Paper Review:

TritonSort: A Balanced and Energy-Efficient Large-Scale Sorting System

Braden D. Licastro
Department of Computer Science
Allegheny College
licastb@allegheny.edu
http://www.fullforceapps.com/

April 2, 2013

1 Summary

This paper describes a new, highly efficient sorting algorithm coined Triton Sort. The authors claim the algorithm to be "a highly efficient, scalable sorting system." [1] The algorithm described was designed to efficiently sort massive data sets, spanning into hundreds of terabytes in size. Many algorithms exist to sort data, but in sets of this magnitude, they are excessively resource hungry and inefficient. To add further complexity to the problem, a dataset of this size will inherently need to be spanned across many disks, and even entire server nodes. This creates an additional speed limit regarding network transfer rates, and each individual nodes I/O rate. The authors created an algorithm that accounts for the numerous variables and is able to choose a median between scalability and performance. By doing this, they show that it is possible to either reduce system cost and complete the same amount of work, or use the same, more cost-prohibitive systems to evaluate larger data sets.

In order to validate their claim of creating a significantly improved sorting function the authors performed numerous tests and found their system was able to far outperform existing systems. After deciding on an optimal hardware configuration and write sizes to create the best environment for testing the algorithm and eliminating external factors the researchers performed their test on several different data sets. Overall they found that the system performed approximately 66% better than the previous record holder while utilizing about 80% of the disks' write speed. [1].

2 Critique

During the testing phase the authors mention a necessary increase in disks per cluster over the previous algorithm in order to run TritonSort in the same dataset size. While the algorithm may run significantly faster, I believe this would introduce a much higher likelihood of data corruption as the number of disks that could fail increases. In addition to this, each node would need additional hardware to accommodate the higher number of disks with additional points of failure. This was not a focus of the paper, but in a real world environment these issues could cause massive data corruption when running large datasets at the speeds discussed. In addition, every aspect of the computer systems and network were optimized specifically for the purpose of the testing. Would the algorithm performance degrade or remain consistent on other hardware? Will a slowdown be insignificant or great enough that a less environment specific algorithm could outperform TritonSort?

3 Synthesis

After reading the research paper I concluded that additional research could be performed in many aspects, primarily testing performance of data residing on nodes in different geographic locations. With data sets ranging in the hundreds of gigabytes it is completely feasible that the information could be spread across several data-centers. This would introduce a vast array of new problems ranging from non-uniform hardware, varying transfer rates between data centers, and possible transfer failures. In this situation I believe that the system could be made significantly more robust and possibly allow an area for automatic optimization of the algorithm to accommodate for an ever changing environment.

References

[1] Gregory M. Kapfhammer, Phil McMinn, and Chris J. Wright. Search-based testing of relational schema integrity constraints across multiple database management systems. In *International Conference on Software Testing, Verification and Validation (ICST 2013)*. IEEE, March 2013.