CHAMPVis: Comparative Hierarchical Analysis of Microarchitectural Performance

Lillian Pentecost*, Udit Gupta*, Elisa Ngan*,
Gu-Yeon Wei, David Brooks, Johanna Beyer, Michael Behrisch
Harvard University
Cambridge, MA

***Abstract*—Performance analysis and optimization are essential tasks for hardware and software engineers. In the age of datacenter-scale computing, it is particularly important to conduct comparative performance analysis to understand discrepancies and limitations among different hardware systems and applications. However, there is a distinct lack of productive visualization tools for these comparisons.**

We present CHAMPVis [1], a web-based, interactive visualization tool that leverages the hierarchical organization of hardware systems to enable productive performance analysis. With CHAMPVis, users can make definitive performance comparisons across applications or hardware platforms. In addition, CHAMPVis provides methods to rank and cluster based on performance metrics to identify common optimization opportunities. Our thorough task analysis reveals three types of datacenter-scale performance analysis tasks: summarization, detailed comparative analysis, and interactive performance bottleneck identification. We propose techniques for each class of tasks including (1) 1-D feature space projection for similarity analysis; (2) Hierarchical parallel coordinates for comparative analysis; and (3) User interactions for rapid diagnostic queries to identify optimization targets. We evaluate CHAMPVis by analyzing standard datacenter applications and machine learning benchmarks in two different case studies.

I. Introduction

Datacenters are the cornerstone of many internet services, including search, online retail, social media, and scientific computing. As these services expand globally, so does the amount of computing resources devoted to them. For example, over the next 20 years, datacenters are expected to consume up to 5% of world-wide energy [2]. At this scale, even small performance optimizations in the software or hardware in a datacenter can translate to large savings [3], [4].

The first step to optimization is identifying performance bottlenecks. The computer systems community has made significant strides in building tools for bottleneck analysis. For example, Eyerman et al. [5] log hardware events symptomatic of poor performance. TopDown [6] records the performance impact of these events and outlines a structured, hierarchical methodology to categorize critical bottlenecks. This methodology was adopted in commercial tools including Intel VTune [7] and Linux *perf* [8]. However, these tools only consider a single application and hardware platform at a time, making comparative analysis laborious and time-consuming. CHAMPVis enables using TopDown-style hierarchical performance analysis across multiple applications.

Supporting comparative performance analysis across applications and hardware platforms is key for datacenter-scale optimizations. Typical datacenters contain thousands of machines which are grouped into tens of machine types, and each essential application spans thousands of lines of code [9]. By analyzing a diverse set of applications or hardware platforms, performance optimizations can be applied to a wider range of inefficiencies. For example, Google and Facebook apply TopDown performance analysis within their datacenters [4], [10]. These works led to solutions that reduced total datacenter cycles by up to 2%. However, researchers are currently limited to studying only the most abstract level in TopDown — deeper evaluations would require visualization techniques tailored to datacenter-scale hierarchical comparisons.

To enable *productive*, *efficient*, and *user-friendly* performance comparisons, this paper presents **CHAMPVis**. Our tool facilitates identifying similarities and differences across different software applications and hardware platforms. We consider key design decisions such as how to visually relate performance metrics at different levels of the hierarchical hardware structure, how to guide users through the analysis process, and how to compare the performance of different applications and hardware platforms across multi-dimensional metrics. CHAMPVis allows users to directly interact with and extract useful details from hierarchical performance analysis data while simultaneously making effective comparisons. This closes the loop between identifying an important performance bottleneck and comparing results across software applications or hardware platforms. This will enable datacenter system engineers and software application developers to more productively optimize performance.

This paper makes three main contributions:

1) High-level similarity analysis across applications by projecting multi-dimensional performance data onto a 1-D feature space, thereby providing a quick comparative analysis view.

2) Interactive hierarchical performance analysis and detailed comparison of trends, which allows users to identify key optimization targets.

3) User-guided performance analysis that leverages a user's

*Authors contributed equally to this work.

978-1-7281-6026-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ProTools49597.2019.00013

domain knowledge by supporting interactive filtering, clustering, and interactive navigation in different comparative views. Users can immediately see the results of their diagnostic queries for searching and filtering, which reduces lengthy performance analysis cycles.

II. Background

Modern datacenters comprise a heterogeneous mix of applications and hardware platforms that makes performance analysis a challenging and laborious task [11]. To gain benefits from performance optimization in a datacenter, optimizations must target multiple applications and hardware platforms. However, each application and hardware platform has unique characteristics which influence the overall performance. This section discusses how performance counters are used to identify these characteristics.

A. Performance counters

Performance Measurement Units (PMUs) in modern CPUs collect statistics about specific hardware events, and these statistics are referred to as performance counters (PCs). Hardware events are recorded during the execution of an application on a given hardware platform. Many performance measurement strategies leveraging PCs have been explored (Section IV). CHAMPVis uses TopDown [6] — a methodology that relates individual, low-level hardware events to the fraction of time each application spends “waiting” on the given hardware resource. For an expert user, analysis of these statistics reveals possible optimization targets.

B. Hierarchical organization of computer systems

Modern CPUs are organized as a hierarchy of hardware components. At the highest level, every CPU cycle (time) spent on a given application is assigned to one of four stages of program execution: frontend, speculation, backend, and retiring. The frontend stage is responsible for decoding the work to be performed. The speculation stage refers to predictive optimizations that enable higher performance. The backend stage comprises loading and storing data to memory, and the retiring stage refers to computations.

These four stages can be broken down hierarchically into more specific sub-systems, as shown in Figure 1. We refer to these sub-systems as the levels of the TopDown hierarchy [6]. The TopDown hierarchical performance analysis methodology uses profiling to determine the fraction of computing cycles the application is stalled in a stage of computation or hardware sub-system. Understanding the relative fractions of compute cycles in the different stages of level 1 (i.e., root) of the hierarchy is interesting and informative. However, to identify optimization targets users have to be able to find and analyze the specific hardware resource that is causing the bottleneck in lower levels of the hierarchy.

III. Task Analysis

This section describes the three categories of tasks that software and hardware engineers must perform in order to optimize datacenter performance: summarization, detailed comparative analysis, and interactivity.

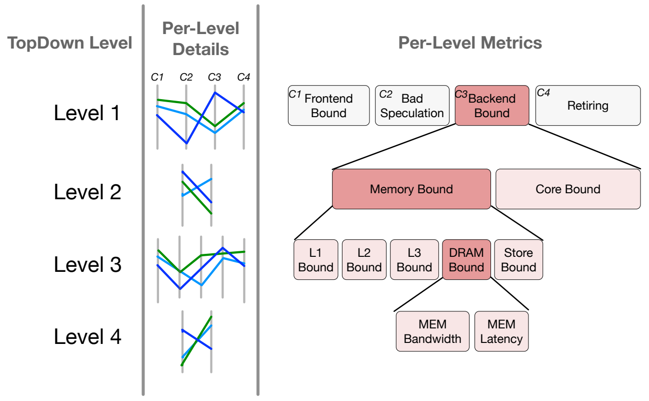


Fig. 1: **TopDown Summary:** TopDown microarchitectural view of computer systems showing their hierarchical structure. Here, we show details and lower-level data for the ‘backend bound’ stage. The parallel plots show performance details for three different applications for each hierarchy level.

Summarization tasks relate to observing and comparing similar and dissimilar applications or hardware platforms.

Task 1.1 - Performance Similarity Overview: How similarly do the applications or hardware platforms profiled perform?

Task 1.2 - Hierarchical Performance Overview: How does the similarity between applications or hardware platforms change across hierarchy levels?

Task 1.3 - Outlier Detection: Which applications or hardware platforms are outliers?

Detailed Comparative Analysis aims to find trends in performance bottlenecks among applications or hardware configurations.

Task 2.1 - Detailed Comparative Analysis: How similar is the trend of hardware resource usage among applications or hardware platforms?

Task 2.2 - Hierarchical Performance Analysis: How do trends vary across hierarchical levels?

Task 2.3 - Trend and Bottleneck Identification: Which features result in differences in performance bottlenecks between applications or hardware?

Interactive Performance Bottleneck Visualization enables comparison of multiple application simultaneously, which is key for datacenter-scale performance analysis.

Task 3.1 - Performance-based Clustering: How many clusters are needed to represent the applications or hardware platforms profiled?

Task 3.2 - Cluster Analysis: How do the performance optimization targets change for each cluster of applications or hardware?

Task 3.3 - Benchmark Comparison: How does the performance of users' applications or hardware compare to standard benchmarks?

IV. Related Work

In this section we analyze previous work based on the performance analysis goals described in Section III.

Microarchitectural Performance Analysis. TopDown [6] is a relatively new technique to view a system as a hierarchy of microarchitectural features. Previous performance visualization tools for high performance computing like Caliper [12] and the HPCToolkit [13] identify bottlenecks across software call stacks. TopDown and CHAMPVis, on the other hand, are hardware-centric. Note that hardware structure is static across applications. By exploiting this static structure, an engineer can quickly determine which portion of the program execution (e.g., computing vs. memory resources) is limiting the applications performance. An alternative abstraction is Hierarchical Cycle Accounting (HCA), as proposed by Nowak et al. [14]. However, HCA lacks an intuitive connection to system's top-level execution and detailed hardware features; this makes TopDown a particularly appealing approach.

The TopDown microarchitectural model has been effectively employed to determine possible bottlenecks in many critical settings, including datacenter-scale computing [4], [10], [15]. Furthermore, the TopDown model is used in VTune [16], a PC-based performance analysis tool developed by Intel. While engineers use Intel VTune to improve application efficiency on a given system, it only allows to view a single application's PCs. Thus, Intel VTune remains insufficient for datacenter-scale analysis as it precludes comparisons or analysis of trends across *multiple* systems or applications.

Performance Visualization. A key result of visualizing PCs is the ability to compare otherwise disaggregated and abstract data across different applications and systems. However, the high dimensionality and mixed data types of performance data along with its hierarchical structure creates challenges. Automatically selecting which performance counters should be visualized is infeasible because expert users typically need to analyze different PCs at different hierarchy levels to understand the data and diagnose problems at both system and detail levels. To that end, several tools have been developed that visualize such high-dimensional data in a format that allows dimensional analysis [17], [18]. However, they lack a structured understanding of performance bottlenecks from the hardware level.

Dynamic Interactivity. Tools like Intel's VTune [16] or the work from Koppelman and Michael [19] both use the TopDown microarchitectural model, and thereby remove the need to manually trace and find relevant paths in their analysis. Existing tools allow users to quickly understand the root cause of performance waste, but they currently only support static visualizations. As a result, the user must rely on their intuition to coarsely interpret the data. For instance, an engineer using VTune still has to make some heuristic-driven decisions rather than data-driven ones, because interactively analyzing a more detailed performance break-down is not supported. Furthermore, since the PC visualizations are static, the engineer needs to mentally connect and fuse information from different views, increasing the user's mental workload. This cognitive load prevents the user from engaging in the higher-order task of intuiting different scenarios and interpretations. Being able to improve and optimize on existing conditions requires a certain level of creativity that is only possible with real-time interactivity [20].

V. CHAMPVis

In this section, we describe the design and the features of CHAMPVis. A web-based version of CHAMPVIS with preloaded reference data is available online [1].

A. Application Configurations

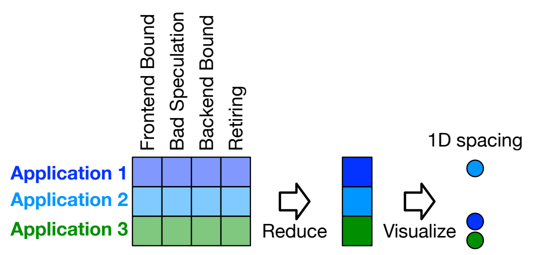
CHAMPVis is designed for expert datacenter engineers. When a datacenter engineer starts using CHAMPVis, the performance data of a default set of applications has been preloaded to quickly allow users to explore CHAMPVis and its features (see Figure 3). Users may also upload their own data (Linux *perf* utility output in CSV format [21]) and configure the set of applications they would like to view (**Task 3.3**). The ability to upload, curate, or reset the datasets being displayed is provided in a top header.

B. Performance Data Summarization View

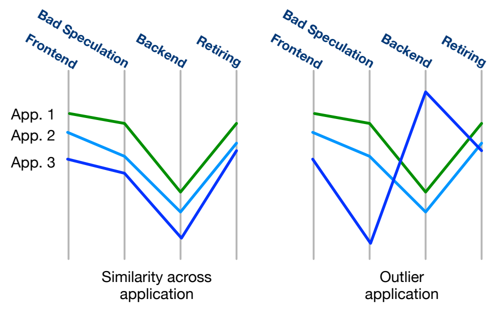
Datacenter engineers often have to determine whether a number of applications exhibit similar performance characteristics, for example to ensure that resource sharing would not overload a particular hardware component. CHAMPVis supports this task by offering a 1-D performance data summarization view (Figure 2, left). This view allows the user to see at a glance which applications are similar, and how similarity changes at different hierarchy levels (**Task 1.1**, **Task 1.2**). We use this data summarization view to guide the analysis steps of users without them having to first analyze multidimensional data at each level of the hierarchy. To create this view, we project multidimensional features of each application into a 1-D space, represented as a vertical line on the left of the CHAMPVis screen, using a single similarity metric. The similarity metric can be user-defined. This summarization (i.e., projection of all data points into a 1-D space) can be calculated for each hierarchy level, allowing users to easily identify outliers (**Task 1.3**).

C. Performance Comparison View

CHAMPVis represents the aggregate performance data for a given TopDown level as parallel coordinates. Each axis in the parallel coordinate plot corresponds to one performance metric of the current hierarchy level (see Figure 1). This supports a detailed comparative analysis of different applications or systems (**Task 2.1**). Users can navigate to lower levels of the hierarchy by clicking on desired axis labels (i.e., performance



(a) 1-D spacing



(b) Parallel coordinates

Fig. 2: **CHAMPVis Elements:** (**Left**) Summarize multi-dimensional data with 1-D spacing. (**Right**) Spotting trends and outliers, respectively, in multi-dimensional performance analysis.

metrics), with the ability to return to Level 1 metrics at any time (**Task 2.2**).

Figure 1 shows the TopDown hierarchy, with corresponding parallel plots for each hierarchy level. As users click through the hierarchy, the selected top-level metric remains as the first axis in the parallel plot as an indicator of the user's filtering choices. Each application is displayed as a line across the parallel coordinate plot. The shape of these lines allows the user to quickly identify trends for applications that are similar for specific ranges of performance metrics or for particular dimensions (**Task 2.3**), as illustrated in Figure 2, right.

D. *Detailed Bottleneck Identification View*

Datacenter engineers need to compare metrics across many different applications to identify optimization targets. To this end, CHAMPVis employs stacked bar charts to show performance details on selected applications at the current hierarchy level (**Task 3.2**). Individual segments in the stacked bar chart represent the performance metrics of the selected level (see Figure 3). This type of visualization is well-known in the systems community and makes CHAMPVis more accessible for domain experts. Furthermore, we allow users to select and cluster applications of interest and selectively view only those applications. This optional adaptive filtering step ensures that investigation can be customized and user-driven and that CHAMPVis scales to a larger number of applications.

E. *Navigation and Interaction*

Interacting with CHAMPVis is a dynamic process of hierarchical navigation, filtering, and clustering, which is supported by the context of the TopDown model. This allows the user to move from micro to macro analysis, explore information, and perform meaningful analysis.

The central view of CHAMPVis is the parallel coordinate representation of a particular performance metric for the current hierarchy level.

Hierachical Navigation. By clicking on an axis' title, users can navigate across levels of the hierarchy based on the TopDown model. This progressive navigation from hierarchy level to hierarchy level prevents the user from being overwhelmed with information and, thus, aides in identifying bottlenecks. As the user steps through each level for a more targeted exploration, the axis and visualization titles, stacked bar chart, and corresponding 1-D summarization view dynamically update.

Filtering. Users can also click and drag on any portion of a parallel axis to select the range of the metrics they are interested in, which automatically filters all other applications. By setting multiple ranges of interest across metrics, users can define detailed criterion of interest. Users can also filter by clicking and dragging a selection in the top-level 1-D summarization view.

Grouping and Clustering. Users can also extract groups or clusters of applications, for example, by clustering all applications with a similar performance metric at a particular hierarchy level (**Task 3.1**). Clustering dynamically updates the 1-D summarization view and the stacked bar chart. Similarly, the 1-D summarization view can be used to define clusters. We encode applications within a cluster with the same color.

F. *Implementation*

CHAMPVis is a web-based visualization tool based on JavaScript and D3.js [22]. Our online demo [1] shows the results of Intel's top-down microarchitectural profiling tools [21] for different applications running on server-class CPUs.

VI. Case Study

To demonstrate how CHAMPVis can be used to accomplish the tasks outlined in Section III, this section examines two example use cases using performance data collected from the Parsec suite of benchmarks [23] and machine learning (ML) workloads, which represent user-defined workloads requiring performance analysis (**Task 3.3**).

Comparing Parsec with user-uploaded applications. Figure 3 illustrates the suite of Parsec [23] benchmarks (purple) and common ML kernels (green) loaded into CHAMPVis.

A typical analysis session begins by looking at the 1-D summarization view of these two sets of applications (Figure 3) top, left). We first notice that the ML kernels are more similar to one another than the diverse Parsec benchmarks (**Task 1.1**). The parallel coordinate view for the top hierarchy level reveals that the majority of execution cycles for our ML applications of interest are spent in the retiring stage. Filtering

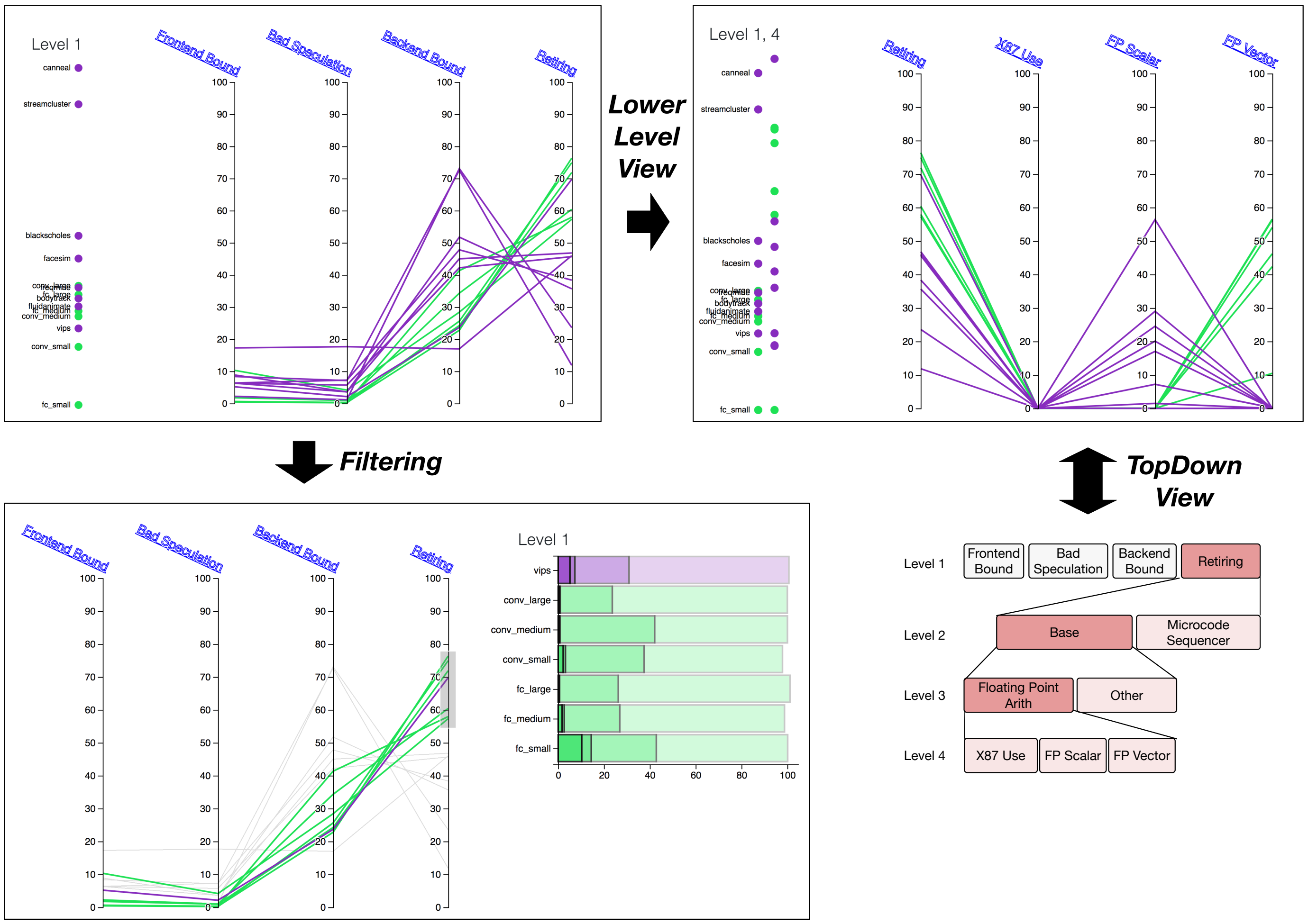


Fig. 3: **Benchmark Comparison:** (**Top, left**) Level 1 main CHAMPVis view without and (**Bottom, left**) with filtering to compare a subset of applications. (**Right**) Level 4 1-D spacing and parallel coordinates highlighting different lower-level performance characteristics between groups of applications when the TopDown hierarchy is navigated as shown below.

the Parsec benchmarks (Figure 3, bottom left) reveals that only vips performs similarly (**Task 2.1**). To identify optimization targets and extract more specific bottleneck information, we iteratively filter to the lowest level hardware components based on where a majority of cycles are spent (Figure 3, right). Using CHAMPVis we are able to make the following observations and takeaways:

- FC and Conv kernels of varying size generally spend more cycles in floating point vector unit (FP Vector) than Parsec workloads (**Task 2.2**).
- Vips (an image processing workload in Parsec) performs most similarly to the ML kernels overall, but does not spend a significant share of cycles in FP Vector (**Task 1.2**, **Task 1.3**).
- ML kernels would benefit from optimizing FP Vector (**Task 3.2**).
- The small FC kernel is disproportionately frontend-bound; contrary to intuition, frontend-related optimizations can benefit ML kernels. This also suggests that optimization strategies differ with neural network size (**Task 2.3**).

Understanding user-uploaded applications. In addition to the performance analysis for machine learning kernels already described, the expert user can also perform more detailed comparisons amongst their workloads of interest, as summarized in Figure 4. Therefore, we create and extract clusters based on neural network type (Conv layers in green and FC in red, **Task 3.1**), leading to the following observations:

- Kernel size tends to dictate similar performance characteristics more than kernel type. For example, the 1-D summary view at Level 1 shows that larger networks are more similar than equivalent smaller kernels (**Task 1.1**).
- Conv layers tend to be more backend bound than FC ones. This warrants further investigation for optimization (**Task 2.1**).
- Lower-level analysis reveals DRAM bottlenecks for Conv

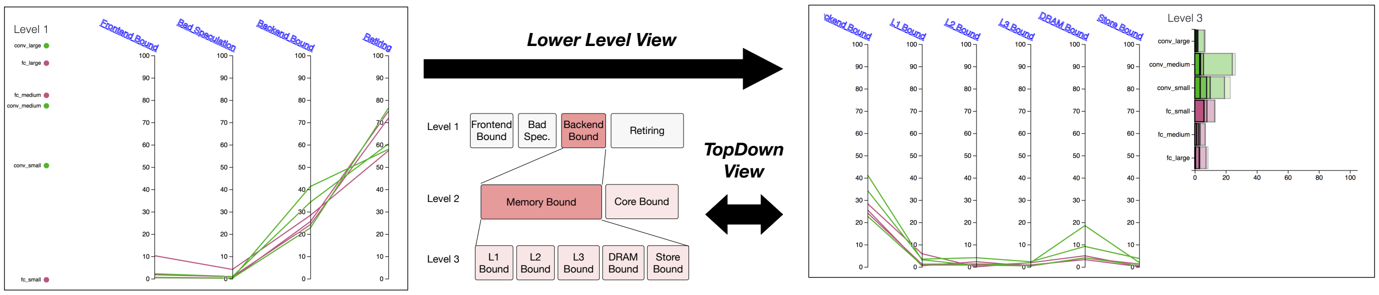


Fig. 4: **Understanding User-Uploaded Applications:** (**Left**) Level 1 main CHAMPVis view comparing among machine learning kernels. (**Right**) Navigating the TopDown hierarchy as highlighted to investigate discrepancies, Level 3 1-D spacing and parallel coordinates reveal behavior of different sets of applications.

layers compared to FC layers (**Task 3.2**).

The case study demonstrates that CHAMPVis allows the users to find unique bottlenecks to performance across the machine learning applications. With CHAMPVis, users are able to not only compare well-understood benchmarks with new applications but also gain deeper understanding of the application to identify new optimization targets.

VII. Conclusion

CHAMPVis is a web-based tool for comparative performance analysis empowered by the TopDown hierarchical framework for categorizing and interpreting performance counters. CHAMPVis enables efficient identification of performance bottlenecks through summarization and similarity analysis, detailed multi-dimensional comparisons, and dynamic user interaction with performance data. While the tool provides an opportunity for productive performance bottleneck analysis in complex datacenters, we hope the work encourages future work from the systems and visualization community to close the gap between detailed and productive datacenter-scale performance analysis. For example, we hope to profile additional applications (e.g., SPEC [24]). In addition, to ease usability, the interactions must be highlighted or clarified with a tutorial. Finally, it would be beneficial to explicitly visualize the connection with the TopDown hierarchy in order to guide the user. The existing visual features of CHAMPVis allow expert computer systems researchers to conduct effective TopDown-style performance comparison and identify optimization targets, and we hope to maintain and extend CHAMPVis in the future.

References

[1] "Champvis," 2019. https://lpentecost.github.io/uarchperfviz/.

[2] N. Jones, "How to stop data centres from gobbling up the worlds electricity," *Nature*, vol. 561, no. 7722, pp. 163–166, 2018.

[3] S. Kanev, J. P. Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks, "Profiling a warehouse-scale computer," *ACM SIGARCH Computer Architecture News*, vol. 43, no. 3, pp. 158–169, 2016.

[4] A. Sriraman, A. Dhanotia, and T. F. Wenisch, "Softsku: optimizing server architectures for microservice diversity@ scale," in *Proceedings of the 46th International Symposium on Computer Architecture*, pp. 513–526, ACM, 2019.

[5] S. Eyerman, L. Eeckhout, T. Karkhanis, and J. E. Smith, "A performance counter architecture for computing accurate cpi components," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 5, pp. 175–184, 2006.

[6] A. Yasin, "A top-down method for performance analysis and counters architecture," in *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 35–44, IEEE, 2014.

[7] J. Reinders, "Vtune performance analyzer essentials," *Intel Press*, 2005.

[8] A. C. De Melo, "The new linuxperftools," in *Slides from Linux Kongress*, vol. 18, 2010.

[9] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang, "Applied machine learning at facebook: A datacenter infrastructure perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 620–629, 2018.

[10] S. Kanev, J. P. Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks, "Profiling a warehouse-scale computer," in *International Symposium on Computer Architecture (ISCA)*, pp. 158–169, 2015.

[11] L. A. Barroso, U. Hölzle, and P. Ranganathan, *The datacenter as a computer: Designing warehouse-scale machines*. Morgan & Claypool Publishers, 3rd ed., 2018.

[12] D. Boehme, T. Gamblin, D. Beckingsale, P.-T. Bremer, A. Gimenez, M. LeGendre, O. Pearce, and M. Schulz, "Caliper: performance introspection for hpc software stacks," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 550–560, 2016.

[13] L. Adhianto, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. Tallent, "Hpctoolkit: Performance measurement and analysis for supercomputers with node-level parallelism," in *Workshop on Node Level Parallelism for Large Scale Supercomputers, in conjuction with Supercomputing '08*, 2008.

[14] A. Nowak, D. Levinthal, and W. Zwaenepoel, "Hierarchical cycle accounting: a new method for application performance tuning," in *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 112–123, March 2015.

[15] A. Yasin, Y. Ben-Asher, and A. Mendelson, "Deep-dive analysis of the data analytics workload in cloudsuite," in *2014 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 202–211, Oct 2014.

[16] Intel Corporation, "Intel vtune," 2018. https://software.intel.com/en-us/vtune.

[17] L. von Rüden, M.-A. Hermanns, M. Behrisch, D. Keim, B. Mohr, and F. Wolf, "Separating the wheat from the chaff: Identifying relevant and similar performance data with visual analytics," in *Workshop on Visual Performance Analysis (VPA)*, pp. 1–8, 2015.

[18] B. Weyers, C. Terboven, D. Schmidl, J. Herber, T. W. Kuhlen, M. S. Muller, and B. Hentschel, "Visualization of memory access behavior on hierarchical numa architectures," in *Workshop on Visual Performance Analysis*, pp. 42–49, 2014.

[19] D. M. Koppelman and C. J. Michael, "Discovering barriers to efficient execution, both obvious and subtle, using instruction-level visualization," in *Workshop on Visual Performance Analysis (VPA)*, pp. 36–41, 2014.

[20] B. Victor, "Inventing on principle," *Canadian University Software Engineering Conference*, 2012.

[21] Intel Corporation, "Pmu tools," 2019. https://github.com/andikleen/pmu-tools.

[22] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, vol. 17, no. 12, pp. 2301–2309, 2011.
[23] C. Bienia, *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, January 2011.
[24] J. L. Henning, "Spec cpu2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1–17, 2006.