Event reconstruction in the LHCb Online cluster

Albert Puig Navarro
Universitat de Barcelona-ICC, Barcelona (Spain)
E-mail: apuig@ecm.ub.es

Markus Frank
CERN, Geneva (Switzerland)
E-mail: Markus.Frank@cern.ch

on behalf of the LHCb Collaboration

Abstract. The LHCb experiment at the LHC accelerator at CERN will collide particle bunches at 40 MHz. After a first level of hardware trigger with output at 1 MHz, the physically interesting collisions will be selected by running dedicated trigger algorithms, the High Level Trigger (HLT), in the Online computing farm. This farm consists of 16000 CPU cores and 40 TB of storage space. Although limited by environmental constraints, the computing power is equivalent to that provided by all Tier-1’s to LHCb. The HLT duty cycle follows the LHC collisions, thus it has several months of winter shutdown, as well as several shorter machine and experiment downtime periods. This work describes the strategy for using these idle resources for event reconstruction. Due to the specific features of the Online Farm, typical processing à la Tier-1 (1 file per core) is not feasible. A radically different approach has been chosen, based on parallel processing the data in farm slices of \(O(1000)\) cores. Single events are read from the input files, distributed to the cluster and merged back into files once they have been processed. A detailed description of this architectural solution, the obtained performance and how it will be connected to the LHCb production system will be described.

1. Introduction
LHCb is the experiment dedicated to B-physics at the Large Hadron Collider (LHC) at CERN [1]. Its main goal is to look for new physics through the study of CP violation and rare decays of beauty hadrons. The LHC will deliver proton-proton collisions at a centre of mass energy of up to 14 TeV to the LHCb detector at a rate of 40 MHz. This data rate is reduced with a two-level trigger system to roughly 2 kHz of particle collisions to facilitate the handling of these data with the available computing and long-term data storage resources. The first level trigger is hardware based and reduces the rate of accepted events to 1 MHz. The second level or High Level Trigger (HLT) is software based and requires significant computing power. The HLT is performed by running special algorithms in a dedicated farm, called the Online Farm, and lowers the event rate to the final 2 kHz.

These computing resources, which actually correspond to the power provided by all Tier-1’s to LHCb, are only used when the LHC is producing collisions and can be assigned to other functions during periods without collisions, such as event reconstruction or Monte Carlo generation. In
the following we describe the usage of the computing resources to perform event reconstruction for reprocessing during periods when LHC does not provide particle collisions. Such an activity may be performed during LHC winter shutdown periods or machine development phases, which in total are expected to account for up to 50% of the available time. The farm resources and the method to use them to perform reconstruction will be outlined in section 2 and in section 3 details of the implementation will be given. In section 4 the performed tests will be discussed, as well as future plans and prospects. Finally, some conclusions will be drawn in section 5.

2. Using the Online Farm for reconstruction

The Online farm is expected to host roughly 16000 CPU cores, typically grouped on processor boards with 8 cores each, which are equally distributed over 50 subfarms, physically and geographically represented by a rack hosting the corresponding processors. The installation, currently 25% complete and limited by space and cooling restrictions, is connected to disk storage system of 40 TB which can be accessed at 400-500 MB/s. All nodes are connected with 1Gb interfaces to the data acquisition network, but only a few, the so-called Storage nodes, can access directly disk storage.

During data taking the accepted events with a size of 30-35 kB are recorded in 2 GB files, containing the detector response of approximately 60000 particle collisions. The first stage reconstruction, which takes 1-2 s per event, yields approximately 20-25 kB of reconstructed data per event and does not contain the raw data. The file size of 2 GB may increase in the future depending on the data handling requirements of the offline processing environment.

Typical offline processing strategies, where each CPU core processes a single file, are not possible in the LHCb Online Farm for three reasons:

- The storage system cannot provide sufficient space to host both input files (16k cores × 2GB = 32 TB) and output files (16k cores × 1.5 GB = 24TB). Thus, there is insufficient disk space or CPU would not used efficiently.
- Short idle periods could not be used for reconstruction because of the necessary time to process one single file, about 1 day.
- The complexity if a storage system that can sustain 32k simultaneous open files is far beyond what is available in the LHCb Online environment.

To overcome these problems a strategy which takes advantage of the fact that events are independent has been developed: each core processes single events instead of whole files. This is achieved by reading sequentially events and sending them to the worker nodes, thus reducing the time consumed to process one file and reducing the load of the storage system with less files being accessed simultaneously. Theoretically, with one open file at a time and using the whole Online Farm, one would be able to reconstruct the events contained in each file in 4-8 seconds. However, such a setup results in a bottleneck, since one single process reading from disk cannot read a single file sufficiently fast to feed all reconstruction processes executing in the Online farm (one per CPU core).

Therefore, a more balanced and flexible setup has been chosen to simultaneously optimize the disk access, the storage requirements and the integrated reprocessing time per file. Such a setup requires several file reading processes each accessing one file at a time. With 3-4 readers the reconstruction time per file is well below one minute and hence a possible change of function of the Online Farm such as (re)starting data taking activities can be quickly achieved.

A flexible implementation must also allow to perform several, although not arbitrary, reconstruction activities in parallel, e.g. test serveral versions of the reconstruction program or prioritize a small reprocessing request. Hence a solution which allows to split the entire Online farm into several logical slices was chosen. Each slice, which consists of a set of subfarms, is an independent entity connected to its own slice of the storage system and can be used...
independently. The only connection point between the slices is a resource allocator, which ensures that any given subfarm is used by one single slice. This is how the Online Farm works when trigger work is being performed. In the following section we describe the allocation and the management of resources and the system used to actually deliver “work” to a reconstruction slice.

3. Implementation

Three main systems, shown in Figure 1, collaborate in order to facilitate event reconstruction in a distributed manner using the LHCb Online Farm. These building blocks are:

- **Resource management**: The management and allocation of the farm or parts thereof, including the storage system, the storage nodes and the working nodes, taken care by the ECS (Experiment Control System). This part deals with the configuration of the processes participating in the reconstruction process, the definition of the different paths through which raw event data are transferred from the storage system to the worker nodes and how reconstructed event data are collected.

- **Workload management**: Definition of the “work” to be performed by the Online reconstruction system, taken care by the Reco Manager. Given a slice of the Online farm already preconfigured for Online reconstruction use, the Workload Management system is responsible to start the data flow and to keep the worker nodes busy as long as there is pending work.

- **Integration in the LHCb Production System**: The Online reconstruction system is not a standalone entity. In order to maximize its usefulness it must integrate into the LHCb Offline data processing facilities, like any other site used for event reconstruction. This integration is done via DIRAC (Distributed Infrastructure with Remote Agent Control) [2], the LHCb Grid solution, which interacts with the Reco Manager.

3.1. Resource management

Both the hardware installation and the controls structure of the Online Farm is hierarchical, as shown in Figure 2. The entire farm is divided into subfarms hosted by one rack. Each subfarm consists of 1 control PC and several multicore PC boards, which execute one reconstruction task per core. These reconstruction tasks take a considerable time to start and configure, so they are not stopped when the file to reconstruct changes, only when the configuration of the reconstruction processes changes.

Farm resources are allocated, configured and operated on using the standard experiment controls software [3][4][5], based on PVSS [6]. This allowed to realize the reprocessing framework with existing components already developed for the operation of the HLT farm. Existing components were reused for the control of the reconstruction processes, but also to transport event data to the worker nodes, to transport reconstructed data back to the storage and to allow the reconstruction processes to access these data.

Whenever a reprocessing activity is to be started the allocation unit dynamically creates reconstruction slices, which consist of

- an input slice responsible to read event data and deliver individual events to the worker nodes,
- a configurable number of subfarms to host the reconstruction processes,
- an output slice responsible to collect reconstructed data from the worker nodes and writing them to disk.
Figure 1. Schema of the various components that constitute the reconstruction system: resource management is done by the ECS, workload management and job steering is taken care of by the Reco Manager, which also interacts (although implementation is still pending) with DIRAC in order to connect the farm with the LHCb Production System and the Tier-0 center (arrows in and out of the figure).

The mechanism to access and transport the event data is described elsewhere [7]. For completeness, Figure 3 recalls the functionality of the two basic building blocks, described by a producer-consumer paradigm with a shared memory buffer manager to exchange data.

All components related to this data flow are realized using the basic building blocks, as can be seen in Figure 4:

- Several Reader processes are started in the storage input nodes, the exact number depending on the number of concurrently open files one wants. When idle, each reader receives a command to open a file and declare sequentially all events to the connected buffer manager.
- Several Sender processes (one per connected subfarm) access the events from the buffer manager and distribute the events to the worker nodes.
- On each worker node a Receiver process collects the data event by event and declares the events to the local input buffer.
- The reconstruction processes access the input data, perform the event reconstruction and declare the results to the output data buffer.
- A Sender process, which is subscribed to the local output buffer in the worker node, sends the result back to output slice on the storage.
- Several Receiver processes (one per connected subfarm) on the storage output node collect the reconstructed data from the network and declare the events to the local data buffer.
- A Writer process writes the reconstructed data to disk.

The Reader and the Writer processes on the Storage system must interact with the job management system described in section 3.2 in order to know which file to open for reading.
Figure 2. The Online Farm has a hierarchical control structure: it is divided into subfarms, formed by 1 control PC and several worker nodes, each having 8 cores. It is intended to run one reconstruction process per core, plus some data transfer and management processes.

Figure 3. Building blocks for data transfer and management in the Online Farm, implementing the producer-consumer paradigm with the use of buffers. On the left, the data processing block: a Producer declares data to the buffer manager in the form of events. Consumers subscribe to these events and receive data whenever a new event is declared by the Producer. On the right, the data transfer block: a Consumer process receives data from a buffer manager located in the source node and sends the data via a network connection to a Receiver process located in the target node. This Receiver acts as a Producer for the buffer manager on the target node.

and writing respectively. Since the reconstruction of one event typically is finished after 1-2 seconds the Writer process closes a file after no events for this file were received for a sufficiently long time. This timeout strategy was chosen because individual events cannot be tracked as they flow through the system. Therefore, crashes are detected at the level of single files, which
Figure 4. Data flow of events from the raw data to the reconstructed file. An event is read by the Storage Reader and sent to the worker node Input buffer. It is then picked by the Reconstruction process, which processes it and puts the resulting data into the Output buffer. These data is sent to the Storage node again, where it is stored to disk by the Storage Writer.

have to be debugged offline, an assumption deemed reasonable for large reprocessings with many files. At closing time the total number of events written is communicated to the job management system and cross-checked with the number of input events. In order to ensure that data from all events in one input file are also written to the corresponding output file, each file open request is accompanied by an identifier, which is added to the event data by the reader. This identifier travels with the event data and is used by the writer to identify the correct output file.

3.2. Workload Management
Though individual events are reconstructed on the worker nodes, the atomic managerial unit are files: only full files are manipulated and reconstructed. Since for each file a number of specific actions must be performed in a specific order it is advantageous to manage each file independently using a Finite State Machine (FSM) mechanism. The FSM executes the following actions with the corresponding state transitions:

(i) File enters the system.
(ii) Reading of the file starts and individual events are sent to the subfarms.
(iii) Reading finishes, input file is closed.
(iv) Writing of the reconstructed data finishes, output file is closed.
(v) File has been correctly reconstructed.

Handling each file to reconstruct with an FSM allows to easily keep track of the status of each reconstruction job. The status of each file, its identifier, the GUID, the number of events and the location of the input and output data are stored together with a timestamp of the last transition as a record in a database. This database allows to:

- Keep track of the history of the tasks performed.
• Protect against system failures such as power cuts. In the event of a system failure, the last known state can be restored with the information in the database.

To take care of steering the reconstruction tasks by managing the FSMs and the database, the Reconstruction Database Manager (Reco Manager) has been created. Its duties are:

• **Steer the reconstruction tasks.** File processing is triggered by the entries in the Reconstruction Database. Each slice has its own Reader, subfarms and Writer. All these resources are connected using buffers (as seen in section 3.1) in such a way that the output from a task goes directly to the input of the next task. That means that once reading of a file starts data will automatically flow through the allocated resources until its reconstructed counterpart is written to disk. The Reco Manager job is to provide the Readers with the files to open and provide the Writers with the names of the reconstructed file they must write to. The processing work is thereby automatically done by the resources allocated by the given slice.

• **Hold the FSM instances** and move them through the various states of the FSM. These FSM instances are virtual representations of the state of the file based on the stage of the reconstruction it is in. When a specific set of conditions –stored in the FSM definition– is met, the Reco Manager toggles state transitions and executes appropriate actions. Therefore, the Reco Manager is reactive to the environment conditions, i.e. the status of the reader/writer clients, and each time conditions change the status of the FSM is reevaluated and, if necessary, transitions are applied.

• **Interact with the Database**, which is kept updated with each change of the states of the FSMs. This database contains information on all the files –processed, being processed and not processed– and it is not only used for storing data but also to query for information, i.e. new files to reconstruct.

• **Interact with the LHCb Production system.** The Reco Manager doesn’t by itself insert any new files into the database, it just modifies the present ones according to their corresponding state. However, in order to trigger the reconstruction, information on files to work on as given by the LHCb production system must be inserted into the database, either before starting reprocessing or on-the-fly during other reconstruction tasks. In this case, the Reco Manager acts as a thin client between the agents that have to insert the information and the reconstruction database.

### 3.3. Integration in the LHCb Production System

Though the Reconstruction Manager steers the processing of files, the entire system described so far is standalone and does not communicate with the world of computing outside the LHCb Online environment. The following are missing from the current implementation:

• Import of raw data files from the Tier-0 (CERN) for reconstruction, which, as described above, results in new work items in the reconstruction database.

• Export of new reconstructed datasets to the Tier-0, and from there to the Tier-1 centers, where the reconstructed files are needed, taking care of registering them in the LHCb bookkeeping and the Grid File Catalogue.

This integration, which is still missing, will be achieved by treating the Online Farm as a DIRAC Computing Element (CE) connected to the LHCb Production System. By doing so, reconstruction is formulated in the same way as usual offline data processing in LHCb, i.e. reconstruction tasks are DIRAC jobs managed by the DIRAC Workload Management System; data moving in and out of the Online farm is managed by DIRAC Data Management System Agents. A DIRAC Agent is also responsible for populating the database with the information of
the new reconstruction jobs through a thin client interface with the Reco Manager, as explained in section 3.2.

4. Status and Future Plans

4.1. Performed tests

In the absence of the connection to the LHCb Production System, a number of files to be reconstructed were provided by hand and the corresponding records were inserted into the Reconstruction Database. The tests performed focused on the ability to use the existing hardware as efficiently as possible and for that reason a high-speed Fiber Channel (FC) interface between the Storage System and the Storage nodes was temporally installed. Several results were obtained:

- One single Reader process connected to the Storage System saturates one CPU core at a speed of 45 MB/s.
- Three concurrent Reader processes were able to saturate the installed FC connection to the storage at 130 MB/s.
- A slice test using three Reader processes and 25 subfarms (800 cores), showed that the system can sustain a stable data throughput 100 MB/s for a total workload of more than 100 files of 2 GB each until all files were processed. The limit was imposed by the 1 GBit/s network interfaces of the Storage nodes to the Farm. To minimize possible interferences with other software components while still ensuring the proper data load, the reconstruction tasks were replaced by stubs simply copying the input data to the output buffer.

The tests have shown that the current performance bottleneck is the Gigabit connection of the Storage nodes. The link to the Tier-0 has a safety factor of roughly two with respect to normal data taking activity. Hence, no bottlenecks should occur if one allows to saturate the data link to the Tier-0 centre.

4.2. Future plans

The system is limited by the data throughput of the processing nodes accessing the storage and distributing the data to the individual worker nodes. These nodes, currently equipped with 1 GBit/s network interfaces, will be equipped this summer with 10 GBit/s network interfaces to distribute/collect the data to/from the worker nodes. Also planned is an improved Fiber Channel connection to the storage media, which will allow to access data for input and output with a speed of 300 MB/s per storage node. With the help of these two improvements most of the current hardware constraints will vanish.

To achieve compliance with the LHCb production system, emphasis will be put on the integration of the Online Farm as a regular computing resource to the LHCb production system.

5. Conclusions

The LHCb Online Farm requires significant computing resources to run the HLT algorithms during data taking but they are only used for a fraction of the time of the year. We have shown that these resources can be used during their idle periods for data processing such as event reconstruction. The turnaround time to reconstruct one file can be significantly reduced by applying a parallelized architecture. Hence the reconstruction activity may be started or stopped within few minutes, allowing to take advantage of idle times much shorter than one day. Though we target event data reprocessing the technique of processing one single event in one core can easily be used for other data processing activities such as Monte Carlo generation. In the chosen implementation, standard components of the experiment controls system and the
event data flow have either been largely reused or could be adapted to provide the required functionality. Only the reconstruction manager, used to access the work repository, and the integration with the LHCb Production system require special developments. The results of system tests show that current performance is mostly constrained by the network connections between the storage nodes—a hardware deficiency which we plan to remove soon.

Acknowledgments
AP would like to thank R. Graciani and A. Casajús for the fruitful discussions. AP acknowledges financial support by the MICINN FPU grant.

6. Bibliography
[1] Amato S et al. 1998 CERN-LHCC-98-04
[2] Tsaregorodtsev A et al. 2003 *Computing in High-Energy Physics and Nuclear Physics* 2003 (La Jolla, California)
[3] Gaspar C and Franck B 1997 *Computing in High-Energy Physics and Nuclear Physics* 1997 (Berlin, Germany)
[4] JCOP Framework Team 2007 CERN-JCOP-2000-008
[5] ITCOBE: Framework Project URL http://itcobe.web.cern.ch/itcobe/Projects/Framework/
[6] ETM 2004 PVSS II, Version 3.0 Eisenstadt, Austria URL http://www.etm.at/
[7] Frank M, Gaspar C, van Herwijnen E, Jost B, Neufeld N, Cherukuwada S and Stoica R 2007 *Computing in High-Energy Physics and Nuclear Physics* 2007 (Victoria, Canada)