Release Management
From Packages to Production

Philip J. Hollenback
philiph@yahoo-inc.com

April 2011

Hi There

- Yahoo! Mail
  - maybe you've heard of it?
- Release Management
  - we install the softwares
  - supported by many others in Service Engineering

Environment

- 10k freebsd machines
- Other teams handle the hardware
- We focus on just the software installs

Houston?

- Core mail backend servers means lots of legacy code
  - About 1500 software packages
- Mail must be 100% reliable
  - People expect email to always be available
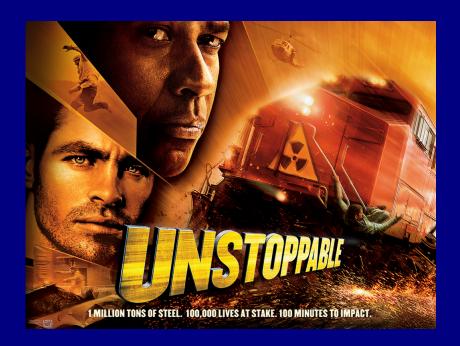- It's hard to load test mail service

Solution

- Distributed, cached software repositories with dist
- Keep system state in Igor
- Assemble releases with SRM
- Massively parallel pushes with Pogo
- Rigorous release cycle
- DO IT FAST
  - (3 week release timeline)

Release Philosophy

- All Packages All Environments
  - Packages must be installed on all servers of a given type
  - Only way we can properly test releases
  - Any other solution leads to many separate 'releases'

Which brings us to. . .

UNSTOPPABLE

1 MILLION TONS OF STEEL. 100,000 LIVES AT STAKE. 100 MINUTES TO IMPACT.

Release Trains

- Start a new train every 3 weeks
- Release Management and QA hand release back and forth
- Release assembly with SRM
- QA does integration
- QA final test & signoff
- Push to Prod in about a week (with Pogo)
  - That's 10k hosts worldwide, folks

Delays?

- Train delays suck
  - See the part about DOING IT FAST
- too much backup means we cancel
  - roll changes into next train
- QA is limiting factor

Massively Parallel Pushes

- Launch software push on lots of hosts at once
  - but with constraints & healthcheck

Starting a push to about 3000 hosts

```
$ pogo run -R Train_34 -I @mail.farm.xset.deploy-stage4-us
Password:
p0000031421; http://pogo.corp.yahoo.com/p0000031421
status: running: constraints computed
```

ⓨ!*Pogo*

Type a webpage address, title, or bookmark

Pogo ID	p0000008600
User	joonkim

Command	: train 28 restore
	yinst -v
	if ! yinst restore -igor -message "igor restore to mail.farm Train 28"
	then
	  echo "igor restore failed!"
	  sleep 30
	  exit 100
	fi

Invoked As	/home/y/bin/pogo run -C /home/joonkim/pogo-recipes -R t28 -I '@mail.farm.set.ac4-qa-6195'
Request Host	push3.mail.corp.sp1.yahoo.com
Retry	3
Timeout	21600
Start Time	Wed Sep 22 2010 18:50:39 GMT-0700 (PDT)
End Time	Wed Sep 22 2010 19:04:37 GMT-0700 (PDT) (+13m58s)
Job State	finished: no more hosts to run - ☑ Auto-Update

done to 6/6, 100% (6) finished, 0 retries

Host	State	RC	Time started	Duration	Timeline
web619502.mail.ac4.yahoo.com	finished	0	18:54:21(+3m42s)	+10m15s	
web619501.mail.ac4.yahoo.com	finished	0	18:52:39(+2m1s)	+1m40s	
vxs619502.mail.ac4.yahoo.com	finished	0	18:51:41(+1m3s)	+55s	
mix619501.mail.ac4.yahoo.com	finished	0	18:50:40(+1s)	+1m43s	
vxs619501.mail.ac4.yahoo.com	finished	0	18:50:40(+1s)	+1m0s	
mix619502.mail.ac4.yahoo.com	finished	0	18:50:40(+1s)	+33s	

Job Complete

- 10 hours to push to just over 3000 servers:

End of Push

```
CMR completed successfully
Start Wed Apr 13 2011 22:24:45 GMT-0700 (PDT)
End Thu Apr 14 2011 08:24:44 GMT-0700 (PDT) (+10h0s)
Total Hosts : 3046 (Skipped 36 invalid hosts)
Finished : 3038
Completion % : 99.73%
```

ps - open source pogo can be found at github.com/nrh/pogo

Yes, We Do Rollbacks

- Rigorous Release Testing
  - forward and backward
  - bare metal
- Full rollback roughly every 6 months

What Works

- massively parallel pushes
- rollback
- rigorous release testing

Challenges

- getting developers to care about old code
- package quality, install/remove scripts
- separating bugfix and feature trains
- moving settings to central config servers

To Sum It Up

DON'T BE CLEVER!

- Install the same packages everywhere
- Test releases thoroughly before rollout
- The simple approach is most likely to succeed

Thanks for Listening!

Phil Hollenback

philiph@yahoo-inc.com

@philiph

www.hollenback.net

This Is A Team Effort

Jen Draper

Jerrod Kensil

Brian McNeff

Joon Kim

Shajeeb Muhammad

Prem Ramnath

Tisha Emmanuel

Pradipta Ghosh

Any Questions?