Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

Constructing a Testing Application for the Glidein Workflow Management System

LeRayah Neely-Brown
SIST 2020 Internship
Supervisor: Bruno Coimbra
05 August 2020

GlideinWMS, HTCondor, & Their Importance

High Throughput Computing

HTC is the use of many computing resources over long periods of time to execute a computational task. Many High Energy Physics experiments require this to carry out simulations, data reconstruction, and data analysis.

Glidein Workflow Management System

GlideinWMS is a tool that distributes High Throughput Computing. GlideinWMS consists of 3 components: Glideins, Factory, and Frontend.

HTCondor

HTCondor is a Workload Management System; a batch system for jobs. HTCondor system is used for scheduling and job control.

GlideinWMS Workflow

Glideins: Scouts for resources and validates the Worker node, customizes node, and prepares it to be executed via HTCondor

Factory: Knows how to submit nodes to the site

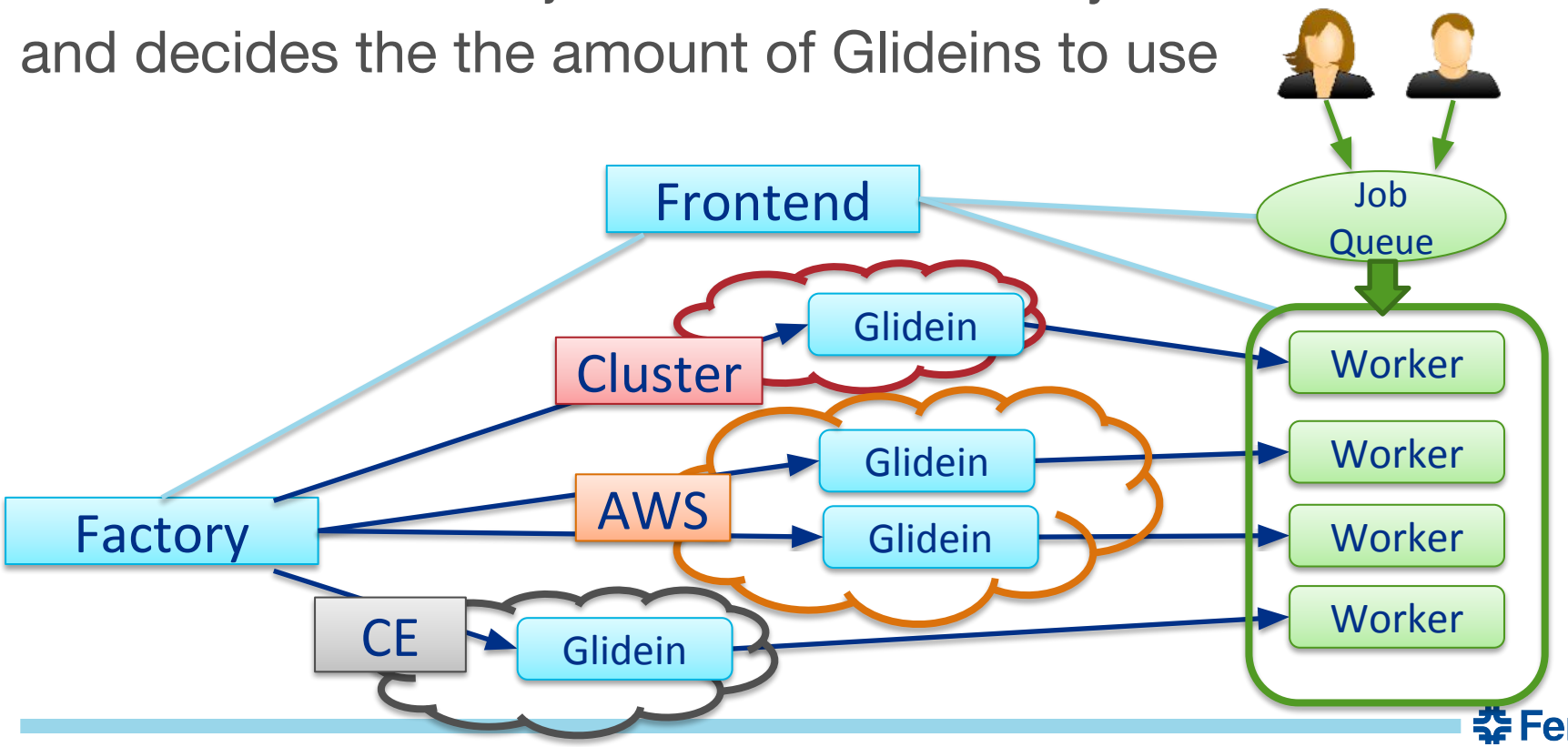
Frontend: Monitors jobs to see how many Glideins are needed and decides the the amount of Glideins to use

GlideinWMS Workflow

Glideins: Scouts for resources and validates the Worker node, customizes node, and prepares it to be executed via HTCondor

Factory: Knows how to submit nodes to the site

Frontend: Monitors jobs to see how many Glideins are needed and decides the the amount of Glideins to use



Issues with GlideinWMS

Current Steps To Test a Job

- ➔ Have the frontend
- ➔ Have to submit a job
- ➔ Submit the job to the factory
- ➔ Submit a Glidein
- ➔ The Glidein may end up to many different sites
- ➔ Need to get the Glidein to the **SPECIFIC** site wanted
- ➔ **THEN** start the actual job at the correct specific job site

Problems

- ➔ Glideins run in different places
- ➔ Software keeps changing, so want to make easier to test new sites and new software

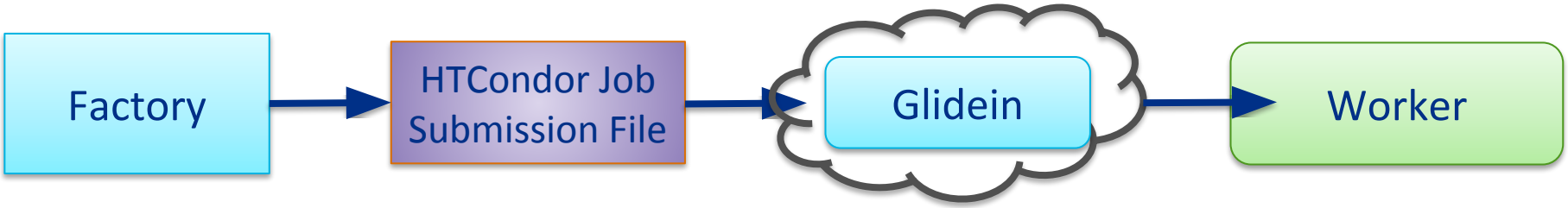
Objective for Building Testing Application

The GWMS Testing Application serves as a streamline tool that will directly submit a Glidein to the desired worker node. The testing application's goal is to efficiently test the worker nodes of a site and test new GWMS configuration.

Objective for Building Testing Application

The GWMS Testing Application serves as a streamline tool that will directly submit a Glidein to the desired worker node. The testing application's goal is to efficiently test the worker nodes of a site and test new GWMS configuration.

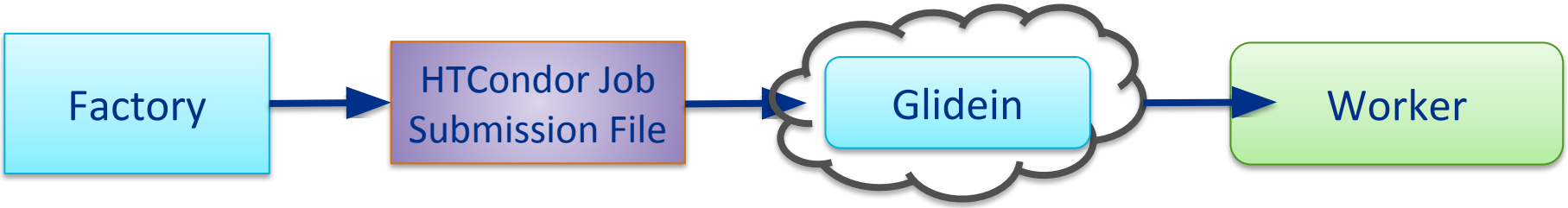
Development of Testing Application



Objective for Building Testing Application

The GWMS Testing Application serves as a streamline tool that will directly submit a Glidein to the desired worker node. The testing application's goal is to efficiently test the worker nodes of a site and test new GWMS configuration.

Development of Testing Application



Technologies Used

Python, Shell, FermiCloud (OpenNebula), Git

Example of Job Submission File

```
# File: job.condor
#
Universe = grid
Grid_Resource = condor fermicloud089.fnal.gov fermicloud089.fnal.gov:9619
Executable = glidein_startup.sh
copy_to_spool = True
Arguments = $ENV(GLIDEIN_ARGUMENTS)
Transfer_Executable = True
transfer_Output_files =
WhenToTransferOutput  = ON_EXIT
stream_output = False
stream_error  = False
+TransferOutput = ""
x509userproxy = $ENV(X509_USER_PROXY)
+GlideinFactory = "$ENV(FACTORY_NAME)"
+GlideinName = "$ENV(GLIDEIN_NAME)"
+GlideinEntryName = "$ENV(GLIDEIN_ENTRY_NAME)"
+GlideinEntrySubmitFile = "$ENV(GLIDEIN_ENTRY_SUBMIT_FILE)"
+GlideinClient = "$ENV(GLIDEIN_CLIENT)"
+GlideinFrontendName = "$ENV(GLIDEIN_FRONTEND_NAME)"
+GlideinCredentialIdentifier = "$ENV(GLIDEIN_CREDENTIAL_ID)"
+GlideinSecurityClass = "$ENV(GLIDEIN_SEC_CLASS)"
+GlideinWebBase = "$ENV(GLIDEIN_WEB_URL)"
+GlideinLogNr = "$ENV(GLIDEIN_LOGNR)"
+GlideinWorkDir = "$ENV(GLIDEIN_STARTUP_DIR)"
+GlideinSlotsLayout = "$ENV(GLIDEIN_SLOTS_LAYOUT)"
+GlideinMaxWalltime = $ENV(GLIDEIN_MAX_WALLTIME)
+fename = "$ENV(GLIDEIN_USER)"
periodic_remove = (isUndefined(GlideinSkipIdleRemoval)==True || GlideinSkipIdleRemoval==False)
ME)>0 && (time() - QDate)>$ENV(GLIDEIN_IDLE_LIFETIME))
Notification = Never
```

Example of Job Submission File

```
# File: job.condor
#
Universe = grid
Grid_Resource = condor fermicloud089.fnal.gov fermicloud089.fnal.gov:9619
Executable = glidein_startup.sh
copy_to_spool = True
Arguments = $ENV(GLIDEIN_ARGUMENTS)
Transfer_Executable = True
transfer_Output_files =
WhenToTransferOutput  = ON_EXIT
```

Challenges With Constructing Testing Application

- ➔ Modifying the startup script for the Glidein
- ➔ While testing the testing application, some Glideins failed to reach the correct worker node or remain idle

Future Work + Conclusions

- ➔ Create a web framework will visualize the result of the testing application and allow remote manual submission of the Glidiein
- ➔ Implement a feature to override some parameters received from the Factory and Frontend of GWMS for the Glidein

Future Work + Conclusions

- ➔ Create a web framework will visualize the result of the testing application and allow remote manual submission of the Glidiein
- ➔ Implement a feature to override some parameters received from the Factory and Frontend of GWMS for the Glidein

```
parser.add_argument(
    '--addfactory', type=str, action='store', dest='custom_name_factory',
    help='Custom option for factory')

parser.add_argument(
    '--addfrontend', type=str, action='store', dest='custom_name_frontend',
    help='Custom option for frontend')
```

Acknowledgements

I would like to thank my mentor Bruno Coimbra, the leader of the GWMS project, Marco Mambelli, and all of the developers who work on the GWMS project. I would also like to thank Fermilab and the SIST Committee for giving me the opportunity to partake in this project. Lastly, a special thanks to my SIST Mentors.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.