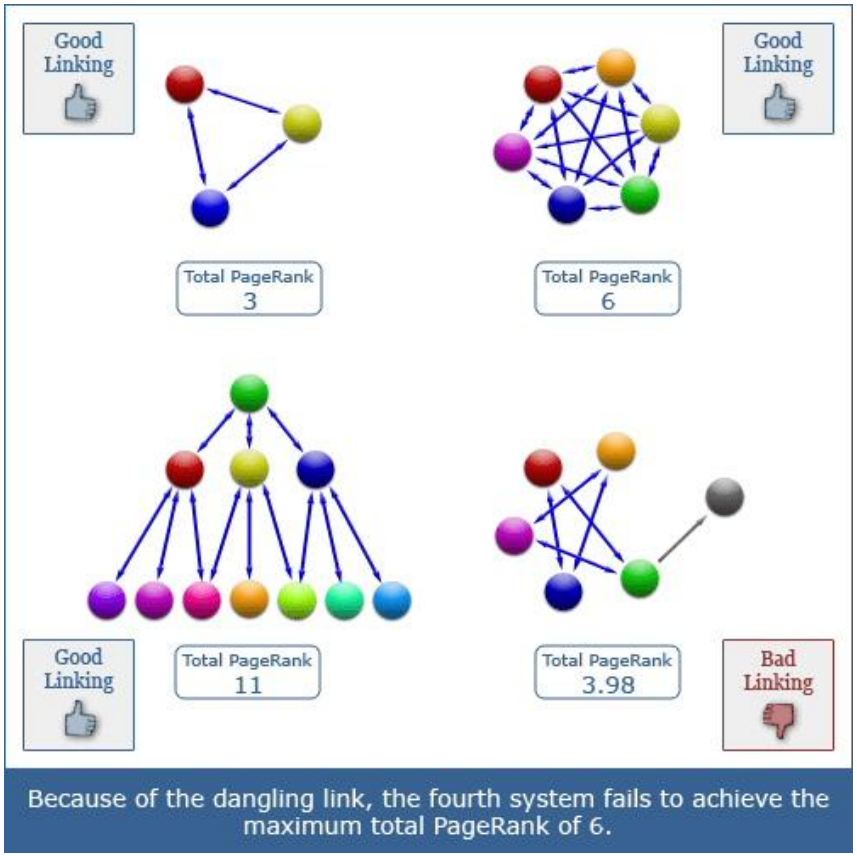
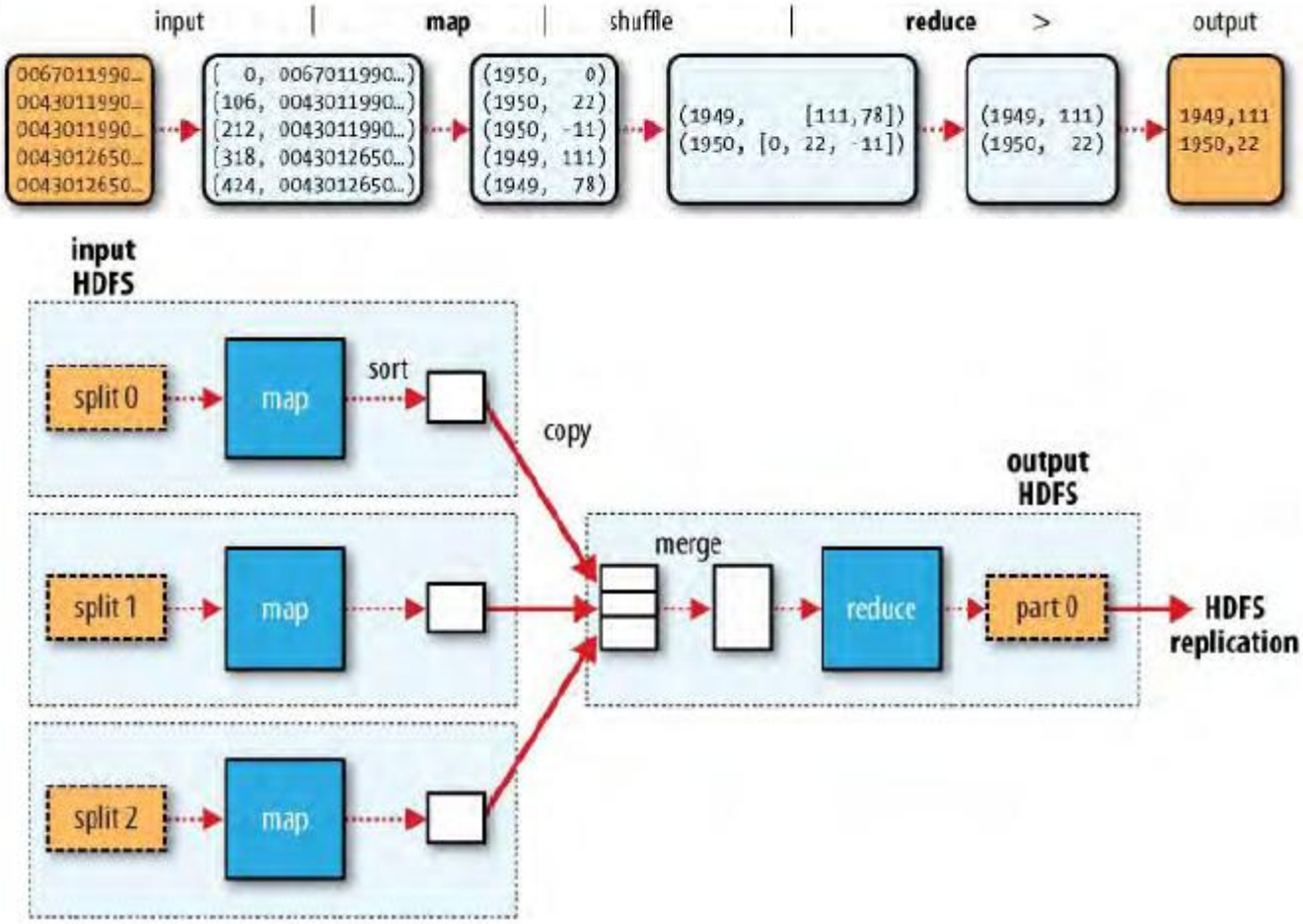


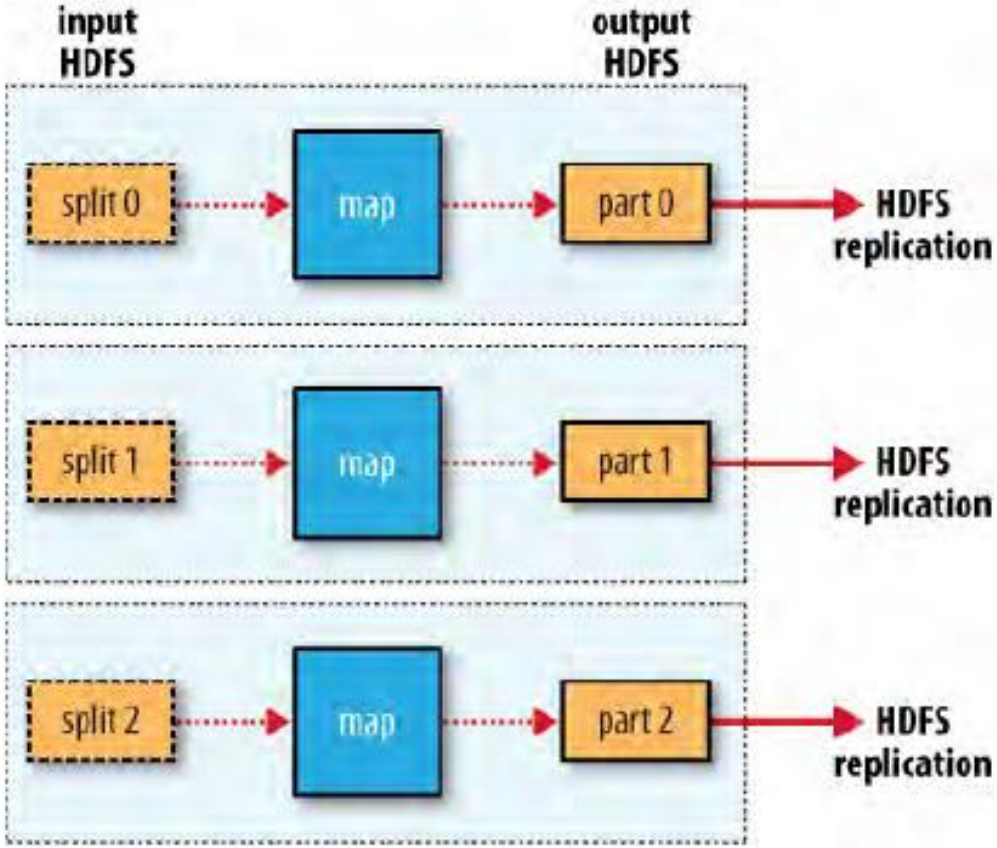
Hadoop数据分析平台 第4周

2012.9.10

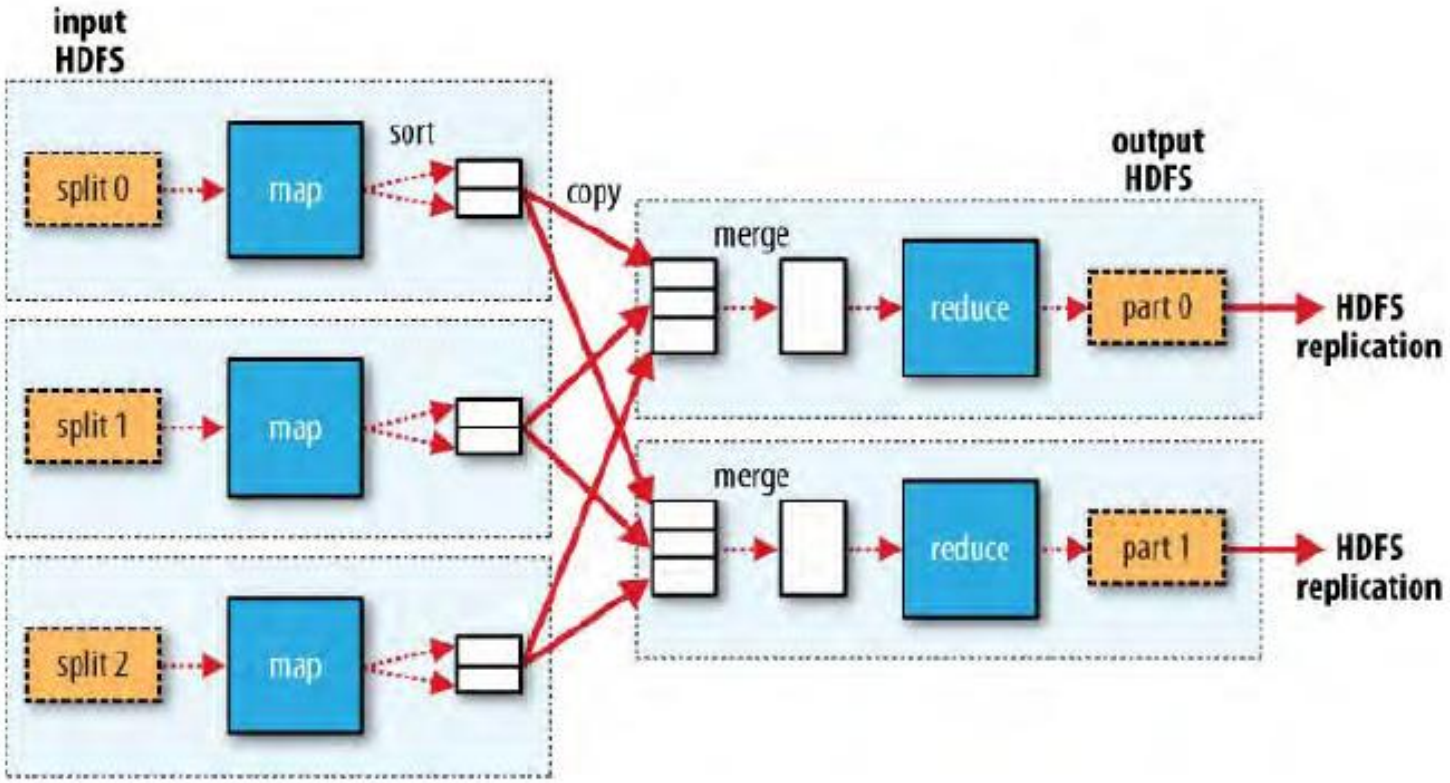
Map-Reduce编程模型



没有reduce的简单编程模型



复杂的编程模型


input
HDFS
split 0
map
sort
split 1
map
split 2
map
copy
output
HDFS
merge
reduce
part 0
HDFS
replication
merge
reduce
part 1
HDFS
replication

Mapper

- Map-reduce的思想就是“分而治之”
- Mapper负责“分”，即把复杂的任务分解为若干个“简单的任务”执行
- “简单的任务”有几个含义：1 数据或计算规模相对于原任务要大大缩小；2 就近计算，即会被分配到存放了所需数据的节点进行计算；3 这些小任务可以并行计算，彼此间几乎没有依赖关系

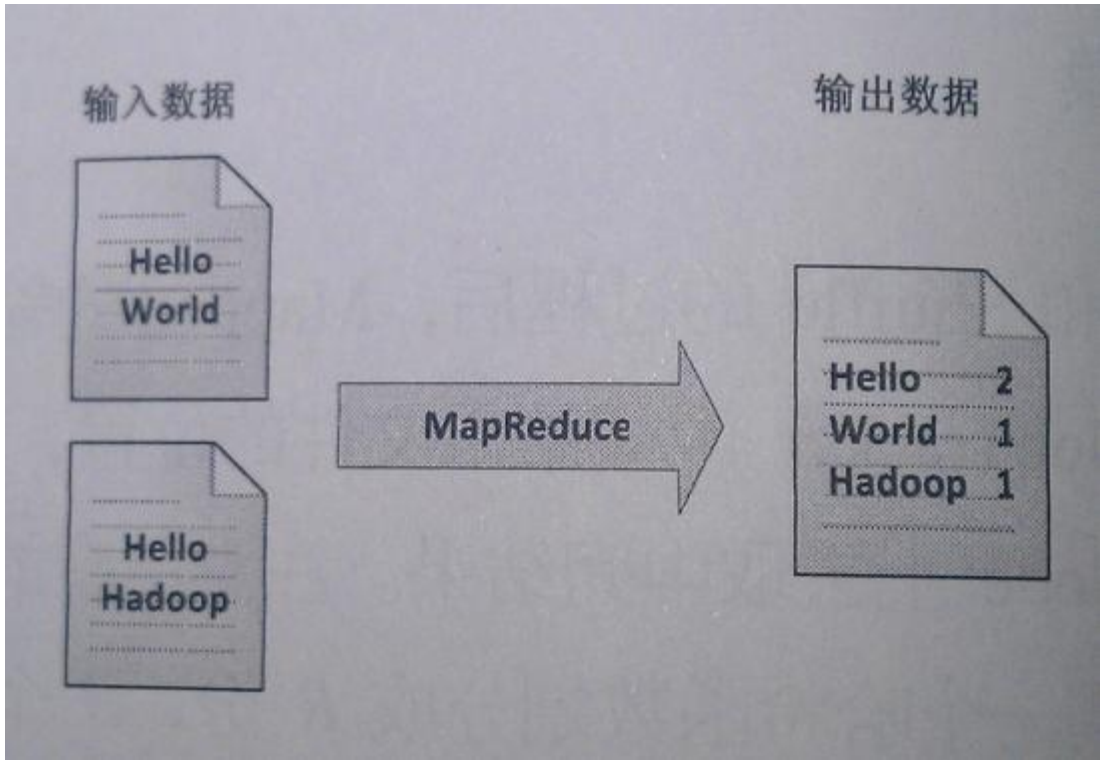
Reducer

- 对map阶段的结果进行汇总
- Reducer的数目由mapred-site.xml配置文件里的项目mapred.reduce.tasks决定。缺省值为1，用户可以覆盖之

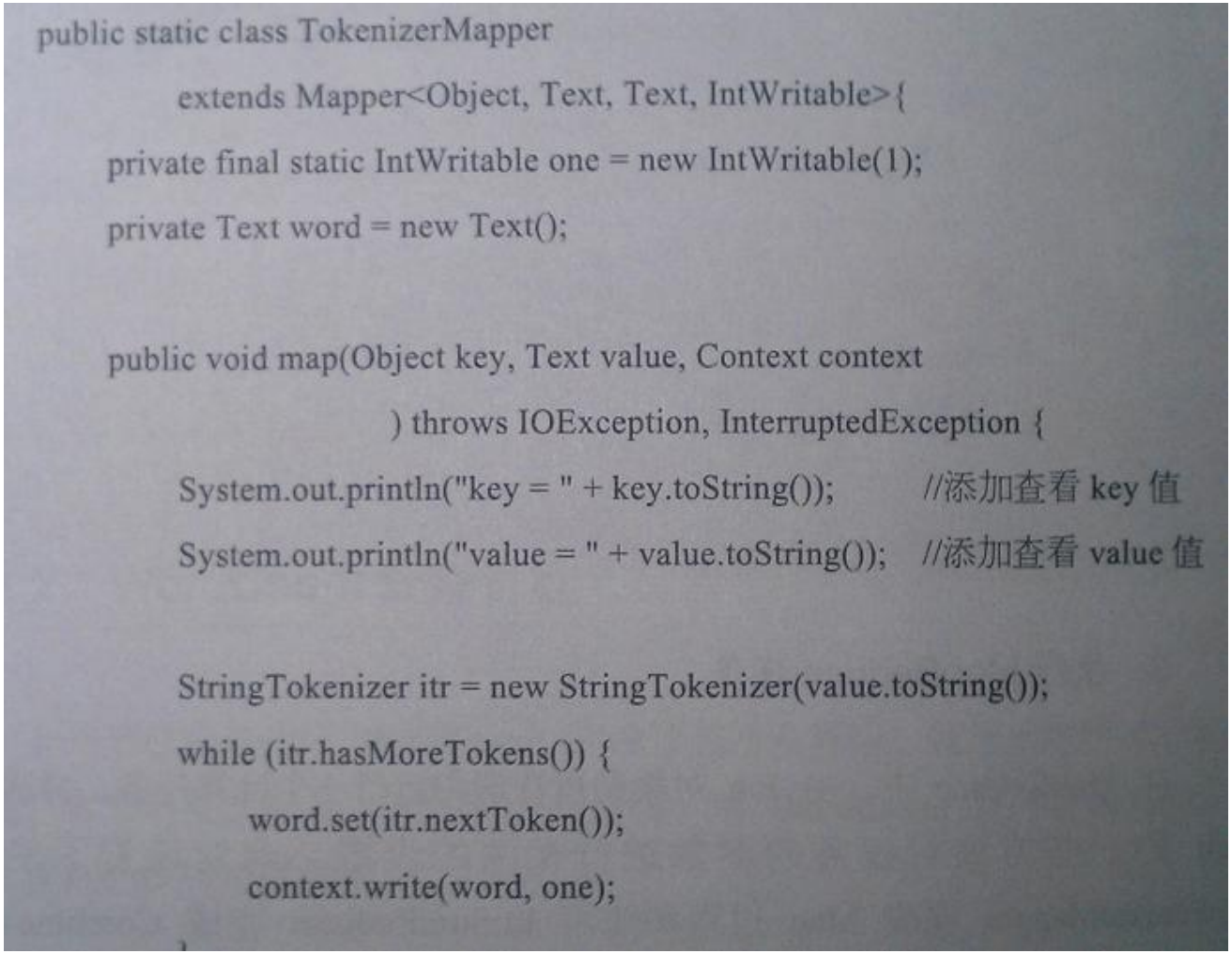
Shuffler

- 在mapper和reducer中间的一个步骤（可以没有）
- 可以把mapper的输出按照某种key值重新切分和组合成n份，把key值符合某种范围的输出送到特定的reducer那里去处理
- 可以简化reducer过程

M-R的现实例子

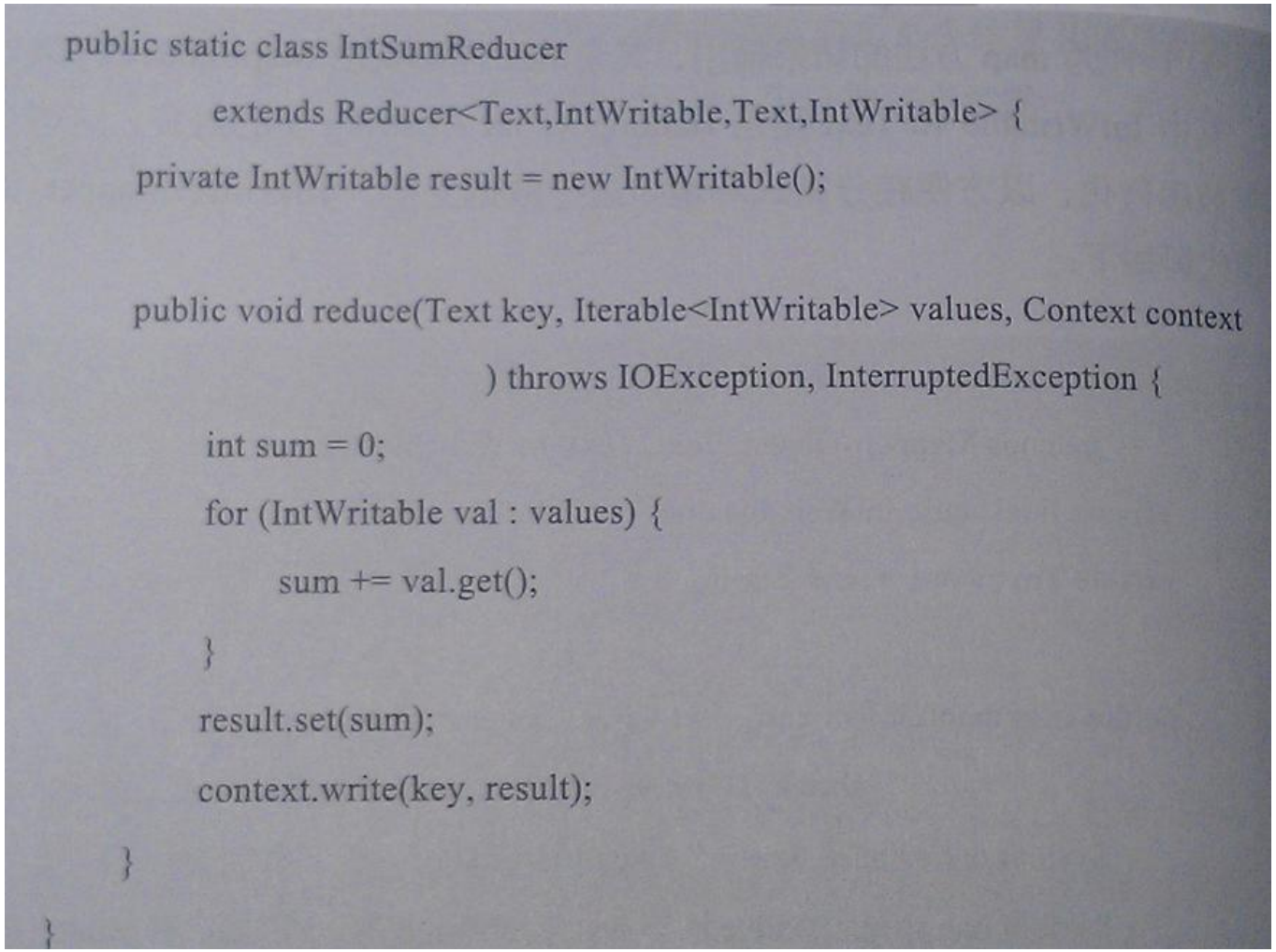

输入数据
Hello
World
Hello
Hadoop
MapReduce
输出数据
Hello 2
World 1
Hadoop 1

例子：mapper



```
public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
        System.out.println("key = " + key.toString());        //添加查看 key 值
        System.out.println("value = " + value.toString());    //添加查看 value 值

        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
```

例子：reducer

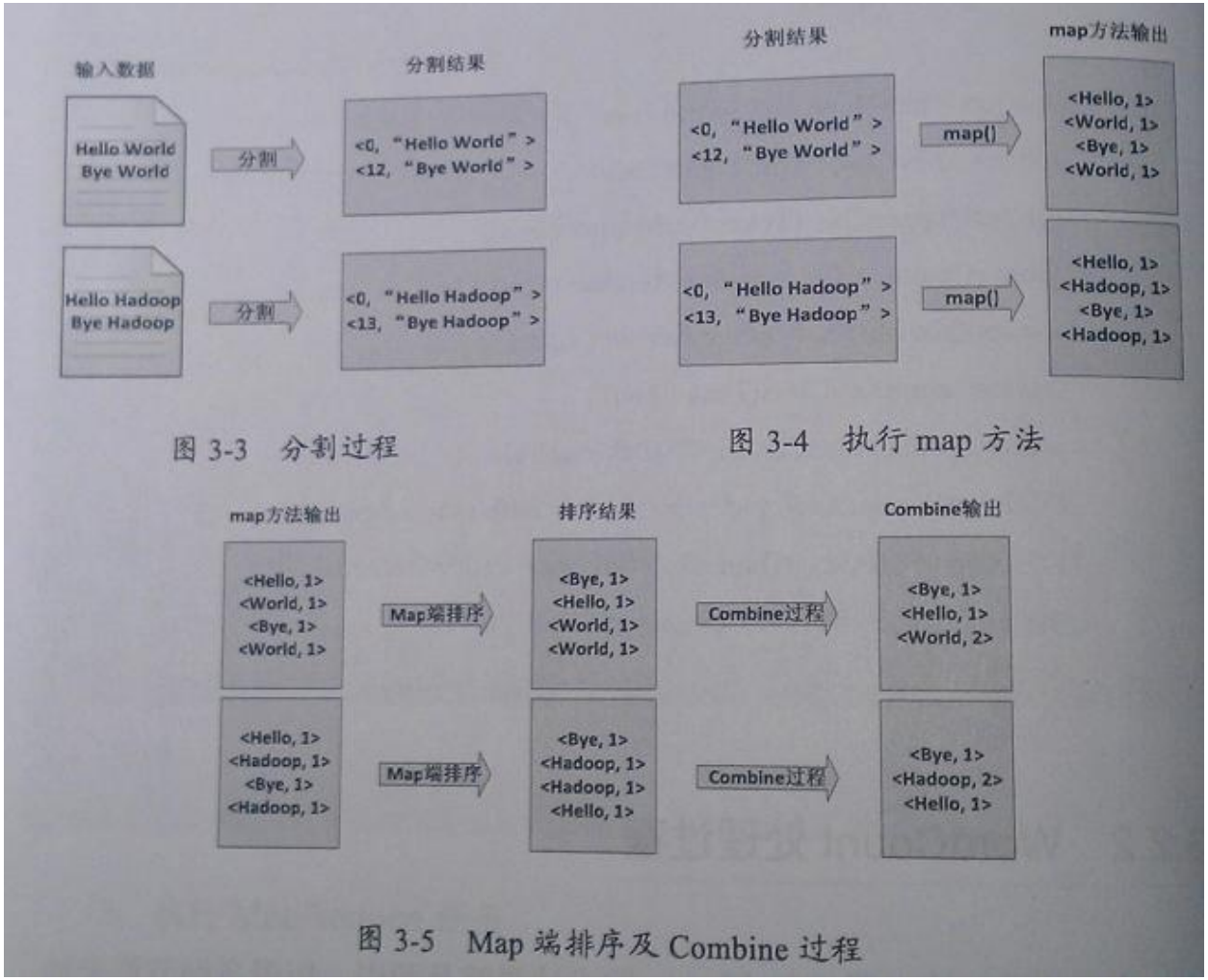


```
public static class IntSumReducer
        extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values, Context context
                       ) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

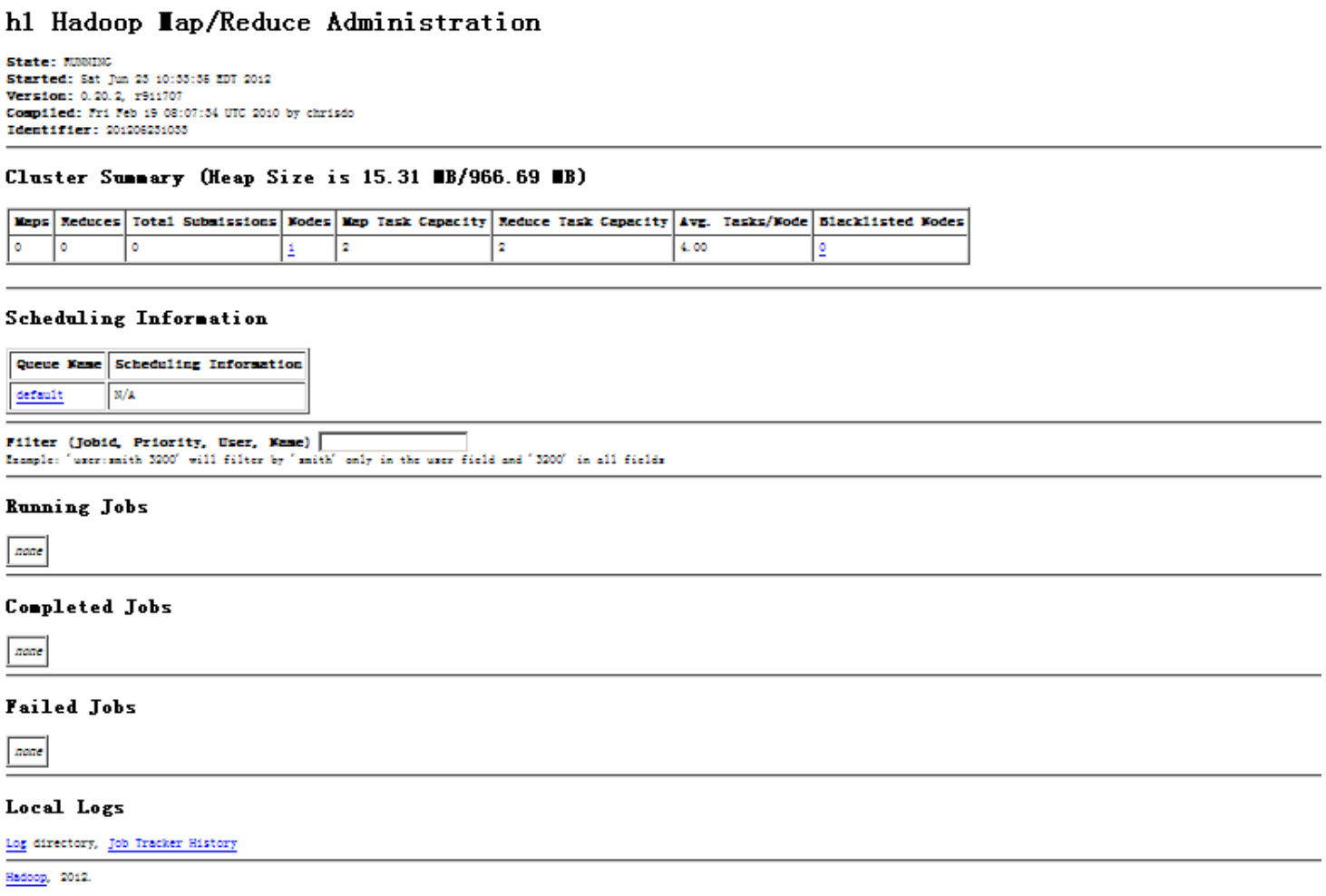
例子：运行mapper和reducer

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
```

例子：流程原理图


输入数据
分割结果
分割结果
map方法输出
Hello World
Bye World
分割
<0, "Hello World" >
<12, "Bye World" >
<0, "Hello World" >
<12, "Bye World" >
map()
<Hello, 1>
<World, 1>
<Bye, 1>
<World, 1>
Hello Hadoop
Bye Hadoop
分割
<0, "Hello Hadoop" >
<13, "Bye Hadoop" >
<0, "Hello Hadoop" >
<13, "Bye Hadoop" >
map()
<Hello, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
图 3-3 分割过程
图 3-4 执行 map 方法
map方法输出
排序结果
Combine输出
<Hello, 1>
<World, 1>
<Bye, 1>
<World, 1>
Map端排序
<Bye, 1>
<Hello, 1>
<World, 1>
<World, 1>
Combine过程
<Bye, 1>
<Hello, 1>
<World, 2>
<Hello, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
Map端排序
<Bye, 1>
<Hadoop, 1>
<Hadoop, 1>
<Hello, 1>
Combine过程
<Bye, 1>
<Hadoop, 2>
<Hello, 1>
图 3-5 Map 端排序及 Combine 过程

网络界面：http://192.168.1.102:50030/jobtracker.jsp



h1 Hadoop Map/Reduce Administration

State: RUNNING
Started: Sat Jun 23 10:55:56 EDT 2012
Version: 0.20.2, r911707
Compiled: Fri Feb 19 08:07:34 UTC 2010 by chrisdo
Identifier: 201206231055

Cluster Summary (Heap Size is 15.31 MB/966.69 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
0	0	0	1	2	2	4.00	0

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Completed Jobs

none

Failed Jobs

none

Local Logs

Log directory, Job Tracker History

Hadoop, 2012.

作业页面

Hadoop job_201206231033_0002 on h1

User: grid
Job Name: word count
Job File: hdfs://h1:9000/tmp/hadoop-grid/mapred/system/job_201206231033_0002/job.xml
Job Setup: Successful
Status: Running
Started at: Sat Jun 23 10:37:41 EDT 2012
Running for: 12sec
Job Cleanup: Pending

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	2	0	0	2	0	0 / 0
reduce	0.00%	1	0	1	0	0	0 / 0

	Counter	Map	Reduce	Total
Job Counters	Launched reduce tasks	0	0	1
	Launched map tasks	0	0	2
	Data-local map tasks	0	0	2
FileSystemCounters	HDFS_BYTES_READ	25	0	25
	FILE_BYTES_WRITTEN	125	0	125
Map-Reduce Framework	Combine output records	4	0	4
	Map input records	2	0	2
	Spilled Records	4	0	4
	Map output bytes	41	0	41
	Map output records	4	0	4
	Combine input records	4	0	4

2012.9.10

作业页面

Map Completion Graph - close

Reduce Completion Graph - close

Go back to JobTracker

Hadoop, 2012.

2012.9.10

任务页面

Hadoop map task list for job_201206231033_0002 on h1

All Tasks

Task	Complete	Status	Start Time	Finish Time	Errors	Counters
task_201206231033_0002_m_000000	100.00%		23-Jun-2012 10:37:46	23-Jun-2012 10:37:52 (6sec)		8
task_201206231033_0002_m_000001	100.00%		23-Jun-2012 10:37:46	23-Jun-2012 10:37:52 (6sec)		8

Go back to JobTracker

Hadoop, 2012.

Hadoop reduce task list for job_201206231033_0002 on h1

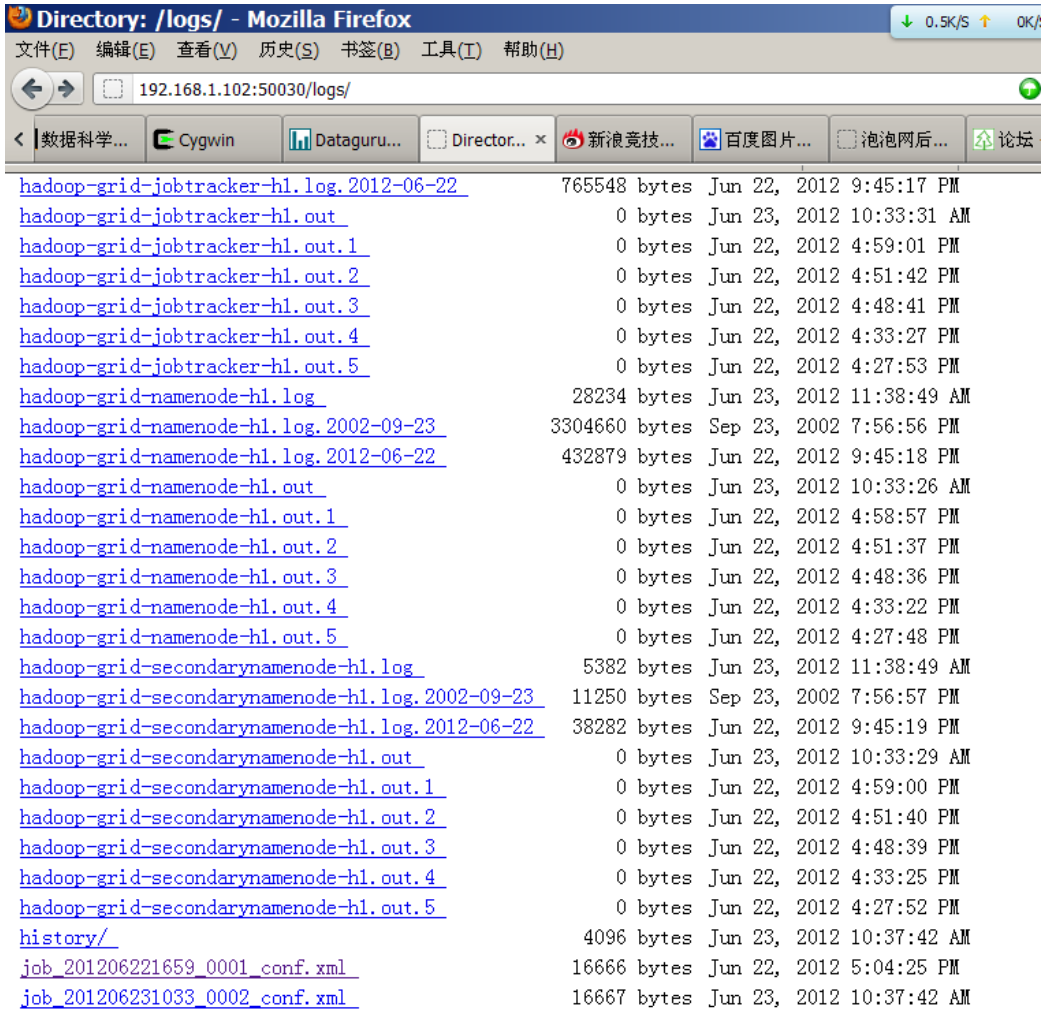
All Tasks

Task	Complete	Status	Start Time	Finish Time	Errors	Counters
task_201206231033_0002_r_000000	100.00%	reduce > reduce	23-Jun-2012 10:37:52	23-Jun-2012 10:38:04 (12sec)		10

Go back to JobTracker

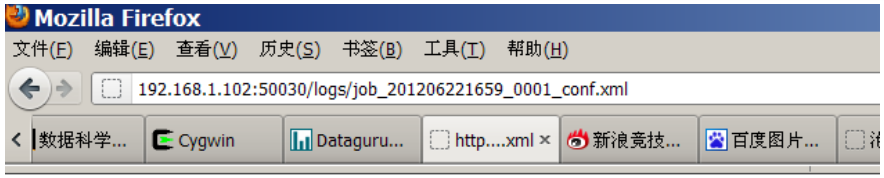
Hadoop, 2012.

观看参数列表



2012.9.10

观看参数列表



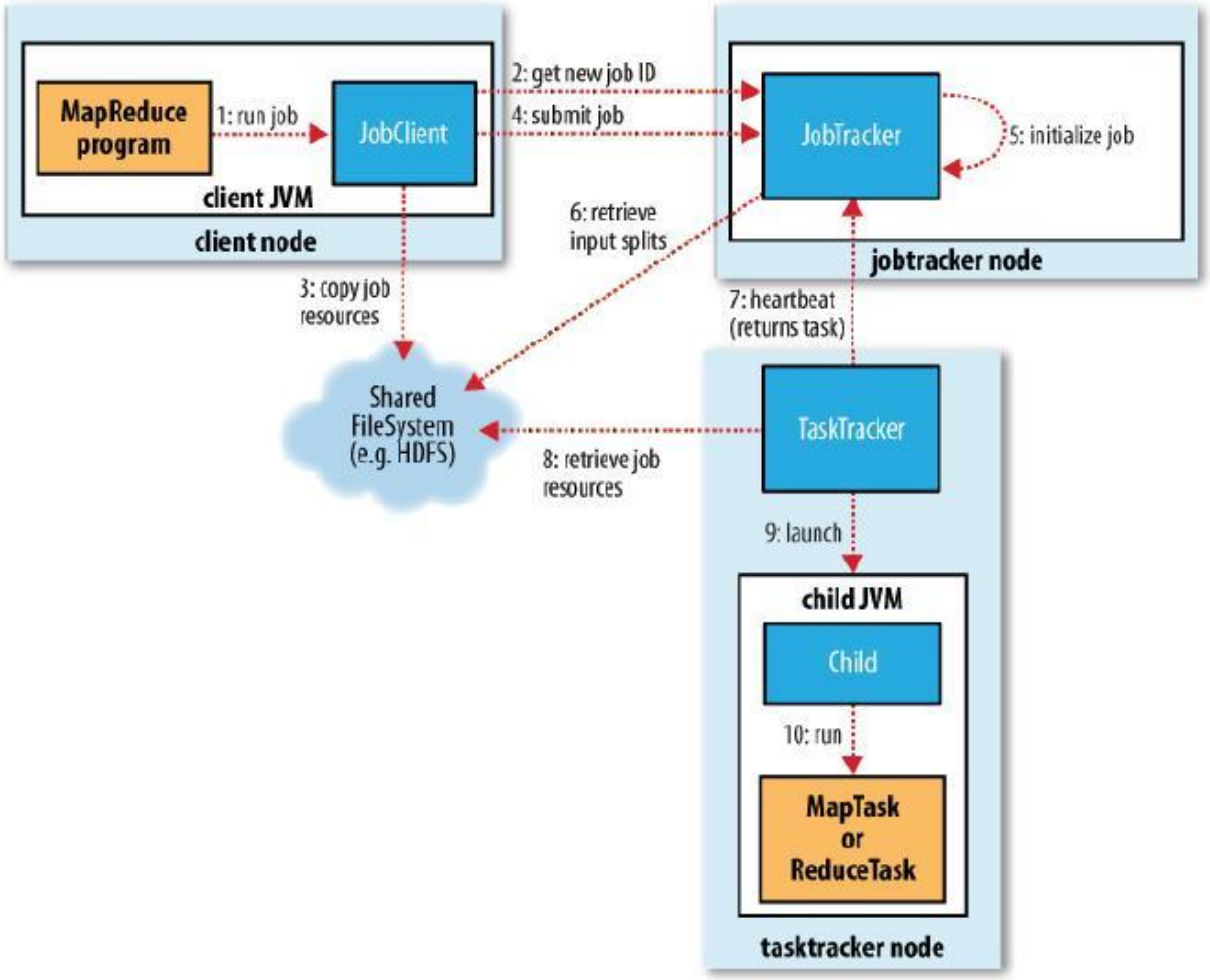

```
</property>
-<property>
  <name>dfs.namenode.logging.level</name>
  <value>info</value>
</property>
-<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:50010</value>
</property>
-<property>
  <name>io.skip.checksum.errors</name>
  <value>false</value>
</property>
-<property>
  <name>fs.default.name</name>
  <value>hdfs://h1:9000</value>
</property>
-<property>
  <name>mapred.reducer.new-api</name>
  <value>true</value>
</property>
-<property>
  <name>mapred.child.tmp</name>
  <value>./tmp</value>
</property>
-<property>
  <name>fs.har.impl.disable.cache</name>
  <value>true</value>
</property>
-<property>
  <name>dfs.safemode.threshold.pct</name>
  <value>0.999f</value>
</property>
```

2012.9.10

性能调优

- 究竟需要多少个reducer？
- 输入：大文件优于小文件
- 减少网络传输：压缩map的输出
- 优化每个节点能运行的任务数：mapred.tasktracker.map.tasks.maximum和mapred.tasktracker.reduce.tasks.maximum （缺省值均为2）

Map-Reduce工作机制剖析



调度机制

- 缺省为先入先出作业队列调度
- 支持公平调度器
- 支持容量调度器

任务执行优化

- 推测式执行：即如果jobtracker发现有拖后腿的任务，会再启动一个相同的备份任务，然后哪个先执行完就会kill去另外一个。因此在监控网页上经常能看到正常执行完的作业有被kill掉的任务
- 推测式执行缺省打开，但如果是代码问题，并不能解决问题，而且会使集群更慢，通过在mapred-site.xml配置文件中设置mapred.map.tasks.speculative.execution和mapred.reduce.tasks.speculative.execution可为map任务或reduce任务开启或关闭推测式执行
- 重用JVM，可以省去启动新的JVM消耗的时间，在mapred-site.xml配置文件中设置mapred.job.reuse.jvm.num.tasks设置单个JVM上运行的最大任务数（1，>1或-1表示没有限制）
- 忽略模式，任务在读取数据失败2次后，会把数据位置告诉jobtracker，后者重新启动该任务并且在遇到所记录的坏数据时直接跳过（缺省关闭，用SkipBadRecord方法打开）

错误处理机制：硬件故障

- 硬件故障是指jobtracker故障或tasktracker故障
- jobtracker是单点，若发生故障目前hadoop还无法处理，唯有选择最牢靠的硬件作为jobtracker
- Jobtracker通过心跳（周期1分钟）信号了解tasktracker是否发生故障或负载过于严重
- Jobtracker将从任务节点列表中移除发生故障的tasktracker
- 如果故障节点在执行map任务并且尚未完成，jobtracker会要求其它节点重新执行此map任务
- 如果故障节点在执行reduce任务并且尚未完成，jobtracker会要求其它节点继续执行尚未完成的reduce任务

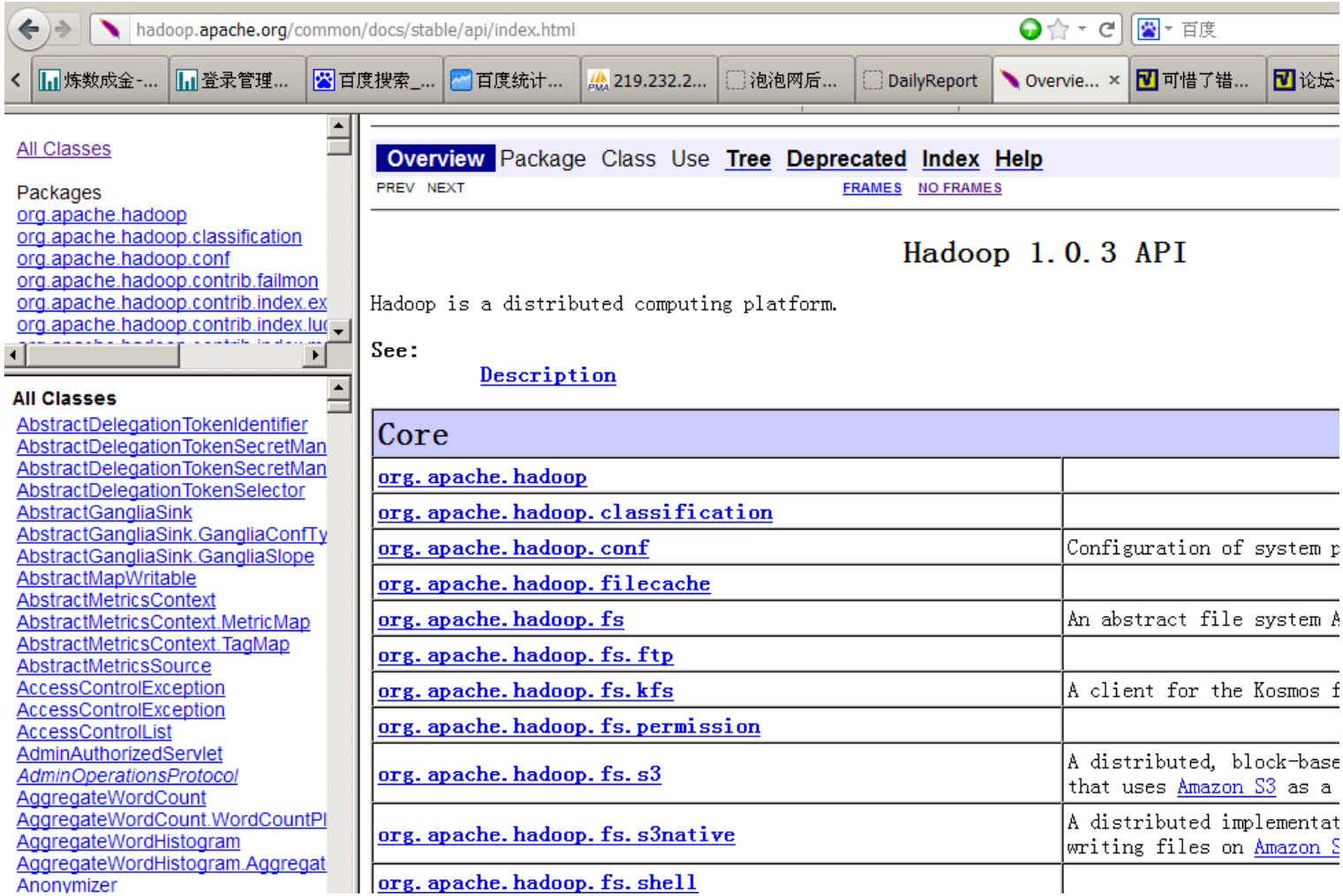
错误处理机制：任务失败

- 由于代码缺陷或进程崩溃引起任务失败
- Jvm自动退出，向tasktracker父进程发送方错误信息，错误信息也会写入到日志
- Tasktracker监听程序会发现进程退出，或进程很久没有更新信息送回，将任务标记为失败
- 标记失败任务后，任务计数器减去1以便接受新任务，并通过心跳信号告诉jobtracker任务失败的信息
- Jobtrack获悉任务失败后，将把该任务重新放入调度队列，重新分配出去再执行
- 如果一个任务失败超过4次（可以设置），将不会再被执行，同时作业也宣布失败

Hadoop命令大全

2012.9.10

Hadoop API大全



2012.9.10

审计日志

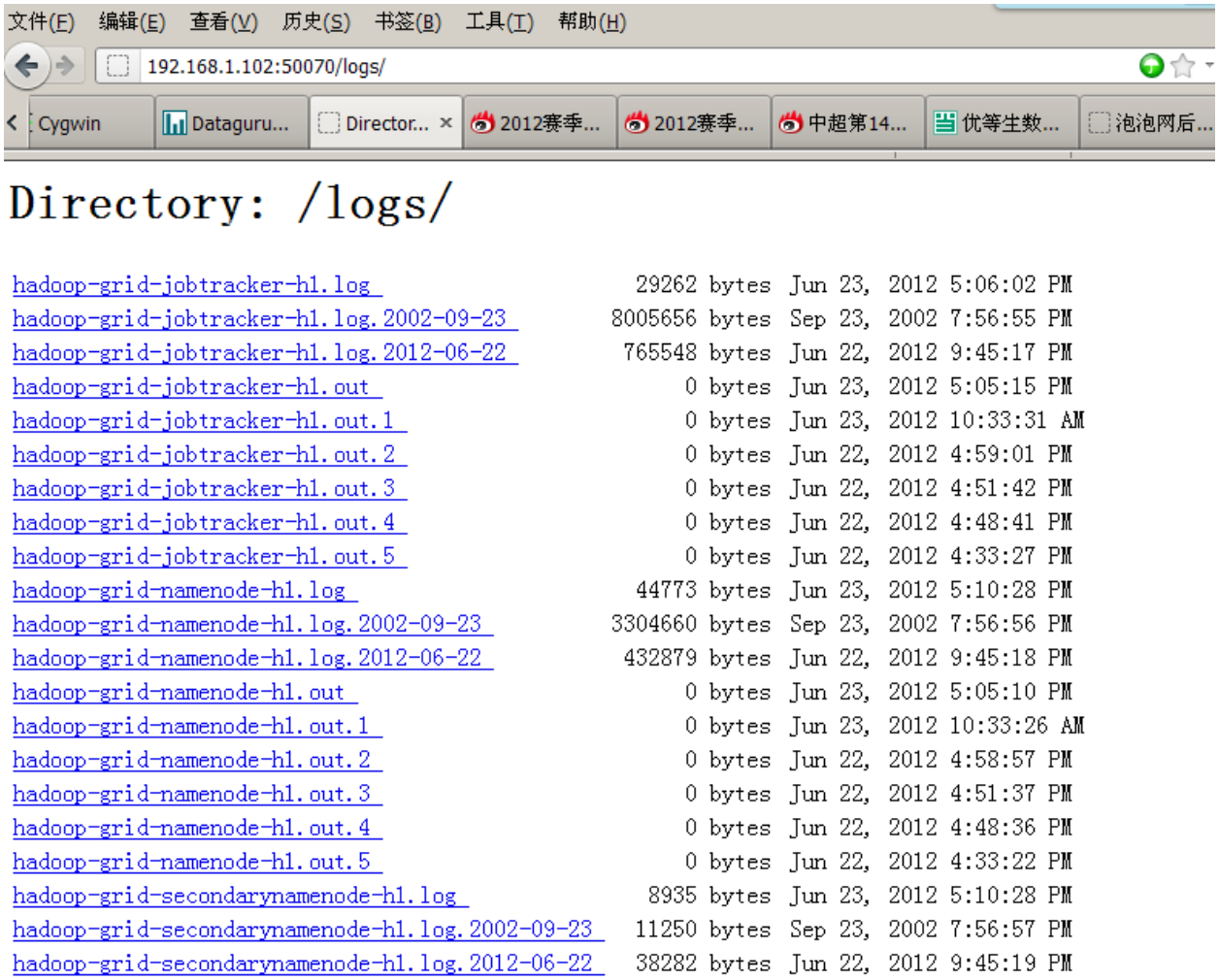
把log4j.properties配置文件中的

log4j.logger.org.apache.hadoop.fs.FSNamesystem.audit=WARN

改为“INFO”可以打开审计日志。每个HDFS事件都会在namenode的log中写入一行记录

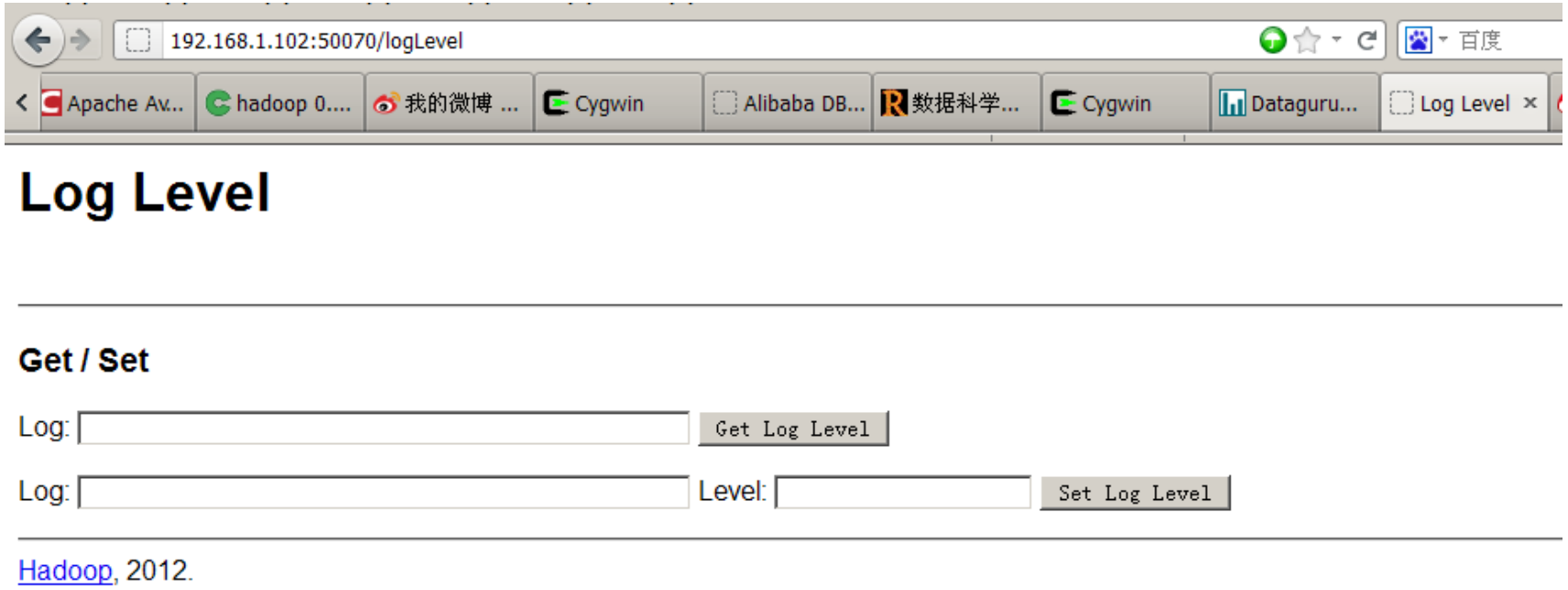
```
#log4j.appender.RFA.MaxBackupIndex=30

#log4j.appender.RFA.layout=org.apache.log4j.PatternLayout
#log4j.appender.RFA.layout.ConversionPattern=%d{ISO8601} %-5p %c{2} - %m%r
#log4j.appender.RFA.layout.ConversionPattern=%d{ISO8601} %-5p %c{2} (%F:%M

#
# FSNamesystem Audit logging
# All audit events are logged at INFO level
#
log4j.logger.org.apache.hadoop.fs.FSNamesystem.audit=WARN

# Custom Logging levels

#log4j.logger.org.apache.hadoop.mapred.JobTracker=DEBUG
#log4j.logger.org.apache.hadoop.mapred.TaskTracker=DEBUG
#log4j.logger.org.apache.hadoop.fs.FSNamesystem=DEBUG

# Jets3t library
log4j.logger.org.jets3t.service.impl.rest.httpclient.RestS3Service=ERROR
```

Namenode日志


文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)
192.168.1.102:50070/logs/
Cygwin
Dataguru...
Director...
2012赛季...
2012赛季...
中超第14...
优等生数...
泡泡网后...
Directory: /logs/
hadoop-grid-jobtracker-h1.log 29262 bytes Jun 23, 2012 5:06:02 PM
hadoop-grid-jobtracker-h1.log.2002-09-23 8005656 bytes Sep 23, 2002 7:56:55 PM
hadoop-grid-jobtracker-h1.log.2012-06-22 765548 bytes Jun 22, 2012 9:45:17 PM
hadoop-grid-jobtracker-h1.out 0 bytes Jun 23, 2012 5:05:15 PM
hadoop-grid-jobtracker-h1.out.1 0 bytes Jun 23, 2012 10:33:31 AM
hadoop-grid-jobtracker-h1.out.2 0 bytes Jun 22, 2012 4:59:01 PM
hadoop-grid-jobtracker-h1.out.3 0 bytes Jun 22, 2012 4:51:42 PM
hadoop-grid-jobtracker-h1.out.4 0 bytes Jun 22, 2012 4:48:41 PM
hadoop-grid-jobtracker-h1.out.5 0 bytes Jun 22, 2012 4:33:27 PM
hadoop-grid-namenode-h1.log 44773 bytes Jun 23, 2012 5:10:28 PM
hadoop-grid-namenode-h1.log.2002-09-23 3304660 bytes Sep 23, 2002 7:56:56 PM
hadoop-grid-namenode-h1.log.2012-06-22 432879 bytes Jun 22, 2012 9:45:18 PM
hadoop-grid-namenode-h1.out 0 bytes Jun 23, 2012 5:05:10 PM
hadoop-grid-namenode-h1.out.1 0 bytes Jun 23, 2012 10:33:26 AM
hadoop-grid-namenode-h1.out.2 0 bytes Jun 22, 2012 4:58:57 PM
hadoop-grid-namenode-h1.out.3 0 bytes Jun 22, 2012 4:51:37 PM
hadoop-grid-namenode-h1.out.4 0 bytes Jun 22, 2012 4:48:36 PM
hadoop-grid-namenode-h1.out.5 0 bytes Jun 22, 2012 4:33:22 PM
hadoop-grid-secondarynamenode-h1.log 8935 bytes Jun 23, 2012 5:10:28 PM
hadoop-grid-secondarynamenode-h1.log.2002-09-23 11250 bytes Sep 23, 2002 7:56:57 PM
hadoop-grid-secondarynamenode-h1.log.2012-06-22 38282 bytes Jun 22, 2012 9:45:19 PM

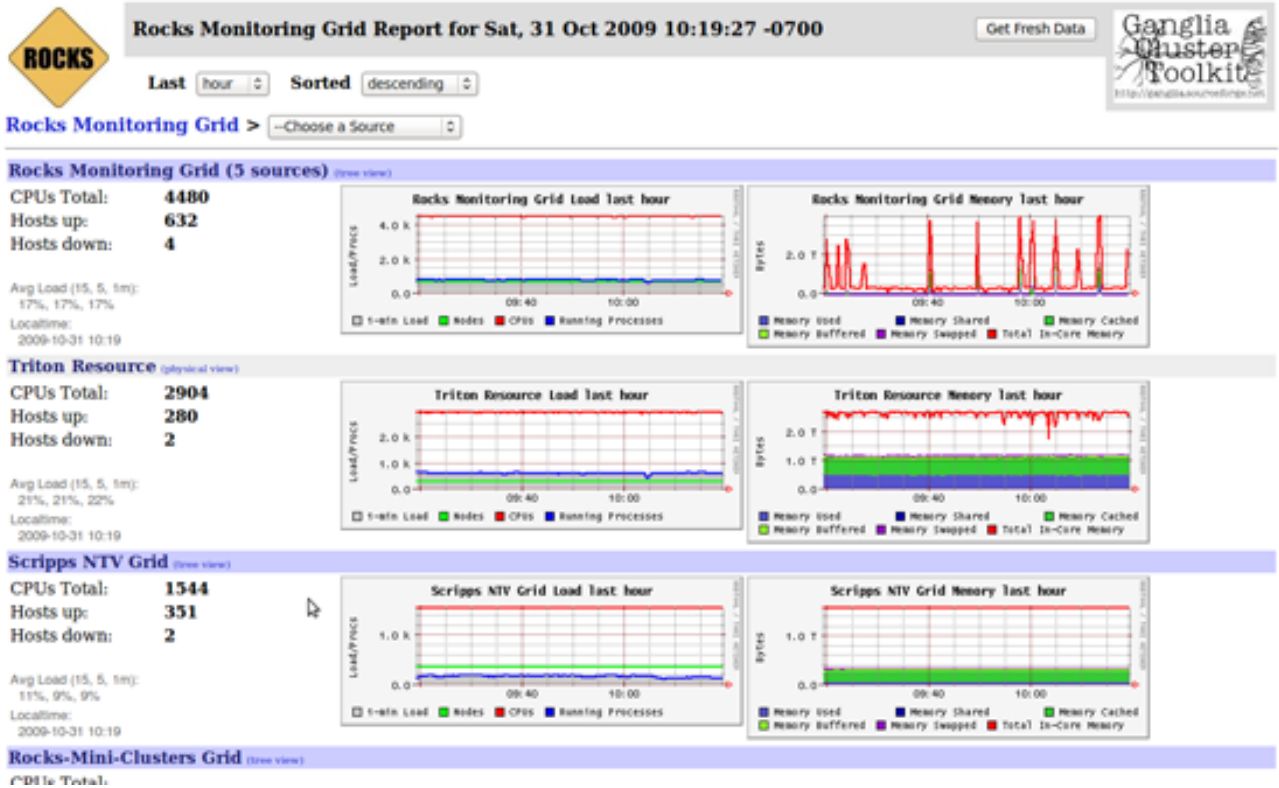
监控日志

- 调整log4j日志级别：在监控网页的url后 加上/logLevel



第三方工具

- Ganglia
- Chukwa
- Openstack



Thanks

FAQ时间