Challenges in Distributed Energy Adaptive Computing

Krishna Kant
Intel Corporation
krishna.kant@intel.com

ABSTRACT

Fueled by burgeoning online services, energy consumption in information technology (IT) equipment is becoming a major concern from a variety of perspectives including the continuation of Moore's Law for hardware design, enabling sophisticated mobile client functionality, mounting utility costs in data centers, and increasing CO_2 emissions associated with IT manufacturing, distribution, usage and disposal. This article discusses an approach where energy consumption and related issues of heat dissipation and sustainability are considered as the primary concerns that drive the way computation and communication is organized at both clients and servers. This article describes the challenges in supporting such a distributed energy adaptive computing paradigm.

1. INTRODUCTION

The rapid proliferation of information technology (IT) continues to increase both the number of computing devices and the energy consumed by them. Coupled with the push towards more compact and higher performing devices, this proliferation has made energy related issues critical in a variety of ways that range from advancement of underlying technologies to global environmental impact. In particular, at the circuit level, as the feature size decreases, so does the interconnect thickness which in turn leads to dramatic increase in resistance and hence power wasted as heat. This effect coupled with more devices per unit area are creating unsustainable power and thermal densities that threaten Moore's law [16], and the situation will worsen with 3-D integration [2] Because of increasing miniaturization and computing power, similar issues arise at higher levels as well. For example, the tight form factors of blade servers, notebooks and PDAs make heat dissipation very challenging. In terms of power consumption per se, the battery technology has not improved much even as the computing requirements of mobile devices continue to go up. On the data center side, the total power consumption can run into multiple megawatts or more, thereby making electricity consumption a very substantial percentage (up to 50%) of the operational costs.

Many of these issues have been well recognized and have resulted in substantial improvements in energy efficiency at a variety of levels – from low-power HW design to aggressive use of available power modes to intelligent load and activity management (e.g., see [6] and references therein), including coordinated power management at multiple levels [13]. These efforts are expected to continue in the foreseeable future. Yet, the sheer increase in the computing base and the rapidly emerging sustainability concerns require that we move beyond energy efficiency to *energy adaptive computing* or EAC. The main point of EAC is to consider energy related constraints at all relevant levels as the primary limitation that determines how much computation we can afford. We then need to come up with appropriate adaptation mechanisms which may range from simple schemes such as slow-down or redistribution of computation to substantially changing the nature of the computation. For example, by loosening the QoS or availability constraints, it is often possible to do the computations in such as way so as to reduce energy requirements significantly beyond what is possible by traditional power management techniques. Although adaptation of computations to cater to a variety of resource constraints and faults has been extensively explored in the literature [14, 3, 1, 12], a cooperative, multi-level distributed adaptation to limited energy in complex environments still poses challenging problems.

2. SUSTAINABILITY AND ENERGY ADAPTIVE COMPUTING

In recent years, the environmental impact of IT has also become an area of increasing concern. Much of the electricity powering the IT infrastructure comes from fossil fuels and thus involves substantial carbon footprint. Furthermore, in spite of aggressive efforts at power consumption reduction of computing systems, we are likely to see 2-4X increase in overall power consumption of servers, clients and the intervening network in the next decade and hence a corresponding increase in the carbon footprint.

It is well recognized by now that much of the power consumed by a data center is actually wasted. In particular, up to 50% of the data center power may go into the non-IT equipment including cooling, air movement, electrical conversion and distribution, and lighting. This energy consumption does not directly contribute to computing and thus can be considered a "waste". Furthermore, the operational energy is not the only energy involved here. Many of these functions are quite resource and infrastructure heavy and a substantial amount of energy goes into the construction and maintenance of the cooling and power conversion/distribution infrastructures. In fact, even the "raw" materials such as water or metals involve hidden energy footprint in form of making those materials available in usable form. It follows that from a sustainability perspective, it is not enough to simply minimize operational energy usage or wastage; we need to minimize the energy that goes into the infrastructure as well. This principle applies not only to the supporting infrastructure but to the IT devices such as

clients and servers themselves. In fact, for the rapidly proliferating small mobile clients such as cell-phones and PDAs, the operational energy used over their useful lifetimes could be less than the energy used in their manufacture, distribution and recycling. Even for servers in data centers, the increased emphasis on reducing operating energy only makes the non-operational part of the energy more important.

Towards this end, it is important to consider data centers that can be operated directly via locally produced renewable energy (wind, solar, geothermal, etc.) with minimal dependence on the power grid or large energy storage systems. Such an approach reduces carbon footprint not only via the use of renewable energy but also by reducing the size and capacity of power storage and power-grid related infrastructure. For example, smaller uninterruptible power supply (UPS) and lower power draw from the grid would reduce data center infrastructure costs. The down-side of the approach is more variable energy supply and more frequent episodes of inadequate available energy to which the data center needs to adapt dynamically.

In large data centers, the cooling system not only consumes a substantial percentage of total power (up to 25%) but also requires significant infrastructure in form of chiller plants, compressors, fans, plumbing, etc. Much of this energy consumption and infrastructure can be done away with by using ambient (or "free") cooling, perhaps supplanted with undersized cooling plants that kick in only when ambient temperature becomes too high. Such an approach requires the energy consumption (and hence the computation) to adapt dynamically to the available cooling ability. Of course, the energy available from a renewable source (e.g., solar) may be correlated with the available cooling capacity, and such interactions need to be considered in the adaptation mechanisms.

For servers and AC operated clients, the power supply and the power distribution infrastructure can also be significant energy wasters. For example a server consuming 500W and sporting a high efficiency power supply with 85% efficiency will still waste 75W of power. (Since this waste is in form of heat, additional power is wasted in removing the resulting heat.) Often, servers run at rather low utilization levels, and the power supply efficiency is typically much poorer at lower utilizations. Smart *phase shedding* power supplies address this problem by providing a number "phases" [4]. As the server utilization dips more and more phases can be turned off, thereby keeping the power supply utilization and efficiency high. For example, a power supply with 8 phases may have all phases active at 90% server utilization, but at server utilization of 45%, only 4 phases need to be active and will still provide the same power supply efficiency. Similar approaches apply to on-board voltage regulators (VR's). Since the power availability is limited by the number of active phases and the phase changes are rather slow, it becomes necessary to adapt computation to various limitations including power/thermal ones.

Yet another sustainability issue is the overdesign and overprovisioning that is commonly observed at all levels of computer systems. For example, the power and cooling infrastructure in servers, chassis, racks, and the entire data center is designed for worst-case scenarios which are either rare or do not even occur in realistic environments. For example, it is very difficult to find workloads where the CPUs, DRAM and network adapter in a server will be running close to their

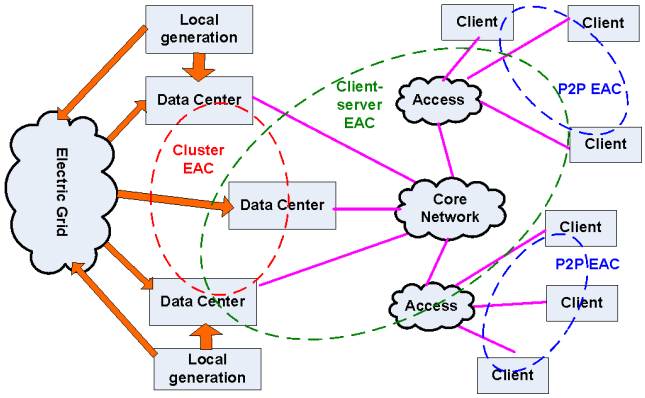

Figure 1: Illustration of energy adaptation loops

capacity simultaneously. Although data centers are beginning to "derate" specified power and cooling requirements to address this lack of realism, it is possible to go significantly beyond this practice and reduce the overall energy footprint of both the servers and the clients. This leanness of the infrastructure could be either static (e.g., lower capacity power supplies and heat sinks, smaller disks, DRAM, etc.), or dynamic (e.g., phase shedding power supplies, hardware resources dynamically shared via virtualization). In either case, it is necessary to adapt computations to the limits imposed by power and thermal considerations.

3. DISTRIBUTED ENERGY ADAPTIVE COMPUTING

It is clear from the above discussion that many advanced techniques for improving energy efficiency of IT infrastructure and making it more sustainable involves the need to dynamically adapt computation to the suitable energy profile. In some cases, this energy profile may be dictated by energy (or power) availability, in other cases the limitation may be a result of thermal/cooling constraints. In many cases, the performance and/or QoS requirements are malleable and can be exploited for energy adaptation. For example, under energy challenged situations, a user may be willing to accept longer response times, lower audio/video quality, less current information, and even less accurate results. These aspects have been explored extensively in specific contexts, such as adaptation of mobile clients to intelligently manage battery lifetime [3]. However, complex distributed computing environments provide a variety of opportunities for coordinated adaptation among multiple nodes and at multiple levels. In general, there are three types of distributed energy adaptation scenarios: (a) Client-server, (b) Peer to Peer (or client to client), and (c) Cluster computing (or server to server). These are shown pictorially in Fig. 1 and discussed briefly in the following. Notice that in all cases, the network is an important part that should also participate in the adaptation.

3.1 Client-Server EAC

The client-server EAC needs to deal with both client-end and server-end adaptation to energy constraints in such a way so that client's QoS expectations are satisfied. A coordinated client-server energy adaptation could even deliver benefits beyond adaptation per se. As the clients become

more mobile and demand richer capabilities, the limited battery capacity gets in the way. The client-server EAC can provide better user satisfaction and service by seamlessly compensating for lack of client resources such as remaining battery, remaining disk space, operating in a hot environment, etc. In fact, such techniques can even help slow down client obsolescence and thus enhance the goal of sustainability.

Client-server EAC can be supported by defining client energy states and the QoS that the client is willing to tolerate in different states and during state switching. This information could be communicated to the server side in order to effect appropriate adaptation actions as the client energy state changes. For a more comprehensive adaptation, the intervening network should also be communicated the QoS requirements and should be capable of exploiting it. The situation here is similar to but more complex than the contract based adaptation considered in [12]. The major challenge is to decide optimal strategy to ensure the desired end-to-end QoS without significant overhead or increase in complexity. In a client-server computing, the client adaptation could also be occasionally forced by the server-side energy adaptation. Since server-side adaptation (such as putting the server in deep sleep state and migrating the application to another server) can affect many clients, the server side adaptation decisions become quite complex when interacting with a large number of geographically distributed and heterogeneous clients. Involving the network also in the adaptation further complicates the problem and requires appropriate protocol support.

3.2 Peer to Peer EAC

In a peer to peer setting involving increasingly mobile clients, energy consumption is becoming an important topic. Several recent papers have attempted to characterize energy consumption of P2P content sharing and techniques to improve their energy efficiency [5, 8, 9]. Energy adaptation in P2P environment is quite different from that in a client-server setting. For simple file-exchange between a pair of peers, it is easy to consider the energy state of the requesting and serving peers and that of their network connections; however, a collective adaptation of a large number of peers can be quite complex. Furthermore, it is important to consider the fundamental P2P issue of get-give in this adaptation. In particular, if a peer is in a power constrained mode, it may be allowed to be more selfish temporarily (i.e., allowed to receive the appropriate low-resolution content that it needs without necessarily supplying content to others). In a more general situation such as BitTorrent where portions of file may come from different clients, deciding and coordinating content properties and assembling the file becomes more challenging. In particular, it might be desirable to offload some of these functions to another client (that is not in energy constrained mode). In general, addressing these issues requires defining appropriate energy related metrics relative to the content requester, all potential suppliers (or "servers"), transit nodes and the intervening network. A framework that allows minimization of global energy usage while satisfying other local performance and energy requirements can be quite challenging.

3.3 Cluster EAC

Cluster EAC refers to computational models where the request submitted by a client requires significant computation involving multiple servers before the response can be returned. That is, client involvement in the service is rather minimal, although the client could certainly provide its limitations to the servers. In this sense, cluster EAC differs significantly from client-server EAC. Cluster EAC also differs substantially from P2P EAC because of AC power operation and cooperative functioning of servers. Cluster EAC involves the data center network, which due to its very high speed can consume a substantial amount of power. Thus, energy adaptation of network is important in cluster EAC.

In cluster EAC, the energy adaptation must happen at multiple levels. For example, the power capping algorithms may allocate a certain power share to each server in a chassis or rack, and the computation must adapt to this limit. In addition, there may be a higher level limit as well – for example, the limit imposed by the power circuits coming into the rack. At the highest level, energy adaptation is required to conform to the power generation (or supply) profile of the energy infrastructure. As usual, the limits placed at the lower level must necessarily be more flexible than at higher levels. Translating higher level limits into lower level limits is a challenging problem and requires a dynamic multi-level coordination [13]. A related issue is that of energy limitation along the software hierarchy (e.g., service, application, software modules, etc.) and corresponding multi-level adaptation.

Although in the above we discussed the three EAC scenarios separately, they all need to be addressed together. In particular, while a server responds to client adaptation needs, it itself may need to adapt due to power/thermal limits being imposed either due to its own actions or that of other interfering applications.

4. CHALLENGES IN DISTRIBUTED EAC

The most fundamental issue in energy adaptive computing is that of setting the energy budgets and using a suitable mechanism for adapting to the energy constraints. In many situations "energy adaptation" can be equated with "power adaptation" over suitably defined intervals; however, a sharper distinction is necessary in some situations. In particular, power spikes over short intervals are important in that they might exceed the power circuit capacities. This is typically of concern at the rack or chassis level when these enclosures are filled up with servers, each being heavily used. This could be more of an issue with less redundant designs as advocated here.

The real energy or power limitation usually applies only at a rather high level – at lower levels, this limitation must be progressively broken down and applied to subsystems in order to simplify the overall problem. For example, in case of a data center operating in an energy constrained environment, the real limitation may apply only at the level of the entire data center. However, this limitation must be broken down into allocations for the physical hierarchy (e.g., racks, servers, server components, ...) and/or logical hierarchy (e.g., service, application, application components, ...). While such a recursive break-down allows independent management of energy consumption at a finer-grain level, a suboptimal allocation that starves certain components while providing more than adequate power to related components could significantly degrade the overall performance and energy efficiency.

Good energy allocation or partitioning requires an accurate estimation of energy requirements at various layers. This is often quite difficult since the energy consumption not only depends on workload and hardware configuration but also on complex interactions between various hardware and software components and power management actions. For example, energy consumed by the CPU depends on how much the CPU stalls due to access latencies to the cache and memory hierarchy. The cache and memory access latencies, in turn, depend on their access patterns and power management actions. Similarly, the overall energy consumption of a set of applications or software modules running on a system could be quite different from the sum of energy consumption of the individual components in isolation.

Good energy allocations become progressively more difficult to achieve, as the available energy (or power) dips significantly below that required for normal (unconstrained) operation. These complications arise from the fact that the optimal operating point depends on a variety of factors including the hardware configurations, nature and importance of the workload, and how frequently the workload characteristics change and interactions between various hardware and software components. The interdependence between various hardware and software components may make their relative energy consumption to change quite substantially as the energy budgets shrink. For example, if CPUs and memory are allocated only 1/2 of their normal power for a workload, the changed workload behavior could make this proportion significantly suboptimal. Thus a continuous monitoring and adjustment to energy needs is required in order to keep energy allocation close to optimal.

When energy availability is restricted, certain applications – particularly those involved in background activities – don't even need to run. Others may run less frequently, with fewer resources, or even change their outputs, and still provide acceptable results. For applications that are driven by client requests and must run, the treatment depends on a variety of factors such as SLA requirements, level of variability in the workload characteristics, latency tolerance, etc. For example, if the workload can tolerate significant latencies and has rather stable characteristics, the optimal mechanism at the server level is to migrate the entire workload to a smaller set of servers so they can operate without power limitations and shut-down the rest. In this case, a tradeoff is necessary with respect to additional energy savings, SLA requirements, and migration overheads.

A comprehensive tradeoff requires accounting for not just the servers but also for the storage and networking infrastructure. As within a single platform, the relative energy consumption behavior between servers, storage and network could change significantly under severe energy constraints and needs to be considered carefully. As the workload becomes more latency sensitive, the latency impact of reconfiguration and power management actions must be taken into account. In particular, if firing up a shut-down server would violate latency and response time related SLA, it is no longer possible to completely shut-down the servers and instead one of the lower latency sleep modes must be used. A less stable workload may also require use of less severe power management actions.

Power management techniques typically take advantage of the low utilization of resources so that idle energy consumption can be minimized. This is done either by putting the devices into inactive low-power mode when idle (including complete shut-off), or running them at lower frequencies and voltages so as raise the device utilization (i.e., the traditional dynamic voltage-frequency scaling or DVFS controls) [15, 6]. Traffic batching can help reduce the overhead of entering and exiting sleep states [11] at the expense of adding additional latencies. In the past, much of the work has focused mostly on a rather narrow application of these techniques, such as DVFS control of CPUs or nap states for DRAM, but more complex scenarios involving coordinated control of multiple subsystems are beginning to be analyzed [10].

It is important to note that in case of EAC, often the problem is not inadequate work, but rather inadequate energy to process the incoming work. Obviously, in order to reduce the average power consumption, we need to slow down processing, except that this slowdown is not triggered by idling. The basic techniques for slowing down the computation still remain the same and may involve either forcing the device into low-power sleep modes or lower DVFS states. Reference [7] compares the effectiveness of the two methods. However, unlike the situation where the goal is to minimize wasted energy, an energy constrained environment requires careful simultaneous management of multiple subsystems in order to make the best use of the available energy. For example, it is necessary to simultaneously power manage CPU, memory and IO adapters of a server in order to ensure that the energy can be delivered where most required.

In addition to power management, the inability to process all of the incoming workload may require some additional load management actions to avoid build up of long queues. In a high-performance computing type of environment driven by long running jobs, delaying completion of running jobs or startup of new jobs usually has no further consequences. In a transactional system driven by user requests, further actions in form of dropping requests, redirecting them to another facility, migrating away entire applications, or reducing processing requirements at the cost of degraded output quality may be necessary. All of these actions require an accurate mechanism for evaluating "before" and "after" energy requirements for making intelligent decisions. The difficulty here is that because of interference between workloads, power consumptions don't necessarily add up, and some notions similar to those used in bandwidth management become necessary. For example, similar to the ideas of equivalent bandwidth or available bandwidth we also need to define and evaluate "equivalent power" and "available power" in order to handle decision making simply.

The admission control, migration and power management of a large number of resources at multiple levels raises a lot of interesting issues in terms of the stability and optimality of the control in addition to the issues of the overhead and lag associated with information exchange. A comprehensive control theoretic framework is required in order to address these issues. When the control extends over multiple physical facilities, perhaps each with differing energy costs, the problem becomes even more complex.

Although much of the above discussion concerns servers, similar issues apply to clients and their subsystems. For example, the partitioning and control of power between CPU, memory, storage and other portions of a client involves the same set of issues as servers. However, the peer-to-peer interaction between clients involves some unique issues as already stated above.

Although much of our discussion has focused on servers and clients, storage and network also need serious consideration in energy adaptive computing because of increasing data intensiveness of most applications. Energy management of rotating magnetic media often involves long latencies (in spinning down or spinning up the drives) and reliability issues resulting from RPM changes or repeated starts and stops. The emerging solid state storage (SSD) can be helpful in this regard. The energy management of network devices such as switches and routers is inherently difficult because of its nonlocal impact. For example, if a router/switch port is placed in a low power mode, every application and endpoint whose traffic goes through this port will be affected. When this energy management is triggered by shortage of available power (as opposed to simply taking advantage of idle periods), the impact is much more severe, since the energy management will result in accumulation of packets and significantly increase flow latencies. The end-to-end admission control required to manage the traffic needs to carefully manage these latencies, performance impact on various applications with varying latency sensitivity, and application timeouts. A significant amount of work remains to be done to address these issues adequately.

While the topic of applications changing their behavior in the face of energy limitations has been explored in several specific contexts such as audio/video streaming, rendering a web page, P2P content sharing [3, 9], and mechanisms to specify and manage the adaptation have been proposed [12, 14], there is scope for considerable further work on how and when to apply various kinds of adaptation mechanisms (e.g., lower resolution, higher latency, control over staleness and/or accuracy, etc.) under various kinds of power/thermal limitation scenarios.

The main theme in EAC has been to cut down "fat" at all levels and thereby lower not only the direct energy consumption but also the entire life-cycle energy costs that are essential to examine from a sustainability perspective. This leanness has a down-side: the increased fragility in the system which can be exploited by attackers. In particular, just as current systems can be victimized by denial of service (DoS) attacks, the systems proposed here can be further victimized by denial of energy (DoE) attacks. For example, it is possible to craft "power viruses" whose aim is to consume as much power as possible. A carefully planned attack using such viruses can significantly disrupt a distributed EAC scheme and lead to instabilities and poor performance. Protection mechanisms against such energy attacks are essential to realize the EAC vision.

5. CONCLUSIONS

In this article, we discussed the notion of energy adaptive computing that attempts to go beyond achieving energy savings based on the minimization of device idling and energy wastage. One of the goals of energy adaptive computing is to make IT more sustainable by minimizing overdesign and waste in the way IT equipment is designed, built and operated. As pointed out in this article, such an approach brings multiple new challenges in the energy management of IT systems that need to be explored more fully.

6. REFERENCES

[1] A. Corradi, E. Lodolo, S. Monti and S. Pasini, "Dynamic Reconfiguration of Middleware for Ubiquitous Computing", Proc. of 3rd Intl. workshop on adaptive and dependable mobile ubiquitous systems, London 2009.

[2] P. Emma, E. Kursun, "Opportunities and Challenges for 3D Systems and Their Design", IEEE Design & Test of Computers, Vol 26, No 5, 2009, pp6-14

[3] J. Flinn and M. Satyanarayanan, "Managing battery lifetime with energy-aware adaptation", ACM trans. on computer systems, Vol 22, No 2, May 2004, pp 137-179

[4] D. Freeman, "Digital Power Control Improves Multiphase Performance", Power Electronics Technology, Dec 2007 (www.powerelectronics.com).

[5] S. Gurun, P. Nagpurkar, B.Y. Zhao, "Energy Consumption and Conservation in Mobile Peer-to-Peer Systems", Proc. of Intl. conf. on mobile computing and networking, Sept 2006.

[6] K. Kant, "Data Center Evolution: A Tutorial on State of the Art, Issues, and Challenges", Computer Networks Journal, Dec 2009.

[7] K. Kant, "Distributed Energy Adaptive Computing", available at www.kkant.net/download.html

[8] I. Kelenyi and J.K. Nurminen, "Energy Aspects of Peer Cooperation Measurements with a Mobile DHT System", Proc. of ICC 2008.

[9] I. Kelenyi and J.K. Nurminen, "Bursty content sharing mechanism for energy-limited mobile devices", Proc. of 4th ACM workshop on Perf. monitoring and measurement of heterogeneous wireless and wired networks, 2009, pp 216-223.

[10] S. Mohapatra and N. Venkatasubramanian, "A game theoretic approach for power aware middleware", Proc. of 5th ACM/IFIP/USENIX Intl. conf. on Middleware, Oct. 2004

[11] A. Papathanasiou and M. Scott, "Energy Efficiency through Burstiness", Proc of the 5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'03), pp. 44-53, Oct 2003.

[12] V. Petrucci, O. Loques, D. Moss, "A framework for dynamic adaptation of power-aware server clusters", Proc. of 2009 ACM Symposium on Applied Computing (Honolulu, Hawaii). SAC '09. pp 1034-1039.

[13] R. Raghavendra, P. Ranganathan, et.al., "No Power Struggles: Coordinated Multi-level Power Management for the Data Center", Proc. of 13th ASPLOS, Mar 2008.

[14] J.P. Sousa, V. Poladian, D. Garlan, et al., "Task based adaptation for ubiquitous computing", IEEE trans. on systems, man & cybernetics, 2006.

[15] V. Venkatachalam and M. Franz, "Power Reduction Techniques for Microprocessors", ACM computing surveys, Vol 37, NO 3, Sept 2005, pp 195-237. (http://www.ics.uci.edu/ vvenkata/finalpaper.pdf)

[16] B.P. Wong, A. Mittal, Y. Cao and G. Starr, *Nano-CMOS Circuit and Physical Design*, John Wiley, 2005, chapter 1.