目录

1. 试验一：优化 Reduce 节点个数

1.1. 实验目的

完成本实验，您应该能够：

- 掌握 Reduce 节点个数的设置方式
- 掌握如何合理设置 Reduce 节点个数

1.2. 实验要求

- 熟悉 Linux 命令
- 熟悉 MapReduce 配置
- 熟悉 HDFS 文件操作

1.3. 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个节点，节点间网络互通，各节点配置：4 核 CPU、2GB 内存、30G 硬盘
运行环境	CentOS 7.4 （gui 英文版本）
用户名/密码	root/password hadoop/password
服务和组件	HDFS、YARN、MapReduce 等，其他服务根据实验需求安装
测试数据大小	13.08MB、130.84 MB、392.52MB

1.4. 实验视图

MapReduce 实验运行流程如图 1-1 所示。

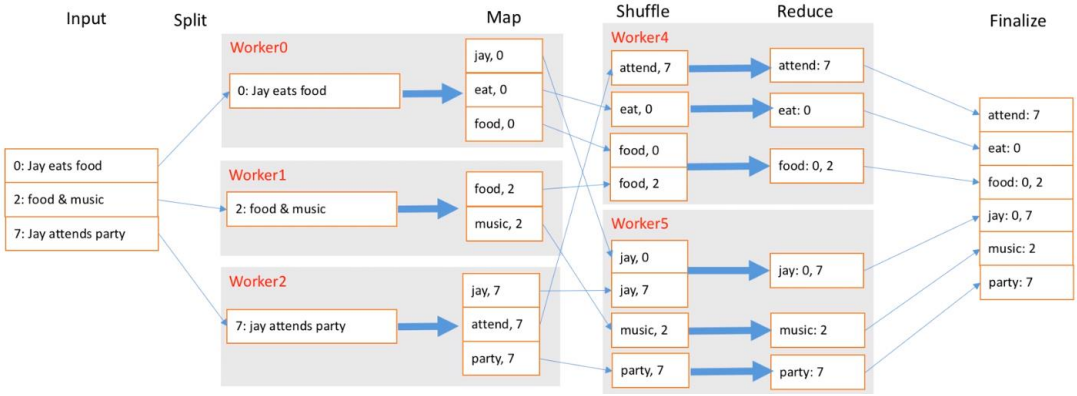


图 1-1 运行流程

1.5. 实验过程

1.5.1 实验任务一：设置 reduce 数量为 1

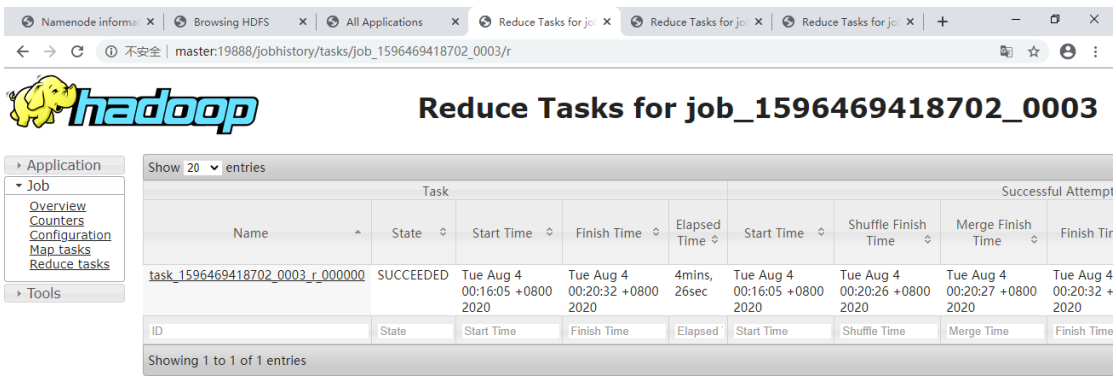
1) 修改配置文件 mapred-site.xml，设置为默认值 1

```
[hadoop@master ~]$ cd /usr/local/src/hadoop
[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml
<property>
        <name>mapreduce.job.reduces</name>
        <value>1</value>
</property>
```

2) 运行官方 MapReduce 实例

```
[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output1
……
Job Counters
        Killed map tasks=4
        Launched map tasks=9
        Launched reduce tasks=1
        Data-local map tasks=9
        Total time spent by all maps in occupied slots (ms)=2140245
        Total time spent by all reduces in occupied slots (ms)=266899
        Total time spent by all map tasks (ms)=2140245
        Total time spent by all reduce tasks (ms)=266899
        Total vcore-seconds taken by all map tasks=2140245
        Total vcore-seconds taken by all reduce tasks=266899
        Total megabyte-seconds taken by all map tasks=2191610880
        Total megabyte-seconds taken by all reduce tasks=273304576
……
File Input Format Counters
        Bytes Read=562759225
File Output Format Counters
        Bytes Written=9240222
```

3) 查看运行结果



结果显示：Reduce 经过时间为 4mins, 26sec。

1.5.2 实验任务二：设置 reduce 数量为 2

1) 修改配置文件 mapred-site.xml，设置 reduce 数量为 2

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

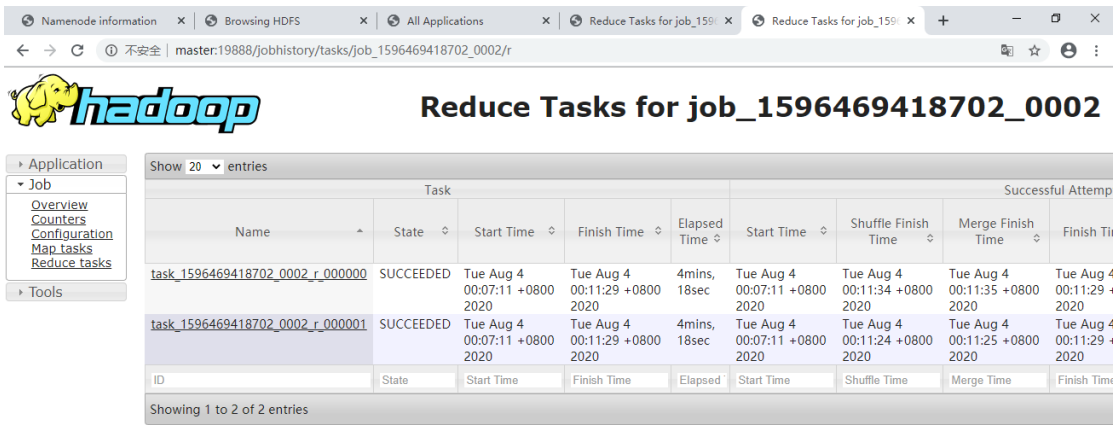
```
<property>
        <name>mapreduce.job.reduces</name>
        <value>2</value>
</property>
```

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output2

```
……
Job Counters
        Killed map tasks=4
        Launched map tasks=9
        Launched reduce tasks=2
        Data-local map tasks=9
        Total time spent by all maps in occupied slots (ms)=1978805
        Total time spent by all reduces in occupied slots (ms)=516342
        Total time spent by all map tasks (ms)=1978805
        Total time spent by all reduce tasks (ms)=516342
        Total vcore-seconds taken by all map tasks=1978805
        Total vcore-seconds taken by all reduce tasks=516342
        Total megabyte-seconds taken by all map tasks=2026296320
        Total megabyte-seconds taken by all reduce tasks=528734208
……
File Input Format Counters
        Bytes Read=562759225
File Output Format Counters
        Bytes Written=9240222
```

3) 查看运行结果



结果显示：Reduce 经过时间为 4mins, 18sec。

1.5.3 实验任务三：设置 reduce 数量为 6

1) 修改配置文件 mapred-site.xml，设置 reduce 数量为 6

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

```
<property>
    <name>mapreduce.job.reduces</name>
    <value>6</value>
</property>
```

设置依据：

mapreduce.tasktracker.reduce.tasks.maximum 参数默认值为 2，优化值：

0.95 * NUMBER_OF_NODES* mapred.tasktracker.tasks.maximum

理由：启用 95%的 Recude 任务槽运行 task，Recude task 运行一轮就可以完成。剩余 5%的任务槽永远失败任务，重新执行。

1.75 * NUMBER_OF_NODES* mapred.tasktracker.tasks.maximum

理由：因为 Recude task 数量超过 Recude 槽数，所以需要两轮才能完成所有 Recude task。

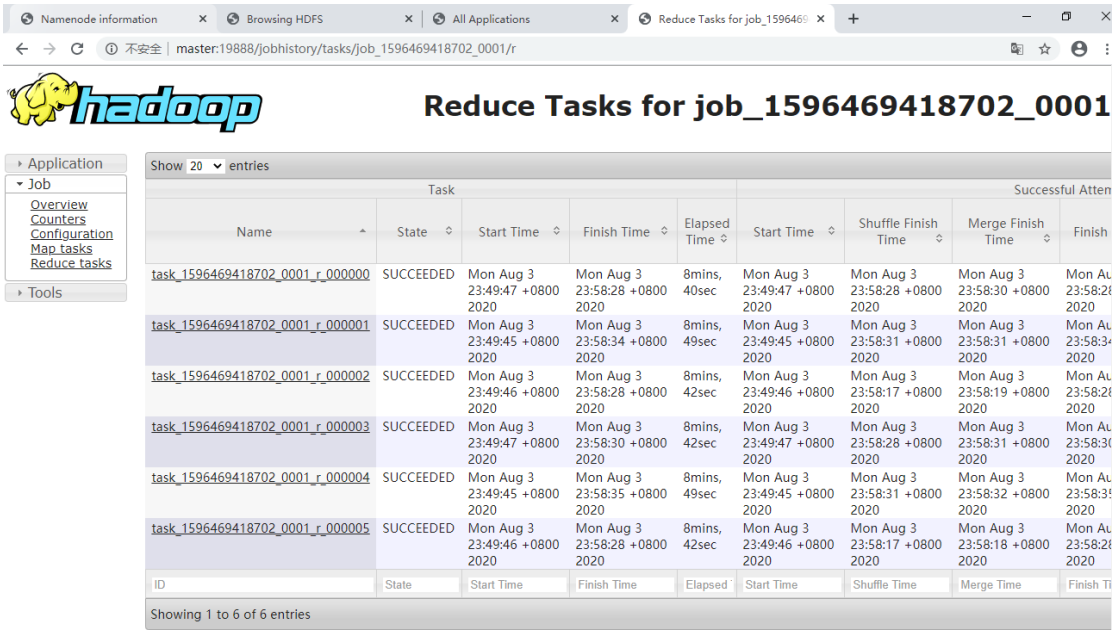
其中，NUMBER_OF_NODES 代表集群中计算节点的个数，mapred.tasktracker.reduce.tasks.maximum 代表每一个节点所分配的 Reducer 任务槽的个数。

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output3

```
……
Job Counters
        Killed map tasks=4
        Launched map tasks=9
        Launched reduce tasks=6
        Data-local map tasks=9
        Total time spent by all maps in occupied slots (ms)=4136256
        Total time spent by all reduces in occupied slots (ms)=3148261
        Total time spent by all map tasks (ms)=4136256
        Total time spent by all reduce tasks (ms)=3148261
        Total vcore-seconds taken by all map tasks=4136256
        Total vcore-seconds taken by all reduce tasks=3148261
        Total megabyte-seconds taken by all map tasks=4235526144
        Total megabyte-seconds taken by all reduce tasks=3223819264
……
File Input Format Counters
        Bytes Read=562759225
File Output Format Counters
        Bytes Written=9240222
```

3) 查看运行结果



结果显示：Reduce 经过时间为 8mins, 49sec。

1.5.4 运行结果分析与总结

1) 设置不同 reduce 任务数得到的运行结果

mapreduce.job.reduces 默认值为 1 的结果：Reduce 经过时间为 4mins, 26sec。

mapreduce.job.reduces 值为 2 的结果：Reduce 经过时间为 4mins, 18sec。

mapreduce.job.reduces 值为 6 的结果：Reduce 经过时间为 8mins, 49sec。

2) 结果分析

通过以上三个结果，mapreduce.job.reduces，设置为 2 时效率最佳。Reduce 任务是一个数据聚合的步骤，数量默认为 1。太少，会导致 task 等待，延长处理时间；太多，会导致 Map、Reduce 任务间竞争资源，造成处理超时等错误，使用过多的 Reduce 任务则意味着复杂的 shuffle，并使输出文件的数量激增。

2 试验二：优化 Reduce I/O 相关参数

2.1 实验目的

优化 Reduce I/O 相关参数：mapreduce.task.io.sort.factor

2.2 实验要求

- 了解 MapReduce 作业过程
- 熟悉 Hadoop 配置文件
- 熟悉 MapReduce 代码运行

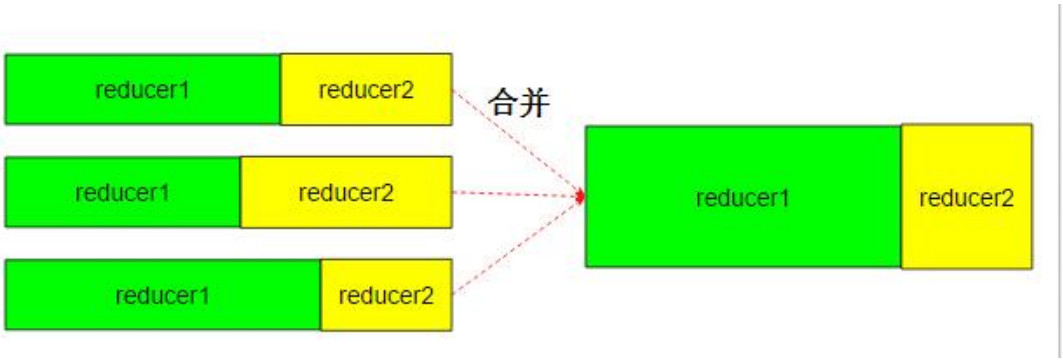
2.3 实验环境

本实验所需之主要资源环境如表 2-1 所示。

表 2-1 资源环境

服务器集群	3 个节点，节点间网络互通，各节点配置：4 核 CPU、2GB 内存、30G 硬盘
运行环境	CentOS 7.4 （gui 英文版本）
用户名/密码	root/password hadoop/password
服务和组件	HDFS、YARN、MapReduce 等，其他服务根据实验需求安装
测试数据大小	631.39 MB

2.4 实验视图



2.5 实验过程

2.5.1 实验任务一：使用参数的默认值

1) 修改配置文件 mapred-site.xml，设置为默认值 10

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

<property>

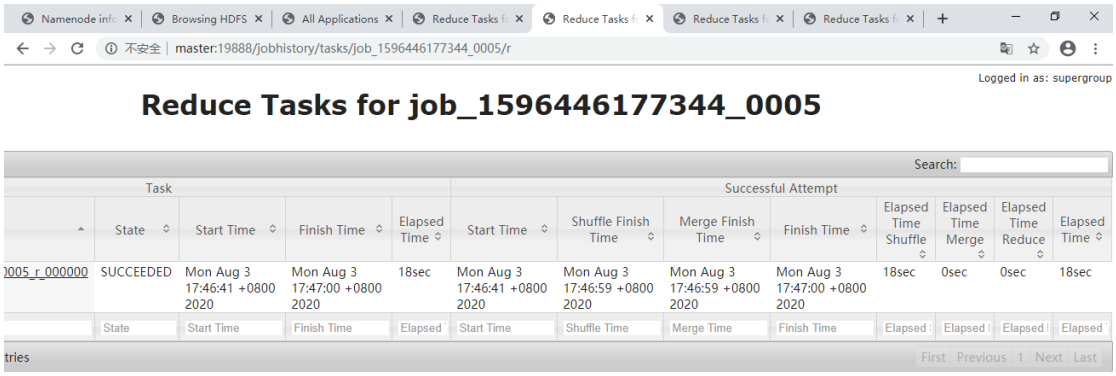
```
        <name>mapreduce.task.io.sort.factor</name>
        <value>10</value>
    </property>
```

默认值为 10，Reduce Task 中合并小文件时，一次合并的文件数据，每次合并的时候选择最小的前 10 进行合并。

2) 运行官方 MapReduce 实例

```
[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output1
    ……
    Job Counters
        Launched map tasks=5
        Launched reduce tasks=1
        Data-local map tasks=5
        Total time spent by all maps in occupied slots (ms)=985700
        Total time spent by all reduces in occupied slots (ms)=18992
        Total time spent by all map tasks (ms)=985700
        Total time spent by all reduce tasks (ms)=18992
        Total vcore-seconds taken by all map tasks=985700
        Total vcore-seconds taken by all reduce tasks=18992
        Total megabyte-seconds taken by all map tasks=1009356800
        Total megabyte-seconds taken by all reduce tasks=19447808
    ……
    File Input Format Counters
        Bytes Read=662588576
    File Output Format Counters
        Bytes Written=777947
```

3) 查看运行结果



结果显示：测试数据 631.39MGB，一次合并的文件数据设置为 10， reduce 合并时间为 18sec。

2.5.2 实验任务二：修改合并参数值为 20

1) 修改配置文件 mapred-site.xml，设置为 20

```
[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml
    <property>
```

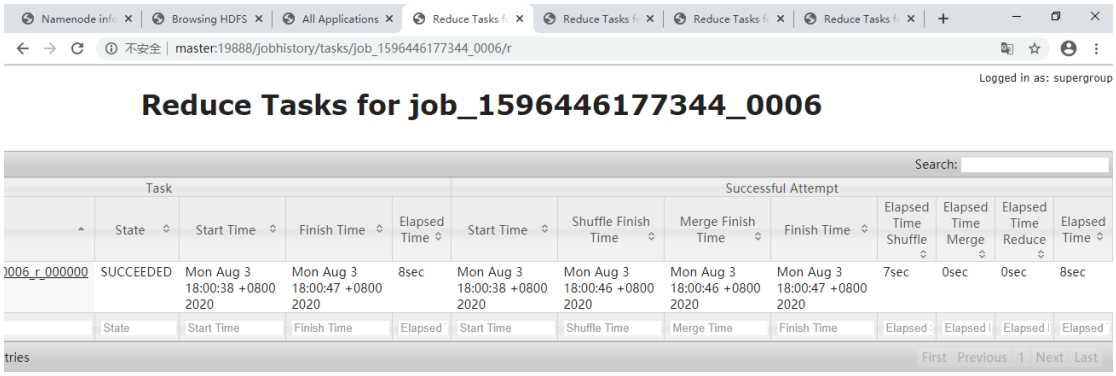
```
        <name>mapreduce.task.io.sort.factor</name>
        <value>20</value>
    </property>
```

这里修改 mapreduce.task.io.sort.factor 属性配置为 20，即一次最多合并 20 个 spill 文件。

2) 运行官方 MapReduce 实例

```
[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output2
    ……
    Job Counters
        Launched map tasks=5
        Launched reduce tasks=1
        Data-local map tasks=5
        Total time spent by all maps in occupied slots (ms)=639321
        Total time spent by all reduces in occupied slots (ms)=8471
        Total time spent by all map tasks (ms)=639321
        Total time spent by all reduce tasks (ms)=8471
        Total vcore-seconds taken by all map tasks=639321
        Total vcore-seconds taken by all reduce tasks=8471
        Total megabyte-seconds taken by all map tasks=654664704
        Total megabyte-seconds taken by all reduce tasks=8674304
    ……
    File Input Format Counters
        Bytes Read=662588576
    File Output Format Counters
        Bytes Written=777947
```

3) 查看运行结果



结果显示：测试数据 631.39MB，一次合并的文件数据设置为 20，Reduce 合并时间为 8sec。

2.5.3 实验结果分析与总结

通过试验对比，设置 mapreduce.task.io.sort.factor 参数为 20 时，reduce 合并时间比默认值 10 时快了 10 秒，每一轮合并不一定合并平均数量的文件数，指导原则是使用整个合并过程中写入磁盘的数据量最小，为了达到这个目的，则需要最终的一轮合并中合并尽可能多的

数据，因为最后一轮的数据直接作为 reduce 的输入，无需写入磁盘再读出。因此我们让最终的一轮合并的文件数达到最大，即合并因子的值，通过 mapreduce.task.io.sort.factor 来配置。

3 试验三：优化 Reduce shuffle 阶段并行传输数据的数量

3.1 实验目的

如果 map task 有 100 个, reducer 有 5 个, 平均每个 reducer 需要拉取 20 个 map task 的输出结果, 但默认情况下 reducer 会初始化 5 个拉取数据的线程, 逐次从 map 端 copy 适当地增加 reduce 端拉取 map 数据的线程数, 让 shuffle 过程执行的更快些。设置 mapreduce.reduce.shuffle.parallelcopies 实现优化 Reduce shuffle 阶段并行传输数据的数量。

3.2 实验要求

- 了解 MapReduce 作业过程
- 熟悉 Hadoop 配置文件
- 熟悉 MapReduce 代码运行

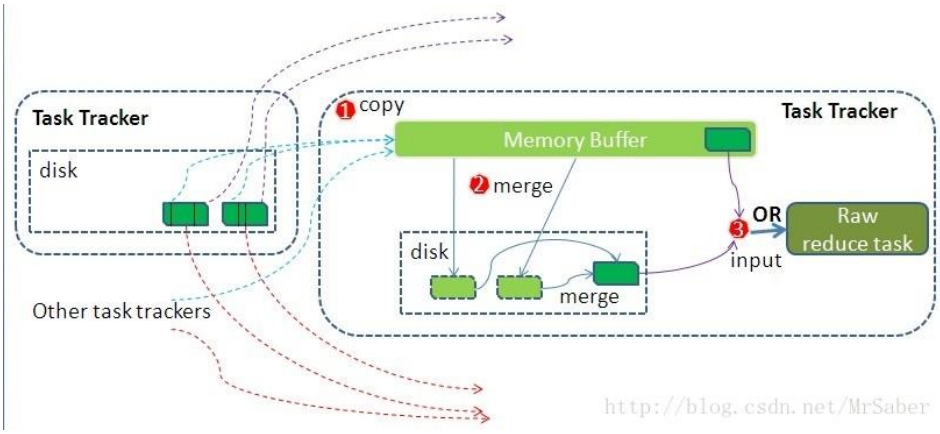
3.3 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个节点，节点间网络互通，各节点配置：4 核 CPU、2GB 内存、30G 硬盘
运行环境	CentOS 7.4 （gui 英文版本）
用户名/密码	root/password hadoop/password
服务和组件	HDFS、YARN、MapReduce 等，其他服务根据实验需求安装
测试数据大小	104.67 MB、130.84 MB、392.52 MB

3.4 实验视图



3.5 实验过程

3.5.1 实验任务一：测试数据 628.03MB，参数值取默认值 5

1) 修改配置文件 mapred-site.xml，设置为默认值 5

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

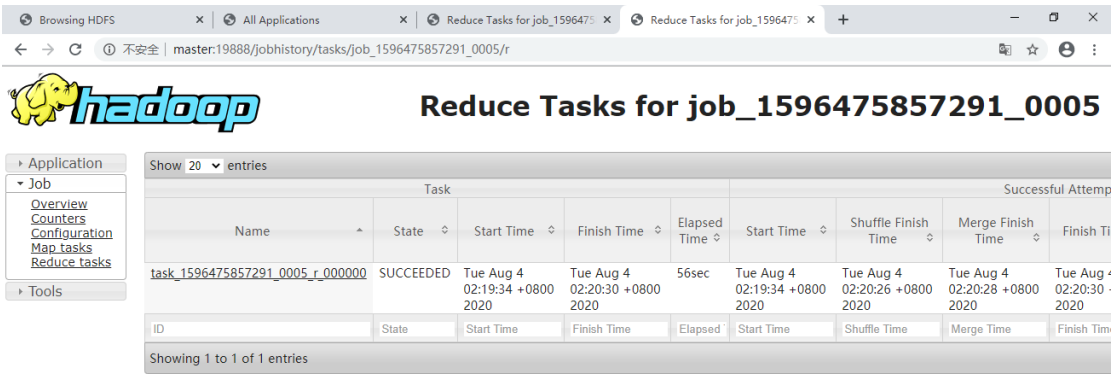
```
<property>
    <name>mapreduce.reduce.shuffle.parallelcopies</name>
    <value>5</value>
</property>
```

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output1

```
……
Job Counters
        Killed map tasks=1
        Launched map tasks=6
        Launched reduce tasks=1
        Data-local map tasks=6
        Total time spent by all maps in occupied slots (ms)=985946
        Total time spent by all reduces in occupied slots (ms)=56247
        Total time spent by all map tasks (ms)=985946
        Total time spent by all reduce tasks (ms)=56247
        Total vcore-seconds taken by all map tasks=985946
        Total vcore-seconds taken by all reduce tasks=56247
        Total megabyte-seconds taken by all map tasks=1009608704
        Total megabyte-seconds taken by all reduce tasks=57596928
……
File Input Format Counters
        Bytes Read=658795312
File Output Format Counters
        Bytes Written=9241288
```

3) 查看运行结果



结果显示：测试数据 628.03MB，Reduce 合并时间为 56 秒。

3.5.2 实验任务二：测试数据 628.03MB，参数值 10

1) 修改配置文件 mapred-site.xml，设置为 10

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

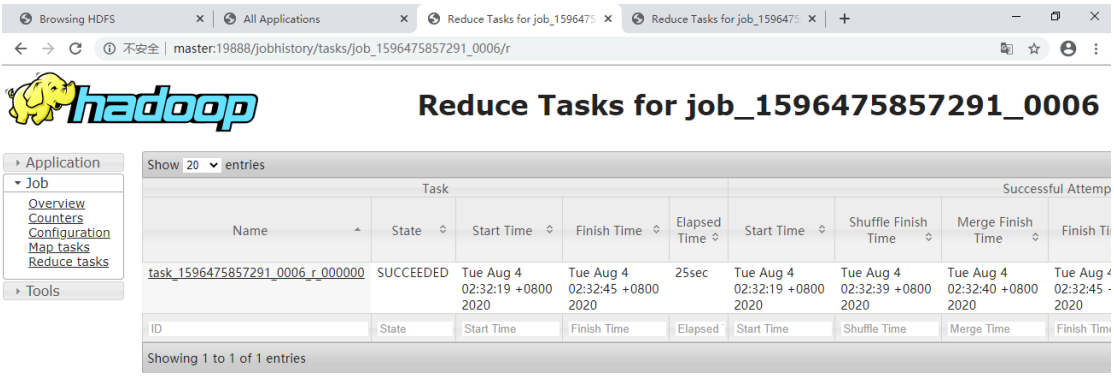
```
<property>
    <name>mapreduce.reduce.shuffle.parallelcopies</name>
    <value>10</value>
</property>
```

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output2

```
……
Job Counters
        Launched map tasks=5
        Launched reduce tasks=1
        Data-local map tasks=5
        Total time spent by all maps in occupied slots (ms)=940827
        Total time spent by all reduces in occupied slots (ms)=25942
        Total time spent by all map tasks (ms)=940827
        Total time spent by all reduce tasks (ms)=25942
        Total vcore-seconds taken by all map tasks=940827
        Total vcore-seconds taken by all reduce tasks=25942
        Total megabyte-seconds taken by all map tasks=963406848
        Total megabyte-seconds taken by all reduce tasks=26564608
……
File Input Format Counters
        Bytes Read=658795312
File Output Format Counters
        Bytes Written=9241288
```

3) 查看运行结果



结果显示：测试数据 628.03MB，Reduce 合并时间为 25 秒。对比任务一效率上得到了提升。

3.5.3实验结果分析与总结

本试验 Reduce tasks 数设置为 1，具体可以根据需求修改，此试验中两个实验任务的结果对比显示并发复制线程数越大效率越高。但是需要说明的是 mapreduce.reduce.shuffle.parallelcopies 值也并不是越大越好，需要根据 Map task 和 Reduce task 具体值来判断设置，当 Map task 与 Reduce task 比例过小时，设置并发复制线程数过大反而影响运行效率。

4 试验四：优化 tasktracker 并发执行的 reduce 数

4.1 实验目的

优化 tasktracker 并发执行的 reduce 数，通过修改配置中 mapred.tasktracker.reduce.tasks.maximum 参数的值实现优化效果，通过验证将参数值修改为 CPU 数量作业运行效率与默认值进行效果对比。

4.2 实验要求

- 了解 MapReduce 作业过程
- 熟悉 Hadoop 配置文件
- 熟悉 MapReduce 代码运行

4.3 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个节点，节点间网络互通，各节点配置：4 核 CPU、2GB 内存、30G 硬盘
运行环境	CentOS 7.4 （gui 英文版本）
用户名/密码	root/password hadoop/password
服务和组件	HDFS、YARN、MapReduce 等，其他服务根据实验需求安装
测试数据大小	104.67 MB、115.3 MB、130.84 MB、392.52 MB

4.4 实验视图

无

4.5 实验过程

4.5.1 实验任务一：测试参数默认值效果

1) 修改配置文件 mapred-site.xml，设置为默认值 2

```
[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml
    <property>
        <name>mapred.tasktracker.reduce.tasks.maximum</name>
        <value>2</value>
    </property>
```

2) 重启 hadoop 集群

```
[hadoop@master hadoop]$ stop-all.sh
```

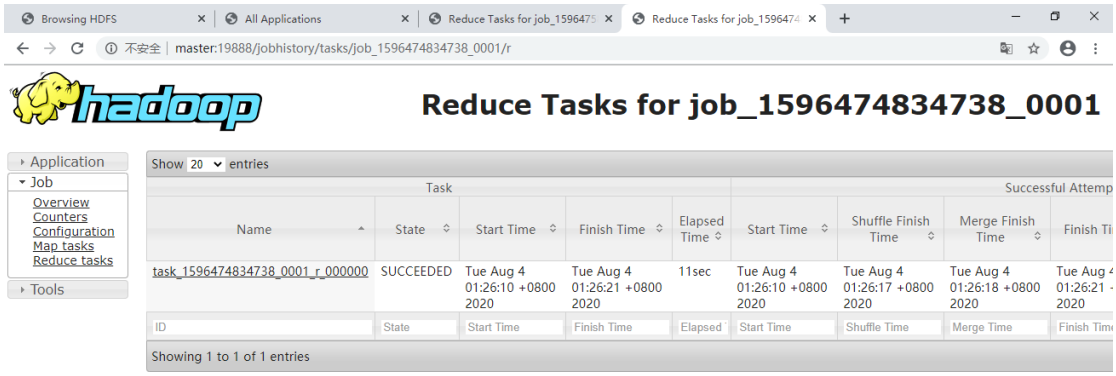
[hadoop@master hadoop]$ start-all.sh

3) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output1

```
……
Job Counters
        Launched map tasks=4
        Launched reduce tasks=1
        Data-local map tasks=4
        Total time spent by all maps in occupied slots (ms)=509547
        Total time spent by all reduces in occupied slots (ms)=9243
        Total time spent by all map tasks (ms)=509547
        Total time spent by all reduce tasks (ms)=9243
        Total vcore-seconds taken by all map tasks=509547
        Total vcore-seconds taken by all reduce tasks=9243
        Total megabyte-seconds taken by all map tasks=521776128
        Total megabyte-seconds taken by all reduce tasks=9464832
……
File Input Format Counters
        Bytes Read=779695757
File Output Format Counters
        Bytes Written=963558
```

4) 查看运行结果



结果描述：测试数据共 743.33 MB，经过时间 11 秒。

4.5.2 实验任务二：根据规则优化参数设置

1) 修改配置文件 mapred-site.xml，设置为 1

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

```
<property>
        <name> mapred.tasktracker.reduce.tasks.maximum </name>
        <value>1</value>
</property>
```

规则：

当 CPU 数量大于 2 则该参数设置为 CPU 数量 * 0.50 否则设置为 1。

2) 重启 hadoop 集群

[hadoop@master hadoop]$ stop-all.sh

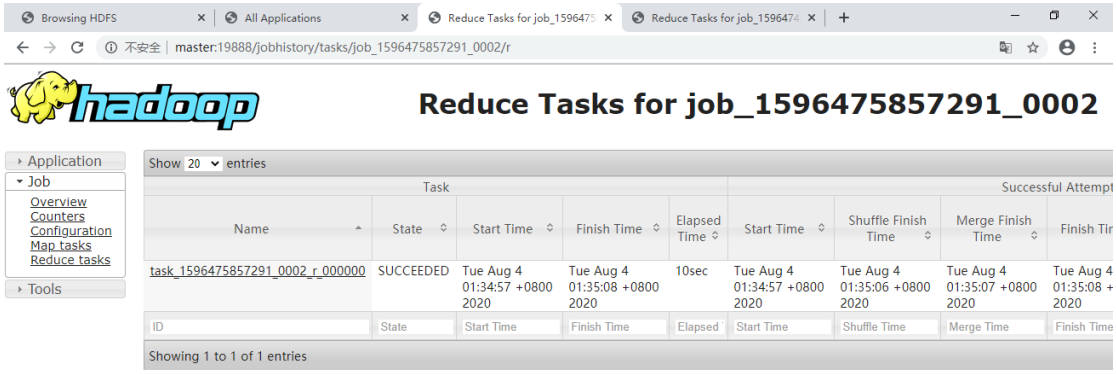
[hadoop@master hadoop]$ start-all.sh

3) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output2

```
……
Job Counters
        Launched map tasks=4
        Launched reduce tasks=1
        Data-local map tasks=4
        Total time spent by all maps in occupied slots (ms)=477573
        Total time spent by all reduces in occupied slots (ms)=10875
        Total time spent by all map tasks (ms)=477573
        Total time spent by all reduce tasks (ms)=10875
        Total vcore-seconds taken by all map tasks=477573
        Total vcore-seconds taken by all reduce tasks=10875
        Total megabyte-seconds taken by all map tasks=489034752
        Total megabyte-seconds taken by all reduce tasks=11136000
……
File Input Format Counters
        Bytes Read=779695757
File Output Format Counters
        Bytes Written=963558
```

4) 查看运行结果



结果描述：测试数据共 743.33 MB，经过时间 10 秒。

4.5.3 实验结果分析与总结

由于本次测试所采用的虚拟机集群配置限制，测试的数据相对较小，得出的效果差距不够明显，但是根据以上两次任务结果对比，当 CPU 数量大于 2 则将 mapreduce.tasktracker.tasks.reduce.maximum 设置为 CPU 数量 *0.50，否则设置为 1 比较合适。

5 试验五：优化可并发处理来自 tasktracker 的 RPC 请求数

5.1 实验目的

在集群相对较大的时候，JobTracker 的处理 RPC 能力就可能有些不足了。JobTracker 需要并发处理来自各个 TaskTracker 的 RPC 请求，管理员可根据集群规模和服务器并发处理能够调整 RPC Handler 数目，以使 JobTracker 服务能力最佳。JobTracker 为每个 TaskTracker 服务的时间差不多是 100ms，默认有 10 个 handler，JobTracker 每秒可处理差不多 100 个 TaskTracker 请求，如果 TaskTracker 数量很大，那么很有必要增加这个设置来提高 JobTracker 的响应。

5.2 实验要求

- 了解 MapReduce 作业过程
- 熟悉 Hadoop 配置文件
- 熟悉 MapReduce 代码运行

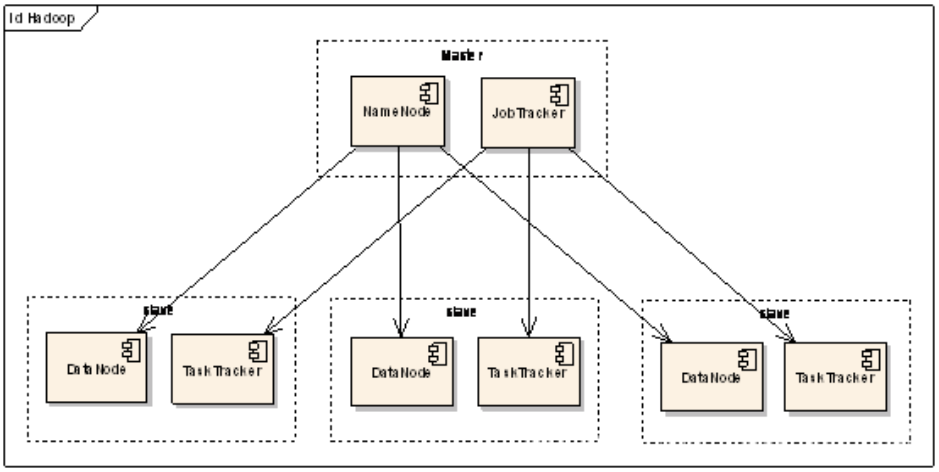
5.3 实验环境

本实验所需之主要资源环境如表 1-1 所示。

表 1-1 资源环境

服务器集群	3 个节点，节点间网络互通，各节点配置：4 核 CPU、2GB 内存、30G 硬盘
运行环境	CentOS 7.4 （gui 英文版本）
用户名/密码	root/password hadoop/password
服务和组件	HDFS、YARN、MapReduce 等，其他服务根据实验需求安装
测试数据大小	104.67 MB、130.84 MB、392.52 MB

5.4 实验视图



5.5 实验过程

5.5.1 实验任务一：参数为默认值 10

1) 修改配置文件 mapred-site.xml，设置参数为默认值 10

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

```
<property>
    <name> mapreduce.jobtracker.handler.count</name>
    <value>10</value>
</property>
```

2) 运行官方 MapReduce 实例

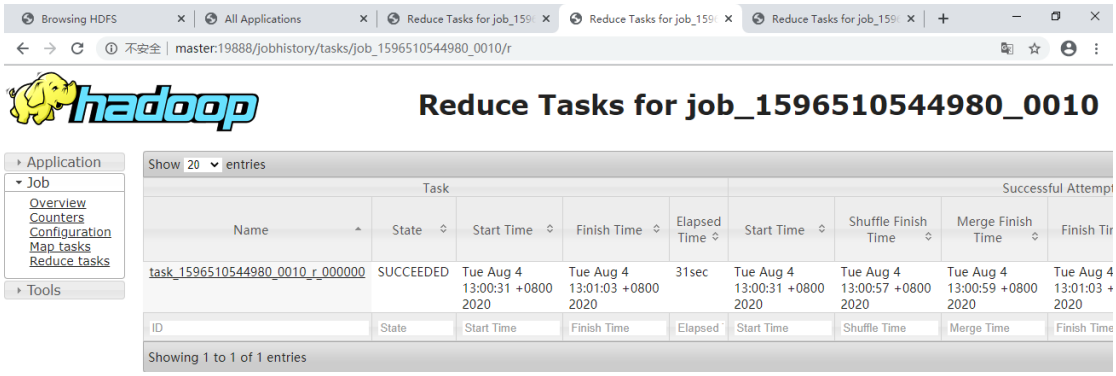
[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output1

```
……
Job Counters
    Launched map tasks=5
    Launched reduce tasks=1
    Data-local map tasks=5
    Total time spent by all maps in occupied slots (ms)=824740
    Total time spent by all reduces in occupied slots (ms)=31946
    Total time spent by all map tasks (ms)=824740
    Total time spent by all reduce tasks (ms)=31946
    Total vcore-seconds taken by all map tasks=824740
    Total vcore-seconds taken by all reduce tasks=31946
    Total megabyte-seconds taken by all map tasks=844533760
    Total megabyte-seconds taken by all reduce tasks=32712704
……
File Input Format Counters
    Bytes Read=658795312
```

File Output Format Counters

Bytes Written=9241288

3) 查看运行结果



结果显示：Reduce 合并时间为：31sec。

5.5.2 实验任务二：设置参数值为 20

1) 修改配置文件 mapred-site.xml，设置参数为 20

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

```
<property>
    <name> mapreduce.jobtracker.handler.count</name>
    <value>20</value>
</property>
```

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output2

……

Job Counters

Launched map tasks=5

Launched reduce tasks=1

Data-local map tasks=5

Total time spent by all maps in occupied slots (ms)=781394

Total time spent by all reduces in occupied slots (ms)=33056

Total time spent by all map tasks (ms)=781394

Total time spent by all reduce tasks (ms)=33056

Total vcore-seconds taken by all map tasks=781394

Total vcore-seconds taken by all reduce tasks=33056

Total megabyte-seconds taken by all map tasks=800147456

Total megabyte-seconds taken by all reduce tasks=33849344

……

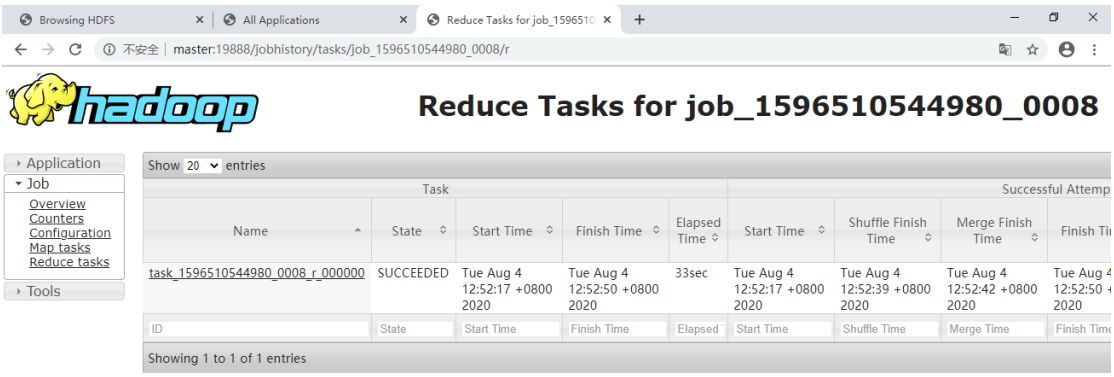
File Input Format Counters

Bytes Read=658795312

File Output Format Counters

Bytes Written=9241288

3) 查看运行结果



结果显示：Reduce 合并时间为：33sec。

5.5.3 实验任务三：设置参数值为 5

1) 修改配置文件 mapred-site.xml，设置参数为 5

[hadoop@master hadoop]$ vi /usr/local/src/hadoop/etc/hadoop/mapred-site.xml

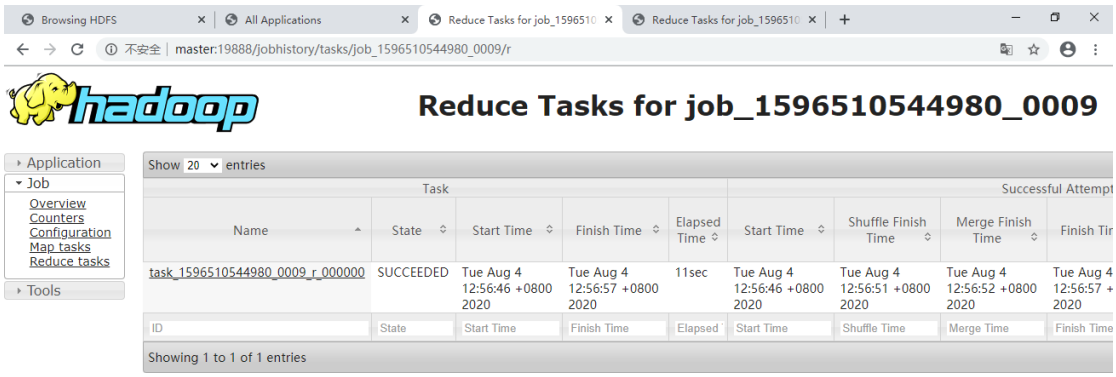
```
<property>
    <name> mapreduce.jobtracker.handler.count</name>
    <value>5</value>
</property>
```

2) 运行官方 MapReduce 实例

[hadoop@master hadoop]$ hadoop jar /usr/local/src/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount /input1 /output3

```
……
Job Counters
        Launched map tasks=5
        Launched reduce tasks=1
        Data-local map tasks=5
        Total time spent by all maps in occupied slots (ms)=872699
        Total time spent by all reduces in occupied slots (ms)=11062
        Total time spent by all map tasks (ms)=872699
        Total time spent by all reduce tasks (ms)=11062
        Total vcore-seconds taken by all map tasks=872699
        Total vcore-seconds taken by all reduce tasks=11062
        Total megabyte-seconds taken by all map tasks=893643776
        Total megabyte-seconds taken by all reduce tasks=11327488
……
File Input Format Counters
        Bytes Read=658795312
File Output Format Counters
        Bytes Written=9241288
```

3) 查看运行结果



结果显示：Reduce 合并时间为：11sec。

5.5.4 实验结果分析与总结

mapreduce.jobtracker.handler.count 参数的表示对于 JobTracker 服务器线程数。这官方建议该值应该是大约 4%的 TaskTracker 节点数。但是由于此次试验的条件限制无法满足该条件的测试，因此本次试验使用的是虚拟机搭建的 hadoop 全分布式集群，测试的可并发处理 RPC 请求数的值分别 5，10 和 20，其中 10 为默认值；根据测试结果，参数值增加时效率下降；参数值减少是效率提高。