CS 267 Homework 0

Peter Birsinger

February 7, 2013

Bio

I am a senior undergraduate here at UC Berkeley, majoring in EECS (heavy focus on CS) and Applied Mathematics. I do research in the Par Lab under Armando Fox with the Sejits (Selective Embedded Just-In-Time Specialization) group. Specifically, I have worked on a specializer for the Bag of Little Bootstraps (BLB) algorithm with a cloud backend (Spark cluster computing system). Generally speaking, I am interested in distributed systems and their use for manipulation of big data. From this class, I would like to improve my ability to write parallel programs and also to spot in problems when and how parallelism should be utilized.

Application: Distributed File Tree Walk of Parallel File Systems

The problem attempted to be solved here is the parallel traversal of large distributed file systems. I find this interesting because this application involves distributed systems, massive data, and a fresh algorithmic approach all in one. Typically, such traversal algorithms are serial and thus well underutilize the potential of most modern computers. Traversing a distributed file system tree is in some ways similar to traversing a graph as a filesystem is organized in a tree. However, already existing parallel graph algorithms are unsuitable for traversing file trees because unlike in many graph algorithms, every node in the file tree must be visited so no subtrees can be ignored. Also, there is no express need for synchronization of the separate slave processes in a file tree traversal, allowing for large speedups. In addition, graphs can often be constructed efficiently, but in order to do so, (nearly) complete graph information must be known a priori, something not always possible with huge distributed filesystems. In terms of already existing parallel algorithms for parallel distributed file tree traversal, they had an overly high communication overhead between a centralized master and slave processes.

The new algorithm framework in addition to the three new novel parallel algorithms achieved its objective of reducing communication overhead. Specifically, on a file tree of 100 million files, the new parallel algorithms exchange two orders of magnitude less bytes than a centralized parallel algorithm and have on average a quarter of the running time with an equal number of processes. The techniques balanced system work load uniformly in real-world experiments while incurring low communication costs and without global process synchronization.
These algorithms were tested on a state of the art parallel file system called Panasas on a supercomputer at the Los Alamos National Laboratory. Panasas distributes file metatdata for files as file components across the storage system for performance and redundancy. Also, these metadata storing components can be accessed in parallel, and the Panasas file systems has multiple optimizations for quick metadata access. The algorithms presented use the MPI standard. The 1.37 petaflop supercomputer used, named Cielo, is currently ranked 18th in the world.

References

J. LaFon, S. Misra, J. Bringhurst, "On Distributed File Tree Walk of Parallel File Systems" in *Supercomputing Conference* , 2012.