



Contribution ID: 330

Type: **Poster presentation**

A GPU offloading mechanism for LHCb

Monday, 14 October 2013 15:00 (45 minutes)

The LHCb Software Infrastructure is built around a flexible, extensible, single-process, single-threaded framework named Gaudi. One way to optimise the overall usage of a multi-core server, which is used for example in the Online world, is running multiple instances of Gaudi-based applications concurrently. For LHCb, this solution has been shown to work well up to 32 cores and is expected to scale up a bit further.

The appearance of many-core architectures such as GPGPUs and the Intel Xeon/Phi poses a new challenge for LHCb. Since the individual data sets are so small (about 60 kB raw event size), many events must be processed in parallel for optimum efficiency. This is, however, not possible with the current framework, which allows only a single event at a time. Exploiting the fact that we always have many instances of the same application running, we have developed an offloading mechanism, based on a client-server design.

The server runs outside the Gaudi framework and thus imposes no additional dependencies on existing applications. It asynchronously receives event data from multiple client applications, coalesces them, and returns the computed results back to callers. While this incurs additional inter-process communication overhead, it allows co-processors to be used within the existing large framework with minimal changes. We present our solution and describe the achieved performance, both at the single-instance and the server levels.

Summary

Primary authors: ZVYAGIN, Alexander (Fakultaet fuer Physik-Ludwig-Maximilians-Univ. Muenchen); BADALOV, Alexey (La Salle - Ramon Llull University); CAMPORA PEREZ, Daniel Hugo (CERN)

Co-authors: NEUFELD, Niko (CERN); VILASIS CARDONA, Xavier (La Salle - Ramon Llull University)

Presenter: CAMPORA PEREZ, Daniel Hugo (CERN)

Session Classification: Poster presentations

Track Classification: Software Engineering, Parallelism & Multi-Core