

# How to Diagnose Tez App

## Yarn App Log

Tez will propagate most of the exceptions to client side. So please first check the client side log to look for any useful information about the errors. If the client side log does not convey much information, you can check for the yarn application logs.

Users can invoke command "yarn logs -applicationId {your\_app\_id}" to fetch the yarn app log to your local directory. This command is only available when the yarn log aggregation is enabled. For how to enable the yarn log aggregation, you can refer to <http://hortonworks.com/blog/simplifying-user-logs-management-and-access-in-yarn/>. If the log aggregation is not enabled, you may have to find the logs on each node manager machines.

After you get the yarn app logs. You can first check the Tez AM log, Tez AM is launched in the first yarn app containers, so it is located in folder like container\_{yarn\_app\_id}\_000001. Under this folder you may find the following files.

- syslog This is the log before AM properly started itself.
- syslog\_dag\_{yarn\_app\_id}\_{dag\_id} This is the log for each dag
- syslog\_dag\_{yarn\_app\_id}\_{dag\_id}\_post This is the log for each dag after dag is completed

So usually you only need to check the last dag log to find the errors.

## Tez-UI

Start from 0.6, tez has its proprietary ui to track the information of the running dag and history of completed dags. Most of time, users can see all the errors of dags on the tez-ui. Here's details of how to set up tez-ui. <http://tez.apache.org/tez-ui.html>

## Hive on Tez

First hive client ( hive cli/beeline will print some info if error happens on tez ), if it's not clear you may check the hive log file. Usually it is located in /tmp/{user}/hive.log. It may be also located in other places if you configure it in \$HIVE\_HOME/conf/hive-log4j.properties.

If the hive log still don't have enough information, you can refer the yarn log ( refer section Yarn App Log )

## Pig on Tez

## Cascading on Tez

TBD

### Enable logging for specific tasks in Tez

- "tez.task-specific.launch.cmd-opts.list" and "tez.task-specific.log.level" can be used to control log levels for specific tasks in the vertex of a DAG. Examples are given below
  - tez.task-specific.launch.cmd-opts.list="Map 1[0]" tez.task-specific.log.level="INFO;org.apache=DEBUG;"
    - Enable DEBUG log for "org.apache" for task-0 in Map 1. Rest of the logs would be at INFO level

### Profiling in Tez

It is possible to profile specific tasks in specific vertices in Tez. Examples are given below.

Example A: Use Xprof. Add "-Xprof" to tez.task.launch.cmd-opts, e.g. tez.task.launch.cmd-opts="-Xprof".

Example B. Use yourkit on specific tasks via "tez.task-specific.launch.cmd-opts.list" and "tez.task-specific.launch.cmd-opts". If your cluster doesn't have yourkit installed, you can config "tez.lib.uris" and "tez.lib.uris.classpath" so that the yourkit tar.gz is localized by yarn and added to the container classpath. With the below configuration, yourkit snapshot files will be created under yarn container log directory based on vertex name and task index. Then you can use "yarn logs" command to download the container log and exact the snapshot portion from the log.

- tez.task-specific.launch.cmd-opts.list
  - Specifies the tasks in different vertices where additional options have to be specified
  - Examples:
    - tez.task-specific.launch.cmd-opts.list="M5[0]" - Specifies task 0 in vertex M5
    - tez.task-specific.launch.cmd-opts.list="Map10[5,20]" - Specifies task 5,20 in vertex Map 10
    - tez.task-specific.launch.cmd-opts.list="M5[]" - Specifies all tasks in M5

- tez.task-specific.launch.cmd-opts.list="M5[1:3,10]" - Specifies 1,2,3,10 tasks in M5
    - However, defining partial ranges like "M5[:3]" is not yet supported.
  - tez.task-specific.launch.cmd-opts.list="M5[0];V2[10]" - Specifies task 0 in vertex M5 and task 10 in vertex V2
- tez.task-specific.launch.cmd-opts
  - Specifies the additional task specific JVM launch options that need to be added.
  - \_\_VERTEX\_NAME\_\_ and \_\_TASK\_INDEX\_\_ can be specified in the option, which would be replaced at runtime.
  - Examples:
    - tez.task-specific.launch.cmd-opts="-agentpath:yjp-2016.02-b46-linux.tar.gz/yjp-2016.02/bin/linux-x86-64/libyjpagent.so=disablej2ee,sampling,disabletracing,disablealloc,periodicperf=60,dir=<LOG\_DIR>/\_\_VERTEX\_NAME\_\_/\_\_TASK\_INDEX\_\_"
- tez.lib.uris
  - Add the yourkit tar.gz HDFS path to tez.lib.uris. Example: tez.lib.uris=/lib/tez-0.9.0-SNAPSHOT-minimal.tar.gz,/lib/yjp-2016.02-b46-linux.tar.gz
- tez.lib.uris.classpath
  - tez runtime tar.gz is defined in tez.lib.uris, config the classpath for the tez runtime tar.gz via tez.lib.uris.classpath=tez-0.9.0-SNAPSHOT-minimal.tar.gz/\*,tez-0.9.0-SNAPSHOT-minimal.tar.gz/lib/.