TIM Lecture Series

Huge Memory and Collection-Oriented Programming: Less Code, More Speed?

Dave Thomas

“ *I'm always trying to build more software faster by writing less code using fewer people.* ”

Dave Thomas
Chief Scientist/CSO, First Derivatives FD Labs

Overview

The TIM Lecture Series is offered by the Technology Innovation Management (TIM; timprogram.ca) program at Carleton University in Ottawa, Canada. The lectures provide a forum to promote the transfer of knowledge between university research to technology company executives and entrepreneurs as well as research and development personnel. Readers are encouraged to share related insights or provide feedback on the presentation or the TIM Lecture Series, including recommendations of future speakers.

The second TIM lecture of 2016 was held at Carleton University on March 8th and was presented by Dave Thomas, Chief Scientist/CSO, First Derivatives FD Labs (firstderivatives.com). The lecture focused on the disruptive aspects of "huge persistent memory", in terms of the technology shift it represents, the impact it has on how developers write software programs, and the corresponding business opportunities it brings about.

Summary

In this lecture, Thomas emphasized that the enormity of datasets used with "Big Data" demand lighter, query-based programs that allocate as much available memory as possible to the data, rather than the overhead of overly complicated programs.

He began by describing how developers over the years must face or least push back, the "memory wall", which is the limit of overall computer speed imposed by the limits the speed of memory. And, opportunities may arise in the quest to overcome this memory wall, particularly when considering the overall cost of an application, which consists of the costs of hardware and software plus the costs of management. Current downward trends in the cost of hardware mean that investing in hardware is an efficient way to bring down costs.

Recent leaps in the amount of memory that can be put on, for example, an DIMM card or gum stick (e.g., 3.5 TB), have reinforced the notion that storage and memory hierarchy must continue to scale. But, NAND/SSS does have its problems, including the required complexity of the software that runs on it, the slow writing speed relative to the fast reading speed, the loss of data on large writes if there is a power failure, the failure of the NAND memory after many writes, and the security vulnerability associated with need for large persistent storage without encryption.

Announced in August 2015, Intel's Micro 3D XPoint memory is 1000 times faster than NAND, has 1000 times more endurance than NAND, and is 10 times denser than conventional memory (Intel, 2015). However, with the greater memory performance in memory technology, there is a need for built-in data protection features to enable enhanced data security. Thomas explained that Intel non-volatile memory has enhanced data security with:

- *power loss data protection:* made so you can turn off power and writes will still complete; prevents data loss during unexpected system power loss while writing data (completes all writes in progress, even during power failure)
- *a surplus array of NAND:* surplus array of NAND Flash on SSDs furthers drive reliability; provides system protection against individual NAND die failure

- *encryption:* 328-Bit AES encryption when used with an ATA drive password; provides an additional layer of security

In step with the advances in memory technology, the Storage Networking Industry Association (SNIA; snia.org) have highlighted a new programming model for non-volatile memory, the NVM Programming Model (tinyurl.com/jpyya3z). This new model takes advantage of memory mapped files to communicate directly with persistent memory instead of using the traditional model, which relies on file systems and disks as intermediary technology. This new, and much simpler, programming model and its associated standard represent a breakthrough in terms of performance and the potential for more interesting applications than were previously possible.

Thomas' own research with the new Intel SSD DC P3700 memory technology found impressive and encouraging performance results against the STAC benchmark, which is used in the financial industry.

The impact on software

The advances in hardware and new programming models have impacts on software, and in particular on the feasibility of using object-oriented techniques. For example, automatic memory management (or "garbage collection") cannot cope in contexts with truly large amounts of data. Unfortunately, current languages, and most current developers, are not yet able to adjust to this new context.

Thomas outlined the differences (Table 1) between the conventional approaches to online transaction processing (OLTP) and a new OLTP approach using Hstore and Estore high-performance SQL database technologies, which offer substantial performance advantages by specifying complete workloads (i.e., collections of transaction classes) in advance (Stonebraker et al., 2007). This new approach addresses the common mismatch between databases and hardware, meaning that many conventional OLTP techniques are not appropriate for use with modern hardware.

Collection-oriented programming

Next, Thomas traced the evolution of our concept of "programs" up today and into the future, when collection-oriented programming will become prominent:

- *1960s and 70s:* Programs = Data Structures + Procedures
- *1970s and 80s:* Programs = Database + SQL
- *1980s:* Programs = Logic + Control
- *1990s:* Programs = Objects + Methods
- *2012:* Programs = Functional Programming - Data Structures + Functions
- *2017:* Programs = Collections + Queries

The drive toward collection-oriented programming is driven by the increasing complexity of software and the need for a simpler approach, particularly when working with Big Data. Applications become small function scripts of collections and queries, making the approach easy to use and accessible to most programmers, who can then write smaller, simpler, and faster programs that are easier to maintain and run. The approach includes:

- Tables, Dictionaries, and Lists
- Operations and Functions for all collections
- Simple value semantics (no pointers)

Table 1. The conventional versus new approach to online transaction processing (OLTP) (Stonebraker et al., 2007)

Disadvantages of Conventional OLTP	Advantages of New OLTP
• Disk-oriented storage >> memory	• Column store
• And indexing structure (B trees)	• Memory based
• Buffer pools to reduce latency	• One-shot single-threaded transactions
• Multithreading to hide latency	• No knobs
• Locking-based concurrency control mechanisms	• Replication instead of logs
• Log-based recovery	• Anti-caching for massive data
• Hardware cache unaware	

- Tables attributes are columns in a column store and can have trillions of rows

- Select, Update, Upsert, Delete for tables with functions in any position; implicit join, group by make it easier than SQL

- Each f(Map), f/ (Reduce), f\scan

As a result, most of the memory is devoted to the data rather than the programs, resulting in high performance with large datasets; however, the remaining challenge is to improve our ability to think in terms of formulating effective queries. Additional tools can be added to make queries even easier, for example: i) faster ETL (extract, transform, and load) without programming, ii) Visual Query simplifies Big Data querying but enables full power, and iii) a Big Data spreadsheet for non-linear analysis. Also, visual data exploration allows iterative, real-time visualization and pattern detection within massive datasets.

Thomas concluded the lecture with four key takeaways that summarize the near-future technologies and programming approaches:

1. Think more, and write less code.

2. Programs now consist of collections and queries.

3. Leverage the hardware: it is fast and inexpensive.

4. Simplicity reduces the time and cost of development and often improves performance.

About the Speaker

Dave Thomas is Chief Scientist/CSO, First Derivatives FD Labs. He is also Founder and Chairman of the YOW! Australia and Lambda Jam conferences, he is a GOTO Conference Fellow, and he is an ACM Distinguished Engineer. With a unique ability to see the future and translate research into competitive products, he is known for his contributions to object technology including IBM VisualAge and Eclipse IDEs, Smalltalk, and Java virtual machines, and more recently, he has been a proponent for the use of applied functional programming. He holds close links to the R&D community as an Adjunct Research Professor at Carleton University in Canada, and he has held past positions at UQ, QUT, and NICTA in Australia. While a professor at Carleton, he formed the Object-Oriented Research Group and established Ottawa's leadership in object-oriented technology. Dave has been a business and technical advisor to many technology local and international technology companies. And, among his past roles, he was Co-Founder and Chairman of Bedarra Research Labs (BRL), Founder and CEO of Object Technology International (OTI), becoming CEO of IBM OTI Labs after its sale to IBM.

This report was written by Chris McPhee.

References

Intel. 2015. 3D XPoint Unveiled – The Next Breakthrough in Memory Technology. *Intel.* Accessed March 15, 2016: http://www.intel.com/content/www/us/en/architecture-and-technology/3d-xpoint-unveiled-video.html

Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N., & Helland, P. 2007. The End of an Architectural Era (It's Time for a Complete Rewrite). In *Proceedings of the 33rd International Conference on Very Large Databases (VLDB '07):* 1150–1160.

Citation: Thomas, D. 2016. TIM Lecture Series – Huge Memory and Collection-Oriented Programming: Less Code, More Speed?. *Technology Innovation Management Review*, 6(3): 41–43. http://timreview.ca/article/974

Keywords: programming, databases, memory, speed, object-oriented programming, collection-oriented programming, queries, huge persistent memory, very large databases, Big Data