Big Data Benchmarking for Yahoo!'s use of Hadoop Map-Reduce

{sriguru,nroberts}@yahoo-inc.com
Yahoo! Inc.

Background

This document is scoped around Yahoo!'s interest in Map-Reduce benchmarks, particularly as applicable to *Apache Hadoop* [1] deployments on large clusters of Linux servers.

Yahoo! runs Hadoop clusters with 4000+ commodity servers (~50K+ cores) that cater to a wide variety of applications including user click analysis, web content processing, ad serving optimization, etc. Given our large server footprint we are primarily interested in (understanding & improving) job throughput, cluster efficiency and operating costs. We are equally interested in (understanding, minimizing & in some cases guaranteeing) job processing latencies in multi-tenanted Hadoop deployments. We are also interested in minimizing the cost of moving data in and out of HDFS and across Hadoop clusters and maximizing storage efficiency on HDFS.

Workloads of Interest

We are interested in a gamut of workload types spanning data & compute intensive algorithms, embarrassingly parallel to partially parallelizable algorithms, native Map-Reduce & PIG/Hive applications, and so forth. We propose to share traces of our representative workloads with the research community along with a workload-replay benchmark named GridMix3 [2], to help study and achieve improvements across our workload types using relatively small (10x smaller than original cluster) test-bed clusters. GridMix3 is available in Apache Hadoop under the *contrib*/ folder. We would like to advocate the use of GridMix3 with a suitable set of workloads from across industry and academia as our preferred benchmarking methodology for Hadoop Map-Reduce. This is covered in the section "Benchmarking Methodology" below.

Metrics of Interest

We are interested in benchmarks as well as metrics (e.g., via instrumentation of Hadoop) that can help quantify and/or understand the following aspects of Hadoop performance.

From a Hadoop user's perspective:

1. Nature and causes of job latency variation across runs.
2. Ideal (range of) map & reduce slots for a given workload.
3. Break-up of Hadoop overheads that contributed to latency.

From a Hadoop provider's perspective:

1. Cluster scalability metrics & sources of bottlenecks.
2. Cluster & scheduling throughput & efficiency.
3. Power consumption. Power vs. performance trade-off costs.
4. Effect of straggler nodes on latency & throughput.

Benchmarking Methodology

GridMix3

GridMix3, available in Apache Hadoop under the *contrib*/ folder, was authored by Yahoo! developers for the purpose of capturing Hadoop Job characteristics and replaying them on a small test-bed cluster to address various benchmarking use cases including:

1. Comparing (performance of) two or more versions of Hadoop.
2. Measuring the effect of a Hadoop code or configuration change.
3. Comparing different hardware configs.
4. Reproducing performance & scalability bugs triggered by a specific workload pattern.

GridMix3 uses a *workload emulation* approach to enable study of Hadoop cluster and workload performance on a small sized testbed cluster. GridMix3 currently implements the following prominent features:

1. Multiple job submission modes: REPLAY (preserves submission timelines), STRESS (keeps cluster fully busy) and SERIAL (runs jobs sequentially)
2. SLEEPJOB mode: Creates tasks with zero I/O load. Useful for scalability measurements by typically running multiple TaskTrackers per node.
3. Multiple users & submission queues.
4. Emulation of bytes IN/OUT of HDFS and Local file system at a task level.
5. Emulation of average user CPU and JVM heap sizes at a task level.
6. Emulation of distributed cache, compressed input & highRAM jobs.

Methodology

The benchmarking methodology using GridMix3 is as follows:

1. Capture workload information from a Hadoop cluster of interest via *JobHistory* logs
2. Extract job characteristics and scale them down into a *trace file* via the *Rumen*[3] tool. Examples of job characteristics are: Job Submission Time, Number of Map & Reduce tasks per job, Job Config Parameters (hadoop-config.xml), HDFS Bytes In/Out per task, local bytes In/Out per task, average CPU & Memory utilization per task, etc.
3. Pick a small test bed cluster – typically less than a 10^{th} of the original cluster size -- and run GridMix3 with the above trace file. GridMix3 will generate random data and re-submit the jobs in the tracefile with custom Map and Reduce tasks that re-create load similar to the original workload.
4. Observe the system.

Steps 3 and 4, when run across multiple configurations of interest, address a majority of the above mentioned use cases.

Workload Traces

Yahoo! recently added log anonymization capability to Rumen and has made a set of anonymized traces available via our Webscope data sharing program. Our hope is that other organizations will also follow suit to release workload traces to help this work group create a "golden set of Hadoop traces" as the de facto Map-Reduce benchmark for real-world applications.

Summary

Yahoo! runs large (50k+ core) Hadoop clusters and is primarily interested in understanding and improving scalability, efficiency & operating costs of Hadoop Map-Reduce clusters.

To address the wide variety of applications using Map-Reduce, we recommend a benchmarking methodology that replays real workloads via GridMix3, in preference to creating synthetic benchmarks. We recommend creating a "golden set of workloads", originating from across industry and academia, as the de facto benchmarking suite for MapReduce.

References

[1] Apache Hadoop: http://hadoop.apache.org/

[2] GridMix3: http://hadoop.apache.org/mapreduce/docs/current/GridMix3.html

[3] Rumen: http://hadoop.apache.org/mapreduce/docs/r0.22.0/rumen.html