



# Implementing Tracing to Monitor Job Execution in GlideinWMS

Mentee: Ralph Ortiz (Arizona State)

Mentor: Marco Mambelli

---



# Abstract

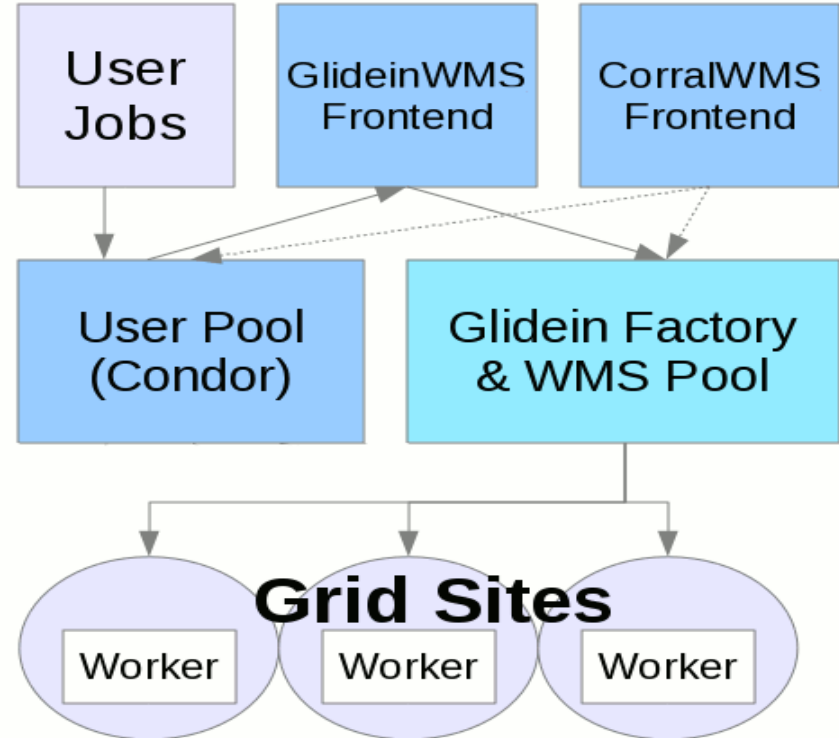
GlideinWMS is a workload management system allowing experiments like CMS, DUNE and others to access and efficiently use vast amounts of computing resources. CMS uses at any given time hundreds of thousands of CPUs for its simulations and analyses, with peaks over a million. And GlideinWMS provisions the required resources, using local or remote batch farms, private and commercial clouds, and supercomputing centers. The Glidein, or pilot job, is a very important component of GlideinWMS: it tests and sets up the computing nodes and makes them available for the scientific computations.

This project aims to extend the GlideinWMS monitoring to include tracing and to evaluate optimizations. Compared to standard monitoring or logging, tracing encompasses a much wider, continuous view of an application. The goal of tracing is to follow the Glidein flow and data progression, producing a lot of information. This activity includes extending the Glidein monitoring framework, instrumenting the Glideins with tracing software, and analyzing the traces with statistical and ML techniques to understand in more detail the Glidein behavior and to suggest optimizations.



# Overview

1. Introduction
2. GlideinWMS
3. Jaeger / Open Telemetry
4. Glidein and Jobs instrumentation
5. Traces collection
6. Data analysis
7. Python / Shell
8. Conclusions





# To Dos

Understand the Virtual Machine via Shell

Propose a tracing infrastructure

- where will the Jaeger agent run?

- What services will be involved (such as Open Telemetry)?

and implementation

- How will trace IDs be propagated?

- Trace sampling strategy?

Implement tracing for Glideins

Collect traces, analyze samples, propose workflow optimizations



# Method

- Run test jobs through FermiCloud Virtual Machines
- Track traces of jobs in Jaeger
- Plan code and architecture modifications
- Develop in Python and Shell using GitHub to collaborate
- Deploy
- Collect trace information
- Analyze and suggest optimizations