2011 IEEE International Parallel & Distributed Processing Symposium

Willow: A Control System for Energy and Thermal Adaptive Computing

Krishna Kant
Intel Corporation
Email: krishna.kant@intel.com

Muthukumar Murugan
University Of Minnesota
Minneapolis, USA-55414
Email: murugan@cs.umn.edu

David H.C.Du
University Of Minnesota
Minneapolis, USA-55414
Email: du@cs.umn.edu

***Abstract*—The increasing energy demand coupled with emerging sustainability concerns requires a re-examination of power/thermal issues in data centers from the perspective of short term energy deficiencies. Such energy deficient scenarios arise for a variety of reasons including variable energy supply from renewable sources and inadequate power, thermal and cooling capacities. In this paper we propose a hierarchical control scheme to adapt assignments of tasks to servers in a way that can cope with the varying energy limitations and still provide necessary QoS . The rescheduling of tasks on different servers has direct (migration related) and indirect (changed traffic patterns) network energy impacts that we also consider. We show the stability of our scheme and evaluate its performance via detailed simulations and experiments.**

I. Introduction

The rapid growth of data centers to support burgeoning online services and cloud computing has led to data centers accounting for an increasing amount of energy consumption and hence an increasing environmental footprint. This has resulted in intensive research efforts on reducing the data center power consumption at all levels from energy efficient hardware design all the way up to power, thermal and cooling management of the entire data center. Much of this research is focused on reducing the direct energy usage of the data center, whereas from an environment impact perspective one needs to consider the entire life-cycle of energy consumption – that is, the energy consumption in the manufacture, distribution, installation, operation and disposal of the entire data center infrastructure including IT assets, power distribution equipment, and cooling infrastructure.

Looking at energy consumption from this larger perspective entails not only low power consumption during operation but also "leaner" designs and operation using renewable energy as far as possible. A leaner design could take many forms including smaller (i.e., lower capacity) power supplies, heat-sinks and fans, ambient cooling which allows the elimination of chiller plants, high-temperature operation, under-engineering uninterrupted power supplies (UPS), under-designed rack power circuits, etc. All these forms of lean design increase the probability that the data center will be occasionally under-powered and thus needs mechanisms to cope with it. The power deficiency could be either real or result from the fact that limited cooling capabilities do not allow adequate dissipation of power even if the power availability is itself plentiful. The variability associated with the direct use of renewable energy could result in similar power deficiencies. The challenge is to adapt the data center operations to such available power variations – as far as possible - while still meeting the desired QoS requirements. We call this *Energy Adaptive Computing* (EAC) [1]. In this paper, we propose a hierarchical control scheme called *Willow* to cope with energy deficiencies. It is assumed here that energy deficiencies are temporary and infrequent, rather than persistent so that their occurrence does not affect the long-term performance or viability of the applications.

In general, we need adaptation to deal with both supply side and demand side variations. The supply side variations result both from the actual variations in energy supply and the variations as a result of varying partitioning of available energy among various components. The demand side variations (which themselves drive variability in partitioning) result from variations in workload intensity and characteristics. It has been noted that as the computing moves towards more real-time data mining driven answers to user queries, the demand side variations could become significantly more severe, thereby further increasing the need for adaptation to available energy.

Willow migrates the workloads away from energy deficient zones to energy surplus zones. Adaptation to the available energy profile could take many forms. In cases of serious and relatively long-lived energy deficiency, the only mechanism to cope is to shut down low - priority tasks. In less severe cases, the nature of the computation can be altered (e.g., reducing the resolution of video, use of coarser audio codecs, or computation of answers to a lower precision). Often, energy consumption can also be reduced via a latency-power tradeoff. For example, consolidation of load on fewest number of servers (without violating local power or thermal constraints) can allow others to be shut down or put in deep sleep modes. Similarly, batched processing is a well known mechanism for creating longer busy and idle intervals and thereby improving power management efficiency. Finally, local energy deficiency can be dealt with by migration of load from energy deficient areas to energy plenty areas of the data center. We allow for all of these variations in our scheme, although we do

This work was partially supported by the following NSF awards : 0934396, 0960833 and 1016350.

1530-2075 2011
U.S. Government Work Not Protected by U.S. Copyright
DOI 10.1109/IPDPS.2011.14

IEEE computer society

not explicitly consider the treatment of low - priority tasks or degraded operational modes of the applications. Instead, we focus on dynamic migrations and load consolidations in large data centers. Additionally we consider thermal limitations of individual devices when making the migration decisions.

Willow tries to adapt to the energy and thermal profile of the data center by managing the migration of tasks between servers. We named our control scheme after the Willow bird that migrates from Europe to Africa during winter and vice versa during summer in order to survive the extreme temperature conditions.

The rest of the paper is organized as follows. Section II highlights some of the related work in the area of power management in data centers. Section III explains the concept of Energy Adaptive computing and describes the relationship between energy and thermal constraints. Section IV explains the multi-level power control architecture and simultaneous energy adaptations on the supply and demand sides. Section V presents the evaluation of *Willow* by analysis and experiments. Section VI concludes the paper. Throughout the paper we use the terms energy and power interchangeably.

II. Related Work

Traditionally power control techniques [2], [3], [4] have focused on harvesting the idle periods in the workloads and either put the devices in low power modes or reduce the operation bandwidth of the components. Research works have explored the use of power control techniques in various components in the data centers, like network links [5], [6] and disks [7], [8].

Heller et al. [9] propose a dynamic change in the number of active components with changing workload patterns. The goal is to use only a required subset of network components and power down unnecessary components. Moore et al. [10] incorporate temperature profiles in data centers to make workload placement decisions. Wang et al. [11] propose an algorithm based on optimal control theory to meet with the energy and thermal constraints in chip multi-processors. Their algorithm exploits the availability of per-core DVFS in current day processors and formulates a MIMO model for multi-core processors. Anderson et al. [12] propose a cluster architecture with low-power, cheap processors and flash storage. This architecture performs best in data intensive application scenarios with small sized random accesses.

Nathuji and Schwan [13] propose a coordinated power management scheme in distributed environments with virtual machines. They leverage the guest level power management techniques and realize them in real time on the host without violating the guaranteed resource isolation between multiple guests. X. Wang and Y. Wang [14] propose a cluster level coordinated control architecture that aims at providing per-VM performance guarantees and a cluster level power control. Govindan et al. [15] present power profiles for benchmark applications. They have designed efficient power provisioning techniques based on this profiling.

While all the above techniques perform well in a situation where the available power supply is stable, these techniques do not efficiently handle situations of continuous variation in energy availability.

K. Kant [1] introduces the concept of *Energy Adaptive Computing* (EAC). Notably, power control in EAC is driven mainly by variability in energy and thermal profiles and not by the presence of idle periods. In other words, the migrations in EAC are constraint-driven. K. Kant [16] addresses the problem of coordinating power supply and demand simultaneously in hierarchical multilevel systems like data-centers. Our current work builds on [16] and presents a complete design and analysis of a control scheme (*Willow*) to achieve the supply side and demand side coordination. *Willow* considers power and thermal constraints simultaneously. We have evaluated the performance of *Willow* via real time experiments on a cluster. Also *Willow* can be seamlessly applied on top of any existing idle power control technique with slight modifications. Section III explains in detail the concept of energy and thermal adaptive computing.

III. Energy and Thermal Adaptive Computing

Energy costs constitute a significant proportion of the operational costs of a data center. The increasing carbon footprint as a result of the enormous energy that is being consumed by today's data centers is an area of growing concern. Use of alternate sources of energy like renewable forms and reducing the size of energy storage systems in these data centers can help reduce the carbon footprint greatly. But the down side of this approach is increased variability in the energy availability and more frequent episodes of energy deficiency in some parts of the data center. The sustainability concerns coupled with the need to guarantee a certain degree of service quality make energy adaptive computing an ideal strategy in data centers.

In addition to the energy availability, the thermal constraints play a significant role in workload adaptation in a data center. In this section, we discuss the coordination issues relative to thermal adaptation. To start with let us consider a power-plenty situation where only the thermal controls go into effect. Traditionally, CPUs are the only devices that have significant thermal issues to provide both thermal sensors and thermal throttling mechanisms to ensure that the temperature stays within appropriate limits. For example, the T states provided by contemporary CPUs allows introduction of dead cycles periodically in order to let the cores cool. DIMMs are also beginning to be fitted with thermal sensors along with mechanisms to reduce the heat load. With tight enclosures such as blade servers and laptop PCs, ambient cooling, and increasing power consumption, other components (e.g. switching fabrics, interconnects, shared cache, etc.) are also likely to experience thermal issues. In challenging thermal environments, a coordinated thermal management is crucial because the consequences of violating a thermal limit could be quite severe. Also, an over - throttling of power to provide a conservative temperature control could have severe performance implications.

Thermal control at the system (e.g., server, client or network/storage element) level is driven by cooling characteristics. For example, it is often observed that all servers in a rack do not receive the same degree of cooling, instead, depending on the location of cooling vents and air movement patterns, certain servers may receive better cooling than others. Most data centers are unlikely to have finer grain mechanisms (e.g., air - direction flaps) to even out the cooling effectiveness. Instead, it is much easier to do their thermal management to conform to the cooling profile. So, the simplest scheme is for each server to manage its own thermals based on the prevailing conditions (e.g., on - board temperature measurements). However, such independent controls can lead to unstable or suboptimal control. A coordinated approach such as the one considered in this paper could be used to ensure satisfactory operation while staying within the temperature limits - or rather within the power limits dictated by the temperature limit and heat dissipation characteristics. The next subsection establishes this relationship.

A. Energy-Temperature relationship

In the design of our control scheme we limit the power consumption of a device based on its thermal limits as follows.

Let t denote time, $T(t)$ the temperature of the component as a function of time, $P(t)$ power consumption as a function of time, and c_1, c_2 be the appropriate thermal constants. Also, let T_a denote the ambient temperature, i.e., temperature of the medium right outside the component. The component will eventually achieve this temperature if no power is supplied to it. Then the rate of change of temperature is given by

$$dT(t) = [c_1 P(t) + c_2(T(t) - T_a)]dt \quad (1)$$

Being a first-order linear differential equation, this equation has an explicit solution. Let $T(0)$ denote the temperature at time $t = 0$. Then,

$$T(t) = [T_a + [T(0) - T_a]e^{-c_2 t}] + c_1 e^{-c_2 t} \int_0^t P(\tau) e^{c_2 \tau} \, d\tau \quad (2)$$

where the first term relates to cooling and tends to the ambient temperature T_a and the second term relates to heating. Let T_{limit} denote the limit on the temperature and P_{limit} is the limit on power consumption so that the temperature does not exceed T_{limit} during the next adjustment window of Δ_s seconds. It is easy to see that,

$$T(\tau) = T_a + P_{limit} c_1 / c_2 [1 - e^{-c_2 \Delta_s}] + [T(0) - T_a] e^{-c_2 \Delta_s} \quad (3)$$

It can be observed that Equation 2 can be used to predict the value of temperature of the device at the end of the next adjustment window and hence can help in making the migration decisions. We use this relationship to estimate the maximum power consumption that can be allowed on a node so that it does not exceed its thermal limits.

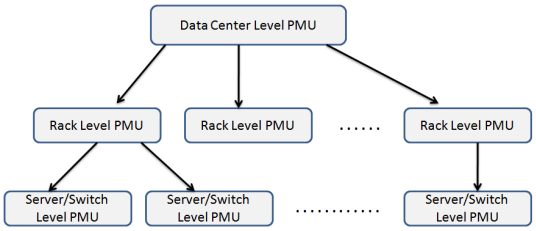

Fig. 1. A simple example of multi-level power control in a datacenter

IV. Supply and Demand side Adaptation in Willow

This section describes the mechanism adopted by *Willow* to adapt the workload to the energy and thermal profiles in a data - center via co - ordination between supply and demand sides.

A. Hierarchical Power Control

Power/energy management is often required at multiple levels including individual devices (CPU cores, memory DIMMs, NICs, etc.), subsystems (e.g., CPU - cache subsystem), systems (e.g., entire servers), and groups of systems (e.g., chassis or racks). In a power limited situation, each level will be expected to have its own power budget, which gets divided up into power budgets for the components at the next level. This brings in extra complexity since one must consider both the demand and supply sides in a coordinated fashion at various levels. In this paper we use such a multilevel power control architecture. One simple such power control model is shown in Figure 1. The data center level power management unit (PMU) is at the level 3. The rack level PMU is at level 2 and server/switch level PMUs are at level 1.

With such a multilevel power control architecture our control scheme attempts to provide the scalability required for handling energy and thermal adaptation in large data centers with minimum impact on the underlying networks.

In the hierarchical power control model that we have assumed, the power budget in every level gets distributed to its children nodes in proportion to their demands. All the leaf nodes are in level 0. The component in each level $l + 1$ has configuration information about the children nodes in level l. For example the rack level power manager has to have knowledge of the power and thermal characteristics of the individual components in the rack. Within a rack, the power and thermal characteristics of a SAN enclosure might be completely different from that of a Gigabit Ethernet switch. The components at level l continuously monitor the demands and utilization levels and report them to level $l + 1$. This helps level $l + 1$ to continuously adjust the power budgets. Level $l + 1$ then directs the components in level l as to what control action needs to be taken. The granularities at which the monitoring of power usage and the allocation adjustments are done are different and are discussed later in Section IV-C. The communication pattern of the control messages is shown in Figure 2.

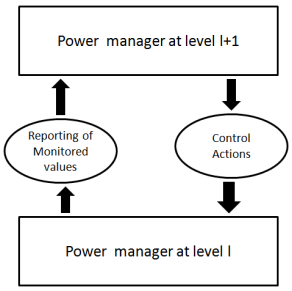

Fig. 2. Communication of control messages

B. Supply And Demand Side Coordination

As mentioned in Section II *Willow* implements a unidirectional hierarchical power control scheme. Migrations of power demands are initiated by the power and thermal constraints introduced as a result of increase in demand at a particular node or decrease in power budget to the node. Simultaneous supply and demand side adaptations are done to match the demands and power budgets of the components.

C. Time Granularity

The utilization of server resources in a data center varies widely over a large scale. If the nature of the workload fluctuates significantly, it is likely that different resources (e.g., CPU cores, DRAM, memory bus, platform links, CPU core interconnects, I/O adapters, etc.) become bottlenecks at different times; however, for a workload with stable characteristics (but possibly varying intensity) and a well-apportioned server, there is one resource (typically CPU and sometimes network adapter) that becomes the first bottleneck and its utilization can be referred to as server utilization. (Recommended server configuration practices attempt to achieve this). We assume this is the case for the modeling presented in this paper, since it is extremely difficult to deal with arbitrarily configured servers running workloads that vary not only in intensity but their nature as well. Under these assumptions, the power consumption can be assumed to be a monotonic function of the utilization. Furthermore, assuming that the bottleneck platform resource does not reach saturation, the relationship can be assumed to be approximately linear.

Because of varying intensity of the workload, it is important to deal with average utilizations of the server at a suitable time granularity. For convenience the demand side adaptations are discretized with a time granularity of Δ_{Dl}. It is assumed that this time granularity is sufficiently coarse to accommodate accurate power measurement and its presentation, which can be quite slow. Typically, appropriate time granularity at the level of individual servers are of the order of tens of milliseconds or more. Coarser granularities may be required at higher levels (such as rack level).

Even with a suitable choice of Δ_{Dl}, it may be necessary to do further smoothing in order to determine trend in power consumption. Although it is possible to use sophisticated ARIMA type of models, a simple exponential smoothing is often adequate. Let $CP_{l,i}$ be the power demand of node i at level l. For exponential smoothing with parameter $0 < \alpha < 1$, the smoothed power demand CP' is given by:

$$CP'_{l,i} = \alpha CP_{l,i} + (1-\alpha)CP'^{old}_{l,i} \tag{4}$$

Note that the considerations in setting up the value of Δ_{Dl} come from the demand side. In contrast, the supply side time constants are typically much larger. Because of the presence of battery backed UPS and other energy storage devices, any temporary deficit in power supply in a data center is integrated out. Hence the supply side time constants are assumed to be $\Delta_{Sl} = \eta_1 \Delta_{Dl}$, where η_1 is an integer > 1. *Willow* also performs workload consolidation when the demand in a server is very low so that some servers can be put in a deep sleep state such as S3 (suspend to memory) or even S4 (suspend to disk). Since the activation/deactivation latency for these sleep modes can be quite high, we use another time constant Δ_{Al} for making consolidation related decisions. We assume $\Delta_{Al} = \eta_2 \Delta_{Dl}$, for some integer η_2 such that $\eta_2 > \eta_1$.

D. Supply Side Adaptation

As mentioned earlier we ignore the case where the data center operates in a perpetually energy deficient regime. The available power budget of any level $l+1$ is allocated among the nodes in level l proportional to their demands. As mentioned in Section IV-C the supply side adaptations are done at a time granularity of Δ_{Sl}. Hence the power budget changes are reflected at the end of every Δ_{Sl} time period. Let TP^{old}_{l+1} be the overall power budget at level $l+1$ during the last period. TP_{l+1} is the overall power budget at the end of current period. $\Delta_{TP} = \text{TP}_{l+1} - \text{TP}^{old}_{l+1}$ is the change in overall power budget. If Δ_{TP} is small we can update the values of $\text{TP}_{l,i}$'s rather trivially. However if Δ_{TP} is large we need to reallocate the power budgets of nodes in level l. In doing so we consider both *hard* and *soft* constraints.

1) Hard Constraints are imposed by the thermal and power circuit limitations of the individual components. In our control scheme the thermal limits play an important role in deciding the maximum power that can be supported in the component.
2) Soft Constraints are imposed by the division of power budget among the other components in the same level.

The power and thermal constraints thus necessitate the migration of demand in level l from power deficient nodes to nodes with surplus power budget.

Any increase in the overall power budget happens at a higher level and is then reflected in its constituent lower levels. This situation can lead to three subsequent actions.

1) If there are any under provisioned nodes they are allocated just enough power budget to satisfy their demand.
2) The available surplus can be harnessed by bringing in additional workload.
3) If surplus is still available at a node then the surplus budget is allocated to its children nodes proportional to their demand.

E. Demand Side Adaptation

The demand side adaptation to thermal and energy profiles is done systematically via migrations of the demands. We assume that the fine grained power control in individual nodes is already being done so that any available idle power savings can be harvested. Our focus in this paper is on workload migration strategies to adapt to the energy deficient situations. For specificity we consider only those type of applications in which the demand is driven by user queries and there is minimum or no interaction between servers, (e.g.,) transactional workloads. The applications are hosted by one or more virtual machines (VMs) and the demand is migrated between nodes by migrating these virtual machines. Hence the power consumption is controlled by simply directing the user queries to the appropriate servers hosting them.

There are a few considerations in designing a control strategy for migration of demands.

1) *Error Accumulation*: Because of the hierarchical nature of the control scheme any small errors and uncertainties that occur in the topmost level add up as we move down the lower levels. As a consequence the worst errors are experienced by the lowermost levels.
2) *Ping-Pong Control*: A migration scheme that migrates demand from server A to B and then immediately from B to A due to erroneous estimations leads to a ping-pong control This is highly undesirable considering the overheads involved.
3) *Imbalance*: The migration scheme should not leave a few servers in the power deficient state while some servers have excess power budgets.

We carefully avoid these pitfalls by allowing sufficient margins both at the source and the destination to accommodate fluctuations after the migrations are done.

In the unidirectional approach that we have implemented, the migrations are initiated only by the tightening of the power constraints and not by their loosening. The tightening of power constraints is handled by reducing the power consumption via workload migration. The migrations are initiated in a bottom up manner. If the power budget $\mathrm{TP}_{l,i}$ of any component i is too small then some of the workload is migrated to one of its sibling nodes. We call this as local migration. If the sibling nodes of component i do not have sufficient surplus to accommodate its excess demand then the workload is migrated to one of the children of another $l+1$ component. We call this non-local migration. Always local migrations are preferred to non-local migrations. This is because of two reasons :

1) The overheads involving networking resources are reduced when the migrations are local rather than non-local. Also instance in some data centers typically the IP addresses are location dependent [17] and so a non-local migration would require reconfiguration of the IP addresses.
2) The VMs might have some local affinity with common resources like hard disks and a non-local migration might affect this affinity.

The migration decisions are made in a distributed manner at each level in the hierarchy starting from the lowermost level. The local demands are first satisfied with the local surpluses and then those demands that are not satisfied locally are passed up the hierarchy to be satisfied non-locally. The final rule in the unidirectional control scheme is that the migrations are destined to a node only if the power budget of the node is not reduced by the event that caused the migration. For instance if the power budget of a rack has gone down because of some reason, no migrations are allowed into that rack. Similarly if the power budget of the entire data center has reduced no migrations are allowed at all.

Now having defined the basic rules for the migration scheme we define a few terms related to the migration decisions.

Power Deficit and Surplus: The power deficit and surplus of a component i at level l are defined as follows.

$$P_{def}(l,i) = [CP'_{l,i} - TP_{l,i}]^{+} \tag{5}$$

$$P_{sur}(l,i) = [TP_{l,i} - CP'_{l,i}]^{+} \tag{6}$$

where $[]^{+}$ means if the difference is negative it is considered zero.

From the above definitions we can define the power surplus and deficit at a particular level to be

$$P_{def}(l) = max(P_{def}(l,i)) \tag{7}$$

$$P_{sur}(l) = max(P_{sur}(l,i)) \tag{8}$$

Power Imbalance: The power imbalance is a measure of the inefficiency in allocation of the power budgets and is defined as

$$P_{imb}(l) = P_{def}(l) + min[P_{def}(l), P_{sur}(l)] \tag{9}$$

The reason for capping the surplus by deficit is simply because any supply that is in excess of deficit is not handled by our control scheme and is left to be taken care of by the idle power control schemes that operate at a finer granularity. If there is no surplus that can satisfy the deficit in a node, the excess demand is simply dropped. In practice this means that some of the applications that are hosted in the node are either shut down completely or run in a degraded operational mode to stay within the power budget.

Power Margin (P_{min}): The minimum amount of surplus that has to be present after a migration in both the source and target nodes of the migration. This helps in mitigating the effects of fluctuations in the demands.

Migration Cost: The migration cost is a measure of the amount of work done in the source and target nodes of the migrations as well as in the switches involved in the migrations. This cost is added as a temporary power demand to the nodes involved.

A migration is done if and only if the source and target nodes can have a surplus of at least P_{min}. Also migrations

are done at the application level and hence the demand is not split between multiple nodes. Finally *Willow* also does resource consolidation to save power whenever possible. When the utilization in a node is really small the demand from that node is migrated away from it and the node is deactivated. Note that any measure of power savings that we attempt to present is a result of this strategy. In reality the power savings can be much more when idle power control techniques are in place.

F. Packing The Bins

Even with the above mentioned constraints there are multiple ways of matching a demand with a surplus. The entire problem now reduces to the classical bin packing problem. The surpluses available in different nodes form the bins. The bins are variable sized and the demands need to be fitted in them. The variable sized bin packing problem is a NP-hard problem as such and numerous approximation schemes are available in literature [18], [19], [20]. We choose one such simple scheme called FFDLR [20]. Let L be the list of demands that are to be satisfied. The FFDLR scheme works as follows.

1) The bin sizes and demand sizes are normalized so that the largest bin has a size of 1.
2) Pack the first demand in L into the first bin of size of 1.
3) Repeat steps 1 and 2 until all demands are matched with a surplus
4) At the end, the contents of all bins are repacked into the smallest possible bins.

FFDLR solves a bin packing problem of size n in time $O(n \log n)$. The optimality bound guaranteed for the solution is (3/2) OPT +1 where OPT is the solution given by an optimal bin packing strategy. We chose this algorithm for two reasons. First, it is simple to implement with guaranteed bounds. Second, repacking into smaller bins means we try to run every server at full utilization. The bins (servers) that are empty can then be deactivated during the consolidation phase.

V. Evaluation

A. Performance Analysis

In this section we discuss some of the properties of *Willow* that impact its performance.

1) Convergence: *Willow* is a distributed control scheme and the decisions are decentralized. Before we present the convergence analysis of *Willow*, a discussion of the following definition from [21] is necessary.

Definition 1: Consider an update to an object X in a system at time t. The system is said to be δ convergent if all sites in the system perceive the same value of X after time $t + \delta$.

In other words any update at time t is propagated to all sites in the system within time $t+\delta$. As mentioned in Section IV-C any update to the demand values is done at a time granularity of Δ_{Dl} at all the levels. A node at any level $l > 2$ obtains the updates in demand values only from its children and hence the communication of update messages is one - way. Assuming that the links carrying the control messages between the levels do not fail or do not suffer from prolonged congestion, the time taken by any message to be communicated from a leaf to the root is much less than Δ_{Dl}. Similarly the supply side updates are done at a granularity of $\Delta_{Sl} (> \Delta_{Dl})$. Here the one way communication is from the root to the leaves.

In order to estimate the value of δ let us consider a conservative approach. Consider a scenario where there are h levels in the power control hierarchy. An update message requires to be propagated through all these h levels. The nodes at every level have to update the old values and send the update to the next level (up or down). Suppose that the update propagation time through each level is at most α. So the overall time taken for propagation of update messages is $h\alpha$. In this case assuming the value of Δ_{Dl} to be much larger than the actual value (say, 10 times $h\alpha$) would avoid instabilities in decision making. Even in a very large data center, the number of levels in the hierarchy is unlikely to be more than 4 or 5, and update at each level can be done in a few tens of milliseconds. Therefore, $\delta \approx 50\,ms$, and a Δ_{Dl} value exceeding $500\,ms$ should be safe in almost all cases.

2) Time Complexity: The major aspect in the decision making process is solving the bin packing problem. The communication times are assumed to be negligible. Let n be the total number of leaf nodes (servers) in the data center.

Consider a problem instance to be a set of demands that have increased simultaneously at the different leaves of the tree. The distributed algorithm solves them independently at the level 1. Let b_l be the maximum branching factor of any node in level l.

Time complexity of FFDLR in level 1 = $O(b_1 \log b_1) = k$.

where k is a constant. Similarly, at every level the simultaneous problem solving takes a constant time ($O(b_l \log b_l)$) in solving the bin packing problem.

If h is the height of the tree and $h = O(\log n)$, the time complexity for making the decisions is $O(\log n)$.

3) Other Properties: We highlight some of the important properties of *Willow* here.

Property 1: The solutions of the different instances of the bin packing problem with given locality constraints are optimal with the bounds guaranteed by the FFDLR algorithm. The locality constraint that we refer to here is the rule that local migrations are preferred to non-local migrations.

Property 2: The introduction of resource constraints along with the locality constraints will still yield optimal solutions. The resource constraints that we refer to here are non - availability of adequate resources like CPU or large affinity to some resources like hard disks.

In general the introduction of any constraints results in the reduction of number of surpluses that are being presented to satisfy the demands. Hence the solutions will be the same irrespective of whether the problem is solved in a distributed manner or it is solved at a centralized location.

Property 3: The number of communication messages on any network link between a node at level l and a node at level $l+1$ in a period of Δ_{Dl} is at most 2 - one on either direction in the

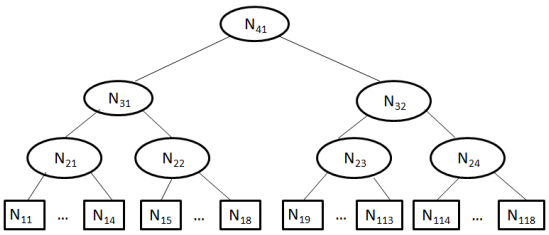

Fig. 3. Configuration used in simulation

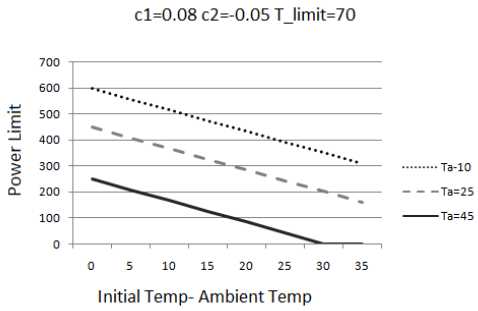

Fig. 4. Setting up the thermal constants

link. This significantly reduces the communication overhead due to the control messages that are exchanged.

Property 4: Let P_{min} be the migration threshold i.e *Willow* migrates a demand from a node n_1 to a target node n_2 only if there is a surplus of at least P_{min} in the source and target nodes after the migration. Let Δ_f be the time for which the power demands do not increase beyond P_{min} in time Δ_f. After a decision is made the validity of the decision remains at least for time Δ_f. In other words, a demand that has migrated from node n_1 to node n_2 remains in node n_2 at least for time Δ_f.

We conducted detailed simulations and real time experiments to evaluate the performance of *Willow*. The simulations were based on generalized assumptions on the power demands and limits. The experiments were done to calibrate the equation that defines the relationship between power and temperature and to test the working of *Willow* when real migrations of VMs is done practically. The simulations and experiments demonstrate the adaptability of *Willow* to varying energy and thermal conditions. Section V-B explains the simulations and Section V-C explains the experimental results.

B. Simulations

1) Simulation Environment: We simulated *Willow* in MATLAB. The simulator can be configured to simulate any number of levels and servers in the data center. For our performance evaluation we used the configuration shown in Figure 3. There are four levels in the power control hierarchy and 18 server nodes. On each server we placed a random mix of 4 different application types that have a relative average power requirement of 1, 2, 5 and 9. The average power demand in a server is the sum of all the average power requirements of the applications that are hosted in it. The power demand in each node was assumed to have a Poisson distribution. The time constant multipliers for discrete time control η_1 and η_2 in Section IV-C are assumed to be 4 and 7 respectively.

2) Setting Up the Thermal Constants: The thermal constants in Equation 1 are dependent on the thermal properties of the individual devices. For our simulations we calculate the values of these constants by assuming reasonable values for the maximum device power and the ambient temperature. We assume that the average server/switch power consumption is around 450 Watts and a typical ambient temperature to be 25°C. Also the thermal limit of the servers and switches is assumed to be 70°C. With these settings, Figure 4 shows one possible set of values for the constants c_1 and c_2. When the power consumption is zero (when the server is in a deep sleep state because of no activity), the component is at the ambient temperature. The power surplus that is presented at this temperature should be approximately the same as the maximum power rating of the component. From Figure 4 it can be seen that the values $c_1 = 0.08$ and $c_2 = -0.05$ show the maximum power limit to be around $450\,W$ when the ambient temperature T_a is 25°C and the initial power consumption is zero. Hence we chose these values for c_1 and c_2 for our experiments. It can be seen from Figure 4 that when the ambient temperature T_a = 45°C and the temperature of the server is at 70°C the power surplus that is presented is almost zero because the server has already reached its thermal limit.

We assume that the time taken for the increase in temperature when the demand increases, is less than Δ_{Dl}. This is a conservative assumption. In reality the temperature might not reach a steady state value before Δ_{Dl}. However such a conservative assumption avoids the tasks that are migrated to a node from being migrated away again within a short period of time. In other words, it reduces the number of migrations and instability in decisions.

3) Energy and Thermal Adaptation: As explained in Section III *Willow* migrates the demand from energy deficient areas to energy surplus areas. In doing so care is taken so that too much workload is not migrated to already hot zones. In order to demonstrate this behavior of *Willow* we set the ambient temperature of servers 1 to 14 to be equal to 25°C and the ambient temperature of servers 15 to 18 as 40°C. This setting reflects a real time scenario where in a data center some servers are in the hot zones and some others are in cooler zones. Figure 5 shows the average power consumption under this setting for different utilizations. The average power consumption of servers 15 to 18 is much lower than the rest of the servers. This is because the power surplus that is being presented in these servers is much less than the others due to the higher ambient temperature. Hence less workload is running on these servers than the others. It can be seen from Figure 5 that at low utilizations the power consumed by the servers in the high temperature zone is low. As the utilization increases, the power consumed by these servers

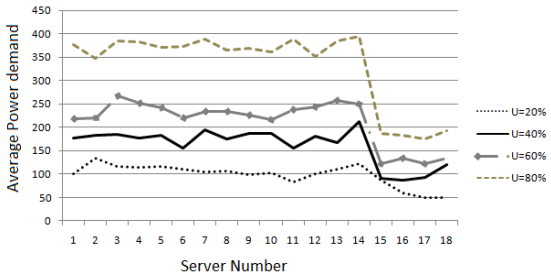

Fig. 5. Average power consumption when T_a=25°C for servers 1-14 and T_a=40°C for servers 15-18

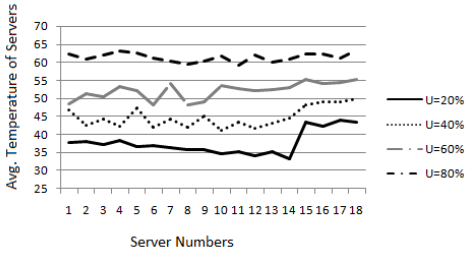

Fig. 6. Average temperature of servers when $T_a = 25°C$ for servers 1 - 14 and $T_a = 40°C$ for servers 15 - 18

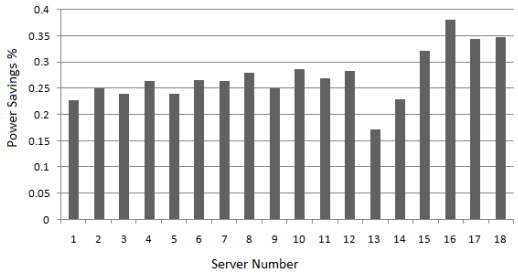

Fig. 7. Power saved in the servers as a result of consolidation when $T_a = 25°C$ for servers 1 - 14 and $T_a = 40°C$ for servers 15 -18 (U=40%)

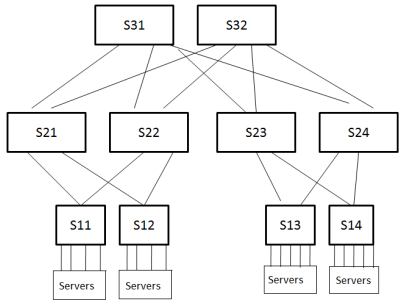

Fig. 8. Switch Configuration

also increases to cope with the higher utilization, but only upto the limit provided by the thermal constraint. Figure 6 shows the average temperature of all the servers with the above mentioned setting. At low utilization levels the servers in the hot zones are maintained at a temperature close to the ambient temperature of $40°C$. The variation in temperature of the servers in the hot and cold zones gradually reduces with the increase in utilization and the temperature of the servers is almost uniform when the utilization is very high. Figure 7 shows the power savings achieved in each server at 40% utilization. Again we emphasize that this power savings is a result of consolidation of workload during low utilization levels. We see that maximum power savings is achieved in the last four servers. This is because *Willow* tries to move as much work away from these servers as possible due to their high temperatures and hence they remain shut down for more time.

4) Migration Traffic: The most important aspect of *Willow* is the migrations that are done to adapt to the energy and thermal profiles in the data center. A measure of the migration traffic is important because migrating a demand is an overhead both in terms of power and network resources. Migrations in *Willow* are either demand driven or consolidation driven. While the former cause is more often seen in high utilization cases the latter is observed a lot in low utilization cases. This trend is shown in Figure 9. Figure 10 shows the proportion of migration traffic normalized with respect to the maximum possible utilization of the network. This normalization is necessary if we need to have an absolute picture of the migration overhead. For instance 2% of the overall traffic at 10% utilization may be much smaller than 2% of the overall traffic at 80% utilization. We see that in Figure 10 the migrations are increasing with increase in utilization. However at high utilization levels the migration traffic is decreasing. This is because at higher utilizations, the servers have no surplus to accommodate the workload from other servers. At 50% utilization there is a sudden increase in the number of migrations. This is because at 50% utilization both demand and consolidation driven migrations occur almost equally and hence the number of migrations shoots up. However at higher utilizations very less number of migrations occur since none of the servers has a surplus to accommodate the deficit that is arising in the other servers.

5) Switch Power Consumption: The migrations have a direct impact on the switch power consumption since the switches are involved directly in the migrations and an indirect impact since after migration the corresponding switch that handles connections to the target node of the migrations may have to handle increased traffic. The switch configuration that we assumed for our simulation is shown in Figure 8. We can easily observe the correspondence of the switch configuration in Figure 8 and the power control hierarchy in Figure 3. Level 1 switches are placed along with the servers in level 1 in the power control hierarchy, level 2 switches are in level 2 in the power control hierarchy and so on. The switches are allocated a power budget from a control component one level

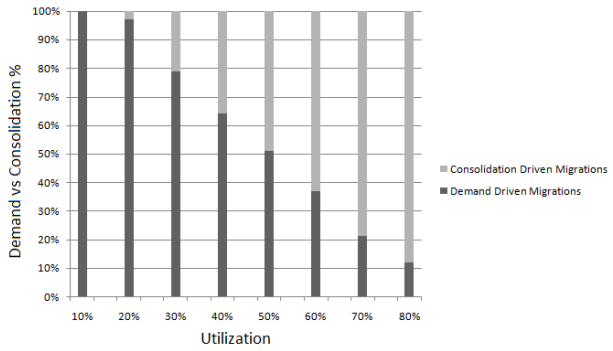

Fig. 9. Demand driven vs Consolidation Driven Migrations

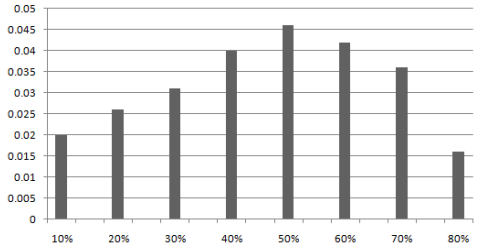

Fig. 10. Migration traffic in the switches normalized to maximum traffic

above. This correspondence of the switch hierarchy and power control hierarchy may not be present always. However the switches draw power from one of the levels in the hierarchy. By placing the switches in the hierarchy and apportioning a power budget to them we can directly control the amount of traffic that can go through a switch. We have a simple power model for the switch power consumption. We assume that the switch power consumption has two parts static and dynamic. The dynamic portion of the power consumption in a switch is directly proportional to the amount of traffic it handles. The static part is fixed and is very small. Assuming the static part to be small is idealistic since the idle power control techniques have to be extremely efficient to achieve this. Nevertheless, we are positive that this is achievable in the near future. We also assume that in the presence of redundant paths with two switches, the load is balanced evenly between the switches. This is typically the case in any data center with redundant network paths. Figure 11 shows the average power consumption of the level 1 switches for different average utilizations in the servers. According to our switch power model the power consumption in switches in the other levels would simply be the aggregation of demands in their children switch nodes. We see that the average power demand is almost the same in all the switches. At lower utilizations one might expect different power consumptions in different switches as a result of consolidation. But the fact that local migrations are preferred to non-local migrations, evenly spreads out the traffic

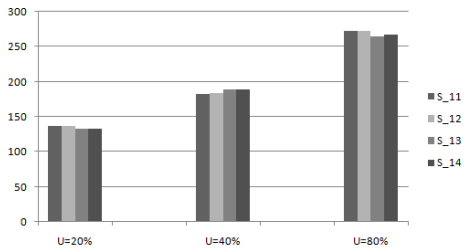

Fig. 11. Power Demand Of Level 1 Switches

across all the switches. Figure 12 shows the migration cost that is directly associated with the switches. This corresponds to the trend in total number of migrations that are done at different utilizations as shown in Figure 10.

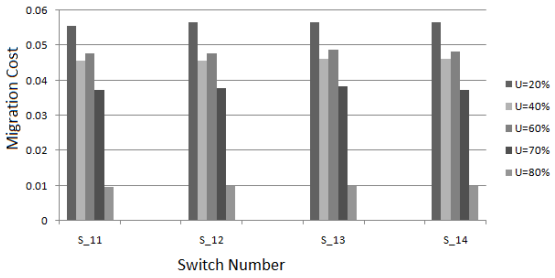

Fig. 12. Migration cost in level 1 switches

C. Experimental Evaluation

TABLE I
UTILIZATION VS POWER CONSUMPTION

Utilization%	Average Power consumed(Watts)
20	175
40	190
60	215
80	230
100	245

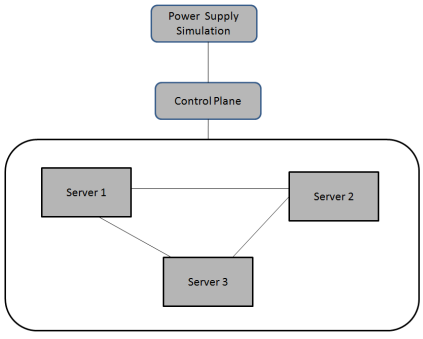

Fig. 13. Experimental Testbed

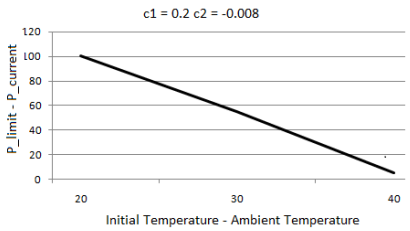

Fig. 14. Experimental estimation of parameters c_1 and c_2

1) Experimental Testbed: We conducted real time experiments to evaluate the performance of *Willow*. Figure 13 shows the architecture of our experimental testbed. A cluster of three Dell servers was formed and VMWare ESX server 3.5 [22] was running on all three of the servers. The three servers were managed from a remote machine. The remote machine is the control plane which simulates the hierarchical power control scenario. Since there were only 3 machines, we simulated a power control hierarchy with two levels - two switches in level 1 and one switch in level 2. CPU temperature was measured from the onboard sensor. CPU utilization of the ESX servers was monitored continuously by a script running on the control plane. The variation in power supply was introduced into the system artificially and the control system made migration decisions to adapt to the variations in power supply. Virtual machines were installed on each of the ESX servers. These virtual machines were hosting web servers that supported one of three applications, each with different power profiles. For simplicity all the applications were CPU bound. Hence the power - temperature measurements that we make correspond to CPU utilization.

2) Baseline Experiments: We conducted baseline experiments to parameterize Equation 1 and to capture the relationship between power and utilization. A CPU intensive application was running in the server. Table I shows the power consumption of the servers for various CPU utilizations. We see that the power consumption is a continuously increasing function of utilization. The static power consumption was found to be almost a constant. With only a single component in the server (in our case CPU) being involved in the computations, this kind of a linear relationship is quiet intuitive. The scenario where power consumption follows only the utilization is an idealistic situation. Fine grained power management techniques aim at achieving this goal by trying to minimize the static power consumption. The scope of *Willow* is not idle power control but to adapt to the power supply and demand. Notably, the servers that we used are old machines and hence VMWare ESX hypervisor was not able do any power management by itself. The power consumption was measured using Extech 380801 power analyzer. The sampling rate of the power analyzer was around 2Hz.

Figure 14 shows the trend in power consumption vs temperature. The parameter values in Equation 1 were determined to be $c_1 = 0.2$ $c_2 = -0.008$. Note that these values are different from those used in the simulation. This is because the maximum wattage in the simulations was assumed to be around $450\,W$. However in practical scenarios, the power that a server consumes even at 100% utilization is much less than the maximum wattage. In our case at 100% CPU utilization the power consumed was around $245 - 250W$. The y-axis in Figure 14 is the maximum power that can be accomodated and the x-axis is the difference between ambient temperature and current temperature of the device. The ambient temperature was assumed to be uniform throughout the room and was equal to $25°\mathrm{C}$.

TABLE II
APPLICATION POWER PROFILE

Application	Increase in power consumption (Watts)
A_1	8
A_2	10
A_3	20

3) Application Profiling: We designed three applications A_1, A_2 and A_3 with different CPU utilizations and hence different power consumptions. Table II shows the increase in power consumption when three applications are running on the server. Each application is run on a single virtual machine. The power consumption of the servers can be controlled by migrating the VMs away from or to the servers.

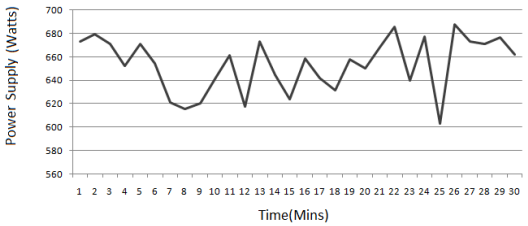

Fig. 15. Power supply variation (Energy Deficient Situation)

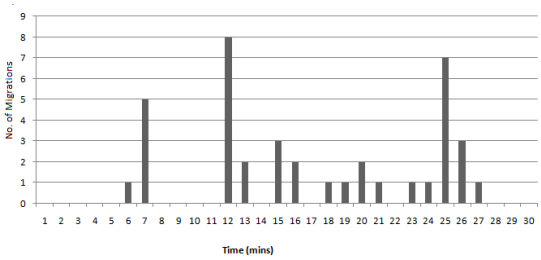

Fig. 16. Number of Migrations

4) Experimental Results: In this section we present the observed results when the servers are running at an overall average utilization level of 60%. Figure 15 shows the power supply variation pattern that was used in the simulation when the average utilization is 60%. As stated in Section I the perpetually energy deficient regime is simply ignored and only

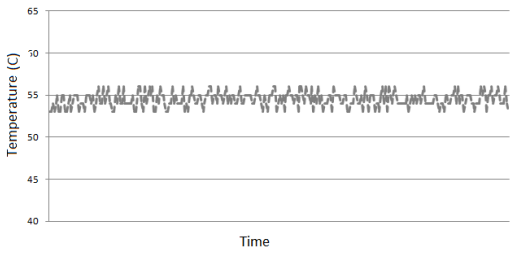

Fig. 17. Temperature Time Series (Server A)

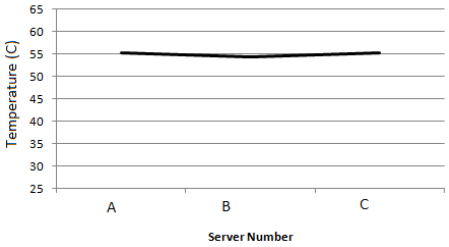

Fig. 18. Average Temperature of Servers

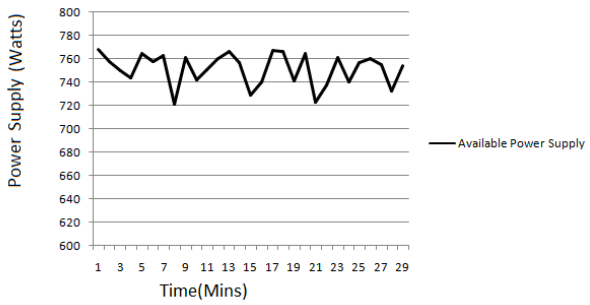

Fig. 19. Power supply variation (Energy Plenty situation)

short term energy deficiencies are considered. The available power supply is divided proportionally between the servers. When the available power supply is reduced the tasks are migrated away from servers running at high utilizations to servers running at a low utilization. In general the number of migrations to or from a server depends on the available power supply and the utilization at which it is running. This can be observed in Figure 16. For instance, in Figure 15 at time unit 7 the power consumption plunges deeply. In a power deficient situation, servers that are running at a higher utilization have to migrate some workload away from them to maintain the same QoS. It can be seen from Figure 16 that the number of migrations is also high in time unit 7. The decreased power supply persists till time unit 10. However there are no migrations between time units 7 and 10. This is because of the decision stability property explained in Section V-A. Once the migrations are done there is enough margin left to handle the demand variations. Hence no more migrations are needed. Similar patterns can be observed in time units 12 and 25. Note that after a deep plunge in power supply there is not much migration activity even if the power supply increases. This is because of the fact that the migrations in *Willow* are always initiated by the tightening of power constraints and not by their loosening. Note that this is true only for constraint driven migrations and does not include the consolidation driven migrations. The initiation of migrations in a power - plenty situation can happen only at low utilization levels where there is a possibility of shutting down a server completely.

5) Workload Consolidation: In Section V-C4 we observe that at an average utilization level of approximately 60% it is not possible to shut down any server. This scenario is different from the case in simulation results where power savings is obtained even at 60% utilization. That is because the average power supply that we used in the simulations is close to the maximum power limit of the servers. However in the case of the experiments in Section V-C4 the average power supply was just enough to support an average utilization of 60%. This demonstrates the ability of *Willow* to adapt to the variations in power supply. In this section we describe the potential power savings achieved by *Willow* as a result of consolidation when there is a large surplus in power supply. In other words this scenario is the case where some servers are running at low utilization levels. The available power supply varied as in Figure 19. Note that the average power supply available is close to the necessary power supply to support the three servers at a utilization level of 100% ($\approx 750W$). Consolidation driven migrations are initiated when the utilization in the server is lower than the migration threshold. We set this migration threshold to be 20%. When the migration threshold of a server falls below this value consolidation driven migrations are initiated. Table III shows the initial utilization levels of the servers and average utilization levels of servers after the run of the experiments. We see that the utilization of Server

TABLE III
UTILIZATION OF SERVERS

Server	Initial Utilization%	Average Utilization at the end of experiment
A	60	80
B	40	40
C	20	0

C is 0%. This is because Servers A and B are running below their power and thermal limits even after the tasks have been migrated from Server C. Hence there is no need to wake up server C. Server C remains shut down for the entire run of the experiments. Let us calculate the power savings achieved by *Willow* in this case.

Without consolidation, if the servers A, B and C were running at utilizations of 80%, 40% and 20% respectively, their average power consumption from Table I is (175+190+215) $580W$. However after consolidation, the average power consumption is (230+190+standby) 420 + standby - power consumption for server C. It is reasonable

to assume that this standby power consumption is negligible. For instance VMWare ESX server has the Dynamic Power Management [23] utility that can shut down an idle host and the power consumed is zero. With that assumption the power savings achieved is around 27.5%.

VI. Conclusion and Future Work

In this paper we have designed *Willow*, a control scheme for energy and thermal adaptive computing. In the process of adapting the workloads according to the thermal and energy constraints of the individual components, *Willow* makes stable decisions. The stability in the decision making directly translates to reduced networking impact since there are no flip-flops in migrations. The power savings that have been presented are only the result of completely deactivating the individual servers or putting them in standby mode as and when *Willow* decides to do so. The goal of *Willow* is not to maximize the power savings but to adapt to the energy profile. In allocating the tasks, *Willow* tries to maximize the power efficiency. The decision making is decentralized and centralized decisions are minimized as much as possible. The thermal constraints were never violated in the simulations or experiments in any component and no ping - pong migrations were observed at least for a time $\Delta_f < 50\Delta_{Dl}$.

In this work, we have tested the performance of *Willow* in the case of transactional workloads in simulations for specificity. In our experiments we used applications that are CPU intensive. A more complete design must be able to measure power consumption and temperature of every component in the server including memory, NIC, hard disks etc. and make fine grained control decisions. We would also like to analyze the performance of *Willow* under more complex workloads where there is excessive IPC traffic among the servers. A real time implementation might need to consider the migrations that are caused as a result of resource constraints as well. In that case a synchronization mechanism between these migrations and migrations caused by *Willow* may be necessary. In order to do a holistic power control, *Willow* must consider the energy consumed by cooling infrastructure as well in the adaptation. We do not explicitly model multiple QoS classes and the impact of changes in the network traffic on the QoS of applications. Nevertheless, the goal of *Willow* is to minimize QoS impact by dynamic energy allocation and task migrations. Dealing with multiple QoS classes is a future direction that we intend to pursue.

References

[1] K.Kant, "Distributed Energy Adaptive Computing," in *Proceedings of International Conf. on Communications, ICC '10*, 2010.

[2] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008, pp. 323–336.

[3] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi, "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," in *39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-39 2006), 9-13 December 2006, Orlando, Florida, USA*, 2006.

[4] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 103–116, 2001.

[5] K. Kant, "Power control of high speed network interconnects in data centers," in *INFOCOM'09: Proceedings of the 28th IEEE international conference on Computer Communications Workshops*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 145–150.

[6] M. Gupta and S. Singh, "Dynamic Ethernet Link Shutdown for Energy Conservation on Ethernet Links," in *ICC*, 2007, pp. 6156–6161.

[7] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: dynamic speed control for power management in server class disks," *SIGARCH Comput. Archit. News*, vol. 31, no. 2, pp. 169–181, 2003.

[8] D. Colarelli and D. Grunwald, "Massive arrays of idle disks for storage archives," in *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*. Los Alamitos, CA, USA: IEEE Computer Society Press, 2002, pp. 1–11.

[9] B.Heller, S.Seetharaman, P.Mahadevan, Y.Yiakoumis, P.Sharma, S.Banerjee, and N. McKeown, "ElasticTree: Saving Energy in Data Center Networks," in *NSDI'10: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2010.

[10] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": temperature-aware workload placement in data centers," in *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2005, pp. 5–5.

[11] Y. Wang, K. Ma, and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 314–324, 2009.

[12] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: a fast array of wimpy nodes," in *SOSP '09*. ACM, 2009, pp. 1–14.

[13] R. Nathuji and K. Schwan, "VirtualPower: Coordinated power management in virtualized enterprise systems," in *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*. New York, NY, USA: ACM, 2007, pp. 265–278.

[14] X. Wang and Y. Wang, "Coordinating Power Control and Performance Management for Virtualized Server Clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 99, 2010.

[15] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers," in *EuroSys '09: Proceedings of the 4th ACM European conference on Computer systems*. New York, NY, USA: ACM, 2009, pp. 317–330.

[16] K.Kant, "Supply and Demand Coordination in Energy Adaptive Computing," in *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, 2010.

[17] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, ser. SIGCOMM '09. New York, NY, USA: ACM, 2009, pp. 51–62.

[18] D. S. Johnson, A. J. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms," *SIAM J. Comput.*, vol. 3, no. 4, pp. 299–325, 1974.

[19] C. Chekuri and S. Khanna, "On Multi-Dimensional Packing Problems," in *SODA*, 1999, pp. 185–194.

[20] D. K. Friesen and M. A. Langston, "Variable sized bin packing," *SIAM J. Comput.*, vol. 15, no. 1, pp. 222–230, 1986.

[21] F. J. Torres-Rojas and E. Meneses, "Analyzing Convergence in Consistency Models for Distributed Objects," in *OPODIS*, 2004, pp. 346–356.

[22] VMWare ESX Server, http://www.vmware.com/products/vsphere/esxi-and-esx/index.html.

[23] VMWare Dynamic Power Management, http://www.vmware.com/virtualization/green-it.