Optimizing the Datacenter for Data-Centric Workloads

Stijn Polfliet Frederick Ryckbosch Lieven Eeckhout
ELIS Department, Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
{stijn.polfliet, frederick.ryckbosch, lieven.eeckhout}@elis.UGent.be

ABSTRACT

The amount of data produced on the internet is growing rapidly. Along with data explosion comes the trend towards more and more diverse data, including rich media such as audio and video. Data explosion and diversity leads to the emergence of data-centric workloads to manipulate, manage and analyze the vast amounts of data. These data-centric workloads are likely to run in the background and include application domains such as data mining, indexing, compression, encryption, audio/video manipulation, data warehousing, etc.

Given that datacenters are very much cost sensitive, reducing the cost of a single component by a small fraction immediately translates into huge cost savings because of the large scale. Hence, when designing a datacenter, it is important to understand data-centric workloads and optimize the ensemble for these workloads so that the best possible performance per dollar is achieved.

This paper studies how the emerging class of data-centric workloads affects design decisions in the datacenter. Through the architectural simulation of minutes of run time on a validated full-system x86 simulator, we derive the insight that for some data-centric workloads, a high-end server optimizes performance per total cost of ownership (TCO), whereas for other workloads, a low-end server is the winner. This observation suggests heterogeneity in the datacenter, in which a job is run on the most cost-efficient server. Our experimental results report that a heterogeneous datacenter achieves an up to 88%, 24% and 17% improvement in cost-efficiency over a homogeneous high-end, commodity and low-end server datacenter, respectively.

Categories and Subject Descriptors

C.0 [**Computer Systems Organization**]: Modeling of computer architecture; C.4 [**Computer Systems Organization**]: Performance of Systems—*Modeling Techniques*

ICS'11, May 31–June 4, 2011, Tuscon, Arizona, USA.

General Terms

Design, Performance, Measurement, Experimentation

Keywords

Datacenter, data-centric workloads, workload characterization, heterogeneity

1. INTRODUCTION

The internet-sector server market is growing at a fast pace, by 40 to 65% per year according to various market trend studies (including by IDC). This fast increase is due to various novel internet services that are being offered, along with ubiquitous internet access possibilities through various devices including mobile devices such as smartphones and netbooks. In particular, smartphones enable their users to be permanently in touch with email, the internet, social networking sites such as Facebook and Twitter, e-commerce, etc. There are around 400 million smartphones worldwide today, and trend analysis estimates the number of smartphones to exceed 1.1 billion by 2013[1]. Hence, the number of people using internet services of various kinds is increasing rapidly and demonstrates the large scale of the applications and systems behind these services. For example, there are more than 500 million active Facebook users of which 50% log in on a daily basis; 200 million Facebook users use mobile devices and these users are twice as active as non-mobile users — according to Facebook's statistics as of Jan 2011[2]. As another example, there are 175M registered Twitter users generating more than 95M Twitter messages a day, as of Sept 2010[3].

Designing the servers to support these services is challenging, for a number of reasons. Online services have hundreds of millions of users, which requires distributed applications to run on tens to hundreds of thousands of servers [4], e.g., Facebook has 60,000 servers as of June 2010[4]. The ensemble of servers is often referred to as a warehouse-scale computer [5] and scaling out to this large a scale clearly is a key design challenge. Because of its scale, warehouse-scale computers are very much cost driven — optimizing the cost per server even by only a couple tens of dollars results in millions of dollars of cost savings and thus an increase in

[1] http://www.i4u.com/29160/11-billion-smartphones-2013
[2] http://www.facebook.com/press/info.php?statistics
[3] http://twitter.com/about
[4] http://www.datacenterknowledge.com/archives/2010/06/28/facebook-server-count-60000-or-more/

millions of dollars in revenue. There are various factors affecting the cost of a datacenter, such as the hardware infrastructure (the servers as well as the rack and switch infrastructure), power and cooling infrastructure as well as operating expenditure, and real estate. Hence, warehouse-sized computers are very cost-sensitive, need to be optimized for the ensemble, and operators drive their datacenter design decisions towards a sweet spot that optimizes performance per dollar. For example, commercial offerings by companies such as SeaMicro[5] as well as ongoing research and advanced development projects such as the EuroCloud project[6], target low-end servers to optimize datacenter cost-efficiency.

The emergence of warehouse-scale computers also leads to a dramatic shift in the workloads run on today's datacenters. Whereas traditional datacenter workloads include commercial workloads such as database management systems (DBMS) and enterprise resource planning (ERP), the datacenters in the cloud now run a new set of emerging workloads for online web services, e.g., e-commerce, webmail, video hosting, social networks. Users accessing these online web services generate huge amounts of data, both text and rich media (i.e., images, audio and video). The workloads running on a warehouse-scale computer not only include the interactive interface with the end user but also distributed data processing and storage infrastructure. In addition, data analytics workloads need to run in the datacenter 'behind the scenes' to manage, manipulate, and extract trends from the vast amounts of online data. For example, an e-commerce application will feature a data mining workload running in the background to collect user profiles and make suggestions to its end users for future purchases. Similarly, web search engines feature indexing workloads running in the background to build up indices. Whereas traditional datacenter workloads are well studied historically, see for example [9, 13, 19], and online interactive workloads have emerged as a workload of interest in recent research efforts [1, 12, 20], data-centric workloads have received limited attention so far.

1.1 Data-centric workloads

In this paper we focus on the data-centric workloads that are likely to run as background processes in datacenters in the cloud, i.e., workloads such as data mining, indexing, compression, encryption, rich media applications and data warehousing. And we study how these data-centric workloads affect some of the design decisions in the datacenter. Through full-system simulations using a validated x86 simulator while simulating minutes of run time, we explore which server type optimizes the performance per dollar target metric. We conclude that there is no clear winner: for some workloads, a high-end server yields the best performance per cost ratio, whereas for others, a middle-of-the-road server is a winner, and for yet other workloads, a low-end server yields the best performance-cost efficiency.

This result suggests the case for heterogeneous datacenters in which a workload is run on its most performance-cost efficient server. For our set of workloads and experimental setup (which assumes equal weight for all workloads), a homogeneous low-end server datacenter improves performance-cost efficiency by 14% compared to a homogeneous high-end server datacenter; we report an 18% better performance-cost efficiency for a heterogeneous datacenter relative to a homogeneous datacenter with high-end servers only. We also observe that a heterogeneous datacenter with a collection of high-end servers and low-end servers achieves most of the benefits that can be achieved through heterogeneity; adding middle-of-the-road servers does not contribute much.

Obviously, the improvement achieved through heterogeneity very much depends on the workloads that co-execute in the datacenter. Considering a wide range of workload mixes, we report performance-cost efficiency improvements for a heterogeneous datacenter up to 88%, 24% and 17% compared to homogeneous high-end, commodity and low-end server datacenters, respectively. Because estimating a datacenter's total cost of ownership is non-trivial, we also report results quantifying the performance-cost efficiency as a function of the cost ratio between the various server types, and by doing so, we determine the sweet spot for heterogeneous datacenters. Finally, we present a comprehensive analysis on where the benefit comes from. In the cases where the high-end server achieves a better performance-cost efficiency, the higher cost is offset by the higher throughput achieved through higher clock frequency, lower execution cycle counts and larger core counts. For the benchmarks for which the low-end processor is more performance-cost beneficial, the higher throughput achieved on the server is not offset by its higher cost.

We believe this is an interesting result given the current debate in the community on high-end versus commodity (middle-of-the-road) versus low-end servers for the datacenter [14, 17]. In particular, Lim et al. [12] conclude that lower-end consumer platforms and low-cost, low-power components from the high-volume embedded/mobile space may lead to a 2× improvement in performance per dollar. Reddi et al. [20] similarly conclude that a low-end Atom processor is more favorable than a high-end Intel Xeon for an industry-strength online web search engine, although these processor would benefit from better performance to achieve better quality-of-service and service-level agreements. In spite of these recent studies pointing towards low-end embedded servers for performance-cost efficient datacenters, there is no consensus as to whether contemporary datacenters should consider high-end versus low-end versus middle-of-the-road server nodes [14, 17]. Some argue for low-end 'wimpy' servers (see T. Mudge's statement in [14]) whereas others argue for high-end servers, and yet others argue for middle-of-the-road 'brawny' servers (see U. Hölzle's statement in [14]). This paper concludes there is no single answer. For some workloads, high-end servers are most performance-cost efficient, whereas for other workloads, low-end embedded processors are most efficient.

1.2 Paper contributions and outline

This paper makes the following contributions.

- We collect a set of data-centric workloads and we study how these workloads affect design decisions in the datacenter. Recent work in architectural studies for the datacenter considered online interactive workloads for the most part, and did not consider data-centric workloads. Running data-centric workloads requires minutes of run time on large data sets. We employ full-system simulation for doing so using a validated architectural simulator.

[5] http://www.seamicro.com/

[6] http://www.eurocloudserver.com/

- We obtain the result that high-end and middle-of-the-road servers can be more cost-efficient than low-end servers for running data-centric workloads. This is in contrast to recent work, see for example [1, 12, 20], which argues for lower-end servers to optimize cost-efficiency and/or energy-efficiency in the datacenter. The reason for this outcome is that data-centric workloads are computation-intensive and frequency-sensitive, hence, high-end and middle-of-the-road servers yield a substantially better performance per cost ratio.

- We demonstrate that for some sets of data-centric workloads, a heterogeneous datacenter in which each workload runs on its most cost-efficient server, can yield significant cost savings.

- We provide detailed sensitivity analyses to gain insight in the benefits of heterogeneity and how it varies with workload mixes, server infrastructure cost and energy cost. In particular, we demonstrate that heterogeneity is beneficial for a range of cost ratios between a high-end versus a low-end server. Further, we demonstrate that the benefit from heterogeneity is higher at lower energy costs.

The remainder of this paper is organized as follows. We first describe the data-centric workloads that we consider in this study (Section 2). We subsequently detail on the datacenter modeling aspects and our experimental setup (Section 3). We then describe our results (Section 4) and provide sensitivity analyses (Section 5). Finally, we discuss related work (Section 6) and conclude (Section 7).

2. DATA-CENTRIC WORKLOADS

2.1 Data explosion and diversity in the cloud

A prominent trend that we observe in the cloud is data explosion. The amount of online data has grown by a factor of 56× over 7 years, from 5 exabytes of online data in 2002 to 281 exabytes in 2009 — a substantially larger increase compared to Moore's law (16× over 7 years) [18]. The reason comes from the emergence of interactive internet services (e.g., e-commerce, web mail) and Web 2.0 applications such as social networking (e.g., Facebook, Twitter), blogs, wikis, etc., as well as ubiquitous access to online data through various mobile devices such as netbooks and smartphones.

Along with data explosion comes the trend of increasingly diverse data, including structured data, unstructured data and semi-structured data. In addition, the data stored in Web 2.0 applications is increasingly rich media, including images, audio and video.

Data explosion and diversity preludes a novel area of data-centric workloads in the cloud to manipulate the data, manage this huge data volume, extract useful information from it, derive insight from it, and eventually act on it. Hence, it is important to study these workloads and understand how this emerging class of workloads may change how datacenters are optimized for performance-cost efficiency.

2.2 A data-centric benchmark suite

Motivated by this observation, we collected a number of benchmarks to represent the emerging application domain of data-centric workloads. We identify a number of categories such as data mining, indexing, security, rich media, compression, and data warehousing. Each of these categories prelude important emerging applications in data-centric workloads. We select benchmarks for each of these categories, see also Table 1.

Data mining.

Analyzing the data is absolutely crucial to gain insight from it and eventually act on it. This requires data mining, statistical analysis and machine learning to extract and understand the underlying phenomena. We include three data mining benchmarks, namely kmeans, eclat and hmmer. The kmeans benchmark is a clustering workload that discovers groups of similar objects in a database to characterize the underlying data distribution. Clustering algorithms are often used in customer segmentation, pattern recognition, spatial data analysis, etc. Our dataset includes 100K data points in an 18-dimensional space and groups these points in 50 clusters. The eclat benchmark is a typical Association Rule Mining (ARM) workload to find interesting relationships in large data sets (466MB in our case). The benchmark tries to find all subsets of items that occur frequently in a database. The hmmer benchmark involves the pfam collection of multiple sequence alignments and hidden Markov models (HMM) covering many common protein domains and families. It is used for running the hmmpfam executable, part of the HMMER package. Its input is a sequence of 9,000 residues that is being compared against 2,000 HMMs.

Indexing.

Analyzing the data often requires indexing the data to enable efficient searching. We include the Apache lucene text search engine. In our case, lucene builds an index for 50K Wikipedia pages (647MB in total). The lucene benchmark is a Java workload and runs on the Open JDK JVM v6.

Data compression.

Storing huge volumes of data requires compression and decompression in order to be able to store the data on disk in an efficient way. Our benchmark suite includes the tarz application which consists of the standard GNU Tar utility to create an archive from, in our case, a set of PDF and text files. The archive is compressed using gzip (GNU zip). Gzip reduces the size of the archive using Lempel-Ziv (LZ77) encoding. The uncompressed input equals 1.2GB in size and is compressed to 273MB.

Data security.

Data stored in the cloud may be proprietary or personal, and third parties should not access this data. Data encryption is thus required to secure the data. We consider gpg (GNU Privacy Guard) as part of our benchmark suite. We sign and encrypt the same 1.2GB archive as for the compression benchmark.

Rich media applications.

As mentioned before, the data stored online is becoming more and more rich media, including audio (e.g., iTunes, MySpace), images (e.g., flickr), video (e.g., YouTube), as well as virtual reality (e.g., online games). We include three benchmarks to cover rich media applications, namely blender, bodytrack and x264. The

category	benchmark	source	description	run time
data compression	tarz	GNU	Create an archive and compress the files	1m10s
data mining	kmeans	MineBench	Mean-based data clustering	1m50s
	eclat	MineBench	Association rule mining to find interesting relationships in large data sets	1m56s
	hmmer	BioPerf	Compares sequence alignments against hidden Markov models	3m30s
data indexing	lucene	Apache	Apache text search indexer library written in Java	1m59s
data security	gpg	GNU	Sign and encrypt files	1m30s
rich media	blender	Blender Foundation	3D graphics rendering for creating 3D games, animated film or visual effects	2m15s
	bodytrack	PARSEC	Body tracking using multiple cameras	1m38s
	x264	PARSEC	Encoding video streams in H.264 format	1m15s
business	SPECjbb2005	SPEC	Middle-tier of server-side Java performance	2m09s

Table 1: Our set of data-centric benchmarks: their category, source, description and run time on a dual-socket dual-core AMD Opteron 2212 machine.

blender benchmark is a 3D graphics rendering application for creating 3D games, animated film and visual effects. We render 40 frames from a 3D scene including objects, and shadow, lightning and mirroring effects. The bodytrack benchmark is a computer vision application that tracks a human body with multiple cameras through an image sequence. As input data we consider 200 frames from 4 cameras with 4,000 particles in 5 annealing layers (input data set of 477MB). The x264 benchmark is an application for encoding videostreams in H.264 format. Its input is a 1.5GB video file.

Classical business logic.

Next to these emerging workloads, classical business logic will remain to be an important workload. We include PseudoSPECjbb2005, a modified version of SPECjbb2005 that executes a fixed amount of work rather than for a fixed amount of time. SPECjbb models the middle tier (the business logic) of a three-tier business system containing a number of warehouses that serve a number of districts. There are a set of operations that customers can initiate, such as placing orders or requesting the status of an existing order. PseudoSPECjbb, in our setup, processes 4M operations in total.

Both multi-threaded as single-threaded workloads.

As mentioned in Table 1, we gathered these benchmarks from various sources. Some benchmarks come from existing benchmark suites (PARSEC [7], MineBench [15], BioPerf [3]), while others were derived from real-life applications (Apache lucene, blender, GNU gpg, GNU tarz). Half the benchmarks are multi-threaded workloads (blender, bodytrack, kmeans, specjbb, x264); the others are single-threaded (hmmer, eclat, gpg, lucene, tarz). The inputs for these workloads were chosen such that the run time on a dual-processor dual-core AMD Opteron 2212 machine is on order of minutes, see also Table 1. We simulate these workloads to completion.

Workload data set sizes.

All the workloads run on data sets with hundreds of MBs or on the order of GBs of data. Although the data sets may be even bigger in real setups, we believe this is a reasonable assumption for our purpose, because these data sets do not fit in the processor's caches anyway. Hence, simulating even larger data sets is unlikely to change the overall conclusions. We simulate these workloads for minutes of real time, see also Table 1, or hundreds of billions of instructions, which is unusual for architecture simulation studies.

3. DATACENTER MODELING

Datacenter design is very much cost driven, and design decisions are driven by two key metrics, namely performance and cost [5]. Cost is not limited to hardware cost, but also includes power and cooling as well as datacenter infrastructure cost. A recently proposed metric for internet-sector environments is performance divided by total cost of ownership (TCO) and quantifies the performance achieved per dollar [12]. We now describe how we quantify cost and performance in the following two subsections, respectively.

3.1 TCO modeling

We build on the work by Lim et al. [12] to quantify datacenter cost. A three-year depreciation cycle is assumed and cost models are provided for hardware cost, as well as power and cooling costs. Hardware cost includes the individual components (CPU, memory, disk, board, power and cooling supplies, etc.) per server. Power and cooling cost includes the power consumption of the various server and rack components. The cooling cost includes infrastructure cost for power delivery, infrastructure cost for cooling, and the electricity cost for cooling.

We consider three server types: a high-end server, a low-end embedded processor and a middle-of-the-road (commodity) server. Table 2 describes their configurations and their cost models. The high-end server that we simulate is modeled after the Intel Xeon X5570; we assume an eight-core machine[7] running at 3GHz with a fairly aggressive out-of-order processor core along with an aggressive memory hierarchy. The low-end processor is a dual-core embedded processor running at 1.2GHz with a modest core and memory hierarchy, and is modeled after the Intel Atom Z515 processor. The commodity system is somewhat in the middle of the road between the high-end and low-end systems. We assume 4 cores at 2GHz and we model it after the Intel Core 2 Quad. The cost for each of the components is derived from a variety of sources[8].

[7]The Intel Xeon X5570 implements 4 cores and 2 hardware threads per core.

[8]http://ark.intel.com/Product.aspx?id=40740; http://www.newegg.com/Product/Product.aspx?Item=N82E16813131358; http://ark.intel.com/Product.aspx?id=40816&processor=Q8200S&spec-codes=SLG9T

Processor configuration			
	high-end	middle	low-end
frequency	3GHz	2GHz	1.2GHz
#cores	8	4	2
OOO core	4-wide	3-wide	2-wide
ROB size (#insns)	160	90	40
mem latency (cycles)	120	80	40
private L1 caches	64KB	32KB	32KB
L1 prefetching	yes	yes	no
private L2 caches	256KB	NA	NA
shared LLC cache	8MB	2MB	1MB
LLC prefetching	yes	no	no
branch predictor	4KB, 14b hist	2KB, 10b hist	1KB, 8b hist
Cost model			
	high-end	middle	low-end
CPU	1,386	213	45
board and mngmnt	330	145	50
memory	265	113	98
total hardware cost	1,981	471	193
CPU power (TDP)	95	65	1.4
server & rack power	300	100	22
cooling	300	100	22
total power (Watt)	600	200	44
power cost 3-year	2,680	894	197
total cost 3-year	4,662	1,365	390

Table 2: Processor configurations and their cost models (in Euro).

We use these default costs for reporting a reasonable design point given today's technology. Note that we do account for the server NIC cost as part of the 'board and management' cost. We do not account for the network itself; we basically assume that network cost is constant across different datacenter configurations. We believe this is a reasonable first-order approximation, given that networking accounts for 8% of the total datacenter cost only [8]. Further, because cost depends on many sources and varies over time, we vary the relative cost ratios across platforms in order to understand cost sensitivity in Section 5. In other words, if server cost and/or network cost were to differ across datacenter configurations, this can be accounted for through these cost ratios.

We consider a default energy cost of 17 Eurocent per kWh, unless mentioned otherwise. This is a typical private tariff rate; industry tariff rate may be as low as 10 Eurocent per kWh and below, hence, we explore a range of electricity costs in the evaluation section of this paper.

3.2 Performance modeling

We use HP Labs' COTSon simulation infrastructure [2] which uses AMD's SimNow [6] as its functional simulator to feed a trace of instructions into a timing model. COTSon can simulate full-system workloads, including the operating system, middleware (e.g., Java virtual machine) and the application stack. In this study, we use the COTSon-based simulator by Ryckbosch et al. [21], which has been validated against real hardware and which runs at a simulation speed of 37 MIPS with sampling enabled. This high simulation speed enables us to run the data-centric workloads on sufficiently large datasets for minutes of real time. The sampling strategy assumed is periodic sampling: we consider 100K instruction sampling units every 100M instructions and 1M instructions prior to each sampling unit for warming the caches and predictors.

We quantify performance as throughput or the number of jobs that can be completed per unit of time. Because the workloads that we consider are supposed to run as background processes in the cloud — these workloads are non-interactive with the end users — we believe throughput is the right performance metric. For each platform we compute the best possible throughput that can be achieved. For the single-threaded benchmarks this means we run multiple copies of the same benchmark concurrently on the multicore processor and we vary the number of copies (e.g., for the high-end server, from one copy up to eight copies), and we then report the best possible throughput that was achieved. For the multi-threaded benchmarks, we vary both the number of copies and the number of threads (e.g., on an 8-core system we consider 1 copy with 8 threads, 2 copies with 4 threads, etc.), and we report the best possible throughput.

4. OPTIMIZING THE DATACENTER

4.1 Which server type is optimal?

Figure 1 quantifies performance per TCO efficiency for the high-end, middle-of-the-road and low-end servers, normalized to the high-end server. Performance per TCO efficiency is defined as TCO divided by performance, or the reciprocal of performance per TCO. Hence, performance per TCO efficiency is a lower-is-better metric. The interesting observation from Figure 1 is that there is no single winner: there is no single server that yields the best performance per TCO across all the workloads. For most workloads, the low-end server results in the lowest performance per TCO efficiency, however, for a couple workloads, the high-end server is the most performance per TCO efficient system, see for example kmeans and x264. It is also interesting to note that for a couple workloads, namely gpg, hmmer and tarz, the middle-of-the-road server yields the best performance per TCO efficiency, albeit the difference with the low-end server is very small. The result that high-end and middle-of-the-road servers are more cost-efficient than low-end servers for some data-centric workloads is surprising and is in contrast to common wisdom and recently reported studies [1, 12, 20] which argued for lower-end servers to optimize cost-efficiency in the datacenter. The reason is that these workloads are computation-intensive which makes the high-end and commodity servers yield a better performance per cost ratio, as we explain next. It must be noted that these conclusions hold true for our workloads, but more study is needed before we can generalize these results to a much broader range of data-centric workloads, and internet-sector workloads in general.

4.2 Where does the benefit come from?

In order to get some insight as to why a particular server type is a winner for a particular workload, we break up the performance per TCO metric into its contributing components, using the following formula:

$$\textit{performance per TCO} = \frac{\textit{no. parallel jobs} \cdot \frac{\textit{freq}}{\#\textit{cycles}}}{TCO}. \quad (1)$$

The denominator quantifies cost for which we assume a 3-year de-

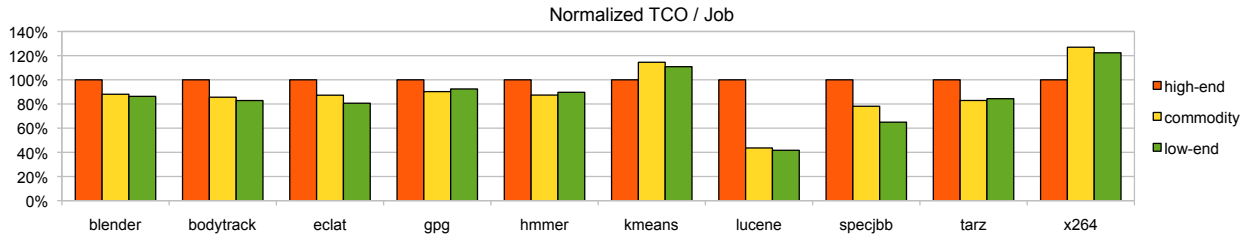

Figure 1: Normalized performance per TCO efficiency (lower is better) for the high-end, the middle-of-the-road and the low-end servers.

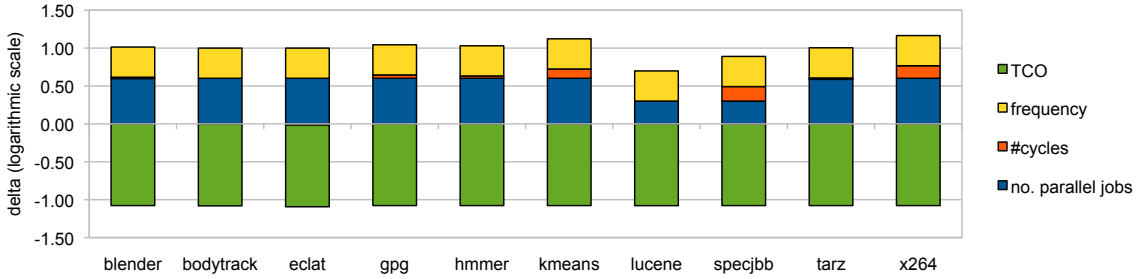

Figure 2: Performance per TCO stacks for quantifying the different factors; high-end versus low-end processors.

preciation cost cycle. The nominator quantifies throughput as the number of parallel jobs multiplied by the performance per job, or the reciprocal of the job's execution time; we measure throughput as the number of jobs that can be completed over a three-year time period. Figure 2 quantifies the contributing components when comparing the high-end versus the low-end server. The vertical axis is on a logarithmic scale. The contributing components are additive on a logarithmic scale, or multiplicative on a nominal scale. A negative component means that the component is a contributor in favor of the low-end server. In particular, TCO is always in favor of the low-end server because the TCO for the low-end server is about 12 times as low as for the high-end server. A positive component implies that the component is a contributor in favor of the high-end server. For example, frequency is a significant positive contributor for the high-end server: 3GHz versus 1.2 GHz, a 2.5× improvement. Also, the number of parallel jobs is a significant contributor for the high-end server for most workloads. This means that the high-end server benefits from its ability to run multiple jobs in parallel, and hence achieve a higher throughput than the low-end server. Note that for some benchmarks, e.g., lucene and specjbb, this component is only half as large as for the other benchmarks. This is due to the fact that 4 copies is the optimum for these benchmarks on the high-end servers versus 2 copies on the embedded server, whereas for the other benchmarks 8 copies is the optimum on the high-end server. Finally, the third positive contributor is the number of execution cycles; this means that the execution time in number of cycles is smaller on the high-end server compared to the low-end server. For most benchmarks, the number of execution cycles is roughly the same for the high-end and low-end servers, which implies that on the low-end server, the reduction in memory access time (in cycles) is compensated for by the increase in the number of cycles to do useful work (smaller processor width on the low-end server) and the increase in the number of branch mispredictions and cache misses (due to a smaller branch predictor and smaller caches on the low-end server). The number of execution cycles is a positive contributor for the high-end server for three benchmarks though, namely kmeans, specjbb and x264. In other words, the high-end server benefits significantly from the larger caches and branch predictor as well as the larger width compared to the low-end server for these workloads.

4.3 Does multi-threading help?

As mentioned before, half the workloads are multi-threaded and we optimize the datacenter for optimum throughput at the lowest possible cost. An interesting question is whether multi-threading helps if one aims for maximizing throughput. In other words, for a given workload for which there exists both a sequential and a parallel version, should we run multiple copies of the sequential version simultaneously, or are we better off running a single copy of the multi-threaded version? This is a non-trivial question for which an answer cannot be provided without detailed experimentation. On the one hand, parallel execution of sequential versions does not incur the overhead that is likely to be observed for the parallel version because of inter-thread communication and synchronization. On the other hand, multiple copies of sequential versions may incur conflict behavior in shared resources, e.g., the various sequential copies may incur conflict misses in the shared cache.

Table 3 summarizes the optimum workload configuration on each of the servers in terms of the number of instances of each workload and the number of threads per workload. For all of the multi-threaded workloads, except for specjbb, running multiple copies of the single-threaded workload version optimizes throughput. It is remarkable to see that multi-threading does not help in maximizing throughput for the data-centric workloads. Running multiple se-

	high-end	middle	low-end
blender	c8t1	c4t1	c2t1
bodytrack	c8t1	c4t1	
eclat	c8t1	c4t1	
gpg	c8t1	c4t1	
hmmer	c8t1	c4t1	
kmeans	c8t1	c4t1	
lucene	c4t1	c4t1	
specjbb	c4t2	c2t2	
tarz	c8t1	c4t1	
x264	c8t1	c4t1	

Table 3: Workload configurations that maximize throughput on the high-end, commodity and low-end servers; 'cxty' means 'x' copies of the same workload with 'y' threads.

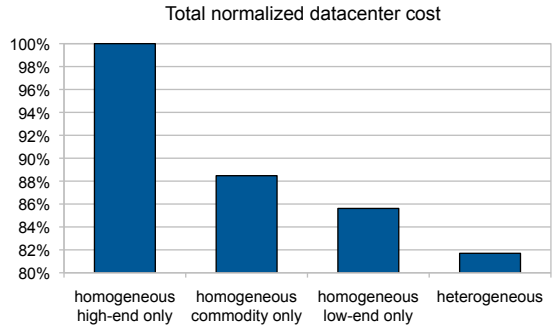

Figure 3: Normalized cost for iso-throughput homogeneous datacenters with high-end, middle-of-the-road and low-end servers only, versus a heterogeneous datacenter.

quential versions yields higher throughput compared to running a single parallel version; co-running sequential versions do not incur significant conflict behavior in shared resources.

4.4 The case for a heterogeneous datacenter

The results shown above suggest that a heterogeneous datacenter in which a job is executed on the most cost-efficient server, may be beneficial. In order to quantify the potential of a heterogeneous datacenter for data-centric workloads, we consider four iso-throughput datacenter configurations. We consider three homogeneous datacenters (with high-end servers only, middle-of-the-road servers only, and low-end servers only) as well as a heterogeneous datacenter. We assume the same workloads as before and we assume that all of these workloads are equally important — they all get the same weight. All of the datacenter configurations achieve the same throughput (for all of the workloads), hence, a datacenter with low-end servers needs to deploy more servers to achieve the same throughput as the homogeneous high-end server datacenter. The heterogeneous datacenter is configured such that it minimizes cost while achieving the same throughput as the homogeneous datacenters.

Figure 3 quantifies datacenter cost normalized to the homogeneous high-end server datacenter. A homogeneous datacenter with commodity servers reduces cost by almost 12% and low-end servers reduce datacenter cost by 14%. A heterogeneous datacenter reduces cost by 18%. Clearly, optimizing the datacenter's architecture has a significant impact on cost. Even homogeneous datacenters with commodity and low-end servers can reduce cost significantly. Heterogeneity reduces cost even further, although not by a large margin. However, this is very much tied to the workloads considered in this study. As shown in Figure 1, only two out of the ten workloads are run most efficiently on the high-end server. Hence, depending on the workloads, cost reduction may be larger or smaller.

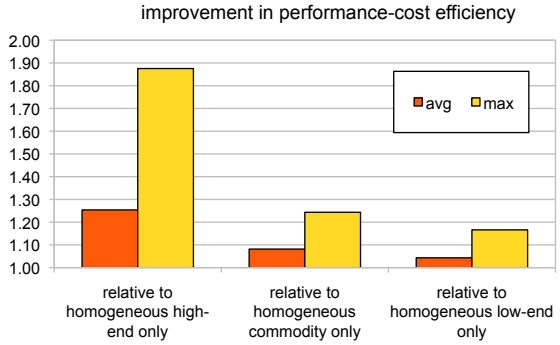

Figure 4: Cost reduction for a heterogeneous datacenter relative to homogeneous datacenter configurations across all possible two-benchmark workloads.

In order to get a better view on the potential of heterogeneity as a function of its workload, we now consider a large variety of different workload mixes. The previous experiment assumed that all the workloads are equally important, simply because we do not have a way for determining the relative importance of these workloads in real datacenters. We now consider a more diverse range of workload types: we consider all possible two-benchmark workload mixes and determine the potential benefit from heterogeneity; this is to study how sensitive a heterogeneous datacenter is with respect to its workload. In other words, for each possible two-benchmark workload mix, we determine the cost reduction through heterogeneity relative to homogeneous datacenters, see Figure 4. On average, a heterogeneous datacenter improves cost by 25%, 8% and 4%, and up to 88%, 24% and 17% relative to a homogeneous high-end, commodity and low-end server datacenter, respectively. (We consider the two-benchmark workload mixes for the remainder of the paper.)

We now zoom in on the architecture of a heterogeneous datacenter. We therefore consider the workload mixes for which we observe a throughput benefit of at least 30% for heterogeneity compared to a homogeneous datacenter consisting of high-end servers only. Figure 5 plots the fraction of low-end and commodity servers in a heterogeneous datacenter; one minus these two fractions is the fraction of high-end servers. The size of the disks relate to the number of cases (workload mixes) for which we observe a particular configuration. We observe that the optimum heterogeneous datacenter typically consists of a relatively large fraction low-end servers and smaller fractions of commodity and high-end servers.

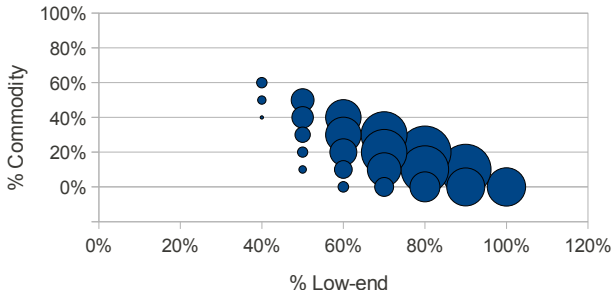

Figure 5: Configuration of the optimum heterogeneous datacenter: the fraction of low-end and commodity servers; the fraction of high-end servers equals one minus the fraction of low-end and commodity servers.

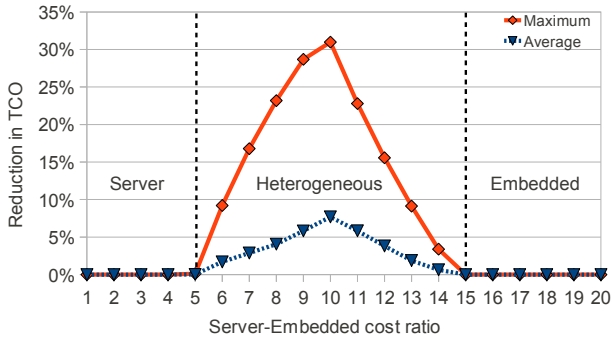

Figure 6: Cost reduction through heterogeneity as a function of the cost ratio between the high-end vs low-end servers.

5. SENSITIVITY ANALYSES

The results presented so far assumed the default parameters relating to datacenter cost mentioned in Section 3. Meaningful cost parameters are not easy to obtain because they are subject to a particular context, e.g., energy cost relates to where the datacenter is located, hardware purchase cost depends on the number of hardware items purchased, etc. In order to deal with the cost uncertainties, we therefore perform a sensitivity analysis with respect to the two main cost factors, hardware purchase cost and energy cost.

5.1 Varying the cost ratio

So far, we considered fixed costs for the various server types, as shown in Table 2. However, cost may vary depending on the number of servers that are bought — we assumed a fixed price per server. In addition, prices fluctuate over time. Hence, making a quantitative statement about which system is most performance-cost efficient at a given point in time, is subject to the cost ratios and thus it is not very informative. Instead, we also report the cost reduction through heterogeneity as a function of the cost ratio between the high-end and the low-end server, see Figure 6. The cost reduction reported here is the cost reduction over the best possible homogeneous datacenter. In case a high-end server is less than 5 times more expensive than a low-end server, then a high-end server is the clear winner, and there is no need for heterogeneity: a homogeneous high-end server datacenter optimizes the performance per dollar metric. In case a high-end server is more than 15 times more expensive that a low-end server, then a homogeneous datacenter with low-end servers is the optimal datacenter configuration. For cost ratios between 5× and 15×, performance per cost is optimized through heterogeneity. A 10× cost ratio yields the best possible benefit through heterogeneity, with an average reduction in cost of 8% and up to 31% for some workload mixes. As a point of reference, the cost ratio between the high-end and low-end server assumed in the rest of the paper equals 12, see Table 2.

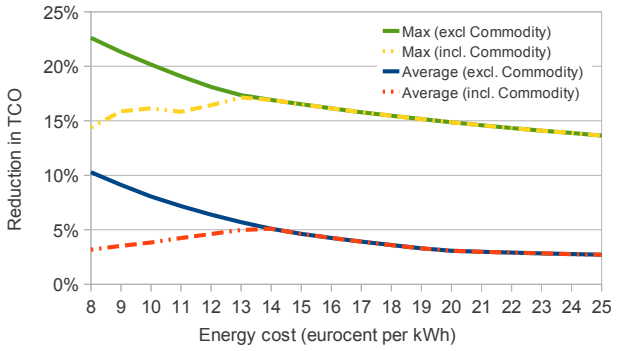

Figure 7: Cost reduction for a heterogeneous datacenter relative to the best possible homogeneous datacenter as a function of energy cost.

5.2 Varying energy cost

Along the same line, energy cost is variable as well: it varies from one location to another, and it varies over time. Figure 7 quantifies the cost reduction of a heterogeneous datacenter relative to the best possible homogeneous datacenter as a function of energy cost. We report the average and maximum cost reduction through heterogeneity, and we consider two heterogeneous datacenter setups: one setup includes high-end, commodity and low-end servers, while the other includes high-end and low-end servers only (no commodity servers). The reason for considering both configurations is that the performance-cost efficiency is comparable for the commodity and low-end servers, as seen in Figure 1, which implies that heterogeneous datacenters with high-end and low-end servers only would achieve most of the benefits from heterogeneity — including commodity servers does not add much benefit. This is indeed the case for the 17 Eurocent per kWh assumed so far.

The interesting observation from Figure 7 is that there is a cost benefit from heterogeneity across a broad range of energy prices. Second, when considering high-end and low-end servers only (i.e., 'excluding commodity' servers in Figure 7) for both the homogeneous and heterogeneous design points, the benefit from heterogeneity tends to be higher at lower energy costs. At lower energy costs, the performance argument outweighs the cost argument, shifting the optimum towards high-end servers for a larger fraction of the workloads. At higher energy costs, the performance per cost metric drives the optimum design point towards low-end servers for most of the workloads, hence, the benefit from heterogeneity is decreasing. Finally, commodity servers fit a sweet spot at lower energy costs (see the 'including commodity' curves in Figure 7). Commodity servers have interesting performance-cost properties at low electricity costs — they yield good throughput at relatively low cost. Nevertheless, heterogeneity is still beneficial and can reduce the datacenter's TCO by up to 15%.

5.3 Discussion

The results discussed so far made the case for heterogeneous datacenters. Significant cost reductions can be obtained compared to homogeneous datacenters while achieving the same overall system throughput. The results also revealed that the extent to which cost is reduced is subject to various factors including the workloads, server cost ratio for different server types, energy cost, etc. Hence, in some cases, depending on the constraints, the benefit from heterogeneity may be limited. However, in a number of cases (for specific sets of workloads, server cost ratios and energy cost), heterogeneity may yield substantial cost benefits, which may translate into millions of dollars of cost savings.

6. RELATED WORK

Prior work in architectural studies for warehouse-sized computers considered online interactive workloads for the most part. In particular, Lim et al. [12] consider four internet-sector benchmarks, namely websearch (search a very large dataset within sub-seconds), webmail (interactive sessions of reading, composing and sending emails), YouTube (media servers servicing requests for video files), and mapreduce (series of map and reduce functions performed on key/value pairs in a distributed file system). These benchmarks are network-intensive (webmail), I/O-bound (YouTube) or exhibit mixed CPU and I/O activity (websearch and mapreduce). The data-centric benchmarks considered in this paper are data-intensive and are primarily compute- as well as memory-intensive, and barely involve network and I/O activity. It is to be expected that cloud datacenters will feature both types of workloads, interactive internet-sector workloads as well as data-intensive background workloads. Lim et al. reach the conclusion that lower-end consumer platforms are more performance-cost efficient — leading to a 2× improvement relative to high-end servers. Low-end embedded servers have the potential to offer even more cost savings at the same performance, but the choice of embedded platform is important. We conclude that heterogeneity with both high-end and low-end servers can yield substantial cost savings.

Andersen et al. [1] propose the Fast Array of Wimpy Nodes (FAWN) datacenter architecture with low-power embedded servers coupled with flash memory for random read I/O-intensive workloads. Vasudevan et al. [22] evaluate under what workloads the FAWN architecture performs well while considering a broad set of microbenchmarks ranging from I/O-bound workloads to CPU- and memory-intensive benchmarks. They conclude that low-end nodes are more energy-efficient than high-end CPUs, except for problems that cannot be parallelized or whose working set cannot be split to fit in the cache or memory available to the smaller nodes — wimpy cores are too low-end for these workloads. Whereas the FAWN project focuses on energy-efficiency, we focus on cost-efficiency, i.e., performance per TCO. While focusing on data-centric workloads, we reach the conclusion that both high-end and low-end CPUs can be cost-efficient, depending on the workload.

Reddi et al. [20] evaluate the Microsoft Bing web search engine on Intel Xeon and Atom processors. They conclude that this web search engine is more computationally demanding than traditional enterprise workloads such as file servers, mail servers, web servers, etc. Hence, they conclude that embedded mobile-space processors are beneficial in terms of their power efficiency, however, these processors would benefit from better performance to achieve better service-level agreements and quality-of-service.

Keys et al. [10] consider a broad set of workloads as well as different processor types, ranging from embedded, mobile, desktop to server, and they aim for determining energy-efficient building blocks for the datacenter. They conclude that high-end mobile processors have the right mix of power and performance. We, in contrast, aim for identifying the most cost-efficient processor type taking into account total cost of ownership (TCO), not energy-efficiency only. We conclude that a mix of high-end servers and low-end servers optimizes performance per TCO.

Nathuji et al. [16] study job scheduling mechanisms for optimizing power efficiency in heterogeneous datacenters. The heterogeneous datacenters considered by Nathuji et al. stem from upgrade cycles, in contrast to the heterogeneity 'by design' in this paper. Also, Nathuji et al. consider high-end servers only and they do not include commodity and low-end servers as part of their design space.

Kumar et al. [11] propose heterogeneity to optimize power efficiency in multicore processors. Whereas Kumar et al. focus on a single chip and power efficiency, our work considers a datacenter, considers total cost (including hardware, power and cooling cost) and data-centric workloads.

7. CONCLUSION

Data explosion and diversity in the internet drives the emergence of a new set of data-centric workloads to manage, manipulate, mine, index, compress, encrypt, etc. huge amounts of data. In addition, the data is increasingly rich media, and includes images, audio and video, in addition to text. Given that the datacenters hosting the online data and running these data-centric workloads are very much cost driven, it is important to understand how this emerging class of applications affects some of the design decisions in the datacenter.

Through the architectural simulation of minutes of run time of a set of data-centric workloads on a validated full-system x86 simulator, we derived the insight that high-end servers are more performance-cost efficient compared to commodity and low-end embedded servers for some workloads; for others, the low-end server or the commodity server is more performance-cost efficient. This suggests heterogeneous datacenters as the optimum datacenter configuration. We conclude that the benefit from heterogeneity is very much workload and server-cost and electricity-cost dependent, and, for a specific setup, we report improvements up to 88%, 24% and 17% over a homogeneous high-end, commodity and low-end server datacenter, respectively. We also identify the sweet spot for heterogeneity as a function of high-end versus low-end server cost, and we provide the insight that the benefit from heterogeneity increases at lower energy costs.

Acknowledgements

We thank the anonymous reviewers for their constructive and insightful feedback. Stijn Polfliet is supported through a doctoral fellowship by the Agency for Innovation by Science and Technology (IWT). Frederick Ryckbosch is supported through a doctoral fellowship by the Research Foundation–Flanders (FWO). Additional support is provided by the FWO projects G.0232.06, G.0255.08, and G.0179.10, the UGent-BOF projects 01J14407 and 01Z04109,

and the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 259295.

8. REFERENCES

[1] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: A fast array of wimpy nodes. In *Proceedings of the International ACM Symposium on Operating Systems Principles (SOSP)*, pages 1–14, Oct. 2009.

[2] E. Argollo, A. Falcón, P. Faraboschi, M. Monchiero, and D. Ortega. COTSon: Infrastructure for full system simulation. *SIGOPS Operating System Review*, 43(1):52–61, Jan. 2009.

[3] D. A. Bader, Y. Li, T. Li, and V. Sachdeva. BioPerf: A benchmark suite to evaluate high-performance computer architecture on bioinformatics applications. In *Proceedings of the 2005 IEEE International Symposium on Workload Characterization (IISWC)*, pages 163–173, Oct. 2005.

[4] L. A. Barroso, J. Dean, and U. Hölzle. Web search for a planet: The google cluster architecture. *IEEE Micro*, 23(2):22–28, Mar. 2003.

[5] L. A. Barroso and U. Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Synthesis Lectures on Computer Architecture. Morgan and Claypool Publishers, 2009.

[6] R. Bedichek. SimNow: Fast platform simulation purely in software. In *Proceedings of the Symposium on High Performance Chips (HOT CHIPS)*, Aug. 2004.

[7] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC benchmark suite: Characterization and architectural implications. In *Proceedings of the International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 72–81, Oct. 2008.

[8] J. Hamilton. Datacenter networks are in my way. Principals of Amazon, Oct. 2010.

[9] K. Keeton, D. A. Patterson, Y. Q. He, R. C. Raphael, and W. E. Baker. Performance characterization of a quad Pentium Pro SMP using OLTP workloads. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 15–26, June 1998.

[10] L. Keys, S. Rivoire, and J. D. Davis. The search for energy-efficient building blocks for the data center. In *The Second Workshop on Energy-Efficient Design (WEED), held in conjunction with the International Symposium on Computer Architecture (ISCA)*, June 2010.

[11] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen. Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction. In *Proceedings of the ACM/IEEE Annual International Symposium on Microarchitecture (MICRO)*, pages 81–92, Dec. 2003.

[12] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 315–326, June 2008.

[13] Y. Luo, J. Rubio, L. K. John, P. Seshadri, and A. Mericas. Benchmarking internet servers on superscalar machines. *IEEE Computer*, 36(2):34–40, Feb. 2003.

[14] T. Mudge and U. Hölzle. Challenges and opportunities for extremely energy-efficient processors. *IEEE Micro*, 30(4):20–24, July 2010.

[15] R. Narayanan, B. Ozisikyilmaz, J. Zambreno, G. Memik, A. Choudhary, and J. Pisharath. MineBench: A benchmark suite for data mining workloads. In *Proceedings of the International Symposium on Computer Architecture (IISWC)*, pages 182–188, Oct. 2006.

[16] R. Nathuji, C. Isci, and E. Gorbatov. Exploiting platform heterogeneity for power efficient data centers. In *Proceedings of the International Conference on Autonomic Computing (ICAC)*, Oct. 2007.

[17] K. Olukotun, J. Laudon, and B. Lee. Mega-servers versus micro-blades for datacenter workloads. Panel debate at the Workshop on Architectural Concerns in Large Datacenters (ACLD), held with ISCA, June 2010.

[18] P. Ranganathan. Green clouds and black swans in the exascale era. Keynote at the IEEE International Symposium on Workload Characterization (IISWC), Oct. 2009.

[19] P. Ranganathan, K. Gharachorloo, S. V. Adve, and L. A. Barroso. Performance of database workloads on shared-memory systems with out-of-order processors. In *Proceedings of the Eighth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Oct. 1998.

[20] V. J. Reddi, B. C. Lee, T. Chilimbi, and K. Vaid. Web search using mobile cores: Quantifying and mitigating the price of efficiency. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 26–36, June 2010.

[21] F. Ryckbosch, S. Polfliet, and L. Eeckhout. Fast, accurate and validated full-system software simulation of x86 hardware. *IEEE Micro*, 30(6):46–56, Nov/Dec 2010.

[22] V. Vasudevan, D. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru. Energy-efficient cluster computing with FAWN: Workloads and implications. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy)*, pages 195–204, Apr. 2010.