FAILURE DIAGNOSIS & VISUALIZATION FOR CLOUD COMPUTING

Jiaqi Tan, Soila Kavulya, Mike Kasick, Elmer Garduno, Xinghao Pan, Nathan Mickulicz, Rajeev Gandhi

Priya Narasimhan

Carnegie Mellon University

Automated Failure-Diagnosis

- Diagnosing the root-cause of failures
  - Creates major headaches for administrators
  - Worsens as scale and system complexity grows
- Goal: automate it and get proactive
  - Fault detection
  - Fault localization ("automated fingerpointing")
  - Problem visualization
- How: Instrumentation plus statistical analysis

Exploration of Fingerpointing

- Current target systems
  - *MapReduce / Hadoop*
    - [HotCloud 09, HotMetrics 09, WASL 08, SysML 08, NOMS 10, ISSRE 09, CCGrid 10, ICDCS 10, ACM CHIMIT 11]
  - *PVFS, Lustre, GPFS*
    - High-performance parallel file/storage system [HotDep 09, USENIX FAST 10, HotDep 10]
    - Real-world production clusters: Intrepid, Argonne National Labs
  - *VoIP Systems*
    - Real-world telecom system [SLAML 11, ACM OSR 11]
- Studied
  - Various types of problems
  - Various kinds of instrumentation
  - Various kinds of data-analysis techniques

Goals & Non-Goals

- Diagnose faulty node to user or system administrator
- Target production environments
  - Use Hadoop logs as-is (*white-box strategy*)
  - Use OS-level metrics (*black-box strategy*)
- Work for various workloads and under workload changes
- Support online and offline diagnosis
- Enable visualization of job progress for root-cause analysis

- Non-goals (for now)
  - Tracing problem down to the offending line of code

Target Hadoop Clusters

- 4000-processor Yahoo!'s M45 cluster
  - Production environment (managed by Yahoo!)
  - Offered to CMU as free cloud-computing resource
  - Diverse kinds of real workloads, problems in the wild
    - Massive machine-learning, language/machine-translation
  - Have harvested all logs and OS data each week for 2 years
- 100-node Amazon's EC2 cluster
  - Production environment (managed by Amazon)
  - Commercial, pay-as-you-use cloud-computing resource
  - Workloads under our control, problems injected by us
    - gridmix, nutch, pig, sort, randwriter
  - Can harvest logs and OS data of only our workloads

M45 Dataset Summary

Job Characteristics	
Log Period	April 2008 – April 2009
Number of jobs	Successful: 165948 (97%)
	Failed: 4100 (2.4%)
	Canceled: 1031 (0.6%)
Average job duration	20 minutes (max: 6.8 days)
Average nodes per job	27 (max: 299)
Dominant job patterns	Map-only jobs: 77%
	Map-mostly jobs: 14%

Job-Failure Statistics

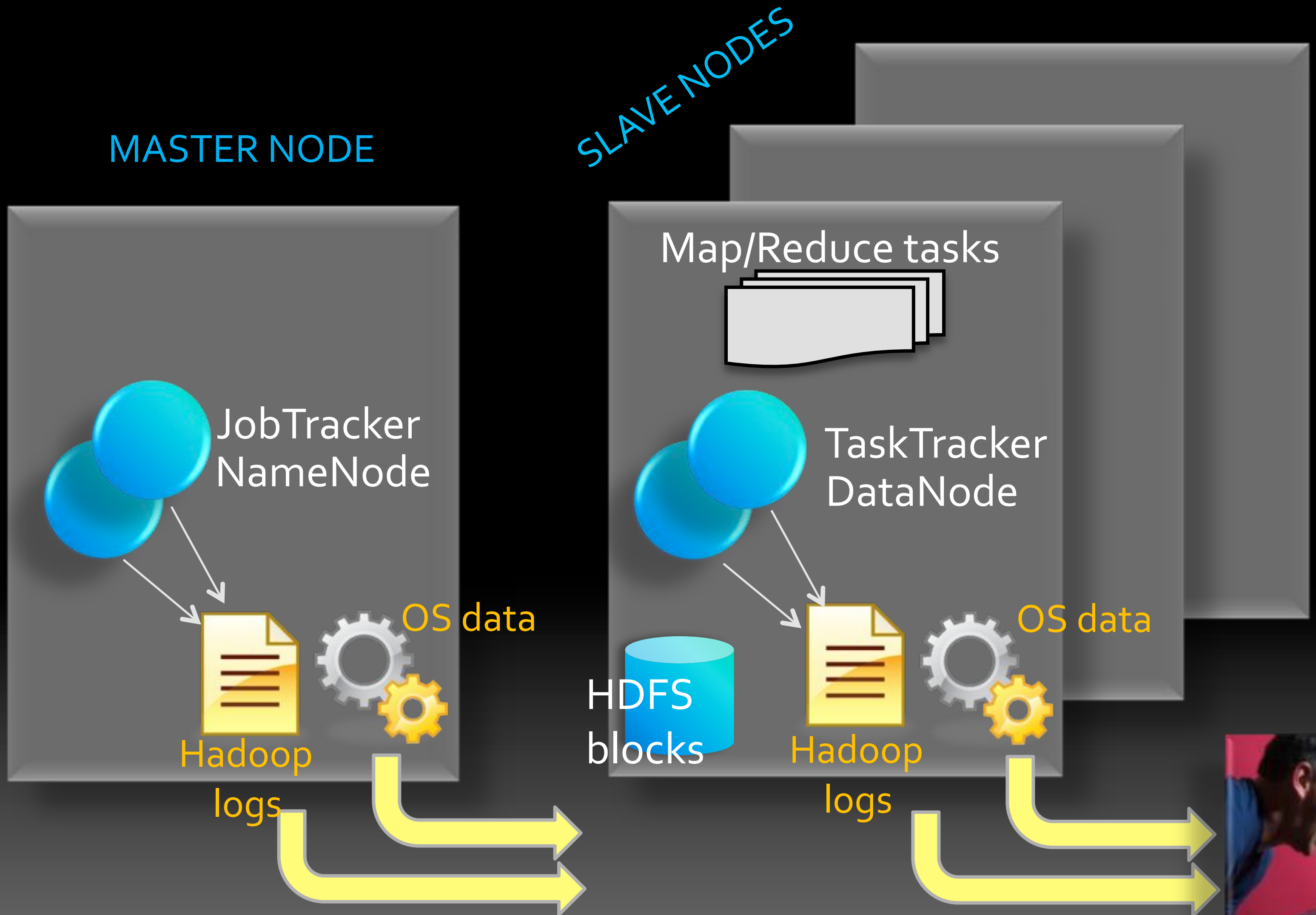
- Failures due to bad config detected quickly
  - But, 30% of failed jobs run for >1hr before aborting
- 5131 (3%) jobs failed or were canceled by user
  - Over 70% of these failures during the Map phase
  - 5% of these failures due to configuration problems, such as missing files, during job initialization
- Performance problems harder to identify
  - Lack of ground truth data
  - Identifying slow jobs through performance prediction [CCGrid 10]

Faults Studied

	Fault	Description
Resource contention	CPU hog	External process uses 70% of CPU
	Packet-loss	5% or 50% of incoming packets dropped
	Disk hog	20GB file repeatedly written to
	Disk full	Disk full
Application bugs Source: Hadoop JIRA	HADOOP-1036	Maps hang due to unhandled exception
	HADOOP-1152	Reduces fail while copying map output
	HADOOP-2080	Reduces fail due to incorrect checksum
	HADOOP-2051	Jobs hang due to unhandled exception
	HADOOP-1255	Infinite loop at Nameode

Studied Hadoop Issue Tracker (JIRA) from Jan-Dec 2007

Hadoop: Instrumentation



How About Those Metrics?

- White-box metrics (from Hadoop logs)
  - Event-driven (based on Hadoop's activities)
  - *Durations*
    - Map-task durations, Reduce-task durations, ReduceCopy-durations, etc.
  - System-wide dependencies between tasks and data blocks
  - Heartbeat information: Heartbeat rates, Heartbeat-timestamp skew between the Master and Slave nodes
- Black-box metrics (from OS /proc & Ganglia)
  - 64 different time-driven metrics (sampled every second)
  - Memory used, context-switch rate, User-CPU usage, System-CPU usage, I/O wait time, run-queue size, number of bytes transmitted, number of bytes received, pages in, pages out, page faults

rika (Swahili), *noun*. peer, contemporary

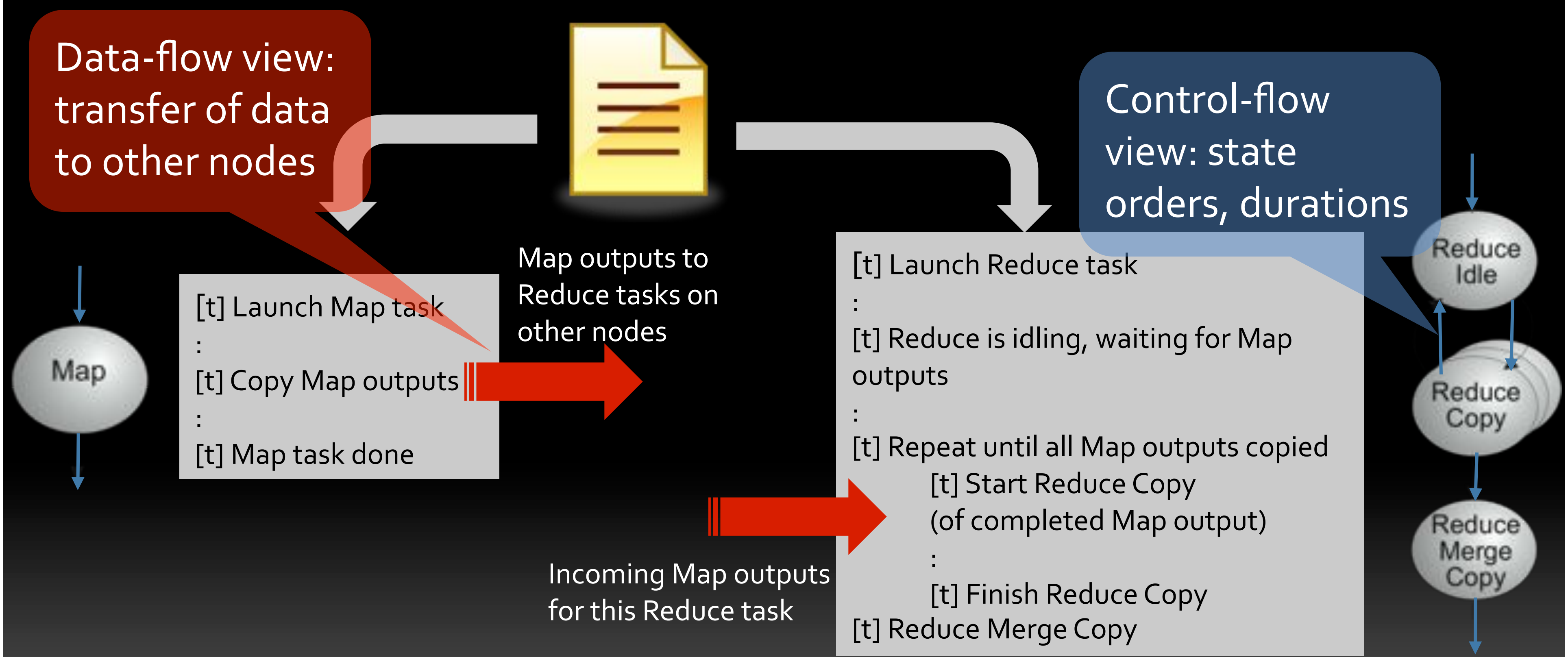
Intuition for Diagnosis

- Peer-comparison algorithm (others underway)
  - Slave nodes are "peers," doing *approximately similar* things for a given job
  - Compare the behavior of peers across the system
- Gather metrics of peers and extract statistics
  - For both black-box and white-box data
- Peer-compare histograms, means, etc., to determine the "odd-man out"
- Extended to cover heterogeneity within a job and its tasks

Log-Analysis Approach

- SALSA: Analyzing Logs as StAte Machines [*USENIX WASL 2008*]
- Extract state-machine views of execution from Hadoop logs
  - Distributed control-flow view of logs
  - Distributed data-flow view of logs
- Diagnose failures based on statistics of these extracted views
  - Control-flow based diagnosis
  - Control-flow + data-flow based diagnosis
- Perform analysis incrementally so that we can support it online

Applying SALSA to Hadoop Logs



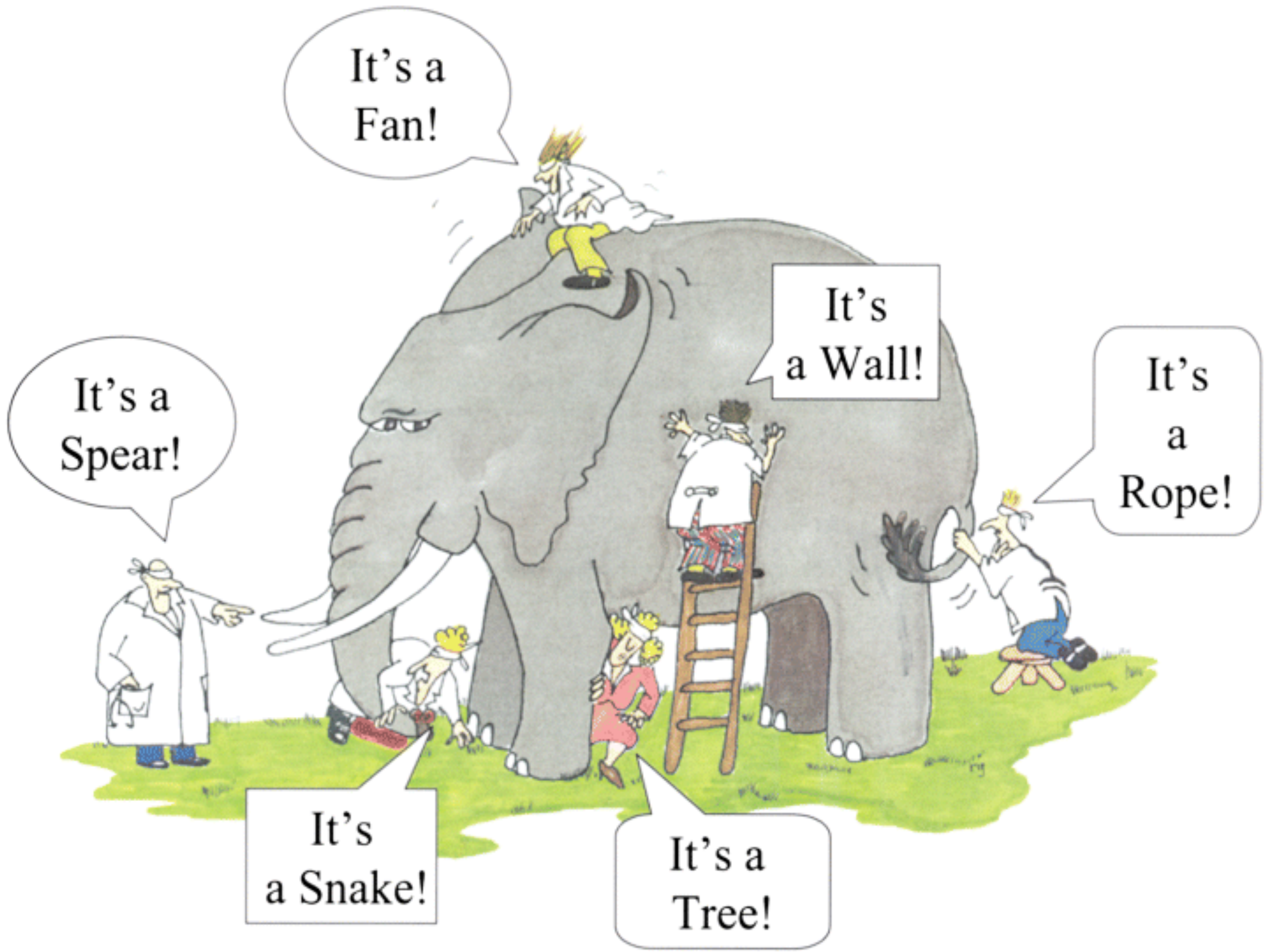
Distributed Control+Data Flow

- Distributed control-flow
  - Causal flow of task execution across cluster nodes, i.e., Reduces waiting on Maps via Shuffles
- Distributed data-flow
  - Data paths of Map outputs shuffled to Reduces
  - HDFS data blocks read into and written out of jobs
- Job-centric causal flow: Fused control+data flows
  - Correlate paths of data and execution
  - Create conjoined causal paths from data source before, to data destination after, processing

On the Black-Box Data Side…

- Analyze black-box data with similar intuition
  - Example method: Derive PDFs, use clustering
    - Distinct behavior profiles of metric correlations
  - Compare distance between histograms across nodes
  - Technique called Ganesha [*HotMetrics 2009*]
- Analyze heartbeat traffic
  - Compare heartbeat durations across nodes
  - Compare heartbeat-timestamp skews across nodes

Different metrics, different viewpoints, different algorithms



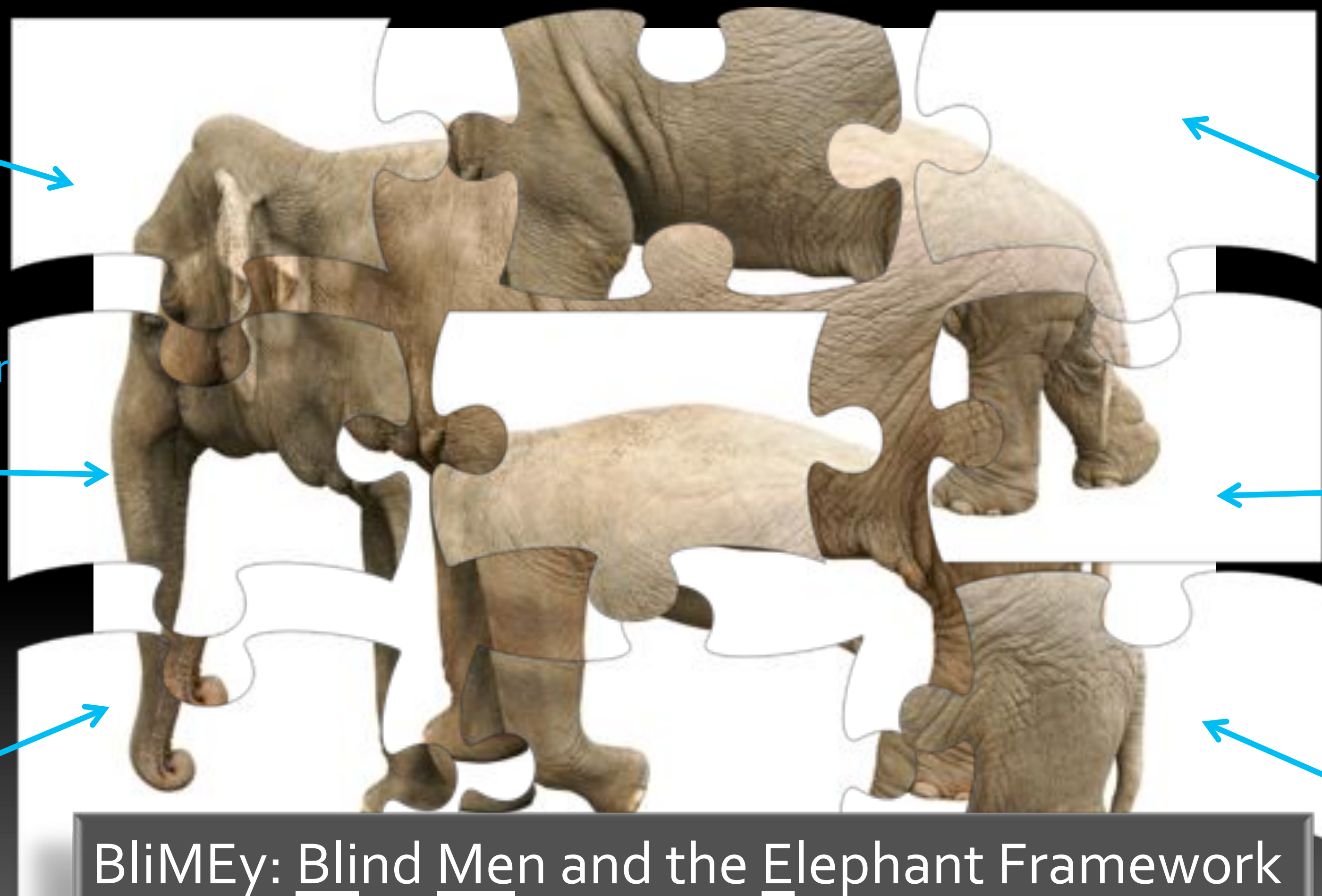
It's a Fan!
It's a Wall!
It's a Spear!
It's a Rope!
It's a Snake!
It's a Tree!

Piecing the Elephant Together

JobTracker Durations views

TaskTracker Durations views

Job-centric data flows



TaskTracker heartbeat timestamps

JobTracker heartbeat timestamps

Black-box resource usage

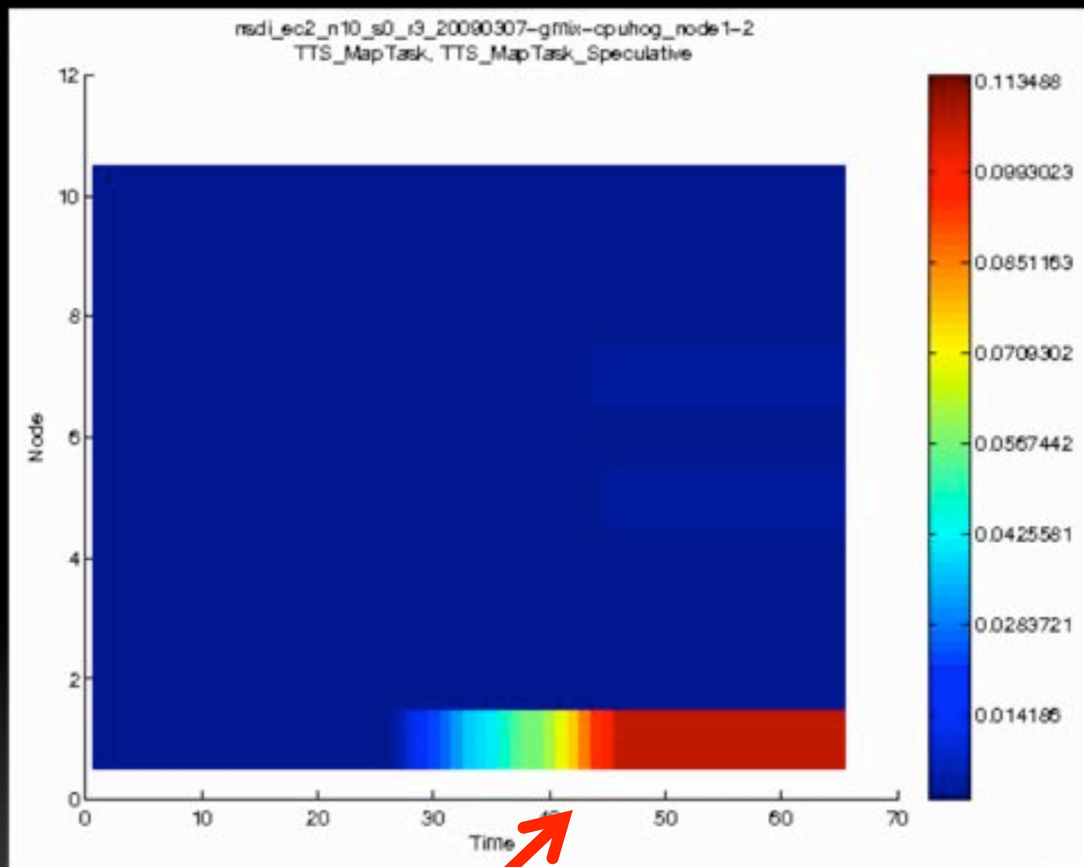
BliMEy: Blind Men and the Elephant Framework

[*CMU-CS-09-135*]

Visualization Tools

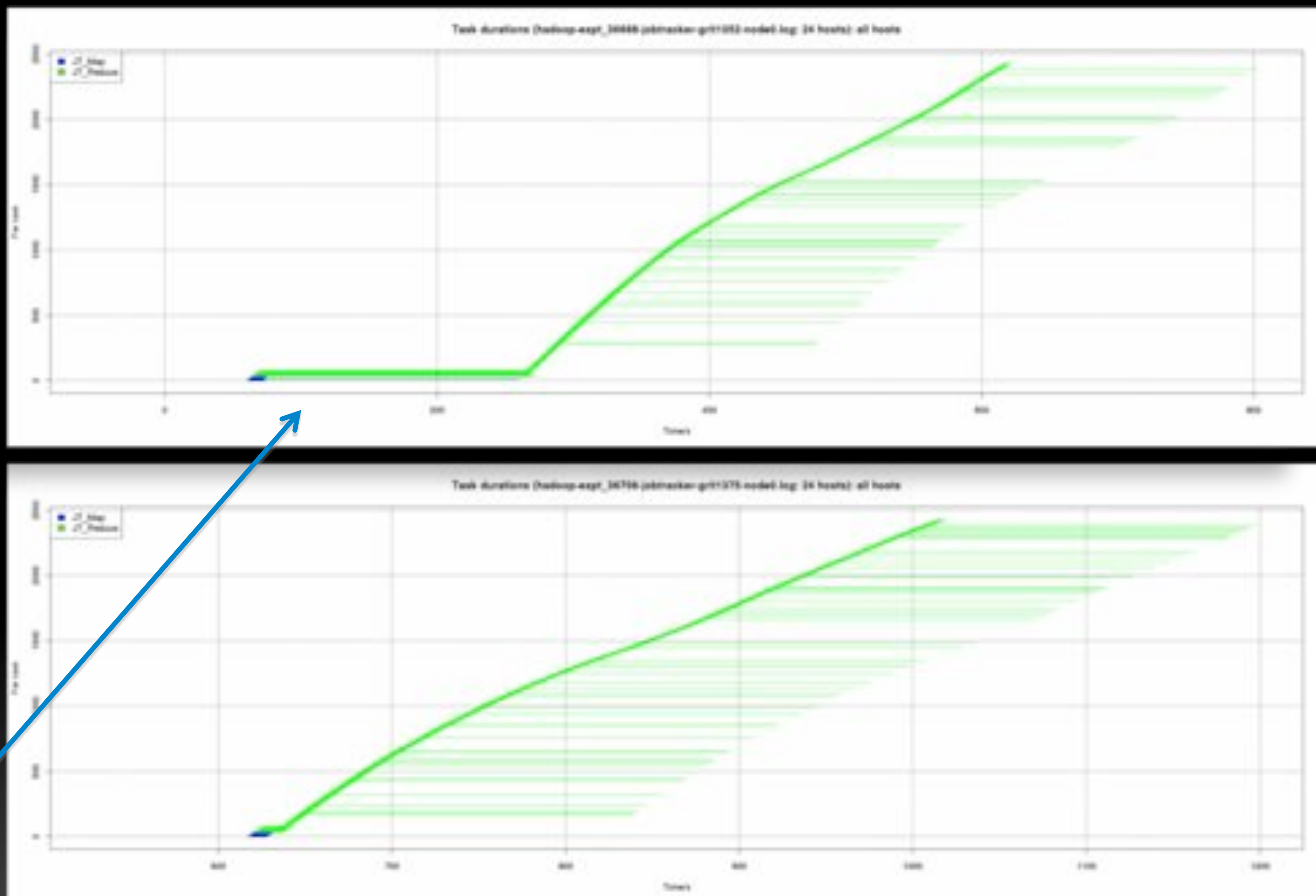
- To reveal system execution and trends of metrics for system administrators
  - Allows them to identify faulty nodes visually
- To reveal program/task execution and resource usage to developers
  - Allows them to spot issues that might assist them in restructuring their code/algorithms
- Developed visualization tools for HICC (Hadoop Infrastructure Care Center)
  - Available for public use via collaboration with Yahoo!

Sample Visualization (*heat-maps*)



CPU hog on node 1 visible
due to markedly (and increasingly)
different Map-task duration

Sample Visualization (*swim-lanes*)



Long-tailed Map task
delaying the overall
job-completion time

Ongoing Work

- Understanding the limits of black-box fingerpointing
  - What failures are outside the reach of a black-box approach?
  - What are the limits of "peer" comparison?
  - What other kinds of black-box instrumentation exist?
- Online diagnosis
  - Latency and scale in running these algorithms online
- Visualization
  - Helping system administrators visualize problem diagnosis
- Trade-offs
  - More instrumentation and more frequent data can improve accuracy of diagnosis, but at what performance cost?
- Virtualized environments
  - Do these environments help/hurt problem diagnosis?

DETOUR

Parallel File/Storage Systems

(DIFFERENT TARGET SYSTEM)

- Parallel file system ideally exhibits balanced load
  - Components should exhibit similar performance
  - Performance imbalance indicates underlying problem
- Intrepid: Located at Argonne National Laboratory
  - 128 GPFS NSD servers, 1152 LUNs across 16 controllers (4.5 PB)
  - Operators need tools to localize the problem
- Performed black-box analysis over three months
  - OS-level metric data
  - Diagnosed 8 independent disk failures (5 controller-failed, 3 operator-failed), 3 lost attachments between controller and server
- More details [HotDep 2009, HotDep 2010, USENIX FAST 2010]

DETOUR

Large-Scale Mobile Video

(DIFFERENT TARGET SYSTEM)

- Mobile streaming video in high-density environment
  - Tens of thousands of sports fans watching a replay overt Wi-Fi on their smartphones in a stadium
  - Smartphone clients, Wi-Fi network, back-end video servers + cloud
- YinzCam: Deployments in 10 NFL/NHL sports venues
  - Ranging from 20,000--80,000 fans inside each venue
  - Typical usage in a venue: 55% of the venue audience
  - Users face video latency, video quality issues, errors in overload
- Performed black-box and log analysis over 2 years
  - Platform-agnostic data, user analytics (across iOS, Android and RIM)
  - Diagnosed network-configuration issues, wireless-router issues, cloud resource-allocation problems

Summary

- Automated failure diagnosis
- Target systems: Hadoop, PVFS, Lustre, GPFS, VoIP
  - Real-world problems in the wild
  - Focus on production environments: M45, Intrepid, VoIP, YinzCam

- Additional details
  - USENIX WASL 2008 (white-box log analysis)
  - USENIX HotCloud 2009, ACM CHIMIT 2011, ICDCS 2010 (visualization)
  - HotMetrics 2009 & ISSRE 2009 (black-box metric analysis)
  - NOMS 2010 (black-box vs. white-box analyses)
  - CCGrid 2010 (M45 data analysis for performance prediction)
  - HotDep 2009 (system-call analysis for PVFS)
  - USENIX FAST 2010 (black-box analysis for PVFS & Lustre)
  - HotDep 2010 (behavior-based analysis for PVFS)
  - SLAML 2011 & ACM OSR 2011 (black-box analysis for VoIP)

Research Collaborators

- AT&T Labs
  - Matti Hiltunen, Kaustubh Joshi, Scott Daniels
- University of Lisboa
  - Antonio Casimiro, Diego Kreutz, Carlos Silva (Portugal Telecom)
- Argonne National Labs
  - Rob Ross, Sam Lang
- Intel Labs
  - Jason Campbell

FOR MORE INFORMATION:

HTTP://WWW.ECE.CMU.EDU/~FINGERPOINTING
PRIYA@CS.CMU.EDU