Scaling a Distributed Container Management System: Diego Performance
Who Are We?

- James Myers
- Luan Santos
What We Will Cover

- A quick introduction to Diego
- Performance as product definition
- Designing performance experiments
- Evaluating Success and iterating to hit targets
- The future of Diego performance
What is Diego?
How much performance?

• Largely a product decision
  - We need N cells with N*K application instances

• Also engineering though
  - How does stress effect the system?
  - How fast can we operate vs how fast should we operate
Evaluating Performance

• Design Experiments
  – How do we artificially generate load in a realistic manner?
  – What types of load do we want to generate?

• Measure Performance
  – What metrics should we watch?
  – How do we characterize success?
Designing Experiments

• Smoke Tests (fezzik)
  – Run a large number of instances as fast as we can
  – Not truly a realistic scenario
  – Exposes obvious performance bottlenecks
Designing Experiments (cont.)

- Full scale performance tests
  - Push N*K CF applications to an environment backed by Diego
  - Perform smaller experiments on a loaded environment
  - Allow environment to “bake”
  - More realistic (production) load
Non Standard Performance Experiments

• Testing failure modes
  – ETCD failure
  – VM failure
  – How well does Diego converge?
Measure Performance

- Metrics emitted to Datadog
  - Bulk loop durations
  - Number of routes, LRP instances, etc

- Logs written to disk
  - Allows us to construct an LRP instance timeline through tagged log lines
Hitting our targets

• Early bottlenecks found quickly
  – Database server performance
  – Poorly tuned concurrency
  – JSON marshalling

• Solutions
  – BBS
  – More configuration
  – Protobufs
Results on 100 Cells

- Smoke Tests (fezzik)
  - 4k LRP Tasks in less than 30 seconds
  - Even less time spent in Diego components
  - No obvious bottlenecks
Results (cont.)

- Stress tests
  - 10k instances run successfully
  - No performance degradation over time
  - Let it sit for a week
Failure Modes

- **Killing ETCD**
  - How long did it take NSYNC to recover?
  - Applications kept running

- **Killing Cells**
  - Applications restarted on available cells until capacity is reached
  - Running favored over crashing
Rolling Deploys

- # of routes overtime stays constant
- # of running applications
- Extra evacuating instances and their impact
Conclusions from tests

- Successfully hit our first performance target
- Diego is extremely resilient to failure
- Room to grow
But what about more performance?

• 1000 cells? 200K instances?
  - Expensive to test
  - Hard to manage

• Benchmark-bbs
  - Focused tests to stress specific aspects of the system

• Astroturf
  - Don’t need to test garden to test Diego performance
Benchmark-BBS & Astroturf

- ETCD is a bottleneck
  - 200k instances is more data than ETCD can handle
  - Slow lookups, no indices, not a real database

- BBS performance degrades as number of instances increases
The Future

• BYOD (Bring Your own Database)
  – Mysql/Postgres
  – Built to handle large amounts of data at scale
  – Operators can choose the level of consistency needed

• Read BBS nodes?
  – Currently only one active

• In memory caching?
  – Remove read load from database
Thanks for listening. Questions?