Structured Comparative Analysis of Systems Logs to Diagnose Performance Problems

Karthik Nagaraj

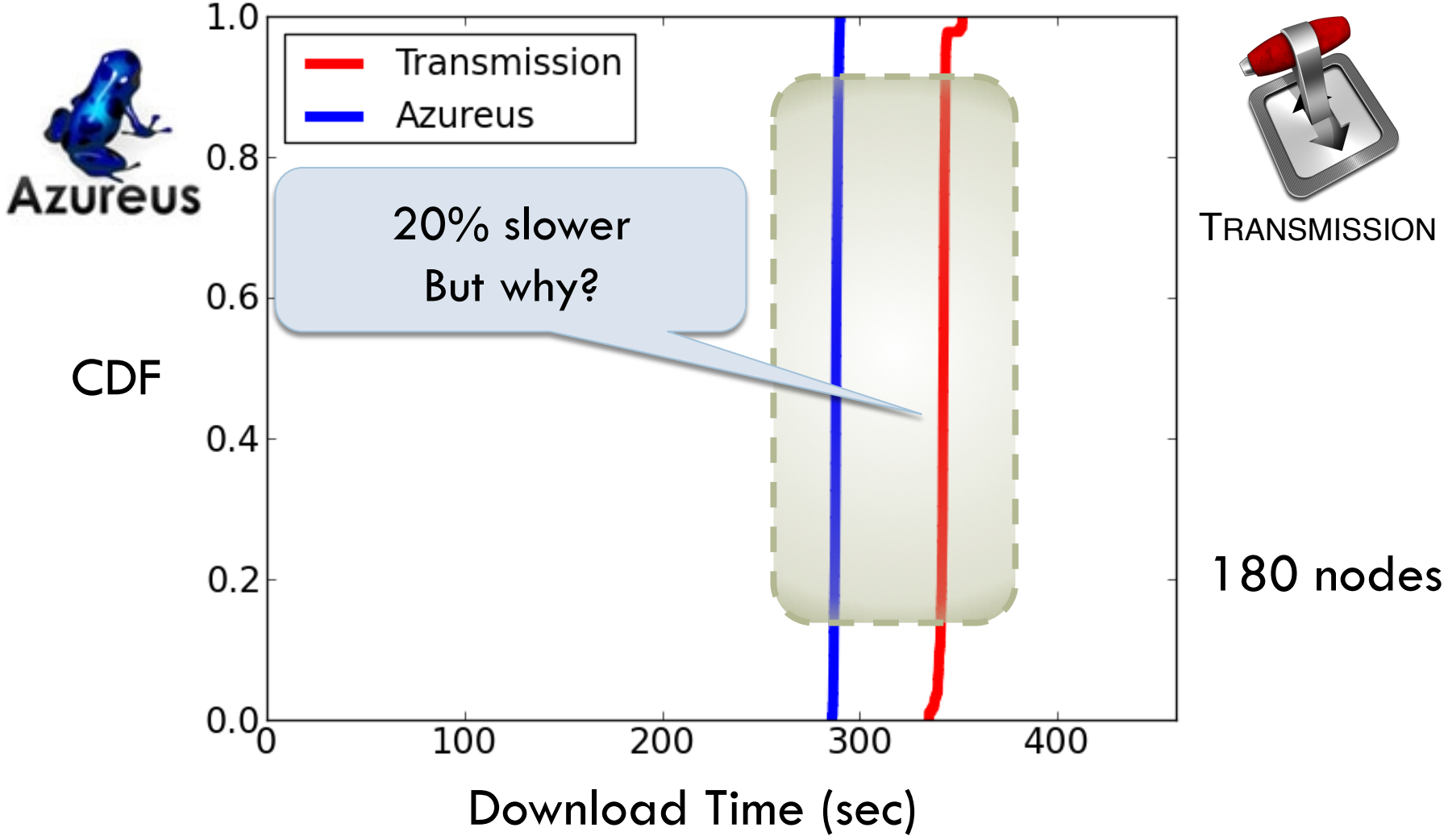
Charles Killian

Jennifer Neville

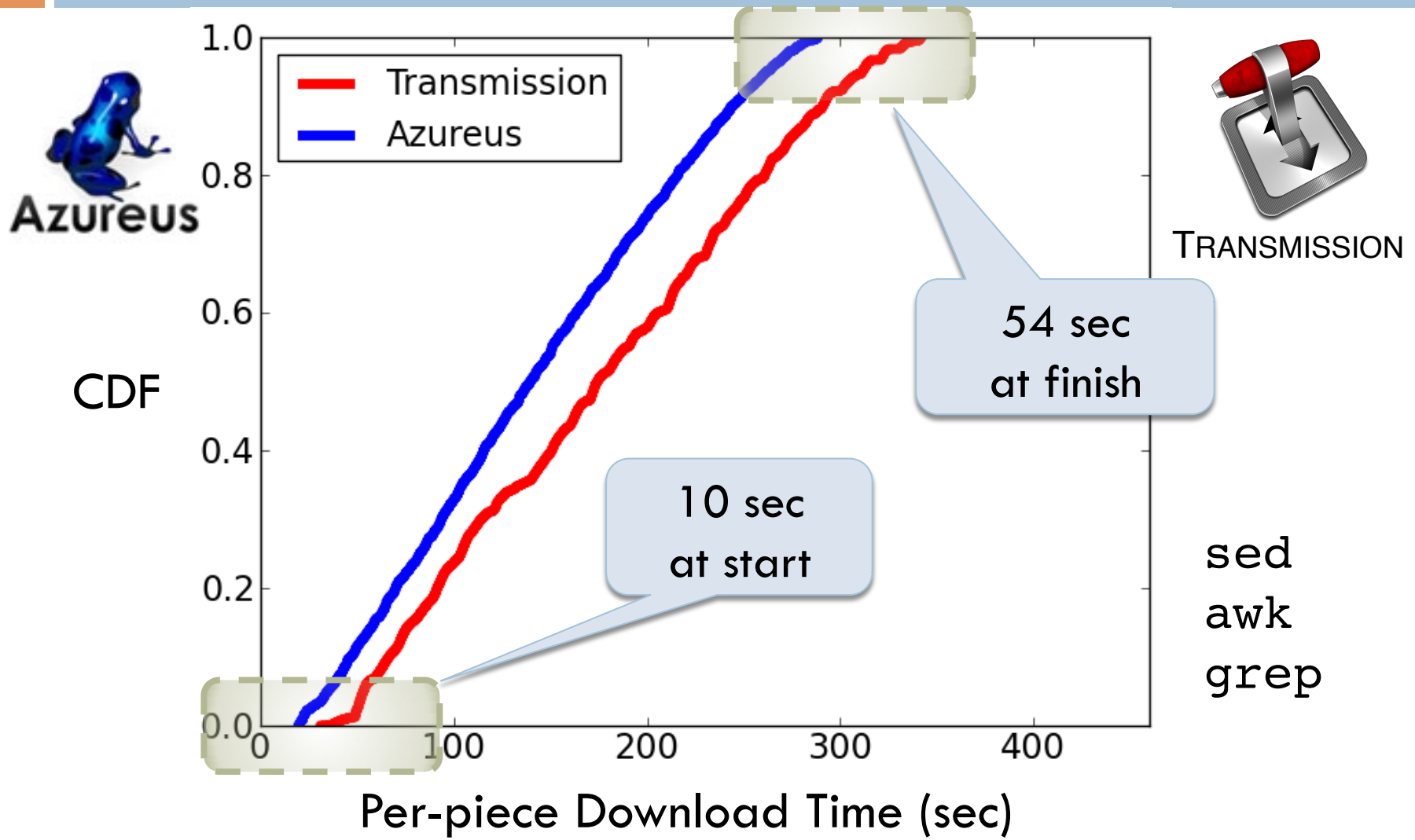
Computer Science

Purdue University

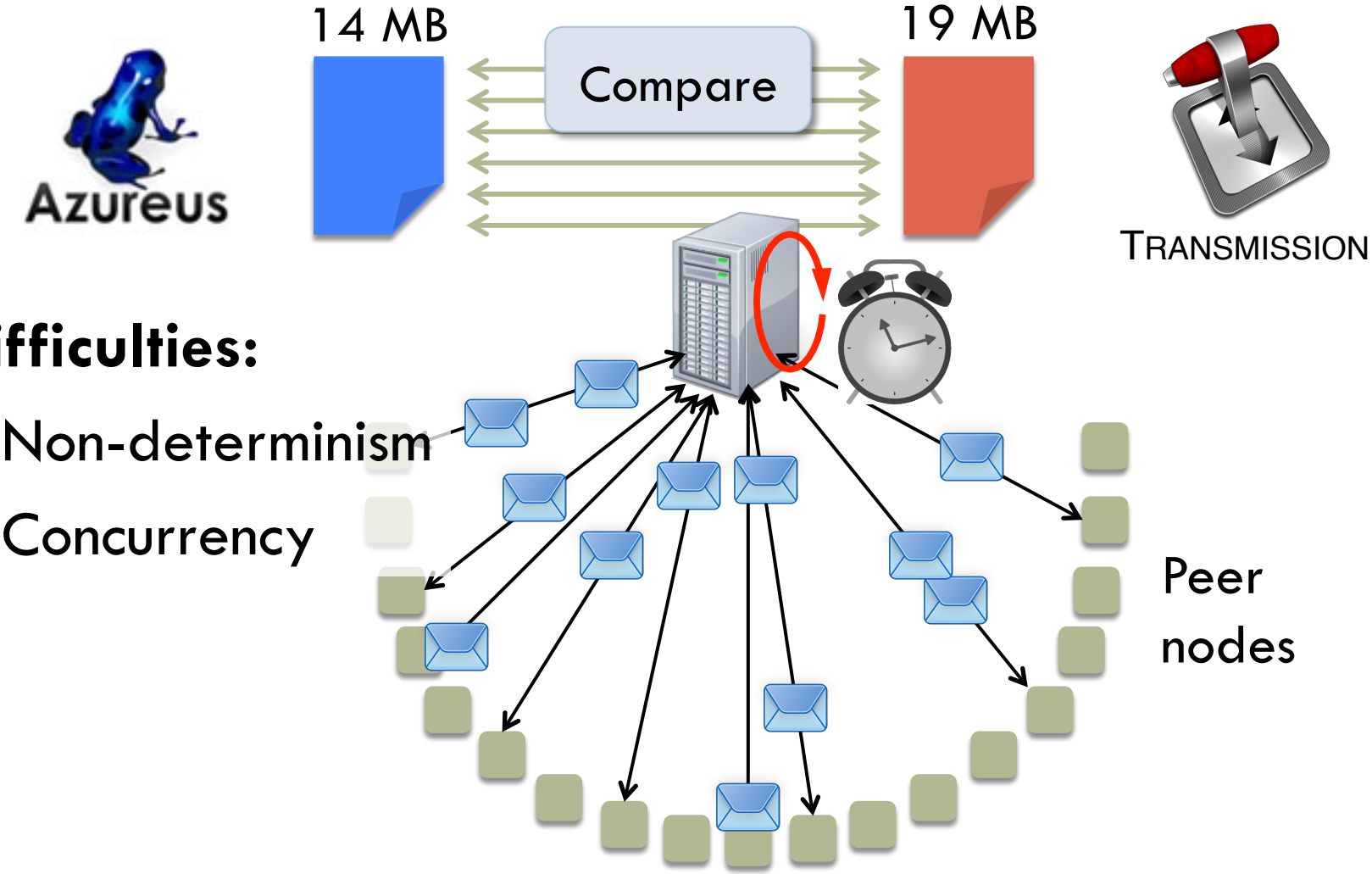
Distributed Systems: BitTorrent



BitTorrent: Compare Piece Times



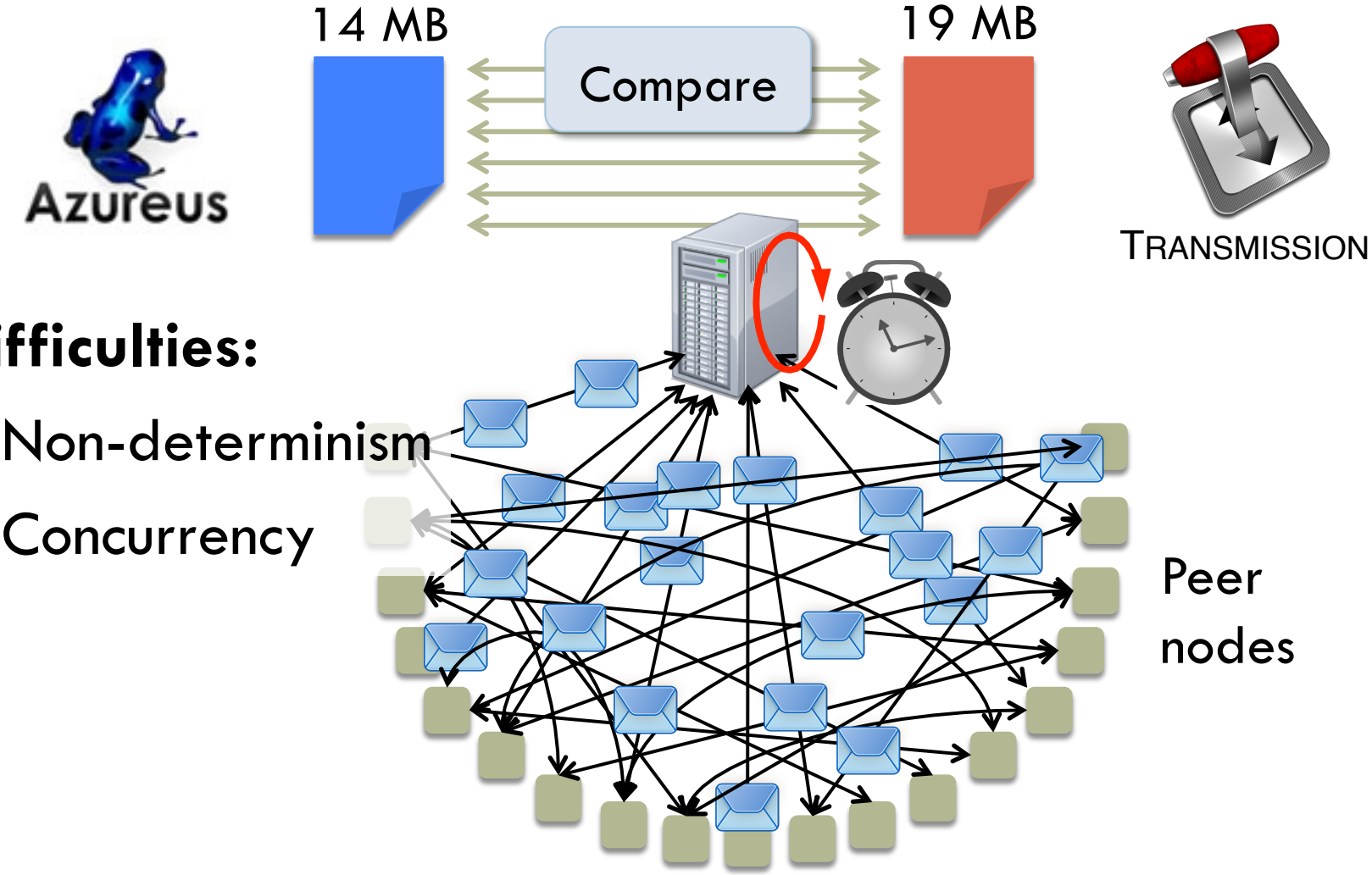
BitTorrent: Compare Logs



Difficulties:

- Non-determinism
- Concurrency

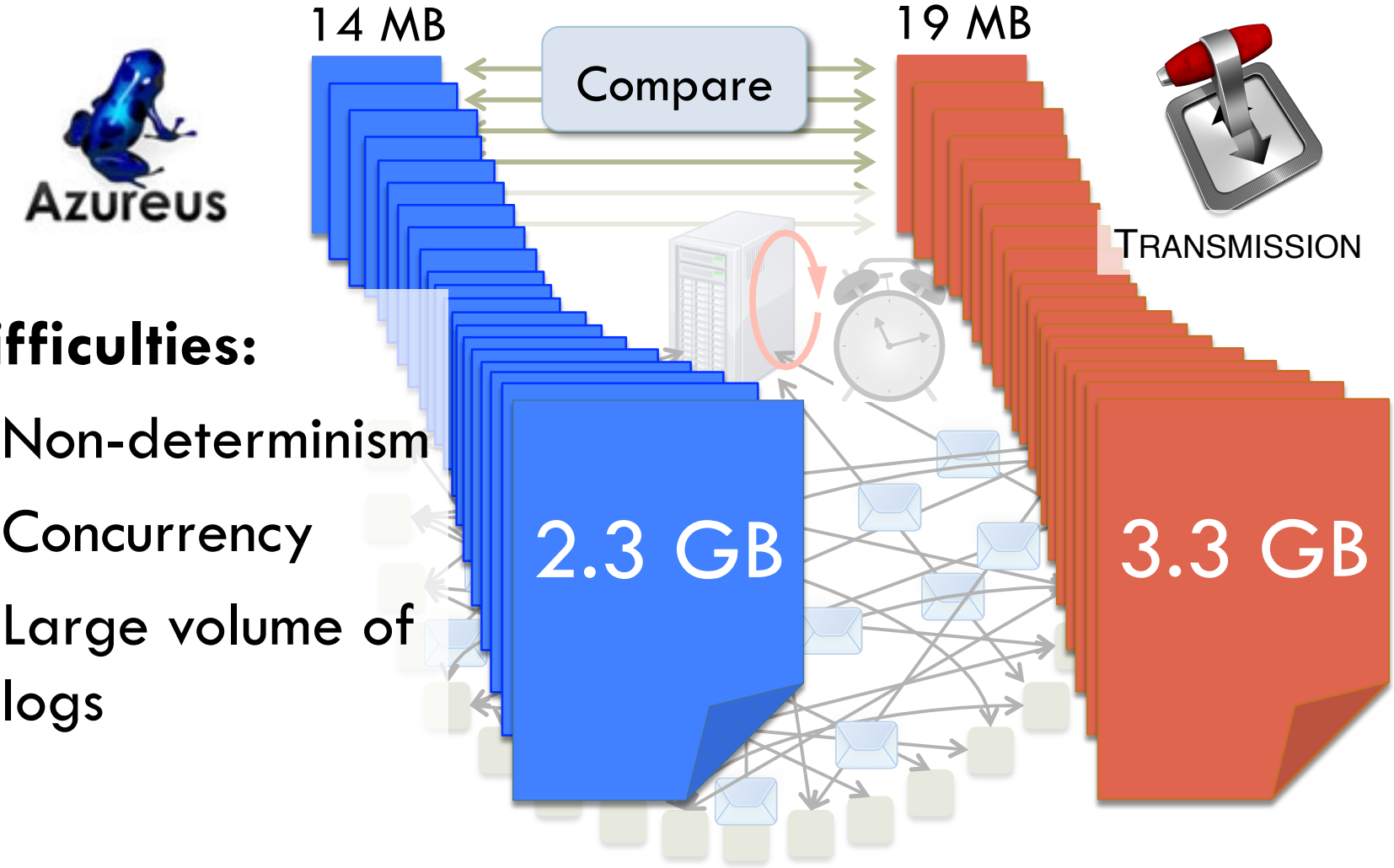
BitTorrent: Compare Logs



Difficulties:

- Non-determinism
- Concurrency

BitTorrent: Compare Logs



Difficulties:

- Non-determinism
- Concurrency
- Large volume of logs

Related Work: Performance Debugging

Detection

Diagnosis

Detecting System Problems
Mining Console Logs
[Xu et. al. SOSP 2009]

DISTALYZER

MACEPC
[Killian. et. al. FSE 2010]

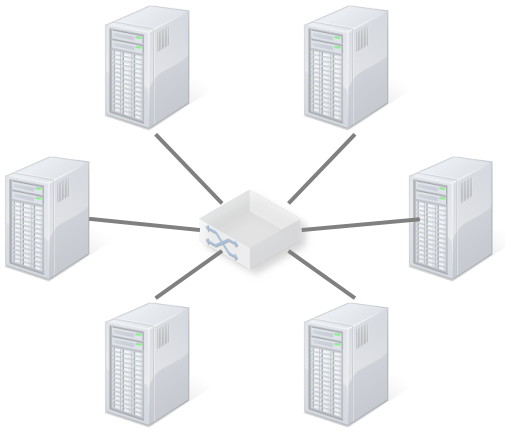
Request Processing

SPECTROSCOPE
[Sambasivan et. al. NSDI 2011]

MAGPIE
[Barham et. al. OSDI 2004]

Correlating instrumentation data to system states
[Cohen et. al. SOSP 2004]

Distalyzer Highlights



DISTALYZER diagnoses performance

Identifies:

- Significant divergences
- Impact on Performance

- Practical
  - Existing logs
- Automatic
  - Machine Learning
- For non-expert developers

Successful demonstration

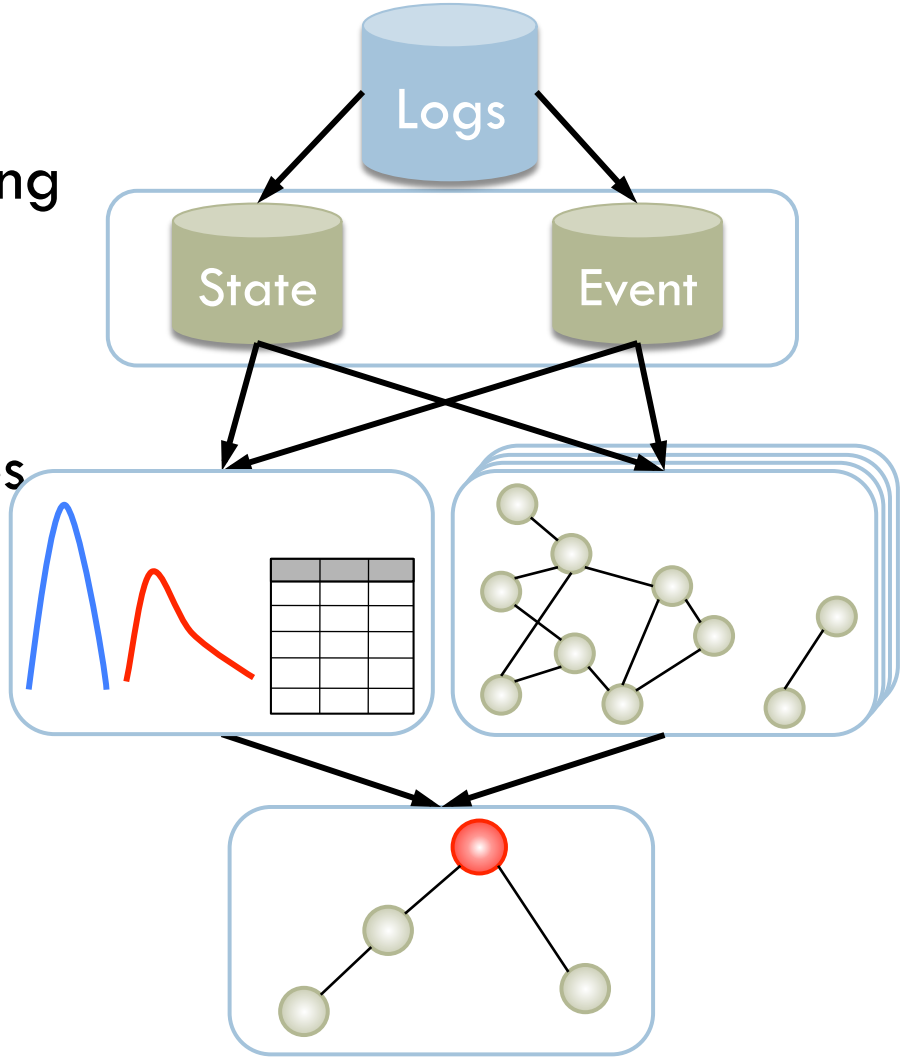
TRANSMISSION

APACHE HBASE

TRITONSORT

Design of Distalyzer

- Feature creation
  - Extract *behaviors* influencing overall performance
- Predictive modeling
  - Find *distinguishing* features
- Descriptive modeling
  - Component *interactions*
- Attention focusing
  - Root cause description



Code Instrumentation

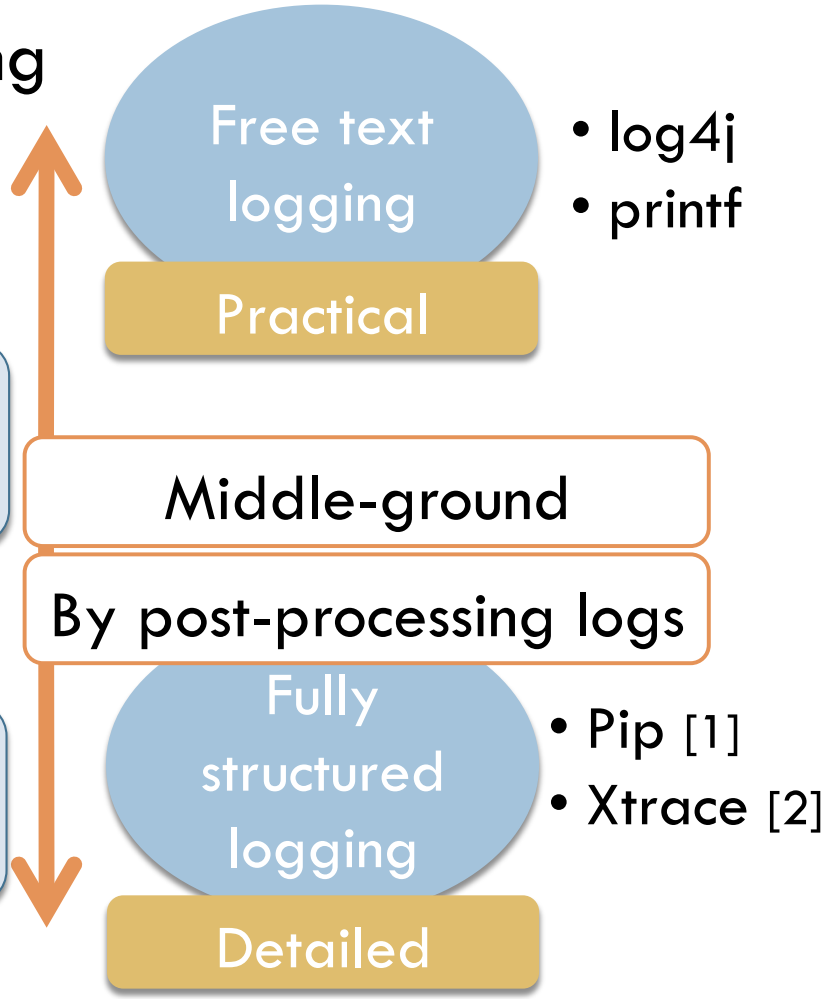
- Systems already have logging
- Categorize logs

Event logs: *some event happened*

```
1309116804.465942 10.0.0.74:8090
Received BT_PIECE for block 1058
```

State logs: *value of system variable*

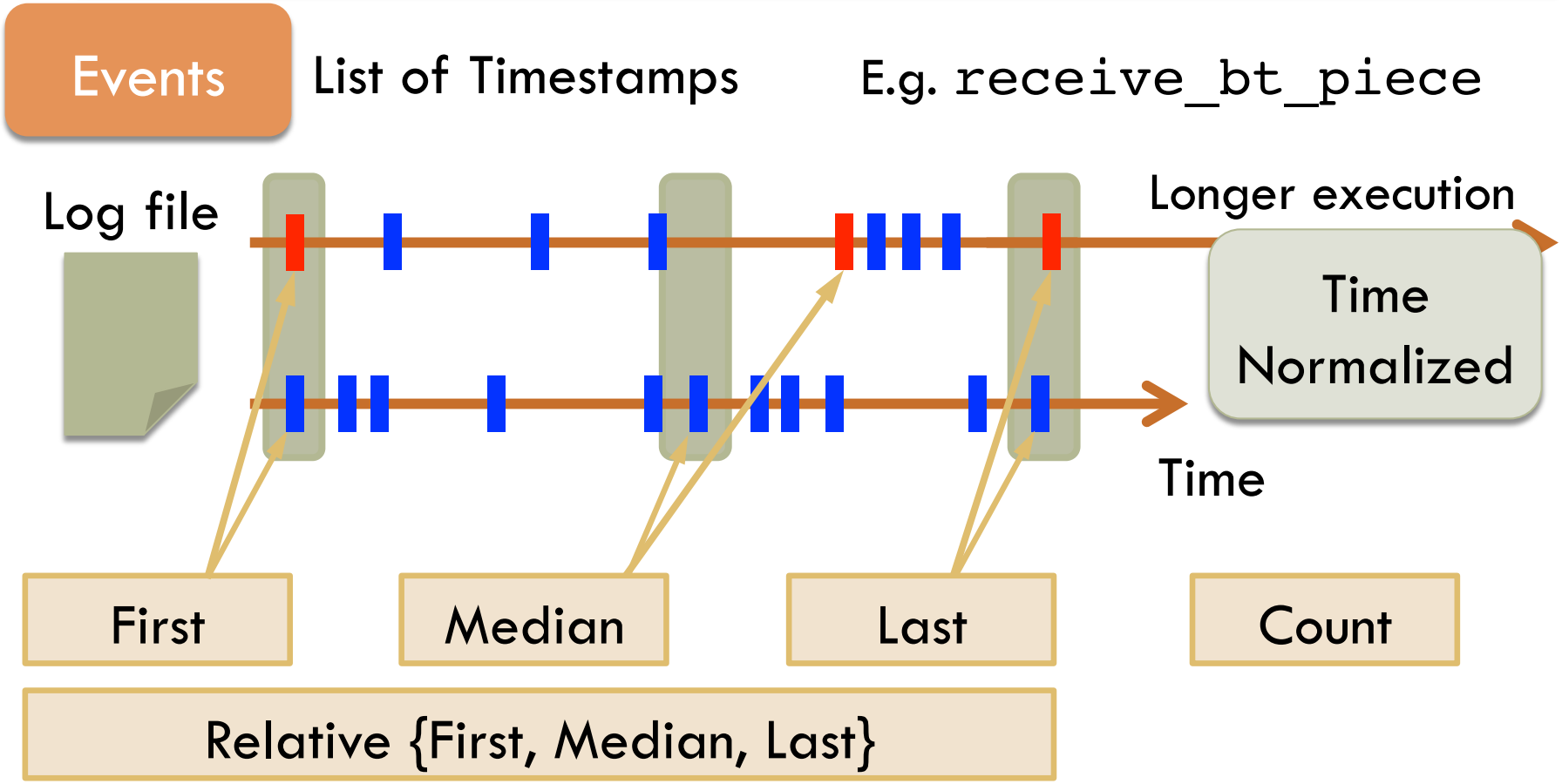
```
1309116871.387768 Progress: 25.05%
Speed: 237 KiB/s / 0 KiB/s
```



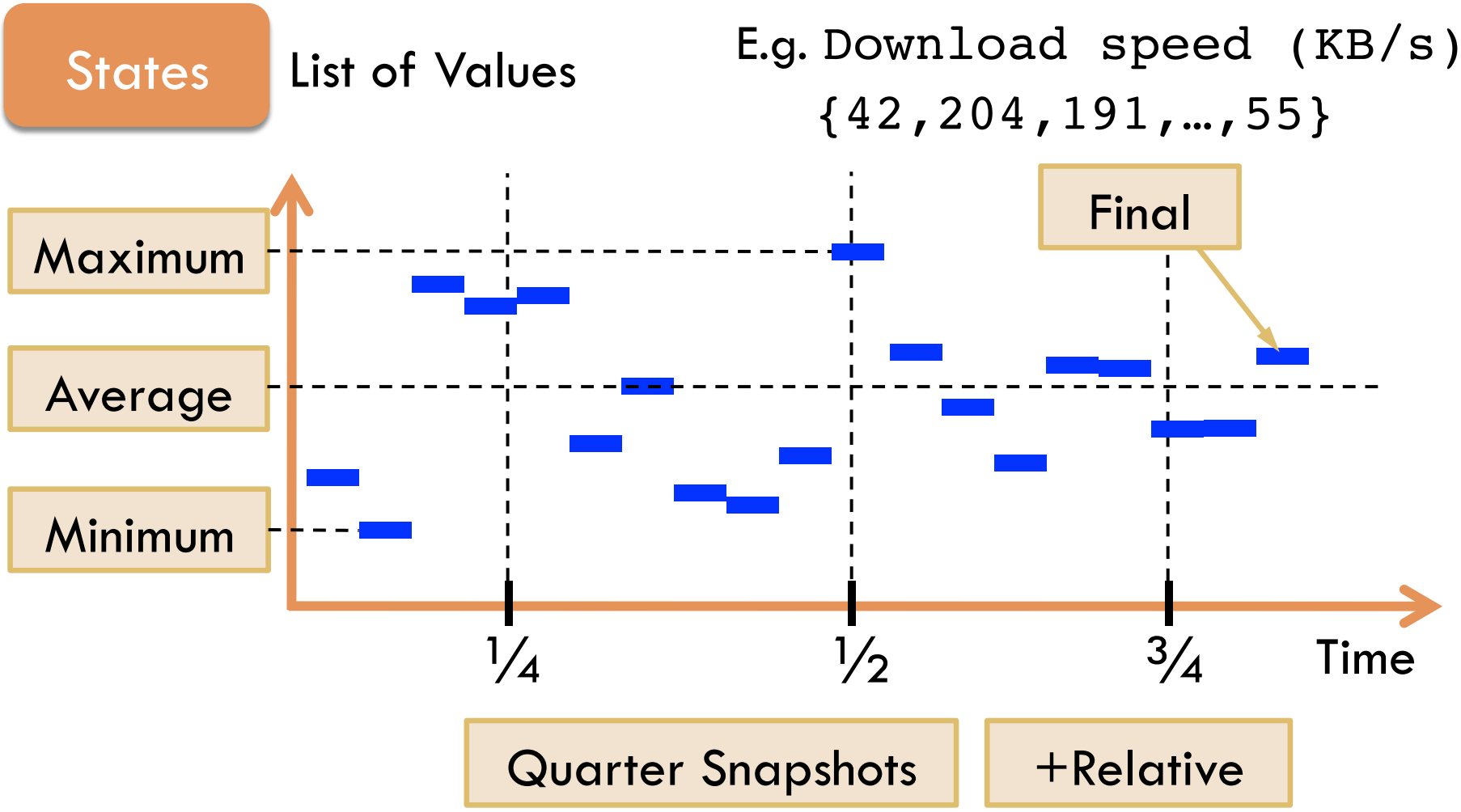
- [1]: Reynolds et. al. NSDI 2006
- [2]: Fonseca et. al. NSDI 2007

Design: Feature Creation

Capture ***summarized*** log characteristics of ***complete*** execution

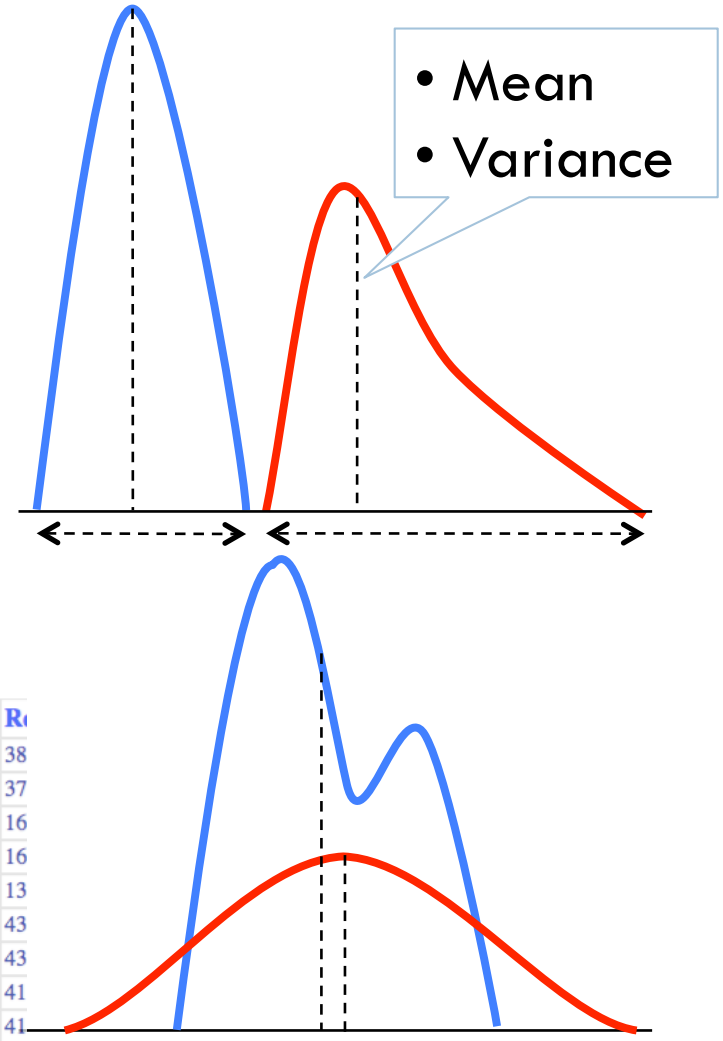


Design: Feature Creation



Design: Predictive Modeling

- Which features differ most?
- *E.g. First*(`recv_bt_piece`)
- *E.g. Max*(`Download speed`)
- Welch's T-Test
  - Significance: Really different?
  - Magnitude of difference
- Rank significant variables

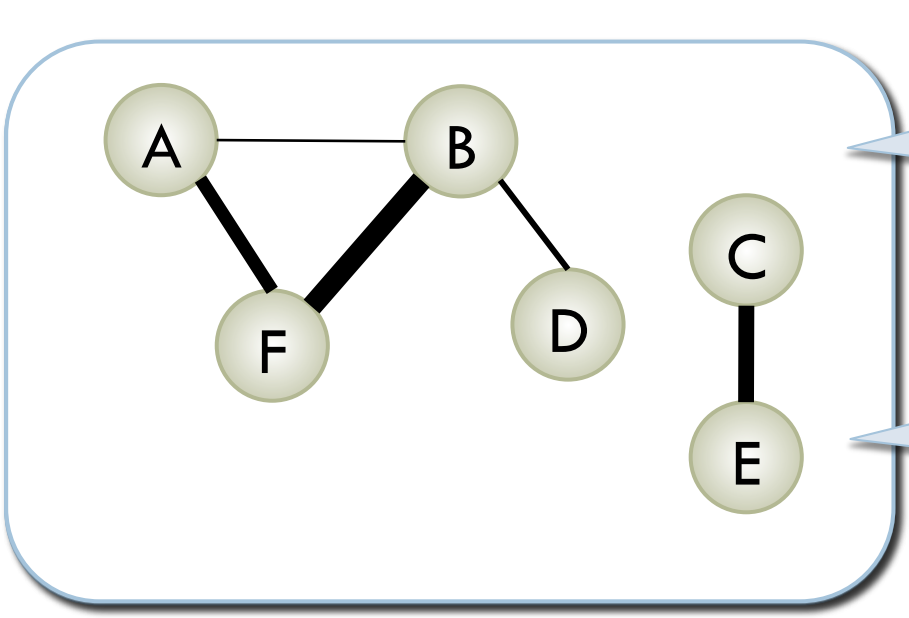


Variables	Count	First	Middle	Last	RelativeFirst	RelativeMiddle	R
Sent_Bt_Bitfield	79.83	-2723.97	-46.49	709.62	-964.50	-13.11	38
Recv_Bt_Bitfield	80.08	-2480.17	-46.65	600.61	-922.98	-13.20	37
Sent_Bt_Unchoke	207.79	5.01	53.87	55.30	34.04	149.48	16
Recv_Bt_Unchoke	137.96	1.88	53.32	55.12	26.66	146.96	16
Announce	-105.96	-0.02	-63.24	-159.33	18.76	38.87	13
Recv_Bt_Choke	20.85	-88.29	-64.78	-45.60	-77.15	-18.72	43
Recv_Bt_Have	176.79	-23.87	-42.08	-45.02	-18.54	8.75	43
Sent_Bt_Have	134.52	-26.35	-40.04	-40.09	-20.10	6.71	41
Sent_Bt_Choke	38.51	-72.77	-53.67	-37.52	-50.43	-11.88	41
Finish	0.00	-34.32	-34.32	-34.32	30.20	30.20	30
Sent_Bt_Not_Interested	31.14	-28.79	-24.93	-26.64	-22.80	13.79	43
Recv_Bt_Not_Interested	34.10	-29.35	-23.46	-29.53	-16.88	11.82	45

Nagaraj, Killian and Neville

Design: Descriptive Modeling

- How do system components interact?
- Dependency networks [1]



Dependency graph

Identify Dependencies

Strength *(edge)*

…re

…vents

A : recv_bt_piece

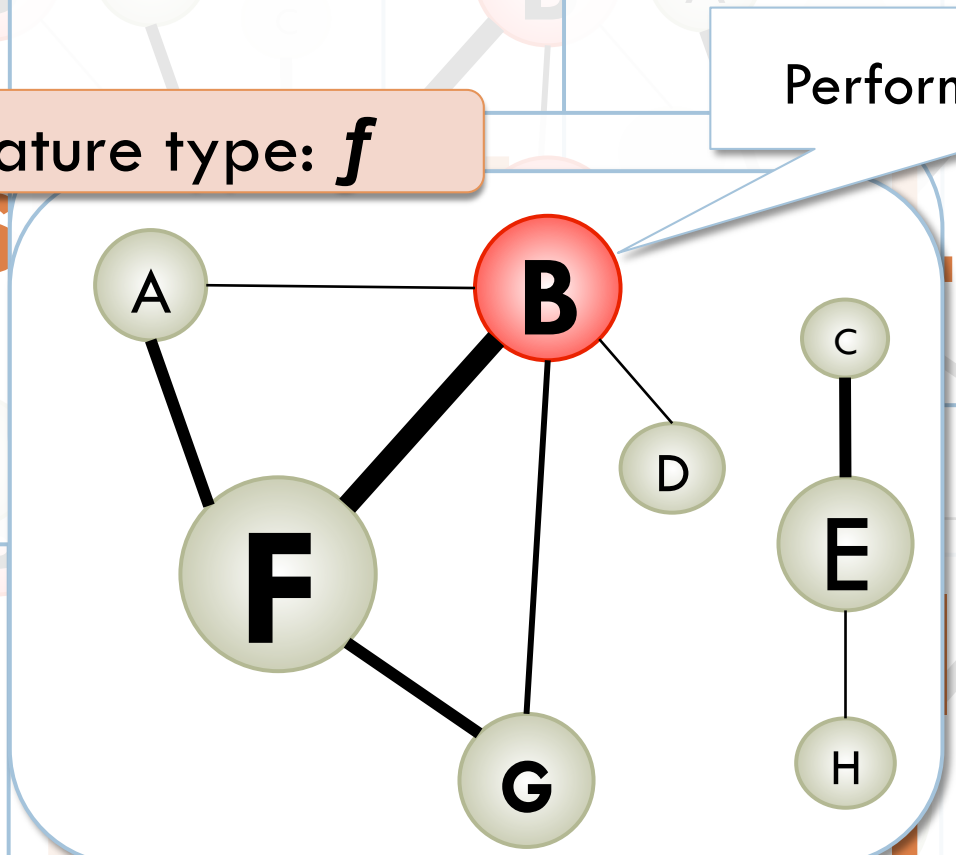
C : recv_bt_request

…

[1]: Heckerman et. al. JMLR 2001

Design: Attention Focusing

- Best smallest explanation of the problem



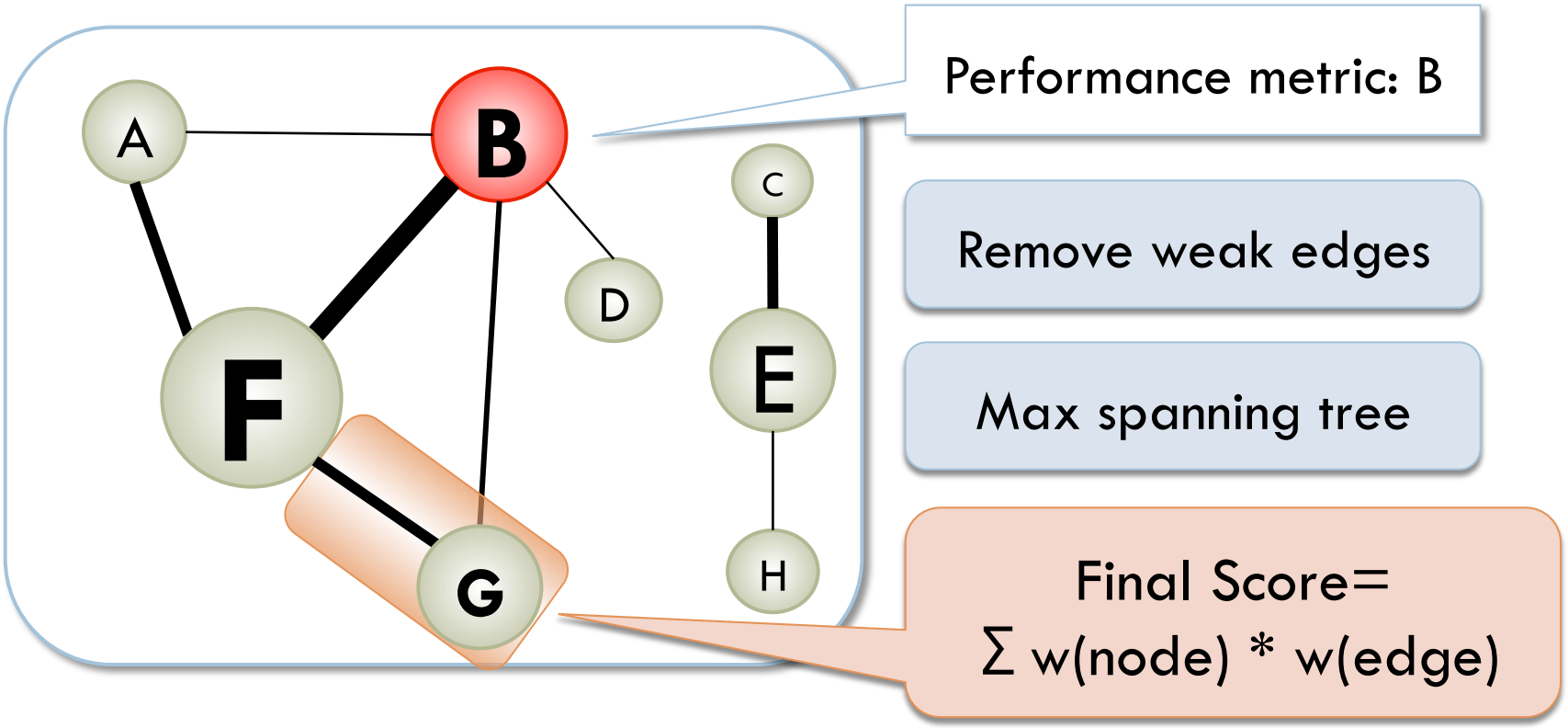
Firs ast

Relat ative

First Median Last

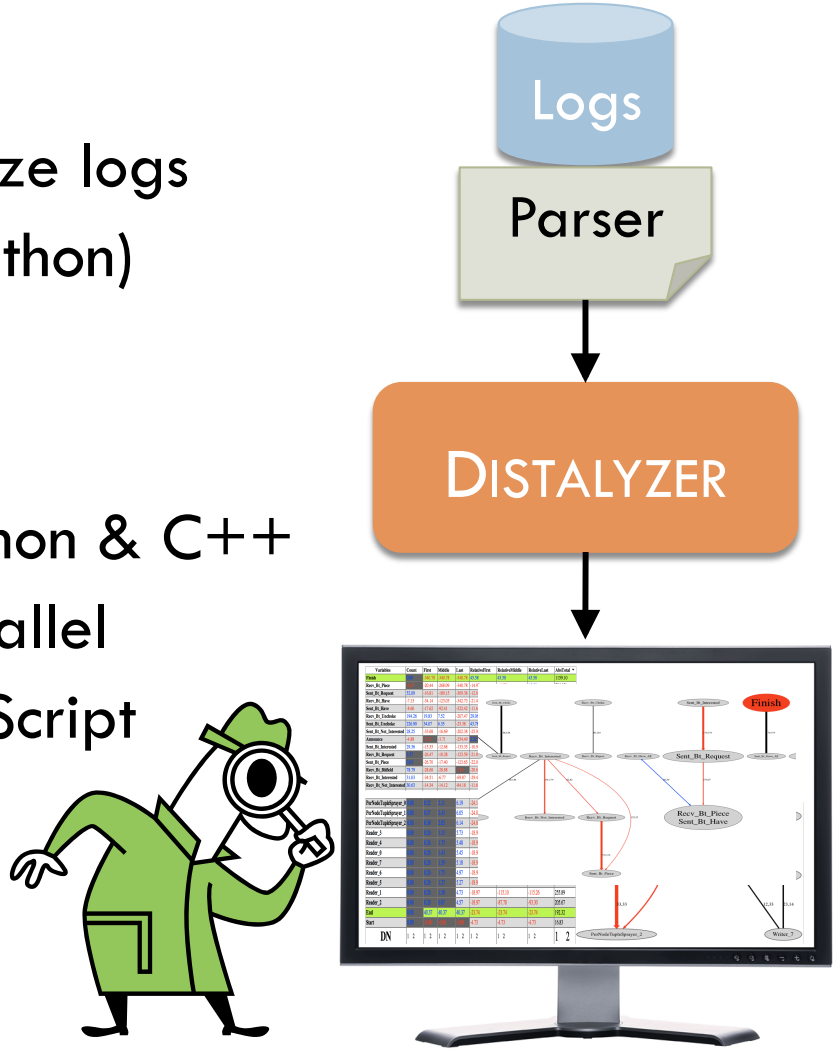
Design: Attention Focusing

- Score divergence and dependence equally



Implementation

- Parsers for text logs
  - Distalyzer API to categorize logs
  - Small (<100 lines Perl/Python)
  - One-time
- Distalyzer
  - 4000 lines of code in Python & C++
  - Offline usage, Highly parallel
  - Web frontend using JavaScript
    - Helps explore logs
  - Publicly available



Diagnosis Case Studies

3 systems
6 performance problems

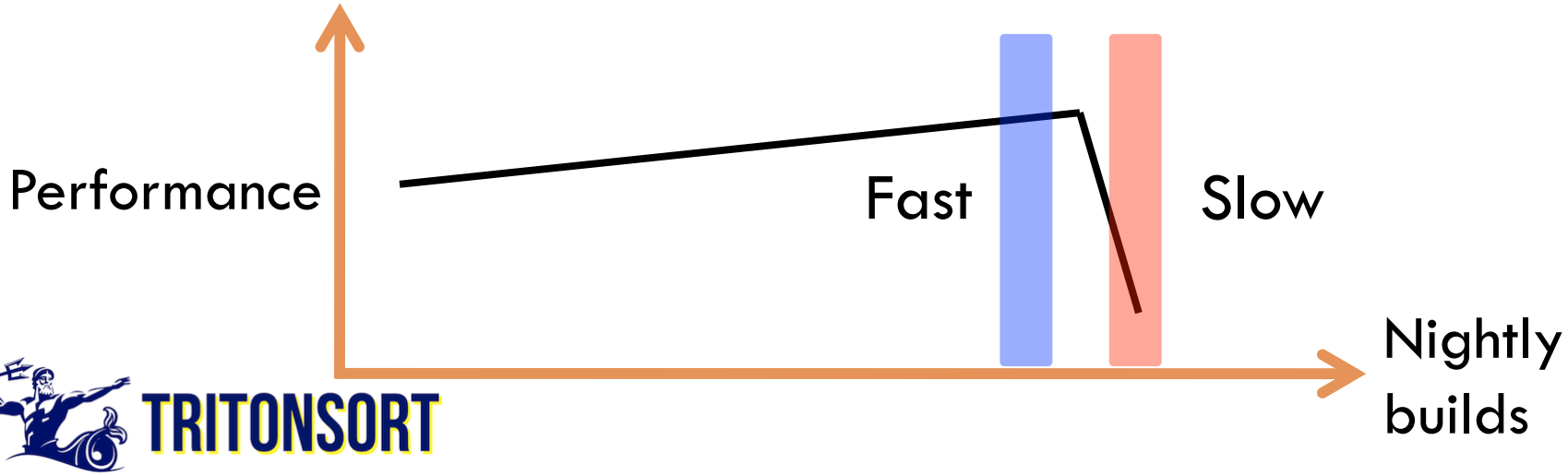
- TritonSort (Sorting) [NSDI 2011]
  - Different versions

- HBase (BigTable)
  - Different requests

- Transmission (BitTorrent)
  - Different implementations

TRANSMISSION

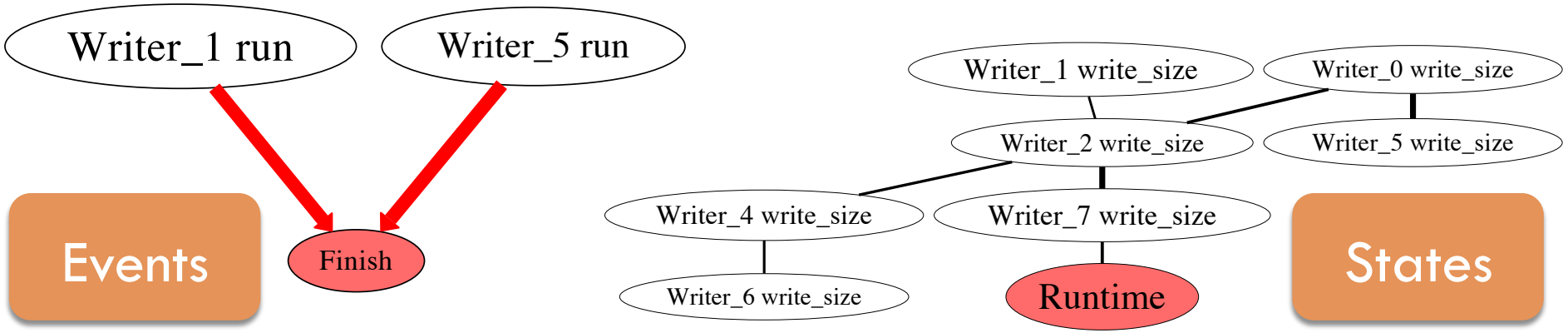
TritonSort (Different Versions)



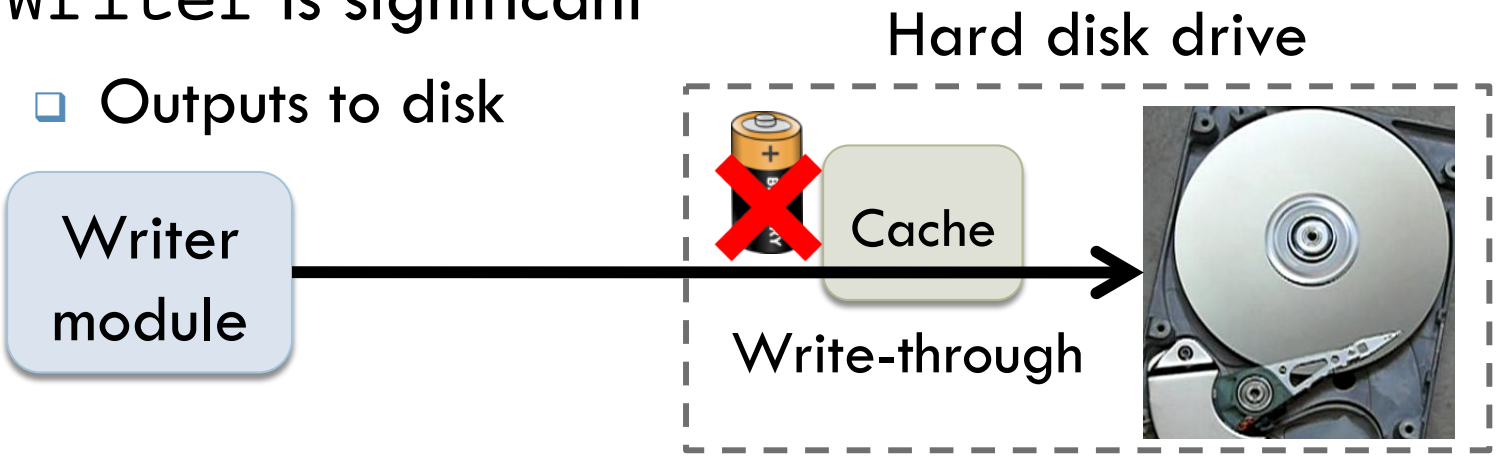
- Suddenly 74% slower on a day
- Baseline: Faster execution with same workload
- Diagnosed in 3-4hrs
  - Manual debugging took *"the better part of 2 days"*

Disclaimer: This plot was generated using fictitious data, and is only meant for illustration purposes.

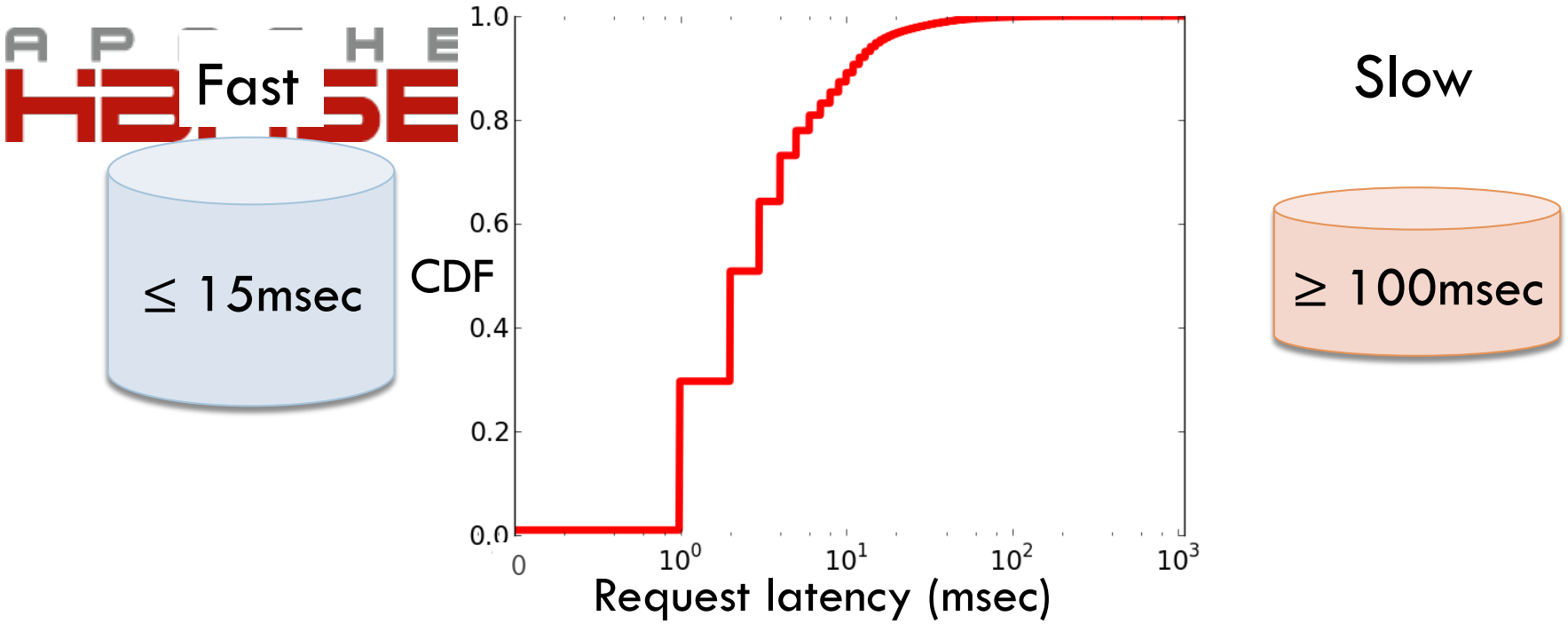
TritonSort



- `Writer` is significant
  - Outputs to disk

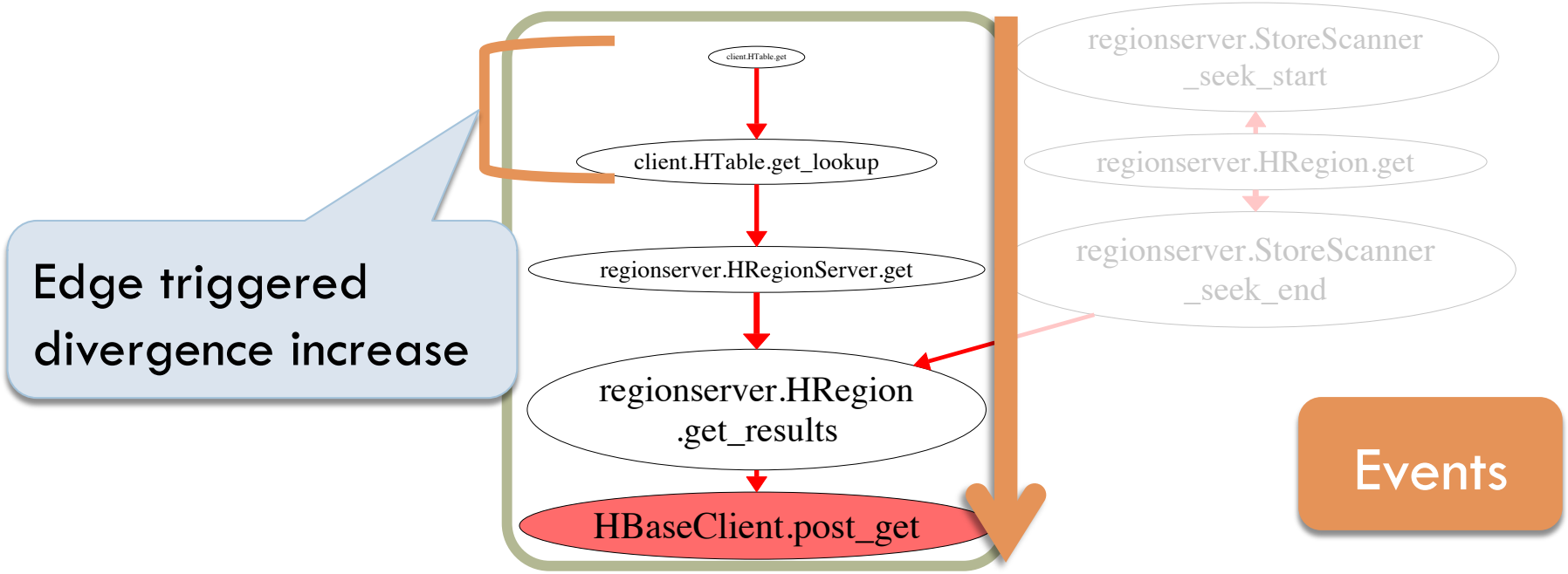


HBase (Different Requests)



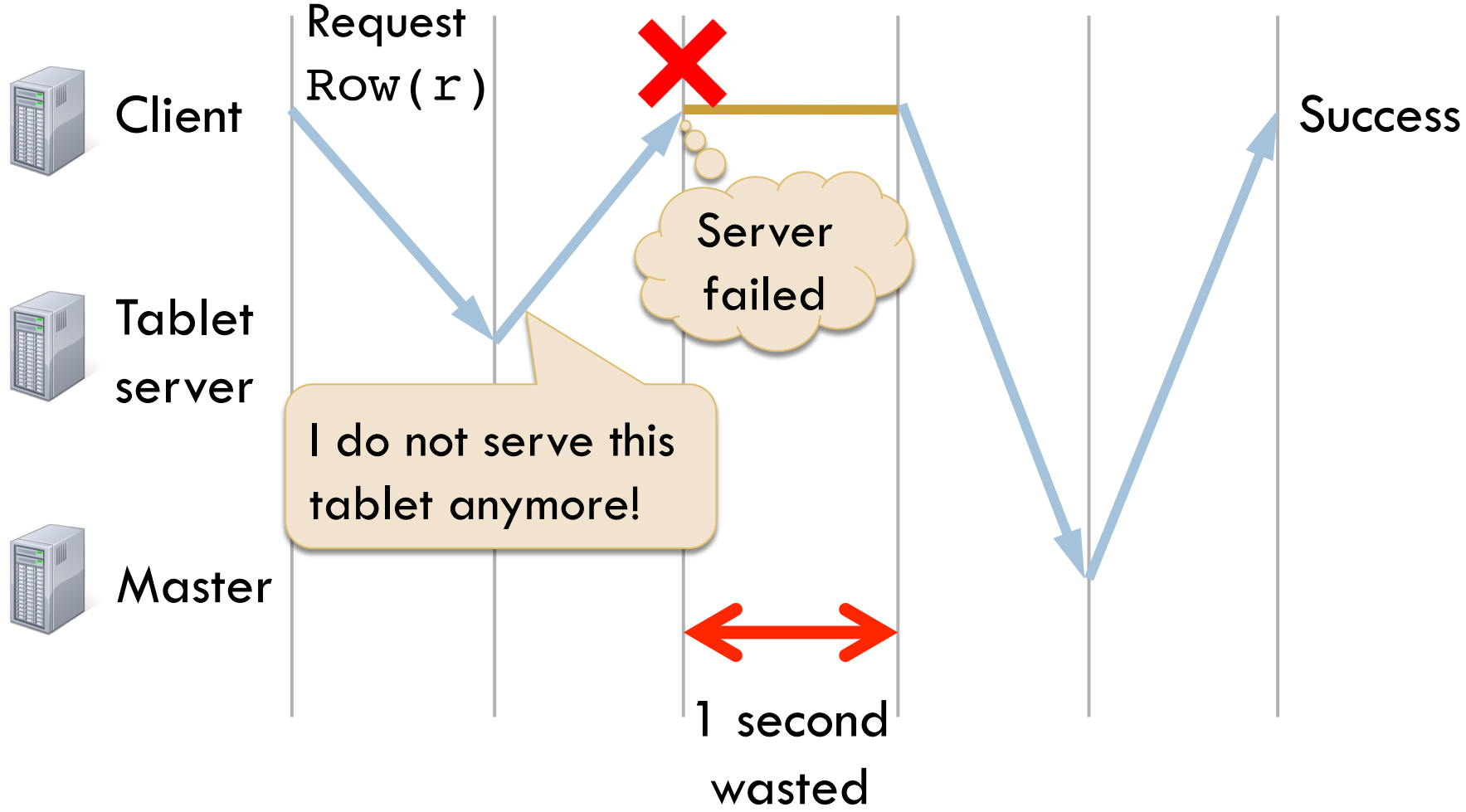
- Yahoo Cloud Storage Benchmark
  - 1Million operations: 95% reads, 5% writes
- Baseline: Fast requests in same execution

HBase: Slow Outliers



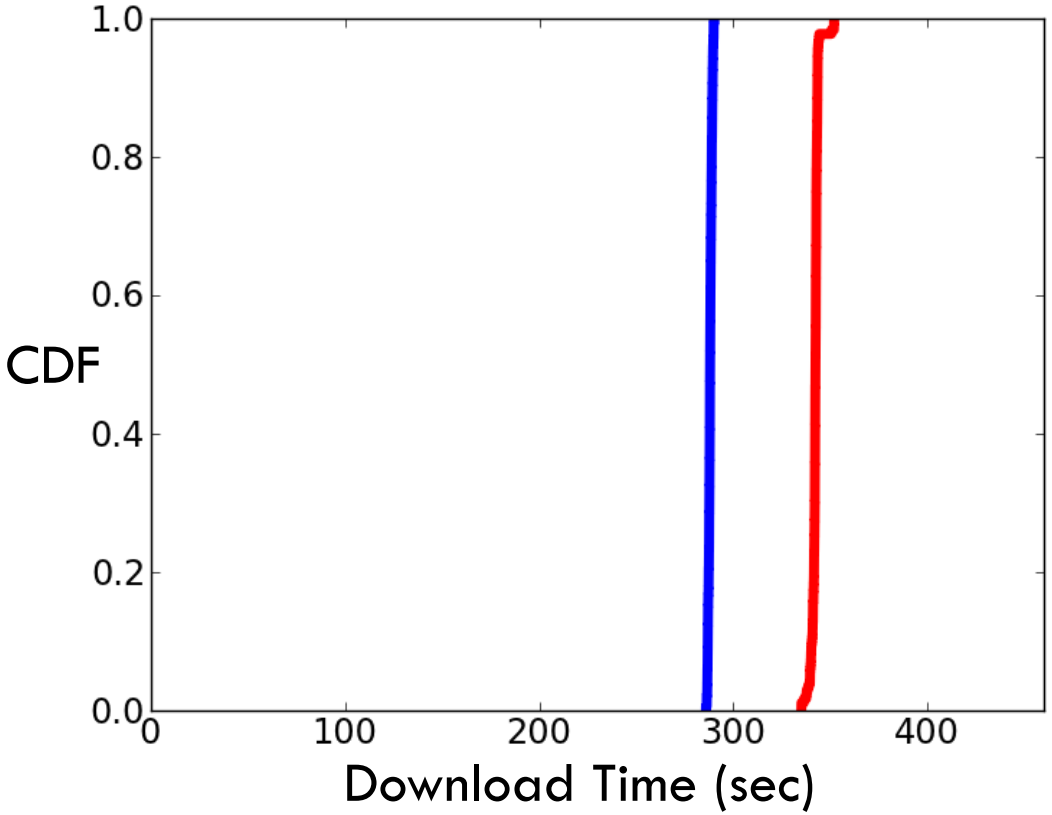
- `client.HTable.get_lookup` suspicious
- Code just preceding this log
- *Region* (Tablet) lookup by the client

HBase: Root Cause



BitTorrent (Different Implementations)

Fast



Slow

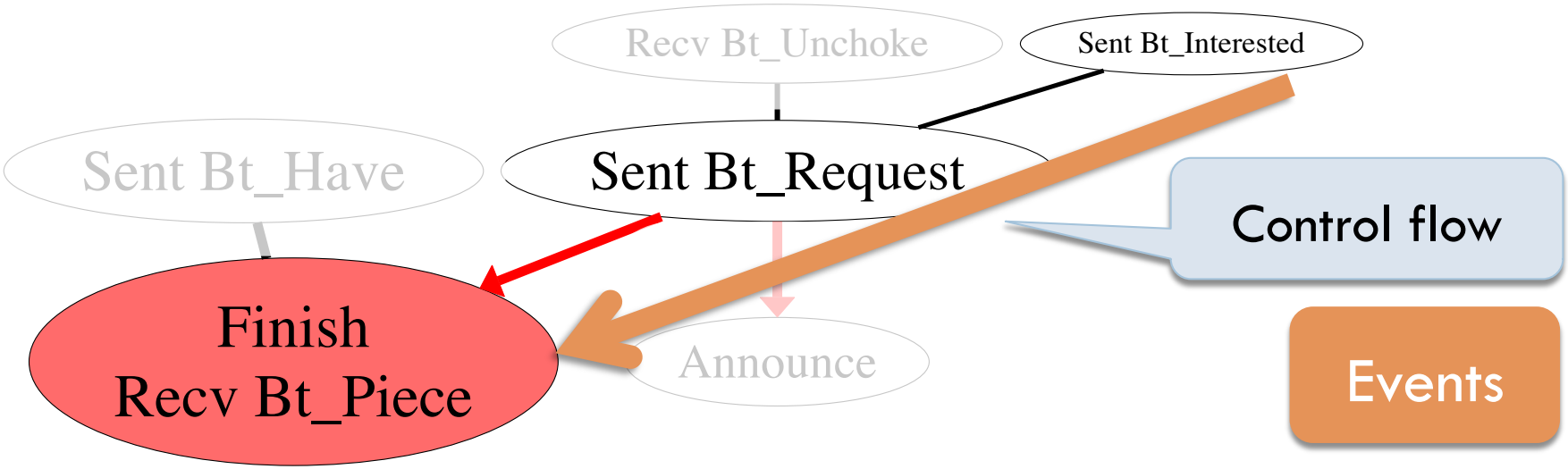
TRANSMISSION

180 nodes

250 KBps upload cap

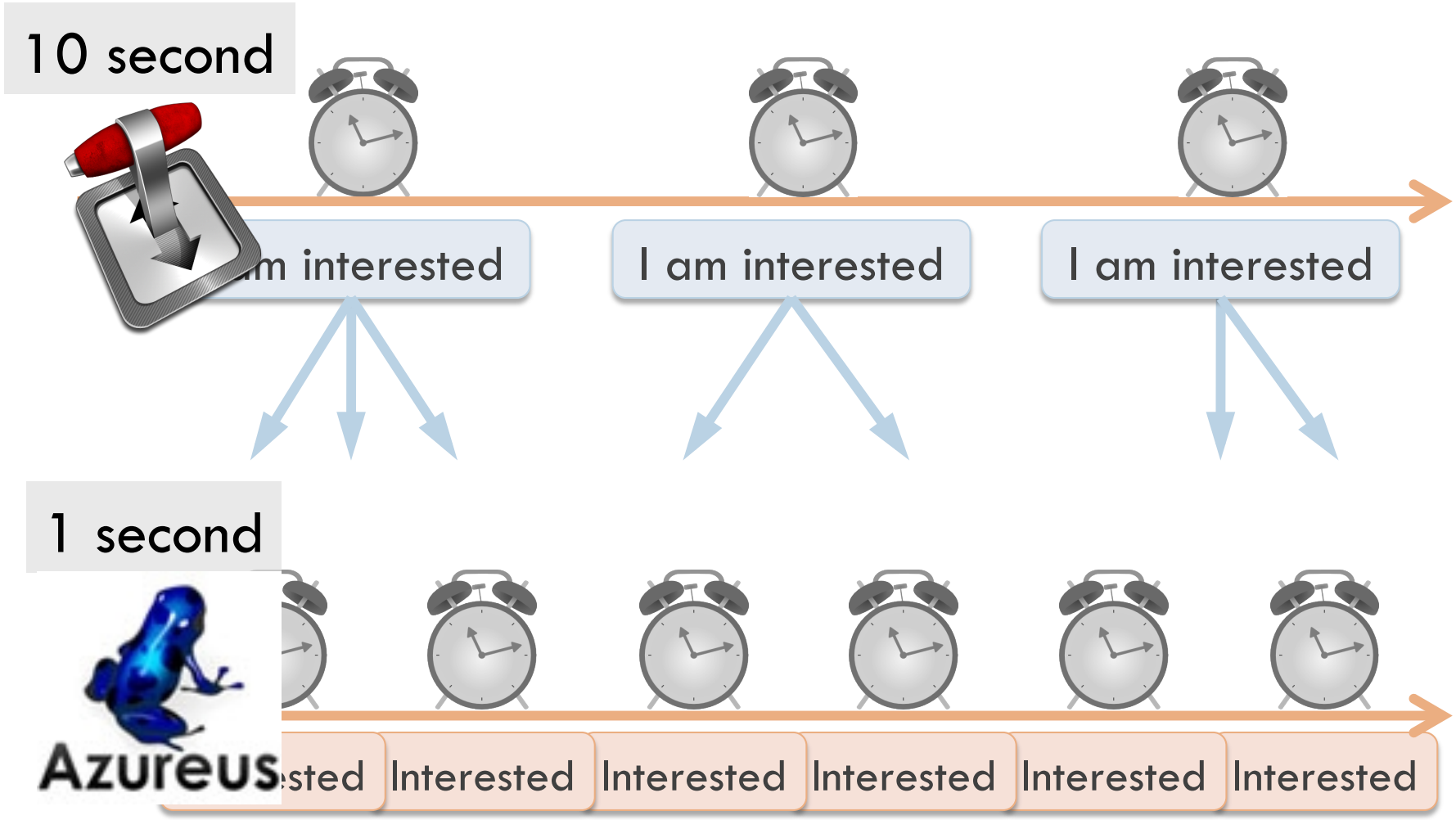
- Transmission 20% slower than Azureus
- Baseline: the Azureus implementation

BitTorrent: Transmission



- Dependency graph for *Last* feature
- `Sent_Bt_Interested` lags in Transmission
- A timer initiates this log

Transmission: Root Cause



Conclusion

- Diagnose performance problems by comparing against baseline
- Different implementations, runs and requests
- DISTALYZER automatically diagnoses performance
- Successful in 3 mature & popular distributed systems

DISTALYZER is free software
http://www.macesystems.org/distalyzer

NSDI 2012

Backup slides

Facts of Life

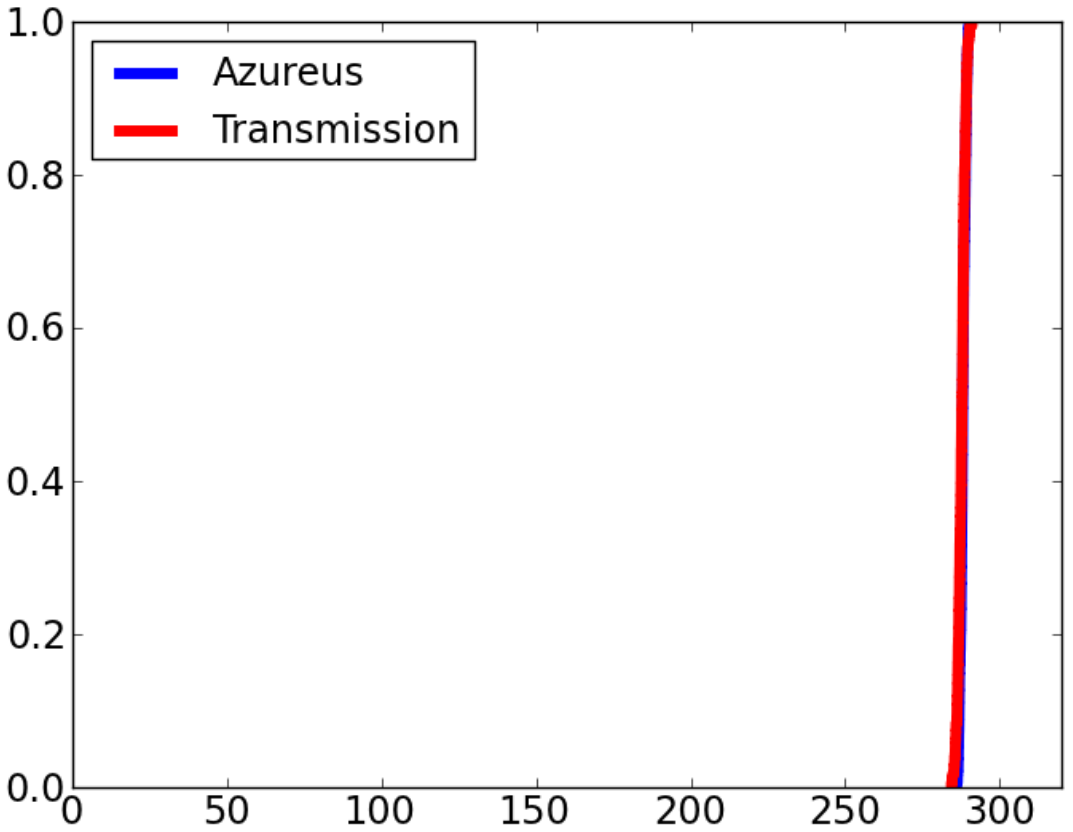
- Need enough samples
  - For statistical significance
- Similar execution environments
  - Experimental artifact or performance problem?
  - Can be relaxed with developer annotations
- Needs log data to succeed!
  - Performance overheads from logging
  - Logs should contain the bug

Transmission Fixed

- 288.0 sec Transmission vs. 287.7 sec Azureus

TRANSMISSION

CDF



Clock Synchrony

- Responsibility of the instrumentation framework
- Relative synchrony
  - Master can broadcast START, and all nodes log it
- No, if granularity is already small

Related work (2)

- NetMedic [Kandula et. Al. SIGCOMM 2009]
- Giza [Mahimkar et. al. SIGCOMM 2009]
- WISE [Tariq et. al. SIGCOMM 2008]
- DISTALYZER
  - Software problems
  - Inter-component Dependencies and faults

Diagnosed Performance Problems

- TritonSort
  - Hard disk loose cache battery (previously diagnosed)
- Hbase (22%)
  - Tablet server lookup mishandled
  - Linux default disk scheduler (CFQ)
  - Network slowdown (Nagle's algorithm)
- Transmission (45%)
  - Same-IP peers mishandled
  - Sent BT_Interested timer