Cputime was chosen in a period when storages loads was the dominant reason single core jobs were inefficient. Pilots didn't even exist then. But now the reasons of inefficiency are others and often dues to experiments way of running things. Some things are difficult to measure or at least it will require development work to report it (i.e. idle cpus) but the immediate way to take correct usage into account is to use the wallclock which is already recorded in APEL (multicore jobs wallclock reporting is now sorted), so moving to wallclock is not a technical problem, but a political one. However there are good reasons to move to use wallclock.

WLCG should change it's primary reporting to wall-time. Wall-time is, and has always been, what sites and funding agencies actually buy (i.e. a machine for 4 real years), and this is even more obvious now we move to expand our sites out into external hosting and clouds, where you clearly and unambiguously pay real cash by the hour, regardless of what the CPU is doing. CPU reporting to understand the efficiency is important, but only for that reason. This was pointed out already long time ago (since 2006) by some administrators.

Regardless of whether you take CPU or wall clock as definitive, it's hard to de-convolute site efficiency from experiment efficiency. The whole thing is a lot easier to reason about when you take [nomalised] wall-clock as the basis, that being the real thing that we all pay for.

Funding agencies care about both what they paid for (the wall-clock), and the efficiency of how it was used. Reweighing these in CPU terms with arbitrary "assumed 0.8 nominal efficiencies", when there actual efficiencies are different, just makes it hard to discuss. Each experiment does need to back-calculate the wall-clock pledge from its CPU needs, given its estimated efficiency, but then, at least those per-experiment efficiency assumptions are exposed to the funding agencies clearly.

Some organisation are moving or moved some time ago to report wallclock time to their funding agencies.

For example GridPP in the UK is moving the performance metrics it uses for handing out hardware money to wall clock time partly because of the way cloud resources are accounted. Furthermore, as we move to jobs accessing data remotely more often, an experiment may choose to prioritise running the job today at a lower efficiency rather than tomorrow at the site(s) where the data is, at a higher efficiency. Sites running inefficient workload are penalised even on standard sites. This was evident with analysis but is even more evident since the introduction of multicore jobs. Sites shouldn't be penalised for that choice by the experiments, and we don't want to create perverse incentives for the sites to resist new technologies because they might lose out in the way the metrics are done.

And OSG has worked hard with its funding agencies to explain that we believe WLCG accounting measures the wrong thing (normalized CPU time); in the past, we even prepared normalized wall time reports for the funding agencies. So, I think there'd be a strong interest to put the focus on walltime. We believe making the migration from CPU time to walltime would be worth the human effort required.

In the MB reports, more interesting is not the "100%" efficiencies in the top right corner (which to me mean exactly what was described earlier, i.e., site reports are not corrected and taken as is). Most interesting are graphs with corrections against pledges. The pledges are compared to uncorrected CPU-time! And this is exactly what sites are complaining about. A fair comparison would be to wall-time, of course, especially since the pledges are derived from requirements, which in turn do have large inefficiencies ***included***.

For example if you check now PIC [1], for CPU they 83% of the pledge this year with the current approach (in which we assume the jobs are 100% efficient). However, if we compare with the used walltime in PIC, that results in 104% of the pledges. Jobs have been using the pledges correctly in this case.

The reason of the low CPU time used in PIC is the very poor performance of the CMS pilot jobs, in particular for the MultiCore jobs, running inefficient payloads (pilot jobs occupying 8-slots, and running a few single-core payloads). This has happened to all of the CMS sites.

The CPU Efficiencies the year 2015 for ATLAS, CMS and LHCb have been:

ATLAS - CPUeff 85%

CMS - CPUeff 62%
LHCb - CPUeff 98%

So, now the question: if in the end we are going to be targeted towards CPU time used by the jobs, it's not fair, for two reasons:

- We don't know which is going to be the CPUeff of the jobs for the VOs in the next year, hence we don't know in advance how to provision CPU resources to fulfill the pledges, if we are accounted towards used CPUtime.
- It's not fair, as the (not standard) job inefficiencies are penalizing sites, in this accouting schema.

Hence we should revise this. The VOs know about the CPU efficiency and they can take this into account when asking for the pledges (indeed they do, afaik), and sites should be targeted towards used WALLTIME. Of course, we *should* keep an eye on the cputimes and the efficiencies, as these are important measures with room for improvements, and should be overviewed.

The same applies to **Disk** and **Tape**: we should be targeted towards INSTALLED CAPACITIES, not usage, without applying factors as well. I mean, it's not the site's fault if ATLAS don't fill the available space which has been asked in the pledges (tape or disk), and provided by the site. But, again, it's very important to measure the usage, and overview this, but sites should targeted towards installed capacities.