Hadoop Benchmarking from a SAS Perspective

Paul.Kent@sas.com
VP Bigdata, SAS
@hornpolish
+1.919.749.6303

SAS High Performance analytics cut their teeth inside MPP DBMS architectures like Netezza, Asterdata, Teradata and Greenplum. Our accumulated learning over these environments caused us to prefer an "alongside" approach when co-habiting; one where the SAS mathematics could establish an MPI network of processes that live as peers alongside the network of DBMS processes.

Many of our early algorithms do not require grouped or ordered data (distribute by, local order by), but they do work most optimally when the amount of local data transferred from the DBMS slice to the SAS slice has a nice even spread from node to node, and it is our experience that reducing the number of OS processes that the data has to traipse through can be very beneficial.

The workload SAS would love to study/benchmark/improve as it relates to Hadoop can be characterized by the "variable selection" process typical of the early phase of many data mining processes… essentially "which of several thousand variables contribute the most effectively towards explaining the variance in a target variable" and as such are worthy of deeper modeling analysis.

This problem stresses several key components of the connectivity between HDFS and the analytics application.

1. The dataset buffer is very wide. 25000 variables is around 20x the number of columns permitted by many RDBMS whose bufferlength and pagelength constraints are typically near 65K bytes.
2. The analytics wishes to consume the entire dataset in a single pass. It will be important to transfer multiples of rows (boxcars is the term we use) into the application rather than one at a time.
3. It is very beneficial to influence the distribution of the blocks in the dataset when it is created – if we can lay out the allocation of primary and secondary blocks to have and nice even distribution across all data-nodes we find this a huge win. Alternatively, a technique that shuffled the minimum number of blocks at read time to achieve balance at the block level would be an interesting study

4. The application would like to mmap the data directly. Querying the underlying OS filenames containing the blocks has shown to be a huge win over reading/streaming/piping the reads thru a controlling process.
5. The result set can be written locally on the same distribution partition scheme, but one has to deal with an incomplete block per datanode possibility
6. The analytics will consume all CPU threads, and as much memory as the dataset requires per node. It might be interesting to apply more sophisticated workload management techniques

Sorry if this isn't completely centered around Hadoop and MapReduce. I come from a place of trying to map our analytics to a diverse set of technologies. We are making some impressive progress with Hadoop, and hope to have a coming out party later this year with even more SAS Software on Hadoop.

Looking forward to the metting and comparing notes with others implementing atop the Hadoop Framework

Paul