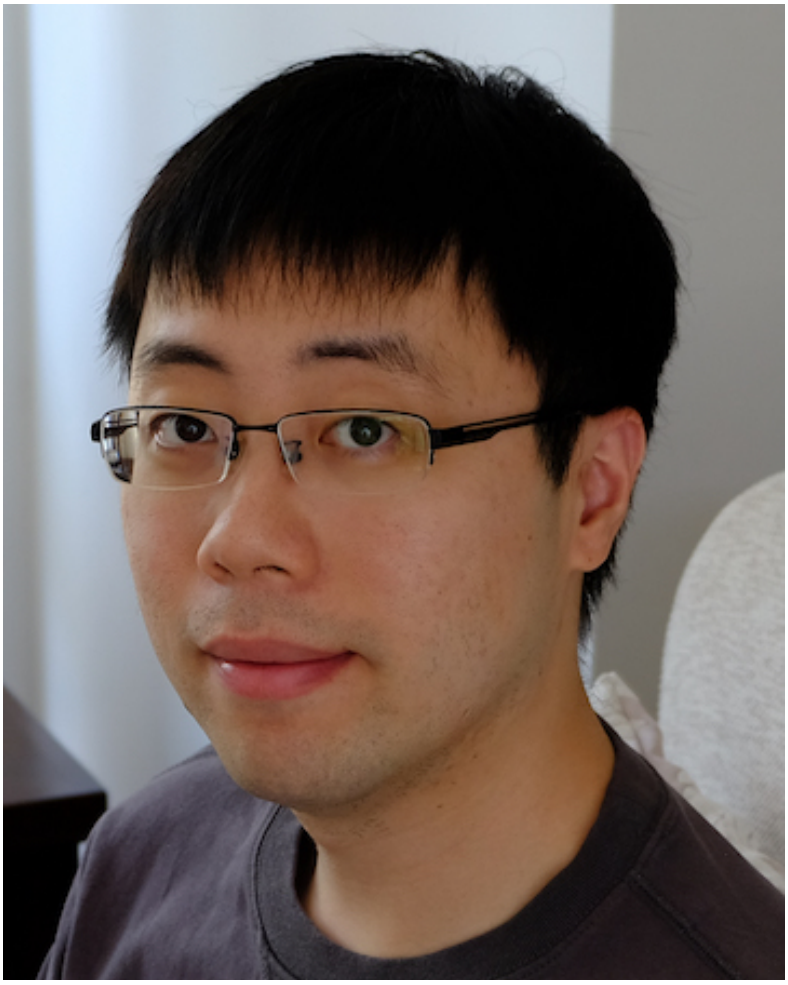
Huanchen Zhang Speaks Out on Memory-Efficient Search Trees

Marianne Winslett and Vanessa Braganholo



Huanchen Zhang

https://people.iiis.tsinghua.edu.cn/~huanchen/

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 Huanchen Zhang, who is the 2021 winner of the ACM SIGMOD Jim Gray Dissertation Award for his thesis entitled Memory-Efficient Search Trees for Database Management Systems. After a postdoc at Snowflake, Huanchen is now an Assistant professor at Tsinghua University. His PhD is from Carnegie Mellon University, where he worked with David Andersen and Andy Pavlo. So, Huanchen, welcome!

Thank you.

What was your dissertation about?

My dissertation was focused on how to reduce the size of in-memory search trees in a database without compromising query performance. In many cases, the techniques introduced improve the end-to-end system performance in addition to the memory saving. We observed this problem back in 2015 when running the TPC-C benchmark on H-Store, which is a main memory OLTP database. And we found in the experiment that the indexes, which are B+ trees, eat up more than half of the database memory.

That's insane, right? The indexes are literally larger than the actual data you store. And someone may argue that the DRAM is getting cheap, so just don't worry about it, but that's not true. If you compare the per-gigabyte cost between DRAM and SSD for the past seven, eight years, the price gap between them is increasing. So, the truth is that DRAM is getting relatively more expensive when compared to storage. Also, because of the rapidly growing database sizes, memory as a resource is actually even more precious than before.

[...] stay healthy. No research is more important than your health. So, get enough sleep, take a vacation when you feel stressed. Life is much more than just research. Enjoy it.

So, under this background, reducing the memory footprint of those search tree indexes makes a lot of sense. It improves the memory efficiency of database systems -- and a better memory efficiency eventually translates to better performance and a smaller bill.

When I talk about compressing indexes, people usually think of using block compression algorithms such as snappy, LZ4, or Zstandard at a page level. This approach works really well for disk-based indexes because the IO latency hides everything. But in the case of in-memory search trees, where we are talking about several million index operations per second, the decompression overhead of these algorithms is just too expensive.

So, my entire thesis is about designing new search tree data structures that are super-small in size and super-fast at the same time. I provided a three-step recipe in my thesis to achieve that goal. As the first step, we focused on static data structures because they are much easier to compress. What we do here is that we borrow the concept of succinct data structures from the algorithms community and sort of reinvent this technique from a systems engineering perspective so that it's fast enough to be used in practice in addition to being succinct. A representative data structure we built is called SuRF. SuRF is a range filter. It's like a bloom filter, but it can also handle range queries. And it's quite useful. It's been used in LSM trees today. Under the cover, it's a static trie data structure that we designed with a space consumption close to the optimal -- optimal meaning the minimum bits required by the information theory. And we managed to engineer it to be really fast, with a performance comparable to some of the fastest trees out there.

Once we have this fast and close-to-optimal-space, static data structure, the next step in our recipe is to relax this constraint of being read-only. Here we introduced the hybrid index design as an efficient way to handle individual inserts, updates, and deletes to the static tree, but with an amortized cost. The key idea is very simple. We put a dynamic tree as a write buffer in front of the static one, and we do periodic merges between them. With a clever merge strategy, we managed to bound the overhead of this extra merge step to be very small.

Now, the last step in my recipe deals with the index keys because the first two steps already pushed the structural overhead of a search tree to the minimum. So, the dominating factor in terms of space shifts towards the index keys. Now, there are several challenges of compressing index keys. The first is that the encoding has to be order-preserving, otherwise you won't be able to perform range queries on the tree. And the second is that you don't know all the keys beforehand: a user can insert arbitrary keys. So, the traditional dictionary encoding doesn't work here. These are the challenges we solved in the last piece in my dissertation. In one sentence, we've built a super-fast compression tool called HOPE that can encode arbitrary input strings while preserving their original order.

Altogether, these three steps form a practical recipe for achieving memory efficiency in search trees, and that's my thesis.

Have you seen impact from your thesis in industry?

Yes. The SuRF range filter is being used by several major internet companies in their LSM-tree engines.

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

Yeah, I do have many because my Ph.D. wasn't that smooth. But according to my observation, the main reasons for Ph.D. students to drop out are: (i) they feel they have accomplished nothing, so they lost confidence; and (ii) they have to give up because of bad health (either physical or mental health). So, my advice would be first, whatever project you're currently doing, finish it. Even if you think it's a dead-end, just wrap it up, publish the results somewhere, maybe on archive, before thinking about what's next. It's very dangerous if you just jump between topics and projects and end up not completing any of them -- it will destroy your confidence in getting things done. And my second advice obviously is to stay healthy. No research is more important than your health. So, get enough sleep, take a vacation when you feel stressed. Life is much more than just research. Enjoy it.

Thank you very much for talking with me today.

Thank you for having me.