Energy-Agility: A New Grid-centric Metric for Evaluating System Performance

Supreeth Subramanya, Zain Mustafa, David Irwin, and Prashant Shenoy
University of Massachusetts Amherst

ABSTRACT

Energy-efficiency has long been considered a "first class" metric for evaluating computer system performance. A high energy-efficiency translates into more work done per unit of energy, which increases the operating time for battery-powered systems and lowers the energy bill for conventional grid-powered systems. Unfortunately, optimizing solely for energy-efficiency leads systems designers to make two implicit assumptions about how the electric grid generates energy: i) that all energy is created equal and ii) that it is available in unlimited quantities at any time. In reality, all energy is not created equal—its cost and carbon footprint vary substantially over time—and, as the penetration of intermittent renewable energy sources in the grid increases, it may not be available in unlimited quantities at any time. Thus, we propose a new grid-centric performance metric, called *energy-agility*, that accounts for the assumptions above. We argue that grid-friendly computer systems are better judged by their energy-agility, rather than their energy-efficiency.

1. INTRODUCTION

Energy-efficiency, which is defined as the amount of work, i.e., computation and I/O, done per joule of energy, has been considered a "first class" metric for evaluating computer system performance for over two decades [?]. Energy-efficiency as a metric originally gained prominence with the introduction of mobile battery-powered systems, such as laptops, which must run off a fixed supply of stored energy [?]. In this case, a more energy-efficient system can run longer without depleting its battery. Energy-efficiency eventually also became important for server-class systems with the rise of warehouse-scale data center facilities [?]. Here, greater energy-efficiency lowers these facilities' large electric bills (assuming that utilities charge a constant price for energy over time) and reduces their carbon emissions (assuming that the energy is created from carbon-based sources).

Energy-efficiency for warehouse-scale data center facilities remains a highly active research area, since their size and number continues to grow to satisfy the increasing demand for cloud-based services. The power demands of the largest data centers now exceed 100 megawatts (MW) [?], and, collectively, they are estimated to consume 1.7-2.2% of electricity in the United States [?]. The scale of data center energy consumption is now large enough to affect the electric grid's operation. Unfortunately, optimizing exclusively for energy-efficiency has led systems designers to make two implicit assumptions about how the grid generates energy: i) that all energy is created equal and ii) that it is available in unlimited quantities at any time. These assumptions are not correct: in reality, all energy is not created equal—its cost and carbon footprint vary over time depending on the mix of generators utilities use to create it—and, as the penetration of intermittent renewable energy in the grid increases, it may not be available in unlimited quantities at any time.

Thus, in the broader context of the electric grid, just because a system is highly energy-efficient does not necessarily mean that its carbon footprint and energy cost are lower than a highly inefficient system. That is, an inefficient system that consumes energy at the "right" times, e.g., when renewable energy is plentiful or electricity prices are cheap, may be cleaner and cheaper than a much more energy-efficient system that uses energy at the "wrong" times, e.g., when renewable energy is limited or electricity prices are high. In addition, energy-efficient systems designers generally do not account for energy constraints, but instead assume they can consume whatever energy is required without limits at any given time to satisfy their workload. While there has been some prior work on enforcing power caps across servers [?, ?] and clusters [?, ?], it is distinct from energy-efficiency optimizations, and instead focuses on over-subscribing power delivery infrastructures with fixed limits on the maximum amount of power they can deliver.

In this paper, we consider energy constraints in a much broader context as they arise in the electric grid. Modern society generally expects the grid to instantly generate however much power is necessary to meet any amount of demand at any time. Unfortunately, meeting this expectation imposes a substantial financial and environmental burden for a number of reasons. First, the grid must massively over-provision its generation capacity to meet its peak demands, which may only occur for a short period. In addition, the "peaking" generators utilities activate to satisfy demand peaks are typically much less efficient than the baseload generators that continuously operate. As a result, estimates attribute as much as 20% of the generation costs in the U.S. to servicing only the top 100 hours of peak demand each year [?]. Put another way, as much as 20% of the generation costs go to satisfying a period of roughly four days per year. Further, different types of generators not only in-

LIMITS'15, June 15–16, 2015, Irvine, California, USA

cur different monetary costs, but also have different carbon emissions and environmental impact. For example, WattTime.org estimates the grid's real-time carbon emissions to encourage consumers to use energy when emissions are low, similar to how variable electricity prices encourage energy consumption when its price is low. Finally, ensuring the grid is capable of satisfying arbitrary power demands discourages high penetrations of renewables, such as wind and solar, since utilities cannot control their output.

Due to the dynamics above, making energy generation more "efficient," that is reducing its cost and carbon footprint, will require consumers to limit their energy usage at particular times, either when the cost and carbon footprint of generation are too high or when renewable energy is not plentiful. There are a variety of ways to implement or incentive such *demand-side management*, which is a focus of recent smart grid efforts. Our purpose here is to highlight that the grid's "efficiency" in terms of its cost and carbon footprint varies substantially over time and is more complex than the conventional notion of energy-efficiency, i.e., work done per joule, in computer systems. In particular, a higher energy-efficiency in computer systems does not necessarily correlate with a higher grid efficiency, i.e., lower energy costs and carbon emissions. As a result, energy-efficiency is not the right metric to quantify a computer system's performance in the context of the grid. Just because a system is energy-efficient does not mean it is "green," as is often implicitly assumed in prior work. In reality, how a system uses energy to accomplish a task (and how that energy was generated) is just as important as how much energy it uses. To properly evaluate performance in this context, we propose a new grid-centric metric, which we call *energy-agility*.

While energy-efficiency is a measure of work done per joule of energy consumed by a platform, energy-agility is a measure of work done per joule of energy *available* to a platform, which may vary *over time*. Thus, as a metric, energy-agility captures the salient characteristics above that i) energy is not always available in unlimited quantities at any time, and i) the availability of energy may vary over time. The energy available to a platform is also independent of how much energy it actually consumes. Whereas energy-efficiency only depends on how much energy a platform consumes, energy-agility applies a "use it or lose it" property to energy that incentivizes platforms to use as much energy as possible, as efficiently as possible, when it is available, or else waste it. Energy-agility captures the grid's basic characteristics, where electricity's supply and demand must be balanced at all times, and the only way to not waste unused energy is to explicitly store it for later use. We argue that, when considered in the broader context of the electric grid, computer systems are better judged by their energy-agility, rather than their energy-efficiency. Note that, since energy-agility incentivizes systems to use available energy as efficiently as possible, it also implicitly encourages energy-efficiency.

2. BACKGROUND

Formally, energy-agility is a measure of the amount of work, e.g., computation and I/O, done by a computer system given a *power signal* $P(t)$ that dictates an energy cap the system must adhere to over each interval $(t-\tau, t]$ for some interval length τ. Thus, for a given amount of work, a greater energy-agility translates into a shorter running time and less aggregate energy used or wasted. Due to the "use it or lose it" property above, the value of τ derives from a platform's energy storage capacity, where a larger capacity implies a higher value. While energy-agility quantifies how well an application is able to adapt to variable amount of available power, it does not dictate the underlying reason for the power variations, e.g., due to demand-response signals, fluctuations in renewable generation, changes in electricity prices, etc. Thus, the characteristics of $P(t)$ may differ widely depending on the scenario. As with energy-efficiency, a platform's software design and its hardware capabilities contribute to its energy-agility. However, we believe designing energy-agile systems differs markedly from designing energy-efficient ones. For example, recent research on energy-efficient system design focuses on ensuring two key characteristics—*balance* [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] and *energy-proportionality* [**?**]—which we briefly describe below.

Balanced systems focus on aligning a hardware platform's capabilities with its software requirements to ensure all hardware is fully utilized and, hence, does not waste power by consuming it when idle, i.e., doing no useful work. To illustrate, GPUs, which combine multiple low-speed cores with high streaming memory bandwidth, are energy-efficient for data-parallel applications [**?**, **?**], but not for applications that require high-speed single threaded performance. Since I/O is often the bottleneck for data-parallel applications and CPU power increases super-linearly with clock speed, high-speed cores simply waste power without improving performance. This theme of balance underlies the node platform design for JouleSort [**?**], FAWN [**?**], and Themis [**?**], which target an energy-efficient sorting system, key-value store, and MapReduce implementation, respectively.

While balanced systems maximize peak performance per watt, e.g., at 100% utilization, energy-proportional systems focus on energy-efficiency across all utilization levels, such that power consumption scales linearly with node utilization. Designing energy-proportional nodes remains a challenging problem, since a variety of hardware components, including the CPU, memory, disk, motherboard, and power supply, consume significant amounts of power. Thus, any power optimization that targets only a single component does not result in energy-proportionality, since it reduces only a fraction of a node's total power consumption [**?**, **?**].

As one example, due to the power consumption of non-CPU components, nodes that aggressively use dynamic voltage and frequency scaling (DVFS) on CPUs at low utilizations may still operate at over 50% of their peak power [**?**, **?**]. As a result, there has been much research on approximating energy-proportionality in multi-node platforms for various applications by activating and deactivating entire nodes as workload demands vary over time, and then concentrating workload on the active set of nodes [**?**]. Thus, if an application requires 5k nodes in a 10k-node system (at 50% system utilization), only 5k nodes will be active and consuming >50% peak power, while the other 5k nodes will stay in a low-power inactive state, such as ACPI's S3 Suspend-to-RAM state, which consumes minimal (<5% peak) power.

Balance and energy-proportionality have been the basis for a wide range of prior research on improving energy-efficiency. However, a key attribute of both types of systems is that they are *workload-driven*: while they use energy efficiently, they consume whatever power is necessary (without limits) to satisfy their workload. For example, if the application above requires all 10k nodes at some point to complete a subtask, it simply activates the remaining 5k nodes, without any constraints. Thus, these systems implicitly assume that the power delivery infrastructure is able to deliver as much power as necessary (up to the system's peak power) at any time. Grid-centric energy-agile systems can make no such assumption. Instead, they are *power-driven*: to reduce power's cost and carbon footprint, they must choose how to use a limited, dynamically changing, and potentially unpredictable power budget to maximize performance.

While researchers have made substantial improvements to the energy-efficiency of computing systems, as noted above, we argue that continuing to achieve significant gains will pose an increasingly challenging problem. Just as with energy-efficiency optimizations, optimizing energy-agility also has the potential to reduce computing's energy cost and carbon footprint. In particular, we believe that designing for energy-agility will yield new avenues of research that can reduce the energy cost and carbon emissions associated with large-scale computing in data center and high-performance computing platforms by correcting gross inefficiencies in the electric grid. As one example, note that energy-proportionality only applies to applications where the workload intensity varies over time based on request volume, e.g., web applications, batch schedulers, etc.; it does not apply to large-scale long-running tasks that have no variance in their workload, but are highly amenable to delays from power variations. In contrast, energy-agility encourages techniques that optimize such tasks to use energy efficiently at all utilization levels and adapt to variations in available energy, e.g., due to cost or renewable fluctuations.

3. DESIGNING FOR ENERGY-AGILITY

There has been a variety of recent research on designing computing systems to better handle power variations, e.g., due to changing power prices. For example, initial research has focused on optimizing a variety of system components for power variations, including distributed caches [**?**], file systems [**?**], virtual machines [**?**, **?**], and batch schedulers [**?**, **?**, **?**, **?**, **?**]. The metric these systems measure themselves against is generally the cost of power, since variable electricity prices are typically lower, on average, than flat prices. Thus, the performance of these systems is dependent on the absolute price of electricity: the more variable the prices, or the wider their range, the more cost savings are possible. Using prices to evaluate systems performance is not ideal, since prices vary significantly by region, by time, and based on external factors. While changes in prices may incentivize or disincentives energy-agile design, they are not a sound basis for evaluating systems that use variable power.

In contrast, energy-agility provides a price-independent metric to evaluate and compare the performance of such systems, similar to how the absolute cost of energy has no bearing on a system's energy-efficiency. However, research into energy-agile design requires new considerations along a number of dimensions, which we summarize below.

Impact of Power Signal. The characteristics of the power signal $P(t)$—its variability, its range, and its magnitude—affect energy-agility. For example, consider an energy-agile server cluster, which must activate and deactivate nodes to stay within the power cap at any time t. With a stable power signal equal to 50% of the cluster's maximum power, the cluster might simply deactivate half its nodes to stay with in the cap. However, with a highly variable power signal that oscillates between near full power and near zero power, the cluster must make choices about when and how to deactivate nodes to cap power. In this case, blinking all nodes nodes [**?**] might improve performance relative to policies that activate and deactivate individual nodes.

Impact of the Energy Storage Capacity. Energy storage capacity affects the stability of the power signal over each interval τ. A more stable power signal allows a platform to better plan its power usage, which may reduce the number of necessary power state transitions and overhead. However, energy storage capacity is expensive to install and maintain [**?**, **?**, **?**]. Thus, a key question in energy-agile design is how close can the energy-agility of a system with little-to-no energy storage come to a system with infinite energy storage capacity. Ideally, a system with little-to-no energy storage would be preferable if its energy-agility is equal to a system with infinite energy storage. Energy-agility provides a metric for assessing the storage necessary to achieve a certain energy-agility independent of electricity or storage costs. Thus far, storage capacity has been considered largely from a cost perspective, i.e., is a certain amount of energy storage profitable given electricity prices?, and not a performance perspective, i.e., how much energy storage does the system require to reach a specific level of energy-agility?

Impact of Efficiency Losses. If an energy-agile system is not able to, or choose not to, consume all the energy available at any given time, it must either store it for later use or instantly transmit it for use by another consumer. The grid currently implements the latter case using "net metering," which permits consumers to transfer surplus energy to the grid for use by others. Both options efficiency losses that waste energy. For example, 20-50% of stored energy is typically wasted due to conversion and efficiency losses depending on the type of energy storage. Losses due to net metering are more difficult to quantify, but include transmission and distribution losses (which are proportional to the square of current) and losses related to running inefficient "peaking" generators to offset renewable energy stochastics. Energy-agility can incorporate these losses by discounting surplus energy by some percentage, rather than assuming it is entirely wasted. By considering these losses, systems designers can accurately quantify the effect of energy storage or net metering on energy-agility.

Impact of the Number and Type of Platform Power States. The ability of a system to fully use a varying amount of available power is largely a function of the number and type of platform power states it has. High-power platforms typically have a multitude of active and inactive power states, including the full range of ACPI system states (S0-S5), inactive processor C states, and active processor P states, e.g., using dynamic voltage and frequency scaling.

As a result, these platforms have a wide dynamic power range that can accommodate a wide range of power caps. However, high-power platforms also tend to have the highest idle power usage due to powering non-CPU components, such as memory, disks, network cards, etc., and are significantly more energy inefficient than lower-power nodes. Unfortunately, low-power nodes tend to have many fewer power states; they often have limited processor C states and few active P states, since their processors are already considered low-power. Thus, while low-power platforms are much more energy-efficient than high-power platforms, they have a narrower dynamic power range, which may prevent them from fully utilizing an arbitrary amount of available power. Thus, an interesting question is, with respect to optimizing energy-agility, whether the efficiency of the low-power platforms outweighs their lack of dynamic power range.

Impact of the Power State Transition Latency. The latency to transition between power states is also a key factor in energy-agile design: the smaller the latency to transition power states, the less overhead a system incurs to adapt to changes in power. Servers often do not support server-wide power state transitions, such as ACPI's S3 Suspend-to-RAM state and its S4 Suspend-to-Disk state, and if they do, they have not been optimized for low-latency transitions. Thus, even though the latency to transition to S3 could take as little as a few hundred milliseconds [**?**], most servers have transition latencies on the order of tens of seconds [**?**]. Such long transitions preclude power management strategies that incur large numbers of transitions, such as blinking [**?**], to maintain a variable power cap. Note that blinking all nodes in tandem between the active and inactive states enables better use of the available network bandwidth than only activating a fraction of the nodes at time. Thus, an interesting research question is at what latency does the benefits of blinking outweigh the overhead of transitioning power states.

Impact of Inter-node Communication Patterns. An energy-agile server cluster continuously shifts the limited power that is available away from resources that are waiting or idle to resources that are performing useful work. Thus, energy-agile design also poses an algorithmic challenge to determine how to shift power based on inter-node communication patterns. For example, if all nodes are shuffling data amongst themselves, then reducing power by deactivating any node or lowering its voltage and clock frequency will simply delay the entire data shuffling phase. To maximize energy-agility, a platform must adjust power states in conjunction with the application to ensure the application is fully utilizing all of the available power. As a result, energy-agile applications should continually monitor their utilization and energy usage in a closed-loop fashion to shift power away from nodes that are not using it to nodes that are using it. A key challenge is determining the optimal algorithms for changing power states based on the available power states and application communication patterns.

Impact on Applications. Finally, applications may require modifications to maximize their energy-agility. For example, if a server cluster platform activates and deactivates nodes as power rises and falls, then an application must be modified (often significantly) to gracefully handle such sudden addition and removal of nodes. Alternatively, a server cluster could only use active power states to enable an application to run unmodified; unfortunately, only using active power states does not permit a wide dynamic power range, which limits energy-agility. Other policies, such as blinking [**?**], which transition all nodes in a server cluster between the active and inactive states in tandem, might also be able to minimize application modifications, such that all nodes are concurrently active. An important research challenge is fully optimizing energy-agility for a wide range of applications without requiring significant modifications.

4. CONCLUSION

This paper proposes a new grid-centric metric, called energy-agility, for evaluating computer system performance. Energy-agility accounts for the fact that all energy is not created equal—its cost and carbon footprint vary over time. These variations are expected to intensify as the penetration of renewable energy sources in the grid increases. As a result, computer (and other) systems will need to take into account these variations in assessing their performance. We argue that the energy-efficiency metric used by computing researchers over the past quarter century is not sufficient for this purpose. Instead, we propose a grid-centric metric, called energy-agility, to assess computer system performance in scenarios where a limited amount of available power varies over time. We then outline a number of considerations for energy-agile systems designers.

5. REFERENCES

[1] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing. Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers. In *HotPower*, October 2011.

[2] D. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: A Fast Array of Wimpy Nodes. In *SOSP*, October 2009.

[3] L. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. *Computer*, 40(12), December 2007.

[4] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *SOSP*, December 2001.

[5] R. Cochran, C. Hankendi, A. Coskun, and S. Reda. Pack & Cap: Adaptive DVFS and Thread Packing Under Power Caps. In *MICRO*, December 2011.

[6] X. Fan, W. Weber, and L. Barroso. Power Provisioning for a Warehouse-sized Computer. In *ISCA*, June 2007.

[7] K. Fehrenbacher. The Era of the 100MW Data Center. In *Gigaom*, January 31 2012.

[8] A. Gandhi, M. Harchol-Balter, R. Das, J. Kephart, and C. Lefurgy. Power Capping via Forced Idleness. In *Weed*, June 2009.

[9] I. Goiri, W. Katsak, K. Le, T. Nguyen, and R. Bianchini. Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. In *ASPLOS*, March 2013.

[10] I. Goiri, K. Le, M. Haque, R. Beauchea, T. Nguyen, J. Guitart, J. Torres, and R. Bianchini. GreenSlot: Scheduling Energy Consumption in Green Datacenters. In *SC*, April 2011.

[11] I. Goiri, K. Le, T. Nguyen, J. Guitart, J. Torres, and R. Bianchini. GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks. In *EuroSys*, 2012.

[12] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar. Benefits and Limitations of Tapping into Stored Energy for Datacenters. In *ISCA*, June 2011.

[13] Jonathan Koomey. Growth in Data Center Electricity Use 2005 to 2010. In *Analytics Press*, Oakland, California, August 2011.

[14] A. Krioukov, C. Goebel, S. Alspaugh, Y. Chen, D. Culler, and R. Katz. Integrating Renewable Energy Using Data Analytics Systems: Challenges and Opportunities. In *Bulletin of the IEEE Computer Society Technical Committee*, March 2011.

[15] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. Sung Kim, T. M. Aamodt, and V. Janapa Reddi. GPUWattch: Enabling Energy Optimizations in GPGPUs. In *ISCA*, June 2013.

[16] C. Li, A. Qouneh, and T. Li. iSwitch: Coordinating and Optimizing Renewable Energy Powered Server Clusters. In *ISCA*, June 2012.

[17] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: Eliminating Server Idle Power. In *ASPLOS*, March 2009.

[18] National Public Radio. How Smart is the Smart Grid?: Interview with Dan Delurey, President Demand Response Smart Grid Coalition, July 7th, 2010.

[19] W. Qadeer, R. Hameed, O. Shacham, P. Venkatesan, C. Kozyrakis, and M. Horowitz. Convolution Engine: Balancing Efficiency and Flexibility in Specialized Computing. In *ISCA*, June 2013.

[20] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level Power Management for Dense Blade Servers. In *ISCA*, June 2006.

[21] A. Rasmussen, M. Conley, R. Kapoor, V.T. Lam, G. Porter, and A. Vahdat. Themis: An I/O-Efficient MapReduce. In *SoCC*, October 2012.

[22] A. Rasmussen, G. Porter, M. Conley, H. Madhyasthay, R. Mysore, A. Pucher, and A. Vahdat. TritonSort: A Balanced and Energy-Efficient Large-Scale Sorting System. *TOCS*, 31(1), February 2013.

[23] S. Rivoire, M. Shah, and P. Ranganathan. JouleSort: A Balanced Energy-Efficient Benchmark. In *SIGMOD*, June 2007.

[24] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: Managing Server Clusters on Intermittent Power. In *ASPLOS*, March 2011.

[25] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. A Distributed File System for Intermittent Power. In *IGCC*, June 2013.

[26] R. Singh, D. Irwin, P. Shenoy, and K.K. Ramakrishnan. Yank: Enabling Green Data Centers to Pull the Plug. In *NSDI*, April 2013.

[27] E. Le Sueur and G. Heiser. Dynamic Voltage and Frequency Scaling: The Laws of Diminishing Returns. In *HotPower*, October 2010.

[28] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu. Delivering Energy Proportionality with Non-Energy-Proportional Systems: Optimizing the Ensemble. In *HotPower*, December 2008.

[29] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam. Optimal Power Cost Management Using Stored Energy in Data Centers. In *SIGMETRICS*, March 2011.

[30] V. Vasudevan, D. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru. Energy-efficient Cluster Computing with FAWN: Workloads and Implications. In *e-Energy*, April 2010.

[31] Vijay Vasudevan, David G. Andersen, Michael Kaminsky, Jason Franklin, Michael A. Kozuch, Iulian Moraru, Padmanabhan Pillali, and Lawrence Tan. Challenges and Opportunities for Efficient Computing with FAWN. *SIGOPS Operating Systems Review*, (1), January 2011.

[32] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy. Energy Storage in Datacenters: What, Where, and How Much? In *SIGMETRICS*, June 2012.

[33] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for Reduced CPU Energy. In *OSDI*, November 1994.

[34] S. Xi, M. Guevara, J. Nelson, P. Pensabene, and B. Lee. Understanding the Critical Path in Power State Transition Latency. In *ISPLED*, September 2013.