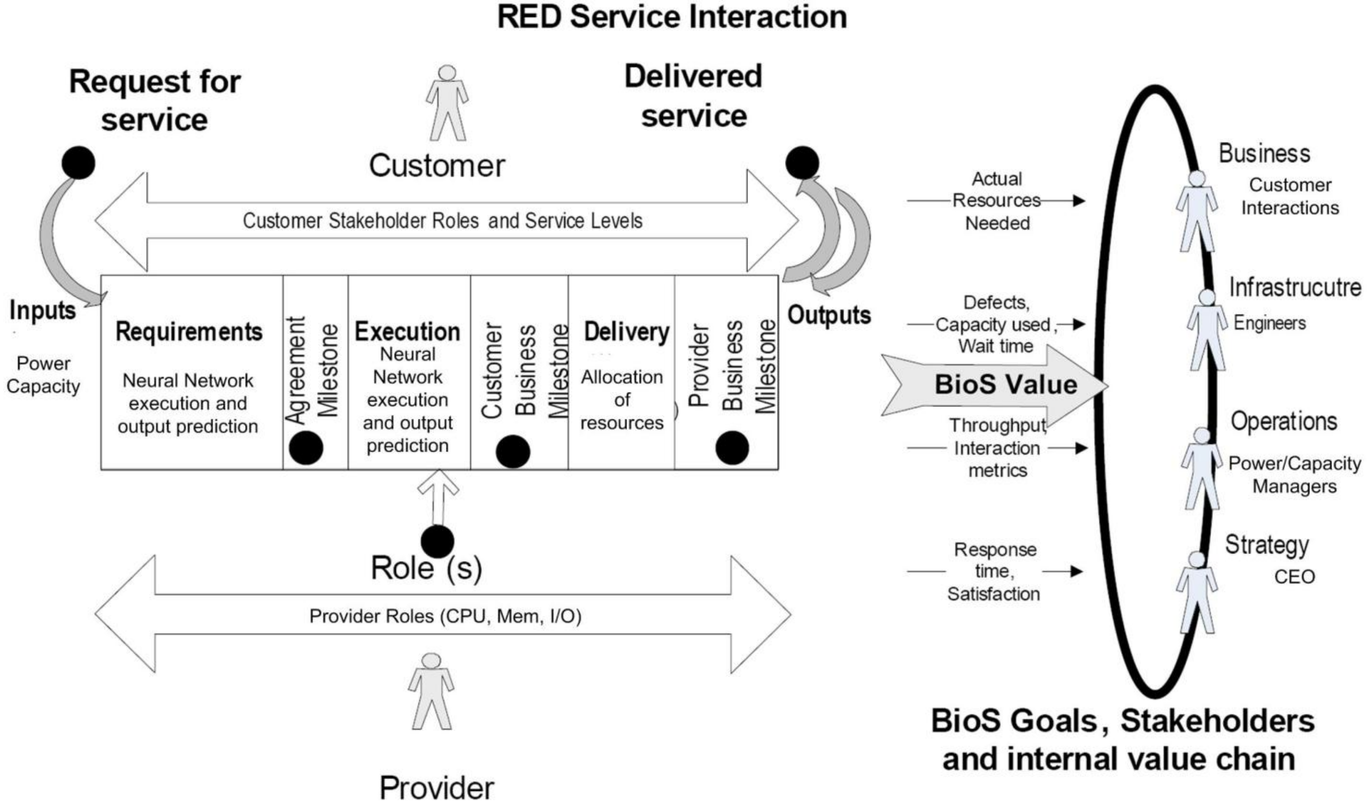

RED Service Interaction
Request for service
Customer
Delivered service
Customer Stakeholder Roles and Service Levels
Inputs
Power Capacity
Requirements
Neural Network execution and output prediction
Agreement Milestone
Execution
Neural Network execution and output prediction
Customer Business Milestone
Delivery
Allocation of resources
Provider Business Milestone
Outputs
Role (s)
Provider Roles (CPU, Mem, I/O)
Provider
Actual Resources Needed
Defects, Capacity used, Wait time
BioS Value
Throughput Interaction metrics
Response time, Satisfaction
Business
Customer Interactions
Infrastrucutre
Engineers
Operations
Power/Capacity Managers
Strategy
CEO
BioS Goals, Stakeholders and internal value chain

Figure 26: Red Service Interactions with BioS Values

The objective of the RED Interaction structure is to make explicit important transitions from all perspectives and points of metrics collection. The perspectives are:

- Customer perspective
- Conceptual RED Interaction and performance perspective
- Provider perspective

RED Transactions can be broken down into smaller sub-transactions. The Requirements, Execution and Delivery may each be considered as a RED transaction of its own. Coming back to our earlier example of a user buying a book at an online store, this transaction can be broken down into creating a session, doing the interaction of buying the book, and having it shipped to an address. These RED interactions can be broken down further into RED transactions on the different applications performing these actions. We can take it a step further by breaking these transactions down into http requests, database queries and other external transactions with external systems such as credit card companies and shipping companies. At every step, we collect metrics that correspond to the amount of work done, and relate this to the BioS goals. We thus have a system where the business value can be traced down to the individual agents working in the enterprise.

For consolidated and virtualized systems, we can divide the transactions into transactions on individual applications on particular virtual machines. Let's look at this process in detail with the help of an example. Let's consider a database server. The server has to meet a particular SLA; in our example of the database server, it could be a certain amount of time within which to accomplish each database transaction. As long as we complete the transaction in no more than the maximum time allowed, we meet the SLA. We can execute the transaction faster, but it does not, strictly speaking, add business value – we just need to meet the SLA, not exceed it.

We have now described our problem in terms of a single server and have to provide the infrastructure and operational managers with a system that helps them make decisions. At the same time, the overall process of collecting and merging this information, which is described later in the paper, can be used to consolidate the information gained and provide the business and other stakeholders with better metrics to make strategic decisions.

6.4.3 Defining an SLA curve

To meet this transaction processing time, we require a certain amount of CPU, memory and I/O. The power consumption of the server can be thought of as the power consumption of these individual components. Servers are designed to be efficient at certain combinations of these parameters. We therefore have two sets of learning to do, before we can establish the correlation between power consumption and the business value generated.

As seen in the diagram, we have the software/application profiling – where we are learning how the SLAs of that particular application can be met by various combinations of utilization of the resources. We also need hardware profiling – which is learning how the resource utilization affects the power consumption of that particular hardware.

The question that remains is how we relate the information learned from the hardware and software profiling, and use it to decrease power consumption. The following diagram describes that methodology.

In Figure 17, the work is divided into two RED transactions – one from the customer's perspective and the other from the provider's perspective. The customer is concerned with obtaining his requested service in a minimum amount of time. As long as the SLA is met, the customer is happy. A faster response, though, would improve the customer experience.

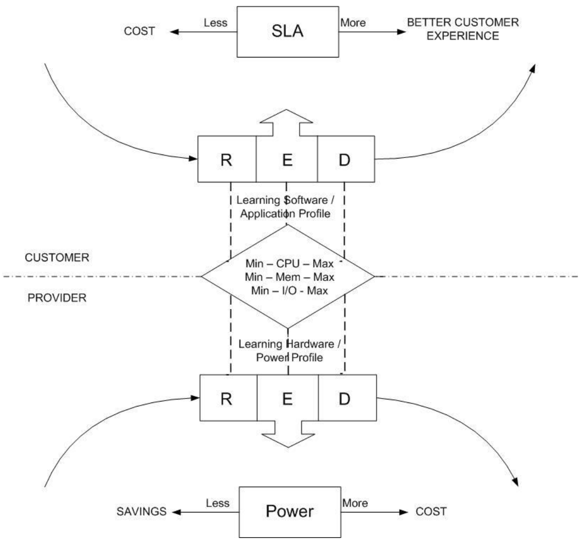


Figure 27: SLA and Power relationship using two RED Transactions

The provider has the task of providing the infrastructure for the customer. He has various ways of satisfying the requirement by designing the system to provide different resource utilizations – each having a different impact on the power consumption. Lower power consumption means increased savings in power and cooling costs; however, increased resource utilization would mean increased usage of power. In addition to increased cost, there is also a physical limit to the amount of resources that can be provided, because of hardware and power capacity constraints.

With these two RED transactions in place, the problem now boils down to selecting the right combination of resources to meet the SLAs while minimizing power consumption. This can be done by the infrastructure and operational managers.

The business and strategic managers can look at this system of metrics and play "what if" games such as reducing the power consumption to a point where the SLAs are not met, and study how much it costs them when they do not meet the SLAs and quantify the savings achieved through a reduction in power consumption. If the SLA is very tight, or overly stringent, then a slight decrease in the SLA might not cost much, but could result in huge energy and cooling savings. This system provides the stakeholders with tools to make intelligent, well-informed choices and would result in significant savings.

6.4.4 OLA for resource

To perform optimization what we need is not a single SLA value, but a curve that displays the relationship between SLA and business value. Having such a curve will help us understand the tradeoffs associated with lowering the SLA and how this effects business value. It will also help us justify the choice of the selected SLA.

Some useful curves would be Business Value vs. SLA and Business Value vs. Performance.

In our model of continuous improvement, we have to continuously try to improve the SLA, but by keeping the costs in mind, and making sure it adds business value. If the required SLA is over-estimated, and the cost of providing that high SLA is not justified, we would need to reduce the SLA and cut costs, especially when the cost involved could be detrimental to the environment, for example, the case of having a larger carbon footprint.

6.4.4.1 Conversion of SLA into OLA Units

The SLA provides a metric which represents the value delivered to the customer. It could be for example, the response time of the company website. This is useful to quantify the satisfaction of the customer, and to achieve business goals. However, to optimize operations, we need to have the SLA translated into a different value called the Operating Level Agreement (OLA). The OLA corresponding to the previous example would be the availability of the system, or some performance metric. Translating the SLA into the OLA is an important step in quantifying and justifying the OLAs in the organization too, since we can now have a mapping of the business value to an OLA, and argue whether the operational costs are justified.

6.4.4.2 Conversion of OLA into Role Sets

The infrastructure departments of the organization are involved in allocating resources to the other business units, and charging those units for the services. Resources include pieces of hardware like servers, and other utilities like power and cooling.

The infrastructure business unit can measure power consumption and cooling costs associated with providing a service and try to increase their efficiency. Inefficient servers can be replaced by efficient ones, and operational costs can be reduced. Sometimes saving power involves an associated performance loss. High business value applications could be impacted by such losses. To make the most cost-effective decisions, we need a mapping of the OLAs to role sets. Role sets enumerate the resources needed to fulfill the OLA. The resources could be the amount of computing power needed, the amount of bandwidth needed, or other such metrics. Having the OLA–Role Set mapping allows us to quantify the business value associated with the roles, and push for continuous improvement in the infrastructure business unit.

6.4.5 Normalization of resource utilization units

Although our approach offers many advantages, as explained above, there is also a complication which must be overcome for it to work well. Specifically, the resource utilization units on different hardware will not be uniform, and therefore must be normalized, so that the management system will have a consistent unit for decision making. For example, if an application’s power profile gives its average CPU utilization as 8% on one piece of hardware, it may be a significantly different value on different hardware with a different clock speed, instruction pipelining design, or instruction set architecture. Although we have not yet completed work on the normalization process, it is clear that it can be done using statistical techniques.

6.4.6 Improved Chargeback Model

Since our approach allows tracing of power use in the data center, and tracing of request servicing through RED transactions, it also offers the benefit of an improved chargeback model. A general difficulty with chargeback is that it is difficult to measure how much capacity or how much of a particular resource a business unit is using, and therefore, it is difficult to do chargeback in a way that accurately reflects this use. The problems created for the enterprise include the occurrence of free riding by some business units, which use more resources than they are charged for. By improving chargeback, free riding can be minimized or eliminated, and this results in incentives for all business units to consume only the resources they need.

6.5 Resource Utilization into costs

We have now gone through the process of converting incoming requests to SLA requirements, SLA requirements to OLA requirements, Converted OLA into resource utilization, and we are now at a point where we have to convert resource utilization into costs.

The major cost associated with resources is power consumption. This is addressed in section 6.5.1. Section 6.5.2 describes the other costs associated with servicing requests. Section 6.5.3 describes the side effects and constraints that contribute to the costs.

6.5.1 Power along with other ROI stuff

The resources in the system have costs associated with them. These costs can be procurement costs, operational costs, and maintenance costs. The enterprise has many shared resources and this makes it very difficult to quantify the business value of these resources.

Role sets are groups of resources needed to perform a task. Using role sets allows us to model the shared resources phenomenon. A single resource can be a part of more than one role set. By quantifying the time, or percentage consumption of a particular resource, we can divide the costs incurred in supporting that resource. The costs can then be compared with the business value provided by the role set, and thus the business value associated with a resource can be quantified.

This provides a good model of the costs associated with the resource, and can be used as an input for continuous improvement.

6.5.2 Other Costs

The chargeback model explained in section 6.4 provides a means for capacity management while maintaining flexibility. Resources are not tightly coupled with the applications, but are provided using various role sets. The chargeback model also takes power consumption into consideration. Quantifying costs of resources allows us to improve throughput and reduce costs.

There are also savings associated with licensing of applications. When we consolidate applications, there is an opportunity to use fewer licenses, and this could also impact the cost.

Other costs to be included in the chargeback model include the fixed costs of the facility, and application licensing costs. The costs of the server hardware also have to be accounted for, and corresponding studies on the ROI need to be done. Having the business value traced to the lower operational levels allows us to quantify the value generated by each resource and prevents the practice of charging the same cost to all the users of the infrastructure. The overhead costs should not be charged evenly, but should be imposed according to utilization of the resource.

An additional benefit that impacts customer relationships is making the enterprise environmentally responsible, by measuring its carbon footprint, and decreasing power consumption. To improve energy efficiency and promote competition, it would be of great value if the organization could track power consumption and move this information up the value chain and deliver this information to customers along with the response to the incoming request. One way of doing this is using a metric for energy use. For example, we might convert the carbon footprint associated with servicing the request to a value like number of leaves – the equivalent carbon footprint of a leaf. This idea could be extended to larger groups of transactions also, (100,000 leaves = 1 Tree, 100,000 Trees = 1 Forest, and so on). Some load could be reduced by making customers aware of the amount of energy being consumed when they visit a website, or use some other computing resources. Tracing metrics like this can increase customer loyalty and promote better customer relationships.

Another issue to be studied is the cost of implementing this kind of traceability. Specifically, studies of the cost of traceability versus the benefits which can be gained from it are needed.

6.5.3 Side effects and Constraints

There are some costs that cannot be related to any particular business application. We call such costs side effects. Examples of side effects include the heat generated by systems and the cost involved with cooling. Side effects are unwanted outputs of the system, which cannot be prevented. One way to trace this cost is to correlate it with the resource utilization and include it as a resource utilization cost. Another way of modeling this cost is to consider side effects as requests generated by the system, and then there is a cost associated with fulfilling this request.

There are also some constraints that prevent the application of this optimization approach to the entire organization. An example of such a constraint is an application that cannot run on a virtual server simply because the vendor of the application does not support the application running on virtual machines. Another example could be an operating system needed for legacy reasons that cannot be virtualized due to technical reasons. There are other such constraints that need to be modeled. One way of doing this is to make the constraints part of the role specification, and treat these roles as separate resources that cannot be shared.

7 Application of Framework

The adoption of the complex framework described in the previous sections requires a number of changes to the existing organization. Building the framework from scratch is easier to implement, but today's enterprise poses a challenge where we need to add traceability to the existing systems, and monitor power for the existing physical infrastructure. We therefore need a methodology with a step by step approach. Section 7.1.4 describes the different metrics that are accepted industry standards. At each stage of the continuous improvement methodology, we come back to these standard metrics to check if the overall efficiency of the data center is increasing. Section 7.2 specifies areas of opportunity, where quick changes to the infrastructure can result in sizable increases in power efficiency. Section 7.1 discusses a continuous improvement methodology.

7.1 Integrated Continuous Improvement

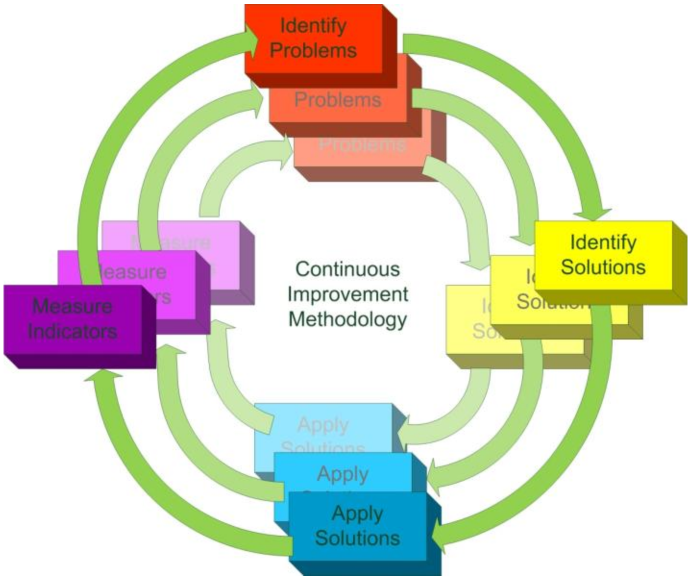
This section describes the various steps in the Integrated Continuous Improvement Methodology.

7.1.1 Identify Problems

The first stage in the continuous improvement methodology is identifying problems. Problems are opportunities to increase efficiency and decrease costs. We start with the problems that give us the greatest gain when solved. The problems can be divided into two types - problems with hardware and problems with software. Section 7.2.1.1and 7.2.1.2 discusses these in detail.

7.1.2 Identify Solutions

If the change involves adding new hardware, we must learn the profiles of the new hardware. A



detailed methodology for doing this is explained in section 6.2.1 and section 6.2.2. The result of this profiling is a power predictor, which can predict the power consumption for any resource request mix. The Section 6.2.3 talks about the power profiler in more detail.

After completing the RED analysis, we build an application profile using the resource utilization logs. The methodology for performing this application profiling is described in Section 6.4.2.

Now that we have the application profiles, and know the resource usage patterns for each application, we can decide on an optimal mix of servers that can be consolidated. This process can be automated, and different algorithms can be used, for example, a bin-packing or heuristic algorithm.

7.1.3 Apply Solutions

After identifying the solution, allocate people and time to apply the solution. The solution could be a simple measure like enabling DVFS, or it could be something as complex as implementing traceability.

7.1.4 Measure Indicators

We need to make sure we are making progress at every iteration of the continuous improvement cycle. If we do not, it could mean that the power saving measures that were just introduced could be interfering with other power saving measures that were already implemented

At every stage in the continuous improvement model, we use industry standard metrics to verify if the data center operation is increasing in efficiency. We use the following Green Grid metrics to help us validate the changes we make at each level. PUE stands for Power Usage Effectiveness and DCIE stands for Data Center Infrastructure Efficiency.

$$\text{PUE} = \frac{\text{Total Facility Power}}{\text{Total IT Power}}$$

$$\text{DCIE} = \frac{1}{\text{PUE}} = \frac{\text{IT Equipment Power}}{\text{Total Facility Power}} \text{ x } 100\%$$

Power profiling of machines eliminates the need to have granular measurements of power. The power consumption can be predicted based on the hardware power profile and resource utilization. However, periodically we need to measure power to make sure that our predictions are right and that there is an actual increase in efficiency in the data center.

The continuous optimization process will consist of continually increasing the granularity at which the improvement cycle is done. Initially, the judgments and evaluation required can be undertaken by subject matter experts in the relevant areas. Next the hardware profiling, application profiling, and power predictor components can be done only for hardware and applications which are believed to consume a lot of power in the data center. As more and more hardware and applications are profiled, the dynamic management capabilities could be implemented partially in the data center, again beginning with large power consumers, and on each subsequent improvement cycle, more and more hardware and applications can be added. This approach spreads the implementation work out over time, but allows the enterprise to reap significant benefits on each optimization cycle.

7.2 Steps to Realize Opportunities

This section provides guidelines on how to prioritize the changes needed in the enterprise and how to get the maximum benefits out of each iteration.

7.2.1 Initially - Static Optimization

To begin with, we look at areas of opportunity where simple measures can lead to significant energy savings. Doing simple things can save a lot of energy and can reduce running costs. Some of the things that can be done are described below.

7.2.1.1 Software Considerations - Business Value vs SLA

Figure 28 is a graph that plots the Business Value versus the Service Level Agreement associated with an application. The graph helps us to identify which applications to consolidate/virtualize first, and gives us the order in which we should consider the various applications.

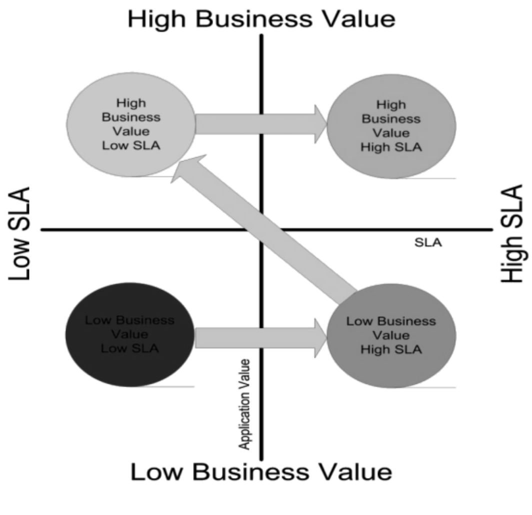

High Business Value
High Business Value Low SLA
High Business Value High SLA
Low SLA
High SLA
SLA
Low Business Value Low SLA
Low Business Value High SLA
Application Value
Low Business Value

Figure 28: Application Business Value v/s SLA

We first consider applications which have a low SLA and low business value. These applications are considered first because they have the least impact on the system – they have low SLAs - they do not demand fast response times, and secondly, they have low business value, so any impact on performance that might arise as a side effect to consolidation will be unlikely to be detrimental. Next, we tackle the applications that have a low business value, but have a high SLA. The high SLA means that we cannot consolidate too many servers together, but we can attempt to consolidate some of them. These applications might sometimes have an SLA that is too high. We can reduce the SLA and thereby reduce the cost associated with maintaining the high SLA. This can be justified because of the low business value of the application.

The next set of applications to consider are the ones that have a high business value but low SLA. These applications can be consolidated/virtualized because of their low SLAs, which allow for consolidation. Finally, we can consider the applications which have high business value, and demand high SLAs. These applications are very crucial to the organization as they generate a lot of revenue. Care should be taken when consolidating/virtualizing these applications because a drop in the SLA can have huge financial implications.

7.2.1.2 Hardware Considerations- Hw cost vs performance

We need to consider the hardware, and replace less efficient pieces of hardware with more efficient ones. Figure 29 shows the relationship between cost and capacity for different servers in the data center. The costs are the fixed costs for the physical hardware and the utility cost of running the servers.

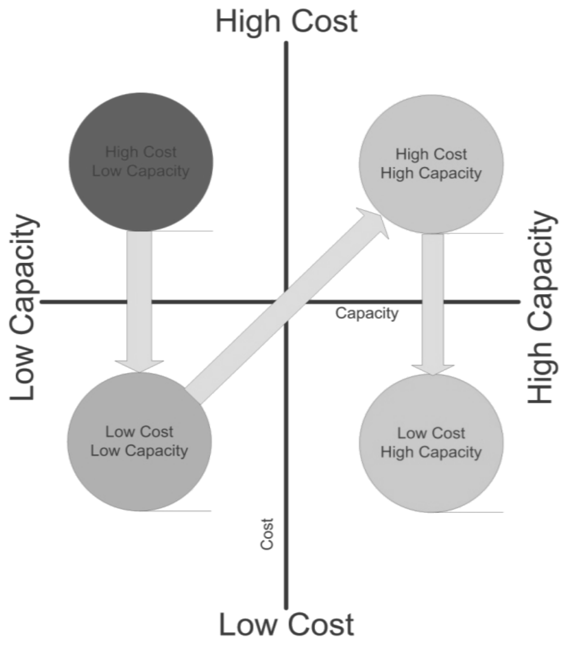


Figure 29: Hardware Cost v/s Capacity

We first identify the servers that have a high cost, but low capacity. These servers consume a lot of energy, but have a very low computing, memory or I/O capacity. These servers must be replaced with more efficient servers that provide a higher capacity for the power they consume. Next, we consider the machines that have a low cost and low capacity. These machines have low costs associated with them, and they also have a low capacity, so ideally they would be running applications that do not require high capacity. These servers could be consolidated and put into a single efficient physical server that has a higher capacity. After this we take a look at the servers that have a high cost associated with them, and demand a high capacity. These servers are usually running high business value applications that require a huge capacity. However, they have a huge cost associated with them. These servers could be running critical applications and so, the effects of changing the hardware must be carefully studied before replacing the hardware. Finally the servers which have a high capacity and low cost are considered for replacement. These servers are efficient, but they have to be refreshed periodically with more efficient machines

7.2.1.3 Allocation of Resources

Allocate resources to increase utilization and minimize consumption of power. This is a problem where we minimize the total number of physical servers running, but maximize the resource utilization of each physical machine by moving the virtual machines around.

7.2.1.4 Consolidation of Servers

Each server runs a different application. For instance, if there are multiple web applications, each one is run on a different physical server. Most of these servers run well below their capacity and if we consolidate multiple applications onto a single server, we can save a lot of power. For example, the machine profiled in Figure 19 consumes 60W at idle, and 100W at peak load. If two machines are running, each will consume at least 60W, totaling 120W. The actual consumption can be anywhere between 120W – 200W. But if we consolidate the servers their consumption will be something between 60W – 100W. This clearly is a huge savings.

The first step therefore, would be to identify servers that are relatively idle, and consolidate them into single machines. We could use the application profile to study the time at which applications consume resources and consolidate servers that use different sets of resources at different intervals in a way that the usage profiles complement each other. We could also use expert opinion to decide which servers have to be consolidated.

7.2.1.5 Replace servers with more efficient ones

Many vendors have started producing power efficient servers. Old servers consume a lot of power, and replacing these servers can save a lot of energy. The company must have a server refresh methodology that promotes the use of more power efficient servers. Servers that have dynamic voltage and frequency scaling (DVFS) must be bought, and the DVFS capabilities must be turned on. Other power saving features must be activated and the latest BIOS updates installed on the systems.

7.2.1.6 Encourage Virtualization as a means for better server consolidation

Virtualization is a technology which allows multiple virtual machines to run on a single physical server. A hypervisor runs on the hardware, and it provides an environment to run many virtual machines. Virtualization must be encouraged in the organization as virtual machines are easier to move around. Consolidation, or running many virtual machines on a single server, increases the utilization of the server and reduces running costs.

7.2.1.7 Cooling Optimizations

Cooling consumes at least half, and typically more than half of the total power consumed by the data center. Possible cooling optimizations include hot and cold aisles, where machines are positioned in a manner where there are alternating hot and cold aisles. In a cold aisle, all the fronts of the servers face each other. Air from the cold aisle is sucked into the server. Hot aisles are aisles that have backs of all the servers facing each other. The aim is to get the hot aisles as hot as possible, and the cold aisle must be as cold as possible. This increases the circulation of air, and allows for better heat transfer from the machines.

7.2.2 Demand Management of Services

Conduct a study on the applications that consume the most power and check to see if the SLA is too high. In other words, try to match the SLA to the business value.

7.2.3 Ongoing - Dynamic Optimizations

After going through all the above changes, the organization has to make sure that it is continuously improving its efficiency. It can make these decisions based on continuous measurements- as the measurements get more granular, the system will be better understood.

Another tool would be the dynamic models of power profiles that would give the organization a better understanding of the way the machine is used and how this impacts power consumption.

Finally, having a joint optimization of power and cooling could make the data center very power efficient. The aim is to put the organization in a “Continuous Improvement” frame of mind

8 Roadmap for Remaining Research

We now identify several areas where further work can be done.

• Human factors analysis: The usual assumption is that better performance by the computer system is preferred by humans using the system. At least in some kinds of cases, though, this may not be true. For example, in interactive settings, the human user can only make use of output provided by the system at a certain rate. In this case, it may not only be *sufficient* if the system response time does not exceed that at which the human user can make use of the system’s response, it may actually be *better*. If this is true, there may be cases where providing a better experience for the user actually uses less power. This area has not been studied, but may yield tremendous benefits by simultaneously saving power and increasing customer satisfaction.

• Converting more of the fixed costs to variable costs. Recently, there has been a trend in enterprises to attempt to convert fixed costs to variable costs, as much as possible. The economic rationale is that fixed costs are inefficient, because they constitute a form of “overhead,” i.e., costs which are incurred in doing business, but which cannot be attributed to any particular transaction(s). Such costs must be recovered somehow, but imposing them on customers by increasing prices may result is the impression that the customer is getting less than value than the price reflects. If these fixed or overhead costs can be converted to variable costs, which can be attributed to individual transactions, pricing becomes more transparent. An additional benefit is that the enterprise does not have to pay the costs until the transaction occurs, and so cost recovery can be faster. One way to convert fixed costs to variable costs is through cloud computing, as discussed earlier. Our approach to power modeling, traceability, and data center power management also supports this effort, because it allows a closer matching than was possible in the past of resource use to current demand. For example, in the traditional data center scheme, where applications are run on single dedicated server platforms, the server can spend significant amounts of time idling. The power that is consumed (and wasted) during these periods of idling is a kind of fixed cost, because it is not readily attributable to any particular transaction or series of transactions, at least not without accurate and granular power measurements (which, as discussed above, have not been available typically in the past). In our approach, by virtualizing and consolidating applications, so that server capacity is more fully utilized, the idle costs are greatly decreased, and most of the power cost for operating the server is directly attributable to identifiable transactions. Thus, the customers who request these transactions can be charged for the power. Further work on how the model can be used to improve this conversion of fixed costs to variable costs needs to be done, so as to maximize this benefit.

- Resource unit normalization: As discussed above, one issue to be resolved in our approach is normalization of the resource utilization units in which hardware power profiles and application power profiles are expressed. These same resource utilization units are also used to predict power. The units on different hardware will not be uniform, though, so a common or normalized unit is required so that meaningful comparisons and evaluations can be made by the power manager, and other parts of the system. One approach which is promising, but which we have just begun to explore, is using benchmarking applications, along with statistical techniques, to normalize the units. For example, a CPU benchmarking suite could be run on two different platforms, and then, through a statistical analysis of the results, a conversion factor to equate the CPU resource utilization units for the two machines could be developed.

9 References

[1] MacKinnon, C. The changing data center life cycle: one day here & the next day gone. Processor, December 22, 2006.

[2] S. Kumar, V. Talwar, P. Ranganathan R. Nathuji, and K. Schwan. M-channels and m-brokers: new abstractions for co-ordinated management in virtualized systems. In Proceedings of the Workshop on Managed Many-Core Systems (MMCS), June 2008.

[3] R. Nathuji, Mechanisms for coordinated power management with application to cooperative distributed systems. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2008.

[4] D. Narayanan. Power by proxy: a new aprpoach to measurement. August 2002. *research.cs.ncl.ac.uk/cabernet/www.laas.research.ec.org/cabernet/workshops/radicals/2003/papers/paper.pdf_5.pdf.*

[5] C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: methodology and empirical data. In Proceedings of the 36th International Symposium on Microarchitecture (MICRO-36'03), December 2003.

[7] S. Niles. Virtualization: optimized power and cooling to maximize benefits. American Power Conversion White Paper, 2008.

[8] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In Proceedings of the 18th Symposium on Operating Systems Principles (SOSP), 2001.

[9] R. Nathuji and K. Schwan. VirtualPower: coordinated power management in virtualized enterprise systems. In Proceedings of Twenty-First Symposium on Operating Systems Principles (SIGOPS), October, 2007.

[10] S. Kumar, V. Talwar, P. Ranganathan, and K. Schwan. vManage: coordinated management in virtualized systems. www.hpl.hp.com/personal/Partha_Ranganathan/papers/2008/2008_hotac_vmanage.pdf

[11] J. Stoess, C. Lang, and F. Bellosa. Energy management for hypervisor-based virtual machines. In Proceedings of the USENIX Annual Technical Conference, June 2007.

[12] R. Nathuji, C. Isci, and E. Gorbatov. Exploiting platform heterogeneity for power efficient data centers. In Proceedings of the IEEE International Conference on Autonomic Computing (ICAC), June 2007.

[13] W. Bircher and L. John. Analysis of dynamic power management on multi-core Processors. In Proceedings of the 22nd Annual International Conference on Supercomputing (ICS), June 2008.

[14] D. Suleiman, M. Ibrahim, and I. Hamarash. Dynamic voltage frequency scaling (DVFS) for microprocessors power and energy reduction. In 4th International Conference on Electrical and Electronics Engineering, December 2005.

[15] P. Kurp. Green computing: are you ready for a personal energy meter? Communications of the ACM. October 2008.

[16] R. Nathuji, A. Somani, K. Schwan and Y. Joshi. CoolIT: coordinating facility and IT management for efficient datacenters. In Proceedings of HotPower '08. December, 2008. Retrieved March 3, 2008 at http://www.usenix.org/events/hotpower08/.

[17] S. Niles. Virtualization: optimized power and cooling to maximize benefits. American Power Conversion White Paper, 2008.

[18] J. Thethi. Realizing the value proposition of cloud computing: CIO's enterprise IT strategy for cloud. Infosys White Paper, 2009.

[19] J. Ramanathan and R. Ramnath. Co-Engineering Applications and Adaptive Business Technologies in Practice Enterprise Service Ontologies, Models, and Frameworks. Igi Global, 2009.

[20] Report to congress on server and data center energy efficiency. Technical report, U.S. Environmental Protection Agency, Energy Star Program, August 2007.

[21] J. Spitaels. Dynamic power variations in data centers and network rooms. American Power Conversion White Paper, 2005.

[22] Carnegie Mellon website, http://www.sei.cmu.edu/architecture/cbam.html

[23] L. A. Barroso and U. Hölzle. The Case for Energy-Proportional Computing – Google.

[24] N. Rasmussen. An Improved Architecture for High-Efficiency, High-Density data centers. The Green Grid White Paper #126.