29th International Cosmic Ray Conference Pune (2005) **3**, 437–440

The data acquisition software system of the 2004/2005 CREAM experiment

Zinn, S.-Y.[b], Ahn, H. S.[b], Allison, P.[c], Bagliesi, M. G.[d], Beatty, J. J.[c], Bigongiari, G.[d], Boyle, P.[e], Childers, J. T.[f], Conklin, N. B.[g], Coutu, S.[g], DuVernois, M. A.[f], Ganel, O.[b], Han, J. H.[h], Hyun, H. J.[h], Jeon, J. A.[h], Kim, K. C.[b], Lee, J. K.[h], Lee, M. H.[b], Lutz, L.[b], Maestro, P.[d], Malinine, A.[b], Marrocchesi, P. S.[d], Minnick, S.[j], Mognet, S. I.[g], Nam, S. W.[h], Nutter, S.[k], Park, N. H.[h], Park, H.[i], Park, I. H.[h], Seo, E. S.[a,b], Sina, R.[b], Swordy, S.[e], Wakely, S.[e], Wu, J.[b], Yang, J.[h], Yoon, Y. S.[a], and Zei, R.[d]

(a) Dept. of Physics, University of Maryland, College Park, MD 20742, USA
(b) Inst. for Phys. Sci. and Tech., University of Maryland, College Park, MD 20742, USA
(c) Dept. of Physics, Ohio State University, Columbus, Ohio 43210, USA
(d) Dept. of Physics, University of Siena and INFN, Via Roma 56, 53100 Siena, Italy
(e) Enrico Fermi Institute and Dept. of Physics, University of Chicago, Chicago, IL 60637, USA
(f) School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA
(g) Dept. of Physics, Penn State University, University Park, PA 16802, USA
(h) Dept. of Physics, Ewha Womans University, Seoul, 120-750, Republic of Korea
(i) Dept. of Physics, Kyungpook National University, Taegu, 702-701, Republic of Korea
(j) Dept. of Physics, Kent State University Tuscarawas, New Philadelphia, OH 44663, USA
(k) Dept. of Physics and Geology, Northern Kentucky University, Highland Heights, KY 41099, USA
Presenter: Hoseok Ahn (syzinn@umd.edu), usa-zinn-SY-abs1-og15-poster

During the 2004/2005 Antarctic campaign, CREAM (Cosmic Ray Energetics And Mass) had a record-breaking flight of about 42 days and of three circumnavigations around the continent. The CREAM data acquisition software system (CDAQ) has provided excellent stability and robustness. The design and overall flight performance of CDAQ is presented. During the flight, CDAQ collected physics, calibration, pedestal, and housekeeping events and sent them to University of Maryland via NASA's Command Data Modules (CDMs), satellites, and support centers. The interfaces, not only to the instrument but also to the telemetry support infrastructures, are described in some details.

1. Introduction

CREAM measures the energy spectra of cosmic rays from ∼1 to 1000 TeV and their composition from protons to iron nuclei [1]. CREAM had its first flight in December 2004. It was launched in Antarctica and flew for about 42 days, which broke the previous record of about 32 days for a long-duration balloon experiment.

The CREAM payload consists of the science instrument and the support systems. The CREAM collaboration built the science instrument containing, from top to bottom, a timing charge detector (TCD), a transition radiation detector (TRD), a silicon charge detector (SCD), a hodoscope (HDS), a calorimeter (CAL), a science flight computer (SFC), and common electronics. The NASA support systems contain two communication and data modules and a power management system including solar panels and batteries. Only one CDM is active at a given time and the other is redundant.

During the flight, the SFC collected various physics, calibration, pedestal, log message, and housekeeping events. The SFC handed over fragments of the data, following a custom protocol, to the active CDM [2]. However, due to limited bandwidth of about 85 kbps (kilobits per second), not all events could be transmitted

via telemetry. So events of a particular high-rate trigger were saved to a flash disk on the SFC and were not delivered to the active CDM.

Upon receiving data from the SFC, the active CDM transmitted the data to a ground station via TDRSS (Tracking and Data Relay Satellite System) or, when the payload was out of TDRSS coverage, just reduced housekeeping packets via IRIDIUM. The ground station that received TDRSS data was located in White Sands, New Mexico. The data were forwarded to the Operational Control Center (OCC) in Palestine, Texas and then to the Engineering Support Center (ESC) in Wallops Island, Virginia. Finally, the ESC forwarded data to the Science Operation Center (SOC) at the University of Maryland. Science commands travelled from the SOC to the SFC using the same route but in the reverse direction. Through this telemetry route, we were able to control the instrument and to collect data of 19 GB in near real-time, on top of the archived.

2. Design and implementation

The interface between the SFC and the rest of the instrument was as follows. Triggering and synchronizing of events was coordinated by a master trigger board. The SFC communicated with the master trigger via a digital I/O board. The command board for the CAL, HDS, SCD, and some of the common electronics was connected via a serial port. The housekeeping board used another serial port. The TCD had nine concentrators which were running a Linux operating system. So the TCD interfaced to the SFC via Ethernet. The TRD employed a FPGA board and a digital I/O board. The CAL, HDS, and SCD were connected to the SFC by using custom PC/104 cards. Finally, the communication to the CDMs was done by Ethernet.

Since telemetry was involved for command and data, it was natural to adopt a server/client model for the CREAM data acquisition software system. Commands or requests were originated from a ground computer so the software running at the SOC was a client. The flight software running on the SFC was a server.

For simplicity, whenever possible, we minimized the number of processes running on the server. CDAQ_SERV was the master process which managed other processes. All other processes were launched by the master process. CDAQ_SERV also created and initialized six buffers used for inter-process communications. More specifically, we implemented circular buffer rings based on shared memory and semaphore lock so that CDAQ processes could pass commands or event data efficiently.

CDAQ_SNIO was the network communication process that talked to the CDMs. Details pertaining to CDM communication are given in Sec. 3 below. CDAQ_SNIO generated packets from events according to the custom protocol set out for communication with the CDM. It also extracted commands from packets received from the CDM. Threading and real-timing schemes were adopted for CDAQ_SNIO to function most efficiently over an extended period of time. Commands from the SOC were forwarded to the CDAQ_CMD process which directed them to the pertinent processes. In addition, the CDAQ_CDM process generated commands for pedestal and calibration runs at commandable intervals. The CDAQ_HK process collected housekeeping data from the housekeeping board and other processes. The housekeeping events were passed to CDAQ_SNIO. Finally, CDAQ_EVT constantly monitored the master trigger and built events whenever a trigger occurred. The data and command channels for CDAQ_HK and CDAQ_EVT were kept separate so that the housekeeping events could be monitored regardless of the status of CDAQ_EVT.

Similarly to CDAQ_SERV, CDAQ_CLI served as the master process on client computers. Depending on our designation for different computers, CDAQ_CLI started processes for remote commanding, for receiving data streams, or for making automated backups of telemetered events. For a remote commanding computer, CDAQ_RCMD and a commanding GUI were run in addition to a remote commanding software provided by NASA. A set of predefined scripts was also stored on this computer. Commands were generated either from

the GUI or from a command-line script processor. CDAQ_RCMD fetched commands from either source, translated them to a predefined format, and handed them over to the NASA software which in turn encrypted and relayed them to the ESC.

For a computer receiving the data stream, CDAQ_RSTR, CDAQ_RSTR2, and CDAQ_RELAYD were run. CDAQ_RSTR received ITOS[1] packets from a TCP port and extracted science packets that the SFC generated. Whenever enough packets were gathered, CDAQ_RSTR reassembled events and saved them to files. For monitoring purposes, a certain portion of physics events and all of the housekeeping events were passed to the online monitoring programs. When the payload went into a zone of exclusion (ZoE) for TDRSS, the real-time data streaming was cut off. Later when TDRSS became available again, the lost portion of the data was played back to another port. CDAQ_RSTR2 listened on this playback port and reassembled events. Playback events were saved to files but they were not used for online monitoring because they were not real-time. During the playback, the bandwidth between the SFC and a CDM was brought down from 85 to 50 kbps and the difference of 35 kbps was used for the playback. CDAQ_RELAYD was a server program for making automated backup copies of the telemetered events on another computer. CDAQ_RELAYD read a circular buffer where events were recorded by CDAQ_RSTR and passed to CDAQ_RELAY running on a computer making backups.

CDAQ_RELAY wrote events to files in the same manner as CDAQ_RSTR and also copied them to a circular ring buffer for online monitoring. The events could be propagated to other computers if the backup computer was running CDAQ_RELAYD. Since the telemetered events were saved to this backup computer in near real-time, the CREAM collaborators could retrieve files for further monitoring or quick off-line analysis. Furthermore, the housekeeping data were posted on a web site every five seconds unless the payload drifted into the TDRSS ZoE. This allowed monitoring the instrument around the globe at any time during the flight.

3. Telemetry

As mentioned in Sec. 1, the SFC communicated with the CDMs via Ethernet. More specifically, a custom UDP protocol was adopted. UDP was selected over TCP because it is connectionless: this makes it easy to re-establish communication between the SFC and the CDMs after losing one side, e.g., due to power cycling the SFC, switching from one CDM to the other, etc. Five types of fixed-length packets were used. These were connection status, data, housekeeping, acknowledgment, and command packets.

The connection status packets were generated every five seconds by both the SFC and the active CDM and were exchanged asynchronously. By examining the connection status packet, the SFC detected switching of the active CDM. When the CDM was switched, the SFC sent packets to the newly activated CDM from then on until another switching occurred. Additionally, the connection status packet served to initiate communication between the SFC and the CDMs. The UDP port numbers of the CDMs were not specified. Only the UDP port number of the SFC side was specified. When the active CDM sent the first connection status packet, the SFC learned which port number to use and the custom connection was established.

The length of the data and housekeeping packets was 255 bytes. Since most events exceeded this packet size, fragmentation and reassembly by CDAQ were necessary. To this end, five bytes were allocated which reduced the usable packet size to 250 bytes. For optimal usage of TDRSS bandwidth, events were tightly packed and were allowed to cross packet boundaries. By design, an event could vary from 4 bytes to 64 kilobytes. Thus, in principle, one data or housekeeping packet could contain at most 62 events or a fragment of an event larger than a usable packet.

When the data or housekeeping packets were sent by the SFC, an acknowledgment packet from the active

[1] Integrated Test and Operations System developed by NASA

CDM was expected. The SFC sent the same packet repeatedly at one second intervals until the packet was acknowledged or thirty attempts were made. After thirty attempts, the SFC dropped the connection and checked both CDMs by pinging. The SFC resumed sending the same packet upon establishing a new connection. However, to prevent an infinite transmission loop, a packet could be timed out eventually and discarded. In such a case the next packet was to be processed. It turned out that during the flight, the packets were acknowledged instantly almost 100% of the time.

The command packets came from the active CDM to the SFC. The SFC acknowledged a command packet after checking CRC[2] by sending an acknowledgment packet. The command packet could contain an eight byte command or one fragment of a parameter file which was 1050 bytes. The length of the command packet served to differentiate these two. Eight parameter file fragments were assembled by CDAQ_SNIO to produce a file. This uploading capability could be used to update sparsification tables or to transfer files from the SOC to the SFC.

Since the telemetry involved RF communications between the CDM and the first ground station via satellites, packets could be corrupted. The packet corruption was detected mostly whenever the payload entered or left the zone of exclusion. This made it important for CDAQ_RSTR and CDAQ_RSTR2 to detect and drop corrupt packets, to recover from such situations, and to proceed with reassembling. For packet checks, the SFC generated a 16-bit CRC for every packet and placed the low byte in the packet. This accounts for one byte among the five byte header. (The other four bytes contained reassembly information and error recovery in case of missing packets.) However, it turned out that this method of checking packet integrity was not perfect. At a very low rate, a corrupt packet passed the test and was reassembled to full events. It was found that a total of only 29 events were corrupted. The level of contamination is thus about 1 per million telemetered events or 1 byte per billion bytes assuming an average size of 1 kB for all telemetered events. Later the contaminated events were identified and replaced when the events were reconstructed from the CDM backup files.

4. Summary

CREAM had its first flight during the 2004/2005 Antarctic campaign. CREAM gathered data in the amount of 19.3 GB, 37.6 GB, and 17.9 GB via telemetry, archiving to the SFC disk, and reconstructing events from the CDM backups, respectively. After having removed duplicate events, the flight data amounted to 48.9 GB. The number of physics, pedestal, calibration, housekeeping, and log message events are 43,539,770, 1,611,631, 34,645, 572,790, and 12,799,928, respectively.

5. Acknowledgments

The financial and telemetry support by NASA is gratefully acknowledged. We also thank the National Scientific Balloon Facility and the National Science Foundation for the Antarctic campaign.

References

[1] E.-S. Seo et al., Advances in Space Research **30**, 1263 (2002).
[2] S.-Y. Zinn et al., To appear in Nucl. Instr. & Meth. A (2005).

[2] Circular redundancy check is a standard method for checking the integrity of data.