

CMS change v11_1 to v12_5

What we have today in Hepscore candidate

- CMSSW_11_2_0 (slc7) for x86_64, ppc64le and aarch64
 - GEN-SIM, DIGI, RECO
- CMSSW_12_1_0 (slc7) for the same architectures + Nvidia GPUs
 - Modified HLT
- All sw releases in CVMFS, all working with simple tests
- The x86 CPU part used during the test campaign (hundreds / thousands of runs)

Why not ok?

- While **x86 (+GPU)**, **ppc64le (+GPU)** behave well on full machines, we have seen rare **segfaults** (memory protections) with **aarch64 (+GPU)**
 - It seems the aarch64 system is less tolerant to memory problems
 - Which is good! Allows us to see more code problems wrt using x86 alone
 - But of course, having failing Hepscore runs is not nice
-
- **This is not impacting Hepscore in the way we will use it in production (x86 and CPU only), but still...**

the problem / the solution(s) ..

- One of the reasons we like the move to Hepscore has to do with the capability to **show(case) differences between architectures in performance (events/sec)**, so that one day a site will be able to decide whether to buy x86 or ARM or PowerPC or ...
- Of course we could fix the problem by moving ONLY AARCH64 WLs to CMSSW_12_5_0 (which fixes the problem)...
- ... but this implies we surrender the capability as above (in which you need to compare apples to apples)

The proposal

- (Eventually) move all the 6 workflows x 3 architectures to CMSSW_12_5_0 (18 containers)
- This implies moving the singularity base image SLC7 → EL8 (alma, rocky, ...)
- The scripts have all been provided to AndreaS ~ 2 weeks ago
- Of course numbers will differ (different algorithms run in the WFs, different compiler, ...)
- **Can it be handled at this stage?**

(on top of this)

- If you look back at WS slides, AndreaS was hinting to the fact that it would be nice to add another (3) WLs, with a realistic HLT@GPU as we run it today @ CMS in P5
- Moving to CMSSW 12_5 we are in position to do that
- We would like to add this to the possible images, whether this is run everywhere / on demand