Spark on Mesos

Tim Chen
Mirantis
tnachen@gmail.com

Dean Wampler
Lightbend
dean.wampler@lightbend.com

SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE
JUNE 6-8, 2016 SAN FRANCISCO

Dean Wampler

- Architect for Big Data Products at Lightbend
  - Early advocate for Spark on Mesos
- O'Reilly author
  - –Programming Scala, 2nd Edition
  - –Programming Hive
  - –Functional Programming for Java Developers

Timothy Chen

- Principal Engineer at Mirantis
- Previously lead engineer at Mesosphere
- Apache Mesos PMC
- Spark contributor, help maintain Spark on Mesos

What’s this all about, then?

- Why Spark on Mesos?
- What’s happened since last year?
- Demo - GPU support
- What’s next for Spark and Mesos?

Why Spark on Mesos

- Hadoop is great, but ...
  - … resource management with YARN is limited to compute engines like MapReduce and Spark.
- What if your clustering system could run *everything*?

Why Spark on Mesos

- Hadoop is great, but ...
  - … Big Data is moving to streaming ("Fast Data") and Spark offers mini-batch streaming.
- What if your cluster system offered dynamic and flexible resource scheduling able to meet the needs of evolving, long-running streams?

Why Spark on Mesos

- Hadoop is great, but ...
  - … it doesn’t support other popular tools like Cassandra, Akka, web frameworks, ...
- Maybe you need the SMACK stack:
  - Spark
  - Mesos
  - Akka
  - Cassandra
  - Kafka

What’s happened since last year?

- What’s new in Mesos
- What’s new in Spark on Mesos
- Getting rid of fine-grained mode?

What’s new in Mesos?

- Maintenance primitives
- Resource quotas, dynamic reservation

Beta

- CNI network Support
- GPU Support
- Unified Containerizer
- More..

What’s new in Spark on Mesos?

- Integration test suite
- New scheduler
- Mesos framework authentication
- Cluster mode now supports Python

Integration Test Suite

- A recent release candidate for Spark broke Mesos integration completely.
  - Better *integration testing* clearly needed.
  - Lightbend and Mesosphere collaborated on an automated integration test suite.

https://github.com/typesafehub/mesos-spark-integration-tests

Integration Test Suite

- “mesos-docker” subproject:
  - –Builds Docker image with Ubuntu, Mesos, Spark, and HDFS.
  - –Scripts to run cluster with 1 master and N slaves, configurable #s of CPUs, memory, etc.
    - •(Not needed if you already have a Mesos cluster ;^)

Integration Test Suite

- “test-runner” subproject:
  - –Executes a suite of tests on your Mesos or DC/OS cluster.
  - – Currently exercises dynamic allocation, coarse-grain and fine-grain modes, etc.

New Coarse Grain Scheduler

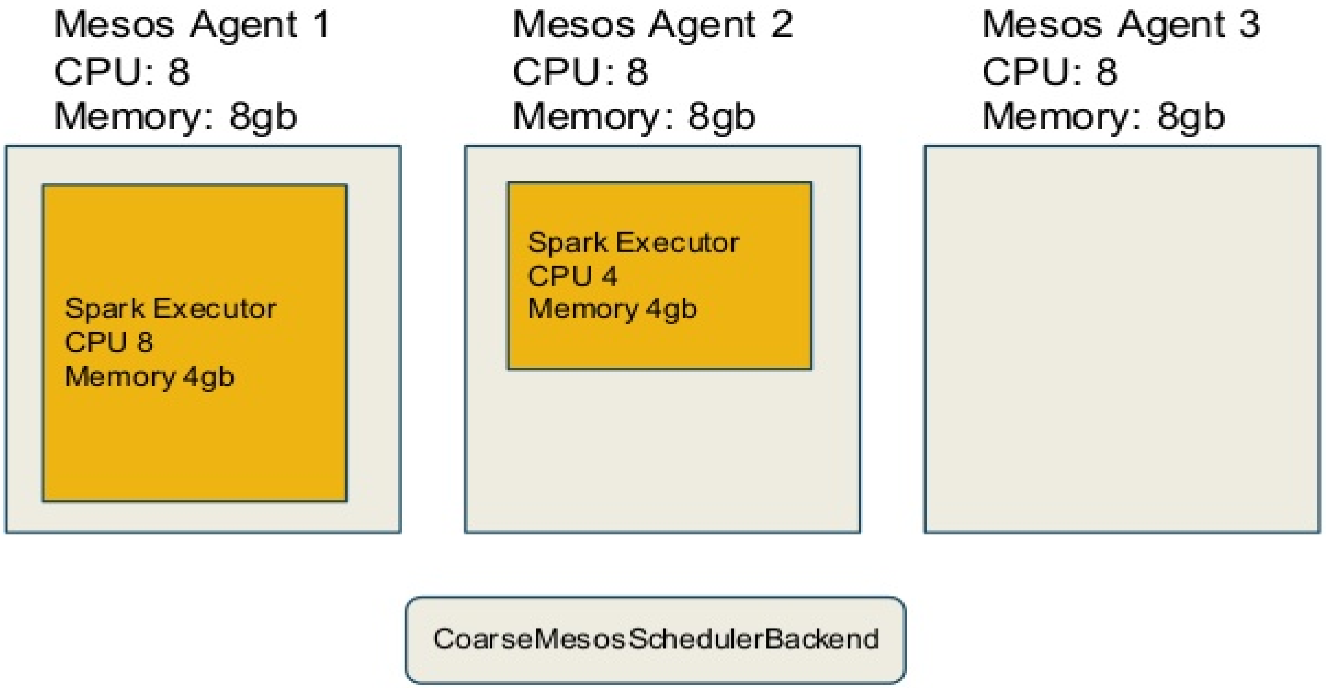
How the old Coarse grain scheduler works?

Launch 1 Spark executor per agent

- Rough steps:
    - Evaluate offers as it comes in from the master
    - Offers that meets min cpu (1) and min memory requirements
    - Use as much cores until meets spark.cores.max
    - Every executor requests fixed memory

New Coarse Grain Scheduler

How the old Coarse grain scheduler works?


Mesos Agent 1
CPU: 8
Memory: 8gb

Spark Executor
CPU 8
Memory 4gb

Mesos Agent 2
CPU: 8
Memory: 8gb

Spark Executor
CPU 4
Memory 4gb

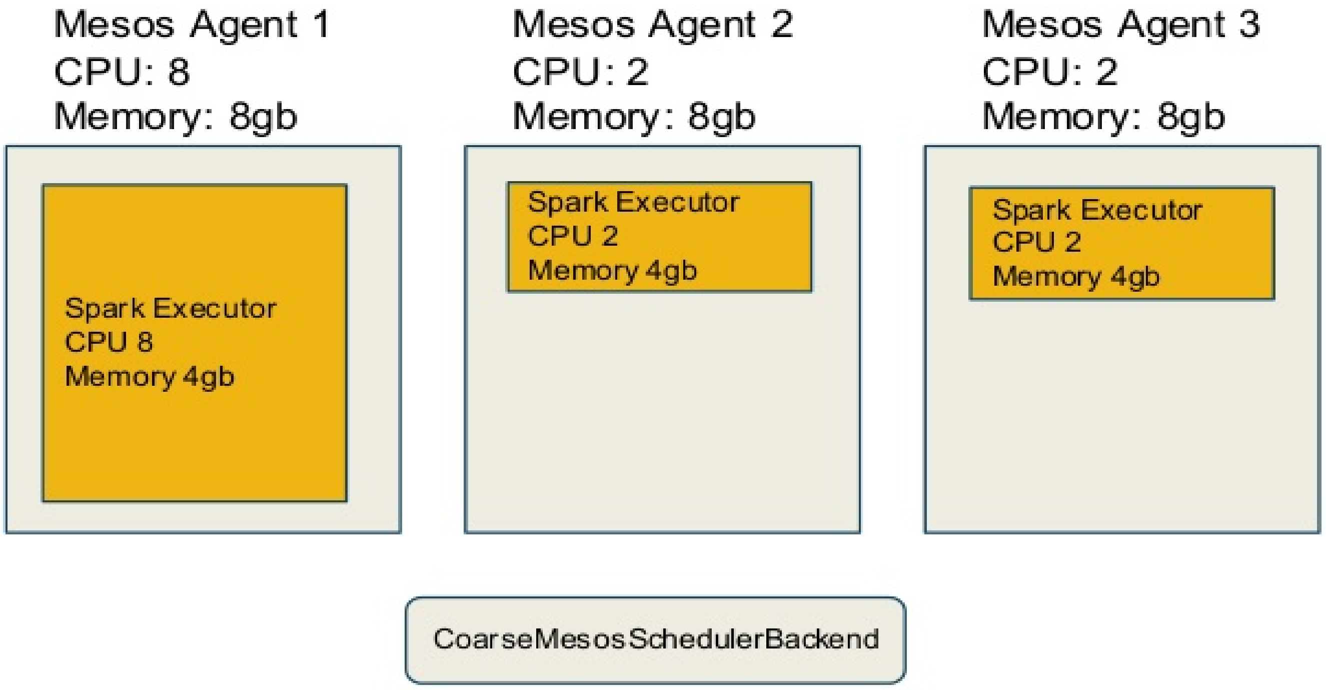
Mesos Agent 3
CPU: 8
Memory: 8gb

CoarseMesosSchedulerBackend

spark.cores.max=12
spark.executor.memory=4gb

New Coarse Grain Scheduler

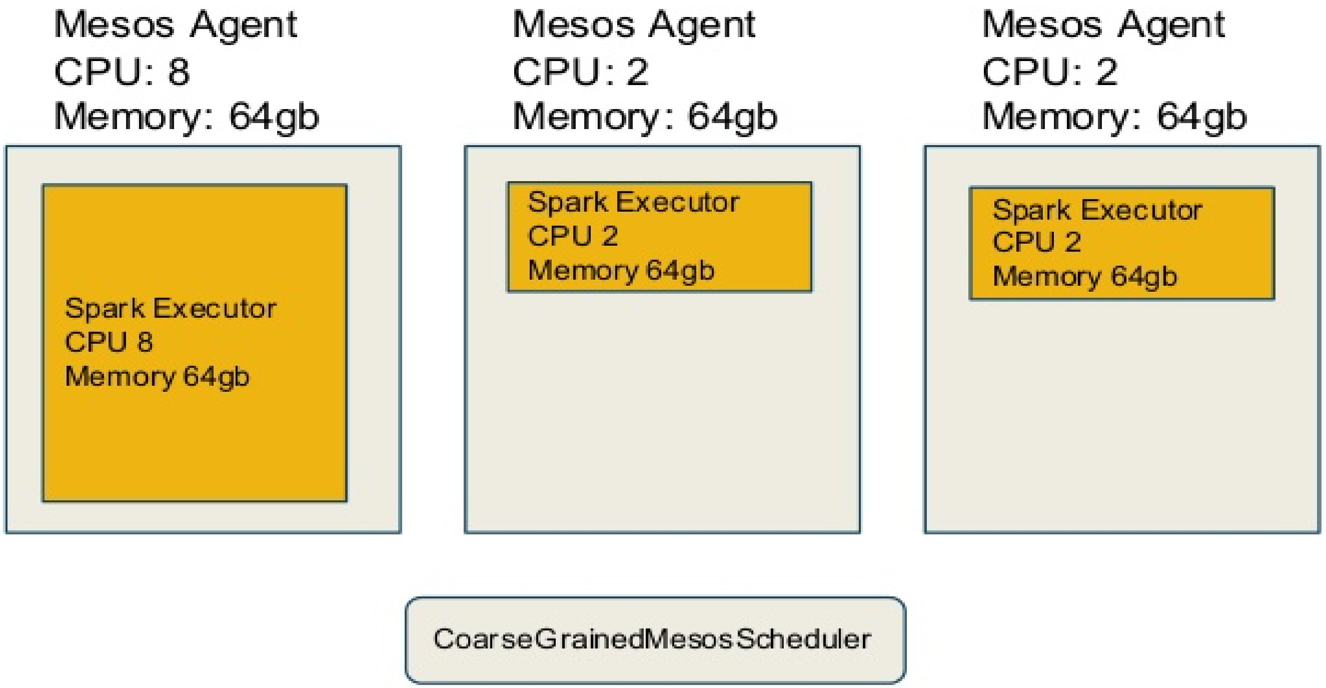
How the old Coarse grain scheduler works?



spark.cores.max=12
spark.executor.memory=4gb

New Coarse Grain Scheduler

How the old Coarse grain scheduler works?



Mesos Agent
CPU: 8
Memory: 64gb

Spark Executor
CPU 8
Memory 64gb

Mesos Agent
CPU: 2
Memory: 64gb

Spark Executor
CPU 2
Memory 64gb

Mesos Agent
CPU: 2
Memory: 64gb

Spark Executor
CPU 2
Memory 64gb

CoarseGrainedMesosScheduler

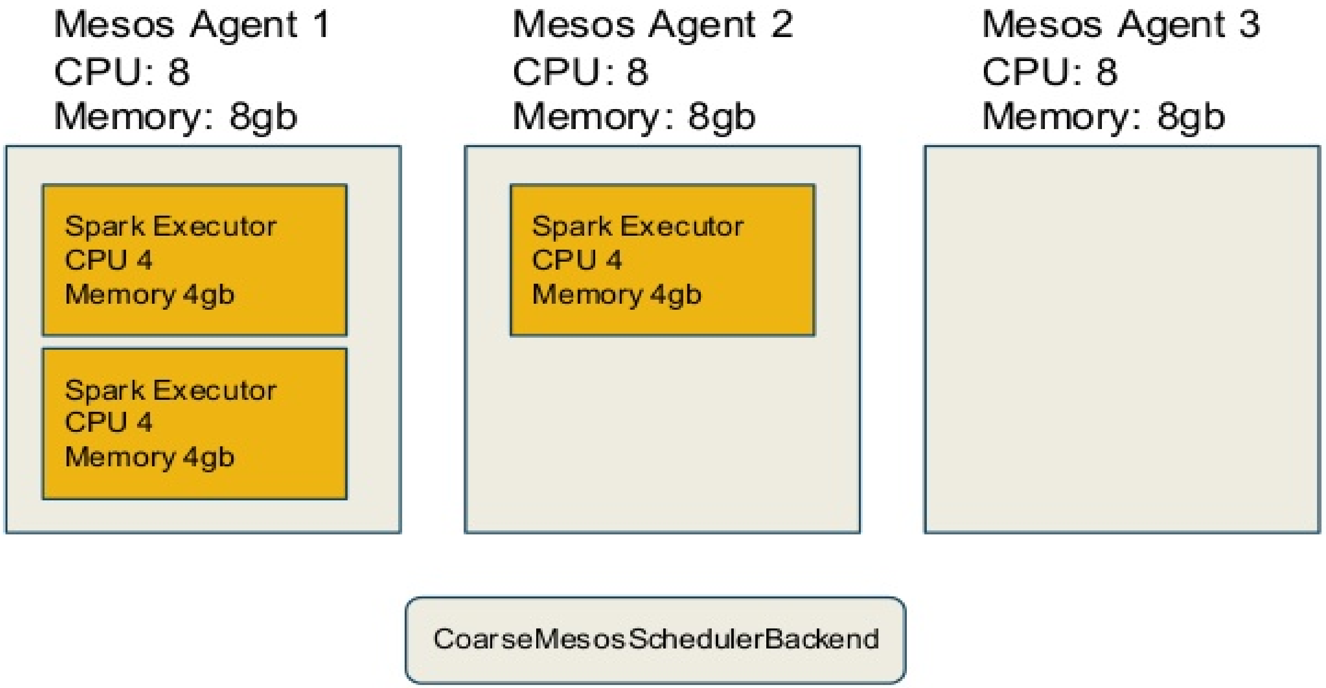
spark.cores.max=12
spark.executor.memory=64gb

New Coarse Grain Scheduler

Problems with the old scheduler:

- Only allow one executor per slave
- Unpredictable performance
- Can skew allocation

New Coarse Grain Scheduler



Mesos Agent 1
CPU: 8
Memory: 8gb

Spark Executor
CPU 4
Memory 4gb

Spark Executor
CPU 4
Memory 4gb

Mesos Agent 2
CPU: 8
Memory: 8gb

Spark Executor
CPU 4
Memory 4gb

Mesos Agent 3
CPU: 8
Memory: 8gb

CoarseMesosSchedulerBackend

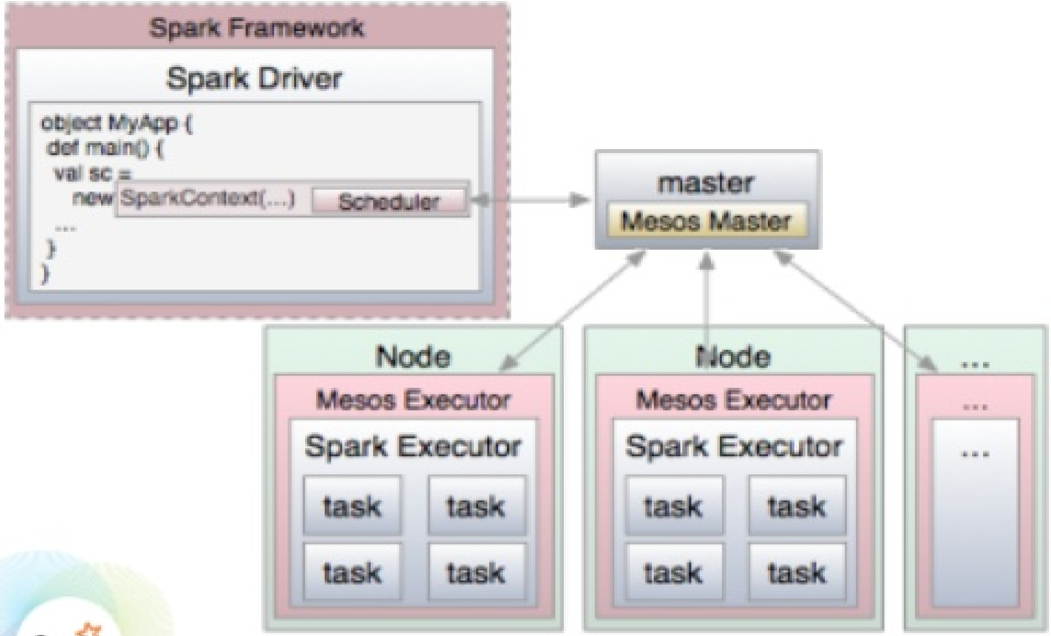
spark.cores.max=12
spark.executor.memory=4gb
spark.executor.cores=4

Mesos Framework Authentication

- Mesos supports framework authentication.
- Roles can be set per framework
  - Impacts the relative weight of resource allocation
- Optional authentication information to allow the framework to be connected to the master.

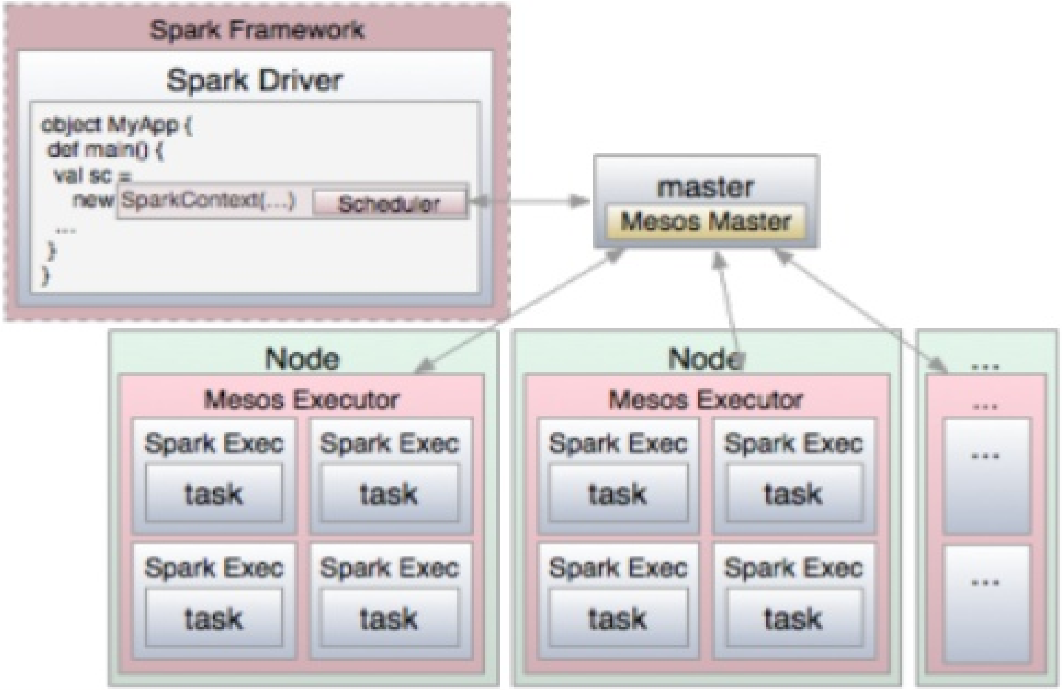
Getting rid of fine-grained mode?

Coarse-grained Mode



```
object MyApp {
 def main() {
  val sc =
    new SparkContext(...)
  ...
 }
}
```

Fine-grained Mode



```
object MyApp {
 def main() {
  val sc =
    new SparkContext(...)
  ...
 }
}
```

Getting rid of fine-grained mode?

- Why two modes?
  - –FG uses resources more efficiently, because of start-on-demand and Spark executor+task are removed when no longer needed.
  - –CG holds onto all allocated tasks until the job finishes.

  - –But *that* makes CG faster to start tasks; nice for interactive jobs (e.g., SQL queries).
  - –While FG has a longer start up time.

Getting rid of fine-grained mode?

- Why two modes?
  - –Until recently, only ONE CG executor allowed per worker node.
  - –Makes it harder to exploit all of the node's resources.

Getting rid of fine-grained mode?

- Today:
  - –*Dynamic Allocation* reclaims unused executors.
    - •(Although running this service on every node is a disadvantage)
  - –Allows more than one CG executor per node.
- Hence, the advantages of FG are becoming less important.

Getting rid of fine-grained mode?

- Spark has lots of redundant code to implement both modes.
- So, to simplify the code base and operations, FG is now *deprecated*, but it can’t be removed yet.

Running Deep Learning on Tensorflow with Spark
on top of Mesos using GPUs in the Cloud!

Demo

What's Next for Mesos?

- Pod support
- Multiple roles support
- Event Bus
- Improved Container Security (capabilities, etc)

What’s Next for Spark on Mesos?

- GPU Support on Mesos
- Use revocable resources
- Better scheduling
  - –Strategies (e.g: Spread, Binpack)
  - –Scheduling metrics
- More integration test coverage:
  - –More cluster and job configuration options.
  - –Roles and authentication scenarios.

What's Next for Spark on Mesos?

- Make "production" easier:
  - –Easier overriding of configuration with config files outside the jars.
  - –Better documentation.
  - –Easier access to Spark UIs and logs from Mesos UIs
  - –Improved metrics.
  - –Smarter acceptance of resources offered.

THANK YOU.

tnachen@gmail.com
@tnachen
dean.wampler@lightbend.com
@deanwampler

SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE
JUNE 6-8, 2016 SAN FRANCISCO