

MapReduce

Simplified Data Processing on Large Clusters

严春伟

互联网研发中心

December 28, 2012

Outline

Motivation

大数据的处理

- ① Google 至少有 $8 * 10^9$ 个网页

Motivation

大数据的处理

- ① Google 至少有 $8 * 10^9$ 个网页
- ② 成效和成本

Motivation

大数据的处理

- ① Google 至少有 $8 * 10^9$ 个网页
- ② 成效和成本
- ③ 利用成百上千的 CPU

MapReduce 的解决方案

- ① 自动化的并行分发控制
- ② 容错性
- ③ I/O 调度

What is Map/Reduce

Map in LISP(Scheme)

- **(map** *f* *list*[*list*₁, *list*₂, \dots , *list*_{*n*}])
- **(map** *square* (list 1 2 3 4))
 - ▶ (1 4 9 16)

Reduce in LISP

- **(reduce** *f* *list*[*list*₁, *list*₂, \dots , *list*_{*n*}])
- **(reduce** *+* 0 (list 1 4 9 16))
 - ▶ (+ 16 (+ 9 (+ 4 (+ 1 0))))
 - ▶ 30

Key/Value

- (**map** mapper list $[(key, val)_1, (key, val)_2, \dots, (key, val)_n]$)
 - ▶ map 对 list 中每一个 (key,val) 键值处理
 - ▶ list $[(key, (mapper \ val))_1, (key, (mapper \ val))_2, \dots]$
- reduce(key, vals)
 - ▶ 对每一个独特的 key 合并 val
 - ▶ 最终的输出

Count Words in Docs

Algorithm Code Demo

```
map(key=url, val=contents):  
    For each word w in contents, emit (w, "1")  
reduce(key=word, values=uniq_counts):  
    Sum all "1"s in values list  
    Emit result "(word,sum)"
```

Count, Illustrated

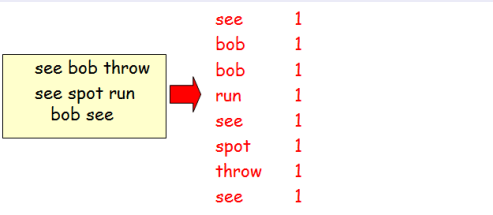
```
see bob throw  
see spot run  
bob see
```


Count Words in Docs

Algorithm Code Demo

```
map(key=url, val=contents):  
    For each word w in contents, emit (w, "1")  
reduce(key=word, values=uniq_counts):  
    Sum all "1"s in values list  
    Emit result "(word,sum)"
```

Count, Illustrated



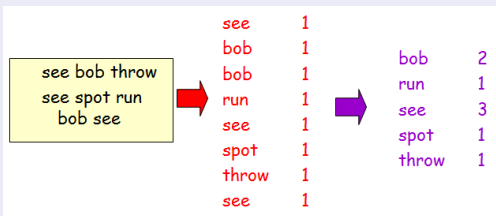
see	1
bob	1
bob	1
run	1
see	1
spot	1
throw	1
see	1

Count Words in Docs

Algorithm Code Demo

```
map(key=url, val=contents):  
    For each word w in contents, emit (w, "1")  
reduce(key=word, values=uniq_counts):  
    Sum all "1"s in values list  
    Emit result "(word,sum)"
```

Count, Illustrated



More Demos

- 倒转网络连接图

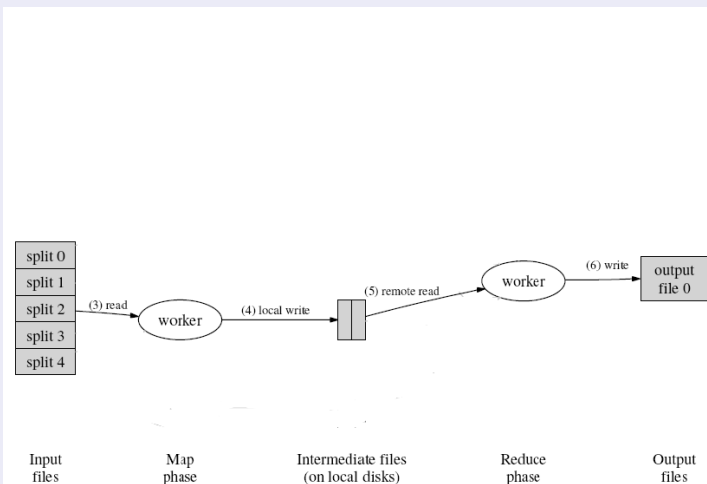
- ▶ Map 在网页 (source) 中搜索链接目标 (target), 输出为 (target, source)
- ▶ Reduce 合并相同的 target, 输出 (target, list(source))

- 计算 URL 访问频率

- ▶ Map 处理日志中 web 页面请求的记录, 输出 (URL, 1)
- ▶ Reduce 把相同 URL 的 value 累加, 产生 (URL, 记录总数) 结果

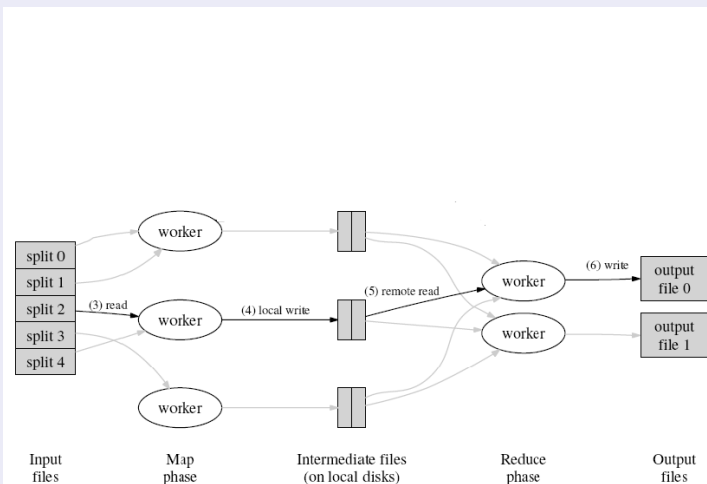
结构

模型实现



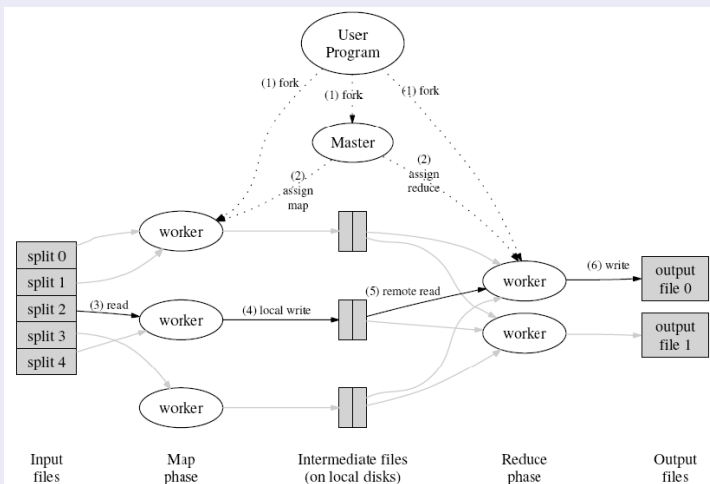
结构

更多的 worker 加入



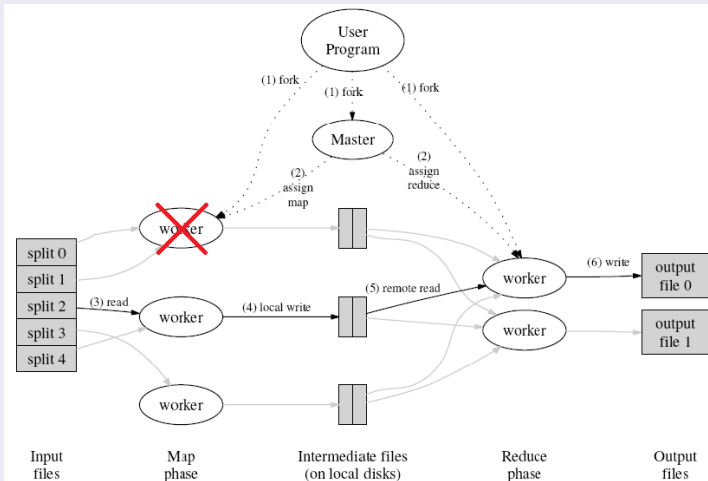
结构

中央控制 master



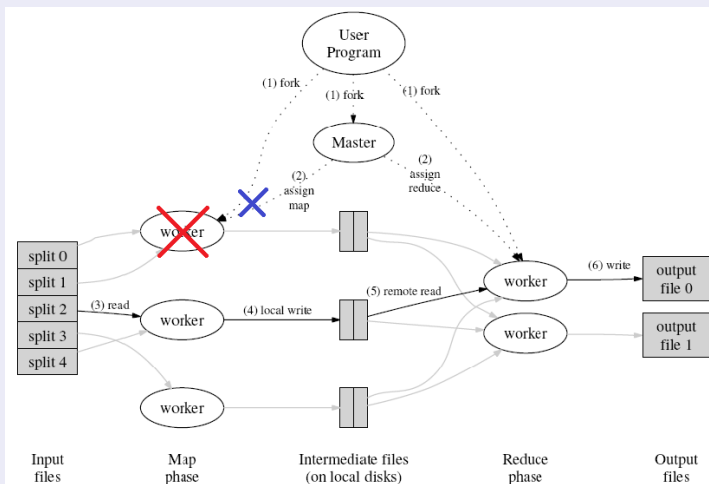
容错

worker 故障



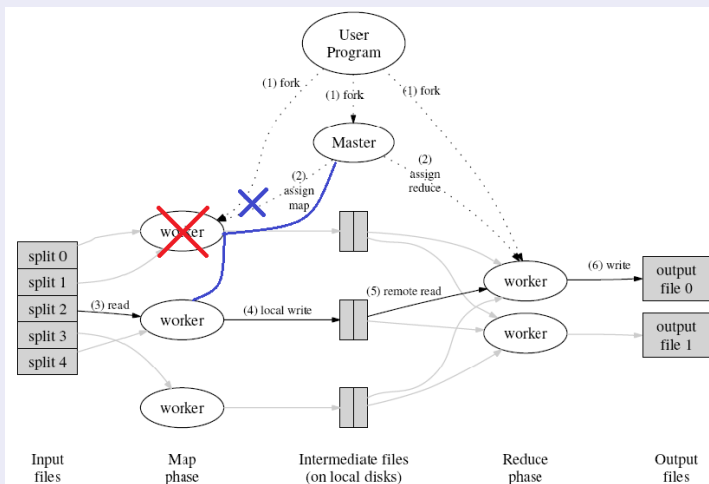
容错

worker 故障



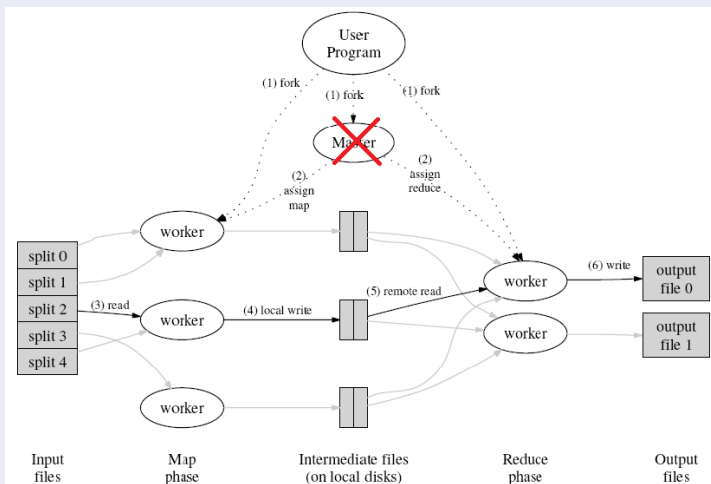
容错

worker 故障



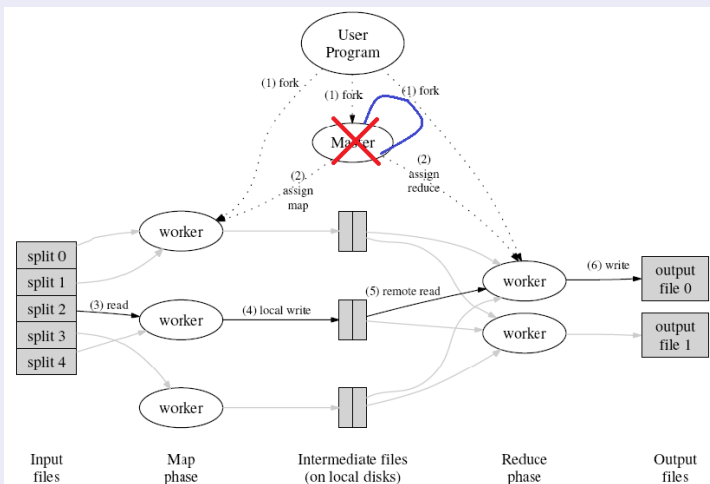
master 失败

master 故障



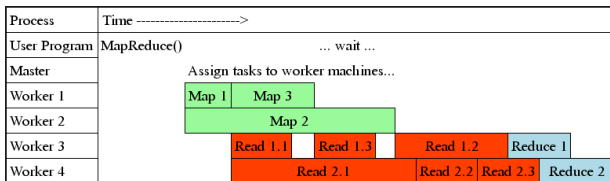
master 失败

根据最后断点，重新建立进程



性能

- 在 2000 台机器上运行
- 使用 2×10^5 个 map 任务 & 5000 个 reduce 任务



MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

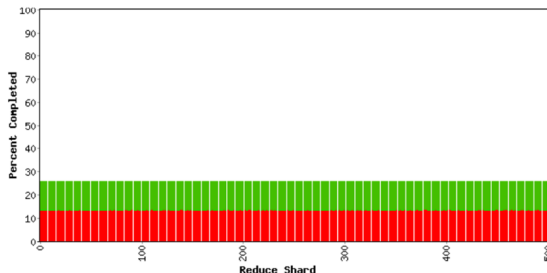
Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 05 min 07 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	1857	1707	878934.6	191995.8	113936.6
Shuffle	500	0	500	113936.6	57113.7	57113.7
Reduce	500	0	0	57113.7	0.0	0.0

Counters

Variable	Minute
Mapped (MB/s)	699.1
Shuffle (MB/s)	349.5
Output (MB/s)	0.0
doc-index-hits	5004411944
docs-indexed	17290135
dups-in-index-merge	0
mr-operator-calls	17331371
mr-operator-outputs	17290135



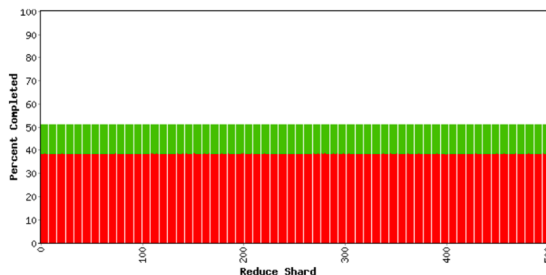
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 10 min 18 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	5354	1707	878934.6	406020.1	241058.2
Shuffle	500	0	500	241058.2	196362.5	196362.5
Reduce	500	0	0	196362.5	0.0	0.0



Counters

Variable	Minute
Mapped (MB/s)	704.4
Shuffle (MB/s)	371.9
Output (MB/s)	0.0
doc-index-hits	5000364228
docs-indexed	17300709
dups-in-index-merge	0
mr-operator-calls	17342493
mr-operator-outputs	17300709

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

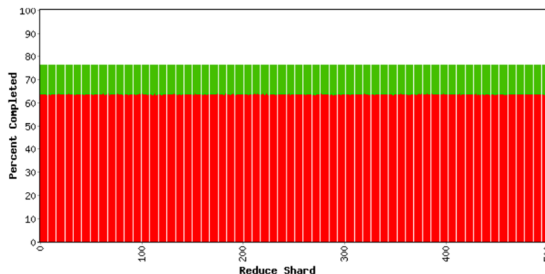
Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 15 min 31 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	8841	1707	878934.6	621608.5	369459.8
Shuffle	500	0	500	369459.8	326986.8	326986.8
Reduce	500	0	0	326986.8	0.0	0.0

Counters

Variable	Minute
Mapped (MB/s)	706.5
Shuffle (MB/s)	419.2
Output (MB/s)	0.0
doc-index-hits	4982870667
docs-indexed	17229926
dups-in-index-merge	0
mr-operator-calls	17272056
mr-operator-outouts	17229926



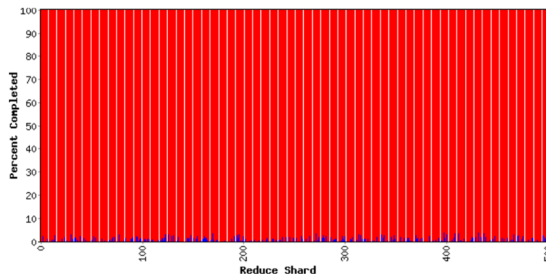
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 29 min 45 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	195	305	523499.2	523389.6	523389.6
Reduce	500	0	195	523389.6	2685.2	2742.6



Counters

Variable	Minute
Mapped (MB/s)	0.3
Shuffle (MB/s)	0.5
Output (MB/s)	45.7
doc-index-hits	2313178
docs-indexed	7936
dups-in-index-merge	0
mr-merge-calls	1954105
mr-merge-outputs	1954105

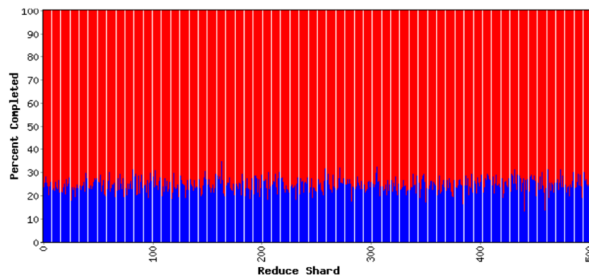
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 31 min 34 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	133837.8	136929.6



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.1
Output (MB/s)	1238.8
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51738599
mr-merge-outputs	51738599

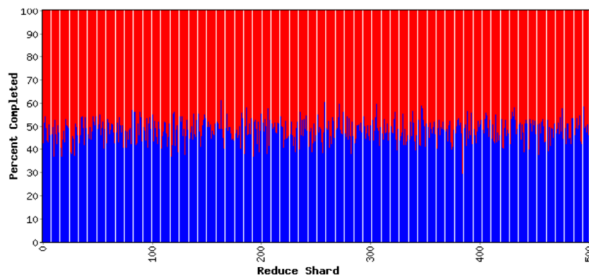
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 33 min 22 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	263283.3	269351.2



Counters

Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	1225.1
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51842100
mr-merge-outputs	51842100

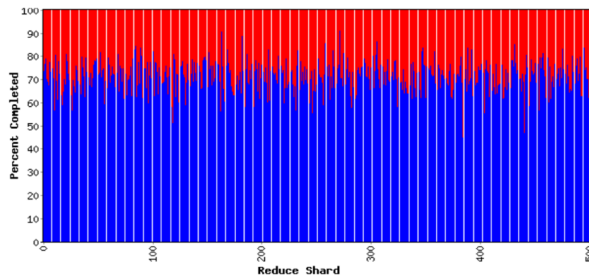
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 35 min 08 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	523499.5	523499.5
Reduce	500	0	500	523499.5	390447.6	399457.2



Counters

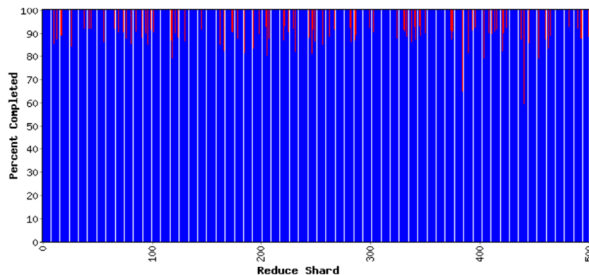
Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	1222.0
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	51640600
mr-merge-outputs	51640600

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 37 min 01 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	520468.6	520468.6
Reduce	500	406	94	520468.6	512265.2	514373.3



Counters

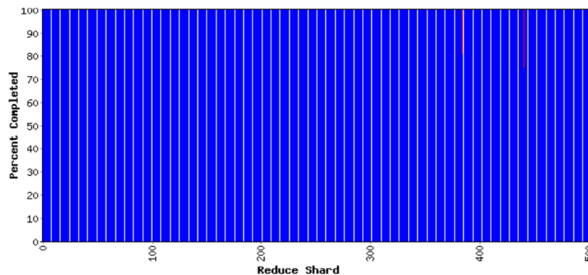
Variable	Minute
Mapped (MB/s)	0.0
Shuffle (MB/s)	0.0
Output (MB/s)	849.5
doc-index-hits	0 10
docs-indexed	0
dups-in-index-merge	0
mr-merge-calls	35083350
mr-merge-outputs	35083350

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 38 min 56 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	519781.8	519781.8
Reduce	500	498	2	519781.8	519394.7	519440.7



Counters

Variable	Minute	
Mapped (MB/s)	0.0	
Shuffle (MB/s)	0.0	
Output (MB/s)	9.4	
doc-index-hits	0	1056
docs-indexed	0	1
dups-in-index-merge	0	
mr-merge-calls	394792	1
mr-merge-outputs	394792	1

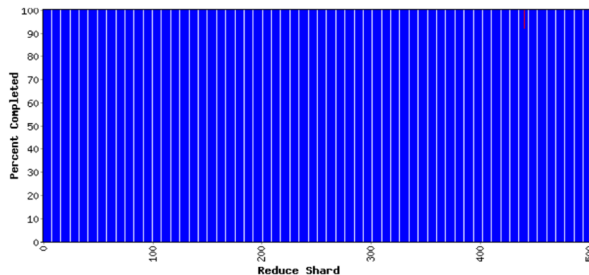
性能

MapReduce status: MR_Indexer-beta6-large-2003_10_28_00_03

Started: Fri Nov 7 09:51:07 2003 -- up 0 hr 40 min 43 sec

1707 workers; 1 deaths

Type	Shards	Done	Active	Input(MB)	Done(MB)	Output(MB)
Map	13853	13853	0	878934.6	878934.6	523499.2
Shuffle	500	500	0	523499.2	519774.3	519774.3
Reduce	500	499	1	519774.3	519735.2	519764.0



Counters

Variable	Minute	
Mapped (MB/s)	0.0	
Shuffle (MB/s)	0.0	
Output (MB/s)	1.9	
doc-index-hits	0	105
docs-indexed	0	:
dups-in-index-merge	0	
mr-merge-calls	73442	:
mr-merge-outputs	73442	:

References

 [Jeffrey Dean and Sanjay Ghemawat, 2004]

Google, Inc

MapReduce: Simplified Data Processing on Large Cluster, 2004

 [Dan Weld]

Dan Weld's class at U. Washington

 [Rajesh Gadipuri]

MapReduce Online

 [Anand Rajaraman, Dan Weld]

Stanford Univ. , Univ. of Washington

Map Reduce Architecture

Conclusion

- Map/Reduce 良好的模型抽象
- 极大地简化了大规模计算
- 弹性容错 & 可靠的计算框架
- 数量取胜，有效而廉价