How will I get this strange motor to fit in my car?

Reflections on what modern computing hardware means for DBMS architecture

Eyal Rozenberg*, DB Architecture Group, CWI

Background & Motivation Our DBMSes run on ever-evolving computers; to process queries more quickly, we need to utilize the computational resources the computing hardware manufacturers put at our disposal. Most of us, for most of the time, take our existing software to new hardware, and expect it to run more of the same, just faster. But when new hardware offers new features; when you switch architectures; or even when the balance between existing capabilities has gradually shifted enough — we have to change our approach. A now 'classical' example is described in the CWI DA group's "memory bottleneck" paper [2]: Sweeping architecture and algebra operator implementation changes proposed, due to a change in just one hardware feature: CPU speed increasing without memory latency decreasing as much.

These days (and for a while now already), there is a great wealth of untapped innovation on the hardware front that has not yet been brought to bear on (analytic) DBMSes: New computational devices, designed for massive parallelism, such as general-purpose GPUs (which I focus on), Xeon Phi's, FPGAs... as well as computational capability becoming available in strange places like disk controllers [4]. CPUs are also evolving differently than in the past: The age of frequency scaling is over, and we're seeing more SIMD and multi-core capabilities, which main memory can't even feed with data.

Research work on many of these subjects is ever-present, of course, in conferences, journals and workshops (especially DaMon and ADMS). However, most DBMSes we work with are (probably) still assume, deep down, multi-threaded serial execution, trusting the compiler and the OS to do well enough on that.

Also, published papers are usually "this thing works for us" kind of statement; it's much less common to find "here are a bunch of problems we're coming up against" and certainly not "we tried X, it doesn't work, need another approach". We also often tend to make mitigating assumptions about the execution scenario, the original DBMS involved, the data et cetera.

Session type This will session will not be about presenting results (my own or others'). It will have a default structure of a presentation followed by questions from the audience; but the presentation itself will consist of some introductory material about hardware and its capabilities, and the raising of some questions, problems and challenges relating to its use (with a weak focus on my own experience). Attendants are encouraged to interrupt the narrative flow with their own insights and comments. Getting side-tracked by a lively discussion will probably mean the session has been a success.

About me I've joined the group at CWI after a two-year stint at Hua Wei research, where we worked on a proof-of-concept query processor using GPUs for accelerating analytic query processing. While Hua Wei's interests shifted and our group disperesed, a swan song paper describing our work [1], related to this proposed session was presented in ADMS 2016 workshop. It is also one of nine or so different systems presented in recent years intended to demonstrate how GPUs are useful for this kind of acceleration, and the jury is still out on which aspects of their different approaches should be adopted; see [3] for a semi-recent survey. In my post-doctoral research I'm working on GPU compression and hoping to prototype a FOSS 'plan execution engine' in a hetergeneous computing system, realizing some of the ideas which this session will bring up.

[1] Adnan Agbaria et al. "Overtaking CPU DBMSes with a GPU in Whole-Query Analytic Processing". In: *International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures - ADMS*. 2016.

[2] Peter A Boncz, Stefan Manegold, and Martin L Kersten. "Database architecture optimized for the new bottleneck: Memory access". In: *VLDB*. Vol. 99. 1999, pp. 54–65.

[3] Sebastian Breß et al. "GPU-accelerated Database Systems: Survey and Open Challenges". In: *The Third ASE International Conference on Big Data Science and Computing, Beijing, China*. ACM/IEEE. 2014.

[4] Insoon Jo et al. "YourSQL: a high-performance database system leveraging in-storage computing". In: *Proceedings of the VLDB Endowment* 9.12 (2016), pp. 924–935.

*E.Rozenberg@cwi.nl