

# Table of Contents

<b>SU3 QCD application</b> .....	<b>1</b>
Second round (Diane 2.0 - April 2008).....	1
Analysis of logs.....	1
Small backup sample.....	1
Large Atlas sample.....	2
Production 2008.....	3
Monitoring crontabs.....	3
How to start a master (and a file server). No set up.....	4
How to submit more.....	4
Submitting using AgentFactory.....	5
Questions.....	5
.fake.snapshots.for.testing.....	5
General talks on the subject.....	5
Old links.....	5

# SU3 QCD application

## Second round (Diane 2.0 - April 2008)

- What's up today?: LogFile08
- Standing issues: ProblemsToBeSolved08
- *Preparation: meeting with Philippe de Forcrand; MeetingWithPhilippe*
- *Creation of MotherSnapshots*
- *Set up of the system: SystemSetUp08*

## Analysis of logs

This is information from 2010 (LQCD paper preparation). It is incomplete.

We analyze the reasons for failures using the failure\_log of the agent factory. This information is not complete, we have only a small snapshot.

Here is the procedure: we extract the ids of each type of failure into the processedNN files, where NN is the failure type number. Then we mkdir processed directory where the logs of each type will be moved (to filter out these logs each time from the main failure\_log directory). To filter out we do: `python move_processed.py processedNN`

The more details on CE and SITES is provided by this script: `python analyze_processed.py processedNN`

The results are committed to CVS files (for Atlas sample): `RUN_41_AF_fail_logs_*`

We store all ids of processed logs in all\_processed file: `ls processed > all_processed`

A cross type analysis is a special case where we do: `python analyze_processed.py all_processed`. It gives all count of types of problems (as described below, according to their ids stored in processedNN files)

The py helper scripts are in CVS: `LatticeQCD2/analysis_tools`

## Small backup sample

```
[lxarda29] /data/lqcd2008/backup_factory_logs/lostman.gangadir
```

Unzip output files:

```
find agent_factory -name stdout.gz -exec gunzip {} \;  
find agent_factory -name stderr.gz -exec gunzip {} \;
```

Number of workers: 249

```
find agent_factory -name stderr -exec echo {} \; | wc
```

Show all logs

```
find agent_factory -name stderr -exec cat {} \; | less
```

Types of problems:

Type 1:

## SU3 < ArdaGrid < TWiki

**126 times** DIANE\_CORBA.XFileTransferError(message="[Errno 2] No such file or directory: '/data/lqcd/apps/output/hmc\_su3\_newmuI.amd\_opteron'")

```
find agent_factory -name stderr -exec grep -l amd {} \; | wc
```

**Type 2: 51 times** This program was not built to run on the processor in your system

```
find agent_factory -name stderr -exec grep -l processor {} \; | wc
```

**Type 3: 23 times** error executing... sh: line 1: 14351 Killed  
./hmc\_su3\_newmuI <parameters

```
find agent_factory -name stderr -exec grep -l 'error executing' {} \; > processed3
```

**Type 4: 12 times** communication error

```
return _omnipy.invoke(self, "download", _0_DIANE_CORBA.FileTransferServer._d_download, args)  
UNKNOWN: CORBA.UNKNOWN(omniORB.UNKNOWN_PythonException, CORBA.COMPLETED_MAYBE)
```

```
find agent_factory -name stderr -exec grep -l 'omniORB.UNKNOWN_PythonException' {} \; > processed
```

**Type 5: 15 times** python version(?) problem: ImportError: No module named logging

```
find agent_factory -name stderr -exec grep -l 'ImportError: No module named logging' {} \; > proo
```

**Type 6: 6 times** = ERROR: could not download the file : omniORB-4.1.2-slc3\_gcc323-diane.tar.gz=

```
find agent_factory -name stderr -exec grep -l 'GANGA_VERSION' {} \; > processed6
```

**Type 7: 11 times** OSError: [Errno 13] Permission denied: '/home/atlas157/diane/submitters'

```
find agent_factory -name stderr -exec grep -l 'Permission denied' {} \; > processed7
```

Remaining errors: 5

1. error with file system permissions
2. http download failure
3. SOCK problem with http download
4. MAINTENANCE file download problem
5. OS type problems (Suse, Debian)

### Large Atlas sample

```
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir
```

It corresponds to a subset of run 41 (first part of RUN4 in LQCD paper), from 27.08.2008 to 18.09.2008. The logs are available in: /data/lqcd2008/apps/output/logs-run4-20081028.tgz

Worker agent numbers, from 1 to 12069. Total number of invalid worker agents reported from AF: 4163-801(success)=\*3354\*. It quite well coincides with the number **3704** reported in the master log in that period (available output by running analyze\_vcard.py script on RUN\_41\_vcard\_summary\_INVALID\_WORKERS.dat)

Number of invalid workers: 1379 with stderr available.

Small backup sample

Type1: **186**

Type2: **41**

Type3: **410**

Type4: **0**

Type5: **87**

Type6: **8**

Type7: **2**

Type8: **398**

Type9: **180**

Type8 + Type9: 578 times a CORBA problem with worker registration

```
find agent_factory -name stderr -exec grep -l 'return _omnipy.invoke(self, "registerWorker", _0_D
```

Type 8: **398** communication problem

```
find agent_factory -name stderr -exec grep -l 'TRANSIENT: CORBA.TRANSIENT(omniORB.TRANSIENT_Conne
```

Type 9: **180** communication problem (timeout)

```
find agent_factory -name stderr -exec grep -l 'TRANSIENT: CORBA.TRANSIENT(omniORB.TRANSIENT_CallT
```

Type 910: **59** http socket error (IOError: [Errno socket error] (110, 'Connection timed out'))

```
find agent_factory -name stderr -exec grep -l 'socket error' {} \; | wc
```

Other types **8** times: including Debian platform and strange socket errors

The reminder of 2784 invalid workers does not have stderr available:

```
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
801 5607 69670
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
2782 22736 253479
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
588 4704 48165
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
282 2824 41041
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
715 6441 62664
[lxarda29] /data/lqcd2008/apps/output/lostman_atlas.gangadir > grep reason agent_factory/failure_
398 3184 32155
```

## Production 2008

### Monitoring crontabs

There is acron on lxplus for publishing the live plots on the wiki page and the cron on each master server (lxarda28) for collecting the data for the plots and doing cleanups.

Check currently defined acrons and crons:

- acrontab -l
- crontab -l

and compare them with: `* /storage/lqcd/apps/output/monitoring-and-cleanup.crontab= *`  
`/afs/cern.ch/sw/arda/install/su3/2009/plots/web-monitoring.acrontab=`

Install missing corntabs.

- **master server:** `crontab /storage/lqcd/apps/output/monitoring-and-cleanup.crontab`
- **lxplus (acrontab for web publishing):** `acrontab < /afs/cern.ch/sw/arda/install/su3/2009/plots/web-monitoring.acrontab`

## How to start a master (and a file server). No set up

**NOTE:** *rev 40 is the last revision of this doc that worked for old lxb1420 machine (new versions should be compatible but...)*

- Log in to **lxarda28** (old servers: **lxb7232 lxb1420**)
- Main sequence of commands:

```
bash

# setup diane
source /storage/lqcd/env.sh

# go to the output area
cd /storage/lqcd/apps/output

# run the file server
./start_file_server

# run master using an open port
./start_master_server

# OPTIONAL:

# install cron to monitor the server processes (connections, cpu, memory etc)
# if you restart the servers then remove the old cron
cron -e
#* * * * * /storage/lqcd/apps/output/servermon PID_MASTER PID_FILESERVER

# you may get the PIDs using this command
netstat -lp | grep python
```

- **diane-master-ping verbosity [INFO|DEBUG]** to switch the debugging level

## How to submit more...

**NOTE:** *rev 41 is the last revision of this doc before the submit\_more command was introduced*

```
bash
cd /afs/cern.ch/sw/arda/install/su3/2009
./submit_more config-kuba.gear lxb1420 LCG.py NUMBER [opts: --delay 10, --CE ce]
```

The name of the configuration file must be `config-PROXYNAME` and must match the proxy file `p/PROXYNAME`

LCG.py may be substituted by any submitter script (LSF.py) etc. If the submitter script is not specified then ganga is run interactively.

## Submitting using AgentFactory

The procedure below uses `$HOME/proxy` to store the long proxy and `extend.sh` script; if you prefer different location, please make the appropriate changes to the scripts!

(i) create the long-lived proxy (this will create a proxy valid for a month) and myproxy

```
mkdir $HOME/proxy
cd $HOME/proxy
/afs/cern.ch/sw/arda/install/su3/2009/agent_factory/mkproxy.sh
```

(ii) add the `extend.sh` script acrontab:

```
acrontab -e
```

add the following line:

```
30 10,22 * * * lxarda28 /afs/cern.ch/sw/arda/install/su3/2009/agent_factory/extend.sh
```

(iii) run Agent Factory (preferably on screen):

```
cd /afs/cern.ch/sw/arda/install/su3/2009
screen
./agent_factory/run_agentfactory.sh CONFIG_FILE HOST WORKER_NUMBER
```

where `CONFIG_FILE` is your configuration file and `WORKER_NUMBER` is the amount of workers you want to have.

## Questions

- !!! diane-master-ping kill should have ask for confirmation 😞
- Is this line (master log) correct?  
<verbatim>2008-05-02 09:03:34,311 INFO: config.main.cache = ~/dianedir/cache></verbatim>

## fake snapshots for testing

```
# TESTING ONLY
cd /storage/lqcd/apps/output
cp /storage/lqcd/apps/LatticeQCD2/test/* .
python /storage/lqcd/apps/LatticeQCD2/make_test_snap.py
```

## General talks on the subject

- Andrew Maier's talk at CHE2007

## Old links

- ProductionLattice2007

-- JakubMoscicki - 26 Jun 2007

---

This topic: ArdaGrid > SU3

Topic revision: r52 - 2010-04-14 - 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](#)