Launching products at massive scale: The DevOps way

Velocity, Amsterdam, 2016.
Who are we?

Kishore Jalleda
Senior Director, Production Engineering, Yahoo!
kjalleda@yahoo-inc.com

Gopal Mor
Software Architect, Yahoo!
gopalmor@yahoo-inc.com
Yahoo Scale

- 1+ Billion MAUs
- 6+ major data centers in strategic locations around the world
- 50+ edge PODs
- 400,000+ servers

Yahoo! News  
Yahoo! Sports  
Yahoo! Finance  
Yahoo! Fantasy  
......
Product redesigns at scale are non-trivial
We take feedback seriously!

@mlaguardia • Aug 30
@bgurley don't give up on Yahoo yet. We're rapidly responding to complaints and delivering new stuff. Keep the gripes coming and we'll fix!
Make sure to run a ton of experiments.
100+ iterations / experiments at any given time - almost every user is in some sort of an experiment

Validating metrics is not easy when you are dealing with a billion users - need to make the right decisions for the user.

Should not cannibalize other services like search and mail
And… there is “DevOps”
What is DevOps?

“DevOps is about eliminating Technical, Process and Cultural barriers between Idea and Execution -- using Software”

-Kishore Jalleda
The DevOps Way

(How)
The DevOps Way

DevOps

Culture

Process

Tools
The DevOps Way

Enable

Culture

Process

Tools

Ownership

Excellence
The DevOps Way

DevOps

Culture

Process

Tools

(Re)Usable

Self-Serve
Functional Pillars

(What)
Functional Pillars

DevOps

Deliver
... products to market quickly

Prevent
... defects from reaching customers

Repair
... production issues quickly
Functional Pillars

DevOps

**goal**
- sustainable velocity

**metric**
- velocity (time to market)

**use cases**
- provision
code
ship

**strategy**
- easy CD
cloudify
platformize

**culture & process**
- Agile
CD practices

**tools**
- CD pipelines
Cloud
Dev Tools

Deliver
Functional Pillars

DevOps

**goal**
prevent defects from reaching users

**use cases**
testing axes: func, perf, resilience, scale...

**culture & process**
test coverage
CD & launch gates

**metric**
quality

**strategy**
self-serve tools
expert services

**tools**
Disruptive Testing
Metrics based promotion

...
**Functional Pillars**

---

**DevOps**

- **Goal**: fix issues fast
- **Metric**: TT(R)
- **Use Cases**: detect, decide / diagnose, alert / remediate

---

**Strategy**
- directed alerting
- auto-remediation
- ...

**Culture & Process**
- Directed Alerting
- postmortems
- user feedback
- ...

**Tools**
- Monitoring
- Auto Remediation
- Product Health-Dashboards
- ...

---

**Repair**
In Summary...
In Summary...

Enable a Culture of Ownership & Excellence

Engineer Agile & Automated Processes

Develop (Re)Usable & Self-Serve Tools
to kick ass at...

Delivery Prevention Repair
(Product) Resilience

Resilience is critical to launching and operating products at a massive scale!

Let’s talk about it in detail!
Resilience at Yahoo Homepage and Media sites
Yahoo Homepage (www.yahoo.com)

- Among top 3 destinations on internet
- Personalized content
- Available in 22 internationals
- Page consists of multiple modules
- 99.999% availability
Availability Challenge

Hard to guarantee availability and latency in a ...

- Distributed multilayer architecture
- 100s of subsystems
- Complex request flow
- Change is the only constant
In this hypothetical example ...

- Each subsystem is highly available
- But combined system availability = 99.50%
- Downtime per year = 1 day, 19 hours, 49 min

The number against each box, in above figure, is availability of individual sub-system.
In this hypothetical example ...

- Each subsystem is highly available
- But combined system availability = 99.50%
- Downtime per year = 1 day, 19 hours, 49 min

The number against each box, in above figure, is availability of individual sub-system.
In this hypothetical example ...

- Each subsystem is highly available
- But combined system availability = 99.50%
- Downtime per year = 1 day, 19 hours, 49 min

**Combined system is weaker than the weakest subsystem.**

The number against each box, in above figure, is availability of individual sub-system.
How we ensure high availability

Four layers of resiliency in serving stack

1. Speculative Retry
2. Per module fallback
3. Fullpage failsafe
4. Failwhale Be-Right-Back page
Speculative Retry

- Trigger a retry when latency is higher than threshold
- High success rate for retry due to low latency at p95
- Addresses long tail latency and intermittent failures

Long tail latency
Speculative Retry

Not drawn to scale
Speculative v/s Backup

Speculative Retry Request

- Retry only when needed
- Need extra servers based on max retry rate

Backup Request

- Always send a backup request
- Need twice number of servers
- Need twice network resources
Speculative Retry

Few more considerations

- Useful for idempotent requests only
- Define max retry rate
- Prefer new connection for retry
- Track retry requests
- Use feature flag to turn on/off
Speculative Retry - Results

Spec retry helps reduce fallback rate by big margin.
Per Module Fallback

- Serve cached content for failed module
- Non personalized content
- Addresses prolonged failure of subsystem(s)
Per Module Fallback

- Not possible for cases like
  - Real time data (Example - sports scores)
  - Personal info (Example - stock tickers)
Non personalized cache, for each module, is always available on frontend servers
Per Module Fallback

Make sure that ...

- Cache is always fresh
- Strong validation needed on cache data
- Check for backward compatibility if TTL is high
Per Module Fallback - Results

- It is a degraded experience
- Keep it as low as possible
Fullpage Failsafe

- Cache entire page
- Non-personalized
- No ads
- Min interactions
- Used when page cannot be served

Entire page served from cache.
Fullpage Failsafe

No single point of failure between serving stack and failsafe stack
Fullpage Failsafe

- Using autoscale on AWS
- Automatic or manual switch
- Fine control on amount or type of traffic
- Helpful during unprecedented traffic spike
- Monitor cache freshness, failsafe traffic
Failwhale

Looks familiar?
Failwhale

- Last resort when everything fails
- All hands on deck situation
- This page is served from edge

Will be right back...

Thank you for your patience.
Our engineers are working quickly to resolve the issue.
Summary

1. Analyze entire range of failure types
2. Understand their rate and impact level
3. Holistic plan to cover all failure types
4. Fire drills - Test, Test, Test

Remember that Murphy's law is not on our side.

Anything that can go wrong, will go wrong.
Thank you!

CREDITS

Shay Holmes
Rashmi Tenginka
Santosh Mandi
Pushkar Sachdeva
Dreux Ludovic
Sandeep Davu
Karthikeyan Thangaraj
Phil Hayward
Natarajan Kannan