Evaluation of a Microrebootable System

George Candea, Shinichi Kawamoto, Yuichi Fujiki,
Greg Friedman, Armando Fox

-- Stanford University --

μRB and Crash-Only Design

- reboot-curable failures
- microreboot at the component level:
  - correct recovery
  - localized + contained recovery
  - fast + correct reintegration into running system
- crash-only design to enable μRB-ing:
  - state segregation (DB, SSM, ...)
  - componentization
  - fine grain resource reclamation

Prototype

- μRB-enabled JBoss (comps = EJBs) using MySQL and extended SSM
- RUBiS: online auction app (132K items, 1.5M bids, 100K users)
- Fault injection:
  - null references
  - deadlocks, infinite loops
  - corruption of volatile metadata
  - resource leaks
  - Java exceptions and errors
- Client-side detection + automated recovery

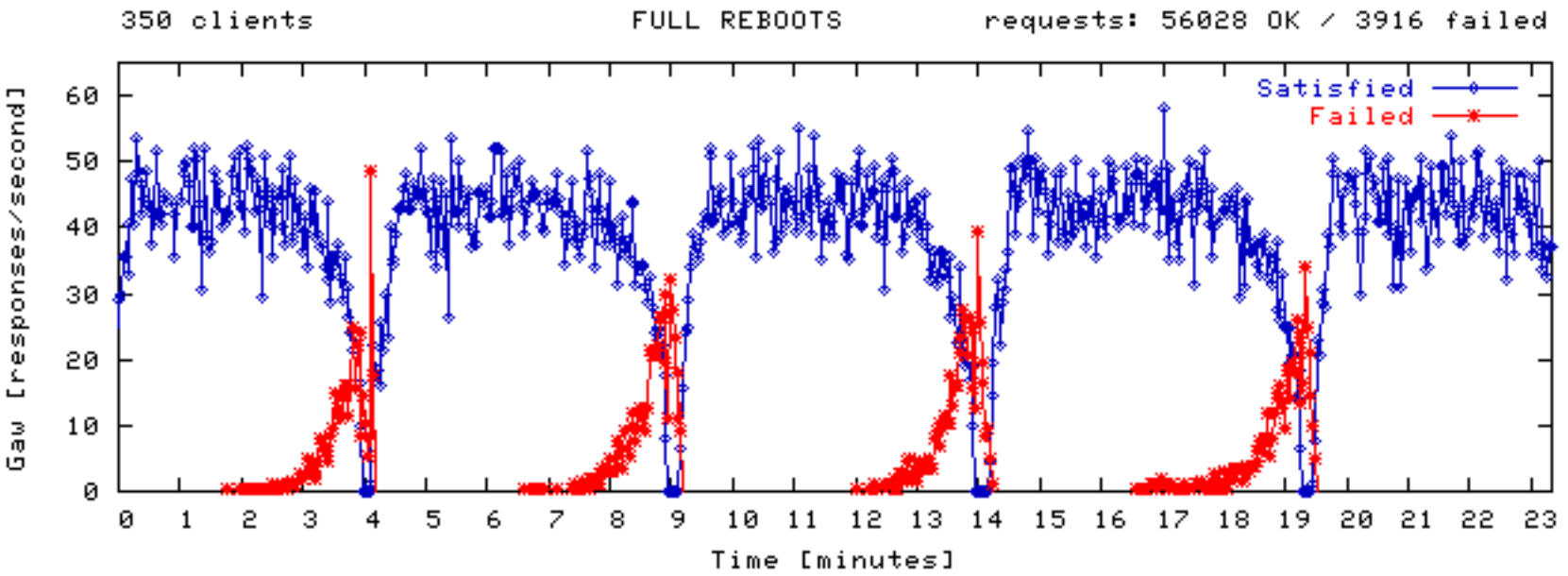
Preview

- μRB-ing vs. FRB-ing on a single node
- μRB-ing complements cluster-based solns.
- Fast recovery tolerates sloppy fault det.
- Microrejuvenation averts failure at low cost
- Insignificant performance impact (1%)

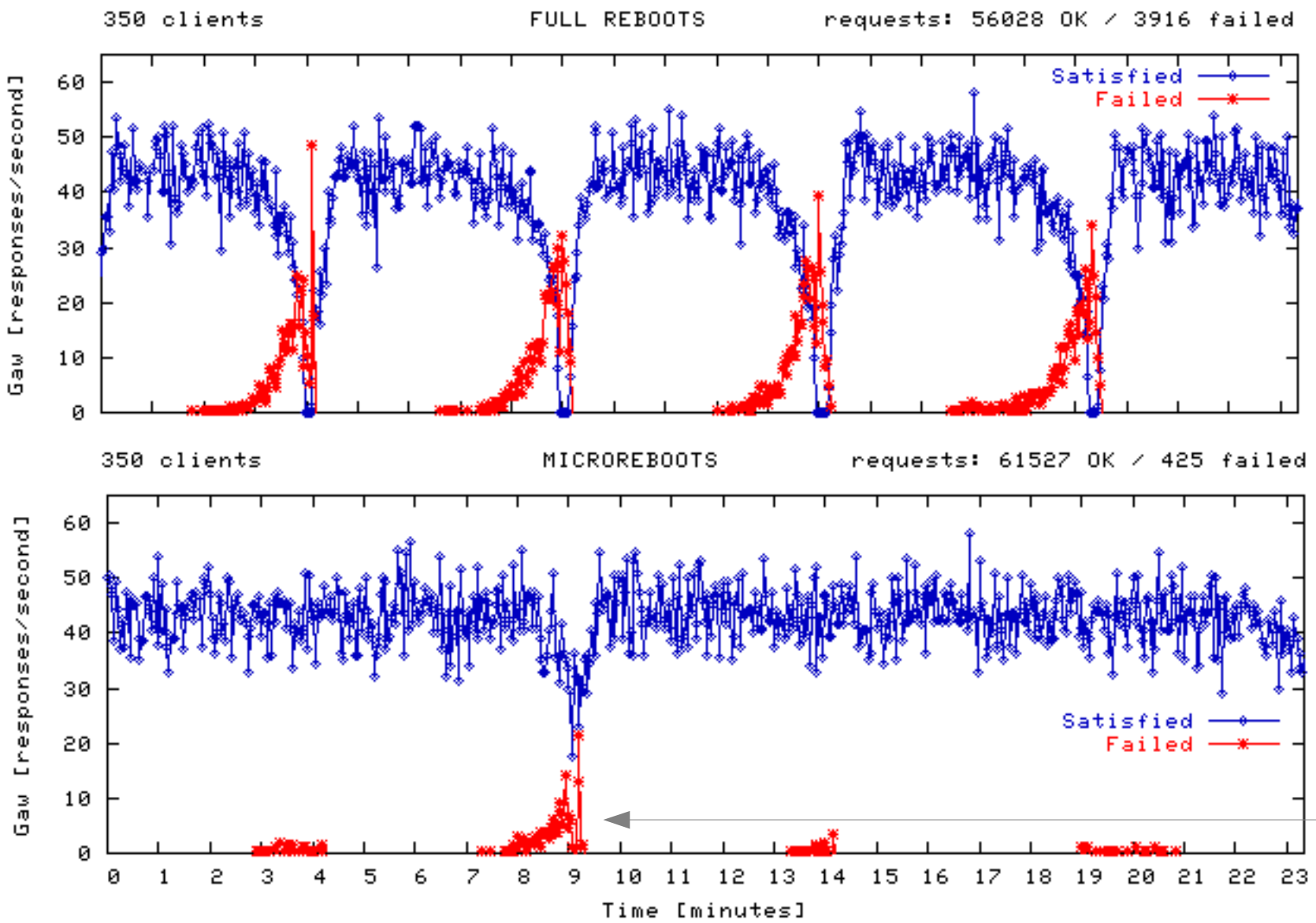
The Gaw Metric

- Goodput x time is not good enough:
  - partial failure can improve metric
  - does not capture user actions
- Action-weighted goodput (Gaw):
  - user session = login ... logout (or abandon)
  - user action = op, op, ..., commit point
  - user actions are atomic
- Emulate 350 clients w/ eBay-like workload

FRB vs. μRB: Gaw



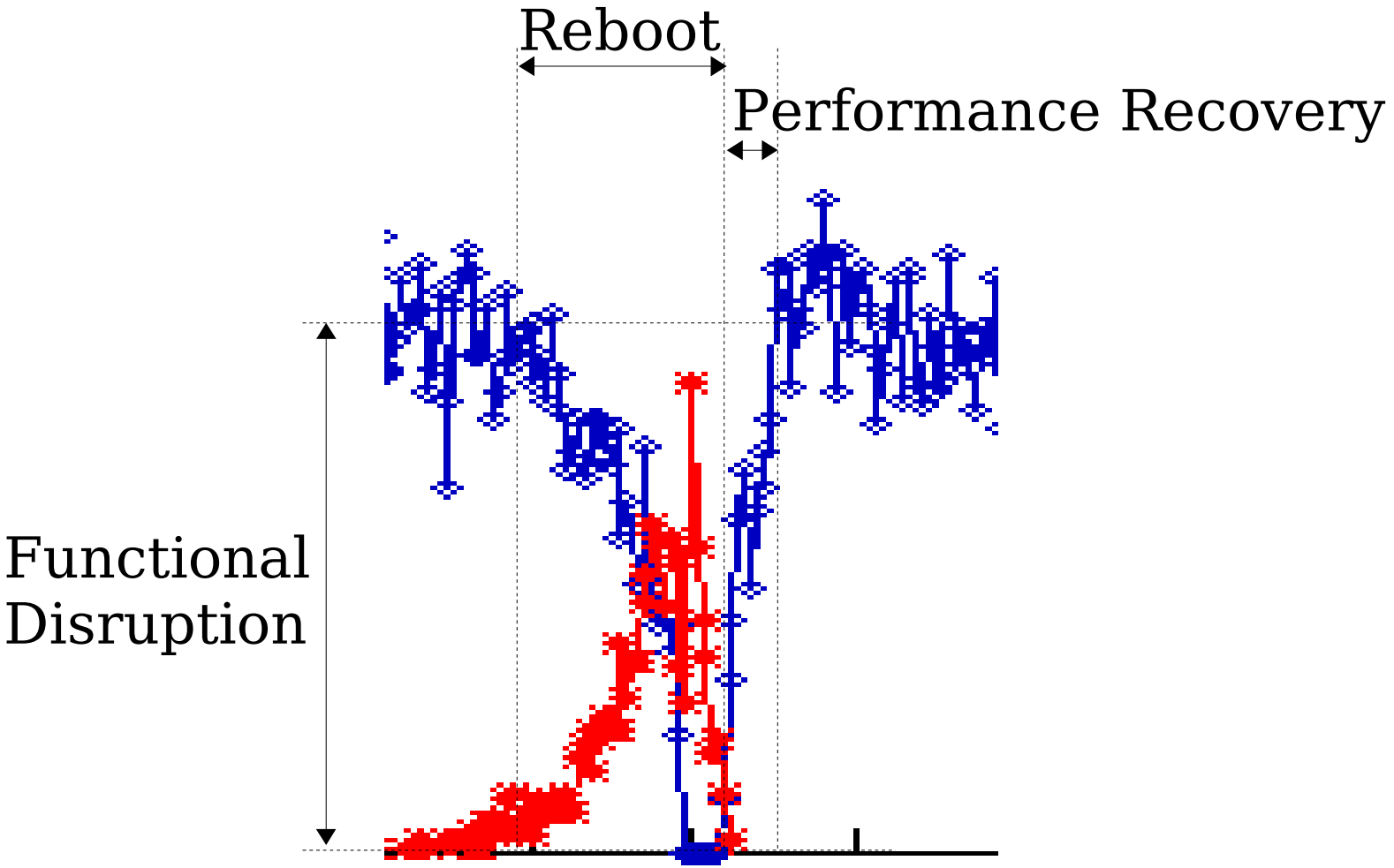
FRB vs. μRB: Goodput



89%

6 false positive μRBs

Gaw Dip (Zoom in)

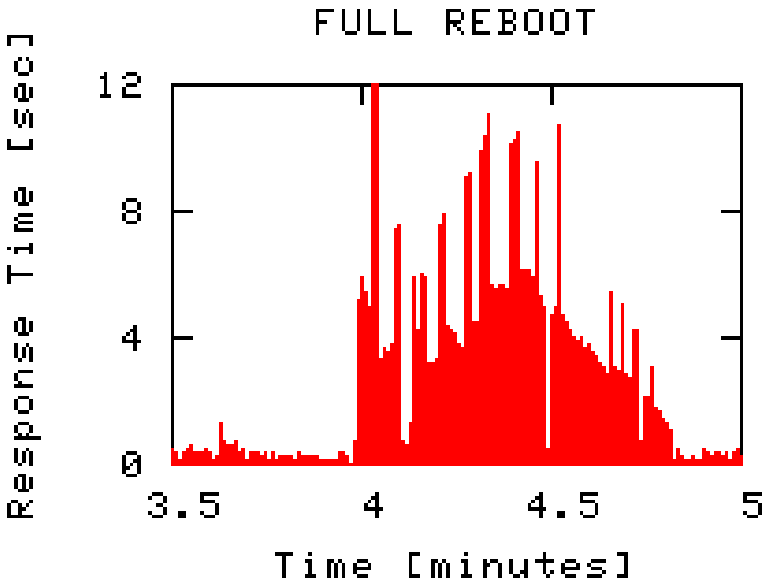


Recovery Unit Size

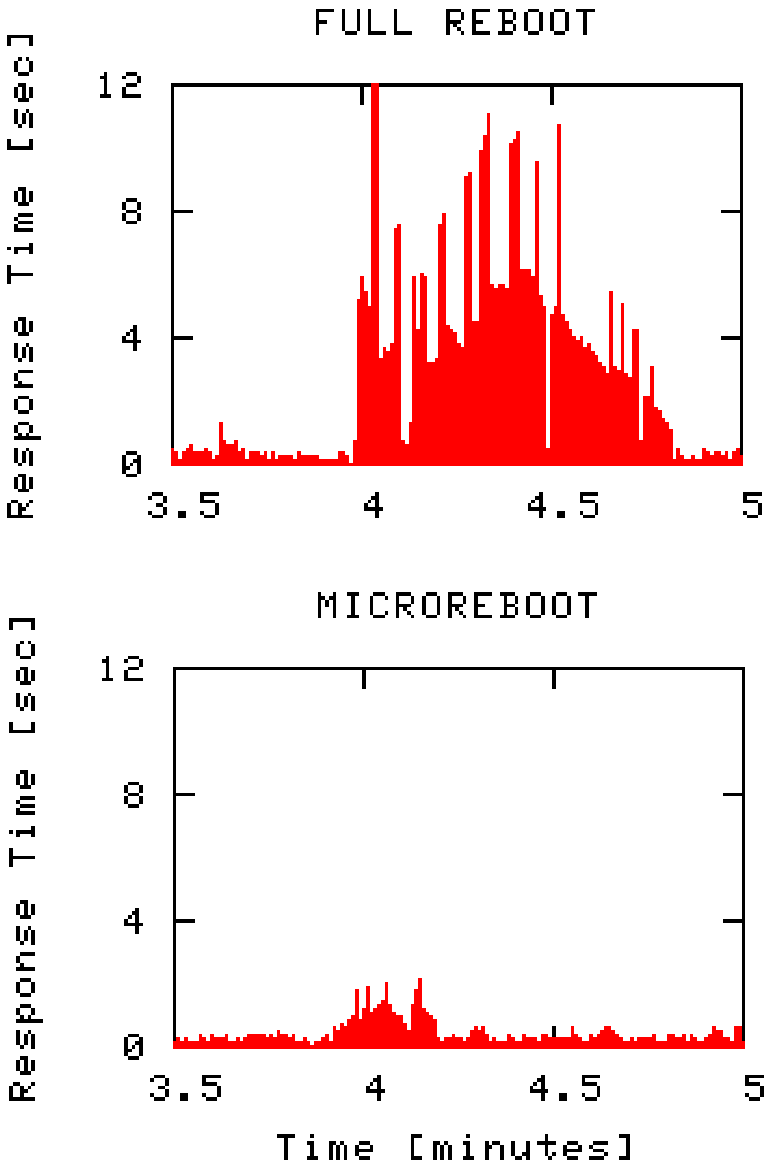
Component	Ave	Min	Max
Jboss restart (JVM)	51800	49000	54000
RUBiS restart (app)	11679	7890	19225
SB_CommitBid	286	237	520
SB_BrowseCategories	340	277	413
.....	....	...	...
SB_SearchItemsByCategory	911	488	3019
IDManager	1059	663	1547
UserFeedback	1248	761	1591
BuyNow	1421	668	4453
User-Item	1828	876	4636

››› order of magnitude

Performance Recovery Time

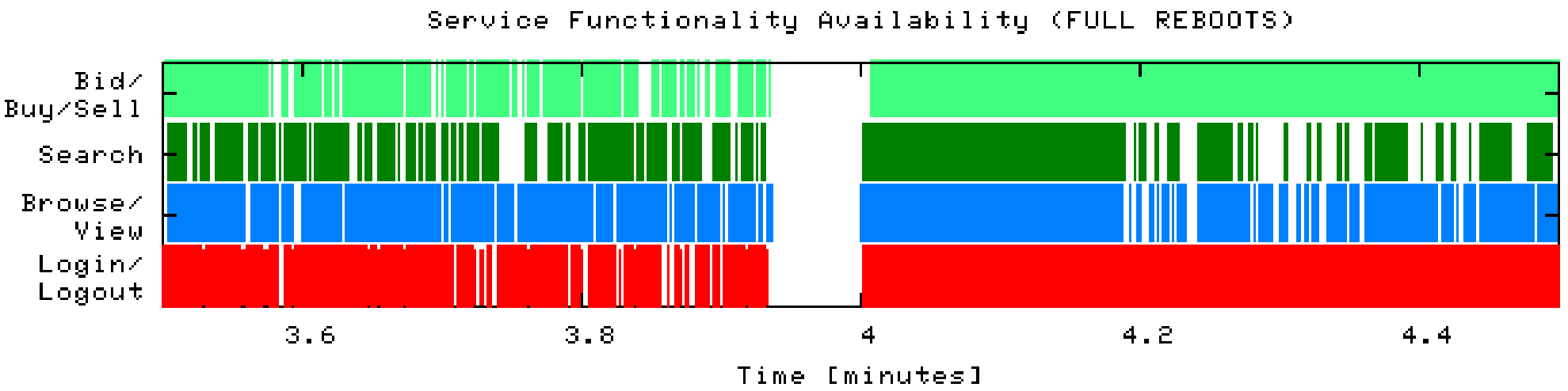


Performance Recovery Time



- 10-sec reboot + 1-minute recovery
- 8 sec threshold for interactivity

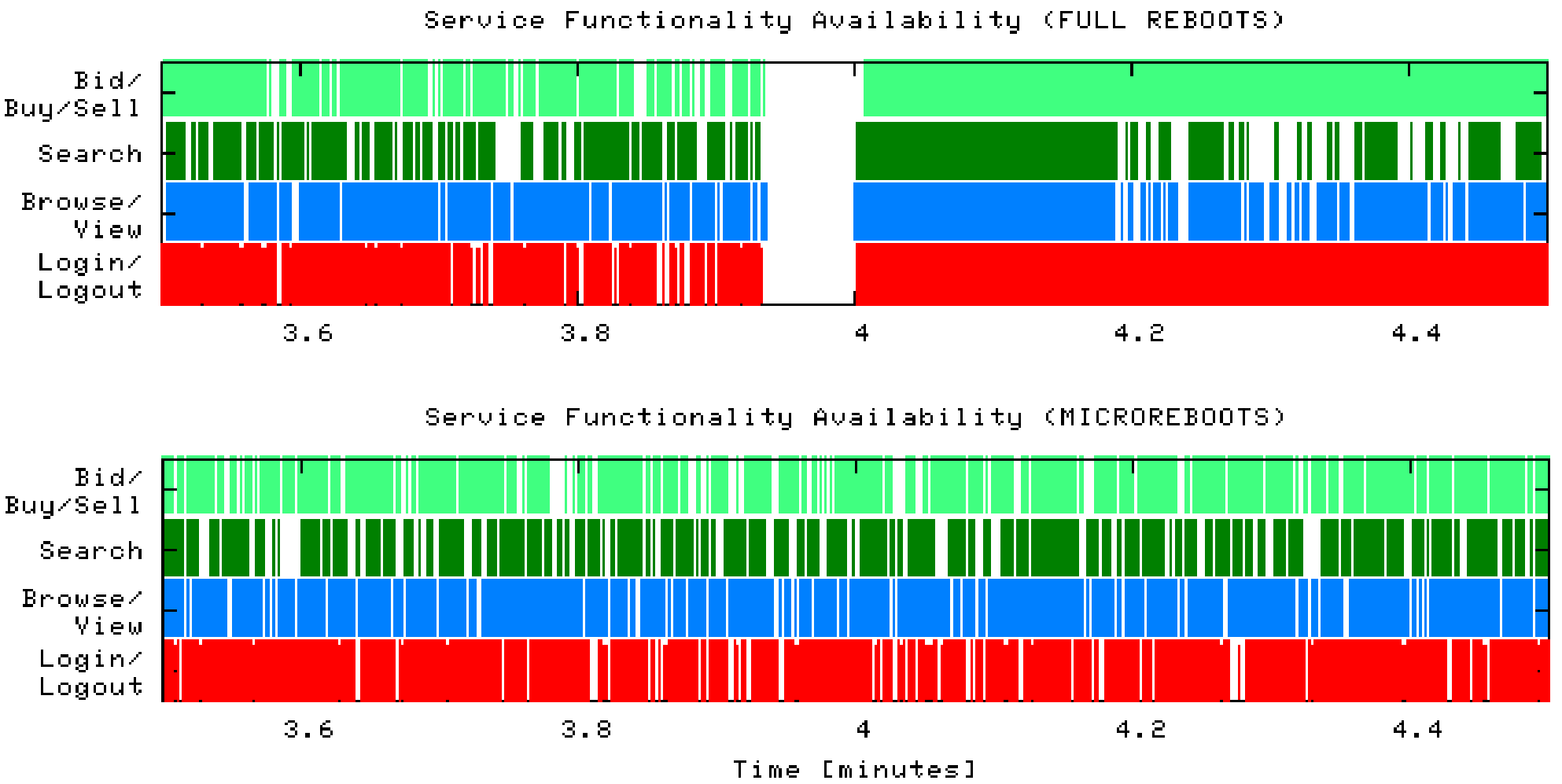
Failure Containment



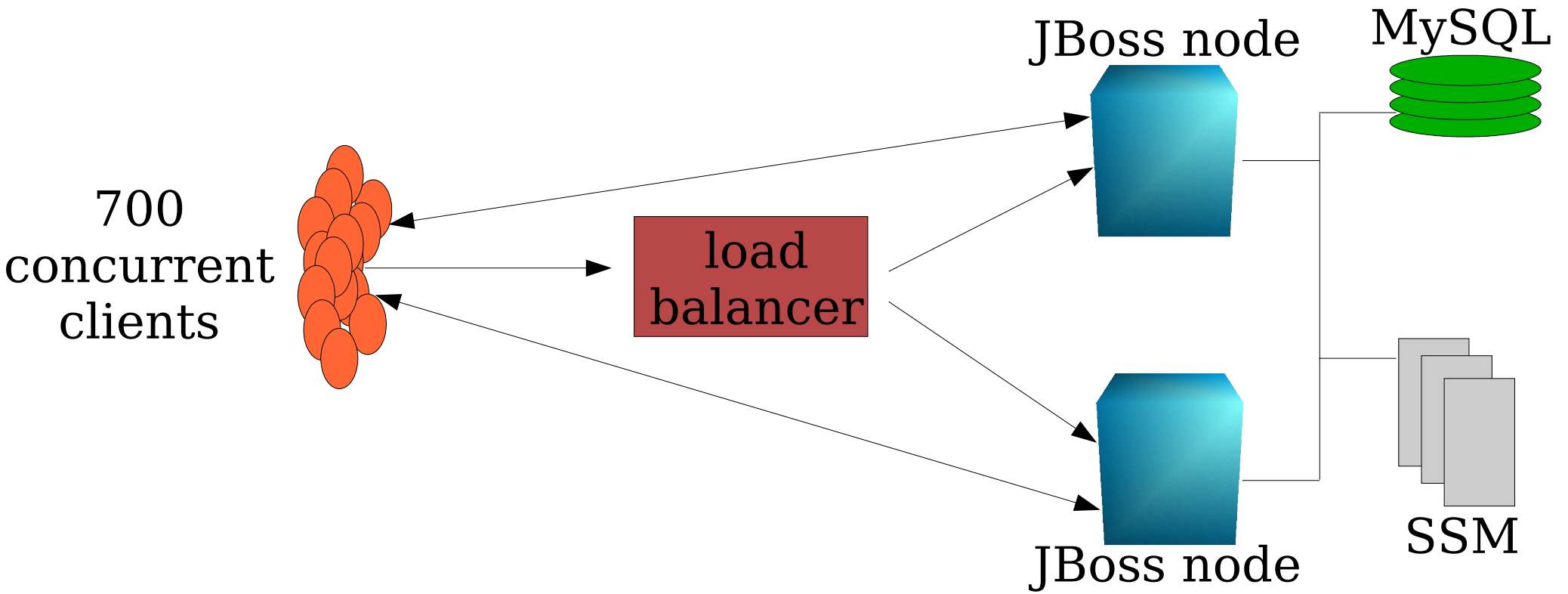
- solid line = activity
  gap=no activity

- Causes: no requests, site down, etc.

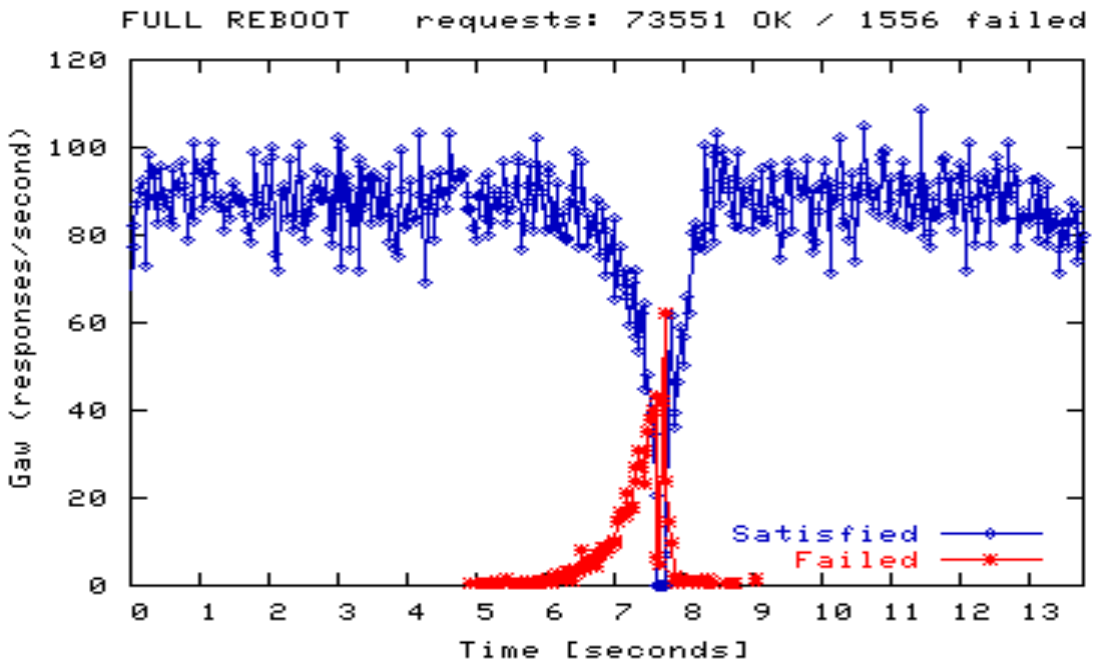
Failure Containment



Cluster Setup

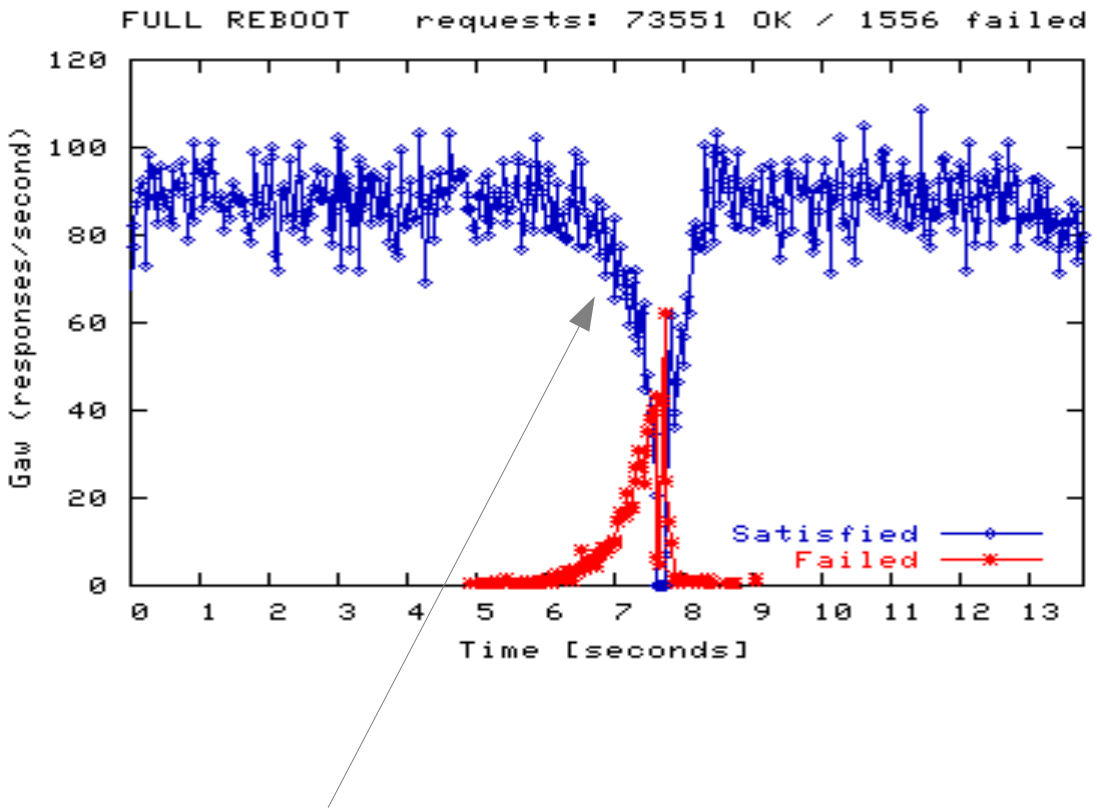


Failover (with SSM)



Aggregate Gaw

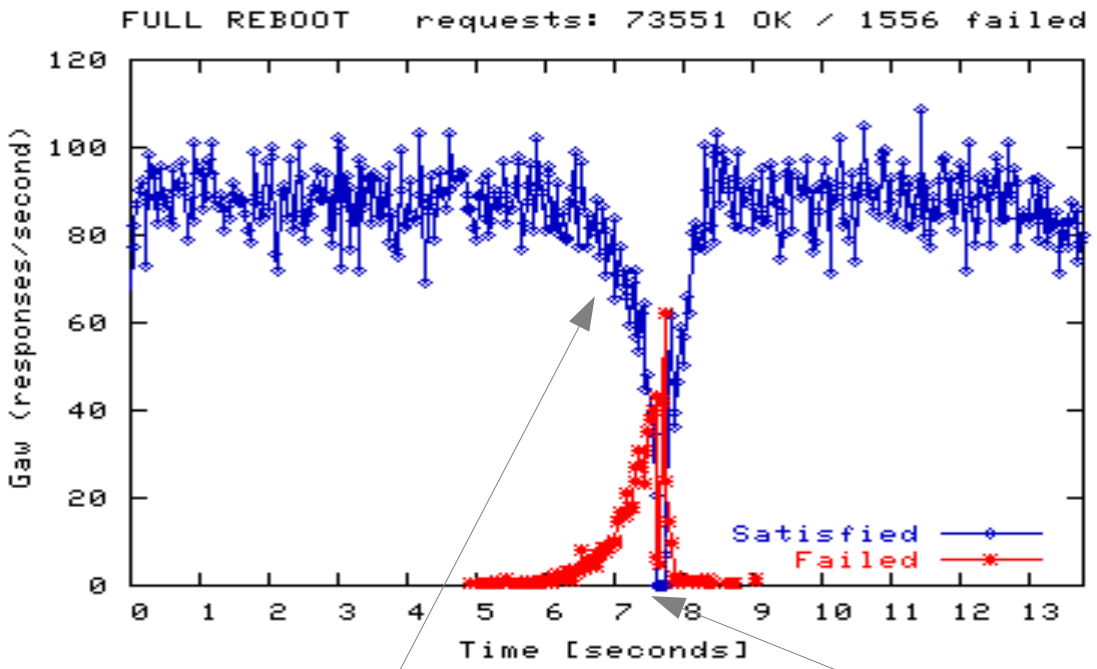
Failover (with SSM)



Failover loads good node w/ 700 clients

☞ response time > 8 sec

Failover (with SSM)



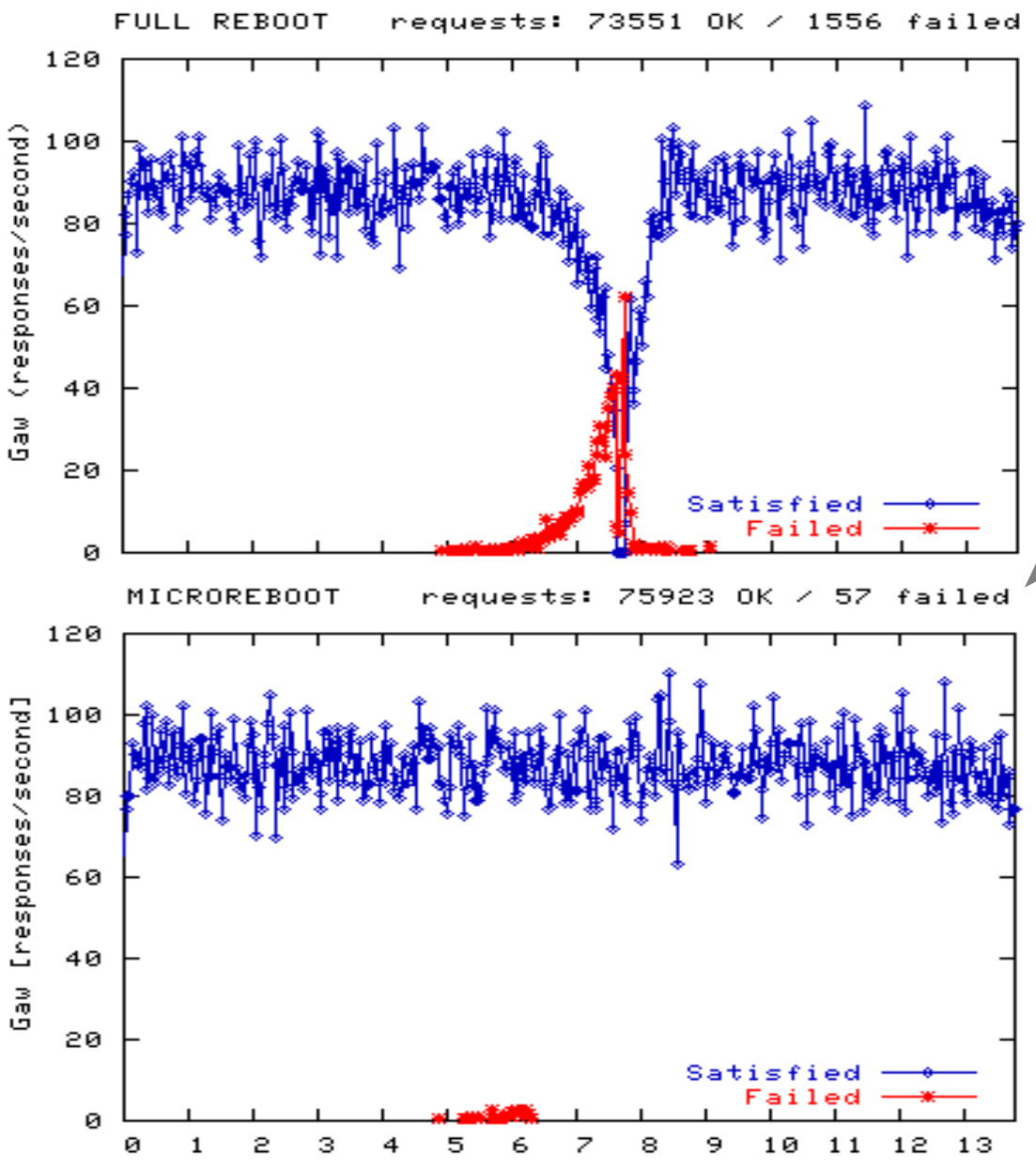
FRB good node

Failover loads good node w/ 700 clients

☞ response time > 8 sec

... oscillation possible

Failover (with SSM)



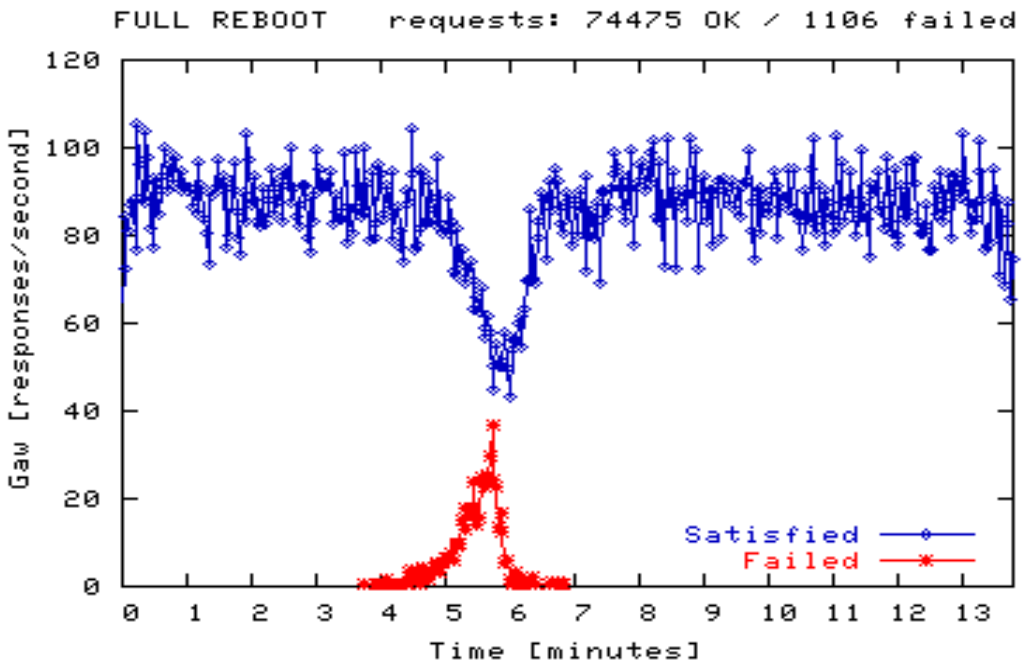
Aggregate Gaw

96%

Failed reqs due to:

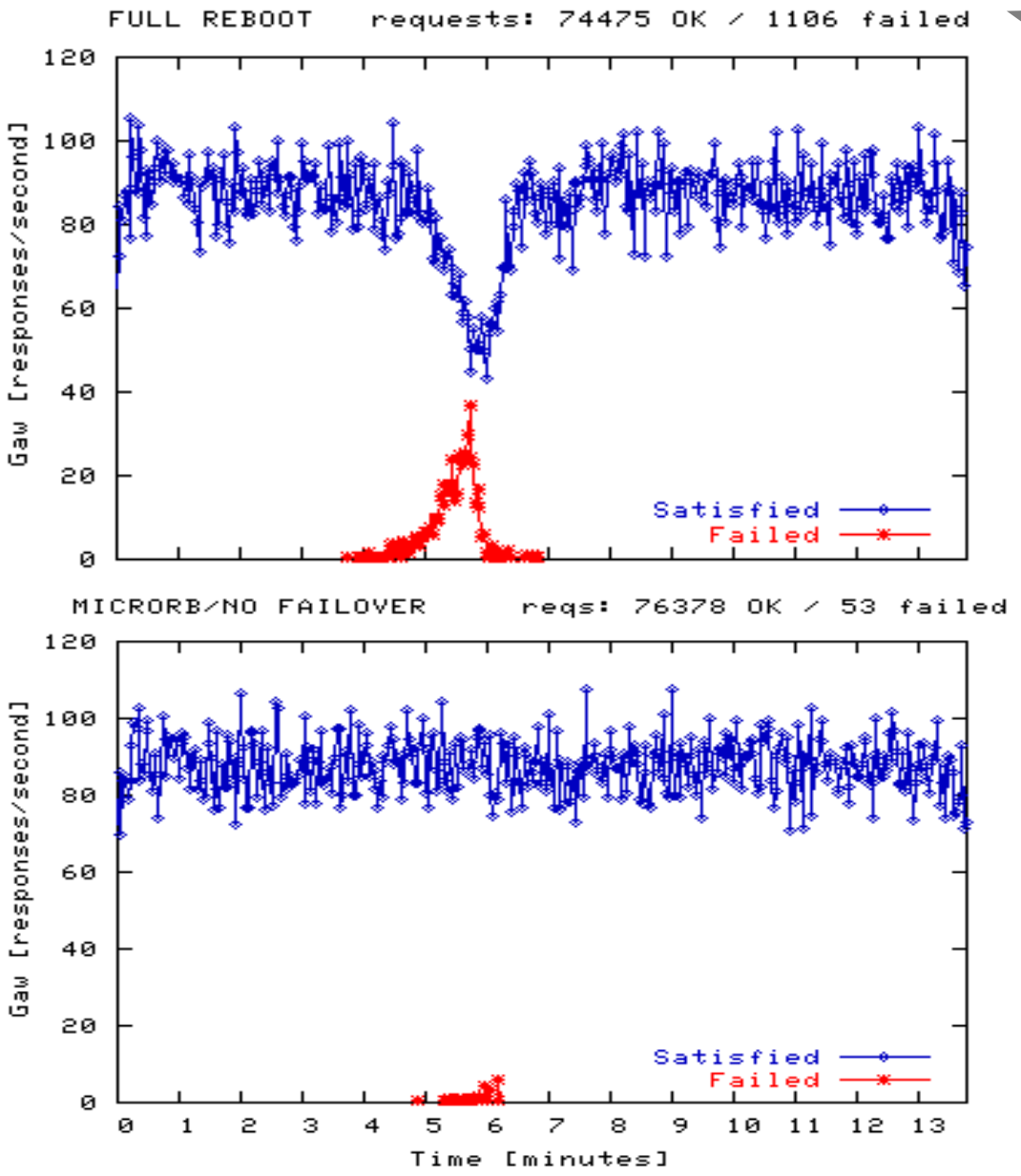
- timeout
- cluster down
- reqs in progress on recovering node

Avoiding Failover Altogether



- SSM not used
- Load bal. fails orphaned sess. right away

Avoiding Failover Altogether



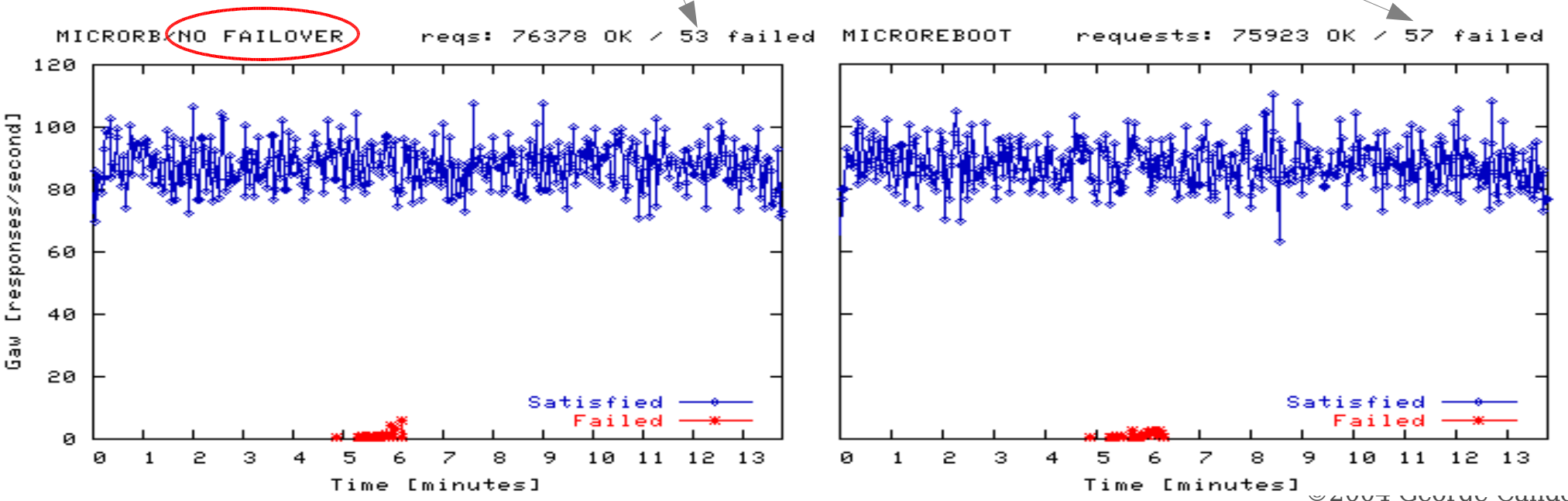
95%
(always uRB prior to failover ?)

- SSM not used
- Load bal. fails orphaned sess. right away

Avoiding Failover Altogether

Failover is coarser than uRB-ing

☞ recovery isolation enables partial avail.



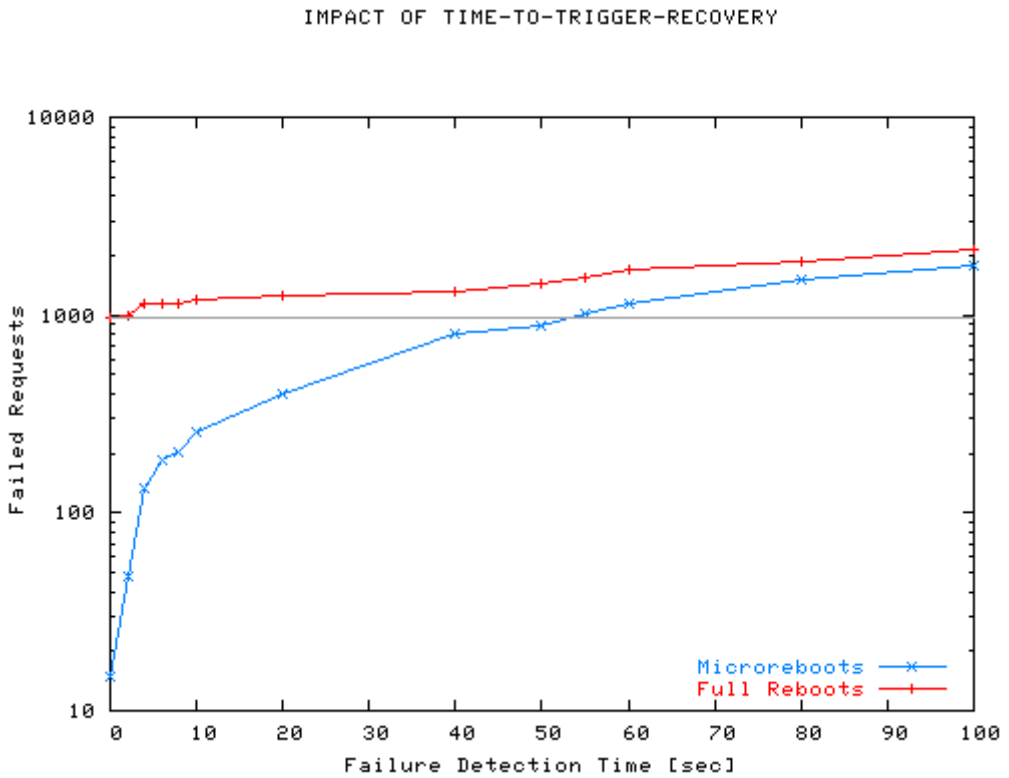
Lax Failure Detection

- Can uRB-ing be cheap enough to employ at the slightest hint of failure ?

- 2 ways we can relax failure detection:
  -- more false positives
  -- longer FD time (think longer)

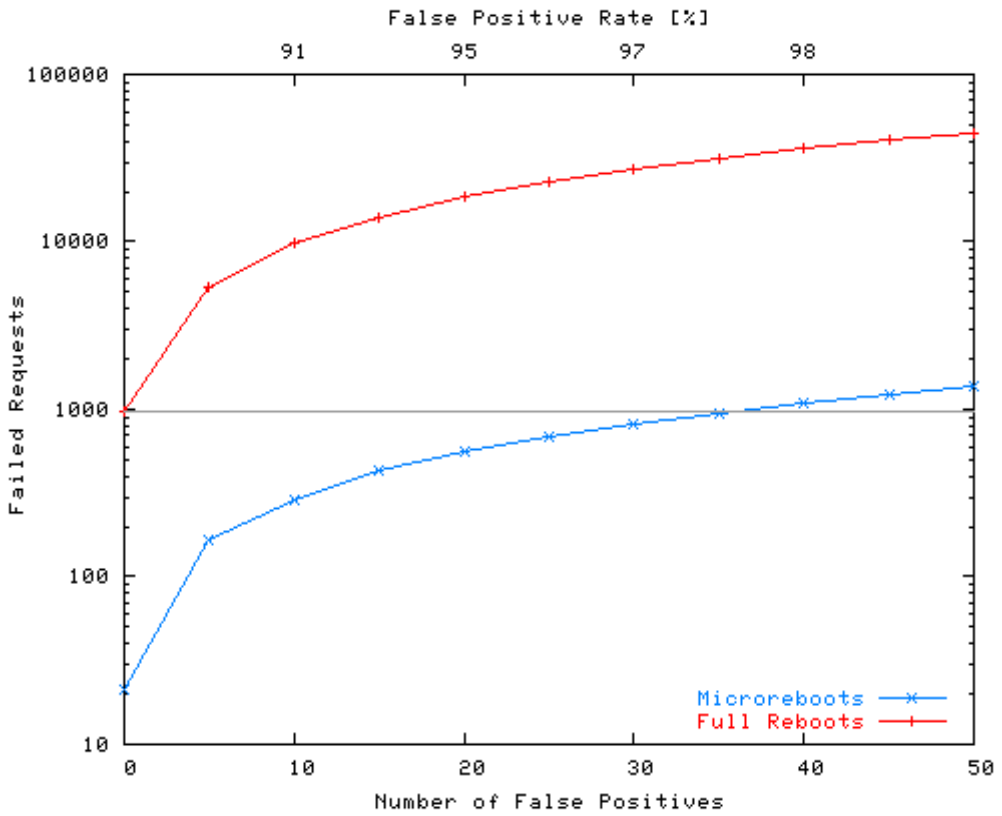
- false negatives not a problem for us

Delayed Detection/Reporting



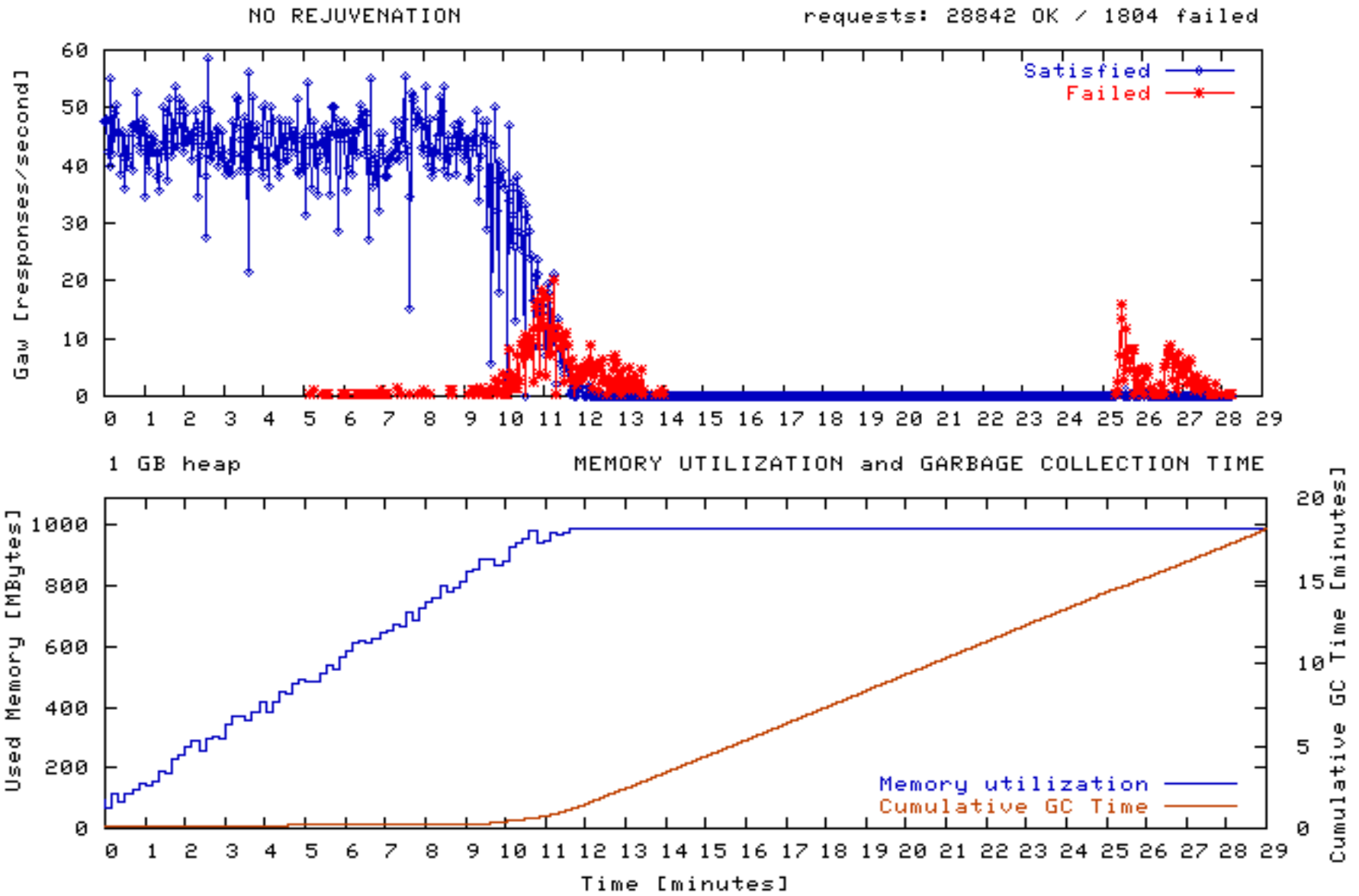
- 0 sec w/ FRB = 53 sec w/ uRB
  -->enable more accurate diagnosis

Increased False Positive Rate

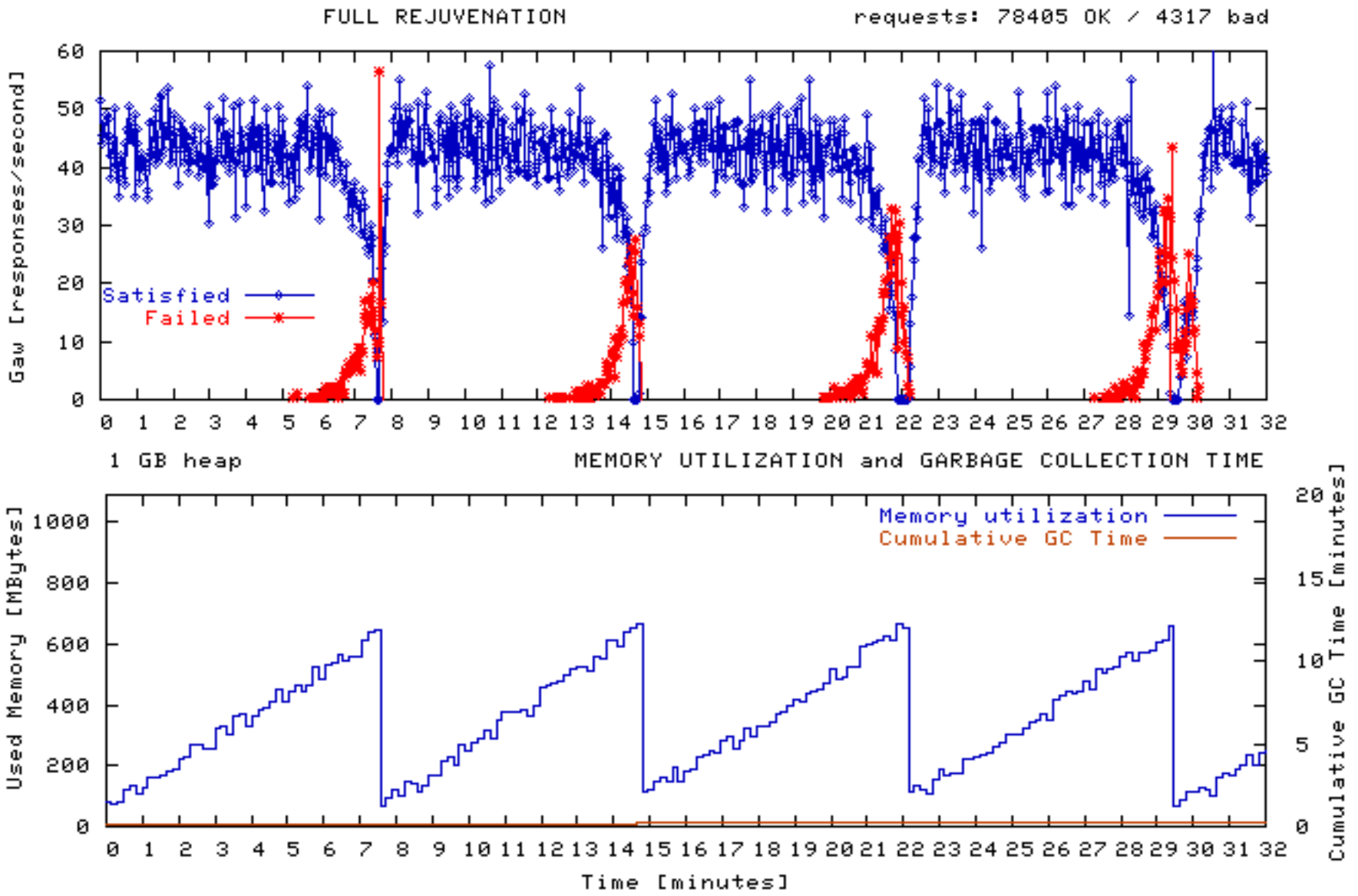


- 0% FP rate w/ FRB = 97.2% FP rate w/ uRB
  --> enable faster detection

Why Prophylactic Rebooting ?

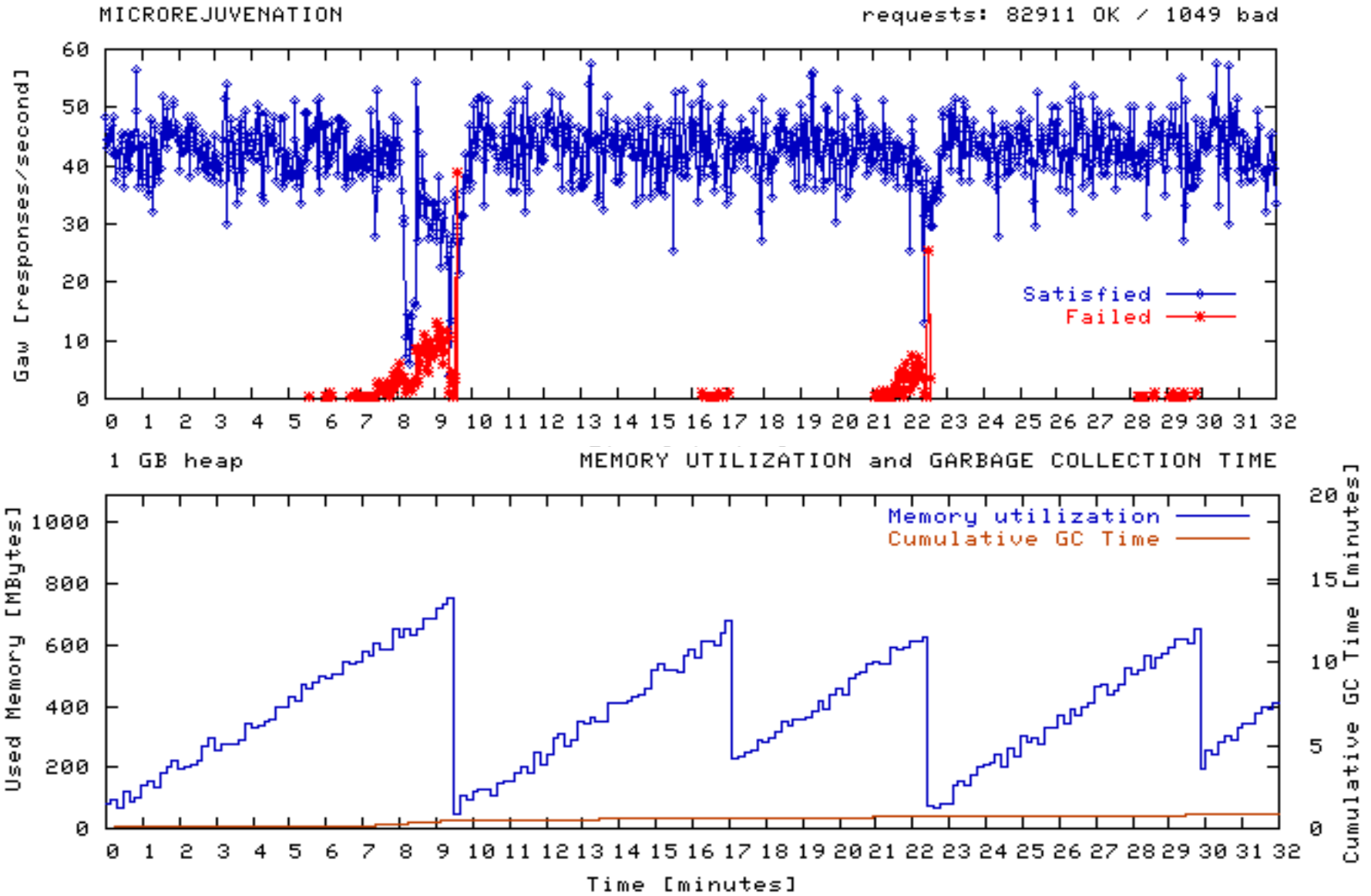


Full Rejuvenation



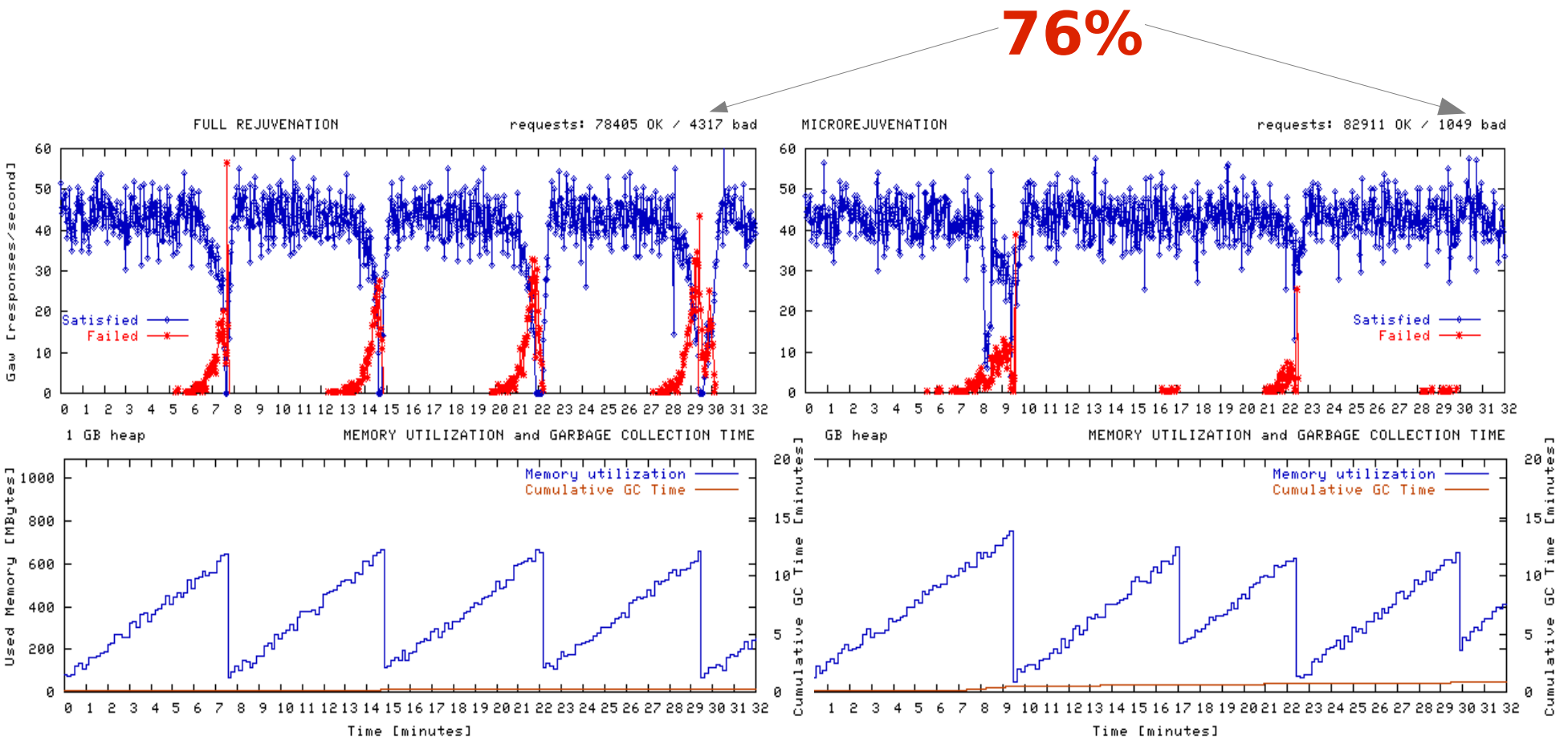
M_alarm = 35%

Microrejuvenation



M_sufficient = 80%

Full vs. Microrejuvenation



unplanned total downtime ---->

planned partial downtime

Performance Overhead

		JBoss 3.2.1 w/ HttpSession-RUBiS	uRB-JBoss w/ SSM-RUBiS	JBoss 3.2.1 w/ SSM-RUBiS
Throughput [req/sec]		44.8	44.5	44.4
Latency [msec]	Avg	39	82	83
	Max	826	1245	1097
	StDev	0.066	0.131	0.142

- 150 clients/node: latency=38 msec (3 -> 7 nodes)
- Human-perceptible delay: 100-200 msec
- Auction site: 41 req/sec, 33-300 msec latency

Summary of Results

- uRB-ing has many of FRB's recovery properties
- uRB is order-of-magnitude less disruptive
- always uRB prior to failing over in clusters
- cheap recovery simplifies detection (97% FP)
- rejuvenate system w/out ever shutting down
- insignificant performance cost: 1%

http://crash.stanford.edu