# Running LIGO onOpen SciStampedeOpen S

15 March 2016

Edgar Fajardo on behalf of OSG Software and Technology

OSG All Hands Meeting 2016



## 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.

**UC** San Diego

- GlideinWMS Team
- The Stampede folks

# 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:

Via general project id tag on the fronted config Tailored glideIns per job 2.



# General project\_id tag on the fronted

#### Open Science Grid 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/0=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"



# 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">



## About Stampede

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

Op	System Component	Specs	n
	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 Glideln at Stampede

 Associate a computing account with the DN of the pilot proxy.
 Have an allocation project\_name at the fronted in any of the two ways mentioned above.
 And voila submit with: +DESIRED\_XSEDE\_Sites="Stampede"

Not that fast. There is a catch.

**UC** San Diego

## How to Glideln 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

**UC** San Diego

## How to Glideln 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

**UC** San Diego

### From the Stampede side it looks like this glidein\_startup.sh glidein\_startup.sh glidein\_startup.sh glidein\_startup.sh glidein\_startup.sh

glidein\_startup.sh

512 times

OSG All Hands Meeting 2016 12

UC San Diego

## How to GlideIn at Stampede

But from each glide in perspective it should think it only has I 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>



## How to GlideIn at Stampede

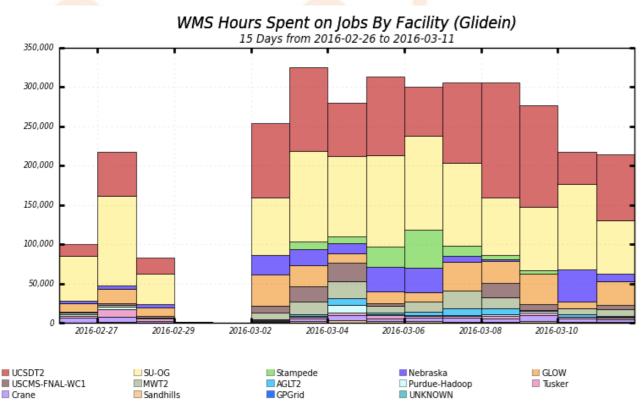
 From then on is business almost as usual CVMFS over NFS.

- gridftping or gfaling the data-in and HTCondor file transfer for the data out.
- /tmp is mounted on all nodes for volatile storage



## LIGO on Stampede

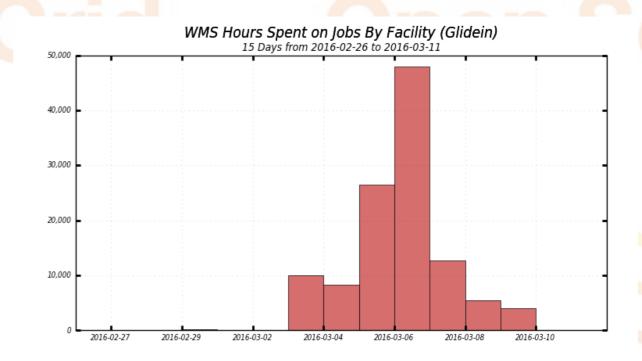
So does this work?



Maximum: 324,732 , Minimum: 496.30 , Average: 213,086 , Current: 214,315

CPU Hours in all OSG Sites by Ligo

OSG All Hands Meeting 2016 15



Stampede

Maximum: 48,006 , Minimum: 0.00 , Average: 8,844 , Current: 4,058

#### 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

**UC** San Diego

### In Summary

Catching a wave through gliding into an Stampede OSG All Hands Meeting 2016

## Questions? Comments?

Open ScieContact us at: 1-900-Stampede-masters



## Just Kidding

### Open ScierContact us: osg-software@opensciencegrid.org

