

Taming Orbitz Logs with Hadoop

SEPTEMBER 15 | 2010



About Orbitz Worldwide

- Operates multiple brands across the globe
 - In the US: Orbitz, Cheaptickets, The Away Network and Orbitz for Business
 - Internationally: ebookers, HotelClub (includes RatestoGo and Asia Hotels)
- Powers multiple supplier sites
 - For air: Northwest and American Airlines
 - For hotel: Southwest Airlines
 - For vacation packages: MSN Travel, Yahoo! Travel, AAA, Marriott and United
- Initial Public Offering on July 20, 2007!
 - Registered as OWW on NYSE
 - Primary shareholders: Travelport and Blackstone

Orbitz History

- Orbitz started in 1999, site launched in 2001
- Initial Public Offering (IPO) at end of 2003 (Jeff Katz, CEO)
- Cendant purchases Orbitz in 2004 to become Cendant Travel Distribution Services (TDS)
- 2006 Cendant dissolves and spins companies; Travelport (formerly TDS) bought by Blackstone (private equity)
- June 2007: IPO of Orbitz Worldwide
- June 2008: Launch of Price Assurance for Flights
- January 2009: New President & CEO Barney Harford
- May 2009: Launch of Hotel Price Assurance



About Me

- Steve Hoffman
 - orbitz: steve.hoffman@orbitz.com
not orbitz: steveh@goofy.net
 - blog: <http://blog.goofy.net/>
 - twitter: [@hoffman60613](https://twitter.com/hoffman60613)
- 2nd developer @Orbitz
 - 2000-2005
 - “The JINI Guy”
 - 2010-present
 - Principal Engineer, Intelligent Marketplace

2004 Dukié - Orbitz won “Miss Con-Jini-ality”



ORBITZ
WORLDWIDE

This is the closest I ever got to somebody famous.
I think they call this the “Duke’s Choice Awards” now.

The Problem

- Too Many Logs - Terabytes (just application/access)
 - Primary storage (SAN, NAS, etc) is expensive
 - Currently only 60 days retention before archive
- Too Many Tools (many 1-off afterthoughts to log processing)
 - loggrep (web based)
 - socgrep (web based)
 - ssh/grep
 - “Top 10” Exceptions
 - Splunk (big \$ at high volume)
 - Confusion...

Why Hadoop?

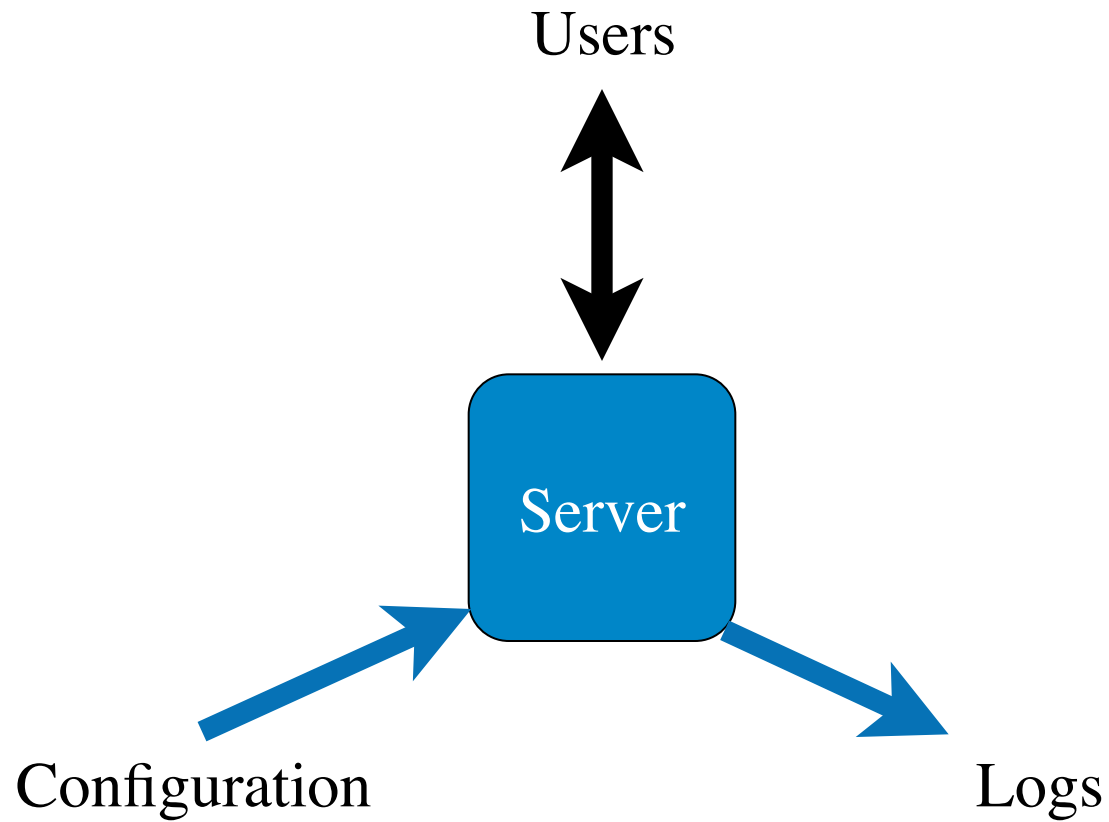
- Cheap reliable Storage
 - Horizontally Scalable
 - Commodity Hardware
- Cheap reliable Computing
 - Horizontally Scalable
 - Map/Reduce Framework for Log Analytics
 - Community Computing Resource

- “Move the search to the logs rather than the logs to the search”

Hadoop @Orbitz

- Two Production Clusters (built late 2009)
 - Operations
 - Intelligent Marketplace
- Ops Cluster
 - 2 NN/JT Nodes + 10 Data Nodes
 - 40TB raw/13TB usable (rep factor of 3)
 - Storage capacity for several months of production logs
 - 20 cpu's/80 cores for log analysis
- IM Cluster similar in size to Ops
- Plans to combine production clusters and grow soon

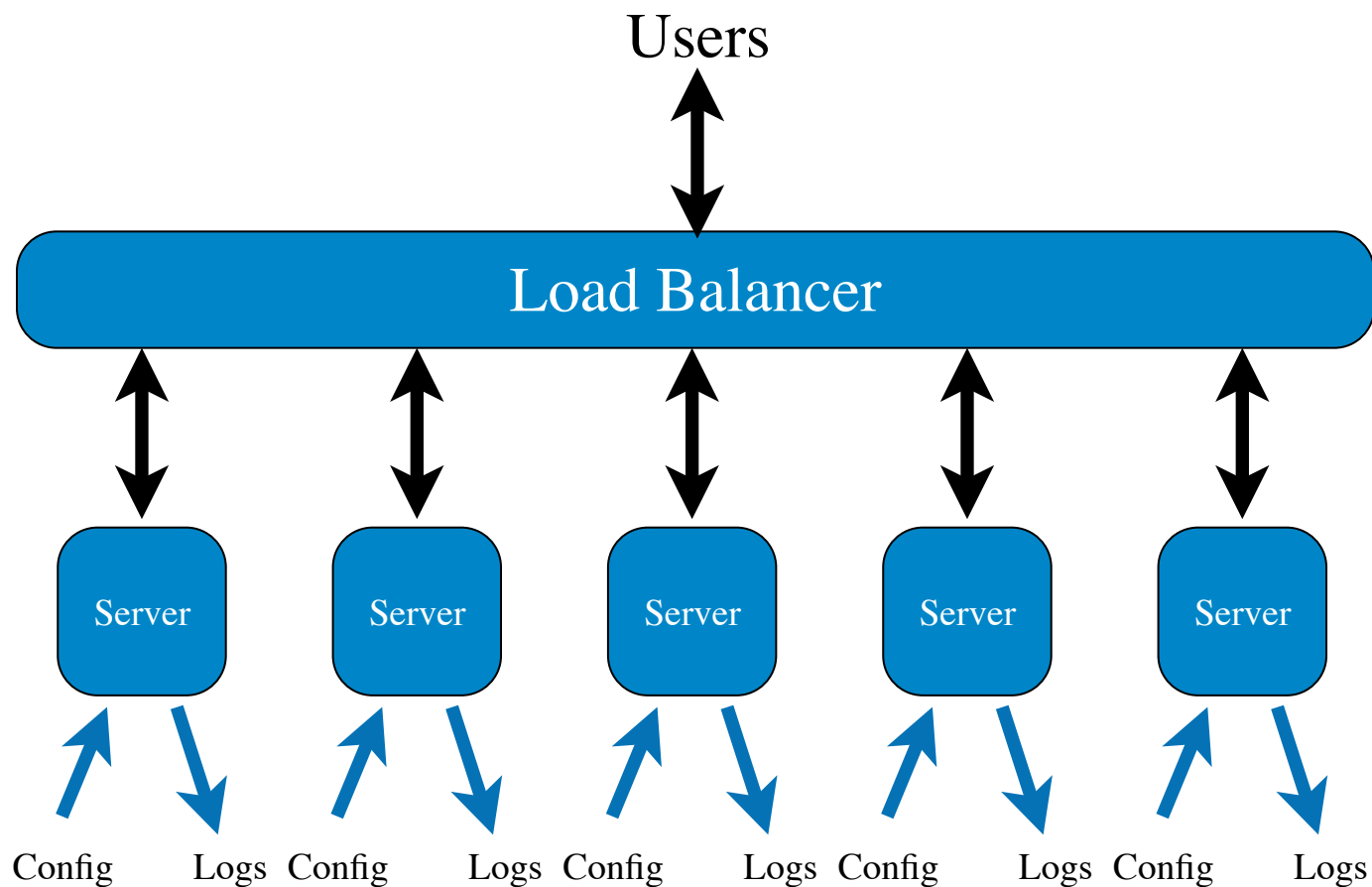
Large System™ : v1.0



ORBITZ
WORLDWIDE

So you want to build a large system?
You start by building the functionality your users see.
You configure via Property files
You use something log4j-ish for logging errors for when stuff is broken

Large System™ : v2.0

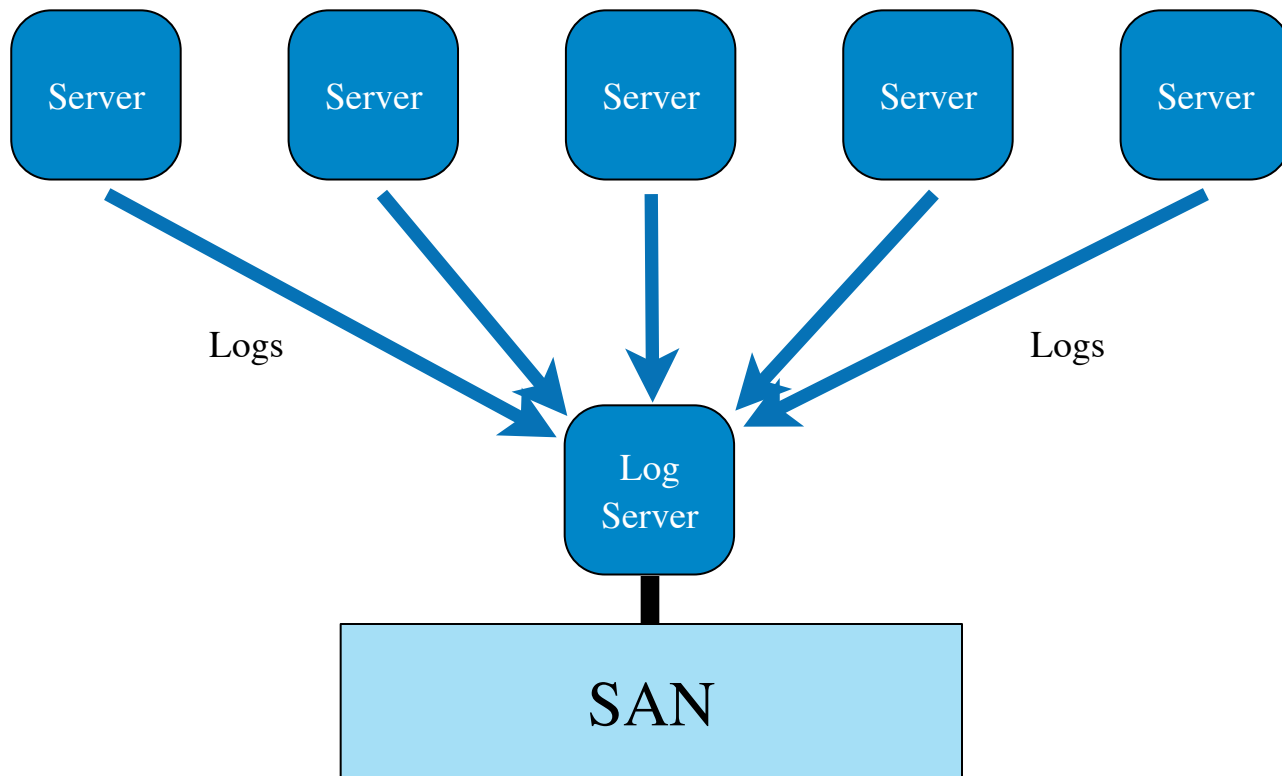


When you need more capacity you put a load balancer in front of multiple instances of your application. Leaving off things like cross application state (replicated session, database, whatever). Eventually manual configuration of hundreds then thousands of machines becomes problematic you turn towards things like centralized configuration (database, http server to pull property files, chef, cfengine, and so on). MorningStar Tech Talk 9/21 on configuration in cloud that should touch on this. When talking logs on thousands of machines, how do you know when something is wrong?

What do you mean when you say “Logs”?

- Are these developer centric logs for problem solving?
 - Stacktraces, DEBUG, TRACE
- Are these key events in your system for behavior tracking?
 - User made an air booking to Las Vegas
 - User searched for Maui hotels
- Are they statistics? (Are we going to make a graph?)
 - Response time to user
 - Time to query DB
- Are they “monitoring” events?
 - OutOfMemory
 - Machine Down
 - Disk Full
- **Answer: Yes**

Logs Logs Everywhere: Consolidate!



ORBITZ
WORLDWIDE

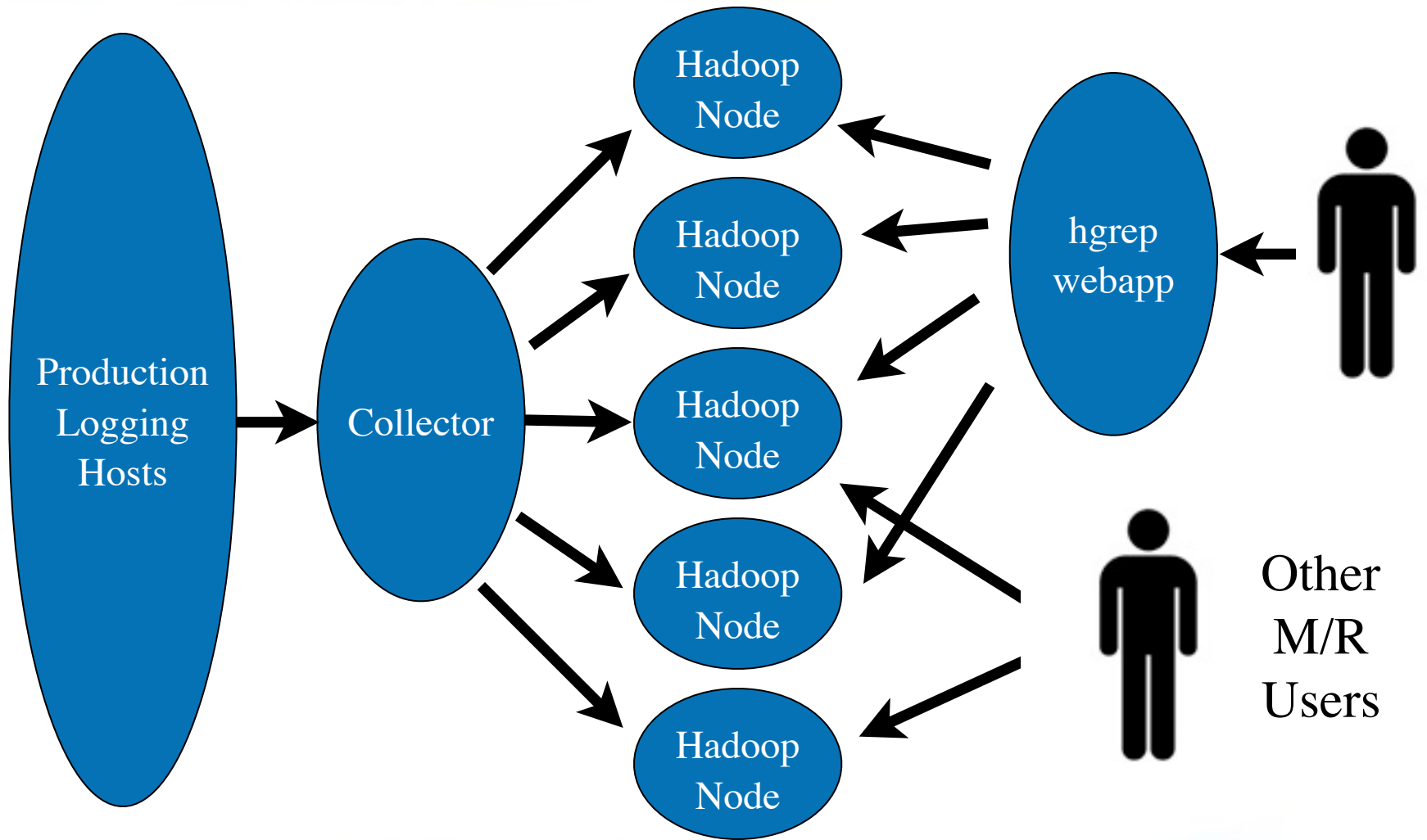
Next step is to move the logs to a central location which usually has a very very large storage device attached. As a consumer, you can get 2TB for a little over \$100 today. Enterprise systems cost much more at scale. There are plenty of ways to get logs from A to B (syslog, syslog-ng). People log into log server to search (ssh/grep family).

Hadoop POC

- Is Hadoop a viable platform for managing OWW logs?
 - Can we get data into Hadoop in real time?
 - Can we use Map/Reduce to replace existing “grep” tools
 - Can Hadoop run 24/7/365?

- Yes!

The Plan



Hadoop SPOF

- Name Node/Job Tracker are central point of failure today (0.20)
- Cloudera shows you how here: <http://bit.ly/hadoopha>
- Uses:
 - linux-ha (aka heartbeat)
 - drbd (replicated block device)
- Our modifications:
 - Used crossover cable on second ethernet interface rather than bonded interfaces (drbd & heartbeat)
 - Made some of the init.d scripts LSB compliant so heartbeat didn't get confused
 - Standby namenode/jobtracker is Primary webapp and vice-versa (2 resource stacks managed by heartbeat)

Our /etc/ha.d/haresources

NamesNodeA \

IPaddr::192.168.1.4 \ (VIP for NN/JT)

drbddisk::r0 \

Filesystem::/dev/drbd0::/hadoop::ext3::defaults \

hadoop-0.20-namenode \

hadoop-0.20-secondarynamenode \

hadoop-0.20-jobtracker

NameNodeB \

IPaddr::192.168.1.5 \ (VIP for WEBAPP)

tomcat \

httpd



tomcat doesn't have an init.d script - really?

Our /etc/ha.d/ha.cf

```
# NamenodeA
# eth0: 192.168.1.2/24
# eth1: 172.16.0.1/30
debugfile /var/log/ha.debug
logfile /var/log/ha.log
logfacility local0
keepalive 1
initdead 60
deadtime 5
ucast eth0 192.168.1.3
ucast eth1 172.16.0.2
ping 192.168.1.1
node NamenodeA
node NamenodeB
auto_failback off
```

```
# NamenodeB
# eth0: 192.168.1.3/24
# eth1: 172.16.0.2/30
debugfile /var/log/ha.debug
logfile /var/log/ha.log
logfacility local0
keepalive 1
initdead 60
deadtime 5
ucast eth0 192.168.1.2
ucast eth1 172.16.0.1
ping 192.168.1.1
node NamenodeA
node NamenodeB
auto_failback off
```

- Heartbeat unicast checks on both interfaces to other machine
- Ping “health” check to gateway IP as test
- Failback turned off (machines are homogenous so doesn’t matter which is active)

Collector v1.0 - The Perl Script

- Multithreaded Perl
 - Reads from consolidated syslog-ng FIFO pipe
 - 500MB 'chunks' (balance of "real time" with hadoop efficiency)
 - gzip compress (to just under 64MB)
 - write to hdfs:/logs/prod/raw/YYYY/MM/DD/HH
 - secondary collector writing to hdfs:/logs/prod/raw/.2ndary/YYYY/MM/DD/HH
 - Manually reconcile if necessary (non-optimal)
- Goal: Get the logs into HDFS ASAP



2-3 minutes is close enough to real-time for POC

So what to use for Collector v2.0?

- Scribe @ <http://github.com/facebook/scribe>
 - Clearly works on large data sets - Facebook
 - Compile yourself - dependencies, packaging, etc.
 - Can't tail a file into hdfs (prototype on a branch)
 - Thrift API (Log4j appender example)

So what to use for Collector v2.0?

- Flume @ <http://github.com/cloudera/flume>
 - Not battle tested like Scribe
 - Part of Cloudera distro (as of CDH3b2)
 - Has log4j appender
 - Has ability to tail 1 or more files as source
 - Defines transport guarantees (best effort, next hop, end-to-end) per logging channel
 - Centralized configuration based on clustered Zookeeper

So what to use for Collector v2.0?

- Chukwa @ <http://incubator.apache.org/chukwa/>
 - Similar in architecture to Flume
 - Includes analysis piece we may or may not want
 - Been around longer than Flume
- Disclaimer: I need to look at this more
 - what's the mysql DB for?
 - can I use just the collection framework?
 - How do you configure it?

Getting Data Out v1.0

- Single Server
- Log into server and run grep
 - Pros:
 - Simple
 - All logs in one place
 - Cons
 - Access to production server
 - Negative impact on customer experience

Getting Data Out v1.1

- Multiple Servers
- Log into server and run grep
 - Pros:
 - Simple
 - Cons
 - Access to production server
 - Negative impact on customer experience
 - Log scattered (ssh/shell for-loop magic?)

Not possible to do cross environment correlation.
If I see this error on 1 server, not necessarily a problem.
If I see it on all my servers, clearly problem.

Getting Data Out v1.2

- Multiple Servers consolidated to single server
- Log into log single server and run grep
- Can also slap web UI on server to do searches (we did)
 - Pros:
 - Simple again!
 - Not logging into production server
 - Not impacting customer experience
 - Cons
 - Multiple searches degrade troubleshooting. Separation of “production troubleshooting” from “analysis”
 - Gonna need a bigger server/expensive “large” storage
 - Web UI not ideal for delivery of large amounts of results



Not possible to do cross environment correlation.
If I see this error on 1 server, not necessarily a problem.
If I see it on all my servers, clearly problem.

Getting Data Out v1.5

- Bring in Splunk (or similar)
- Nice Web UI, graphs, indexing, etc.
 - Pros:
 - Searches faster
 - Not logging into production server
 - Not impacting customer experience
 - Cons
 - Not free (maybe, but definitely not at volume)
 - Gonna need a bigger server/expensive “large” storage
 - At some point DB backing store will strain under volume

Hadoop + grep = hgrep

Impedance Mismatch?

- MR jobs are run as batch jobs
 - Existing loggrep tools are interactive/webUI
 - Request to Server/Result to Browser
 - Hadoop Input/Output is HDFS
- What is the Key/Value for Mapper/Reducer when we are talking log entries?
 - Usually mapper used to find needles in haystacks in parallel. Reducer small work of combining individual results.
 - With logs, any record matching criteria, output whole record

Webapp for a batch world

- Web Form submits job with parameters to Job Tracker (gets back a JobID)
 - Call `JobClient.submitJob()` instead of `runJob()`
 - Write job metadata to HDFS (jobID, parameters, location of output, etc)
- Web Page monitors progress of job querying JobTracker
 - Also shows other jobs running/state w/ links to results
 - Progress so far (matches found, % done) with ability to kill
 - Results live in HDFS so can be accessed anytime
- Details Page
 - Results are LARGE. Just show a few at a time (page results)
 - Link for full dump of result set (try before you buy)

Feature Overload

- When we talked with people, everybody wanted different features
 - grep style output (developer)
 - trending (is “XXX” out of norm?)
 - graphs
- Start with simple grep and go from there
- Provide help/training so others can use Hadoop to get what they want.
 - Hive
 - Pig
 - Map/Reduce (Java or Other™ via streaming)



If it turned out a large enough population needed something, then add to standard webapp

Input Form

Results are [here](#)

Grep Search **Raw Logs**

Dates

***From:**
***To:**

Criteria

Hostname:

Session ID:

Class:

Message:

Log Level

FATAL
 ERROR
 WARN
 INFO
 DEBUG
 VERBOSE
 UNKNOWN

Memo

***Enter a memo to help find your results later:**

*=Required Field

Job Running

Refreshing every 10 seconds.... [Stop refreshing](#)

[New Search](#)

[Today](#) | [Previous Day](#) | 2010-06-25 | [Next Day](#)

Submitted Time	Memo	Job ID	Status
14:56:25	RTE Last Hour	job_201006251417_0001	Running Map 89.06% Reduce 18.23%

All Done!

[New Search](#)

[Today](#) | [Previous Day](#) | 2010-06-25 | [Next Day](#)

Submitted Time	Memo	Job ID	Status
14:56:25	RTE Last Hour	job_201006251417_0001	Results

Details

[Back to summary](#)

job_201006251417_0001 started 2010-06-25 at 14:56:25

Memo: RTE Last Hour

Page 1

[Download All!](#)

[Next 10](#)

Time	Log Level	Host	Instance	POS	Session ID	Thread ID	
2010-06-25T13:55:00.051-05:00	ERROR		air_search-31.8-0	EBFR	46891DF9335680276	b470c99f	com.orbitz.air.

DEMO



ORBITZ
WORLDWIDE

What about Real-Time search?

- Your site may not have the volume of logs to approximate real-time.
- Perhaps a combination of approaches?
 - Rackable
 - They build Solr indexes for last 7 days
 - Then fall back to brut-force MR beyond that
- Perhaps commercial offerings for last N days is an option rather than roll-your-own?
- Some other intermediate transformation of your data?
 - i.e. Hbase/BigTable tables
- We are still figuring this out...



Some logs tell a story of how you interact with your customers, so create a model (ala bigtable).

User searched for Vegas. User booked Air. User booked hotel 3 days later.

So What's the Key/Value?

- The Log message is the Key (sortable by time)
- There is no Value (NullWritable)



OrbitzLogWritable

```
public class OrbitzLogWritable implements WritableComparable {  
  
    private final LongWritable timestamp = new LongWritable(-1L);  
    private final Text type = new Text();  
    private final Text hostname = new Text();  
    private final Text instance = new Text();  
    private final Text pos = new Text();  
    private final Text sessionId = new Text();  
    private final Text threadId = new Text();  
    private final Text serviceName = new Text();  
    private final Text clazz = new Text();  
    private final IntWritable lineNumber = new IntWritable(-1);  
    private final Text message = new Text();  
  
    // Getters and Setters  
}
```

Filter Helper Class

```
public boolean accept(OrbitzLogWritable entry) {
    // Check each criteria in turn
    DateTime entryTime = entry.getEventTime();
    if (entryTime.isBefore(criteria.getStartDateTime())) {
        // Entry is before the start time so don't include
        return false;
    }
    if (entryTime.isAfter(criteria.getEndDateTime())) {
        // Entry is after the end time so don't include
        return false;
    }
    // Check log level match
    if (!criteria.getLogLevels().contains(entry.getLogLevel())) {
        // entry's log level not in the set we care about so don't include
        return false;
    }
    // Check for hostname match
    if (!patternMatch(criteria.getHostNamePattern(), entry.getHostname())) {
        return false;
    }
}

// and so on...
}
```

Grep Mapper

```
public class GrepMapper extends MapReduceBase
    implements Mapper<OrbitzLogWritable, NullWritable,
                    OrbitzLogWritable, NullWritable> {

    public void map(OrbitzLogWritable key, NullWritable value,
        OutputCollector<OrbitzLogWritable, NullWritable> output,
        Reporter reporter) throws IOException {
        if (key.isValidRecord()) {
            if (filter.accept(key)) {
                output.collect(key, value);
            }
            reporter.progress();
        } else {
            reporter.setStatus("Ignoring invalid record: " + key);
            reporter.incrCounter(Parsing.ERROR, 1);
        }
    }
}
```

Grep Reducer

```
public class GrepReducer extends MapReduceBase
    implements Reducer<OrbitzLogWritable, NullWritable,
                    OrbitzLogWritable, NullWritable> {

    public void reduce(OrbitzLogWritable key, Iterator<NullWritable> values,
        OutputCollector<OrbitzLogWritable, NullWritable> collector,
        Reporter reporter) throws IOException {
        // Only want to output the key once so can't use IdentityReducer.
        collector.collect(key, NullWritable.get());
        reporter.progress();
    }
}
```

Glue Code

```
JobConf conf = new JobConf(getConf(), getClass());
conf.setJobName("grep");

OrbitzLogInputFormat.setLogRoot(conf, args[0]);

Path outputPath = new Path(args[1]);
FileOutputFormat.setOutputPath(conf, outputPath);

conf.setInputFormat(OrbitzLogInputFormat.class);
conf.setMapOutputKeyClass(OrbitzLogWritable.class);
conf.setMapOutputValueClass(NullWritable.class);
conf.setOutputKeyClass(OrbitzLogWritable.class);
conf.setOutputValueClass(NullWritable.class);
conf.setOutputFormat(SequenceFileOutputFormat.class);

conf.setMapperClass(GrepMapper.class);
conf.setReducerClass(GrepReducer.class);

jobConf.setPartitionerClass(HourlyPartitioner.class);
jobConf.setNumReduceTasks(HourlyPartitioner.totalPartitions(criteria));

JobClient jc = new JobClient(conf);
RunningJob job = jc.submitJob(jobConf);
JobID id = job.getID();
// Save id away for later
```



Final output is a SequenceFile of OrbitzLogWritable.
OrbitzLogInputFormat used to parse prior to Map phase. Also to embed knowledge of files based on dates searched. Only need root. Rest scanned from search criteria. Rather than using FileInputFormat which doesn't do nested directories, just files in given directory.

Custom InputFormatter

```
protected FileStatus[] listStatus(JobConf job) throws IOException {
    String logRoot = job.get(ROOT_KEY);
    GrepCriteria searchCriteria = new GrepCriteria(job);
    DateTime startTime = searchCriteria.getStartDateTime();
    DateTime endTime = searchCriteria.getEndDateTime();
    int hours = Hours.hoursBetween(startTime, endTime).getHours();
    List<Path> paths = new ArrayList<Path>(hours + 1);
    for (int hour = 0; hour <= hours; hour++) {
        DateTime t = startTime.plusHours(hour);
        String path = String.format("%s/%04d/%02d/%02d/%02d/", logRoot,
t.getYear(), t.getMonthOfYear(), t.getDayOfMonth(), t.getHourOfDay());
        paths.add(new Path(path));
    }

    // Now iterate over paths and look for files in each directory
    // (if it exists). In the end you should have a list of FileStatus
    // to return
}
```

Custom InputFormatter

```
// Override getRecordReader to return OrbitzLogWritable to Map phase
public RecordReader<OrbitzLogWritable, NullWritable> getRecordReader(
    InputSplit split, JobConf job, Reporter reporter) throws IOException
{
    reporter.setStatus(split.toString());
    LogParser parser = new SyslogChunkedRecordParser();
    return new OrbitzRecordReader(job, (FileSplit) split, parser);
}
```

Custom RecordReader

```
public class OrbitzRecordReader implements RecordReader<OrbitzLogWritable, NullWritable>
{
    public synchronized boolean next(OrbitzLogWritable key, NullWritable value) throws
    IOException {
        Text line = new Text();
        while (pos < end) {
            int newSize = in.readLine(line, maxLineLength, Math.max((int) Math.min
            (Integer.MAX_VALUE, end - pos), maxLineLength));
            parser.parse(line.toString(), key);
            if (newSize == 0) { return false; }
            pos += newSize;
            if (newSize < maxLineLength) {
                return true;
            }
            // line too long. try again
            LOG.info("Skipped line of size " + newSize + " at pos " + (pos - newSize));
        }
        return false;
    }
}
```



Most of the code is cut/paste job from LineRecordReader
Parser passed into Constructor. Does the work usually cut/paste into map()

Input Parser

```
public interface LogParser {  
    // Parse 1 line into structured object  
    public void parse(String line, OrbitzLogWritable target);  
}  
  
public class SyslogChunkedRecordParser implements LogParser {  
    public void parse(String line, OrbitzLogWritable target) {  
        // Set fields in target from parsed line  
        // what all the examples show done in map()  
        ...  
    }  
}
```

Multiple Reducers via Time Slicing Trick

```
// Since the key is time ordered, send 1 hour worth to each reducer so no
// one machine has to do more than 1 hour's sorting.
// part-00000 has first hour sorted, part-00001 has second, and so on.
// If you read files in order, total ordering is preserved.

public class HourlyPartitioner
    implements Partitioner<OrbitzLogWritable, NullWritable> {
    public int getPartition(OrbitzLogWritable key, NullWritable value,
        int numPartitions) {

        int h = Hours.hoursBetween(criteria.getStartDateTime(),
            key.getEventTime()).getHours();

        if (h < 0) {
            // Can't be before first partition
            return 0;
        }

        if (h >= numPartitions) {
            // Can't be after last partition
            return numPartitions-1;
        }
        return h;
    }

    public static int totalPartitions(GrepCriteria c) {
        return Hours.hoursBetween(c.getStartDateTime(),
            c.getEndDateTime()).getHours()+1;
    }
}
```



Large map() output to single reducer is sorted by key – in this case the log entry.

Use multiple partitioner to split the work (we picked hourly).

Results in multiple output files, but since most M/R stuff takes all files in a directory in order, isn't a problem until you have 100,000 hours which is 11.4 years so we'll worry about it later.

Thanks!

Questions

ORBITZ
WORLDWIDE