A Case for Energy-Aware Accounting in Large-Scale Computing Facilities

Cost Metrics and Implications for Processor Design

Víctor Jiménez † Francisco J. Cazorla † Roberto Gioiosa † Eren Kursun *
Canturk Isci * Alper BUYuktosunoglu * Pradip Bose * Mateo Valero †

† Barcelona Supercomputing Center, Spain {victor.javier, francisco.cazorla, roberto.gioiosa, mateo.valero}@lsc.es
* IBM T. J. Watson Research Center, Yorktown Heights, USA {kursun, canturk, alperb, pbose}@us.ibm.com

1. INTRODUCTION

Energy and power trends in large-scale computing facilities (LSCF) are likely to shape the way next-generation facilities are designed, built and maintained. The electricity demand from LSCF shows the fastest growth among all sectors. In fact, U.S. Environmental Protection Agency (EPA) estimates that national energy consumption due to LSCF will soon reach more than 100 billion kWh annually [1], with an associated $30 billion electrical cost [6]. Given the fact that the cost of energy is on the rise, recent studies show that energy already accounts for 20% of the total cost of ownership (TCO) in a LSCF. This cost increases up to 40% if we add the cost for the cooling infrastructure [3].

Despite the above energy consumption trends, user- or task-specific accounting for energy or power consumption is very limited. The accounting method applied for user-level billing, is usually based simply on time and size of resource usage. However, the exact level of resource utilization is typically not considered, and power consumption attributable to a specific user job is estimated based on peak (or nameplate) values for used resources. This is clearly not fair, since different customers may incur different utilizations across similarly allocated resources, and yet result in near-identical usage time (and bill amount). Additionally, the cost for the facility owner may significantly vary as well.

In order to elaborate upon the need for accurate, energy-aware accounting principles, we consider several benchmarks as proxies for the behavior of applications executed by different users on a small system. We execute all the SPEC CPU2006 benchmarks on an Intel quad-core server system. A 10% variation in power across workloads is typical, with the maximum variation being 20%. Figure 1a shows a subset of the results with one example for low, medium and high consuming workloads. So, user workloads executing for the same length of time would incur energy usage levels that may actually differ by a margin of 20%; yet, current accounting practices would bill them equally. Another illustrative example is shown in Figure 1b, showing the results of executing the SPECpower benchmark [4], on two different Intel Xeon systems. This example is representative of variable-demand workloads, with considerable different power consumption for different CPU utilization levels.

A desirable solution for the static energy consumption problem is to obtain energy-proportional systems [2], in which power is close to zero when the system is idle and power linearly increases as performance increases as well. Although current systems are not energy-proportional yet, the trend is to move towards this kind of systems. In the presence of truly energy-proportional systems, the static power cost would be almost entirely eliminated, and the dynamic cost would account for most of the energy consumption. As all the energy consumed by the systems will be a consequence of application activity, considering energy consumption for accounting purposes becomes very attractive.

In this paper, we make a case for energy-aware accounting in LSCF. The adoption of accounting metrics, based on accurate measurements of actual resource utilization levels, by the facility owner would drive up energy-efficiency in computing facilities, without hurting the owner’s bottom-line profit margins. Moreover, directly including the energy cost in the bill, with a detailed breakdown of resource usage would increase the energy-awareness within the user community. This would motivate users to optimize their codes and deployment configurations; and, competition would drive users towards progressively “greener” computing facilities.

1.1 Target Facilities

Different LSCF employ vastly different provisioning models with distinct quality of service and cost models. In this work we differentiate between systems where the provisioned resources can be dedicated or shared. With dedicated provisioning, commonly employed in HPC clusters, some number of physical nodes are leased to the end-user. In this model, the overall operation and power cost of the leased nodes can be easily attributed to the running applications, using a per-node energy accounting. A second approach is to provide shared hardware resources where the applications can share nodes with other applications, usually via virtualization. As the applications are not directly linked to physical hardware, direct hardware profiling is not generally available at the application level. In this case, the contribution of each application and virtual machine (VM) to energy consumption depends on provisioned virtual resources, the imposed resource constraints and the underlying resource sharing mechanism. In addition, many virtualization technologies also employ additional resource optimizations (e.g., page sharing) that difficult per-application energy tracking.
we have to decide how much overhead we allow in
order to track energy consumption.

**Fairness:** From the user/client perspective is important to ob-
tain the same energy-accounting result for the same input, re-
gardless of the applications it is co-scheduled with. However, in
reality a number of factors complicate the ideal-case.

**Accuracy vs. Variation:** Supply Heat Indexes (SHI) indicate
unequal cooling profiles for servers at different locations of the
computing facility. As a result the underlying variation in the
computing facility may dominate the power dissipation vari-
aton, causing significant differences among identical applications
running on same type of servers.

Next we develop some of the previous points and discuss the
design trade-offs for effective energy accounting.

### 2.1 Static/Dynamic Power

In order to accurately track energy consumption, we need
to break down power-related costs between static and dynamic
costs. The former accounts for power that does not depend on
the activity on the system while the latter accounts for the ex-
tra power consumed when there is activity on the system. When
nodes are not shared among users, that distinction is not really
necessary as total power consumption can be typically measured
at the node level. For shared environments we must estimate the
fraction of these components that must be attributed to ev-
ery running application.

**Static Power.**

Splitting the cost of static power consumption among applica-
tions depends on the level at which resources are shared, leading
to several possibilities with different associated accuracies and
overheads. The easiest solution is to evenly split static con-
sumption among the applications mapped to that node. If a
higher accuracy is desired it is possible to individually look at
the subcomponents making up the system. We differentiate two
subcomponent types based on their nature:

**Spatial-sharing:** In subcomponents spatially shared (e.g., cache,
memory) there is a linear relation between the amount of space
demanded by a user and the cost of static power. If in a given
instance a resource with an associated space of $M_{total}$ bits has a
static power consumption of $S_{total}$ Watts, it can be broken down
among $N$ users as follows: $S_i = (M_i/M_{total}) \cdot S_{total}$, where $M_i$
and $S_i$ are respectively the amount of space used and the static
consumption incurred by user $i$.

**Temporal-sharing:** Temporally shared components (e.g., CPU,
hard drive) consume static power proportionally to the duration
they are enabled. In this case we can use an interval-based ac-
counting approach: we divide the time into intervals of fixed
length $I$. If during a given interval a certain amount of ap-
lications access the device, all its static power consumption is
charged to those applications. The other running applications
are not charged anything, since we assume that the subcom-
ponents can go into a low-power mode if it is not accessed for an
interval $I$.

**Dynamic Power.**

Splitting the dynamic power consumption between applica-
tions is a complex task that in some cases may require hardware
and/or software support.

**Request-based:** CPU utilization and the number of requests
per unit of time are high level metrics that typically correlate
well with power consumption, and hence energy consumption as
well. If a higher accuracy level is desired, energy consumption
be estimated based on lower-level metrics by using perform-
ance counters or OS statistics.

**CPU-intensive:** In this case, high-level generic metrics are
generally less useful. CPU utilization for this kind of applica-
tions is mostly close to 100%, rendering utilization-based power
estimation inapplicable. Application-specific, high-level metrics
can be used, but this solution is not portable among different
applications. For this kind of applications event-based metrics
are a much better fit to accurately estimate energy consumption.

### 2.2 Interferences

In shared environments, although application’s output will not
change from run to run, the actions taken by the system to
obtain this output could differ from run to run. For instance,
the aggregated memory footprint of both applications can exceed
the amount of cache or memory installed in the system, leading
to memory or disk accesses that would not take place if the
applications ran in isolation. Another source of interferences
is system activity due to housekeeping (freeing virtual memory,
cleaning system logs, etc.) Finally, VM optimizations across
VMs create interactions among user environments as well. The
challenge here is to determine how to account for the energy
that the system consumes considering such interference.

### 2.3 Hardware/Software Support

Some currently available systems already allow to obtain power
measurements at the processor level. A standard and accurate
way to obtain similar measurements for the most consuming
subcomponents in a system can greatly enhance the accuracy of
energy accounting. Although performance counters can be used
as a power-proxy, other possibilities exist: including hardware
support to obtain the instruction mix per thread can already
provide a considerably accurate power consumption estimation.
Hardware support to overcome application interference can also
help to improve the accuracy of energy accounting.

Software mechanisms can enhance/complement hardware mech-

enisms used to mitigate the effect of application interference, by
tracking the time that resources are being used by the OS itself,
without contributing to a direct profit for the user. Interaction
between the accounting system and the VM monitor can help
to track energy usage in the presence of VM optimizations.

### 3. PUTTING IT ALL TOGETHER

Energy consumption in LSCF is increasing and it is becoming
a bigger fraction of their TCO. We argue that, in this scenario,
introducing energy accounting will benefit both end users and
facility owners. Additionally, energy accounting can trigger a
spiral process that leads to “greener” facilities and reduce the
carbon footprint associated with these facilities.

### 4. REFERENCES

[1] EPA Report to Congress on Server and Data Center


A. Buyuktosunoglu, and M. Valero. ITCA: Inter-task

X. Zhu. No “power” struggles: coordinated multi-level