Hadoop Tutorial
- 설치 및 실행

2008. 7. 17

한 재 선 (NexR 대표이사)
jshan0000@gmail.com
http://www.web2hub.com
H.P: 016-405-5469

NexR
TOWARD OPEN PLATFORM

Hadoop 소개

- **Brief History**
  - 2005년 Doug Cutting(Lucene & Nutch 개발자)에 의해 시작
    - Nutch 오픈소스 검색엔진의 분산확장 이슈에서 출발
  - 2006년 Yahoo의 전폭적인 지원 (Doug Cutting과 전담팀 고용)
  - 2008년 Apache Top-level Project로 승격
  - 현재(2008년4월) 0.16.3 release

- **Hadoop**
  - Java 언어 기반
  - Apache 라이선스
  - 많은 컴포넌트들
    - HDFS, HBase, MapReduce, Hadoop On Demand(HOD), Streaming, HQL, Hama, Mahout, etc

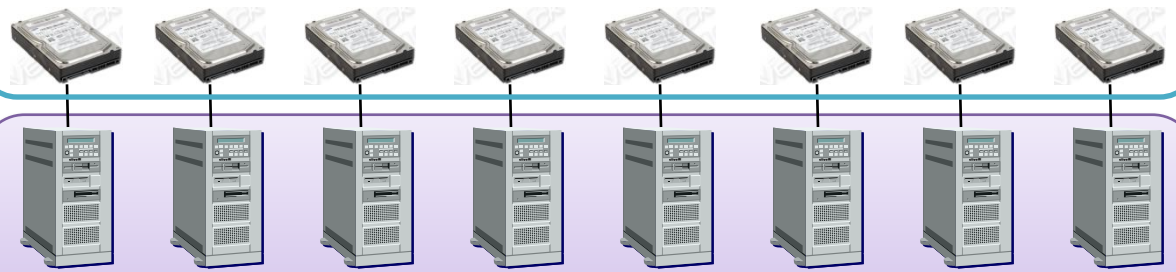
Hadoop 구조

Nutch: Open Source Search Engine → Google Search

MapReduce: 분산 데이터 처리 시스템 → MapReduce

HBase: 분산 데이터베이스 → Bigtable

HDFS: 분산 파일 시스템 → GFS



Commodity PC 서버 클러스터

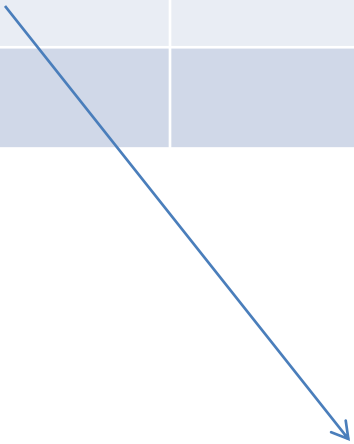
Hadoop Versions

Version	Release
0.16.3	Release 2008.4.16
0.16.4	Release 2008.5.5
0.17.0	Release 2008.5.20
0.17.1	Release 2008.6.23
0.17.2	Not released yet
0.18.0	Not released yet
0.19.0	Not released yet

Current stable version

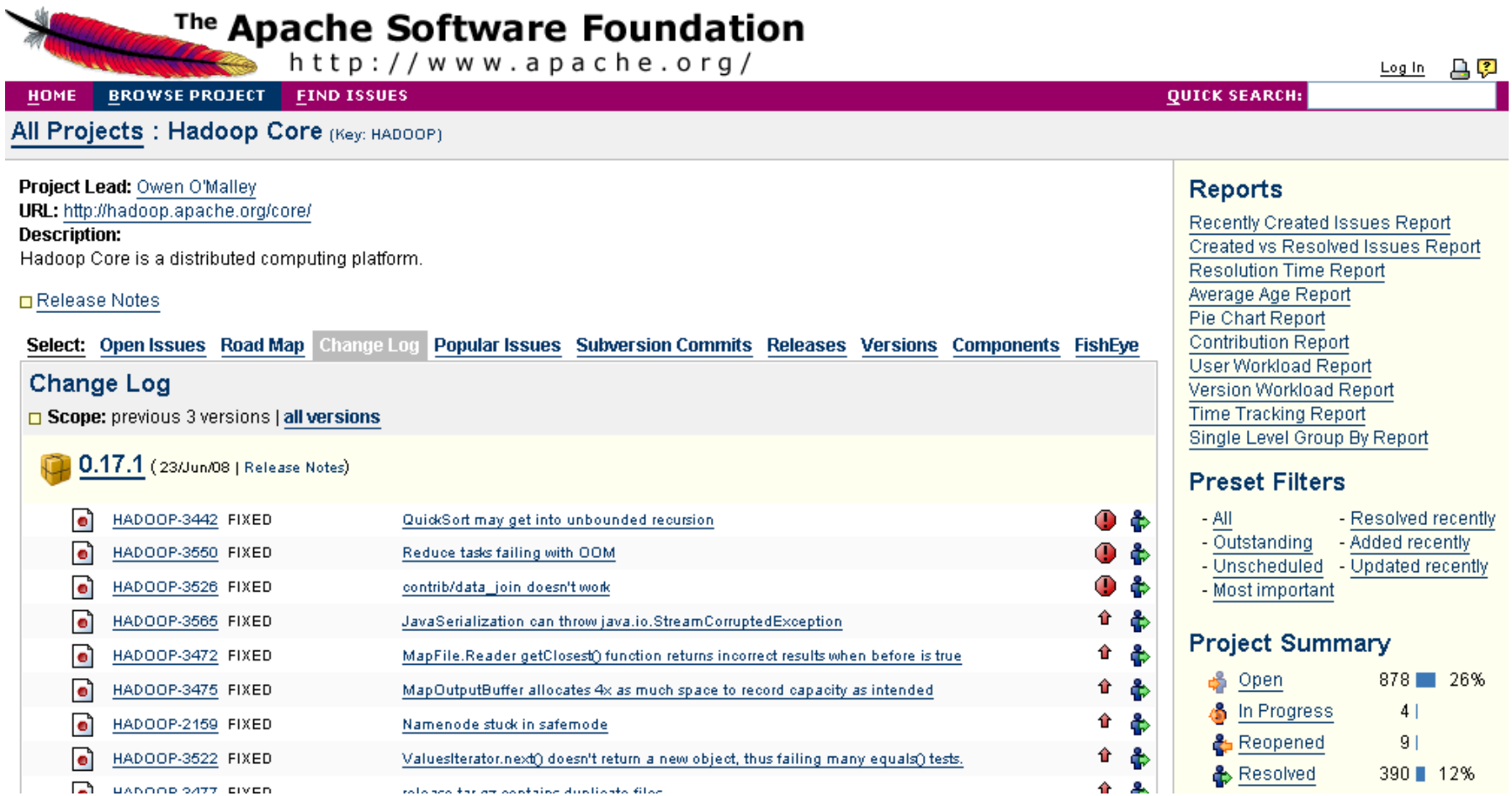
HBase Versions

Version	Release
0.1.0	Release 2008.3.27
0.1.1	Release 2008.4.11
0.1.2	Release 2008.5.13
0.1.3	Release 2008.6.27
0.2	Not released yet



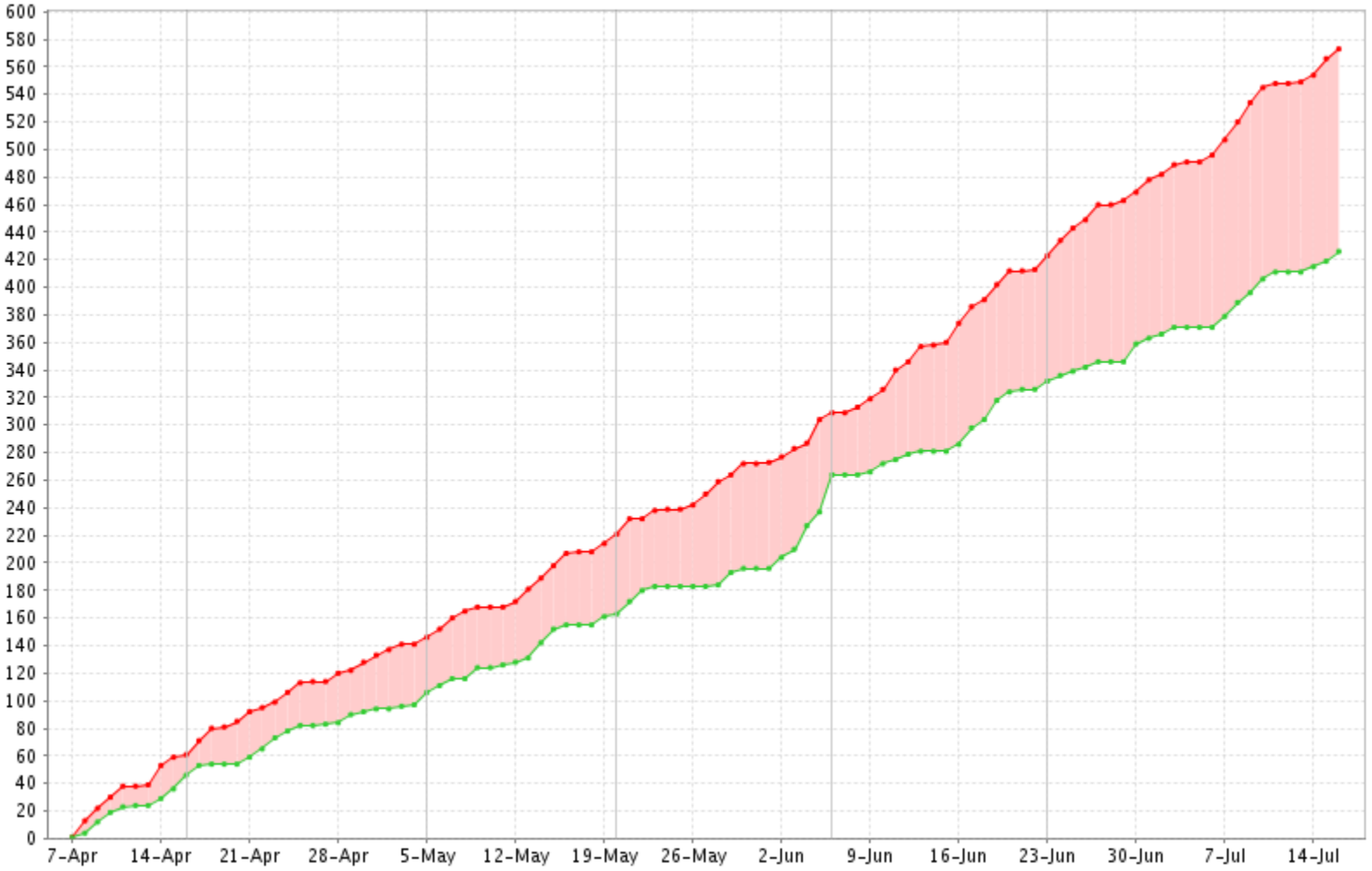
Current stable version

Hadoop Project Issue Tracking



http://issues.apache.org/jira/browse/HADOOP

Hadoop Project 상황



지난 100일간 생성된 이슈와 해결된 이슈들 누적분포

NEXR
TOWARD OPEN PLATFORM

Hadoop 설치

1. hadoop-0.17.1.tar.gz 다운 및 압축해제
2. conf/hadoop-env.sh 편집
export JAVA_HOME=/usr/java/jdk1.6.0_03
3. conf/hadoop-site.xml 편집 (conf/hadoop-default.xml 에서 필요한 내용 가져와 편집)

```
<property>
<name>hadoop.tmp.dir</name>
<value>/home/${user.name}/tmp/hadoop-0.17.1-${user.name}</value>
<description>A base for other temporary directories.</description>
</property>

<property>
<name>fs.default.name</name>
<value>hdfs://192.168.1.2:9000/</value>
</property>

<property>
<name>mapred.job.tracker</name>
<value>192.168.1.2:9001</value>
</property>

<property>
<name>dfs.replication</name>
<value>3</value>
<!-- set to 1 to reduce warnings when running on a single node -->
</property>
```

Hadoop 설치

4. conf/masters 편집
192.168.1.2
5. conf/slaves편집
192.168.1.3
192.168.1.4
192.168.1.5
6. ssh pub-key 등록
$ ssh-keygen
$ ssh-copy-id -i ~/.ssh/id_rsa.pub id@server // 접속하려는 서버의 id/주소
7. HDFS format
$ bin/hadoop namenode –format
8. 실행
$ bin/start-all.sh
9. 정지
$ bin/stop-all.sh

% HADOOP_HOME 경로는 master와 slave 모두 동일하게 하자.
master에서 실행시 slave들과 rsync를 통해 sync 맞춤
% 문제 있는 경우 iptable 설정 확인
iptable 설정을 제거하거나 hadoop이 쓰는 포트들을 등록

Hadoop 실행 - DFS

```
$ bin/hadoop dfs
Usage: java FsShell
        [-ls <path>]
        [-lsr <path>]
        [-du <path>]
        [-dus <path>]
        [-count <path>]
        [-mv <src> <dst>]
        [-cp <src> <dst>]
        [-rm <path>]
        [-rmr <path>]
        [-expunge]
        [-put <localsrc> ... <dst>]
        [-copyFromLocal <localsrc> ... <dst>]
        [-moveFromLocal <localsrc> ... <dst>]
        [-get [-ignoreCrc] [-crc] <src> <localdst>]
        [-getmerge <src> <localdst> [addnl]]
        [-cat <src>]
        [-text <src>]
        [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>]
        [-moveToLocal [-crc] <src> <localdst>]
        [-mkdir <path>]
        [-setrep [-R] [-w] <rep> <path/file>]
        [-touchz <path>]
        [-test -[ezd] <path>]
        [-stat [format] <path>]
        [-tail [-f] <file>]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-chgrp [-R] GROUP PATH...]
        [-help [cmd]]
```

Hadoop 실행 - MapReduce

1. 작성한 MapReduce class들을 jar 파일로 묶기

```
$ jar -cvf wordcount.jar -C wordcount_classes/ .
```

2. DFS에 input 디렉토리 생성 및 input file들 복사

```
$ bin/hadoop dfs –mkdir wordcount/input
$ bin/hadoop dfs –ls wordcount
  /user/jshan/wordcount/input     <dir>
$ bin/hadoop dfs –put file01.txt wordcount/input     // file01.txt는 로컬파일
$ bin/hadoop dfs –put file02.txt wordcount/input     // file02.txt는 로컬파일
$ bin/hadoop dfs –ls wordcount/input
  /user/jshan/wordcount/input/file01 .txt     <r 1>
  /user/jshan/wordcount/input/file02 .txt     <r 1>
$ bin/hadoop dfs –cat wordcount/input/file01.txt
  Hello World Bye World
$ bin/hadoop dfs –cat wordcount/input/file02.txt
  Hello Hadoop Goodbye Hadoop
```

3. MapReduce 실행

```
$ bin/hadoop jar wordcount.jar org.myorg.WordCount /user/jshan/wordcount/input
  /user/jshan/wordcount/output
$ bin/hadoop dfs -cat /user/jshan/wordcount/output/part-00000
  Bye 1
  Goodbye 1
  Hadoop 2
  Hello 2
  World 2
```

Source: http://hadoop.apache.org/core/docs/r0.17.1/mapred_tutorial.html

Hadoop MapReduce 프로그래밍

```
1.  package org.myorg;
2.
3.  import java.io.IOException;
4.  import java.util.*;
5.
6.  import org.apache.hadoop.fs.Path;
7.  import org.apache.hadoop.conf.*;
8.  import org.apache.hadoop.io.*;
9.  import org.apache.hadoop.mapred.*;
10. import org.apache.hadoop.util.*;
11.
12. public class WordCount {
13.
14.   public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
15.     private final static IntWritable one = new IntWritable(1);
16.     private Text word = new Text();
17.
18.     public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
19.       String line = value.toString();
20.       StringTokenizer tokenizer = new StringTokenizer(line);
21.       while (tokenizer.hasMoreTokens()) {
22.         word.set(tokenizer.nextToken());
23.         output.collect(word, one);
24.       }
25.     }
26.   }
```

NEXR TOWARD OPEN PLATFORM

Hadoop MapReduce 프로그래밍

```
28.   public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

29.     public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

30.       int sum = 0;
31.       while (values.hasNext()) {
32.         sum += values.next().get();
33.       }
34.       output.collect(key, new IntWritable(sum));
35.     }
36.   }
37.
38.   public static void main(String[] args) throws Exception {
39.     JobConf conf = new JobConf(WordCount.class);
40.     conf.setJobName("wordcount");
41.
42.     conf.setOutputKeyClass(Text.class);
43.     conf.setOutputValueClass(IntWritable.class);
44.
45.     conf.setMapperClass(Map.class);
46.     conf.setCombinerClass(Reduce.class);
47.     conf.setReducerClass(Reduce.class);
48.
49.     conf.setInputFormat(TextInputFormat.class);
50.     conf.setOutputFormat(TextOutputFormat.class);
51.
52.     FileInputFormat.setInputPaths(conf, new Path(args[0]));
53.     FileOutputFormat.setOutputPath(conf, new Path(args[1]));
54.
55.     JobClient.runJob(conf);
57.   }
58. }
```

Hadoop DFS 관리도구

NameNode '192.168.2.252:9000'

Started:	Sun Jul 13 11:35:12 KST 2008
Version:	0.17.2-dev, r
Compiled:	2008. 07. 07. (i) 16:09:00 KST by jshan
Upgrades:	There are no upgrades in progress.

Browse the filesystem

Cluster Summary

29 files and directories, 11 blocks = 40 total. Heap Size is 16.31 MB / 888.94 MB (1%)

Capacity	:	94.59 GB
DFS Remaining	:	26.1 GB
DFS Used	:	10.23 MB
DFS Used%	:	0.01 %
Live Nodes	:	1
Dead Nodes	:	0

Live Datanodes : 1

Node	Last Contact	Admin State	Size (GB)	Used (%)	Used (%)	Remaining (GB)	Blocks
localhost	2	In Service	94.59	0.01		26.1	11

Hadoop MapReduce 관리도구

192 Hadoop Map/Reduce Administration

State: RUNNING
Started: Sun Jul 13 11:35:19 KST 2008
Version: 0.17.2-dev, r
Compiled: 2008. 07. 07. (i) 16:09:00 KST by jshan
Identifier: 200807131135

Cluster Summary

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
0	1	39	1	2	2	4.00

Running Jobs

Running Jobs								
Jobid	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed
job_200807131135_0039	jshan	Pearson Correlation MR	100.00%	2	2	0.00%	1	0