# Energy management of highly dynamic server workloads in a heterogeneous datacenter

Efraim Rotem[1,2]
Uri C. Weisser[1]

Avi Mendelson[1]
Ahmad Yasin[1,2]
Ran Ginosar[1]

[1]Technion – Israeli institute of Technology, Haifa, Israel

[2]Intel corporation, Haifa, Israel

***Abstract*— we propose a hybrid management model to address energy efficiency in heterogeneous datacenters with highly dynamic workloads. A central dispatch and control algorithm with distributed system energy management was implemented and validated on real processor and system. We demonstrate up to 20% energy savings (11% average) without compromising quality of service. Additional 5% average energy savings was achieved by exploiting system heterogeneity.**

***Keywords—Power management, energy efficiency, datacenter***

## I. INTRODUCTION

Energy efficiency is a fundamental consideration in building and management of a datacenter. Energy consumption and energy efficiency are not independent parameters, and are subject to optimization only in the context of Quality of Service (QoS). Cloud computers such as Microsoft Azure and Amazon Elastic Compute Cloud guarantee their customers a predefined Service Level Agreement (SLA). Failing to meet the guaranteed SLA bares business and financial implications. Custom solutions such as Google web search or financial services are also required to meet QoS in order to maintain customer satisfaction and avoid revenue lose. Lowering system performance and saving energy is acceptable as long as the guaranteed user QoS is achieved. Workload characteristics also impact the energy management opportunities. Throughput workloads are less sensitive to latency and can employ energy management actions with longer latency while Web search or online trading are highly sensitive to latency and require fast responding energy saving states. Furthermore, completion of distributed latency sensitive workloads often pending for "the long tail" [5][6] and slowing down a single thread may compromise QoS of the entire task. The focus of this study is throughput class of workloads. The methods that are presented in this study will be extended to latency sensitive workloads in a future work.

Datacenter workloads experience high dynamic range of activity [8]. The capacity of the datacenter is determined by the worst case demand. Unutilized systems can be put into various idle states e.g. platform S-states. Deeper idle states conserve more energy but require longer time to restore, and at a cost of higher activation energy. Therefore, systems are put in deep power saving states only if they are expected to remain in this state for a long time. Online systems can be either active or in a shallow sleep state that can start executing computational tasks within micro- to few milli-seconds. This study aims at minimizing energy of these online systems while meeting QoS.

Heterogeneity exists in the datacenter as a result of natural evolution. Servers are deployed at different times and from different manufacturers, older systems replaced by new, better energy-efficient systems etc. [3][4]. Recently, Heterogeneity has been embraced as a mean for energy conservation where heterogeneous CPUs have been proposed to address energy efficiency of highly dynamic QoS demand. Our proposed energy management techniques further exploit datacenter heterogeneity in order to minimize energy consumption.

## II. OVERVIEW OF THE STUDY

We propose a hierarchical approach for managing the datacenter. Modern CPUs and compute platforms are equipped with highly optimized power management mechanisms. These mechanisms are tightly optimized to the individual CPU, platform and workload micro-architectural behavior. Obviously each platform model is different and managed differently. However, the knowledge about the required QoS is usually applied either externally by user preference, service agreement or by management agents that track incoming requests for example. On big data applications e.g. MapReduce the QoS is often not a single platform metric but the accumulation of multiple platforms. We study a hierarchical approach with a central agent that performs job dispatching, tracks the overall QoS and closes a formal control loop that notifies the underlying systems whether to accelerate or decelerate and the target system (leafs) perform a local power management algorithm.

The proposed system power management algorithm is an extension to the Energy Aware Race to Halt (EARtH) algorithm [2]. We evaluated the system power management algorithm on a set of 101 workloads consisting of SPEC2000, SPEC2006, HPC and big data workloads (e.g. DGEMM, k-means, Google PageRank, Pattern Matching) and Apache Hadoop implementation of MapReduce. We implemented the system power management in a driver on a 35nm and 22nm Intel® Core™ processor. We also coded the

algorithm into the power management firmware of the 14nm processor soon to be shipped and collected energy and performance measurements of these workloads. We have constructed a Monte-Carlo simulator to randomly accept sets of workloads and perform the central dispatching and control module.

The main contributions of this study are:

- A novel hierarchal datacenter management that combines DVFS and job dispatching to optimizes energy consumption while maintaining throughput QoS.
- The local system algorithm was implemented in the firmware of a production high volume 14nm Intel® core™ processor and validated on multiple real systems
- An average of 11% energy with up to 20% energy savings have been demonstrated.
- Introduced a dispatch mechanism that utilizes datacenter heterogeneity. The use of a single meta-data parameter conserved an average of 16% energy with up to 21.7% energy.

## III. Related Work

Kansal et al. [7] have proposed a hybrid energy management algorithm for the datacenter. They propose a mechanism to collect the actual QoS and control the individual systems QoS. In their work, energy control is done by modifying the application and compromising QoS (e.g. lower frame rate in video processing). We propose hybrid approach to conserve energy by changing the voltage and frequency of the processor without compromising QoS. Meisner et al. [10] propose PowerNap algorithm, running the processor of all the systems at the highest frequency, and then put the entire system in idles state (often named Race to Halt – RtH). We have shown in previous work [2] that the highest frequency is not the most optimal frequency. In this work we make use of this observation and set the processor of each system at its most energy efficient frequency. Meisner et al. [5] and Jeffrey et al. [6] also distinguish the throughput vs. responsiveness workload characteristics. Meisner et al. [5] introduce the online data intensive workload characteristics (e.g web search) and describe the energy management challenges of such workloads. One of these challenges is the need to keep enough systems active in order to meet the latency requirements, limiting the use of PowerNap [10]. Our work addresses this challenge and provides energy management algorithm that keeps enough systems active and offers energy savings for these active systems. However, our work does not fully address the Long Tail challenge [6] and the extension of the proposed algorithm is a subject of future work. Mars et al. [3][4] surveyed existing datacenters and demonstrated the existence of heterogeneity in datacenters. They proposed dispatch algorithm to exploit this heterogeneity for the purpose of energy management by selecting the right system for the right workload. Our study further extends this idea by utilize the powerful energy savings capability of Dynamic Voltage and Frequency Scaling (DVFS). We show that proper use of DVFS more than doubles the energy savings (Figure 5 and Table 1). Fan et al. [8] profiled system utilization and load over time in the datacenter. Barroso et al. [9] evaluate the energy cost and performance in the datacenter. We base our online system requirements and energy optimizations on this observation.

## IV. The Study Details

We address a datacenter model described in Figure 1. The focus of the proposed method is at leaf nodes since most of the datacenter energy is consumed by leaf systems [5]. The number of online leaf systems is predefined by the peak required QoS and the usage profiles of the datacenter [5] and may vary from time to time. Changing the number of online systems occurs at long time intervals and is not the scope of this study. The unutilized online systems are put in a shallow sleep state such as S0ix, S1/2 [1] and participate in the power management algorithm.

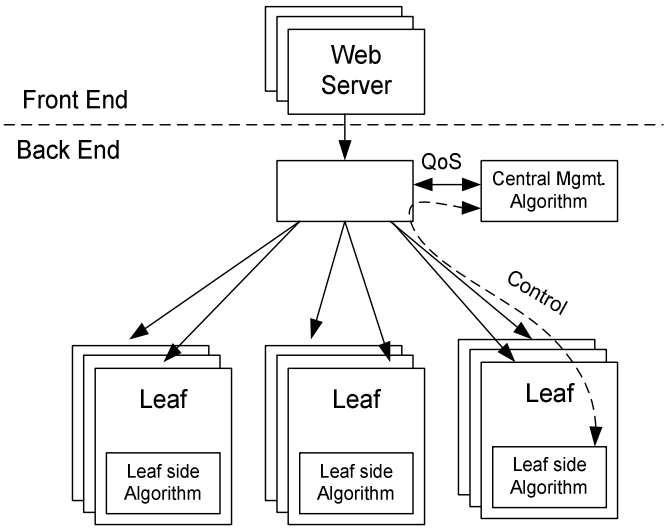


**Figure 1: The datacenter model of this study**

We propose a hierarchical energy management algorithm. The central management algorithm tracks the overall application QoS of all participating leaf systems and applies a formal control algorithm to maintain the desired QoS. The control is performed by sending notifications to the leaf systems over the interconnect fabric of the datacenter (Figure 1). The individual leaf systems respond to these notifications by increasing or decreasing their performance and energy. We have shown that EARtH [2] offers energy management within QoS constraints and therefore we choose it for the leaf side algorithm. The performance to energy profile is described by Equation 1 where SCA is the workload scalability with frequency due to memory access patterns and CPR is the workload power relative to the rest of the platform power. The rest of the platform power in this implementation is measured at the shallow idle state of the system. The average CPR and SCA of the workload are saved as metadata about each workload. The metadata represents the average values over the entire workload run while the local optimization is done at intervals of 1mSec. More details can be found in [2].

**(1)**

$$\frac{E_f}{E_{f0}} = \left(SCA \cdot \frac{1}{f_c} + 1 - SCA\right) \cdot \left(CPR \cdot F\left(f_c\right) + 1 - CPR\right)$$

We instrumented 35nm, 22nm and 14nm Intel® Core™ systems for power measurements at 5mSec rate. These processors incorporate internal performance monitors to report SCA and power consumption used for CPR calculation. We collected power performance measurements on the above 101 workloads at 8 different frequencies. We used Intel® Xeon® E5-2697 V2 with Apache Hadoop implementation of MapReduce running the CloudSuite's Data Analytics workload. We implemented the algorithm in Figure 2 in an offline simulator that uses the power performance runs as an input. A computational task, consisting of multiple processes or threads is required to deliver some guaranteed known quality of service (SLA). SLA is defined in this study as sum of all components ($SLA = \sum_1^k SLA(i)$ ). All the benchmarks that we tested are measured by either time to complete or 1/time. In order to sum scores of different benchmarks and have equal weigh for each one, we normalize each benchmark to its score at fixed reference frequency and same polarity (i.e. higher value is better).

Central Management side Algorithm

```
// Implemented in a central manager that communicates
// with leaf systems, reading throughput and sending respond

Every time interval {
  Read and calculate Actual_SLA = ∑1k SLA(i);
  if Actual_SLA < Target_SLA {
      If systems are available
        Dispatch workload to a free leaf;
      else
        Send leafs x = F(Target_SLA- Actual_SLA);  // inc. frequency
      }
  else
        Send leafs x = F(Target_SLA- Actual_SLA);  // dec. frequency
}

Leaf side algorithm
Every time interval{
  Update SLA(i)
  Get x
  If X>0 increment frequency by X*gain;
  If X<0 decrement frequency by -X*gain;
}
```

**Figure 2: Central management algorithm**

## V. DISPATCHING ALGORITHM AND RESULTS

There are several possible policies to dispatch M jobs/second to N systems having P power/performance states in order to achieve a given SLA. We described above that enough systems are set online to meet the required SLA. One existing policy [10] is to set each system in the highest possible frequency (P0 [1]), dispatch the jobs to as many systems as needed to meet SLA. This policy is referred to race to halt (RtH) in this study. If the system power is dominating the overall power consumption, this policy is the most efficient. If the CPU dominates the system power, a better policy would be to set all the systems at the lowest possible voltage and frequency (LFM) and to activate as many systems as needed. In this section we evaluate the case that SLA can be satisfied with the existing systems at LFM. We refer to this scenario as **Light Load**. If not, Algorithm 1 need to increase the frequency of the individual systems until SLA is satisfied. This scenario is referred to as **High load workload** and is evaluated in Section VI.

It is possible to build a local system algorithm to find the optimal frequency that performs a computational task with minimum energy [2]. We extend it to a datacenter scope with many systems. Jobs are dispatched to the individual systems. Each system picks the optimal frequency for the specific workload on this specific system and dispatching continues until SLA is honored. This policy is referred to as Fopt.

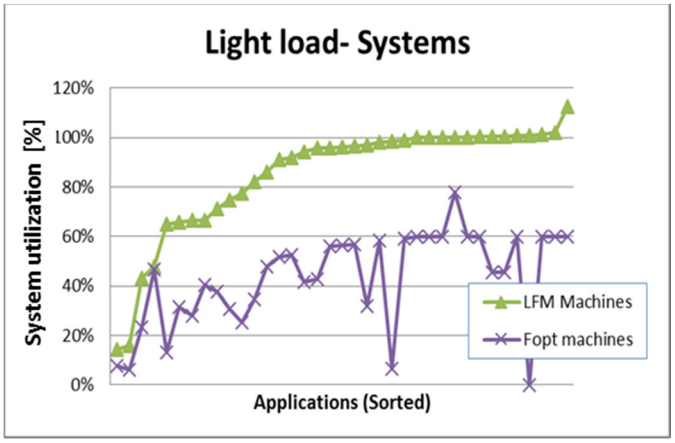


**Figure 3: System utilization of LFM and Fopt policies compared to RtH in a light load scenario**

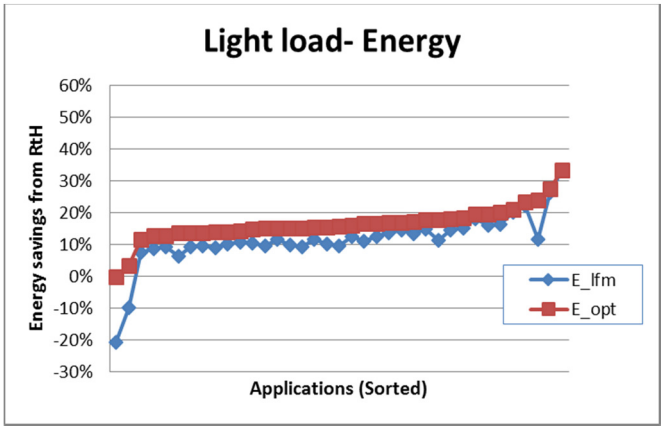


**Figure 4: Energy savings of LFM and Fopt policies compared to RtH in a light load scenario**

We assume unbounded number of systems. Each run, the Monte Carlo algorithm picked a random set of 10 types of workloads out of the 101 with infinite number of jobs of each type. The above tree dispatching policies where applied. The number of systems was used to control the SLA i.e. the fewest systems where used at RtH, more systems where used for LFM and some intermediate value at Fopt yielding the same

throughput. Each point in Figure 3 and 4 represents one set of workloads. The baseline number of systems and energy is RtH. We can see (Figure 3) that LFM policy requires an average of 85% more systems with up to 113% more systems on the worst case scenario. Fopt yielded 45% more systems with up to 80% compared to RtH. In our system, LFM policy is 12% (33% max) more energy efficient then RtH. Fopt yielded 16.4% lower energy average (33% max).

## VI. HIGH LOAD WORKLOADS

Section IV evaluated lightly loaded datacenter that could meet SLA requirements while running at low voltage and frequency. While running at higher load, the systems need to run at higher frequencies in order to meet SLA. We implement the algorithm in Figure 2 in the simulator. In this study we assumed 0.75% load out of full capacity datacenter running at the maximum frequency. We used the same random workload selection procedure described in Section IV. In this section, the dispatched distributes the workloads randomly among systems. It than applies a control loop tracing the delivered SLA and notifies the individual systems weather to increase or decrease frequency until desired SLA is maintained. We implemented three different leaf side policies. All the policies start with the Fopt point calculated by the EARtH algorithm. Running at a lower frequency is not energy efficient. Each system may have a different frequency as a function of the system type and the workload characteristics. The leaf side policies differ in gain (figure 2). The policies evaluated are:

- EARtH: Gain = 1. Base algorithm, increment or decrement frequency without applying any additional information
- EARtH + SCA: Gain = SCA. If higher frequency is needed, this policy favors the more scalable workloads. Increasing frequency comes at a cost of increased power and energy. The scalable workloads are expected to achieve a shorter run time for the same frequency change. This both contributes to SLA and cost less energy due to the shorter run time.
- EARtH + dE/dSLA: Similar rational to the above policy. The workload that gains the most performance for the lowest energy cost is favored.

The power management algorithm at the leaf can calculate the gain from the EARtH algorithm parameters. Figure 5 and Tale I summarize the energy gain of the above algorithms relative to RtH. Baseline EARtH algorithm achieves 9.1% average (19.6% max) energy savings compared to RtH at the same SLA. Adding SCA knowledge to the leaf algorithm improves average energy savings to 10.5% (max 20%) while accounting for dE/dSLA slightly improves energy savings by 0.6%.

## VII. HETEROGENEOUS DATACENTER

In a practical datacenter, the CPU systems are not identical. Systems' installation evolves over time; some systems are replaced due to maintenance etc. This heterogeneity opens the opportunity to assign the workloads to the right systems and further improve the energy efficiency at datacenter scale. Different systems behave differently with DVFS [2]. In the previous studies (Section IV and V) the central management algorithm and the dispatcher randomly picked systems without any preference, whereas the leaf side algorithm accounted for the system and workload characteristics (EARtH algorithm). Many parameters can impact the runtime energy efficiency. For example, the characteristics of some CPU2006 workloads have changed significantly over two subsequent generations of Intel Core products [15]. Workload-dependent parameters like memory access patterns, code and data foot prints interact differently with microarchitecture parameters like cache sizes or prefetchers.

Performing the best workload to system match at run-time is a complex task. We evaluated a single parameter matching using SCA. We assumed that a higher scalability workload will benefit the most from more capable system and contribute to the overall throughput and save energy due to a reduced runtime. In this study we evaluated a datacenter consisting of two types of systems. High power, high performance and low power lower performance systems. More systems' details are available in [2]. Same Monte Carlo simulator is used to pick a set of individual workloads at each run. The individual workloads were sorted by scalability and dispatched in that order i.e. highest scalability workload is dispatched to the high performance system first, and so on in a decreasing order. The average SCA over the entire workload run need to be collected and stored as metadata about the workloads. Dispatching according to SCA improved average energy savings to 16% with max of 21.7% (Figure 5 and Table 1)

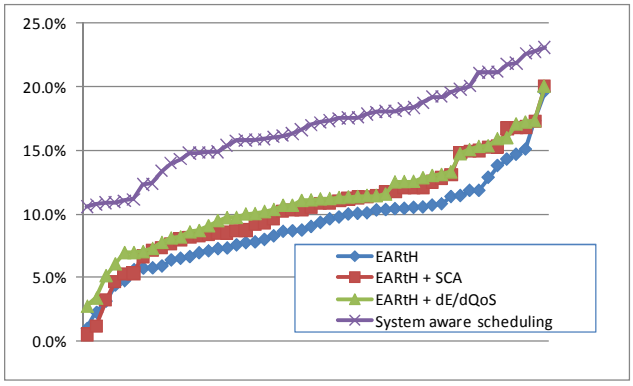


**Figure 5: Energy savings of different leaf management policy and heterogeneous datacenter scheduling algorithm**

TABLE 1

| | EARtH | EARtH + SCA | EARtH + dE/dSLA | Heterogeneous Scheduling |
|---|---|---|---|---|
| Average | 9.1% | 10.5% | 11.1% | 16.0% |
| Max | 19.6% | 20.0% | 20.0% | 21.7% |
| Min | 1.0% | 0.5% | 2.7% | 10.2% |

## VIII. SUMMARY AND CONCLUSIONS

We evaluated a hybrid energy management system for heterogeneous datacenter. Highly dynamic workload profile force to keep enough systems online to meet required QoS. Most of the time the workload is lighter and energy management

techniques such as DVFS can be applied, while meeting QoS requirements. A central dispatch and control algorithm distributes jobs to the leaf systems and controls the overall QoS. Distributed power management on each system optimizes the energy consumption of the system. We have shown that the proposed partition offers a simple and yet powerful energy management tool. Local knowledge of the workload and system characterizations is utilized at the leaf systems with minimal communication and central management.

We evaluated the algorithm on real state of the art Intel® Core™ processors systems. An average of 11% energy with up to 20% energy savings have been measured. Utilizing minimal central knowledge about the leaf systems type and the workload conserved an average of 16% energy with up to 21.7% energy.

## ACKNOWLEDGMENT

This work was partially funded by Intel Collaborative Research Institute - Computational Intelligence

## REFERENCES

[1] Advanced Configuration and Power Interface (ACPI) Specification, [online], Available: www.acpi.info/

[2] E. Rotem, R. Ginosar, U. C. Weiser, A. Mendelson, " Energy Aware Race to Halt: A Down to EARtH Approach for Platform Energy Management," IEEE Computer Architecture Letters, vol. 99

[3] Mars, J.; Lingjia Tang; Hundt, R., "Heterogeneity in "Homogeneous" Warehouse-Scale Computers: A Performance Opportunity," Computer Architecture Letters , vol.10, no.2, pp.29,32, July-Dec. 2011

[4] Jason Mars and Lingjia Tang. 2013. Whare-map: heterogeneity in "homogeneous" warehouse-scale computers. In Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA '13).

[5] Meisner, D.; Sadler, C.M.; Barroso, L.A.; Weber, W.; Wenisch, T.F., "Power management of online data-intensive services," Computer Architecture (ISCA), 2011 38th Annual International Symposium on , vol., no., pp.319,330, 4-8 June 2011

[6] Jeffrey Dean and Luiz André Barroso, "The Tail at Scale," Communications of the ACM, Vol 56, Issue 2, February 2013

[7] Aman Kansal, Jie Liu, Abhishek Singh, Ripal Nathuji, and Tarek Abdelzaher. 2010. Semantic-less coordination of power management and application performance. SIGOPS Oper. Syst. Rev. 44, 1 (March 2010)

[8] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. SIGARCH Comput. Archit. News 35, 2 (June 2007), 13-23.

[9] L. A. Barroso. The price of performance: An economic case for chip multiprocessing. ACM Queue, 3(7), September 2005.

[10] David Meisner, Brian T. Gold, and Thomas F. Wenisch. 2009. PowerNap: eliminating server idle power. In *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems* (ASPLOS '09). ACM, New York, NY, USA

[11] B. Lin, A. Mallik, P. A. Dinda, G. Memik, and R. P. Dick. Power reduction through measurement and modeling of users and cpus: summary. pages 363–364, 2007.

[12] Frederico Alvares de Oliveira, Jr. and Thomas Ledoux. 2011. Self-management of applications QoS for energy optimization in datacenters. In *Green Computing Middleware on Proceedings of the 2nd International Workshop* (GCM '11),

[13] www.spec.org/power_ssj2008/results/ (Server results)

[14] Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat, and Ronald P. Doyle. 2001. Managing energy and server resources in hosting centers. In *Proceedings of the eighteenth ACM symposium on Operating systems principles* (SOSP '01). ACM, New York

[15] Yasin, Ahmad. "A top-down method for performance analysis and counters architecture." In *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, pp. 35-44. IEEE, 2014.