Tez History/UI

Overview

Tez stores its History via a configurable History Logging Service. There are multiple implementations of this available as part of the Tez Framework:

- SimpleHistoryLoggingService
    - This writes the history into a log-file. The location is configurable. It can be logged as a file which will become part of the YARN Application's aggregated logs (i.e. within the Application Master's container logs) or to a defined location on a Distributed FileSystem such as HDFS.
    - This is more of a prototype (useful for quick testing/analysis) and not fully supported.
- ATSHistoryLoggingService and ATSV15HistoryLoggingService
    - These impls make use of YARN Timeline to store the history data
    - The ATSV15 impl (or to be more clear v1.5) makes use of enhancements done as part of YARN-4233. The enhancements are mainly around the use of a DistributedFileSystem for most of the data being written and stored instead of using LevelDB based storage.

The Tez UI currently only works with the YARN Timeline based stores. It will not be able to display any data for Tez DAGs that were configured to use the SimpleHistoryLoggingService. There is no current support for YARN Timeline v2 (being developed currently and potentially available in a future version of hadoop-3.x)

Tez History Data in YARN Timeline

YARN Timeline supports a notion of Entities. Entities are uniquely identified by a "Type" and an "Id". Entities can be related to other entities. Tez uses a set of Entity types:

- TEZ_APPLICATION
    - Application-level data
    - Contains configuration used to initialize the Tez session/AM.
- TEZ_APPLICATION_ATTEMPT
    - Application attempt specific data
- TEZ_DAG_ID
    - DAG-specific data
    - Contains the dagPlan info as well as additional configuration info
    - Also contains the final status of the DAG, counters, diagnostics, etc.
    - Has event information on when it started, finished, etc.
- TEZ_VERTEX_ID
    - Vertex-specific data
    - Also contains the final status of the Vertex, counters, diagnostics, etc.
    - Has event information on when it finished initializing, started, finished, etc.
- TEZ_TASK_ID
    - Task-specific data
    - Also contains the final status of the Task, counters, diagnostics, etc.
    - Has event information on when it started, finished, etc.
- TEZ_TASK_ATTEMPT_ID
    - TaskAttempt-specific data such as which container/node the attempt ran on.
    - Also contains the final status of the Attempt, counters, diagnostics, etc.
    - Has event information on when it started, finished, etc.

From a relationship perspective, you can think of TaskAttempt being a child of Task, Task a child of Vertex and Vertex a child of DAG. This relationship allows for pulling down all the necessary info for a given DAG.

Other then the above, Tez UI at times uses one more entity. And that is for displaying Hive query text in DAG details. When a hive query is run using latest version of Tez (0.7.1+), the query text would be added as part of the DAG JSON. For older versions when this is not available, the UI loads its from HIVE_QUERY_ID entity.

Tez History Data in YARN Timeline v1.5

Securing Access to Tez History Data

Securing access to data stored in Timeline is done via Timeline ACLs. For more details, please refer to the Access Controls Guide. The SimpleHistoryLoggingService provides no such protection apart from enforcing the necessary access controls on the configured DFS location where the logs are being written to.

Tez UI

Hosting the Tez UI

Hosting the Tez UI for Secure Clusters

Tez UI as an Ambari View

How to Build the Tez Ambari View

Tez UI and CORS (Cross-Origin Resource Sharing):

Tez UI is a pure web application that runs completely in the browser, and by default the browser restricts loading data from a different domain/origin. For instance if the UI is hosted at http://localhost:8080/tez-ui, the UI wont be able to load data from ATS that's running at http://localhost:8188. In such a case, to load data from ATS, you would have to enable Cross-Origin Resource Sharing at the ATS side. When ATS is provides history data (data for completed applications), Tez UI directly communicate with Tez Application Master via the RM web proxy for real-time data (running application). Hence you might hit a CORS issue with running application if CORS is not enabled in RM and the required configurations are not set.

UI errors for CORS:

In the latest UI, an yellow error bar will pop-up at the bottom with following errors if CORS is not enabled.

- When CORS is not enabled in ATS side: Adapter operation failed » Timeline server (ATS) is out of reach. Either it's down, or CORS is not enabled.
- When CORS is not enabled in RM side: Application Master (AM) is out of reach. Either it's down, or CORS is not enabled for YARN ResourceManager. & Resource Manager (RM) is out of reach. Either it's down, or CORS is not enabled.

How to confirm if its a CORS issue:

Click on the error message in the error-bar. It would display the error details with the REST endpoint URL that caused the error (In older versions of the UI, URL was part of the message). Try accessing this URL from a new tab, and if you are able to access it. Then you can confirm that its a CORS issue.

Fixing CORS issue:

By setting the following configurations and restarting the respective services you must be able to enable Cross-Origin Resource Sharing.

1. Enable CORS in timeline server side by following https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/TimelineServer.html. CORS configuration can be found under "Web and RPC Configuration".
2. Enable CORS in the RM side by following https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Enabling_CORS_support
3. As mentioned in https://tez.apache.org/tez-ui.html, ensure that the tez.tez-ui.history-url.base configuration is set to the address where UI is hosted?
    - If the UI is hosted in the default web server port (Port 80), and if the UI is accessed from the browser without the port number, then ensure that the tez.tez-ui.history-url.base values is set without the port number. In other words, its better to use http://tez.ui.origin.com /tez-ui/ than http://tez.ui.origin.com:80/tez-ui/.