AGILE Real-Time Analysis Pipeline Setup

N.Parmiggiani[(1)], A. Bulgarelli[(1)], S. Tampieri[(1)], L. Baroncelli[(1)], A. Addis[(1)]

[(1)]INAF/OAS – Bologna, Via P. Gobetti 93/3, 40129 Bologna

Change history:

Version	Date	Notes
1.0	Jan 23th, 2019	
1.1	Sep 18th, 2020	Update of procedures

Table of contents

1. Introduction **4**

2. Containers: layers **4**

3. Installation of basic packages inside the container **5**

3.1. Anaconda3 5

3.2. Services 5

3.3. Daemons with crontab 7

3.4. Add diskusage scripts 8

4. AGILE DeepVar Installation **8**

5. Science GUI dependencies **10**

6. APPENDIX: installation procedure for old pipelines **11**

6.1. Installation of AGILE pipelines **11**

6.1.1. AGILE ALERT PIPE 11

6.1.2. Anaconda2 11

6.1.3. MCAL results DB 12

6.1.4. GCN DAEMON 13

6.1.5. AGILE MCAL PIPE 2 **13**

1. Introduction

AGILE pipelines setuo

AGILE pipelines can be downloaded with this setup https://github.com/AGILESCIENCE/rta-sci-agile-Setup

2. Containers: layers

AGILE pipelines are deployed into a Singularity container environment.

To build the containers, download this repository: https://github.com/ASTRO-EDU/AGILE-containers

Each container has four files.

- .build is used to build a layer.
- .start file is used to start a container and contains different commands.
- .simg is the container image.
- .recipe contains all commands needed to build the container

The container environment is created with different layers, each with a specific function.

1) Operative system and base libraries
2) Science tools installation
3) System services
4) Dummy
5) Pipeline installation
6) Layer with aspecific configuration for a particular machine (IP, port etc)

To build the container, the user can be root inside the machine and can use the script buildall.sh, which contains a list of commands. Each command runs the build of a single layer.

sh agile_layer1_core.build
sh agile_layer2_scitools.build
sh agile_layer3_service.build
sh agile_layer5_instance.build
sh agile_layer6_[machine]_[user]_instance.build

3. Installation of basic packages inside the container

The following commands can be performed into a machine or written into the recipe to build a container.

3.1. Anaconda3

Install anaconda3 https://www.anaconda.com/download/#linux

Install Manually:

```
root@machine# cd ..
root@machine# wget https://repo.anaconda.com/archive/Anaconda3-5.3.0-Linux-x86_64.sh
Execute installer
Setup install directory  when program ask for: /opt/prod/anaconda3
root@machine# bash Anaconda3-5.3.0-Linux-x86_64.sh
```

Create /opt/prod/modulefiles/anaconda-3.6

```
root@machine# vim /opt/prod/modulefiles/anaconda-3.6
```

```
#%Module
set path /opt/prod/anaconda3
prepend-path PATH $path/bin
unsetenv PYTHONPATH
prepend-path LD_LIBRARY_PATH $path/lib
```

Create environment

```
rt@machine $ module load anaconda-3.6
rt@machine $ conda create -n pipe_manager_py36 python=3.6 astropy=3.0 lxml=4.1.1
mysql-connector-python=2.0.4
```

3.2. Services

SETUP MySQL /etc/my.cnf

```
[mysqld]

sql_mode =
'NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGI
NE_SUBSTITUTION'
```

```
default-character-set=utf8
federated
symbolic-links=0
bind-address =  0.0.0.0
innodb_buffer_pool_size=1024M
innodb_log_file_size=64M
max_connections = 500
query_cache_size = 64M
slow_query_log = ON
query_cache_type = 1
max_heap_table_size = 64M
tmp_table_size = 64M
query_cache_limit= 64M
slow_query_log=ON
log_output=TABLE
event_scheduler=ON
```

Restart mysql

```
root@agilepipe# systemctl restart mysqld
```

If not restart check /var/log/mysqld.log

SETUP pipeline_manager database

```
rt@machine$ cd /opt/prod/pipeline_manager/scripts
rt@machine$ cp conf.py.default conf.py
rt@machine$ vim conf.py
rt@machine$ cp repository.conf.default repository.conf
rt@machine$ vim repository.conf
```

Setup conf.py and repository.conf -> look specific pipe manual

Setup Slurm

Install slurm https://www.slothparadise.com/how-to-install-slurm-on-centos-7-cluster/

```
rt@machine$ cd ../slurm
root@machine# cp slurm.conf /etc/slurm/slurm.conf
root@machine# cp slurmdbd.conf /etc/slurm/slurmdbd.conf
```

Configure the last two files.

Follow this tutorial https://slurm.schedmd.com/accounting.html

Create database, cluster and user

```
rt@machine$ mysql -uroot -p --execute="create database slurm_acct_db;"
root@machine# sacctmgr add cluster [clustername]
root@machine# sacctmgr create user name=rt cluster=[clustername] account=root
rt@machine$ cd /opt/prod/pipeline_manager/database
rt@machine$ cp trigger_slurm_status_update.sql.default trigger_slurm_status_update.sql
Open the trigger and change pipe-db name for each pipe to connect
rt@machine$ mysql -uroot -p slurm_acct_db < trigger_slurm_status_update.sql
```

Create reservation if needed

```
rt@machine$ vim slum_init.sh
Update info
root@machine# sh slurm_init.sh
```

Setup website optionally

```
rt@machine$ cd /opt/prod/pipeline_manager/website
rt@machine$ cp config.php.defautl config.php
Update config
root@machine# ln -s /opt/prod/pipeline_manager/website /var/www/html/website
```

3.3. Daemons with crontab

Add following lines to crontab

```
PIPEMANAGER = /opt/prod/pipeline_manager/scripts

### PIPE MANAGER

# update repository
* * * * * pgrep  -u "$(whoami)" -xf "python ./update_repository.py 1" > /dev/null || (. ~/.bashrc ; date >>
$LOG/update_repository.log ; cd $PIPEMANAGER ; source ../skeleton.ll ; python ./update_repository.py 1
>> $LOG/update_repository.log 2>&1)

# submit run
* * * * * pgrep  -u "$(whoami)" -xf "python ./submit_run.py 1" > /dev/null || (. ~/.bashrc ; date >>
$LOG/pipe_manager_submit.log ; cd $PIPEMANAGER ; source ../skeleton.ll ; python ./submit_run.py 1 >>
$LOG/pipe_manager_submit.log 2>&1)

# cancel run
* * * * * pgrep  -u "$(whoami)" -xf "python ./cancel_run.py 1" > /dev/null || (. ~/.bashrc ; date >>
$LOG/pipe_manager_cancel.log ; cd $PIPEMANAGER ; source ../skeleton.ll ; python ./cancel_run.py 1 >>
$LOG/pipe_manager_cancel.log 2>&1)
```

3.4. Add diskusage scripts

In root crontab add the following line

0 */2 * * * /root/scripts/diskusagealert.sh > /dev/null 2>&1

The script /root/scripts/diskusagealert.sh is the following:

Chmod u+x /root/scripts/diskusagealert.sh

```
#!/bin/bash

ADMIN="[email]"
THRESHOLD=90
TMP_THRESHOLD=10000

df -PkH | grep -vE '^Filesystem|tmpfs|cdrom|media' | awk '{ print $5 " " $6 }' | while read output;
do
    usep=$(echo $output | awk '{ print $1}' | cut -d'%' -f1 )
    partition=$(echo $output | awk '{print $2}' )
    if [ $usep -ge $THRESHOLD ] ; then
        echo "$(df -h)" |
        mail -s "Alert from $(hostname): disk usage $usep% on partition $partition" $ADMIN
    fi
done

tmp_mb=$(du -sm /tmp/ | cut -f1)
if [ $tmp_mb -ge $TMP_THRESHOLD ] ; then
    echo "$(du -sh /tmp)" |
    mail -s "Alert from $(hostname): /tmp usage above $TMP_THRESHOLD MB" $ADMIN
fi
```

4. AGILE DeepVar Installation

Download DeepVar BUILD repository

```
rt@machine $ cd /opt/prod
rt@machine $ git clone https://[user]@github.com/ASTRO-EDU/DeepVar.git DeepVar_B1
rt@machine $ cd DeepVar_B1
rt@machine $ git checkout BUILD1
rt@machine $ ./download_modules.sh [git username] -t
```

Create deepvar_build1

```
root@machine# module av
root@machine# vim [module path]/deepvar_b1
```

```
#%Module
module load agile-B25-r5
setenv DEEPVAR /opt/prod/DeepVar_B1
```

Dowload cat2 scripts

```
rt@machine $ cd /home/rt/builds
rt@machine $ git clone https://github.com/ASTRO-EDU/AGILE-GRID-scripts-cat2.git
rt@machine $ cd AGILE-GRID-scripts-cat
rt@machine $ git checkout v1.0.1
rt@machine $ module load deepvar_b1
rt@machine $ ./install.sh
```

Create link for website

Create deepvar_build1

```
root@machine# cd /var/www/html
root@machine# module load deepvar_b1
root@machine# ln -s $DEEPVAR/website spot6
```

Create Database

```
rt@machine $ cd $DEEPVAR/database
rt@machine $ mysql -uroot -p --execute='create database cat2'
rt@machine $ mysql -uroot -p --execute='create database deepsearch1'
rt@machine $ mysql -uroot -p --execute='create database spot6'
rt@machine $  mysql -uroot -p  cat2 < cat2_schema.sql
rt@machine $ mysql -uroot -p  deepsearch1 < cat2_schema.sql
rt@machine $ mysql -uroot -p  spot6 < spot6_schema-28-2-18.sql
```

Create rt user and grant permission to these databases

```
rt@machine $ mysql -uroot -p --execute='CREATE USER rt@% IDENTIFIED BY '[PASS]';'
rt@machine $ mysql -uroot -p --execute='GRANT ALL PRIVILEGES ON cat2.* TO 'rt'@'%';'
rt@machine $ mysql -uroot -p --execute='GRANT ALL PRIVILEGES ON deepsearch1.* TO 'rt'@'%';'
rt@machine $ mysql -uroot -p --execute='GRANT ALL PRIVILEGES ON spot6.* TO 'rt'@'%';'
rt@machine $ mysql -urt -p cat2 < sqrtts_radius.sql
```

```
rt@machine $ mysql -urt -p deepsearch1 < sqrtts_radius.sql
rt@machine $ mysql -urt -p cat2 < import_source.sql
rt@machine $ mysql -urt -p cat2 < import_source.sql
rt@machine $ cd $DEEPVAR/generate_lightcurves/utilities/chi_square/
rt@machine $ ruby import_chi_square.rb [host] [user] [pass] [database]
```

Create config file

```
rt@machine $ cd $DEEPVAR/website/cat2
rt@machine $ cp config.php.default config.php
rt@machine $ cp config.php.default deepsearch1.php
Vim to modify
```

```
rt@machine $ cd $DEEPVAR/import/
rt@machine $ cp config.rb.default config.rb
Vim to modify
```

Start daemon for upload command from website

```
#submit run from cat2
* * * * * pgrep -xf "ruby ./command_daemon.rb" > /dev/null || (. ~/.bashrc ; .
/usr/share/Modules/init/bash ; date >> $LOG/daemon_cat2.lo ; cd
/opt/prod/DeepVar/website/cat2/backend/  ; ruby ./command_daemon.rb >> $LOG/daemon_cat2.log
2>&1)

### import spot6 on agilehost1
*/15 * * * * pgrep -xf "ruby ./check_for_import_spot6.rb" > /dev/null || (. ~/.bashrc ; date >>
$LOG/check_import.log ; cd $PIPE/crontab/ ; pwd >> $LOG/check_import.log ; ruby
./check_for_import_spot6.rb >>  $LOG/check_import.log 2>&1)
#
```

5. Science GUI dependencies

dependecies:

1. jquery-3.0.0
2. bootstrap-3.3.7
3. datatables-1.10.12
4. d3-3.5.17

5. celestial-0.6.5
6. plotly-1.38.3

Custom JS9 parameters:

Edit the following file: /var/www/html/js9/js9prefs.js
The js9prefs.js “keys' option values can be found at js9.js or via the display plugin View->Preferences.

6. APPENDIX: installation procedure for old pipelines

The new installation procedures are inside the singularity recipes.

6.1. Installation of AGILE pipelines

6.1.1. AGILE ALERT PIPE

6.1.2. Anaconda2

Install Anaconda 2 https://conda.io/docs/user-guide/install/linux.html

Install Manually:

```
root@machine# cd ..
root@machine# wget https://repo.anaconda.com/archive/Anaconda2-5.3.0-Linux-x86_64.sh
Execute installer
Setup install directory  when program ask for: /opt/prod/anaconda2
root@machine# bash Anaconda2-5.3.0-Linux-x86_64.sh
```

Create module /opt/prod/modulefiles/anaconda-2.7

```
#%Module
set path /opt/prod/anaconda2
prepend-path PATH $path/bin
unsetenv PYTHONPATH
prepend-path LD_LIBRARY_PATH $path/lib
```

Create module /opt/prod/modulefiles/alert_pipe_27

```
#%Module
module load anaconda-2.7
setenv ALERTPIPE /opt/prod/AGILEPIPE/
prepend-path PYTHONPATH $/opt/prod/pipeline_manager:$env(ALERTPIPE)
```

Create environment

```
rt@agilepipe $ module load alert_pipe_27
rt@agilepipe $ conda create -n py_alert_pipe_27 python=2.7.5 numpy=1.12 matplotlib=2.0.2
astropy=2.0.3 scipy=0.18.1 pytest=3.8.0 mysql-connector-python=2.0.4
rt@agilepipe $ source activate  py_alert_pipe_27
rt@agilepipe $  pip install healpy==1.11.0
rt@agilepipe $ pip install pyastronomy==0.12.0
```

Clone the AGILE-ALERT-PIPE repository

6.1.3. MCAL results DB

Create database for MCAL results

```
rt@agilepipe$ cd AGILE-ALERT-PIPE/database
rt@agilepipe $ mysql -uroot -p [pipe db] < init_pipe_database.sql
rt@agilepipe$ mysql -uroot -p --execute='create database rt_alert_results_db;'
rt@agilepipe $ mysql -uroot -p rt_alert_results_db <
AGILE-MCAL-PIPE/database/mcal_results_db_schema_alert.sql
Check if exist mcal_notice_db or create it
rt@agilepipe $ mysql -uroot -p --execute='create database mcal_notice_db'
rt@agilepipe $ cd /opt/prod/AGILE-MCAL-PIPE/database
rt@agilepipe $ mysql -uroot -p mcal_notice_db < mcal_notice_db_schema.sql
If AGILE pipe Check that in database the instrument TEST and network TEST have the id = 0
```

Setup cluster table in database

Check and change the database data if needed to setup all configuration.

Add the configuration to update tstop of EVT, LOG, 3908, 3916, 3913 e 3201 on database

```
rt@agilepipe$ cd /opt/prod/pipeline_manager/scripts/
rt@agilepipe$ cp repository.conf.default repository.conf
rt@agilepipe$ vim  repository.conf
Add this row:  ALERT /opt/prod/AGILEPIPE/update_alert_repo/update_repository.py 1
rt@agilepipe$ cd /opt/prod/AGILEPIPE/update_alert_repo
rt@agilepipe$ cp conf.py.default conf.py
rt@agilepipe$ vim conf.py
```

6.1.4. GCN DAEMON

```
rt@agilepipe$ cd /opt/prod/AGILEPIPE/gcn
rt@agilepipe$ make
rt@agilepipe$ cp conf.txt.default conf.txt
rt@agilepipe$ vim conf.txt
```

6.1.5. AGILE MCAL PIPE 2

Check that this module exists:

```
[rt@agilepipedev ~]$ cat /opt/prod/modulefiles/root_v5.34.24
#%Module

setenv ROOTSYS     "/opt/prod/root/root_v5.34.24_py"

prepend-path PATH                  $env(ROOTSYS)/bin

set libs                    "/usr/lib64:/usr/lib:$env(ROOTSYS)/lib"
prepend-path LD_LIBRARY_PATH    $libs
prepend-path SHLIB_PATH        $libs
prepend-path LIBPATH           $libs
prepend-path PYTHONPATH        $libs

prepend-path MANPATH           $env(ROOTSYS)/man
```

If there is no root _py copy it from another machine, install it and change the module

root@vm # rsync -av rt@[machine]:/opt/prod/root/root_v5.34.24_py .

Install Anaconda 2 https://conda.io/docs/user-guide/install/linux.html

Create /opt/prod/modulefiles/anaconda-2.7

```
#%Module
set path /opt/prod/anaconda2
prepend-path PATH $path/bin
unsetenv PYTHONPATH
prepend-path LD_LIBRARY_PATH $path/lib
```

Create module /opt/prod/modulefiles/mcal_pipe_2.0

```
#%Module
module load anaconda-2.7
```

```
setenv MCALPIPE /opt/prod/AGILE-MCAL-PIPE/pipe2
prepend-path PYTHONPATH $/opt/prod/pipeline_manager:$env(MCALPIPE)
```

Create environment

```
rt@agilepipe $ module load anaconda-2.7
rt@agilepipe $ conda create -n py_mcal_27 python=2.7.5 scipy=0.18.1 numpy=1.12 matplotlib=2.0.2
basemap=1.0.7 lxml=3.2.1 mysql-connector-python=2.0.4 astropy=2.0.3
rt@agilepipe $ source activate  py_mcal_27
rt@agilepipe $ pip install pyastronomy==0.12.0
```

Create pipe database from schema in pipeline_manager repository (check relative manual).

```
rt@agilepipe$ cd pipeline_manager/database
rt@agilepipe $ mysql -uroot -p --execute='create database mcal_pipe_db'
rt@agilepipe $ mysql -uroot -p mcal_pipe_db < pipe-db-schema.sql
rt@agilepipe $ mysql -uroot -p --execute='create database mcal_notice_db'
rt@agilepipe $ mysql -uroot -p mcal_notice_db < mcal_notice_db_schema.sql
rt@agilepipe $ cd AGILE-MCAL-PIPE/database
rt@agilepipe $ mysql -uroot -p mcal_pipe_db < initialize_database_pipe.sql
```

Add this row in pipe_manager database with right path

mcal_pipe localhost [user] [password] mcal_pipe_db_test
/opt/prod/AGILE-MCAL-PIPE/pipe2 /opt/prod/AGILE-MCAL-PIPE/pipe2/skeleton.ll 1

Insert default value for this pipe

```
rt@agilepipe$ cd AGILE-MCAL-PIPE/pipe2/database
rt@agilepipe $ mysql -uroot -p --execute='create database  mcal_results_db;'
For next step check the foreignkey database name
rt@agilepipe $ mysql -uroot -p mcal_results_db < mcal_results_db_schema.sql
```

Check and change the database data if needed to setup all configuration.

Setup cluster table in database

Add the configuration to update tstop of 3908 and 3916 on database

```
rt@agilepipe$ vim /opt/prod/pipeline_manager/scripts/repository.conf
```

```
rt@agilepipe$ cd /opt/prod/AGILE-MCAL-PIPE/pipe
rt@agilepipe$ cp conf.py.default conf.py
rt@agilepipe$ vim conf.py
Add this row to pipeline_manger repository.txt  MCAL
/opt/prod/AGILE-MCAL-PIPE/pipe/update_repository.py 1
```

Setup website

```
rt@agilepipe$ cd /opt/prod/AGILE-MCAL-PIPE/website
rt@agilepipe $ cp config.php.default config.php
Setup config.php
  $base_restults_dir = "/data01/ANALYSIS3.local":
  $web_results_dir = "../analysis/";

root@machine# cd /var/www/html
root@machine# ln -s /opt/prod/AGILE-MCAL-PIPE/website agilemcal
```