OOZIE

Orchestrating your Hadoop jobs

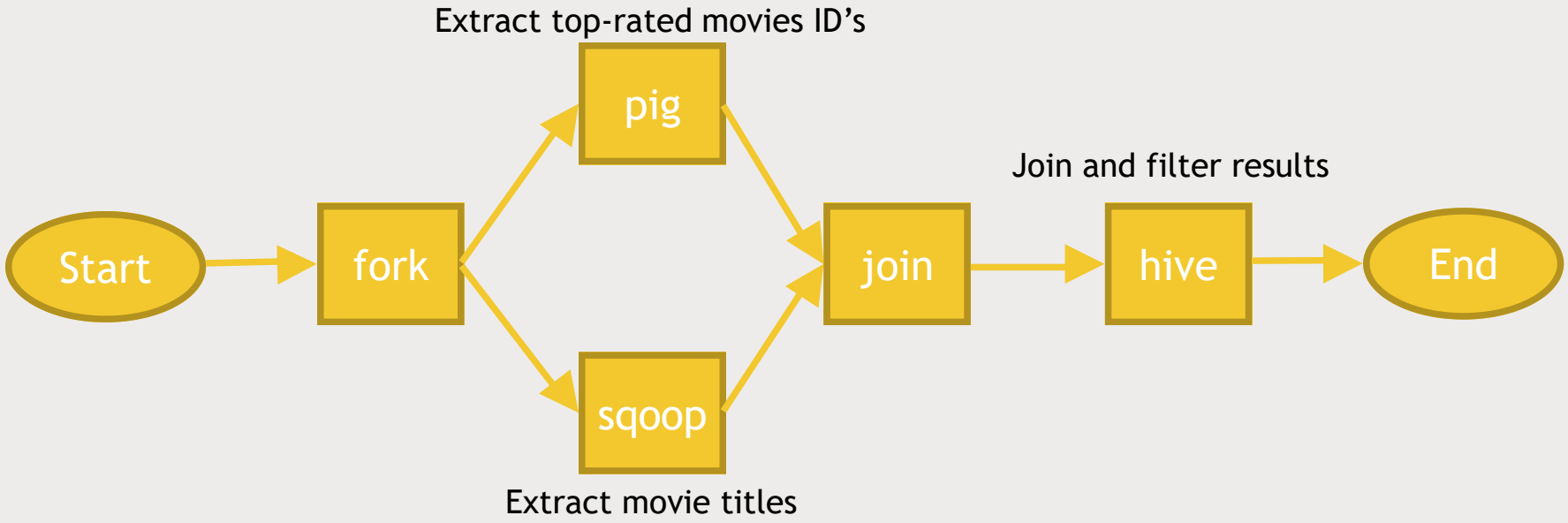
What is Oozie?

- Burmese for “elephant keeper”
- A system for running and scheduling Hadoop tasks

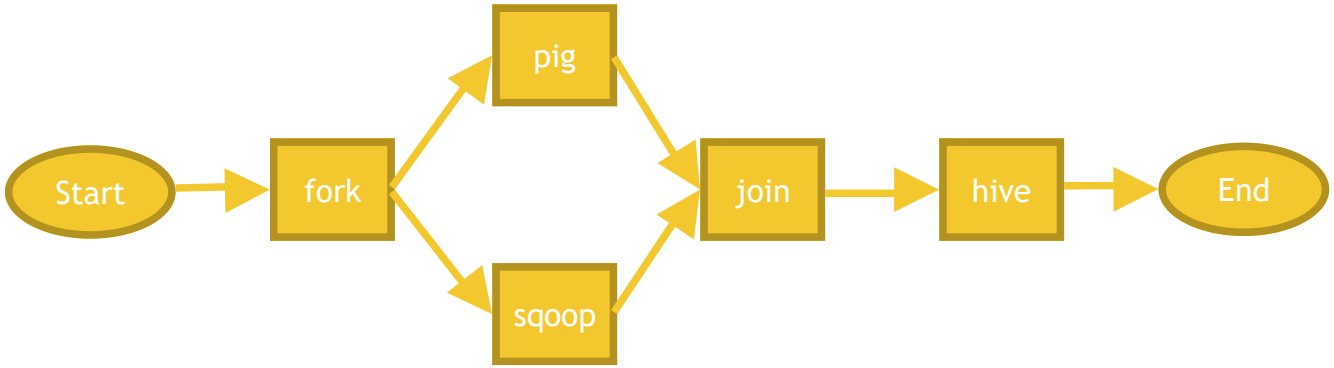
Workflows

- A multi-stage Hadoop job
  - *Might chain together MapReduce, Hive, Pig, sqoop, and distcp tasks*
  - *Other systems available via add-ons (like Spark)*
- A workflow is a Directed Acyclic Graph of actions
  - *Specified via XML*
  - *So, you can run actions that don't depend on each other in parallel.*

Workflow example



Workflow XML structure




```
<?xml version="1.0" encoding="UTF-8"?>
<workflow-app xmlns="uri:oozie:workflow:0.2" name="top-movies">
    <start to="fork-node"/>

    <fork name="fork-node">
        <path start="sqoop-node" />
        <path start="pig-node" />
    </fork>

    <action name="sqoop-node">
        <sqoop xmlns="uri:oozie:sqoop-action:0.2">

        … sqoop configuration here …

        </sqoop>
        <ok to="joining"/>
        <error to="fail"/>
    </action>

    <action name="pig-node">
        <pig>

         … pig configuration here …

        </pig>
        <ok to="joining"/>
        <error to="fail"/>
    </action>

    <join name="joining" to="hive-node"/>

    <action name="hive-node">
        <hive xmlns="uri:oozie:hive-action:0.2">

        … hive configuration here …

        </hive>
        <ok to="end"/>
        <error to="fail"/>
    </action>

    <kill name="fail">
        <message>Sqoop failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
    </kill>
    <end name="end"/>
</workflow-app>
```

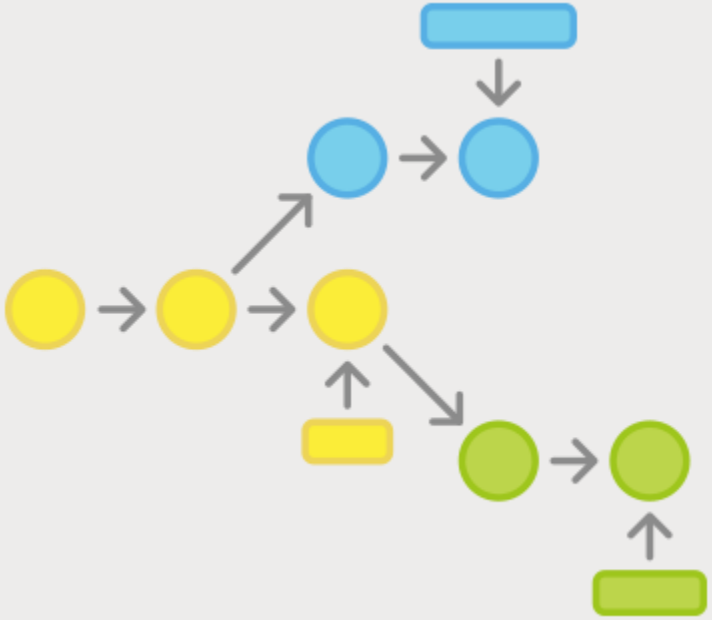
Steps to set up a workflow in Oozie

- Make sure each action works on its own
- Make a directory in HDFS for your job
- Create your workflow.xml file and put it in your HDFS folder
- Create job.properties defining any variables your workflow.xml needs
  - *This goes on your local filesystem where you'll launch the job from*
  - *You could also set these properties within your XML.*

```
nameNode=hdfs://sandbox.hortonworks.com:8020
jobTracker=http://sandbox.hortonworks.com:8050
queueName=default
oozie.use.system.libpath=true
oozie.wf.application.path=${nameNode}/user/maria_dev
```

Running a workflow with Oozie

- oozie job --oozie http://localhost:11000/oozie -config /home/maria_dev/job.properties –run
- Monitor progress at http://127.0.0.1:11000/oozie



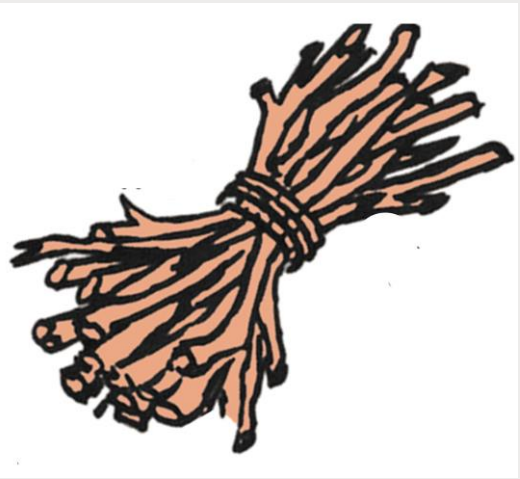
Oozie Coordinators

- Schedules workflow execution
- Launches workflows based on a given start time and frequency
- Will also wait for required input data to become available
- Run in exactly the same way as a workflow

```
<coordinator-app xmlns = "uri:oozie:coordinator:0.2" name = “sample coordinator" frequency = "5 * * * *" start = "2016-00-18T01:00Z" end = "2025-12-31T00:00Z"" timezone = "America/Los_Angeles">
        <controls>
                <timeout>1</timeout>
                 <concurrency>1</concurrency>
                <execution>FIFO</execution>
                <throttle>1</throttle>
        </controls>
        <action>
                <workflow>
                        <app-path>pathof_workflow_xml/workflow.xml</app-path>
                </workflow>
        </action>
</coordinator-app>
```

Oozie bundles

- New in Oozie 3.0
- A bundle is a collection of coordinators that can be managed together
- Example: you may have a bunch of coordinators for processing log data in various ways
  - *By grouping them in a bundle, you could suspend them all if there were some problem with log collection*



Let’s set up a simple workflow in Oozie.

- We’ll get movielens back into MySQL if it’s not still there
- Write a Hive script to find all movies released before 1940
- Set up an Oozie workflow that uses sqoop to extract movie information from MySQL, then analyze it with Hive