RAVEN – Boosting Data Analysis for the LHC Experiments

Michael Schmelling[1]*, Markward Britsch[1], Nikolai Gagunashvili[2],
Hans Kristjan Gudmundsson[2], Helmut Neukirchen[3], and Nicola Whitehead[2]

[1]Max-Planck-Institute for Nuclear Physics, Heidelberg, Germany
michael.schmelling@mpi-hd.mpg.de,markward@mpi-hd.mpg.de,
[2]University of Akureyri, Akureyri, Iceland
nikolai@unak.is,hgk@unak.is,nicolaw@unak.is
[3]University of Iceland, Reykjavik, Iceland
helmut@hi.is

Abstract. The analysis and visualization of the LHC data is a good example of human interaction with petabytes of inhomogeneous data. After outlining the computational requirements for an efficient analysis of such data sets, a proposal, RAVEN – a Random Access, Visualization and Exploration Network for petabyte sized data sets, for a scalable architecture meeting these demands is presented. The proposed hardware basis is a network of "CSR"-units based on off-the-shelf components, which combine Computing, data Storage and Routing functionalities. At the software level efficient protocols for broadcasting information, data distribution and information collection are required, together with a middleware layer for data processing.

Keywords: LHC, Particle Physics, RAVEN, data analysis, CSR-unit

1 Introduction

In particle physics the basic units which make up a data set are so-called "events". In former times an event did correspond to a photograph showing the interaction of a high energy particle with an atomic nucleus in a bubble chamber, at the LHC [1] it is the information recorded from a single bunch crossing of the two proton beams. A bubble chamber photograph is shown in Fig.1.

The bubble chamber is a detector device which allows to collect and to display the full information about a high energy particle physics interaction in a very intuitive form. Its main drawback is that it can only record events at a rate of a few Hz, which renders it unsuitable to look for really rare types of interactions. As a consequence, over the last 30 years they have been replaced by electronic detectors which nowadays are able to scrutinize high energy interactions with rates up to 40 MHz and to store information from potentially interesting events

* corresponding author

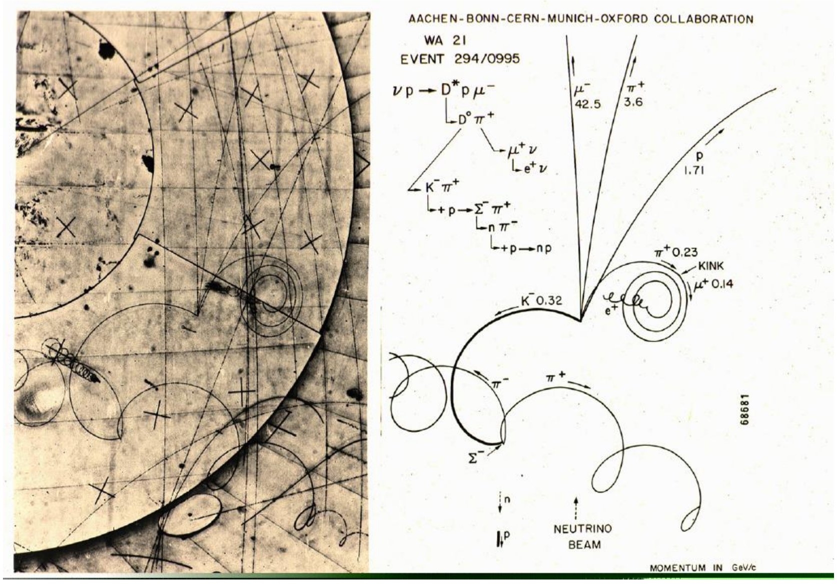

Fig. 1. Bubble chamber photograph of a high energy collision between elementary particles. Secondary particles are created from the available energy and travel before they decay or induce secondary interactions. The information about the reaction is contained in the momentum vectors of the final state particles, their charges and the points (vertices) of interactions. (Photo by CERN).

with a few kHz. For example at LHCb [2], one of the four large LHC experiments at CERN, the typical amount of information corresponds to 50 kB per event and events can be stored with a rate of 2 kHz. With an expected number of 2×10^{10} events per year, the annual data volume amounts to O(1) PB.

Individual events are reconstructed by means of sophisticated numerical algorithms. Those start from the raw information collected by the, depending on the specific experiment, 1 - 200 million readout channels of the detector. From those they extract the equivalent information one would have from a bubble chamber photograph, i.e. particle trajectories, vertices, decay chains etc. An example how the information from a modern electronic detector can be visualized is shown in Fig.2.

The final analysis of the reconstructed data is conceptually simple in the sense that all events are equivalent, i.e. at the event-level it parallelizes trivially. Also the information content of a single event has a relatively simple structure, consisting of lists of instances of a few basic elements such as “tracks” or “vertices” which contain the measured information about the final state particles created in a high energy collision. Different events will differ in the number of those objects and the relations between them.

In contrast to analysis tasks in other branches science where the main problem is accessing the relevant data items, data analysis in particle physics is

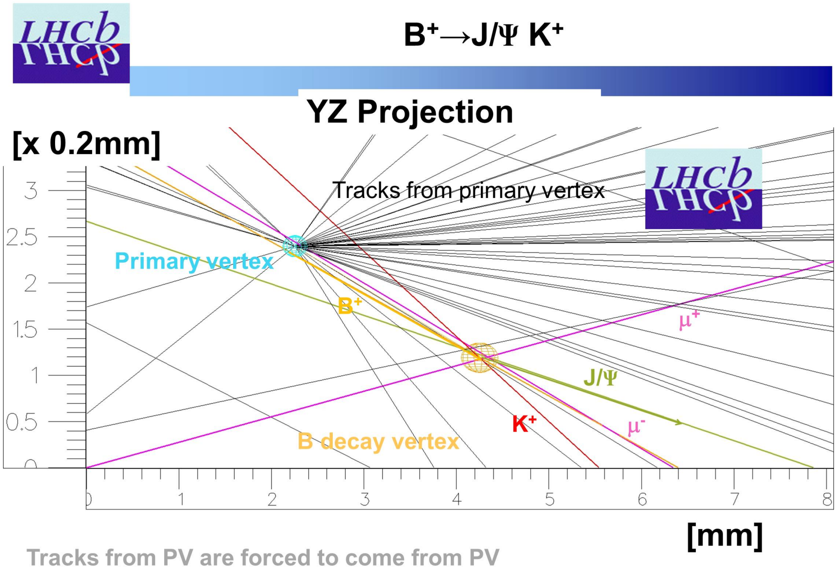

Fig. 2. Visualization of an interaction recorded by the LHCb experiment. Information from electronically read out detectors is used to reconstruct trajectories of particles created in a high-energy proton-proton collision. The computer generated image shows a zoom to the region of the primary vertex where two protons from the counter-rotating beams did collide. In a addition to a large number of particle created at the primary vertex, the reconstruction also shows the decay of a so-called B-meson which after creation at the primary vertex travels a few millimeters before decaying into three longer lived final state particles (K^+, μ^+, μ^-).

completely dominated by the processing of the event information. Compared to the processing step reading and decoding the data usually requires negligible CPU resources. To illustrate this, consider the problem of finding for example decays of so-called D^0-mesons into a pair of final state particles in LHCb. Finding such decays in an event requires checking all combinations of two tracks and to decide whether or not this pair is consistent with coming from a D^0-decay. In LHCb one has to check on average 73 combinations per event, with each single check requiring up to hundreds of floating point operations. In addition, D^0 decays into two final state particles, though still frequent compared to many other decay channels of interest, are already rather rare. Only about 1 percent of all events contains a D^0, and only about 4 percent of those decay into the specific two-particle final state. Already for this “easy” example the data analysis has to cope with a situation where the background is about 200,000 times larger that the signal.

The basic mode of data analysis in particle physics is characterized by two steps. In the first step the data set is scrutinized for events containing a specific signature. Events with this signature then are analyzed in detail, either by extracting some characteristic information or by iterating the selection process with additional criteria.

It is evident that depending on the selection criteria the size of the event sample used in a specific analysis can vary by many orders of magnitude. On the other hand, the maximum communication bandwidth available to return information back to the user will essentially by fixed, i.e. the interaction between user and the full data set must be such that the network traffic stays below a certain limit.

The quantities of interest in a typical particle physics analysis are probabilities or probability density functions for a certain process or configuration to occur. Numerical estimates are obtained by means of histograms, i.e. simple counters for how often a certain condition is observed. The analysis framework thus must be able to handle this kind of cumulative information, which even for very large event samples reduces to a limited set of numbers.

In addition to cumulative information from many or even all events, the system must be able to transmit some or all information from a few selected events. This is of particular relevance for very rare types for final states, such as for example events with a candidate Higgs decay or other exotic processes and which require an in depth analysis of single events.

The combination of the two access modes becomes particularly relevant in the context of interactive searches for special event types starting from the full data set. Here powerful visualization tools and user interfaces are required, which provide an intuitive representation of the properties of the event set, together with the possibility of interactive select-and-zoom schemes to focus on certain candidates.

2 Computing Requirements

During the construction of the CERN Large Hadron Collider (LHC) it was realized that the analysis of the data produced by the LHC experiments requires a computing infrastructure which at the time went beyond the capabilities of a single computing center, and which since then has been built up in the framework of the Worldwide LHC Computing Grid (WLCG) [3–5]. The design of the WLGC was driven by the requirement to allow a sharing of the effort between many partners and the ability to cope with future increases of the computing demands.

Despite the fact that many new concepts regarding data distribution and sharing of computing load have been implemented, the computing models for the analysis of the LHC data (see e.g. [6]) are still very close to the approach by earlier generation particle physics experiments. They focus on filtering the huge initial data sets to small samples relevant for particular physics question, which then are handled locally by the physicist doing the analysis.

While making efficient use of limited resources, this scheme has some obvious shortcomings.

- At a given time direct access is possible to only a small fraction of the total event sample. This reduced sample also has to serve to define and check the selection criteria for the selection jobs. As a consequence the selection may be biased or inefficient.
- The time constant for full access to the data is given by the frequency of the selection runs which go through the complete data set. Programming errors or missed deadlines for code submission can easily result in serious delays for the affected analyses.
- High statistics measurements, i.e. analysis which use information from more than a small fraction of all events, are not feasible. The same holds for finding exceptional rare events which are not caught by selection criteria based on prior expectations.

What is needed is a framework which allows random access on petabyte-size datasets. It should have a scalable architecture which allows to go to real time information retrieval from the entire data set. The initial use case of this infrastructure will be faster and more efficient access to the data for classical analysis scenarios. Beyond that, however, also novel ways of interacting with the data and new ways of data visualization will evolve.

3 Design Aspects

The requirements outlined above suggest a design similar to that of a biological brain: a dense network of many "simple" nodes combining data storage, processing and the routing of information flow. For the use case of particle physics, each node would store a small fraction of the total event sample, have the possibility to run an analysis task on those events and route information back to the user having submitted the analysis query. In the following these nodes will be referred to as Computing-Storage-Routing (CSR) units, which at the hardware level are standard commodity CPUs. With an appropriate middleware-layer a network of such CSR-units will then constitute a RAVEN system: a Random Access, Visualization and Exploration Network for petabyte sized data sets.

The key feature which guarantees exact scalability is a peer-to-peer architecture [7] where every node is able to perform every functionality required by the system. This departs significantly from the current Grid installation which is built around a system of services that are associated with distinct units, such as for example "worker nodes", "storage elements" or "work-load management systems". While the current Grid-approach is natural in the sense that different functionalities are identified and implemented separately, it results in a rather complex infrastructure with corresponding requirements in terms of maintenance. The RAVEN approach, in contrast, aims at defining a protocol or rule-set which allows the system to organize itself.

Another important aspect of RAVEN is redundant and possibly also encrypted data storage. While encryption should simply ensure the confidentiality of the data also in case that public computing resources are used, redundant storage assures that the entire data set can still be processed even if some nodes

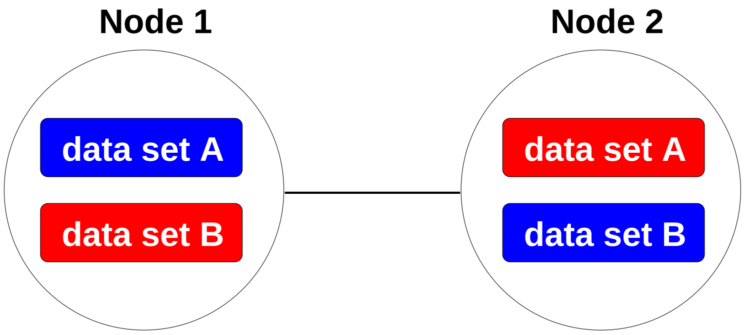

Fig. 3. Simple of example of redundant data storage on two nodes. If both nodes are present, the analysis starts in parallel on different subsets. Node 1 will start on data set A, node 2 will start with B. If one node is unavailable, either because it's down or busy with another task, then the other node will process the entire data set.

become unavailable. A simple sketch how duplication of data between two nodes can serve these purposes is shown in Fig. 3. Although encryption adds to the computing costs, typical applications in particle physics analysis are such that the decoding step only adds a small overhead to the actual analysis.

For a particular analysis or visualization task, instructions would be broadcast to all CPUs. These instructions will then be executed on the local event samples, and the information retrieved from those events routed back to the user.

As discussed before, with respect to the information that is returned one has to distinguish between cumulative data, and per-event data. Since all data have to go back to a single node, per-event data should either be of only limited volume per event or should be transmitted for only a subset of all events. Cumulative data on the other hand, such as histograms, flowing back through the network can be accumulated on-the-fly such that the total amount of information transmitted over the network stays small even for very large event samples. Figure 4 illustrates the case.

4 Implementation Aspects

A central feature of the design of a RAVEN system is its scalability, which almost automatically comes from the fact the different events are independent and thus can be spread over as many CSR-units as are available. Scalability allows to develop RAVEN on a small test system and later expand the working system to the size required for a particular application, possibly also taking advantage of cloud-computing infrastructures.

A particular implementation dealing with 1 PB of data spread over 10^5 CSR-units would correspond to 10 GB per node. Assuming a processing speed of 10 MB/s, which seems possible today, the data set could be processed within a quarter of an hour. A test system should typically have one percent of the capacity of the 1 PB system.

One problem that has to be addressed for RAVEN is the creation of ad-hoc routing and communication topologies for a given analysis query to be used both to distribute the query to all nodes and to collect the results of the analysis. Here a big challenge arises from the fact that logically certain next-neighbor topologies may be required which then have to be mapped to actual routing schemes by taking into account the existing hardware capabilities and the data flow that needs to be handled. Furthermore, since many analyses will only access subsets of the full data set, the system should be able to process multiple queries simultaneously.

Another issue is the distribution of data, analysis code and actual query of a specific analysis. One big challenge is the distribution of the full data set. Here different data items have to go to different nodes, which in view of the total data volume that has to be distributed is a non-trivial task. Uniform distribution can be achieved by some hashing scheme, where a hash-code of every event or file determines on which node it will be stored and analyzed [8]. If the RAVEN system is able to autonomously distribute the address-space spanned by the hash code among its members, then an event entering the system via any node can be routed to its proper destination. It is also easy to check whether a particular event is already stored on the system. The data distribution scheme also should take care of the redundant storage scheme. Finally, the RAVEN system must be able to automatically detect new CSR-units joining the system and to migrate part of the data to the new resources.

Apart from data distribution also bookkeeping of available data has to be addressed. Although particle physics analyses can be performed on subsets of the total event sample, a proper interpretation of the results requires the knowledge about the actual events that have been processed. Even if the redundancy built into the system will normally guarantee access to the full data set, a monitoring of which events contribute to a particular result has to be foreseen.

While the distribution of the full data set will happen only rarely, updates of the analysis code will be more frequent, though still rare compared to analysis queries. The latter two can be distributed via a broadcast mechanism. The splitting into analysis code and query is motivated by the goal to minimize the network traffic. Instead of distributing the full analysis code, which for a typical LHC experiments amounts to O(1) GB, with each query, a layered ("middleware") approach suggests itself. Here the (in general machine dependent) analysis code forms a software layer on top of the operating system. This "analysis middleware" then provides a machine independent high level language to perform the actual physics analysis.

While the mapping of the classical analysis models based on histograms or n-tuples on a RAVEN infrastructure is relatively straightforward, the system calls for novel approaches to exploit its real-time capabilities in new visualization tools for the interaction of a human being with petabytes of data.

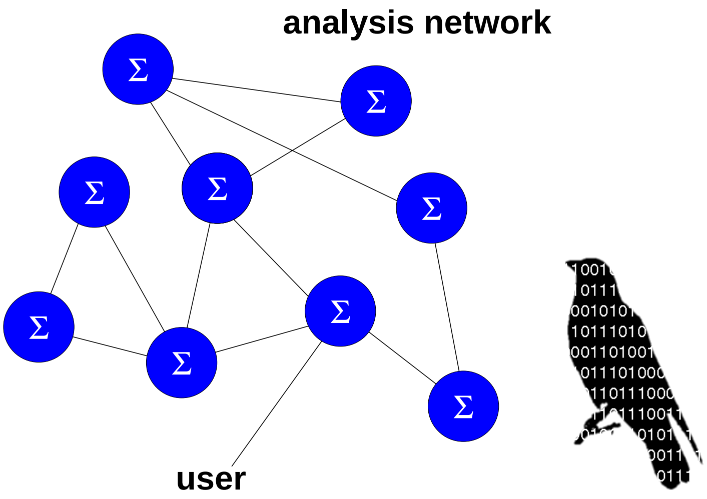

Fig. 4. Sketch of a RAVEN network. Histogram data, for example, produced by the analysis jobs on the different CSR-units are routed back to the node connected to the user, and updated on-the-fly on the way back. The links show which nodes are aware of their neighbors, i.e. the network topology routing and data distribution have to deal with.

The performance of the system can be optimized by making sure that events falling into the same class with respect to a specific selection are distributed as evenly as possible. An analysis query addressing only that subset then will harness a large number of CPU simultaneously and finish with minimal time. Providing analysis jobs with the possibility to tag events as belonging to a certain class should lead to a system which is able to automatically migrate data between nodes in order to minimize access times.

Another level of optimization would be to store event related information which is created by a specific analysis for further use. Information that should be kept in a persistent store can either be specified by the user, or selected automatically, e.g. storing by default all information that is determined with computational cost above a certain threshold.

5 Prior Work

Realization of the RAVEN project will benefit greatly from already existing knowledge in networking, middleware design, distributed data storage and computing. Projects which are in principle interesting from the point of view of RAVEN are for example `BitTorrent` [9, 10] for broadcasting information over a network, the `Apache Hadoop` [11] project addressing MapReduce-based [12] scalable, distributed computing, the `BOINC` [13, 14] framework for volunteer computing and grid computing, or the `xrootd` [15, 16] server for low latency high bandwidth data access in the `root` [17, 18] framework, which defines the de-facto standard for data analysis in particle physics. Additional input could come from the Grid-middleware developers e.g. `gLite` [19, 20] or the `Linux` community [21].

6 Summary

Physics analysis of the data recorded by the LHC experiments calls for new computing architectures which ares scalable to allow fast parallel access to petabytes of data. One possible approach is the RAVEN system, featuring redundant storage, on-the-fly accumulation of results and a rigorous middleware-approach to the data analysis.

References

1. Evans L. and Bryant P. (editors): LHC Machine. JINST 3 (2008) S08001.
2. The LHCb Collaboration, A. Augusto Alves Jt. et al.: The LHCb Detector at the LHC. JINST 3 (2008) S08005.
3. The LHC Computing Grid: LCG Website, `http://lcg.web.cern.ch/lcg/`.
4. Lamanna, M.: The LHC computing grid project at CERN. Nucl. Instrum. Meth. **A** 534 (2004) 1–6.
5. Eck, C., et al.: LHC computing Grid: Technical Design Report. Version 1.06. LCG-TDR-001 CERN-LHCC-2005-024.
6. The LHCb Collaboration, Antunes Nobrega R. et al.: LHCb Computing Technical Design Report. CERN/LHCC 2005-019.
7. Oram, A.: Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly & Associates, Inc., Sebastopol, CA, USA (2001); ISBN 059600110X.
8. Balakrishnan H., Kaashoek, M. F., Karger, D., Morris, R. and Stoica, I.: Looking up data in P2P systems, Commun. ACM **46** (2003) 43–48.
9. BitTorrent, Inc.: BitTorrent Website, `http://www.bittorrent.com`.
10. Cohen, B.: Incentives Build Robustness in BitTorrent. 1st Workshop on Economics of Peer-to-Peer Systems, University of California, Berkeley, CA, USA (2003).
11. The Apache Software Foundation: Apache Hadoop Website `http://hadoop.apache.org/`.
12. Dean, J. and Ghemawat, S.: MapReduce: simplified data processing on large clusters, Commun. ACM **51** (2008) 107–113.
13. BOINC project: BOINC website, `http://boinc.berkeley.edu/`.
14. Anderson, D. P.: BOINC: a system for public-resource computing and storage. Fifth IEEE/ACM International Workshop on Grid Computing 2004.
15. XRootD project: Scalla/XRootD Website, `http://project-arda-dev.web.cern.ch/project-arda-dev/xrootd/site`.
16. Hanushevsky A., Dorigo A. and Furano, F.: The Next Generation Root File Server. Proceedings of Computing in High Energy Physics (CHEP) 2004, Interlaken, Switzerland, 2004.
17. The ROOT team: ROOT Website, `http://root.cern.ch/`.
18. Antcheva I., et al.: ROOT – A C++ framework for petabyte data storage, statistical analysis and visualization. Computer Physics Communications **180** (2009) 2499–2512.
19. gLite Open Collaboration: gLite Website, `http://glite.web.cern.ch`.

20. Laure E., et al.: Programming the Grid with gLite. Computational Methods in Science and Technology **12** (2006) 33–45.
21. The Linux Foundation: The Linux Foundation Website, `http://www.linuxfoundation.org`.