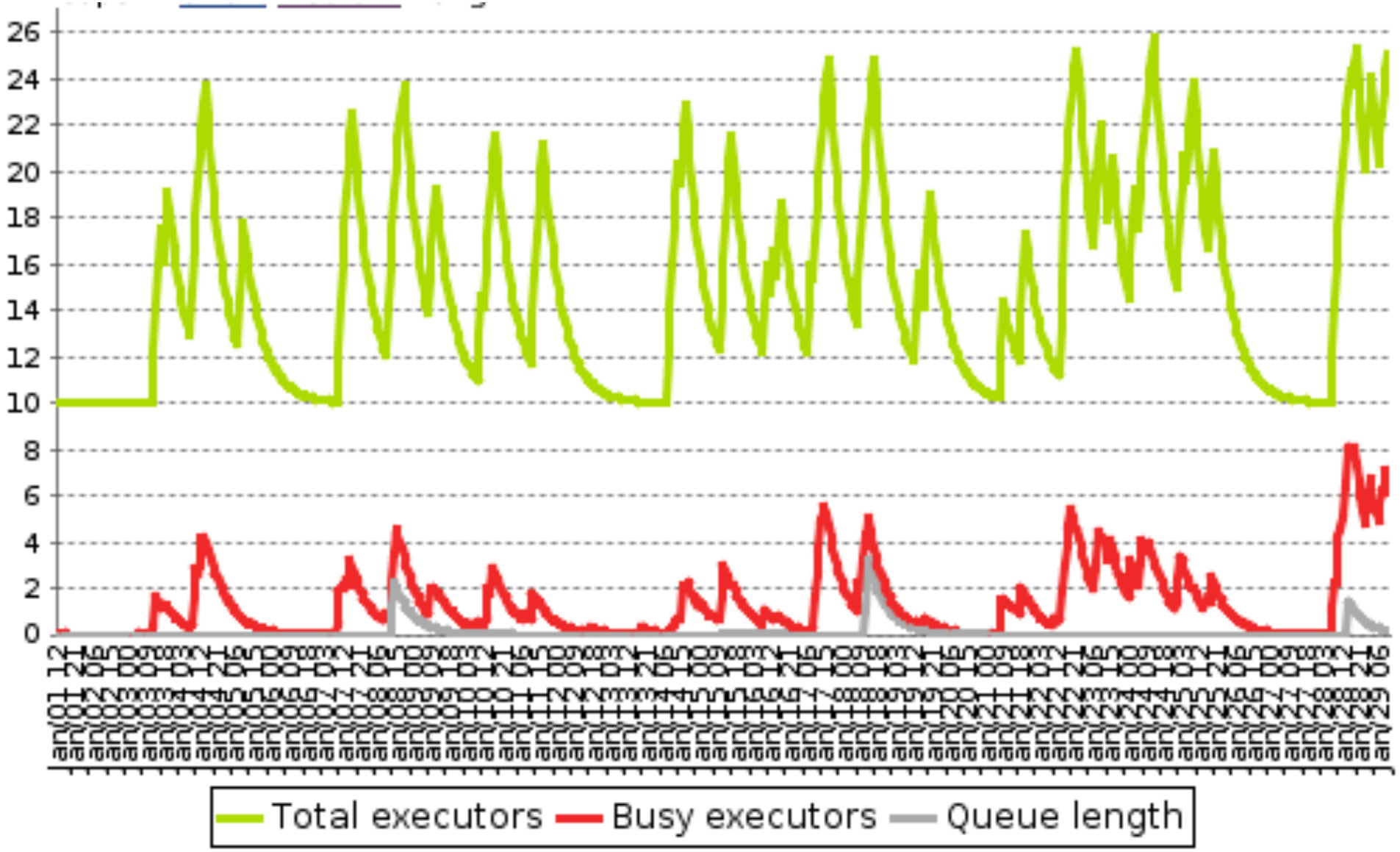
Scale your Jenkins pipeline

FOSDEM 2013

Anders Nickelsen, PhD
QA engineer @ Tradeshift
@anickelsen, ani@tradeshift.com

TRADESHIFT®


IT WORKED FINE IN TEST
OPS PROBLEM NOW


26
24
22
20
18
16
14
12
10
8
6
4
2
0
Total executors
Busy executors
Queue length

Tradeshift

TRADESHIFT®

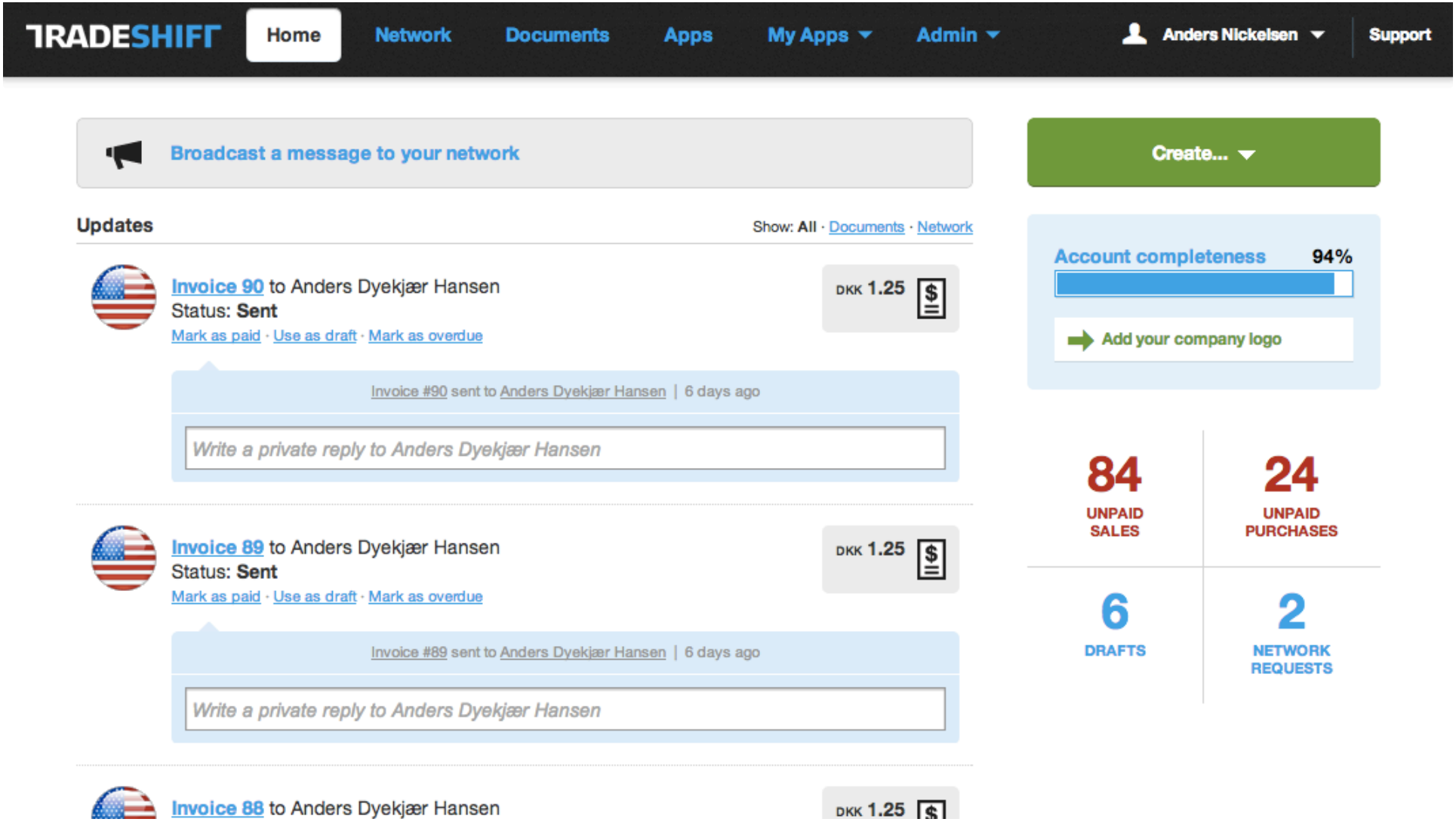
Tradeshift

Platform for your business interactions
Core services are free

~3 years old

One product – one production environment

~20 developers
    Copenhagen, DK and San Francisco, CA


TRADESHIFT
Home
Network
Documents
Apps
My Apps
Admin
Anders Nickelsen
Support
Broadcast a message to your network
Create...
Updates
Show: All · Documents · Network
Invoice 90 to Anders Dyekjær Hansen
Status: Sent
Mark as paid · Use as draft · Mark as overdue
DKK 1.25
Invoice #90 sent to Anders Dyekjær Hansen | 6 days ago
Write a private reply to Anders Dyekjær Hansen
Invoice 89 to Anders Dyekjær Hansen
Status: Sent
Mark as paid · Use as draft · Mark as overdue
DKK 1.25
Invoice #89 sent to Anders Dyekjær Hansen | 6 days ago
Write a private reply to Anders Dyekjær Hansen
Invoice 88 to Anders Dyekjær Hansen
DKK 1.25
Account completeness 94%
Add your company logo
84
UNPAID SALES
24
UNPAID PURCHASES
6
DRAFTS
2
NETWORK REQUESTS

Slow test is slow

TRADESHIFT®

Why scale? Fast feedback

Integration tests (IT): Selenium 2 from Geb (Java/Groovy)
Run by Jenkins on every change

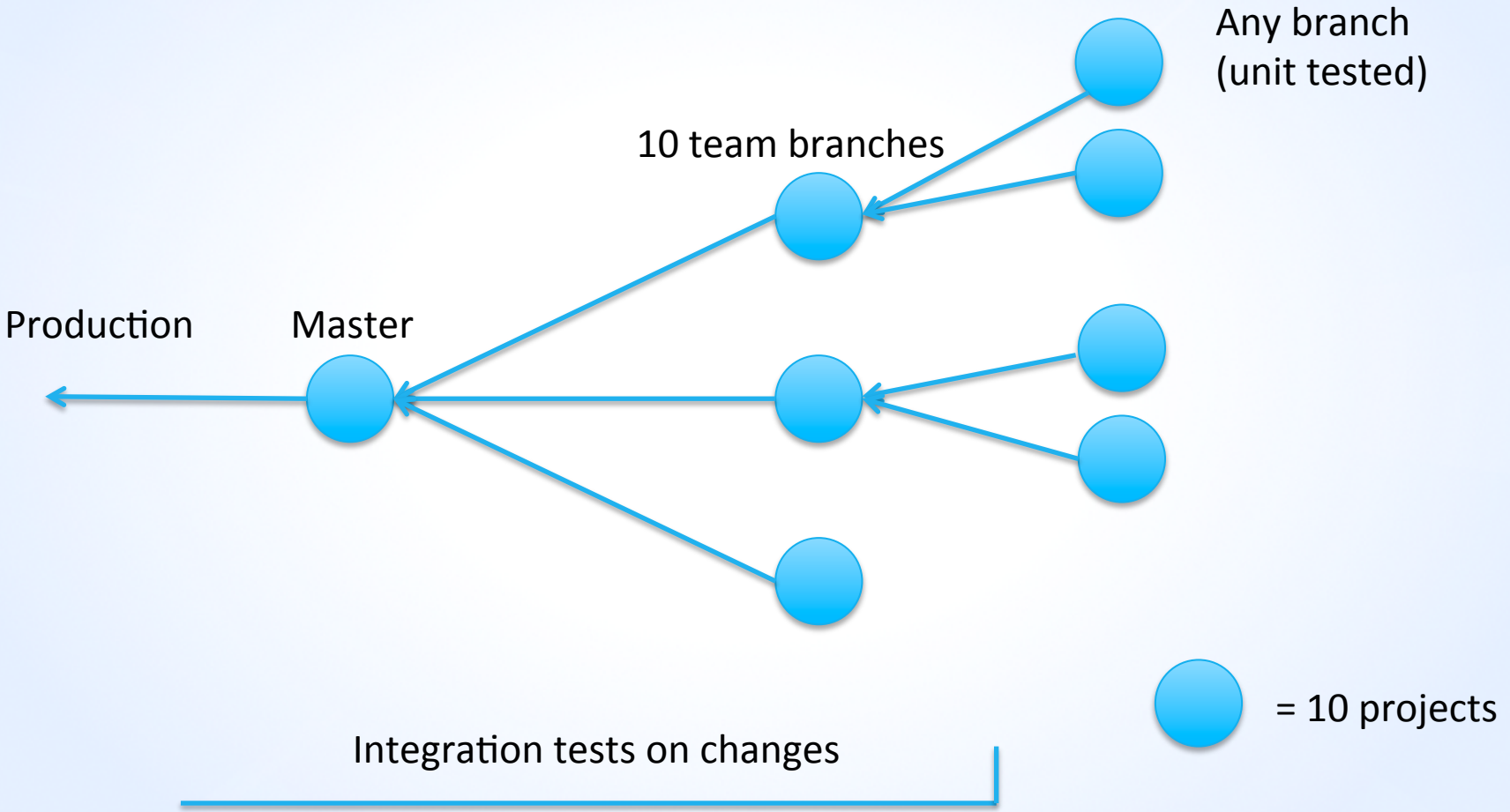
Takes **2 hours** to run all in sequence

10+ team branches, each trigger IT
Verified merges to master, also trigger IT

Pipeline is congestion point

Feedback not fast enough

Release pipeline



The swarm

TRADESHIFT®

What?

12 Jenkins slaves
- New slaves join swarm on boot
- Orchestrated by ec2-collective
- Configured by Puppet at boot

Tests distributed in sets
Longest processing time
Uses test run-time from last stable build

1 set per online slave (dynamic)

```
13:11:29  Creating 12 test sets with durations:
13:11:29  Test set 1: 565.9100000000001 seconds.
13:11:29  Test set 2: 565.9580030000001 seconds.
13:11:29  Test set 3: 567.809022 seconds.
13:11:29  Test set 4: 569.2750401 seconds.
13:11:29  Test set 5: 574.8729999999999 seconds.
13:11:29  Test set 6: 575.310014 seconds.
13:11:29  Test set 7: 576.8799930000001 seconds.
13:11:29  Test set 8: 577.4440030000001 seconds.
13:11:29  Test set 9: 578.6210100000001 seconds.
13:11:29  Test set 10: 578.6440090000001 seconds.
13:11:29  Test set 11: 579.663997 seconds.
13:11:29  Test set 12: 610.90393 seconds.
```

Fast tests!

All sets are put into Jenkins build queue

Picked up by any slave

- throttled to one per slave

12 slaves => 20 minutes

- Tests: 10 min
- Overhead: 10 min

24 slaves => 15 minutes

Post processing

Join when all sets complete
- Triggered builds are blocking

Collect test results
- JUnit, HTML, screenshots, videos, console logs = 500 MB/run
- Curl request to Jenkins API

Process test results
- JVM outputs, slice console logs, archive artifacts
- Custom groovy script

Optimizations

TRADESHIFT®

Optimizations

Only on ITCase file-level, file scan

Only on spec level

- min time = longest running spec

Only scale test processing time

- 10 minutes today

Lessons learned

TRADESHIFT®

Parallelization overhead

Swarm setup and tear down

- node initialization and test result collection

Tests break a lot when run in parallel

Fixing tests and code hurts

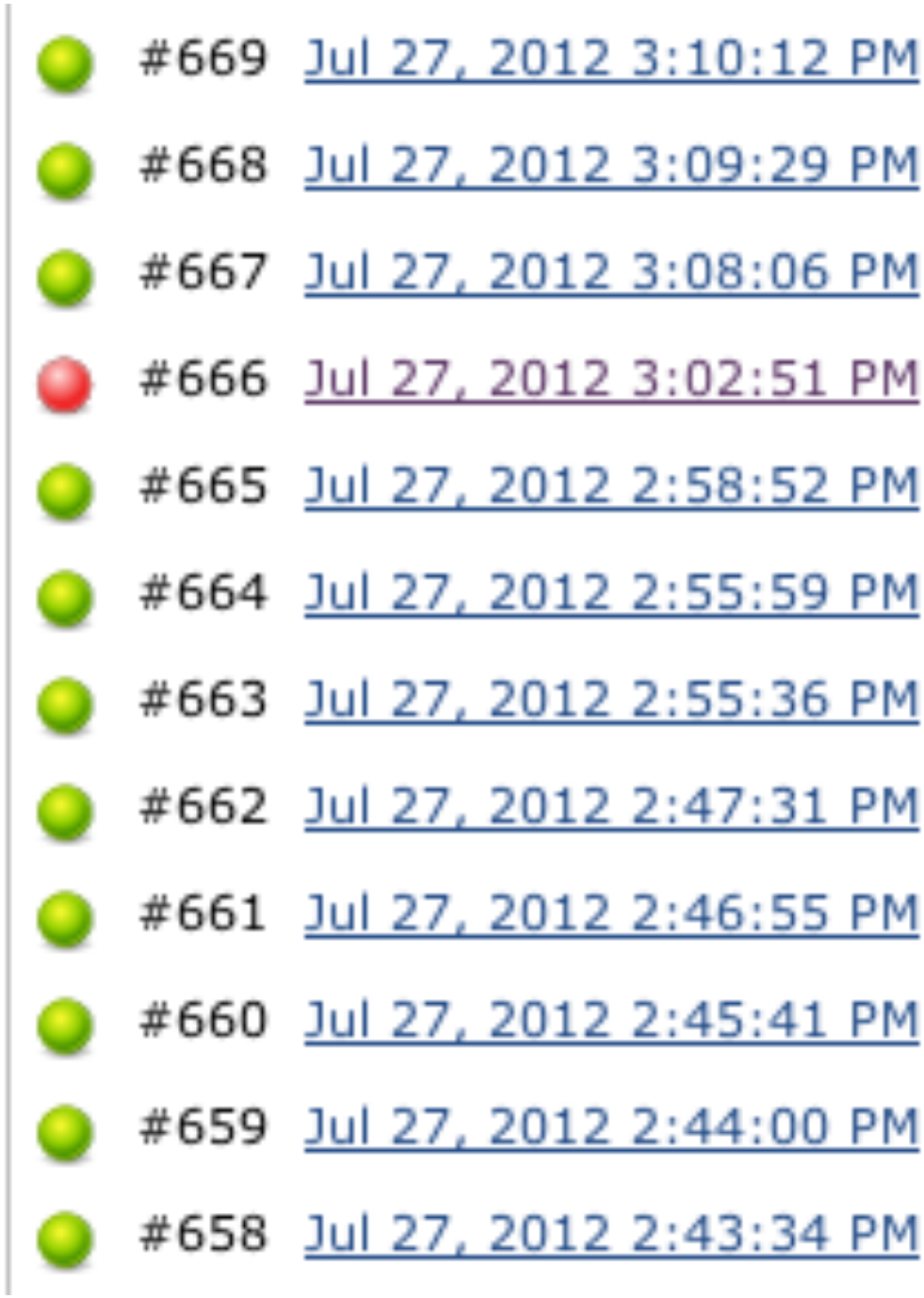
‘Random failures’

Failures appear probabilistic / random

- Order dependencies
- Timing issues
- Rebuild ‘fixes’ the tests

Slave failures

- Weakest link breaks the pipeline
- 24 slaves => 1 filled up due to slow mount => pipeline broken


#669 Jul 27, 2012 3:10:12 PM
#668 Jul 27, 2012 3:09:29 PM
#667 Jul 27, 2012 3:08:06 PM
#666 Jul 27, 2012 3:02:51 PM
#665 Jul 27, 2012 2:58:52 PM
#664 Jul 27, 2012 2:55:59 PM
#663 Jul 27, 2012 2:55:36 PM
#662 Jul 27, 2012 2:47:31 PM
#661 Jul 27, 2012 2:46:55 PM
#660 Jul 27, 2012 2:45:41 PM
#659 Jul 27, 2012 2:44:00 PM
#658 Jul 27, 2012 2:43:34 PM

Cost optimization

AWS EC2 instances:

- 1/2 price gave 3/4 performance
- 4/3 swarm size gave 1/4 price reduction
- Equivalent performance

Swarm is offline when not used

- Kept online for 1 hour after use


25
20
15
10
5
0
Total executors
Busy executors
Queue length

Credits

puppetlabs.com
github.com/andersdyekjaerhansen/ec2_collective

Jenkins and plugins

- Swarm plugin, parameterized trigger, envinject, rebuild, join, throttle concurrent build, conditional build step

More details of our setup

- tradeshift.com/blog/just-add-servers/

We're hiring!

- **tradeshift.com/jobs**