

Kubernetes vs Mesos

evaluation and testing results





My personal opinion

- Both technologies make it possible to use containers to deploy, manage, and scale applications (orchestration)
 - Kubernetes: natively
 - Mesos: through Marathon (and even Kubernetes) built on top of Mesos

	Kubernetes	Mesos + Marathon
Learning curve	middle/high	middle
documentation	full	poor
debugging (system & container)	easy	difficult
System stability	full	Marathon seems not so stable

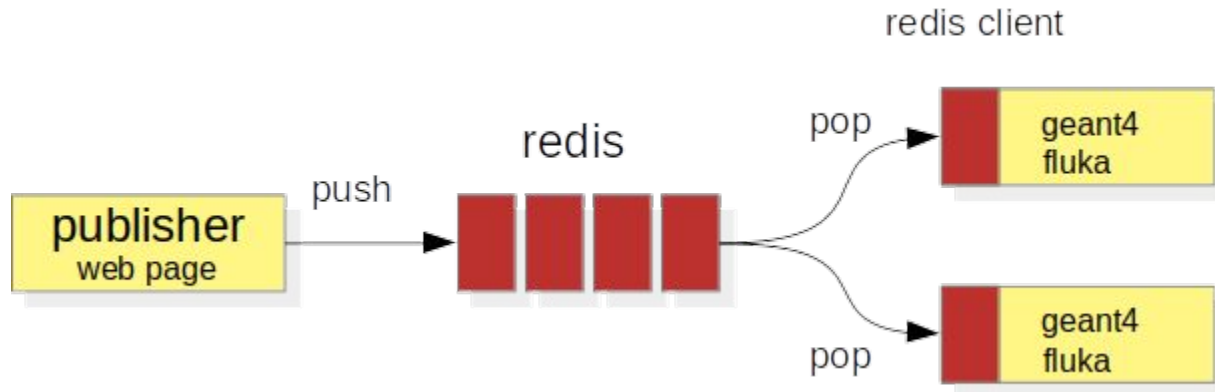


Kubernetes (k8s) testbed

- Kubernetes testbed deployed in ISOLPHARM_Ag
 - 1 master and 3 slaves
- Executed the same tests done for Mesos (based on geant4)
- Deployed the Nuclides Chart as long running service with replica 3
- Storage solution still based on NFS server
 - Kubernetes supports several storage backend (S3, ceph, etc)
 - NB: volumes can be shared among different containers/hosts
- Implemented a task queue (Redis) serving parallel tasks (geant4 / fluka) defined by a single job
 - Redis is an open source, in-memory data structure store, used as a database, cache and message broker (e.g. RabbitMQ)
 - Added custom Redis client (python) to our Geant4 and Fluka containers

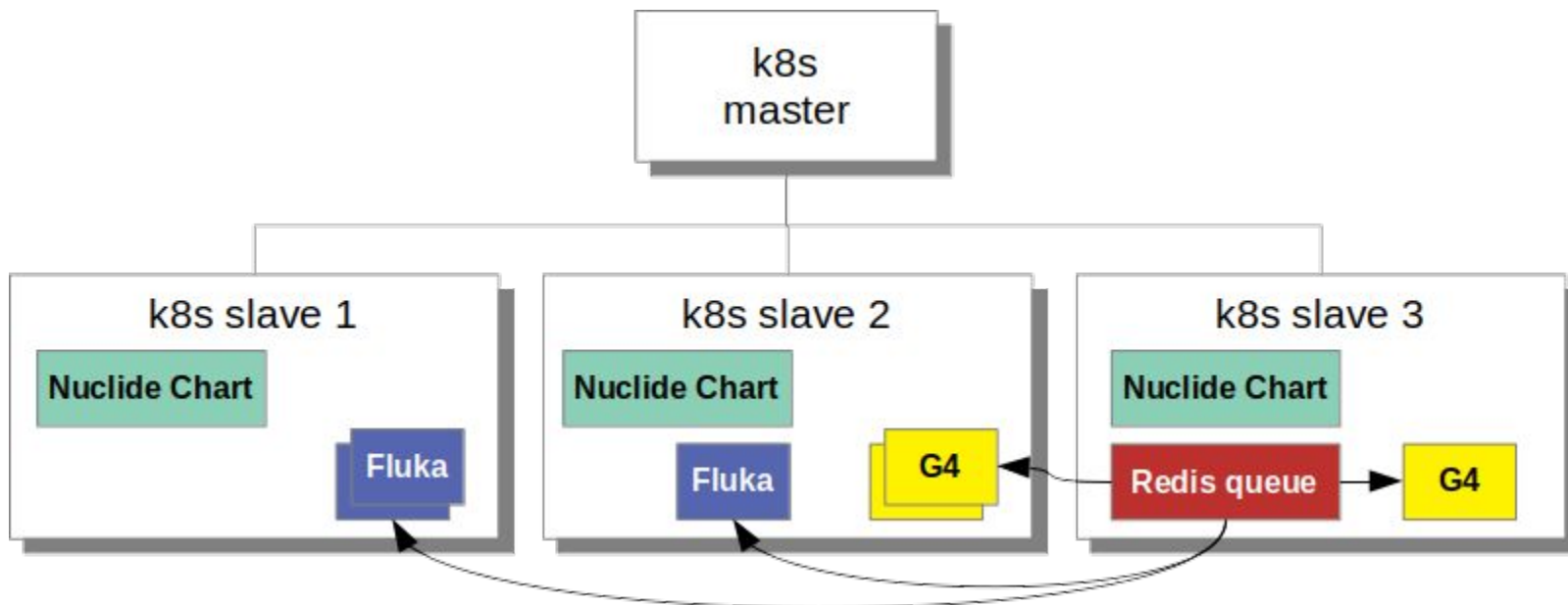


Redis as message queue





The k8s testbed





Cluster horizontal autoscaling

- K8S provides a horizontal autoscaling mechanism for Google (GCE) and Amazon (AWS) cloud providers
 - Openstack (OS) not supported
- Clues, Magnum, Supergiant etc, manage the Kubernetes autoscaling in OS
- We have to investigate the better and proper solution for our needs
- I will start with Clues



Questions?

