Large-Scale Merging of Histograms using Distributed In-Memory Computing

Primary author: J.Blomer
Co-author: G.Ganis

Most high-energy physics analysis jobs are embarrassingly parallel except for the final merging of the output objects, which are typically histograms. Currently, the merging of output histograms scales badly. The running time for distributed merging depends not only on the overall number of bins but also on the number partial histogram output files. That means, while the time to analyze data decreases linearly with the number of worker nodes, the time to merge the histograms in fact increases with the number of worker nodes.

On the grid, merging jobs that take a few hours are not unusual. In order to improve the situation, we present a distributed and decentral merging algorithm whose running time is independent of the number of worker nodes. We exploit full bisection bandwidth of local networks and we keep all intermediate results in memory. We present benchmarks from an implementation using the parallel ROOT facility (PROOF) and RAMCloud, a distributed key-value store that keeps all data in DRAM. Our results show that a real-world collection of ten thousand histograms with overall ten million non-zero bins can be merged in less than one minute.