MAGIC-TDAS 13-08
130809/MAGIC-ONSITE

MAGIC On-Site analysis program. Update.

A. Lorca
<alejandro.lorca@fis.ucm.es>
K. Satalecka
<satalk@gae.ucm.es>
V. Scapin
<valeria.scapin@gmail.com>
J. L. Contreras
<contrera@gae.ucm.es>
OSA contact
magic-onsite@gae.ucm.es

August 9, 2013

Abstract

This document is an overview of the MAGIC on-site analysis chain (OSA) and an update of the MAGIC TDAS 09-03. Since its installation in 2005, the on-site analysis is an essential part in the observatory's data pipeline, providing the preliminary data output to the analyzers, detecting hardware and software problems and, in many cases, helping to decide on observation strategies. The data is automatically calibrated and processed at the MAGIC site using a combination of services and scripts running on a multi-processor computing cluster. After the upgrade of the MAGIC readout system and MAGIC-I camera in 2012, up to 3TB of raw data are taken during a single night of observation. The OSA chain, along with the computer cluster at the MAGIC site, was upgraded: to cope with the large amount of generated data, to meet the needs of a fast data reduction and to improve the software quality.

Contents

1 Introduction

Due to the size of the daily recorded data, transferring the raw data (even compressed) through the network connection from La Palma island to continental Europe have been an issue . Therefore a fast on-site analysis (OSA) chain, described in this document, have always performed a key role in the MAGIC experiment data pipeline. OSA reduces the raw data at MAGIC site, so that the high level data can be delivered by internet to the experiment data center at PIC (Port d'Informació Científica) [1] with the needed promptness. Raw files are now transferred via network and arrive within few hours, but having reduced data, ready to perform analysis the same day, remains a big ad essential advantage.

The raw files are also taped, as back up solution, and usually arrive to the data center by airmail several weeks after the data is recorded.

The Magic On-line Analysis (MOLA) [2], is commonly used to detect high state activity of the sources and it is very useful to decide on observation strategies. It is, on the other hand, a simplified version of the analysis, not as sensitive as the full analysis used by OSA. The observation strategies for weak sources benefit from the possibility of having a full analysis of the data reduced by OSA.

2 La Palma onsite computing and network

The computing system at the MAGIC site is a cluster of computers linked by internal networks and linked to the outer world through a firewall (`wwwint.magic.iac.es`) [3] and accessible from outside via the public server (`www.magic.iac.es`).

The operations and data processing at the MAGIC observatory are carried out in the susbsystem servers and in the computer cluster respectively. The cluster is composed of a set of HPC servers running the same operating system (Scientific Linux Cern 6.3) and sharing both, network and disk access. The network is divided into two different LANs. The first one, labelled "private", is used as the default communication mechanism between the site machines. The other one, traditionally labelled "GFS2", provides a class C internal network for intensive data transferring and the distributed resource management. Both networks are connected via high speed Gigabit/s ethernet cards, ports and switches. Six 2x quad-core machines are dedicated to perform OSA. Details of the machines are given in Table 1.

The storage capacity of the cluster is provided by four RAID systems and local space for scratch. Each RAID system has different capacity, configuration and purpose. Volumes devoted to DAQ service (raw data storage raid1 and raid2) are partitioned using XFS filesystem and are handled by the DAQ machines. Shared volumes store the raw compressed data, software and user data (raid3 and raid4). The sharing of the storage is done by means of a fibre-channel dedicated network accessing a GFS2-formatted filesystem accessible by every machine of the cluster. OSA has access to 58 TB of storage space, which gives it a buffer of 5-7 days in case of transfer problems and allows to host data that could need special analysis on site due to hardware problems or special needs (i.e. GRB data).

Details of the storage system can be found in Table 2.

A deeper and more comprehensive description of the MAGIC site computing system can be found elsewhere [4].

Host	CPU			RAM	Other
	[model]	[threads]	[HS06/thr.]	[GB]	[services]
`ana0`	Intel Xeon –	8	6.0	2	Copy/Compress, Taping
`ana1`	Intel Xeon –	8	6.3	8	Copy/Compress, Taping
`ana2`	Intel Xeon X3470	8	16.5	4	*torque* WN
`ana4`	Intel Xeon E5405	8	9.6	4	DataCheck, *torque* WN
`ana5`	Intel Xeon E5405	8	10.0	2	Ubuntu test, Back-up
`ana6`	Intel Xeon E5405	8	9.3	4	Luci, Cluster quorum leader, Nagios, MySQL slave server
`ana7`	Intel Xeon E5405	8	9.4	4	*torque* SN, MySQL master, NFS server
`ana8`	Intel Xeon E5405	8	9.4	4	*torque* WN
`ana9`	Intel Xeon E5405	8	9.4	4	*torque* WN
`ana10`	Intel Xeon W3520	8	14.3	12	MOLA, *torque* WN
`grid`	Intel Xeon E5504	4	8.1	4	Transfer

Table 1: Main characteristics of the computers available for the on-site analysis as of June 2013. SN and WN stand for server and worker node respectively.

Volume label	Server	Storage [TB]	Filesystem [type]	Use
raid1_p1	RAID1	8	XFS	M1 raw
raid1_p2	RAID1	4	XFS	M2 raw failover
raid2_p1	RAID2	8	XFS	M2 raw
raid2_p2	RAID2	4	XFS	M1 raw failover
raid3_data	RAID3	39	GFS2	M1, ST and Other data
raid3_opt	RAID3	0.5	GFS2	Software
raid3_home	RAID3	0.5	GFS2	User space
raid4_data	RAID4	19	GFS2	M2 data

Table 2: Space available in RAIDs the on-site analysis as of June 2013.

3 Data flow

Data created at the observatory are processed in a well defined chain. The data flow comprises different "activities" according to the logical actions and subsystems involved. The responsability for each activity relies in a group, which takes care of the full development and deployment of the services and tools required for the correct processing. The data flow of the activities carried out on-site is sketched on Fig. 1.

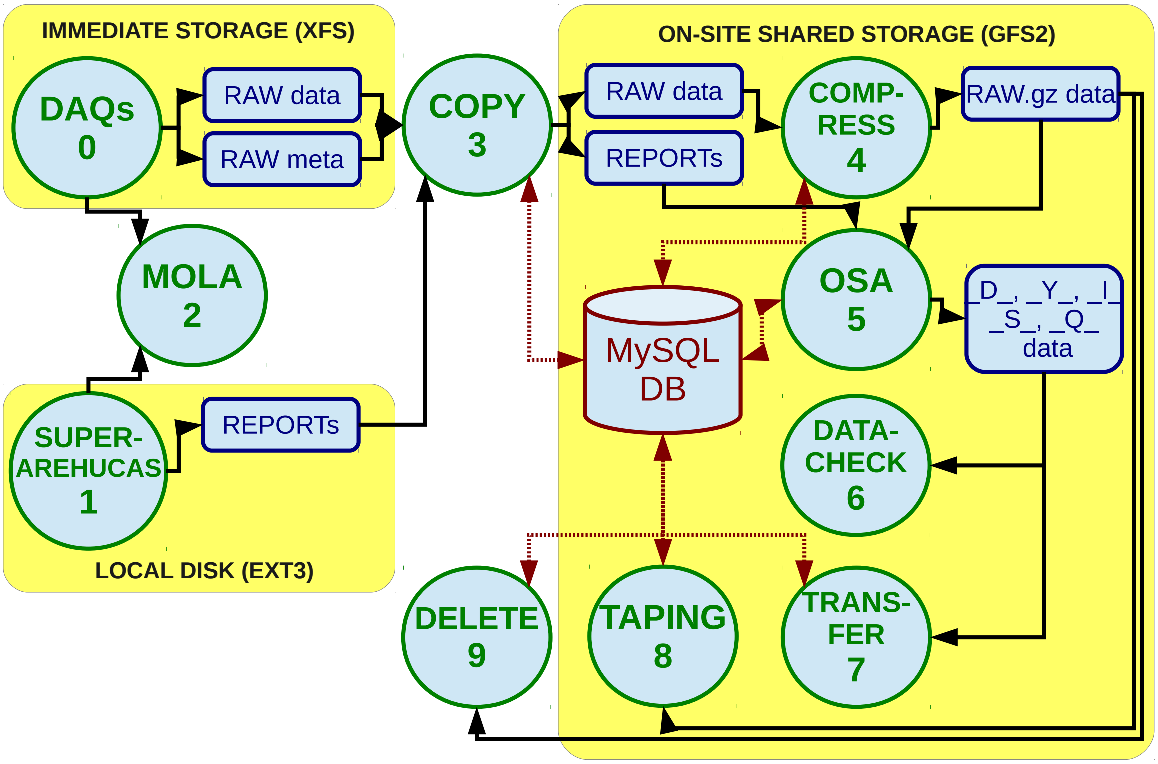


Figure 1: Data flow scheme on-site as of June 2013. The copy and compress activities (3, 4) are done simultaneously, while the others follow approximately the sequential ordering indicated by the number inside the blob.

The OSA team is responsible for the second subset of activities after the immediate data generation: Copy/Compress and OSA itself. The creation of the data either by the DAQs (raw or metadata, being this last data about data) or SuperArehucas (reports), triggers the Copy/Compress activities. The compressed files and reports are registered in the database and placed in a longer-lived shared storage (GFS2 storage in Table 2) . From there on, the OSA can act independently from the data acquisition. This decoupling is important for two reasons:

a) OSA and the subsequent activities start by reading the `raw.gz` files (which are still of 1.9 GB in size and written to the shared storage), thus impacting the I/O performance of the shared-storage but not of the storage provided for DAQ.

b) A decoupling of the storage arrays allows for independent management of the underlying architecture. Major improvements on the shared storage were achieved upgrading from GFS to GFS2. Also the shared filesystem can be checked, replaced or shutdown without affecting data acquisition.

3.1 Copy/Compress

There is a script called `rawcopy.py` running daily as a cron job on `ana0` for M2 and `ana1` for M1. It reads data from both telescopes (raid1 and raid2) and compresses them on the fly using a parallel implementation of the gzip (pigz) and profiting from the multi-core architecture of the machines. When the data is read, a forked process computes the md5sum sumcheck of both, original and compressed files for a later integrity check before deletion. Compressed data is copied to raid3 (M1) and raid4 (M2).

Due to strong limitations on the stability and capabilities of the DAQ machines, all the raw data and metadata written to the XFS partitions is read directly by `ana0` and `ana1` from GFS2. This process requires mounting/unmounting of the XFS raid partitions to update consistently the filesystem information. A similar approach using a NFS client from the DAQ machines turned out to be highly demanding and prune to errors when the DAQs were rebooted often.

If at the time of processing a given raw file and its corresponding reports are available, they get copied and register in the DB as well, so that a full run can be analyzed immediately.

3.2 OSA

There are two scripts, also running as cron jobs from `ana7`, which analyze M1, M2 and ST data. The whole set of utilities is located in the `/opt/OSA/` directory. The usage of a distributed resource manager like *torque* permits an efficient and trackable way to handle the individual jobs for each data sequence. Once the whole night has been correctly processed or when the analyzers manually decide the analysis is over, the OSA activity ends using an specific tool called *closer* and the rest of the activities can continue. In the following sections we will describe the OSA chain in more detail.

3.3 MySQL database

On november 2012, a database was deployed along with the cluster. It was designed to provide a set of useful tables which store information relative to files produced by the observatory, and to keep track of the daily activities. Currently most of the activities use the database for reading and writing, though there are still many operations which require checks on files and the filesystem itself. In addition to the main database `MAGIC`, a development copy of the database `MAGIC_devel` is present in order for the application developers and integrators to test the newly implemented solutions.

The mysql server runs in a master-slave configuration where `ana7` serves as master and `ana6` as slave. It is from the slave from where a security backup is processed daily as a cron job.

All the mysql queries are handled through the internal net for which the server name is `fcana7`. Thus a query from any `ana` machine could be issued from the shell directly as `mysql -h fcana7 -D MAGIC -e "QUERY"`.

It is important to note that regular users (i.e. regular users with access to La Palma machines, not a general analyzer) have only INSERT, UPDATE and SELECT rights to the database. In case that a wrong or corrupted entry has been added, only the MySQL superuser has rights to DELETE. It's important to mention that the MySQL database is meant for internal users on charge of particular activity in La Palma and not for general analyzers.

4 MARS analysis software

The standard analysis software for the MAGIC experiment data is called MARS (Magic Analysis and Reconstruction Software). It is a set of C++ (object-oriented framework) classes based on the well known ROOT package from CERN (`root.cern.ch`). Basically it can be run in two modes, either inside the ROOT framework using the specific containers and tasks designed to analyze the MAGIC data, or as compiled macros (MARS executables). Some of these MARS executables use a Graphical User Interface (GUI).

The on-site analysis uses the following MARS executables:

- *sorcerer*: calibrates the data. The calibrated files have the key-name `*_Y_*.root`. It is able to calibrate directly the "raw" files, including those ones that are compressed using the UNIX *gzip* utility. When compressed, the extension of the raw files is `raw.gz`.
- *merpp*: adds the central control reports to the calibrated files. It can also convert the raw data of MAGIC subsystems to ROOT package format. Merpped files have a key-name `*_D_*.root`.
- *star*: Performs the image cleaning and parametrization of the images. The executable output files have the key-name `*_I_*.root`. They are so called *star files*.
- *superstar*: Combines the M1 and M2 star files into one file. Performs the purely geometrical stereo reconstruction. The executable output files have the key-name `*_S_*.root`.
- *melibea*: Calculates the final event parameters: energy, hadronness and direction using Random Forest (RF) matrices and look up tables (LUTs) (in the case of energy). The executable output files have the key-name `*_Q_*.root`.

Sorcerer, *star*, *superstar* and *melibea* use input cards to define the analysis configuration. *Melibea* requires also matrices and look up tables. In the case of OSA a standard set of pre-trained RF matrices and LUTs is used.

The MAGIC data analysis is based on sequences. A sequence is a group of files marked by a common run number (05XXXXXN if the data has been taken in stereo mode, and 01XXXXXN or 02XXXXXN if taken in mono mode). The run numbers of pedestal and calibration files are 05XXXXX(N-1) and 05XXXXX(N-2) respectively, where 5XXXXXN is the first data run for a given source. Usually (i.e. not for GRB observation, for example) the same pedestal and calibration files are used to calibrate all data runs from one particular source.

OSA uses only the latest, official MARS release version unless a special situation arises, generally related to hardware or input files format changes, which require temporal solutions. The history of MARS versions used by OSA can be found at:
`http://www.magic.iac.es/operations/osa/README_MarsVersion.txt`

5 OSA overview

The aim of the OSA software is to perform the first part of the data reduction (calibration, image cleaning and parametrization) of MAGIC data, allowing a prompt transfer of image files to PIC. The OSA software achieves its objective by a high degree of parallelization of the data reduction and starting to work shortly after[1] the first data files are copied to the shared storage.

[1] Tipically it may take up to an hour, depending on the cron job settings.

If there are no unexpected problems during the analysis process, all star files are available at around 12:00 UTC the morning after the data taking night. These reduced data are available to be downloaded to the MAGIC data center at PIC.

In this section, MAGIC-I and MAGIC-II data processing software will be described. The three first steps of the analysis are essentially the same for both telescopes and the analysis is done in parallel, independently for each telescope. The stereo analysis starts as soon as the mono star files are available.

5.1 Description of the main programs

The OSA chain is run by the analysis computers described in section 2, and belongs to the user *analysis*. The programs are located at the directory `$OSAHOME=/opt/OSA` in the `ana` machines. The sub-directory `./python` contains the python scripts and the configuration files. The configuration details are located in `./python/cfg`. Compiled python scripts (commands) can be found in the `./bin` sub-directory. Other macros and scripts are stored in `./macros` and `./perl`. A rough description of the `$OSAHOME` sub-directories can be found in Table 3.

Subdirectory inside `/opt/OSA`	Content
`bin/`	Binary files for convenience (default-configured python scripts)
`macros/`	Mars self created macros
`Mars_extras/`	
`- inputcards_OSA/`	Configured input cards
`- Melibea_matrices/`	Matrices for Melibea
`perl/`	Perl scripts
`python/`	Python scripts
`- cfg/`	Configuration files
`utils/`	Utilities for the web
`- activities/`	Summary of activities today
`- closer_incidences/`	OSA Incidences book
`- cpu_status/`	CPU Status
`- disk_space/`	Disk Space
`- osa_sequencer/`	OSA Sequencer today
`- style_css/`	OSA css style file for web
`- torque_status/`	Torque Job Status

Table 3: Scheme of the directories relevant for the on-site analysis programs.

For every day, from which data are going to be analysed, OSA builds dedicated directories (YYYY/MM/DD) under predefined locations, relevant for the analysis and enumerated in Table 4. Let us describe them briefly. During the execution of `sequencer` the data and other required files are stored in the "Analysis" directory. Correctly processed sequences are copied to the "Calibration", "Star", "SuperStar" and "Melibea" directories, where they await to be tranferred to PIC. The transfer takes place when all sequences are processed succesfully and the day is "closed". Later, when all the data is transferred, the deletion of these directories and their content occurs.

The automatic operation of OSA is managed through the Linux "cron daemon" at `ana7` using the crontab utility for user *analysis* defined at `ana7` machine. The "cron daemon" launches the following

Subdirectory	Content
`/data/{M1,M2,ST}/OSA/Analysis`	Working directory for sequencer
`/data/{M1,M2}/OSA/Calibrated`	Calibrated files
`/data/{M1,M2}/OSA/CC`	Copy of the CC reports
`/data/{M1,M2,ST}/OSA/Closer`	Closer flag files
`/data/{M1,M2,ST}/OSA/devel`	Directory with replicated structure for development
`/data/{M1,M2,ST}/OSA/log`	Log for sequencer and other scripts
`/data/{M1,M2}/OSA/Star`	Starred files
`/data/ST/OSA/SuperStar`	Superstarred files
`/data/ST/OSA/Melibea`	Files after running melibea

Table 4: Scheme of the directories relevant for the on-site analysis data.

script responsible for starting the different tasks of OSA:

- `sequencer`: this python script takes cares of performing a check of the raw data and produces the list of data sequences to be analyzed, establishing relationships between them according to the Pedestal-Calibration procedure. It sends `calibrationsequence.py`, `littlesequence.py` and `stereosequences.py` processes to the *torque* queue system. The "cron daemon" only executes `sequencer` to process the data taken the night before, and it works hourly analyzing the corresponding night according to the current day and the offset defined in the configuration file. The combinemuons macro is run at the end of each run of sequencer for single telescope.

Other auxiliary scripts and macros are:

- `closer`: sets a "finished" flag on a given day (for each telescope separately) and signals that the analyzed data are ready for transfer to PIC.
- `combinemuons.cpp` macro: runs over star*.root files, combines the histograms used to obtain muon arc width and sizes, which are used to monitor telescope performance (size scale and PSF).
- `monolith`: checks if there is remaining data from previous days to be processed and allows to start sequencer in case we want to analyze the data.
- `suspend-osa, resume-osa` allow to immediately halt/resume all the on-site analysis processes in all the analysis machines. With the jobs suspended, there is no need to stop the cron daemon. These scripts can be called by the operators in case it is needed to stop the on-site analysis program. DAQ and OSA are decoupled, and should not interfere with each other, so this is just an emergency solution, if some unexpected problem occurs.
- `utils` *web scripts*: launched by the "cron daemon" every 10 minutes in `ana1`, perform auxiliary tasks useful for the on-site analysis administrators. They copy relevant information to the webpage `http://www.magic.iac.es/operations/osa`, so the status of the on going processes can be tracked without the need of a *ssh* connection to La Palma computing system.

A schematic view of the OSA program components is shown in Fig. 2.

All the OSA program code can be found at the protected subversion repository:
`https://duperier.gae.ucm.es/OSA`. The current version is tagged as 2.0.0.

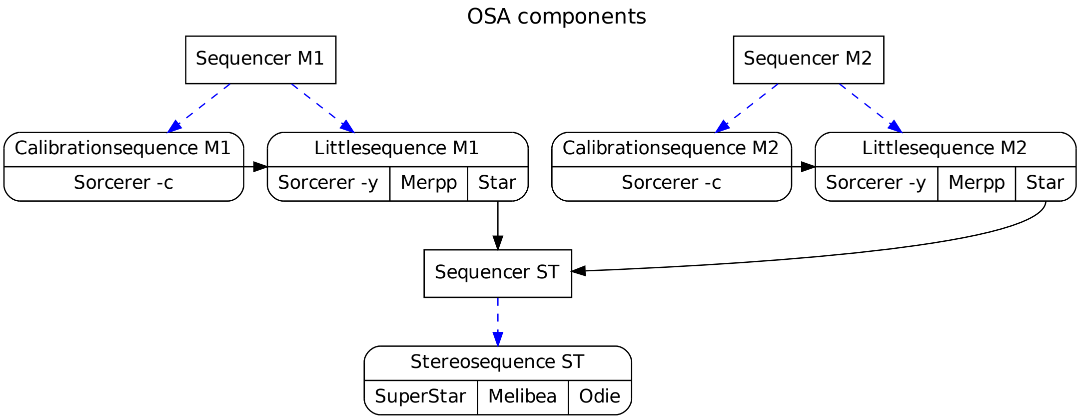


Figure 2: On-site analysis program blocks with the relevant Mars modules called. Squared boxes indicate programs run in the user interface in `ana7`, while rounded boxes are jobs executed anywhere in the other `anaX` working nodes. Solid lines indicate dependencies while dashed lines stand for job submissions.

5.2 *Torque* queue system

The jobs submitted by `sequencer` go to the batch system called *`torque`*. The system takes care of a proper distribution of the jobs to the available CPUs following the Calibration− >Data dependency pattern (see Fig.3). Each analysis machine has a maximum number of processes that can be run simultaneously. This number depends on the characteristics of each machine, and also on the disk access capacity – it is important to avoid any possible interferences with the DAQs of both telescopes during data taking.

The analysis user does not need to issue the *torque* commands by hand, but it is interesting to know them. The main utilities are:

- `qsub script_name` (to submit a job script by hand)
- `qstat` (to see the queued jobs)
- `qdel -p job_number` (to kill a job and remove it from the queue)
- `clean_hold_jobs.sh` (to get rid of old jobs on hold)

The full documentation of the *torque* queue system can be find at:
`http://docs.adaptivecomputing.com/torque/2-5-12/help.htm`.

5.3 Sequencer tasks

The task of `sequencer` is to create the list of sequences to be analyzed, establish relationships between them and send them to the *torque* queue system. By the means of the `.veto` files (see below) this script also controls the correct execution of the analysis.

There are three types of sequences according to the content of the input files.

- Calibration. Files of the form `*_C_*` which have to be preceeded by pedestals `*_P_*` files.
- Data. Files of the form `*_D_*` which are preceeded by calibration sequences.
- Stereo. Files which have two different sequence parents coming from data sequences of M1 and M2 sharing the same run number.

The calibration sequences are analyzed first, in order to produce the `scalib<run>_<tel_id>.root` files, which contain the calibration parameters for the data. The analysis of the data runs depends on the existance of the appropriate `scalib<run>_<tel_id>.root` file.

In an unusual situation of multiple `*_P_*` and `*_C_*` files taken for the same data runs (e.g. 2 `*_P_*` runs, or 2 `*_C_*` runs, or `*_P_* *_C_* *_P_* *_C_*`,...., `*_D_*` , `*_D_*` ,....) the last couple of pedestal and calibration file is chosen by the program. If this combination is not valid, the OSA team has to manually select the best one.

The `sequencer` script is normally launched by the "cron daemon" as said before. It can also be manually executed to start the on-site analysis for any datataking night. If no argument is given, it process the whole input in blocks by telescopes, first M1, then M2 and finally the stereo mode ST. Also an argument can be provided with the `<tel_id>` (M1, M2 or ST) and the previous night's data analysis is launched for the chosen telescope. If one specific night is given as an option following the flag '`-d`', the program will be launched for the selected night. As an example, the command: `sequencer -d 2012_04_01 M1` will launch OSA for M1 data from 2012 April 1st.

One can also use sequencer to monitor the progress of data processing, using the option `-s` (simulate). The command:
`sequencer -s M2` will show the status of the M2 analysis for the previous night (example in table 5). The last three columns show the percentage of files processed at each level (calibration, merpp, star). The "Action" column shows: "Check", if the job finished successfully, "Simulate", if the job should be run again (error exit in a previous try), "Veto", if the sequence finished with the same error for the third time. The "State" column shows: C in case of a completed job, R if the job is still running, H if the job is on hold (waiting in the queue for a free node or for a parent sequence to be ready).

This can be also monitored on the OSA Sequencer web page:
`http://www.magic.iac.es/operations/osa/osa_sequencer.xhtml`

For more `sequencer` options type: `sequencer -h`

The script `sequencer` initially calls a *perl* script called `nightsummary.pl` located in `$OSAHOME/perl`. This script lists the raw files from the raw directory of each telecope
(`/data/<tel_id>/DAQ/RAW/<YYYY_MM_DD>`), and creates a list of the runs to be analyzed. The output of this script written in to a file called: `NightSummary_<YYYYMMDD>_<tel_id>.txt`, stored in the analysis directory. For each subrun, the file contains a line with the following information:

`run_number subrun_number run_type start_date start_time project_name number_events calibration_script zd_deg source_ra source_dec L2_table test_run`

```
[analysis@ana7 ~]$ sequencer -s M2
============================ Starting sequencer.py at 2013-04-02 16:08:44 UTC for MAGIC, Telescope: M2, Night: 2013_04_02 ==============================
Files for 2013_04_02 M2 are completely transferred to raid
-----------------------------------------------------------------------------------------------------------------------------------------------------------
Tel Seq  Parent  Type         Run      Subruns  Source       Wobble     Action   Tries  JobID  State  Host       CPU_time  Walltime  Exit  _Y_\%  _D_\%  _I_\%
M2    0  None    CALIBRATION  5024701        1  1ES0927+500  0.40+000   Check        1  11879  C      fcana2/0   00:00:23  00:00:29     0  None  None  None
M2    1       0  DATA         5024702       19  1ES0927+500  0.40+000   Check        3  12053  C      fcana2/6   01:05:49  02:04:18     0   100   100   100
M2    2  None    CALIBRATION  5024704        1  1ES0927+500  0.40+000   Check        1  11888  C      fcana2/2   00:00:27  00:00:35     0  None  None  None
M2    3       2  DATA         5024705       28  1ES0927+500  0.40+000   Check        1  11889  C      fcana2/3   01:21:54  01:27:53     0   100   100   100
M2    4       2  DATA         5024706       28  1ES0927+500  0.40+180   Check        1  11890  C      fcana2/4   01:21:37  01:27:53     0   100   100   100
M2    5       2  DATA         5024707       20  1ES0927+500  0.40+090   Check        3  12054  C      fcana9/0   01:08:21  02:15:39     0   100   100   100
M2    6       2  DATA         5024708       27  1ES0927+500  0.40+090   Check        1  11899  C      fcana2/5   01:21:59  01:31:22     0   100   100   100
M2    7       2  DATA         5024709       26  1ES0927+500  0.40+270   Check        1  11900  C      fcana2/6   01:20:14  01:30:37     0   100   100   100
M2    8       2  DATA         5024710       15  1ES0927+500  0.40+000   Check        1  11901  C      fcana10/0  00:42:24  00:47:49     0   100   100   100
M2    9  None    CALIBRATION  5024712        1  Mrk421       0.40+270   Check        1  11909  C      fcana2/2   00:00:34  00:00:38     0  None  None  None
M2   10       9  DATA         5024713       20  Mrk421       0.40+270   Check        1  11910  C      fcana2/0   00:58:48  01:09:17     0   100   100   100
M2   11       9  DATA         5024714       20  Mrk421       0.40+000   Check        1  11911  C      fcana2/4   00:57:58  01:08:58     0   100   100   100
M2   12       9  DATA         5024715        8  Mrk421       0.40+180   Check        1  11912  C      fcana10/0  00:22:18  00:31:44     0   100   100   100
M2   13  None    CALIBRATION  5024717        1  1ES0927+500  0.40+180   Check        1  11920  C      fcana2/2   00:00:31  00:00:44     0  None  None  None
M2   14      13  DATA         5024718       27  1ES0927+500  0.40+180   Check        1  11921  C      fcana2/2   00:44:14  00:45:38     0   100   100   100
M2   15      13  DATA         5024719       27  1ES0927+500  0.40+090   Check        1  11922  C      fcana2/3   00:44:08  00:45:38     0   100   100   100
M2   16      13  DATA         5024720       27  1ES0927+500  0.40+270   Check        6  11950  C      fcana2/2   01:23:32  03:08:05     0   100   100   100
M2   17      13  DATA         5024721       25  1ES0927+500  0.40+000   Check        6  11951  C      fcana2/3   01:17:53  02:58:01     0   100   100   100
M2   18      13  DATA         5024722       20  1ES0927+500  0.40+000   Check        6  11952  C      fcana2/4   01:04:55  02:29:21     0   100   100   100
M2   19  None    CALIBRATION  5024724        1  M87          0.40+000   Check        1  11966  C      fcana2/6   00:00:37  00:00:44     0  None  None  None
M2   20      19  DATA         5024725       22  M87          0.40+000   Check        1  11967  C      fcana10/3  01:23:17  01:59:11     0   100   100   100
M2   21      19  DATA         5024726       23  M87          0.40+180   Check        1  11968  C      fcana10/4  01:28:49  02:04:22     0   100   100   100
M2   22      19  DATA         5024727       23  M87          0.40+090   Check        1  11987  C      fcana2/2   01:38:35  02:27:10     0   100   100   100
M2   23      19  DATA         5024728       22  M87          0.40+270   Check        1  11988  C      fcana2/6   01:40:00  02:28:42     0   100   100   100
M2   24      19  DATA         5024729       20  M87          0.40+000   Check        1  11989  C      fcana10/2  01:27:00  02:19:25     0   100   100   100
M2   25      19  DATA         5024730       19  M87          0.40+180   Check        1  11990  C      fcana10/5  01:26:05  02:19:25     0   100   100   100
M2   26      19  DATA         5024731       18  M87          0.40+090   Check        4  12093  R      fcana2/6   02:13:39  02:18:49  None   100   100   100
M2   27      19  DATA         5024732       17  M87          0.40+270   Check        3  12064  R      fcana9/3   02:24:36  03:30:04  None   100   100   100
M2   28      19  DATA         5024733        6  M87          0.40+000   Veto         4  12065  C      fcana9/4   01:17:08  01:39:24    34   100   100   100
M2   29  None    CALIBRATION  5024735        1  DarkPatch30  None       Check        1  12008  C      fcana9/2   00:00:40  00:00:53     0  None  None  None
M2   30      29  DATA         5024736       16  DarkPatch30  None       Simulate     3    -31  C      fcana9/5   02:39:12  03:44:13    34   100   100   100
M2   31      29  DATA         5024737        3  DarkPatch30  None       Veto         4  12067  C      fcana9/6   00:39:08  00:50:20    34   100   100   100
M2   32      29  DATA         5024738       15  DarkPatch30  None       Simulate     3    -33  C      fcana8/0   02:39:53  03:40:58    34   100   100   100
M2   33      29  DATA         5024739       15  DarkPatch30  None       Simulate     3    -34  C      fcana8/1   02:39:09  03:40:11    34   100   100   100
M2   34      29  DATA         5024740       12  DarkPatch30  None       Check        3  12094  R      fcana10/0  02:05:11  02:09:48  None   100   100   100
M2   35      29  DATA         5024741       12  DarkPatch30  None       Check        1  12071  R      fcana8/3   00:47:45  01:05:31  None   100   100    50
```

Table 5: Typical execution of the `sequencer` script as monitor tool.

Any run which contains a pattern different from `*_P_*`, `*_C_*` or `*_D_*` in the file name will be ignored by OSA: these are technical runs requested by experts that are not intended to be analyzed in the standard way.

Reading the information summarized in NightSummary_<YYYYMMDD>_<tel_id>.txt, sequencer will create a logical workflow of jobs. The workflow includes some dependencies allowing the job scheduler to know which jobs are to be executed first (parent jobs) and which other jobs depend on their output (child jobs). A vector graphic file called Workflow_<YYYYDDMM>_<ted_id>.svg is generated helping to visualize the structure of the sequences for that night and telescope. Examples of these graphs are shown in Fig. 3. The NightSummary_<YYYYMMDD>_<tel_id>.txt is updated every time sequencer is run and contains all but most recent data run. This prevents OSA from analyzing partially copied sequences. The last data run is included only when the rawcopy activity is completed. In case of problems one can force the program to use an existing NightSummary_<YYYYMMDD>_<tel_id>.txt by running sequencer -n.

The application further generates a set of ASCII text files. All files have a common name convention: sequence_<tel_id>_<run>*, and they are distinguished by their extension:

- `.txt`: This file is the sequence itself, in the standard MARS format, a text file containing a list of all data subruns in the sequence and the relevant pedestal and calibration runs.
- `.sh`: a shell script containing the command line to run *littlesequencer.py* on the specified sequence. This script will be send to the *torque* batch system.
- `.history`: a text file with a brief summary of the processes run over the sequence. The file con-

tains the following information: <sequence> <process> <day> <time> <input calibration constant file> <input card> <error code>. The <process> field can be: *sorcerer*, *merpp* or *star*. For *star* processes the <input calibration constant file> field is empty, as well as, for *merpp* processes, the <input card> field. This file allows to control how many times and with what configuration input card each process has been launched, enabling the creation of veto files, if required.

- .veto: This file is empty. It is created, if a problem than cannot be solved with the available input card arises during processing of a sequence. Only three trials (with each adequate input card) are allowed for each *sorcerer* and *star* process before the veto is created. This way the number of consecutive times that a failed process can be again run is limited. A sequence with an existing veto file will never be processed by the OSA automatic scripts.

Apart from the above mentioned files the *torque* batch system creates for each sequence two ASCII files: sequence_<tel_id>_<run>_<job_id>.anaX.magic.iac.es

- .out contains output of the analysis chain sorcerer, merpp and star;
- .err contains errors and warnings from the analysis chain;
- .log contains information about the python execution (both error and output).

Before any further work, the script also checks if the sequence has already finished the *sorcerer*, *merpp* and *star* processes. This is achieved by looking into the different .history files and checking the level achieved and the exit status. A completed block of calibrationsequence.py, littlesequence.py or stereosequence.py is checked as well through the job exit status from the qstat command of *torque*.

5.4 *Calibrationsequence* tasks

For all the sequences after a pedestal and calibration pair, the best degree of parallelization is obtained if the calibration constants from the pedestal-calibration pair can be shared as starting values for the data runs. These calibration parameters are stored at the scalib<run>_<tel_id>.root file, obtained from a *sorcerer* execution with the option '-c'. The script accomplishing this task is called calibrationsequence.py. It takes *_C_* and *_P_* files as input, executes the sorcerer -c and gives the scalib<run>_<tel_id>.root file as output.

5.5 *Littlesequence* tasks

The purpose of the script littlesequence.py is to execute the first analysis steps on a sequence, unless a veto file is found. It is normally called by sequencer for every sequence of the data taking night. It can also be manually executed by specifying the following parameters:
<scalib root file to use> <run> <tel_id>
An example on how to run manually littlesequence.py is:

```
/opt/OSA/python/littlesequence.py -c /opt/OSA/python/osa.cfg -d 2013_01_09
/data/M1/OSA/Analysis/2013/01/09/scalib05022257_M1.root 05022260 M1
```

Below we give more details on the specific tasks of littlesequence.py: calibration, joining of the data and relevant reports, image cleaning and image parameters calculation. Those tasks are performed by calling the appropriate MARS executables in the appropriate order.

5.5.1 Calibration: sorcerer

The goal of the calibration is to obtain physical quantities from the digitized information. This is done through the process which determines the conversion factor from digitized readout counts to incident photons, and the arrival time delay for each pixel. Detailed information can be found in [9].

The script `littlesequence.py` will first try to perform the calibration of the sequence using the default configuration with the input card sorcerer_<tel_id>_DRS4.rc, which can be found at MAGIC CVS at PIC, according the the MARS version used at the moment.

Note that for every run the first 20-40 seconds are calibrated using the pedestal RMS and conversion factors from the `*_P_*` and `*_C_*` runs (which can be taken up to some hours before), but then the interleaved calibration kicks in and most of the data are calibrated using the interleaved events.

In case the calibration fails, sorcerer returns an error code through the *exit status* variable. Most of the errors are well known and there are dedicated input cards to allow the calibration of the data avoiding the error (see the sorcerer wiki page:
`http://magic.pic.es/priv/wiki/index.php/MAGIC_software:Sorcerer`).

Nevertheless, the detected problem, which has commonly a hardware problem origin, remains in the calibrated files. These input cards allow to have the data calibrated, even if the quality of the recorded data is not optimal.

5.5.2 Merging control reports and data: merpp

After the calibration is performed, the central control reports have to be added to the calibrated data. These reports contain, among all the subsystems information, the drive and starguider systems reports. Having this information added, the telescope mispointing can be corrected. The subsystem report information is added to the calibrated files by the MARS executable *merpp* using the '`-u`' option.

Please note that OSA do the merpping after the calibration, in order to optimize the CPU. This means that the reports are directly attached to `*_Y_*` files, i.e. the calibarted and calibrated+merpped files would have the same name. In order to keep track of the merpping process, for each merpped file OSA create `*_D_*`.root file which is just a link to a real `*_Y_*`.root file.

The most common merpp error is 255 when a SuperAreucas report is missing. In this case, the most important starguider and drive Cosy reports have to be copied from pc4 (M1) or pc14 (M2) and added to the data files. This solution is not yet automatic.

5.5.3 Image cleaning and parametrization: star

The next step of the data reduction is to clean the calibrated shower images from pixels which contain only night sky background noise. After the images are cleaned, a parametrization of the images is performed. Both image cleaning and image parameter calculation are done by the MARS executable *star*. The image parametrization will be needed at the final steps of the analysis for the signal/background discrimination and for the energy estimation.

The script `littlesequence.py` runs *star* for the sequence with the standard configuration obtained from input cards star_<tel_id>_OSA.rc. The image is cleaned according to the number of phe (photoelectrons) of each pixel and its neighbours, and also according to the arrival times of the pulses at the pixels [10]. The used configuration is the MARS default one. For more details consult the star input cards.

After the image cleaning the image parameters, called Hillas parameters [11] are computed. They are basically statistical moments up to the third order of the cleaned light distribution on the camera in phe units. Further description of the image parameters can be found in references [12] and [13].

The executable *star* also uses the information from the starguider subsystem to correct the data from the possible telescope mispointing, when this information is correct. It happens often that for twilight or moon observations the obtained information from *starguider* is not adequate for the mispointing correction. If this happens, *star* gives the output error code “6” and *littlesequence.py* tries again to run *star* with the input card `star_<tel_id>_nostgcal.rc`, which will create output files without mispointing correction.

5.6 Stereosequencer tasks

The `stereosequencer.py` script takes care of the stereo part of the analysis. It is executed, if star processes for the same sequence from both: M1 and M2 data finished with an exit 0.

5.6.1 Superstar

The task of *superstar* is to join the information from M1 and M2 star files and perform a purely geometrical stereo reconstruction of the events. As an input it takes the M1 and M2 star files. As an output we get one `*_S_*` file for each sequence. Superstar needs an input card called: `superstar.rc`.

5.6.2 Melibea

Melibea takeas care of the energy and direction reconstruction as well as the calculation of the Hadronness parameter used for gamma/hadron separation. MAGIC uses the Random Forest (RF) method [19] to separate gamma- and hadron-like events and to reconstruct the direction of the event. This method requires so-called RF matrices, i.e. a set of decision trees which contain the reconstruction/separation pattern. The energy reconstruction is done by the means of Look-Up Tables (LUTs) [20]. OSA uses the pre-calculted, standard RF matrices and LUTs. The excution of melibea also requires an input card `melibea.rc`. As an output we get one `*_Q_*` file for each sequence.

5.7 File copy and verification: `closer`

After the *star* or *melibea/odie* processes has finished succesfully the data, calibration coefficient and status display files are copied automatically to:

- `/data/<tel_id>/OSA/Calibrated/<YYYY>/<MM>/<DD>`,
- `/data/<tel_id>/OSA/Star/<YYYY>/<MM>/<DD>`.
- `/data/<tel_id>/OSA/SuperStar/<YYYY>/<MM>/<DD>`,
- `/data/<tel_id>/OSA/Melibea/<YYYY>/<MM>/<DD>`,

For all of them, an entry in the `STORAGE` table in the database is also inserted or updated. This is done sequence by sequence and can be used to transfer the “preliminary” data to PIC.

Once all of the sequences have been completely processed one have to set a “finished” flag on the data analysis for a given day and telescope. This is done by the `closer` script, that, up to now, have to be run manually from OSA users. The script checks that all sequences finished with exit 0 and re-register them in the data base, while moving the calibrated, star, superstar and melibea files to the directories, independently of their exit status (the processing errors are listed in the Incidences - see below). Also the `scalib*.root`, `ssignal*.root`, `star*.root`, `superstar*.root` and `melibea*.root` files are moved to the final Calibrated, Star, SuperStar and Melibea directories.

The day-telescope combination which has been closed cannot be analyzed again. In exceptional case, if the data has to be re-analyzed, one has to "open" the day again by removing manually the "finished" flag and the corresponding entry in the DB. This is only possible as long as the raw files are still on-site.

5.8 Common errors

The most common error codes are:

2 (*sorcerer*): Interleaved calibration is not ok. OSA will exclude short (1 sub run) sequences presenting this problem. For longer sequences the OSA team will try to calibrate manually the data without interleaved calibration.

235 (*sorcerer*): When there are corrupted events. OSA team tries to calibrate the data manually with -ff option, skipping the corrupted events.

255 (*merpp*): Report file missing for at least one sub run. OSA team recovers the DRIVE reports and merpp the data manually.

6 (*star*): Starguider calibration error. OSA will automatically try to run star without starguider calibration.

7 (*star*): Not fatal, data analysis continues. Tipically appears when the exclusion of starguider calibration is not enough and there is a problem in the cleaning due to too much light shining in the camera (data taken when moon is present).

34 (*star*): Not fatal, data analysis continues. Problem with Image Cleaning levels used by OSA. More than 20 precent of the events will contain spurious islands only due to noise. Data has to be cleaned with higher cleaning thresholds.

Other possible errors may happen during the calibration process[2], and they can be printed by using the MARS executable *ape* (Analysis Problems & Errors) :
`ape -human *_I_*.root`

At the closing of the day/telescope the summary of the encountered problems and their solutions is published in the `Incidences.txt` file at the MAGIC webpage at La Palma:
`http://www.magic.iac.es/operations/osa/closer_incidences.xhtml`. We strongly encourage the analyzers to consult this archive before starting the data quality selection and analysis.

The *sorcerer* and *star* files are copied from MAGIC to the PIC database as soon as the day is closed. Also the raw files are copied to tapes and deleted from the RAID.

6 Conclusion

The software presented here performs a fast standard data reduction that allows the calibrated and *star* files to be transferred quickly to the MAGIC experiment data center at PIC. To achieve this objective, the OSA software uses the computer cluster at MAGIC site at La Palma to optimize, using a high degree of parallelization at sequence level, the processing of the raw data files.

[2]Their meaning can be found in the MARS class MAnalysisProblem.cc `http://magic.pic.es/priv/cvs`.

Major hardware updates like the installation of the Domino4 readout system and new MAGIC-I camera have increased dramatically the volume of the raw data stored and processed every night. Therefore an update of the computer system in La Palma and the OSA software was needed. The actual version of the on-site analysis is a result of an evolution of the original code [14] [16] [17] [18] that, together with the update of the needed computing and storage capacity, has successfully coped with the mentioned hardware changes.

7 Acknowledgments

We are grateful to all the people who have helped us to develop and to maintain the OSA software. We would like to thank in particular Julian Sitarek and Emiliano Carmona for their essential and always fast help on solving the problems we have faced during these years.

8 Frequently Asked Questions

1. **How do I know, if my data from last night was processed?**
   If you want to know at which stage of processing the data from today are, have a look here:
   `http://www.magic.iac.es/operations/osa/activities.xhtml`
   In order to have a sequence-by-sequence information of OSA processing look here:
   `http://www.magic.iac.es/operations/osa/osa_sequencer.xhtml`

2. **Where are my data?**
   OSA processed data are transfered to PIC (usually on the same day). You can search for them here:
   `http://data.magic.pic.es/Data/`
   OSA processed data are v1. Due to hardware/software problems the data could have been re-processed at PIC. Please check the correct version of your data on this wiki page:
   `http://magic.pic.es/priv/wiki/index.php/Data_Center:Storage:Recommended_Versions`

3. **My data are not at PIC, where are they?**
   First, consult the Runnbook - maybe there was no data taken... Second, if the data are from today, have a look here:
   `http://www.magic.iac.es/operations/osa/activities.xhtml`
   and here:
   `http://www.magic.iac.es/operations/osa/osa_sequencer.xhtml`
   to check if the processing was finished. Third, if the processing is finished, or you ask about past data, have a look into the Incidences book to check, if there were any problems with data analysis (e.q. Test data, GRB alert, major hardware problem...) and the data was excluded:
   `http://www.magic.iac.es/operations/osa/closer_incidences.xhtml`
   Last, contact us at: `magic-onsite@gae.ucm.es` or PIC support at: `support@magic.pic.es`. Some exceptional hardware/software/transfer problem might have occurred...

4. **Which MARS version was used to process my data?**
   MARS versions used by OSA are listed here:
   `http://www.magic.iac.es/operations/osa/README_MarsVersion.txt`
   Remember that for data process with Mars V2-5-0 (2011-09-26) or higher you can always use inforc: inforc –mars files.

5. **Which input cards were used to process my data?**
   OSA uses the standard input cards corresponding to the MARS version with which your data were processed. In case of wrong starguider calibration OSA will use automatically a standard OSA input card with MJStar.CalibStarguider: no and MJStar.UseStarguider: no. If non-standard input cards were used it is listed in the Incidences:
   `http://www.magic.iac.es/operations/osa/closer_incidences.xhtml`
   In any case for data process with Mars V2-5-0 (2011-09-26) or higher you can always use inforc.

6. **What should a shifter know about OSA?**
   In principle nothing :). Tha data aquisition and OSA are decoupled and should not interfere with each other. The OSA jobs are running in the anaX machines:
   `http://www.magic.iac.es/operations/osa/cpu_status.xhtml` If you see problems with one of them, please contact .... You might be asked to restart one of the computers or suspend/resume OSA jobs using the scripts described on page 9.

References

[1] Reichardt et al., 2009, *The MAGIC data center*, 31st ICRC, Lódź, Poland.

[2] Tescaro, D., Lopez Oramas, A., Moralejo, A. , Mazin, D. *The MAGIC telescopes DAQ software and the on-the-fly online analysis client*, 33rd ICRC, Rio de Janeiro, Brasil.

[3] Coarasa J.A., 2005, *Online PC Farms for MAGIC-II and Data Transfer*, MAGIC Collaboration Meeting, Tenerife.

[4] Carmona E. et al., 2009, *A Flexible High Demand Storage System for MAGIC-I and MAGIC-II using GFS*, 31st ICRC, Lódź, Poland.

[5] `http://magic.pic.es` (MAGIC private URL).

[6] `http://magic.pic.es/priv/cvs` (MAGIC private URL).

[7] Cortina, J., Oya, I. 2006, *DailyCheck Manual* , MAGIC internal documentation, TDAS 08-02.

[8] *MAGIC Wiki* `http://magic.pic.es/priv/wiki/index.php/Main_Page`.

[9] Gaug M., PhD Thesis, Universitat Autonoma de Barcelona, March 2006. `http://magic.mppmu.mpg.de/publications/theses/index.html`.

[10] E. Aliu et al., *Improving the performance of the single-dish Cherenkov telescope MAGIC through the use of signal timing* , Astropart. Phys. 30 (2009) 293.

[11] Hillas A.M., 1985, *Cherenkov Images of EAS produced by primary gamma rays and by nuclei*, Proc. 19th ICRC (La Jolla), 3, 445.

[12] Gaug M., Moralejo, A. , *A handbook of the standard MAGIC analysis chain*, MAGIC internal documentation, TDAS 08-02.

[13] Oya I., et al, 2009, *MAGIC Data Check program*, MAGIC internal documentation, MAGIC-TDAS 09-02.

[14] De los Reyes, R., Oya, I., *MAGIC data check and on-site analysis program* MAGIC internal documentation, TDAS 06-11.

[15] J. Albert et al., *FADC Signal reconstruction for the MAGIC telescope* J. Albert et al., Nucl. Instr. Meth. A 594 (2008) 407.

[16] De los Reyes, R. , PhD Thesis, Universidad Complutense de Madrid, July 2008. `http://magic.mppmu.mpg.de/publications/theses/index.html`.

[17] Oya, I., Diploma thesis, UCM, September 2006.

[18] Oya, I. et al, *MAGIC On-Site analysis program*, MAGIC internal documentation, TDAS 09-03.

[19] J. Albert et al., Nucl. Instr. Meth. A 588, 424 (2008), *Implementation of the Random Forest Method for the Imaging Atmospheric Cherenkov Telescope MAGIC* Performance of the MAGIC stereo system obtained with Crab Nebula data

[20] J. Aleksi et al., Astropart. Phys. 35, 435 (2012), *Performance of the MAGIC stereo system obtained with Crab Nebula data*

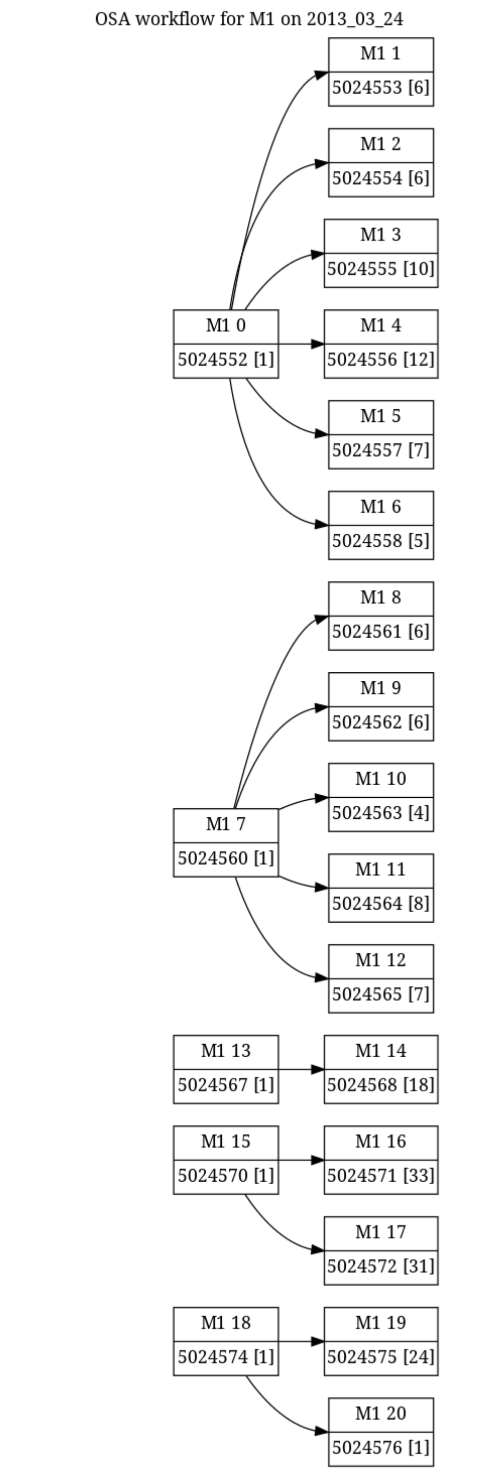
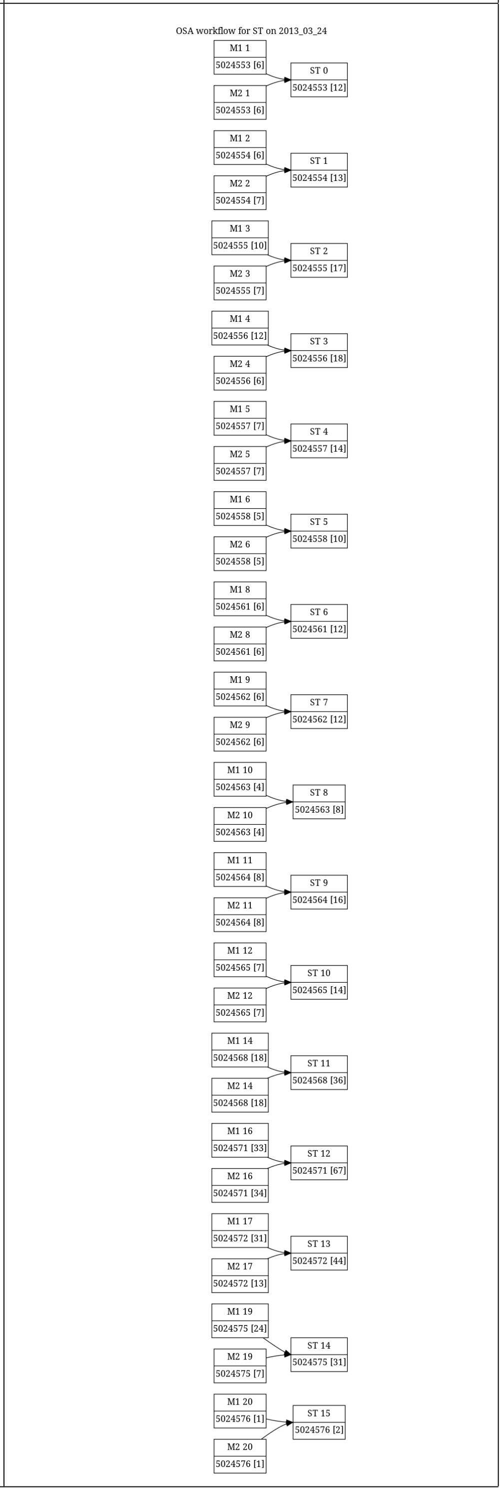
OSA workflow for M1 on 2013-03-24	OSA workflow for ST on 2013-03-24
	

Figure 3: Graph showing the dependencies between sequences. On the left, for a single telescope, dependencies are of the kind *Calibration* $\rightarrow$ *Data*. On the right, for a stereo analysis sequence, both single telescope data sequences are parent and have to be present for stereo: $\substack{Data\rightarrow \\ Data\rightarrow} Stereo$).