APPENDIX IV: USEFUL UTILITIES AND TRICKS

Utilities

Copying files from the Grid to Hep01: Makefiles

```
all: gridfiles.md5


gridfiles.xml: query.sh
    ./$< > $@


gridfiles.md5: gridfiles.xml
    xsltproc /alice/data/util/xml2md5.xsl $< > $@


download: $(shell cut -c 49- files.md5)


/alice/data/%:
    mkdir -p $(dir $@)
    alien_cp alien:$@ file:$@
```

Copying directories efficiently from a server to a local machine: rsync

```
rsync -av --stats --progress --include="*/" --include="*.txt" --exclude="*.C"
user@server.host.url.ac.za:/path/to/directory/ .
```

User specified aliases in ~/.bashrc

```
# User specific aliases and functions to quickly enter AliPhysics from
CVMFS
alias init_ali='/cvmfs/alice.cern.ch/bin/alienv enter
VO_ALICE@AliPhysics::vAN-20180902-1'
# Or to enter your own installation of AliPhysics
alias my_alice='alienv -w /alice/user/alice/sw enter
VO_ALICE@AliPhysics::latest'
```

Remote Editing

X11 Forwarding

Atom packages:

- Remote Atom Server
- PlatformIO-IDE-Terminal

Killing a process being listened to on the remote port 52698:

List processes that are owned by me:

```
ps aux | grep gviljoen
```

Find the sshd process being listened to on port 52698 and kill it, by running:

```
kill -9 $processid
```

Using Keras in R on the UCT HPC Cluster (not recommended)

Compiler variables set in ~/.R/Makevars

```
CC = gcc -std=gnu99
```

Install.packages command needs to be modified to write packages in a directory where there are permissions and where the CRAN mirror is set, dependencies=TRUE allows R to read the Makevars compiler variables.

install.packages(pkgs="keras",lib="/scratch/username",

repos="https://cloud.r-project.org",dependencies=TRUE)

APPENDIX V: RUNNING AND MONITORING ROOT ANALYSIS TASKS

Once one is happy with the analysis task defined, one first needs to enter AliPhysics, by using one of the user-defined aliases, e.g.:

```
initialize_aliroot
```

Then, one gets a token from alien, to access the grid. This token will be valid for 24 hours. Since my CERN username is not the same as my username on HEP01, the command is:

```
alien-token-init username
```

Once the above commands have been run, one can run the analysis task on the grid, by setting the following parameters in the analysis macro (ana.C):

```
  Bool_t local = kFALSE;
  Bool_t gridTest = kFALSE;
```

Adding the appropriate run number and output directory:

```
alienHandler->AddRunNumber(265377);
```

```
alienHandler->SetGridWorkingDir("new-wd-momentum-test");
alienHandler->SetGridOutputDir("outDir265378");
```

Setting the run mode and starting the analysis:

```
alienHandler->SetRunMode("full");
//alienHandler->SetRunMode("terminate"); //this is run for merging stages
mgr->StartAnalysis("grid");
```

Assuming that one has added the appropriate CERN certificates, one can then view, manage and download the output of one's jobs on the MonALISA grid monitoring site for ALICE see **Figure 1** for an example screenshot of user job monitoring and **Figure 2** for the user interface for viewing the directory structure for the ALICE grid, in particular the user's working directory:
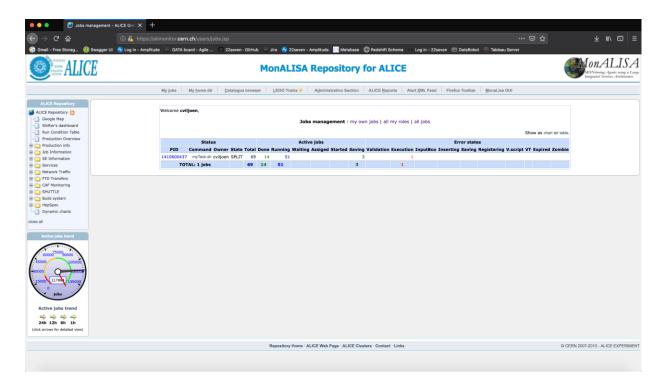

Figure 1: MonALISA Alice grid monitoring site, user jobs at url:
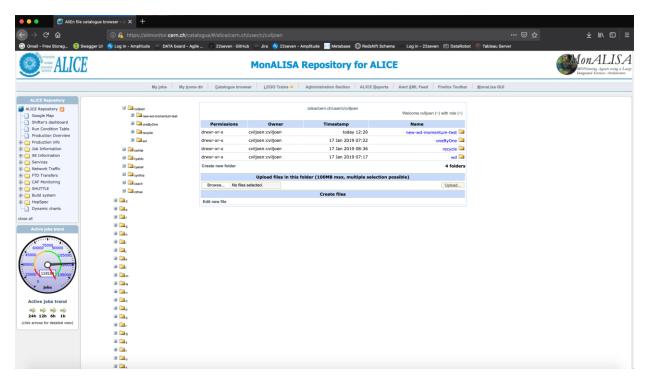
https://alimonitor.cern.ch/users/jobs.jsp

Figure 2: User working directory structure on MonALISA at url:

https://alimonitor.cern.ch/catalogue/#/alice/cern.ch/user/c/cviljoen

In **Figure 3**, a screenshot shows how subjobs belonging to a masterjob can be tracked by clicking on the process ID on the MonALISA jobs management webpage:
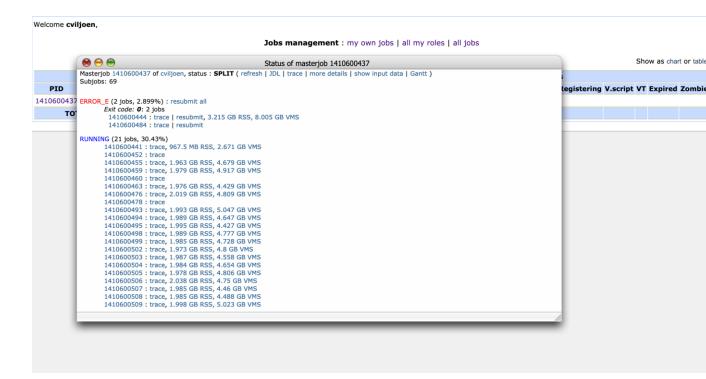

Figure 3: Tracking the status of subjobs of a master-job, by clicking on the process id (PID)

One can resubmit errored subjobs by browsing through the various error states in the "Status of masterjob" view and clicking on "resubmit all" for all processes that are in a specific error state.

The trace of a subjob (see **Figure 4** for an example screenshot) can give hints as to what caused a specific subjob to fall into an error state. In this case the job has an error state "ERROR_E", i.e. "Error in Execution", since the job is using too much memory (memory and storage limits are allocated to each user and overusing either can downgrade the priority of a user's jobs).

The alien shell can be accessed by running

```
aliensh
```

This gives access to the alien terminal, which is not strictly a bash terminal, but has similar commands, for instance the shell command to forcefully and recursively remove a directory:

```
rm -rf directory
```

would be achieved on an alien terminal by running:

```
rmdir directory
```

Killing a job is done in a similar fashion to the normal shell workflow, i.e. running

```
ps
```

To list the currently active processes and

```
kill $(process-id)
```

To kill a process and its attendant subprocesses, in case you figured out that you made a mistake and want to terminate a running process early for whatever reason.

Figure 4: Example trace of a subjob on MonALISA