Benchmarking and optimising large scale parallel workflows

J. P. Hernández, G. Amadio

ROOT
Data Analysis Framework
https://root.cern
Run on olshwep04.cern.ch (Intel Xeon E5-2698, dual socket)
- 16 cores/socket, 64GB RAM, 2.3GHz base, 3.6GHz boost

Run with ROOT nightly build from Gentoo prefix in CVMFS
- /cvmfs/sft.cern.ch/lcg/contrib/gentoo/linux/x86_64/startprefix

ROOT compiled with -O2 -g -gdwarf -fno-omit-frame-pointer

Tutorials: dimuon, higgs

Profiling tools: VTune and perf
Dimuon Analysis

Matching muon pairs and producing an histogram of the dimuon mass spectrum showing resonances up to the Z mass
Core usage seems to be OK
Quantifying NUMA effects

- Pinning is faster
- More threads, lower execution time
- Performance not scaling properly
Time and speedup comparison

- SSD & NVME the fastest
- Pinning has a big influence

- Best speed-up: SSD & NVME
Jitted vs compiled

- Compiled runs faster to a certain point
- Jitted code has a bigger speed-up

**Compiled & jitted execution times**

**Compiled & jitted speed-up**
Higgs Analysis

Reconstructing the Higgs boson decaying to two Z bosons from events with four leptons
Pinning vs no pinning

- Pinning is faster
- More threads, lower execution time

- Performance not scaling properly (even worse than dimuon analysis)
Some contributions

- Scalability issues detected
- “Problematic” code identified
- NUMA effects are important
- Pull request with the new compiled tutorial
Thank you!

Any questions?

jesus.perales.hernandez@cern.ch