PROCEEDINGS OF SCIENCE

Computing and Data Handling

Doris Y. Kim*
Soongsil University, South Korea
E-mail: dorisykim@ssu.ac.kr

The modern particle physics experiments produce vast amount of data sets, which require up to date software libraries, large scale computing resources, and well-defined management schemes. In this article, the status of computing and data handling systems utilized by the particle physics experiments are discussed, and some of the most recent developments in the field are summarized.

38th International Conference on High Energy Physics
3-10 August 2016
Chicago, USA

*Speaker.

PoS(ICHEP2016)011

1. Introduction

The data sets produced by particle experiments are huge. Their operations require extraordinary amount of computing resources, compared to those used by the past experiments. The cost of procuring computing resources is increasing rapidly, projected to be near to the cost of detector construction. Various new ideas have been tested to speed up the computing system and to increase its efficiencies. Usages of distributed computing systems over the worldwide networks are common. Software libraries and management schemes are being updated continuously. Collaboration among different experiments has become more important, to share useful ideas and exploit limited computing resources together.

In this article, recent developments in the computing and data handling system are discussed. The author did not attempt to cover all the available topics, which are simply too many to summarize in a restricted space.

2. Overview of collider experiments

2.1 Status of Large Hadron Collider (LHC) Run II

The Run II operation of the LHC experiments at CERN started in 2015. The operation is scheduled to continue until 2018. The major differences between Run I and Run II are accelerator parameters: The center of mass energy of the colliding beams is raised from 8 TeV to 13 TeV. The bunch spacing in the beams is reduced from 50 ns to 25 ns. The LHC experiments updated their computing and data handling systems accordingly. The following paragraph summarizes some of these changes, with the examples from the CMS experiment [1].

Due to the changes in the accelerator parameters, pile-up of events in the detectors is increased to 40 events per bunch crossing. The CMS experiment responded to this challenge by optimizing the reconstruction algorithms for calorimeters and tracking [1]. Another change is the introduction of a new analysis object data format (Mini-AOD), which is smaller than the predecessor in size but flexible enough to satisfy most of the analysis needs [2]. The reconstruction of data is required to run at the 48 hour frame after data acquisition. The offline calibration should be completed within this window [3]. The Monte Carlo (MC) simulation events are produced according to the prioritized schedule. Monitoring of data quality is done continuously and certification of good data is conducted regularly to provide the best data set to users [4].

2.2 New experiments: ILC and Belle II

ILCSOft is a collection of common software tools developed for the Linear Collider (ILC) groups [5]. Numerous packages are created by the collaboration of the Linear Collider (LC) projects. Recently, there has been extensive development on the geometry and simulation framework, DD4hep (Detector Description for High Energy Physics) [6, 7]. One of the requirements for the framework is that it should support the entire life cycle, from the concept design stage till the data taking operation and data analysis stages. DD4hep has subpackages for simulation, alignment, conditions data, and reconstruction. For the distributed computing system, iLCDirac is created as a user interface [8]. The system is based on DIRAC [9, 10] with additional functionality.

The SuperKEKB collider at KEK commenced its operation in 2016. The full Belle II detector will start physics run in 2018. The Belle II software frame (basf2) is a modular structure and is capable of parallel processing. Currently, basf2 is at the later stage of its development [11]: Two strategies have been devised for simulation of background hits: The first one is based on simulation provided by the accelerator group. The second one, currently being developed, is based on random trigger events. An alternative algorithm on the electromagnetic calorimeter reconstruction is being investigated for the higher background environment. The conditions database is being constructed with emphasis on scalability. The distributed computing system of Belle II is also based on DIRAC. Large scale Monte Carlo productions and cosmic ray data replication have been conducted to test the production system. The global network has been tested between Japan, US, and European collaborators with data challenges and the traffic speed is verified. More data challenges are planned for further testing of network configuration.

3. Algorithms and software languages

Developments in computing algorithms and software languages affect the way physicists conduct analysis. Some of these effects could be profound and overhaul the entire analysis frame. In this section, two interesting topics involving recent developments in the field are discussed.

3.1 Deep learning

Usage of artificial neural network (ANN) is not new to particle physics. It has been gaining popularity and became a main analysis tool. The tool is used to distinguish signal events from background events in a sample or used to identify particle types. The inputs to an ANN are usually analysis features extracted or constructed from a raw data set.

The concept of multi-layer (deep) neural network is not new, but it could not implemented properly due to limited computing resources. However, advent of big data and rapid development in computing architecture such as GPU's changed the situation, make the application of multi-layer neural network possible. Convolutional neural network (CNN) is a type of deep neutral network inspired by visual cortex of animals. CNN can be used to recognize images, as shown in Figure 1.

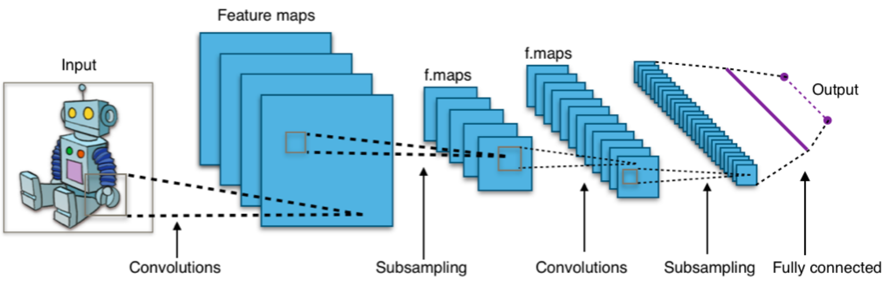

Figure 1: An example of CNN architecture. By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374

In particle physics experiments, deep neural networks or CNN can be used to learn features from raw data. There has been a surge of interest in their applications, especially in the neutrino experiments [12]. A case study was conducted on the application of CNN to the Daya Bay data, where CNN could classify raw data events into different physics categories with 97 % accuracy [13].

3.2 Metaprogramming

Metaprogramming is a technique to write computer programs using another program as its input data. For example, C++ template is metaprogramming. Another example is Nim [15] . According to the official website, Nim can compile on various platforms including Windows, Linux, and Mac OS X, and it can generate C++ codes. And bindings to popular software packages such as GTK2[1] [16], the Windows API, OpenCL[2] [25], Python, mySQL are provided.

QEX (Quantum EXpressions) [14] is a new framework for lattice field theories, written mostly in Nim. The tests of QEX on the staggered conjugate gradient (CG) algorithm showed good performance with an Intel Knights Landing (KNL) architecture[3], Intel Xeon Phi CPU 7210. The performance tests were performed with variations in lattice volumes, number of threads, and precision.

4. Infrastructure

New models in computing architecture and availability of various online resources are changing how particle physics experiments process data. Innovative strategies are developed, such as LArSoft[4], a software collaboration for liquid argon time projection chamber detectors [17]. Large institutes such as Femilab are deeply involved in providing toolkits and services for smaller experiments, who do not have enough resources to develop their own framework [18]. In this section, several examples of emerging technologies used by particle physics experiments are discussed.

4.1 Cloud computing

The time when one or a few institutes could satisfy all the computing needs of a particle physics experiment had ended several years ago. The amount of resources required by experiments are huge and innovative strategies should be applied to fulfill the requirements. Three major strategies have emerged recent years, which are listed as follows:

- **High performance computing center (HPC)**
  An institute with large computing resources. Usually funded by government grants. The access to the facility is controlled by the funding agency or by the user committee.

- **Grid computing**
  A virtual organization or a collaboration of sites to grant access of resources to users over the network. The size of sites can be diverse, from one research group to a HPC.

[1] GIMP toolkit

[2] Open Computing Language

[3] The KNL architecture belongs to Many Integrated Core (MIC) category and specifically designed for supercomputers or workstations.

[4] Liquid Argon Software

- **Cloud computing**
  There exist two types of cloud computing: Community cloud and commercial cloud. Community clouds are similar to grid computing. Resources provided by commercial clouds such as Amazon Web Services (AWS) can be rented by either set or spot price.

Femilab has a HPC on-site and is a Tier-1 host for the CMS experiment. In addition, Femilab has started to provide access to cloud services as a gate interface in 2015 [19]. This project is named *HEPCloud* and based on the recommendation by the 2014 Particle Physics Project Prioritization Panel (P5) [20]. The amount of resources provided by a HPC is fixed and inflexible, which is not adequate to satisfy ever increasing and fluctuating demands by physics experiments. On the other hand, cloud services can be rented by a flexible schedule, which can fill the gap between the user demands and the fixed resources at a HPC. Another advantage of cloud computing is that its price has been falling, becoming more attractive to renters.

4.2 Utilization of Graphic Process Units (GPU)

The graphic process units (GPU), which have been developed to handle intensive visualization on computer screens, are also excellent in executing parallel computational operations. These units are used to run multitudes of relatively simple tasks, such as calculating matrix operations. Central processing units (CPU) are still better for handle complex algorithms. There are GPU's specifically developed for scientific purposes, such as Nvidia Tesla K40. One of the limiting factors of the GPU operation is the speed of input/output (IO) data transfer. Therefore, a motherboard with PCIe[5] version 3.0 is preferred [21]. For the GPU software, there are two major players: CUDA[6] and OpenCL. CUDA is the proprietary software for Nvidia units [24], while OpenCL is an open source software compatible with many GPU's including Nvidia units.

GPU's have been used for pattern recognition and reconstruction by several experiments. For example, the NA62 experiment used the units for the ring-Cerenkov detector [22] and the LHCb experiment used them for track and vertex reconstruction [23]. The Muon g-2 experiment is constructing a DAQ system consisting of multi-core CPU's and GPU's for the 2017 run at Fermilab [21].

4.3 GeantV: Vectorized Geant

Vectorization is an algorithm jargon, which means applying similar operations as a group, rather than applying one by one in a sequence. A similar concept can be employed to update the Geant4 [7] simulation package [27]. This update project is named as GeantV or Vectorized Geant. The main algorithm of the Geant4 library as follows: During the simulation of a detector, the trajectory of a particle inside the detector proceeds step by step. When there are many particle inside the detector, the simulation of particle trajectories are done one by one. The basic idea of Geant4 is to transport particles in groups instead. Grouping is done either by geometry volume or by physics processes. This grouping procedure is acted as an overhead for the performance of the library. To get benefit from vectorization, the overhead should not be larger than the gains obtained

[5] Peripheral Component Interconnect Express
[6] Compute Unified Device Architecture
[7] acronym: for GEometry ANd Tracking

by vectorization [26]. The update project also considers parallel operations seriously. The new library should be comfortable with various types of processing units including CPU's, GPU's, Intel Xeon Phi, Intel Atom, etc.

Three components of the library have developed and tested: A multi-thread scheduler which controls the particle baskets, a vectorized geometry library and navigator, and a vectorized Compton scattering process and a tabulated physics list. The test on these components show minimal overhead and improved performance, generating optimistic expectation for the new library [26].

5. Data management

In this section, a couple of select topics on management of large data sets are summarized, with emphasis on the LHC experiments.

5.1 Utilization of the high level trigger (HLT) data

In large scale experiments, typically, a selected raw event is designated as a physics event based on two level trigger system. The first level (Level 1) consists of hardware trigger modules to filter out unwanted background events. Subsequently, the filtered events are fed into the high level trigger (HLT) system with dedicated algorithms to filter in only the events suitable for physics analysis. The final event candidates are saved as physics data for offline reconstruction and further analysis later.

With the LHC experiments, it became difficult to analyze all the physics event candidates due to limited computing resources. One strategy to go around this obstacle is to analyze HLT trigger samples online. When an event is found as a discovery, it may be stored in a tape, but will not be saved for offline analysis. The CMS collaboration started using the strategy, *Data Scouting*, for jet events with high energy calorimeter deposits during Run I. For Run II, they expanded the strategy to include events with the particle flow topology, and have produced several promising physics results [28].

The LHCb collaboration developed a similar scheme where they emphasize the analysis on the HLT data sets. They built three streams at the HLT level: *Full Stream* in which events are saved for full offline reconstruction. *Turbo Stream* in which events are only subject to online reconstruction. Since no raw data is saved, these events cannot be reprocessed later. Finally, *Calibration Stream* in which events are subject to both online and offline reconstruction. The concept of Turbo Stream was developed during Run II and it will be used as the only analysis stream for coming Run III [29].

5.2 Analytics platform for distributed computing

Nowadays, data sets of one experiment would be saved and accessed over globally distributed network of storage centers. The formats of data sets are various, from physics data sets to network monitoring status. It became important to monitor the status of the distributed data, to improve the access performance and to create new distributed network tools. The ATLAS collaboration constructed an analytics platform, which can be maintained by a small number of people [30]. Modern big data tools are utilized to construct the system: Hadoop [31], Flume [32], Elasticsearch

[33], Kibana [34], Jupyter [35], and others. It was shown that the platform is easy to use and can provide a base for alerting and reporting services.

6. Conclusion

We are in the era of big data. The overall structure in the computing and data handling sector is changing rapidly. Researchers have dealt with huge amount of data sets already, and more sets are expected to be generated. The data traffic on the global research network is heavy, demanding efficient strategies and wider collaboration. Next generation technologies are being introduced continuously, making long term planning more complicated.

The particle physics researchers have responded to these challenges rigorously: Detailed plans are employed to control needs for computing resources. Many new ideas are incorporated and adopted by the researchers. There have been efforts to share common software/computing toolkits among relevant experiments to increase stability and efficiency in operation. All these efforts have been utilized successfully to build fast responding data analysis frameworks for particle physics experiments, and have made it possible to produce prominent physics results.

Acknowledgments

The author is supported by the National Research Foundation of Korea (NRF) Grant No. 2016R1D1A1B02012900. The author also appreciates support from the Supercomputing Center/Korea Institutes of Science and Technology Information with their facilities (supercomputing, GSDC, and KREONET) including technical support.

References

[1] G. Cerminara [The CMS Collaboration], *CMS operations for Run II: Preparation and commissioning of the offline infrastructure*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)169.

[2] G. Petrucciani, A. Rizzi, and C. Vuosalo [The CMS Collaboration], *Mini-AOD: A new analysis data format for CMS*, in proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015), *JPCS* **664** (2015) 072052.

[3] G. Cerminara and C. van Besien [The CMS Collaboration], *Automated workflows for critical time-dependent calibrations at the CMS experiment*, in proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015), *JPCS* **664** (2015) 072009.

[4] M. Rovere [The CMS collaboration], *The Data Quality Monitoring Software for the CMS experiment at the LHC*, in proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015), *JPCS* **664** (2015) 072030.

[5] *ILC Soft*, http://ilcsoft.desy.de/portal.

[6] Advanced European Infrastructures for Detectors at Accelerators (AIDA), *Detector Description for HEP; AIDA Common Software Tools*, http://aidasoft.web.cern.ch/DD4hep.

[7] N. Nikiforou [The CLICdp and ILD Collaborations], *Linear collider software and computing*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)172

[8] C. Grefe et al., [THE CLICdp Collaboration] *ILCDIRAC, a DIRAC extension for the Linear Collider community*, in proceedings of the 20th International Conference on Computing in High Energy and Nuclear Physics (CHEP2013), *JPCS* **513** (2014) 032077.

[9] Advanced European Infrastructures for Detectors at Accelerators (AIDA), *Detector Description for HEP; AIDA Common Software Tools*, http://aidasoft.web.cern.ch/DD4hep.

[10] A. Tsaregorodtsev et al., [THE CLICdp Collaboration] *DIRAC3 - the new generation of the LHCb grid software*, in proceedings of the 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09), *JPCS* **219** (2010) 062029.

[11] M. Schram [The Belle II Collaboration], *Overview and highlights of the Belle II computing and software*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)173.

[12] A. Farbin, *Event reconstruction with deep learning*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)180.

[13] E. Racah et al., *Revealing fundamental physics from the Daya Bay neutrino experiment using deep neural networks* [arXiv:1601.07621]

[14] X.-Y. Jin and J. C. Osborn, *QEX: A framework for lattice field theories*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)187. Also documents at https://github.com/jcosborn/qex.

[15] A. Rumpf and contributors, *Nim Programming Language*, http://nim-lang.org, 2015.

[16] The GTK+ Team, *The GTK+ project*, https://www.gtk.org.

[17] R. Pordes and E. Snider [The LArSoft Collaboration], *Liquid argon software tookkits (LArSoft): Goals, status, and plan*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)182. Also documents at http://larsoft.org.

[18] K. R. Herner, *The FIFE project at Fermilab: Computing for experiments*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)176

[19] B. Holzman, *Femilab HEPCloud: An elastic computing facility for high energy physics*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)175.

[20] Particle Physics Project Prioritization Panel (P5), *Building for discovery; Strategic plan for U.S. particle physics in the global context*, DOE Office of Science and NSF, May 2014.

[21] W. Gohn, *Data acquisition with GPUs: The DAQ for the Fermilab Muon g-2 experiment*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)174.

[22] R. Ammendola et al., *GPU-based real-time triggering in the NA62 experiment* [arXiv:1606.04099].

[23] A. Badalov et al., *LHCb GPU acceleration project*, *J. Instrum.* **11** (2016) P01001.

[24] J. Nickolls et al., *Scalable parallel programming with CUDA*, *ACM Queue* **6** (2008) 40.

[25] Khronos Group, *OpenCL - The open standard for parallel programming of heterogeneous systems*, https://www.khronos.org/opencl.

PoS(ICHEP2016)011

[26] P. Canal, *GeantV: From CPU to accelerators*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)177.

[27] S. Agostinelli et al., *GEANT4 - a simulation toolkit*, *Nucl. Instr. Meth. Phys. Res. A* **506** (2003) 250; J. Allison et al., *Geant4 developments and applications*, *IEEE Trans. Nucl. Sci.* **53** (2006) 270.

[28] D. Anderson [The CMS Collaboration], *Data Scouting in CMS*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)190.

[29] A. Falabella [The LHCb Collaboration], *LHCb distributed computing in Run II and its evolution toward Run III*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)191.

[30] I. Vukotic, G. Robert, B. Lincoln [The ATLAS Collaboration], *Getting the most from distributed resources: An analytics platform for ATLAS computing services*, in proceedings of the 38th International Conference on High Energy Physics, PoS(ICHEP2016)192.

[31] The Apache Software Foundation, *Welcome to Apache™ Hadoop®!*, http://hadoop.apache.org.

[32] The Apache Software Foundation, *Welcome to Apache Flume™*, http://flume.apache.org.

[33] Elasticsearch, *Elasticsearch: RESTful, Distributed Search & Analytics*, https://www.elastic.co/products/elasticsearch.

[34] Elasticsearch, *Kibana: Explore, Visualize, Discover Data*, https://www.elastic.co/products/kibana.

[35] Project Jupyter, *Project Jupyter*, https://jupyter.org.