T 62: DAQ NN/ML – GRID II

Time: Wednesday 15:50–17:20 Location: HSZ/0301

T 62.1 Wed 15:50 HSZ/0301

Development of machine-learning based topological algorithms for the CMS level-1 trigger — •Finn Labe, Johannes Haller, Gregor Kasieczka, Artur Lobanov, and Matthias Schröder — Institut für Experimentalphysik, Universität Hamburg

At the CMS experiment, a two-level trigger system is used to decide which collision events to store for later analysis. Due to the large fraction of non-interesting, low-energy collisions, currently used triggers often rely on momentum thresholds, only selecting events containing at least one highly-energetic object. In many cases, such as searches for di-Higgs production, this can substantially reduce the signal efficiency of the trigger selection. Targeting the upgraded CMS detector for the High Luminosity LHC, novel techniques are presented that utilize machine learning inside the first hardware layer of the trigger, which is based on FPGAs. Instead of individual objects, these triggers rely on the full event topology to select previously inaccessible events. The usage of these algorithmns in the context of the second trigger layer and offline analysis is studied.

T 62.2 Wed 16:05 HSZ/0301

LHCb's Topological Trigger in Run 3 — Johannes Albrecht[1], Gregory Max Ciezarek[2], Blaise Delaney[3], Niklas Nolte[3], and •Nicole Schulte[1] — [1]TU Dortmund University, Dortmund, Germany — [2]CERN, Switzerland — [3]Massachusetts Institute of Technology, Massachusetts, USA

The data-taking conditions expected in Run 3 of the LHCb experiment present unprecedented challenges for the software and computing systems. Consequently, the LHCb collaboration is pioneering an entirely software-based trigger system to efficiently manage the increased event rate. The beauty physics programme of LHCb is heavily dependent on topological triggers. These are dedicated to the inclusive selection of b-hadron candidates based on the characteristic beauty decay topology and their expected kinematic properties.

We present the Run 3 implementation of the topological triggers using Lipschitz monotonic neural networks. This architecture offers robustness under varying detector conditions and sensitivity to long-lived candidates, opening the possibility of discovering New Physics at LHCb.

T 62.3 Wed 16:20 HSZ/0301

APEL accounting with AUDITOR — Michael Böhler, Stefan Kroboth, •Dirk Sammel, and Markus Schumacher — Albert-Ludwigs-Universität Freiburg

Institutions that are part of the Worldwide LHC Computing Grid (WLCG) offer computing resources to analyse the data recorded by experiments at the Large Hadron Collider (LHC), and to produce simulated samples. An important task is the accounting of the utilized resources. For this, the used CPU time among other information is reported to the APEL (Accounting Processor for Event Logs) server, which publishes it on the Accounting Portal of the European Grid Infrastructure (EGI).
In Freiburg, for example, the accounting to APEL is performed by ARC CE (Advanced Resource Connector Compute Element). The compute element accepts compute jobs from WLCG and submits them to the local batch system. In addition it can also report the utilized resources. An alternative approach, independent from ARC CE, is the use of AUDITOR (Accounting Data Handling Toolbox For Opportunistic Resources). AUDITOR uses a "collector" to gather job information from the local batch system. These "job records" are stored in a PostgreSQL database, which can be accessed by plug-ins.
This talk first gives a short overview of the accounting system of the ARC software, and then presents an AUDITOR plug-in for accounting to APEL. The plug-in receives job records from the AUDITOR database, formats them according to APEL specifications, and submits them to the APEL server.

T 62.4 Wed 16:35 HSZ/0301

Accounting opportunistic resources with AUDITOR — •Stefan Kroboth, Michael Boehler, Dirk Sammel, and Markus Schumacher — Albert-Ludwigs-Universität Freiburg

The increasing computational demand and concerns about energy efficiency in high performance/throughput computing are driving forces in the search for more efficient ways to utilize available resources. A measure for achieving high efficiency is the sharing of idle resources of under-utilized sites with fully occupied sites. The software COBalD/TARDIS can automatically, transparently, dynamically and opportunistically integrate and disintegrate such resources. However, sharing resources also requires accounting. In this work we present AUDITOR (AccoUnting DatahandlIng Toolbox for Opportunistic Resources), a flexible and extensible accounting system that is able to cover a wide range of use cases and infrastructure. AUDITOR gathers accounting data via so-called collectors which are designed to monitor batch systems, COBalD/TARDIS, cloud schedulers or other sources of information. The data is stored in a database and provided to so-called plugins, which take an action based on accounting records. Actions can range from creating a bill, computing the CO_2 footprint, adjusting parameters of a service (i.e. priorities in a batch system) to forwarding accounting information to other accounting systems. Depending on the use case, one simply selects a suitable collector and plugin from a growing ecosystem of collectors and plugins. To facilitate the development of collectors and plugins for yet uncovered use cases by the community, libraries for interacting with AUDITOR are provided.

T 62.5 Wed 16:50 HSZ/0301

Containerization of the ATLAS HammerCloud setup — •Benjamin Rottler, Michael Böhler, and Markus Schumacher — Universität Freiburg

HammerCloud (HC) is a testing service and framework for continuous functional tests, on-demand large-scale stress tests, and performance benchmarks. It checks the computing resources and various components of distributed systems with realistic full-chain experiment workflows.

The current deployment setup based on RPMs allowed a stable deployment and secure maintenance over several years of operations for the ATLAS and CMS experiments. However, the current model is not flexible enough to support an agile and rapid development process. Furthermore, we wanted to be more independent of software versions that are provided by the package manager of the host system.

Therefore, we have decided to use a solution based on containerization, and switched to industry-standard technologies and processes. Having an "easy to spawn" instance of HC enables a more agile development cycle and easier deployment. With the help of such a containerized setup, CI/CD pipelines can be integrated easily into the automation process as an extra layer of verification. Furthermore, the container-based setup allows for quick onboarding of new team members, as developers can now work locally with a quick turnaround without the need to set up a production-like environment first.

In this talk we present the container-based setup for HammerCloud and discuss the process that led to our containerized solution.

T 62.6 Wed 17:05 HSZ/0301

Sapphire - Small-file aggregation for the dCache tape interface — •Svenja Meyer[1], Krishnaveni Chitrapu[3], Dmitry Litvintsev[2], Paul Millar[1], Tigran Mkrtchyan[1], Lea Morschel[1], Albert Rossi[2], and Marina Sahakyan[1] — [1]Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany — [2]Fermi National Accelerator Laboratory, Batavia, USA — [3]National Supercomputer Center, Linköping University, Sweden

dCache is an open source distributed storage system used to manage and store scientific data in the scale of hundreds of petabyte. Archiving data on tertiary storage, for example tape, is a main feature of this software. Unfortunately the performance of writing data to tape decreases for small sized files, which are produced more and more by experiments.

To circumvent this problem, *Sapphire*, an advancement of *SmallFiles* for *dCache*, was developed. Working as a plugin for *dCache*, these small files are bundled into bigger archives without needed intervention by the user. Flushing files to tape as well as staging them back works transparently to the user.