

# Table of Contents

<b>Scalability with SAGA, Ganga, DIANE through WLCG, and (Cloud) resources.....</b>	<b>1</b>
Planning.....	1
Tests.....	1
Application.....	1
Full machinery: Ganga, SAGA, DIANE.....	2
Using.....	2
Interactive access to clusters.....	3
Using resources without SAGA.....	3
Options for RANGER/TACC.....	3
Test runs of plain.executable on.....	3
Abe.....	4

# Scalability with SAGA, Ganga, DIANE through WLCG, and (Cloud) resources

Milestone: talk and demonstration at EGEE User Forum, beginning of April 2010

## Planning

Work plan to be completed by March 15th 10am C (=5pm CET)

1. access to TeraGrid and FutureGrid (Cloud) resources
  - ◆ KUBA: Ranger - still need to get access
  - ◆ KUBA: interactive access to Kraken be processed via a notary
  - ◆ OLE, SHANTENU: Kraken - GRAM submission to be investigated and solved
2. test of openMP application on TeraGrid: how it scales?
  - ◆ tests done Abe, QB
  - ◆ OLE: test on ranger
  - ◆ KUBA: testing connectivity (diane workers): need to verify the worker nodes on Abe, QB, RANGER
3. GangaSAGA/GRAM submission
  - ◆ OLE: improvement the GangaSAGA plugin (file handling + monitoring)
  - ◆ OLE: testing of GRAM access to all the machines in TG (submitting from lxplus using GangaSAGA backend)
4. KUBA: replicate the mother snapshots

## Tests

Define application level metrics and design and conduct the experiments.

Basic objectives:

1. **interoperability**
2. **scale-out:**
  - ◆ how many of runs we can perform concurrently?
3. **use of DIANE for smart scheduling decisions**
  - ◆ use the performance measurements and feedback to a) task scheduler (optimal placement of tasks on a set of resources), b) agent factory (optimal influx of new resources)
  - ◆ we have one additional parameter to steer the system: number of OMP threads (cores), this poses an extra complication on "squeezing" the tasks into CPU-slots of nodes which are typically fixed size (8)

## Application

LQCD 2010 application is a natural continuation of 2008 simulations and links with it in the scope of theoretical physics research.

It requires openMP and at CERN has been compiled with Intel Fortran Compiler version 11.

Code: `/afs/cern.ch/sw/arda/install/su3/2010/su3`

Tarfile available for download (with precompiled executable and ganga helper scripts for local batch submission): `wget http://cern.ch/diane/lqcd_2010/su3_RHMC_Nt6_omp.tgz`

Compilation instructions on lxplus: [README\\_lxplus](#)

Timing tests (conducted by the user, PhdF):

```
46490.739u 914.793s 35:05.14 2251.8%   NCPU=32
25441.683u 368.346s 36:06.08 1191.5%   NCPU=16
20342.263u 147.546s 51:12.31 666.9%   NCPU=8
12567.237u 79.296s 1:00:01.40 351.1%   NCPU=4
7688.528u 26.305s 1:09:12.17 185.8%   NCPU=2
7840.101u 0.416s 2:10:41.77 99.9%    NCPU=1
```

Timing tests in TG:

OMP_THREAD_NUM	queenbee	abe
1	90m15s	97m4s
2	54m57s	61m38s
4	49m33s	50m29s
8	51m16s	50m11s

## Full machinery: Ganga, SAGA, DIANE

The master is running on lxarda28 in /data/lqcd/apps/output

The worker agents may be submitted from lxplus to TG like this (change the tguser name!):

```
cd /afs/cern.ch/sw/arda/install/TeraGrid
source env.sh
$(/afs/cern.ch/sw/arda/diane/install/2.0-beta20/bin/diane-env)

diane-env -d ganga ./SAGASubmitter.py --tguser jmoscick --diane-master=./MasterOID
```

To submit from another host you need

- install DIANE: <http://it-proj-diane.web.cern.ch/it-proj-diane/install.php>
- make sure you have SAGA and GangaSAGA configured

The steps are then the same as on lxplus, you need to get the SAGASubmitter.py and MasterOID files from AFS to your local system.

## Using

The latest **1.4.1** release of SAGA is deployed in:

```
/afs/cern.ch/sw/ganga/external/SAGA/1.4.1/
```

and will be automatically sourced by env.sh script below.

```
> cd /afs/cern.ch/sw/arda/install/TeraGrid
> ./myproxy_logon #you may use a different user name here
> source env.sh
> ganga
```

More info how to submit jobs here: <http://faust.cct.lsu.edu/trac/saga/wiki/Applications/GangaSAGA>

List of GRAM services in TeraGrid: <https://www.teragrid.org/web/user-support/gram>

## Interactive access to clusters

Web resources:

- [http://www.teragrid.org/web/user-support/login\\_quickstart](http://www.teragrid.org/web/user-support/login_quickstart)
- (QueenBee) [http://www.loni.org/teragrid/users\\_guide.php](http://www.loni.org/teragrid/users_guide.php)

Interactive access from lxplus (SLC4):

```
>ssh lxplus4
> cd /afs/cern.ch/sw/arda/install/TeraGrid
> ./myproxy_logon #you may use a different user name which is hardcoded in this script
Enter MyProxy pass phrase:
A credential has been received for user jmoscick in /tmp/x509up_u979.

> ./gssh login1-qb.loni-lsu.teragrid.org # login to queenbee
```

Ranger: for Kuba gssh/TG access does not work. Connect directly with ssh and password.

## Using resources without SAGA

First time setup:

1. Login interactively to a TG machine.
2. Install latest version of DIANE: <http://cern.ch/diane/install.php>
3. Get additional files, MasterOID and PBSSubmitter.py, from /afs/cern.ch/sw/arda/install/TeraGrid
4. PBSSubmitter.py script was developed for QueenBee. If needed change the j.backend.extraopts options in PBSSubmitter.py, depending on the configuration and requirements of local PBS installation
5. set environment: `$(~/diane/install/2.0-beta20/bin/diane-env)`

Command to add 200 workers to the system:

```
diane-env -d ganga ./PBSSubmitter.py --diane-master=./MasterOID --diane-worker-number=200
```

## Options for RANGER/TACC

Documentation: <http://services.tacc.utexas.edu/index.php/ranger-user-guide>

```
j.backend=PBS()
j.backend.extraopts='-V -l h_rt=24:00:00 -pe 4way 16'
j.queue='normal'
```

Note: these settings are probably wasting some cores in the node...

## Test runs of plain executable on

```
> gssh login1-qb.loni-lsu.teragrid.org

[jmoscick@qb1 ~]$ qsub -l nodes=1:ppn=8 test.sh
212566.qb2
```

Note: QueenBee supports allocation of entire nodes only (ppn=8 always)

## Abe

System information:

<http://www.ncsa.illinois.edu/UserInfo/Resources/Hardware/Intel64Cluster/Doc/Jobs.html#PBS>

I installed ganga 5.5.0, PBS hello world jobs run fine.

Compiled the app in ~/su3. It looks like there are no limits for stacksize in the system. Here is my application wrapper:

```
[jmoscick@honest3 ~/su3]$ cat run_su3.sh
#!/usr/bin/env bash

#increase stack size if needed
#ulimit -s 50000
ulimit -a
export OMP_NUM_THREADS=$1
./hmc_su3_omp <$2
```

Here is a small utility to submit test jobs:

```
In [100]:execfile("su3.py")
usage:
su3submit (OMP_THREADS_NUM)
su3test ()
su3show (j)

In [101]:su3show(jobs(10))
Out[101]: job 10: elapsed time: 1:09:30.595613 for 2 OMP threads

In [102]:su3show(jobs(11))
Out[102]: job 11: elapsed time: 1:09:30.600562 for 4 OMP threads

In [103]:su3show(jobs(12))
Out[103]: job 12: elapsed time: 1:09:30.606223 for 8 OMP threads
```

-- JakubMoscicki - 08-Feb-2010

---

This topic: ArdaGrid > LatticeQCDTeraGrid2010

Topic revision: r12 - 2010-07-06 - JakubMoscicki



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)