SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)

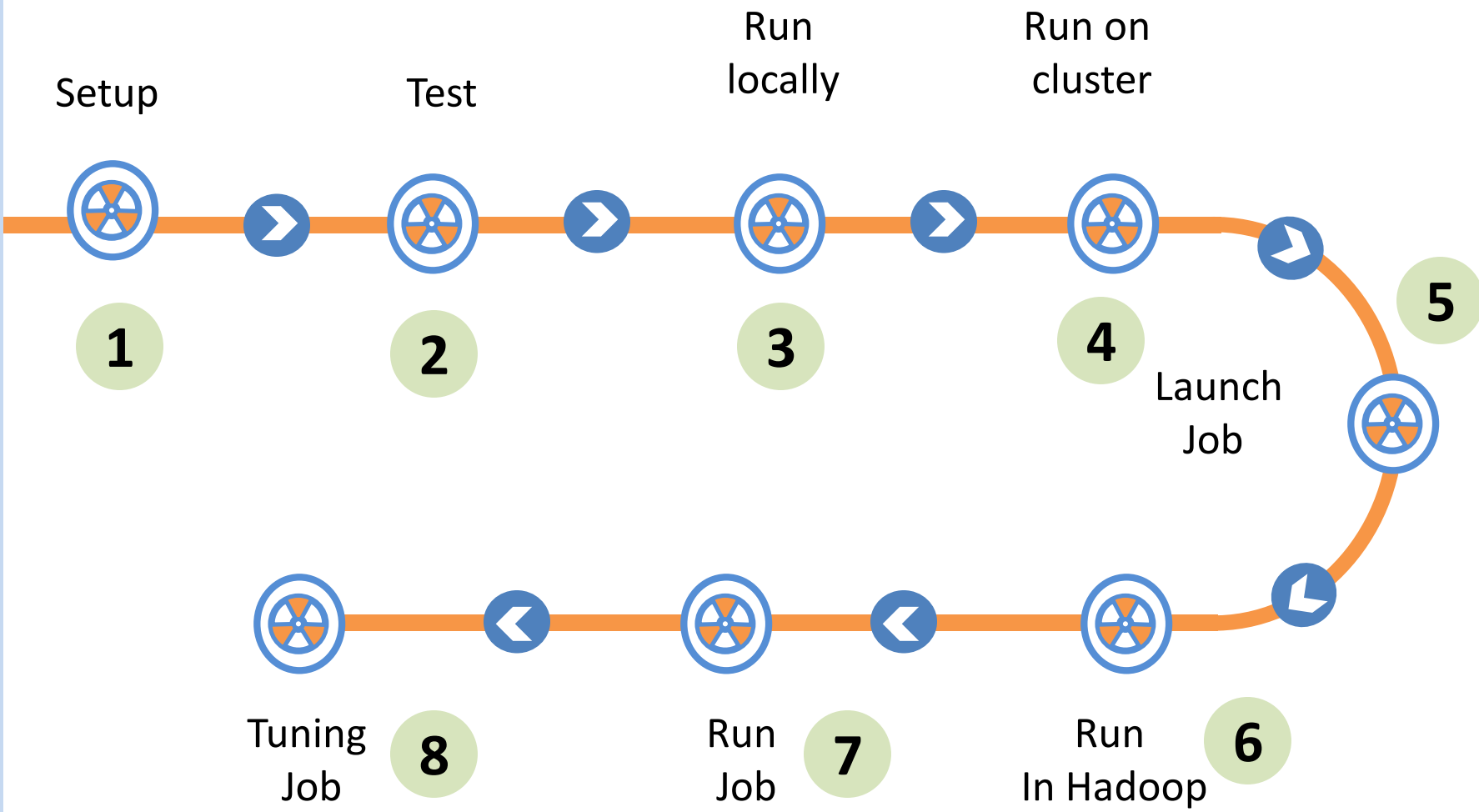
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Topic: Developing a MapReduce Application

COURSE	UNIT - II	CLASS
16CA917 Big Data Analytics	Hadoop	V Semester / III MCA

Procedure


Setup
Test
Run locally
Run on cluster
1
2
3
4
5
Launch Job
Tuning Job
8
Run Job
7
Run In Hadoop
6

Procedure

1. Set up and configure the development environment
2. Writing unit test for both map and reduce function
3. Running locally on test data using local job runner (Tool Interface)
4. Running on a cluster - Packaging into JAR file
5. Launching a job – run driver by specifying cluster
6. *% hadoop jar job.jar v3.MaxTemperatureDriver -conf conf/hadoop-cluster.xml \input/ncdc/all max-temp*
7. runJob() method on JobClient launches the job and polls for progre
8. Tuning a job

Setup & Configuration

Components in Hadoop are configured using Hadoop's own configuration API. (found in org.apache.hadoop.conf package)

```
<?xml version="1.0"?>
<configuration>
  <property>
    <name>color</name>
    <value>yellow</value>
    <description>Color</description>
  </property>

  <property>
    <name>size</name>
    <value>10</value>
    <description>Size</description>
  </property>
```

Setup & Configuration

Components in Hadoop are configured using Hadoop's own configuration API. (found in org.apache.hadoop.conf package)

```
<property>
  <name>weight</name>
  <value>heavy</value>
  <final>true</final>
  <description>Weight</description>
</property>

<property>
  <name>size-weight</name>
  <value>${size},${weight}</value>
  <description>Size and weight</description>
</property>
</configuration>
```

Setup & Configuration

Assuming this configuration file is in a file called *configuration-1.xml*, we can access its properties using a piece of code

```
Configuration conf = new Configuration();
conf.addResource("configuration-1.xml");
assertThat(conf.get("color"), is("yellow"));
assertThat(conf.getInt("size", 0), is(10));
assertThat(conf.get("breadth", "wide"), is("wide"));
```

Setup & Configuration

- Conf directory contains three configuration files: hadoop-local.xml, hadoop-localhost.xml, and hadoop-cluster.xml
- hadoop-local.xml file contains the default Hadoop configuration for the default filesystem and the jobtracker

```
<?xml version="1.0"?>
<configuration>

  <property>
    <name>fs.default.name</name>
    <value>file:///</value>
  </property>

  <property>
    <name>mapred.job.tracker</name>
    <value>local</value>
  </property>

</configuration>
```

Setup & Configuration

- hadoop-localhost.xml point to a namenode and a jobtracker both running on localhost
- hadoop-cluster.xml contains details of the cluster's namenode and jobtracker addresses

```
<?xml version="1.0"?>
<configuration>

  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost/</value>
  </property>

  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:8021</value>
  </property>

</configuration>
```

MapReduce WI

Cluster Summary (Heap Size is 53.75 MB/888.94 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
53	30	2	11	88	88	16.00	0

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_200904110811_0002	NORMAL	root	Max temperature	47.52%	101	48	15.25%	30	0	NA

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_200904110811_0001	NORMAL	gonzo	word count	100.00%	14	14	100.00%	30	30	NA

Failed Jobs

none

Tuning a Job

Area	Best practice
Number of mappers	How long are your mappers running for? If they are only running for a few seconds on average, then you should see if there's a way to have fewer mappers and make them all run longer, a minute or so, as a rule of thumb. The extent to which this is possible depends on the input format you are using.
Number of reducers	For maximum performance, the number of reducers should be slightly less than the number of reduce slots in the cluster. This allows the reducers to finish in one wave and fully utilizes the cluster during the reduce phase.
Combiners	Can your job take advantage of a combiner to reduce the amount of data in passing through the shuffle?

Tuning a Job

Intermediate compression	Job execution time can almost always benefit from enabling map output compression.
Custom serialization	If you are using your own custom `Writable` objects, or custom comparators, then make sure you have implemented `RawComparator`.
Shuffle tweaks	The MapReduce shuffle exposes around a dozen tuning parameters for memory management, which may help you eke out the last bit of performance.

Assessment

Primary interface for a user to describe a MapReduce job

A. JobConfig B. JobConf C. Job_Config D. Job_Configure

Run MapReduce first at

A. Cluster B. Locally C. Next node in network D. Any node

References

- Tom White, “ Hadoop: The Definitive Guide” Third Edition, O’reilly Media, 2012

https://www.informit.com/articles/article.aspx?p=2017060

https://www.youtube.com/playlist?list=PLf0swTFhTI8p7ZYBB5DmRHK3L6aq8Mvcv

https://energie.labs.fhv.at/~repe/bigdata/introduction-to-big-data-projects/tutorials/developing-a-mapreduce-application/


Thank you