Timeline Server: Next Generation Log Management in Hadoop

Job execution logs and profiles are important when troubleshooting Hadoop errors, tuning job performance, and planning cluster capacity. In the past, the Job History Server has been the primary source for this information, providing logs of important events in MapReduce job execution and associated profiling metrics. With the advent of YARN, which enables execution frameworks beyond MapReduce, the responsibilities of the Job History Server will eventually be taken over by the new Timeline Server. Currently, the main producer of Timeline Server data is Tez, an optimized execution engine used by the latest version of Hive, but support for Spark and MapReduce is under development.

The Timeline Server provides application-generic data and application-specific data through two different interfaces. Application-generic data relates to YARN resource management, such as the YARN containers used by each application and metrics related to those containers. This interface was previously supported by the generic Application History Server but is being subsumed under the umbrella of the Timeline Server.

For example, to list nodes consuming YARN resources, you can use the command:

[root@iteblog.com ~]$ yarn node -list

Id   State Http-Address   Running-Containers
s1.altiscale.com:45454 RUNNING s1.altiscale.com:8042 1

Then, to see a node's resource consumption:

[root@iteblog.com ~]$ yarn node -status s1.altiscale.com:45454

Node Report :
 Node-Id : s1.altiscale.com:45454
 Rack : /default-rack
 Node-State : RUNNING
 Node-Http-Address : s1.altiscale.com:8042
 Last-Health-Update : Tue 19/Aug/14 03:59:36:360PDT
 Health-Report :
 Containers : 1
 Memory-Used : 250MB
 Memory-Capacity : 2250MB
 CPU-Used : 1 vcores

CPU-Capacity : 8 vcores

Collection of application-specific data is a new feature of the Timeline Server motivated by YARN's support of a wide range of distributed execution engines (in addition to MapReduce). Each engine has different notions of execution components and workflow that don't always fit the model supported by the current Job History Server. The Timeline Server provides storage and retrieval of application-specific history for a wide range of execution engines.

Timeline Server data is structured around two concepts: Entities and Events. A REST API is available to query for Entities and Events using a variety of query filters. For example, querying a Tez DAG by an ID, i.e.

GET /ws/v1/timeline/TEZ_DAG_ID/dag_1406085934814_0004_1

Generates the following JSON response (details elided):

```
{ "entity" : "dag_1406085934814_0004_1",
  "entitytype" : "TEZ_DAG_ID",
  "events" : [
   { "eventinfo" : { ... },
     "eventtype" : "DAG_STARTED",
     "timestamp" : 1406087349969
   },
   ...
  ],
  "primaryfilters" : {
   "dagName" : [ "root_2014..." ],
   "user" : [ "root" ]
  },
  "otherinfo" : {
   "counters" : { ... },
   "dagPlan" : {
    "edges" : [ ... ],
    "vertices" : [ ...],
   }
  }
}
```

Entities describe major processing components or resources and associated attributes. The top-level structure of entities is standardized as in the example to include: an id ("entity"), a type ("entitytype"), associated "events", indexed attributes ("primary filters"), and unindexed attributes ("otherinfo"). The distinction between "primary filters" and "other info" is made to allow applications to choose which attributes should be supported by indexes in the underlying Timeline Server database. Entity types will vary by application. For example, MapReduce would include map/reduce task and task attempt entity types. Tez has DAGs, vertices, tasks, task attempts, etc. while Spark's entity types would include stages, tasks, and RDDs.

Events are time-stamped entries that record data related to important points in an Entity's lifetime. For example, significant events in Tez are initializing, starting or finishing a vertex, while Spark's significant events include submitting/completing stages and persisting/un-persisting an RDD.

We are deploying the Timeline Server at Altiscale as part of our log collection infrastructure. As Hadoop grows beyond its Map-Reduce roots, the ability to capture and analyze logs for a growing set of execution frameworks is critical to our ability to provide value to our users. In particular, the Timeline Server makes it easier for us to offer services, such as proactive job monitoring and tuning, in a scalable manner, as our user base increases and as newer tools are added to the Hadoop ecosystem.

Other Resources

Timeline Server is under active development and you can learn more from several resources.

The design of Timeline Server is described in the attachments at: https://issues.apache.org/jira/browse/YARN-1530.

A draft set of instructions for using the API is available at: https://issues.apache.org/jira/secure/attachment/12637256/YARN-1876.3.patch.

See https://issues.apache.org/jira/browse/TEZ-1066 for information on support for Tez.

The state of Spark support is being tracked at: https://issues.apache.org/jira/browse/SPARK-1537.

Information about Map Reduce support is available at: https://issues.apache.org/jira/browse/MAPREDUCE-5858.

Readers can checkout the Timeline Server as a technical preview in latest Hadoop. Instructions for configuration are available here: http://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/TimelineServer.html

4 / 4