The Performance of the H.E.S.S. Target of Opportunity Alert System

A. Balzer$^1$, M. Füßling$^2$, P. Hofverberg$^3$ and R. D. Parsons$^3$
for the H.E.S.S. collaboration

$^1$ GRAPPA, Anton Pannekoek Institute for Astronomy, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
$^2$ DESY, D-15735 Zeuthen, Germany
$^3$ Max-Planck-Institut für Kernphysik, P.O. Box 103980, D-69029 Heidelberg, Germany

E-mail: arnim.balzer@uva.nl

Abstract. The High Energy Stereoscopic System (H.E.S.S.) is an array of five imaging atmospheric Cherenkov telescopes located in the Khomas Highland in Namibia. Very high energy gamma rays are detected using the Imaging Atmospheric Cherenkov Technique. Using the fifth, larger telescope of the array with a huge mirror area of 600 m$^2$, it was possible to lower the energy threshold down to $\approx 30$ GeV. With this unique ability to observe large amounts of gamma rays in the high energy gamma-ray regime ($< 100$ GeV) by using the large effective area of the fifth telescope at these energies, the H.E.S.S. experiment is ideally suited to observe short time scale transient events like gamma-ray bursts (GRBs). Originally detected by the Vela satellites in 1967, GRBs are among the most energetic processes in the known Universe. Extrapolating the spectrum of long duration GRBs (i.e. a GRB duration of the order of a few seconds or above) measured by current satellite experiments like Fermi, which measured gamma rays up to 95 GeV for GRB 130427A, a detection of these phenomena with the H.E.S.S. array is possible.

This paper will give an update on the H.E.S.S. Target of Opportunity (ToO) alert system. It is used for an immediate and fully automatic response to a prompt GRB alert received via the Gamma-Ray Coordinates Network (GCN). The key feature of this system is a fast repointing of the whole array to a new observation position. We will discuss the implementation of the ToO alert system as well as its overall performance. Moreover, we will show that software improvements alone reduced the average response time to a ToO alert to below 60 s on average, a decrease by more than 50%.

1. Introduction

Very high energy (VHE; $> 100$ GeV) gamma-ray astronomy reached maturity with the introduction of the current generation of ground-based Imaging Atmospheric Cherenkov Telescopes (IACTs) and has therefore become an important area of Astroparticle Physics. The High Energy Stereoscopic System (H.E.S.S.) [1] is the most successful of the current IACT arrays [2]. It is located in the Khomas Highland of Namibia at an altitude of roughly 1800 m above sea level featuring four 100 m$^2$ mirror surface area telescopes as well as a later added fifth, larger telescope with a mirror surface area of 600 m$^2$. The resulting huge collecting area of the H.E.S.S. II telescope makes it ideally suited to observe transient events in the VHE gamma-ray domain. This is especially true for long-duration gamma-ray burst (GRBs) [3, 4], their main characteristics...
being a duration of around 30 s [5] and a random distribution within the sky. As a result, the response time of an IACT array to a GRB alert issued by satellites has to be as fast as possible due to the small field of view of IACTs (up to 5° for H.E.S.S.).

This paper will give an overview of the H.E.S.S. Target of Opportunity (ToO) alert system, its integration into the H.E.S.S. Data Acquisition (DAQ) system [6], its recent response time speed-ups due to software changes alone as well as its performance during real and fake ToO alerts.

2. The H.E.S.S. Data Acquisition System

IACT arrays can only operate during dark, clear and cloud free nights. Therefore, the available observation time during a year is limited to roughly 1000 h. To use this time as effectively as possible, a highly efficient array control software is of utmost importance. For the H.E.S.S. array, this task is fulfilled by the H.E.S.S. DAQ, a multi-process, multi-core, multi-machine software. Its main programming language is C++ with some processes also written in Python. For inter-process communication, the omniORB implementation [7] of the CORBA standard [8] is used. The necessary details about the H.E.S.S. DAQ to explain the ToO alert system will be given below, for more details on the H.E.S.S. DAQ implementation, the reader is referred to Balzer et al. [6].

Each component of the detector is represented by at least one Controller process. Complicated hardware, like the Cherenkov cameras, are represented by multiple Controllers at once (i.e. one for the camera high voltage, one for the camera trigger). Each Controller has to implement a common, flat state machine consisting of four different states, connected via eight different state transitions, which represent the current state of the array. The hierarchy of the processes within the DAQ can be freely configured using a MySQL database and can be adjusted to changes in the array configuration. An example hierarchy is shown in Figure 1b.

The H.E.S.S. array can be split into multiple SubArrays, each using a sub set of the telescopes to allow for the observation of multiple targets at once. Moreover, a dedicated SlowControl context is used to monitor the current status of the array at all times. The available observation time during a night is split in 28 min segments called runs. For a run to start properly, the hardware components, and therefore the processes used to control them, have to be prepared for data taking in the right order (the same is true in reverse order for stopping a run). This sequence of processes relying on other processes to be in a given state is called the dependency scheme of the DAQ and is configurable via a MySQL database.

This scheme is enforced by a central managing process for each subset of processes in the DAQ called a Manager. Currently two different implementations of Managers are used, the SubArray-Managers and the SlowControl-Manager. These Managers are also responsible for error handling, i.e. bringing the array to a safe state after an error in one of the components of the detector has occurred.

The RunManager is used to keep track of the available resources in the detector and to distribute them according to the observation schedule for a given night, i.e. which telescopes are used to observe which target for how long and in which order. To ensure high data taking efficiency, two types of processes are used within the DAQ, required and optional processes. An error of a required process will stop the current data taking whereas an error of an optional process does not influence data taking except that the affected component is removed from the run until the problem is fixed and it can rejoin the run.

Note that each component is responsible for its own safety and fatal problems have to be handled directly in firmware first, after which they notify the DAQ of the error.
3. The H.E.S.S. Target of Opportunity Alert System

In case of GRBs, IACT arrays rely on other experiments to trigger them. A dedicated communication network can be used for this purpose, the Gamma-Ray Burst Coordinates Network (GCN). A ToO alert system suitable to react to GRB alerts for an IACT array has to react as fast as possible and, ideally, fully automatic to an incoming alert. Moreover, it has to be able to react to an alert in any given state of the array during data taking.

The H.E.S.S. DAQ is able to react to a ToO alert during ongoing runs, during the starting and stopping of a run as well as while being idle. It also features a dedicated process called the GCNAlerter which listens to the alarms it receives from the GCN network. Once a suitable alarm is received (i.e. passing several criteria like minimum altitude, red-shift, etc.), the RunManager is informed of an incoming ToO alert. The RunManager then notifies all SubArray-Managers of the ToO alert which in turn notify each of their processes of the ToO alert. This allows each process in the DAQ to immediately react to an incoming alert. For instance this is used by the Tracking Controller of the H.E.S.S. II telescope to immediately slew to the coordinates of the new target regardless of its current state in order to minimize the response time of the array. Moreover, special flags accessible by each Controller during the transitions can be used to speed-up a ToO alert response. Once all processes have been notified, the RunManager will stop all ongoing runs, if any, and start the ToO observation run with all available telescopes.

Note that optional processes are not waited upon during a ToO alert in contrast to normal behavior, i.e. a process will only wait for its required dependencies, not for its optional ones. To make use of the faster slewing time of the H.E.S.S. II telescope, all small telescopes are marked as optional processes during a ToO alert run, thus also speeding-up the response time of the array. As a side effect, an error in one of the smaller telescopes will not stop the data taking of the H.E.S.S. II telescope as well. For further information on the H.E.S.S. ToO alert system the reader is referred to Balzer et al. [10].

---

2 The Gamma-Ray Burst Coordinates Network (GCN) distributes information about the location of a Gamma-Ray Burst detected by various spacecraft, see [http://gcn.gsfc.nasa.gov/](http://gcn.gsfc.nasa.gov/)
4. Software Speed-ups

To be able to benchmark the performance of the H.E.S.S. DAQ and to calculate the data taking efficiency of the H.E.S.S. array, each Controller in the DAQ writes several time stamps to a MySQL database. For each state transition a Controller performs, three time stamps are written, one at the start of the transition, one once all dependencies of the current transition are fulfilled and one at the end of the transition. A Python analysis framework called the Transition Time Tools (TTT) is used to analyze these time stamps. Moreover, the response time of the system to a ToO alert can be determined with microsecond precision. Additionally, the contribution of each component of the detector to the total response time can be calculated.

The goal for the H.E.S.S. DAQ is to achieve a negligible total software overhead of less than 5% of the total response time of the system, i.e. all of the response time is spent on hardware activity (telescopes slewing, camera and trigger configuration, etc.) and not on software. While analyzing the data obtained during the data taking using the first implementation of the fully automatic ToO alert system in the H.E.S.S. DAQ (daq-3-0), a significant software overhead induced by the DAQ framework was found. Software overhead can occur, for example, when processes are waiting for other processes (their dependencies) and are not immediately aware of state changes of their dependencies. Any time passing between the slowest dependency of a process finishing its transition and the process itself starting its own transition is wasted by the DAQ and is called software overhead. As seen in Figure 2 for the daq-3-0 implementation, an overhead of up to several seconds per process could occur. This resulted in a summed software overhead of more than 50% of the total response time of the system for a ToO alert.

![Diagram](a) DAQ overhead distribution
![Diagram](b) DAQ overhead histogram

Figure 2: The DAQ overhead (the time spent by the core DAQ implementation on synchronizing, book keeping, etc.) is compared between the daq-3-0 (blue) and the daq-4-0 (green) implementation, each for a time period of ≈ 7 month. A sorted distribution (left) and a histogram (right) of the DAQ overhead in seconds for the relevant processes in the DAQ is shown. A significant reduction in DAQ overhead of more than two orders of magnitude is observed.

To rectify the problem, a redesign of the process hierarchy within the H.E.S.S. DAQ was done (daq-4-0) as shown in Figure 1. For the old daq-3-0 implementation, each sub-context was assigned its own Manager process. Moreover, each Controller was told its list of dependencies for each transition and started to poll its dependencies until they were done (using an increasing time out up to 1 s). The new design introduced a single Manager per SubArray. Instead of

---

3 Note, that an additional time span of around 20 s to 30 s has to be added to the total response time due to a delay from the detection of the GRB to the arrival of the GRB alert on site.
each process taking care of its own dependencies, the *SubArray-Manager* is taking care of the dependency handling. Moreover, the polling approach has been changed to a signaling one, i.e. a process waits to be notified by its *SubArray-Manager* that its dependencies are fulfilled. Only then it start with its own transition and notifies the *SubArray-Manager* once it is finished with the transition. Therefore, if multiple processes are waiting for the same dependency, the amount of sent messages has been greatly reduced in addition to the response time for these messages. In total, the average software overhead per process was reduced from 348.4 µs to 2.2 µs, a speed-up of almost 160. The reduction in DAQ software overhead is clearly visible in Figure 2. The new implementation contributes with less than 2 s on average to the total response time to a ToO alert which is negligible in comparison to the overall response time of more than 60 s on average.

5. Performance

To quantify the performance of the new DAQ implementation, the transition times for observation runs before and after the change were compared. The transition time is defined as the time needed to start and stop a given run. This includes all hard- and software activity of the detector components used for this run, i.e. telescope slewing time, DAQ software overhead, etc. The result of the comparison is shown in Figure 3. A clear reduction in transition time is visible from an average of 176.4 s to 118.4 s, a speed-up of 1.49.

![Figure 3](image-url)

(a) transition time distribution  
(b) transition time histogram

Figure 3: The transition time for observation runs (the time spent to start and stop a run) is compared between the daq-3-0 (blue) and the daq-4-0 (green) implementation, each for a time period of ≈ 7 month. A sorted distribution (left) and a histogram (right) of the transition time in seconds for each observation run is shown. The significant improvement between the two implementations is clearly visible. The different slope of the tail of the distributions is due to the slewing time of the telescopes which starts to dominate at long transition times.

Regular full system tests of the H.E.S.S. ToO alert system, called GRB fire drills, are performed to ensure a working system, especially after changes to any of the software involved. Moreover, these test can be used to benchmark the ToO alert system. A crucial measurement is the response time of the system to a ToO alert which is shown in Figure 4. The response time should be dominated by the telescopes slewing time which is strongly dependent on the angular distance of the current target and the ToO alert target. Therefore, the response time scaled by the angular distance between old and new observation position is also shown in Figure 4. A clear trend towards lower response times is obvious. Large spikes in response time can occur due to long slewing times of the telescopes in case of ToO alert targets located above an elevation of 90°. In this case, the telescopes have to move up to 180° in azimuth to reach the new target. This
behavior is due to the H.E.S.S. telescopes not tracking in reverse mode at the moment, i.e. the usable elevation range of the telescopes is between $0^\circ$ and $90^\circ$).

### 6. Summary & Outlook

The H.E.S.S. ToO alert system has been working smoothly since its implementation and is able to react fully automatic to incoming GCN alerts. The response time of the ToO alert system has been significantly sped-up due to software improvements in the H.E.S.S. DAQ alone by reducing the total software overhead to below 2 s on average. By now, response times smaller than 60 s have been reached which is closer to the goal of the average duration of long duration GRBs of around 30 s. However, with the recent activation of the reverse tracking capability of the H.E.S.S. II telescope, a further reduction in slewing time, and therefore response time, is expected from an average of 112 s to 52 s for 90% of incoming GRB alerts as shown in [7]. Moreover, further speed-ups in the Cherenkov cameras starting and stopping procedures are planned. This should reduce the response time even further until the H.E.S.S. ToO alert system response time is only dependent on the slewing time of the telescopes and other influences are negligible.

![Figure 4: The performance of the H.E.S.S. ToO alert system is shown for a given set of real and fake GRB alerts. The upper plot shows the response time of the whole array for the different alerts, the middle plot is the response time scaled by the angular distance to reduce the influence of the slewing time of the telescopes. The lower plot shows the total overhead of the software (book keeping, process synchronization, run scheduling, alert processing, etc.). Green circles indicate an alarm during an ongoing run, red squares show an alarm while the array was idle and yellow triangles indicate an alarm during the start of a run. A white star indicates real GRB alerts, all other data points are fake alerts. The dotted lines show a linear/exponential fit to the data. A clear improvement over time is visible, note that the change from daq-3-0 to daq-4-0 was done between GRB140818 and (fake) GRB140819181509.](image-url)
Acknowledgments
The authors would like to acknowledge the support of their host institutions. We want to thank the whole H.E.S.S. Collaboration for their support.

The support of the Namibian authorities and of the University of Namibia in facilitating the construction and operation of H.E.S.S. is gratefully acknowledged, as is the support by the German Ministry for Education and Research (BMBF), the Max Planck Society, the German Research Foundation (DFG), the French Ministry for Research, the CNRS-IN2P3 and the Astroparticle Interdisciplinary Programme of the CNRS, the U.K. Science and Technology Facilities Council (STFC), the IPNP of the Charles University, the Czech Science Foundation, the Polish Ministry of Science and Higher Education, the South African Department of Science and Technology and National Research Foundation, and by the University of Namibia. We appreciate the excellent work of the technical support staff in Berlin, Durham, Hamburg, Heidelberg, Palaiseau, Paris, Saclay, and in Namibia in the construction and operation of the equipment.

References
[1] F. Aharonian et al. “Observations of the Crab nebula with HESS”. In: A&A 457 (Oct. 2006), pp. 899–915. DOI: 10.1051/0004-6361:20065351 eprint: astro-ph/0607333.
[2] J. A. Hinton and W. Hofmann. “Teraelectronvolt Astronomy”. In: ARA&A 47 (Sept. 2009), pp. 523–565. DOI: 10.1146/annurev-astro-082708-101816 arXiv: 1006.5210 [astro-ph.HE].
[3] R. W. Klebesadel et al. “Observations of Gamma-Ray Bursts of Cosmic Origin”. In: ApJ 182 (June 1973), p. L85. DOI: 10.1086/181225.
[4] C. Kouveliotou et al. “Identification of two classes of gamma-ray bursts”. In: ApJ 413 (Aug. 1993), pp. L101–L104. DOI: 10.1086/186969.
[5] D. Lennarz et al. “Searching for TeV emission from GRBs: the status of the H.E.S.S. GRB programme”. In: ArXiv e-prints (July 2013). arXiv: 1307.6897 [astro-ph.HE].
[6] A. Balzer et al. “The H.E.S.S. central data acquisition system”. In: Astroparticle Physics 54 (Feb. 2014), pp. 67–80. DOI: 10.1016/j.astropartphys.2013.11.007 arXiv: 1311.3486 [astro-ph.IM].
[7] Sai-lai Lo and Steve Pope. “The Implementation of a High Performance ORB over Multiple Network Transports”. In: In Middleware 98: IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing. 1998, pp. 157–172.
[8] OMG. The Common Object Request Broker: Architecture and Specification. Tech. rep. 2.0. Object Management Group, 1995.
[9] P. Hofverberg et al. “Commissioning and initial performance of the H.E.S.S. II drive system”. In: ArXiv e-prints (July 2013). arXiv: 1307.4550 [astro-ph.IM].
[10] A. Balzer et al. “The H.E.S.S. data acquisition system”. In: Journal of Physics Conference Series 513.1, 012003 (June 2014), p. 012003. DOI: 10.1088/1742-6596/513/1/012003.