Orchestrating containers and GPU accelerated resources

Brief activity overview

EGI Conference
Amsterdam
6-8 May 2019

Marica Antonacci
Alessandro Costantini
INFN

DEEP-Hybrid-DataCloud is funded by the Horizon 2020 Framework Programme of the European union under grant agreement number 777435

Summary

- **Container orchestration in DEEP-HDC**
- **Mesos and Kubernetes GPU support**
- **DEEP PaaS and TOSCA**
  - Deploy a GPU cluster and application
- **Conclusions**

Container orchestration

- **DEEP** exploits some of the most used **Container Orchestration Platforms** to deploy and run user applications (both ***long-running services*** and ***batch-like jobs***)
- The usage of production-grade platforms, like **Mesos** and **Kubernetes**, brings crucial advantages (e.g. fault-tolerance, load-balancing, auto-healing, etc.) but can introduce some complexities for the users
- DEEP provides documentation, recipes and high-level tools to **hide this complexity and take advantage of these platforms** in an easy and straightforward way
- GPUs are easily used and exploited for running **ML/DL applications**

GPU support in Mesos

- Apache **Mesos** is an open-source **cluster manager** that can be deployed on heterogeneous and hybrid environments: bare-metal, virtual machines, containers
- **Different workloads** can be run on top of a Mesos cluster, ensuring **isolation** and resource fair **sharing**
- Mesos provides **native GPU support** through a dedicated module that mimics the *nvidia-docker* behavior: the proper drivers and tools are automatically mounted into the container

GPU support in Kubernetes

Hybrid DataCloud

- **Kubernetes** is an open-source platform for automating deployment, scaling and managing containerized applications
- **Different workloads** can be run on top of a Kubernetes cluster, ensuring **isolation** and resource fair **sharing**
- Kubernetes provides **native GPU support**
  - through *nvidia-docker*
  - GPU-specific container images for Kubernetes built by Nvidia

Deploying a GPU-enabled cluster through DEEP PaaS

- DEEP provides a set of **ansible roles** and **TOSCA templates** that allow to deploy **a complete Mesos/K8s cluster with GPU support** automatically
  - The **TOSCA templates** are used to deploy the cluster in **cloud** (OpenStack, OpenNebula, Amazon, etc.) through the **DEEP PaaS Orchestration layer**
    - Orchestration layer collects high-level deployment requests and translate them into action to coordinate resources interacting with the underlying cloud infrastructures.
    - Ansible used for cluster deployment in the TOSCA templates
      - **Ansible roles** can be also used to deploy the cluster "manually" on **bare-metal**
    - Tested successfully also towards EGI Fed Cloud
  - TOSCA Templates and DEEP PaaS Layer enable to exploit auto-scaling and auto-provisioning of the computing resources
    - depending on the load requested by the Mesos Users.

Deploying GPU applications/jobs with DEEP PaaS

- The DEEP PaaS allows users to deploy **dockerized long-running services** or **batch-like jobs** on distributed Mesos clusters
- The user requirements (e.g. requested num of GPUs, CPUs, RAM Memory) are expressed in the **TOSCA template** and analyzed by the **PaaS Orchestrator that selects the infrastructure to submit the deployment among those available**:
  - Service endpoints and capabilities (e.g. gpu support enabled or not) are published in the Configuration Management Database (CMBD)
  - Service health and metrics are collected and published by the Monitoring system
  - All these information are consumed by the Orchestrator
  - The authentication/authorization is managed through OIDC tokens issued by INDIGO IAM

Conclusions

- **DEEP** exploits some of the most used **Container Orchestration Platforms** to deploy and run user applications (both ***long-running services*** and ***batch-like jobs***) on distributed Mesos and K8S clusters
  - Enabling GPU support for **ML/DL** applications
- User requirements (e.g. requested num of GPUs, CPUs, RAM Memory) are expressed in the **TOSCA template** and analyzed by the **PaaS Orchestrator that selects the cloud infrastructure to submit the deployment among those available**
- DEEP high-level tools **hide the inherit complexity and take advantage of these platforms** in an easy and straightforward way
- Newly developed Python-based Dashboard Orchestrator GUI

  - https://github.com/indigo-dc/orchestrator-dashboard

References

DEEP-HDC:

- https://deep-hybrid-datacloud.eu/

Documentation:

- https://docs.deep-hybrid-datacloud.eu/en/latest/

TOSCA templates:

- https://github.com/indigo-dc/tosca-templates
- https://github.com/indigo-dc/tosca-types/tree/master/examples

Ansible roles:

- https://galaxy.ansible.com/indigo-dc

Thank you

Any Questions?

https://deep-hybrid-datacloud.eu

DEEP-Hybrid-DataCloud is funded by the Horizon 2020 Framewo  of the European union under grant agreement number 777435