



System-Level Integrated Server Architectures for Scale-out Datacenters

Sheng Li, Kevin Lim, Paolo Faraboschi, Jichuan Chang, Parthasarathy Ranganathan, Norman P. Jouppi

HP Laboratories
HPL-2012-189

Keyword(s):

System-on-Chip; server; datacenter; cost; TCO

Abstract:

A System-on-Chip (SoC) integrates multiple discrete components into a single chip, for example by placing CPU cores, network interfaces and I/O controllers on the same die. While SoCs have dominated high-end embedded products for over a decade, system-level integration is a relatively new trend in servers, and is driven by the opportunity to lower cost (by reducing the number of discrete parts) and power (by reducing the pin crossings from the cores to the I/O). Today, the mounting cost pressures in scale-out datacenters demand technologies that can decrease the Total Cost of Ownership (TCO). At the same time, the diminishing return of dedicating the increasing number of available transistors to more cores and caches is creating a stronger case for SoC-based servers. This paper examines system-level integration design options for the scale-out server market, specifically targeting datacenter-scale throughput computing workloads. We develop tools to model the area and power of a variety of discrete and integrated server configurations. We evaluate the benefits, trade-offs, and trends of system-level integration for warehouse-scale datacenter servers, and identify the key "uncore" components that reduce cost and power. We perform a comprehensive design space exploration at both SoC and datacenter level, identify the sweet spots, and highlight important scaling trends of performance, power, area, and cost from 45nm to 16nm. Our results show that system integration yields substantial benefits, enables novel aggregated configurations with a much higher compute density, and significantly reduces total chip area and dynamic power versus a discrete-component server. Finally, we use utilization traces and architectural profiles of real machines to evaluate the dynamic power consumption of typical scale-out cloud applications, and combine them in an overall TCO model. Our results show that, for example at 16nm, SoC-based servers can achieve more than a 26% TCO reduction at datacenter scale.

External Posting Date: September 6, 2012 [Fulltext] Approved for External Publication

Internal Posting Date: September 6, 2012 [Fulltext]

Published in MICRO-44: 44th Annual IEEE/ACM International Symposium on Microarchitecture (2011)

© Copyright MICRO-44: 44th Annual IEEE/ACM International Symposium on Microarchitecture (2011)

System-Level Integrated Server Architectures for Scale-out Datacenters

Sheng Li, Kevin Lim, Paolo Faraboschi,
Jichuan Chang, Parthasarathy Ranganathan, Norman P. Jouppi
Hewlett-Packard Labs
{sheng.li4, kevin.lim2, paolo.faraboschi,
jichuan.chang, partha.ranganathan, norm.jouppi}@hp.com

ABSTRACT

A System-on-Chip (SoC) integrates multiple discrete components into a single chip, for example by placing CPU cores, network interfaces and I/O controllers on the same die. While SoCs have dominated high-end embedded products for over a decade, system-level integration is a relatively new trend in servers, and is driven by the opportunity to lower cost (by reducing the number of discrete parts) and power (by reducing the pin crossings from the cores to the I/O). Today, the mounting cost pressures in scale-out datacenters demand technologies that can decrease the Total Cost of Ownership (TCO). At the same time, the diminishing return of dedicating the increasing number of available transistors to more cores and caches is creating a stronger case for SoC-based servers.

This paper examines system-level integration design options for the scale-out server market, specifically targeting datacenter-scale throughput computing workloads. We develop tools to model the area and power of a variety of discrete and integrated server configurations. We evaluate the benefits, trade-offs, and trends of system-level integration for warehouse-scale datacenter servers, and identify the key “uncore” components that reduce cost and power. We perform a comprehensive design space exploration at both SoC and datacenter level, identify the sweet spots, and highlight important scaling trends of performance, power, area, and cost from 45nm to 16nm. Our results show that system integration yields substantial benefits, enables novel aggregated configurations with a much higher compute density, and significantly reduces total chip area and dynamic power versus a discrete-component server.

Finally, we use utilization traces and architectural profiles of real machines to evaluate the dynamic power consumption of typical scale-out cloud applications, and combine them in an overall TCO model. Our results show that, for example at 16nm, SoC-based servers can achieve more than a 26% TCO reduction at datacenter scale.

Categories and Subject Descriptors

C.0 [Computer Systems Organizations]: GENERAL

General Terms

Design, Performance, Verification

Keywords

System-on-Chip, server, datacenter, cost, TCO

1. INTRODUCTION

In the last decade, System-on-Chip (SoC) designs have become the dominant technology in high-end embedded, consumer, and telecommunication markets. A typical SoC includes a combination of several components: heterogeneous processor cores, memory, network and I/O controllers, VLIW cores, DSPs, and graphics processors, and special-purpose accelerators. The significant savings in power, total die area, and cost are paramount to the embedded market and have been a key force driving the adoption of system-level integration. Moreover, the opportunities to customize, differentiate and optimize for a given target domain provide an additional set of benefits for embedded SoCs targeting a specific application domain.

To date, SoCs have remained relatively absent from the general purpose processor mainstream, where power efficiency and cost are traditionally sacrificed for higher performance. However, if we look at historical CPU trends, we can observe a slow but steady pace of integration of system-level features [37], as demonstrated by the appearance of on-chip L2/L3 caches, memory controllers and GPUs. Other system components (such as bus interfaces and interconnect support) have started appearing in consumer-class [11, 26, 47], and server [46] CPUs, and we expect this trend to continue in the future.

In our work, we focus on the use of system-level integration for the warehouse-scale datacenter server market. As datacenters grow larger and cloud computing scale-out workloads become more important, there is a relentless pressure to increase cost efficiency at all levels [20], with the goal of reducing the datacenter Total Cost of Ownership (TCO). TCO consists of several components, such as amortization of facilities capital costs, management, software licensing, and real-estate space, but two primary contributors are hardware acquisition and energy costs. Because SoC-based systems can reduce the bill of material (number of components) of a server and a large fraction of power-hungry pin crossings,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MICRO’11, December 3-7, 2011, Porto Alegre, Brazil

Copyright (c) 2011 ACM 978-1-4503-1053-6/11/12 ...\$10.00.

they have the potential to address two major TCO contributors.

While the trend towards SoC is clear, a quantification of the benefits that integration provides remains an open question. The many design options for SoCs, such as the number of cores and their organization, the core complexity, or the choice of the integrated controllers, lead to a very large design space to be explored.

In this paper, we explore the implications of SoC designs on the scale-out server market and make the following contributions:

- We present a comprehensive framework for modeling and evaluating the impact of SoC integration designs ranging from the technology level, through system and cluster/rack level. To our knowledge, this is the first modeling framework that addresses the effect of system-level integration across these levels, and it lays the groundwork for the community to embark in future explorations of SoC designs and their implications at the scale-out level.
- We explore the design space and quantify benefits and technology-level effects of SoC integration for datacenter servers. We identify scaling trends of power, area, and cost of “core” and “uncore” components from 45nm to 16nm that reveal integration becomes more important as technology advances. We show that integration choices in servers require careful thinking, and highlight the important sweet spots.
- We extend our analysis to the impact of SoC integration on the TCO of a large-scale datacenter. We show that SoC-based individual servers provide tangible TCO savings of more than 26% in future large scale datacenters.

2. AN OVERVIEW OF SOC DESIGN

To understand the implications of System-on-Chip based design, we begin by looking at a traditional server design and introduce the concepts of “core” versus “uncore” components and define a taxonomy of the SoC design space while identifying the key design parameters.

Traditionally, a server is built out of several discrete components, including a processor with caches, a controller of the *fast* subsystem (sometimes called “northbridge”) usually connecting the DRAM and the graphics controller, and a controller of the *slow* subsystem (sometimes called “southbridge”) dealing with I/O devices and Flash/ROM. As seen in Figure 1(a), each of these functions is implemented in discrete chips which are traditionally kept physically separate to allow flexibility in mixing and matching different CPUs and devices. Over time, some of the system functionality has been moved into the CPU die: for example, memory controllers are today commonly integrated with the processor [7, 25, 31].

Several reasons are pushing previously discrete functionality onto the main processor die. First, a fully discrete design leads to major energy inefficiencies: driving pins to address (possibly distant) off-chip devices requires large amounts of power. Additionally, the different scaling and evolution speed of cores and I/O components has changed the balance of cost and power in the system, requiring new thinking to

address these issues. Whereas the processor core was by far the most expensive and power-hungry component, now other system elements, such as DRAM or graphics processing units (GPUs), can often rival (or exceed) the processor in cost and power [43, 47]. Finally, as we start seeing diminishing returns in adding more cores and core complexity, more transistors become available for additional functionality and motivate rethinking some of the traditional “core” and “uncore” boundaries. In general, as these cost and energy balances evolve, the “uncore” components of the system pose additional challenges to improving overall system efficiency beyond the “core” microprocessor.

System-level integration addresses the inefficiency of uncore components by placing CPUs and components on the same die. This reduces the *latency* by placing cores and components closer to one another, *cost* by reducing parts in the bill of material, and *power* by decreasing the number of chip-to-chip pin-crossings. Figure 1(b) shows a simplified block diagram of a single SoC that integrates all of the components shown previously in Figure 1(a).

SoCs offer a very broad design space of components that can be integrated onto a single chip, even considering the limited “general purpose” subset that is appropriate for the server market: processor cores, memory controllers, network controllers, storage controllers, PCI express (PCIe) controllers, GPUs, and some special-purpose accelerators (such as security or compression), and more. For the purposes of our study, we classify SoCs by the type of components in the system that are integrated. We define this space along six parameters: (1) core type and count; (2) cache configuration; (3) on-chip interconnect; (4) “near” controllers; (5) “far” controllers; and (6) accelerators.

2.1 Related work

While the traditional monolithic processor with discrete chipsets largely dominate the server processor market, some examples of SoC-based designs have been proposed. In industry, the Sun Niagara 2/3 and the Tilera Tile64 are among the few general purpose processor SoC designs. For example Niagara integrates a PCIe controller, two 10Gb Ethernet ports, and many heavily multi-threaded simple cores; it is targeted towards high-end enterprise markets, as evidenced by its very large ($>340\text{mm}^2$) die size [25]. On the CPU/GPU front, AMD and Intel have both recently announced processors aimed at the desktop and mobile markets which integrate GPUs and other system-level functionality on the same die [11, 26, 47]. ARM processor cores, which are the dominant core in embedded SoC designs, are starting to appear in server-targeted SoC products. For example, the Marvell Armada XP [36] is an example of an ARM-based quad-core SoC that also integrates network interfaces and storage controllers. Recent announcements by Calxeda [1] also show a focus on a (yet undisclosed) ARM-based highly integrated server-based design. Finally, the SeaMicro SM10000 [2], while not based on a processor SoC, is a pioneer in the use of an “army of ants” of low-power processors coupled with a proprietary interconnect and management functionality, using similar principles as found in SoC design.

In research, most prior work has primarily focused on low-power processor and system architectures. Work such as Microblades [35] and FAWN [8] focused on using low-power processors as the basis of platforms for scale-out in-

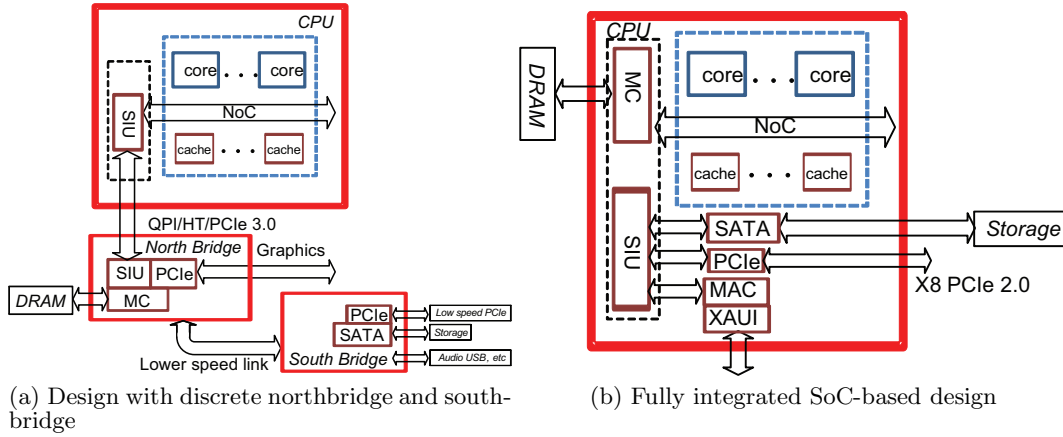


Figure 1: A simplified block-diagram view of traditional and SoC-based server designs.

frastructures. Similarly, Gordon [15] used low-power processors along with optimized Flash interfaces to provide an energy-efficient server building block. Other work such as Picoservers [30] used low-power cores paired with 3D stacked memory, and the on-going EuroCloud [3] project is looking at utilizing SoCs with 3D stacked memory to create a “server-on-chip”.

Several challenges remain for widespread adoption of system-level integration in the general purpose server market. In comparison to an embedded SoC, server CPUs tend to be much more complex, and require higher volumes and longer lifetimes to justify the larger engineering investment to build them. Because an SoC is by definition more specialized, its narrow applicability works against the economics of large high-performance chips. In turn, targeting general-purpose high performance pushes towards aggressive out-of-order microarchitectures and large caches, which negatively impacts power dissipation, and can cause thermal issues that may prevent integrating additional functionality on the same die.

3. THE SOC ARCHITECTURE SPACE

Several open questions remain on the specific design of SoCs for servers, and pruning the large design space is necessary to identify the most promising architectures. First and foremost, a server SoC must provide an adequate level of performance for the target workloads. In our work, we focus on cloud-like throughput workloads, where a scale-out (distributed) design is usually more efficient than scale-up (shared memory) designs. For these shared-nothing workloads, the overall throughput scales almost linearly with the number of nodes; however, the individual nodes are still required to maintain a reasonable single-threaded performance [38], as well as the appropriate balance of compute and I/O resources.

In contrast to the embedded market, where systems are designed for a limited purpose, an important aspect of servers is the general-purpose nature of the application target, and this has important implications on the range of applicable SoC designs. For example, high-end embedded SoCs normally include special accelerators and/or VLIW or DSP engines (such as TI’s OMAP5 [14]). Instead SoCs for servers must rely on components offering general-purpose functionality, and focus on reducing the overall system cost and power. While the use of accelerators in servers is an in-

teresting and important research area, its use is still fairly limited, and we consider it outside of the scope of this work.

To be viable, an SoC-based server must provide tangible TCO benefits over existing discrete designs. Deploying an SoC-based server architecture can have deep implications on both the hardware and software ecosystems, and may reduce the overall system flexibility. For example, integrating the Network Interface Controller (NIC) saves cost and energy, but is not compatible with configurations that share NICs across multiple nodes, or wish to use a more powerful NIC, or support a different protocol (e.g., Infiniband over Ethernet). While these challenges are difficult to quantify, it is important to be aware of them in the context of SoC-based servers. We believe that our results show that SoC-based servers can achieve significant benefits that overcome potential system flexibility drawbacks, but at the same time we want to highlight the importance of paying special attention to these considerations when addressing the architecture of server SoCs.

3.1 Base Architecture

We chose to base our analysis on a next-generation, high-end, low-power multi-core design. The microarchitecture we model has characteristics similar to the upcoming ARM Cortex A15 [32] core, whose parameters are shown in Table 1(a).

Our choice was based on several considerations. The trend towards simpler cores due to well-known power limitations will likely continue into the foreseeable future. Similarly, the thermal issues of SoC designs also push the market towards low-power cores, thus making it a natural choice for our evaluation. Finally, previous studies [8, 35] have shown a design style based on simpler cores to be well suited for throughput-oriented server workloads.

As we further discuss in Section 4, we look at multiple die sizes and technology generations using this style of microarchitecture building block¹. We start with 2 cores at a 45 nm technology using a 50 mm² die size, and scale up the number of cores as the die size increases and as technology generations advance.

The basic multicore substrate assumes a tiled architec-

¹For practical reasons, we actually use an x86 ISA in the evaluation, and also model the x86 decoder.

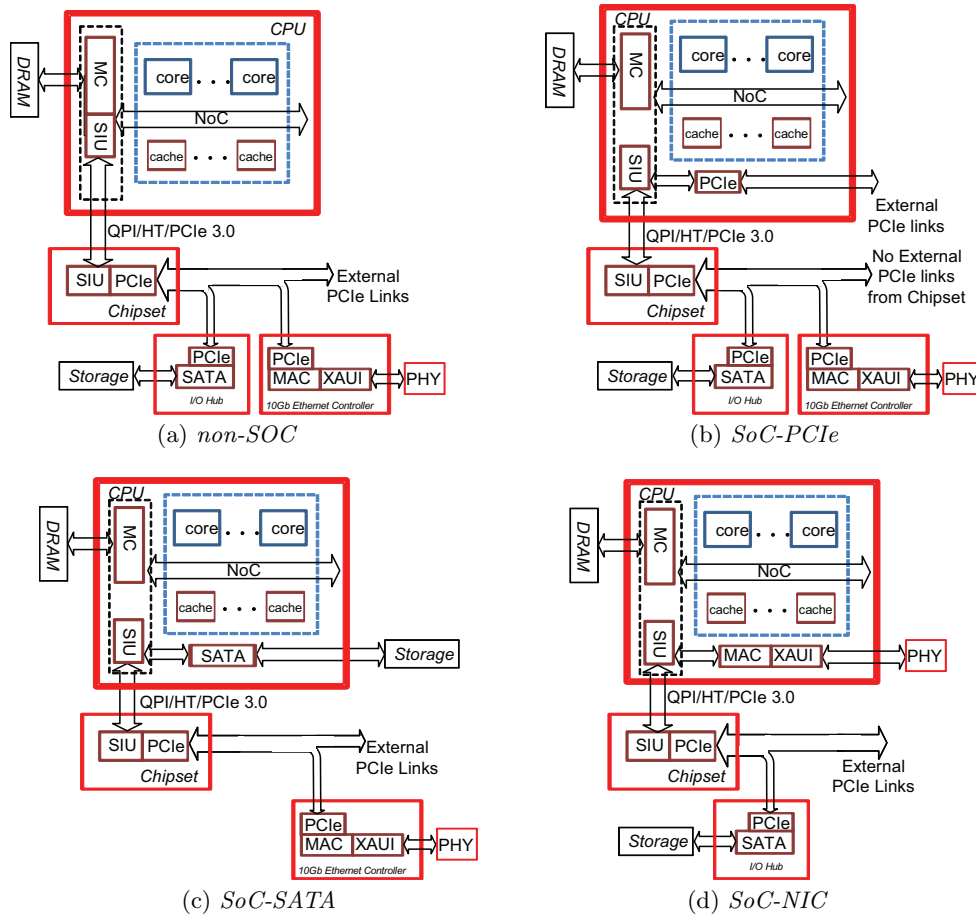


Figure 2: Baseline architecture plus four different integrated SoC options. (a) Discrete chipsets with no SoC integration, (b) integrated PCIe controller, (c) integrated disk I/O controller, (d) integrated NIC.

Table 1: Target SoC architecture parameters.

(a) Baseline processor

Processor	
Frequency	2 GHz @45nm
Issue width	3
Architecture	Out-of-Order
L2 cache size	1 MB per Core

(b) Taxonomy for our design space

Taxonomy	Options	Integrated
Onchip interconnect	Network-on-chip/bus	-
“Near” controllers	Memory controller, PCIe	Yes, {Yes/no}
“Far” controllers	Disk controller, NIC	{Yes/no}, {Yes/no}
Accelerators	None	-

ture with a flat memory subsystem, with one core per tile and a private L2 cache (to match the throughput computing workload characteristics). With 8 cores or fewer, all cores and their private caches are connected by a network-on-chip (NoC) that is fully interconnected; with more than 8 cores, they are connected by a 2D-mesh. The fact that our multi-core template does not share any important resource (such as last-level caches) allows us to linearly scale the throughput based on the analysis of a single core.

Figure 2(a) shows a baseline *non-SoC* (discrete components) server using this architecture. We assume a typical cost-optimized server design similar to those found in warehouse-scale data centers [20] with PCIe channels, a network controller, and a locally attached SATA/SAS storage subsystem. Since we are looking at a cost-optimized server design, we assume a common SATA I/O hub, a 10Gb Ether-

net controller with an integrated XAUI-PHY² interface, and no peripheral ports (such as audio or USB). Both the I/O hub and Ethernet controller utilize PCIe to communicate with the CPU and/or its chipset. The XAUI-PHY interface can connect to different types of Ethernet PHY chips (e.g., copper, UTP, fiber). As we discussed before, since we do not model accelerators, we also do not consider high-speed x16 PCIe lanes whose purpose in servers is often limited to support GP-GPUs accelerators. Table 1(b) shows the design space we evaluate.

3.2 SoC Integration Configurations

In Figure 2, we show the three SoC partial integration configurations we consider in our study. We first look at inte-

²XAUI is a chip-to-chip interface standard between MAC and the physical layer of 10Gb Ethernet.

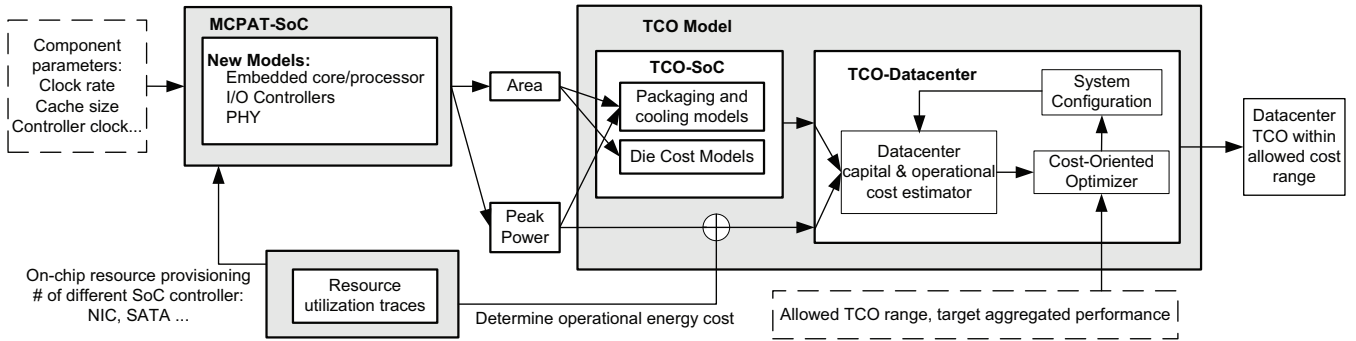


Figure 3: Evaluation framework

grating each of the controllers individually, shown in Figure 2(b)-(d). Those designs –*SoC-PCIe*, *SoC-SATA*, and *SoC-NIC*– integrate just the PCIe controller, disk I/O controller, and NIC, respectively: *SoC-PCIe* reduces the complexity of the chipset by removing the PCIe links; *SoC-SATA* uses a chipset with narrower PCIe links and does not need an I/O hub; *SoC-NIC* reduces chipset complexity and removes the separate Ethernet controller. We also consider an architecture integrating all three controllers, *SoC-all*, shown in Figure 1(b) that does not require external high-speed links (e.g., QuickPath Interconnect, HyperTransport, or the forthcoming PCIe 3.0) since all relevant high-speed communication remains on-chip.

When components are integrated, communication between the processor and the components no longer need to go off-chip. This change eliminates the need to serialize and deserialize the signals through high-speed Serializer/Deserializer (SerDes) and the power to drive the pins to communicate off-chip, by using more parallel and energy-efficient on-chip communication instead. By moving these components on chip, the motherboard and system designs also become simpler, cheaper, and smaller, and enable a much higher physical compute density.

Using these SoC designs, we examine the effect of their differing degrees of integration on total die area, power, and cost in Section 5, comparing their impacts across several technology generations.

4. MODELING SOC-BASED DATACENTER SERVERS

There are many challenges to evaluating the impact of SoC servers at the datacenter level, which are made even more complex as technology generations advance. At the more basic level, there are fundamental challenges that must be addressed to enable a successful study of future SoC-based servers. An understanding of SoC implications is required at multiple levels: at the chip level for area, power, and cost of SoCs compared to CPUs with discrete chipsets; at the system level for changes to system architecture and cost due to SoC; and at the cluster/datacenter level for the impact of SoC servers on the Total Cost of Ownership (TCO).

In order to address these challenges, we have developed a new, comprehensive methodology for modeling and evaluating SoC-based servers using multiple tools and models, as shown in Figure 3. To the best of our knowledge this is the first framework supporting a comprehensive cost analysis

from chip to datacenter scale. Its goal is to provide in-depth modeling insights of SoCs, ranging from the technology level (die area, power), to the cluster level (performance, total cost of ownership). As we aggressively enter the system integration era, tools that can model and quantify different SoC configurations are going to become as important to the architecture community as tools like Cacti have been to understand the cache memory hierarchy tradeoffs. From this perspective, we believe that our set of tools and proposed evaluation framework represent a core contribution of this work.

4.1 Framework Overview

Our framework is composed of (1) *McPAT-SoC*—a detailed modeling framework for power, area and timing of SoC chips at the technology and architecture level; (2) resource utilization traces, obtained from both real systems running our throughput workloads and simulations; and (3) a thorough model of the total cost of ownership (TCO) of SoC-based designs in a datacenter, along with an SoC die and packaging cost estimator.

The workflow of our framework, shown in Figure 3, is as follows: *McPAT-SoC* (which extends the original McPAT [33]) is used to model the power, area, and timing of a parametric SoC design. Resource utilization traces obtained from real servers are used with *McPAT-SoC* to determine on-chip resource provisioning. Given a design target and the resource provisioning, *McPAT-SoC* computes the peak power and area, which are then fed to the *SoC cost model* to estimate the die and packaging cost. The peak power numbers are then combined with *workload profiles* to compute the dynamic power usage of the target workloads, and together with the cost estimates are finally fed to the datacenter *TCO model* to obtain the aggregated TCO of a target scale-out cluster using SoC-based servers.

4.2 Chip-level Models: McPAT-SoC

McPAT is an integrated power, area, and timing analytical modeling framework for multi-/many-core processors. When combined with a performance simulator, McPAT can also produce runtime power figures based on the statistics (instruction counts, accesses to register files, and memory, etc.) provided by the simulator.

For our SoC evaluations, we extended McPAT to *McPAT-SoC* by adding power (dynamic and leakage) and silicon area models of a broad set of embedded cores, such as the Intel Atom Diamondville [22] and the ARM Cortex A9 [9] cores.

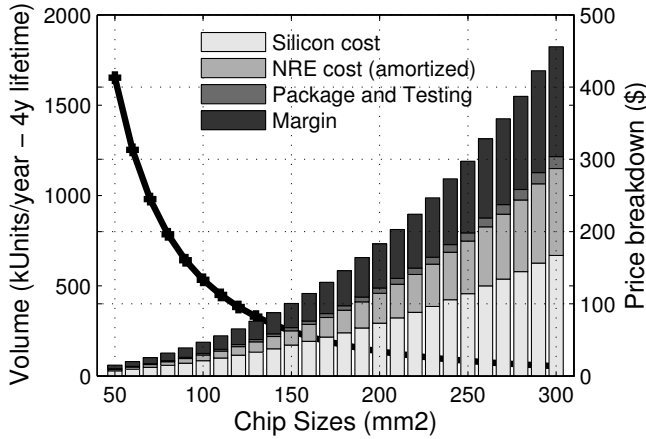


Figure 4: Example of cost and volume trends over chip size showing the non-linear cost-area relationship generated by the chip cost model, assuming a 32nm technology, a TAM of 200M\$/year, 4-year lifetime, and a gross margin of 50%.

In McPAT-SoC, we also developed new models for the SoC components described in Section 2: NIC, PCIe, and storage controllers (SATA), as well as high speed system interfaces such as Intel QuickPath Interconnect (QPI) [23] and AMD HyperTransport [6].

To model the control logic of the new SoC components we use Cadence’s ChipEstimator [13], which provides power, area, and timing information of different controllers taken from an extensive library of IP (Intellectual Property) SoC blocks. To account for the additional RAM buffers and DMA channels, we adjust the power and area results by modeling these buffers with the latest version of CACTI [34]. The 10Gb Ethernet controller is a special case because the controller usually contains extra TCP/IP offloading accelerators and packet filters since the processor overhead of supporting a NIC without accelerators is very high—a 10Gb NIC port can easily saturate a 2.33 GHz Intel Xeon E5345 core [41]. Thus, it is common for server class NICs to include TCP/IP accelerators, which can be seen in Niagara2 [25] and Niagara3 [46], as well as off-chip server-grade NICs such as the Broadcom 10Gb MAC controller [12]. We follow this design choice and assume a TCP/IP-accelerated NIC in our designs and models.

The physical interfaces (PHYs) in these controllers are high speed point-to-point SerDes links which we model using the parameters of a high performance design from Texas Instruments [19] (0.45 mm² and 26.4mW/Gbs per lane at 65nm). We then scale the area, bandwidth, and power to different technology generations based on SerDes scaling trends, as summarized in [39]. We did not consider the use of a lower power SerDes [17, 39] since they are only used for short distance communications (channel attenuation of −12 dB/−15 dB and maximum bit-error rate of $10^{-12}/10^{-15}$) and are inadequate in typical blade-based server designs [21] that support a backplane crossing (channel attenuation of −24 dB and a maximum error rate of 10^{-17}).

4.3 Chip Cost Modeling

An important factor of the cost component of the TCO is

the cost of the processor SoC itself. For this, we have built a processor cost model which takes into account the silicon area, an approximation of the non-recurring engineering (NRE) amortized over an estimated part volume, and packaging and cooling (e.g. fan and heatsink). The packaging and cooling model includes a power component (up to 1\$/W [29], depending on chip power and power density) and a pin component (depending on the die size, based on ITRS projections [44]).

We base our die cost estimate on standard silicon yield equations [27], using a negative binomial distribution of the defect density with industry standard defect clustering parameters, such as those used in ASIC cost estimation spreadsheets [28]. Our NRE approximation includes mask costs ranging from \$1.5M (65nm) to \$3.5M (16nm), and two Silicon spins per design. To estimate the design costs, we compute a design effort for the logic gates from the die size, the ratio of logic/non-logic (such as memory and pads) area, and the gate density for each process node. We also assume a productivity in gates/engineer-month that improves with technology, to account for more IP reuse and a larger degree of design regularity. We then scale the design effort with industry standard engineering costs.

To compute the volume over which NRE costs are amortized, our model assumes a fixed Total Addressable Market (TAM) spending for the chip over a standard 4-year lifetime. Specifically, we use a TAM of 200M\$/year, which is consistent with aggressive industry projections for the cloud datacenter market. In terms of our business model, we assume that the SoC (over which the NRE is amortized) is sold to OEMs who can spend for the SoC up to TAM\$ to build the servers they will sell to multiple customers, whose datacenter properties are the entities that ultimately see the TCO reduction benefits.

For example, Figure 4 shows an example of our computed costs for a 32nm node. To account for the difference in price vs. cost, we assume a flat 50% gross margin on top of the computed cost. Note that the assumed 50% gross margin is smaller than the average gross margin of general purpose CPUs (e.g. Intel has 65% gross margin for general purpose CPUs) because of the relatively smaller market size compared to the general purpose CPUs. Since we assume a fixed TAM, the volume (line on the left axis) of the chip decreases rapidly as the SoC size increases. This, in turn, increases the amortized NRE fraction per SoC, hence requiring a larger margin per part to meet the target gross margin (50% in the example), and causing a higher chip price (bars on the right axis) that then negatively impacts the acquisition cost fraction of the TCO computation.

4.4 Factoring in Usage Scenarios

One important aspect of our framework is framing the target design space. Because SoCs offer such a broad design space, it is important to determine the necessary resources and architecture based on the expected usage scenarios. To this goal, we ran several throughput-oriented workloads on a cluster of real servers to obtain utilization traces. Our workloads include a combination of typical applications in all three tiers of a scale-out datacenter: a video-hosting web server based on modified SPECweb2005 to model YouTube-like video-hosting (Tier 1); distributed sort implemented in Parallel DataSeries library, a Key Value Store running on top of VoltDB, and Hadoop’s grep/wordcount benchmarks

Table 2: Key characteristics of our usage scenario workloads. CPI, L2 and memory data comes from system-level sampling with *oprofile*. CPU, network and disk usage data comes from *sar*, averaged over several runs.

Benchmark	Time(s)	CPI	L2/1k inst	Mem(B/cycle)	CPU Usage	DISK Usage	Net Usage
key-value store	458	2.384	57.74	0.39	48%	0.0%	48%
distributed sort	1,046	2.927	79.93	0.49	63%	92.6%	25%
transactional OLTP	9,699	1.917	37.71	0.26	63%	0.3%	11%
web video serving	211	3.308	78.65	0.33	19%	0.2%	17%
hadoop wordcount	12,492	1.365	16.50	0.21	76%	2.2%	1%

(Tier 2); and a transactional in-memory OLTP workload with VoltDB (Tier 3). Each of these workloads was configured to process several gigabytes of data and run for multiple minutes.

We carefully selected the machines to run these workloads to ensure that they matched our target SoC core architecture well. We picked an older class of dual-core AMD Opteron 2.6 GHz based blade servers (with 8 GB of RAM, SAS local disks, and Debian Linux OS). The CPUs in these servers have a peak issue width of three instructions per cycle and a private 1MB L2 cache per core, and very closely approximate the A15-like processors that we assume in our study.

Running clusters of up to 16 real servers, we obtained traces using the *sar* logging tool for CPU, memory, disk, and network utilization. We additionally ran our workloads using the system-level profiler *Oprofile* to obtain more fine-grained CPU-level data including IPC, L2 access frequency, and memory access frequency. Table 2 summarizes the workload characteristics. We also used M5 [10] to run SPEC CPU2006, PARSEC, and SPLASH2 to both test memory bandwidth usage more comprehensively (as the throughput computing workloads may not represent worst-case memory subsystem usage) and increase our confidence of not under-provisioning the memory bandwidth to the cores.

The utilization traces and architecture profiles were first used as input to McPAT-SoC, along with the parameters of the SoC design targets. Based on the SoC target, McPAT determined a balanced resource allocation for best performance, and reported the area and peak power of the final design. Then, we assumed a workload distribution from recent IDC datacenter analysis [16] mimicking a real datacenter (45% Tier-1, 50% Tier-2, 5% Tier-3) and computed the relative usage frequency of each SoC component, which was used to modulate the peak power data to estimate the dynamic power consumption for the TCO model.

4.5 Datacenter TCO Modeling

In order to compare various SoC designs, we model the total cost of ownership (TCO) of large clusters for each type of SoC design. Our TCO model factors in multiple aspects of the individual elements of the clusters: processors, systems, boards, trays, and racks. At the SoC level we use the cost and power estimates computed as we described above. At the system level, we add DRAM and storage costs. Since we assume a design with multiple SoCs per tray, we then add motherboard, chipset and DC/DC power converter costs. At the tray level, we include AC/DC power supplies, fans, and material costs. Finally, at the rack level, we account for enclosure and top-of-rack (TOR) switch costs.

We use component pricing information collected from various public sources. While any pricing information comes with a significant range and can change over time (for example, we do not account for preferred partner or volume

discounts), we pick representative values that most importantly capture the ratio of costs between the various components. For PCB motherboards, since no comparable SoC-based data is publicly available, we assume a cost reduction of 40% compared to non-SoC design to account for the smaller board with fewer parts. To address potential sensitivity issues, we tested values between 10-40%, and found that it impacts our results by at most 5%, and in many cases less than 3%.

Capital costs are amortized with a 4 year depreciation schedule. Power costs are computed at \$70 per MWh. For the SoC and chipset components, the power consumption comes from the results of our real machine measurements and McPAT-SoC models, whereas the other components such as top of rack (TOR) switches use published numbers. We assume that the facility can provision the total dynamic runtime power, obtained using McPAT-SoC and usage scenario workloads discussed earlier. Power overhead (delivery, cooling, etc.) is modeled through a power usage effectiveness (PUE) factor of 1.2, the state of the art for efficient warehouse-scale datacenters. While our model can model sophisticated provisioning models (e.g., power capping) and include facility-level elements (e.g. power delivery infrastructure, real estate), we leave them out to better isolate microarchitecture impacts. More importantly, the power and cost savings of SoC servers that we showed will motivate rethinking the entire power delivery, packaging and cooling infrastructure. This could be an important future research direction that our work enables, and from this perspective the TCO savings analysis in following sections (Figure 8) are conservative.

4.6 Modeling Framework Validation

Our modeling framework targets a vast space, and we approached validation in layers to make the problem tractable. We validated the accuracy of our modeling framework at chip-level (mainly the McPAT-SoC), server level, and datacenter level.

The base McPAT accuracy was discussed in the original paper [33]; and we extended McPAT to model embedded cores and SoC components. We validated the output of McPAT-SoC against published data for the 45nm dual-core Diamondville [22] Atom processor running at 1.6GHz with a 1.0V power supply, and the 40nm Cortex A9 dual-core hard IP implementation running at 2.0GHz [9]. We also validated the I/O controller models in McPAT-SoC against published data (total area and power of all I/O controllers) of the Niagara 2 SoC. Table 3 (upper) shows the comparison of total power and area for modeled results against the target processors. The modeling and validating process against the A9 core ensures the accuracy of our models on embedded-class cores. The differences between the total peak power generated by McPAT-SoC and reported data are under 5% and the area differences between the die area generated and

Table 3: Validation results of the modeling framework. At the processor level, McPAT-SoC results are validated with regard to total power and area of target processors and controllers. At the server level, modeled results are validated against two server configurations from top vendors. At the datacenter level, the modeled results are validated against public knowledge of datacenters [18, 20].

Processor power and Area	Published Power and Area	McPAT-SoC Results	McPAT-SoC error (%)
Atom Diamondville	8 W / 51.92 mm ²	7.74 W / 49.8 mm ²	-3.2 / -4.1
Cortex A9 Hard IP	1.9 W / 6.7 mm ²	1.86 W / 6.52 mm ²	-2.3 / -2.7
Niagara 2 I/Os	11.1 W / 96.8 mm ²	10.7W / 92.4 mm ²	-3.6 / -4.5
Server Price	Listed price (\$)	Tool estimated price (\$)	Difference (%)
Dell PowerEdge R410	4468	4375	-2
HP ProLiant DL360	6139	6223	1
Datacenter Cost Breakdown	Prior TCO model (%) [18]	Tool TCO model (%)	Difference (%)
Hardware	67	59	-8 %
Networking	13	14	1 %
Power	20	27	7 %

reported data from industry are all under 5% as well.

At the system level, we augmented our SoC/chipset prices with component prices collected from public sources. Figure 4 shows the SoC chip cost model which is found to be quite aligned to the (publicly available) prices of commercially available CPUs. To validate our tool at server level, we modeled two server configurations from top vendors [4, 5] and found our models to be within a few percent of the listed price, assuming a margin within industry standard typical range (Table 3 (middle)). At the datacenter level, we built our models upon public knowledge of datacenters [18, 20]. Although we model a slightly different scenario compared to the prior work [18] (our server configuration has higher power consumption), we compared the high-level breakdown of hardware, networking, and power costs to ensure our tools captured similar overall trends. Using the assumptions outlined in section 4.5, we obtained the breakdown shown in Table 3 (lower), demonstrating our model to be within a few percent of the previous work [18].

5. EVALUATION AND RESULTS: FROM SOCS TO DATACENTER TCO

Evaluating the impact of chip-level technology changes at the datacenter scale is a very challenging endeavor. For this reason, we present our results in layers, starting from the quantification of cost benefits and efficiency at the single-

chip SoC level, and then build up from single SoCs to datacenter TCO.

We want to answer three key questions: (1) how much benefit can SoCs provide at a single-chip and a datacenter level; (2) how will this benefit scale with different technologies and chip sizes; and (3) what are the most cost-effective SoC configurations?

5.1 Experimental Setup

Table 4 shows the architecture parameters we used for each technology generation. As mentioned in Section 3, we chose a high-end embedded-class core (similar to ARM A15) as the SoC building block. We start from a core frequency of 2GHz at 45nm, and increase it conservatively by around 15% every technology generation to achieve power efficiency. We also start from a die size around 50 mm² at 45nm (two cores and a private 1MB L2 cache each), and double the baseline number of cores when moving to the next technology generation. At each generation, we also consider increasing the number of cores until the die size reaches the approximately 300 mm² (corresponding to 16 cores at 45nm).

The non-SoC and SoC designs use the same many-core substrate—all cores share uncore components except the private L2 caches. To achieve balance, we allocate cores and uncore components based on average resource utilization ratios extracted from our workloads on real systems and the simulations as discussed in Section 4.4. For example (Table

Table 4: SoC parameters across technology generations. We consider four core counts (separated by “/”) per technology and one memory channel per controller. The core count determines the number of I/O controllers (for SoC and non-SoC designs). On-chip modules include interface logic to handle the communication with cores and on-chip cache. Off-chip modules include a PCIe channel with adequate bandwidth to communicate to the chipsets. High-speed links connect the chipsets to the processors. Based on PCIe and SATA PHYs roadmaps, SoCs use PCIe2.0 and SATA2.0 at 45nm, but PCIe3.0 and SATA3.0 at 32nm and beyond. The volume production times (years) of different technologies are based on the ITRS projections [44].

Parameters	45nm (2010)	32nm (2013)	22nm (2016)	16nm (2019)
Core count	2/4/8/16	4/8/16/32	8/16/32/64	16/32/64/128
Clock rate (GHz)	2.0	2.3	2.7	3.0
Memory controller count	1/1/2/4	1/2/4/7	2/3/6/11	2/3/6/12
10Gb Ethernet controller	1/1/1/1	1/1/1/2	1/1/2/4	1/2/4/8
PCIe controller	1x x8PCIe2.0	1x x8PCIe3.0	1x x8PCIe3.0	1x x8PCIe3.0
Storage controller	1/1/2/3	1/1/2/3	1/2/3/6	2/4/7/14
Main memory type	DDR3-1333	DDR3-1600	DDR4-2133	DDR4-4266

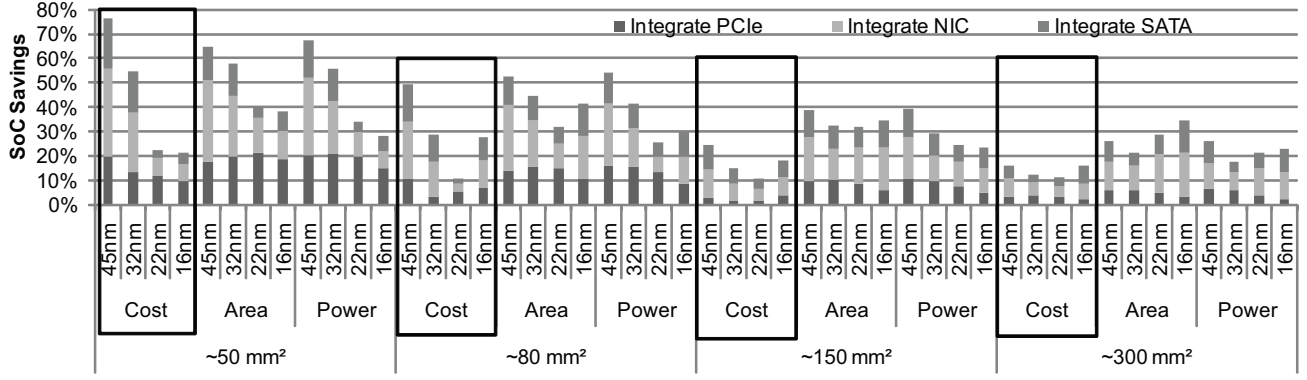


Figure 5: Single chip SoC benefits of all configurations across technology generations. The core count starts from 2 at 45nm for 50 mm², then doubles within the same technology and across different technologies. Thus, we show four different die sizes at each technology: around 50, 80, 150, and 300 mm². Each bar in the chart shows the total savings on cost, aggregated area, and aggregated power, compared to its *non-SoC* alternative. Each bar segment shows the individual contributions of integrating PCIe, NIC, and SATA controllers. Power savings include both dynamic and leakage power, with leakage adding up to about 20% to 30% of dynamic power across technology generations.

2) the CPU throughput to network speed ratio is around 6%, the CPU throughput to disk I/O ratio is around 7%, and the CPU throughput to memory bandwidth ratio is around 40%, which aligned with previous observations for similar scale-out workloads [42]. For our non-SoC designs as in Figure 2(a), PCIe channels are used to connect the off-chip modules to chipsets. In contrast, for our SoC-all designs as in Figure 1(b), all I/O modules are already on chip with dedicated PHYs, and do not require additional PCIe channels. We do however reserve one x8 PCIe channel for possible expansion to external high speed devices. The bandwidth for memory channels, PCIe PHYs, and SATA PHYs are assumed to scale based on the expected parameters at each technology node according to the JEDEC roadmap [24], PCIe roadmap [40], and SATA roadmap [45], respectively. Using the CPU throughput and bandwidth of I/O channels, we provision the I/O controllers at each technology node as shown in Table 4.

5.2 SoC Chip-level Analysis

Using McPAT-SoC, we first compute the power and area of core and uncore components across technologies from 45nm to 16nm covering through 2019 according to the ITRS roadmap. Based on general historical trends [12, 31], we assume the off-chip discrete components lag the CPU by two technology generations. When off-chip components are integrated, they use the same technology. Using the area and power results from McPAT-SoC and our chip cost model, we compute the cost of processors as well as the chipsets and off-chip components (if needed).

In addition to bridging the technology gap for off-chip components, SoCs dramatically reduce the expensive and power-hungry interface controllers and SerDes links needed for chip-to-chip pin crossings. These savings are substantial for high bandwidth chip-to-chip communication: for example, communicating to an off-chip 10Gbps NIC requires area and power comparable to a fully functional x4 PCIe2.0 controller (with PCIe control logic and link SerDes). Figure 5 shows the SoC savings on cost, aggregated area, and power, compared to the base non-SoC alternative as in Figure 2(a).

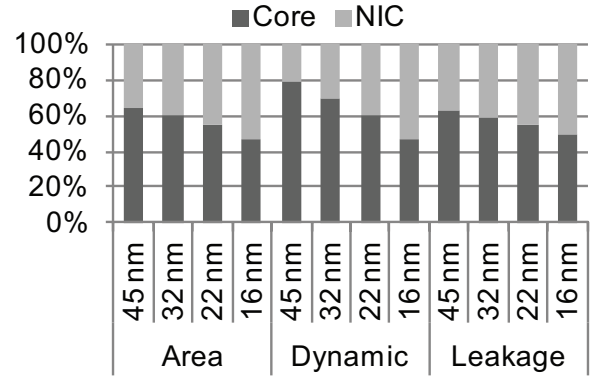


Figure 6: Scaling trends for the core and SoC I/O components. A 10Gb NIC (with its XAUI interface) is used to show the scaling trend of SoC I/O controllers. Based on McPAT-SoC, the area, maximum dynamic power, and leakage power of the core at 45nm are 7.6 mm², 2.3 W, and 0.45 W, respectively. The area, maximum dynamic power, and leakage power of the NIC at 45nm are 4.1mm², 0.63 W, and 0.27 W, respectively.

From non-SoC, we start by integrating PCIe, NIC, and storage controllers (SATA) onto the processor chip and form the configurations of SoC-PCIe as in Figure 2(b), SoC-NIC as in Figure 2(d), and SoC-SATA as in Figure 2(c). When integrating all system components, (*SoC-all*), chip-level cost savings range from 11% to 30%. This is due to the combination of a power savings of 26% to 54% and area savings of 13% to 41% across all technology generations and chip sizes. It is important to note that the lower bounds of the cost savings are not smaller than 11% for all cases.

Determining which system components to integrate is one of the major design decision points for server SoC architects. A first look at our results reveals that integrating the NIC is the most efficient choice due to the large area and power overhead of a discrete 10Gb NIC with its TCP offloading

Table 5: Modeling results of SoC-all configurations starting from 8 cores at 45nm (the configurations are chosen for their more visible turning points across technology generations). Uncore includes on-chip interconnects and all I/O controllers. The volume production times (years) of different technologies are based on ITRS projections [44].

Parameters	45nm (2010)	32nm (2013)	22nm (2016)	16nm (2019)
Core count	8	16	32	64
Core (area(mm ²) / Peak power (W))	61 / 22	63 / 28	65 / 35	67 / 45
L2 (area(mm ²) / Peak power (W))	42 / 6	42 / 7	43 / 8	43 / 9
Uncore (area(mm ²) / Peak power (W))	42 / 13	49 / 19	44 / 20	46 / 24
Chip (area(mm ²) / Peak power (W))	145 / 41	154 / 54	152 / 63	156 / 76

engine. Figure 5 also demonstrates the importance of integrating other system components. For example, integrating storage controllers (SATA controllers) and PCIe controllers can achieve more than 20% cost savings at 45nm. Although the benefits of integrating PCIe controllers drops as technology advances as shown in Figure 5, we have to keep in mind that we made rather conservative assumptions on the system requirement of PCIe links, assuming one x8 PCIe channel is sufficient for all configurations across all technologies. Thus, the impact of integrating PCIe becomes less important with technology.

Figure 5 also reveals the scaling trend of the SoC benefits, which start diminishing with technology, reach a turning point (22nm for area and 32nm for power), and increase again. This behavior is due to two contrasting trends. On one hand, core components show different scaling trends across technologies in power and area from on-chip uncore system components. The area and power of I/O controllers shrinks more slowly than cores and caches, eventually causing the the system I/O components to become larger contributors to the total chip area/power. For example, Figure 6 shows the area, maximum dynamic power, and leakage power of the core and SoC I/O components. Considering a 10 Gb NIC with its XAUI interface (which we call “the NIC” from here on), we can see how the area ratio versus the core is significant at 45nm but diminishes with technology as a result of different scaling behaviors. For each technology generation a typical core (or the NIC logic) achieves a 45% area reduction, but the analog XAUI part only shrinks by less than 20%. Overall, the NIC area only shrinks by 37% from 45nm to 32nm, and only by 26% from 32 nm to 16 nm. As a result, the core area at 45nm is 1.85 \times larger than the NIC, but at 16nm they are about the same size. Similar trends can be seen for dynamic and leakage power.

Were this the only dictating factor across technologies, scaling of SoCs should exhibit monotonic trends. The observed non-monotonic trend is, however, caused by another factor. The scaling of the number of system components per chip across different technologies does not keep up with the scaling trend of cores per chip as shown in Table 4, thereby reducing the proportion of components area/power to total chip area/power (e.g., doubling the number of cores doubles the required network bandwidth, but does not necessarily double the number of 10Gb NICs during technology or chip size scaling). The reason is the number of I/O controllers can only be changed in integer steps, which may cause under- or over-provisioning system resources vs. the cores and produce a non-optimal resource balancing. However, this is a constraint that an SoC architect would also have to face in real life, since performance scaling on cores is inherently different from that of system controllers across technology

generations. This is especially true in sever designs that rely on standard parts. Thus, SoC architects will have to pay attention to all the uncore components and understand how they scale for future technology generations in a similar way in which CPU architects learned processing elements and cache tradeoffs.

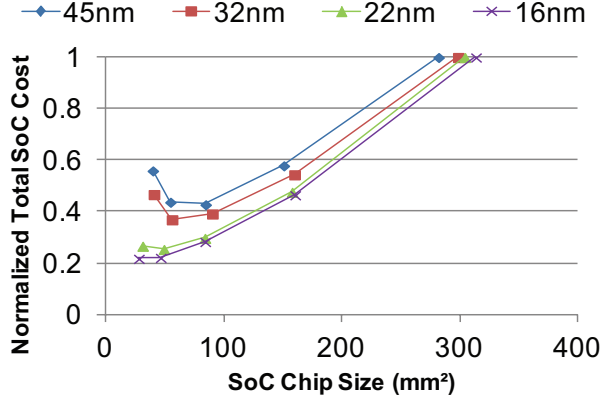
While Figure 5 shows the comparison between *SoC-all* and non/partial-SoC designs, Table 5 shows the detailed modeling results of *SoC-all* configurations across four technology generations. As shown in Figure 5, the configurations with 80 mm² and 150 mm² die sizes have more visible turning points than other configurations. Thus, in order to study both the overall chip scaling trends and the implications of different scaling trends of cores and I/O controllers, we chose the configurations with around 150 mm² die sizes (starting from 8 cores at 45nm and doubling the core count for each generation) as shown in Table 5. System components, including memory controllers, contribute to the non-monotonic scaling trend of the uncore part of the SoC across the technology generations, which in turn leads to the non-monotonic scaling of the chip size and the uneven scaling of the chip power. Despite the number of system components scaling slower than the number of cores and caches, they occupy significant portions of chip area and power, being more than 30%. Overall, the effects of these trends is for SoC components to remain important compared to cores and caches, and they indicate that system-level integration will bring significant benefits across technology generations.

5.3 Datacenter-level TCO Analysis

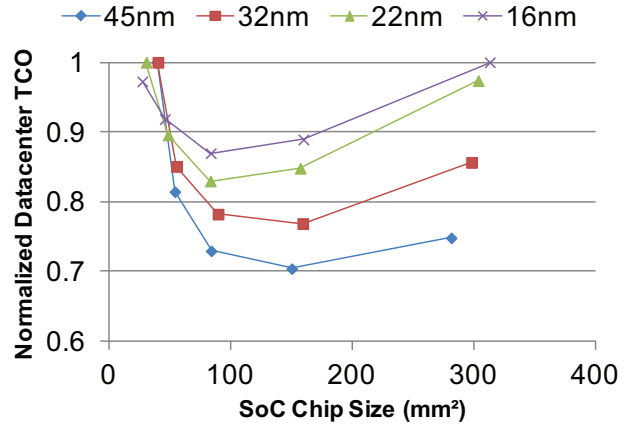
Since the design space of datacenters is large, when doing datacenter TCO analysis we first sweep the SoC configurations and establish which SoC-based design offers the most significant saving opportunities. For our datacenter-scale study, we assume a fixed target throughput performance with a cluster of 10,000 cores for each technology generation. Considering the per-core performance, it corresponds roughly to 4 to 9 racks (42U) across all our configurations.

First, we look at the component level with SoC chips only, and sweep the SoC sizes to find the die size yielding the lowest cost at each technology for the target performance. Figure 7(a) shows that, at the component level, smaller chips are more efficient, and this trend is even more visible at smaller (and more expensive) geometries. This is largely due to the nonlinear dependence of cost versus die size previously discussed.

Then, we want to see whether the “best” SoC at the component level is also the one that gives the highest savings in datacenter-level TCO. Since a datacenter contains more than just processors (including memories, storages, and interconnects), it turns out that the most TCO-efficient SoC



(a) Normalized cost for at component level for SoC only



(b) Normalized TCO at datacenter level

Figure 7: Cost analysis to identify the optimal configurations to reach a target performance at: (a) the component level, considering the SoC only, and (b) the datacenter TCO level including all the additional elements. SoC die sizes are varied across a range up to around 300 mm². All values are normalized to the most expensive configuration at each technology generation (lower is better). We assume a fixed target throughput performance with a datacenter of 10,000 cores.

chip size does not match the die-size sweet spot, when considering that several other elements (such as the DRAM/storage capacity, PCBs, trays, enclosures, and networks) also need to adequately scale with different SoC chip size. Figure 7(b) shows the normalized datacenter cost when using different sized SoCs. In that figure we can see that the “best” chip size shifts to the middle ground between 80 and 160 mm². For example, in 32nm the best SoC configuration is 154 mm², with 16 cores, 1 NIC, 2 storage (SATA) controllers, 1 x8 PCIe channel, and 4 memory controllers. In the server each tray has 8 SoCs.

Finally, we also want to quantify the benefits of SoC vs non-SoC configurations at the datacenter level. For that purpose, we examine the two primary components of TCO: per-year amortized acquisition capital costs and the ongoing energy cost (to run and cool the systems). In Figure 8, for each technology we pick the best configuration (yielding the lowest TCO as defined above), and we compare it to a non-SoC server design with the same performance (number of cores).

We can see that the savings are substantial for future datacenters. At 16nm *SoC-all* designs can provide a 23% improvement in capital costs versus the baseline non-SoC design. These cost savings come from a reduced bill of material due to integration (30% according to our die cost estimator), as well as a reduction in PCB costs and power supplies due to the lower total power usage achieved by eliminating pin-crossings. Moreover, the use of SoCs enables denser server designs compare to the non-SoC based server designs, since the processor count (with its chipsets and offchip components) is limited by the form factor of the server PCB and power supply. The denser server design also reduce the number of racks and expensive top of rack (TOR) switches, which further reduces datacenter cost. Thus the power savings are more than 35%, and the overall TCO savings are more than 26%. What is important to observe is the substantial impact of TCO improvements on a datacenter scale

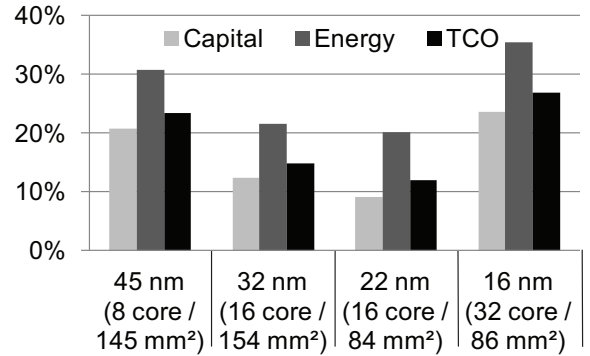


Figure 8: Savings on capital cost, energy cost, and TCO of the datacenter with a cluster of 10,000 cores and built using the best SoC configuration obtained from earlier analysis, compared to corresponding non-SoC baselines. Each data point shows the SoC configuration and corresponding chip size along the x-axis.

in terms of absolute TCO (\$) savings. For example, assuming a 150-rack warehouse-size datacenter, the 26% TCO reduction on the datacenter at 16 nm (year 2019) translates to a net savings of over 2.4M\$ per year.

6. CONCLUSIONS

We are entering the server system-level integration era where substantial system functionality will be integrated together with the processor die. This is a major paradigm shift that opens the door to the next level of power and cost optimizations that scale-out datacenters demand.

With this work, we have developed a comprehensive evaluation framework, and explored the design space from cores to datacenters to find the best improvements that the adop-

tion of system-level integration can provide. Our initial results are very promising for future data centers: reductions of more than 23% in capital cost, 35% in power costs, and more than 26% in overall datacenter TCO at 16 nm in year 2019. These are substantial improvements and translate to millions of dollars of yearly savings when scaled to warehouse-sized cloud datacenters.

We have just scratched the surface of the architectural issues around system-level integration for servers, and some of the directions we outlined deserve much deeper scrutiny. We hope this work will inspire possible future research in this area, such as evaluating I/O sharing and aggregation through advanced on-chip networking, or the integration of accelerators.

7. ACKNOWLEDGEMENTS

The authors would like to thank Steven K. Reinhardt at AMD and Benjamin C. Lee at Duke University for the feedback on the early draft of this paper and the anonymous reviewers for their constructive comments.

8. REFERENCES

- [1] <http://www.calxeda.com/>.
- [2] <http://www.seamicro.com/>.
- [3] <http://www.eurocloudserver.com/>.
- [4] <http://www.dell.com/>.
- [5] <http://www.hp.com/>.
- [6] AMD, "HyperTransport Technology: Simplifying System Design," Tech. Rep., 2002.
- [7] AMD, "AMD Opteron Processor Benchmarking for Clustered Systems," *AMD WhitePaper*, 2003.
- [8] D. G. Andersen, *et al.*, "FAWN: a Fast Array of Wimpy Nodes," in *SOSP '09*, 2009, pp. 1–14.
- [9] ARM, <http://www.arm.com/products/processors/cortex-a/cortex-a9.php>.
- [10] N. L. Binkert, *et al.*, "The M5 Simulator: Modeling Networked Systems," *IEEE Micro*, vol. 26, no. 4, pp. 52–60, 2006.
- [11] M. Bohr, "Silicon Technology for 32 nm and Beyond System-on-Chip Products," in *IDF'09*, 2009.
- [12] Broadcom, "BCM57710 - Dual-Port 10G/2500/1000BASE-X TOE, RDMA, iSCSI PCI Express Ethernet Controller," Tech. Rep., 2008.
- [13] Cadence InCyte Chip Estimator, "<http://www.chipestimate.com/>."
- [14] B. Carlso, "Going Beyond a Faster Horse to Transform Mobile Devices," Texas Instruments, Tech. Rep., May 2011.
- [15] A. M. Caulfield, L. M. Grupp, and S. Swanson, "Gordon: Using Flash Memory to Build Fast, Power-efficient Clusters for Data-intensive Applications," in *ASPLOS '09*, 2009.
- [16] M. Eastwood and M. Bailey, "Server Workloads Forecasts and Analysis Study, 2005 - 2010," IDC Special Study, Tech. Rep., 2010.
- [17] K. Fukuda, *et al.*, "A 12.3mW 12.5Gb/s complete transceiver in 65nm CMOS," in *ISSCC'10*, 2010, pp. 368–369.
- [18] J. Hamilton, "Overall Data Center Costs," <http://perspectives.mvdirona.com/2010/09/18/OverallDataCenterCosts.aspx>.
- [19] M. Harwood, *et al.*, "A 12.5Gb/s SerDes in 65nm CMOS Using a Baud-Rate ADC with Digital Receiver Equalization and Clock Recovery," in *ISSCC'07*, 2007, pp. 436–439.
- [20] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.
- [21] HP, "HP BladeSystem c-Class SAN connectivity technology brief," Tech. Rep., 2009.
- [22] Intel, <http://www.intel.com/products/processor/atom/techdocs.htm>.
- [23] Intel, "An Introduction to the Intel QuickPath Interconnect," Tech. Rep., 2009.
- [24] JEDEC Solid State Technology Association, "<http://www.jedec.org/>."
- [25] T. Johnson and U. Nawathe, "An 8-core, 64-thread, 64-bit Power Efficient Sparc SoC (Niagara2)," in *ISPD*, 2007.
- [26] R. Jotwani, "An x86-64 Core Implemented in 32nm SOI CMOS," in *ISSCC'10*, 2010.
- [27] H. Kaeslin, *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*, 1st ed. Cambridge University Press, April 2008.
- [28] Kaeslin, Hubert, "ASIC Cost Estimator webpage at <http://www.dz.ee.ethz.ch/?id=1592>."
- [29] A. Kahng, "The Road Ahead: The significance of packaging," *IEEE Design and Test of Computers*, vol. 19, pp. 104–105, 2002.
- [30] T. Kgil, *et al.*, "PicoServer: Using 3D Stacking Technology to Enable a Compact Energy Efficient Chip Multiprocessor," in *ASPLOS*, 2006.
- [31] R. Kumar and G. Hinton, "A family of 45nm IA processors," *ISSCC*, pp. 58–59, 2009.
- [32] T. Lanier, "Exploring the Design of the Cortex-A15 Processor," ARM, Tech. Rep.
- [33] S. Li, *et al.*, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," in *MICRO '09*, 2009, pp. 469–480.
- [34] S. Li, K. Chen, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "CACTI-P: Architecture-Level Modeling for SRAM-based Structures with Advanced Leakage Reduction Techniques," in *ICCAD*, 2011.
- [35] K. Lim, *et al.*, "Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments," in *ISCA '08*, 2008.
- [36] Marvell, "Marvell Unveils 1.6GHz Quad-Core ARMADA XP Platform for Enterprise Class Cloud Computing Applications, The Wall Street Journal, November 2010."
- [37] C. R. Moore, "Microarchitecture in the system-level integration era," *Keynote at MICRO-34*, 2008.
- [38] T. Mudge and U. Holzle, "Challenges and opportunities for extremely energy efficient processors," *IEEE Micro*, vol. 30, 2010.
- [39] R. Palmer, *et al.*, "A 14mW 6.25Gb/s Transceiver in 90nm CMOS for Serial Chip-to-Chip Communications," in *ISSCC'07*, 2007, pp. 440–441.
- [40] PCI Special Interest Group, "<http://www.pcisig.com/>."
- [41] K. K. Ram, J. R. Santos, Y. Turner, A. L. Cox, and S. Rixner, "Achieving 10 Gb/s using safe and transparent network interface virtualization," in *VEE*, 2009, pp. 61–70.
- [42] E. Riedel, G. A. Gibson, and C. Faloutsos, "Active storage for large-scale data mining and multimedia," in *VLDB '98*, 1998, pp. 62–73.
- [43] S. Rusu, *et al.*, "A 65-nm Dual-Core Multithreaded Xeon Processor With 16-MB L3 Cache," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, Jan 2007.
- [44] Semiconductor Industries Association, "International Technology Roadmap for Semiconductors./ Model for Assessment of CMOS Technologies and Roadmaps (MASTAR) <http://www.itrs.net/>."
- [45] Serial ATA International Organization, "<http://www.sata-io.org/>."
- [46] J. Shin, *et al.*, "A 40nm 16-Core 128-Thread CMT SPARC SoC Processor," in *ISSCC'10*, 2010, pp. 98–99.
- [47] M. Yaffe, E. Knoll, M. Mehal, J. Shor, and T. Kurts, "A Fully Integrated Multi-CPU, GPU and Memory Controller 32nm Processor," in *ISSCC*, Feb. 2011, pp. 264–266.