Data Management Systems on Future Hardware: Challenges and Opportunities

Bingsheng He
National University of Singapore

Hardware development is a major driver for the development of data management systems. For example, due to the increased capacity and low cost of main memory, data management systems have shifted from disk-based to in-memory systems [1]. Recently, databases on emerging hardware have gained a lot of attractions in both academia and industry. A significant amount of research has been devoted to the design and implementation of high-performance hardware-conscious systems. We can summarize them into three main lines of research, mainly according to processor architectures:

- *General CPU architectures.* Researchers have studied a wide range of aspects in CPU-based systems, including data storage [2], logging [3], concurrency control [4] and NUMA effect in multi-socket systems [5].
- *Commodity accelerators.* This line of research goes beyond the CPU, and attempts to leverage commodity accelerators including GPUs [6] and FPGAs [7], [8]. Compared with CPUs, these accelerators have many advantages including massive parallelism, high memory bandwidth and/or high energy efficiency. This trend is evident from the fact that more and more cloud providers such as Amazon, Microsoft and Alicloud have already deployed accelerators in their data centers.
- *Specialized hardware.* Recently, there have been a few proposals and implementations on specialized hardware solutions for databases. Oracle has adopted "SQL in Silicon" to integrate data de-compression pipelines into the processor. Q100 [9] is another example of implementing databases using specialized hardware architectures.

Orthogonal to processor architectures, systems on emerging interconnects and memory architectures have also been studied, such as RDMA networking for distributed systems [10] and non-volatile RAM (NVRAM) [11].

In this talk, we will summarize the recent efforts in the above three lines of research and systems on other related interconnects and memory architectures. We have clearly seen from the studies that "one size does not fit all". Each software and hardware solution has its own pros and cons. For each line, there are still many challenges and opportunities on the research of data management systems on future hardware. Particularly, we will discuss the following challenges and outline the future research agenda.

Challenge 1: Even though hardware becomes very powerful in recent years, it is still just one side of the coin, and software is the other side. Careful design of data management systems is important in order to fully utilize the hardware.

Challenge 2: As data management systems and future hardware become more complex, it causes a large system design and exploration space. Thus, the gap between a research prototype and real-production systems may widen, as many other factors besides performance are also important in production environments, such as total ownership cost, cooling and space.

Challenge 3: Data management systems are already "monsters", in number of lines of source code. Many data processing systems like Apache Hadoop and Spark already have millions of lines of source code. How can these systems take advantage of future hardware without a total rewrite?

Challenge 4: Generalization vs. Specialization. The comparison among general and specialized solutions is still inconclusive in terms of performance, flexibility and energy efficiency etc., and more efforts are needed to help the community to make a wise choice.

Challenge 5: Many (supposedly great) hardware architectures did not survive in the history. How to predict the success of future hardware?

In summary, data management systems on future hardware continue to be a challenging and exciting research area. Hardware and software co-design will invoke a lot of interesting research on data management systems.

ACKNOWLEDGEMENT

This work is partially funded by a MoE AcRF Tier 1 grant (T1 251RES1610) and a startup grant of NUS in Singapore.

REFERENCES

[1] K.-L. Tan and et al., "In-memory databases: Challenges and opportunities from software and hardware perspectives," *SIGMOD Rec.*, vol. 44, no. 2, 2015.
[2] Y. Li and J. M. Patel, "Bitweaving: Fast scans for main memory data processing," in *SIGMOD*, 2013.
[3] R. Johnson and et al., "Aether: A scalable approach to logging," *PVLDB*, 2010.
[4] X. Yu and et al., "Staring into the abyss: An evaluation of concurrency control with one thousand cores," *PVLDB*, 2014.
[5] S. Zhang and et al., "Revisiting the design of data stream processing systems on multi-core processors," in *ICDE*, 2017.
[6] J. Paul and et al., "GPL: A GPU-based pipelined query processing engine," in *SIGMOD*, 2016.
[7] Z. Wang and et al., "A study of data partitioning on OpenCL-based FPGAs," in *FPL*, 2015.
[8] L. Woods and et al., "Ibex: An intelligent storage engine with support for advanced sql offloading," *PVLDB*, 2014.
[9] L. Wu and et al., "Q100: The architecture and design of a database processing unit," in *ASPLOS*, 2014.
[10] X. Wei and et al., "Fast in-memory transaction processing using RDMA and HTM," in *SOSP*, 2015.
[11] S. Pelley and et al., "Storage management in the NVRAM era," *PVLDB*, 2013.