論文の内容の要旨

論文題目

Effective Deployment of Data-intensive Frameworks on Supercomputers
(スーパーコンピュータ上へのデータ集約型計算フレームワークの効果的展開)

氏名： ダオ　タンチュン

(本文)

The goal of this research is to achieve a better performance of popular data-intensive frameworks, for example Hadoop and Spark, with only small modifications when running on modern supercomputers. Big data analytics applications have been becoming more important and widely being demanded to process large-scale datasets in both industry and academia. Compared with developing a new data-intensive application from scratch, using existing popular data-intensive application frameworks to develop is a better choice in aspects of productivity and maturity. Supercomputers are potentially faster than commodity clusters, such as Amazon EC2 cloud, when running data-intensive applications due to their high-performance and high-cost hardware. However, the current supercomputer design focuses more on compute-intensive applications rather than data-intensive ones, so it is hard to achieve the best performance of the hardware when running data-intensive applications on supercomputers. This is partly because it is also important to keep the original data-intensive frameworks' source code as much as possible since minimizing the cost of changes in the architecture helps increase productivity and easily upgrade to newer versions.

We observe two mismatches of the execution environment on a lack of MPI-friendly dynamic process creation and local disks when running those frameworks on supercomputers since they are designed to run on the commodity clusters, but supercomputer design is different from commodity clusters. The first mismatch raises a question of how to provide MPI-compatible fast dynamic process creation for popular data-intensive frameworks but satisfy the standard way of creating processes on supercomputers. The second mismatch brings into question of when using in-memory storage to provide virtual local disks as a replacement of physical local disks, how to deploy that in-memory storage and what deployment strategy is good on supercomputers.

To overcome the first mismatch, we propose HPC-Reuse located between YARN-like and PBS-like resource managers in order to provide better support of dynamic management with MPI. YARN, which is a key component of Hadoop, our targeted data-intensive framework, is responsible for resource management. YARN adopts dynamic management for job execution and scheduling. We identify three Ds (3D) dynamic characteristics from YARN-like management: on-Demand (processes created during job execution), Diverse job, and Detailed (fine-grained allocation). The dynamic management does not fit into typical resource managers on supercomputers, for example PBS, that are identified having three Ss (3S) static characteristics: Stationary (no newly created process during execution), Single job, and Shallow (coarse-grained allocation). Our experimental results show that HPC-Reuse can reduce execution time of iterative workloads by 26% and speed up data shuffle up to 10% by using MPI.

Regarding to the second mismatch, we report our experiments to compare various deployment strategies of memcached-like in-memory storage for our focused Hadoop framework on supercomputers, where each node often does not have a local disk but shares a slow central disk in order to find the best strategy. For the experiments, we develop our own memcached-like file system, named SEMem, for Hadoop. Since SEMem is designed for supercomputers, it uses MPI for communication. SEMem is configurable to adopt various deployment strategies and our experiments reveal that a good deployment strategy is allocating some nodes that work only for in-memory storage but do not directly perform MapReduce computation, with up to 10-13% improvement in comparison with deploying the memcached-like file system on every computation node.