What CPUs to support, at what cost

Helge Meinhard / CERN-IT and
Mattias Wadenstein / Umeå University
Grid Deployment Board 15-Jan-2020

This Presentation: A meta-Proposal

- Not a firm proposal to take a decision
- Rather intended to trigger thoughts, discussions and work needed to shape a firm proposal

Problem Statement (1)

- CPU capacity accounts for about 1/3 of expenditures in WLCG
  - Storage is the main expenditure, and is being looked at in view of optimisations e.g. in the context of DOMA
- CPU Server park changing constantly
  - New powerful machines with new CPU features coming in, old machines being retired
    - Machine lifetime increasing, but that doesn't change the principle

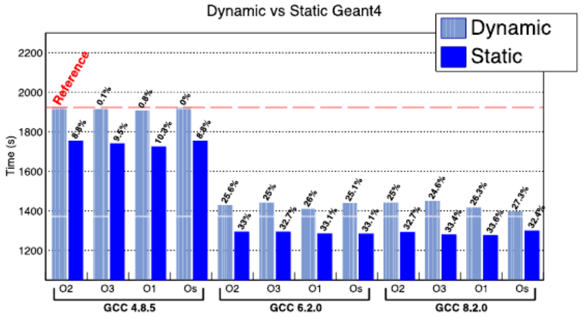
Problem Statement (2)

- Workload presumably still compiled very conservatively (gcc –O2?)
  - Physics verification labour-intensive
  - Early days of WLCG: Desire to be as inclusive as possible
- The world (and with it x86) has moved on:
  - 2013: AVX2, FMA3, TSX, BMI1, and BMI2
  - 2015: AVX512
  - …

Why Bother?

- Numerous indications of potential performance increases when building workload differently and using modern processor features
  - E.g. CHEP 2019: C. Marcon (Lund U): Impact of different compilers and build types on Geant4 simulation execution time `https://indico.cern.ch/event/773049/contributions/3473317/`
  - Or GDB November 2019: A. Naumann (CERN) on ROOT performance improvements with AVX 2 `https://indico.cern.ch/event/739884/contributions/3632250/`

Static vs dynamic performance with **full detector geometry**



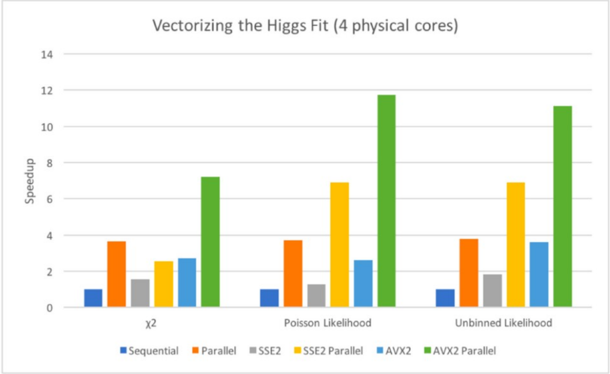
- The computations were carried out **on CERN machine** considering 5000 initial events and using 4 threads. The computation was repeated 3 times for each configuration.
- The static approach, for all the GCC versions, reduces the execution time by more than **10%** in some cases.
- Regardless of the build approach, switching from GCC 4.8.5 to GCC 6.2.0 and GCC 8.2.0 results in an average of **30%** improvement in the execution time.
- A static build with GCC 8.2.0 leads to an improvement of almost **34%** with respect to the default configuration (GCC 4.8.5, dynamic, O2).
- The different GCC optimizations do not seem to have visible effects on the execution time.

LUND UNIVERSITY

5

mov %avx2 ?

Performance of the Higgs Fit on your laptop



Xavi Valls, https://indico.cern.ch/event/567550/contributions/2631612/

15 Axel Naumann axel@cern.ch: ROOT, GDB 2019-11-20

The Meta-Proposal

- This issue should be addressed systematically
  - Common recommendation about CPU features to use, compilers, flags, ...
- A one-off is close to useless – need to agree on a regular activity
  - Once per year? (Natural coincidence with resource cycle)
- If reactions positive, work out a detailed proposal for such regular review, and conduct the first cycle

(Some of the) Questions to Address

- Is our assumption right about the experiments' current build practice?
- Are builds using newer CPU features, other compilers or more aggressive optimisation feasible?
  - Effort for validation the same, or increases?
- What workload to use in order to measure potential improvements?
  - Workloads submitted to benchmarking WG could be a good start
- What is the impact on accounting, resource requests, pledges, …?
  - My first assumption is no change to HS06 baseline (or successor) – the impact would hence be a reduced need for resources

(More) Questions to Address

- Who does the work? Who drives it regularly?
  - Obviously some common points with benchmarking, cost modelling, Markus' performance team in CERN-IT, ...
- How to decide on the "cut-off"? Who decides?

Comments, Reactions?

- Should we try and go in this direction, or forget about the idea immediately?

- Opinions, interest? Contact any one of us (Mattias, Helge)