



Contribution ID: 346

Type: Poster

GoCxx: a tool to easily leverage C++ legacy code for multicore-friendly Go libraries and frameworks

Thursday, May 24, 2012 1:30 PM (4h 45m)

Current HENP libraries and frameworks were written before multicore systems became widely deployed and used.

From this environment, a 'single-thread' processing model naturally emerged but the implicit assumptions it encouraged are greatly impairing our abilities to scale in a multicore/manycore world.

Writing scalable code in C++ for multicore architectures, while doable, is no panacea. Sure, C++11 will improve on the current situation (by standardizing on `std::thread`, introducing lambda functions and defining a memory model) but it will do so at the price of complicating further an already quite sophisticated language. This level of sophistication has probably already strongly motivated analysis groups to migrate to CPython, hoping for its current limitations with respect to multicore scalability to be either lifted (Grand Interpreter Lock removal) or for the advent of a new Python VM better tailored for this kind of environment (PyPy, Jython,...)

Could HENP migrate to a language with none of the deficiencies of C++ (build time, deployment, low level tools for concurrency) and with the fast turn-around time, simplicity and ease of coding of Python ?

This paper will try to make the case for Go - a young open source language with built-in facilities to easily express and expose concurrency - being such a language.

We will first present a status update on go-gaudi, a framework written in Go loosely modeled after the C++ framework Gaudi used by two LHC experiments, and how its event loop was modified to expose more concurrency.

Then, benchmarks fed with data flows extracted from current C++ frameworks and with different toy-components (thread-safe/non-thread-safe, I/O bound, CPU bound, ...) will be discussed.

Finally, we will introduce GoCxx, a tool leveraging gcc-xml's output to automatize the tedious work of creating Go wrappers for foreign languages, a critical task for any language wishing to leverage legacy and field-tested code.

We will conclude with the first results of applying GoCxx to real C++ Gaudi components, effectively enabling go-gaudi with LHC know-how.

Student? Enter 'yes'. See <http://goo.gl/MVv53>

no

Summary

We present GoCxx, a tool to automatize the wrapping of C++ libraries, and its impact on next-generation parallel event processing frameworks.

Primary author: Dr BINET, Sebastien (LAL/IN2P3)

Presenter: Dr BINET, Sebastien (LAL/IN2P3)

Session Classification: Poster Session

Track Classification: Software Engineering, Data Stores and Databases (track 5)