CSCI2952-F

Microservices.. Day 3: Migrations
Summary

• Administrivia (HW#1, Paper signups)

• Background Continued
  • eBPF
  • YAML
    • Service Discovery

• Migration
  • Motivation
  • Challenges

• Migration UseCases
• **Borg: the next generation** Submit Review
• **Systems @Scale 2019 - Scaling Cluster Management at Facebook with Tupperware**
• Resource Management with Deep Reinforcement Learning
• **Synthesizing Cluster Management Code for Distributed Systems**
• Summary
  • 2 paragraphs

• Interesting
  • One paragraph each:
    • observations/insights,
    • Assumptions
    • Design choices

• Limitations: one paragraph

• Future Directions: one paragraph

• Graded on: completeness, clarity, innovations
Background
Netronome Primes High-Performance Firewalls Based on eBPF/XDP as Data Center Operators Upgrade From Iptables

Taking Advantage of Extended Berkeley Packet Filter (eBPF) and Express Data Path (XDP) Technologies, New Hardware-Accelerated Solutions Pave the Way for Tighter Security and Faster Innovation for Data Center Core and Enterprise Edge Applications

Tigera Embraces eBPF to Advance Container Networking

https://containerjournal.com/topics/container-networking/tigera-embraces-ebpf-to-advance-container-networking/
eBPF Primer

- Allows user defined code to run in kernel

- Kernel containers predefined hook points

- Write program in restricted language (no loops, pointers, function calls)

- Program is compiled and loaded into hook point

YAML (YAML Ain't Markup Language)

What It Is: YAML is a human friendly data serialization standard for all programming languages.

Kubernetes Configuration file === YAML

You will see a bit of YAML in HW#1 but hopefully you’ll use tools that abstract YAML

https://itnext.io/why-should-we-prefer-the-yaml-file-over-the-properties-file-for-configurations-in-spring-boot-f31a273a923b
Develop: Service creation

Auto-generate starter material:

**Service sources**
- Python/Go/Node service stub
- Dockerfile
- CI configs

**Helm Charts**
- Friends don’t let friends write YAML!
Migration.....
• Why migrate?

• Which migration strategies?

• How to decompose a monolithic?

• What are the technical challenges?
Why do Companies migrate to a microservices?
Intentions

• Maintainability
  • Can’t update/understand/hard to change

• Scalability
  • Perf bottlenecks

• Functional Requirements
  • Outdated software -> hard to get correct functions

• Operability
  • Smaller components → easier to manage

• Company Strategy
  • Easily support multivendor strategy

• Team Scalability

• Time to Market
  • Need faster development (6months to get changes to production)

• Interoperability

• Reliability
Intentions

- Maintainability
  - Can’t update/understand
- Scalability
  - Perf bottlenecks
- Functional Requirements
  - Outdated software
- Operability
  - Smaller components
- Company Strategy
  - Easily support multiv
- Team Scalability
- Time to Market
  - Need faster development (6months to get changes to production)
- Interoperability
- Reliability

Which do you think is the most popular?
Intentions

- Maintainability
  - Can’t update/understand/hard to change
- Scalability
  - Perf bottlenecks
- Functional Requirements
  - Outdated software -> hard to get correct functions
- Operability
  - Smaller components → easier to manage
- Company Strategy
  - Easily support multivendor strategy
- Team Scalability
- Time to Market
  - Need faster development (6months to get changes to production)
- Interoperability
- Reliability
What does a migration consist of?
Strategies for Migration

- Rewrite
- Strangler Pattern
  - Gradually replace existing systems with microservices
- Extension
  - Extending with microservices
- Parallel Operation
  - Old and new running side by side
- Greenfield
  - Starting from no code

https://www.travelblog.org/Photos/8691056
https://www.infoq.com/articles/microservices-intro/
Strategies for Decomposition

• Other (or non-systematic)

• Functional Decomposition

• Existing System’s Structure
  • Reuse preexisting partition system

• Domain-Driven Design
  • Business specific domain boundaries

http://www.waynecliffordbarker.co.za/functional-decomposition-for-microservices/
Strategies for Decomposition

• Other (or non-systematic)

• Functional Decomposition

• Existing System’s Structure
  • Reuse preexisting partition system

• Domain-Driven Design
  • Business specific domain boundaries

Strategies for Decomposition

• Functional versus Domain-Driven
  • What are the pros/cons?

Domain-Driven Design

- Business specific domain boundaries

Function design

- Break down into functionality
- Identify the different roles in your business and functionality to support them
Strategies for Decomposition

• Functional versus Domain-Driven
  • Functional: more components with lots more interactions
  • Domain-driven: redundancy in each microservices
What makes a migration challenging?
Migration Challenges

• Decomposition 8
  • Unclear how to correctly decompose service

• Lack of Expertise 8
  • Insufficient knowledge about microservices OSS

• DevOps and Automation 6
  • Old dev tools don’t work in microservice environment

• Integration of Services 4

• Legacy System 4

• Security 4
Group Exercise
Imagine you have to do a migration

• Which Migration strategy would you use? And why?
  • Rewrite, Stranger, Extension, Parallel Operation, Greenfield

• Which strategies would you use for decomposition?
  • Functional, existing system structure, domain-driven design

• https://forms.gle/aDNUYGmj8Mua4BXd8
Strategies for Migration

- Rewrite 9
- Strangler Pattern 7
  - Gradually replace existing systems with microservices
- Extension 7
  - Extending with microservices
- Parallel Operation 5
  - Old and new running side by side
- Greenfield 4
  - Starting from no code

https://www.travelblog.org/Photos/8691056
https://www.infoq.com/articles/microservices-intro/
Strategies for Decomposition

- Other (or non-systematic) 9
- Functional Decomposition 7
- Existing System’s Structure 5
  - Reuse preexisting partition system
- Domain-Driven Design 3
  - Business specific domain boundaries

Migration Case Studies.....
Financial Times (@2018)

• Motivation: lower costs (80% reduction in ec2 costs)
  • Adopt OSS and reduce cost of maintain in-house tools

• Approach: Run in parallel (comes at a cost)
  • Over 2000 code release while running in parallel
  • Increase costs of running in parallel

• Challenges (how to keep thinks in parallel)
  • How to commit code changes: Separate branches vs **if/else in code**
  • Deployment mechanisms: **Separate (decouples deployments)** vs a single mechanism (avoid sync problems)
  • Testing: both or just the production (**same testing code+mirror**)

• Lessons
  • A standard that isn’t enforced may will not be complied with.
  • Services that hadn’t been built for a long time may not work
  • K8 moves services all the time but service may not work after moved

https://www.youtube.com/watch?v=HU6qrNmGqyE
Tinder (@2019)

- **Motivation:**
  - Fast scale times (VMs versus container)
  - Low touch operations

- **Approach:** Existing system structure
  - Change service to service calls to use ELB
  - Move individual services over

- **Challenges**
  - Managing code (polygon): new code build infrastructure (CI/CD)

- **Lessons**
  - Initially one node type → multiple node types

GoPay (@2019)

- **Motivation:**
  - Better Cluster Management
  - Easier and faster cluster setup
  - Faster deployments

- **Approach:**
  - one service at a time
  - Run in Parallel (for fault tolerance— at least one instance in non-kubernetes)

- **Challenges**
  - Managing code (polygon): new code build infrastructure (CI/CD) – language specific helm charts + GitLab script for deployment
  - New logging (BaritoLog)
  - No health checks

Reddit (@2019)

• Motivation:
  • Small infra team: engineering too dependent on infra team
  • Infra team is bottleneck to deployments (need more automation)

• Approach:
  • one service at a time
  • Run in Parallel (for fault tolerance— at least one instance in non-kubernetes)

• Challenges
  • Setup standards that are irrespective of languages (RPC, metrics, tracing, log output)
  • Code deployment (spec → YAML)
  • Security/policy enforcement/specification

Summary

• Discussed background: eBPF, ServiceDiscovery, YAML

• Overview of Migration
  • Motivation
  • Approaches
  • Challenges

• Discussed migration cases studies
  • Financial Times, Tinder, Reddit, GoPay