MapReduce和Yarn技术原理

www.huawei.com

HUAWEI

目标

- 学完本课程后，您将能够:
  - 熟悉**MapReduce**和**Yarn**是什么；
  - 掌握**MapReduce**使用的场景及其原理；
  - 掌握**MapReduce**和**Yarn**功能与架构；
  - 熟悉**Yarn**的新特性；

目录

MapReduce 基本定义

MapReduce是面向大数据并行处理的计算模型、框架和平台。

它包含以下三层含义：

1）MapReduce是一个基于集群的高性能并行计算平台（**Cluster Infrastructure**）。

2）MapReduce是一个并行计算与运行软件框架（**Software Framework**）。

3）MapReduce是一个并行程序设计模型与方法（**Programming Model & Methodology**）。

MapReduce 应用场景

- **MapReduce**基于**Google**发布的分布式计算框架**MapReduce**论文设计开发，用于大规模数据集（大于**1TB**）的并行运算，特点如下：
  - 易于编程：程序员仅需描述做什么，具体怎么做交由系统的执行框架处理。
  - 良好的扩展性：可通过添加节点以扩展集群能力。
  - 高容错性：通过计算迁移或数据迁移等策略提高集群的可用性与容错性。

Yarn 产生背景

MRv1几个方面的缺陷：

- 扩展性受限
- 单点故障
- 不支持**MR**之外的计算

多计算框架之间无法数据共享，资源利用率低

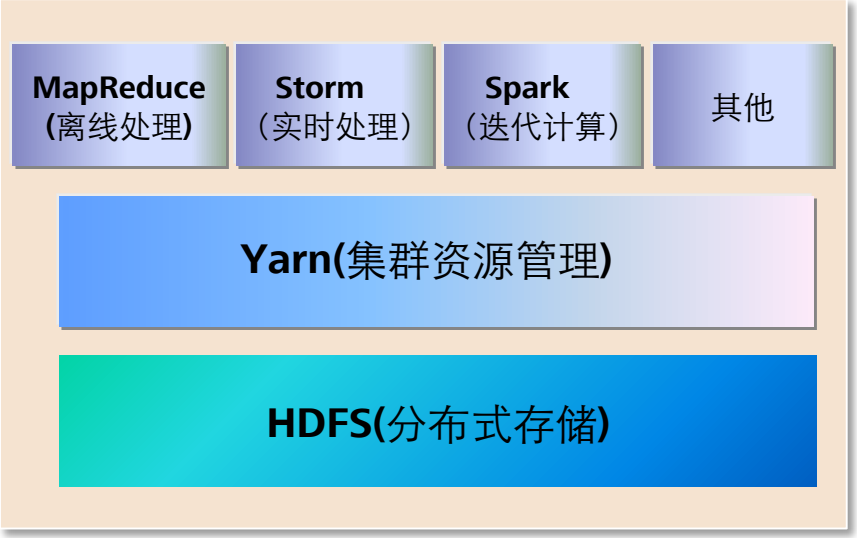
- **MR**：离线计算框架
- **Storm**：实时计算框架
- **Spark**：内存计算框架

Yarn 基本定义

Apache Hadoop YARN （Yet Another Resource Negotiator，另一种资源协调者）是一种新的 **Hadoop** 资源管理器，它是一个通用资源管理系统，可为上层应用提供统一的资源管理和调度，它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。

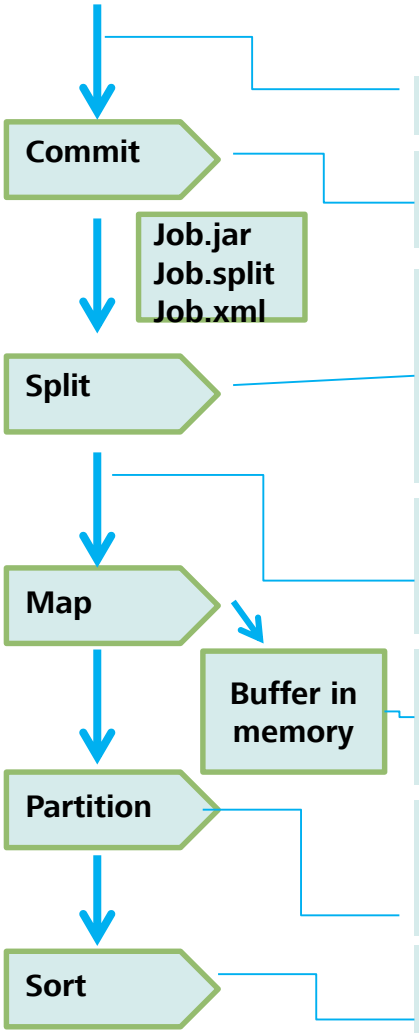
在产品中定位

- **Yarn**是**Hadoop2.0**中的资源管理系统，它是一个通用的资源管理模块，可为各类应用程序进行资源管理和调度
- **Yarn**是轻量级弹性计算平台，除了**MapReduce**框架，还可以支持其他框架，比如**Spark**、**Storm**等
- 多种框架统一管理，共享集群资源：
  - 资源利用率高
  - 运维成本低
  - 数据共享方便



目录

MapReduce的过程-MR过程详解



在启动**MapReduce**之前，确保待处理的文件放在**HDFS**上面

MapReduce应用将请求提交给**RM**，由**RM**创建对应的**Job**，一个应用对应一个**Job**（**jobID**形如，**job_201431281420_0001**）

Job在提交之前，首先要对待处理的文件进行分片（**Split**）。**MR**框架们默认将一个快（**Block**）作为一个分片（**Split**）。客户端应用也可以重新定义块与分片的映射关系。分片的必要性：**MR**框架将一个分片和一个**Map Tast**对应，即一个**Map Task**只能负责处理一个分片的数据。数据分片确定了将要为这个**Job**创建**Map Task**的个数

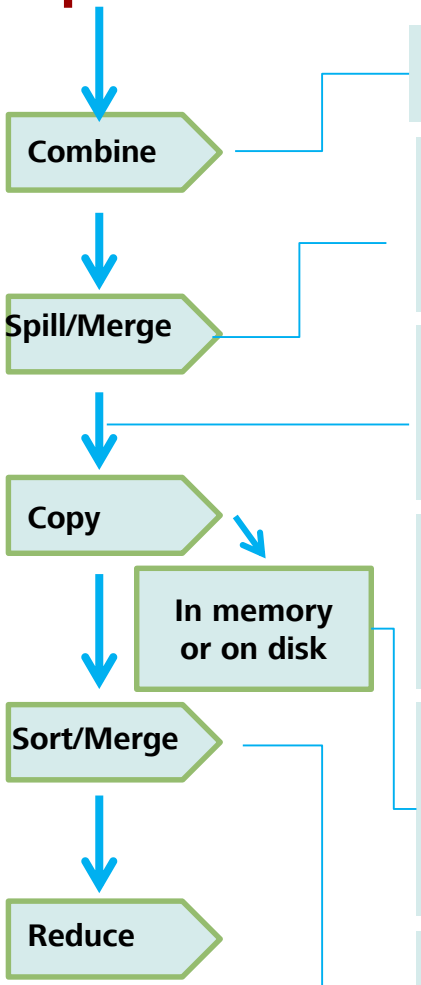
Job提交给**RM**，**RM**根据**NM**的负载在**NM**集群中挑选合适的节点调度**AM**，**AM**负责**Job**任务的初始化并向**RM**申请资源，由**RM**调度合适的**NM**启动**Container**,**Container**来执行**Task**。

Map的输出放入一个环形内存缓冲区中，当缓冲区的数据溢出时，需将缓冲区中的数据写入到本地磁盘中，写入本地磁盘之前通常需要做如下处理：

1.分区（**Partition**）---默认采用**Hash**算法进行分区，**MR**框架根据**Reduce Task**个数来确定分区个数。具备相同**Key**值的记录最终被送到相同的**Reduce Task**来处理。

2.排序（**Sort**）---将**Map**输出的记录排序，例如将**('Hi','1'),('Hello','1')**重新排序为**('Hello','1'), ('Hi','1')**。

MapReduce的过程-MR过程详解



3.组合(Combine)---这个动作MR框架默认是可选的。例如将('Hi','1'), ('Hi','1'),('Hello','1'), (Hello','1')进行合并操作为('Hi','2'),('Hello','2')。

4.合并(Spill)---Map Task在处理后能产生很多的溢出文件(spill file)，这时需将多个溢出文件进行合并处理为一个经过了分区和排序的Spill File（MOF:MapOutFile）。为减少写入磁盘的数据量，MR支持对MOF进行压缩后再写入

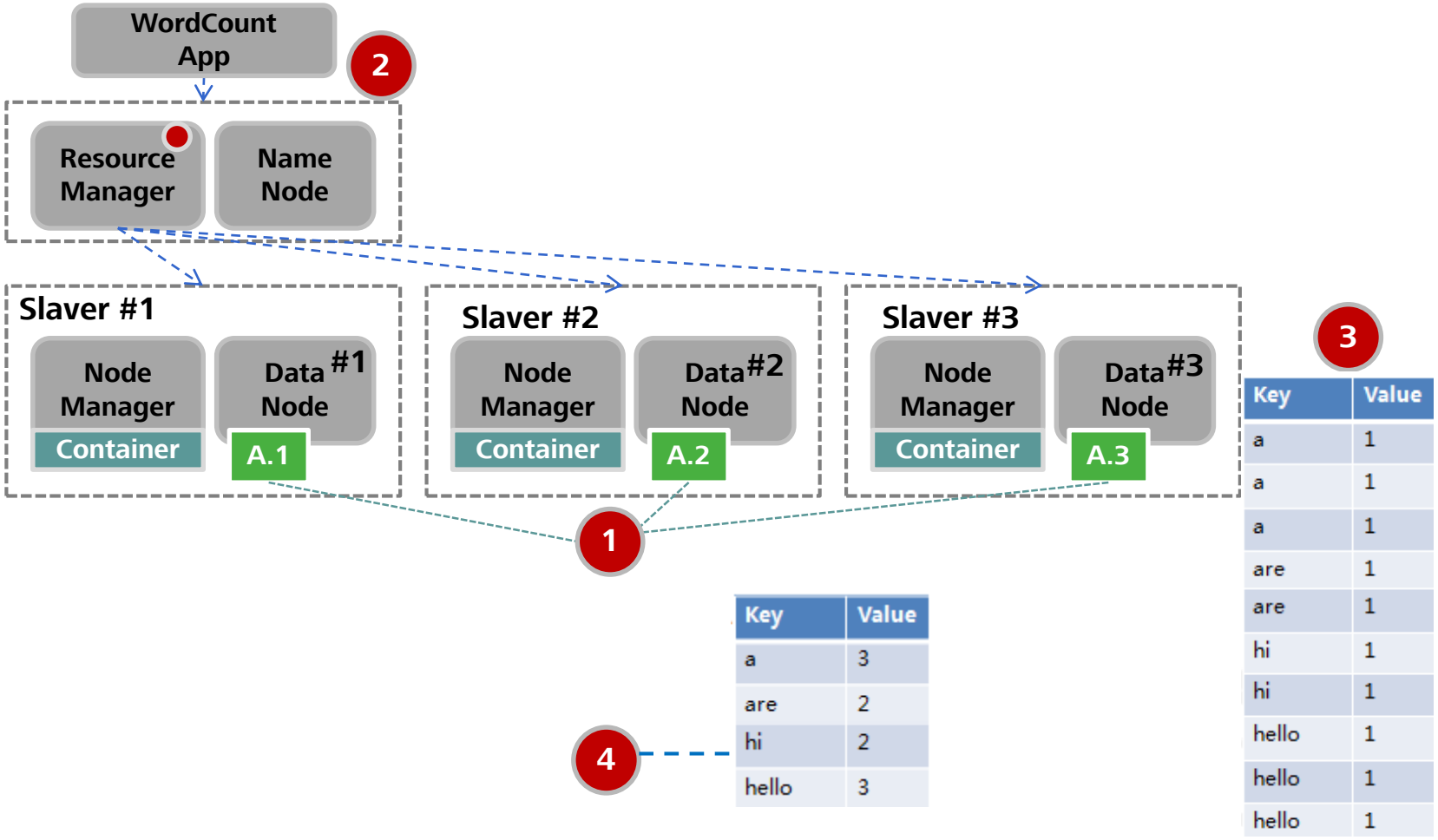
通常在Map Task任务完成MOF输出进度到3%时启动，从各个Map Task获取MOF文件。前面提到Reduce Task个数由客户端决定，Reduce Task个数决定MOF文件分区数。因此Map Task输出的MOF文件都能找到相对应的Redcue Task来处理。

Reduce阶段的三个过程：
1.Copy---Reduce Task从各个Map Task拷贝MOF文件。
2.Sort---通常又叫Merge，将多个MOF文件进行合并再排序。
3.Reduce---用户自定义的Reduce逻辑。

前面提到的MOF文件是经过排序处理的。当Reduce Task接收的数据量不大则直接存放在内存缓冲区中，随着缓冲区文件的增多，MR后台线程将它们合并成一个更大的有序文件，这个动作是Reduce阶段的Merge操作，的过程中会产生许多中间文件，最后一次合并的结果直接输出到用户自定义的reduce函数。

Shuffle的定义：Map阶段和Reduce阶段之间传递中间数据的过程，包括从Reduce Task各个Map Task获取MOF文件的过程，以及对MOF的排序与合并处理。

典型程序WordCount举例



Key	Value
a	3
are	2
hi	2
hello	3

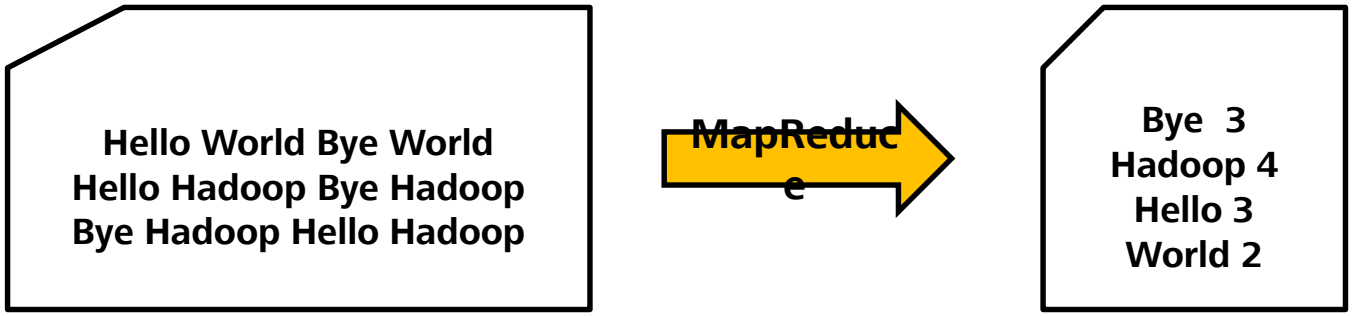
Key	Value
a	1
a	1
a	1
are	1
are	1
hi	1
hi	1
hello	1
hello	1
hello	1

HUAWEI

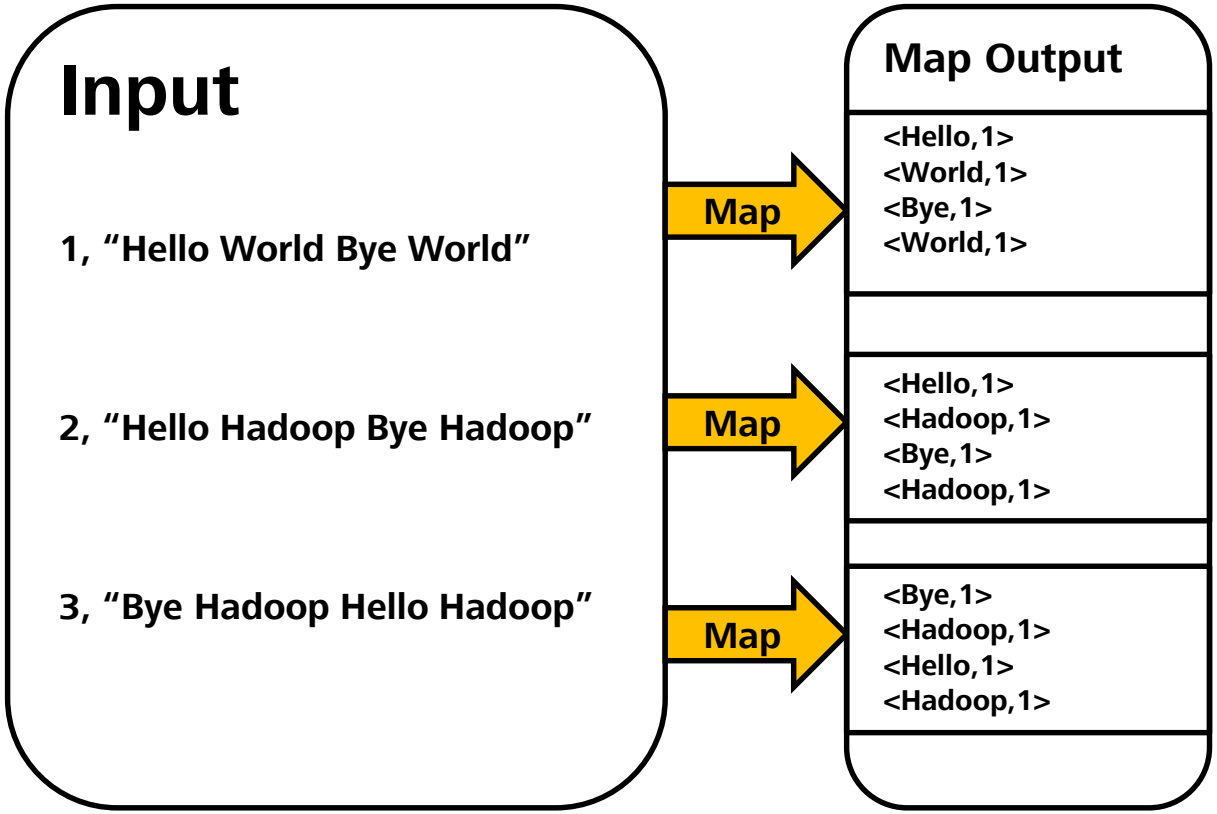
WordCount程序功能

Input 包含单词的文件

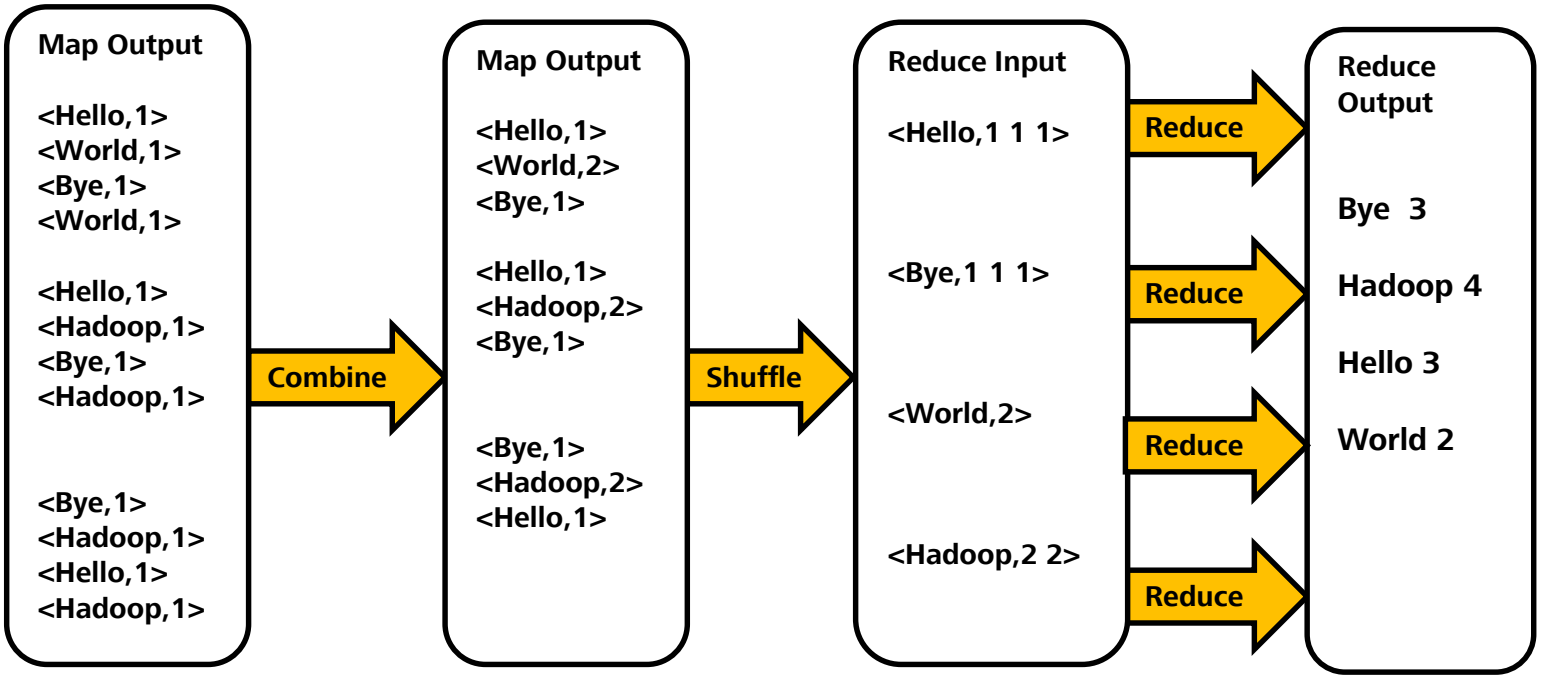
Output 每个单词出现的次数



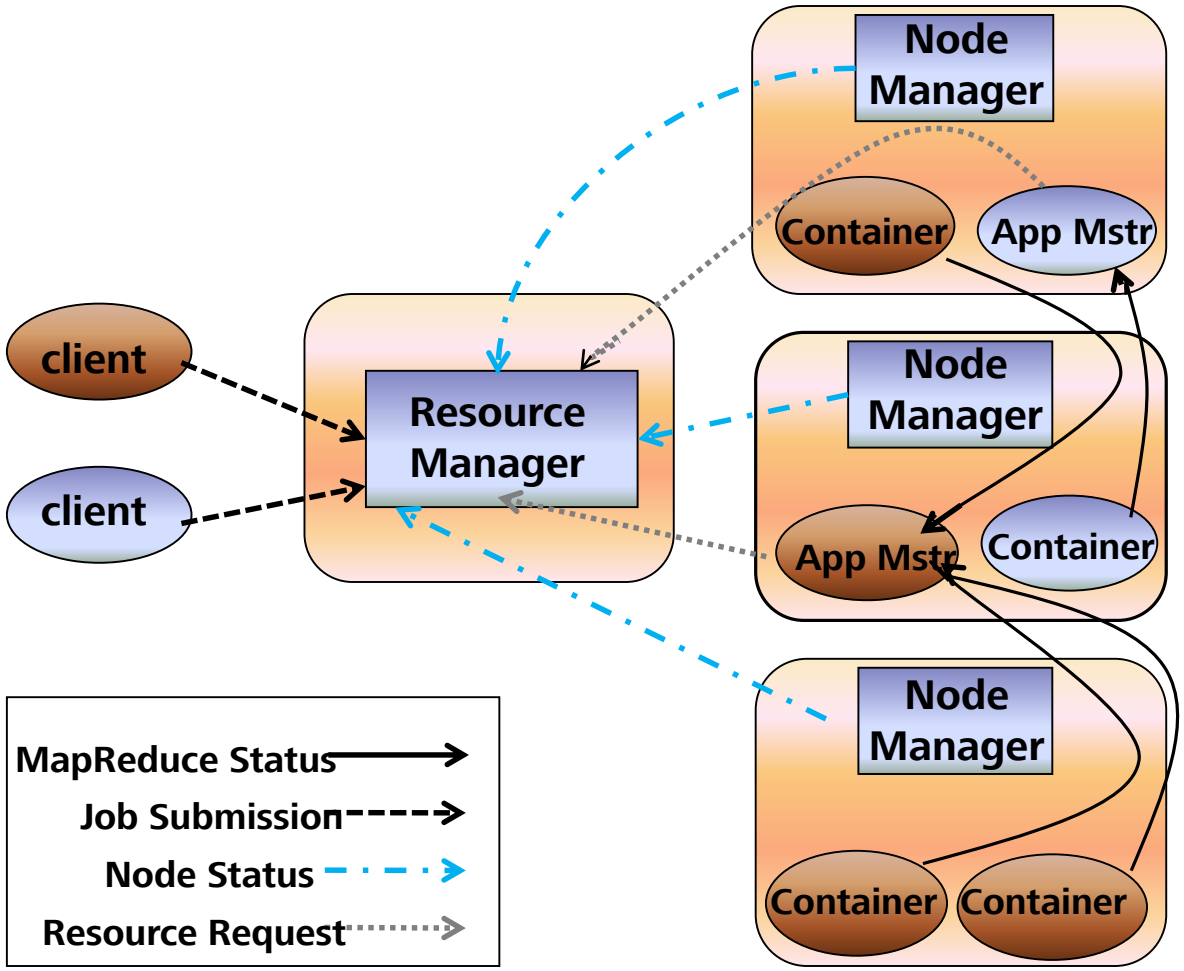
WordCount 的Map过程



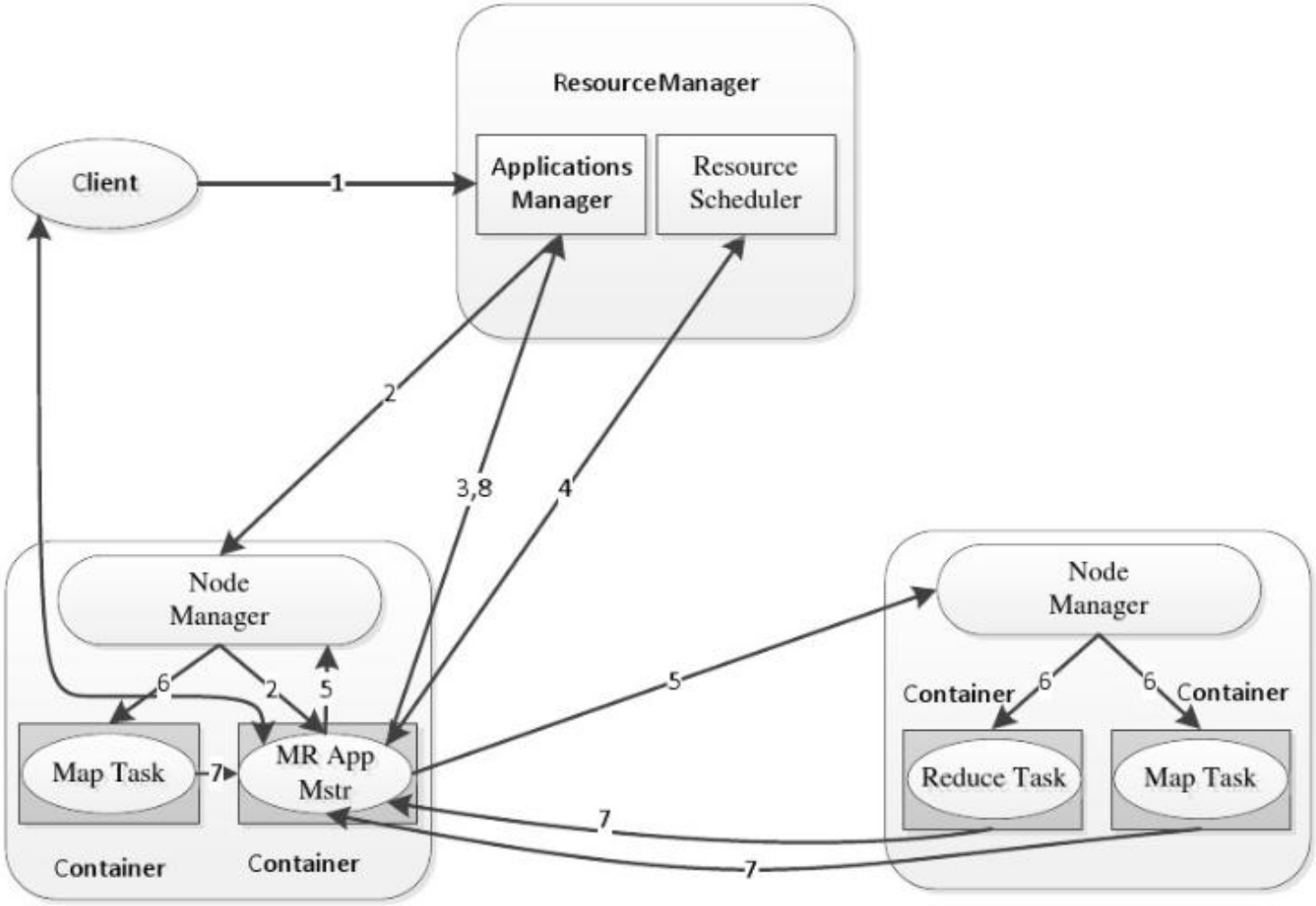
WordCount的Reduce过程



Yarn的组件架构

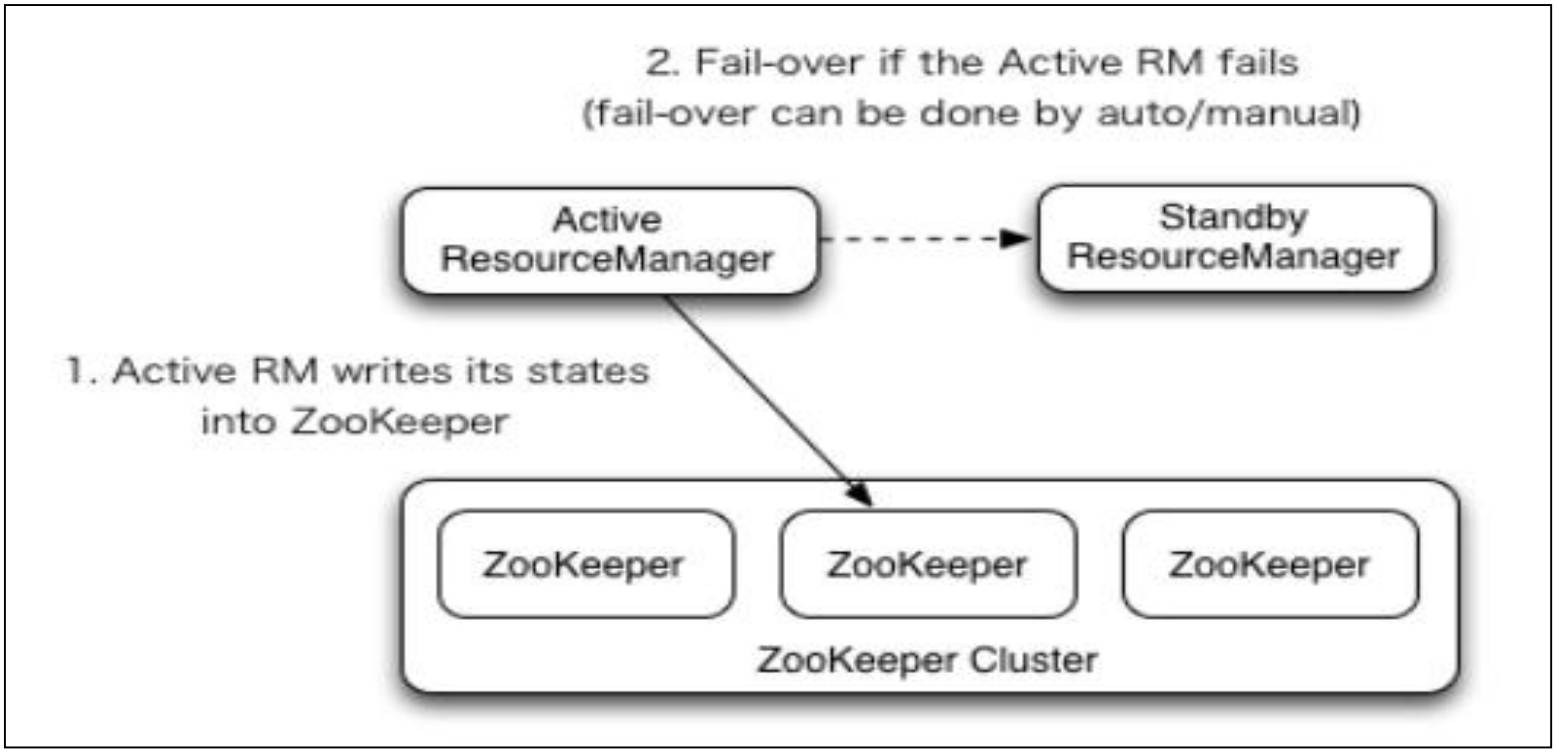


MapReduce On Yarn任务调度

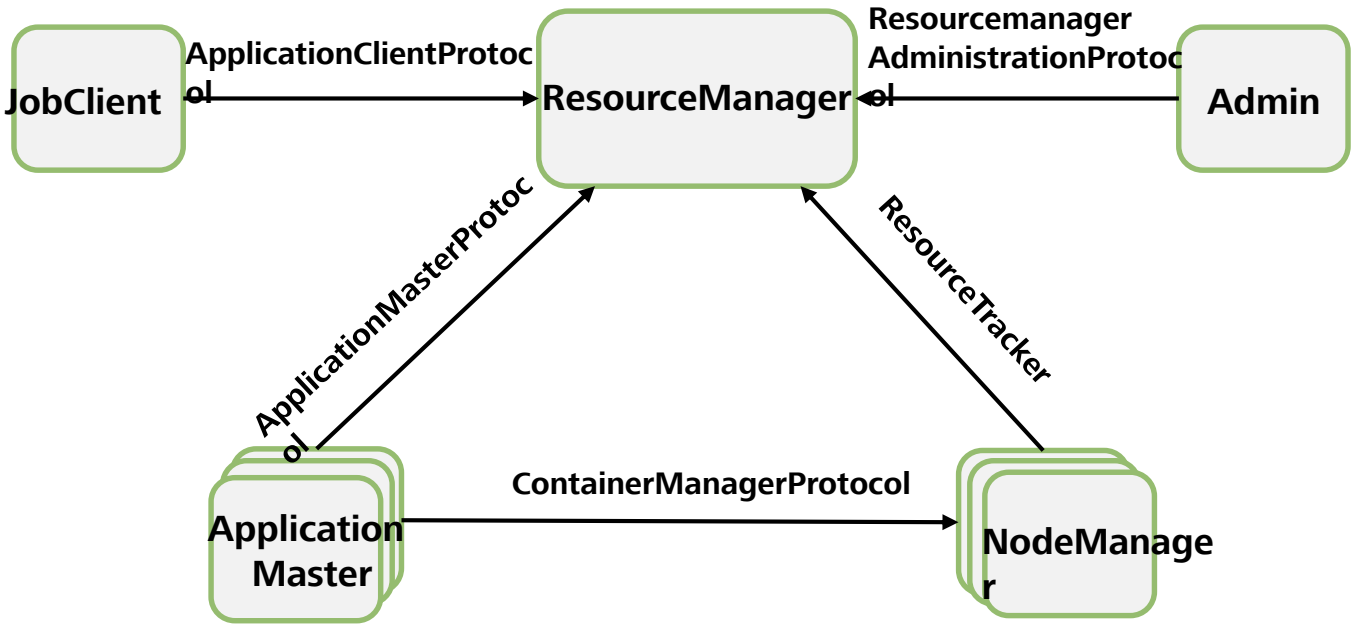


Yarn HA方案

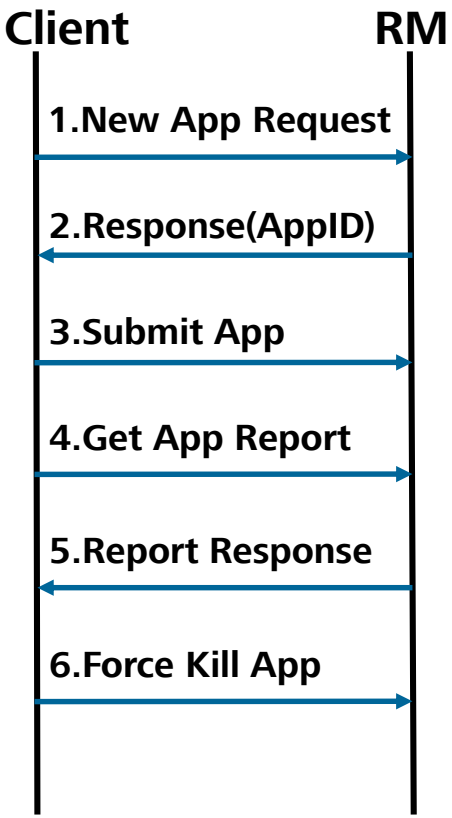
YARN中的**ResourceManager**负责整个集群的资源管理和任务调度，在以前的版本，**ResourceManager**在**YARN**集群中存在单点故障的问题。**YARN**高可用性方案通过引入冗余的**ResourceManager**节点的方式，解决了这个基础服务的可靠性和容错性问题。



Yarn通信协议

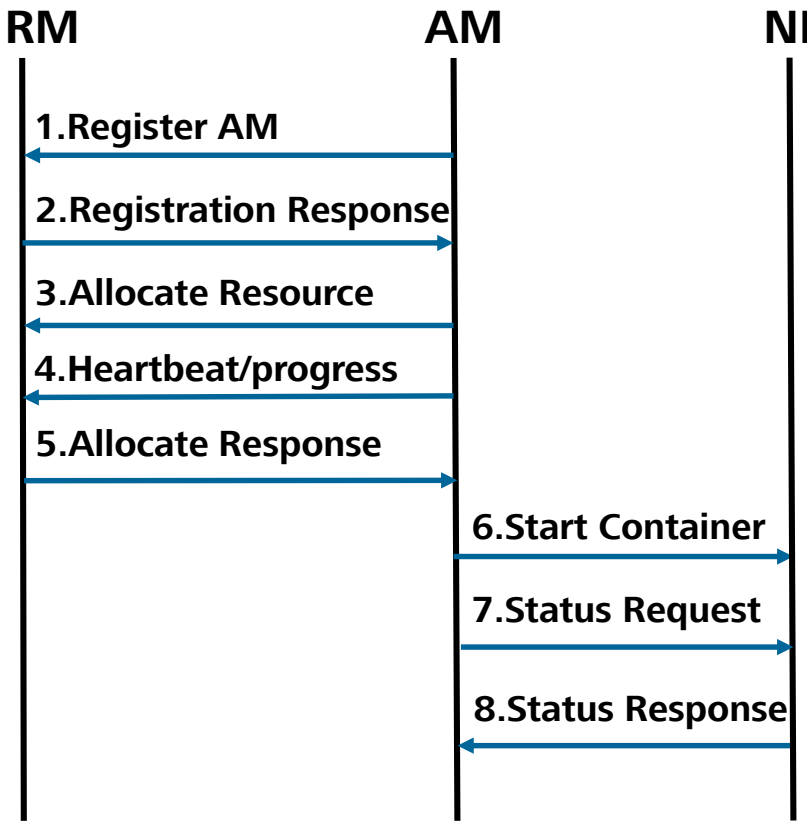


Yarn 客户端与RM的内部交互原理



1. 客户端通知**RM**提交一个应用。
2. **RM**生成一个唯一标识的应用**ID**，又叫**Job ID**，同时将当前**NM**集群的资源描述信息反馈给客户端。
3. 客户端根据**RM**的反馈信息，开始**Job**提交之前的初始化过程，包括调度队列、用户及优先级信息，和**RM**创建、启动**AM**所需的信息（例如应用**Jar**文件、**Job**资源信息、安全**Token**或其他资源描述）。
4. 客户端向**RM**查询、获取应用的执行进展报告。
5. **RM**将应用执行进展报告发送给**MR Client**。
6. 如有必要，客户端可直接通知**RM**终止**Application**的运行。

Yarn 关键组件内部交互原理



1. RM调度资源并在合适的NM节点上启动对应的AM。AM向RM注册，包含二者之间的握手信息、AM侦听端口，及后续进行AM管理和监控的URL。
2. RM接收AM注册信息，并反馈响应给AM，包含集群资源信息。
3. AM向RM发起资源分配请求，包含需要使用的Container个数，同时附带归属于本AM的Container信息。
4. AM向RM获取资源分配进度信息，并保持与RM之间的心跳。
5. RM根据资源调度策略，分配容器资源给AM。
6. AM根据RM反馈信息，指示对应NM完成Container的拉起。一个NM上可以启动多个Container。
7. 在Container运行过程中，AM向NM获取Container的运行状态报告。
8. NM将Container的运行状态信息反馈给AM。

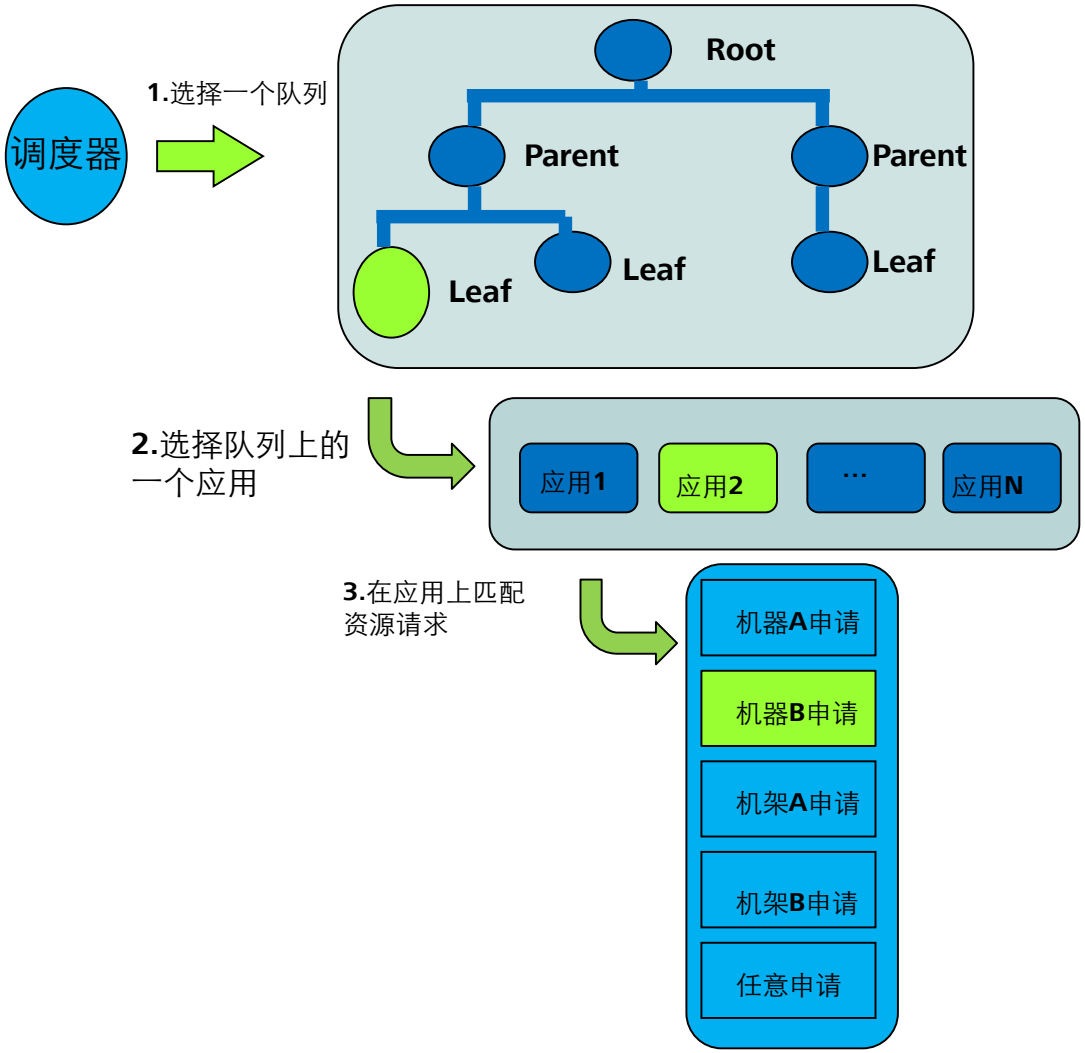
目录

资源管理

- 当前**YARN**支持内存和**CPU**两种资源类型的管理和分配。
- 每个**NodeManager**可分配的内存和**CPU**的数量可以通过配置选项设置（可在**yarn**服务配置页面配置）
  - **yarn.nodemanager.resource.memory-mb**
  - **yarn.nodemanager.vmem-pmem-ratio**
  - **yarn.nodemanager.resource.cpu-vcore**

资源分配模型

- 调度器维护一群队列的信息。用户可以向一个或者多个队列提交应用。
- 每次**NM**心跳的时候，调度器根据一定的规则选择一个队列，再在队列上选择一个应用，尝试在这个应用上分配资源。
- 调度器会优先匹配本地资源的申请请求，其次是同机架的，最后是任意机器的。



HUAWEI

容量调度器的介绍

- 容量调度器使得**Hadoop**应用能够共享的、多用户的、操作简便的运行在集群上，同时最大化集群的吞吐量和利用率。
- 容量调度器以队列为单位划分资源，每个队列都有资源使用的下限和上限。每个用户可以设定资源使用上限。管理员可以约束单个队列、用户或作业的资源使用。支持作业优先级，但不支持资源抢占。

容量调度器的特点

- **容量保证**：管理员可为每个队列设置资源最低保证和资源使用上限，所有提交到该队列的应用程序共享这些资源。
- **灵活性**：如果一个队列中的资源有剩余，可以暂时共享给那些需要资源的队列，当该队列有新的应用程序提交，则其他队列释放的资源会归还给该队列。
- **支持优先级**：队列支持任务优先级调度（默认是**FIFO**）。
- **多重租赁**：支持多用户共享集群和多应用程序同时运行。为防止单个应用程序、用户或者队列独占集群资源，管理员可为之增加多重约束。
- **动态更新配置文件**：管理员可根据需要动态修改配置参数，以实现在线集群管理。

容量调度器的任务选择

- 调度时，首先按以下策略选择一个合适队列：
  - 资源利用量最低的队列优先，比如同级的两个队列**Q1**和**Q2**，它们的容量均为**30**，而**Q1**已使用**10**，**Q2**已使用**12**，则会优先将资源分配给**Q1**；
  - 最小队列层级优先，例如：**QueueA**与**QueueB.childQueueB**，则**QueueA**优先；
  - 资源回收请求队列优先。
- 然后按以下策略选择该队列中一个任务：
  - 按照任务优先级和提交时间顺序选择，同时考虑用户资源量限制和内存限制。

队列资源限制

队列的创建是在多租户页面，当创建一个租户关联**Yarn**服务时，会创建同名的队列。比如先创建**QueueA,QueueB**两个租户，即对应**Yarn**两个队列。

队列资源限制

队列的资源容量（百分比）：

例如，有**default**、**QueueA**、**QueueB**三个队列，每个队列都有一个**[**队列名**].capacity**配置；

Default队列容量为整个集群资源的**20%**，

QueueA队列容量为整个集群资源的**10%**，

QueueB队列容量为整个集群资源的**10%**，后台有一个影子队列**root-default**使队列之和达到**100%**。

在集群的**Manager**页面点击“租户管理”》“动态资源计划”》“资源分布策略”可以看到下图页

资源分配

租户名（队列）	资源容量
QueueA(root.QueueA)	10%
QueueB(root.QueueB)	10%
TestParent(root.TestParent)	10%
testchild(root.TestParent.testchild)	10%
default(root.default)	20%
testyarn(root.testyarn)	5%

队列资源限制

共享空闲资源

- 由于存在资源共享，因此一个队列使用的资源可能超过其容量（例如**QueueA.capacity**），而最多使用资源量可通过该参数限制。
  - **yarn.scheduler.capacity.root.QueueA.maximum-capacity**（此参数也是在上页胶片展示的**FusionInsight**页面配置）
- 如果某个队列任务较少，可将剩余资源共享给其他队列，例如**QueueA**的**maximum-capacity**配置为**100**，假设当前只有**QueueA**在运行任务，理论上**QueueA**可以占用整个集群**100%**的资源。

用户限制和任务限制

用户限制和任务限制的参数可通过“租户管理”>“动态资源计划”>“队列配置”进行配置

用户限制

每个用户最低资源保障（百分比）

任何时刻，一个队列中每个用户可使用的资源量均有一定的限制，当一个队列中同时运行多个用户的任务时，每个用户的可使用资源量在一个最小值与最大值之间浮动，其中，最大值取决于正在运行的任务数目，而最小值则由**minimum-user-limit-percent**决定。

例如，设置队列**A**的这个值为**25**，即**yarn.scheduler.capacity.root.QueueA.minimum-user-limit-percent=25**，那么随着提任务的用户增加，队列资源的调整如下：

第**1**个用户提交任务到**QueueA**	会获得**QueueA**的**100%**资源
第**2**个用户提交任务到**QueueA**	每个用户会最多获得**50%**的资源
第**3**个用户提交任务到**QueueA**	每个用户会最多获得**33.33%**的资源
第**4**个用户提交任务到**QueueA**	每个用户会最多获得**25%**的资源
第**5**个用户提交任务到**QueueA**	为了保障每个用户最低能获得**25%**的资源，第**5**个用户将无法再获取到**QueueA**的资源，必须等待资源的释放。

用户限制

- **每个用户最多可使用的资源量（所在队列容量的倍数）**
- **queue**容量的倍数，用来设置一个**user**可以获取更多的资源。

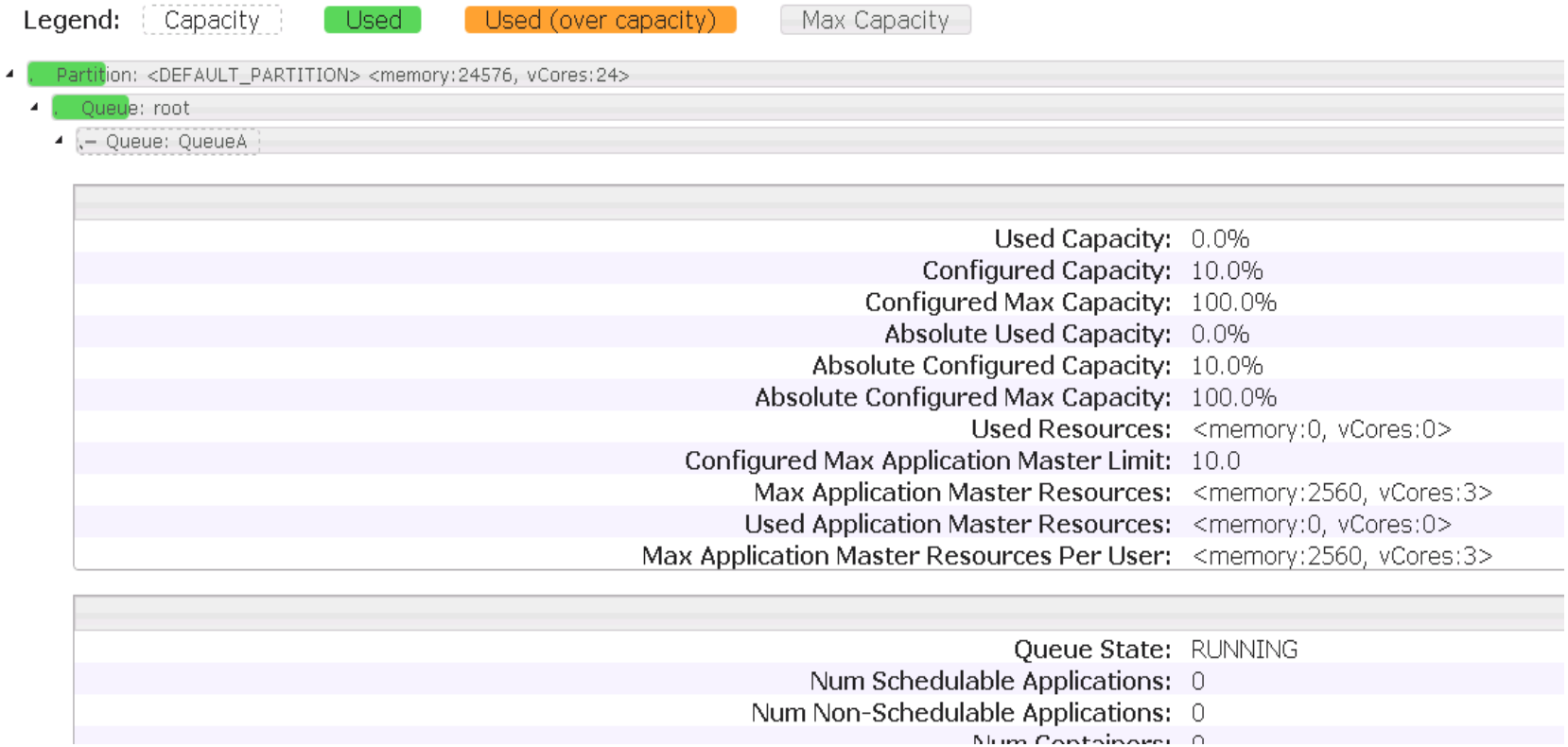
  yarn.scheduler.capacity.root.QueueD.user-limit-factor = 1
- 默认值为**1**，表示一个**user**获取的资源容量不能超过**queue**配置的**capacity**，无论集群有多少空闲资源，最多不超过**maximum-capacity**。

任务限制

- **最大活跃任务数**
  - 整个集群中允许的最大活跃任务数，包括运行或挂起状态的所有任务，当提交的任务申请数据达到限制以后，新提交的任务将会被拒绝。默认**10000**
  - **yarn.scheduler.capacity.maximum-applications=10000**
- **每个队列最大任务数**
  - 对于每个队列，可以提交的最大任务数，以**QueueA**为例，可以在队列配置页面配置，默认是**1000**，即此队列允许最多**1000**个活跃任务。
- **每个用户可以提交的最大任务数**
  - 这个数值依赖每个队列最大任务数，假设根据上面的结果， **QueueA**最多可以提交**1000**个任务，那么对于每个用户而言，可以向**QueueA**提交的最大任务数为
  - **1000* yarn.scheduler.capacity.root.QueueA.minimum-user-limit-percent* yarn.scheduler.capacity.root.QueueA.user-limit-factor**

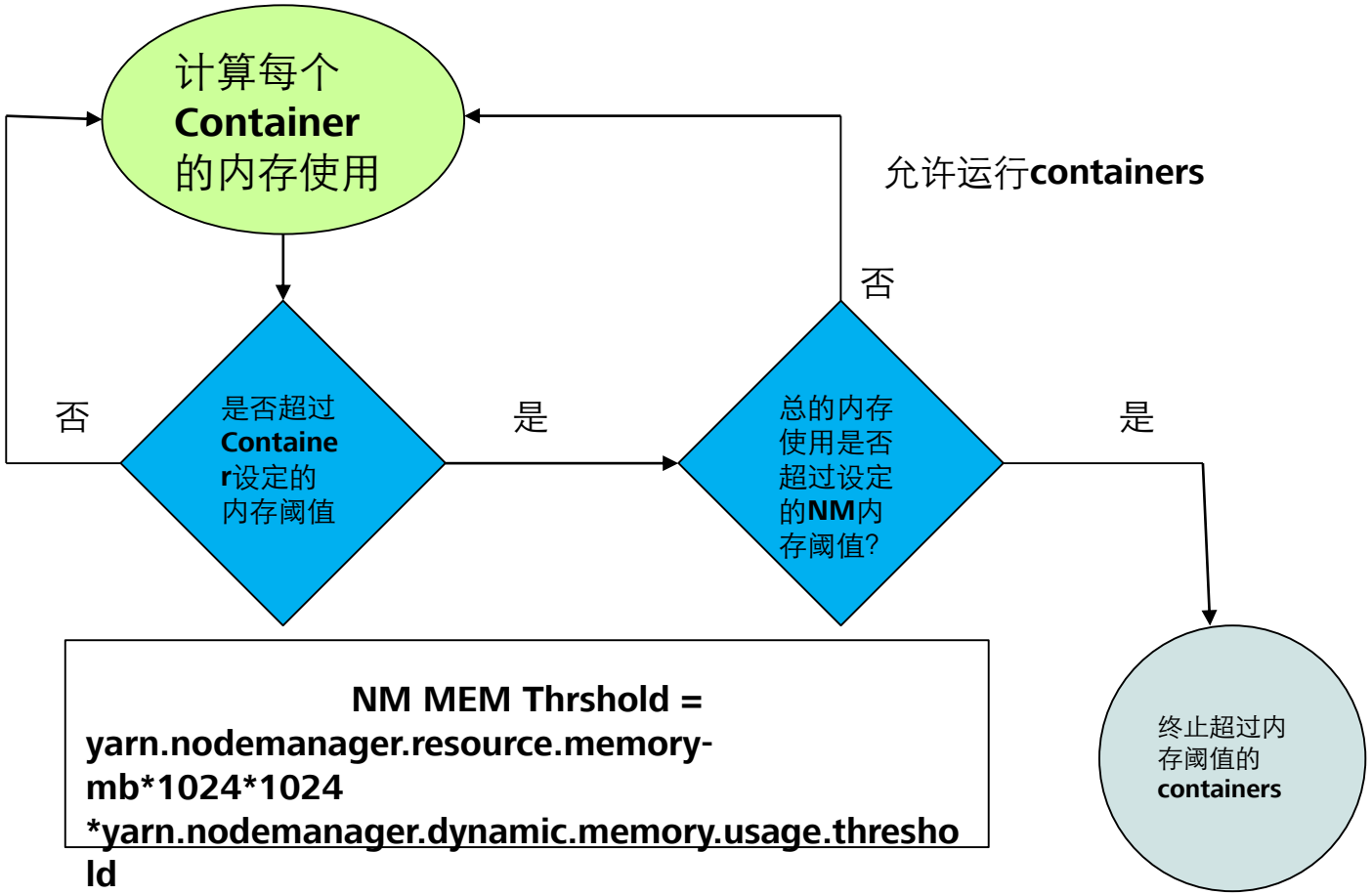
队列信息

队列的信息还可以通过**Yarn webUI**进行查看，进入方法是“服务管理”>“**Yarn**”>“**ResouceManager**（主）”>“**Scheduler**”

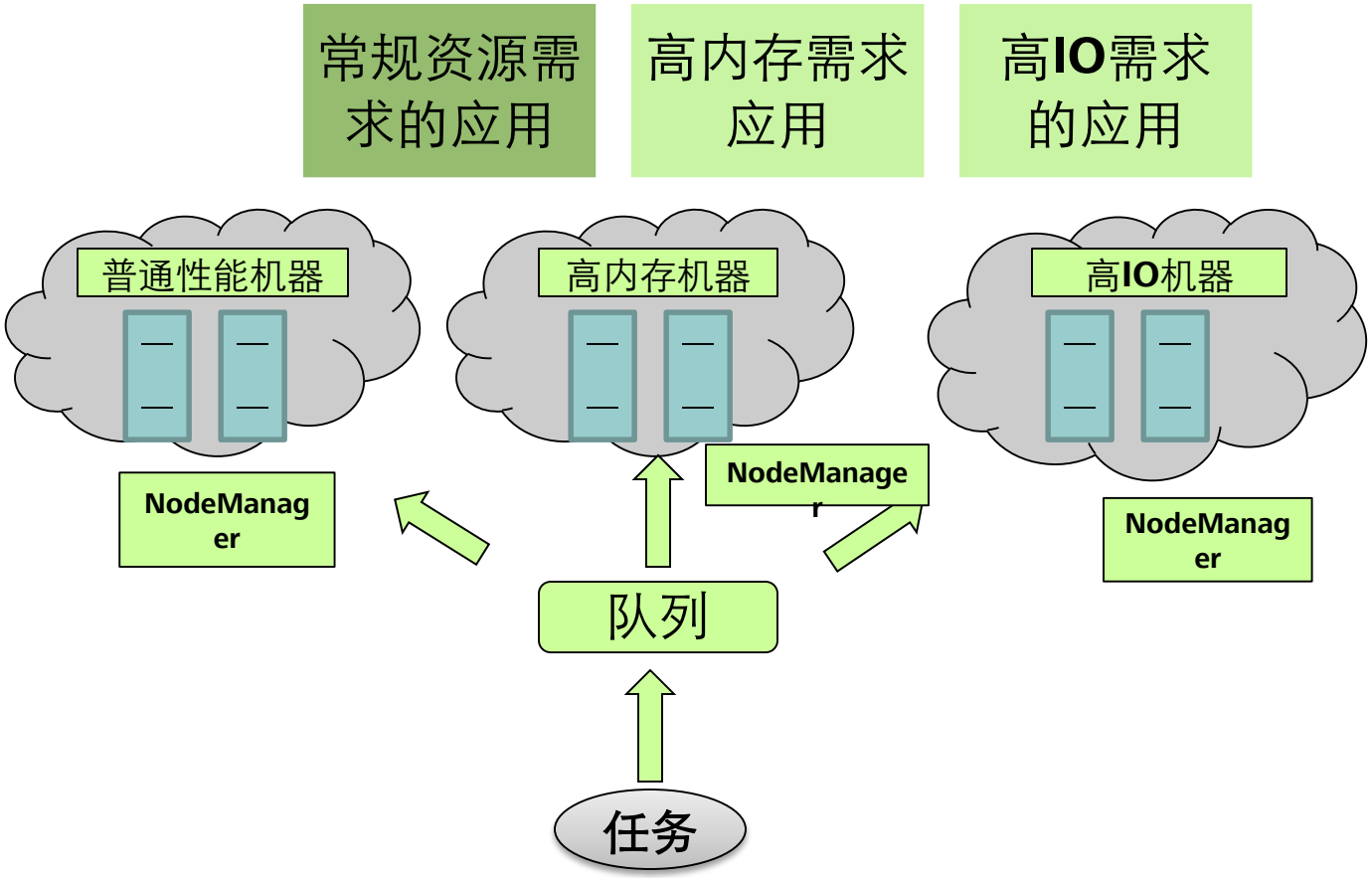


目录

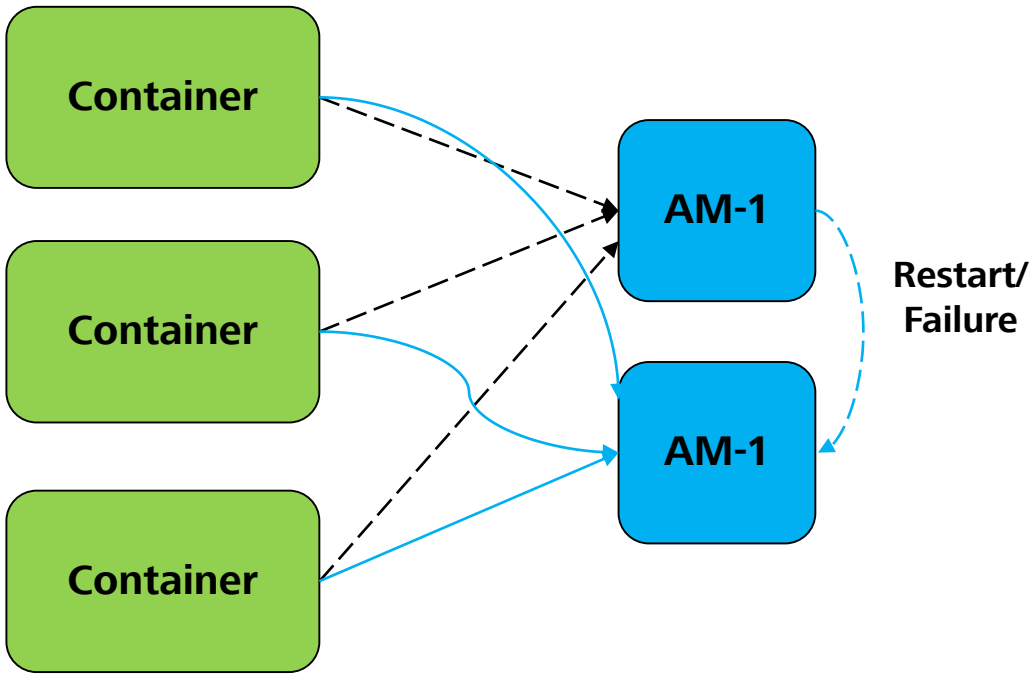
增强特性—Yarn动态内存管理



增强特性—Yarn基于标签调度



增强特性—Yarn AM作业保留



思考题

1.请简述**MapReduce**的工作原理。

2.相比于**MapReduce**，**Yarn**有哪些有点?

3.请简述**Yarn**的工作原理。

思考题

1. 下面哪些是**MapReduce**的特点？（　　）

   A. 易于编程

   B. 良好的扩展性

   C. 实时计算

   D. 高容错性

2. **Yarn**中资源抽象用什么表示？（　　）

   A. 内存

   B. **CPU**

   C. **Container**

   D. 磁盘空间

思考题

3. 下面哪个是**MapReduce**适合做的？（　　）

   A. 迭代计算

   B. 离线计算

   C. 实时交互计算

   D. 流式计算

4. 容量调度器的特点？（　　）

   A. 容量保证

   B. 灵活性

   C. 多重租赁

   D. 动态更新配置文件

本章总结

- 讲述了**MR**和**YARN**的应用场景
- 讲述了**MR**和**YARN**的基本架构
- 讲述了**YARN**资源管理与任务调度
- 讲述了**YARN**的增强特性

学习推荐

- 华为**Learning**网站
  - **http://support.huawei.com/learning/Index!toTrainIndex**
- 华为**Support**案例库
  - **http://support.huawei.com/enterprise/servicecenter?lang=zh**

Thank you

www.huawei.com