Contribution ID: **83** Type: **Parallel talk**

Multicore in Production: Advantages and Limits of the Multi-process Approach.

Thursday 8 September 2011 14:00 (25 minutes)

The shared memory architecture of multicore CPUs provides HENP developers with the opportunity to reduce the memory footprint of their applications by sharing memory pages between the cores in a processor. ATLAS pioneered the multi-process approach to parallelizing HENP applications. Using Linux fork() and the Copy On Write mechanism we implemented a simple event task farm which allows to share up to 50% memory pages among event worker processes with negligible CPU overhead.

By leaving the task of managing shared memory pages to the operating system, we have been able to run in parallel large reconstruction and simulation applications originally written to be run in a single thread of execution with little to no change to the application code. In spite of this, the process of validating athena multi-process for production took ten months of concentrated effort and is expected to continue for several more months. In general terms, we had two classes of problems in the multi-process port: merging the output files produced by the event workers, and assuring the reproducibility of the results, especially of Montecarlo simulations, when running with different configurations, in particular with different number of event workers.

Besides validating the software itself, an important and time-consuming aspect of running multicore applications in production is to configure the production system to handle multicore jobs. This entails defining multicore batch queues, where the unit resource is not a core, but a whole computing node; monitoring the output of many event workers; and adapting the job definition layer to handle computing resources with very different event throughputs (depending on the number of cores used).

To conclude, we will present scalability and memory usage studies, based on data gathered both on dedicated hardware and on ATLAS production nodes. From these it should become apparent that the most promising development to improve performance will be to transition from a simple, flat, event task farm in which all processes handle events independently to a task farm with specialized worker processes, which will be in charge of event I/O. This approach will further reduce the memory footprint of our multicore applications, and at the same time address the issue of merging event worker outputs, at the cost of some increase in the complexity of the ATLAS core software.

Authors: CALAFIURA, Paolo (LBL); TSULAIA, Vakhtang (LBL)

Co-authors: WASHBROOK, Andy (Univ of Edinburgh); LEGGETT, Charles (LBL); LESNY, David (Univ of Illinois Urbana-Champaign); SMITH, Douglas (SLAC); SEVERINI, Horst (Univ of Oklahoma); JHA, Manoj Kumar (INFN Napoli); TATARKHANOV, Mous (LBL); VAN GEMMEREN, Peter (ANL); SNYDER, Scott (BNL); BINET, Sebastien (LAL); LAVRIJSEN, Wim (LBL)

Presenter: TSULAIA, Vakhtang (LBL)

Session Classification: Thursday 08th - Computing Technology for Physics Research

Track Classification: Track 1: Computing Technology for Physics Research