Joy Arulraj Speaks Out on Non-Volatile Memory Database Systems

Marianne Winslett and Vanessa Braganholo

Joy Arulraj

https://www.cc.gatech.edu/~jarulraj/

Welcome to this installment of the ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today I have here with me Joy Arulraj, who won the 2019 ACM SIGMOD Jim Gray Dissertation Award for his thesis entitled The Design and Implementation of Non-volatile Memory Database Management Systems. Joy is now an Assistant Professor at Georgia Tech, and his PhD is from the Carnegie Mellon University, where he worked with Andy Pavlo, who won this same award in his time. So, Joy, welcome!

Thanks for the kind introduction, Marianne. Thanks for having me here.

What was your dissertation about?

My thesis focuses on the implications of a new class of memory technologies known as non-volatile memory or persistent memory for database systems. In particular, I focused on improving the run-time performance, availability, and operational cost associated with the data management system by leveraging these new technologies.

What parts of traditional database technology do you find particularly problematic for non-volatile memory-based systems?

The main component of the database system that affects performance is the write-ahead logging protocol. That is the bottleneck from a run-time performance standpoint because there is a huge gap between computing and storage in a traditional data-processing system. But with the non-volatile memory (NVM), we have this opportunity to revisit this protocol, and that's why we redesigned that protocol in particular for NVM. And we showed that it actually does not only help to improve the performance by a significant margin but also enables instantaneous recovery from system failures, which was not really possible with traditional storage technologies.

Wow. Okay. Tell me about your new logging protocol.

So, the key idea behind the protocol that enables instantaneous recovery is that you periodically propagate the changes made by the database system to persistent memory. And that is feasible because persistent memory supports fast, random writes unlike canonical storage technologies. So, the database on disk is nearly always in sync with the changes that are being made by the transactions. And you also have some additional metadata that is being maintained to just keep track of the transactions that are running at the time of a failure. So, using this minimal metadata, when you come back from a system failure, you can just figure out, "Hey, what were the changes that were made by these transactions?" and just ignore them and immediately start processing new transactions. So, that's the key reason why this protocol enables significantly faster recovery from failures.

This surprises me because non-volatile memory was invented so long ago. You would think that people would've done this already, but they hadn't, obviously.

That's a great question. So, early in grad school, I actually learned a lot by reading papers from H.V. Jagadish and Hector Garcia-Molina on memory-oriented database systems. So, they were kind of anticipating the arrival of non-volatile memory in the early 1990s. The main game-changer from a practical standpoint is that the price-performance of non-volatile memory has really become commercially viable in the last decade. So, manufacturers like Intel have only started manufacturing these devices over the last five years, and major operating systems like Linux and Windows have native support for non-volatile memory right now. So, these were some key changes that enabled us to focus on this problem again with this fresh perspective.

Did your thesis touch on any other pain points?

Yeah. So, the other interesting aspect that was enabled by non-volatile memory is that traditional data structures like B+ Tree have kind of assumed that all the data resides on a hard disk drive. Now, with persistent memory, if you have a single-tier of storage hierarchy with just persistent memory and your indexing structure is backed by persistent memory, there is an opportunity to design a concurrent, persistent indexing data structure that is easy to maintain and delivers great performance by leveraging the properties of NVM.

I had this great opportunity to do an internship at Microsoft Research, and my mentors at MSR, Justin Levandoski and Umar Farooq Minhas encouraged me to focus on how indexing data structures have to be redesigned for persistent memory. And one of the interesting aspects we focused on that goes beyond just performance and availability was on how database developers design and implement these data structures. It is very challenging to actually design a concurrent, persistent data structure that delivers great performance. And we came up with a new primitive for managing data on persistent memory that allows us to atomically update multiple memory locations in a transactionally consistent manner called Persistent Multi-word Compare and Swap. We then used this programming primitive to quickly design a concurrent B+ Tree that's designed specifically for persistent memory. We found that it's actually easier to design this data structure because of the unique properties of NVM and that allows us to kind of get good performance, durability, and also maintainability.

Did your thesis have any impact in the industry?

Yeah. So, I have had great conversations with industry practitioners at Oracle Labs and Samsung Research. In fact, some of my graduate school research was influenced by the questions that were raised by these folks. And we found that the protocols and the new data structures that we designed are being adapted, enhanced, and used in practice in these companies.

[...] appreciate your journey through graduate school and not just focus on the destination.

Do you have any words of advice for today's graduate students?

So, first, and for most, it is super important to find a connecting theme in your research, and that's where it's good to get helpful inputs from your advisor. So, I had a really spectacular advisor, Andy Pablo. When we started this project in 2013, it was a long shot because NVM was not commercially available. But we decided that it was the right time to explore this project. And it was through this connecting theme that we explored the different components of this thesis.

Another suggestion that I have for students would be to actively reach out to mentors who can help shape their thesis. So, I had this great opportunity to do an internship at Microsoft Research, and my mentors at MSR encouraged me to focus on a new component of persistent memory database systems. And I got helpful pointers from Donald Kossmann and Phil Bernstein during my time there. My research was also supported by the Intel Science & Technology Center for Big Data that's based in MIT. And I was also part of the Parallel Data Lab at CMU. So, through these centers, I got several helpful pointers from Sam Madden, Greg Ganger, Garth Gibson, and Mike Stonebraker.

Last, but probably the most important suggestion would be to also try to appreciate your journey through graduate school and not just focus on the destination. You'll always have to learn to cope with rejections and hiccups in your research, but it is important to enjoy the scientific process and to always try to better balance your personal and professional aspirations.

Thank you very much for talking with me today.

It was a pleasure, Marianne. Thank you for having me here.