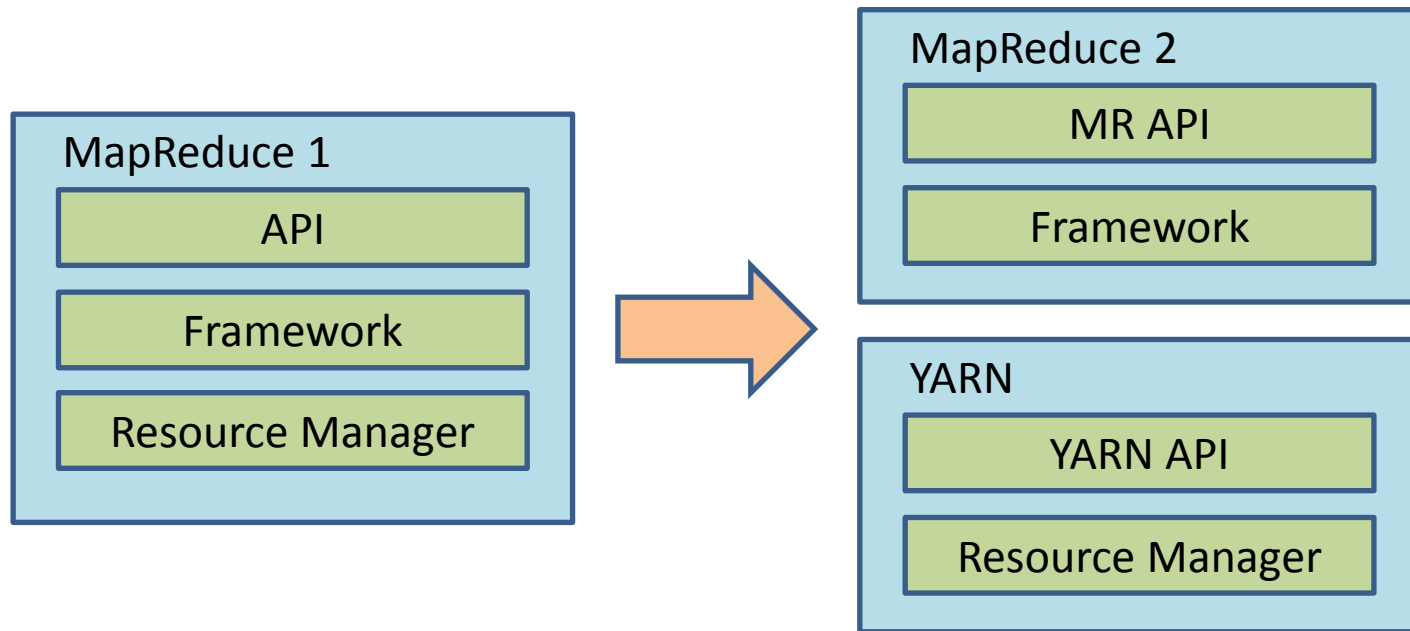


YARN

Yet Another Resource Negotiator

YARN и MapReduce 2.0



YARN

- Yet Another Resource Negotiator
- Разработан в Yahoo! в 2010 году
- Берет на себя функции управления ресурсами
- Может запускать различные типы задач (MapReduce один из вариантов)

Недостатки MapReduce v1

- Жесткое разделение ресурсов кластера
- Нет возможности разделять ресурсы с не MapReduce задачами
- Ограничения в масштабируемости:
 - 4 000 серверов
 - 40 000 запущенных задач

Жесткое разделение ресурсов

- Пусть у нас есть 10 серверов по 16 ядер
- 1 ядро для TaskTracker
- Всего 150 ядер
- Пусть ресурсы делятся следующим образом:
 - 100 мапперов
 - 50 редьюсеров

Жесткое разделение ресурсов

Пусть есть задача:

- нужно 300 мапперов
- каждый маппер работает 10 минут
- если все данные для редьюсеров обрабатывать одним редьюсером, то это займет 100 минут

Какое время потребуется для работы задачи?

Жесткое разделение ресурсов

Меняем распределение мапперов и редьюсеров:

- мапперов: **120**
- редьюсеров: **30**

Время работы задачи:

Жесткое разделение ресурсов

Еще меняем:

- мапперов: **140**
- редьюсеров: **10**

Время работы задачи:

Жесткое разделение ресурсов

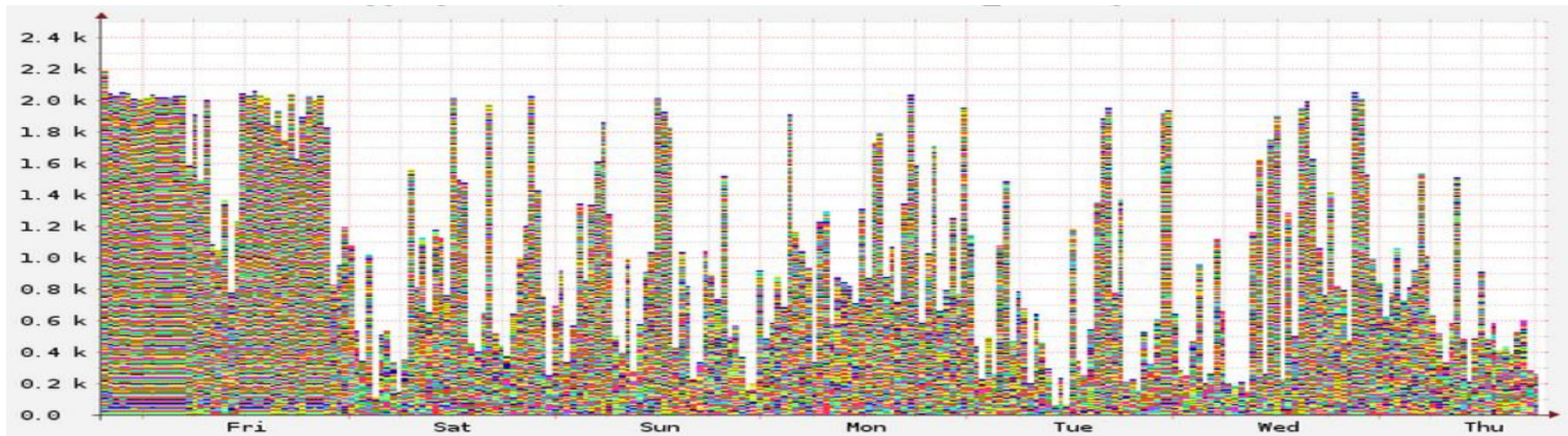
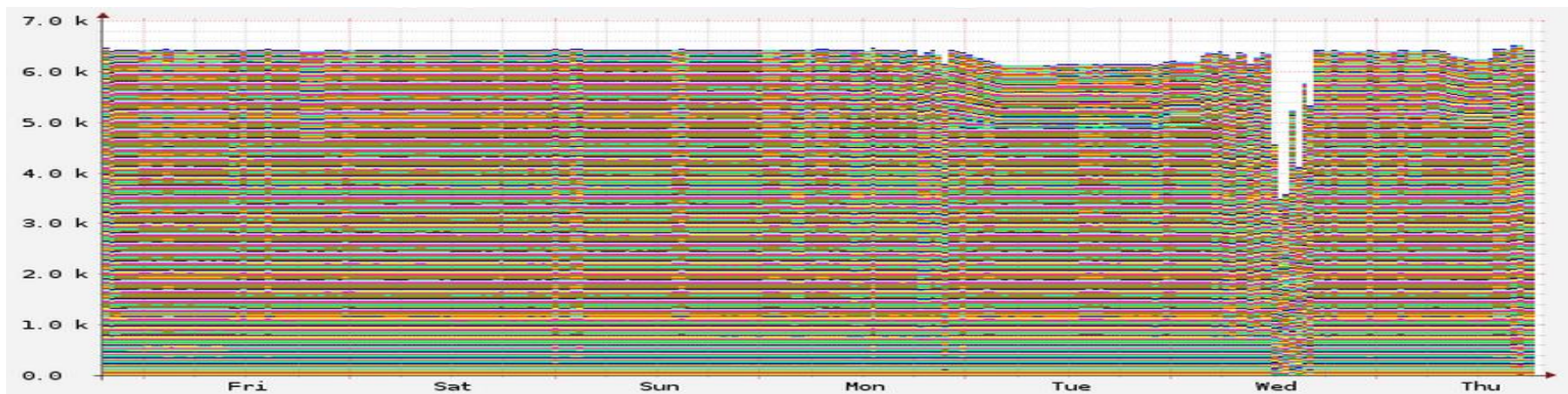
Можно найти оптимальное разбиение для данной задачи, найдя минимум функции:

$$\begin{cases} T(x, y) = \frac{300}{x} * 10 + \frac{100}{y} \\ x + y = 150 \end{cases}$$

Динамическое распределение ресурсов

Пусть любой ресурс может быть как
маппером, так и редьюсером

Время работы задачи:



YARN

- Нет понятия “слоты”
 - Сервер имеет “ресурсы” (память, процессор)
- Нет деления на мапперы и редьюсеры
 - Можно запускать не только MR задачи

Компоненты YARN

Resource Manager (RM)

- Запущен на отдельном сервере
- Управляет глобальным распределением ресурсов
- Разрешает конфликты между конкурирующими приложениями



Компоненты YARN

Node Manager (NM)

- Запущен на всех нодах кластера
- Взаимодействует с RM

Node
Manager



Node
Manager



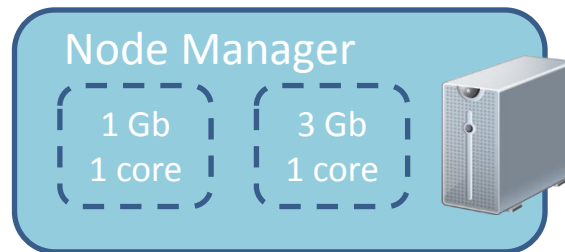
Node
Manager



Компоненты YARN

Containers

- Создается RM по запросу
- Захватывает определенное число ресурсов на ноде (память, CPU)
- Приложение запускается на одном или нескольких containers

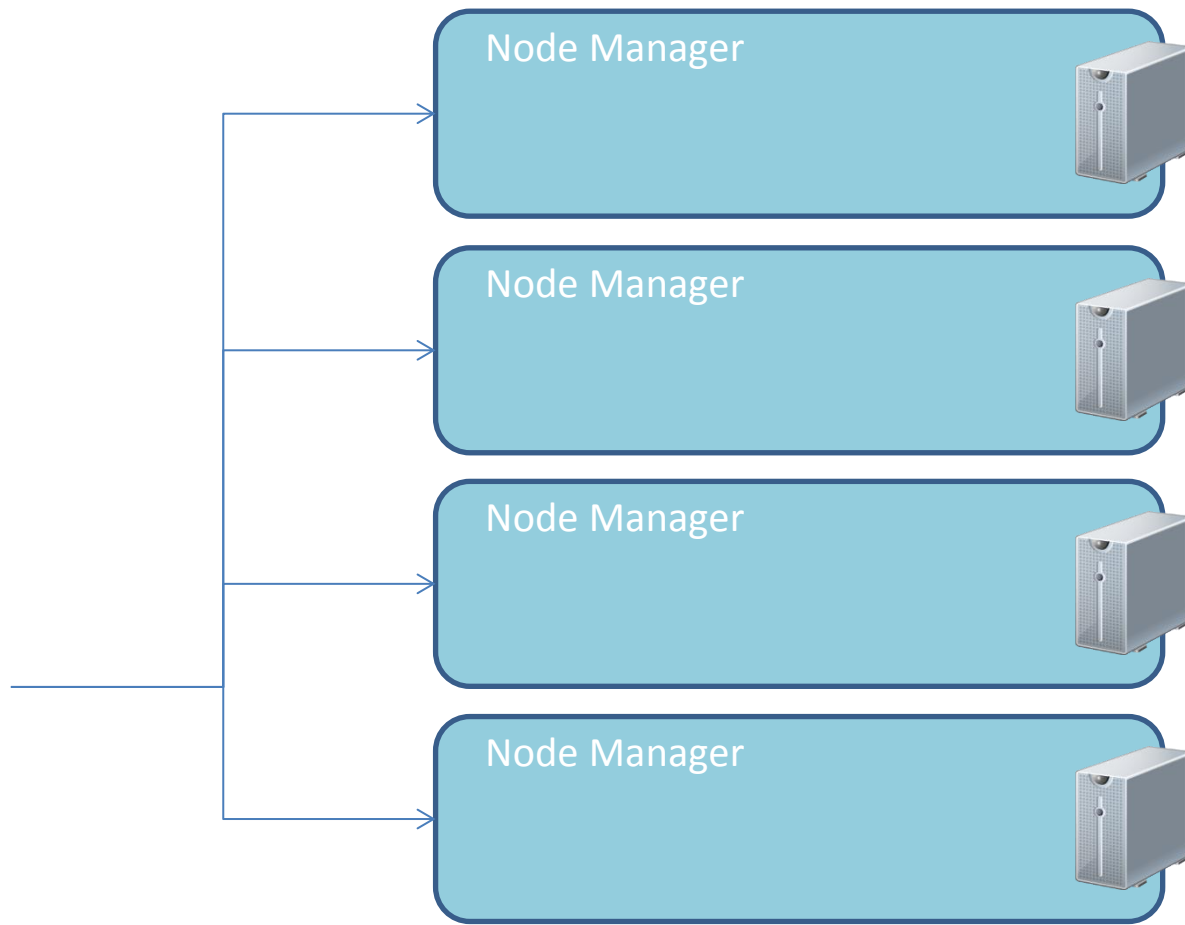


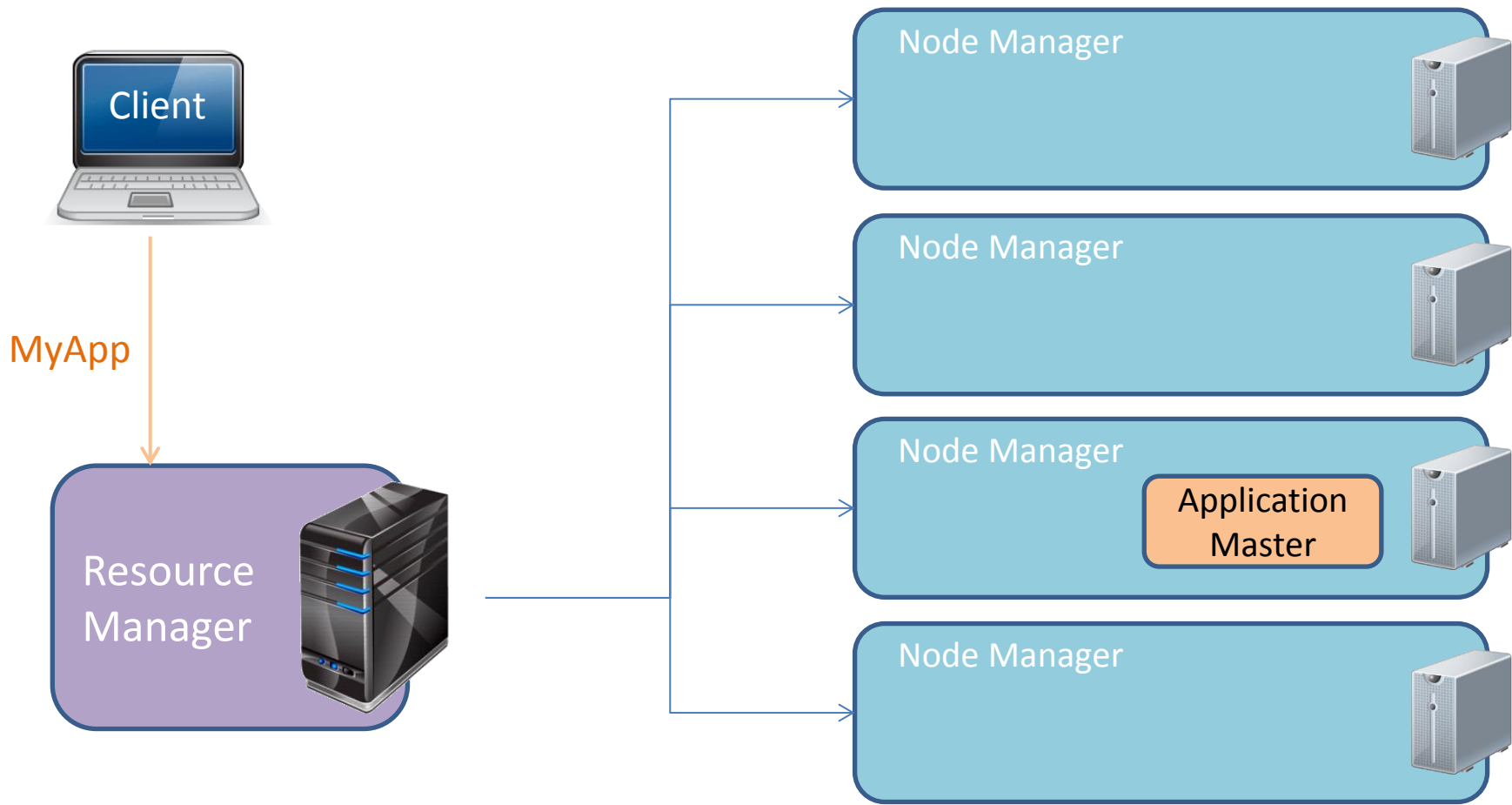
Компоненты YARN

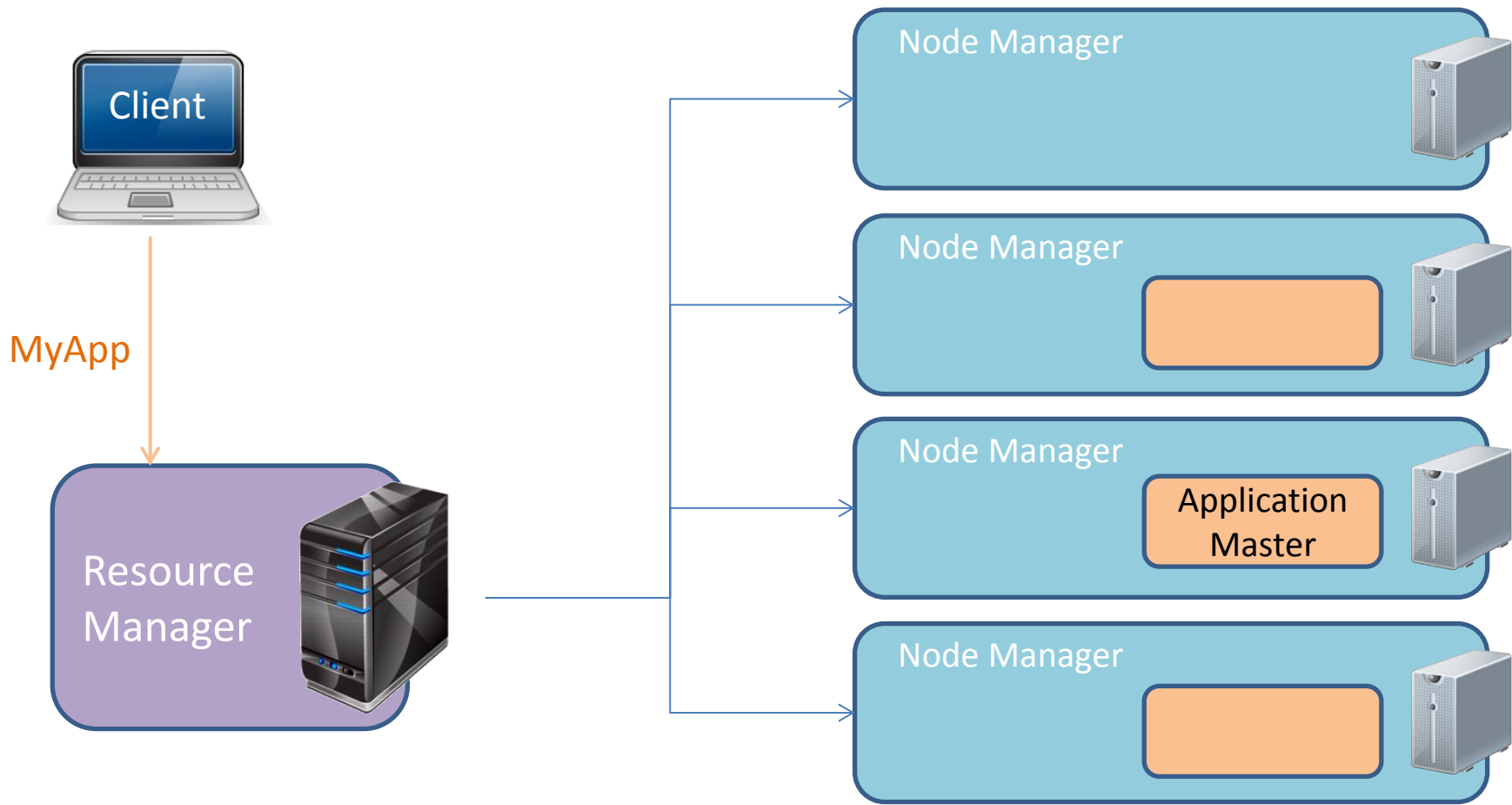
Application Master (AM)

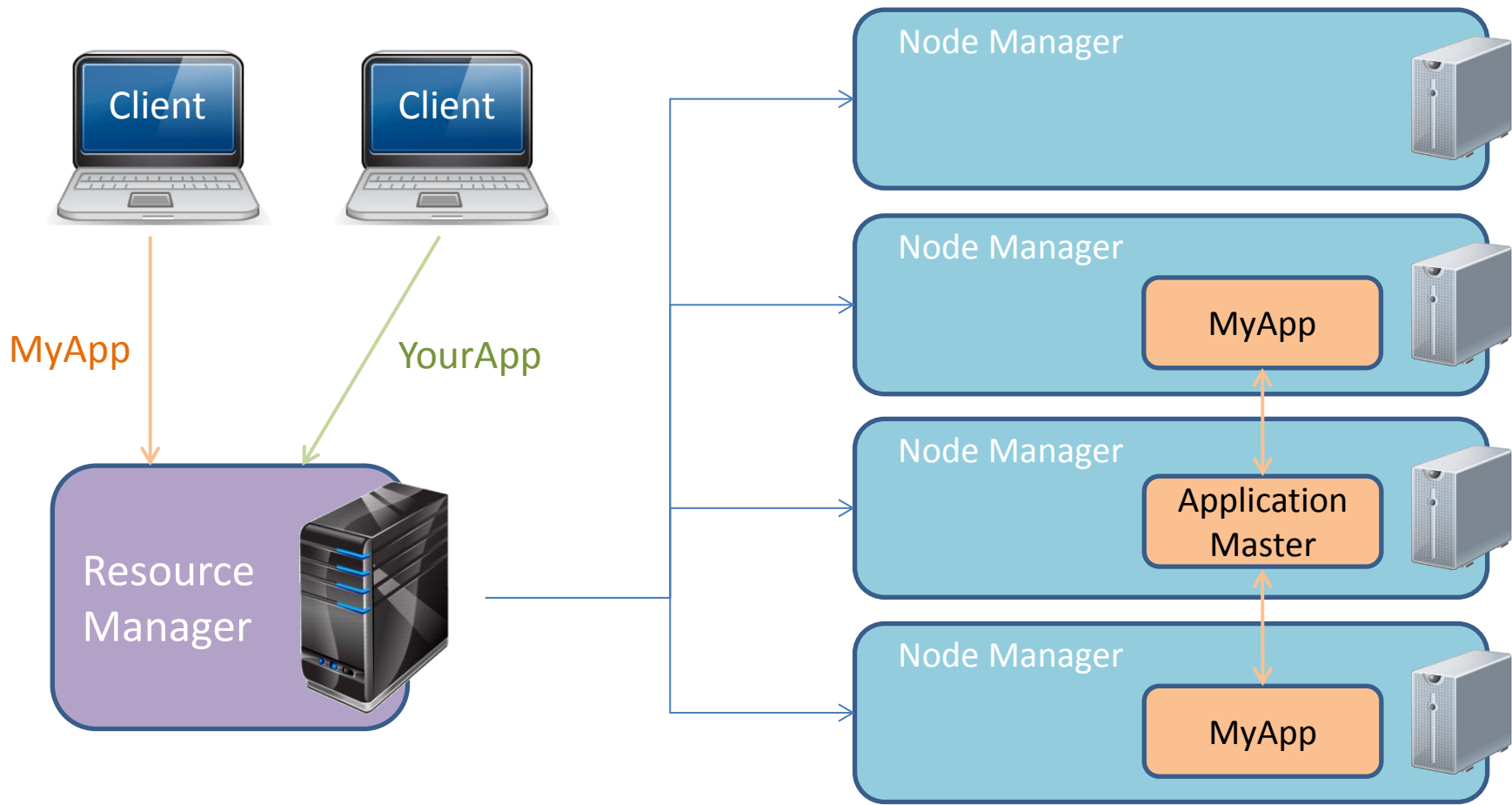
- Один для каждого приложения
- Зависит от типа задачи (свой для MR задач)
- Запускается в container
- Запрашивает другие containers для запуска приложения

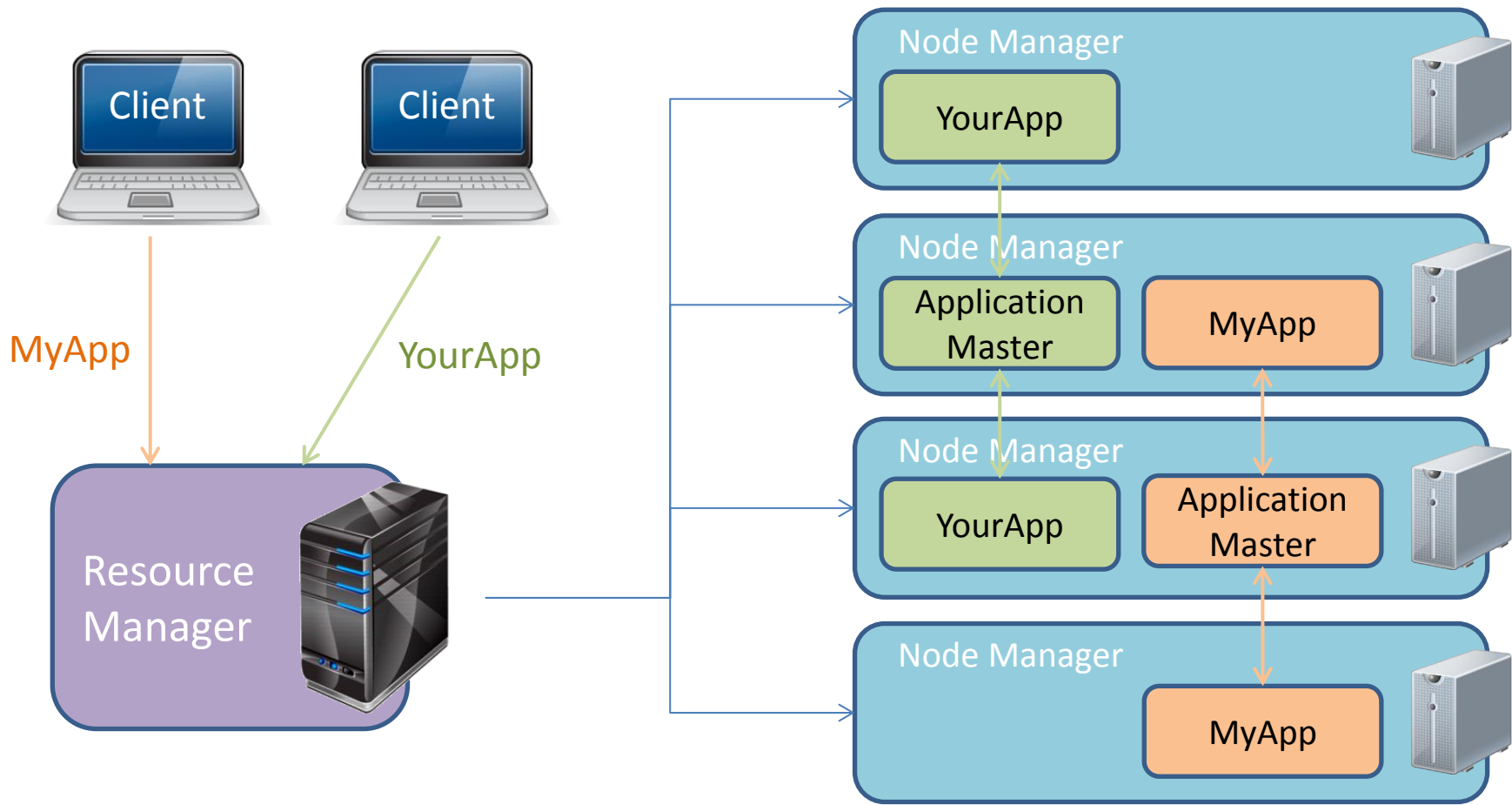












Функции Resource Manager

- Управляет нодами
- Управляет containers
- Взаимодействует с Application Manager
- Отвечает за защиту данных

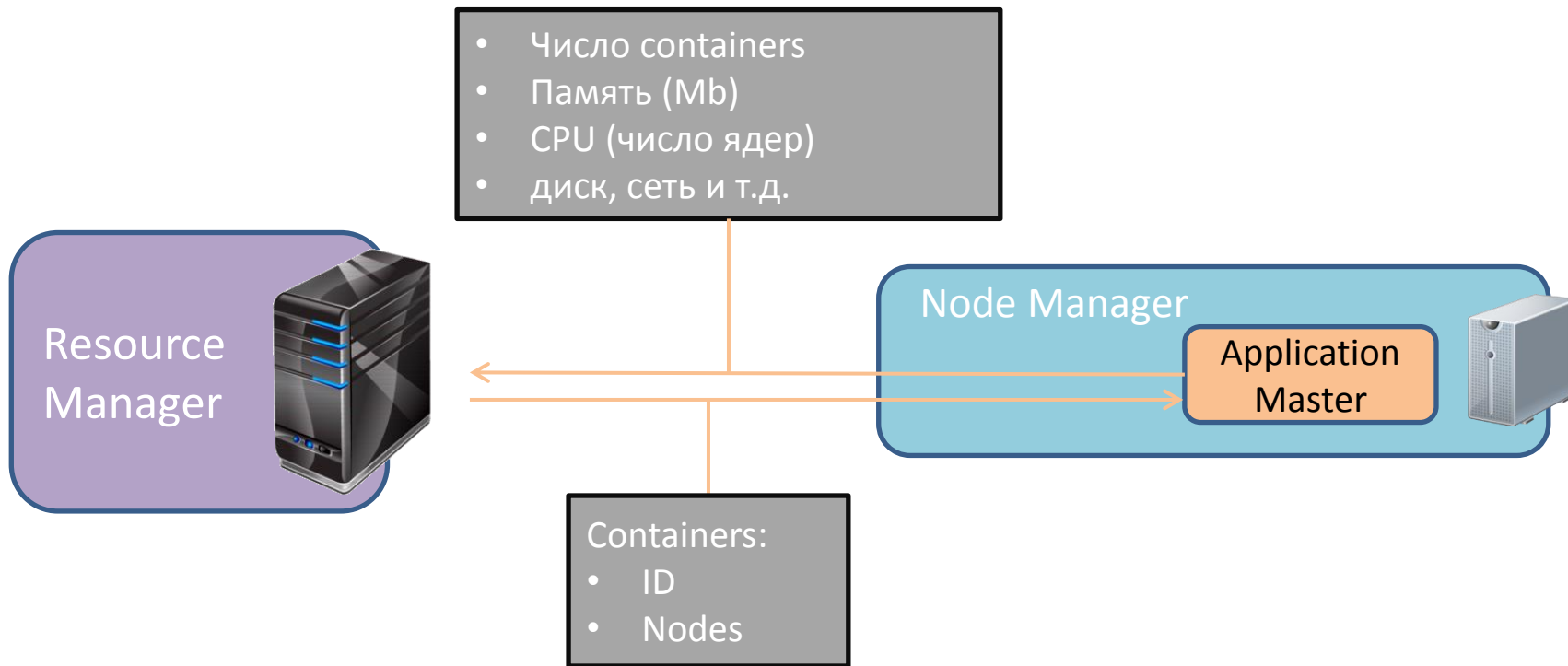


Функции Node Manager

- Взаимодействует с RM
- Управляет процессами в контейнере
- Логирует данные приложений
- Поддерживает защиту ACL на уровне ноды

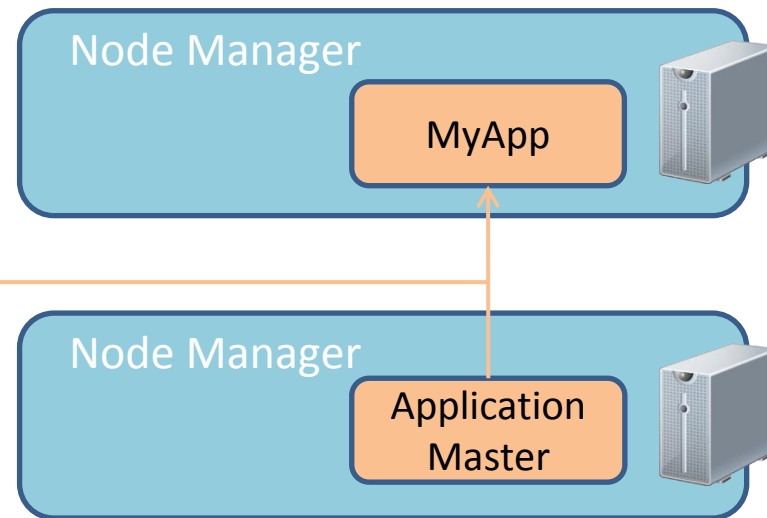


Запрос ресурсов



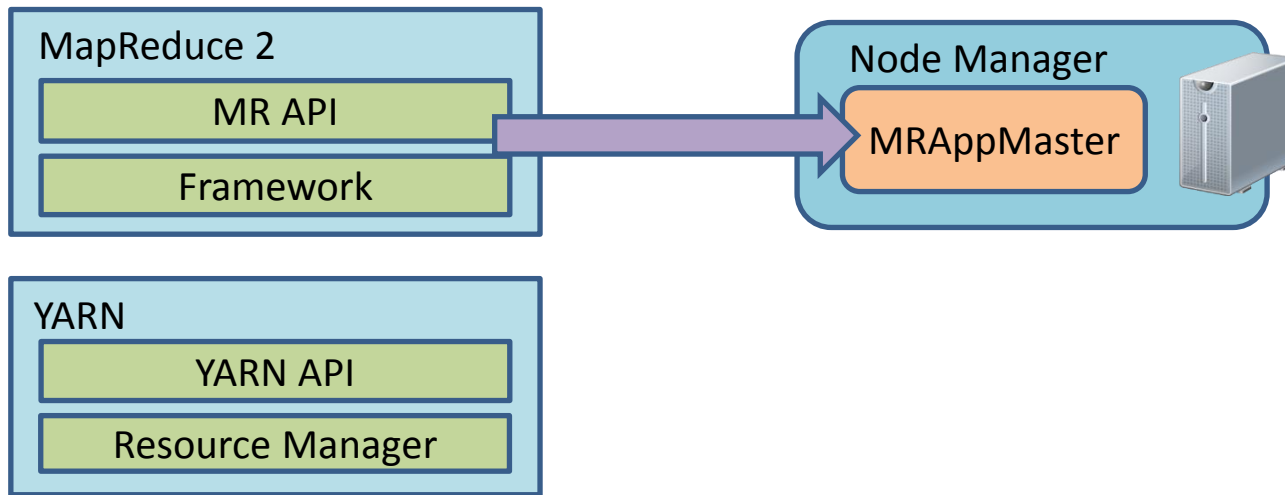
Запуск container

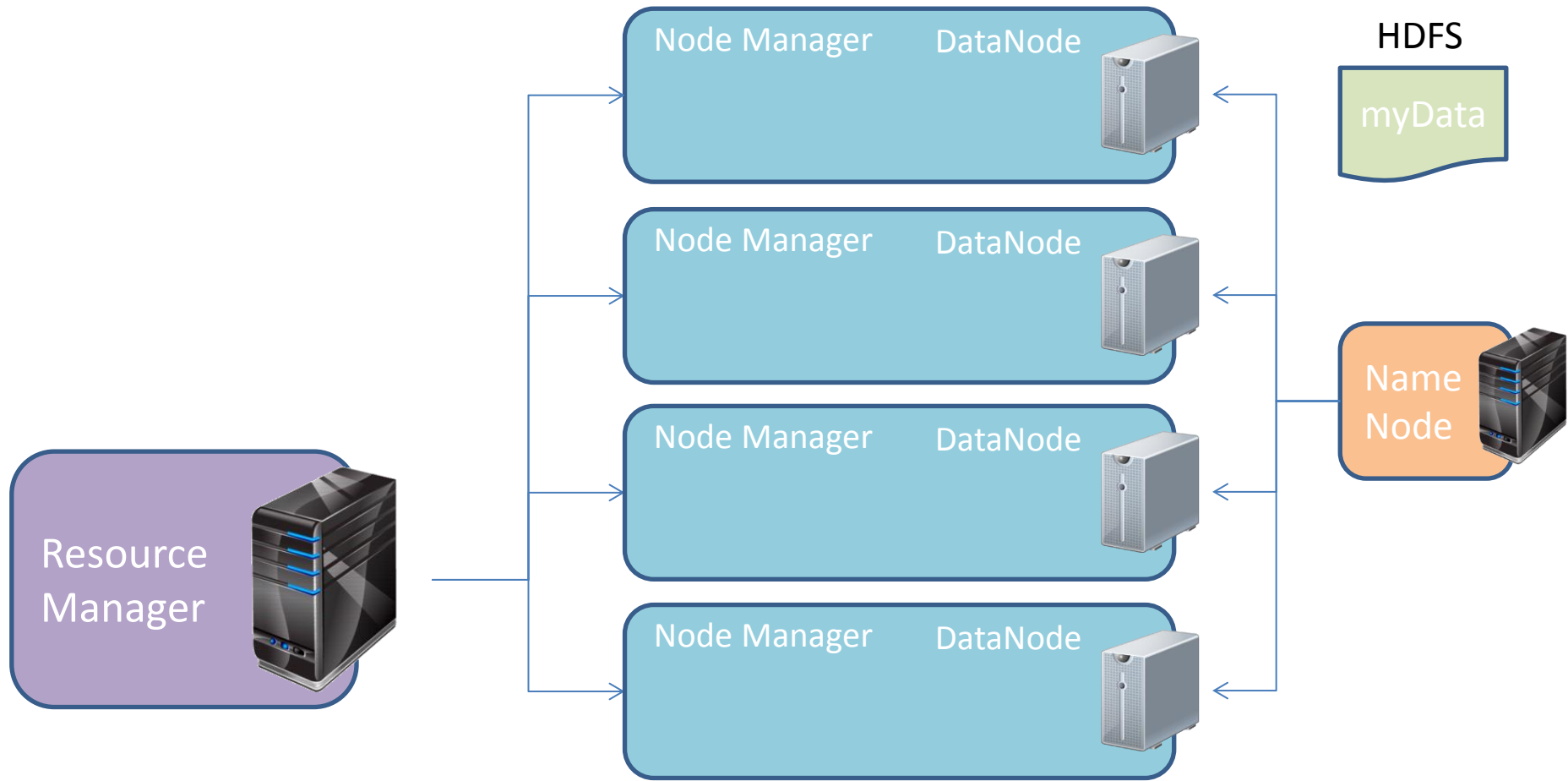
- Container ID
- Команды (для запуска приложения)
- Окружение (конфигурация)
- Локальные ресурсы (код задачи, HDFS файлы)



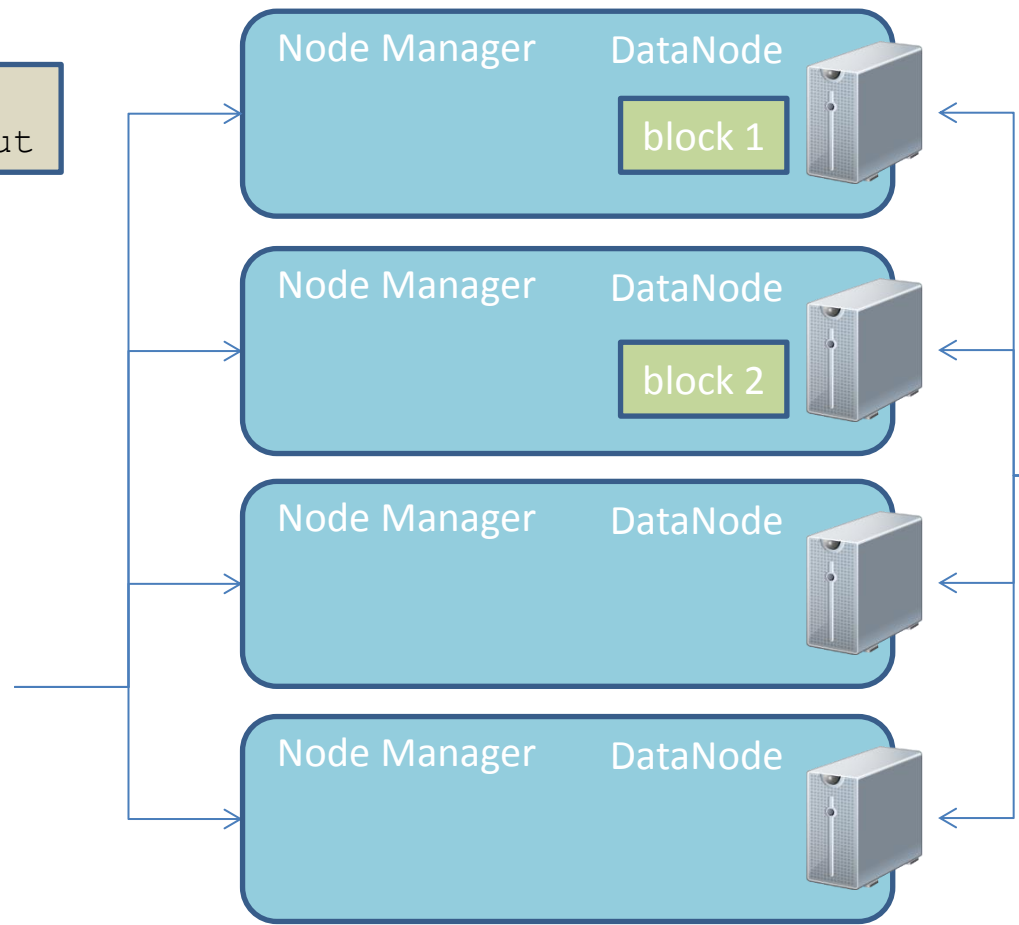
YARN и MapReduce

- YARN не знает какой тип приложения он запускает
- MRAppMaster – AM для MapReduce





```
$ hadoop jar wc.jar  
WordCount mydata output
```



```
$ hadoop jar wc.jar  
WordCount mydata output
```



Resource
Manager



Node Manager

DataNode

block 1



Node Manager

DataNode

block 2



Node Manager

DataNode

MRAppMaster

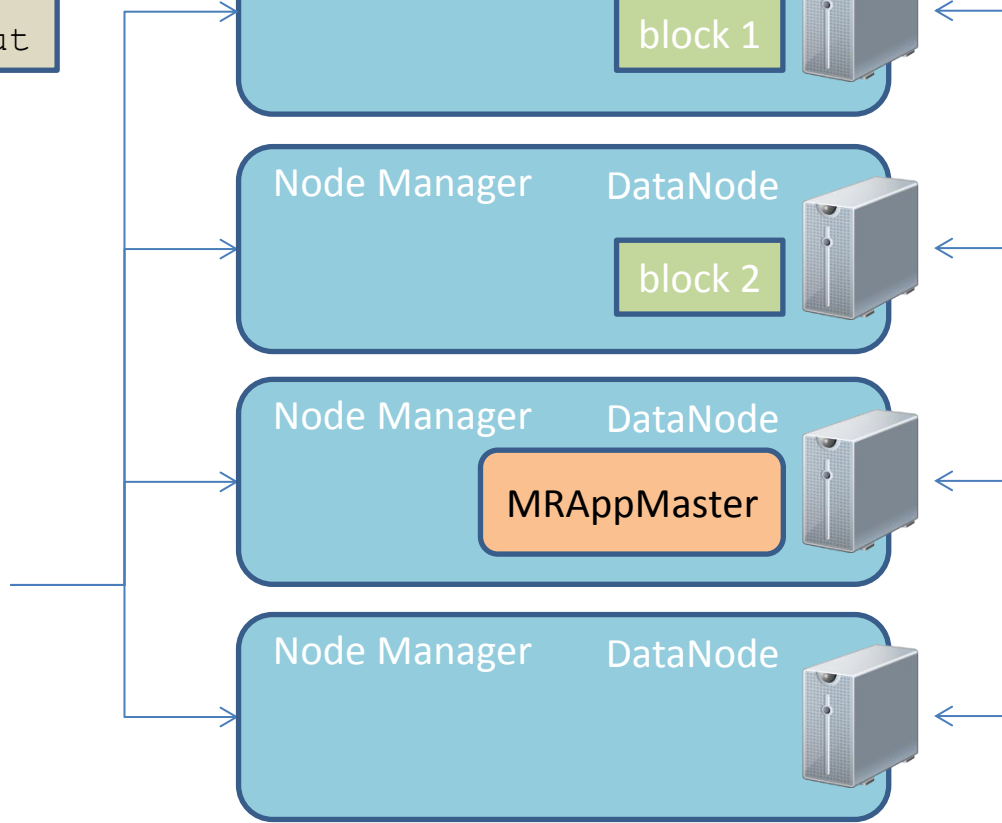


Node Manager

DataNode



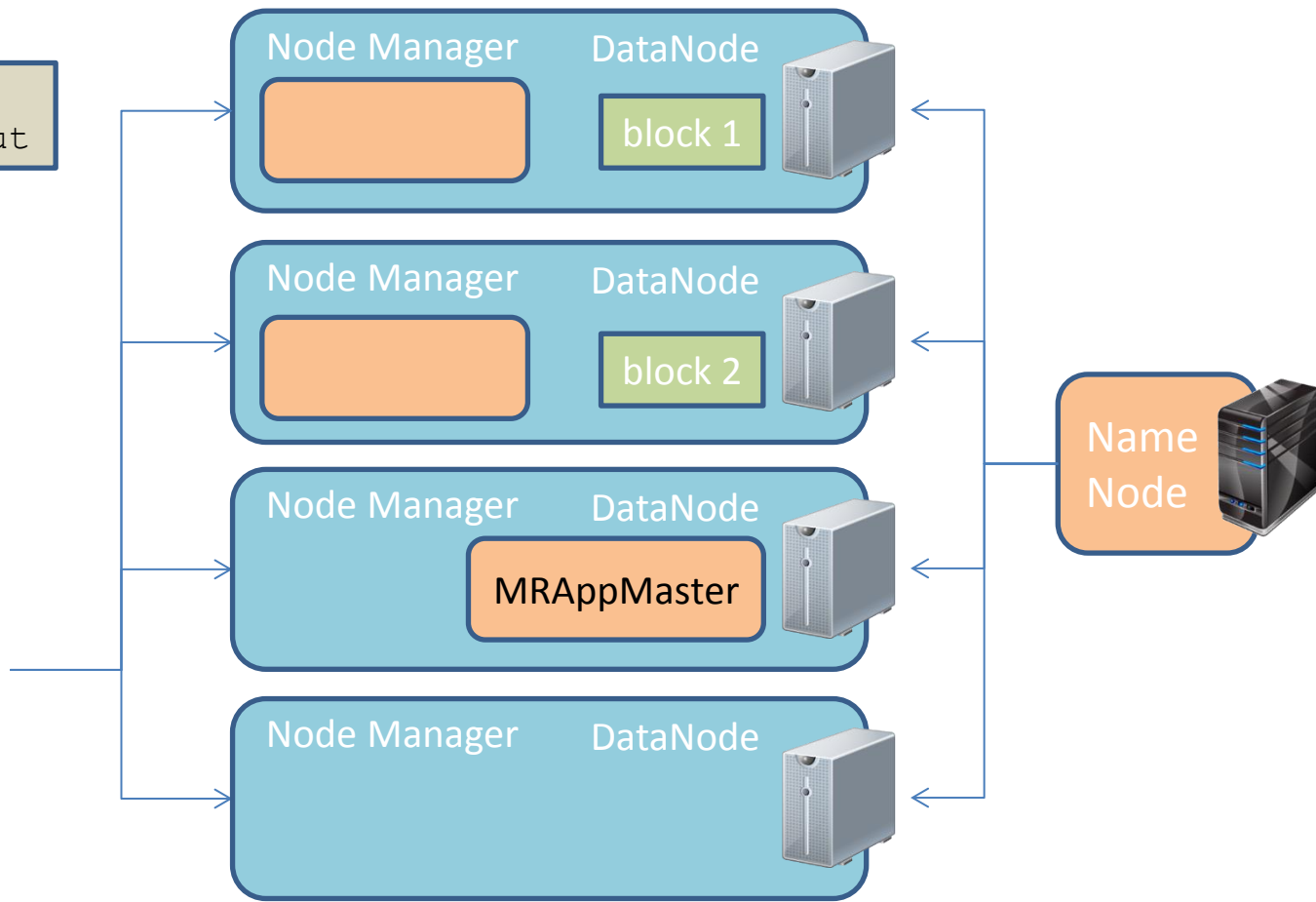
Name
Node



```
$ hadoop jar wc.jar  
WordCount mydata output
```



Resource
Manager



```
$ hadoop jar wc.jar  
WordCount mydata output
```



Resource
Manager



Node Manager

WordCount
Map Task

DataNode

block 1



Node Manager

WordCount
Map Task

DataNode

block 2



Node Manager

MRAppMaster

DataNode

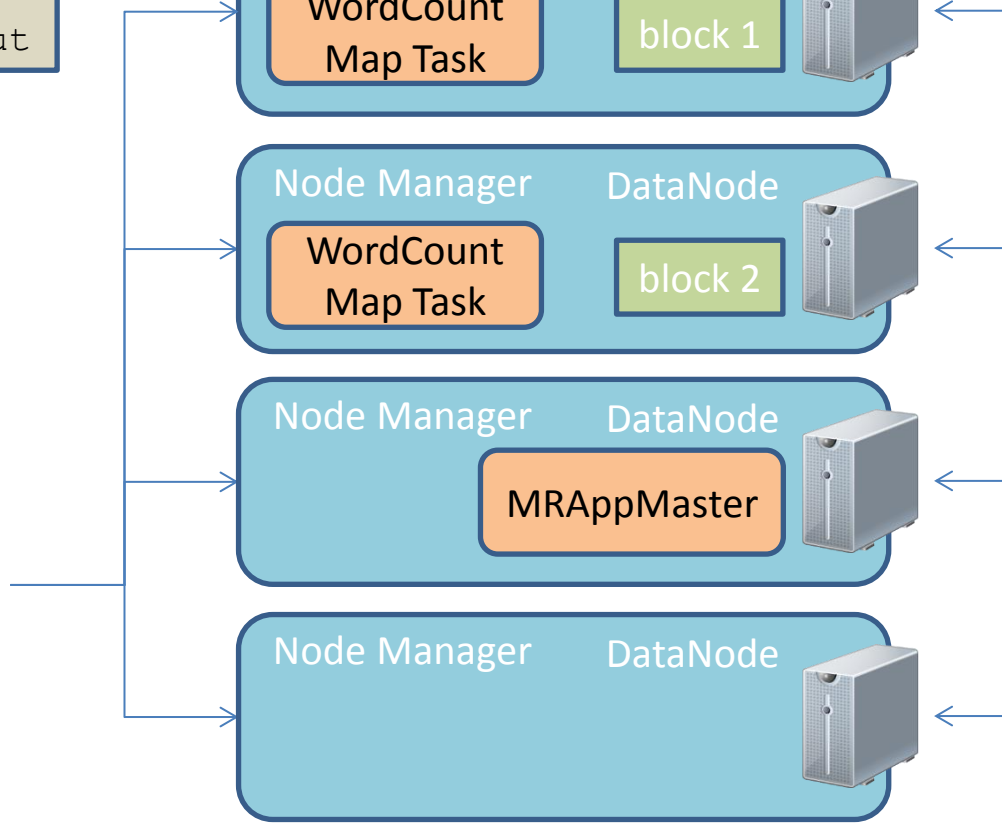


Node Manager

DataNode



Name
Node



```
$ hadoop jar wc.jar  
WordCount mydata output
```



Resource
Manager



Node Manager

DataNode

block 1



Node Manager

DataNode



Node Manager

DataNode

MRAppMaster

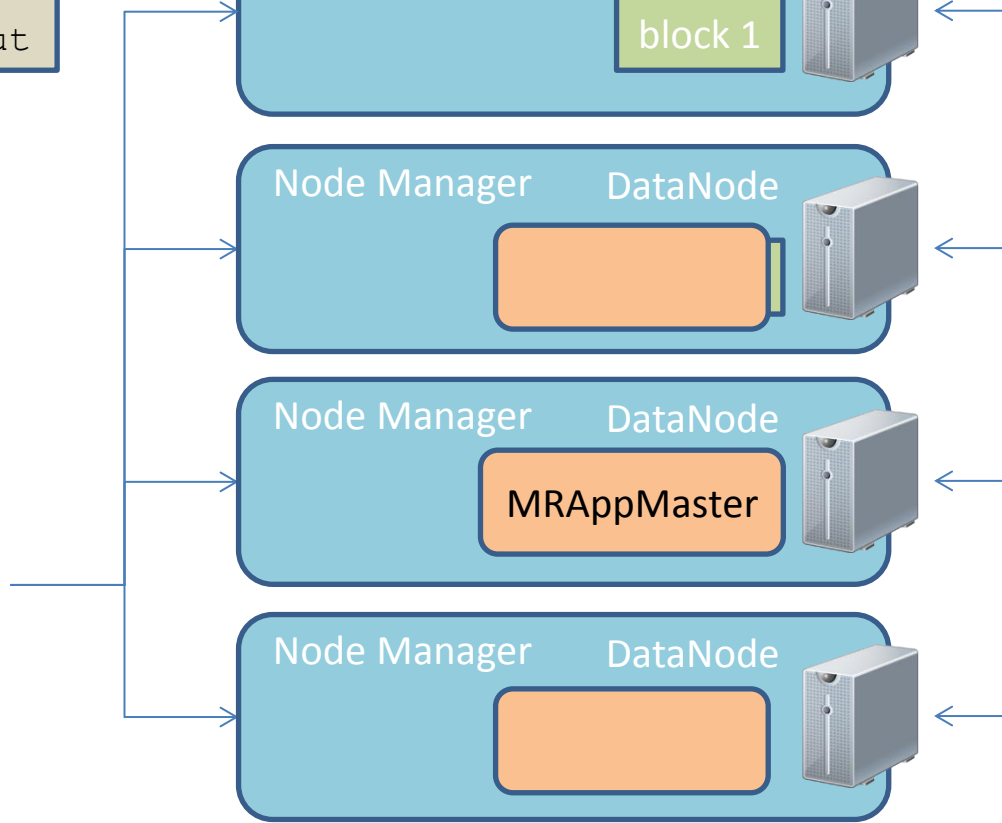


Node Manager

DataNode



Name
Node




```
$ hadoop jar wc.jar  
WordCount mydata output
```



Resource
Manager



Node Manager

DataNode

block 1



Node Manager

DataNode

WordCount
Reduce Task



Node Manager

DataNode

MRAppMaster



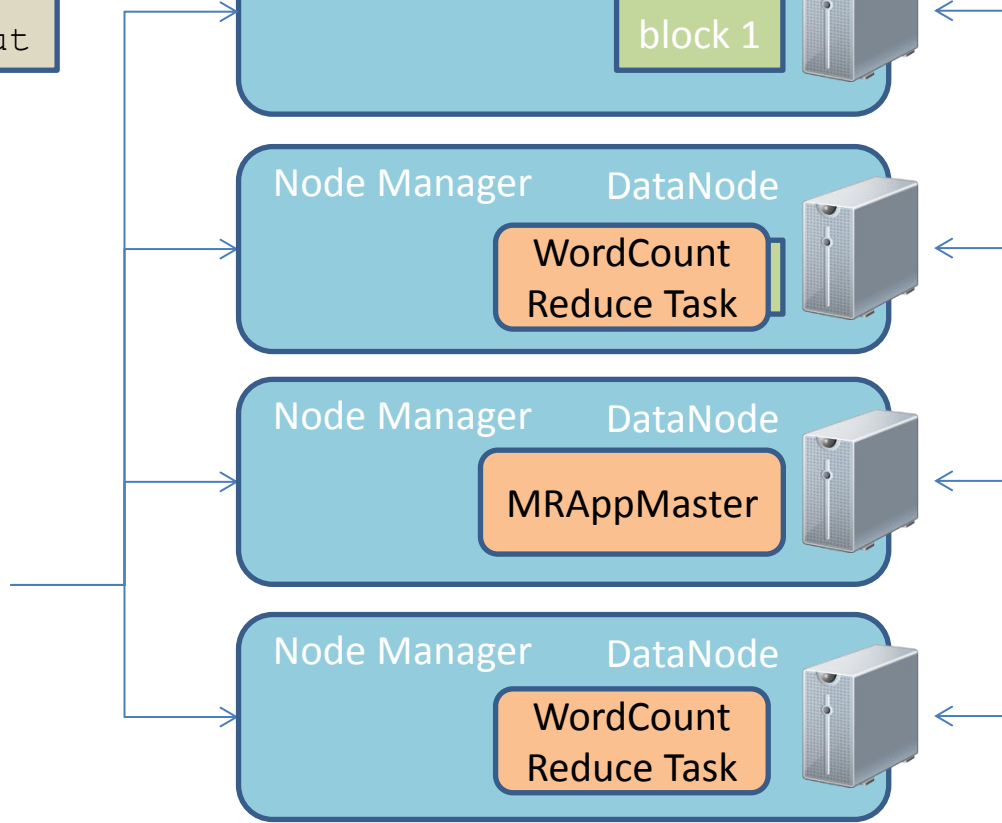
Node Manager

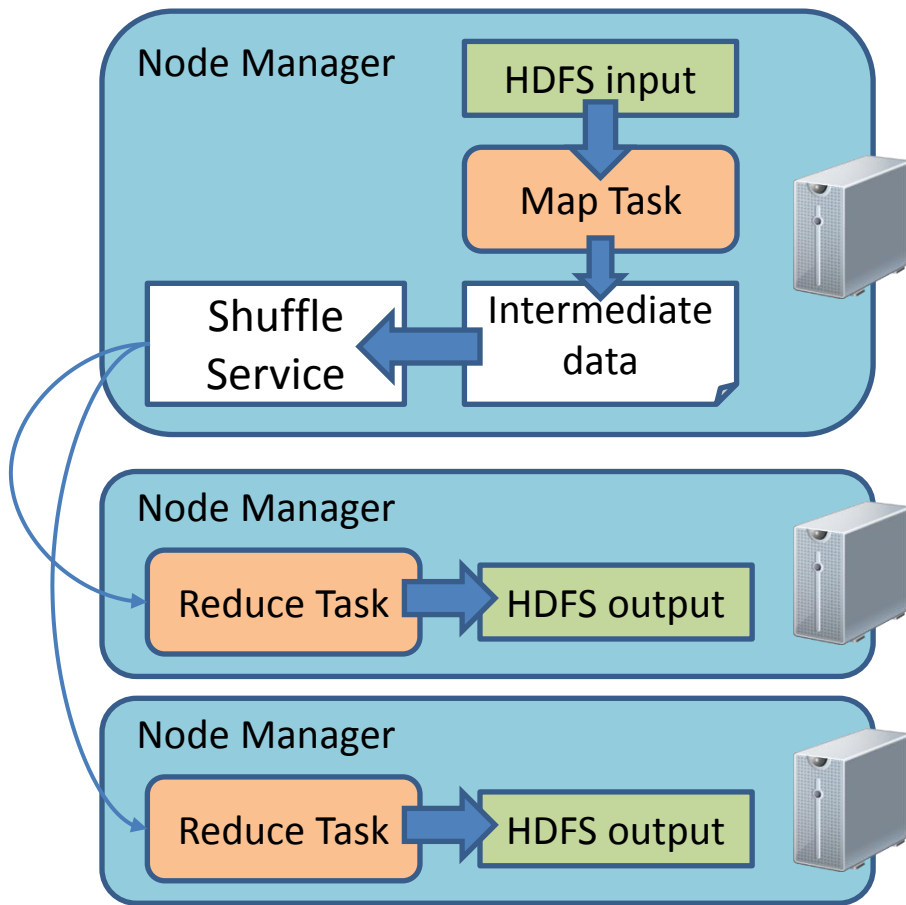
DataNode

WordCount
Reduce Task



Name
Node





Uber задачи

- **MRAppMaster** запускает **map** и **reduce** внутри одной JVM
- Условия:
 - Менее 10 мапперов
 - *mapreduce.job.ubertask.maxmaps*
 - Один редьюсер
 - *mapreduce.job.ubertask.maxreduces*
 - Размер входных данных меньше размера 1 HDFS-блока
 - *mapreduce.job.ubertask.maxbytes*
- Запуск uber-задач можно отключить
 - *mapreduce.job.ubertask.enable=false*

Управление памятью в YARN

Resource Manager выделяет память:

- минимум: ***yarn.scheduler.minimum-allocation-mb*** (1024 Mb)
- максимум: ***yarn.scheduler.maximum-allocation-mb*** (8192 Mb)

Управление памятью в YARN

- Каждый Node Manager имеет лимит на размер выделенной памяти:
 - ***yarn.nodemanager.resource.memory-mb*** (8192 Mb)
- Сумма всех containers не может превышать этот лимит
- Node Manager не создает container, если нет достаточно памяти

Управление памятью в YARN

- Размер containers для map и reduce:
 - ***mapreduce.map.memory.mb=1536***
 - ***mapreduce.reduce.memory.mb=3072***
- Размер JVM Heap:
 - ***mapreduce.map.java.opts=-Xmx1024m***
 - ***mapreduce.reduce.java.opts=-Xmx2560m***

Управление памятью в YARN

- Размер виртуальной памяти задается как коэффициент к физической памяти:
 - **yarn.nodemanager.vmem-pmem-ratio=2.1**

Управление памятью в YARN

Если **map** или **reduce** превышает лимит виртуальной или физической памяти, то **Node Manager** убивает соответствующий контейнер

```
Current usage: 2.1gb of 2.0gb physical memory used; 1.6gb  
of 3.15gb virtual memory used. Killing container.
```


Resource Manager

yarn.scheduler.maximum-allocation-mb=**8192**

yarn.scheduler.minimum-allocation-mb=**1024**

Node Manager

yarn.nodemanager.resource.memory-mb=**24576**

Max Heap
2048Mb

Max Virtual
3225.6Mb

Map JVM
-Xmx1024m

Max Heap
3072Mb

Max Virtual
6451.2Mb

Reduce JVM
-Xmx2560m

Max Heap
2048Mb

Max Virtual
3225.6Mb

AM JVM
-Xmx1024m

mapreduce.map.memory.mb=**1536**

mapreduce.map.java.opts=**-Xmx1024m**

yarn.nodemanager.vmem-pmem-ratio=**2.1**

mapreduce.reduce.memory.mb=**3072**

mapreduce.reduce.java.opts=**-Xmx2560m**

yarn.nodemanager.vmem-pmem-ratio=**2.1**

yarn.app.mapreduce.am.resource.mb=**1536**

yarn.nodemanager.vmem-pmem-ratio=**2.1**

Обработка сбоев

Сбои могут случиться:

- Container (Task)
- Application Manager (MRAppMaster)
- Node Manager
- Resource Manager

Сбой в работе контейнера

- Что может случиться:
 - Исключительная ситуация (exception)
 - Падение JVM / выключение сервера
 - Зависание процесса
- MRAppMaster перезапускает задачу
- Число перезапусков определяется через:
 - *mapreduce.map.maxattempts*
 - *mapreduce.reduce.maxattempts*

Сбой в Application Manager

- Что может случиться:
 - Работа АМ завершается с ошибкой
 - Сигналы от АМ не доходят до RM
- Resource Master перезапускает АМ
 - *yarn.resourcemanager.am.max-retries=2*
- MRAppMaster может восстановить состояние задачи до перезапуска:
 - *yarn.app.mapreduce.am.job.recovery.enable=true*

Сбой в Node Manager

- Что происходит:
 - NM перестает посылать сигналы RM
 - Контейнер на NM не посылает сигналы AM
- RM убирает NM из списка активных NM, если от него не приходит сигнал 10 минут
 - *yarn.resourcemanager.nm.liveness-monitor.expiryinterval-ms*
- MRAppMaster заносит NM в “черный список”, если произошло 3 падения контейнера
 - *mapreduce.job.maxtaskfailures.per.tracker*

Сбой в Resource Manager

- Никакие задачи и приложения не могут быть запущены
- Приводит к простоя кластера

Resource
Manager
(активный)



Resource
Manager
(резерв)



Использование YARN

- **Apache Giraph**
 - Итеративная система обработки графов
 - Facebook обрабатывает триллион ребер за ~4 мин
- **Spark**
 - Платформа для быстрой аналитики данных
- **Apache HAMA**
 - Фреймворк для массивных научных вычислений над матрицами, графами и сетями
- **Open MPI**