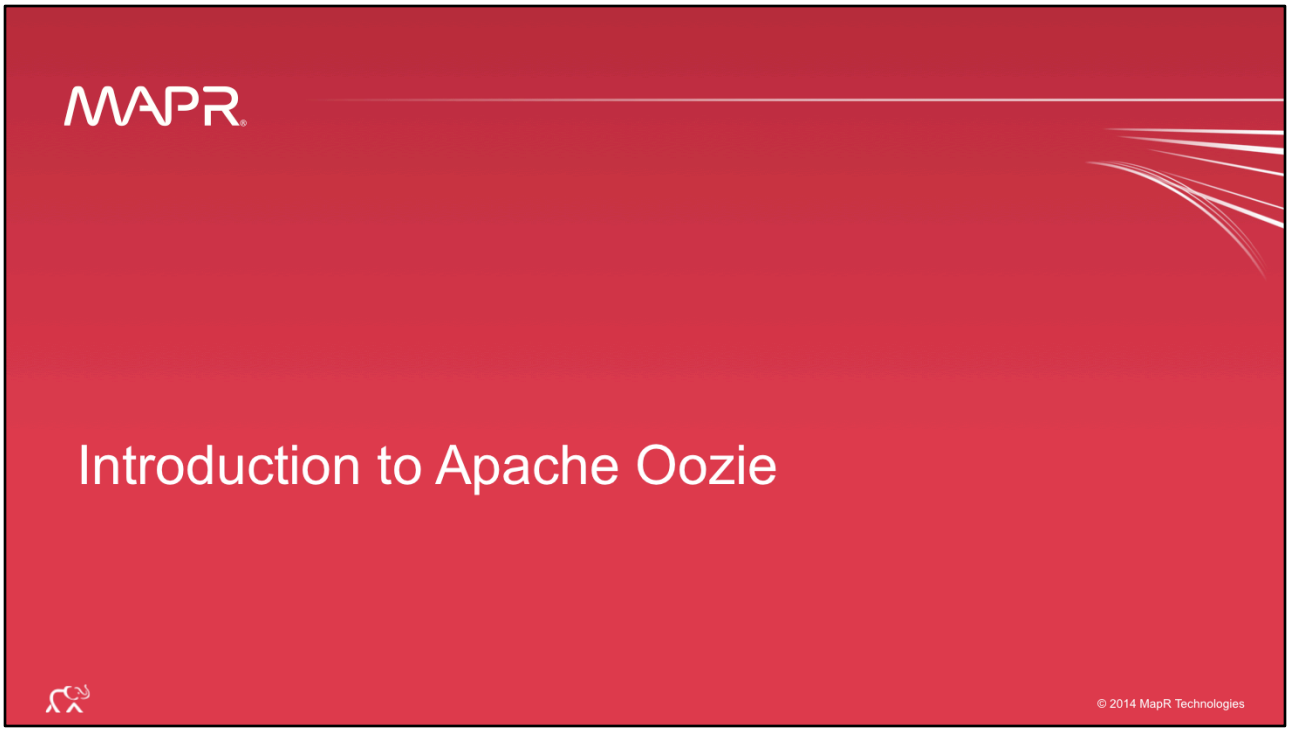
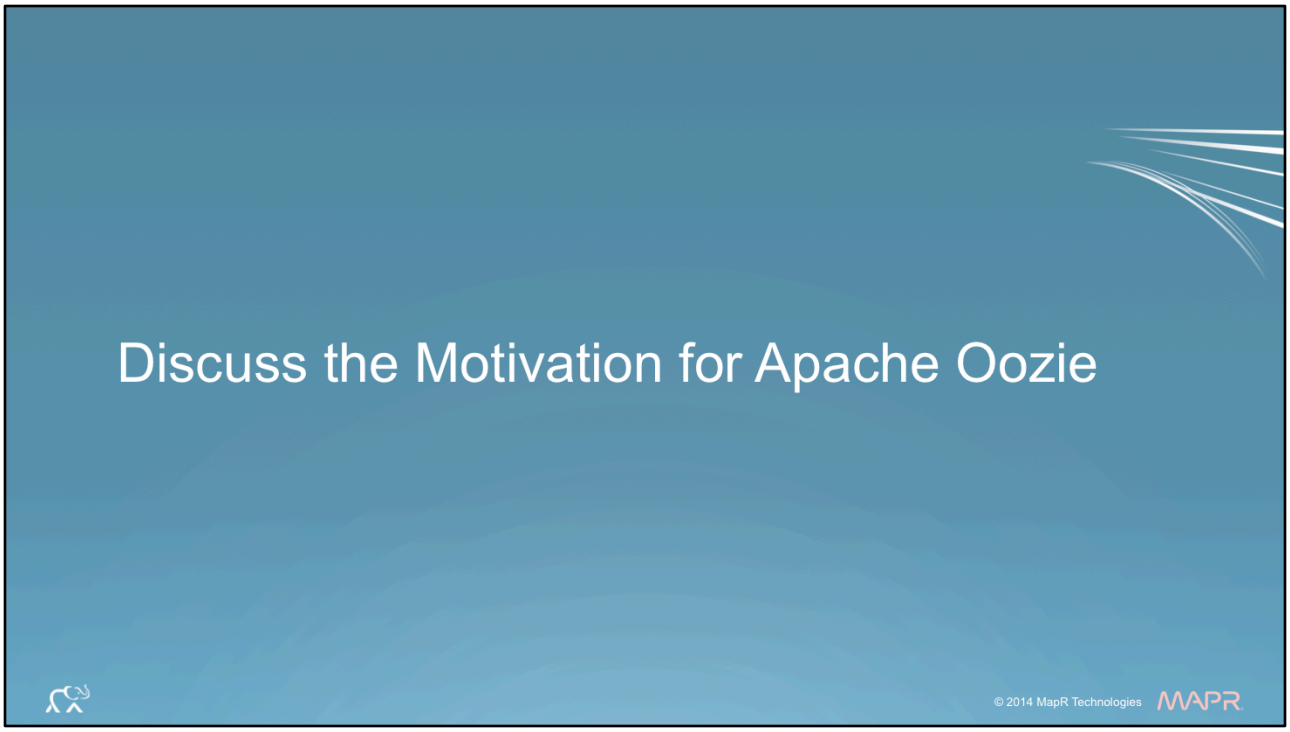


This lesson is a short introduction to Apache Oozie.

Learning Objectives

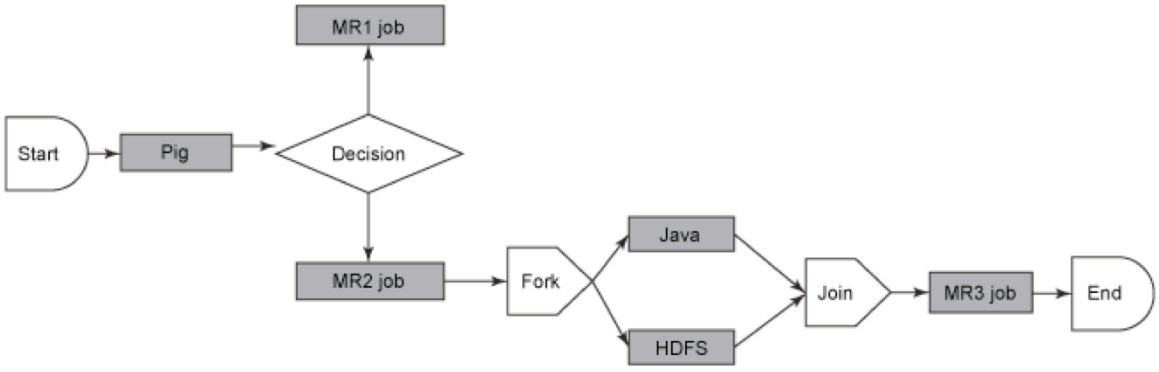
- Discuss the motivation for Oozie
- Describe how to use Oozie
- Perform a basic demonstration of Oozie

The objectives for this lesson are defined above.



This section briefly motivates the need for Apache Oozie.

Here's a Sample Hadoop Workflow



The diagram above (called an acyclic directed graph, or DAG) represents a simple Hadoop workflow. How would you manage such a workflow?

You Can Use Shell Scripts to Manage the Workflow

```
#!/bin/bash

rm -rf ~/DISTRIBUTED_CACHE/UNIV_OUT
ARGS=$1

hadoop jar University.jar University.UniversityDriver -D
var1="verbal" -D var2="math"  ~/DISTRIBUTED_CACHE/DATA/
university.txt ~/DISTRIBUTED_CACHE/UNIV_OUT

rm -rf ~/DISTRIBUTED_CACHE/STAT_OUT

hadoop jar Stats.jar Stats.StatsDriver ~/DISTRIBUTED_CACHE/DATA/
university.txt ~/DISTRIBUTED_CACHE/UNIV_OUT ~/DISTRIBUTED_CACHE/
STAT_OUT
```

 MAPR 5

The script above is a simple example of using a shell script to manage a simple MapReduce workflow comprised of just 2 jobs. Imagine if there were many more tasks to launch, with branch points and decision logic, using tools other than or in addition to MapReduce.

But Scripts Will Have Issues

- Scripts are difficult to make robust
- Scripts are difficult to modify
- Scripts don't scale well with the number of jobs in workflow
- Every Hadoop ecosystem CLI has a unique syntax

As described in the slide above, using shell scripts will be problematic. For one, it's difficult to make a shell script work properly, much less identify and react to failures. Moreover, once you do get your shell script working, it is difficult to modify. It's not easy to add new functionality or modify existing functionality once you get something working. Hadoop workflows can get complex, and using a shell script to manage complex workflows doesn't scale well. Last, and perhaps most importantly, every Hadoop ecosystem has a unique CLI syntax. This makes the learning curve steep just to write the script.

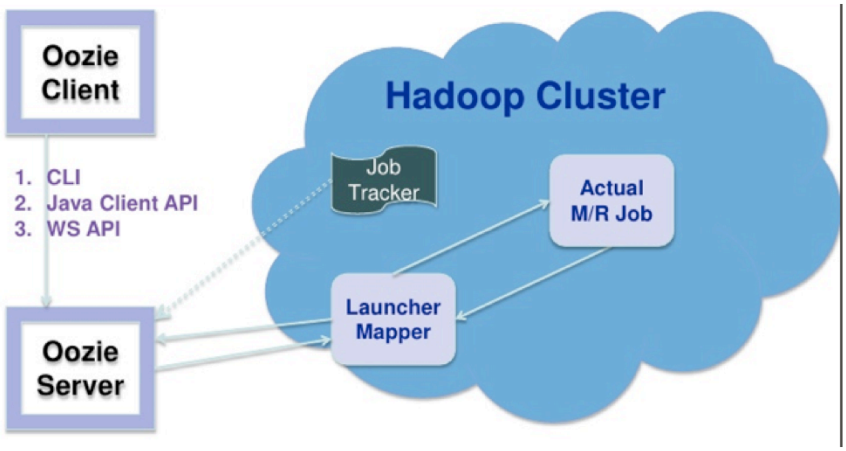
This section provides a short introduction on using Apache Oozie.

What is Oozie?

- Scalable and extensible workflow scheduler system to manage Hadoop jobs within a cluster
- Hadoop workflows are represented as DAGs and encoded in XML
- An Oozie workflow can manage:
  - MapReduce
  - Pig
  - Hive
  - Sqoop
  - Distcp
  - Arbitrary scripts and Java programs

Oozie is a scalable and extensible scheduling system for complex Hadoop workflows. Workflows in Oozie are represented as directed acyclic graphs (or DAGs) and are encoded in a XML file. Oozie can be used to manage multiple types of jobs within a workflow, including MapReduce, Pig, Hive, Sqoop, distcp, and arbitrary scripts and Java programs.

Oozie Architecture



The graphic above depicts the Oozie architecture. The client can be written using a CLI, a Java API, or Web service (REST) API. The client submits workflows to the Oozie server, which then contacts the appropriate endpoint in the Hadoop cluster to launch the job. In the slide above, the Oozie server is submitting a simple MapReduce job.

Use Cases for Oozie

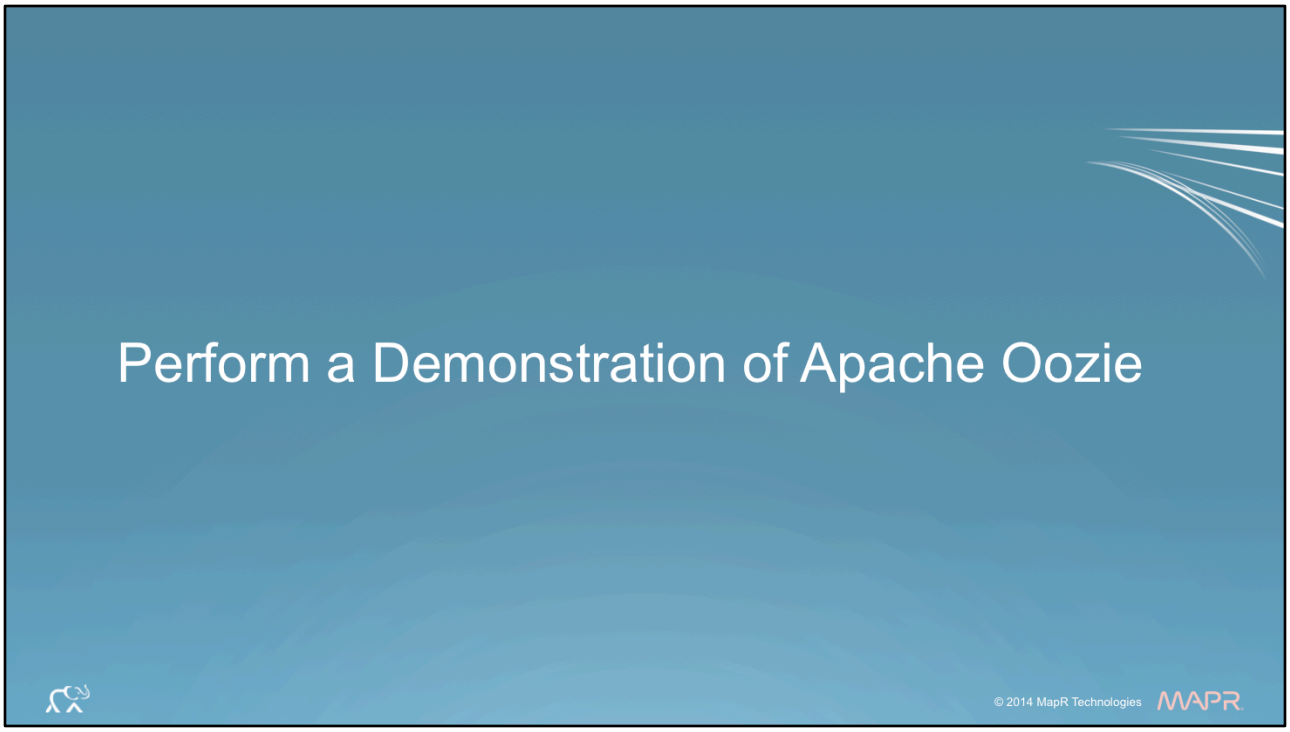
Use Case	Example
Complex workflow	Most Hadoop solutions require multiple programs to run in a series or in parallel or both
Time dependency	Jobs need to run at a certain regular frequency
Data dependency	Jobs shouldn't run until data is available

There are several cases where you can use Oozie in Hadoop. The primary use case for Oozie is to manage a complex workflow comprised of several different types of jobs. You may also have a dependency in your workflow that must be met first, including a time dependency or data dependency, as described in the table above.

References for Oozie

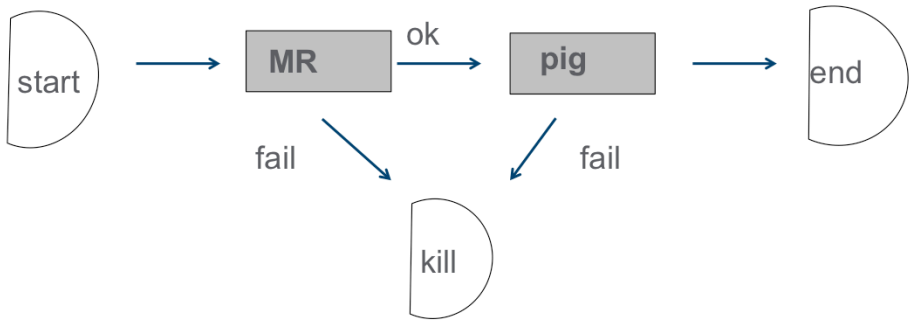
- http://doc.mapr.com/display/MapR/Oozie
- http://oozie.apache.org
- Hadoop: The Definitive Guide (O'Reilly)

The slide above identifies a few references for you to get started using Oozie. There are, however, many more resources available for Oozie. Just google it!



This last section provides a short demonstration of Oozie.

Demo: Using Oozie to Manage a Hadoop Workflow



 13

The DAG above shows a simple workflow that we will demonstrate with Oozie. After the start of the workflow, Oozie will launch a MapReduce job. Once that MapReduce job is finished, Oozie will launch a Pig job. If either job fails, Oozie will short-circuit the workflow and terminate in the "kill" state. If both jobs run successfully, the workflow ends in the "end" state.

Q&A

Engage with us!

@mapr			maprtech
mapr-technologies			MapR
yourname@mapr.com			maprtech