Only Aggressive Elephants are fast Elephants

Jens Dittrich, Jorge-Arnulfo Quiané-Ruiz, Stefan Richter, Stefan Schuh, Alekh Jindal, Jörg Schad

Saarland University

One Day in Bob´s Life

bob$ head UserVisits.dat

170.122.19.7 | mqt.html | 1994-4-27 | 336.43 | NetAnts/1.2 | [URL] [SOURCE IP] [DURATION]
163.110.31.8 | espdf.html | 2003-7-13 | 176.33 | Mozilla/4.0 (X11; U; Linux x86_64) [URL] [SOURCE IP] [DURATION]
160.103.36.25 | [URL] | 1991-2-14 | 182.95 | [SOURCE IP] [DURATION]
175.101.2.45 | [URL] | 1998-5-4 | 130.32 | Mozilla/4.0 (X11; U; Linux x86_64) [URL] [SOURCE IP] [DURATION]
154.123.46.7 | [URL] | 1992-12-5 | 346.96 | Mozilla/1.2 [URL] [SOURCE IP] [DURATION]
174.118.37.24 | [URL] | 1976-10-11 | 71.28 | Mozilla/1.2 [URL] [SOURCE IP] [DURATION]
173.123.36.22 | [URL] | 1997-4-27 | 15.58 | [URL] [SOURCE IP] [DURATION]
168.114.15.1 | [URL] | 2002-4-18 | 7.930 | Mozilla/4.03 [URL] [SOURCE IP] [DURATION]
158.118.25.1 | [URL] | 1977-5-23 | 160.077 | RSSOwl/1.2.3 [URL] [SOURCE IP] [DURATION]
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;

/**< 
* User: Bob 
* Date: 29.08.12 
* Time: 10:45 
*/
public class BobsMapper extends Mapper<IntWritable, Text, Text, Text> { 
    @Override 
    public void map(IntWritable key, Text value, Context context) 
    throws IOException, InterruptedException 
    { 
        // implement me! 
    } 
}
30 minutes later...

Uploading UserVisits.dat (2.6 TB) to HDFS...

Stop

Uploading 2.6 TB of UserVisits.dat completed
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;

/**
 * User: Bob
 * Date: 29.08.12
 * Time: 10:45
 */
public class BobsMapper extends Mapper<IntWritable, Text, Text, Text> {
    @Override
    public void map(IntWritable key, Text value, Context context)
        throws IOException, InterruptedException {
        String[] lineFields = value.toString().split("\|");
        if ("168.114.15.1".equals(lineFields[0])) {
            context.write(key, value);
        }
    }
}
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;

/**
 * User: Bob
 * Date: 29.08.12
 * Time: 10:45
 */
public class BobsMapper extends Mapper<IntWritable, Text, Text, Text> {
    @Override
    public void map(IntWritable key, Text value, Context context)
        throws IOException, InterruptedException {
        String[] lineFields = value.toString().split("\|"卓越");
        if ("168.114.15.1".equals(lineFields[0]) {
            context.write(key, value);
        }
    }
}
MapReduce Job 4711 canceled

Creating Trojan Index on Attribute 0...

Stop

Trojan Index Query Time

![Graph showing comparison between Hadoop and Hadoop++ (Trojan Index) on query time with 100 nodes. The graph indicates that Hadoop++ (Trojan Index) has a significantly shorter query time compared to Hadoop.]
Creating Trojan Index on Attribute 0...
Stop
Stop
Creating Trojan Index on Attribute 0...
Stop
MapReduce Job 1337 canceled
Trojan Index Creation Time

[Image - 54x-119 to 396x579]

HDFS
+ MapReduce

Bob's Perspective

HDFS

horizontal partitions

HDFS

horizontal partitions

HDFS blocks 64MB (default)

HDFS
map(row) -> set of (ikey, value)

MapReduce

HDFS

Map Phase

MapReduce

HDFS
Map Phase

MapReduce  
map(row) -> set of (key, value)

HDFS
Map Phase

MapReduce: map(docID, document) -> set of (term, docID)

HDFS

HDFS + MapReduce

HAIL

HDFS + MapReduce

HDFS + MapReduce

HAIL + MapReduce
HAIL

horizontal partitions

HDFS blocks
64MB (default)
Failover

Details
HAIL Upload Pipeline

HAIL Query Pipeline

Experiments
Upload Times

Upload Time

Number of created indexes

Upload time [sec]

Hadoop
Hadoop++
HAIL

Upload time [sec]

Number of created indexes

all with 3 replicas

Hadoop
Hadoop++
HAIL

all with 3 replicas
Upload Time

Replica Scalability
Scale-Out

Query Times

Individual Jobs: Weblog, RecordReader
Individual Jobs: Weblog, Job

![Job Runtime Chart]

Scheduling Overhead

![Scheduling Overhead Chart]

HAIL Scheduling
Hadoop Scheduling

| MapReduce | map(row) -> set of (ikey, value) |

HAIL

- sort order
  - 7 map tasks (aka waves)
  - 7 times scheduling overhead

HAIL Scheduling

| MapReduce | map(row) -> set of (ikey, value) |

HAIL

- sort order
  - 1 map task (aka wave)
  - 1 times scheduling overhead

HAIL Split

Query Times

*with* HAIL Scheduling
Individual Jobs: Weblog

 Failover

 Failover

 10 nodes, one killed
fast indexing AND fast querying