Oozie

What is Oozie?

workflow.xml

job.properties

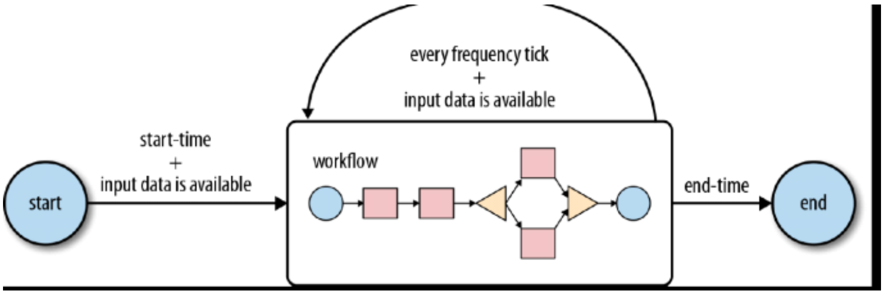
Oozie concepts

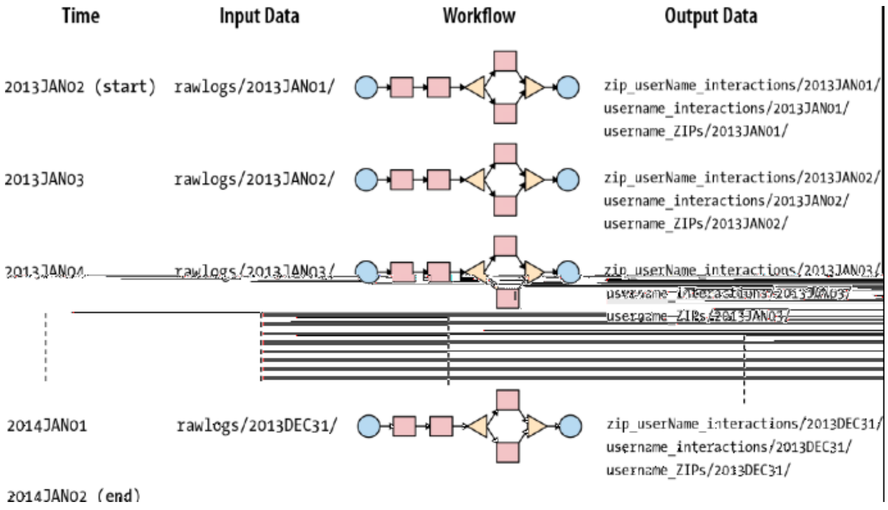
Oozie application

Oozie workflow

Oozie coordinators

-
-
-
-
    ○



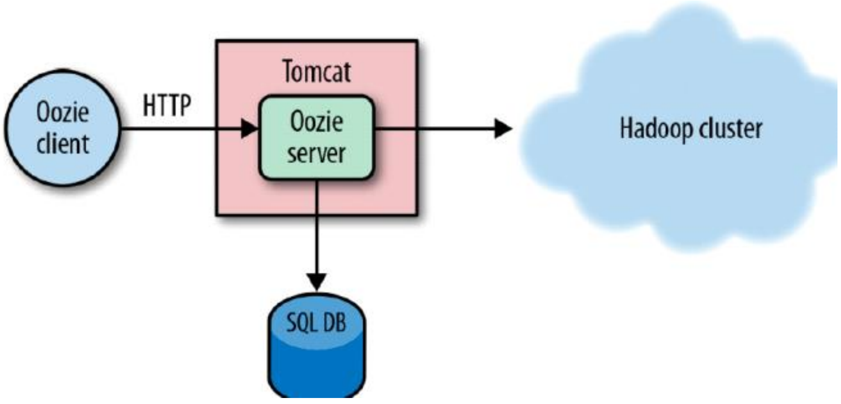


Time	Input Data	Workflow	Output Data
2013JAN02 (start)	rawlogs/2013JAN01/		zip_userName_interactions/2013JAN01/ username_interactions/2013JAN01/ username_ZIPs/2013JAN01/
2013JAN03	rawlogs/2013JAN02/		zip_userName_interactions/2013JAN02/ username_interactions/2013JAN02/ username_ZIPs/2013JAN02/
2013JAN04	rawlogs/2013JAN03/		zip_userName_interactions/2013JAN03/ username_interactions/2013JAN03/ username_ZIPs/2013JAN03/
2014JAN01	rawlogs/2013DEC31/		zip_userName_interactions/2013DEC31/ username_interactions/2013DEC31/ username_ZIPs/2013DEC31/
2014JAN02 (end)			

Oozie bundles

-
-

Oozie architecture



-
-
-

c) It basically manages the scheduling of Oozie jobs

c) It Manages the execution of Oozie jobs

Oozie deployment

-
-
    -
    -
    -

-
-
-

-
- oozie admin –share libupdate is the way to do it and ensures that Oozie uses the latest version.

Chapter 4

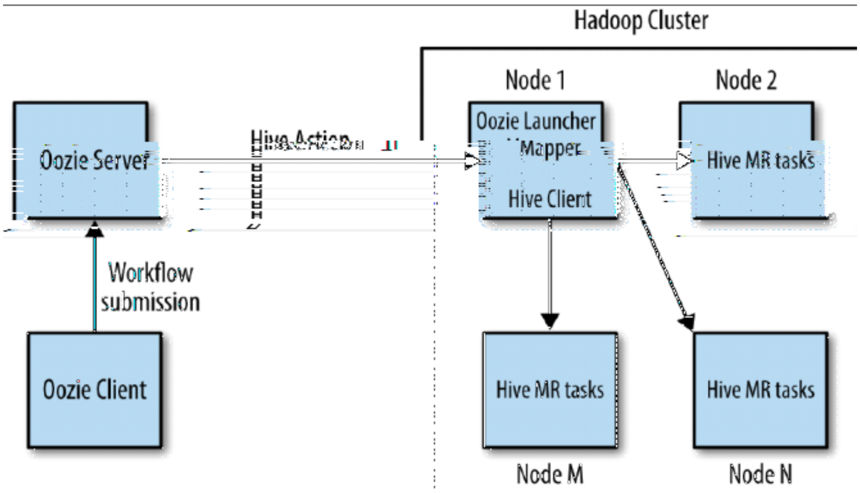
Ooze workflow actions

Hadoop DistCp, for example, is a common tool used to pull data from S3

Workflow actions are a set of DAG actions, mentioned in workflow.xml file

Action- execution model in Oozie : How are actions launched in Oozie?

- When a user submits a oozie job through Oozie CLI or any other client tools, the instruction is then passed onto the Oozie se rver.
- The Oozie server does not launch the action jobs (Hive, shell etc.) on its own server. Instead the Oozie server launches laun cher jobs on Hadoop cluster for every action
- Each Launcher job occupies space on the Hadoop cluster
- Each launcher job in turn launches the appropriate action job on Hadoop (Hadoop job) and waits for all the hadoop jobs to com plete.
- Each launcher job is a separate self-contained application that can run on any node in the hadoop cluster.
- Hence with this mechanism, Oozie server does not take the headache of running all Hadoop jobs. But the output of most hadoop jobs are redirected to the STDOUT/STDERR of Oozie server, and hence the outputs of these hadoop jobs can be viewed on Oozie web interface.



Action components

Each Action has a name property and 2 other routes :

- OK <ok>
- Error <error>

Depending on if the action fails/sucedes it goes to the corresponding next action

Oozie documentation :

https://oozie.apache.org/docs/4.1.0/WorkflowFunctionalSpec.html

Map-reduce action

- Every action has a defined XML structure in Oozie depending on the XST followed (XML version of Oozie followed)
- ***We can use <file> and <archive> tags to create symlinks to files, and can be used within the action code.***
- ***Oozie automatically looks for file and folders under the lib/ sub-directory under the workflow directory. Hence you don't need to create***

specific symbolic links using <file> and <archive> as long as the files that you need in the actions are present inside the l ib/ sub-directory of the root workflow directory.

-

<u>Java action</u>

-
-
    ○

-

<u>Pig action</u>

FS Action

-
-

Sub-Action wf

-
-
-

Hive action

-

Shell action

-
-

-
-

```
<action name="Impala_job" cred="hcat,hs2-creds">
        <shell xmlns="uri:oozie:shell-action:0.1">
            <job-tracker>${jobTracker}</job-tracker>
            <name-node>${nameNode}</name-node>
            <exec>${invalidate_metadata_shell_script}</exec>
            <argument>${mapdt}</argument>
            <argument>${invalidate_metadata_impala_script}</argument>
            <argument>${keytab_path}</argument>
            <file>${hiveSite}#hive-site.xml</file>
            <file>${hiveConfig}#hive-config.xml</file>
            <!--file>${keytab_path}#sunda29.keytab</file-->
            <file>${invalidate_metadata_shell_script}#invalidate_metadata_impala.sh</file>
            <!--file>${invalidate_metadata_impala_script}#invalidate_metadata_impala_script.sql</file-->
            <!--capture-output/-->
        </shell>
        <ok to="EMAIL_SUCCESS"/>
        <error to="KILL"/>
    </action>
```

```
<action name="Impala_job" cred="hcat,hs2-creds">
        <shell xmlns="uri:oozie:shell-action:0.1">
            <job-tracker>${jobTracker}</job-tracker>
            <name-node>${nameNode}</name-node>
            <exec>/usr/bin/python</exec>
              <argument>test.py</argument>
            <argument>07/21/2014</argument>
           <env-var>TZ=PST</env-var>
            <file>test.py#test.py</file>

            <!--capture-output/-->
        </shell>
        <ok to="EMAIL_SUCCESS"/>
        <error to="KILL"/>
    </action>
```

Distcp action

-

Email action

-

Ssh action

-
-
-

-

Sqoop action

-
-

Synchronous vs Asynchronous actions

Chapter 5

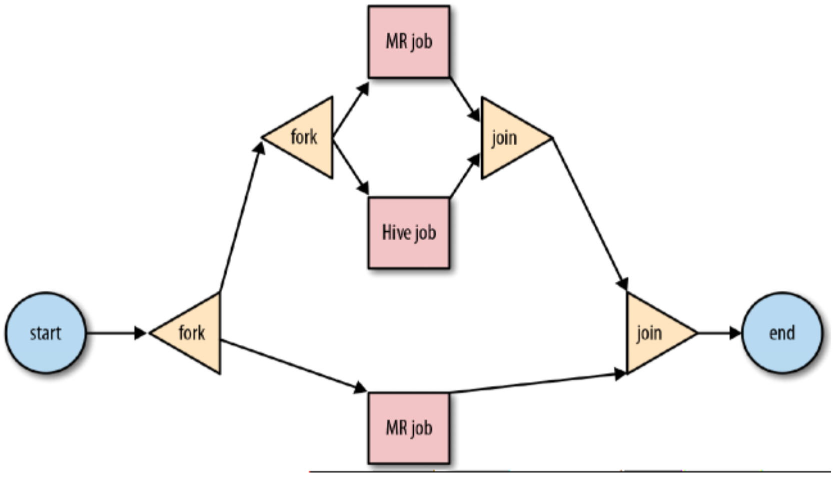
Workflow applications

Outline of a basic workflow

Control nodes

<fork> and <join>

-
-
-



<decision> node

<kill> node

-
-
-

Job XML

Job.properties

Command line option

-

oozie job -oozie http://localhost:4080/oozie/ -DqueueName=research -config job.properties –run

Config-default.xml

config-default.xml file can be used to keep all default values for all workflows

- We can have separate job.properties for each of the other workflows
- This config-default.xml file is most probably overridden by other inline configurations and -D option

Parameters section inside workflow.xml

```
<workflow-app name="parametersWF" xmlns="uri:oozie:workflow:0.5">
<parameters>
        <property>
        <name>queueName</name>
        <value>production</value>
        </property>
</parameters>
```

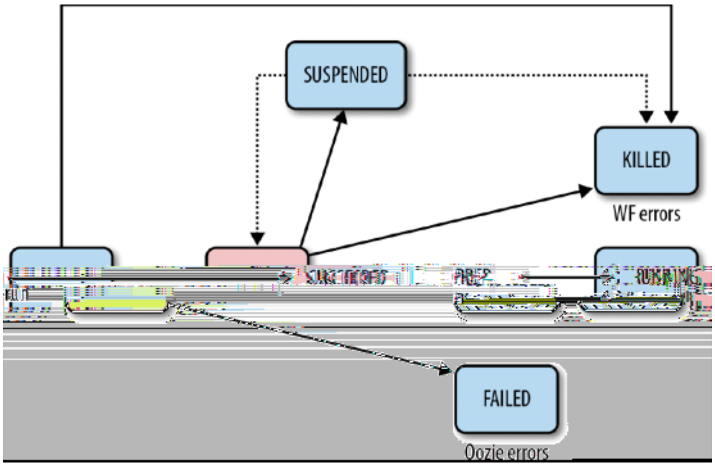
NOTE :

Please do not use config-default.xml and parameters together. Use one or the other

Lifecycle of a workflow :

When we give -run command on oozie, the wf will start executing right away. When we give -submit command the workflow is in the PREP stage, and will execute when we provide -start option

1. PREP
2. RUNNING
3. SUCCEEDED
4. KILLED : When there is an error in the wf
5. FAILED : When there is an Oozie error
6. SUSPENDED : When the workflow is suspended using the suspend command



Chapter 6

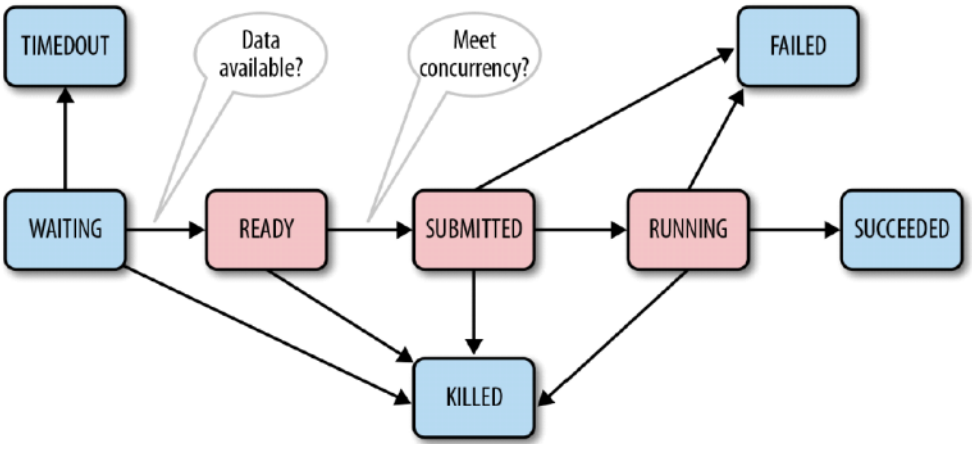
Oozie coordinator

-
-
    ○
    ○
    ○

<u>Coordinator action :</u>

<u>Nominal time of a coordinator action</u>

<u>Coordinator action lifecycle</u>



-
-

<u>Coordinator execution controls</u>

Chapter 7

Data trigger coordinators

Dataset :

-
-
    ○
    ○
        ▪

Defining a dataset

-
    ○
    ○

<include>hdfs://localhost:8020/user/joe/shares/common_datasets.xml</include>

Input events

Output events

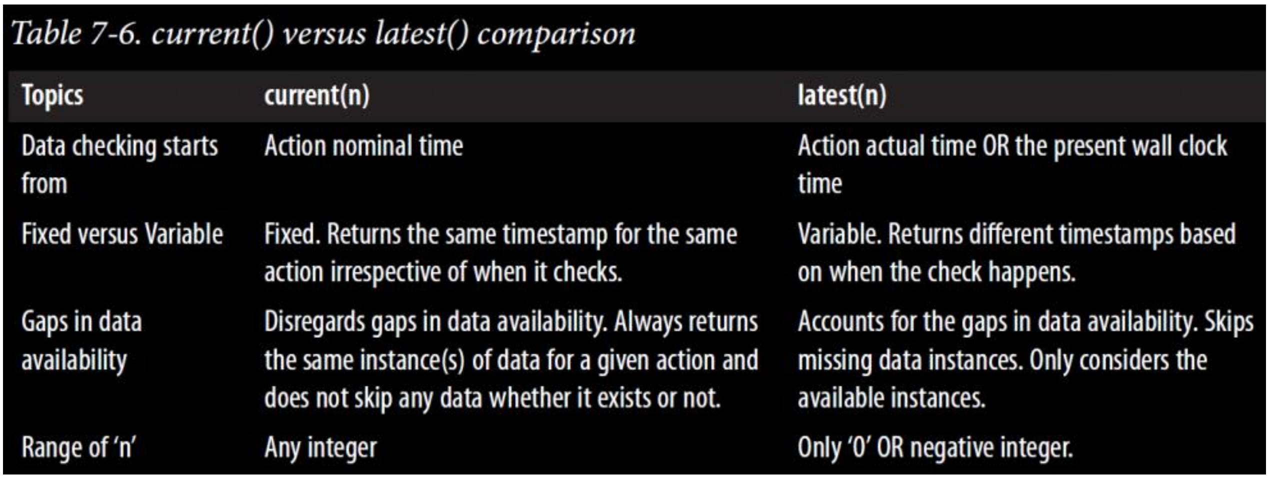
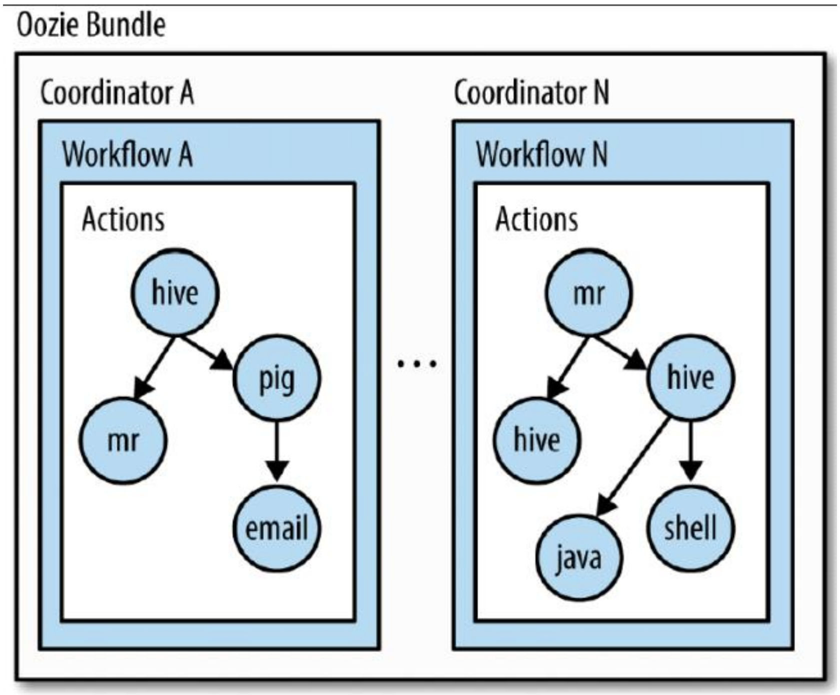


Table 7-6. current() versus latest() comparison

Topics	current(n)	latest(n)
Data checking starts from	Action nominal time	Action actual time OR the present wall clock time
Fixed versus Variable	Fixed. Returns the same timestamp for the same action irrespective of when it checks.	Variable. Returns different timestamps based on when the check happens.
Gaps in data availability	Disregards gaps in data availability. Always returns the same instance(s) of data for a given action and does not skip any data whether it exists or not.	Accounts for the gaps in data availability. Skips missing data instances. Only considers the available instances.
Range of 'n'	Any integer	Only '0' OR negative integer.

A complete coordinator

-
-



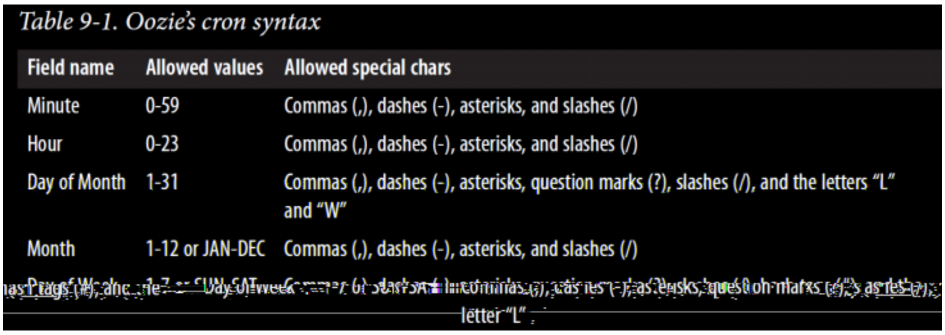


Table 9-1. Oozie's cron syntax

Field name	Allowed values	Allowed special chars
Minute	0-59	Commas (,), dashes (-), asterisks, and slashes (/)
Hour	0-23	Commas (,), dashes (-), asterisks, and slashes (/)
Day of Month	1-31	Commas (,), dashes (-), asterisks, question marks (?), slashes (/), and the letters "L" and "W"
Month	1-12 or JAN-DEC	Commas (,), dashes (-), asterisks, and slashes (/)
Day of Week	1-7 or SUN-SAT	Commas (,), dashes (-), asterisks, question marks (?), slashes (/), pound signs (#), and the letter "L"

-

-
-
-
    ○
    ○
-
-

Oozie CLI

```
$ oozie job -config ./job.properties –submit

$ oozie job –suspend 0000006-130606115200591-oozie-joe-W
$ oozie job –resume 0000006-130606115200591-oozie-joe-W
$ oozie job –kill 0000006-130606115200591-oozie-joe-W

$ oozie job -info 0000084-141219003455004-oozie-joe-C -len 10

$ oozie job -dryrun -config wf_job.properties

$ oozie job -
```

Bundle reprocessing