Cloud Security in Map/Reduce

An Analysis

July 31, 2009

Jason Schlesinger
ropyrusk@gmail.com

Presentation Overview

Contents:

1. Define Cloud Computing
2. Introduce and Describe Map/Reduce
3. Introduce Hadoop
4. Introduce Security Issues with Hadoop
5. Discuss Possible Solutions and Workarounds
6. Final Thoughts

Goals:

- Raise awareness of Hadoop and its potential
- Raise awareness of existing security issues in Hadoop
- Inspire present and future Hadoop users and administrators to be aware of security in their Hadoop installation

Defining Cloud Computing

- Distributed across multiple machines that are linked together either through the Internet, or across an internal network
- Fault tolerant to hardware failure, which is inevitable to happen in a large cluster scenario
- Applications are abstracted from the OS (more or less)
- Often used to offload tasks from user systems that would be unreasonable to run, or unfavorable to maintain.

Considering the above:
Hadoop is an incarnation of Map/Reduce in a Cloud environment.

Map/Reduce: What It Is

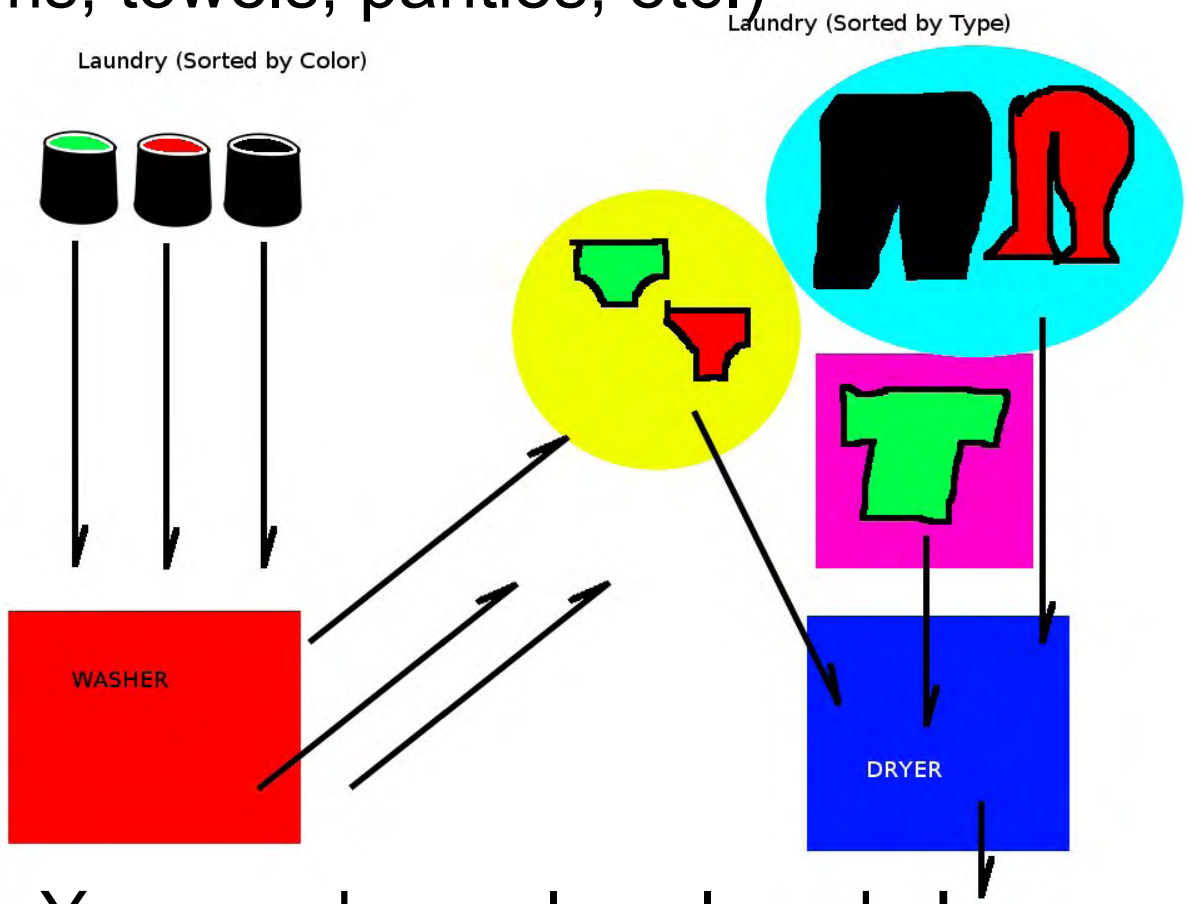
Map/Reduce is for huge data sets that have to be indexed, categorized, sorted, culled, analyzed, etc. It can take a very long time to look through each record or file in a serial environment. Map/Reduce allows data to be distributed across a large cluster, and can distribute out tasks across the data set to work on pieces of it independantly, and in parallel. This allows big data to be processed in relatively little time.

Funfact: Google implemented and uses a proprietary Map/Reduce platform. Apache has produced an open source Map/Reduce platform called Hadoop

Laundromat analogy of Map/Reduce

Imagine that your data is laundry. You wash this laundry by similar colors. Then you dry this laundry by similar material (denims, towels, panties, etc.)

Serial Operation:

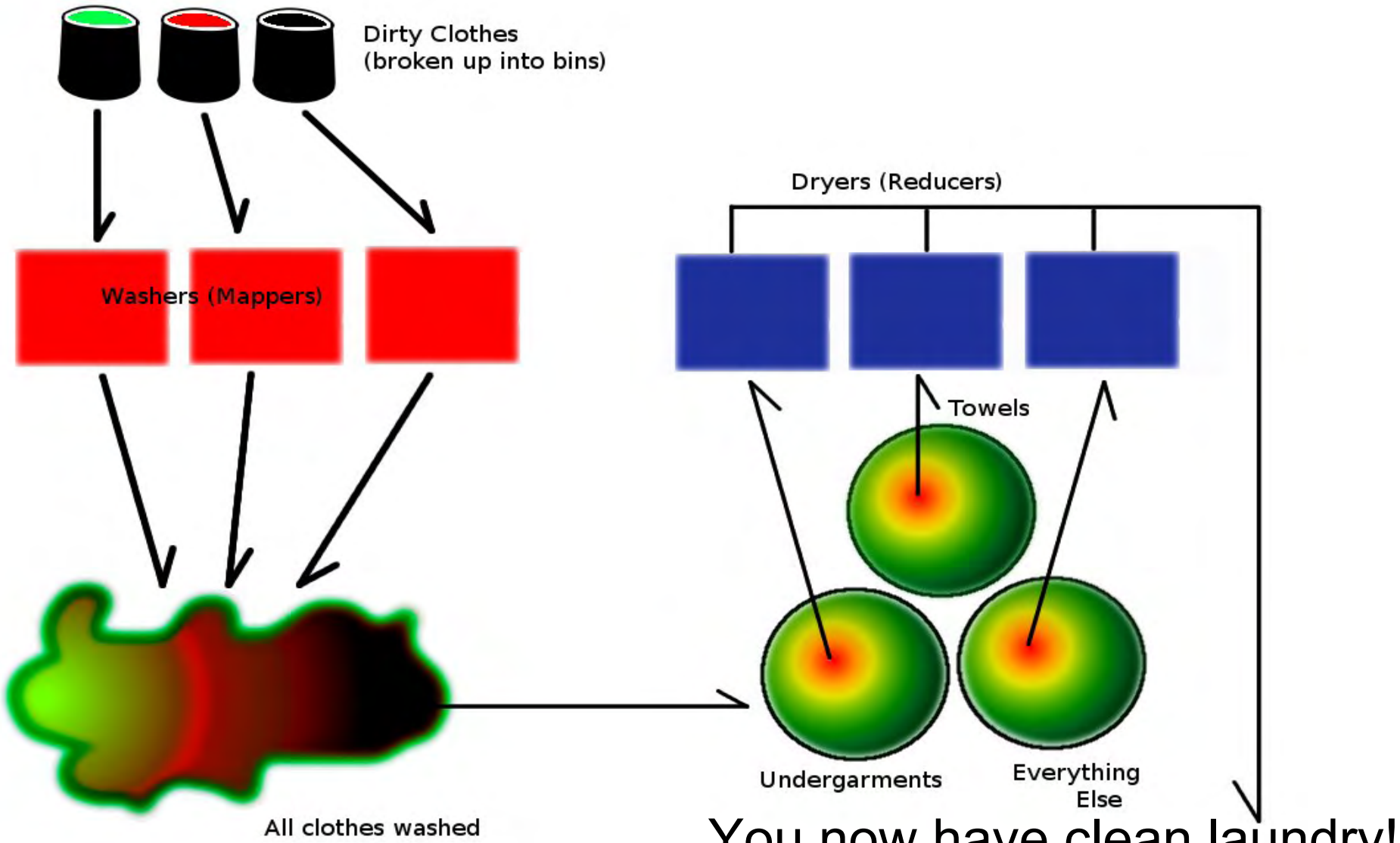


You now have clean laundry!

(Time Elapsed: 2-3 hrs)

Laundromat analogy of Map/Reduce

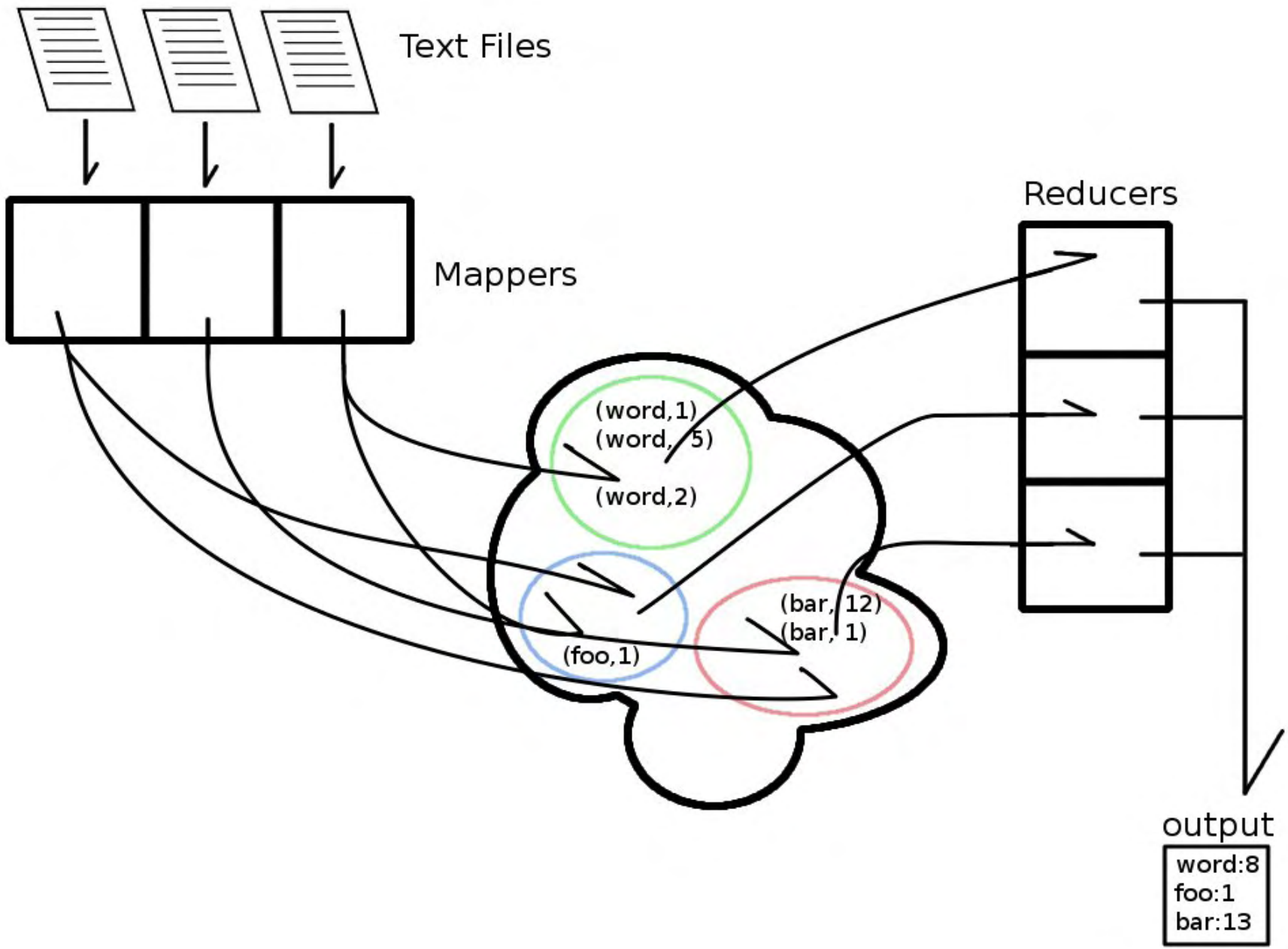
Map/Reduce operation:



You now have clean laundry!

(Time elapsed: 1.25 hrs.)

Word Count example of Map/Reduce



Other Potential uses of Map/Reduce

Since it takes a large data set, breaks it down into smaller data sets, here are some potential uses:

- indexing large data sets in a database
- image recognition in large images
- processing geographic information system (GIS) data - combining vector data w/ point data (Kerr, 2009)
- analyzing unstructured data
- analyzing stock data
- Machine learning tasks

Any situation where processing a data set would be impractical due to its size.

Map/Reduce is a little more confusing...

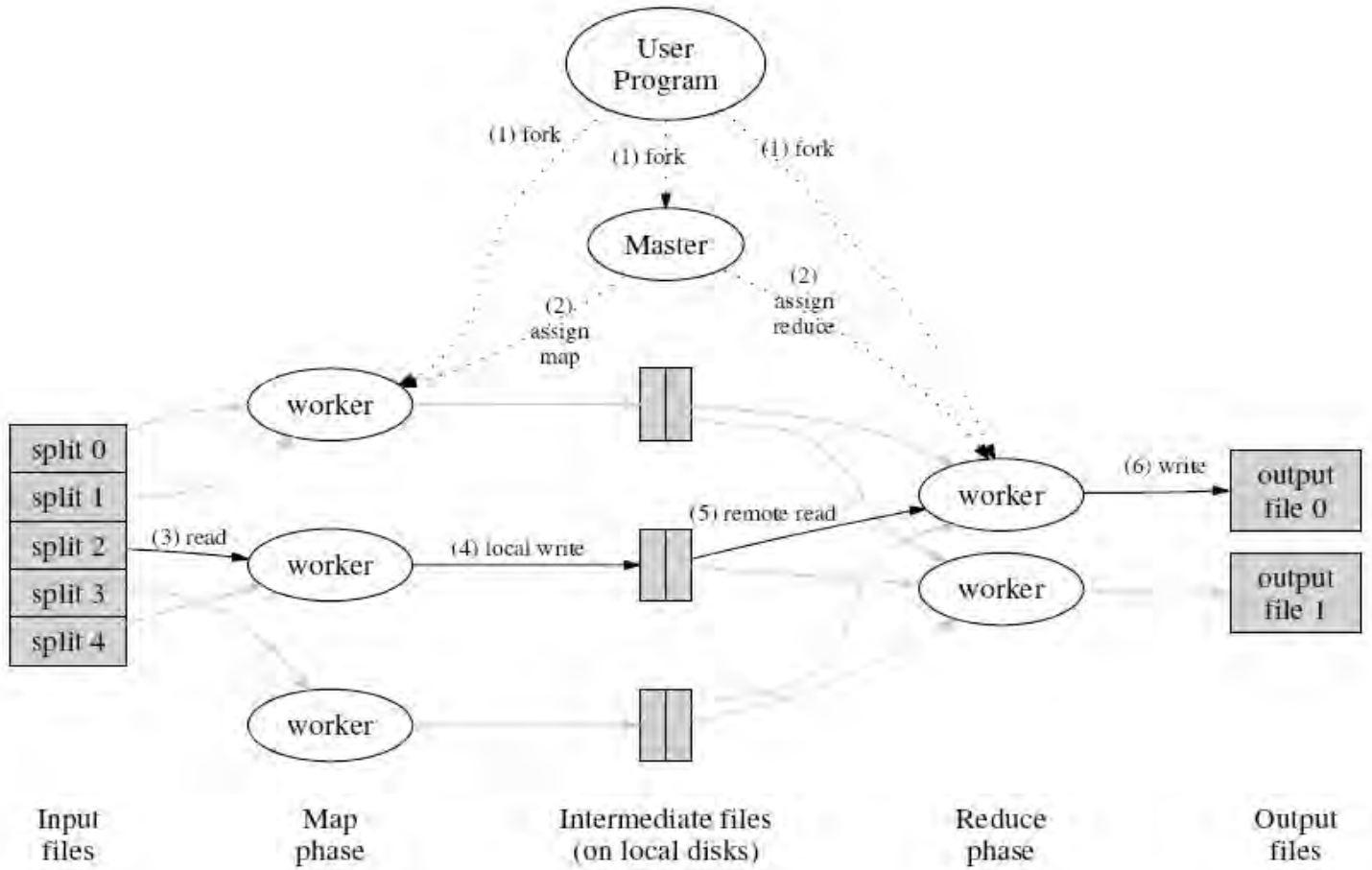


Figure 1: Execution overview

Source: (Dean & Ghemawat, 2004)

Hadoop - A Brief Overview

- Developed by Apache as an open source distributed Map/Reduce platform, based off of Google's MapReduce as described in Dean and Ghemawat, 2004.
- Runs on a Java architecture
- Hadoop allows businesses to process large amounts of data quickly by distributing the work across several nodes.
- One of the leaders of open source implementations of Map/Reduce.
- Good for very large data sets and on large clusters.
- Growing as a business and research tool

Hadoop - A Key Business Tool

Used by Large Content-Distribution Companies, such as...

- Yahoo! - Recently released their version of Hadoop
  - Hadoop is used for many of their tasks, and over 25,000 computers are running Hadoop. (PoweredBy, 2009)
- A9
  - Hadoop is good for Amazon, they have lots of product data, as well as user-generated content to index, and make searchable. (PoweredBy, 2009)
- New York Times
  - Hadoop is used to perform large-scale image conversions of public domain articles. (Gottfrid, 2007)
- Veoh
  - Hadoop is used to "reduce usage data for internal metrics, for search indexing and for recommendation data." (PoweredBy, 2009)

Hadoop - Why I Care (and so can you!)

Used by non-content-distribution companies, such as

- Facebook
- eHarmony
- Rackspace ISP

Other early adopters include anyone with big data:

- medical records
- tax records
- network traffic
- large quantities of data

Wherever there is a lot of data, a Hadoop cluster can generally process it relatively quickly.

Security Framework and Access Control

Now that we know that Hadoop is increasingly useful, here are the security issues with it:

- Hadoop holds data in HDFS - Hadoop Distributed File System. The file system as of version 0.19.1 has no read control, all jobs are run as 'hadoop' user, and the file system doesn't follow access control lists.
- The client identifies the user running a job by the output of the 'whoami' command - which can be forged (Kerr, 2009)
- HBase (BigTable for Hadoop) as of ver. 0.19.3 lacks critical access control measures. No read or write control.
- The LAMP analogue, any application can access any database by simply making a request for it.

Implications of this Accusation

- Any business running a Hadoop cluster gives all programmers and users the same level of trust to all the data that goes into the cluster.
- Any job running on a Hadoop cluster can access any data on that cluster.
- Any user with limited access to the jobs they can run, can potentially run that job on any data set on the cluster.

Pause for Demonstration

(Or similar substitue.)

Possible Workarounds

- Keep each data set on its own Hadoop Cluster
  - If attackers can only access data they have rights to, then the point is moot
  - It is possible to run each job in its own cluster on **Amazon Web Services** with the Elastic MapReduce service, which sits on the Elastic Cloud Computing platform. Simply upload your data to Amazon, give it a job, tell it how many nodes to use, and run it.
  - **Hadoop on Demand -** load data into a real cluster, and generate a virtual cluster every time a job is run.

Possible Workarounds

- Don't store any confidential, secret, private data in Hadoop
  - No one cares if the group that's indexing the forum data can access the knowledge base data (actually, we wish they would more often)
- Encrypt all your sensitive data
  - This will make it difficult to analyze sensitive fields
  - Sensitive data is not always defined as such, and may leak into ustructured fields (such as comments sections)
  - Adds overhead of moving data that most jobs won't read.

Possible Solution

- Develop a Solution that sits on the file system, or write a concerned email to Hadoop developers
  - The problem is that access control is held at the client level, when it should be at the file system level.
  - Access control list checks should be performed at the start of any read or write.
  - User authentication should use a more secure method, such as a password or RSA key authentication.

Final Thoughts

Hadoop is a rising technology, not quite mature, and still has plenty of its own issues. However it's starting to take hold in the marketplace, and now is the time to quell bigger issues like this.

We have the power to shape the future today, let us learn from the mistakes of the past.

Bibliography

Dean, J., & Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters (2004). Google, Inc..

Gottfrid, D., 2007. Self Service, Prorated, Super Computing Fun. Retrieved 6 29, 2009, from http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/

Hadoop 19.1 API, 2009. Hadoop Documentation. Apache Software Foundation. Retrieved 3 10, 2009, from http://hadoop.apache.org/core/docs/r0.19.1/api/

Hadoop Map/Reduce Tutorial, 2009. Apache Software Foundation. Retrieved 3 10, 2009, from http://hadoop.apache.org/core/docs/r0.19.1/mapred_tutorial.html

Powered By, 2009. Apache Software Foundation. Retrieved 3 10, 2009, from http://wiki.apache.org/hadoop/PoweredBy

Kerr, N., 2009. http://nathankerr.com/