



Contribution ID: 61

Type: **Oral presentation to parallel session**

Parallelization of Common HEP patterns with PyPy (cancelled)

Monday 14 October 2013 16:50 (21 minutes)

The Python programming language brings a dynamic, interactive environment to physics analysis. With PyPy high performance can be delivered as well, when making use of its tracing just in time compiler (JIT) and cppy for C++ bindings, as cppy is able to exploit common HEP coding patterns. For example, ROOT I/O with cppy runs at speeds equal to that of optimized, hand-tuned C++.

Python does not, however, offer an easy way to exploit computational parallelization, because of the global interpreter lock (GIL). In PyPy this could be solved using software transactional memory (STM). With STM in place, the patterns in cppy can be employed to automatically parallelize user code when the interpreter deems them, and the underlying libraries, safe. The work described in this paper takes the existing ROOT I/O patterns in cppy and shows how they can be parallelized using STM.

Primary author: LAVRIJSEN, Wim (Lawrence Berkeley National Lab. (US))

Presenter: LAVRIJSEN, Wim (Lawrence Berkeley National Lab. (US))

Session Classification: Software Engineering, Parallelism & Multi-Core

Track Classification: Software Engineering, Parallelism & Multi-Core