International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 7 ǁ July 2022 ǁ PP. 643-651

Towards Energy Efficiency in Computing

Kanan Dwivedi[*1]

[*1] *Department of Computer Engineering, QIS College of Engineering and Technology, Andhra Pradesh, India*
Corresponding Author:kanandwivedi@mail.com

Abstract

Today, almost everyone is aware of the energy problem facing humanity: our primary energy sources are being depleted, while demand, both commercial and domestic, is increasing. Moreover, the side effects of using traditional energy sources bring significant global environmental considerations. This paper offers an answer to the question: What is the role of computers in energy demand, and where we need to focus to reduce consumption and improve energy efficiency?

Keywords:*Energy efficiency, computing, power, energy.*

Date of Submission: 08-07-2022 Date of acceptance: 22-07-2022

I. INTRODUCTION

The expectation to develop new sources of sustainable energy has at least three decades of constant work. Steve Chu [1], US Secretary of Energy, put this situation in context: "A two-pronged approach is needed to solve the energy problem: 1) maximize energy efficiency and decrease energy consumption. and 2) develop new clean energy sources. The former will follow clean energy sources. The former will remain the lowest fruit on the tree for decades to come."

What role does IT equipment play in energy demand, and where should we focus to reduce consumption and improve energy efficiency?

In August 2007, the Environmental Protection Agency EPA [2] sent a report to the US Congress on the energy efficiency of servers and data centers. Some of the report's key findings include: - Servers and data centers consumed $61 trillion kWh in 2006, accounting for the 1.5% total energy consumption that year in the US, and $ 4.51 trillion in costs; on average equivalent to the energy needs of 5.8 millions of homes.

- The electricity consumed in this sector doubled between 2000 and 2006, a trend that is expected to continue.
- The infrastructure systems required to support the operation of IT equipment - for example protection equipment and cooling systems - also consume a significant amount of power, comprising 50% of annual energy consumption.

Excerpts from this report are shown in Figure 1 and in Table 1, and in them two points stand out: the first refers to the fact that the very infrastructure of the spaces for the operation of the computer equipment consumes a lot of energy. This infrastructure consumption is represented mainly in heating, ventilation and air conditioning; heating, ventilation and air conditioning; in addition, the one used to convert and transmit power and maintain its continuity -the latter includes transformers and power and transmission switching equipment, as well as power conditioning equipment and uninterruptible power supplies. This factor is of great importance, but it cannot be the main area faced by IT professionals.

The second point that stands out in the report is the computer equipment itself. Only the volume of servers, of the five types of IT equipment studied, was responsible for the majority of energy used: 68%. Assuming that the annual growth rate -Compound Annual Growth Rate CAGR- of the volume of servers remains at 17%, they will become the main objective to reduce energy in the IT area. The same report details the rate of the same report details the growth rate of storage devices at 20%- a rate that the most recent data suggests is accelerating - indicating another important trend.

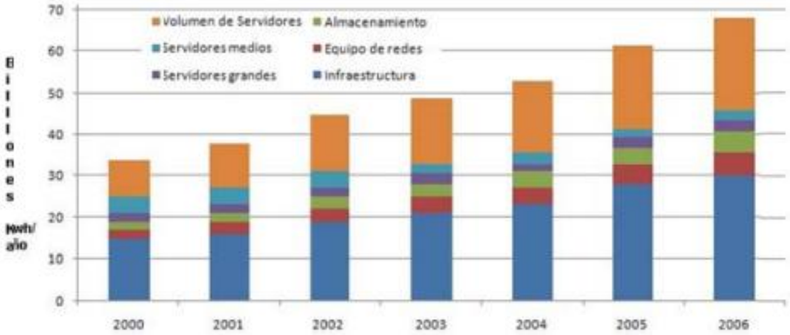

Figure 1: Electricity consumption of end-use components 2000-2006 Source: Environmental Protection Agency

Table 1. Electricity consumed by end use components 2000 to 2006

	2000		2006		2000/2006
Component	**Use of Energy**	**%**	**Use of Energy**	**%**	**Use of Energy**
Infrastructure	14.1	50	30.7	50	14
Networking team	1.4	5	3.0	5	14
Storage	1.1	4	3.2	5	20
Big servers	1.1	4	1.5	2	5
Medium servers	2.5	9	2.2	4	-2
ServerVolume	8.0	29	20.9	34	17
Total	28.2		61.4		14

If the exponential growth of data center computing equipment revealed in this study continues, it is expected that by 2011 data centers will need approximately twice the energy demanded in 2006. This poses obvious challenges beyond economic ones. For example, instantaneous peak demand is expected to increase from 7GW2006 to 12GW in 2011, so 2006 to 12GW2011 will be needed, so ten new power plants will be needed.

In some areas, physical limitations on power availability are already limiting data centers; An IT managing director at Morgan Stanley [3] noted not long ago that the company was no longer physically able to get the power to run a new data center in Manhattan. The situation is so serious that corporations such as eBay, Google, Amazon, Microsoft and Yahoo are looking for suitable places where they can build data centers, necessary to run their web applications and offer the services they currently have.

Some of these companies have already negotiated with certain states in the US, as well as internationally, for the construction of these facilities in addition to the power plants necessary to supply them. A few years ago, Google caused what some journalists Google caused what some journalists called "a modern arms race" when it placed a new facility along the Columbia River in Washington. The combined benefits of low land costs, low outdoor temperatures, ythe availability of potable water for cooling, and hydroelectric power generation could provide Google with an antidote to acute energy availability problems and associated costs.

There is some evidence 1 that the amount of energy consumed by desktop and mobile computing equipment is roughly the same magnitude as that used by servers in data centers, although there is no comprehensive and authoritative study to refer to. The EPA data presented here offers a detailed perspective on where the energy is going. in the computing landscape, in the increasingly in the computing landscape, in the increasingly important segment of servers. Furthermore, some substantiation will have this information, when the EPA's EnergyStar® program in consumer electronics put, as a result of the previous report, mobile and desktop computing in the same space as the rest of computing components.

II. METHODOLOGY

Perhaps the key factor to consider with today's computing systems is that the amount of power they consume does not sufficiently match the amount of work the system performs. The primary design goal for most computer systems in general purpose, to date, has been to maximize performance -0- perhaps performance at a given price - with very little regard for energy use. This has changed rapidly as we approach the point where the cost of computer equipment will outweigh the cost of power to operate it, even during its relatively short -35-year payback period; it is time to pay enough attention to energy in system design.

Although the exercise has been done to calculate proportional energy [4] -that is, the amount of power needed corresponds directly to the degree to which the system or component is used-, it is very far from the current situation. Many components of today's computing systems have very low utilization levels, and most systems spend much of their operating time at relatively low utilization levels.

Power supplies have been notorious for their inefficiency, especially at low load, and fans can waste a lot of power when used carelessly. However, in the last four years the efficiency of power supplies has improved [5]. In fact, algorithms have emerged that more continuously adjust fan speeds relative to thermal needs, rather than just using a few discrete speed points. However, most hardware components in today's computing systems still need to be managed explicitly, and the widely deployed conceptions and facilities for managing power in computing systems, they are still rudimentary.

Power Management

The option of operation vs. Suspension is often called "energy states", of a component or system. While there is a single state to represent operation, there can be more than one for suspend, allowing power to be suspended more progressively, if there is any power structure relevant to your implementation, from the hardware associated with the component or system . CPUs, for example, can suspend execution simply by stopping issuing instructions or turning off their clock circuitry. However, some deeper power states can successively remove power from processor caches, Translation Lookaside Buffers (TLBs), memory controllers, and so on. The more hardware components are suspended, the more power is saved, but then there will be a higher latency to resume operation, or additional power will be required to save and restore the contents of the hardware upon reboot, or both.

Performance tuning options, while running, are more naturally known as "power states" of a component. A widely applied technique for tuning performance is to change the operating frequency of the component. When the clock speed is low, the operating voltage levels can also be reduced, and these two factors together - usually called Dynamic Voltage and Frequency Scaling DVFS - are Voltage and Frequency Scaling DVFS - performance states were first introduced for CPUs, since processors are among the most important consumers of power on the hardware platform - a modern multi-core CPU is in the range of 35" " Wand 165W. Performance states can also be used to control active cache size, number and/or memory operation rates, interconnects I/O. and other similar.

The most widely implemented architecture for power management is the Advanced Configuration and Power Interface ACPI, which has evolved along with the Intel architecture, based on the most widely available hardware platforms for CPUs and their related components. Although there are many detailed aspects to the specification, ACPI primarily provides the controls necessary to implement the two modes of power management just described. In addition, it defines the power states: seven at the level of the entire system, called S-states -S0 to S6-, and four at the level of each device called D-states -D0 to D3. The state 0-S0for system and D0 for each device indicates running or active state, while higher numbers are non-running or idle states with successively lower power - and consequently decreasing levels of availability or readiness to execute. ACPI also defines performance states, called P-states -P0 through P15, allowing a maximum of 16 per device-, which affect the component's operational performance during execution. Both affect power consumption.

Energy Efficiency in the Calculation

Although ACPI is an important standard with wide support from manufacturers, it only provides a mechanism to control aspects of the system in order to affect its power consumption. This allows for but does not explicitly state energy efficiency. Higher level aspects of the overall system architecture are needed to exploit this or any similar mechanism.

How much does calculating energy efficiency differ from power management? How can you know that the problem of energy efficiency for a computer system has been solved? Here's a simple view: "The system consumes the minimum amount of energy required to perform any task." In other words, energy efficiency is an optimization problem. Said system must dynamically adjust its hardware resources, so that only those needed to perform those tasks are enabled -either to complete them on time, or by analogy, to provide the performance required to maintain an established service level-, and as a result, the total energy used will be reduced to a minimum.

Traditionally, systems have been designed to get the most performance for the workload. In energy efficient systems, in some cases you still want maximum performance for some tasks or for the entire workload, but the system must also minimize power usage. It is important to understand that performance and energy efficiency are not mutually exclusive. For example, even when maximum performance is achieved, a resource that needs to be turned off or whose individual performance can be reduced without affecting time to best complete the workload or performance, constitute power optimization.

In fact, in any system there are few situations in which the full capacity of the hardware resources is exploited - that is, all operate at their maximum performance levels. Systems that strive for maximum performance at all times are notoriously over-saturated and consequently underutilized. However, people involved in the design of practical computing systems may note that our science is weak in this area - an area that could be call "dynamic capacity planning and of dynamic provisioning.

Energy optimization is obviously subject to certain restrictions. Some examples are:

Levels should be maintained required performance

Assignments with deadlines must be completed on time. In the general case, a due date is specified for a task or workload. When a deadline is specified that is less than or equal to the optimum in which the system can accomplish that job with any or all hardware resources, it also implies a maximum performance of all hardware resources.

Maximum performance for a task or workload provides an implicit stipulation of the optimal deadline, or "as soon as possible"3. In this case, power optimization is limited to resources that can be turned off or whose individual performance can be reduced, without affecting the time to best complete the workload.

If you specify a deadline less than the best achievable deadline, the calculation can take some time until this date, and the system can look for a more global energy minimum for the task or job. The deadlines could be considered "difficult", in which case the allocator of resources to optimize system energy must somehow guarantee their compliance -which poses problems that are difficult to apply-; or considered "soft", in which case only greater effort can be tolerated.

Services must operate according to the volume of work. For online services, the notion of workload, in order to characterize the level of performance required, may be more appropriate than achieving a lead time. Since services, in their implementation, may ultimately be decomposed into individual tasks to complete, an analogous technique is expected - although the most appropriate means of specifying their performance constraint may be different.

The system must respond to changes in demand

Real workloads are not static. The amount of work set, and resources required to achieve a given level of performance will vary as they run. Dynamic response is an important practical consideration related to service level.

The volume of work (T)must be achievable within latency (L). Specifying the maximum latency at which reserved hardware capacity can be activated or its performance level increased seems like a clear requirement, but it should also be related to the performance needs of the task or job in question.

The volume of work depends on the type of task. A metric such as TPS -Transactions Per Second- may be relevant to the operation of database systems, frames per second to the rendering component of an imaging subsystem, or corresponding measures to a classification, interconnection 1/O, or other service. network interface. Interactive use imposes response criteria such as real-time distribution media: computational, storage, and capacity 1/Orequired to meet audio and video execution rates. Here is a suggested way by which such diverse workload requirements could, in practice, be handled.

The instantaneous power must never exceed its limit (P). A maximum power limit can be specified to meet your practical availability limits - either for an individual system or for a data center as a whole. In some cases, it may be permissible to briefly exceed this limit.

Combinations of these restrictions mean that over-tightening should be expected in some circumstances, and therefore a policy to restrict easing will also be required. A strict precedence of constraints or a more complex trade-off between them might be preferable.

A Solution Proposal

Taking into account this concept to calculate energy efficiency, how should the systems be built? How would an efficient power system be expected to operate?

A system should have three main aspects that could solve this problem:

1. You must be able to build a power model that allows you to know how and where power is consumed, and how you can manipulate it - this component is the basis for the enactment of any form of power management. 2. You must have a means of determining task or workload performance requirements—either by observation or by some more explicit means of communication. These are the limitations -of determination and performance- of the evaluation of components.

2. Lastly, you must implement a power optimizer - a means of deciding a power, efficient configuration of the hardware for all times that it is running. An optimization that can be relative -decided

heuristically- or absolute -based on analytical techniques. This is the planning and dynamic capacity of the provisioning component.

The first aspect is relatively easy to build. The third is certainly immediately accessible, especially when the optimization technique(s) is(are) based on heuristic methods. The second consideration is of greater proportion, and represents an important detrimental consequence for the calculation of energy efficiency, and could demand more formal bases - programming - to communicate the workload requirements to the system. For system workload requirements. This requires a description of the basic provisioning needs of the workload, along with a way to indicate performance at the moment.

A way to indicate a priori the expected sensitivity of different system resources to changes in provisioning might also be useful. Fortunately, there are a number of practical approaches to tracking energy efficiency before enabling the expected refinements.

Power Model

In order to manage the energy efficiency of the system hardware, the system must know the specific power details of the physical devices under its control. Manageable power components must display the control they offer in the ACPI architectural model), such as their power and performance states, respectively -D-states and P-states-. However, to allow modeling of power in relation to performance and availability—that is, in relation to its wake response, the relationship to its wake response—the minimum of the following:

The state transition latency, or the time required to perform each state transition.

Energy for state change, or the energy consumed when changing state.

Once the system has a power model, consisting of all the hardware that manages power, it has the basic foundation to operationalize energy optimization. Importantly, it has knowledge of the components that consume the most power and those with the highest capacity feedback controls that can be used to affect power usage.

Workload restrictions and performance evaluation

In your desire to limit the amount of active hardware and reduce its performance to minimize power consumption, how should a system be in order to know if the running tasks are still reaching the workload to maintain adequate service levels or to achieve their goals? deadlines?

The evaluation of the workload is subject to the task or application in question. The operating system can observe the degree to which its various resources have been and are being used, and could use these observations as its best basis for predicting future resource needs, both to reduce and to expand those available to it. This is a relatively weak basis for determining what workload will be needed, especially for anticipating its dynamic response sensitivity. As a result, the system will have to be much more conservative in its reduction of available resources or performance levels. It seems clear that the best result will be achieved if applications assess their workload against their service level requirements or lead times, and can pass that information to the operating system through an interface. The system can then use this information to potentially make much more potentially much more aggressive resource adjustments and apply a fix accordingly.

Here is the crucial dichotomy: the system is responsible for solving the power optimization problem according to the resources it allocates, while the application is responsible for monitoring its own level of performance and informing the system so that they are put at its disposal. provision of appropriate resources.

Energy optimization by the System

Once it knows the power characteristics of the hardware, and possibly with descriptive information about its limitations from the software application level, the operating system must begin the process of dynamically tuning hardware performance and dynamically tuning performance. hardware and availability levels to control power consumption and improve the energy use of the entire system. How can the operating system make such decisions?

heuristic methods. Provisioning power for a peak workload can, in some cases, optimize it. This is the "(maximum) performance is green" conjecture, which is reflected in the ideas of running to leisure or running to sleep [8]. Although there is evidence that this approach has merit on the client side when the system is idle - especially for embedded and mobile systems where even 95%power can be saved if the entire system is put into a sleep state - it is not. clear how it can be applied to calculate the side of the server. In some cases there is a non-linear increase in power required to obtain linear acceleration-workload, for example the turbo mode of contemporary Intel processors-and therefore, in all cases, the optimal power is not found in a point of provisioning and performance according to the maximum workload.

Dynamically adjusting the hardware performance level, based on its utilization, is a widely used heuristic for power upgrading in active systems: down with low utilization or up with high utilization -

utilization below or above a certain threshold for a certain duration. This can be an effective technique, but it is limited to situations where, to make the state change, both latency and power are so low as to be negligible.

Constraint-based optimization for a focus. In some cases it may be possible to offer a complete analytical solution to simplify the problem to such a degree. For example, considering only a single task on a single CPU with a well-understood power/performance tradeoff, it is relatively easy to fully specify a schedule in which it is possible to match the task to its due date with a minimum total energy. More generally formal results are also possible [9]. However, this is based on a number of assumptions that are often not made in practice, such as good estimates of the total work required by a process. Weaker assumptions require online optimization algorithms to achieve energy-aware scheduling. There is some work in this area, but not yet enough to support a general purpose operating system [10].

For an optimization-based approach to be generally applicable, a number of techniques are required. In the simplest cases, autonomous operation at the device level is possible; for example, a GPU -Graphics Processing Unita hardware level, can power down without using hardware facilities aggressively, based solely on instantaneous appreciations of its utilization levels, because while latency is needed to back up of those facilities, will be inconsequential. Similar practices seem to apply in the use of CPU P-states - CPU performance and cost-energy tuning based on voltage and frequency scaling - since both state transition energy and latency are Very low.

Hardware state changes that affect power but have a much higher latency, and/or a much higher amount of energy to make the state change, require different treatment. An obvious example is a hard drive slowing down, a situation in which you have to take into account the long latency required to restart it; but this latency is not the only concern. Semiconductor memory systems in which part of the total physical memory could be turned off if not required, and where the turn-on latency may be close to zero, will still have a resulting transition energy, since many transactions may be required. in memory to collect the working set on those physical pages that remain active. Resources of this class require, to ensure that the activation latency can be tolerated or maintained and that the state change energy will be outweighed by the energy that will be saved while in that state, greater knowledge of the task or behavior of the workload, as well as an anticipatory treatment of the required hardware resources.

Some common optimization techniques can be based on state change latency, their energy demands, and so on, and from these a taxonomy of them could emerge - some formal or analytical, others based on more numerical or heuristic methods.

Although specific techniques are expected, for the proper optimization of energy to different hardware resources or subsystems, is somewhat different and subject to the properties of the hardware resources in question, the hope is that the composition of energy efficiency optimizers for all resources will add up to form an efficiency scheme for the system in question.

How to do it

The vision of full system concessions for energy efficiency cannot be achieved in one single step. Current software systems are not equipped in the ways described, nor are applications written in a way that takes advantage of that capability. In pragmatic terms, how can this result be achieved, and what measures are already in place for this result?

As a first consideration, systems should be reviewed to pay attention to energy use; the operating system itself, which is always running, has not yet optimized its own power usage. To date, almost all software, including system software, has been optimized for performance, robustness, and scalability without considering power. An initial step, therefore, is to redesign and implement energy efficiency in operating system operation. This is an important task, and its implications are not yet fully understood.

It's not clear whether it's feasible to modify existing operating systems to treat power as a first-class hurdle, which would certainly be preferable. Experience with security systems shows that it is very difficult to introduce such considerations after the fact. To be sure, and due to the pressure of energy efficiency, fundamental new structures can be anticipated within systems software, and even within systems software, and even new operating systems will emerge. At the very least, the facilities for energy awareness and optimization should be adapted.

Processors. Given the significant fraction of power attributed to CPUs on contemporary platforms - and the consequent early introduction of power management features into them - much progress has been made in programming operating systems. The careless activation of hardware when it is not useful for some work should be removed. The constant revision of a program "director" within the operating system -0of applications - is an obvious example; but it uses a high-frequency clock with base interrupts to time events, handle normal time, and schedule thread execution, which can be quite problematic. The goal is to keep the hardware idle until it is needed.

The "tickles" kernel project [11] on Linux introduces an initial implementation of dynamic tick marks. By rescheduling the periodic per-CPU interrupt timer to remove clock tick marks during sleep, you can improve the average amount of time a CPU spends to maintain its sleep state by a factor of 10 or more after each idle state input.

Beyond the good ideas coming from Linux, the Tesla project on Open Solaris is also considering what the transition to a broader event-driven approach to developing software on the operating system means.

The confluence of features in modern processors CMT -Chip Multi threading-, CMP Chip Multiprocessor-, and NUMA -Non-Uniform Memory Access- for multi-socket multi-processor systems, is generating from developers the presentation of a large number of new works aimed at implementing energy efficiency [12]. Taking into account the ability to alter performance levels, power efficiency and the introduction of heterogeneous multi-core CPUs, only this component remains to be added [13,14].

Storage. Compared to CPUs, the power consumed by a disk drive doesn't seem particularly large. A typical 3.5" 7200 RPM drive consumes between 7" " W4W and 8W - just 10%as much as a typical multi-core CPU consumes. 14W of 15000 RPM maybe you can use around alarming growth rate relative to alarming growth rate relative to storage, you could rapidly change the percentage of total power that storage devices represent Reliability performance factors have given rise to the application of spindles multiple, so common today, even in desktop systems to implement simple RAID solution Storage solutions are scaling up much faster in the data center.

Low-end server chassis, routinely used in homes, have room for a dozen or more drives. For example, Sun's rack 4Ufor storage arrays holds 46 3.5" units. So using industrial 10000 RPM or 15000 RPM units could increase power consumption to between 1.088W and 1.6kW.

Currently, storage subsystems are obviously on energy's watchful radar. There are at least two immediate steps that can be taken to help improve the power consumption of storage devices. The first is direct attention to power usage in traditional disk-based storage. Part of this work was started by vendors to introduce power states to their disk drives, and part of this work was started by vendors working on current operating system developers. file systems -such as Zettabyte File System ZFS-, and in the management of storage resources. The second stems particularly from the recent introduction of large, low-cost Flash memory devices. Flash memory fills a significant capacity/performance gap between main memory devices and disks [15,16], but it also has huge energy efficiency advantages over mechanically rotating media.

Memory. Due to its relatively low power requirement - for example, DIMM - main memory appears at first glance to be an even less important concern than disks. Its average size on current hardware platforms, however, current hardware platforms, however, may position it to grow faster. With the focus of hardware system builders on performance levels - to keep up with the corresponding performance demands of multi-core CPUs - it is critical to always maintain the bandwidth between the CPU and memory. . The consequence has been an evolution in DIMM modules from single to dual and now to triple channel, in the corresponding DDR, DDR2 and DDR3 SDRAM technologies. Although the reductions in the size of the manufacturing processes - DDR3 is in the 50 nanometer technology- has allowed the clock frequency to be raised and the power per DIMM to be lowered a little. The desire for even higher performance by increasing DIMMs per memory channel continues to increase the total power consumption in the memory system.

For example, a server with a four-socket system, based on Sun's Niagara2 octa-core CPU, with 16 DIMMs per socket and using dual-channel DDR2 memory technology, has a total of 64 DIMMs; and could be increased to 24 DIMMs per socket -96in total - if its faster successor uses triple-channel DDR3 memory instead. A representative DDR2 DIMM consumes 1.65W--3.3W per pair, while the lowest power edition of current DDR3 DIMMs consumes 1.3"W-3.0W per pair. The result appears to be an increase 20%in power consumption only - in total between 100" " W120W and 120W for our example.

However, considering that the next generation of CPUs will also have twice as many cores per socket, one possible resulting scenario is the desire to have twice the number of memory pools per socket for a possible total of 192 DIMMs. with the goal of balancing overall memory system performance. The result, therefore, could be anincrease of 100" " Wup to 240" " W140% more power consumption for the entire memory system! This trend is seen even on desktop-class machines, no doubt on a much smaller scale, as systems have emerged containing hyperthreaded quad-core CPUs - such as Intel's Nehalem.

If available physical memory can be turned on and off, and perhaps correspondingly reconfigured as a system processing capability, some new functionality that will be required of the operating system's memory management subsystem can be dynamically tuned. Looking to the future, it is an open problem to design a virtual memory system that is energy aware, and that is capable of adjusting the physical resources of the memory while it is running.

I/O. The energy aspects of the system I/O on hardware platforms are probably more important. As a simple example, current local area network interconnected systems and subsystems have developed in two important respects: link aggregation, which is increasingly used to power networks and broadband reliability;

and individual interconnect speed, which has advanced from 1GB to 10GB, and is expected to hit 40GB. A transceiver for a network interface card 10GB today may require, at most, 14W when running at full speed, with a consequent reduction in power when its link speed drops to 1GB or below. Other high-speed interconnections can be expected. like InfiniBand, have similar power considerations for the entire system. Little attention has been paid to the energy implications of communication interconnections in any of their different architectural manifestations, from a chip to wide area networks.

The evolution of application software

The most strategic aspect of energy efficient computing will be the evolution of application software to facilitate energy efficiency of the entire system. While we can certainly expect new application interfaces for system software by supporting the development of new energy-efficient applications, the transition from historical and current applications represents a long-term evolution. In the meantime, how are we going to address the issue of increased energy efficiency for the rest of the installed base? Obviously, it won't happen as a result of a one-time deployment of all existing applications.

One possibility to address the energy agnosticism of existing applications is to perform an extrinsic analysis of their behavior at runtime. Empirical data can be collected on the degree to which application performance is sensitive to different levels and types of resource provisioning. For example, one can observe the degree to which performance 8 is increased by the addition of performance is increased by the addition of resources to the CPU, or the allocation of a CPU with a higher performance micro-architecture, and so on [14]. The application can then indicate, in its binary form, what the measure of its degree of sensitivity is, without the need to alter the existing implementation. The operating system could use the data to allocate resources to target a specified level of performance, or to locate a suitable tradeoff of power consumption vs. performance.

Inevitably, a combination of both techniques is expected to be needed, in which the application itself explicitly informs the system of its workload and resource provisioning needs; and implicitly that static and dynamic analyzes are used to model the need for resources, relative to performance and power consumption.

III. CONCLUSION

We are still in the debut of computational energy consciousness; with much attention from industry, which has enabled the introduction and use of power management mechanisms and controls in individual hardware components, but little attention to broader issue of energy efficiency: minimizing the total energy required to run computational workloads on a system.

In this work, a general approach to energy efficiency in computational systems is proposed, and the implementation of energy optimization mechanisms in the system software, equipped with a power model for its hardware, is proposed; and describes the applications that suggest adjustment of describe the applications that suggest adjustment can reach their workload levels and/or required execution times.

In the short term, a series of heuristic techniques designed to reduce the more obvious energy waste associated with high-power components, such as CPUs, will be required for practical application. In the longer term, and for a more effective optimization of total power, techniques will be needed to model the performance relative to the system's hardware configuration - and thus its power consumption - along with a better understanding and predictive knowledge of workloads.

REFERENCES

[1]. S. Chu, "The energy problem and Lawrence Berkeley National Laboratory." Talk given to the California Air Resources Board. 2008. In: http://www.arb.ca.gov/research/seminars/chu/chu.pdf May2010.

[2]. Environmental Protection Agency, "EPA report to Congress on server and data center energy efficiency". In: http://www.energystar.gov/index.cfm?c=prod_devel opment.server_efficiency_study, Feb, 2010.

[3]. R. H. Katz, "Tech titans building boom". IEEE Spectrum, Vol. 46, No. 2, pp. 40-54. 2009. 4. L. Barroso & U. Holzle, "The case for energyproportional computing". IEEE Computer, Vol. 40, No. 12, pp. 33-37. 2007.

[4]. M. Kyrnin, "PC Power Supply Efficiency. How The Efficiency Rating of a Power Supply Can Save You Money". About.com guide. 2010. In: http://compreviews.about.com/od/cases/a/PSUEffici ency.htm, May. 2010.

[5]. Energy Information Administration, "Residential Energy Consumption Surveys 2001". 2001. In: http://www.eia.doe.gov/emeu/recs/byfuels/2001/byfu els_2001.html, May 2010 .

[6]. Mariam Manakkadu, Sheheeda. "POWER-AWARE PERFORMANCE OPTIMIZATION ON MULTICORE ARCHITECTURES."

[7]. K. W. Roth & K. McKenney, "Energy consumption by consumer electronics in U.S. residences". Cambridge: TIAX LLC. 2007. 147 p.

[8]. M. Garrett, "Powering down". Communications of the ACM, Vol. 51, No. 9, pp. 42-46. 2008.

[9]. D. J. Brown & S. C. Reams, "Toward energyefficient computing". Communications of the ACM, Vol. 53, No. 3, pp. 50-58. 2010.

[10]. F. Yao, A. Demers & S. Shenker, "A scheduling model for reduced CPU energy". Proceedings of 36th Annual Symposium on Foundations of Computer Science. Milwaukee, WI, USA, pp. 374382. 1995.

[11]. S. Siddha, "Getting maximum mileage out of tickles". Proceedings of the Linux Symposium. Ottawa, Canada, pp. 201-208. 2007.

[12]. A. Fedorova, "Operating system scheduling for chip multithreaded processors". Ph.D. dissertation. Harvard University, The Division Of Engineering And Applied Sciences. 2006.

[13]. A. Fedorova, J. C. Saez, D. Shelepov& M. Prieto, "Maximizing Power Efficiency with Asymmetric Multicore Systems". ACM Queue Magazine, Vol. 7, No. 10. 2009.

[14]. D. Shelepov, J. C. Saez, S. Jeffery, A. Fedorova, N. Perez, Z. F. Huang, S. Blagoduro& V. Kumar, "HASS: a scheduler for heterogeneous multicore systems". ACM Operating System Review, Vol. 43, No. 2, pp. 66-75. 2009.

[15]. A. Leventhal, "Flash storage today". ACM Queue, Vol. 6, No. 4, pp. 25-30. 2008.

[16]. J. Mogul, E. Argollo, M. Shah & P. Faraboschi, "Operating system support for NVM + DRAM hybrid main memory". Proceedings of UsenixHotOS XII. Monte Verità, Switzerland.2009.

[17]. Jayaprakash, Stanly, Manikanda Devarajan Nagarajan, Rocío Pérez de Prado, Sugumaran Subramanian, and ParameshachariBidareDivakarachari. "A systematic review of energy management strategies for resource allocation in the cloud: Clustering, optimization and machine learning." Energies 14, no. 17 (2021): 5322.

[18]. Manakkadu, Sheheeda, Sourav Dutta, and Nazeih M. Botros. "Power aware parallel computing on asymmetric multiprocessor." In 2014 27th IEEE International System-on-Chip Conference (SOCC), pp. 35-40. IEEE, 2014.

[19]. Muneeswari, G., Prabha Selvaraj, Vijay Kumar Burugari, and P. Kanmani. "Hybrid Scheduling and Authentication for Legacy Uniprocessor and Multicore System." In 2022 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5. IEEE, 2022.

[20]. Srivastava, Gautam, and Revathi Venkataraman. "A review of the state of the art in business intelligence software." Enterprise Information Systems 16, no. 1 (2022): 1-28.