

hadoop2.0介绍

麦树荣
技术部数据组



hadoop2.0的产生背景

hadoop2.0的基本架构

hadoop2.0的资源管理和分配

hadoop1.0升级到2.0

hadoop2.0应用程序要注意的问题和优化

hadoop1.0与2.0的兼容性

hadoop的版本

- * **hadoop1.0**

hadoop-0.20.x, hadoop-CDH3, hadoop-1.x

- * **hadoop2.0**

hadoop-0.23.x, hadoop-CDH4, hadoop-2.x

- * **现在用的hadoop版本**

hadoop-2.2.0 stable release

hadoop2.0 产生背景

hadoop2.0 产生背景

- * 扩展性受限
- * 单点故障
- * 不能支持其他计算框架

hadoop2.0 的基本架构

hadoop2.0 的组件

◆ HDFS

HA, Federation

◆ YARN

独立的资源分配和管理框架

◆ MapReduce

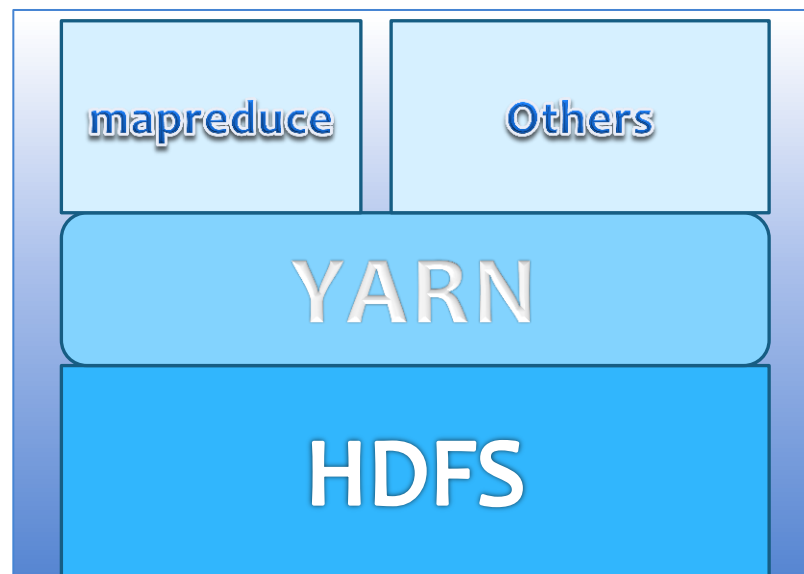
运行在YARN上面的离线计算框架

hadoop1.0和hadoop2.0的比较

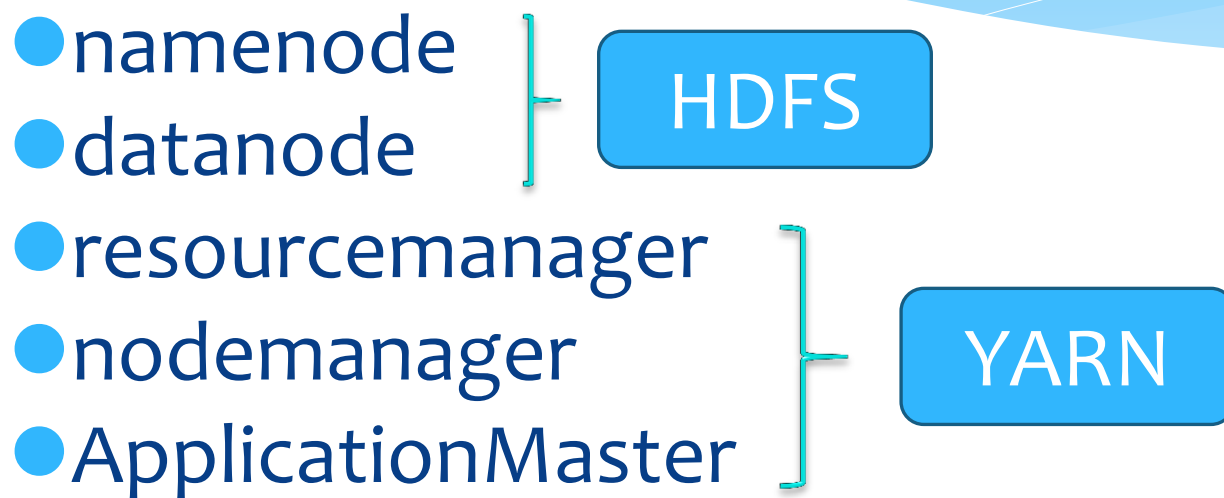
hadoop1.0



hadoop2.0



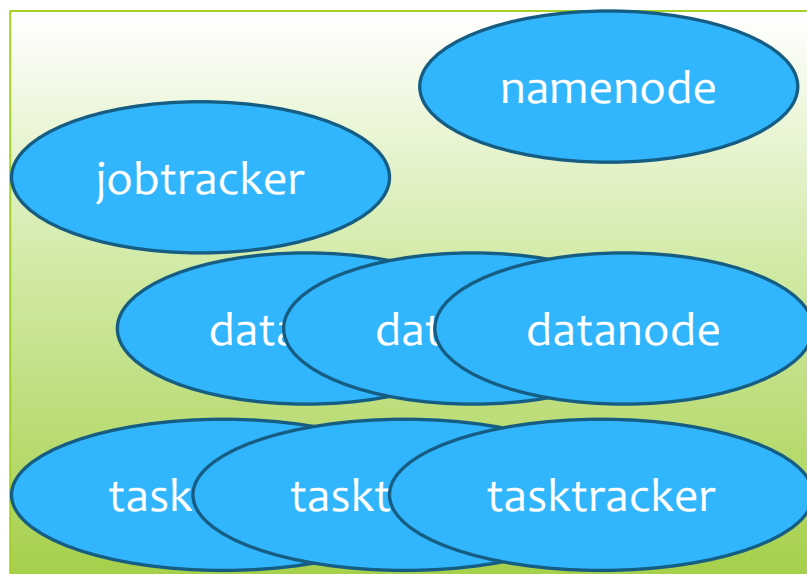
hadoop2.0 的服务进程



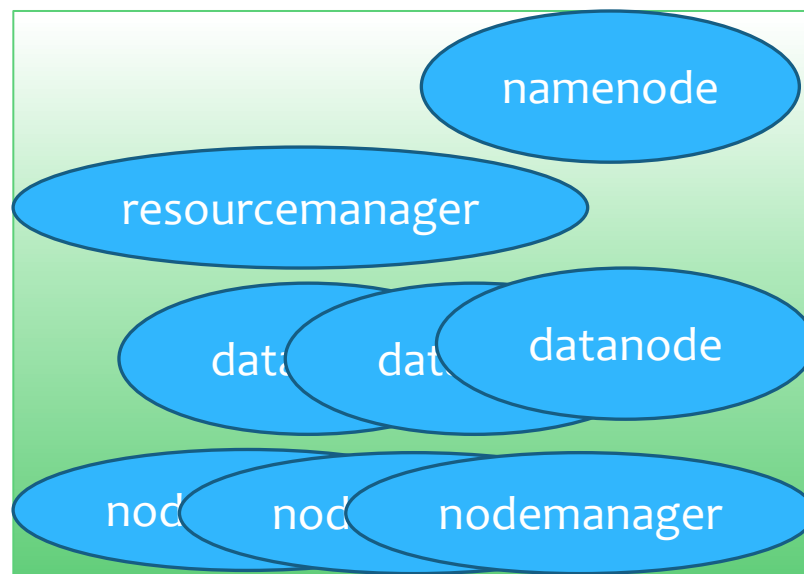
ApplicationMaster运行某个应用时才会运行，
如一个MR job会有一个MRAppMaster，
用于管理和监控该MR job

hadoop1.0和hadoop2.0的服务进程

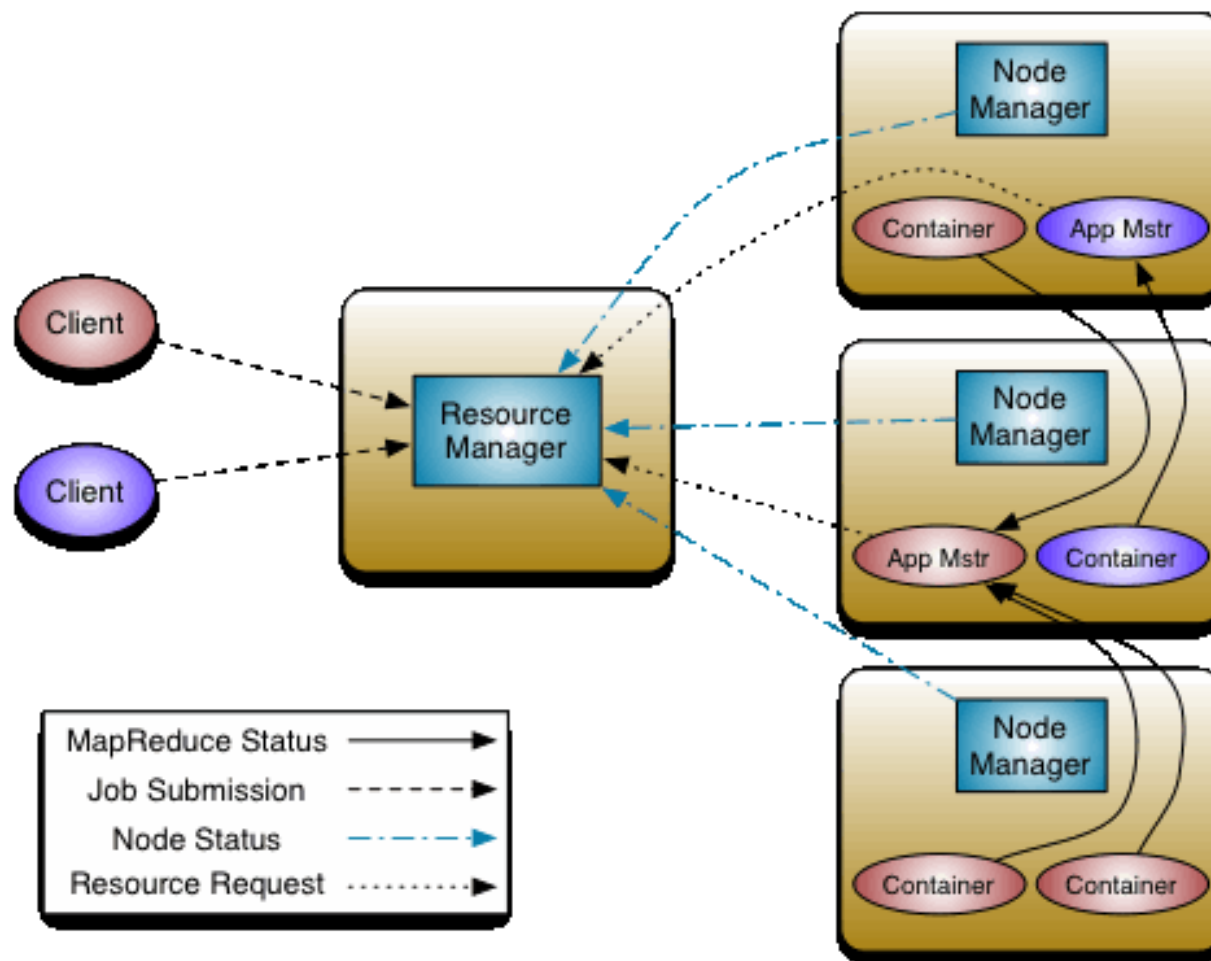
hadoop1.0



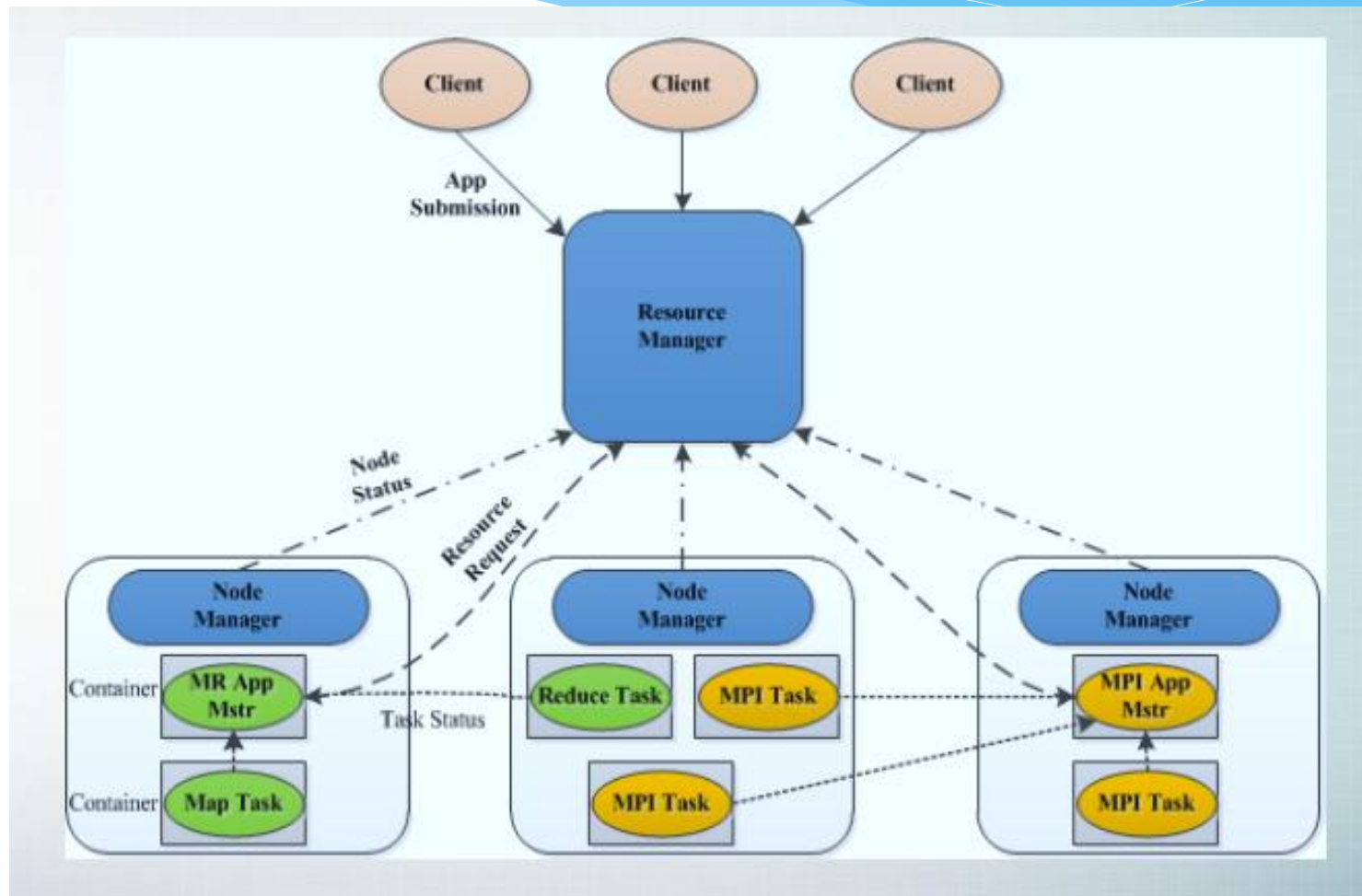
hadoop2.0



YARN的基本架构



YARN上同时运行2个计算框架



hadoop2.0的资源管理和分配

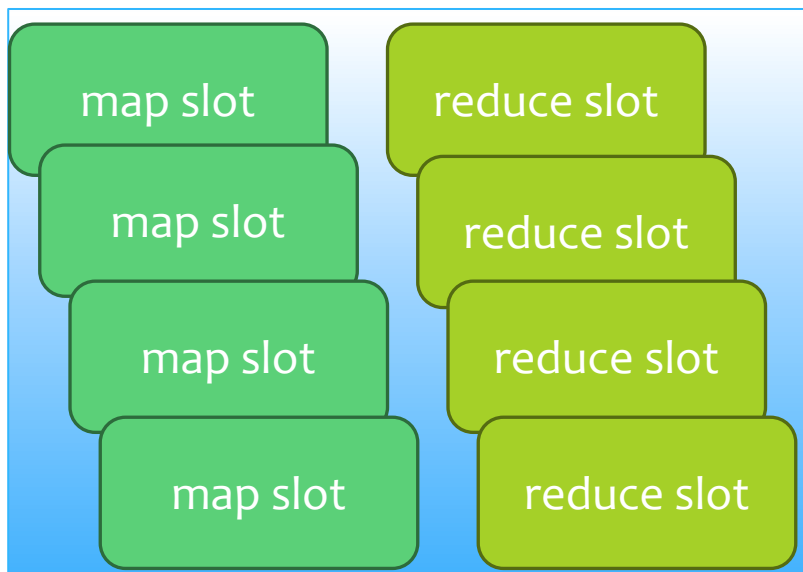
更合理更通用的资源模型

- ◆ Container/容器是YARN中资源的抽象和封装（CPU和内存两类资源）
- ◆ 每个计算节点由多个固定大小的内存块（512MB或者1GB）的容器组成
- ◆ 整个集群的资源形成一个资源池，供各个应用申请
- ◆ ApplicationMaster可以申请该内存整数倍大小的容器
- ◆ 资源池的资源都是对等的（不区分是map的还是reduce的）

hadoop1.0和hadoop2.0资源模型比较

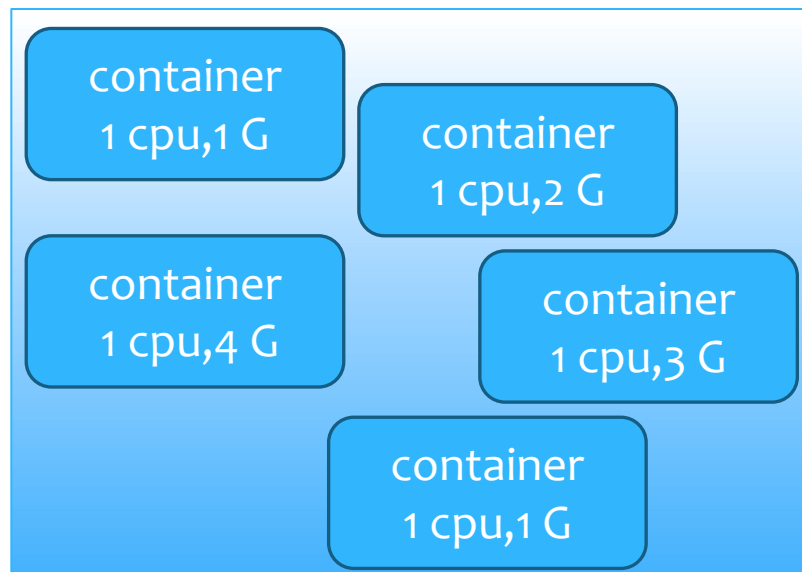
hadoop1.0

- 区分map slot 和 reduce slot
- map和reduce之间不能共享
- 以slot作为资源单元来分配



hadoop2.0

- ◆ 都是对等的container
- ◆ map和reduce之间可以共享
- ◆ 以真实的资源来分配



资源的分配和调度

- ◆ 调度器/scheduler（公平调度器/fairscheduler）
- ◆ 每个队列分配一定的资源量
- ◆ 用户向某些队列提交作业的权限
- ◆ 每个作业从队列中获得相应的资源
- ◆ 运行完成后把资源归还

资源分配的相关参数

minResources

每个队列的最小资源数，保证可使用的最少资源量

maxResources

每个队列允许使用的最大资源数，防止资源滥用

maxRunningApps

每个队列允许同时运行的作业数

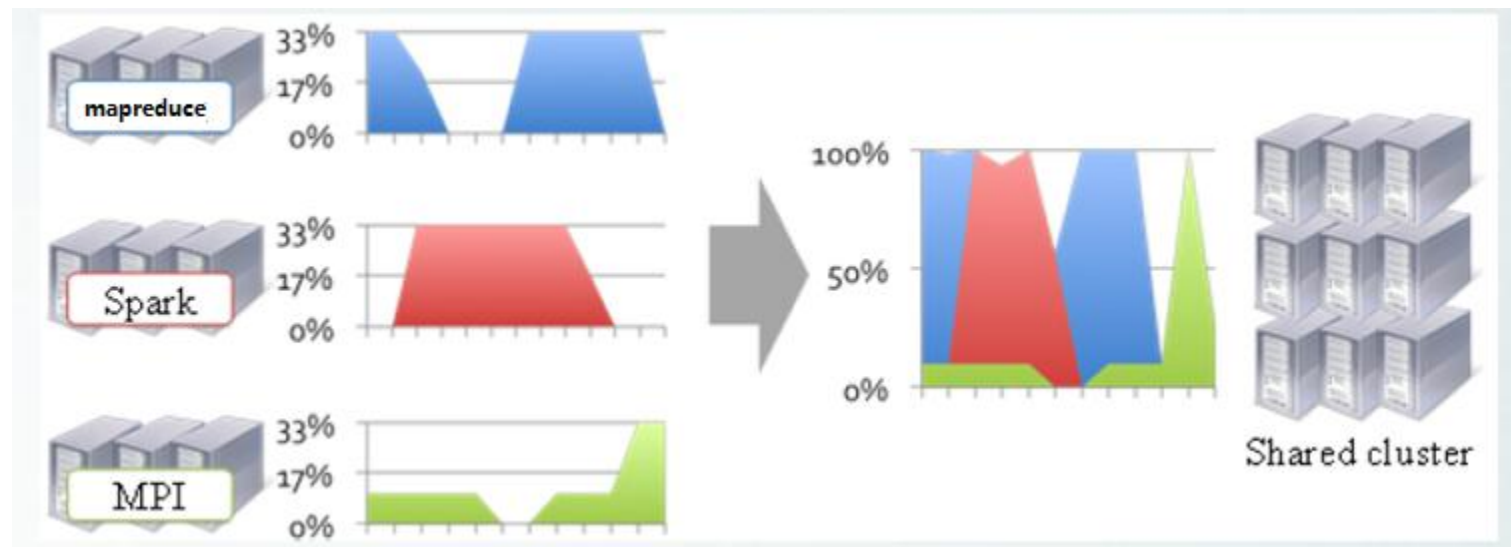
每个用户允许同时运行的作业数

为什么要升级到hadoop2.0



hadoop2.0 的优势

- * 更好的扩展性
- * HDFS的HA
- * 更合理的资源模型
- * 可运行多个计算框架
- * 更好的资源利用



运行在YARN上的计算框架

- MapReduce-on-YARN

离线计算框架

- Storm-on-YARN

流式计算框架

- Spark-on-YARN

内存计算框架

- Tez-on-YARN

DAG作业的计算框架

<http://wiki.apache.org/hadoop/PoweredByYarn/>

hadoop2.0 的不足

- resourcemanager 单点
- 现在在YARN上运行多个计算框架的实践比较少，可参考的资料不多

hadoop2.0 应用程序 要注意的问题和优化

怎样更好的用hadoop2.0



从资源使用的角度来优化

申请内存和CPU的相关参数

ApplicationMaster的内存参数

- ✓ `yarn.app.mapreduce.am.resource.mb=1536`
- ✓ `yarn.app.mapreduce.am.command-opts=-Xmx1024m`

CPU个数设置，一般不设置

- ✓ `mapreduce.map.cpu.vcores=1`
- ✓ `mapreduce.reduce.cpu.vcores=1`

申请内存的相关参数

向yarn申请内存的参数

- ✓ `mapreduce.map.memory.mb`
- ✓ `mapreduce.reduce.memory.mb`

JVM堆内存参数

- ✓ `mapred.child.map.java.opts`
- ✓ `mapred.child.reduce.java.opts`
- X `mapreduce.map.java.opts`
- X `mapreduce.reduce.java.opts`

参数的建议值

可以把建议值写到客户端的配置文件mapred-site.xml中，作为默认参数。具体的job如果有需要，可以在job中设置。

- ✓ `mapreduce.map.memory.mb=1024`
- ✓ `mapreduce.reduce.memory.mb=2048`
- ✓ `mapred.child.map.java.opts=-Xmx900M`
- ✓ `mapred.child.reduce.java.opts=-Xmx1900M`

更节省的设置

- `mapreduce.map.memory.mb=512`
- `mapreduce.reduce.memory.mb=1024`
- `mapred.child.map.java.opts=-Xmx460M`
- `mapred.child.reduce.java.opts=-Xmx900M`



mapper和reducer个数的设置

在mapreduce job中，map个数一般不需要设置，reduce个数一般需要指定

- * `mapreduce.job.reduces`

- * `mapreduce.job.maps`

在hive中，有需要的话可以人工指定reduce的个数，以避免reducer内存不够，同时增加并行度

- * `set mapreduce.job.reduces = 32`

尽量申请适合的资源

- * cpu的个数使用默认值1，即不用设置
- * 尽量申请适合的内存大小
- * 大内存的任务启动慢
- * 大内存消耗队列的资源较多
- * 大内存使队列可同时跑的作业/任务数量变少
- * 作业等待启动的时间变长

客户机的内存设置

- * 设置环境变量

```
export HADOOP_CLIENT_OPTS="-Xmx2048m "
```

- * 修改配置文件

```
$HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

小作业的优化—uber mode

- 小作业
- 由一个container来运行
- 作业运行时间减少
- 优化队列资源的使用
- 注意：
 - `yarn.app.mapreduce.am.resource.mb >= mapreduce.map.memory.mb`
 - `yarn.app.mapreduce.am.resource.mb >= mapreduce.reduce.memory.mb`

何为小作业—uber mode 参数

- * `mapreduce.job.ubertask.enable=true`

- * `mapreduce.job.ubertask.maxmaps=9`

Map的个数，默认值是9

- * `mapreduce.job.ubertask.maxreduces=1`

Reduce的个数，默认值是1，

注意：目前也只能是1，不能设置其他值

- * `mapreduce.job.ubertask.maxbytes=80000000`

默认值是 `dfs.block.size`，64M

jvm 重用

- 适用场合

map/reduce 的运行时间比较短

map/reduce 的数量比较多

- 参数设置

mapreduce.job.jvm.numtasks

查看运行的作业

← → ↺ l-hdpm2.data.cn6.qunar.com:8088/cluster/apps/RUNNING ★



RUNNING Applications

Cluster

[About](#)
[Nodes](#)
[Applications](#)
[NEW](#)
[NEW SAVING](#)
[SUBMITTED](#)
[ACCEPTED](#)
[RUNNING](#)
[REMOVING](#)
[FINISHING](#)
[FINISHED](#)
[FAILED](#)
[KILLED](#)
[Scheduler](#)

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	U
148589	9	11	148569	228	670 GB	2.79 TB	136 GB	57	0	6	0

User Metrics for hadoop

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pe
39435	9	11	148569	0	0	3	0 B	0 B

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progr
application_1387813121465_150391	qhstats	create table tmp_hotel_search_20120221 a...x(Stage-15)	MAPREDUCE	root.hoteldev	2013年12月29日 下午7:35:33	N/A	RUNNING	UNDEFINED	
application_1387813121465_150390	qhstats	hotel_price_monitor_wrapper.jar	MAPREDUCE	root.hoteldev	2013年12月29日 下午7:35:27	N/A	RUNNING	UNDEFINED	
application_1387813121465_150389	qhstats	create table tmp_hotel_search_20130218 a...x(Stage-4)	MAPREDUCE	root.hoteldev	2013年12月29日 下午7:35:25	N/A	RUNNING	UNDEFINED	
application_1387813121465_150385	qhstats	create table tmp_hotel_detail_2013041...g.id(Stage-1)	MAPREDUCE	root.hoteldev	2013年12月29日 下午7:35:23	N/A	RUNNING	UNDEFINED	
application_1387813121465_150370	qhstats	create table tmp_hotel_search_20130218 a...x(Stage-16)	MAPREDUCE	root.hoteldev	2013年12月29日 下午7:33:55	N/A	RUNNING	UNDEFINED	

查看队列状态

Cluster

About

Nodes

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

REMOVING

FINISHING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	A
43537	11	20	43506	283	465 GB	2.45 TB	17 GB	51

User Metrics for hadoop

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Cor
10102	11	20	43506	243	162	0

Application Queues

Legend:

Fair Share

Used

Used (over fair share)

Max Capacity

root

+ root.hoteldev

+ root.datadev

+ root.flightdev

+ root.search

+ root.secdev

+ root.traveldev

+ root.wirelessdev


+ root.default

Show 20 entries

ID	User	Name	Queue	Fair Share	Star
application 1387813121465 43865	wirelessdev	ETL_mbserver_companyKy3. 2013-12-25-10	root.wirelessdev	383147	2013-12-25

查看历史作业

← → ↻ l-hdpm4.data.cn6.qunar.com:19888/jobhistory/ ☆ ☰



JobHistory

Logged in as: root

▼ Application

[About](#)
[Jobs](#)

► Tools

Retired Jobs

Show 20 entries Search:

Start Time ↕	Finish Time ↕	Job ID ↕	Name ↕	User ↕	Queue ↕	State ↕	Maps Total ↕	Maps Completed ↕	Reduces Total ↕	Reduces Completed ↕
2013.12.25 14:26:43 CST	2013.12.25 14:26:52 CST	job 1387813121465 47022	SELECT m.stat_date, COUNT(DIST...m.stat_date(Stage	hadoop	default	SUCCEEDED	8	8	0	0
2013.12.25 14:26:25 CST	2013.12.25 14:26:42 CST	job 1387813121465 47021	create table tmp_dw_hot...x.customer_ip=b.ip(Stage	qhstats	hoteldev	SUCCEEDED	2	2	1	1
2013.12.25 14:26:23 CST	2013.12.25 14:26:37 CST	job 1387813121465 47020	SELECT m.stat_date, COUNT(DIST...m.stat_date(Stage	hadoop	default	SUCCEEDED	6	6	1	1
2013.12.25 14:26:18 CST	2013.12.25 14:26:46 CST	job 1387813121465 47019	create table tmp_hotel_ord...o.user_id=tu.id(Stage	qhstats	hoteldev	SUCCEEDED	3	3	1	1
2013.12.25 14:26:17 CST	2013.12.25 14:26:33 CST	job 1387813121465 47018	select count(distinct f2.uid) from (...null(Stage	hadoop	default	SUCCEEDED	25	25	1	1
2013.12.25	2013.12.25	job 1387813121465 47017	select qunar_global from	qhstats	hoteldev	SUCCEEDED	1	1	1	1

hadoop1.0升级到2.0

升级过程遇到的问题

原来的版本 `hadoop-CDH3`

升级后的版本 `hadoop-2.2.0`

HDFS的升级遇到的问题：

还有客户端在连接HDFS，导致升级不了。

解决办法：

暂时切换端口，切断外面的连接，升级成功。

hadoop1.0与2.0的兼容性

参数的兼容性

- * 参数的名字发生改变
- * 旧的参数依然可以使用，deprecated，warning
- * 旧参数和新参数的对应关系

<http://hadoop.apache.org/docs/r2.2.0/hadoop-project-dist/hadoop-common/DeprecatedProperties.html>

mapreduce api的兼容性

- * 使用hadoop1.0的旧 mapred包 api的程序，不需编译直接在hadoop2.0上跑
- * 使用hadoop1.0的新mapreduce包 api的程序，需要重新编译；
- * 如果用到以下几个不兼容的api，需要修改程序。
- * http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduce_Compatibility_Hadoop1_Hadoop2.html

Problematic Function

Incompatibility Issue

org.apache.hadoop.util.ProgramDriver#drive

Return type changes from void to int

org.apache.hadoop.mapred.jobcontrol.Job#getMapredJobID

Return type changes from String to JobID

org.apache.hadoop.mapred.TaskReport#getTaskId

Return type changes from String to TaskID

org.apache.hadoop.mapred.ClusterStatus#UNINITIALIZED_MEMORY_VALUE

Data type changes from long to int

org.apache.hadoop.mapreduce.filecache.DistributedCache#getArchiveTimestamps

Return type changes from long[] to String[]

org.apache.hadoop.mapreduce.filecache.DistributedCache#getFileTimestamps

Return type changes from long[] to String[]

org.apache.hadoop.mapreduce.Job#failTask

Return type changes from void to boolean

org.apache.hadoop.mapreduce.Job#killTask

Return type changes from void to boolean

org.apache.hadoop.mapreduce.Job#getTaskCompletionEvents

Return type changes from o.a.h.mapred.TaskCompletionEvent[] to o.a.h.mapreduce.TaskCompletionEvent[]

Hadoop2.0应用程序依赖的jar包

- `${HADOOP_HOME}/share/hadoop/common`
- `${HADOOP_HOME}/share/hadoop/mapreduce`
- * `hadoop-mapreduce-client-core-2.2.0.jar`
- * `hadoop-common-2.2.0.jar`
- * `hadoop-mapreduce-client-common-2.2.0.jar`
- * `hadoop-mapreduce-client-jobclient-2.2.0.jar`

Hadoop2.0程序

That is all