大数据Hadoop高薪直通车课程

Spark 高阶应用

讲师：轩宇（北风网版权所有)

课程大纲

1. Spark 应用开发
2. Spark HistoryServer
3. Spark on YARN
4. Spark Streaming
5. Spark Streaming 案例

课程大纲

1. Spark 应用开发
2. Spark HistoryServer
3. Spark on YARN
4. Spark Streaming
5. Spark Streaming 案例

Spark Application Monitor

Web Interfaces

Every SparkContext launches a web UI, by default on port 4040, that displays useful information about the application. This includes:

- A list of scheduler stages and tasks
- A summary of RDD sizes and memory usage
- Environmental information.
- Information about the running executors

You can access this interface by simply opening `http://<driver-node>:4040` in a web browser. If multiple SparkContexts are running on the same host, they will bind to successive ports beginning with 4040 (4041, 4042, etc).

Spark Application Monitor

Spark's Standalone Mode cluster manager also has its own web UI. If an application has logged events over the course of its lifetime, then the Standalone master's web UI will automatically re-render the application's UI after the application has finished.

If Spark is run on Mesos or YARN, it is still possible to reconstruct the UI of a finished application through Spark's history server, provided that the application's event logs exist. You can start the history server by executing:

```
./sbin/start-history-server.sh
```

When using the file-system provider class (see spark.history.provider below), the base logging directory must be supplied in the `spark.history.fs.logDirectory` configuration option, and should contain sub-directories that each represents an application's event logs. This creates a web interface at `http://<server-url>:18080` by default. The history server can be configured as follows:

服务器端配置

配置在spark-env.sh中的SPARK_HISTORY_OPTS

Property Name	Default	Meaning
spark.history.provider	org.apache.spark.deploy.history.FsHistoryProvider	Name of the class implementing the application history backend. Currently there is only one implementation, provided by Spark, which looks for application logs stored in the file system.
spark.history.fs.logDirectory	file:/tmp/spark-events	Directory that contains application event logs to be loaded by the history server
spark.history.fs.updateInterval	10	The period, in seconds, at which information displayed by this history server is updated. Each update checks for any changes made to the event logs in persisted storage.
spark.history.retainedApplications	50	The number of application UIs to retain. If this cap is exceeded, then the oldest applications will be removed.
spark.history.ui.port	18080	The port to which the web interface of the history server binds.

http://spark.apache.org/docs/latest/monitoring.html

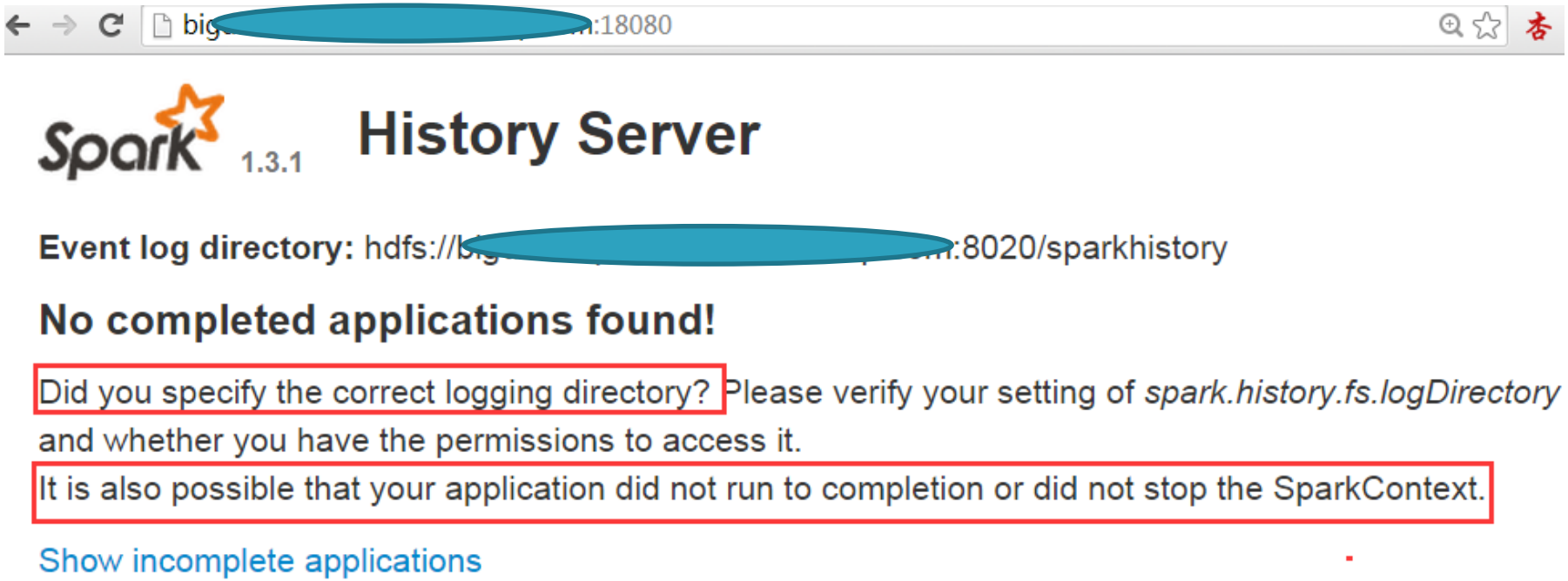
客户端配置

配置在**spark-defaults.conf**

Property Name	Default	Meaning
spark.eventLog.compress	false	Whether to compress logged events, if spark.eventLog.enabled is true.
spark.eventLog.dir	file:///tmp/spark-events	Base directory in which Spark events are logged, if spark.eventLog.enabled is true. Within this base directory, Spark creates a sub-directory for each application, and logs the events specific to the application in this directory. Users may want to set this to a unified location like an HDFS directory so history files can be read by the history server.
spark.eventLog.enabled	false	Whether to log Spark events, useful for reconstructing the Web UI after the application has finished.

http://spark.apache.org/docs/latest/configuration.html#spark-ui

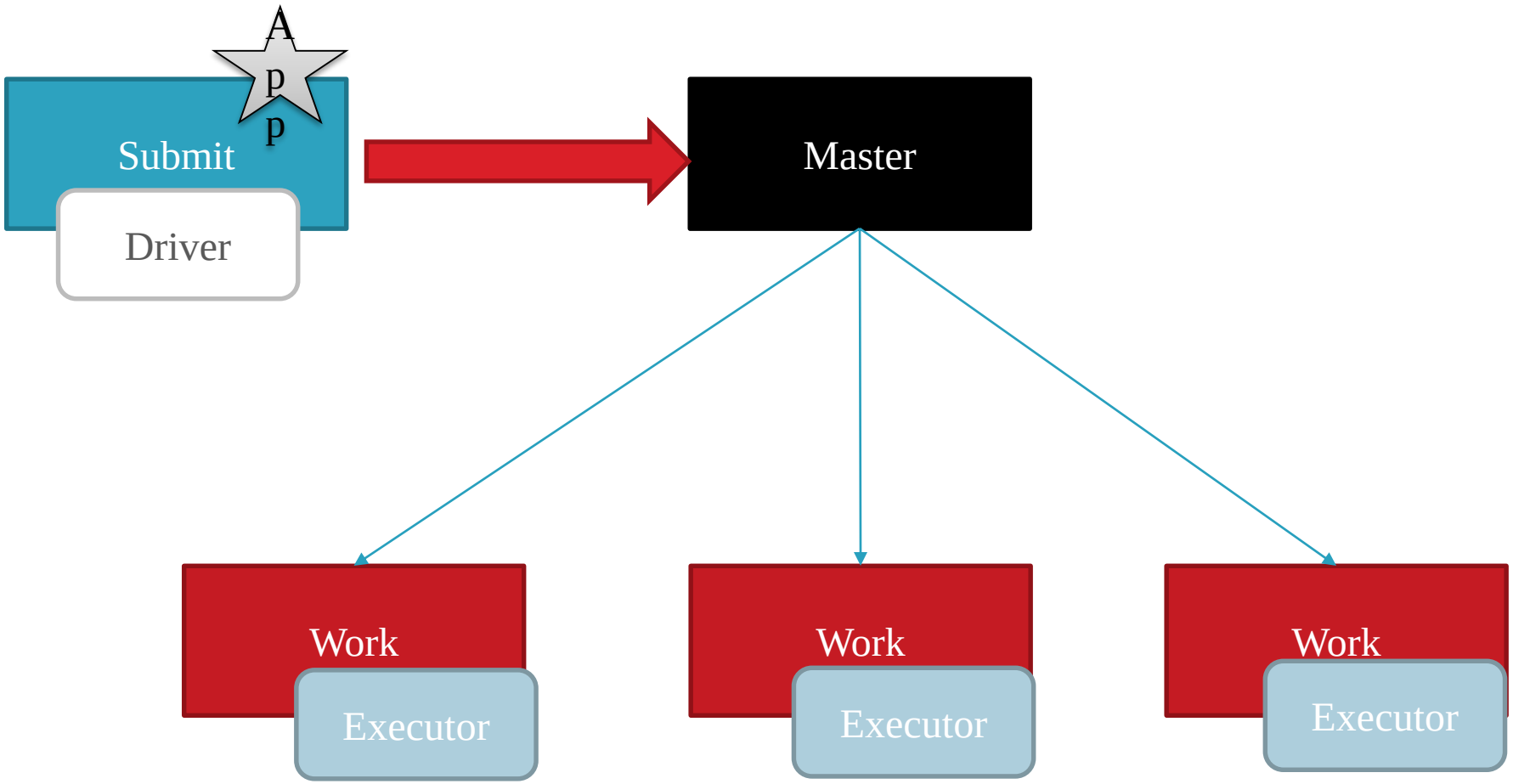
Spark History Server



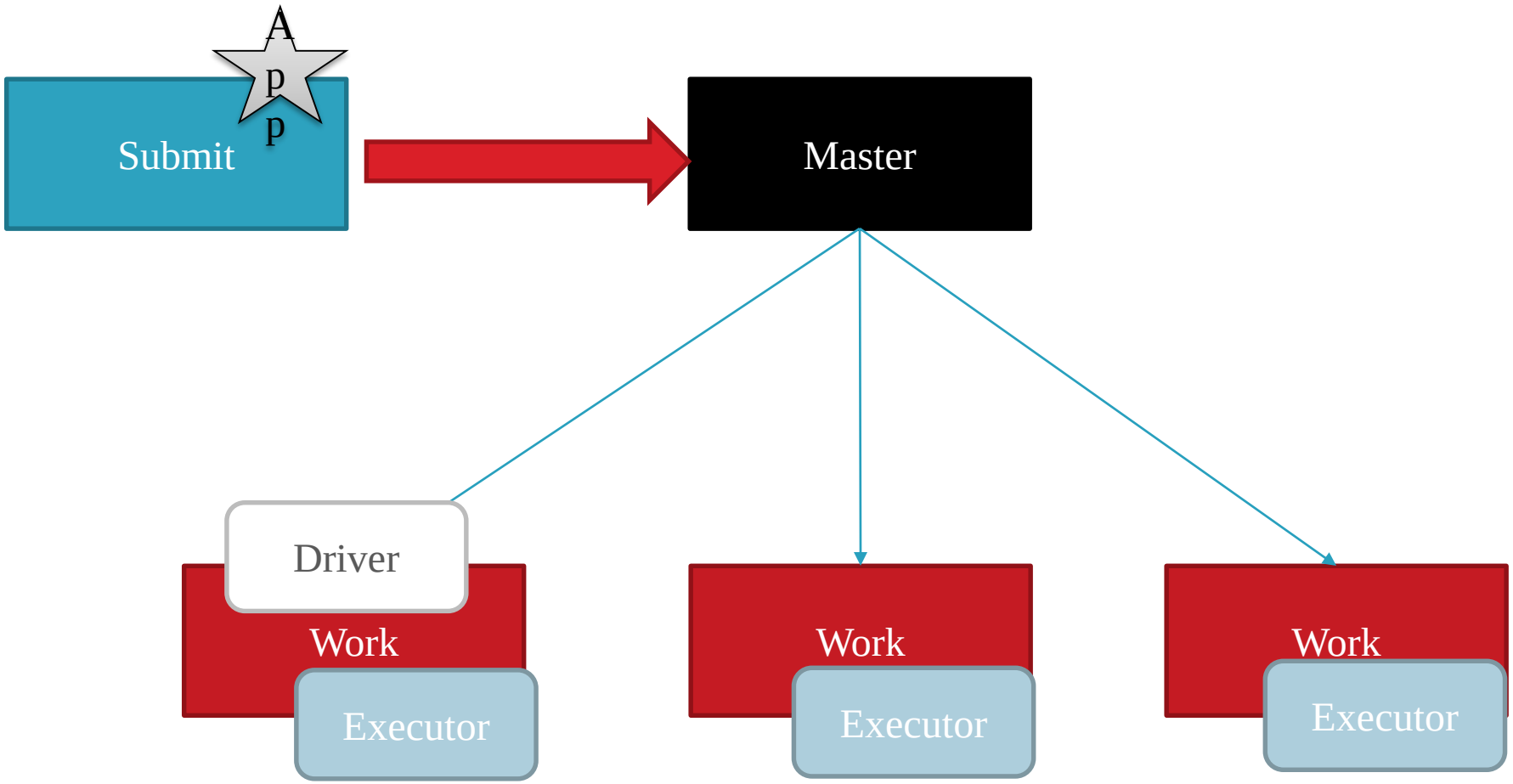
课程大纲

1. Spark 应用开发
2. Spark HistoryServer
3. Spark on YARN
4. Spark Streaming
5. Spark Streaming 案例

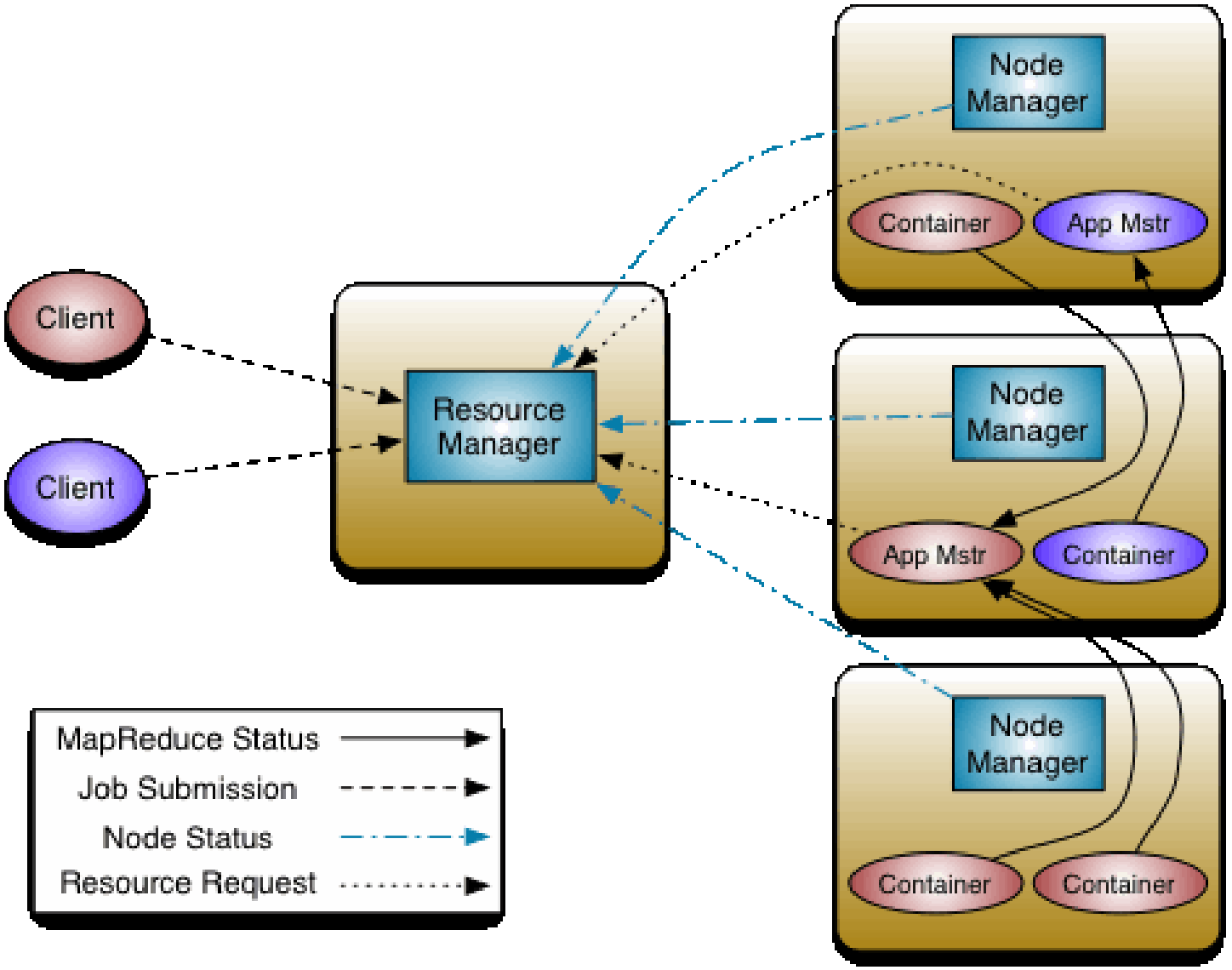
Client


A
p
p
Submit
Driver
Master
Work
Executor
Work
Executor
Work
Executor

Cluster


A
p
p
Submit
Master
Driver
Work
Executor
Work
Executor
Work
Executor

YARN 架构图


Client
Client
Resource Manager
Node Manager
Container
App Mstr
Node Manager
App Mstr
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request

Launching Spark on YARN

Launching Spark on YARN

Ensure that `HADOOP_CONF_DIR` or `YARN_CONF_DIR` points to the directory which contains the (client side) configuration files for the Hadoop cluster. These configs are used to write to the dfs and connect to the YARN ResourceManager.

There are two deploy modes that can be used to launch Spark applications on YARN. In yarn-cluster mode, the Spark driver runs inside an application master process which is managed by YARN on the cluster, and the client can go away after initiating the application. In yarn-client mode, the driver runs in the client process, and the application master is only used for requesting resources from YARN.

Unlike in Spark standalone and Mesos mode, in which the master's address is specified in the "master" parameter, in YARN mode the ResourceManager's address is picked up from the Hadoop configuration. Thus, the master parameter is simply "yarn-client" or "yarn-cluster".

To launch a Spark application in yarn-cluster mode:

```
./bin/spark-submit --class path.to.your.Class --master yarn-cluster [options] <app jar> [app opti
ons]
```

Launching Spark on YARN

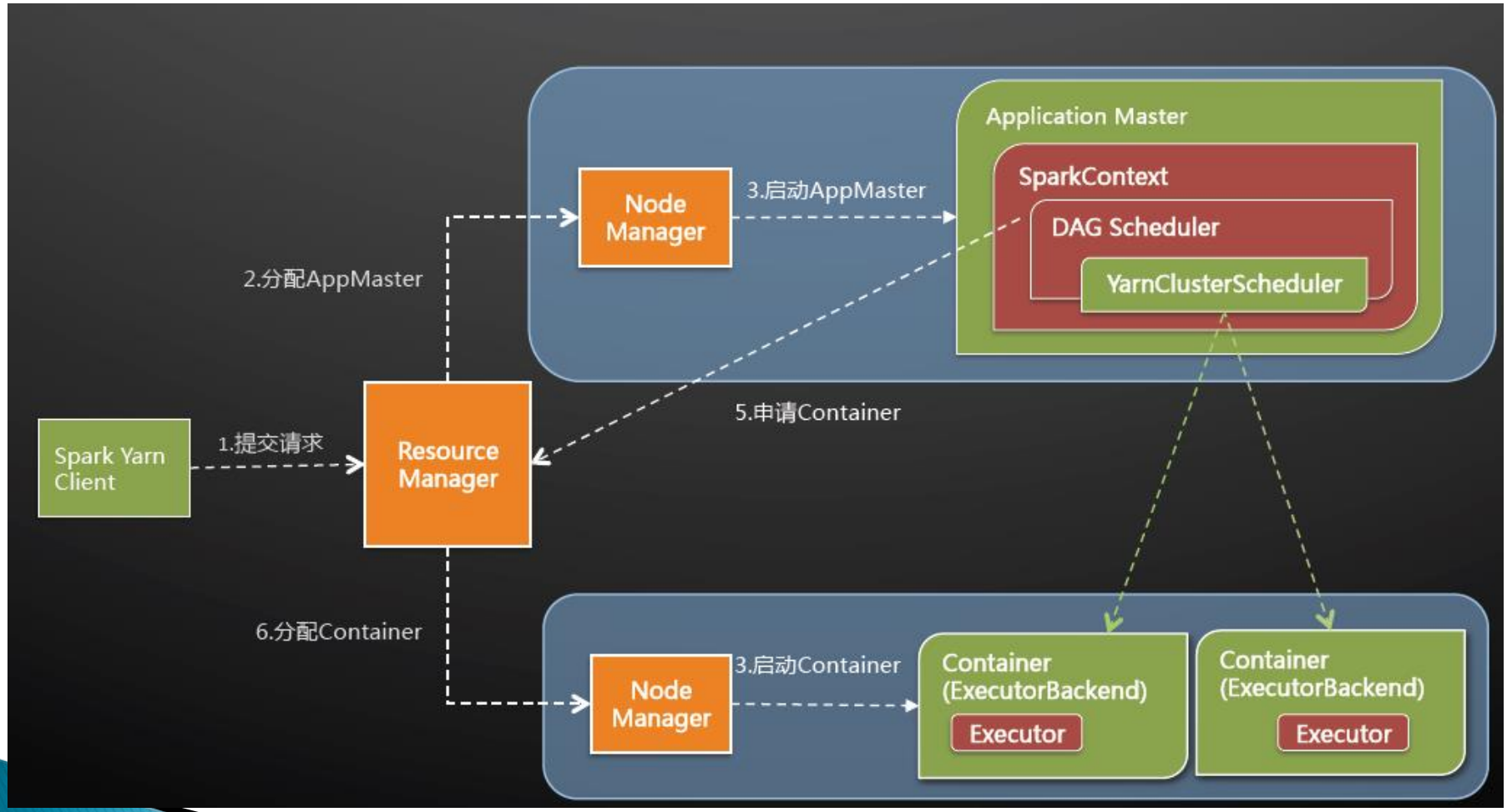
```
$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi \
    --master yarn-cluster \
    --num-executors 3 \
    --driver-memory 4g \
    --executor-memory 2g \
    --executor-cores 1 \
    --queue thequeue \
    lib/spark-examples*.jar \
    10
```

The above starts a YARN client program which starts the default Application Master. Then SparkPi will be run as a child thread of Application Master. The client will periodically poll the Application Master for status updates and display them in the console. The client will exit once your application has finished running. Refer to the "Debugging your Application" section below for how to see driver and executor logs.

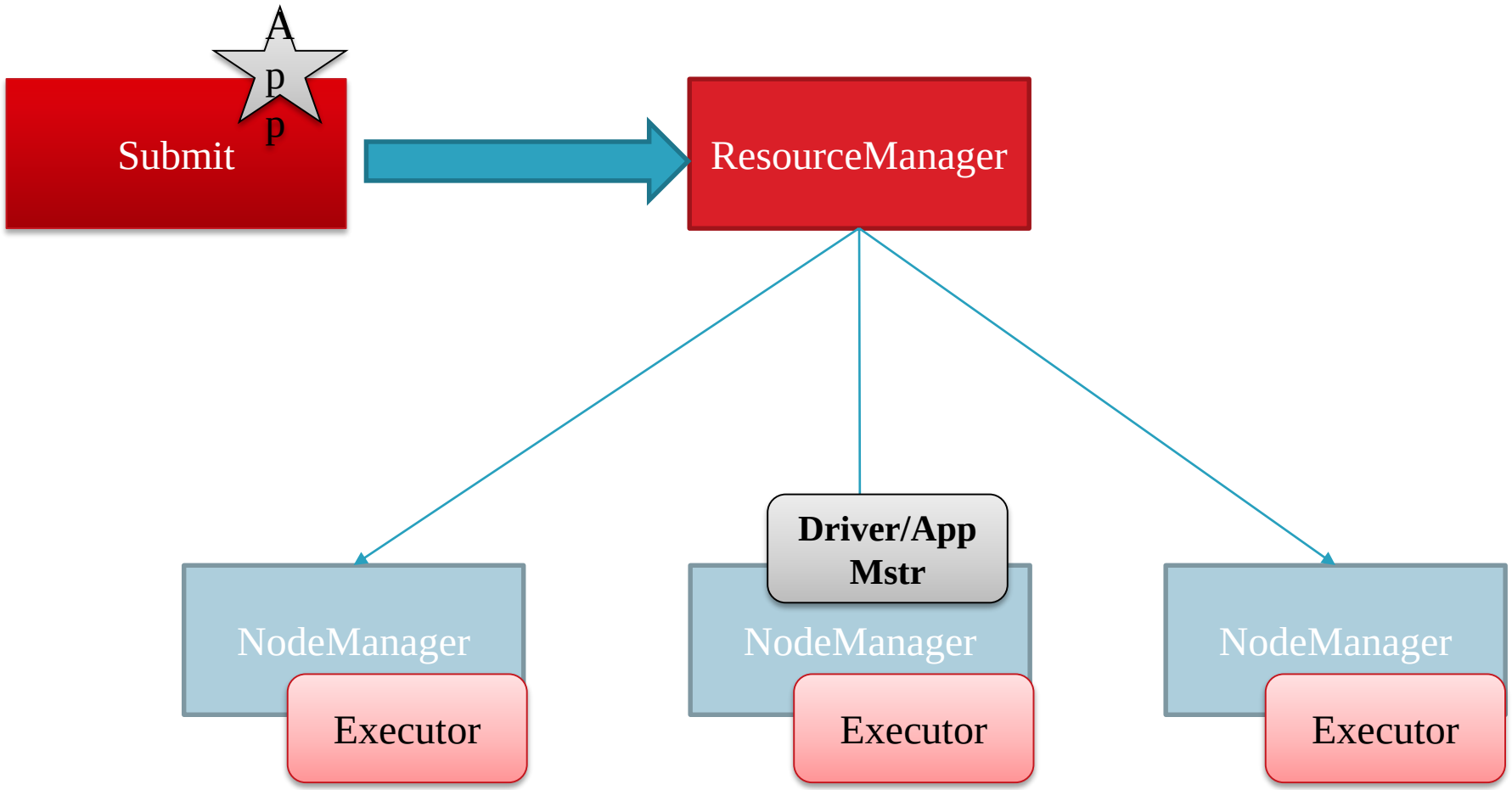
To launch a Spark application in yarn-client mode, do the same, but replace "yarn-cluster" with "yarn-client". To run spark-shell:

```
$ ./bin/spark-shell --master yarn-client
```

Spark on YARN



Spark on YARN



本课程版权归北风网所有

欢迎访问我们的官方网站

www.ibeifeng.com