Gridmix

Table of contents

1. Overview

Gridmix is a benchmark for live clusters. It submits a mix of synthetic jobs, modeling a profile mined from production loads.

There exist three versions of the Gridmix tool. This document discusses the third (checked into contrib), distinct from the two checked into the benchmarks subdirectory. While the first two versions of the tool included stripped-down versions of common jobs, both were principally saturation tools for stressing the framework at scale. In support of a broader range of deployments and finer-tuned job mixes, this version of the tool will attempt to model the resource profiles of production jobs to identify bottlenecks, guide development, and serve as a replacement for the existing gridmix benchmarks.

2. Usage

To run Gridmix, one requires a job trace describing the job mix for a given cluster. Such traces are typically genenerated by Rumen (see related documentation). Gridmix also requires input data from which the synthetic jobs will draw bytes. The input data need not be in any particular format, as the synthetic jobs are currently binary readers. If one is running on a new cluster, an optional step generating input data may precede the run.

Basic command line usage:

```
bin/mapred org.apache.hadoop.mapred.gridmix.Gridmix [-generate <MiB>]
<iopath> <trace>
```

The `-generate` parameter accepts standard units, e.g. `100g` will generate 100 * 230 bytes. The <iopath> parameter is the destination directory for generated and/or the directory from which input data will be read. The <trace> parameter is a path to a job trace. The following configuration parameters are also accepted in the standard idiom, before other Gridmix parameters.

2.1. Configuration parameters

Parameter	Description	Notes
`gridmix.output.directory`	The directory into which output will be written. If specified, the `iopath` will be relative to this parameter.	The submitting user must have read/write access to this directory. The user should also be mindful of any quota issues that may arise during a run.

`gridmix.client.submit.th`	The number of threads submitting jobs to the cluster. This also controls how many splits will be loaded into memory at a given time, pending the submit time in the trace.	Splits are pregenerated to hit submission deadlines, so particularly dense traces may want more submitting threads. However, storing splits in memory is reasonably expensive, so one should raise this cautiously.
`gridmix.client.pending.q`	The depth of the queue of job descriptions awaiting split generation.	The jobs read from the trace occupy a queue of this depth before being processed by the submission threads. It is unusual to configure this.
`gridmix.min.key.length`	The key size for jobs submitted to the cluster.	While this is clearly a job-specific, even task-specific property, no data on key length is currently available. Since the intermediate data are random, memcomparable data, not even the sort is likely affected. It exists as a tunable as no default value is appropriate, but future versions will likely replace it with trace data.

3. Simplifying Assumptions

Gridmix will be developed in stages, incorporating feedback and patches from the community. Currently, its intent is to evaluate Map/Reduce and HDFS performance and not the layers on top of them (i.e. the extensive lib and subproject space). Given these two limitations, the following characteristics of job load are not currently captured in job traces and cannot be accurately reproduced in Gridmix.

Property	Notes
CPU usage	We have no data for per-task CPU usage, so we cannot attempt even an approximation. Gridmix tasks are never CPU bound independent of I/O, though this surely happens in practice.
Filesystem properties	No attempt is made to match block sizes, namespace hierarchies, or any property of input, intermediate, or output data other than the bytes/records consumed and emitted from a given task. This implies that some of the most

	heavily used parts of the system- the compression libraries, text processing, streaming, etc.- cannot be meaningfully tested with the current implementation.
I/O rates	The rate at which records are consumed/emitted is assumed to be limited only by the speed of the reader/writer and constant throughout the task.
Memory profile	No data on tasks' memory usage over time is available, though the max heap size is retained.
Skew	The records consumed and emitted to/from a given task are assumed to follow observed averages, i.e. records will be more regular than may be seen in the wild. Each map also generates a proportional percentage of data for each reduce, so a job with unbalanced input will be flattened.
Job failure	User code is assumed to be correct.
Job independence	The output or outcome of one job does not affect when or whether a subsequent job will run.

4. Appendix

Issues tracking the implementations of gridmix1, gridmix2, and gridmix3. Other issues tracking the development of Gridmix can be found by searching the Map/Reduce JIRA