RETROSPECTIVE: Profiling a warehouse-scale computer

Svilen Kanev, Juan Pablo Darago, Kim Hazelwood, Parthasarathy Ranganathan, Tipp Moseley, Gu-Yeon Wei, David Brooks

We wrote “Profiling a WSC” for ISCA 2015 in order to summarize and share our experiences extracting performance information from Google’s datacenter fleet. The paper outlined performance bottlenecks and trends that we observed, both in software and hardware, and sketched out a handful of areas for future research. It was intentionally light on solutions and meant to inspire directions for others’ work. Almost ten years later, we reflect on the predictions the paper made, check back on which ones panned out, and which of the trends have caught on in industry or academia.

IMPACT AND CONTRIBUTIONS

Datacenter tax. Arguably the biggest contribution of the paper was coining the term “datacenter tax” – referring to the non-application logic required to run distributed services. Since the paper was published, many others have validated it as an industry-wide phenomenon and used the term to justify reducing these overheads. As just one example, multiple key services in Meta datacenters were shown to spend 50+% of their compute cycles in “tax” code.

Hardware “tax” optimizations. We initially proposed classifying functions as “datacenter tax” mostly based on how easy they would be to accelerate in hardware (this is ISCA after all!) – this is why we opted for broadly applicable, low-level code, that is relatively mature and self-contained. Multiple follow-up efforts have taken up designing accelerators for different tax routines. These have included both commercial designs – e.g. Intel’s Infrastructure Processing Unit (which accelerates compression and encryption) [8]; as well as open-source academic ones – accelerators for protocol buffers [10], compression [11], memory allocation [9].

One takeaway from these efforts is that broad acceleration of “tax”-like routines is qualitatively different from more traditional deep acceleration. Accelerating an ensemble of different bottlenecks, worth 5% each, is very different from speeding up the whole application “deeply”. Not only does it take more effort, but the potential gains are smaller by definition. It also requires very careful thinking about accelerator placement and communication requirements, and enforces much tighter budgets (in area/power) due to the limited opportunity.

Software “tax” optimizations. While our main motivation for classifying cycles as “datacenter tax” initially was hardware accelerators, we have since realized that optimizing tax functions in software can be incredibly fruitful. Due to these low-level functions’ ubiquity, a small number of optimization experts can focus their attention on “tax” code and release optimizations to thousands of services at a time. If the same engineers were to go binary-by-binary and focus on application logic, they would have to spend a lot of effort on each application’s idiosyncrasies, and the sophistication of optimizations could suffer. Focusing on a very small set of shared routines instead allows for very outsized impact.

Optimizing datacenter tax in software has been a very concerted effort inside Google’s WSCs. As just one example data point, since the paper was written, a gaggle of different optimizations over 3 years has resulted in a nearly 2x reduction in memory allocation cycles. After the relatively low-hanging optimization fruit, the scope of that work has evolved from “reduce time spent in datacenter tax” to “optimize tax routines holistically for application productivity” [6]. These sometimes make tradeoffs that increase tax cycles, but are a net positive because of positive externalities (like reduced TLB pressure).

Workload diversification. Another trend we identified was the increasing workload diversity in warehouse-scale computers. That has continued over the years, with compute cycles spread over more and more services. It has specifically accelerated with the broader usage of public clouds.

Recent trends on machine learning workloads, however, add a bit more nuance to our point about diversification. Since the paper was written in 2014, of course, machine learning has taken off, with an insatiable appetite for compute cycles. Training and serving large models does concentrate cycles on a small number of workloads. However, the models themselves still change so rapidly that, over yearlong periods, we still observe significant workload diversity.

General-purpose CPU trends. The paper also analyzed the performance bottlenecks for general-purpose CPUs in a lot of detail. Two of these deserve special mention.

CPU frontends. While not the first work to bring significant attention to the CPU frontend as an increasingly important bottleneck for scale out workloads [4], “Profiling a WSC” confirmed that trend and was often used to motivate frontend improvements. Some follow-up work has included deeper characterization [3], new frontend structures [13], [17], compiler solutions [15], [16], and the adoption of software prefetching for code as standard in the industry [7].

1Note that this is made possible by Google’s single-repository, code-lives-at-head approach to software development [19].
Memory bandwidth vs latency. The paper predicted that memory bandwidth was going to be less important than latency, and that’s a trend we got wrong. Since 2014, several factors have contributed to bandwidth becoming a major bottleneck in WSCs. On the one hand, there is more demand for memory bandwidth – due largely to the rise in machine learning, ever growing core counts, and the mainstream adoption of chiplet architectures (which cause more duplication than a monolithic cache). On the other, bandwidth supply is still limited, so bandwidth is becoming ever so scarce.

Does it even matter? When the authors were discussing the contents of this document, an interesting debate formed. One camp argued the original paper was too heavy on microarchitectural details, and that is less and less important in today’s environment dominated by system-wide concerns (disk, network, memory bandwidth) and accelerators (GPUs, TPUs). The other camp argued that general-purpose architecture is less glorious, but still continues to make large strides over time. Since the IvyBridges we profiled in 2014, the amount of compute per socket has improved by a factor of 10, through a series of 1% incremental roofshots. Eventually, while we all agreed that the paper could have benefited from a more stronger systems focus, we didn’t reach a resolution on the role of future microarchitectural innovation – solving philosophical debates is an exercise left to the reader.

Methodology contributions. Benchmarks. The paper joined a small choir arguing that using SPEC CPU is not representative of WSC workloads. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representative. For a long time we even considered a small choir arguing that using SPEC CPU is not representativ...