

Работа с большими данными



01

Введение. Распределенные вычисления. MapReduce.

02

HDFS. Apache Spark. RDD.

03

Базы данных. Spark SQL. Хранение больших данных.

04

Подробнее о модели вычислений Spark. Знакомство со Scala.

05

Алгоритмы машинного обучения на больших данных. spark.ml.

06

Рекомендательные системы. Виды. Их метрики. Spark.ML

07

Обработка потоковых данных. Structured streaming и интеграция с spark.ml.

08

Модели в продакшен. Управление кластером.

8. Модели в продакшен. Управление кластером.

План:

1. Вспоминаем. Кластер Spark. Data locality.
2. Dependency Hell - проблема зависимостей. Как решать?
3. Начало - виртуальные окружения. Виртуальные машины.
4. Контейнеры. Docker.
5. Кластер Spark. Типы менеджеров.
6. Mesos - аналог ОС для кластера.
7. ZooKeeper quorum.

Вспоминаем. Кластер Spark. Data locality.

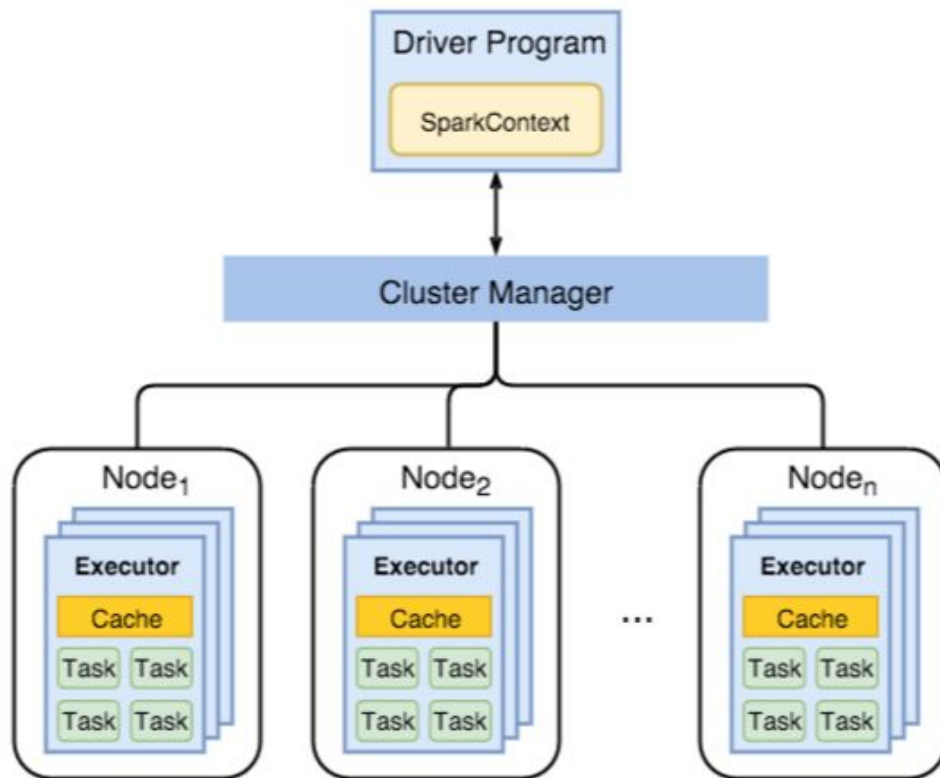
Локальность данных

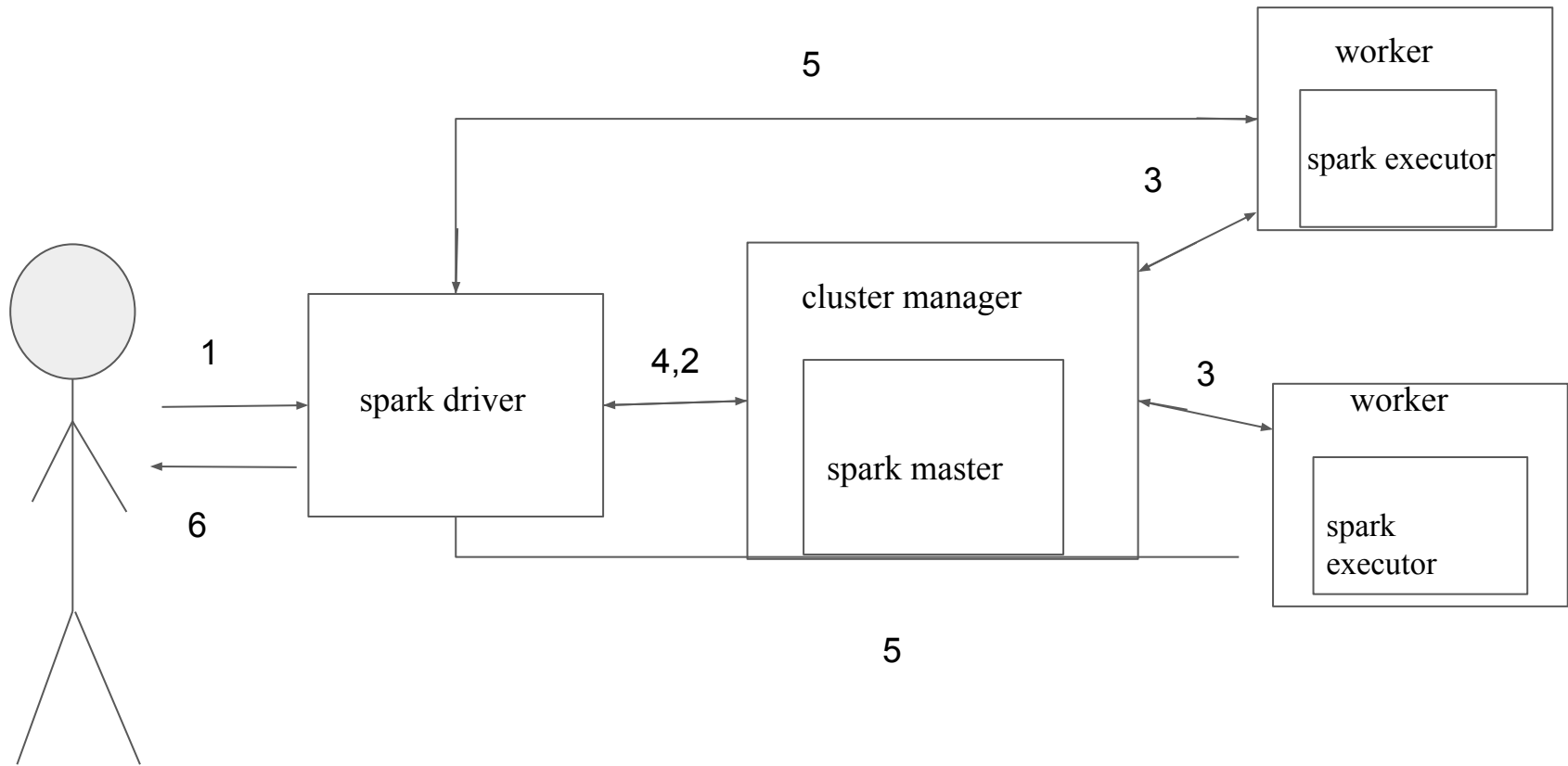
Локальность данных обеспечивается сериализацией кода.

Сериализация — процесс перевода какой-либо структуры данных в последовательность байтов. Обратной к операции сериализации является операция десериализации (структуризации) — восстановление начального состояния структуры данных из битовой последовательности.

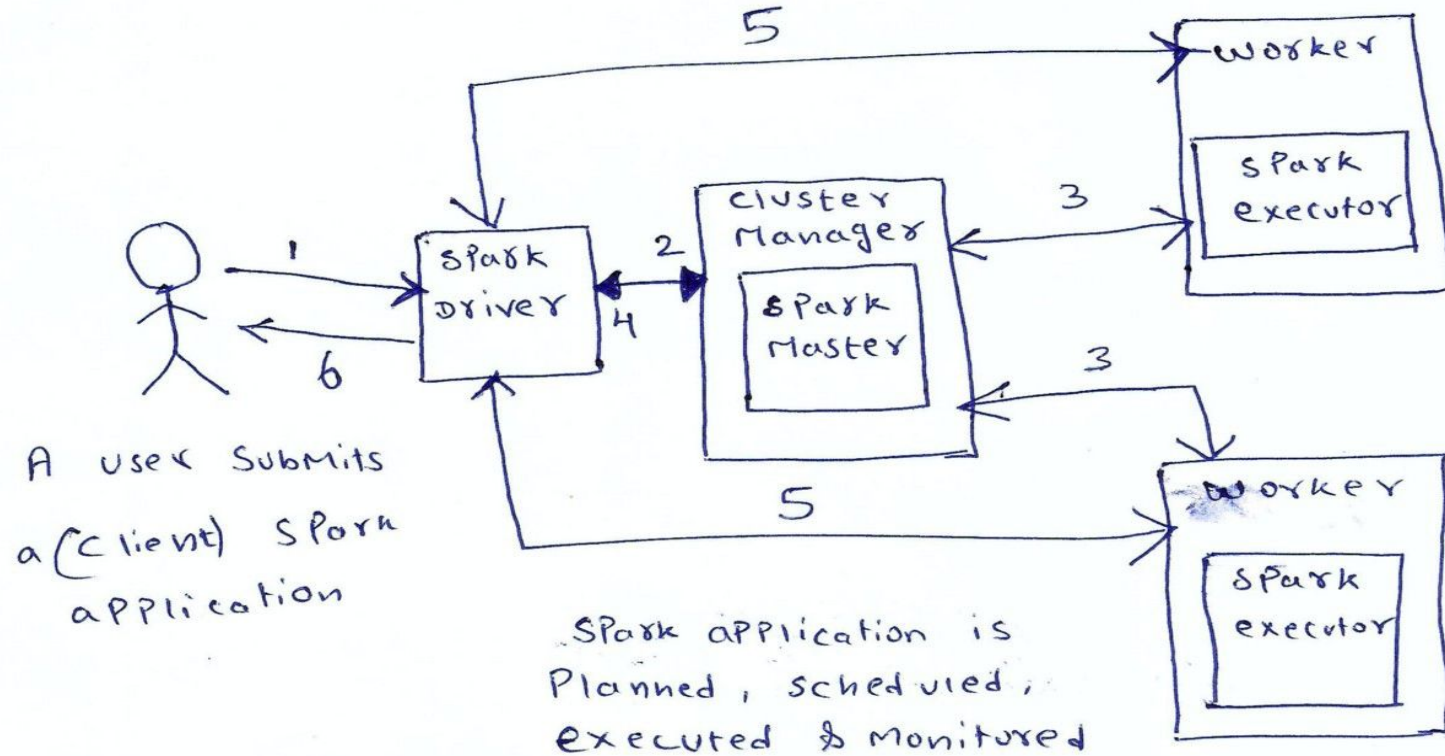
Кластер Spark.

Кластер Spark





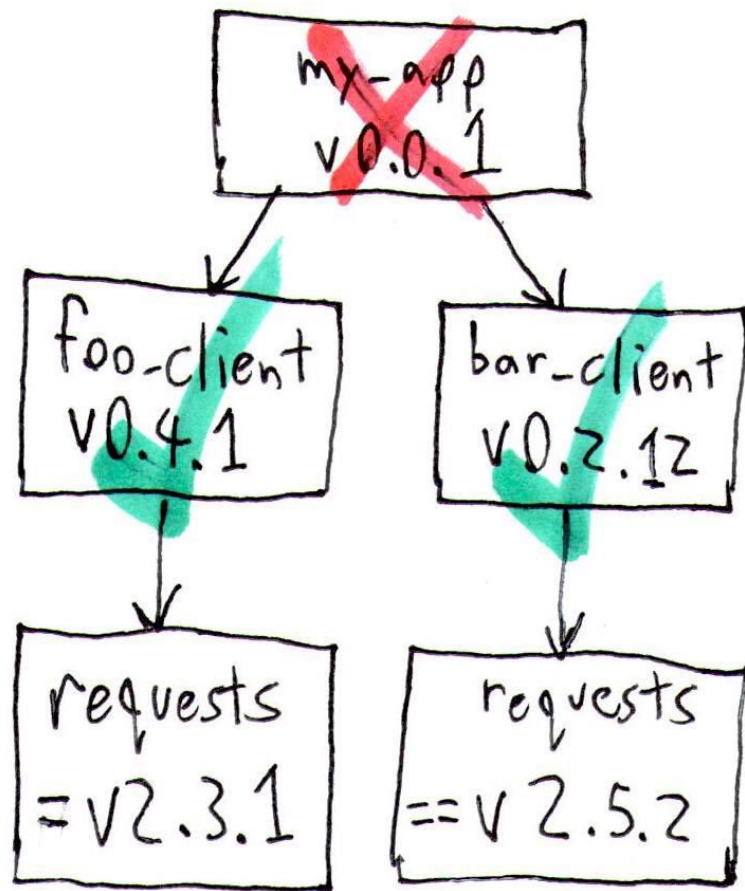
Anatomy of A SPARK Application



Кластер Spark

Приложения Spark работают как независимые наборы процессов в кластере, координируемые объектом `SparkContext` в вашей основной программе (называемой программой драйвера).

Dependency hell



Как решать?

Изолируем python - виртуальное окружение

Изолируем python - виртуальное окружение

Виртуальное окружение создается путём копирования бинарного файла Python в локальную папку (ENV/bin/python). Родительская папка содержит символические ссылки на файлы стандартной библиотеки питона. По сути - делаем свежую копию чистого питона.

<https://habr.com/ru/post/418579/>

Начало - виртуальные машины.

Начало - виртуальные машины.

Виртуальные машины делят между собой физические ресурсы хоста:

- процессор,
- память,
- дисковое пространство,
- сетевые интерфейсы.

На каждой ВМ устанавливаем нужную ОС и запускаем приложения. Недостатком такого подхода является то, что значительная часть ресурсов хоста расходуется не на полезную нагрузку(работа приложений), а на работу нескольких ОС.

Контейнеры. Docker.

Docker

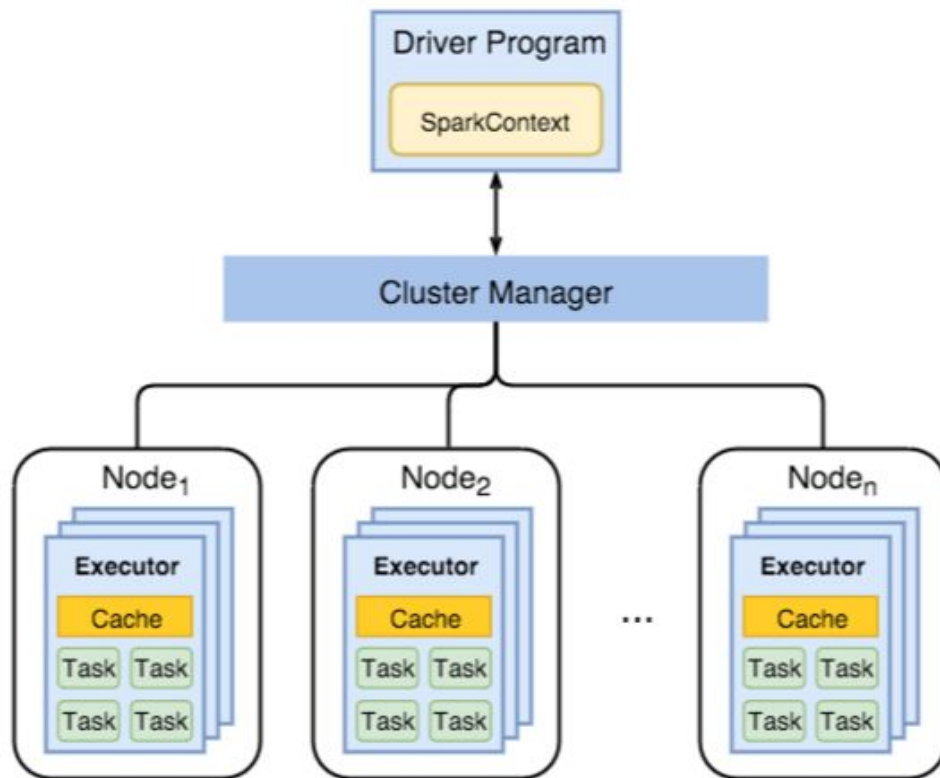
В отличие от ВМ контейнеры изолированно используют свой кусочек ОС:

- файловая система
- дерево процессов
- сетевые интерфейсы

Т.о. приложение, запущенное в контейнере думает, что оно одно во всей ОС. Изоляция достигается за счет использования таких Linux-механизмов, как **namespaces** и **control groups**.

Кластер Spark. Типы менеджеров.

Кластер Spark



Типы менеджеров кластеров

В настоящее время система поддерживает несколько менеджеров кластеров:

- Standalone – простой менеджер кластеров, входящий в состав Spark, который позволяет легко настроить кластер.
- Apache Mesos – это общий менеджер кластеров, который также может запускать приложения Hadoop MapReduce и service.
- Hadoop YARN – менеджер ресурсов в Hadoop 2.
- Kubernetes – система с открытым исходным кодом для автоматизации развертывания, масштабирования и управления контейнерными приложениями.

Типы менеджеров кластеров

В частности, для работы в кластере SparkContext может подключаться к нескольким типам менеджеров кластеров (собственный автономный менеджер кластеров Spark, Mesos или YARN), которые распределяют ресурсы между приложениями. После подключения Spark получает исполнителей на узлах кластера, которые являются процессами, выполняющими вычисления и хранящими данные для вашего приложения. Затем он отправляет ваш код приложения (определенный файлами JAR или Python, переданными в SparkContext) исполнителям. Наконец, SparkContext отправляет задачи исполнителям для запуска.

Mesos - аналог ОС для кластера.

Apache Mesos

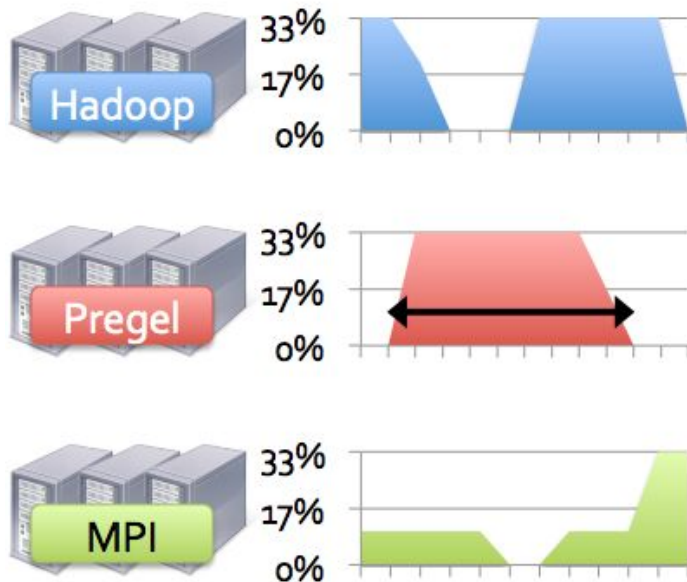
Apache Mesos — это централизованная отказоустойчивая система управления кластером. Она разработана для распределенных компьютерных сред с целью обеспечения изоляции ресурсов и удобного управления кластерами подчиненных узлов (mesos slaves)

вместо деления физической машины на кучу виртуальных, Mesos предлагает их объединять в одно целое, в единый виртуальный ресурс.

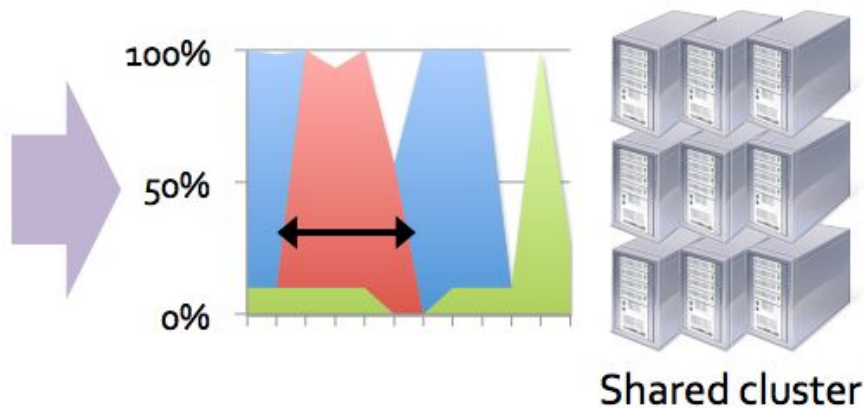
<https://habr.com/ru/post/308812/>

Необходимость выполнить различные типы задач

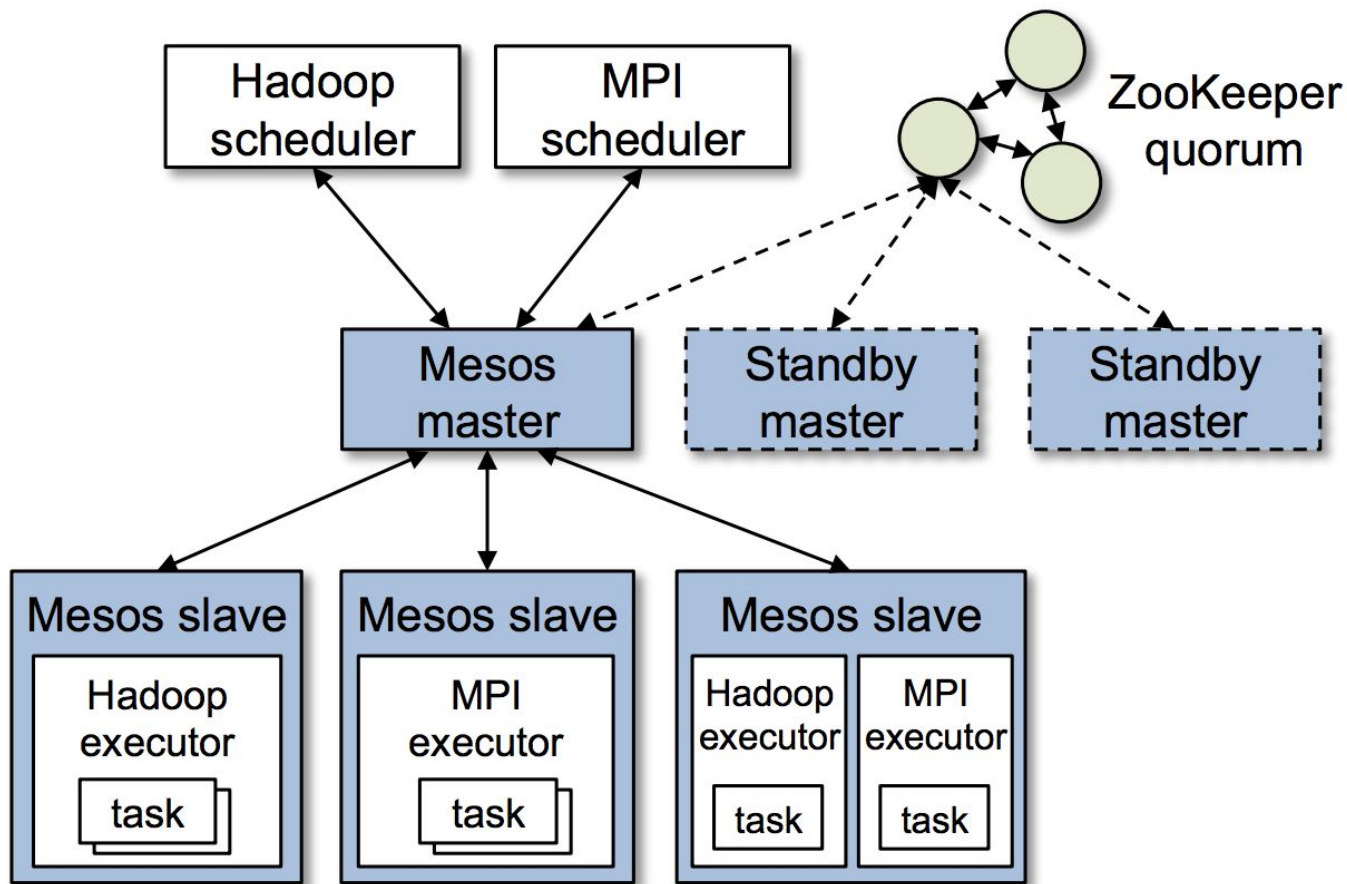
Today: static partitioning



Mesos: dynamic sharing



ZooKeeper quorum.



ZooKeeper

ZooKeeper - это централизованная служба для хранения информации о конфигурации, именования, обеспечения распределенной синхронизации и предоставления групповых услуг.