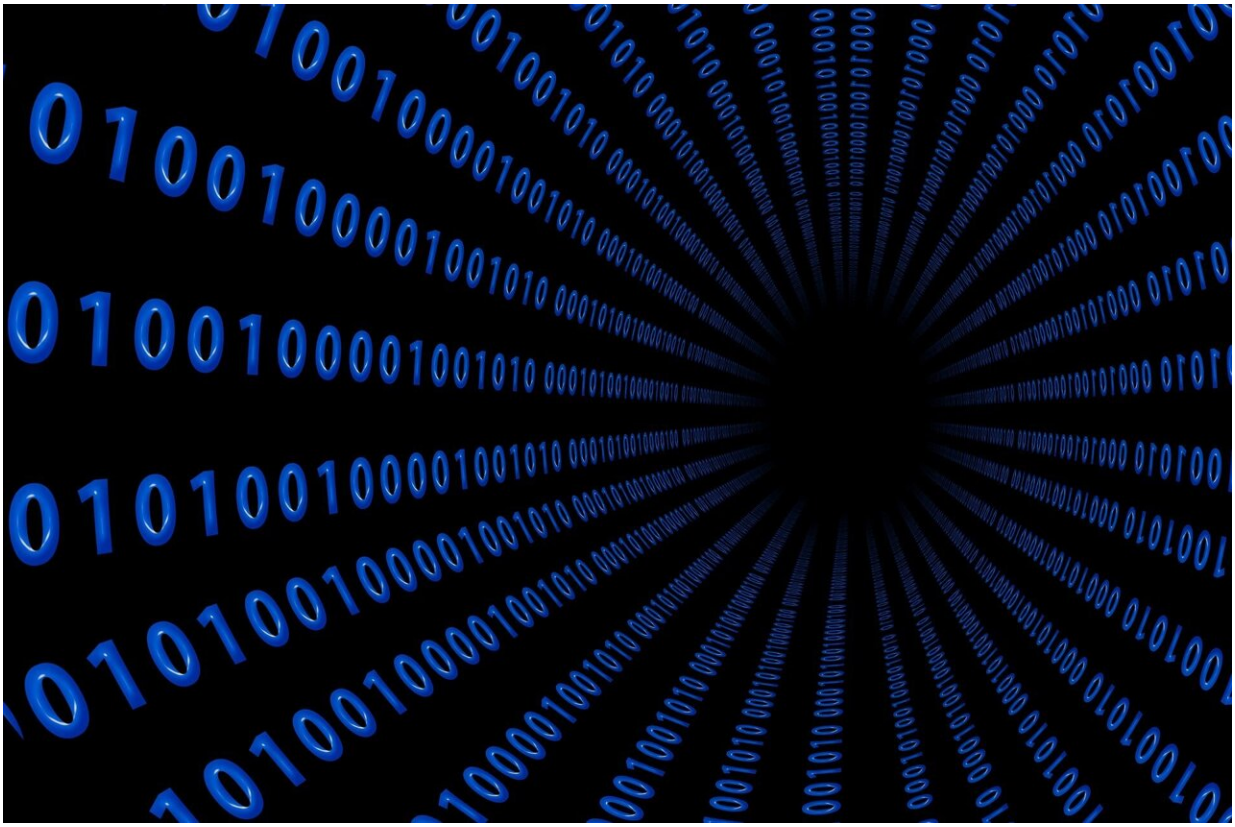
Theoretical breakthrough could boost data storage

November 16 2021, by Steve Nadis



Credit: Pixabay/CC0 Public Domain

A trio of researchers that includes William Kuszmaul—a computer science Ph.D. student at MIT—has made a discovery that could lead to more efficient data storage and retrieval in computers.

The team's findings relate to so-called "linear-probing hash tables," which were introduced in 1954 and are among the oldest, simplest, and fastest data structures available today. Data structures provide ways of organizing and storing data in computers, with hash tables being one of the most commonly utilized approaches. In a linear-probing hash table, the positions in which information can be stored lie along a linear array.

Suppose, for instance, that a database is designed to store the Social Security numbers of 10,000 people, Kuszmaul suggests. "We take your Social Security number, x, and we'll then compute the hash function of x, h(x), which gives you a random number between one and 10,000." The next step is to take that random number, h(x), go to that position in the array, and put x, the Social Security number, into that spot.

If there's already something occupying that spot, Kuszmaul says, "you just move forward to the next free position and put it there. This is where the term 'linear probing' comes from, as you keep moving forward linearly until you find an open spot." In order to later retrieve that Social Security number, x, you just go to the designated spot, h(x), and if it's not there, you move forward until you either find x or come to a free position and conclude that x is not in your database.

There's a somewhat different protocol for deleting an item, such as a Social Security number. If you just left an empty spot in the hash table after deleting the information, that could cause confusion when you later tried to find something else, as the vacant spot might erroneously suggest that the item you're looking for is nowhere to be found in the database. To avoid that problem, Kuszmaul explains, "you can go to the spot where the element was removed and put a little marker there called a 'tombstone,' which indicates there used to be an element here, but it's gone now."

This general procedure has been followed for more than half a century.

But in all that time, almost everyone using linear-probing hash tables has assumed that if you allow them to get too full, long stretches of occupied spots would run together to form "clusters." As a result, the time it takes to find a free spot would go up dramatically—quadratically, in fact—taking so long as to be impractical. Consequently, people have been trained to operate hash tables at low capacity—a practice that can exact an economic toll by affecting the amount of hardware a company has to purchase and maintain.

But this time-honored principle, which has long militated against high load factors, has been totally upended by the work of Kuszmaul and his colleagues, Michael Bender of Stony Brook University and Bradley Kuszmaul of Google. They found that for applications where the number of insertions and deletions stays about the same—and the amount of data added is roughly equal to that removed—linear-probing hash tables can operate at high storage capacities without sacrificing speed.

In addition, the team has devised a new strategy, called "graveyard hashing," which involves artificially increasing the number of tombstones placed in an array until they occupy about half the free spots. These tombstones then reserve spaces that can be used for future insertions.

This approach, which runs contrary to what people have customarily been instructed to do, Kuszmaul says, "can lead to optimal performance in linear-probing hash tables." Or, as he and his coauthors maintain in their paper, the "well-designed use of tombstones can completely change the … landscape of how linear probing behaves."

Kuszmaul wrote up these findings with Bender and Kuszmaul in a paper posted earlier this year that will be presented in February at the Foundations of Computer Science (FOCS) Symposium in Boulder, Colorado.

Kuszmaul's Ph.D. thesis advisor, MIT computer science professor Charles E. Leiserson (who did not participate in this research), agrees with that assessment. "These new and surprising results overturn one of the oldest conventional wisdoms about hash table behavior," Leiserson says. "The lessons will reverberate for years among theoreticians and practitioners alike."

As for translating their results into practice, Kuszmaul notes, "there are many considerations that go into building a hash table. Although we've advanced the story considerably from a theoretical standpoint, we're just starting to explore the experimental side of things."

More information: Linear Probing Revisited: Tombstones Mark the Death of Primary Clustering, arXiv:2107.01250 [cs.DS] arxiv.org/abs/2107.01250

Provided by Massachusetts Institute of Technology