Building Scalable Systems from Startups to Enterprises



Jeremy Edberg Founder and CEO MinOps https://minops.com https://sql.bot

Email: <u>{anything}@jedberg.net</u> Twitter: @jedberg Web: <u>www.jedberg.net</u> Facebook: facebook.com/jedberg Linkedin: www.linkedin.com/in/jedberg

About me





26

2011		2010 2010 Off 4 2014 0 Other 2 ¹⁰¹		
2011 Qtr_4 87899182	2011 Qtr_2 50774121	3261657T	2010 Qfr 2499893	4
			2010 Qtr 2140771	-1 6
		2009		2008
2011 Qtr_3 63598243	2011 Qtr_1 41864535	2009 Qtr 21584569	4 2009 Qtr_2 16699861	12660
		2009 Qtr 16966484	3 2009 Qtr_1 15498622	2008 0 82:3072 2008 0h 655 1849

60 -50 -

10 -





Netflix Countries





Starting with the cloud







Monthly Page Views and Costs

Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr

US\$20 000,00





Monthly Page Views and Costs

US\$20 000,00

Cloud Native (Some folks call this Microservices)

10s of thousands of instances, thousands created and removed daily

Thousands of storage nodes, petabytes of data, nodes can be removed without harm





Advantages to a Service Oriented Architecture

• Easier auto-scaling

Easier capacity planning

Identify problematic code-paths more easily

Narrow in the effects of a change

More efficient local caching

Highly aligned, loosely coupled

- provide.

• Services are built by different teams who work together to figure out what each service will

• The service owner publishes an API that anyone can use and returns proper response codes

Mature companies spend 25% of their engineering resources on their internal platform



It's all about building a culture around DevOps

My DevOps way

- Everything is "built for three"
- Fully automated build tools to test and make packages
- Fully automated deployments and testing

both Dev and Ops

deployment

• A team that builds tools to make all this possible.

My DevOps way

- Independent teams responsible for
- Redundancy through multi-region

Freedom and Responsibility

production at any time

- Hire responsible adults and keep rules and policies to a minimum
- Developers can change any code in
- And things don't break (usually)

Freedom and Responsibility

Developers deploy when they want

 They also manage their own capacity and autoscaling

• And fix anything that breaks at 4am!

Freedom and Responsibility

Developers own their product from beginning to end

If the customer isn't happy, the developer shouldn't be happy





Prescriptive

- Inflexible
- Slow to change

Policies (How They Usually Work)

• Determined by others

Policies @nflx



- Descriptive
- As flexible as we are
- Describe what we choose to do/get
- Evolve quickly

Policies @nflx



Setting up your infrastructure

Infrastructure as Code

- Changes are routine, small, easy, and repeatable
- Resources are easily managed by users and disposable
- Enables continuous deployment and improvement
- Solutions can be easily tested, measured, and then rolled back



Automate all the things!

http://hyperboleandahalf.blogspot.com/2010/06/this-is-why-ill-never-be-adult.html

- Application startup
- Configuration 0
- Code deployment

• System deployment

Automate all the things!

Infrastructure as Code Challenges

- Losing track of servers and resources
- Configuration drift
- Snowflakes
- Fear of a fully automated system (lack of trust in oneself)

Frame: PT12H, End: 2012-01-01T12:00 UTC

previous search

lean startup

advanced search: by author, community...

search reddit

previous search

lean startup

advanced search: by author, community...

search reddit

Self Serve is the Key

- Let developers choose what metrics to submit
- What graphs they put on their dashboards
- What to alert on
- They are closest to the app, so they know best

Self Serve is the Key

• Allows business metrics to be tracked easily

 Can be very powerful combined with decorators

armory STAGES OF SOFTWARE DELIVERY EVOLUTION

Traditional Deployments

Data Center Mutable Deployments SSH into Prod Deployments = Events Manual & Error Prone Dev vs. Ops

STAGE 2

Evaluating Continuous Delivery

Lift & Shift to Cloud Complicated Rollbacks No Service Ownership No Self-Service Inconsistent Deploys Non-standardized SLA Failures

Frequent Outages 20+ Manual Steps Weeks/Months to Deploy 1-2 Deployments/Month

Some Outages 10+ Manual Steps Days/Weeks to Deploy 2-10 Deployments/Month

Delivery Adoption Dedicated DevOps Immutable Deploys Confident Rollbacks Manual Judgements Manager Approvals Strong Integration Test Coverage

Few Outages 1 to 3 Manual Steps Hours to Deploy 10-20 Deployments/Month

Continuous

Continuous Deployment Adoption

Deploy Continuously in Background

> Full Embrace of DevOps Culture

Monolith Apps into Microservices

App Teams Fully Self-Service

All Teams Deploy with Same Platform

Intelligent Deployments

Automated Canaries

Automated Rollbacks

Machine-Learning Powered Anomaly Detection

> SLA Transparency on Per-App Basis

Chaos Engineering

Automated Dependency Analysis

Minimal Outages 0 Manual Steps Minutes to Deploy 100+ Deployments/Month

Rare Outages 0 Manual Steps Minutes to Deploy 1000+ Deployments/Month

Choose business metrics, not machine metrics
Choose groups of machines, not individual machines.

Alert on increase of failure, not lack of success



Increase in 500s



Decrease in 200s



P50, P90, P99



You can estimate percentiles on large (or streaming) datasets

- t-Digest
- Trades accuracy for speed and memory footprint
- Actually more accurate with larger datasets
- Error is worse at 50th percentile than at 95th or 5th





Anscombe's quartet







Anscombe's quartet



<u>Coefficient of determination</u> of the linear regression

- Mean of x 9
- Sample <u>variance</u> of x 11
 - Mean of y 7,50
- **Sample variance of y** 4,125
- **Correlation** between x and y
 - **Linear regression** line
- 0,816 y = 3.00 + 0.500x
- 0,67

Use queues as often as possible

Put Queues Everywhere and then Monitor Queue Lengths

Queuing

- Queue anything you are writing to a data store
- Monitor your queue lengths for great insight and scaling!





ltems

Seconds





Capacity utilization increases queues exponentially

- Every time you reduce the excess capacity by 1/2, you double the average queue size.
- This has a direct effect on the ratio of wait time to work time for a single work unit
- Use this to balance cost vs. latency

10 8 6 4 50 20 30 60 80



• Variability increases queue sizes linearly

• Operating at high utilization increases variability

The price of variability





The price of variability



The price of variability











Chaos Engineering

- Simulate things that go wrong
- Find things that are different

Two most important things to test



Increased Latency







NETFLIX

All systems choices assume some part will fail at some point.

The Monkey Theory





Simulate things that go wrong
Find things that are different

The simian army



- Chaos -- Kills random instances
- Chaos Gorilla -- Kills zones
- Chaos Kong -- Kills regions
- Latency -- Degrades network and injects faults
- Conformity -- Looks for outliers

- Circus -- Kills and launches instances to maintain zone balance
- Doctor -- Fixes unhealthy resources
- Janitor -- Cleans up unused resources
- Howler -- Yells about bad things like Amazon limit violations
- Security -- Finds security issues and expiring certificates

Chaos Monkey

• The first simian

- Randomly kills instances in production
- Used to catch a lot of errors, now everyone builds around it.





 Kills whole zones (datacenters)

- Finds errors with imbalanced request rates and data synchronization problems
- Also finds missing and bad fallbacks and backoffs.

Chaos Kong

- Shifts traffic from one region to another
- Find scaling misconfigurations, load shedding issues and load balancer
 issues.





Latency Monkey

 Injects random latency into network calls



Janitor Monkey

Cleans up unused resources



Howler Monkey

• Complains about things you don't have control over and require human intervention

• ex. Amazon limits



Best Practices for Data



- Have multiple copies of all data
- Keep those copies in multiple datacenters
- Avoid keeping state on a single instance
- No secret keys on the instance

Incident Reviews Ask the key questions:

- What went wrong?
- How could we have prevented it?
- How can we prevent this class of problem in the future?
- next time?

• How could we have detected it sooner?

How can we improve our behavior for

Serverless computing is all about speeding up development by allowing rapid iteration and removing management overhead



Remote Policy vs Remote Work

- Your culture needs to be "Remote First". You can't just dictate culture through policy
- You can't just have an odd person out, or it will fail
- reddit



Distributed Computing and a Distributed Workforce









- The two go hand in hand when you have a good distributed systems culture
- Microservices and Micro Teams



Timezones Matter Less, but Still Matter

- There needs to be some overlap to get things done
- More than about nine hours span across the company becomes tough to deal with
- At least try to have people who work on the same things overlap a few hours a day





Large Open Source Projects are a Great Model

This repository Search		Pull requests	Issues Gist			📌 +• 🚆•		
📮 torvalds / linux					★ Star	27,445 % Fork 10,885		
<> Code 1 Pull requests 83								
Linux kernel source tree								
🕝 560,620 commits	T 560,620 commits		S 441 releases			ଙ୍କି 5,669 contributors		
					-			
Branch: master - New pull re	New	file Find file	HTTPS +	https://github.com/	torvald: 🔁	Download ZIP		
torvalds Merge tag 'hwmon-for-linus-v4.4-rc2' of git://git.kernel.org/pub/scm/ Latest commit 0d77a12 30 minutes ago								
Documentation	Merge branch 'upstream' of git://git.linux-mips.org/pub/scm/ralf/upst 3 days ag							
arch	Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net 19 he							
block	block: fix blk-core.c kernel-doc warning					7 days ago		
certs	certs: add .gitignore to stop git nagging about x509_certificate_list					28 days ago		
Crypto	X.509: Fix the time validation [ver #2]					6 days ago		
drivers	Merge tag 'hwmon-for-linus-v4.4-rc2' of git://git.kernel.org/pub/scm/ 31 minutes ago							
Firmware	firmware: Update information in linux.git about adding firmware 7 months ago							
🖬 fs	FS-Cache: Add missing initialization of ret in cachefiles_write_page() 2 days ago							
include	Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net 19 hours ago							
init	sys_membarrier(): system-wide memory barrier (generic, x86) 2 months ago							
ipc	ipc,msg: drop dst nil validation in copy_msg 12 day					12 days ago		
kernel	Merge branch 'perf-urgent-for-linus' of git://git.kernel.org/pub/scm/					3 days ago		
🖬 lib	Merge branch 'akpm' (patches from Andrew)					8 days ago		
🖿 mm	Merge branch 'akpm' (patches from Andrew)					8 days ago		
net	Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net					19 hours ago		
samples	Merge git://git.kernel.org/pub/scr	m/linux/kernel/git/d	davem/net			19 hours ago		
scripts	Merge branch 'misc' of git://git.kernel.org/pub/scm/linux/kernel/git/ 8 days ago							
security	Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net 8 days ag							

- The Linux Kernel
- Motivated people who communicate only via email and chat.
• Use the cloud

Microservices and DevOps

Empowered engineers

Infrastructure Automation

Pulling it all together

- Monitoring the right things
- Chaos testing and reliability
- Serverless
- Remote first culture

Questions?

Email: <u>{anything}@jedberg.net</u> Twitter: @jedberg Web: <u>www.jedberg.net</u> Facebook: facebook.com/jedberg Linkedin: www.linkedin.com/in/jedberg Company: <u>minops.com</u>

