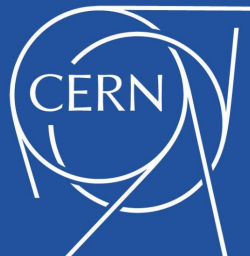




Profiling Tools@LHCb

Joint WLCG & HSF Workshop
28/03/2018

B.Couturier, M.Hadji, F.Lemaitre, S.Ponce, M.Szymanski



Profiling tools landscape

Measuring performance and resource use is crucial to improving software

Many profiling tools are available for this purpose:

[gprof](#), [gperftools](#), [igprof](#), [jemalloc](#), [valgrind/callgrind](#),
[Intel® VTune™ Amplifier](#), [linux perf](#) (c.f. Andrea's talk for more details)

And there is a reason for this:

- **Different type of profiles:** memory, call stack, CPU counters
- **Different methods to gather measures:** sampling, emulation
- **Vendor specific tools:** e.g. Intel vtune

No reasons for this to change:

- **Large variety of needs**
- **Evolving environments**
- **New architectures...**

LHCb experience with the current tools

The tools do work well BUT it's not always easy to get started

- Installation of the tools not always easy (e.g. some tools require licenses)
- Sometimes requires physical machines (e.g. to get CPU counters)
- And root access for specific kernel modules
- Need integration within the Frameworks (e.g. to measure one specific algorithm)
- Not all measures are easy to extract (e.g. use of SIMD vectors...)

Medium long term follow up (with continuous integration) requires work to setup

- Comparing profiles is not always so easy
- LHCb's Performance and Regression testing framework (LHCbPR) uses Callgrind, perf and igprof
- Had to write code to gather/publish profiling results

Outlook

- We need to be flexible and allow using all the profiling tools available
- Using them **should** be a no-brainer (to make sure they are used from the start of developments)
 - Installed by default
 - Integration in event processing framework really helpful
 - Better documentation
- We need to continuously measure performance
 - And publish/keep the results

Would common work on any of those points be helpful?

