Understanding the Performance of SGX-based Computations

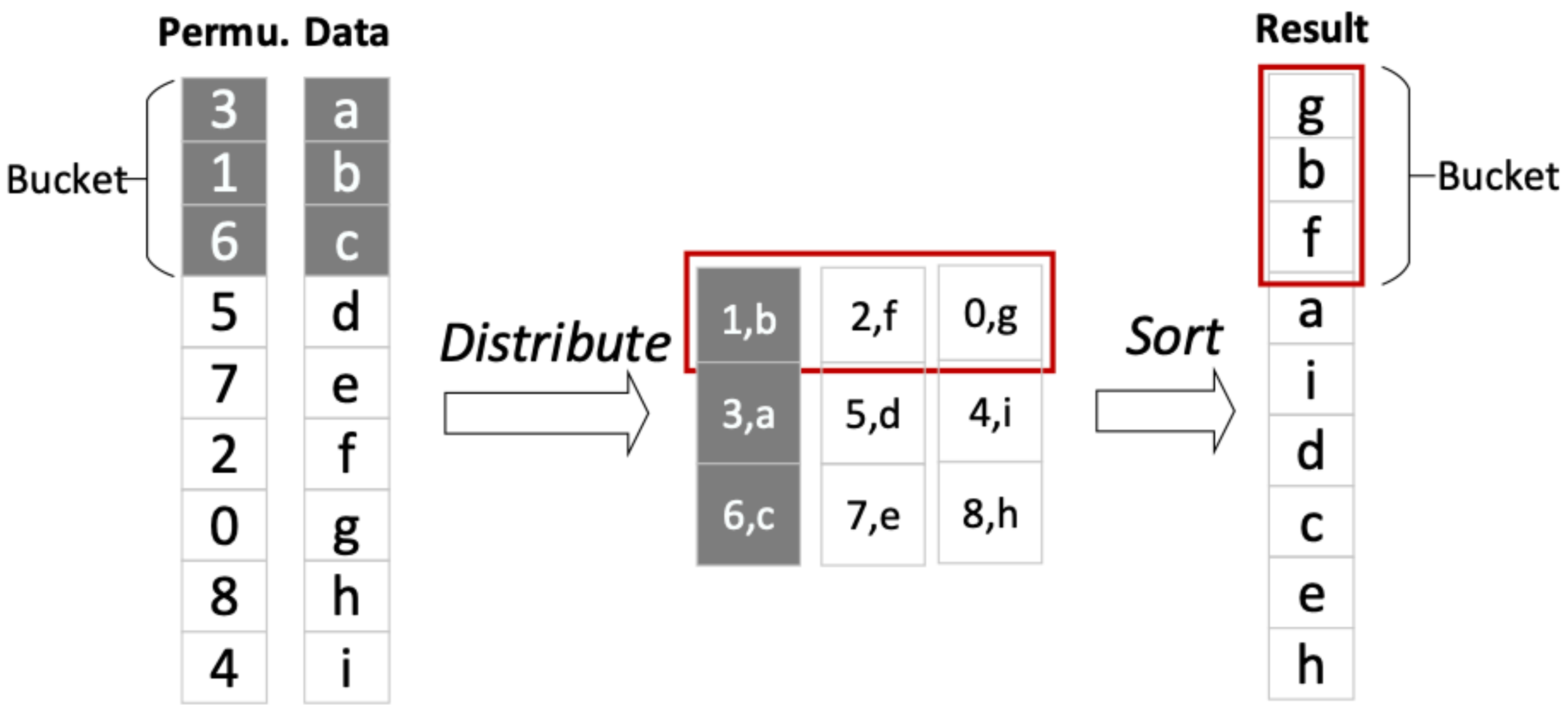
Jeongbin Oh

As technology is developing faster these days, the use of cloud computing has increased a lot such as storing personal and significant information in there. Since valuable information are stored, various side-channel attacks are proposed and occurred. To prevent these attacks, the SGX (software Guard eXtension) is used to protect the environment. Especially, we focused on using the cache-miss obliviousness for the defense. Usually, the hardware enclave load data from the cache, but if the data isn't in the cache, then it loads from the memory and this process is called cache-miss. The cache-miss obliviousness let enclave features a trusted processor issuing cache misses to access the memory in untrusted world so that it makes side-channel attack more difficult although it loads data from untrusted environment.

While in the process of cache-miss obligation, we should be aware that cache-miss should only touch the untrusted memory which are made oblivious. Therefore, we chose to use Melbourne shuffle algorithm which shuffle the data and divide them into oblivious and non-oblivious. Especially, oblivious ones goes to external storage which is untrusted world and this leads the cache-miss to be generated safely.

An example of Computations:

Shuffling data arrays



Instruction Count	2 million	20 million
Compute Sum **Outside** enclave	6705 / 6820	101796 / 1013757
Compute Sum **Inside** enclave	62305 / 62249	817413 / 792673
Compute Sum **Inside** enclave (array outside)	6594 / 6791	102354 / 103407

The chart above is and example of performance difference for simply computing the sum of array with 3 different ways. Computing sum in outside of enclave, inside of enclave, inside of enclave but load the data from outside of the array. Storing the array of data in inside enclave and also computing it takes much more time compare to doing it in outside of enclave. However, by putting the memory data outside of enclave, it induces better performance than other methods.

The modified version of SGX with using Melbourne Shuffle has lower time complexity which leads to better performance compare to classic word-oblivious algorithms that is used widely. Therefore, the use of new version of SGX will be more safe to all users using cloud computing.

In educational point of view, the use of our SGX would bring more trust in uploading significant information in cloud computing system. Especially, it has lower time complexity which takes less time, so the modified SGX would bring faster and safer outcomes.

Since our SGX uses Melbourne shuffle to generate the cache-miss obliviousness which is different from classic SGX, once the efficiency is proved, this can lead to better security by preventing side-channel attacks more effectively.

The 4th NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting (2019 SaTC PI Meeting)
October 28-29, 2019 I Alexandria, Virginia