Research Seminar Report

What does the operating system ever do for me?
Systems Challenges in Graph Analytics
Tim Harris of Oracle Labs
November 7, 2016

Dr. Tim Harris spoke to Texas A&M's computer science department about his work at the Oracle Labs group in Cambridge, in the UK.[1] He summarized some of his group's research over the past few years pertaining to in-memory graph analytics. In this area of research, the aim is to develop more efficient and scalable approaches to large-scale graph problems in terms of both time and system resources. Three specific areas of exploration were put forward: (1) parallel work distribution, or how different batches of vertices are apportioned among threads; (2) memory allocation, which relates to storing graph data efficiently and accessibly in computer memory; and (3) thread placement, which focuses on sharing system resources efficiently among threads in the course of operations.

The main algorithm discussed was the PageRank algorithm, whose main application is to quantify the relevance of web pages for search engines, among other uses. The "inner loop" of this algorithm involves processing batches of vertices using multiple threads.[2] A problem arises in determining how many vertices to allocate per thread. Larger batch sizes, i.e., more vertices per thread, can avoid the high cost of distributing vertices among many threads, while smaller "fine-grained" batch sizes expedite individual threads' runtimes.[3]

Dr. Harris's group proposes a technique to address this problem that it calls *efficient fine-grained distribution*. According to this approach, threads distribute vertices at the start of the loop according to per-core counters. When a thread is approaching the end of its batch of work, it sets a certain flag. Newly finished threads proactively allocate unprocessed vertices among themselves and soon-to-be-finished threads according to those threads' flags. In this way, one thread is able to efficiently perform the work distribution operations of multiple others so that the majority of threads can focus on actually processing vertices.

While our course has not delved into parallel processing implementations, it has covered many graph-based algorithms. The PageRank algorithm discussed in this seminar constitutes a more advanced example of a graph-based algorithm. As with several algorithms discussed in class, such as Dijkstra's algorithm, it involves an inner loop that processes items of data conceptualized as graph vertices. Because of the scale of PageRank problems, the algorithm's real-world execution often demands multiple processors. This seminar gave me an idea of the challenges posed by adding parallel processing considerations to calculating graph algorithm time complexities. Namely, the problem of efficiently coordinating graph operations between different threads must be addressed. In addition, one must consider the problems that might emerge when simultaneous graph processes share the same memory.

I certainly did not follow all the intricacies of the seminar. Discussion of methods of distributing memory allocation for PageRank were particularly foggy. However, the talk provided me with a

greater understanding of and appreciation for parallel computing. Throughout the seminar there was an assumption of using parallel processing and an emphasis on optimizing methods of parallel processing. Going in, I do not think I appreciated the significance or widespread use of parallel computing. Likely I simply have not encountered it enough, since it is a more advanced tool that requires grounding in other areas. However, its usefulness is obvious considering the size of real-world data sets processed by graph algorithms like PageRank. It seems that once an optimal algorithm is determined theoretically, parallel processing provides the greatest or sometimes the only opportunity for significant further improvement outside of upgrading the underlying hardware.

Bibliography

[1] Harris, Tim. "Tim Harris - Architect." *Oracle Labs*. Oacle, n.d. Web. 01 Dec. 2016. <https://labs.oracle.com/pls/apex/f?p=labs%3Abio%3A0%3A296>.

[2] "PageRank." *Wikipedia*. Wikimedia Foundation, n.d. Web. 01 Dec. 2016. <https://en.wikipedia.org/wiki/PageRank>.

[3] Brian. "Graph PageRank: SPARC M7-8 Beats X86 E5 V3 Per Chip." *Graph PageRank: SPARC M7-8 Beats X86 E5 V3 Per Chip (BestPerf)*. Oracle, 26 Oct. 2015. Web. 01 Dec. 2016. <https://blogs.oracle.com/BestPerf/entry/20151025_graphpagerank_m7_8>.