Major Atmospheric
Gamma Imaging
Cerenkov Telescope

MAGIC-TDAS 09-03
090824/IOya

MAGIC On-Site analysis program

I. Oya
<oya@gae.ucm.es>
D. Nieto
<nieto@gae.ucm.es>
R. De los Reyes
<reyes@gae.ucm.es>
J. L. Contreras
<contrera@gae.ucm.es>
J. A. Barrio
<barrio@gae.ucm.es>

August 22, 2009

Abstract

This document is an overview of the MAGIC experiment quick on-site analysis program (QOSA). Since its installation in 2005, the on-site analysis is an important tool in the experiment's data pipeline. Due to their low energy threshold, the MAGIC telescopes event rate is in average around 300 Hz. More than 1.5 TB of raw data is expected for each data taking night. A fast on-site data reduction is needed to have the data quickly available for analyzers, detect hardware problems, and in many cases to decide on observation strategies. The data is automatically calibrated and processed at the MAGIC site using automated scripts on a multiprocessor computing system.

Contents

1 Introduction

MAGIC (Major Atmospheric Gamma Imaging Cerenkov) is a system composed of two 17 m diameter Imaging Atmospheric Cerenkov Telescopes (IACT) located at El Roque de los Muchachos in the Canary island of La Palma (28.8°N, 17.9°W).

MAGIC telescopes acquire images of the fast flashes of Cherenkov light originated by atmospheric showers. These showers are initiated by high energy gammas and the much more numerous cosmic rays background. MAGIC's current energy threshold is around 50 GeV with the standard trigger setup.

The second telescope of MAGIC experiment, MAGIC-II, is now in commissioning phase and the system of two detectors has started to operate in stereo mode.

Due to the size of the recorded data volume, it is still not possible to transfer by Internet the raw files. The program presented in this document performs a key role in the MAGIC experiment data pipeline, reducing the raw data at MAGIC site so they can be delivered by Internet to the experiment data center at PIC (Port d'Informació Científica) [1] with the needed promptness. The raw files are taped and will usually arrive to the data center by airmail several days after the data is recorded.

The Online Analysis [2] is commonly used to detect high state activity of the sources and it is very useful to decide observation strategies. It is, in the other hand, not as sensitive as a full analysis. The observation strategies for weak sources benefit from the possibility of having a fast full analysis of the data reduced by the QOSA software.

2 La Palma computing and network

The computing system at the MAGIC site is a cluster of computers linked by an internal network connected to Internet through a firewall (wwwint.magic.iac.es) [3] and an external machine (www.magic.iac.es).

The internal network consists of the subsystems PCs and the on-site analysis computers. All the computers save their subsystem data locally, sharing the data through NFS (local mount point in /remote). The subsystems computers (PC1-PC7) are software "clones" with OS Suse 7.2 connected to the internal network at 10/100 Mbytes, while the analysis computing (*muxana*, *muxana*{*2,3,4,5*}) and are also "clones" among them but with OS Red Hat 3.4. Another two machines are used as auxiliary processing machines, and these are the two telescope DAQ computers: *muxdaq* for MAGIC-I, and *pc19* for MAGIC-II. The first one has OS Red Hat 3.4 installed while the second one uses OS CERN SLC 4.5. These latter two machines can only be used when the data taking is over. Some details of the analysis machines capacities are described at table 1.

The storage capacity of the cluster is provided by four RAID systems. Each raid system partition has 6.4 Tbytes capacity. RAIDs 1 and 2 have a single partition, RAID 3 has two partitions and RAID 4 has three partitions. In total, 45 Tbytes are available. The I/O speed of the system is between 250 and 300 MByte/s.

The analysis machines are connected with the RAID system through a high speed Gigabit Ethernet connection. Further details on MAGIC site computing system can be found in reference [4].

The central control reports for both telescopes are stored during the data taking at /remote/pc1/home/control/SuperArehucas_CC/logbooks/ and after 8:30 UTC moved to the /local disk at PC15 (or /remote/pc15/local/ccdata/ for the rest of computers). Raw data from MAGIC-I are stored at the RAID system /mnt/raid1/muxdata. MAGIC-II raw files are stored at

Machine	CPU cores	CPU (GHz)	RAM (GB)	Max. proc. 7:00-18:00 UTC	Max. proc. 18:00-7:00 UTC
muxana	8	2.8	8	6	2
muxana2, muxana3	4	3	3	4	2
muxana4, muxana5	8	2	4	8	4
muxdaq	8	2.8	4	8	0
pc19	4	3	3	4	0

Table 1: *Main characteristics of the computers available for the on-site analysis as of August 2009. Last two columns show the number of maximum simultaneous on-site analysis processes that are allowed at each machine.*

/mnt/raid4_1/M2rawdata/ directory.

The processed files are copied to the MAGIC data center at PIC [5]. The transference of calibrated files starts as soon as one full night is processed. On the other hand, image parameter files (*star* files) are copied to PIC and synchronized as soon as they are generated. The transference of the files is performed by using *rsync* processes. More details about the data center can be found at reference [1].

3 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. 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 deploy a Graphical User Interface (GUI).

MAGIC standard analysis tools are several MARS executables. The on-site analysis uses three of these executables:

- MERPP: converts the "raw" and ASCII format of MAGIC subsystems to ROOT package format. The output files have the ".root" extension.
- CALLISTO: 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".
- 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*.

Callisto and star use input cards to define the analysis configuration.

The analysis executables *callisto* and *star* run based on "sequences". A sequence is a text file that lists the basic information of a data sample to analyze. To find an example of a sequence, see class *MSequence* in [6].

The QOSA program only uses the latest official checked MARS release version unless during special situations, generally related to hardware or input files format changes, that require temporal solutions. This is the case presently (August 2009) analysis of MAGIC-II data, that can only be reduced using the development version of MARS.

4 Program description

The aim of the QOSA software is to perform the first part of the data reduction (calibration, image cleaning and parametrization) of MAGIC data, allowing a prompt transfer to PIC. The QOSA software achieves its objective by a high degree of parallelization of the data reduction and starting to work as soon as the first data files are recorded.

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

In this section, MAGIC-I data processing software will be described. The software for MAGIC-II is now in development status, being rather similar to MAGIC-I one. Its particularities are described at section 5.

4.1 Program control scripts

The QOSA program is run by the analysis computers described in section 2, and belongs to the user *analysis*. The program is located at the directory /home/analysis/DataCheck/QOSA in the *muxana* machine. The sub-directories *./bin*, *./macros* and *./awk* contain, respectively, the *csh*, *root* and *AWK* scripts, while the */config* directory contains the configuration details. Reduced data are stored at the analysis directory /mnt/raid1/analysis/DataCheck/Data/analysis/<month>/<day>, while calibrated files are moved to the directory /mnt/raid1/analysis/CalibRootFiles/<year>_<month>_<day> after they are generated. A schematic view of the most relevant directories is shown in figure 1.

The QOSA program automatic operation is managed through the Linux "cron daemon" at muxana using the crontab file from user "analysis" defined at *muxana* machine. The following two scripts, that are launched in the *muxana* computer, are the responsibles of starting the different tasks of QOSA:

- *sequencer.csh*: this c-shell script takes care of the sequence creation and of the launching of the *littlesequence.csh* processes that take care of the data reduction. The "cron daemon" only executes *sequencer.csh* to process the data taken the night before, and it works from 9 UTC to 23 UTC, being launched every 10 minutes

- *monolith.csh*: checks if there is remaining data from previous days to be processed. It also controls the processing while the data taking process is on course. In the case there is data to be processed, this script launches *sequencer.csh* for the corresponding date. The script *monolith.csh* is launched every 15 minutes.

Other auxiliary scripts are:

- *eraser.csh*: allows to immediately halt all the on-site analysis processes in all the analysis machines and prevents the cron daemon to launched any more on-site analysis processes. Option "stop" halts all processes, while "start" resumes the on-site analysis (the processes will be launched by the cron daemon jobs explained before). This script can be also called by the operators in case it is needed to stop the on-site analysis program. The /home/operator/osakiller executable at *muxana* machine halts all processes, while /home/operator/osastarter, at the same machine, cancels the previous halt order, so the on-site analysis processes can be launched by the "cron daemon" again. The operators might want to stop the on-site analysis in certain situations, as for example, in case the disc access of the QOSA processes is interfering with the data acquisition.

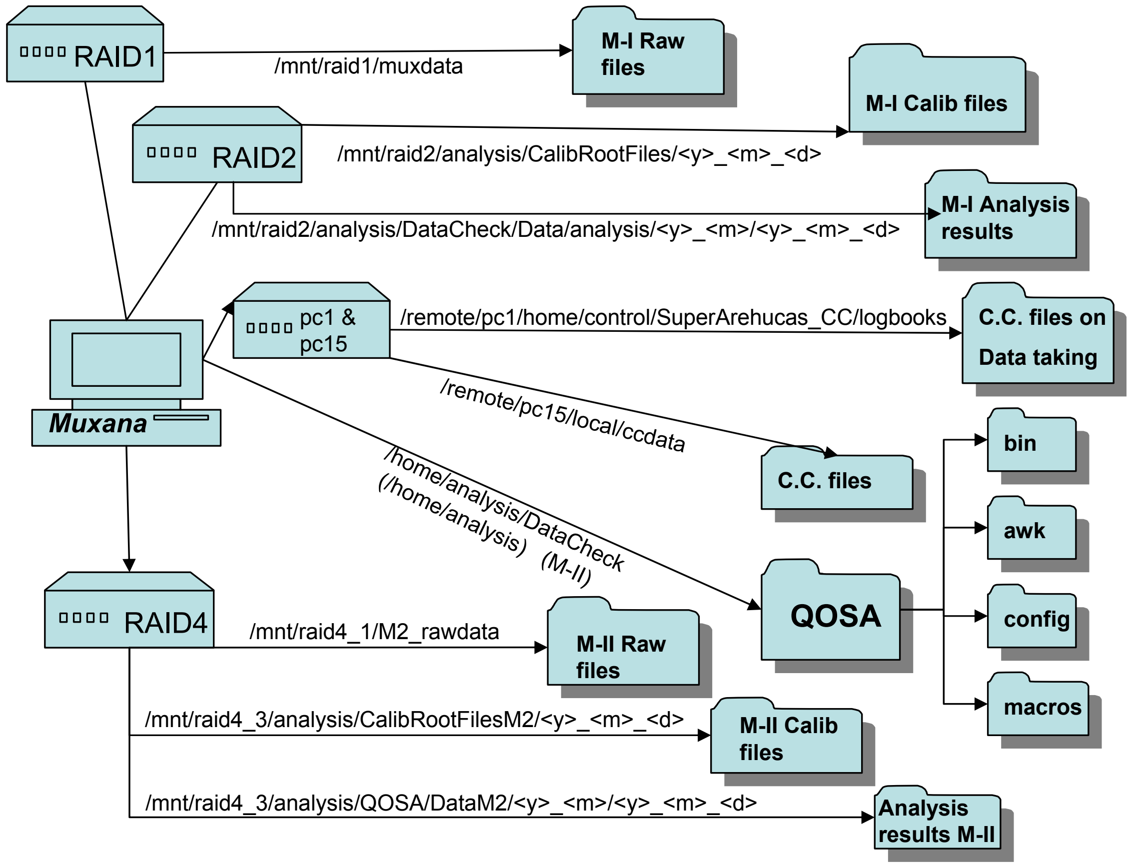

Figure 1: *Scheme of the directories relevant for the on-site analysis.*

- *Status2Web.sh*, launched by the "cron daemon" every 10 minutes from 9:00 UTC to 19:00 UTC, performs some auxiliary tasks useful for the on-site analysis administrators. It copies some information to the webpage http://www.magic.iac.es/operations/datacheck/info/, so the status of the on going processes can be tracked without the need of a *ssh* connection to La Palma computing system. This information is the amount of free disk space on the RAID systems, the number of processes running and the list of raw files not yet processed.

A schematic view of the QOSA program components is shown in figure 2. All the QOSA program code can be found at the MAGIC CVS [6].

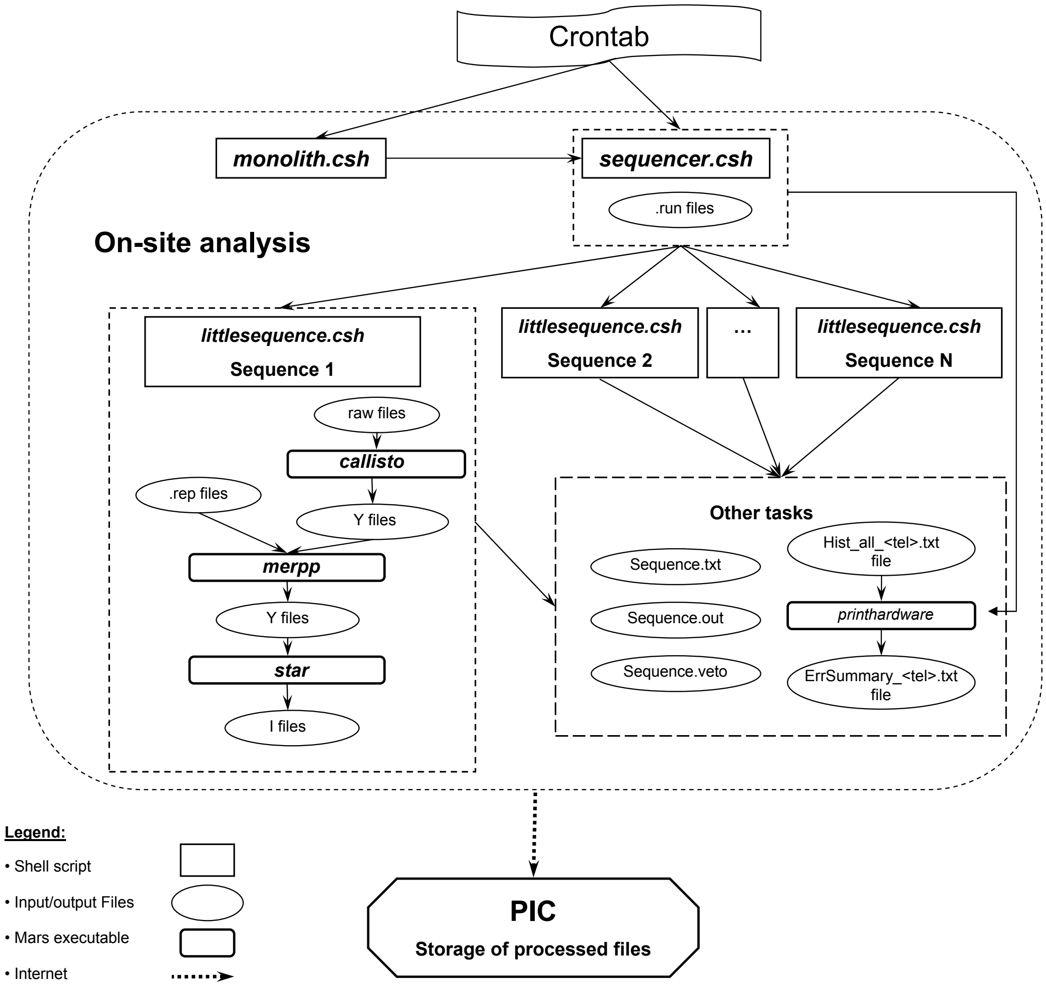

Figure 2: *On-site analysis program. Most relevant scripts, executables and input/output files are shown in a schematic view. The products of the on-site analysis are transferred to the PIC.*

4.2 *Sequencer* tasks

The *sequencer.csh* 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 specified, previous night's data analysis is launched. If one specific night is given as an argument, the program will be launched for the selected night. As an example, the command:
`$PROGRAM_PATH/sequencer.csh 2009_04_01` will launch QOSA for 2009 April 1st

The script *sequencer.csh* first creates a list of the runs to be analyzed, using the *AWK* script called "nightsummary.awk" that reads the run summary files (extension *.run*) present at the central control files directory. The output of this script is stored in a file called
NightSummary_<tel>_<YYYY>_<MM>_<DD>.txt, present in the analysis directory. For each sub run, 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`

Reading the information summarized at *nightsummary.txt*, *sequencer.csh* will create the sequences to be analyzed. In the on-site analysis context, a sequence contains one data run with all its sub-run files. If the two runs before the data run of the sequence are a pedestal and a calibration runs, then both are included in the sequence. Any run tagged as *test* will be ignored: these are technical runs requested by experts that are not intended to be analyzed in the standard way. Before any further work, the script also checks if the sequence has already finished the *callisto* and *star* processes. This is achieved verifying if the first and last callisto and star files have been correctly created.

For all the sequences after a pedestal and calibration pair, the best degree of parallelization is obtained if the the calibration constants from the pedestal-calibration pair can be shared as starting values for the data runs. If the temperature at the camera center remains constant (within 1 degree) along these sequences, then the calibration parameters are assumed to remain stable enough and therefore it is possible to share them for all this group of sequences. Any small variation on the calibration parameters is then corrected by the interleaved calibration and pedestal events. For this parallelization, we use the calibration parameters stored at the file *calib.root*, obtained from a *callisto* (option `-c`) process. In case the temperature variation is above 1 degree, then the starting calibration parameters for sequences without pedestal and calibration pair is the output *signal.root* file from previous callisto process. In this last scenario, one process has to wait until the previous one has finished. In both cases, the best calibration of the data is guaranteed

Each analysis machine has a maximum number of processes that can be run simultaneously. This has been decided according to the characteristics of each machine, and also according to the disk access capacity, avoiding any possible interferences with the DAQs of both telescopes during data taking. Table 1 shows the maximum number of process allowed by each machine. The program manages the CPU availability by checking the number of on-site analysis processes already running on each of the computers, and in case there is room for an additional process, it will launch *littlesequence.csh* (see section 4.3) for the corresponding sequence, unless a veto file is found. A veto file is created during the processing of a sequence if a problem than cannot be solved with the available input card arises. Only three trials (with each adequate input card) are allowed for each callisto and star process before the veto is created.

After the *littlesequence.csh* process is run, two summary files are created:

- *hist_all_<tel>.txt*. This file contains a summary of all the analysis processes already launched by *littlesequence.csh*, obtained by adding each sequences summary. See *littlesequence.csh* section to see the format description.
- *ErrSummary_<tel>.txt*. This file contains an explanation and possible solution for any problem found during all *littlesequence.csh* processes. This file is used also for the daily check procedure [7].

As a final step, *sequencer.csh* launches the root macro *GetAtimes.C*. This macro performs a task needed for the MAGIC Data Check program [13]. It obtains the arrival times distributions from all *callisto* processes of the night and copies the resulting plots to the data check webpage calibration section:
http://www.magic.iac.es/operations/datacheck/daqdata/<YYY_MM>/<YYYY_MM_DD>/

4.3 *Littlesequence* tasks

The purpose of the script *littlesequence.csh* is to execute the first analysis steps on a sequence. It is normally called by *sequencer.csh* for every sequence of the data taking night. It can also be manually executed by specifying the following parameters:
<day> <analysis directory> <signal/calib root file to use> <sequence to process> <telescope>
An example on how to run littlesequence.csh is:

```
littlesequence.csh 2009_07_02 $stardhoy calib01007345_M1.root 1007347 M1
```

4.3.1 Calibration

The goal of the calibration is to find the response of the whole light detection and amplification chain to get the correspondence between the incident photons from the Cherenkov light and the digitized information. This is done through the process which determines the conversion factor from digitized FADC counts to incident photons, and the arrival time delay for each pixel. Some general information of the calibration process can be found in the MAGIC Wiki page [8] and more detailed information at [9].

The script *littlesequence.csh* will first try to perform the calibration of the sequence using the default configuration. The default configuration for MAGIC-I calibration is set by using the input card *callisto_MUX.rc*, that can be found at MAGIC CVS at PIC [6], according the the MARS version used at the moment.

In case the calibration fails, callisto returns an error code through the *exit status* variable. Most of the errors are well known and there are dedicated input cards to allow teh calibration of the data avoiding the error. 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 original quality of the recorded data is not optimum. The most common error codes are:

- Error code 5: This error happens when too many unsuitable pixels have been found during the calibration. The input card *callisto_MUX_clusped.rc* allows to calibrate data with non optimal pedestal files and when a cluster of bad pixels is found.

- Error code 13: This happens when the pedestals have not been correctly recognized with the chosen extractor. The input card *callisto_MUX_pedestal.rc* allows to ignore this error and continue calibrating the data with the non optimum extracted pedestals.

- Error code 15: This happens when the pedestal run has too many events not triggered by the pedestal trigger. The files can still be calibrated by using the input card: *callisto_MUX_pedcal.rc*. The data will be calibrated using the triggered pedestals.

Other possible errors may happen during the calibration process, their meaning can be found in the MARS class MAnalysisProblem.cc [6] and they can be printed by using the MARS executable *printhardware*.

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.

4.3.2 Image cleaning and parametrization

The next step of the data reduction is to clean the calibrated shower images of the sequence from pixels only containing 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. These last steps will be performed by the source analyzers.

The script *littlesequence.csh* runs *star* for the sequence with the standard configuration obtained from input cards *star_timing.rc* for MAGIC-I. The image is cleaned according to the number of phe (photo-electrons) of each pixel and its neighbours, and also according to the arrival times of the pulses at the pixels. The used configuration is the MARS default one: absolute 6/3 phe for core/boundary pixels, 4.5 ns maximum time difference between mean core pixels arrival times and single core pixels arrival times, and 1.5 ns maximum time difference between a boundary pixel arrival time and its core pixel neighbor arrival time. More details on the used image cleaning can be found in reference [10].

The list of image parameters is rather long. The Hillas parameters [11] are the best known ones. They are basically statistical moments up to 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 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.csh* tries again to run *star* with the input card *star_nostgcal.rc*, what will create output files without mispointing correction.

4.3.3 File copy and verification

After the *star* process has finished, the calibrated files of the sequence are copied to the calibrated files directory /mnt/raid1/analysis/CalibRootFiles/<year>_<month>_<day>.

For each one of the processed sequences, a set of ASCII text files is generated. All these files are named as *sequence_<tel>_<runnum>*, and they are distinguished by their extension:

- *.txt*: This file is the sequence itself in the standard MARS format.
- *.out*: This is the log file of the sequence. All output messages from the *littlesequence.csh* and the MARS executables launched by it are stored here.
- *.history*: This is a file with a brief summary of the processes run over the sequence. This allows to control how many times and with what configuration input card has a root executable process been launched, allowing the creation of the previously described veto files. The files contains the following information:
  `<sequence> <process> <day> <time> <input calibration constant file>`
  `<input card> <error code>`
  For star processes the `<input calibration constant file>` field is empty, as well as, for *merpp* processes, the `<input card>` field.
- *.veto*: This file is empty and it has a twofold purpose: First one is to limit the number of consecutive times that a failed process can be again run. The actual configuration only allows three trials before the veto file is created. The second use of this file is to prevent *littlesequence.csh*

to launch *callisto* while the calibrated files are being transferred to the calibrated files directory. A sequence with an existing *.veto* file will never be processed by the QOSA automatic scripts.

4.4 Other tasks

The *star* files are synchronized by rsync processes to the MAGIC PIC database as soon as they are generated. Due to the big size of the *callisto* files, they are only transferred once, when all the sequences have been fully processed. In order to mark a day as finished, an empty text file has to be created at the directory aT *muxana* machine/home/analysis/DataCheck/osa_finished/, named *<year>_<month>_<day>.osa-finished*.
When the calibrated files are transferred to PIC, they are automatically deleted from the RAID system. After the raw files are copied to optical tapes, they are also automatically deleted from the RAID. The transfer and data erasing are tasks under the responsability of other groups in the collaboration.

Finally, the file with the found errors (*ErrSummary_tel.txt*) is copied to the MAGIC webpage at La Palma www.magic.iac.es/operations/datacheck/CCDAQCheck/<month>/<day>.

5 MAGIC-II update

MAGIC-II on-site analysis software is in a development phase. This software is mainly the adaptation of the existing software to the second telescope requirements. The obtained calibrated files are stored at /mnt/raid4_3/analysis/CalibRootFilesM2/<year>_<month>_<day> and the analysis directory, where the *star* files are stored, is /mnt/raid4_3/analysis/QOSA/DataM2/<month>/<day>.

The calibration of MAGIC-II files is slightly different from the first telescope ones. Raw files have to be first "domino-calibrated", by using the so called "Domino coefficients" to take into account the non linearity and temperature dependence of the domino ring sampler. The files containing these coefficients can be found at /mnt/raid4_1/calib_coeff/. Raw files are domino calibrated using the MARS executable *merpp* with the option `--CalibrateDomino`.

A new script, *merpper.csh*, launched from *littlesequence*, takes care of the two needed *merpp* processes, that will be both done before the calibration with *callisto*. The history file for the sequences of MAGIC-II data have a new meaning for the last field of the file in the case of the *merpp* process: 0 if the process went right, and 1 if failed. Due to computer power and available disk constrains, only one trial of the *merpp* process is allowed for each sequence, and therefore if it fails the sequence is vetoed.

The calibration configuration input card used for callisto is *callisto_domino.rc*, The image cleaning form MAGIC-II data is for the time being a very strict one, and absolute 16/6 phe for core/boundary pixels. The time information is not used. The used input card is *star_M2.rc*, found at the program configuration directory, and also in the MAGIC CVS.

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, this 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.

The data volume has increased considerably since the installation of the first version of the on-site analysis: Major hardware updates like the installation of the MUX FADCs, the use of the sum-trigger, and the incorporation of MAGIC-II have increased dramatically the amount of data stored every night.

The actual version of the on-site analysis, QOSA, is the result of the evolution of the original code [14] [16] [17] that, together with the update of the needed computing and storage capacity, has successfully survived the mentioned hardware changes.

7 Acknowledgments

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

References

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

[2] Zanin, R., Moralejo, A. , Hsu, C.C. *Online Analysis*, MAGIC internal documentation, TDAS 08-01.

[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.