Classical and Iterative MapReduce on Azure

Judy Qiu, Thilina Gunarathne, Geoffrey Fox

We describe experiences on Azure, Amazon and academic systems (FutureGrid) on different applications (mostly from the Life Sciences) using various implementations of MapReduce. In particular we discuss MapReduceRoles4Azure that implements MapReduce on Azure building on Azure Queues for task scheduling; Azure Blob storage for input, output and intermediate data storage; Azure Tables for meta-data storage and monitoring. MRRoles4Azure supports a combiner step; dynamically scaling up and down of the compute resources; Web based monitoring console; testing and deployment using Azure local development fabric. We show that Azure gets excellent performance on parallel data intensive applications not requiring the microsecond latency of classic MPI-based simulations.

There exists many data analytics as well as scientific computation algorithms that rely on iterative computations, where each iterative step can be easily specified as a MapReduce computation. Twister4Azure extends the MRRoles4Azure to support such iterative MapReduce executions, drawing lessons from the Java Twister iterative MapReduce framework that we introduced earlier in thesis of Jaliya Ekanayake. Iterative extensions include a merge step; in-memory caching of static data (between iterations); cache aware hybrid scheduling using Azure Queues as well as a bulletin board (special table). Twister4Azure and MRRoles4Azure offer the familiar MapReduce programming model with fault tolerance features similar to traditional MapReduce and a decentralized control model without a master node implying no single point of failure. We test on data mining algorithms applied to Metagenomics and requiring parallel linear algebra in their compute intensive kernel.

MRRoles4Azure and an alpha version of Twister4Azure can be downloaded from http://salsahpc.indiana.edu/mapreduceroles4azure. Twister can be downloaded at http://www.iterativemapreduce.org/ .