appfolio™

Web Operations @ AppFolio

Outline

- About AppFolio.
- Who am I?
- Web Operations in General.
- Web Operation @ AppFolio.
- War Stories.
- Interested in WebOps?

About AppFolio

About AppFolio

- AppFolio creates complete, easy-to-use, cloud-based software for multiple vertical markets.
- Our products include cloud-based property management software, rental syndication and lead tracking software, and legal practice management software.
- Dr Mutz is our Chief Scientist!

Who am I?

Who am I ?

- Principal Software Engineer, Tech Lead @APPF
- Live and work in Dallas now.
- Joined AppFolio full-time in 2009.
- Played multiple roles:
    - application development.
    - tier-3 tech support.
    - database operations.
    - web operations.
    - ...

Web Operations

Web Operations

A domain of expertise that involves deployment, operation, maintenance, tuning, and repair of web-based applications and systems. (From Wikipedia)

If you add product management and application development, you get a full picture of web-based software.

Web Operations

What about these buzzwords?

- DevOps
- Site Reliability (Google)
- Production Engineering (Facebook)

Web Operations

Software engineering approach to web operations.

- Agile
- Automation
- Testing
- Cloud
- ...

Web Ops @AppFolio

Web Operations @ AppFolio

Web operations @ AppFolio include (but not limited to):

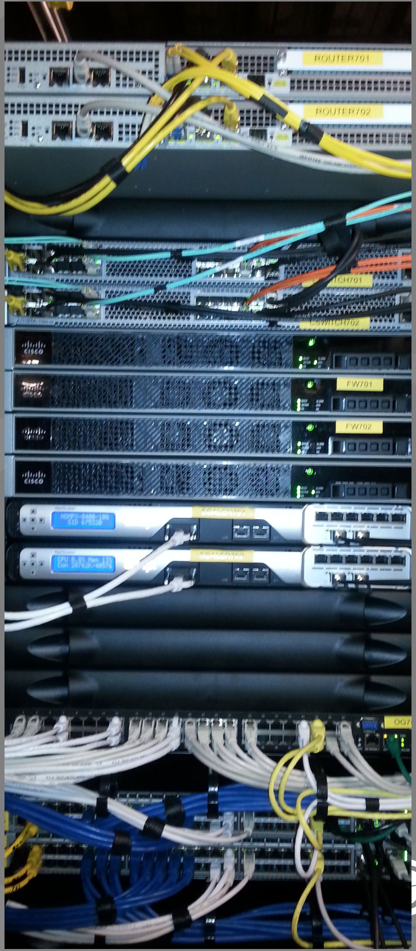
- DataCenter colocation.
- Application Infrastructure.
- Database operations.

DataCenter Colocation

DataCenter Colocation

DataCenter Colocation

DataCenter Colocation



DataCenter Colocation

- Choose colo vendor/location.
- Power.
- Cooling.
- Network.
- Cage and rack spec.
- Hardware spec.
- Capacity planning.
- Procurement.
- Rack hardware.
- Replacement/Repair.
- Cloud?

Application Infrastructure

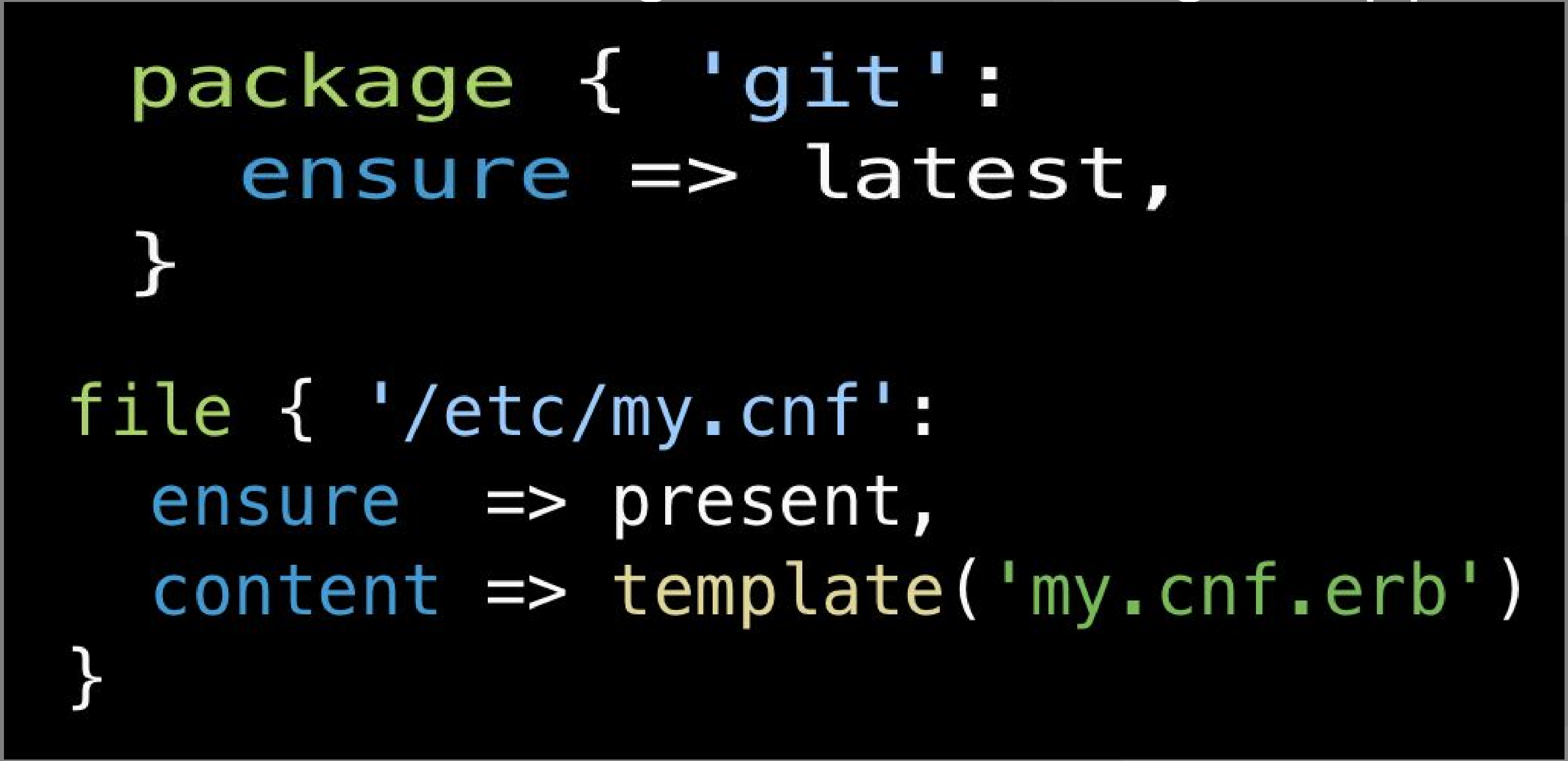
Application Infrastructure

Infrastructure to support AppFolio products.

- Software installation and configurations.
- Application deployment.

Application Infrastructure

Installation and configuration is through Puppet.



```
  package { 'git':
    ensure => latest,
  }

file { '/etc/my.cnf':
  ensure  => present,
  content => template('my.cnf.erb')
}
```

Application Infrastructure

Application Deployment.

- Fully automated.
- Canary release.

Database Operations

Database Operations

We provide reliable datastore for applications.

- NEVER lose business critical data!
- Secure.
- Available.
- Fast.
- Flexible.

Database Operations

- Installation.
- Configuration.
- Upgrade/Downgrade.
- Monitoring.
- Optimization.
- Backup/Restore.
- Replication.
- Debugging.
- Database Consulting.

Database Operations

- Automation is the key.
- Understand business logic.
- Understand best practices for relational databases.
- Understand how MySQL works behind the scenes.
- Occasionally *gdb mysqld* is needed.

War Stories

Mysterious Slowdown

Mysterious Slowdown

00:15 Received a flood of alerts.

00:17 Customer sites were down.

00:20 Most servers were unreachable.

01:00 There was a power outage in our colo data center.

01:30 Power had restored.

01:30 Servers started to boot.

02:10 All services were up.

02:15 Received some alerts.

02:20 Servers were reachable, but sites on our brand new servers were very slow.

Mysterious Slowdown

Top 3 reasons for slowness:

- Slow network at client-side. (A.K.A Blame the wifi).
- Slow network at our ISP. (A.K.A Blame the vendor).
- Slow SQL queries in database.

Mysterious Slowdown

Slow network at client-side.

- Google was slow too!
- Use wired network. (Do not use hotel WiFi!)
- Google became fast.
- But, AppFolio sites were still slow in my browsers.

Mysterious Slowdown

Slow network at our ISP.

- *ping* from multiple locations in the US did not show any noticeable delay.
- *traceroute* results were fine too.
- Our ISP confirmed that they did not experience any slowness.
- AppFolio sites were still slow in my browsers.

Mysterious Slowdown

Slow SQL queries in database.

- Listed all running SQL queries in our database.
- There were not many queries running.
- Queries were not very slow.
- Database was lightly loaded.
- Confirmed with our application performance monitoring tool.
- Tired and confused now.

Mysterious Slowdown

What's next?

- Read the request log to see if we could figure out some pattern.
- Most time of a request was spent in Rails.
- Confirmed with application performance monitoring tool.

Mysterious Slowdown

Rails was slow.

- Garbage collection was too long?
- APM tool showed GC was normal.
- *strace* output was slow.
- *perf* generated a flame graph which we could not really decipher.

Mysterious Slowdown

Other symptoms:

- *ssh* to a server was not as fast as before. It took a while to establish SSL connection.
- Installing *perf* was slow too.
- *top* refresh also lagged.
- My guess: the whole server was slow.

Mysterious Slowdown

To test my theory.

- *openssl speed* result was a surprise.
- Our expensive servers were as slow as the cheapest AWS EC2 instances.
- CPUs were too busy?

Mysterious Slowdown

Busy CPU

- But, *top* showed CPUs were not very heavily loaded.
- CPUs were slow?

Mysterious Slowdown

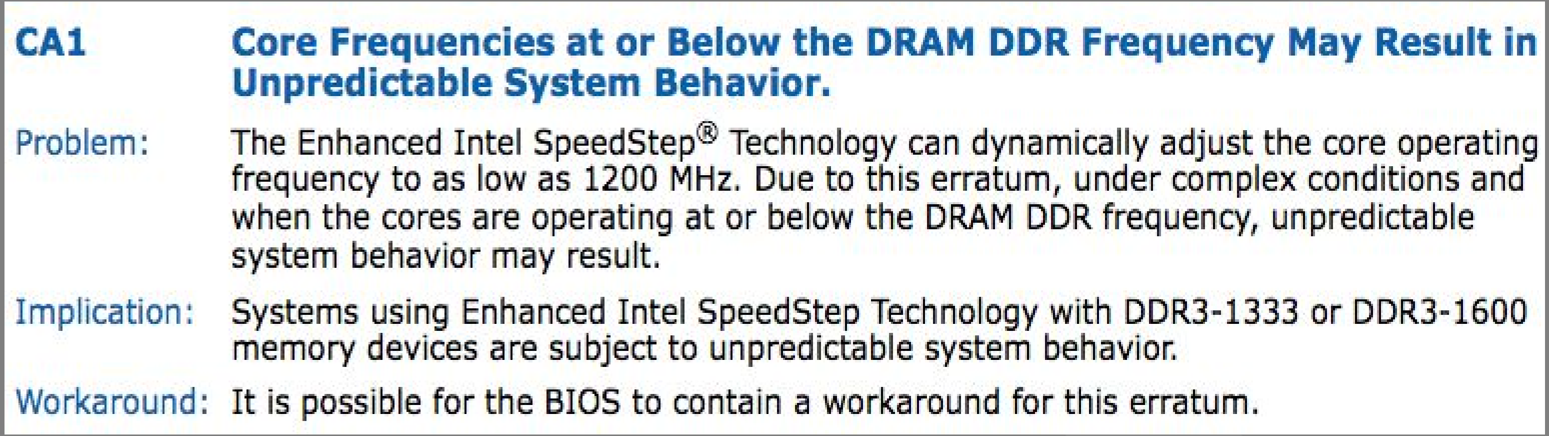
Slow CPU

- *i7z* showed all CPU cores were at 1.2GHz.
- All cores were also in the lowest power consuming mode.
- *openssl speed* did not make CPU cores to step up.
- Let’s disable CPU power saving feature in BIOS and reboot.
- Everything was fast again!
- Re-enabling CPU power saving feature in BIOS could reproduce the slowness.
- No more power saving. Sorry about planet Earth □

Mysterious Slowdown

Why?

- There was a bug reported in our model of Intel CPU.
- Power instability may make the bug more likely to happen.



CA1 **Core Frequencies at or Below the DRAM DDR Frequency May Result in Unpredictable System Behavior.**

Problem: The Enhanced Intel SpeedStep® Technology can dynamically adjust the core operating frequency to as low as 1200 MHz. Due to this erratum, under complex conditions and when the cores are operating at or below the DRAM DDR frequency, unpredictable system behavior may result.

Implication: Systems using Enhanced Intel SpeedStep Technology with DDR3-1333 or DDR3-1600 memory devices are subject to unpredictable system behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Mysterious Slowdown

What we learned?

- Double check power supply.
- Everything could go wrong.
- CPU monitoring needed to be improved.
- Better performance troubleshooting checklist.
- The same issue happened randomly in our testing environment, but we did not pay enough attention.
- Check Intel errata before purchasing CPUs.

What You Read
Is Not
What You Write

WYRINWYW

Background:

- HTTP requests were processed by a pool of processes which run Ruby on Rails code.
- Memcached is a process which provides key-value storage in memory (ideal for caching data).
- Each Ruby on Rails process used TCP to write data to or read data from the Memcached process.

WYRINWYW

Symptoms:

- A lot of time-out exceptions from Memcache-related code in Rails processes.
- Sometimes when a Rails process read from Memcached, the result was unexpected. For example, when an integer was expected, Memcached returned an array.
- A number of legitimate requests ended up with errors due to the cache anomaly.
- Our customers were not happy.

WYRINWYW

Initial attempt:

- *top* showed Rails processes used more memory than usual.
- Garbage collection should be slower.
- Time-out exceptions should be possible.
- Still no clue about cache anomaly.

WYRINWYW

Second attempt:

- *tcpdump* to inspect the network traffic between the Memcached process and Rails processes.
- Reads from Memcached always returned expected data!
- No cache anomaly when Memcached-related methods are executed in Rails console.
- But we still got errors.
- Totally confused!

WYRINWYW

Third attempt:

- Inspect *tcpdump* output while cache anomaly occurred.
- Check Rails logs.

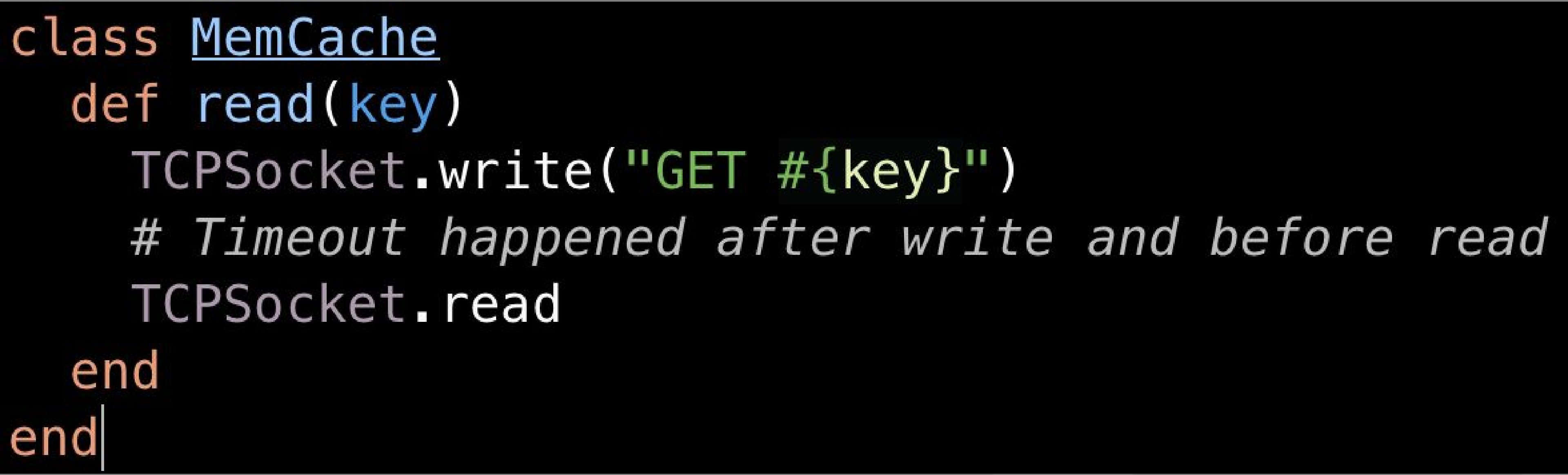
Findings:

- Cache anomaly usually followed time-out exception immediately.
- Both happened in the same Rails process.

WYRINWYW

Fourth attempt:

- Read Memcached-related code.



```
class MemCache
  def read(key)
    TCPSocket.write("GET #{key}")
    # Timeout happened after write and before read
    TCPSocket.read
  end
end
```

WYRINWYW

We had an idea:

- Each Rails process maintained a persistent TCP connection with the Memcached instance.
- Rails process wrote GET message into the connection.
- The Memcached instance sent the reply to the connection.
- The reply were in the OS’ socket buffer.
- Rails process had a time-out exception. So, *TCPSocket.read* was not executed.
- Next time, when the Rails process read from Memcached, it sent GET and read the socket buffer.
- However, the reply from the last Memcache GET was still in the socket buffer.
- Bingo!

WYRINWYW

Why time-out:

- Long garbage collection.
- Rails process object space dump.
- Found a lot of objects from our application performance monitoring tool.

WYRINWYW

What we learned:

- The overhead of performance monitoring could be huge.
- Exception related to IO should be handled carefully.

War Stories

How to deal with outages like this?

- Know the system well.
- Know your tools well.
- Plan ahead (checklist, cheat sheet, ...).
- Practice.
- Don’t panic and be patient under heavy pressure.
- Reason confidently with your computer science knowledge/skill.
- Team player.
- Business priority first.
- No blame.
- Learn from outages.

Interested in WebOps/DevOps/SRE/PE?

WebOps/DevOps/SRE/PE

Traits employers are looking for:

- Be able to understand complex systems.
- Mental toughness.
- Business first.
- Passionate about detective work.
- Attention to details.
- Well-rounded.

WebOps/DevOps/SRE/PE

- Take Algorithm, Data Structure, OS, Networks, PL/Compilers, Databases, and this class.
- Write a simple file listing/upload/download web service from scratch and host it somewhere. Do not use any existing software.
- Read some complex software source code like NGINX and try to figure out how a network request is processed using logging and debugging tools.

Q&A