NVIDIA GPU Support for Apache Mesos and DC/OS

GPU Technology Conference - 2017
Kevin Klues
klueska@mesosphere.com
Kevin Klues is a Tech Lead Manager at Mesosphere working with both the Mesos core team as well as the DC/OS Provisioning and Management team. Since joining Mesosphere, Kevin has been involved in the design and implementation of a number of Mesos’s core subsystems, including GPU isolation, Pods, and Attach/Exec support. Prior to joining Mesosphere, Kevin worked at Google on an experimental operating system for data centers called Akaros. He and a few others founded the Akaros project while working on their Ph.Ds at UC Berkeley. In a past life Kevin was a lead developer of the TinyOS project, working at Stanford, the Technical University of Berlin, and the CSIRO in Australia. When not working, you can usually find Kevin on a snowboard or up in the mountains in some capacity or another.
What is Apache Mesos?

- An open-source, distributed systems kernel (a.k.a cluster manager) for fine-grained management of cluster resources and tasks.
What is Apache Mesos?
What is Apache Mesos?

- Mesos provides its own containerization technology (called the Mesos containerizer)
- It supports the standard docker image format, but relies on its own internal implementation for building containers
- A separate docker containerizer is also available, but not relevant to this presentation

**THE TALE OF TWO CONTAINERIZERS**

- MesosContainerizer (default)
- DockerContainerizer
- Dynamically chosen based on ContainerInfo if both are specified via --containerizers.
What is DC/OS?

- DC/OS (Data Center Operating System) takes the Mesos "kernel" and builds upon it with additional services and functionality
What is DC/OS?

- DC/OS (Data Center Operating System) takes the Mesos “kernel” and builds upon it with additional services and functionality
  - Built-in support for service discovery, load balancing, security, and ease of installation
What is DC/OS?

- DC/OS (Data Center Operating System) takes the Mesos “kernel” and builds upon it with additional services and functionality
  - Built-in support for service discovery, load balancing, security, and ease of installation
  - Extra tooling (e.g. a comprehensive CLI and a GUI)
What is DC/OS?

DC/OS (Data Center Operating System) takes the Mesos “kernel” and builds upon it with additional services and functionality:

- Built-in support for service discovery, load balancing, security, and ease of installation
- Extra tooling (e.g. a comprehensive CLI and a GUI)
- Built-in frameworks for launching long running services (Marathon) and batch jobs (Metronome)
What is DC/OS?

- DC/OS (Data Center Operating System) takes the Mesos “kernel” and builds upon it with additional services and functionality
  - Built-in support for service discovery, load balancing, security, and ease of installation
  - Extra tooling (e.g. a comprehensive CLI and a GUI)
  - Built-in frameworks for launching long running services (Marathon) and batch jobs (Metronome)
  - A repository (app-store) for installing other common packages and frameworks (e.g. Spark, Kafka, Cassandra)
What is DC/OS?
What is DC/OS?
Overview of Talk

- Brief intro to docker and nvidia-docker
- Challenges of supporting Nvidia GPUs in docker containers
- How **nvidia-docker** addresses these challenges
- How **Apache Mesos** addresses these challenges
- **DC/OS** GPU Demos
- Future Work
Docker

- Extremely popular image format for containers
  - Build once → run everywhere
  - Configure once → run anything

Source: DockerCon 2016 Keynote by Docker’s CEO Ben Golub
Nvidia-docker

Wrapper around docker to allow GPUs to be used inside docker containers
Nvidia-docker

Machine Learning Frameworks

Support exists for many popular machine learning frameworks with nvidia-docker (including TensorFlow, Caffe, CNTK, etc.)

Source: https://data-shaker.com/docker-tensorflow-with-jupyter-notebook-on-windows/
Overall Goal

Test locally with nvidia-docker

Deploy to production with DC/OS
Challenges of Supporting Nvidia GPUs in Docker containers

• Before containers it was easy
Before containers it was easy
  ○ **Buy some GPUs**
Challenges of Supporting Nvidia GPUs in Docker containers

- Before containers it was easy
  - Buy some GPUs
  - **Install them on your box**
Challenges of Supporting Nvidia GPUs in Docker containers

- Before containers it was easy
  - Buy some GPUs
  - Install them on your box
  - Install the base nvidia drivers
Before containers it was easy

- Buy some GPUs
- Install them on your box
- Install the base nvidia drivers
- Install some advanced toolkit libraries
Challenges of Supporting Nvidia GPUs in Docker containers

● Before containers it was easy
  ○ Buy some GPUs
  ○ Install them on your box
  ○ Install the base nvidia drivers
  ○ Install some advanced toolkit libraries
  ○ **Link a GPU accelerated application against these libraries**
Challenges of Supporting Nvidia GPUs in Docker containers

- Before containers it was easy
  - Buy some GPUs
  - Install them on your box
  - Install the base nvidia drivers
  - Install some advanced toolkit libraries
  - Link a GPU accelerated application against these libraries
  - **Run your application**
Challenges of Supporting Nvidia GPUs in Docker containers

- So what about containers?
  - Buy some GPUs
  - Install them on your box
  - Install the nvidia-kernel-modules

nvidia-kernel-module
Linux Kernel
So what about containers?
○ Buy some GPUs
○ Install them on your box
○ Install the nvidia-kernel-module
○ **Build a docker image**
  ■ Bundle the base nvidia libraries
  ■ Bundle some advanced toolkit libraries
  ■ Bundle a GPU accelerated application to use these libraries
Challenges of Supporting Nvidia GPUs in Docker containers

- So what about containers?
  - Buy some GPUs
  - Install them on your box
  - Install the nvidia-kernel-module
  - Build a docker image
    - Bundle the base nvidia libraries
    - Bundle some advanced toolkit libraries
    - Bundle a GPU accelerated application to use these libraries
  - Run your docker container
Challenges of Supporting Nvidia GPUs in Docker containers

- Straightforward, right?
Challenges of Supporting Nvidia GPUs in Docker containers

- Will only work if the kernel / user driver versions match
Challenges of Supporting Nvidia GPUs in Docker containers

- Won’t work if they don’t
Challenges of Supporting Nvidia GPUs in Docker containers

- Either way, you have to map in the GPU devices somehow
nvidia-docker and GPUs

- Components of **nvidia-docker**
  - Set of docker images that set custom labels / environment variables
  - **nvidia-docker-plugin** (standard docker volume plugin)
  - **nvidia-docker** (wrapper script around **docker** itself)
nvidia-docker and GPUs

- nvidia-docker-plugin

  Finds all standard nvidia libraries / binaries on the host and consolidates them into a single place as a docker volume

  /var/lib/docker/volumes
  └── nvidia_XXX.XX (version number)
      ├── bin
      │   └── lib
      │       └── lib64
nvidia-docker and GPUs

- **nvidia-docker** wrapper script

  Looks for the label:
  
  ```
  com.nvidia.volumes.needed = nvidia_driver
  ```

  When found, it maps the `nvidia_XXX.XX` volume into the container at:
  
  `/usr/local/nvidia`

  Enumerates all GPUs on the machine and maps them into the container as available devices

  Passes all other docker options straight through to **docker**
nvidia-docker and GPUs

**Diagram:**
- **Container**
  - Application
  - CUDA / TensorFlow libraries
  - nvidia base libraries (v1)
  - nvidia-kernel-module (v1)
  - Linux Kernel

- **Container**
  - Application
  - CUDA / TensorFlow libraries
  - nvidia base libraries (v2)
  - nvidia-kernel-module (v2)
  - Linux Kernel
**nvidia-docker and GPUs**

- **nvidia-kernel-module (v1)**
  - Linux Kernel
  - nvidia-base-libraries (v1)

- **CUDA / TensorFlow libraries**
  - Container

- **nvidia base libraries (v2)**
  - Container

- **nvidia-kernel-module (v2)**
  - Linux Kernel

© 2017 Mesosphere, Inc. All Rights Reserved.
Apache Mesos and GPUs

- Mimics functionality of **nvidia-docker**
  - Supports nvidia docker images with custom labels
  - Maps consolidated volume of binaries / libraries into `/usr/local/nvidia`
  - Enumerates GPUs and injects them into containers
  - **Isolates** access to GPUs between tasks
Apache Mesos and GPUs

Mesos Agent

Containerizer API

(Unified)
Mesos Containerizer

Isolator API

CPU
Memory
Apache Mesos and GPUs

Mesos Agent

Containerizer API

(Unified)
Mesos
Containerizer

Isolator API

CPU
Memory
GPU
Apache Mesos and GPUs

Mesos Agent

Containerizer API

(Unified) Mesos Containerizer

Isolator API

CPU
Memory
GPU

Nvidia GPU Isolator

Nvidia GPU Allocator
Nvidia Volume Manager

Linux devices cgroup
Apache Mesos and GPUs

Mesos Agent

- Containerizer API
- Isolator API

Nvidia GPU Isolator

- Allocates GPUs to tasks
- Isolates Access to GPUs between tasks

Mimics functionality of nvidia-docker-plugin

- Nvidia GPU Allocator
- Nvidia Volume Manager
- Linux devices cgroup
DC/OS GPU Demos

- Simple isolation demo
- Single node Tensorflow demo
- Distributed Tensorflow demo (Future work)
Simple Isolation Demo

- 1 master, 1 agent - 8 GPUs

- `ssh` into agent and run `nvidia-smi` locally to show all 8 GPUs present

- Launch 2 container instances and allocate 4 GPUs to each

- Run `nvidia-smi` in each container to show allocation of 4 GPUs to each
Simple Isolation Demo

https://youtu.be/z9gzzbjE-JE
Single Node Tensorflow Demo

- 1 master, 1 agent - 8 GPUs
- Launch standard Tensorflow docker image
- Show standard Jupyter Notebook running
- Exec into the running container and download Tensorflow Examples [https://github.com/aymericdamien/TensorFlow-Examples](https://github.com/aymericdamien/TensorFlow-Examples)
- Run the Multi-GPU example
Single Node Tensorflow Demo

https://youtu.be/wumsAoUy0cQ
Future Work

- Integrate Different Machine Learning Frameworks with the DC/OS SDK
  - [https://github.com/mesosphere/dcos-commons](https://github.com/mesosphere/dcos-commons)

- Distributed TensorFlow with TFMesos
  - [https://github.com/douban/tfmesos](https://github.com/douban/tfmesos)

- Distributed Mxnet
  - [https://github.com/dmlc/mxnet](https://github.com/dmlc/mxnet)

- One click install in the DC/OS Universe (The DC/OS notion of an app-store)
Future Work

- Topology aware scheduling
- GPU Sharing (virtual GPUs)
- GPU consumption metrics
Future Work

- Topology aware scheduling
- GPU Sharing (virtual GPUs)
- GPU consumption metrics

Contributions Welcome!
Special Thanks to All Collaborators

Vikram Ditya
Andrew Iles
Jonathan Calmels
Felix Abecassis
Rob Todd
Rajat Phull
Shivi Fotedar
Yubo Li

Seetharami Seelam
Yong Feng
Guangya Liu
Ian Downes
Niklas Nielson
Connor Doyle
Benjamin Mahler
Tim Chen
Questions and Links

- Apache Mesos
  - http://mesos.apache.org/

- Open DC/OS
  - https://dcos.io/

- Enterprise DC/OS
  - https://mesosphere.com/product/

- GPU Related Documentation for Mesos and DC/OS
  - https://github.com/apache/mesos/blob/master/docs/gpu-support.md
  - https://dcos.io/docs/1.9/deploying-services/gpu/config/