Distributed Tracing in GlideinWMS

Rafael Ortiz (Arizona State University)

Dr. Marco Mambelli (Fermilab)

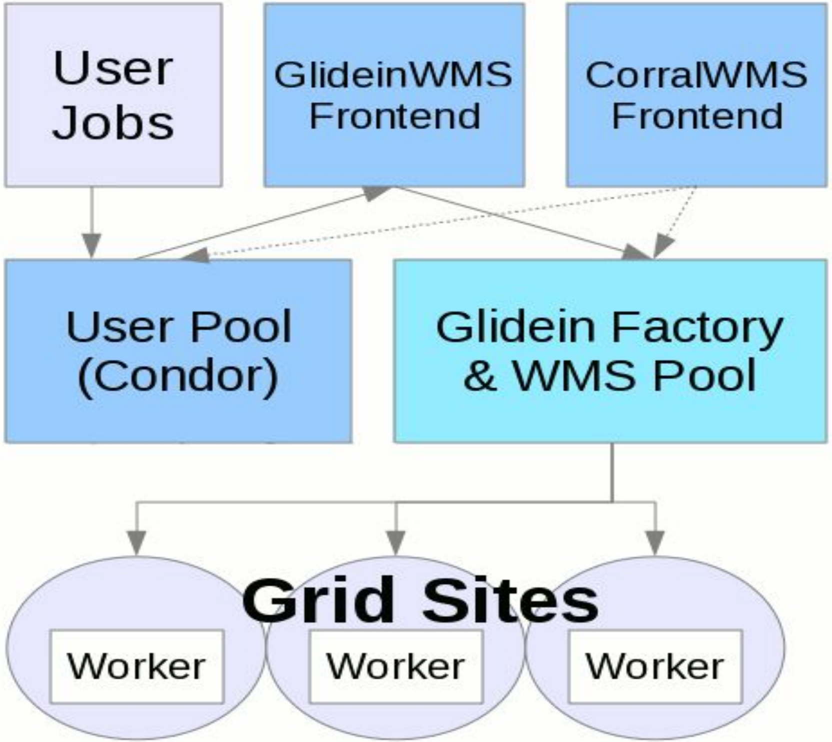
Project Proposal

GlideinWMS (http://glideinwms.fnal.gov/) 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.

GlideinWMS

- GlideinWMS provisions jobs
- Allocates resources
  - CMS & Universities
  - All sorts of computers
  - Simulations and Analyses



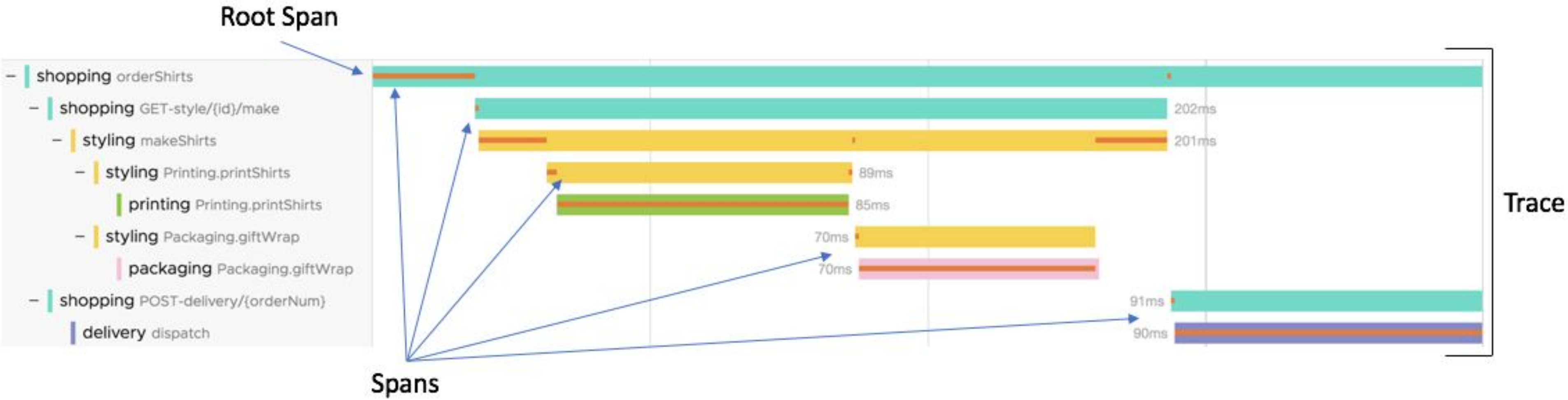
Tracing

- **Tracing** collects temporal instances and time points throughout the execution of a program and the jobs that are running on the system

- Glide-ins run for a long time and on different systems
  - Tracing is important so that we can see the progress of the Glide-ins

- Tools and Software:
  - OpenTelemetry
  - Jaeger
  - pfeil

Parent and Child Spans



Root Span

Trace

Spans

- shopping orderShirts
  - shopping GET-style/{id}/make 202ms
    - styling makeShirts 201ms
      - styling Printing.printShirts 89ms
        - printing Printing.printShirts 85ms
      - styling Packaging.giftWrap 70ms
        - packaging Packaging.giftWrap 70ms
  - shopping POST-delivery/{orderNum} 91ms
    - delivery dispatch 90ms

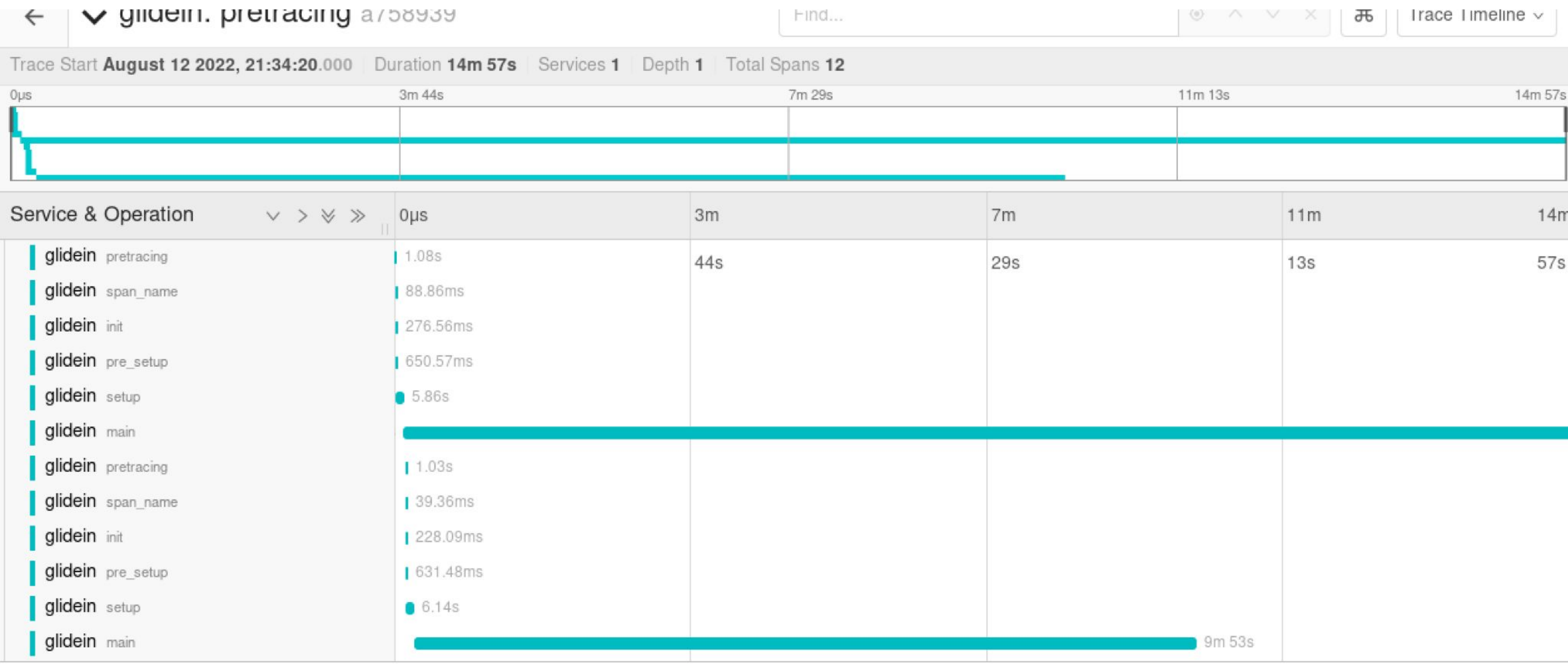
Research

- OpenTelemetry is a library that allows a comprehensive way to instrument tracing
- We send traces to Jaeger running on Docker via TCP
- We initialize a parent trace in the Factory via the OpenTelemetry Python API and continue to send child spans in the Glidein using the pfeil command
- glideinTracer.py
  - Class Methods to initialize a trace and get the Trace ID
  - Implemented into GlideinWMS
- glideFactoryLib.py modified to instrument Glidein submission
- Pfeil traces the main sequences of the Glidein in glidein_startup.sh
  - Initialization, Set Up, Start Up, and Clean Up

Research Cont.

- When the Glidein trace data is in Jaeger, we can tabulate the multiple parameters of the trace data
  - Data Analysis
  - Optimizations
- Trace information will allow the GlideinWMS framework behavior to be more understood and more reliable
- We found that we can fully automate tracing and our prototype allows for manual data analysis
- Remaining parts to revise and adjust for GlideinWMS for future data analysis of Glideins everywhere

Glideins in Jaeger UI

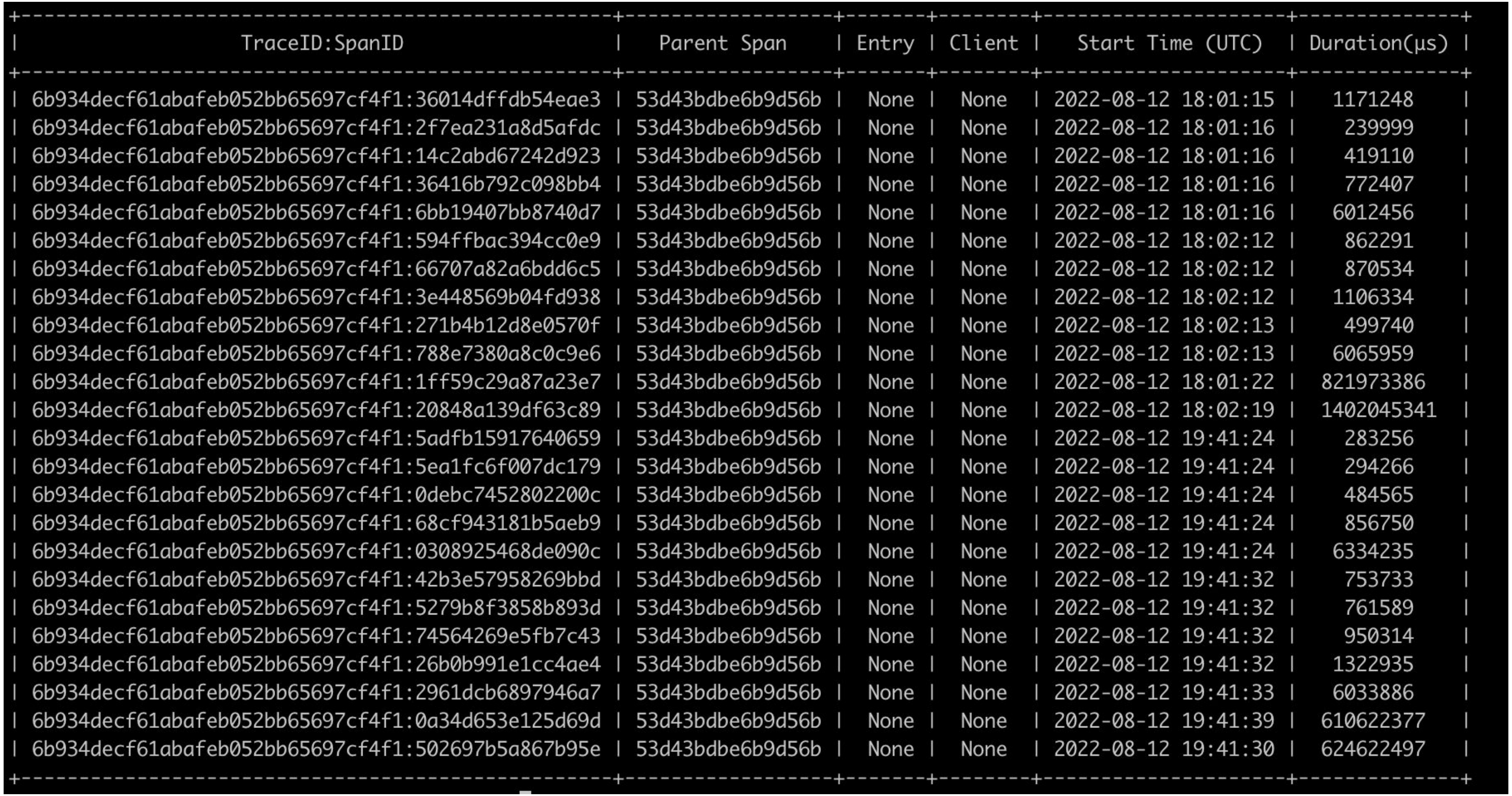


glidein: pretracing a758939

Trace Start **August 12 2022, 21:34:20**.000 | Duration **14m 57s** | Services **1** | Depth **1** | Total Spans **12**

glidein pretracing 1.08s
glidein span_name 88.86ms
glidein init 276.56ms
glidein pre_setup 650.57ms
glidein setup 5.86s
glidein main
glidein pretracing 1.03s
glidein span_name 39.36ms
glidein init 228.09ms
glidein pre_setup 631.48ms
glidein setup 6.14s
glidein main 9m 53s

Tabulation



```
+---------------------------------------------------+------------------+-------+--------+---------------------+--------------+
|                  TraceID:SpanID                   |   Parent Span    | Entry | Client |   Start Time (UTC)  | Duration(μs) |
+---------------------------------------------------+------------------+-------+--------+---------------------+--------------+
| 6b934decf61abafeb052bb65697cf4f1:36014dffdb54eae3 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:15 |   1171248    |
| 6b934decf61abafeb052bb65697cf4f1:2f7ea231a8d5afdc | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:16 |    239999    |
| 6b934decf61abafeb052bb65697cf4f1:14c2abd67242d923 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:16 |    419110    |
| 6b934decf61abafeb052bb65697cf4f1:36416b792c098bb4 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:16 |    772407    |
| 6b934decf61abafeb052bb65697cf4f1:6bb19407bb8740d7 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:16 |   6012456    |
| 6b934decf61abafeb052bb65697cf4f1:594ffbac394cc0e9 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:12 |    862291    |
| 6b934decf61abafeb052bb65697cf4f1:66707a82a6bdd6c5 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:12 |    870534    |
| 6b934decf61abafeb052bb65697cf4f1:3e448569b04fd938 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:12 |   1106334    |
| 6b934decf61abafeb052bb65697cf4f1:271b4b12d8e0570f | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:13 |    499740    |
| 6b934decf61abafeb052bb65697cf4f1:788e7380a8c0c9e6 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:13 |   6065959    |
| 6b934decf61abafeb052bb65697cf4f1:1ff59c29a87a23e7 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:01:22 |  821973386   |
| 6b934decf61abafeb052bb65697cf4f1:20848a139df63c89 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 18:02:19 |  1402045341  |
| 6b934decf61abafeb052bb65697cf4f1:5adfb15917640659 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:24 |    283256    |
| 6b934decf61abafeb052bb65697cf4f1:5ea1fc6f007dc179 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:24 |    294266    |
| 6b934decf61abafeb052bb65697cf4f1:0debc7452802200c | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:24 |    484565    |
| 6b934decf61abafeb052bb65697cf4f1:68cf943181b5aeb9 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:24 |    856750    |
| 6b934decf61abafeb052bb65697cf4f1:0308925468de090c | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:24 |   6334235    |
| 6b934decf61abafeb052bb65697cf4f1:42b3e57958269bbd | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:32 |    753733    |
| 6b934decf61abafeb052bb65697cf4f1:5279b8f3858b893d | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:32 |    761589    |
| 6b934decf61abafeb052bb65697cf4f1:74564269e5fb7c43 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:32 |    950314    |
| 6b934decf61abafeb052bb65697cf4f1:26b0b991e1cc4ae4 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:32 |   1322935    |
| 6b934decf61abafeb052bb65697cf4f1:2961dcb6897946a7 | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:33 |   6033886    |
| 6b934decf61abafeb052bb65697cf4f1:0a34d653e125d69d | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:39 |  610622377   |
| 6b934decf61abafeb052bb65697cf4f1:502697b5a867b95e | 53d43bdbe6b9d56b |  None |  None  | 2022-08-12 19:41:30 |  624622497   |
+---------------------------------------------------+------------------+-------+--------+---------------------+--------------+
```

What I have learned!

- Python!
  - Class Methods and different API
  - Syntax
- Linux
  - Navigation
  - Workflow and Troubleshooting
  - Commands
  - Git
- GitHub
  - Workflow and Management
  - Git within a project
- Software Development
- Computer Science
  - Virtual Machines, Software Architecture
  - Reading Docs, Docker, Deepened Comprehension in General