Microarchitectural Analysis of Database Workloads

Anubhav Jana
Department of CSE,
IIT Bombay, Mumbai
Email: 22m2109@iitb.ac.in

Sunanda Somwase
Department of CSE,
IIT Bombay, Mumbai
Email: 22m2107@iitb.ac.in

Manish Kumar
Department of CSE,
IIT Bombay, Mumbai
Email: 22m2110@iitb.ac.in

***Abstract*—Microarchitectural intricacies significantly impact the performance of Database Management System (DBMS) applications, particularly in the context of compute and memory-intensive workloads. This paper, authored by Manish Kumar, Sunanda Somwase, and Anubhav Jana of Team ACA, presents a meticulous exploration of the microarchitectural aspects influencing DBMS performance. Leveraging tools such as HammerDB, VTune, and IntelPin, the study delves into data access patterns, cache utilization, and control flow structures within PostgreSQL, a widely used DBMS.**

The research methodology involves the simulation of real-world workloads, in-depth performance analysis, and trace generation to uncover optimization opportunities. Experiments, including TLB decoder setup variations and prefetching strategies, contribute to a comprehensive understanding of performance bottlenecks. The findings offer valuable insights into the intricate relationship between microarchitecture and DBMS efficiency, providing a foundation for future optimizations in this critical domain.

1. Introduction

This paper addresses this gap by presenting a thorough analysis of microarchitectural aspects impacting DBMS applications. Our approach involves the utilization of HammerDB for realistic workload simulation, VTune for in-depth performance analysis, and PostgreSQL as the focal point for microarchitectural exploration. Additionally, we employ IntelPin for trace generation, overcoming challenges associated with its integration with PostgreSQL.

The experiments conducted encompass variations in TLB decoder setup sizes and the exploration of prefetching strategies. Through these experiments, we aim to unravel the intricate relationships between microarchitectural components and DBMS performance, ultimately identifying optimization opportunities.

The outcomes of this study contribute not only to the understanding of microarchitectural nuances in DBMS applications but also provide a basis for future research and development in the pursuit of enhanced performance in compute and memory-intensive database workloads.

2. Motivation

In recent years, the demand for compute and memory-intensive database applications has surged, necessitating a deeper exploration of performance optimization strategies. Despite the critical role of microarchitecture in influencing Database Management System (DBMS) performance, there exists a notable gap in research, prompting our investigation. [1] [2]

3. Contribution

In this paper, the authors contribute the following:

- Framework explored and set up a using independent existing tools for generating Champsim traces database applications.
- Champsim simulator traces generated for PostgreSQL database application for two database query benchmarks TPC-C and TPC-H.
- Performed micro-architectural and hotspot analysis using Intel's VTune.
- Experiments performed for microarchitectural optimizations using the Champsim simulator.

4. System Overview: Trace Generator

The system used to generate the traces is as follows: First, we have a physical machine on which Ubuntu VM is set up. Inside the Ubuntu VM, there is another VM

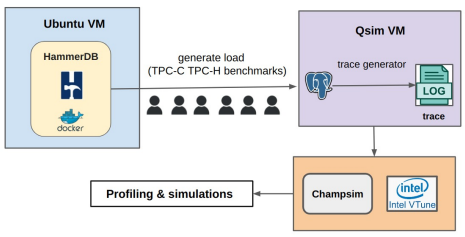

Figure 1. Trace Generator

called qsim. The load generator is set up inside the docker container which is running on an Ubuntu VM with the

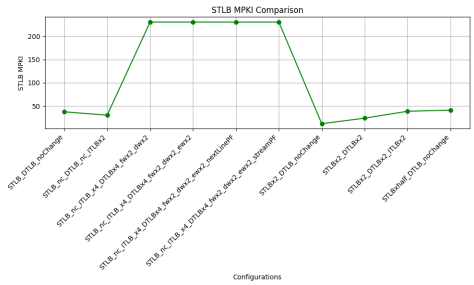

Figure 2. STLB-MPKI - TPCC

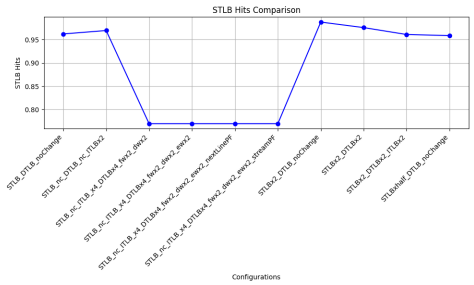

Figure 3. STLBhits - TPCC

network same as the host. Inside the Qsim VM, PostgreSQL is set up using the source code [3] with a well-defined TCP port exposed to the outside VM. The load generator runs on another port mapped to the PostgreSQL port. Once the load generation has started the trace file generator enable function is executed which in return generates a trace.log file. The generated trace.log file is preprocessed separately to create the pgtrace.champsim.gz file.

5. Profiling Database Application using VTune

Profiling a PostgreSQL database using Intel VTune [4] Profiler provided valuable insights into the performance characteristics of the PostgreSQL database system, helping to identify bottlenecks and optimization potentials. VTune Profiler is a powerful performance profiling tool that allows us to analyze the behavior of your PostgreSQL database at the code level, providing detailed information about microarchitecture level usage of the application. Although VTune approximates and predicts the values of many parameters, those are just indications of the possible areas of optimizations. After performing a micro-architectural analysis the relevant experiments on architectural setup are performed using the Champsim simulator. Details of experiments are discussed in section 6

6. Experiments and Results

For the experimental setup following are the details of our baseline architecture.

- L1I : sets=64, ways=8
- L1D : sets=64, ways=12
- L2C : sets=1024,ways=8
- ITLB: sets=16, ways=4
- DTLB: sets=16, ways=4
- STLB: sets=128, ways=12
- Fetch Width: 6
- Decode Width: 6

Various experiments are conducted with the TLB sizes, fetch width, decode width, and data prefetchers at the L2C cache, keeping the size of L1I, L1D, L2C cache same as baseline.

The rest of the results for TPC-C and TPC-H benchmarks are uploaded on the GitHub repo.

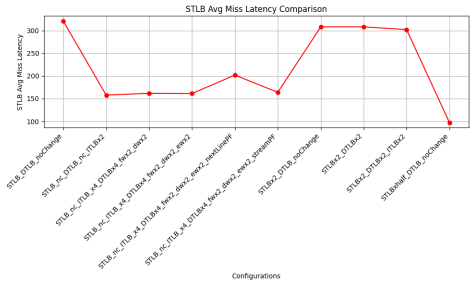

Figure 4. STLBAvg-Miss-Latency - TPCC

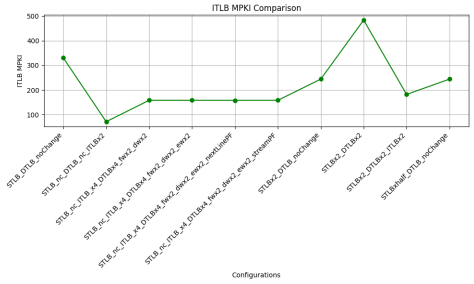

Figure 5. ITLB-MPKI - TPCC

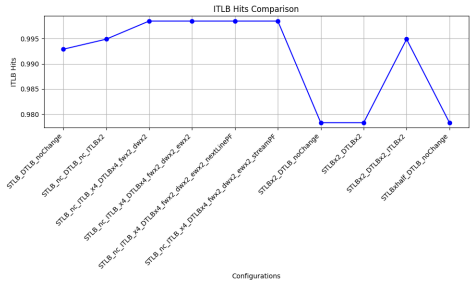

Figure 6. ITLBhits - TPCC

7. Conclusion

In conclusion, our paper introduces a framework for PostgreSQL database applications, utilizing independent tools to generate Champsim simulator traces for TPC-C and TPC-H benchmarks. Leveraging Intel's VTune Profiler, we conducted micro-architectural and hotspot analyses,

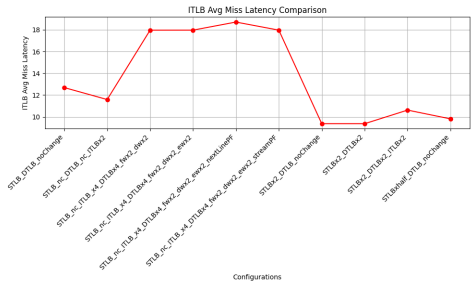

Figure 7. ITLBAvg-Miss-Latency - TPCC

pinpointing optimization opportunities. Through Champsim simulations, we explored and validated microarchitectural optimizations, demonstrating the practical impact on PostgreSQL performance. This work provides a comprehensive approach to database optimization, offering insights for researchers and practitioners in enhancing the efficiency of database applications.

Acknowledgments

We would like to thank Mr. Hari Sharan, PhD student at CSE IIT Bombay for providing us with the trace generator tool qsim. We would also like to thank Prof. Biswabandan Panda for guiding us throughout the project.

References

[1] U. Sirin and A. Ailamaki, "Micro-architectural analysis of olap: Limitations and opportunities," *Proc. VLDB Endow.*, vol. 13, no. 6, p. 840–853, feb 2020. [Online]. Available: https://doi.org/10.14778/3380750.3380755

[2] H. Caminal, Y. Chronis, T. Wu, J. M. Patel, and J. F. Martínez, "Accelerating database analytic query workloads using an associative processor," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ser. ISCA '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 623–637. [Online]. Available: https://doi.org/10.1145/3470496.3527435

[3] postgres. postgres. [Online]. Available: https://github.com/postgres/postgres

[4] Intel. intel. [Online]. Available: https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler-download.html