A Comprehensive Empirical Study of Query Performance Across GPU DBMSes

Young-Kyoon Suh
yksuh@knu.ac.kr
Kyungpook National University
Daegu, Republic of Korea

Junyoung An
jyan@knu.ac.kr
Kyungpook National University
Daegu, Republic of Korea

Byungchul Tak
bctak@knu.ac.kr
Kyungpook National University
Daegu, Republic of Korea

Gap-Joo Na
funkygap@etri.re.kr
ETRI
Daejeon, Republic of Korea

ABSTRACT

In recent years, GPU database management systems (DBMSes) have rapidly become popular largely due to their remarkable acceleration capability obtained through extreme parallelism in query evaluations. However, there has been relatively little study on the characteristics of these GPU DBMSes for a better understanding of their query performance in various contexts. To fill this gap, we have conducted a rigorous empirical study to identify such factors and to propose a structural causal model, including key factors and their relationships, to explicate the variances of the query execution times on the GPU DBMSes. To test the model, we have designed and run comprehensive experiments and conducted in-depth statistical analyses on the obtained data. As a result, our model achieves about 77% amount of variance explained on the query time and indicates that reducing kernel time and data transfer time are the key factors to improve the query time. Also, our results show that the studied systems still need to resolve several concerns such as bounded processing within GPU memory, lack of rich query evaluation operators, limited scalability, and GPU under-utilization.

CCS CONCEPTS

• **Information systems → DBMS engine architectures**; • **Computing methodologies → Massively parallel algorithms**; **Model verification and validation**.

KEYWORDS

GPU DBMS, query time, causal model, performance evaluation

ACM Reference Format:
Young-Kyoon Suh, Junyoung An, Byungchul Tak, and Gap-Joo Na. 2022. A Comprehensive Empirical Study of Query Performance Across GPU DBMSes. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE '22 Abstracts), June 6–10, 2022, Mumbai, India.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3489048.3522644

SIGMETRICS/PERFORMANCE '22 Abstracts, June 6–10, 2022, Mumbai, India
ACM ISBN 978-1-4503-9141-2/22/06.
https://doi.org/10.1145/3489048.3522644

The Full Paper appears at https://dl.acm.org/doi/10.1145/3508024.

1 EXTENDED ABSTRACT

In recent years, the General Purpose computing on Graphics Processing Units (GPGPU) has received heightened attention from the data management community, thanks to its massive parallelism and high bandwidth [14]. Accordingly, the community has also been making strenuous efforts to accelerate the analytical query processing capability by jointly harnessing the power of CPU and GPU. We are starting to reap the benefits of these efforts. Several market players have productized their respective GPU-accelerated database management systems (DBMSes) [2, 9, 12, 15, 20]. Others in academia have developed their prototype engines [3, 5, 13, 16, 17] enabling query executions under co-processing with GPU.

In this context, several research questions arise in terms of how much we understand the performance characteristics of the state-of-the-art GPU DBMSes:

- *What factors affect the query evaluation on the real GPU DBMSes and how do the factors interact?*
- *Does the GPU DBMSes benefit from employing an increasingly advanced GPU model for enhancing the query performance? Are there any limitations to the GPU DBMSes in supporting a wide spectrum of queries?*
- *Is the increase of RAM or the use of multiple GPUs effective in enhancing the performance of the GPU DBMS?*
- *Can DBMSes scale well with the growing volume of data?*
- *How much do the DBMSes exploit the GPUs during their query execution? Does the under-utilization of GPU still persist in the modern GPU DBMSes?*

While seeking for answers to these questions, we found that the performance implications of the latest GPU-based query engines have not yet been well-studied. There has been little research done to systematically explore and identify the performance bottlenecks *across* the engines.

On one hand, performing database operations on GPUs is quite different from and as challenging as executing other types of computations—scientific computing and machine learning—on GPUs. First, the former concentrates more on the 'performance' (or 'efficiency') in evaluating a query plan while the latter focuses more on the 'accuracy' in solving a complex equation and training a model. In addition, for a given query, the utilization of GPUs in a DBMS is *dynamically* determined, as query evaluation operators leveraging GPUs may or may not be included in the execution plan. Moreover, if the size of intermediate results produced by operations such as join, sort, and/or aggregation on GPUs is overly large, the DBMS will suffer from substantial performance degradation in overall query processing. For these reasons, the use of GPUs in the databases may not always lead to obvious benefits, and thus, it is of critical necessity to better understand the query evaluation performance on GPU DBMSes in various contexts.

To this end, this paper takes an empirical evaluation approach [1, 4, 6, 10, 11, 21] for better understanding the query processing performance of multiple, modern GPU DBMSes (BlazingSQL [2], OmniSciDB[1] [12], and PG-Strom [15]) as a general class but not as a specific system. More specifically, we identify key factors that can potentially impact the query processing and propose a structural causal model using these factors and their associations, to explicate the variances of query execution time in GPU DBMSes. In turn, we present a set of hypotheses regarding the correlations of the factors from the model. To test the model, we design and run comprehensive experiments and conduct statistical analyses based on the empirical data. As a result, we find that our proposed model is supported by the empirical data and can explain about 77% of the variances of the query time on GPU DBMSes. Most hypothesized correlations predicted by the model are statistically significant. In particular, the strengths of some correlations are at high levels—0.7 or more. Also, we demonstrate that the GPU DBMSes expose several critical bottlenecks, such as lack of query operators, limited scalability, and under-utilization of GPU. From our analysis results, we draw several performance implications as a guide for further engineering of the GPU DBMSes. To the best of our knowledge, this paper is the *first* work to propose and test the causal model through comprehensive in-depth analyses on the query performance across 'multiple' modern GPU DBMSes. Similar modeling techniques [7, 8, 18, 19] have been previously presented, but they have not been applied to the GPU DBMSes of our interest in this paper. We emphasize that our study can help researchers and engineers gain new insights into performance characteristics of the GPU-based query engines and mitigate any potential performance degradation. Also, our methodology can be applied to conventional CPU-based DBMSes (i.e., non-GPU based).

The paper's contributions are:

- We explore and identify key factors that impact the query performance of GPU DBMSes based on the widely-held assumptions and existing papers.
- We propose a structural causal model, which includes the factors and their relationships, to explicate the variance of query evaluation time on a GPU DBMS and present a set of hypotheses drawn from the model.
- We introduce the detailed operationalizations of the factors and conduct comprehensive experiments across three disparate GPU DBMSes.
- We test the model through rigorous statistical analyses on the obtained empirical data.
- We draw several performance implications that are useful for further engineering of the GPU DBMSes.
- We review relevant literature and suggest promising research directions to expand our work in future.

ACKNOWLEDGMENTS

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [22ZS1300, Research on High Performance Computing Technology to overcome limitations of AI processing].

REFERENCES

[1] Richard Bieringa, Abijith Radhakrishnan, Tavneet Singh, Sophie Vos, Jesse Donkervliet, and Alexandru Iosup. 2021. An Empirical Evaluation of the Performance of Video Conferencing Systems. In *Companion of the ACM/SPEC International Conference on Performance Engineering*. 65–71.

[2] BlazingSQL, Inc. 2021. BlazingSQL - The Official Homepage. URL: https://blazingsql.com/.

[3] Sebastian Breß. 2014. The Design and Implementation of CoGaDB: A Column-oriented GPU-accelerated DBMS. *Datenbank-Spektrum* 14, 3 (2014), 199–209.

[4] Zhifeng Chen, Yan Zhang, Yuanyuan Zhou, Heidi Scott, and Berni Schiefer. 2005. Empirical Evaluation of Multi-level Buffer Cache Collaboration for Storage Systems. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. 145–156.

[5] Periklis Chrysogelos, Panagiotis Sioulas, and Anastasia Ailamaki. 2019. Hardware-conscious Query Processing in GPU-accelerated Analytical Engines. In *Proceesings of the 9th Biennial Conference on Innovative Data Systems Research*. www.cidrdb.org.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555* (2014).

[7] Sabah Currim, Richard T. Snodgrass, Young-Kyoon Suh, and Rui Zhang. 2016. DBMS Metrology: Measuring Query Time. *ACM Transactions on Database Systems* 42, 1, Article 3 (2016), 42 pages.

[8] Moises Goldszmidt and Rebecca Isaacs. 2011. More Intervention Now!. In *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*. USENIX Association, 25.

[9] Kinetica DB Inc. 2021. Kinetica High Performance Analytics Database. URL: https://www.kinetica.com/.

[10] Stefan Manegold. 2008. An Empirical Evaluation of XQuery Processors. *Information Systems* 33, 2 (2008), 203–220.

[11] Michele Mazzucco and Isi Mitrani. 2012. Empirical Evaluation of Power Saving Policies for Data Centers. *ACM SIGMETRICS Performance Evaluation Review* 40, 3 (2012), 18–22.

[12] OmniSci, Inc. 2021. OmniSciDB - The Official Website. URL: https://omnisci.com/platform/omniscidb.

[13] Johns Paul, Jiong He, and Bingsheng He. 2016. GPL: A GPU-based Pipelined Query Processing Engine. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*. 1935–1950.

[14] Johns Paul, Shengliang Lu, and Bingsheng He. 2021. *Foundations and Trends®in Databases* 11, 1 (2021), 1–108.

[15] PG-Strom Development Team. 2021. PG-Strom Manual - Home. URL: https://heterodb.github.io/pg-strom/.

[16] Holger Pirk, Oscar Moll, Matei Zaharia, and Sam Madden. 2016. Voodoo-A Vector Algebra for Portable Database Performance on Modern Hardware. *Proceedings of the VLDB Endowment* 9, 14 (2016), 1707–1718.

[17] Syed Mohammad Aunn Raza, Periklis Chrysogelos, Panagiotis Sioulas, Vladimir Indjic, Angelos Christos Anadiotis, and Anastasia Ailamaki. 2020. GPU-accelerated Data Management under the Test of Time. In *Proceesings of the 10th Conference on Innovative Data Systems Research*. www.cidrdb.org.

[18] Raja R. Sambasivan, Ilari Shafer, Jonathan Mace, Benjamin H. Sigelman, Rodrigo Fonseca, and Gregory R. Ganger. 2016. Principled Workflow-Centric Tracing of Distributed Systems. In *Proceedings of the Seventh ACM Symposium on Cloud Computing*. ACM, 401–414.

[19] Richard T. Snodgrass, Sabah Currim, and Young-Kyoon Suh. 2021. Have Query Optimizers Hit the Wall. *The VLDB Journal* (2021), 1–20. https://doi.org/10.1007/s00778-021-00689-y

[20] SQream Technologies. 2021. SQream - The Official Website. URL: https://sqream.com/.

[21] Yingjun Wu, Joy Arulraj, Jiexi Lin, Ran Xian, and Andrew Pavlo. 2017. An Empirical Evaluation of In-memory Multi-version Concurrency Control. *Proceedings of the VLDB Endowment* 10, 7 (2017), 781–792.

[1] Changed to HeavyDB as of February 2022