Evaluation of the components of a distributed workload managers on new platforms

Rami Aboushadi – Jackson State University – MS – OMNI Internship

Introductions: GlideinWMS

The purpose of the GlideinWMS is to provide a simple way to access the Grid resources. GlideinWMS is a Glidein Based WMS (Workload Management System) that works on top of HTCondor. Glideins are like placeholders, a mechanism by which one or more remote resources temporarily join a local HTCondor pool. The HTCondor system is used for scheduling and job control.

To evaluate the GlideinWMS and HEPCloud components on REHL8 I followed a series of procedures:

- Setup of AlmaLinux 8 VMs with Python 3.9
- Running the continuous integration tests on RHEL8 with Python 3.9. And troubleshooting and fixing some failing unit tests
- Installation, testing and adaptation of GlideinWMS Frontend and Factory on RHEL8 with Python 3.9
- Installation, testing and adaptation of Decision Engine on RHEL8 with Python 3.9
- Benchmarking of new Hyper-threaded worker nodes with different loads from a real DUNE workflow

Methods:

To evaluate the GlideinWMS and HEPCloud components on REHL8 I followed a series of procedures:

- Setup of AlmaLinux 8 VMs with Python 3.9
- Running the continuous integration tests on RHEL8 with Python 3.9. And troubleshooting and fixing some failing unit tests
- Installation, testing and adaptation of GlideinWMS Frontend and Factory on RHEL8 with Python 3.9
- Installation, testing and adaptation of Decision Engine on RHEL8 with Python 3.9
- Benchmarking of new Hyper-threaded worker nodes with different loads from a real DUNE workflow

Conclusions:

I learned ...
I've got more experienced ...
I was able to troubleshoot the unit tests and fix some problems with deprecated Python constructs and with a different handling of multi processing. We decided to skip the test of epoll and remove it from the possible alternatives to handle multi processing.
I was able to install all the systems on Alma Linux 8 (RHEL8 compatible) with Python 3.9: GlideinWMS Frontend, VO Pool, and Factory, and HEPCloud's Decision Engine The DUNE workflow was able to scale on the Worker node using all cores