Distributed scheduler hell
Story of ..

How we moved 100(s) of virtual machines, onto containers
Take aways

- What do Distributed schedulers do?
- Why?
- Best Practices
- What not to use it for
What is a distributed scheduler

Your cloud provider: Digital Ocean, AWS, Google.
For containers: Kubernetes, Nomad, Mesos
Single process
Kernel Provides

- Virtual Memory
- Process Isolation
- Disk storage
- Networking
- CPU scheduling
Distributed Scheduler Provides

- Container Deploy
- Virtual Machine Deploy
- Memory Quota
- Disk storage
- Networking
- CPU scheduling
- Scaling Instances
Why?

Microservices...
Deployment...
Lifecycle management
Vulcan : Distributed timeseries database

https://github.com/digitalocean/vulcan
Requirements

• 3 Gbits a second of Network traffic
• 20 TB Storage
• Sub 100ms Read times
• 100k writes a second
Mesos
From Static Partitioning to Elastic Sharing

Static Partitioning

Elastic Sharing
Deploying our App On Mesos
<table>
<thead>
<tr>
<th>ID</th>
<th>Memory (MB)</th>
<th>CPUs</th>
<th>Tasks / Instances</th>
<th>Health</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>/kibana</td>
<td>512</td>
<td>0.5</td>
<td>0 / 1</td>
<td></td>
<td>Deploying</td>
</tr>
<tr>
<td>/mesos-dns</td>
<td>512</td>
<td>0.2</td>
<td>3 / 3</td>
<td></td>
<td>Running</td>
</tr>
<tr>
<td>/search</td>
<td>2048</td>
<td>1</td>
<td>2 / 2</td>
<td></td>
<td>Running</td>
</tr>
</tbody>
</table>
Marathon Architecture
Kafka

Custom Scheduler
Cassandra
Pinning to specific Mesos nodes
Mesos Failure Modes

Network Partition FAIL
Counter Example: Nomad
Nomad Architecture
Custom Json Language

```json
{
  "application": {
    "name": "timeseries-ingestor",
    "scale": 1,
    "ingresses": {
      "timeseries-ingestor-health": {
        "scheme": "http",
        "container_port": 8001
      }
    },
    "containers": {
      "timeseries-ingestor": {
        "image": "docker.com/timeseries/ingestor",
        "image_tag": "fcb24ca",
        "ports": [8001, 9090],
        "env": {
          "CASSANDRA_ADDRESS_LIST": ""
        },
        "resources": {
          "cpu": "4",
          "memory": "4000"
        }
      }
    },
    "metrics": {
      "port": 8001,
      "path": "/debug/metrics"
    },
    "maintainer": "dummy@digitalocean.com"
  }
}```
Command Line Deploys

docc --contexts dev_env deploy timeseries-microservice1.json

Docc is our internal Kubernetes tool
Deployment tool with diffs to Kubernetes

"env": {
  "CASSANDRA_CQL_VERSION": "3.1.7",
  "CASSANDRA_KEYSPACE": "staging_vulcan",
  "CASSANDRA_KEYSPACE": "staging_vulcan_123",

Simliar to KubeDiff
Final Architecture
Load balancing

Pushing 3 Gbs to kubernetes using Flannel
Metrics
Upsides to Distributed Schedulers
How to choice your abstractation
Questions ?