

AVX2 on the WLCG infrastructure

Collecting CPU flags

Jaroslava Schovancová, Julia Andreeva, Maarten Litmaath
(CERN IT)

Motivation: AVX2 and ROOT

- Axel Naumann's talk about AVX2 for ROOT
 - November 2019 GDB <https://indico.cern.ch/event/739884/>

mov %avx2 ?

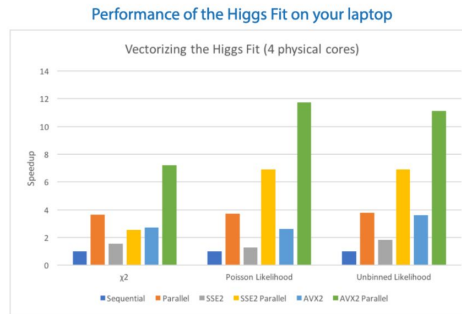
mov %avx2 ?

Grid == AVX2!

- Faster analyses thanks to vectorized hot loops, even though global benefits from vectorization unlikely
- Cannot build ROOT for anything but SSE2 == x86_64
- Would benefit significantly from general AVX2 availability!
 - Which exists in reality
 - But we cannot count on it
- Architecture-specific plugin libraries: validation concern for experiments

14

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



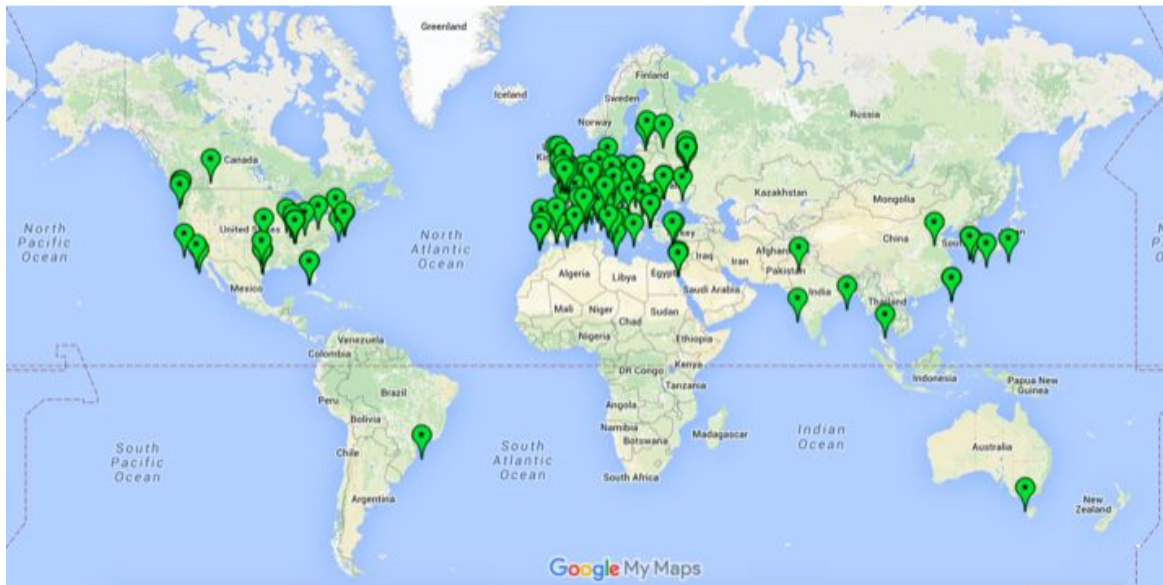
- ROOT (and experiments' frameworks) need to target smallest common denominator
- Currently, lots of under-used silicon
- Plans for AVX2 requirement for grid nodes?

16

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

⇒ What portion of WLCG resources supports AVX2???

WLCG resources



- Let's start with ATLAS (~50 % sites) and CMS (~30 % sites)

Expectations...

- AVX2 is around for 5+ years, could be fairly available, no?

- AVX since 2011, AVX2 since 2016

- https://en.wikipedia.org/wiki/Advanced_Vector_Extensions

Advanced Vector Extensions (AVX, also known as Sandy Bridge New Extensions) are extensions to the x86 instruction set architecture for microprocessors from Intel and AMD proposed by Intel in March 2008 and first supported by Intel with the Sandy Bridge processor shipping in Q1 2011 and later on by AMD with the Bulldozer processor shipping in Q3 2011. AVX provides new features, new instructions and a new coding scheme.

AVX2 expands most integer commands to 256 bits and introduces fused multiply-accumulate (FMA) operations. AVX-512 expands AVX to 512-bit support using a new EVEX prefix encoding proposed by Intel in July 2013 and first supported by Intel with the Knights Landing processor, which shipped in 2016.

- In 2014: <https://software.intel.com/en-us/articles/how-intel-avx2-improves-performance-on-server-applications>

- **Well...**

1st approach: a HammerCloud test

- Set up a template to check: `grep avx2 /proc/cpuinfo`
 - Low frequency test, run at analysis sites
- Collected info over Xmas, small sample of resources
 - Sample of 25k jobs @ 13.7k distinct WNs
 - Failed jobs @ 4.9k distinct WNs

... **~36 % probed WNs did not have 'avx2' in flags in /proc/cpuinfo**

2nd approach: collect flags

- Can we collect all flags from /proc/cpuinfo in all jobs?
 - `grep flags /proc/cpuinfo`
 - Enhanced Pilot wrapper, now every ATLAS job collects flags
 - **Thanks** to Peter Love, Xin Zhao, Chris Lee, David Cameron, Frank Berghaus, Nicolo Magini!
- Sparse sample over few days, **62.5k unique hostnames**
 - 38.4k (**61.5%**) have **'avx2'** (or, 'avx2' and 'avx') in /proc/cpuinfo flags
 - 22.8k (**36.5%**) have only **'avx'** in /proc/cpuinfo flags
 - 1.2k (**2%**) do not have neither 'avx2' nor 'avx' in /proc/cpuinfo flags
- Collected sample
 - WN-CPUmodel: <https://cernbox.cern.ch/index.php/s/D3mnTg4Xs13IAWe>
 - PQ-WN-CPUmodel: <https://cernbox.cern.ch/index.php/s/JJWT374MiKchGfS>

OK, that's interesting...

- Not the full coverage yet (WIP!), on limited sample of WNs:
 - ~60-65% of probed WNs have the 'avx2' flag
 - ~35-40% of probed WNs do not have the 'avx2' flag
- N.B.: **avx2 != avx**
 - We may need to consult Linux & other experts
- Next steps?
 - Extend the sample to ensure full coverage
 - and understand that what we see is what we really get in terms of CPU features
 - We are collecting all flags, i.e. not only 'avx2' ⇒ organize in a repository?
 - e.g. WLCG CRIC?
 - Flags collection campaigns?

