Running Nutch on Tez

Introduction

This tutorial covers running Nutch jobs on Apache Tez (instead of MapReduce).

> This tutorial is a work in progress and the document should be considered in DRAFT status. The work to evaluate Tez as an appropriate execution engine for Nutch jobs was initiated in December, 2020.
>
> Hadoop version: 3.1.4 released 03 Aug 2020
>
> Tez version: 0.10.0-SNAPSHOT (commit 849e1d7694cdfd2432d631830940bc95c6f26ead)
>
> Nutch version: 1.18-SNAPSHOT (commit 88a17f26b4160720bacb3ead1cad71ae24a559bc)

Audience

This tutorial will appeal to Nutch administrators looking to improve runtime speed whilst maintaining MapReduce's ability to scale to petabytes of data. Readers are encouraged to share their experienced using Nutch on Tez.

Related JIRA Tickets

NUTCH-2838 - Getting issue details... STATUS

What is Apache Tez?

Apache Tez is described as an application framework which allows for a complex directed-acyclic-graph (DAG) of tasks for processing data. It is currently built atop Apache Hadoop YARN.

The 2 main design themes for Tez are:

- **Empowering end users by:**
  - Expressive dataflow definition APIs
  - Flexible Input-Processor-Output runtime model
  - Data type agnostic
  - Simplifying deployment
- **Execution Performance**
  - Performance gains over Map Reduce
  - Optimal resource management
  - Plan reconfiguration at runtime
  - Dynamic physical data flow decisions

By allowing projects like Apache Hive and Apache Pig to run a complex DAG of tasks, Tez can be used to process data, that earlier took multiple MR jobs, now in a single Tez job as shown below.

blocked URLblocked URL

Configuring and Deploying Hadoop Services

Hadoop was configured and deployed in pseudo-distributed mode. The following assumes that you have already established a pseudo-distributed cluster and will make the following configuration changes before launching the new cluster.

yarn-site.xml

```xml
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,
HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>

<property>
  <name>yarn.nodemanager.resource.memory-mb</name>
  <value>8000</value>
  <description>Amount of physical memory, in MB, that can be allocated for containers.</description>
</property>

<property>
  <name>yarn.scheduler.minimum-allocation-mb</name>
  <value>500</value>
</property>

<property>
  <description>Indicate to clients whether Timeline service is enabled or not.
  If enabled, the TimelineClient library used by end-users will post entities
  and events to the Timeline server.</description>
  <name>yarn.timeline-service.enabled</name>
  <value>true</value>
</property>

<property>
  <description>The hostname of the Timeline service web application.</description>
  <name>yarn.timeline-service.hostname</name>
  <value>localhost</value>
</property>

<property>
  <description>Value must be the IP:PORT on which timeline server is running.</description>
  <name>yarn.timeline-service.webapp.address</name>
  <value>localhost:8188</value>
</property>

<property>
  <description>Enables cross-origin support (CORS) for web services where
  cross-origin web response headers are needed. For example, javascript making
  a web services request to the timeline server.</description>
  <name>yarn.timeline-service.http-cross-origin.enabled</name>
  <value>true</value>
</property>

<property>
  <description>Publish YARN information to Timeline Server</description>
  <name> yarn.resourcemanager.system-metrics-publisher.enabled</name>
  <value>true</value>
</property>
</configuration>
```

mapred-site.xml

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn-tez</value>
    </property>
    <property>
        <name>mapreduce.application.classpath</name>
        <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*<
/value>
    </property>
</configuration>
```

hadoop-env.sh

```
export JAVA_HOME=/path/to/JDK
export HADOOP_HOME=/path/to/hadoop
export TEZ_JARS=/path/to/tez/tez-dist/target/tez-0.10.1-SNAPSHOT
export TEZ_CONF_DIR=/path/to/tez/conf
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$TEZ_CONF_DIR:$TEZ_JARS/*:$TEZ_JARS/lib/*
```

You can then start all Hadoop services as follows

Start Hadoop Services

```
$HADOOP_HOME/sbin/start-dfs.sh
$HADOOP_HOME/sbin/start-yarn.sh
$HADOOP_HOME/bin/yarn --daemon start timelineserver
```

Configuring and Deploying Tez

First install Tez and ensure you can run the examples. Then progress to the following

copy tez-0.10.0-SNAPSHOT to HDFS

```
hadoop fs -copyFromLocal tez/tez-dist/target/tez-0.10.1-SNAPSHOT.tar.gz /apps/tez-0.10.1-SNAPSHOT/
```

You can then evolve the tez-site.xml below

tez-site.xml

```
<configuration>

<property>
  <name>tez.lib.uris</name>
  <value>${fs.defaultFS}/apps/tez-0.10.1-SNAPSHOT/tez-0.10.1-SNAPSHOT.tar.gz#tez,${fs.defaultFS}/apps/nutch
/apache-nutch-1.18-SNAPSHOT-bin.tar.gz#nutch</value>
</property>

<property>
  <name>tez.lib.uris.classpath</name>
  <value>./tez/tez-0.10.1-SNAPSHOT/*:./tez/tez-0.10.1-SNAPSHOT/lib/*:./nutch/apache-nutch-1.18-SNAPSHOT/*:.
/nutch/apache-nutch-1.18-SNAPSHOT/conf/*:./nutch/apache-nutch-1.18-SNAPSHOT/lib/*:./nutch/apache-nutch-1.18-
SNAPSHOT/plugins/*/*</value>
</property>

<property>
  <name>tez.use.cluster.hadoop-libs</name>
  <value>true</value>
</property>

<property>
  <name>plugin.folders</name>
  <value>nutch/apache-nutch-1.18-SNAPSHOT/plugins</value>
</property>

<property>
  <description>Enable Tez to use the Timeline Server for History Logging</description>
  <name>tez.history.logging.service.class</name>
  <value>org.apache.tez.dag.history.logging.ats.ATSHistoryLoggingService</value>
</property>

<property>
  <description>URL for where the Tez UI is hosted</description>
  <name>tez.tez-ui.history-url.base</name>
  <value>http://localhost:8080/tez-ui-0.10.1-SNAPSHOT</value>
</property>

<property>
   <name>tez.runtime.convert.user-payload.to.history-text</name>
   <value>true</value>
</property>

</configuration>
```

Deploy tez web application to Tomcat

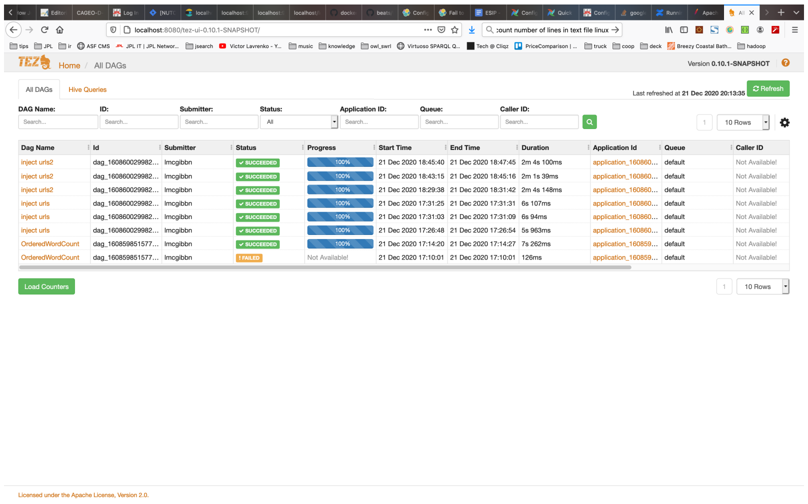
```
//assuming that Apache Tomcat is installed and running
cp $TEZ_HOME/tez-ui/target/tez-ui-0.10.1-SNAPSHOT.war $TOMCAT_HOME/webapps
```

Configuring and Deploying Nutch

Build Nutch and Copy to HDFS

```
cd $NUTCH_HOME && ant clean tar-bin
hadoop fs -copyFromLocal dist/apache-nutch-1.18-SNAPSHOT-bin.tar.gz /apps/nutch-1.18-SNAPSHOT
```

You can then run Nutch jobs as usual e.g. **nutch inject crawldb urls**. Once the job is submitted to YARN you can use the the tez-ui application deployed into Tomcat (should be at http://localhost:8080/tez-ui-0.10.1-SNAPSHOT/) to view Tez jobs. The screenshots below show some examples



localhost:8080/tez-ui-0.10.1-SNAPSHOT/

TEZ Home / All DAGs — Version 0.10.1-SNAPSHOT

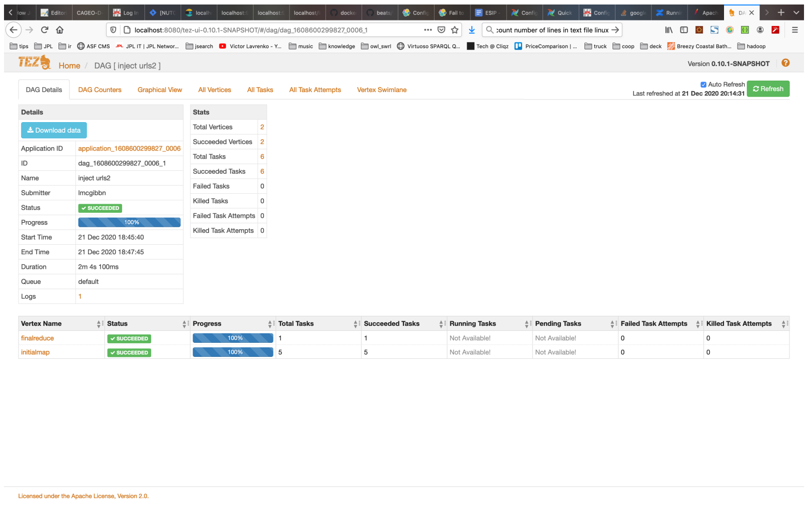
All DAGs | Hive Queries

Last refreshed at **21 Dec 2020 20:13:35** Refresh

DAG Name: | ID: | Submitter: | Status: All | Application ID: | Queue: | Caller ID: — 10 Rows

Dag Name	Id	Submitter	Status	Progress	Start Time	End Time	Duration	Application Id	Queue	Caller ID
inject urls2	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 18:45:40	21 Dec 2020 18:47:45	2m 4s 100ms	application_160860...	default	Not Available!
inject urls2	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 18:43:15	21 Dec 2020 18:45:16	2m 1s 39ms	application_160860...	default	Not Available!
inject urls2	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 18:29:38	21 Dec 2020 18:31:42	2m 4s 148ms	application_160860...	default	Not Available!
inject urls	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 17:31:25	21 Dec 2020 17:31:31	6s 107ms	application_160860...	default	Not Available!
inject urls	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 17:31:03	21 Dec 2020 17:31:09	6s 94ms	application_160860...	default	Not Available!
inject urls	dag_160860029982...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 17:26:48	21 Dec 2020 17:26:54	5s 963ms	application_160860...	default	Not Available!
OrderedWordCount	dag_160859851577...	lmcgibbn	✓ SUCCEEDED	100%	21 Dec 2020 17:14:20	21 Dec 2020 17:14:27	7s 262ms	application_160859...	default	Not Available!
OrderedWordCount	dag_160859851577...	lmcgibbn	! FAILED	Not Available!	21 Dec 2020 17:10:01	21 Dec 2020 17:10:01	126ms	application_160859...	default	Not Available!

Load Counters — 10 Rows



localhost:8080/tez-ui-0.10.1-SNAPSHOT/#/dag/dag_1608600299827_0006_1

TEZ Home / DAG [inject urls2] — Version 0.10.1-SNAPSHOT

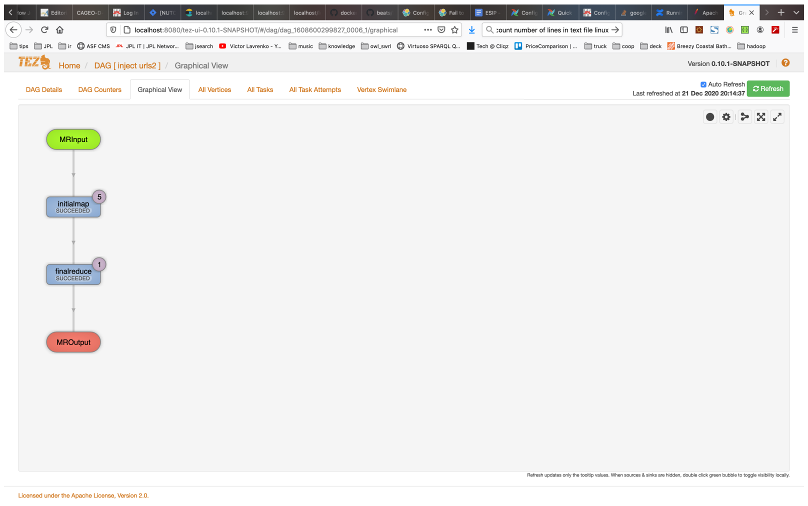
DAG Details | DAG Counters | Graphical View | All Vertices | All Tasks | All Task Attempts | Vertex Swimlane

Auto Refresh — Last refreshed at **21 Dec 2020 20:14:31** Refresh

Details	
Download data	
Application ID	application_1608600299827_0006
ID	dag_1608600299827_0006_1
Name	inject urls2
Submitter	lmcgibbn
Status	✓ SUCCEEDED
Progress	100%
Start Time	21 Dec 2020 18:45:40
End Time	21 Dec 2020 18:47:45
Duration	2m 4s 100ms
Queue	default
Logs	1

Stats	
Total Vertices	2
Succeeded Vertices	2
Total Tasks	6
Succeeded Tasks	6
Failed Tasks	0
Killed Tasks	0
Failed Task Attempts	0
Killed Task Attempts	0

Vertex Name	Status	Progress	Total Tasks	Succeeded Tasks	Running Tasks	Pending Tasks	Failed Task Attempts	Killed Task Attempts
finalreduce	✓ SUCCEEDED	100%	1	1	Not Available!	Not Available!	0	0
initialmap	✓ SUCCEEDED	100%	5	5	Not Available!	Not Available!	0	0



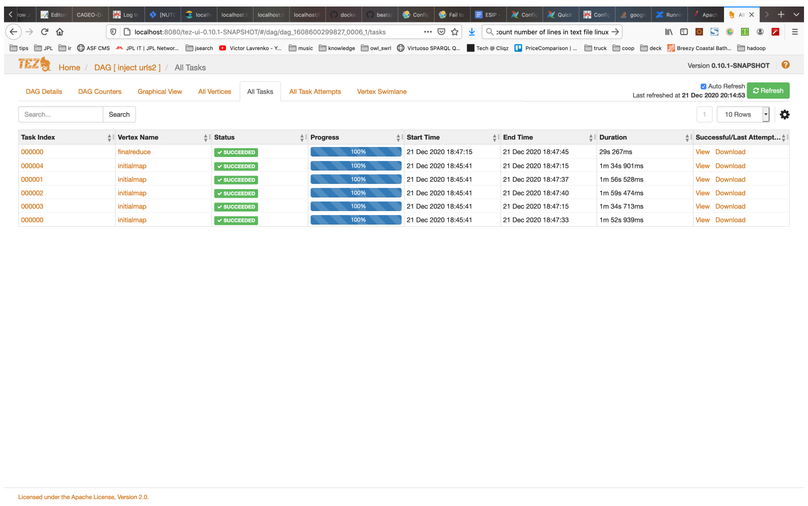
localhost:8080/tez-ui-0.10.1-SNAPSHOT/#/dag/dag_1608600299827_0006_1/graphical

TEZ Home / DAG [inject urls2] / Graphical View — Version 0.10.1-SNAPSHOT

DAG Details | DAG Counters | Graphical View | All Vertices | All Tasks | All Task Attempts | Vertex Swimlane

Auto Refresh — Last refreshed at **21 Dec 2020 20:14:37** Refresh

Refresh updates only the tooltip values. When sources & sinks are hidden, double click green bubble to toggle visibility locally.


localhost:8080/tez-ui-0.10.1-SNAPSHOT/#/dag/dag_1608600299827_0006_1/tasks
Home / DAG [inject urls2] / All Tasks
Version 0.10.1-SNAPSHOT
DAG Details
DAG Counters
Graphical View
All Vertices
All Tasks
All Task Attempts
Vertex Swimlane
Auto Refresh
Refresh
Last refreshed at 21 Dec 2020 20:14:53
Search
10 Rows

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration	Successful/Last Attempt...
000000	finalreduce	✓ SUCCEEDED	100%	21 Dec 2020 18:47:15	21 Dec 2020 18:47:45	29s 267ms	View Download
000004	initialmap	✓ SUCCEEDED	100%	21 Dec 2020 18:45:41	21 Dec 2020 18:47:15	1m 34s 901ms	View Download
000001	initialmap	✓ SUCCEEDED	100%	21 Dec 2020 18:45:41	21 Dec 2020 18:47:37	1m 56s 528ms	View Download
000002	initialmap	✓ SUCCEEDED	100%	21 Dec 2020 18:45:41	21 Dec 2020 18:47:40	1m 59s 474ms	View Download
000003	initialmap	✓ SUCCEEDED	100%	21 Dec 2020 18:45:41	21 Dec 2020 18:47:15	1m 34s 713ms	View Download
000000	initialmap	✓ SUCCEEDED	100%	21 Dec 2020 18:45:41	21 Dec 2020 18:47:33	1m 52s 939ms	View Download

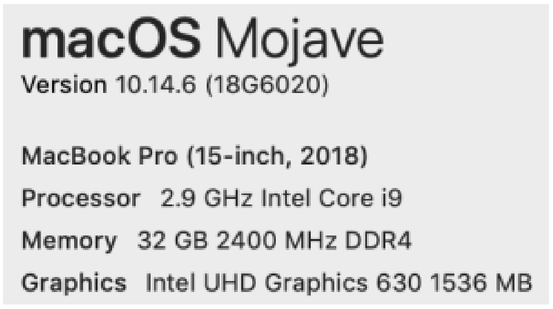
Licensed under the Apache License, Version 2.0.

Evaluating Tez as a Replacement for MapReduce

The following content relates to ongoing experiments which have been run by members of the Nutch community.

Experiments

Running the Injector job on Tez



macOS Mojave
Version 10.14.6 (18G6020)

MacBook Pro (15-inch, 2018)
Processor 2.9 GHz Intel Core i9
Memory 32 GB 2400 MHz DDR4
Graphics Intel UHD Graphics 630 1536 MB

Run #	YARN Engine	# of URLs	Elapsed Time
1	MapReduce	11523	`00:00:34`
2	MapReduce	11523	`00:00:32`
3	MapReduce	11523	`00:00:34`
4	Tez	11523	`00:00:42`
5	Tez	11523	`00:00:13`
6	Tez	11523	`00:00:14`
7	MapReduce	15763469	00:03:21
8	MapReduce	15763469	00:03:13
9	MapReduce	15763469	00:02:38
10	MapReduce	15763469	00:02:37
11	MapReduce	15763469	00:02:48
12	Tez	15763469	00:02:14
13	Tez	15763469	00:02:10
14	Tez	15763469	00:02:13

Both Tez and MapReduce appear to eventually gain performance improvements after a few runs. For shorter tasks we already find a performance improvement because of the default ***tez.am.container.reuse.enabled=true*** configuration property. This especially applies for shorter runtimes, where e.g. JVM startup time/warmup really counts. The above runtimes represent a cold -> warm pattern. Clearly after warm up, Tez appears to offer significant runtime improvements over MapReduce. This is very promising however much more experimentation is required.

Running the Generator job on Tez

Run #	YARN Engine	# of URLS	Elapsed Time
1	MapReduce	11322	00:01:19
2	MapReduce	11322	00:01:18
3	MapReduce	11322	00:01:22
4	MapReduce	11322	00:01:23
5	Tez	N/A	N/A
6	Tez	N/A	N/A
7	Tez	N/A	N/A
8	Tez	N/A	N/A

As of 22 Dec 2020 it was discovered that the Generator job is incompatible with Tez. The job execution log below details the outcome.

Generator job incompatible with Tez

```
$ nutch generate crawldb segments5
...
2020-12-22 10:17:05,168 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
2020-12-22 10:17:05,759 INFO crawl.Generator: Generator: starting at 2020-12-22 10:17:05
2020-12-22 10:17:05,759 INFO crawl.Generator: Generator: Selecting best-scoring urls due for fetch.
2020-12-22 10:17:05,759 INFO crawl.Generator: Generator: filtering: true
2020-12-22 10:17:05,759 INFO crawl.Generator: Generator: normalizing: true
2020-12-22 10:17:05,955 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-22 10:17:06,071 INFO client.AHSProxy: Connecting to Application History server at localhost/127.0.0.1:
10200
2020-12-22 10:17:06,308 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn
/staging/lmcgibbn/.staging/job_1608661005352_0001
2020-12-22 10:17:07,115 INFO input.FileInputFormat: Total input files to process : 1
2020-12-22 10:17:07,161 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-22 10:17:07,387 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1608661005352_0001
2020-12-22 10:17:07,388 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-22 10:17:07,531 INFO client.YARNRunner: Number of stages: 2
2020-12-22 10:17:07,597 INFO conf.Configuration: resource-types.xml not found
2020-12-22 10:17:07,598 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-22 10:17:07,809 INFO counters.Limits: Counter limits initialized with parameters:  GROUP_NAME_MAX=256,
MAX_GROUPS=500, COUNTER_NAME_MAX=64, MAX_COUNTERS=1200
2020-12-22 10:17:07,809 INFO counters.Limits: Counter limits initialized with parameters:  GROUP_NAME_MAX=256,
MAX_GROUPS=500, COUNTER_NAME_MAX=64, MAX_COUNTERS=120
2020-12-22 10:17:07,809 INFO client.TezClient: Tez Client Version: [ component=tez-api, version=0.10.1-
SNAPSHOT, revision=849e1d7694cdfd2432d631830940bc95c6f26ead, SCM-URL=scm:git:https://gitbox.apache.org/repos/asf
/tez.git, buildTime=2020-12-17T01:41:13Z, buildUser=lmcgibbn, buildJavaVersion=1.8.0_221 ]
2020-12-22 10:17:07,825 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-22 10:17:07,825 INFO client.AHSProxy: Connecting to Application History server at localhost/127.0.0.1:
10200
2020-12-22 10:17:07,826 INFO client.TezClient: Submitting DAG application with id:
application_1608661005352_0001
2020-12-22 10:17:07,828 INFO client.TezClientUtils: Using tez.lib.uris value from configuration:
hdfs://localhost:9000/apps/tez-0.10.1-SNAPSHOT/tez-0.10.1-SNAPSHOT.tar.gz#tez,hdfs://localhost:9000/apps/nutch
/apache-nutch-1.18-SNAPSHOT-bin.tar.gz#nutch
2020-12-22 10:17:07,828 INFO client.TezClientUtils: Using tez.lib.uris.classpath value from configuration: ./tez
/tez-0.10.1-SNAPSHOT/*:./tez/tez-0.10.1-SNAPSHOT/lib/*:./nutch/apache-nutch-1.18-SNAPSHOT/*:./nutch/apache-
nutch-1.18-SNAPSHOT/conf/*:./nutch/apache-nutch-1.18-SNAPSHOT/lib/*:./nutch/apache-nutch-1.18-SNAPSHOT/plugins/*
/*
2020-12-22 10:17:07,842 INFO client.TezClient: Tez system stage directory hdfs://localhost:9000/tmp/hadoop-yarn
/staging/lmcgibbn/.staging/job_1608661005352_0001/.tez/application_1608661005352_0001 doesn't exist and is
created
2020-12-22 10:17:08,413 INFO client.TezClient: Submitting DAG to YARN,
applicationId=application_1608661005352_0001, dagName=generate: select from crawldb
2020-12-22 10:17:08,787 INFO impl.YarnClientImpl: Submitted application application_1608661005352_0001
2020-12-22 10:17:08,790 INFO client.TezClient: The url to track the Tez AM: http://localhost:8088/proxy
/application_1608661005352_0001/
^[[C2020-12-22 10:17:50,693 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-22 10:17:50,693 INFO client.AHSProxy: Connecting to Application History server at localhost/127.0.0.1:
10200
2020-12-22 10:17:50,720 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy
/application_1608661005352_0001/
2020-12-22 10:17:50,721 INFO mapreduce.Job: Running job: job_1608661005352_0001
2020-12-22 10:17:51,729 INFO mapreduce.Job: Job job_1608661005352_0001 running in uber mode : false
2020-12-22 10:17:51,731 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-22 10:17:56,764 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-22 10:17:56,766 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-22 10:17:56,768 INFO mapreduce.Job: Job job_1608661005352_0001 completed successfully
2020-12-22 10:17:56,775 INFO mapreduce.Job: Counters: 0
2020-12-22 10:17:56,776 INFO crawl.Generator: Generator: number of items rejected during selection:
2020-12-22 10:17:56,806 WARN crawl.Generator: Generator: 0 records selected for fetching, exiting ...
```

Observed Issues

1. When using Tez, counters are not populated. This makes sense as all existing counters are created using MapReduce framework Context objects. This presents a major issue. Counters are a requirement to have as they are key to regular inspections of ongoing crawls, finding errors and debugging. The org.apache.tez.common.counters package may offer a equivalent replacement but this has still to be investigated.
2. As of 22 Dec 2020 it was discovered that the Generator job is incompatible with Tez. Again there are no counters so this could be the expected behaviour.