Raising Cloud Computing in a HPC production environment

• Use the Cloud to virtualize GRID/HPC architectures to run established workflows and/or migrate workflows from HPC to Cloud?
• Workflows already under construction, turning towards container orchestration engines. Atomic design: Do one thing and do it good.
• Growing demand for scalable visualisation, graphical interaction and dynamic resource management
• Data simply too big: Bring the computing to the data, not vice versa
The European Open Science Cloud for Research pilot project is funded by the European Commission, DG Research & Innovation under contract no. 739563.
Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment.
Magnum clusters on OpenStack

- Per project networking based on native OpenStack Objects: Neutron networks, software router, LbaaS, DNS
- Bare Metal Clusters and Infiniband supported. Overhead HPC O(1%) [1]

- Small clusters ready in less than 5 minutes
- Magnum does NOT run containers. Container management through native COE client: kubectl, docker swarm
- Performance and efficiency of native code execution with the abstraction, security and immutability of virtualization

[1] https://www.openstack.org/assets/science/OpenStack-CloudandHPC6x9Booklet-v4-online.pdf
OpenWhisk: Cloud functions as a Service

1. User Api Gateway
2. LoadBalancing, URL dispatcher
3. Authentication
4. Authorization
5. Discovery
6. Message buffer
7. Invoke Client Code
8. Obtain JSON result

FROM base-action
RUN install more tools
RUN mount filesystem
ADD more stuff
USER bind-to-UID
ENV runtime=argvs

% wsk action create my-workflow --docker my-action
% wsk action create my-workflow my-start-script.zip --docker my-action
% wsk action create my-workflow my-start-script.zip --native
FROM centos:latest
RUN yum -y install python-pip && pip install flask
ADD actionProxy/actionproxy.py #runtime
ADD exec action/exec #executable, start-script

- Native CentOS Linux for Client Code deployment
- action/exec anything callable from start script to large binary
- actionProxy/actionproxy.py for serverless service integration
- JSON arguments in
- JSON arguments out
Packaging functions for event driven computation

- Execute code in response to events
- Elastic auto scaling of swarms and clusters
- Suited for data, API, Experiment Computing
- Efficient resource sharing
- Per usage billing
Science DevOps challenges

- **Software**
  - Deploy codes as microservices, stateless units
  - Licences to include new use cases and distribution channels
  - Non-public Container registries not freely-redistributable codes
- **Skills and training:**
  - Science DevOps
  - Science User
- **On-premise/hybrid cloud**
  - Infrastructure DevOps
  - Platform DevOps

Scaling takes effort

@csantanapr

OpenWhisk

1. Increase of infrastructure cost footprint
2. Increase of operational management cost and complexity
Science user perspective

- EOSCpilot Photon & Neutron Science Demonstrator: Process and index CrystFEL diffraction images
- High portability and reproducibility for large-scale distributed scientific workflows
- User-facing RESTful HTTP API
- Asynchronous and synchronous modes, periodic
- Efficient software distribution over CVMFS (file-based transfer using CernVM-FS graphdriver plugin for Docker)
- Function repository through docker registries
- Co-development, distributed teams
- Function-as-a-Service
Science user perspective

- Users interact with Jupyter Notebook spawned in individual container
  - Develop functions iteratively, tests locally
  - Export and share executable functions using FaaS
- Users can download docker container with pre-configured software and run locally, co-develop, inherit, derive
  - Distribute again on the docker registry
  - Implement on a function server
- Easily cherry pick micro services from other project namespaces
- Deploy same functions for event driven computational pipelines
Images borrowed from:
• https://github.com/apache/incubator-openwhisk
• https://github.com/apache/incubator-openwhisk-devtools
• https://github.com/openfaas/faas
• https://github.com/openstack
• https://github.com/cncf/landscape

Thank you!