Customizing Servers for Emerging Scale-out Workloads Using CloudSuite

Almutaz Adileh, Michael Ferdman‡, Onur Kocberber, Stavros Volos, Mohammad Alisafaee,
Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Boris Grot, Anastasia Ailamaki, and Babak Falsafi

EcoCloud
École Polytechnique Fédérale de Lausanne

‡CALCM
Carnegie Mellon University

Emerging global-scale online services require extensive amounts of computational resources. Today's popular online services, such as web search, social networks, and video sharing, are all hosted in large datacenters. With the industry rapidly expanding, service providers are building new datacenters, augmenting the existing infrastructure to meet the increasing demand. However, while the demand for datacenter infrastructure continues to grow, the semiconductor manufacturing industry has reached the physical limits of voltage scaling [6], and is no longer able to reduce power consumption or increase power density in new chips. Physical constraints have therefore become the dominant limiting factor for datacenters, because their sheer size and electrical power demands cannot be met. Therefore, service providers now optimize their datacenters for compute density and power consumption, and this mandates optimizing server efficiency to ensure that server hardware closely matches the needs of emerging workloads.

Analyzing popular datacenter services, we found that datacenter servers are running scale-out workloads that share similar characteristics; all applications we examined (1) operate on large data sets that are split across a large number of machines, typically into memory-resident shards, (2) serve large numbers of completely independent requests that do not share any state, and (3) use inter-machine connectivity only for high-level task management and coordination. In general, the scale-out characteristics of datacenter applications place them into a distinct class of applications that cannot be represented by conventional benchmark suites that have driven server designs in the past.

Scale-out workloads also exhibit a different micro-architectural footprint than desktop, parallel and even traditional server workloads. Our recent study [2] shows that popular datacenter workloads differ from traditional ones in their compute requirements and their usage of the memory hierarchy. In general, the study identifies several inefficiencies of modern processors running scale-out workloads:

- Data working sets of scale-out workloads considerably exceed the capacity of on-chip caches. Therefore, processor real-estate and power are misspent on large last-level caches that do not contribute to improved scale-out workload performance.
- Scale-out workloads suffer from high instruction-cache miss rates that cannot be mitigated by next-line prefetchers. Both a modest second-level cache and a large last-level cache can only worsen the problem by increasing the effective instruction access latency.
- Instruction- and memory-level parallelism in scale-out workloads is low. Modern aggressive out-of-order cores are excessively complex, consuming power and on-chip area without providing performance benefits to scale-out workloads.
- On-chip and off-chip bandwidth requirements of scale-out workloads are low. As a result, scale-out workloads see no benefit from fine-grained coherence and high memory and core-to-core communication bandwidth.

The results demonstrate critical mismatches between the modern server-class processors, built for a broad range of conventional workloads, and the requirements of scale-out applications.

In this era, characterized by strict energy and physical limits and the emergence of global-scale, data-intensive scale-out workloads, it is clear that designing datacenters should strictly follow a workload driven approach. The processing and memory components that populate datacenters should be designed and even specialized starting from a clear understanding of the workload demands. Having a standard suite of benchmarks that represent scale-out, data-intensive workloads and target their emerging trends is an essential first step.

We have taken the first step toward realizing this vision by creating CloudSuite [1], a benchmark suite that represents emerging scale-out applications. We propose CloudSuite as a basis of a standard big data benchmark suite. To find a set of applications that dominate today's datacenter infrastructure, we examined a selection of internet services based on their popularity [3]. For each popular service, we analyzed the class of application software used by the major providers to offer these services. The first release of CloudSuite includes:

- Data Analytics: Big data analytics using machine learning on Hadoop MapReduce framework. Representing big data mining for useful information.
- Data Serving: Serving data using a NoSQL database. Representing backend stores in large-scale applications.
- Media Streaming: Representing popular media streaming services.
- Software Testing: Software testing distributed and provided as a cloud service.
- Web Search: Representing search engines powering popular services such as Google and Bing.
- Web Serving: Web frontend serving a web2.0 application.

CloudSuite is designed to represent the popular applications that run in today's datacenters. The software stacks of CloudSuite cover emerging datacenter processing and storage techniques, e.g., MapReduce and NoSQL. Moreover, CloudSuite benchmarks capture a variety of dataset types, e.g., structured, unstructured and multimedia. Furthermore, CloudSuite includes interactive and patch job applications. The interactive applications deploy flexible client simulators that can emulate real-world distributions and request mixes. Overall, CloudSuite has the diversity and flexibility that allows it to benchmark a wide variety of testing scenarios.

We consider CloudSuite as a first step towards scale-out benchmarking. The first release will help drive research on scale-out applications that will lead to improving datacenter efficiency. For server designers, supporting large-scale hardware simulation of scale-out applications is a necessity to pursue in the near future. We are in the process of porting CloudSuite to the Flexus [4][5] simulation infrastructure, paving the road for researchers to put their new designs and techniques under evaluation.

Our hope is that CloudSuite forms the basis of a standard and widely accepted benchmark suite for both industrial and academic research. Our future plans include expanding the current set of benchmarks to represent popular workloads and reflect the state-of-the-art. Currently we are evaluating a number of potential additions to the suite, including distributed caching and machine translation. The more representative the benchmarks are, the higher impact our research will have on datacenter efficiency improvement.

References:

1. http://parsa.epfl.ch/cloudsuite
2. Michael Ferdman, Almutaz Adileh, Onur Kocberber, Stavros Volos, Mohammad Alisafaee, Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Anastasia Ailamaki, and Babak Falsafi. Clearing the Clouds: A Study of Emerging Workloads on Modern Hardware. *In 17th International Conference on Architectural Support for Programming Languages and Operating Systems, March 2012.*
3. Alexa, The Web Information Company. http://www.alexa.com/
4. Thomas F. Wenisch, Roland E. Wunderlich, Mike Ferdman, Anastassia Ailamaki, Babak Falsafi, and James C. Hoe. SimFlex: Statistical Sampling of Computer Architecture Simulation. *In IEEE Micro special issue on Computer Architecture Simulation, July/August 2006.*
5. CloudSuite on Flexus ISCA tutorial. http://parsa.epfl.ch/cloudsuite/isca12-tutorial
6. Nikos Hardavellas, Michael Ferdman, Anastasia Ailamaki, and Babak Falsafi. Toward Dark Silicon in Servers. *In IEEE Micro, 31(4):6–15, July-Aug, 2011.*

Onur Kocberber is a third-year PhD candidate in the Parallel Systems Architecture Laboratory at EPFL, where he is advised by Prof. Babak Falsafi. He is the release co-manager of CloudSuite and a co-developer of the Flexus simulation framework. His research interests target utilizing dark silicon for emerging data-oriented services.