Flink meet DC/OS
Deploying Apache Flink at Scale

Elizabeth K. Joseph, @pleia2
Ravi Yadav, @RaaveYadav
Part 1
Introduction to Apache Mesos, Marathon, and DC/OS

Part 2
Demonstration of demo data pipeline + Installing Flink on DC/OS

Part 3
DC/OS 1.9 key features for data services and beyond
Apache Mesos: The datacenter kernel

http://mesos.apache.org/
Mesos can’t run applications on its own.
A Mesos framework is a distributed system that has a scheduler.
Schedulers like Marathon start and keep your applications running. A bit like a distributed init system.
Mesos mechanics are fair and HA
Learn more at https://mesosphere.github.io/marathon/
Introducing DC/OS

Solves common problems

- Resource management
- Task scheduling
- Container orchestration
- Self-healing infrastructure
- Logging and metrics
- Network management
- “Universe” of pre-configured apps (including Flink, Kafka…)
- Learn more and contribute at https://dcos.io/
DC/OS Architecture Overview

DC/OS Services & Containers

- HDFS
- Jenkins
- Marathon
- Cassandra
- Flink
- Spark
- Docker
- Kafka
- MongoDB
- +30 more...

DC/OS

- Container Orchestration
- Security & Governance
- Monitoring & Operations
- User Interface & Command Line

ANY INFRASTRUCTURE

- Physical Servers
- Virtual Servers
- Private Cloud
- Public Cloud Providers (Google, AWS, Azure)
Interact with DC/OS (1/2)

Web-based GUI

https://dcos.io/docs/latest/usage/webinterface/
<table>
<thead>
<tr>
<th>CLI tool</th>
<th>API</th>
</tr>
</thead>
</table>
According to the December 2016 data Artsans-organized Apache Flink user survey just under 30% of respondents were running Flink on Apache Mesos


You may already be using Apache Mesos!

Version 1.2 of Flink includes support for Apache Mesos and DC/OS, “it is now possible to run an highly available Flink cluster on Mesos”

DEMOS

Demo data pipeline + Installing Flink on DC/OS
DC/OS 1.9 - Data Services Ecosystem

DATA SERVICES ECOSYSTEM

- Alluxio
- Couchbase
- Datastax DSE
- Elastic (ELK)
- Redis
- Apache Flink

OPERATIONS

WORKLOADS
DC/OS 1.9 - Operations

DATA SERVICES ECO SYSTEM

OPERATIONS

- Remote Container Shell
- Unified Metrics
- Unified Logging
- Deployment Failure Debugging
- Upgrades & Configuration updates

WORKLOADS

© 2017 Mesosphere, Inc. All Rights Reserved.
REMOTE CONTAINER SHELL

- Open encrypted, interactive, remote session to your containers
- Remotely execute commands for real time app troubleshooting
- Provide developers access to their own applications, not the entire host or cluster

```
my-laptop$ dcos task exec my-task /bin/bash
Starting /bin/bash in my-task ...
Connecting to remote my-task ...
```
UNIFIED LOGGING

- Access application, DC/OS and OS logs
- Easily troubleshoot applications with critical metadata such as container id and app id
- Integrate easily with existing logging systems
UNIFIED METRICS

- Single API for system, container and application metrics
- Metadata such as host id and container id are automatically added to assist in debugging
- Integrate easily with existing metrics systems
DC/OS: OPERATIONS

DEPLOYMENT FAILURE DEBUGGING

- Understand why your application is not deploying
- Understand which nodes in the cluster can accommodate the role, constraints, cpu, mem, disk and port requirements for your app
• Generate new config for cluster nodes

    $ dcos_generate_config.sh --generate-node-upgrade-script <installed_cluster_version>

• Single command upgrade script for individual nodes

    $ curl -O <Node upgrade script URL>
    $ sudo bash ./dcos_node_upgrade.sh
DC/OS 1.9 - Workloads

**DATA SERVICES ECOSYSTEM**

**OPERATIONS**

**WORKLOADS**

- Pods
- GPU based scheduling
PODS

- Schedule, deploy and scale multiple containers on the same host(s) while sharing IP address and storage volumes
- All containers in a pod instance run as if they are running on a single host in pre-container world
- Useful for migrating legacy applications or building advanced micro services (side car containers)
● Traditional monolithic apps on VMs usually have support services such as log shipper, message queuing clients

● Many support services assume col-location on same host, and local-host access to networking and storage

● Pods simplify moving legacy monolithic apps to containers, reducing risk and accelerating migrations
Advanced Micro Services patterns require colocating containers together

Support services include for example:

- Logging or monitoring agents,
- Backup tooling & Proxies
- Data change watchers & Event publishers

Pods simplify the building and maintenance of complex such microservices
GPU: WHY GPU?

- GPUs are needed for many machine learning and deep learning applications.
- GPUs are essential for real-time or near real-time machine learning models.
- GPUs deliver from 10X to 100X performance for some applications, resulting in lower $$$/IOPS and more productivity to data science teams.
- GPU applications include real-time fraud detection, genome sequencing, cohort analysis, and many others.
GPU BASED SCHEDULING

- Test Locally with Nvidia-Docker, deploy to production with DC/OS
- Isolate GPU instances and schedule workloads just like CPU and memory, guaranteeing performance
- Efficiently Share GPU resources across data science team
- Simplify migrating machine learning models across from dev to production, and across clouds
OTHER IMPROVEMENTS

- Mesos 1.2
- Marathon 1.4
- Docker 1.12 and 1.13 (17.03-ce) support
- Centos 7.3 and CoreOS 1235.12.0 support
- Performance improvements across all networking features.
- CNI support for 3rd party CNI plugins.
- 100s of additional bugfixes and tests