Viktor Leis Speaks Out on Concurrency and Parallelism on Multicore CPUs

Marianne Winslett and Vanessa Braganholo

Viktor Leis

https://www.cs6.tf.fau.de/person/viktor-leis/

Welcome to 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 Viktor Leis who won the 2018 ACM SIGMOD Jim Gray Dissertation Award for his thesis entitled Query Processing and Optimization in Modern Database Systems. Viktor is now at the University of University of Erlangen-Nuremberg and his Ph.D. is from the Technical University of Munich, where he worked with Thomas Neumann and Alfons Kemper. So, Viktor, welcome.

Can you tell me what your thesis is about?

Sure! My thesis is in the context of the HyPer system, which is a main memory database system, which we've been building at TU Munich. And in my thesis, specifically, I looked at a couple of different aspects. One big theme is really how do you exploit multicore CPUs. So, a big part of the thesis is how do you do that for concurrency control, for index structures, but also on query processing. How do you parallelize queries, so that you really exploit all these many core CPUs that you have nowadays? And so, in the thesis, you find lots of different techniques that really make it possible to get really good performance and really good speed-ups on modern CPUs.

Nowadays, almost any server CPU has dozens of cores. And so, that's really what I would say is the main focus of my thesis. It also has a chapter on query optimization. But I would say, if there was really one big thing, it's really concurrency and parallelism on multicore CPUs.

So, what was the state of the art like before you did this? Why was this work needed?

So, at that time – and this is also how HyPer started – the initial idea was that in traditional systems you have lots of locks and latches, and this really doesn't scale. And so, the idea was you have to get rid of all of them, and you have to partition your data. And you also have to partition your workload and that obviously can give you very good performance if your workload and your data are actually partitionable. That may work very well for microbenchmarks and for many macro benchmarks that we use in research. But I'm a little bit skeptical: in the real world, a lot of workloads are very messy, very complicated, and it's very hard to find a partitioning that will make this thing actually work. Another way of saying it is that if your workload is partitionable, you can just run five different instances of your database. That would work as well. That's all.

Yeah.

That's, in a way, a little bit cheating. And so, I think 10 years ago, it was not really clear if you could actually achieve very good synchronization without partitioning. And I think now we can say that we can actually do that.

So, what kind of speed-ups do you get now compared to before with your techniques?

So, for the intra-query parallelization: you have one query that scans lots of data, does lots of joins, and so on. Actually, my goal is always if you have a system with 50 CPU cores, then you want a speed up close to 50. Now, you don't always get it, right, but this not totally unrealistic. For most queries, you will actually get that. And to achieve that actually requires a lot of work because what you see at these kinds of speed-ups is that Amdahl's law really kicks in. Because any small part of your query that doesn't scale, that is not parallelized, means you will not get that speed-up of 50.

So, it requires a lot of work, and every single operator needs to be parallelized. A lot of these operators are described in my thesis. And not just joins, but also aggregation, sorting, and window functions. All these operators. And I think there's a lot of work required, both in terms of algorithms and in terms of a lot of low-level techniques.

What kind of impact do you see your thesis having in industry?

> ***[…] pick the right advisor. And they must be someone that you respect […] technically, scientifically […]***

So, my thesis was about the system HyPer. We made a startup, and pretty quickly after my graduation it was acquired by Tableau. And they then put it into their data visualization products. So, if you now click around in Tableau, then it will use HyPer behind the scenes. So, it's actually pretty cool! Particularly, the code of the query parallelization and the query engine that I was talking about is actually now used every day by hundreds of thousands of people. I don't know the exact number, but that's something I am really proud of.

That's really cool. I'm so glad to hear that. Do you have any words of advice for today's graduate students?

Sure. And I should say that, nowadays, you hear a lot of scary stories about how terrible a Ph.D. is. I actually had the best time ever in my Ph.D. But I think that depends mostly on your thesis advisor. So, my main advice, basically, is to pick the right advisor. And they must be someone that you respect, like technically, scientifically, their skills, and also what they're interested in. I think that it's very important that these things are aligned with your interests, and also on a personal level. A Ph.D. takes a long time, and it's also a very personal experience, and so, that is important.

So, I think that's maybe the most important thing. It's probably much more important than whether it's a very fancy school or something in comparison with somebody's who's maybe not in an as fancy university. But if you have a good relationship and you respect the person, that will go a very long way. So, I think that that's a really big thing.

The second piece of advice I could give is – actually, I did my Ph.D. like five years ago, so now I also have a lot of experience on the other side. And so, one thing that I see from a lot of students is, when they have an idea or did something and it's reasonably okay, they immediately want to publish it. They think that's a great thing. I think there's a risk in that because not everything that you can publish – and it turns out you can actually publish everything if you spend a lot of effort there – should be published in the sense that it's much better to try a couple of things, and then pick the best one and then spend that half a year or whatever it takes to polish the paper and so on. I think that's a much better approach.

And I think paradoxically, this can lead to you publishing more work. You will have stronger results that are easier to publish. It will actually make you happy if you believe in the stuff that you are working on. And so, it's pretty easy to do a couple of prototypes. Maybe work on something for a month, and see, "is that just an okay idea or is it actually really good?" It's much better to reiterate and do a couple of prototypes. So, I think that's something more people should do and not just decide on one topic and then fight it through. Because you can do it, but it's just not a lot of fun. So, that would be a second thing I can recommend.

Thank you very much for talking to me today.

Thank you.