

facebook

America's Most Wanted - a metric to detect persistently faulty machines in Hadoop

Dhruba Borthakur and Andrew Ryan
dhruba, andrewr1@facebook.com

Presented at IFIP Workshop on Failure Diagnosis, Chicago
June 25, 2010

Overview

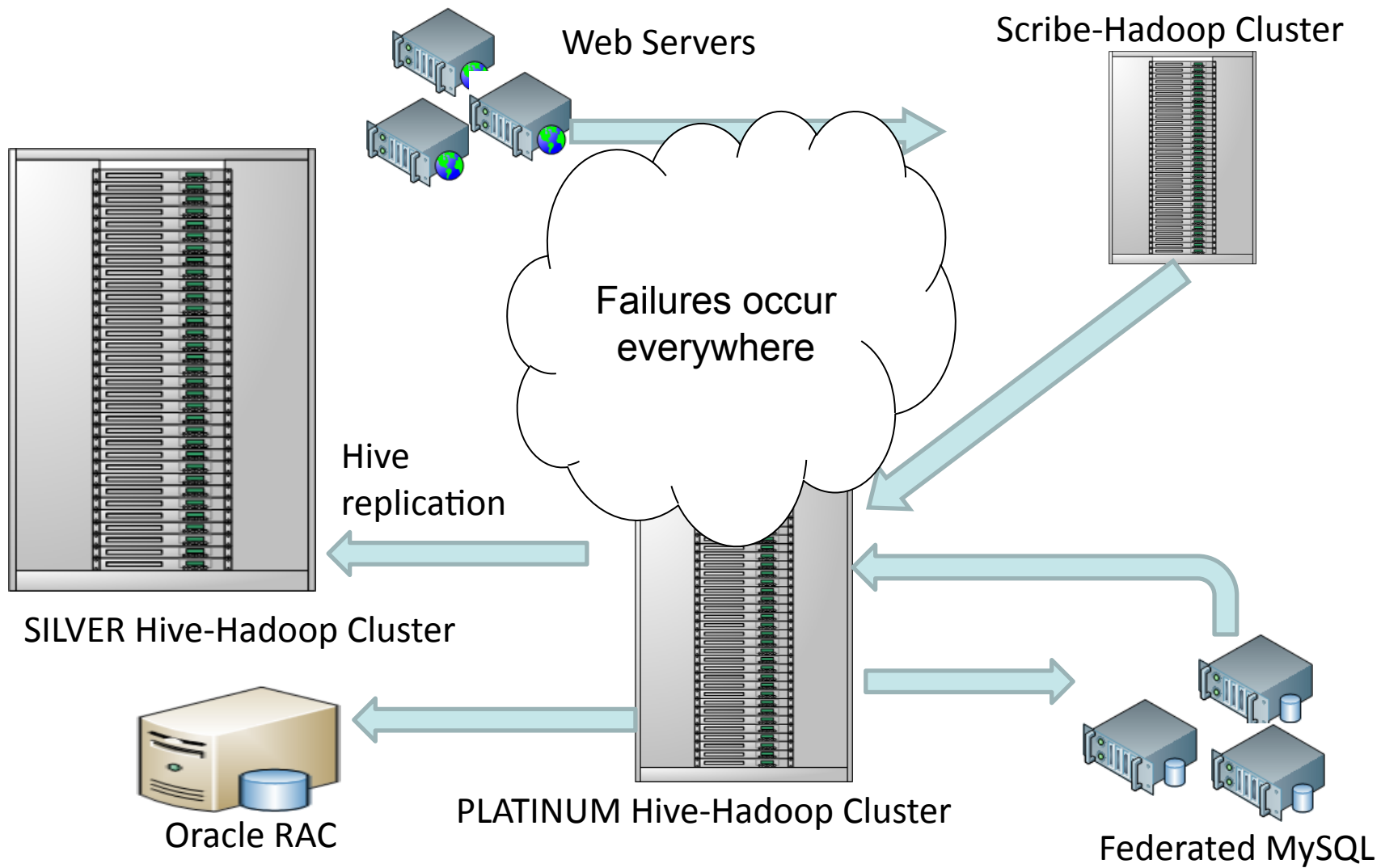
- **Size and scale of Hadoop warehouse cluster**
- **Categorize failures that re-occur**
- **Monitoring tools for system health**
- **Americas Most Wanted Metric**

Primary Challenge: Growth and more Growth!



Recent Growth of Hadoop Data (TB of data)

Data Flow Architecture at Facebook



Hadoop Warehouse @ Facebook

- **Hadoop Warehouse**
 - 16000 cores
 - Raw Storage capacity ~ 21PB
 - 8 cores + 12 TB per node
 - 32 GB RAM per node
 - Two level network topology
 - 1 Gbit/sec from node to rack switch
 - 10 Gbit/sec to top level rack switch
- **Statistics per day:**
 - 800TB of I/O per day via Hive queries
 - 10K - 25K Hadoop jobs per day

Types of Applications

- **Reporting**
 - Daily/Weekly aggregations of impression/click counts
 - Measures of user engagement
 - Microstrategy reports
- **Ad hoc Analysis**
 - how many Page administrators per state or country
- **Machine Learning (Assembling training data)**
 - Ad Optimization
 - User Engagement as a function of user attributes
- **Index Generation**
- **A/B Testing**

Analysis and Data Organization

- **99% of analysis through Hive on Hadoop**
- **Hive**
 - Easy to use
 - Familiar SQL interface with Data as Tables and Columns
 - Easy to extend
 - Can embed map/reduce user programs in the data flow
 - Support for user defined functions
 - Flexible
 - Supports user defined data formats and storage formats
 - Support user defined types
 - Interoperable
 - JDBC, ODBC and thrift interfaces for integration with BI tools

Types of Failures

- **System Errors**
 - Hardware, OS, jvm, hadoop, compiler, etc
 - Hadoop aims to reduce the effect of this broad category of errors
- **User Application Errors**
 - Bad code written by an user
 - Bloated memory usage
- **Anomalous Behaviour**
 - Not working according to expectation
 - Slow nodes
 - Causes most harm to Hadoop cluster

System Errors

- **Operating System Errors & Hardware Errors**
 - Bad network card on rack switch
 - ECC memory corruption
- **Hadoop Framework Errors**
 - JVM bugs
 - Fails to fetch map output
 - No live datanodes contain block
- **Configuration Errors**
 - Code not deployed on some nodes (e.g. older version of jetty)
 - Gcc libraries on some nodes incompatible with LZ0 libraries

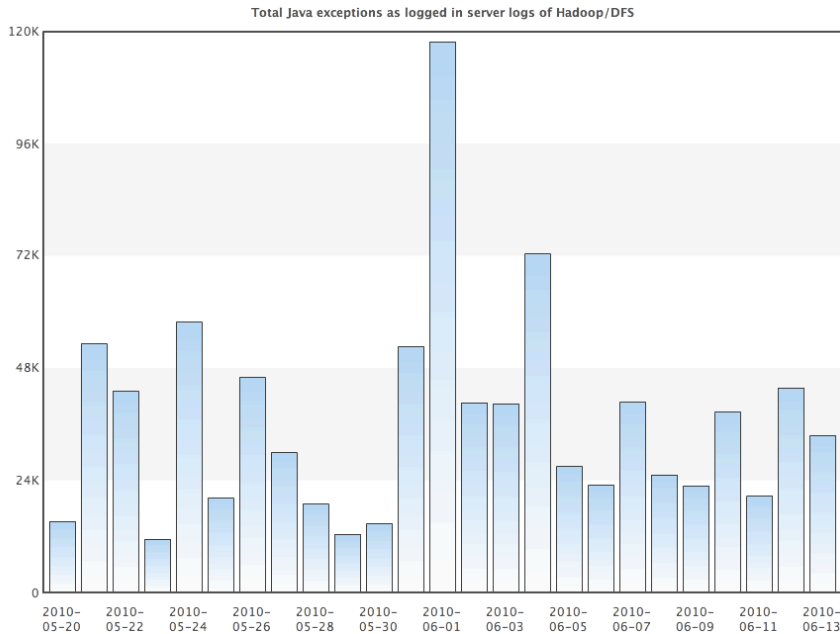
User Errors

- **Hadoop Tasks can be written in any language**
 - A Python dictionary can consume lots of memory
 - Python might not be installed on some nodes
 - A map script written in Python has a syntax error
 - Logical errors
- **More frequent than System Errors**
 - Very important to propagate appropriate message to user
 - Challenge to propagate error messages from lower levels in the software stack to the user

Monitor Hadoop Health

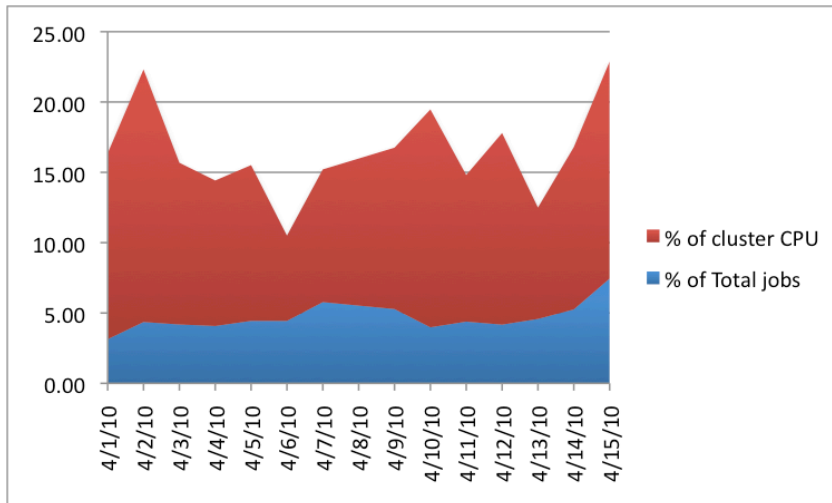
- **Trends and offline analysis**
 - Hadoop History logs contain information about completed jobs
 - Push History logs into Hive tables every hour
 - Produce reports into a mysql database
 - Plot visual reports
- **Online analysis**
 - Generate extended metrics via Hadoop Metrics
 - Hadoop Server's expose metrics via JMX
 - Pull metrics into an RRD Tool
 - Visual dashboard

Plot of Hadoop Server Exceptions



- Exceptions peaked on 06/01 when a bad app deleted mapred system dir in /tmp

Adhoc queries: does python jobs eat more CPU?



- 5% of all jobs in cluster are written in Python
- 15% of cluster CPU is consumed by Python jobs
- 20% of all failed jobs are written in python

Warehouse Utilization and Workload

- **Compute Map-Reduce cluster is CPU bound**
 - Peak usage of 95% CPU utilization
 - Peak network usage is 70% (network is not a bottleneck)
 - 70% tasks find data on local rack
 - We do not compress map outputs
- **Storage HDFS cluster is capacity bound**
 - 75% storage full (current size is 21 PB, 2000 nodes)
 - Disk bandwidth used is 20% of capacity (IO is not a bottleneck)
 - All Hive tables compressed using gzip

How do we fix Errors?

- **Hadoop is fault-tolerant**
 - Node is blacklisted if tasks from multiple jobs fail repeatedly
- **Homegrown Diagnostics scripts**
 - Pings nodes periodically
 - Checks disks, memory, etc. to find bad nodes
 - Instructs hadoop to exclude bad nodes

Anomalies

- **Types of observed anomalies**

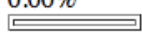
- A anomalous node runs slower than other nodes
- Time on a anomalous node progresses slower than the other nodes
- Data transfer rate from an anomalous node is an abysmal 10 KBps
- Transient Bursty ECC errors from memory module
- Rebooting a machine may appear to fix most of these problems, but may actually mask the real problem.

- **Anomalies are difficult to detect**

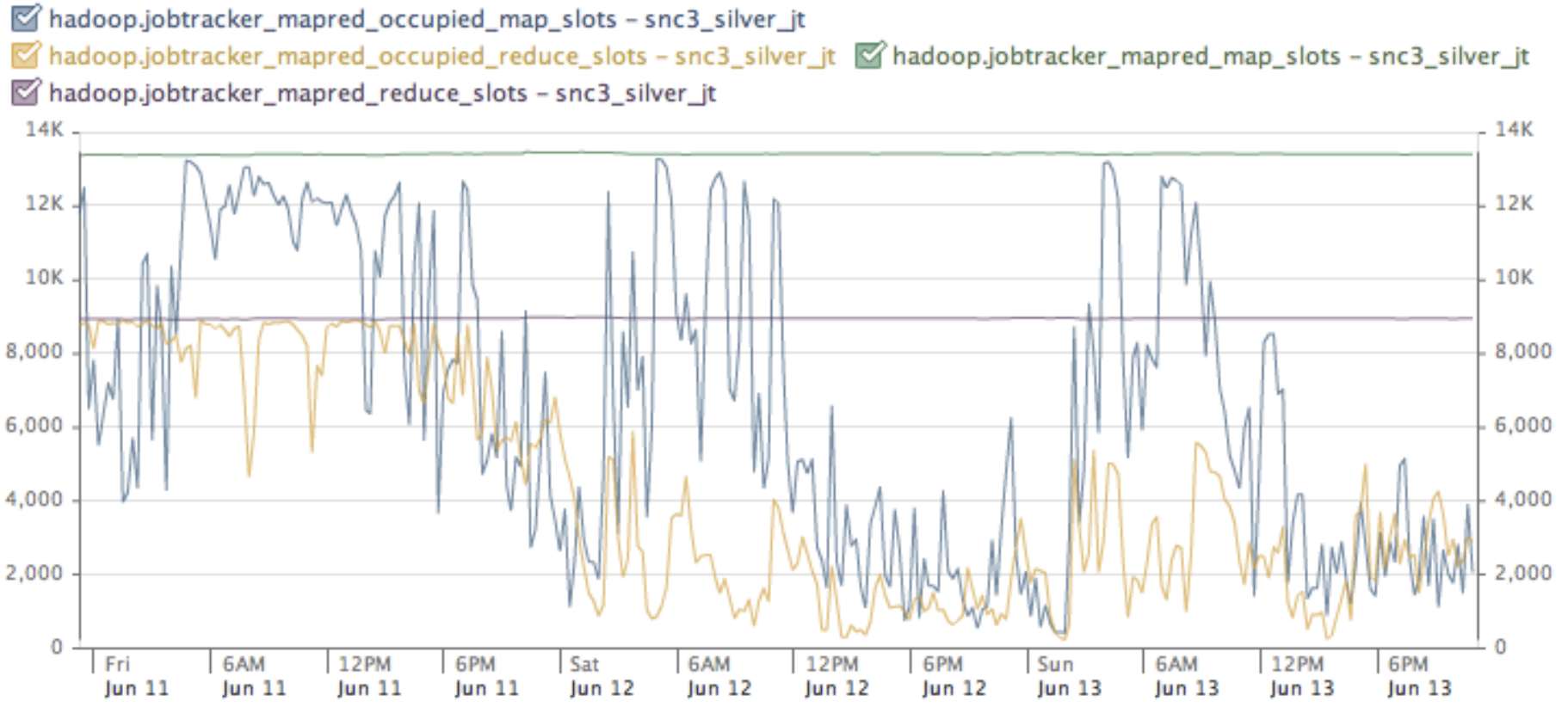
- Hadoop is not very good at handling these scenarios
- Difficult to configure diagnostics tools to account for different types of hardware: How fast should this machine be processing jobs?

Anomalies : Crowd Sourcing to the rescue

- **Harness Hadoop users to detect Anomalies**
 - Our cluster has 50 - 100 simultaneous users at peak load
 - A interactive user sees that task on machineX take a longer time
 - Clicks on *speculate-this-task* button on a UI screen
 - Hadoop speculatively starts execution of another instance of that task
- **Detect Persistent Anomalies (work-in-progress)**
 - Many users sees slow behavior of their tasks on machineX
 - All those users click on *speculate-this-task* button
 - Blacklist machineX from Hadoop cluster

Task Attempts	Machine	Status	Progress	Start Time	Shuffle Finished	Sort Finished	Finish Time	Errors	Task Logs	Counters	Actions
attempt_000_0	machineXm.	RUNNING	0.00% 	7-Jun-2010 04:30:37					Last 4KB Last 8KB All	7	Kill Fail Speculate

Hadoop Workload is bursty



Our observation: failures increase when load increases

Failure rates caused us to split Warehouse

- **Started with one Hadoop cluster**
 - Bad adhoc jobs consume tons of memory, machine hangs
 - Large adhoc job prevented fairshare of resources
 - Impacts periodic pipeline jobs that affects company's revenue
- **Solution: split cluster into two**
 - PLATINUM Hadoop Cluster
 - High SLA, only approved jobs can run here
 - SILVER Hadoop cluster
 - Lower SLA
 - Optimize latency for small jobs
 - Optimize cluster utilization and fair-share for larger jobs
 - Resource aware scheduling (CPU and memory resources only)

America's Most Wanted (AMW)

■ Human Social System

- Small percentage of criminals in society
- Responsible for large percentage of crimes
- Repeat offenders
- Requires intervention by law enforcement
 - Three-strikes law



■ Machine Recidivism

- Small percentage of bad machines in cluster
- Responsible for large percentage of failures
- Requires intervention from automated tools
 - Weed them out
 - Escalate to vendor, or to internal hardware/kernel teams

- **What are their characteristics?**
 - “Repeat offenders”
 - These machines occasionally hang or processes tasks slowly
 - They issue more alarms
 - They require more intervention from our automated tools (reboots, reimages, restarting processes, etc.)
 - They generate more repair events
- **We are building a metric which takes all these “likely repeat offender” characteristics into account**

Analysis of Hardware Repair Events

- **Repair events mean “manual intervention”**
 - Machine cannot be fixed by automated tools
 - Requires a “touch” by a datacenter technician
 - Means we have exhausted our automated remedies
 - Could be an actual hardware problem
 - Could also be configuration error
- **Repair events are usually precipitated or accompanied by some loss of functionality, speed, or data**
 - Slow tasktrackers can substantially reduce job completion rates
 - Datanodes with failing disks should be decommissioned
- **Is Hadoop inherently “tougher” on machines than other applications?**

Hardware Repair Rates Across Tiers

- We studied repair rates for a large sample of servers in Facebook's infrastructure
- Repair rates and frequencies are highly tier-dependent

Tier	Machines in repair at least once	Machines never in repair
Hadoop	18%	82%
Web	70%	30%
Database	8%	92%
Photo Storage	15%	85%

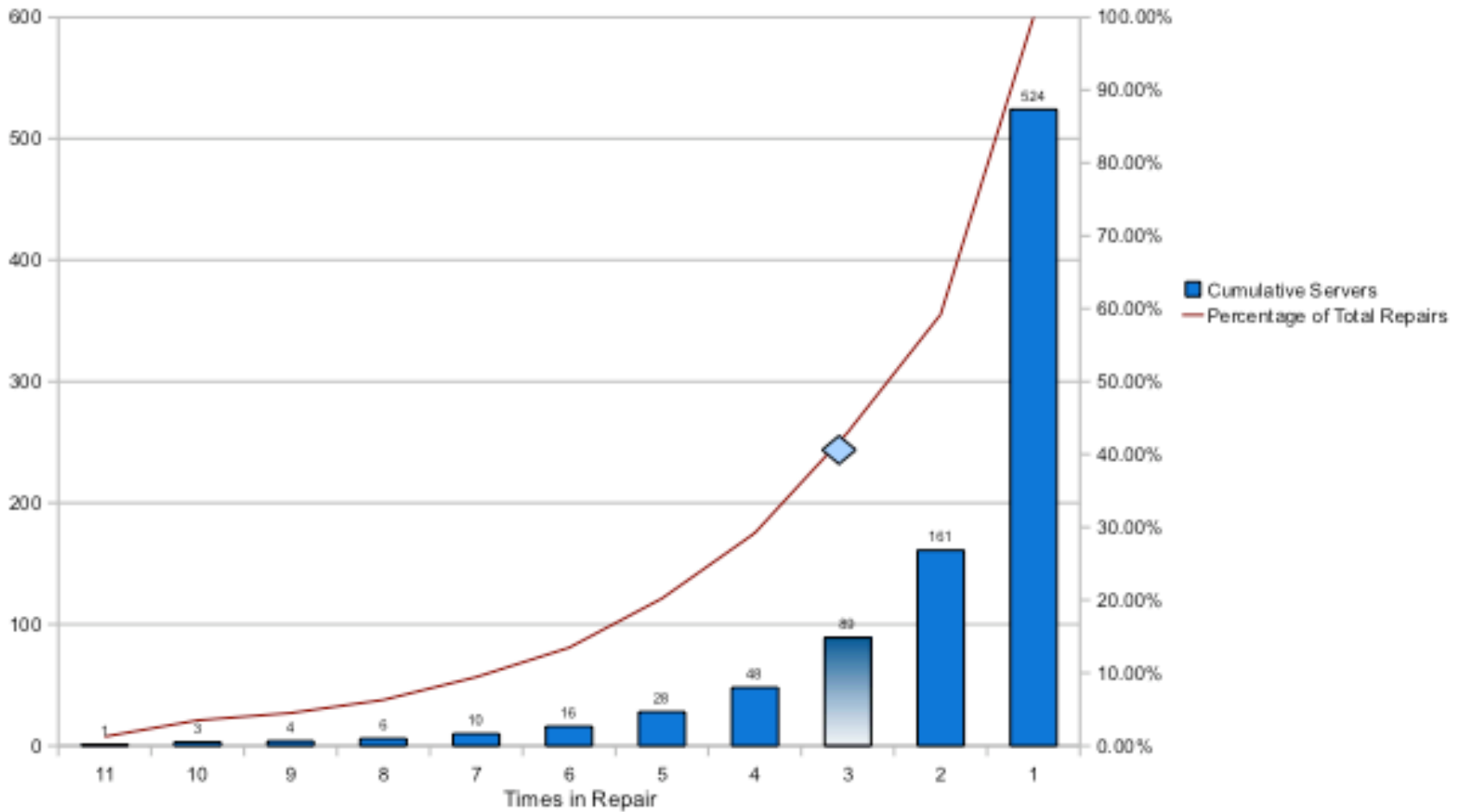
AMW Work in Progress

- **Develop comprehensive scores for tiers and SKU's**
- **Identify repeat offenders earlier**
 - get them out of the system
- **Gather better data on root causes of failures**
 - especially multiple failures

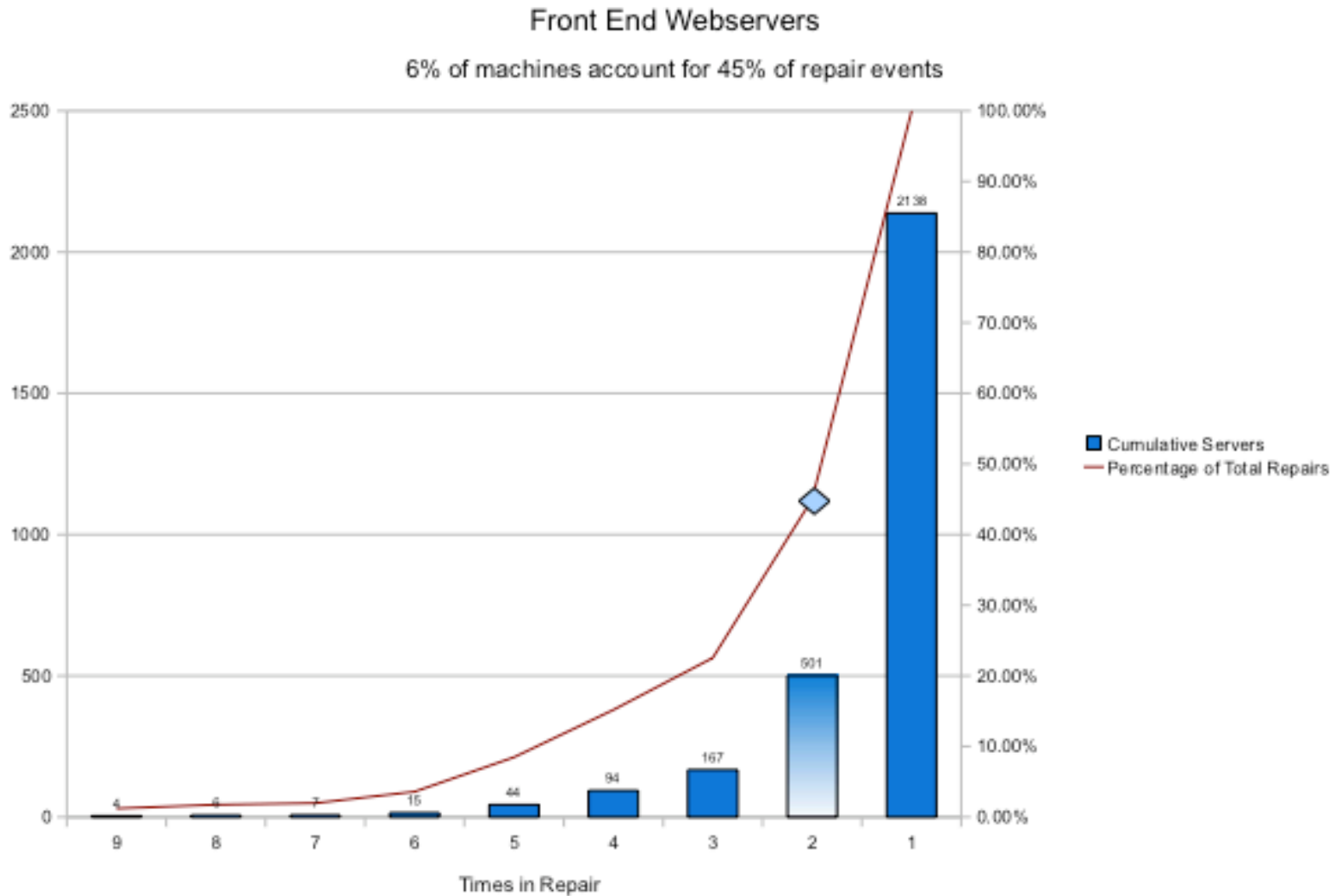
Hadoop Datanode/Tasktracker Tier

Hadoop Datanodes

3% of machines account for 43% of repair events



Front End Webservers Tier



Database Tier

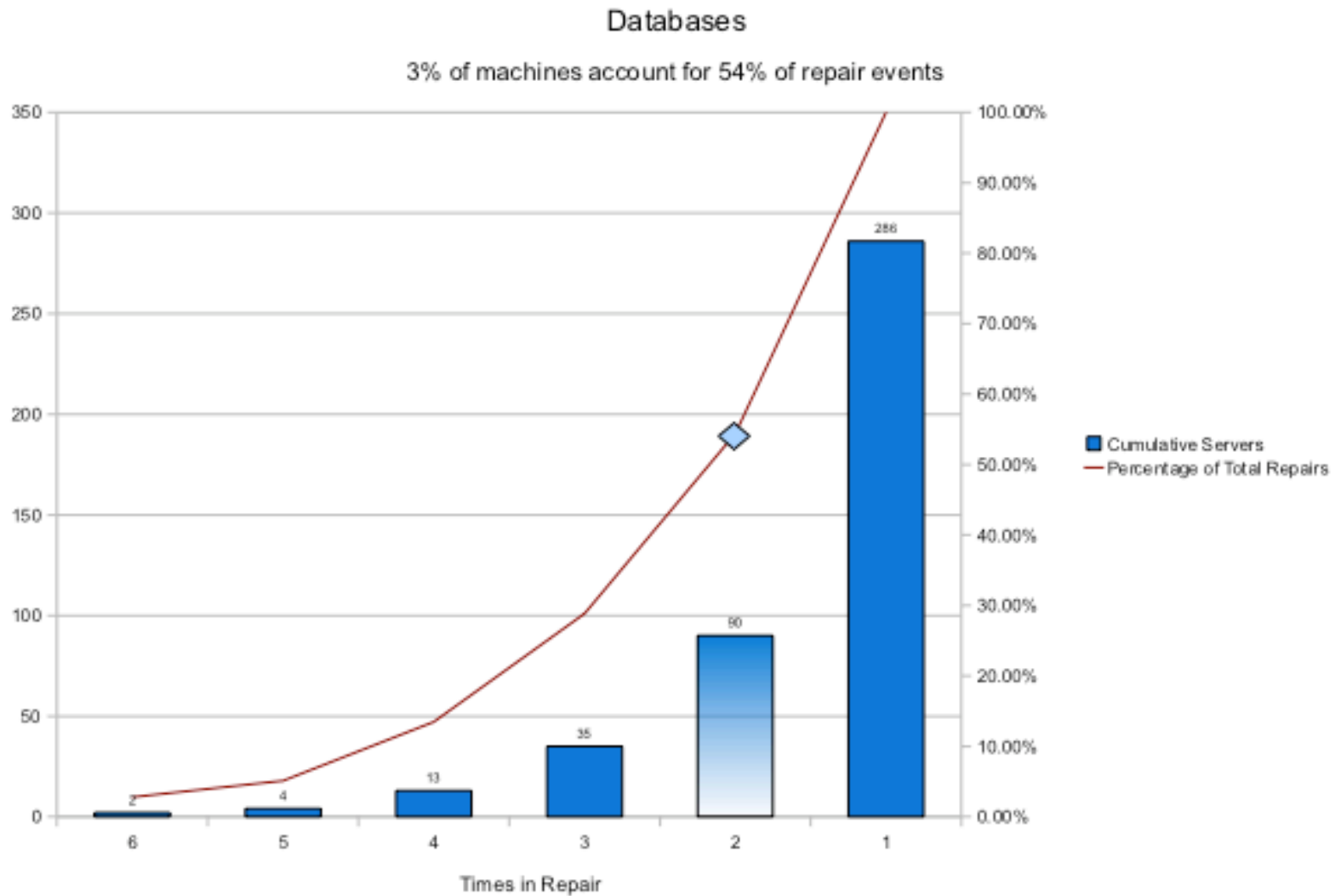
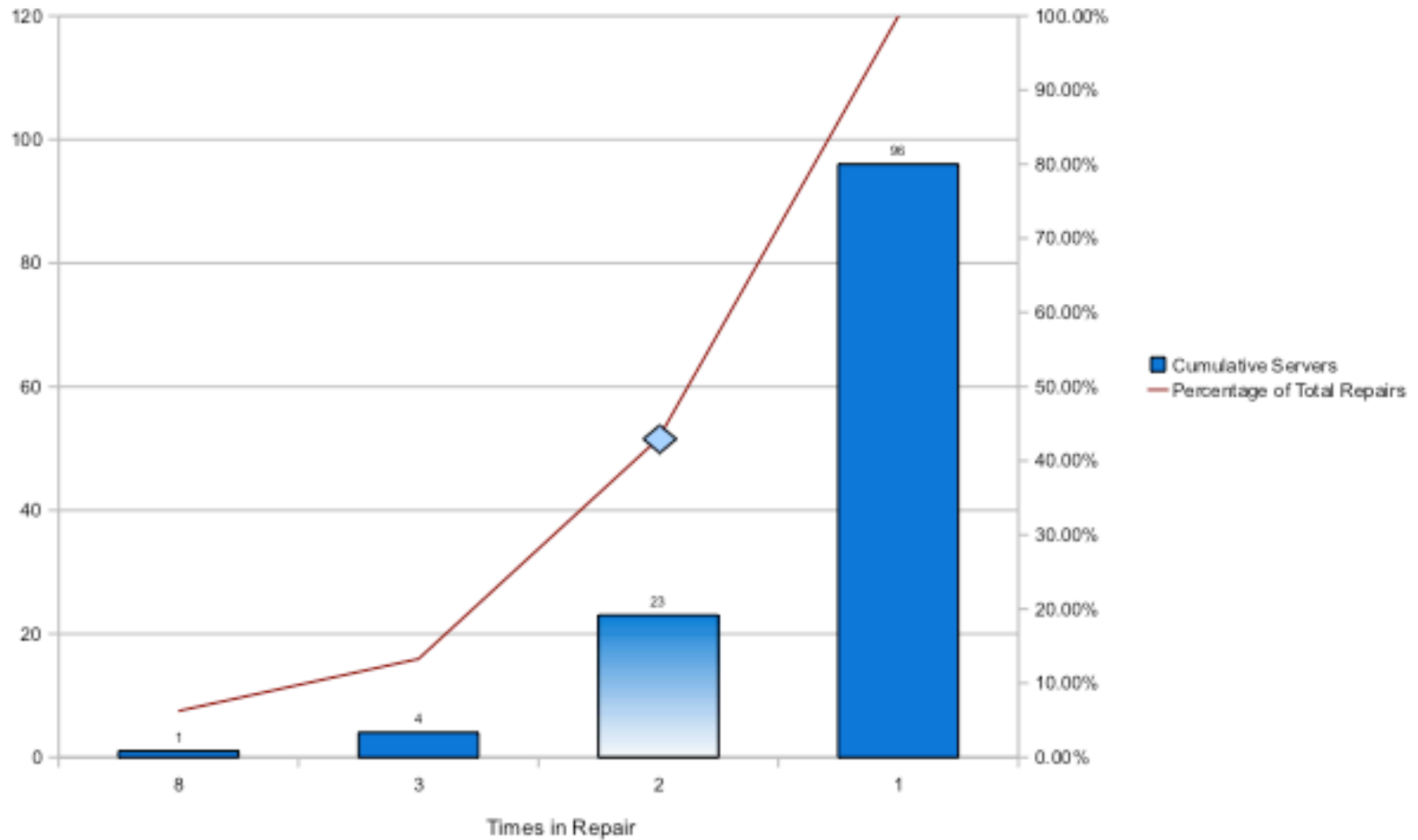


Photo Storage Tier

Photo Storage

3% of machines account for 43% of repair events



What happens if Disaster Strikes?

- **What if there is an oil-spill in California?**
 - Actually, earthquakes are more likely!
 - Entire cluster can be out of service
 - Backing up 20PB is impossible
- **Separate a small storage & compute cluster**
 - Select small subset of data from production warehouse
 - Move this to a remote geo (work-in-progress)
 - Poor man's Disaster Recovery Solution (DR)

Conclusion

- Failure Analysis
 - Monitoring of failures is a must for distributed systems
 - Crowd Sourcing can lend a helping hand
 - Repeat offenders quickly quarantined
- More details
 - Facebook blog at <http://blog.facebook.com/>
 - Hadoop blog at <http://hadoopblog.blogspot.com/>