Title

Peak Efficiency Aware Scheduling for Highly Energy Proportional Servers

ISBN

9781467389471

Author

Wong, Daniel

Publication Date

2016-06-01

DOI

10.1109/isca.2016.49

Peer reviewed

Peak Efficiency Aware Scheduling for Highly Energy Proportional Servers

Daniel Wong
Systems Optimization and Computer Architecture Laboratory (SoCal)
Department of Electrical and Computer Engineering
University of California, Riverside
dwong@ece.ucr.edu

***Abstract*—Energy proportionality of data center severs have improved drastically over the past decade to the point where near ideal energy proportional servers are now common. These highly energy proportional servers exhibit the unique property where peak efficiency no longer coincides with peak utilization. In this paper, we explore the implications of this property on data center scheduling. We identified that current state of the art data center schedulers does not efficiently leverage these properties, leading to inefficient scheduling decisions. We propose Peak Efficiency Aware Scheduling (PEAS) which can achieve better-than-ideal energy proportionality at the data center level. We demonstrate that PEAS can reduce average power by 25.5% with 3.0% improvement to TCO compared to state-of-the-art scheduling policies.**

Keywords-servers; energy efficiency; scheduling;

I. Introduction

Energy efficiency of data centers is a first-class design constraint. Historically, energy efficiency improvements of data center servers has focused on peak and idle utilization, neglecting the low, but non-zero, utilization region where data centers spend the majority of the time. This critical problem led to the call for energy proportional computing [1]; ideally, servers should consume power proportional to its utilization. Energy proportional computing has been a major area of research over the past decade. The main goal of energy proportional computing is to improve energy efficiency in this low, but non-zero, utilization region.

Energy proportional innovations have targeted improvement at the component-level [2]–[5], server-level [6]–[9], and cluster-level [10]–[15]. Component-level innovations typically leverage techniques such as dynamic voltage frequency scaling (DVFS) and power gating. Server-level techniques have explored full-system low power modes [6], [7] and heterogeneity [9]. Cluster-level techniques typically focus on concentrating workloads into a subset of server, in order to turn off idle servers [15], [16].

Over the last decade, energy proportionality has made significant strides to the point where near-ideal servers are becoming common. Figure 1 plots the energy proportionality of 426 servers, from published SPECpower benchmark results [17], from December 2007 to September 2015. Energy proportionality is measured by the energy proportionality (EP) metric [9], where a value of 0 represents an energy disproportional server with constant power regardless of utilization, and 1 represents an ideally-equivalent energy proportional server. From 2008 to 2012, energy proportionality improved drastically from an average of ~0.3 to ~0.8. Since 2012, the observed best energy proportionality has topped out at 1.0, with the range and average energy proportionality of new servers approaching ideal.

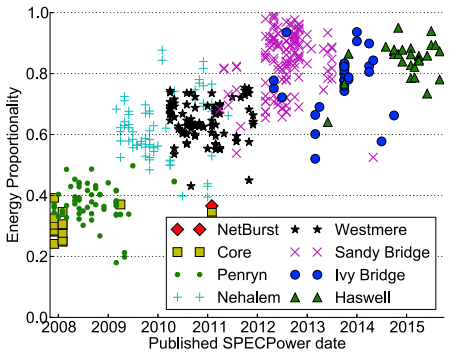


Figure 1. Energy proportionality experienced significant improvement between 2008 and 2012. Since then energy proportionality has leveled off with near-ideal servers becoming the norm.

During the era of poor server energy proportionality, cluster-level scheduling techniques, such as dynamic capacity management [15], were critical in providing high cluster-level energy proportionality. Such techniques are highly effective in improving cluster-wide energy proportionality by packing work into a subset of servers in order to turn off idle servers. Dynamic capacity management techniques are able to mask the poor energy proportionality of individual servers. With the emergence of high energy proportional servers, it has been argued that cluster-level scheduling techniques may no longer play a significant role in improving cluster-wide energy proportionality [18]; simply distributing requests uniformly can achieve better energy proportionality than with dynamic capacity management. The ability for dynamic capacity management to mask individual server's energy proportionality, which once was a beneficial property, is now the limiting factor.

In this paper, we demonstrate that cluster-level scheduling techniques will continue to play a significant role in the high energy proportionality era. With the leveling-off of energy proportionality in servers, the onus is on cluster-level scheduling techniques to continue advancing energy

1063-6897/16 $31.00 © 2016 IEEE
DOI 10.1109/ISCA.2016.49

proportionality. Specifically, this paper makes the following contributions:

Energy Efficiency Properties For Current and Future Energy Proportional Servers: In section II, we identify that current and future energy proportional servers exhibit peak energy efficiency at non-peak utilization. Current servers observe peak energy efficiency at ~60% utilization. By deriving a pareto-optimal frontier for energy proportionality, we identified a best possible achievable EP level of 1.35. The utilization level for peak energy efficiency grows even more profound with these super energy proportional servers, occurring at ~50% utilization.

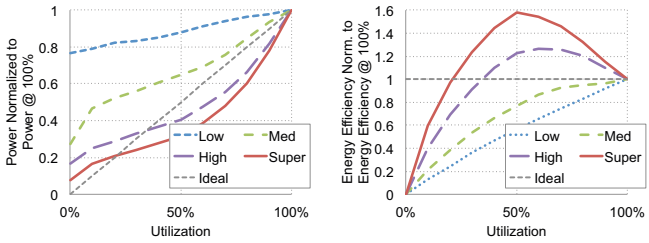
Strengths and Limitations of Packing and Uniform Scheduling Policies: In section III, through analytical analysis, we determine the beneficial and harmful properties of packing and uniform cluster-level scheduling policies. Specifically, we identified that cluster-level schedulers for highly energy proportional servers should 1) expose underlying server's energy proportionality, 2) provide sustained energy efficiency at all utilization levels, and 3) be aware of underlying server's unique peak energy efficiency properties.

Peak Efficiency Aware Scheduling: In section IV, we present the Peak Efficiency Aware Scheduler (PEAS) for highly energy proportional servers. PEAS consist of a global load scheduler and local server profiling daemon. The profiler is able to dynamically capture the energy efficiency profile of the server, which the global scheduler will then utilize as a heuristic for scheduling. In section V, we demonstrate that PEAS can achieve better-than-ideal energy proportionality, sustain a high level of energy efficiency, and significantly outperform state-of-the-art dynamic capacity management and uniform scheduling techniques. Furthermore, we show in section VI that PEAS can improve TCO by 3.0%.

Maximizing Compute Capacity: In section VII, we demonstrate how PEAS can maximize compute capacity under power capping. We show that PEAS can provide up to 20% more compute capacity for a given power budget. In addition, PEAS can provide greater compute capacity than any other scheduling policy, under any given power budget.

II. Energy Efficiency Properties For Current and Future Energy Proportional Servers

In this section, we will explore the energy efficiency properties of current and future energy proportional servers. Figure 2 depicts the utilization vs power curve (also called the energy proportionality curve) and the energy efficiency curve (energy efficiency is defined as operations per watt) of servers with varying levels of energy proportionality. Four servers are presented: "Low" depicts a server with EP=0.24 [18], "Med" depicts a server with EP=0.73, and "High" depicts a server with EP=1.0 [17]. "Super" depicts a hypothetical server which we introduce later this section.



(a) Energy Proportionality Curves (b) Energy Efficiency Curves

Figure 2. Energy efficiency properties of servers with Low (EP=0.24), Medium (EP=0.73) and High (EP=1.0) levels of energy proportionality. The Super energy proportional case (EP=1.2) represents a hypothetical server presented in section II-B.

From the energy proportionality curve in figure 2a, we notice two trends. First, as energy proportionality of servers increase, the utilization vs power curve becomes less linear. A major contributor to the improvement of energy proportionality is processor power management such as DVFS [19], [20]. The non-linearity can be attributed to the cubic reduction in processing power with linear reduction in performance for DVFS and dynamic overclocking (such as Intel TurboBoost) [19], [21]. With a linear power curve, the energy efficiency curve tend to have a linear tradeoff with utilization and energy efficiency, as demonstrated by the "Low" curve in figure 2b. As the power curve becomes non-linear, so does the energy efficiency curve. This non-linear power curve, leads us to our second observed trend; the point of peak efficiency is no longer at peak utilization. Highly energy efficient servers now typically exhibit peak energy efficiency at ~60% utilization. Not only that, but we observe that the level of peak energy efficiency is also 1.27x that of the energy efficiency at peak utilization. Many traditional cluster-level energy proportionality techniques assume that peak efficiency coincides with peak utilization. For example, a goal of dynamic capacity management is to pack servers to increase utilization as much as possible [15]. This assumption and subsequent goal may be more damaging than beneficial for current and future high energy proportional servers. In this section, we will show that as energy proportionality improves, it will be even more necessary to have scheduling policies that are aware of peak energy efficiency utilization.

A. The Energy Proportionality Limit

In order to build a case for a novel scheduling policies for high energy proportional servers, we will first identify the practical upper limits of energy proportionality. This limit study will allow us to demonstrate and quantify what opportunities existing scheduling policies are leaving off the table.

Based on historical SPECpower results, we derive a pareto-optimal frontier to provide a tradeoff between dynamic range (DR), linear deviation (LD), and energy pro-

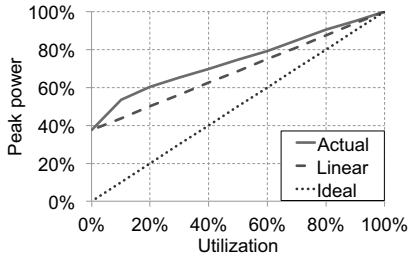


Figure 3. Energy proportionality curves used for calculating dynamic range (DR), linear deviation (LD), and energy proportionality (EP) metrics.

portionality (EP). For completeness, we present these metrics [9] below.

Dynamic range (DR) is defined as:

$$DR = \frac{P_{100} - P_0}{P_{100}} \quad (1)$$

where P_{100} is the peak power at 100% utilization and P_0 is the idle power at 0% utilization.

Linear deviation (LD) is defined as:

$$LD = \frac{A_{actual}}{A_{linear}} - 1 \quad (2)$$

where A_{actual} and A_{linear} is the area under the server's actual and linear energy proportionality curve, respectively. These curves are shown in figure 3. A server is considered *linearly energy proportional* if LD = 0, *superlinearly energy proportional* if LD $>$ 0, and *sublinearly energy proportional* if LD $<$ 0. LD measures how linear the energy proportionality curve is.

Energy proportionality (EP) is defined as:

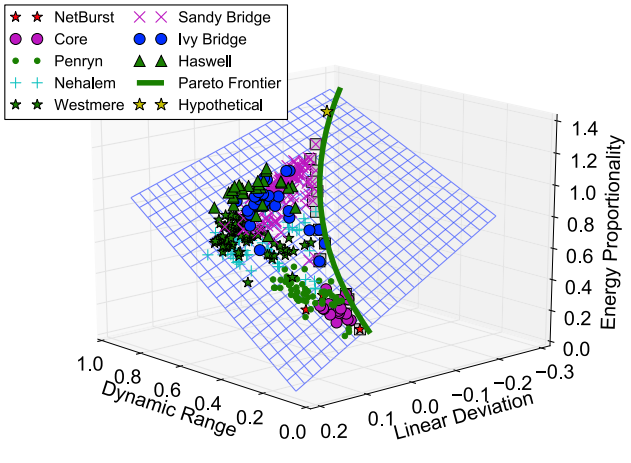
$$EP = 1 - \frac{A_{actual} - A_{ideal}}{A_{ideal}} \quad (3)$$

where A_{actual} and A_{ideal} is the area under the server's actual and ideal energy proportionality curve, respectively. An ideal energy proportional server would have EP = 1.

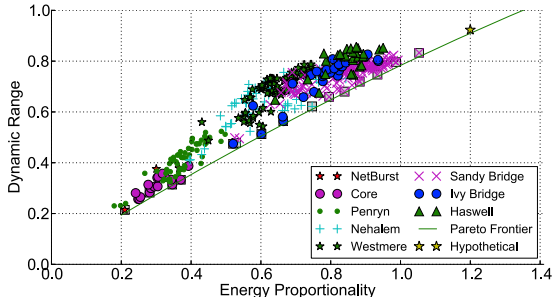
Figure 4a shows the results of the Pareto-optimal frontier analysis. We plot 426 servers from SPECpower results. It turns out that all of these data points fall on a plane in 3D-space as indicated by the wireframe plane. This plane exists because there exists a mathematical relationship between DR, LD, and EP [22]. This plane is given by:

$$EP = 2 - (2 - DR)(LD + 1) \quad (4)$$

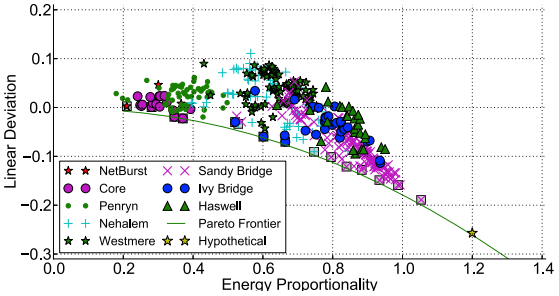
Since all points fall on this plane, it simplifies our 3-dimensional Pareto frontier to a 2-dimensional frontier on the plane. Pareto-optimal server designs that fall on the Pareto frontier are indicated by an enclosing gray box. These server designs were identified by iterating through all servers, sorted by ascending DR, and identifying subsequent servers with better DR, LD, and EP than the last identified Pareto-optimal point. We then fit a quadratic polynomial, broken down into two components DR(*ep*) and LD(*ep*), to these points using the least square regression method such that all design points are enclosed by the frontier. In DR(*ep*) and LD(*ep*), *ep* is the energy proportionality of a measured server in SPECpower. The Pareto frontier represents optimal



(a) DR/LD/EP Tradeoff Frontier



(b) Dynamic Range/Energy Proportionality Frontier



(c) Linear Deviation/Energy Proportionality Frontier

Figure 4. Pareto-optimal frontier derived from historical SPECpower results labeled with processor generation.

design point in terms of their respective components. In other words, a design point that falls on the Pareto frontier has a component that is not dominated by another design point.

For simplicity in visualization and comprehension, we decompose the 3-dimensional DR/LD/EP tradeoff into two 2-dimensional DR/EP and LD/EP tradeoffs. Each decomposed tradeoff represents a component of the DR/LD/EP tradeoff.

Figure 4b shows the DR/EP server design space. The Pareto-optimal frontier for the DR/EP tradeoff is given as:

$$DR(ep) = -0.1025 * ep^2 + 0.8594 * ep + 0.026 \quad (5)$$

Figure 4c shows the LD/EP server design space. The servers along this Pareto frontier are the same points as those

in figure 4b. We identified LD(ep) in a similar manner as before through a fitted quadratic polynomial. The Pareto-optimal frontier for the LD/EP tradeoff is given as:

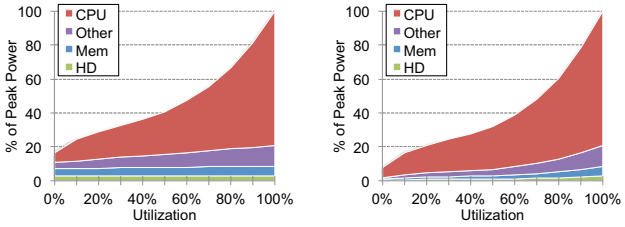
$$LD(ep) = -0.197 * ep^2 + 0.022 * ep - 0.004 \quad (6)$$

Using these derived Pareto frontiers, we are now able to identify a limit for energy proportionality. The possible values of dynamic range are $[0, 1]$. Given this and DR(ep), we see that if we can achieve a server design with dynamic range of 1, then we can achieve an EP of 1.35, the maximum EP values within the pareto frontier.

B. Limit Study: A Hypothetical Super EP Server

Our prior limit study provided us with a maximum attainable EP value. The value by itself does not give significant insight into the energy efficiency properties of future super energy proportional servers. With this in mind, we present a case study on a hypothetical super energy proportional server to tease out energy efficiency properties. Furthermore, this hypothetical case study can give confidence to the derived Pareto frontiers by demonstrating that radically different servers with aggressive energy proportionality profiles still falls within the Pareto frontier.

Figure 5a shows the component level EP curves of the highest EP server reported from SPECpower. This server has EP of 1.05 with DR of 0.833 and LD of -0.1888 (Upper-right most Pareto-optimal point in figure 4b and lower-right most pareto-optimal point in figure 4c). In order to estimate the component-level power breakdown, we collect empirical component-level power breakdown by instrumenting a highly energy proportional server modeled after a Dell R520 server from SPECpower results. We instrumented each individual component by intercepting the power rails and measuring the current with LTS 25-NP current sensors. The outputs of the current sensors are sampled at 1kHz using a DAQ and logged using LabView. To measure CPU power, we inserted a current sensor in series with the 4-pin ATX power connector. To measure memory power, we inserted a current sensor in series with pins 10 and 12 of the 24-pin ATX power connector which supplies power to the motherboard [23]. To measure the power of the hard drive, we inserted a current sensor in series with the hard drive backplane power connector. Our empirical component-level power breakdown is similar to other studies [20].



(a) High EP server (b) Hypothetical Super EP server

Figure 5. EP curve with component-level breakdown of a High EP server with proportional processor (a) and hypothetical server with all proportional components (b).

Figure 5a clearly shows that CPU is the most proportional component in the server, while other components nearly consume the same amount of power regardless of utilization. Now let's assume that all of these non-compute components are just as energy proportional as the processor. This would result in a hypothetical server with an EP curve as shown in figure 5b. This radically different server would represent the case where if all future EP improvements are contributed by non-compute components. This hypothetical server would have an EP of 1.20. By having all components being as energy proportional as the processor, the dynamic range has improved to the point where the server would only consume 7.7% of the peak power at idle. Despite this server's aggressive energy proportionality profile of all components, it still falls along the Pareto-optimal frontier. Therefore, we're confident that our derived Pareto frontier will still hold true for future server platforms with aggressive energy proportionality mechanisms.

Observations: This server is labeled as "Super" in figure 2. This server represents a possible practical limit of server energy proportionality. From the energy efficiency curve of this super energy proportional server, it is clear that the observed trends in server energy efficiency is even more apparent. The peak energy efficiency is now achieved at $\sim$ 50% utilization, with the peak efficiency being 1.58x the efficiency at peak utilization. Clearly, as server energy proportionality improves, there is a growing need for peak efficiency aware scheduling policies.

III. Strengths and Limitations of Packing and Uniform Scheduling Policies

To design a peak efficiency aware scheduling policy for high energy proportional servers, we must first understand the strengths and limitations of existing Packing and Uniform scheduling policies in order to identify desirable properties. Our findings are summarized in table II. Specifically, we identify that cluster-level schedulers for highly energy proportional servers should 1) expose underlying server's energy proportionality, 2) provide sustained energy efficiency at all utilization levels, and 3) be aware of the underlying server's unique peak energy efficiency properties.

A. Best-case Cluster-wide EP Analysis

We will use best-case cluster-wide energy proportionality analysis [18] to identify the benefits and limitations of existing scheduling policies. This analysis relies on *best-case cluster-wide energy proportionality curves*, which are defined as the idealized theoretically achievable cluster-wide energy proportionality if there are no power mode transition penalties or workload migration penalties.

The authors in [18] made an empirical case to forego cluster-level packing techniques in favor of uniform load

Variable	Description
$f(util)$	Server Util. vs Power Curve
$g(util)$	Cluster-wide Util. vs Power Curve
x	Cluster-wide utilization
N	Number of Servers in Cluster
P	Power at Peak Utilization, $f(100)$
U	Peak efficiency utilization

Table I
VARIABLE DEFINITIONS FOR BEST-CASE CLUSTER-WIDE ENERGY PROPORTIONALITY ANALYSIS.

balancing as server energy proportionality improves. Since the load balancing cases covered in [18] were simple, they were able to reason about the best-case cluster-wide energy proportionality curve. Here, we will formalize best-case cluster-wide energy proportionality analysis in order to analyze more complicated load balancing schemes to draw insight into the beneficial properties of scheduling policies for future highly energy proportional systems.

We define the variables in table I to aid our description. $f(util)$ and $g(util)$ represents the utilization vs power curve (also called the EP curves) for individual servers and the entire cluster, respectively, and x is the utilization, which ranges from 0% to 100%. With uniform scheduling, all servers within the cluster will be operating at the same utilization as the cluster. Therefore the best-case cluster-wide utilization vs power curve is simply:

$$g(x) = f(x) \times N \tag{7}$$

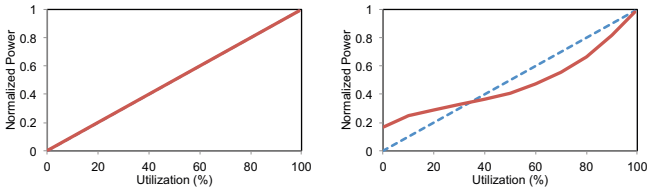
Clearly, the cluster-wide energy proportionality curve is directly proportional to, and exposes, the individual server's energy proportionality curve. This was observed as beneficial as server-level energy proportionality improvements directly translate to cluster-level improvements.

For the case of Packing load balancing, it is assumed that each server can be packed until peak utilization. In this scenario, we will not turn on and load a new server unless all the available servers are completely packed already. Therefore, the best-case cluster-wide EP curve resembles steps where each step is shaped like the individual server's utilization vs power curve. The best-case cluster-wide EP curve for packing scheduling is:

$$g(x) = \left\lfloor \frac{x}{100} \times N \right\rfloor \times P + f\left(\left(x \bmod \frac{100}{N}\right) \times N\right) \tag{8}$$

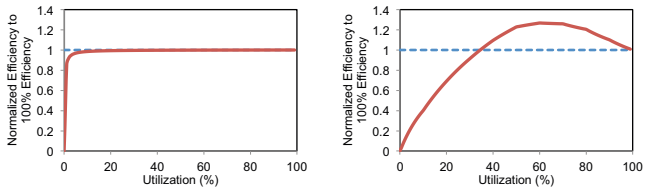
The first term in the equation represents the impact of turning on a subset of servers (N) to operate at peak utilization, and the second term represents the power consumption of a partially loaded server. For instance, if the cluster size is 10 servers and the cluster utilization is 45%, then we would turn on four servers to operate them at 100% utilization, and the last server will be running at 50% utilization. Clearly, the first term dominates the second term, highlighting how Packing load balancing mask the underlying server's energy proportionality profile.

Figure 6a and 6b shows the best-case cluster-wide energy proportionality curve for these two different load balancing schemes, with EP=1.05 servers. The x-axis is the cluster-wide utilization, while the y-axis is the cluster power normalized to the power at 100% utilization. The solid red line represents the cluster-wide EP curve, while the dotted blue line represents the ideal linearly energy proportionality curve. Using uniform load balancing (figure 6b), the cluster-wide EP is 1.05, equivalent to the EP of the servers that make up the cluster. In an idealized case of packing, only the minimum number of servers required to meet a certain load is active, with all other servers off. Therefore, the number of servers required is proportional to the utilization, essentially allowing the cluster to have linearly ideal energy proportionality with EP of 1 as $N \to \infty$.



(a) Packing (b) Uniform

Figure 6. Best case cluster-wide energy proportionality curve for various cluster-level load balancing schemes (red solid line). The blue dotted line represents linearly ideal energy proportionality.



(a) Packing (b) Uniform

Figure 7. Cluster-wide energy efficiency curves

Figure 7 shows the energy efficiency across utilization levels for the uniform and packing scheme. The x-axis represents the cluster-wide utilization, while the y-axis is the cluster-wide energy efficiency normalized to the energy efficiency at 100% utilization. The solid red line represents the efficiency of the best-case load balancing scheme, and the dotted blue line represents the efficiency of an ideal linear energy proportional cluster. Most notably, the packing scheme is able to sustain peak cluster-wide efficiency across all utilization levels (figure 7a). This is a desirable property since the cluster can run at high efficiency regardless of utilization. Unfortunately, uniform load balancing can only operate at high efficiency within a certain "sweet spot".

To summarize, schedulers for highly energy proportional servers should 1) expose underlying server's energy propor-

Properties	Packing	Uniform	PEAS
Expose EP		✓	✓
Sustain Energy Eff.	✓		✓
Peak Eff. Aware			✓

Table II
DESIRED PROPERTIES FOR SCHEDULER FOR HIGH ENERGY PROPORTIONAL SERVERS

tionality so server-level EP improvements can translate to cluster-level EP improvement, 2) provide sustained energy efficiency at all utilization levels, and 3) be aware of the underlying server's unique peak energy efficiency properties.

IV. PEAS: Peak Efficiency Aware Scheduling

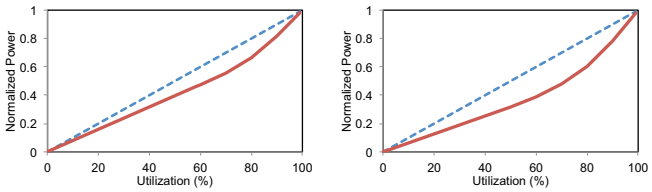
In this section, we present the Peak Efficiency Aware Scheduler (PEAS) for highly energy proportional servers. Drawing from our insight in prior sections, PEAS exhibits all the desirable properties required for scheduling of highly energy proportional servers. PEAS consist of a global peak energy efficiency aware scheduler and a local per server energy efficiency profiling daemon. The profiler is able to dynamically capture the energy efficiency profile of each individual server, which the global scheduler will then utilize as a heuristic for scheduling. Rather than naively packing each server until peak utilization, or uniformly spreading out the load to all servers, it may be more efficient to pack servers up until their peak efficiency point. Then once all servers are operating at their peak efficiency point, if more requests come, issue those request uniformly. The insight here is to quickly get servers to the point of peak efficiency, and then once they reach that point, move away from that point as slowly as possible. The PEAS scheduler will have a best-case EP curve of:

$$g(x) = \begin{cases} \left\lfloor \frac{x}{U} \times N \right\rfloor \times f(U) + f\left(\left(x \bmod \frac{U}{N} \right) \times \frac{100N}{U} \right) & x < U \\ f(x) \times N & x \geqslant U \end{cases} \quad (9)$$

This is a piecewise function where the scheduler packs below the peak efficiency utilization, and uniformly schedules above the peak efficiency utilization. Note that the first part of the piecewise function is a generalized version of the packing equation (8), where packing load balancing has $U = 100$. Essentially, packing load balancing provides the optimal best-case cluster-wide energy proportionality when peak efficiency coincides with peak utilization. Another interesting note is that if $U = 0$, then this equation reduces to the uniform load balancing equation (7). In this scenario, $U = 0$ represents the case where peak efficiency is at 0% utilization, or more concretely, that of a server where it has peak efficiency across all utilization levels (efficiency remains constant at all utilization). An example server with this property is that of an ideal linearly energy proportional server. Peak efficiency scheduling is able to capture the behavior of both packing and uniform load balancing, and therefore provide the optimal best-case cluster-wide energy proportionality for any given peak efficiency utilization.

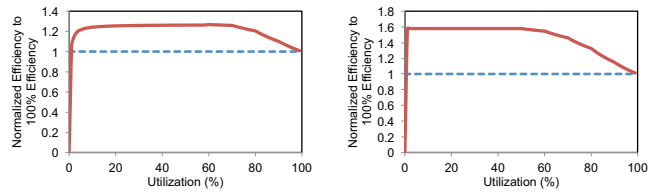
A. A Case for PEAS

We will now present a case for PEAS with a high energy proportionality server (EP=1.0) and a super energy proportional server (EP=1.2, derived from section II-B). The energy and efficiency curves for these servers are shown in figure 2. Figures 8 shows the best-case EP curves for PEAS. With



(a) EP=1.0 (b) EP=1.2

Figure 8. EP curve for Peak Efficiency Aware Scheduling

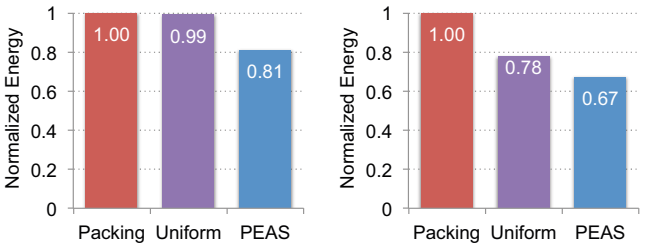


(a) EP=1.0 (b) EP=1.2

Figure 9. Energy efficiency curve for Peak Efficiency Aware Scheduling

high energy proportional servers, PEAS is able to achieve a cluster-wide EP of 1.16. With super energy proportional servers, PEAS is able to achieve a cluster-wide EP of 1.26. As the underlying server's EP improve, so does the cluster-wide EP under PEAS. This demonstrates the desired property of exposing the underlying server's EP. Figures 9 shows the energy efficiency curves for PEAS. As with uniform scheduling, PEAS achieve a peak efficiency level greater than that of efficiency at peak utilization. PEAS achieve peak efficiency levels of 1.26x and 1.58x the efficiency at 100% utilization, for EP=1.0 and EP=1.2, respectively. Unlike uniform scheduling, PEAS is able to sustain a high level of energy efficiency across all utilization levels. By packing requests to the peak efficiency utilization, PEAS captures this desired property from packing scheduling.

Figure 10 shows the potential energy savings for PEAS compared to packing and uniform scheduling. We assume a workload utilization distribution similar to that observed by Google [1]. The energy of each scheduling policy is normalized to the energy of packing scheduling. In figure 10a we show the energy savings using high energy proporitonal servers of EP=1.0. Under this scenario, both packing and uniform scheduling achieves similar cluster-wide EP, resulting in near-identical energy savings. With PEAS, the cluster-wise EP improved to 1.16, translating to 19% energy savings. In figure 10b we show the results for a super energy proportional server of EP=1.2. Since uniform scheduling exposes the underlying server's EP, it is able to achieve 22% energy savings with respect to packing scheduling, which is limited to a cluster-wide EP of 1.0. With PEAS, cluster-wide EP improved to 1.26, translating to overall energy savings of 33% compared to packing and 14% relative to uniform scheduling. Clearly, as server energy proportionality improves, the need for PEAS will become even more apparent.



(a) EP=1.0 (b) EP=1.2

Figure 10. Energy savings for various scheduling policies assuming Google workload distribution [1]. As server EP improves, the need for PEAS becomes more apparent.

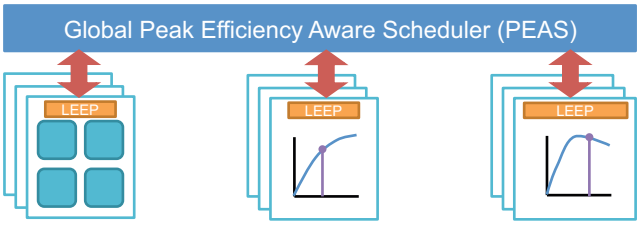


Figure 11. The PEAS runtime framework consists of the LEEP profiler and the PEAS global scheduler. Individual server's EP curve and utilization are used as heuristics for PEAS scheduling decisions.

B. Design of PEAS

We will now discuss the design and implementation of PeaK Efficiency Aware Scheduling. Figure 11 shows an overview of the PEAS runtime system. PEAS consists of two major components, a per server local energy efficiency profiler (LEEP) and a global peak efficiency aware scheduler (PEAS). During runtime, LEEP will profile the dynamic energy proportionality curve of the server. This information, along with the server's utilization, periodically updates the global scheduler. The global PEAS scheduler will then use this information in order to schedule in the most energy efficient way possible.

1) Local Energy Efficiency Profiler (LEEP): PEAS requires the energy efficiency curves for each server in the cluster. The energy efficiency profile of each server can vary dynamically due to differences in workload and individual server configuration [24]. It would be impractical for PEAS to make scheduling decisions based on off-line profiled energy efficiency curves for various server and workload configurations as the overhead required to explore such a large design space is significant. To this end, we develop an online energy efficiency profiler, LEEP, which dynamically captures the dynamic energy efficiency curve of the individual server configuration and workload.

LEEP is implemented as a daemon, which periodically samples the utilization and power consumption of the server, and logs the energy efficiency. We found empirically that sampling every 1 second can provide sufficient granularity to provide an accurate energy efficiency curve. Since the power level can vary at a specific utilization level, the profiler will keep track of a cumulative moving average for each utilization level. For example, a server running at 50% utilization may consume differing power levels due to the usage of different hardware resources. For example, floating point units consume more power than integer units. By storing a cumulative moving average for each utilization level, we can estimate the average power consumption at a given utilization level. The peak efficiency point of a server is initialized at 100% utilization, therefore, the server by default will pack requests as much as possible, enabling the server to observe a wide range of utilization levels. This enables the server to deduce a peak efficiency point based on observed utilization points and interpolating for non-observed utilization points. The profiled energy efficiency curve, and the current utilization, will then periodically update the global scheduler, which will use these profiles as a heuristic for peak efficiency scheduling.

Server utilization information requires more frequent updates compared to energy efficiency curve updates. We implemented the PEAS profiler on real servers and observed minimal profiling overheads in the order of ms. We observed that the energy efficiency curve doesn't change drastically after reaching a steady state, and therefore does not require frequent updates to the global scheduler. To minimize update overhead, we can measure the difference in energy efficiency curves between the last updated curve and the current profiled curve, and trigger an update once the efficiency curve difference reaches a certain threshold.

2) Global Peak Efficiency Aware Scheduler: The PEAS runtime relies on a global Peak Efficiency Aware Scheduler in order to make energy efficiency scheduling decisions. The PEAS scheduler utilizes the profiled information as shown in Algorithm 1.

Algorithm 1: PEAS Scheduling policy

```
1  if Periodically Update Server Profiles then
2  |   Update Server Utilization;
3  |   Update Server Energy Efficiency Profiles;
4  |   Sort Servers based on Peak Energy Efficiency;
5  end
6  if All Servers Above Peak Efficiency Utilization then
7  |   Uniform Scheduling;
8  else
   |   /* Pack to Peak Efficiency Util. */
9  |   for servers in sortedServers do
10 |   |   if Server less than Peak Efficiency Utilization
   |   |    then
11 |   |   |   Schedule requests;
12 |   |   |   break;
13 |   |   end
14 |   end
15 end
```

The scheduler periodically receive updates of the server's utilization and energy efficiency curve from the profiling daemon. In addition to taking into account the peak efficiency utilization of servers, this algorithm also captures the heterogeneity of servers. The scheduler will maintain a sorted list of the servers based on the peak energy efficiency point. When scheduling a request, the scheduler will first pack the servers that are at less than peak efficiency utilization, giving preference to the servers with higher absolute energy efficiency (ops/watt). The intuition is that the energy efficiency observed at peak efficiency utilization can vary due to heterogeneity. Therefore, running high efficiency servers at peak efficiency is better than running lower efficiency servers at peak efficiency. If no servers are running at below peak efficiency utilization, then the PEAS scheduler will simply uniformly schedule requests. This essentially captures the behavior in equation 9.

3) Multi-level PEAS: Many existing schedulers rely on server-level profiling as heuristics for global scheduling, such as temperature [25], availability of green energy [26], application performance monitoring [27] and sensitivity to application interference [28]. These prior works all experienced negligible and manageable overheads; which we also observed with LEEP profiling. Thus, we do not expect profiling overhead to be a major concern. Nevertheless, to address such challenges, we provide a multi-level PEAS runtime to minimize the scheduler overhead.

The global scheduler requires knowledge of the individual server's utilization to make informed scheduling decisions. For large number of servers, this knowledge may result in significant metadata and message passing overheads. To overcome this, we propose a multi-level scheduler design, where subsets of clusters are managed by a PEAS scheduler, which in turn are managed by a higher-level PEAS scheduler. Profiling information will then be aggregated at the subset level, and are then sent to higher levels of the scheduler, reducing message passing overheads and distributing the scheduling complexity across the different layers. At the subset level, the PEAS scheduler will report the sub-cluster utilization and a generate best-case efficiency curve to the top level PEAS scheduler.

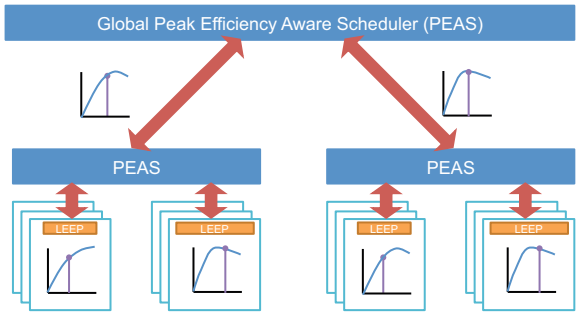


Figure 12. In multi-level PEAS, best-case efficiency curves are generated per sub-cluster, which will be used as the top-level scheduling heuristic.

V. Evaluation

A. Methodology

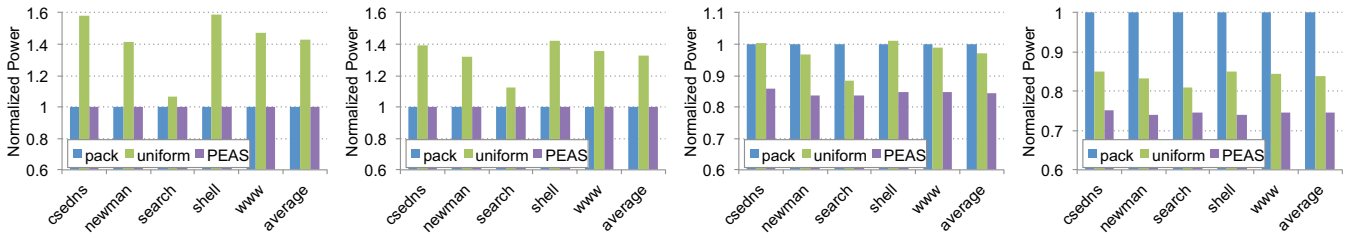
To evaluate PEAS, we use the BigHouse data center simulator [29]. BigHouse is based on stochastic queuing simulation [30], a validated methodology for simulating the power-performance behavior of data center workloads. BigHouse use synthetic arrival/service traces that are generated through empirical inter-arrival and service distributions. These synthetic arrival/service traces are fed into a discrete-event simulation of a G/G/k queuing system that models active and idle low-power modes through state-dependent service rates. Output measurements, such as 95th percentile latency, and energy savings, are obtained by sampling the output of the simulation until each measurement reaches a normalized half-width 95% confidence interval of 5%. We evaluate five workload distributions, DNS (`csedns`), Mail (`newman`), Apache (`www`), Search (`search`), and Shell (`shell`), provided with the BigHouse simulator.

We model a data center with 100 servers, each containing dual-socket 18-core processors, with two threads per core. In total, each server can handle 72 thread contexts, which is in line with many recently reported SPECpower server configurations. We modified the server power model to model four levels of energy proportionality as presented in section II. These four servers are labeled as "Low" (EP=0.24) [9], "Med" (EP=0.73), "High" (EP=1.0) [17], and "Super" (EP=1.2). Each of these servers have a wake up / sleep transition time of 20 seconds [9]. We adopt the sleep policy from AutoScale [15], where the servers will go to sleep after idling for 60 seconds. The time a server waits to go to sleep is largely insensitive between 60 seconds and 260 seconds. In order to handle utilization spikes, we always have a single idle standby servers.

B. Effect on Power

Figure 13 shows the average power across the workloads for various scheduling policies, normalized to packing scheduling. Each figure depicts a cluster with a different underlying server energy proportionality. For Low and Medium energy proportional servers, packing and PEAS scheduling perform similarly. As servers exhibit peak efficiency at peak utilization. With Low and Medium energy proportionality, packing and PEAS significantly outperforms uniform scheduling, due to its ability to mask the underlying server's poor energy proportionality. For highly energy proportional servers, this trend is reversed. For High energy proportionality, PEAS is able to reduce average power consumption by 15.5%. For Super energy proportional servers, the results are even more drastic, echoing our observations in section IV-A. Here, PEAS is able to reduce average power by 25.5%, significantly out-pacing uniform scheduling with 16.3%.

In figure 14, we explore how varying server transition time can impact average energy savings. Figure 14 shows



(a) Low EP (b) Medium EP (c) High EP (d) Super EP

Figure 13. Average power consumption

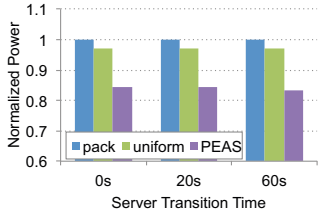
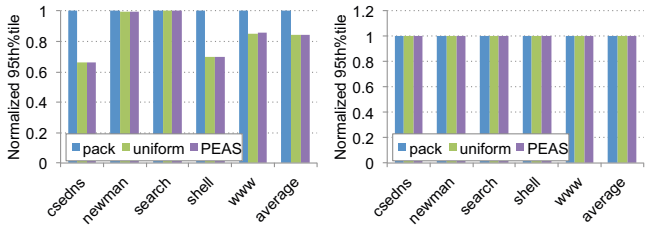


Figure 14. Average power consumption w/ various server transition times



(a) 20 second transition time (b) 0 second transition time

Figure 15. 95th percentile response time

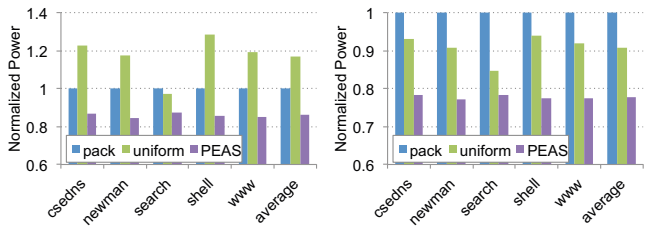
the average power consumption normalized to the power of packing scheduling for three different server transition time: 0 seconds (ideal), 20 seconds, and 60 seconds. The average power savings remains consistent regardless of the server transition time. The only time that transition time can affect power consumption is for the need to have significant numbers of servers on standby in order absorb request spikes. In our experiment, we statically set a single server to be on standby.

C. Effect on Tail Latency

Figure 15 shows the effect of various scheduling policies on 95th percentile tail latency. In all cases, uniform scheduling and PEAS observed the same tail latency. In certain cases, packing scheduling observed significantly higher tail latency. On average, PEAS observed tail latency that is 16% less than packing scheduling. The reason for this is due to how packing absorbs request spikes. The packing scheduler would pack servers until it's peak utilization, and relies on a single standby server to absorb any unexpected request spikes. If this standby server gets utilized, it will have to wake up another server, and potentially incur tail latency penalties while wait for a server to wake up. This can by avoided by assigning a larger number of standby servers in order to handle the incoming request spikes, but at the cost of energy savings. For uniform scheduling and PEAS, all servers are already on, and therefore, there is compute capacity readily available to handle any request spikes. In addition, we also explored the tail latency if servers are able to transition on/off instantaneously. In this scenario, packing scheduling can wake up a server instantly to handle request spikes and therefore experience the same tail latency as uniform and PEAS scheduling. Therefore, PEAS is able to capture the desirable properties of packing scheduling, without the difficulty of dynamically managing capacity.

D. Heterogeneous Servers

Until now, we have only considered the case of homogeneous data centers, where each server exhibits the same energy proportionality profile. It is common for data centers to be made up of servers which exhibit heterogeneous performance and power characteristic [24], [28]. To this end, we explore how servers exhibiting heterogeneous energy efficiency profiles will affect scheduling policies. We explore two scenarios, 1) a mix of 25% Low, Medium, High, and Super energy proportional servers, and 2) a mix of 50% High and Super energy proportional servers. Figure 16 presents the normalized average power of these scenarios. For the mix with all four types of energy proportional servers, uniform scheduling performs significantly worse than packing or PEAS due to its inability to mask the poor energy proportional servers. Under this scenario, PEAS is able to reduce average power by 14% compared to packing scheduling. This demonstrates that PEAS is able to extract opportunities that was being left off the table by non-peak efficiency aware scheduling policies. The case with only High and Super energy proportional servers represents the potential heterogeneous data center environment of the near future.



(a) 25% Low, Med, High, Super (b) 50% High, Super

Figure 16. Average power consumption for heterogeneous mix of servers

With only highly energy proportional servers, packing is limited as we observed earlier. Uniform scheduling is able to reduce power by 9.1%, and PEAS is able to reduce power by 22.2% relative to packing.

VI. TCO Impact

To study the effect of PEAS on TCO, we use a publicly available cost model [32]. The model assumes an 8MW power budget where facility and IT capital costs are amortized over 15 and 3 years, respectively. We assume the cost of electricity to be $0.07 [33] and a PUE of 1.45. Table III, present our cost breakdown for a highly energy proportional server which we used for empirical measurements in section II-B. We broke down cost into memory, storage, processor, and other system components, corresponding to our component-level power measurement capability. Performance is based on the SPECpower benchmark metric of ssj_ops. We present TCO on a monthly basis as Performance per TCO Dollar spent (Perf/$), an important metric in TCO-conscious data centers [34].

We modeled a data center with four different levels of server energy proportionality. For each type, we modeled Packing, Uniform, and PEAS scheduling. Figure 17 presents the percentage change to Perf/TCO$ spent per month normalized to the Packing scheduling case. With low and medium server energy proportionality, uniform scheduling consumes more power than with packing, leading to significant degradation of TCO, up to -4.6%. With highly energy proportional servers, both uniform and PEAS scheduler provides TCO benefits. With High energy proportional servers, PEAS provides a 1.8% improvement to TCO. With Super energy proportional servers, PEAS provides up to 3.0% improvement to TCO, further strengthening the case for PEAS in highly energy proportional data centers.

	Server Breakdown	
	Cost	Power(W)
Memory	$390	16
Storage	$500	7
Processor	$1440 [31]	206
Other	$775	31
Total	$3,105	260

Table III
COST BREAKDOWN OF A HIGHLY ENERGY PROPORTIONAL SERVER. COMPONENT PRICE FROM ORDER INVOICE. POWER BREAKDOWN COLLECTED THROUGH EMPIRICAL MEASUREMENTS (SECTION II-B)

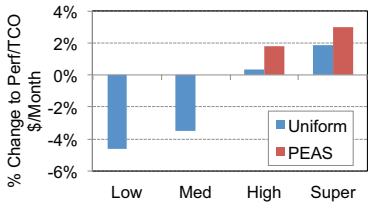


Figure 17. Percentage TCO change to Performance per TCO $ spent per month, normalized to Packing scheduling.

VII. Maximizing Compute Capacity Under Power Capping

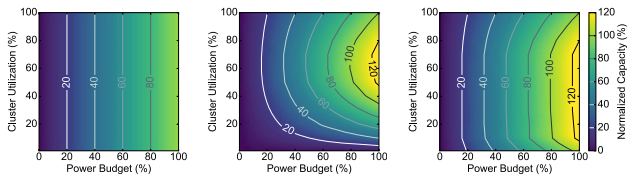
In the prior sections, we demonstrated that highly energy proportional servers can be managed in a way where we can sustain peak energy efficiency across all utilizations. Highly energy proportional server properties can impact more than just workload scheduling schemes. In this section, we will explore the implication of highly energy proportional servers on data center power capping. Specifically, we make a case that Peak Efficiency Aware Scheduling can be used to maximize compute capacity under power capping scenarios. We define compute capacity as the amount of request processing that the cluster is capable of. For example, request handling workloads measure processing capability in terms of queries per second (QPS). Therefore the compute capacity of the cluster is the QPS that the cluster is able to handle.

When data centers are provisioned based on the nameplate power of servers, there exist a significant gap between raw consumed power and available data center power. This gap allows data centers to increase the compute capacity by provisioning more servers assuming typical operating conditions, which consume less than the nameplate power. In such scenario, the data center may have enough power to supply to all of these servers. But, under peak utilization periods, there may not be enough power for all of these servers. Under power emergency scenarios, where servers are all running at peak and can violate the data center power budget, power capping is enforced. The goal is to avoid violating the data center power budget by capping the power of data center servers, while still maintaining quality-of-service requirements.

In this section, we explore under ideal conditions, how various load balancing techniques can affect the compute capacity of the data center at various power capping levels.

We model a cluster of 100 servers, with each server exhibiting a power profile with EP of 1.0. We sweep through a range of cluster power budget from 0% to 100%. The power budget is normalized to the power consumption of all the servers in the cluster. For example, with a cluster size of 100 servers each consuming 100W at 100% load, then the cluster power is 10,000W. With a power budget of 50%, then there is only 5,000W of power available to the cluster. Under Uniform scheduling, this power budget would be equally distributed to each server, where each server can only consume 50% of it's peak power. Under Packing scheduling, a 50% power budget translates to 50% of the servers being able to run at max power, while another 50% of the servers are off.

Figure 18 shows the cluster's compute capacity normalized to the compute capacity of when all servers are running at peak utilization with no power budget. The x-axis shows the power budget available to the cluster, the y-axis shows the cluster utilization and the coloring shows



(a) Packing (b) Uniform (c) PEAS

Figure 18. Compute capacity (QPS) available normalized to capacity at peak utilization with max power budget, plotted across various power budget (x-axis) and cluster utilization (y-axis).

the compute capacity normalized to the compute capacity at peak utilization with no power budget.

For Packing scheduling, the compute capacity remains the same across all cluster utilization levels at a given power budget due to all active servers running at peak (or near-peak) utilization. In this scenario, the cluster's compute capacity is solely determined by the power budget.

For Uniform load scheduling, the compute capacity is determined by both the power budget available and the utilization of the cluster. Under this scheme, the optimal compute capacity occurs when servers are running at ~60% utilization, where the server's peak efficiency occurs. Furthermore, within this operating range, the compute capacity actually exceeds the compute capacity of running at peak utilization. Unfortunately, for utilization less than 40%, Packing load scheduling outperforms Uniform load scheduling.

Figure 18c shows the compute capacity using Peak Efficiency Aware Scheduling. Most notably, the PEAS Scheduling scheme is able to provide greater compute capacity than Packing across all utilization and power budget. In certain scenarios, PEAS can achieve over 20% more compute capacity at the same power budget. This implies that servers that are active are running at optimal efficiency, requiring lesser number of active servers compared to Packing technique to handle a certain throughput. Compared to Uniform scheduling, Peak Efficiency Scheduling provides a larger utilization "sweet spot" than Uniform scheduling. Uniform scheduling achieves its best compute capacity between the 50% - 80% utilization range. In comparison, Peak Efficiency Scheduling can achieve its best compute capacity between 10% - 80%. Clearly, the varying behaviors of different cluster load scheduling schemes makes a significant difference on the compute capacity available.

VIII. Related Work

Energy proportional computing: Energy proportional computing have been a major area of research [1], [5], [6], [9], [15], [18], [35]–[37]. In addition, measurements, metrics and trends of energy proportionality have all been well studied [19], [21], [22], [35], [38]. Energy proportionality techniques has been proposed at the cluster level through workload consolidation and dynamic capacity management with the goal of "right-sizing" the amount of servers in the data center [10], [14]–[16], [36], [37], [39]. These migration-based techniques are best suited for coarse-grain workload fluctuations in the order of minutes - hours, and typically assumes a stable power budget. Our PEAS technique capture the desired behaviors of dynamic capacity management techniques, without the need for difficult "right-sizing".

At the server level, energy proportionality have been achieved through various low power modes. Inactive low power modes are activated when servers are idle, for example, sleep and hibernation. More recently, techniques were proposed to take advantage of millisecond idleness through rapid transitioning between active and idle states [6], [7], but become decreasingly effective as multi-core scaling continues. Active low power modes, on the other hand, remains effective even with multi-core scaling [8], [18]. Commercially available active low power modes includes DVFS [5], where the processor's voltage and frequency are scaled down as the server's workload decreases. While this work does not explore component and server-level energy proportionality techniques, it does rely on high energy proportional servers.

Power Capping: Power capping and power shaving are techniques utilized to limit server power consumption during such power emergencies. Power capping can be achieved through DVFS [40], [41], thread packing [42], power gating [43], and turning servers off [44]. A goal of this work is to maximize the compute capacity given a power constraint. Unlike other prior works where the goal is to turn off components to meet a power constraint without regard of the current energy efficiency of the servers, our proposed technique takes into account the peak energy efficiency profile. This enables our technique to maximize compute capacity even under a power budget.

Super Proportional Servers. Better-than-proportional servers appeared in a few prior works. [45] presented an arbitrary illustrative better-than-proportional curve, to illustrate what is desirable in WSC. [46] discussed briefly the possibility of better-than-proportional servers due to optimistic coordinated scaling of CPU and memory. Our new contribution is formalizing what a better-than-proportional server can be by identifying pareto-optimal limits for EP.

IX. Conclusion

Highly energy proportional servers exhibits unique energy efficiency properties that current state-of-the-art schedulers do not take advantage of. We identify that schedulers for highly energy proportional servers should 1) expose underlying server's energy proportionality, 2) provide sustained energy efficiency at all utilization levels, and 3) be aware of the underlying server's unique peak energy efficiency properties. We present Peak Efficiency Aware Scheduling (PEAS) which provides significant cluster-wide energy proportionality improvements over existing schemes. We demonstrate that PEAS can reduce average power consumption by 25.5%

and improve TCO by 3.0%. This work shows that even though server energy proportionality is nearing the practical limits, there still exists significant opportunities for energy proportional computing advancements.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their valuable feedback and Murali Annavaram for his guidance on preliminary versions of this work.

REFERENCES

[1] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, dec 2007.
[2] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *ISLPED*, 2007.
[3] Z. Hu, A. Buyuktosunoglu *et al.*, "Microarchitectural techniques for power gating of execution units," in *ISLPED*, 2004.
[4] Q. Deng, D. Meisner *et al.*, "Memscale: active low-power modes for main memory," in *ASPLOS*, 2011.
[5] D. Lo and C. Kozyrakis, "Dynamic management of turbomode in modern multicore chips," in *HPCA*, 2014.
[6] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: eliminating server idle power," in *ASPLOS*, 2009.
[7] D. Meisner and T. F. Wenisch, "Dreamweaver: architectural support for deep sleep," in *ASPLOS*, 2012.
[8] V. Anagnostopoulou, S. Biswas *et al.*, "Barely Alive Memory Servers: Keeping Data Active in a Low-power State," *J. Emerg. Technol. Comput. Syst.*, vol. 8, no. 4, Nov. 2012.
[9] D. Wong and M. Annavaram, "Knightshift: Scaling the energy proportionality wall through server-level heterogeneity," in *MICRO*, 2012.
[10] C. Clark, K. Fraser *et al.*, "Live migration of virtual machines," in *NSDI*, 2005.
[11] P. Ranganathan, P. Leech *et al.*, "Ensemble-level power management for dense blade servers," *ISCA*, 2006.
[12] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA*, 2007.
[13] R. Nathuji and K. Schwan, "Virtualpower: coordinated power management in virtualized enterprise systems," *SOSP*, 2007.
[14] C. Subramanian, A. Vasan, and A. Sivasubramaniam, "Reducing data center power with server consolidation: Approximation and evaluation," in *HiPC*, 2010.
[15] A. Gandhi, M. Harchol-Balter *et al.*, "Autoscale: Dynamic, robust capacity management for multi-tier data centers," *ACM Trans. Comput. Syst.*, vol. 30, no. 4, Nov. 2012.
[16] M. Lin, A. Wierman *et al.*, "Dynamic right-sizing for power-proportional data centers," in *INFOCOM*, 2011.
[17] www.spec.org/power_ssj2008/, "Spec power_ssj2008," 2012.
[18] D. Wong and M. Annavaram, "Implications of high energy proportional servers on cluster-wide energy proportionality," in *HPCA*, 2014.
[19] C.-H. Hsu and S. Poole, "Revisiting Server Energy Proportionality," in *ICPP*, 2013.
[20] S. Kanev, K. Hazelwood *et al.*, "Tradeoffs between power management and tail latency in warehouse-scale applications," in *IISWC*, 2014.
[21] D. Wong, J. Chen, and M. Annavaram, "Retrospective look back on the road towards energy proportionality," in *IISWC*, 2015.
[22] C.-H. Hsu and S. W. Poole, "Measuring Server Energy Proportionality," in *ICPE*, 2015.
[23] V. Tseng, "Measuring the power consumption in a server at component level," Hogeschool van Amsterdam, Software Improvement Group, Tech. Rep., 10 2012.
[24] J. Mars and L. Tang, "Whare-map: Heterogeneity in "Homogeneous" Warehouse-scale Computers," in *ISCA*, 2013.
[25] J. Moore, J. Chase *et al.*, "Making scheduling "cool": Temperature-aware workload placement in data centers," in *USENIX ATC*, 2005.
[26] I. n. Goiri, K. Le *et al.*, "Greenslot: Scheduling energy consumption in green datacenters," in *SC*, 2011.
[27] D. Lo, L. Cheng *et al.*, "Towards energy proportionality for large-scale latency-critical workloads," in *ISCA*, 2014.
[28] C. Delimitrou and C. Kozyrakis, "Paragon: Qos-aware scheduling for heterogeneous datacenters," in *ASPLOS*, 2013.
[29] D. Meisner, J. Wu, and T. F. Wenisch, "Bighouse: A simulation infrastructure for data center systems," in *ISPASS*, 2012.
[30] D. Meisner and T. F. Wenisch, "Stochastic queuing simulation for data center workloads," in *EXERT*, 2010.
[31] Intel Xeon e5-2470, http://ark.intel.com/products/64623/Intel Xeon-Processor-E5-2470-20M-Cache-2_30-GHz-8_00-GTs-Intel-QPI
[32] J. Hamilton, http://perspectives.mvdirona.com.
[33] P. Nikolaou, Y. Sazeides *et al.*, "The implications of different dram protection techniques on datacenter tco," in *SELSE*, 2015.
[34] P. Lotfi-Kamran, B. Grot *et al.*, "Scale-out processors," in *ISCA*, 2012.
[35] G. Varsamopoulos and S. K. S. Gupta, "Energy proportionality and the future: Metrics and directions," in *ICPP Workshops*, 2010.
[36] N. Tolia, Z. Wang *et al.*, "Delivering energy proportionality with non energy-proportional systems: optimizing the ensemble," in *HotPower*, 2008.
[37] G. Prekas, M. Primorac *et al.*, "Energy Proportionality and Workload Consolidation for Latency-critical Applications," in *SOCC*, 2015.
[38] F. Ryckbosch, S. Polfliet, and L. Eeckhout, "Trends in Server Energy Proportionality," *Computer*, vol. 44, no. 9, Sep. 2011.
[39] B. Urgaonkar, P. Shenoy *et al.*, "Dynamic provisioning of multi-tier internet applications," in *ICAC*, 2005.
[40] C. Lefurgy, X. Wang, and M. Ware, "Power capping: a prelude to power shifting," *Cluster Computing*, vol. 11, no. 2, pp. 183–195, Jun. 2008.
[41] J. D. Davis, S. Rivoire, and M. Goldszmidt, "Star-cap: Cluster power management using software-only models," Microsoft Research, Tech. Rep. Microsoft Technical Report MSR-TR-2012-107, October 2012.
[42] R. Cochran, C. Hankendi *et al.*, "Pack&Cap: Adaptive DVFS and Thread Packing Under Power Caps," in *MICRO*, 2011.
[43] K. Ma and X. Wang, "PGCapping: Exploiting Power Gating for Power Capping and Core Lifetime Balancing in CMPs," in *PACT*, 2012.
[44] A. Gandhi, M. Harchol-Balter *et al.*, "Power capping via forced idleness," in *WEED*, 2009.
[45] L. A. Barroso, J. Clidaras, and U. Hoelzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*, Morgan and Claypool Publishers, 2013.
[46] D. Meisner, C. M. Sadler *et al.*, "Power management of online data-intensive services," in *ISCA*, 2011.