Paris Koutris Speaks Out on a Theoretical Model for Query Processing

Marianne Winslett and Vanessa Braganholo

Paris Koutris

http://pages.cs.wisc.edu/~paris/

Welcome to the ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we're at SIGMOD 2017 in Chicago. I have with me Paris Koutris, who won the 2016 ACM SIGMOD Jim Gray Dissertation Award for his thesis entitled "Query Processing in Massively Parallel Systems." Paris is now a professor at the University of Wisconsin-Madison, and he did his Ph.D. work with Dan Suciu at the University of Washington.

So, Paris, welcome!

It is great to be here.

What is your thesis about?

My thesis has to do with query processing in massively parallel systems. The key observation is that due to the data explosion we've seen over the last years, there's a massive volume of data being around, and we have to process this data. In order to process this data fast, one way is to use parallelism. This has led to an explosion of different types of systems – distributed systems, parallel systems – that try to improve performance. My work has to do with how we can theoretically model these types of systems and how we can formally reason about these systems.

My first contribution was introducing a model which we called the Massively Parallel Computation model (or MPC for short) that basically creates a theoretical framework to analyze query processing. This model has two main parameters. The first one is communication, so it measures how much data is being exchanged, and the other is the number of rounds or synchronizations. This measures how often does the system have to synchronize and wait for all the machines to reach the same point before moving forward. Using this model, my thesis analyzed different types of algorithms for join processing. Joins are the backbone of any database system. And so, what we did is try to find out if there is a tradeoff between communication and synchronization, and how can we model it, and not only try to create new algorithms, but also try to give lower bounds on how well these algorithms can perform. This is the main part of my thesis.

The second part has to do with what we can do further. For example, many times, data has skew, which means that there are some values in the data that appear more often than the others, and that can create an imbalance in query processing. In this case, we have to use different types of techniques to deal with skew. And my thesis also tried to reason about these types of problems.

Does that mean that you introduced new join algorithms yourself or improved the existing ones?

It's actually both. In the thesis, we both analyzed existing algorithms and proved new bounds on how well they can do, and also introduced some techniques that were novel and could be actually used in practice.

What kind of new join technique did you use?

For example, the new technique is how we can deal with skew. Typically, when we are doing a hash join algorithm, we are distributing the elements by hashing a particular value. Now if this value appears very often, there will be skew, so there's going to be a struggler in a machine that will end up doing more work than the other machines. In order to deal with this problem, we essentially have to find out which are the values that have skew and split up their work in more machines. And we have to do it in a very particular way so that we can get the best possible performance.

Given that this is a classic issue, I find it very surprising that people hadn't already come up with techniques to do a better job of spreading key values.

There are existing techniques to do that, but what we did is we showed which are the theoretical optimal techniques that you can use. So, for some cases, we did use some existing tools. For some others, we had to introduce new ways of balancing that were theoretically optimal.

How close are we now to the theoretical optimal lower bound?

This is an excellent question. In some cases, some of these new theoretical ideas have proved to be faster in practice than the typical algorithms. But there are cases where the constants in the theoretical analysis are so large that going back to some of the classic techniques is faster.

The issue here – and this is generally an issue with theoretical analysis – is that we make worst-case assumptions about the data: for example, we are assuming that we're analyzing the worst case that can happen. And of course, for many real-world datasets this is far away from the truth. So, a very exciting direction is to try to incorporate this assumption in the analysis and try to see how you can prove that the analysis of an algorithm theoretically matches the behavior that we see in the real systems.

What was your model like? Is it based on queuing theory or another approach?

The model is actually very close to the BSP model by Valiant, the Bulk Synchronous Model[1]. The idea is that processing operates in rounds, and at each round there is some communication, some computation, and then

[1] Leslie G. Valiant. A bridging model for parallel computation. Communications of the ACM, 33(8):103–111, 1990.

there is a synchronization barrier. But in order to make our analysis feasible, we abstracted away some of the parameters of Valiant's model. For example, we ignore the computation and try to figure out how well the data is balanced across the different machines that we have.

That topic sounds quite classical and not very Dan Suciu like. Where did the topic come from?

Excellent question. The story is interesting. When I started my Ph.D. at the University of Washington, I started talking with Dan on possible projects I could do. And he was talking about probabilistic databases and all the other things he has been doing, and then he also mentioned this idea of "Oh, people like doing parallel join processing, and we don't know yet how to analyze this", and I got immediately attracted to that problem and started working on that. And I think that turned out very well. So, it was kind of by luck that I started working on this project, but it was very interesting.

> ***[...] by talking with more people and collaborating with more people, you're going to come across with many different ideas. And that may actually improve your research.***

Do you have any words of advice for graduate students?

Yes. One thing that I think is very important, and sometimes in this competitive environment where you are trying to publish as many papers as possible it is kind of lost, is not only to do research but to also try to talk with as many people as possible and try to network with as many people as possible. And also try to develop collaborations with many people – other students in your department, possibly other professors in your department, or also other people and other students from other departments. And the way I view it is that, if you plan to stay in academia, these people will actually be your colleagues for the rest of your career. That's one thing. And second, by talking with more people and collaborating with more people, you're going to come across with many different ideas. And that may actually improve your research.

So, the second thing that I want to say is that students should not be afraid to tackle new problems. It's probably easier to look at some existing papers and then try to improve upon these or try to think about a new technique that gets an improvement of 10 percent in the performance. But I think it's much more impactful if you try to go to new areas and try to introduce new problems, new frameworks, and in general, try to explore new things. The disadvantage of that is that it will be harder, possibly, to convince the database community that this is an important problem, and we need to do research on that. But on the other hand, the results – the potential of this type of research is much higher.

Did you have trouble convincing the community that your particular topic was something they should care about?

I would say no for my case, but I've come across many other cases where this has happened. So, I know that this is an issue and a danger if you're trying to do these types of things. And my point is that you should not be discouraged if this happens, and you should try to push through these directions.

Alright. Well, thank you very much for talking with us today.

Yeah, it was very nice being here. Thank you very much