An Energy Case for Hybrid Datacenters

Byung-Gon Chun†, Gianluca Iannaccone†,
Giuseppe Iannaccone⋆, Randy Katz‡, Gunho Lee‡, Luca Niccolini⋆

†Intel Labs Berkeley, ‡University of California at Berkeley, ⋆University of Pisa

ABSTRACT

Reducing energy consumption in datacenters is key to building low cost datacenters. To address this challenge, we explore the potential of hybrid datacenter designs that mix low power platforms with high performance ones. We show how these designs can handle diverse workloads with different service level agreements in an energy efficient fashion. We evaluate the feasibility of our approach through experiments and then discuss the design challenges and options of hybrid datacenters.

1. INTRODUCTION

Energy consumption of the computing infrastructure has become a major concern for industry and society. Today's datacenters, the backbone of the computing infrastructure, are limited in scale by the costs associated with power (distribution, cooling, density). Studies estimate that power-related costs represent already almost 50% of the operating cost of a datacenter and they are growing faster than compute-related costs (i.e., server and network equipment). Energy efficiency is now a first-class design concern at all levels – computation and data processing, power distribution at the rack and server level, power generation and transmission, etc. Companies such as Microsoft and Google are deploying new datacenters near cheap power sources to mitigate energy costs. Processor manufacturers are pursuing their roadmap of multi-core architectures [9] and low-power designs [14]. Several research proposals deal with power efficient designs and protocols for specific workloads [12], office environments [6, 20] and high speed networks [19].

In this paper, we look at one specific aspect: energy efficient clusters for large datacenters. As a first step, we consider the current trends in server designs and try to exploit them to our advantage. Traditionally, power efficient designs attempt to find the right balance between two distinct, and often conflicting, requirements: (i) deliver high performance at peak power (i.e., maximize compute capacity for a given power budget) and (ii) scale power consumption with load (i.e., energy proportionality and very low power operations). A fundamental challenge in finding a good balance is that, when it comes to processor design, the mechanisms that satisfy the two requirements above are significantly different. High performance requires mechanisms to mask memory and I/O latencies using large multi-level caches (today's server processors use three cache levels with the last-level cache projected to soon reach 24MB [3]), large translation lookaside buffers, out-of-order execution, high speed buses, and support for a large number of pending memory requests. These mechanisms result in large transistor counts leading to high leakage power and overall high power consumption. In a modern processor, less than 20% of the transistor count is dedicated to the actual cores [13, 21].

Low power designs, on the other hand, focus on those processor features with low power operations. For example, the Atom processor [14] includes an in-order pipeline that can execute two instructions per cycle, a small L2 cache and power-efficient clock distribution. This results in a strongly reduced transistor count with low leakage power and limited power consumption at low load. Further, Atom design is focused on allowing quick and frequent transitions to a very low power state (e.g., 80 mW with less than 100 μs exit latency [14]). Proposals like FAWN [12] and Marlowe [5] explore these features to build arrays of low power servers that operate efficiently for specific I/O bound workloads.

In summary, we observe a dichotomy between low power and high performance system designs. Choosing the most appropriate design for an energy efficient datacenter is far from straightforward. First, datacenter workloads are diverse – some (e.g., I/O-bound map/reduce like) lend themselves rather easily to low power designs while others (e.g., transactions, encryption) depend on high performance and fast response times to satisfy stringent service-level agreements (SLAs). Second, the workload dynamics—including job arrival patterns and completion times—may reverse the conclusion of static workload analysis. Finally, the processor is just one contributor to the overall power consumption. Other system components such as the motherboard (e.g., I/O and memory controllers), DRAM banks and power supplies contribute to a large fraction of the overall power consumption and tend not to be optimized for low power operation.

Given these challenges, we propose a hybrid datacenter architecture that mixes low power systems and high performance ones. In Section 2 we perform a preliminary evaluation to highlight the potential of hybrid solutions. Then in Section 3 we layout the challenges of such solutions and explore the design options in this space by giving a quick overview of the spectrum of solutions. Finally, we conclude the paper with a summary of related work.

2. THE CASE FOR HYBRID APPROACHES

As a first step, we are interested in comparing the performance of different systems under datacenter-like workloads. For this task, we consider a quad-core, dual-socket Xeon system and two low-power Atom-based PCs. Table 1 summarizes the characteristics of the three systems. They are representative of high performance systems currently common in datacenters and of low power platforms that are used today to build energy efficient netbooks.

Name	Xeon L5420	Atom 330	Atom N270
Frequency	2.5GHz	1.6GHz	1.6GHz
Cache	2x6MB	2x512KB	512KB
CPU	2	1	1
Cores/CPU	4	2	1
Threads	1	2	2
RAM	16GB	2GB	1GB
Storage	15k SAS	5.4k SATA	SSD

Table 1: Servers under test

Defined the platforms, we pick the set of workloads that are representative of large datacenters. We classify workloads in three broad categories:

Web Services—this is the classical web workload to serve pages to users. The data requested is usually a small object in a large dataset (e.g., an item on sale in an e-commerce site such as Amazon). The first request may lead to a database query but subsequent requests are cached in memory for fast retrieval – memcached [11] is an example of this approach for large clusters and currently used by LiveJournal, Facebook, and others. We use a simple Apache/PHP benchmark to emulate this class of applications.

Data Mining—this second class is representative of large-scale data analysis workloads that process a data set in a distributed fashion. This is typically done to populate the index used in search engines or for machine learning operations, e.g., to drive recommendation engines. To emulate this workload, we use Hadoop [1], an open-source MapReduce [10] implementation, with a pseudo cluster configuration on a single server. The maximum number of mappers and reducers running on a node is set to twice the number of cores in the servers: this setup showed the best performance. We consider two applications that make large use of disk I/O and are available in the Hadoop distribution: "word count" over 10GB of data and "sort" over 1GB.

Compute Intensive—the third type of workloads model CPU intensive applications such as image processing or video encoding. They may operate on a smaller data set but require a significant amount of computation for each data object. To emulate this class of workloads on a datacenter environment we use the Hadoop "pi" application that estimates the value of π using the Monte Carlo method and use `ffmpeg` to convert a file from Windows Media (.wmv) to Flash Video (.flv).

Performance per Watt. Figure 1 compares the performance/watt of the three platforms over the above workloads. Except Web/PHP, the performance is measured as the rate of execution (i.e., one over total execution time). For Web/PHP, we measured the number of concurrent users supported under a certain latency SLA (99% of requests are served within 100ms). The performance is compared to the power consumption of the system measured at the wall socket. To easily compare the workloads in one graph, we normalized the results to the performance/watt of the dual-core Atom 330 system.

A few observations can be made from Figure 1. First, there is no clear winner in terms of performance per watt. Depending on workload, different platforms show best performance per watt (i.e., power efficiency). For data mining workloads (I/O bound) both low power architectures show a clear advantage (Atom 330 is 3-4x

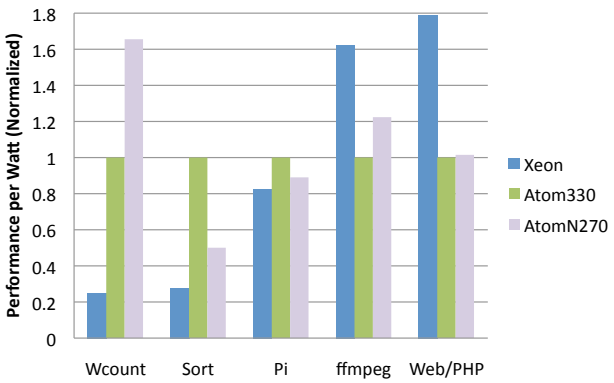

Figure 1: Performance per watt (normalized) of the platforms across different workloads

better than Xeon), while for more traditional web or compute intensive workloads the Xeon server is still the platform of choice. This suggests that mixing different platforms can be better in terms of power efficiency with diverse workloads.

Further, the least power hungry architectures, the Atom N270, exhibit the best performance/watt for WordCount compared to the other servers, but very little gain in performance/watt compared to Xeon processors for other workloads. This is due to the specific mini-PCI solid state drive (SSD) used in the system that provides good read throughput but very low write throughput. WordCount benefits from this characteristics as it is mainly reads.

Energy Proportionality. A second important aspect is to understand how power consumption scales with load. To explore this further we used the SPECpower benchmark [4] that can issue a variable number of transactions to test the platforms under various load levels. Figure 2 shows the power consumption of a Xeon server and multiple Atom servers as a function of the number of transactions per second. To compare the performance over the entire range of load levels we consider the case of adding more Atom-based PCs to handle the additional load, thus having the step shape of the Atom curves in the graph. These numbers are optimistic, estimated numbers when we assume that there is little overhead in using a cluster of Atom servers.

We can make two main observations from this figure. First, from the experiments we can see how a set of Atom-based platforms could be used to mimic an energy proportional system. As load increases, the aggregate consumes power proportional to load at a macro level.

Second, at a micro level, both platforms in isolation show a quite narrow range of power consumption across a wide range of load levels. This is in line with prior work [7] that indicate that other system components, not the CPU are responsible for the poor scaling of power consumption. The Atom motherboards we used in this test come with power-hungry chipsets (the consumption when idle is in the order of 30 W). In particular the memory controller (a.k.a. "Northbridge") is a three year old design built with a 90 nm process resulting in a power consumption of around 22 W of the chipset alone. More recent chipset designs (e.g., Intel 945GSE Chipset) manage to reduce the power consumption to around 6 W.

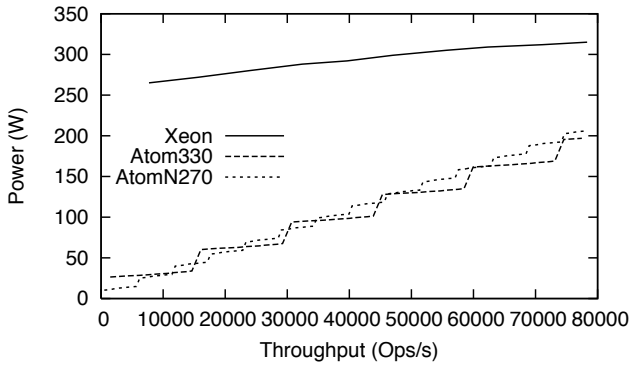

Figure 2: Throughput vs. power under SPECpower

Temporal Characteristics of Workload. So far we have only looked at performance of the system under a static workload, i.e., ignoring the task arrival process. Reports from datacenter operators indicate that servers run between 10% and 50% of their maximum utilization levels [7]. Servers process a continuous stream of task requests and operators try to distribute evenly across the datacenter to avoid high loads and meet latency SLAs.

To explore the potential of hybrid solutions, we investigate two possible hybrid solutions using one Xeon and one Atom platform by comparing them with Xeon-only or Atom-only solutions. Our goal here is not to design a specific strategy but rather to explore the performance and feasibility of such solutions for simple scenarios. The hybrid solutions that work well with multiple Atom and multiple Xeon platforms will be more challenging.

- $Hybrid_1$ implements task migration from the Atom to Xeon when the load exceeds the capacity of the Atom platform. Specifically, when the number of concurrent tasks exceed the value of T_{atom}, the Atom platform wakes up the Xeon server, suspends the execution of the tasks and migrate them to Xeon and finally goes to standby state. Tasks are migrated back to Atom when the number of running tasks goes below $T_{back} < T_{atom}$.

  For $Hybrid_1$, we look at two cases "H_0" and "H_1". The difference between H_0 and H_1 is only in the time required to migrate tasks and to wake up the platforms: H_0 represents the ideal case where both migration and waking up is instantaneous; H_1 describes another extreme where the time to wake up the server is 10% of the duration of the task and the migration cost is 50% of the task duration. $T_{back} = 3$ in all experiments. During the execution of tasks only one platform is running and the consumption of the other is given by P_{sleep} (Table 2).

- $Hybrid_2$ (denoted as "H_2") represents the case where there is no migration but all tasks are completed on the same platform where they started. In this scenario the Atom always runs and wakes up the Xeon only when the number of concurrent tasks exceeds T_{atom}.

To evaluate these, we create a simplified model of task arrivals to simulate dynamic workloads and derive latency and power consumption over time of several simple hybrid designs. We generate tasks with interarrival times derived from a Pareto distribution

Platform	T	C	P_{idle}	$P_{100\%}$	P_{sleep}
Xeon	8	1	259.5W	315.0W	18.0W
Atom	4	0.37	25.6W	33.8W	2.0W

Table 2: Model parameters (from SPECpower results)

(with shape parameter $\alpha = 1.3$). We have also used other distributions (e.g., exponential); we omit results from this paper since their trends are similar.

To model the computing capabilities of the low power and high performance platforms we use two parameters (Table 2): number of threads (T) that corresponds to the number of task requests that can be served in parallel, and the computing capacity (C) normalized to Xeon performance. The actual ratio between Xeon and Atom is derived from the SPECpower experimental results. We use only Atom 330 in our modeling because Atom 330 and Atom N270 showed similar execution times. For simplicity, in this synthetic workload we assume identical tasks that require a constant processing time. Finally, we compute the power consumption $P(t)$ as $P_{idle} + (P_{100\%} - P_{idle})U(t)$. It grows linearly between the minimum idle power (P_{idle}) and the power at full utilization ($P_{100\%}$) as observed experimentally. The utilization $U(t)$ is defined as the ratio between the number of active tasks and the maximum number of threads T.

Figure 3 shows the power consumption and the 99.9th percentile of the response time as a function of the average load (varied using the scale parameter of the Pareto distribution of the task interarrival times). The execution time (τ) is normalized to the execution time on an unloaded Xeon server. The curves labeled "Xeon" and "Atom" correspond to solutions where only Xeon or Atom platforms are used. As expected the curves follow the same trend as in Figure 2 for power consumption while response times grow in an uncontrolled fashion as the load approaches 70%.

Figure 3 (left) shows that very simple hybrid solutions (even with just one Atom and one Xeon platform) achieve good energy proportionality. Never migrating tasks appears to be a feasible strategy that leads us to believe that simple software solutions could be within reach. In Figure 3(right) we plot the 99.9th percentile of the task completion time. Interestingly, H_0 and H_2 show a latency equal to the minimum Atom latency for an utilization well above 50% (the upper limit according to [7]). The average latency (not shown in the figure) approaches the latency of the Xeon server. This shows that if running a task on an Atom platform satisfies datacenter SLAs, then a hybrid solution can preserve that guarantee. However, not all strategies are feasible as shown by H_1 where the latency is dominated by the migration cost and wake up time.

Figure 4 shows one run of H_2. As the load on the Atom system reaches the maximum capacity, the Xeon system is used to handle excess requests. This mode of operation is akin to considering the Xeon an accelerator for the Atom platform.

Summary. We have shown that (i) low power and high performance platforms exhibit different power performance based on the workload and clearly a single solution cannot satisfy the wide range of applications seen in today's datacenters; (ii) many components contribute to the overall power consumption and servers have a narrow dynamic range; (iii) the use of (simple) hybrid solutions may help in designing a datacenter architecture that gives low latency,

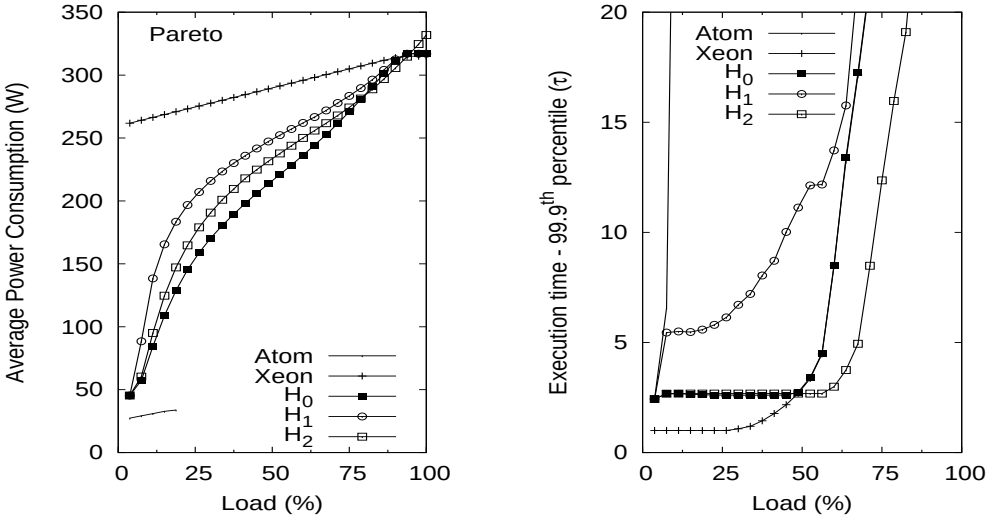

Figure 3: Average power consumption (left) and 99.9th percentile of execution time (right)

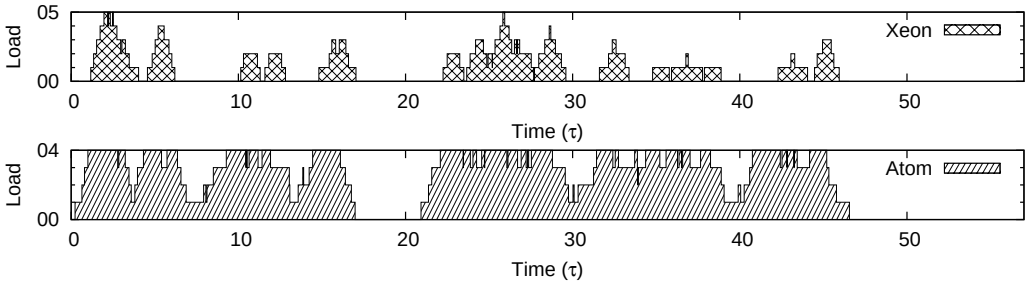

Figure 4: Example of hybrid H_2 operation. Top: Xeon load; Bottom: Atom load.

good performance/watt, and energy proportionality in a wide range of workloads.

3. DESIGN OPTIONS

In this section we set out to define the design questions to be addressed and the challenges involved in exploiting the full potential of hybrid datacenters in practice. Our intent is to understand what a research agenda around hybrid datacenters could look like.

Capacity Planning and Resource Scheduling. Given the initial capital investment in datacenters, workload characterization is crucial to decide to what degree a datacenter should be hybrid, e.g., what should be the mixture of systems with different performance-power tradeoffs? how fast should the high performance system be compared to the low-power one? A direction could be to cast this as an optimization problem: given the capital cost to buy machines, workloads, and SLAs, what is the mixture of servers that minimize the total energy consumption?

At a finer granularity, hybrid datacenters incur harder resource scheduling problems than traditional datacenters. Given an incoming workload, the resource scheduler should choose which servers (with different performance-power tradeoffs) to allocate. Furthermore, if the choice is not correct, the scheduler may migrate the workload from a server of certain type to another.

Hardware and Software Architecture. One possible dimension useful for classifying and evaluating hardware and software designs is the extent to which the high performance and low power platform share common components. Shared components have a direct impact on the complexity of the software architecture, on the degree of changes required in today's operating systems as well as on the overall cost, form factor and reliability of the hybrid platform. All these aspects are very important concerns for datacenter operators.

A first approach could consist of using complete and discrete systems. This solution is the simplest from a hardware perspective. A rack may contain different servers (Atom and Xeon-based) connected using standard network interfaces (e.g., Ethernet). It is also the most expensive as unshared components (e.g., disk, DRAM) may go under-utilized depending on the workload. From a software perspective, it would require to define a resource allocation scheme to choose which server to use for any incoming task. In addition, for datacenters that run cloud computing services, there would be a need to find ways to avoid using virtualization, as Atom servers may not need to multiplex through virtualization. But there would still be the need for some lightweight facility to migrate between Atom-based physical machine and traditional Xeon virtual machines. In this context, it is worth exploring operating system migration without virtualization [16].

Moving towards more integration, one could fit the low power platform on a single PCIe card and use it akin to the way graphics or crypto-acceleration boards are used today. There is a clear advan-

tage in terms of form factor and ease of deployment (no need to run external cables). On the software side, it would require new device drivers to make use of the new board (that would communicate with the main CPU via DMA) plus all the other virtualization-related challenges discussed above.

Finally integration at the processor level is also possible. For example, Intel's QuickAssist [2] allows to connect FPGA-based chips to the processor front-side bus (FSB). One could extend this approach to connect Atom chips directly to the Xeon FSB. This way the two processors would share all the rest of the platform memory, I/O hub, disks, etc. Going further along this path, one can imagine placing Xeon and Atom cores next to each other on the same die. Such heterogeneous core architectures have been proposed in several research projects as a way to improve power consumption [17, 22]. The main challenge with integration of Atom and Xeon cores or chips is that the remaining system components would have to be optimized for low power operations — not the case today where motherboard chipsets do not implement power management features. In addition, a radical change in the hardware architecture would require deep changes to the operating systems (e.g., Barrelfish [8]) to make them aware of the heterogeneous nature of the hardware and implement efficient context switching and data sharing between cores.

4. RELATED WORK

Building energy efficient datacenters is an active research area. FAWN [12] is an example of a cluster architecture that consists of a large number of slow, low-power embedded devices (AMD Geode) coupled with flash storage. The system is efficient in terms of queries per joule for particular seek-bound and I/O throughput-bound applications. Lim et. al [18] evaluated an alternative server architecture design built using embedded components and showed improvement in performance per dollar. Hamilton [15] made a similar argument using embedded or client-side components. To our knowledge, our approach is the first to argue for the case for hybrid approaches and to explore different hybrid datacenter design options.

5. CONCLUSION

In this paper, we have shown how hybrid datacenters have the potential to provide energy efficient operations without sacrificing the performance levels that today's datacenter provide. There exists a wide spectrum of possible solutions — some reachable in the short term (discrete solutions) others that require large investments (heterogeneous cores). They all come with a different set of trade offs and design challenges and further work is required to carefully evaluate each solution. However, we believe that hybrid datacenters represent a good opportunity for future green computing infrastructure.

6. REFERENCES

[1] Hadoop. `hadoop.apache.org`.

[2] Intel quickassist technology. `www.intel.com/technology/platforms/quickassist`.

[3] Nehalem-EX. `www.intel.com/pressroom/archive/releases/20090526comp.htm`.

[4] Specpower_ssj2008. `www.spec.org/power\_ssj2008/`.

[5] Microsoft Studies the Big Sleep. New York Times, `bits.blogs.nytimes.com/2009/02/24/`, Feb. 2009.

[6] Y. Agarwal et al. Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage. In *NSDI*, Apr. 2009.

[7] L. Barroso and U. Holzle. The case for energy-proportional computing. In *Computer*, 2007.

[8] A. Baumann et al. Your computer is already a distributed system. why isn't your os? In *HotOS*, 2009.

[9] S. Borkar. Thousand Core Chips - A Technology Perspective. In *DAC*, June 2007.

[10] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *OSDI*, 2004.

[11] B. Fitzpatrick. Distributed Caching with Memcached. *Linux Journal*, Aug. 2004.

[12] D. g. Andersen et al. FAWN: A fast array of wimpy nodes. In *SOSP*, 2009.

[13] V. George et al. Penryn: 45-nm Next Generation Intel Core 2 Processor. In *IEEE Asian Solid State Circuits Conference*, 2007.

[14] G. Gerosa et al. A Sub-2W Low Power IA Processor for Mobile Internet Devices in 45nm High-k Metal Gate CMOS. *IEEE Journal of Solid-State Circuits*, Jan. 2009.

[15] J. Hamilton. Cooperative expendable micro-slice servers: Low cost, low power servers for internet-scale services. 2009.

[16] M. A. Kozuch, M. Kaminsky, and M. P. Ryan. Migration without virtualization. In *HotOS*, 2009.

[17] R. Kumar et al. Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction. In *IEEE/ACM International Symposium on Microarchitecture*, 2003.

[18] K. Lim et al. Understanding and designing new server architectures for emerging ware house-computing environments. 2008.

[19] S. Nedevschi et al. Reducing network energy consumption via rate-adaptation and sleeping. In *NSDI*, Apr. 2008.

[20] S. Nedevschi et al. Skilled in the art of being idle. In *NSDI*, Apr. 2009.

[21] S. Rusu et al. A 65-nm Dual-Core Xeon Processor With 16-MB L3 Cache. *IEEE Journal of Solid-State Circuits*, Jan. 2007.

[22] H. Wong et al. Pangaea: A tightly-coupled ia32 heterogeneous chip multiprocessor. In *PACT*, 2008.