Running LIGO on Stampede

15 March 2016

Edgar Fajardo
on behalf of OSG Software and Technology

UC San Diego

Acknowledgments

Although I am the one presenting. This work is a product of a collaborative effort from:

- The OSG Factory Ops who debug the GRAM ends.
- GlideinWMS Team
- The Stampede folks

UC San Diego

What this talk is about

- How to run through GlideInWMS at Xsede resources
- Some details about Stampede
- How to run GlideIns at Stampede
- Show as a use Case the LIGO VO Running at Stampede

This talk is NOT about Gravitational waves

UC San Diego

How to run through GlideInWMS at Xsede resources

There are now two ways of doing this:

1. Via general project_id tag on the fronted config
2. Tailored glideIns per job

UC San Diego

General project_id tag on the fronted

It looks like this:

```
<credential absfname="/tmp/vo_proxy" project_id="TG-PHY123456"
security_class="frontend" trust_domain="grid" type="grid_proxy"/>
```

This implies that all pilots from the fronted or group share the same project_id. For example LIGO.

However that is not always the case: aka OSG VO

UC San Diego

Project_id per Job

In the fronted config looks like this:

```
<security classad_proxy="/tmp/vo_proxy" proxy_DN="/DC=com/DC=DigiCert-Grid/O=Open Science Grid/
OU=Services/CN=osg-ligo-1.t2.ucsd.edu" proxy_selection_plugin="ProxyProjectName"
security_name="LIGO" sym_key="aes_256_cbc">
    <credentials>
      <credential absfname="/tmp/vo_proxy" security_class="frontend" trust_domain="grid"
type="grid_proxy"/>
    </credentials>
  </security>
```

And in the job submit file:

```
executable = /bin/sleep
arguments = 1600
error = test-$(Process).error
log = test-$(Process).log
output = test-$(Process).out
+DESIRED_Sites="Stampede"
+is_itb = True
+ProjectName="TG-PHY123456"
```


UC San Diego

From the factory point of view

It looks like any other gram5 entry except for the authentication method:

```
<entry name="Ligo_US_Stampede_gt5" auth_method="grid_proxy+project_id" comment="Added for
LIGO 2015-12-05 note this is an experimental entry! --Jeff" enabled="True"
gatekeeper="login5.stampede.tacc.utexas.edu:/jobmanager-slurm" gridtype="gt5" rsl="(job_type=multiple)
(count=512)(host_count=32)(maxWallTime=2880)" schedd_name="schedd_glideins1@glidein-itb.grid.iu.edu"
trust_domain="grid" verbosity="std" work_dir="/tmp">
```


UC San Diego

About Stampede

Stampede is an XSEDE resource in the Texas Advanced Computing Center at the University of Texas at Austin.

System Component	Specs
Number of Racks	160
Computes nodes per rack	6400
Cores per Node	16 x Xeon E5-2680@ 2.7GHz
Ram per Node	32GB
Total number of Cores	100000

UC San Diego

How to GlideIn at Stampede

1. Associate a computing account with the DN of the pilot proxy.
2. Have an allocation project_name at the fronted in any of the two ways mentioned above.
3. And voila submit with:

```
+DESIRED_XSEDE_Sites="Stampede"
```

Not that fast. There is a catch.

UC San Diego

How to GlideIn at Stampede

Stampede only allows up to 40 jobs (pilots) per user

Yet a job can spawn multiple hosts

Solution: MultiHost GlideIn. Thanks to Brian B and Jeff D who came up with the hack. I mean the solution

How to GlideIn at Stampede

At the factory configuration:

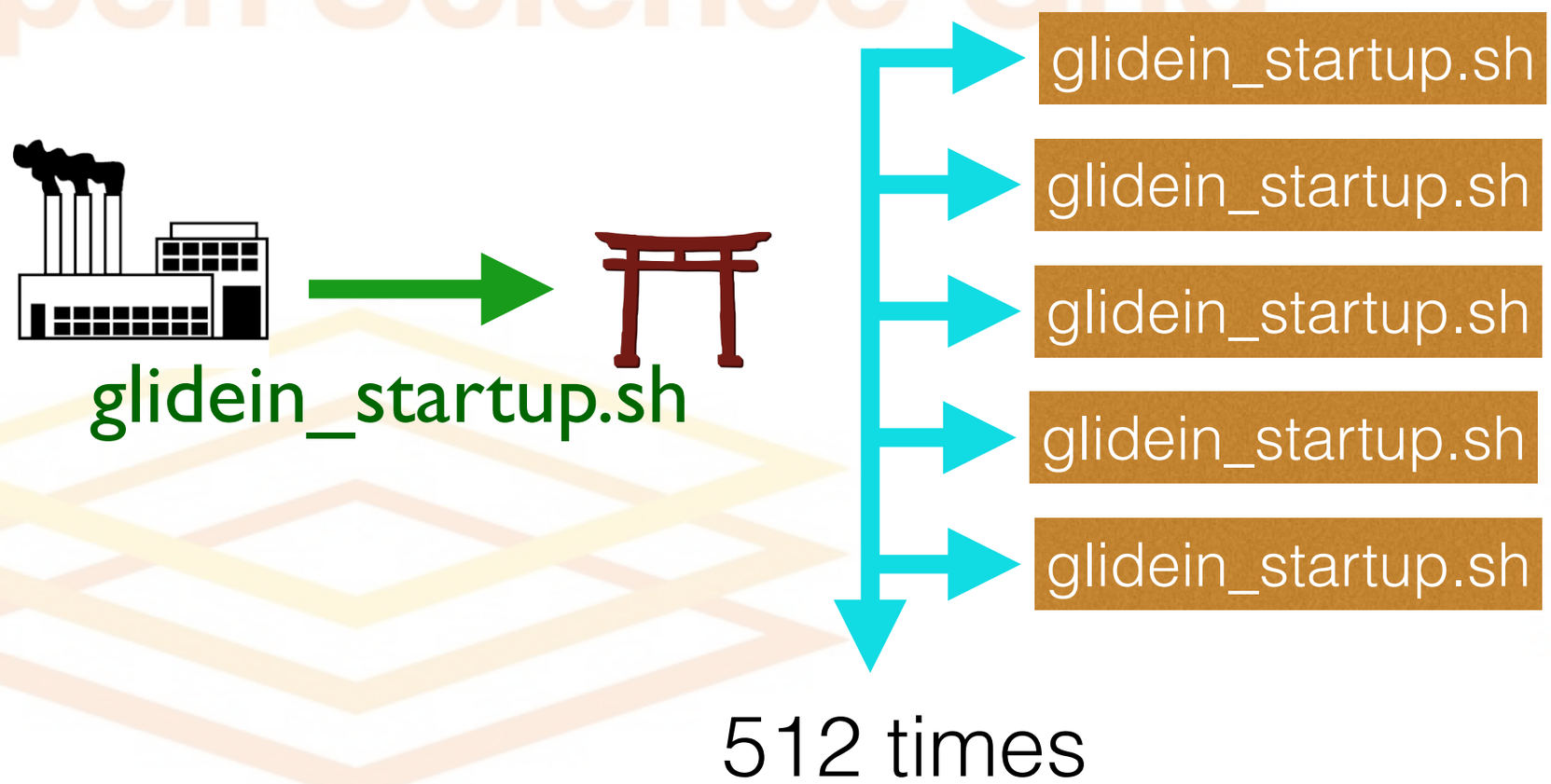
```
<entry name="Ligo_US_Stampede_gt5" auth_method="grid_proxy+project_id" comment="Added for LIGO 2015-12-05 note this is
an experimental entry! --Jeff" enabled="True" gatekeeper="login5.stampede.tacc.utexas.edu:/jobmanager-slurm"
gridtype="gt5" rsl="(job_type=multiple)(count=512)(host_count=32)(maxWallTime=2880)"
schedd_name="schedd_glideins1@glidein-itb.grid.iu.edu" trust_domain="grid" verbosity="std" work_dir="/tmp">
```

This tells gram+SLURM we will use 512 cores

```
<attr name="GLIDEIN_CPUS" const="True" glidein_publish="False" job_publish="True" parameter="True" publish="True" type="string"
value="512"/>
```

This tells the frontend that the pilot is getting 512 cores. In order not to overprovision

From the Stampede side it looks like this



UC San Diego

How to GlideIn at Stampede

But from each glide in perspective it should think it only has 1 core not 512. So on the Stampede entry:

```
<files>
            <file absfname="/etc/gwms-factory/force_one_cpu.sh" const="True" executable="True" period="0" untar="False"
wrapper="False">
               <untar_options cond_attr="TRUE"/>
            </file>
         </files>
```


UC San Diego

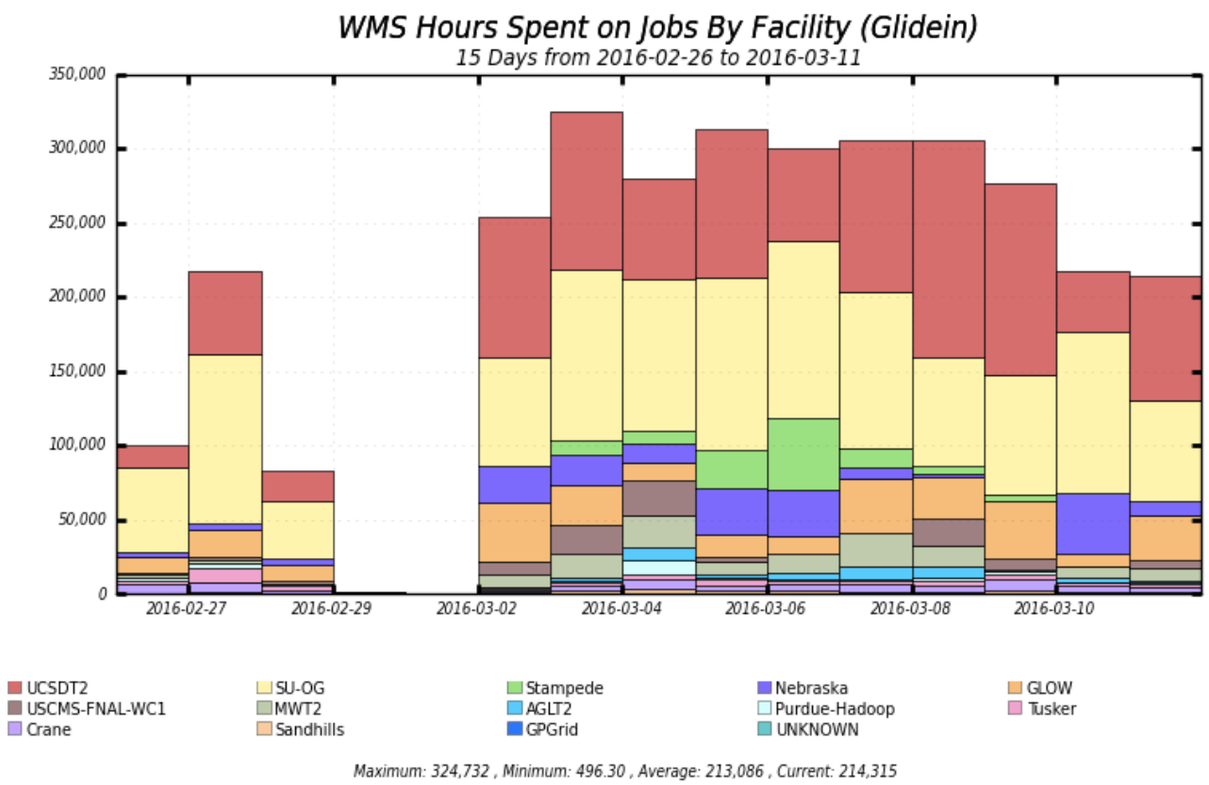
How to GlideIn at Stampede

- From then on is business almost as usual CVMFS over NFS.
- gridftping or gfaling the data-in and HTCondor file transfer for the data out.
- /tmp is mounted on all nodes for volatile storage

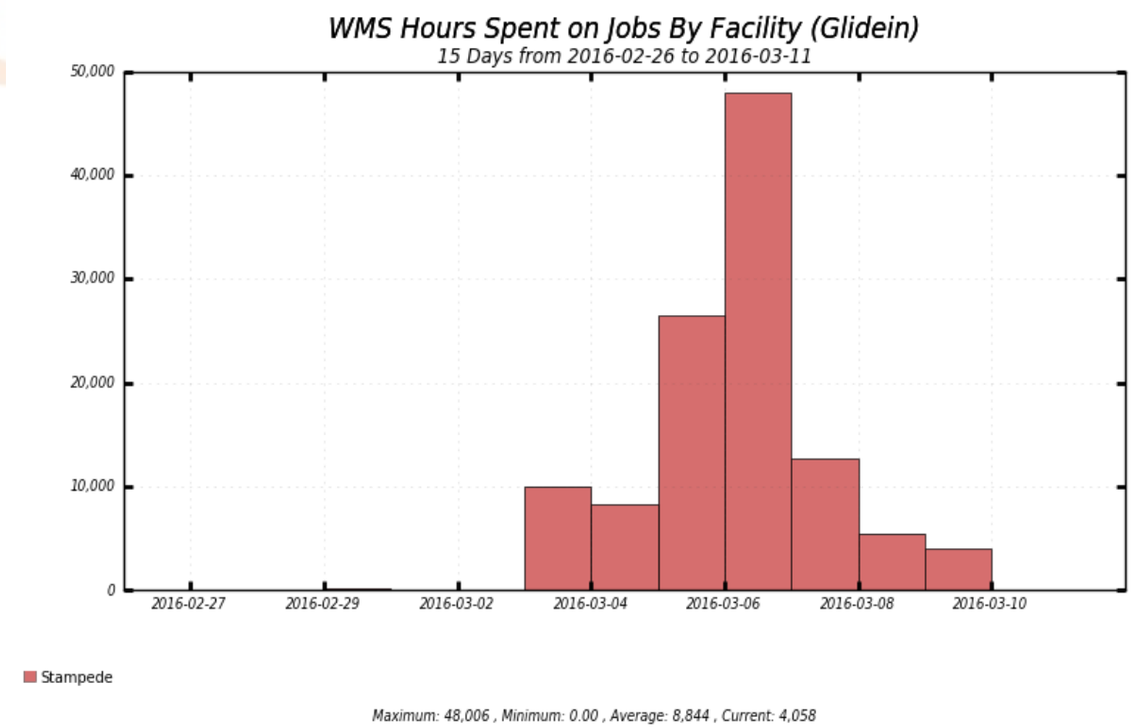
UC San Diego

LIGO on Stampede

So does this work?



CPU Hours in all OSG Sites by Ligo



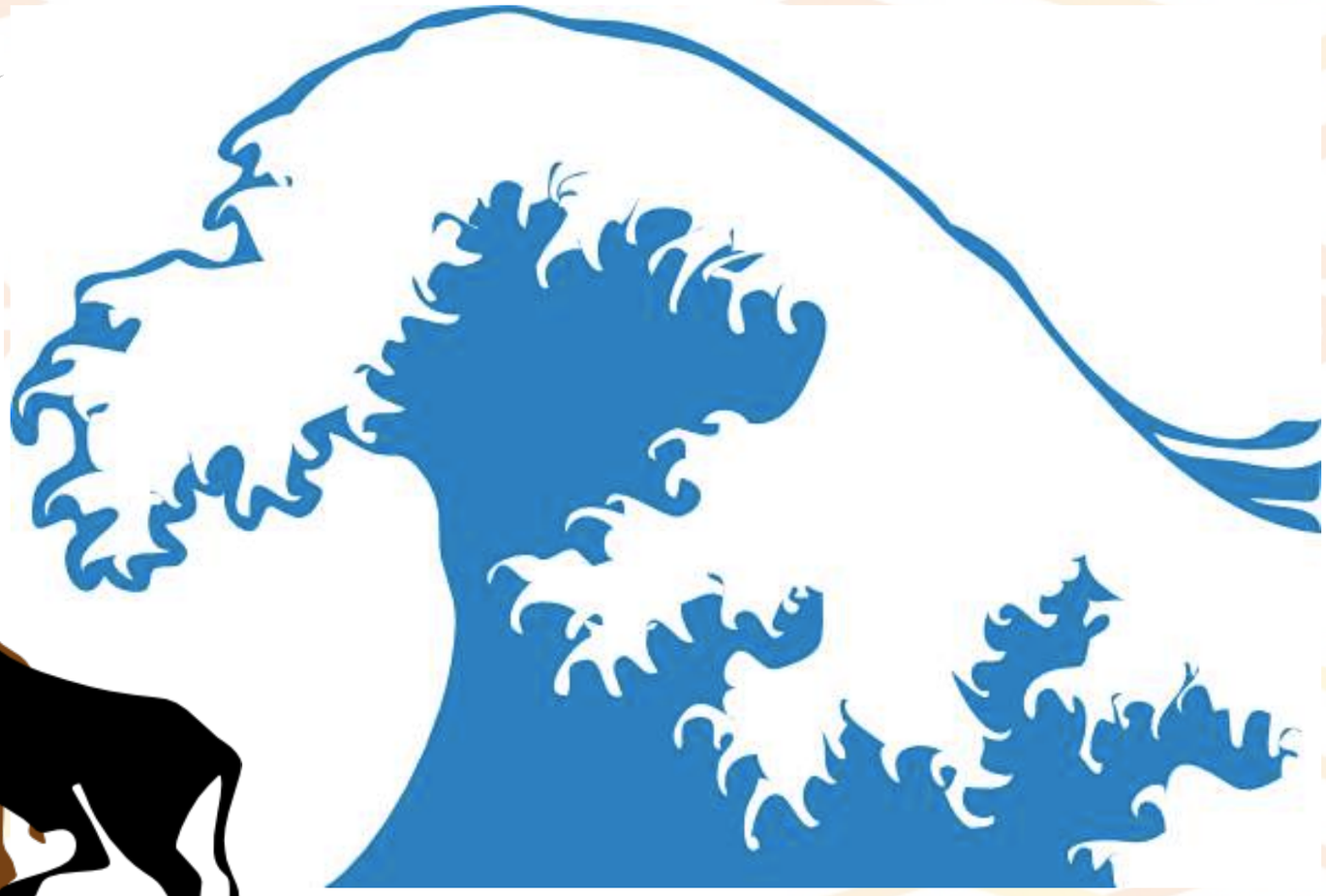
CPU Hours in Stampede by Ligo

UC San Diego

LIGO on Stampede

- From LIGO's perspective their jobs can run potentially in all of the OSG Sites + the XSEDE_SITES: aka late binding
- Its proven to work: after all they found the gravitational waves.
- But the multiple host glidein creates a nightmare for factory ops

In Summary



Catching a wave through gliding into an Stampede

UC San Diego

Questions? Comments?

Contact us at:

1-900-Stampede-masters

UC San Diego

Just Kidding

Contact us:

osg-software@opensciencegrid.org

UC San Diego