

Job Allocation and Handling working group report

Antonio Pérez-Calero Yzquierdo (CIEMAT & PIC), for the WG team

WLCG Workshop, May 7th 2025



The mission of the WG

The WG on Job Allocation and Handling

- Proposed after [presentations and discussions in 2024 WLCG WS](#) concluded this forum was needed
- Approved by the October 2024 WLCG Operations Coordination meeting, the WG started its activity in November 2024
- The WG aims at providing a discussion platform for experts from LHC VOs and WLCG sites on matters of compute resource management and workflow to resource matchmaking
- Contributing members from WLCG VOs and site experts
- Meetings: 6 topical meetings since kick-off, attended typically by 5-10 people

References: [Indico category](#), [Twiki](#) for the WG, meeting minutes [document](#)

Contributions from R. Walker, F. Stagni, F. Barreiro, M. Storetvedt, A. Delgado, B. Jones, M. Kühn, C. Acosta, T. Birkett, O. Smirnova. THANKS!

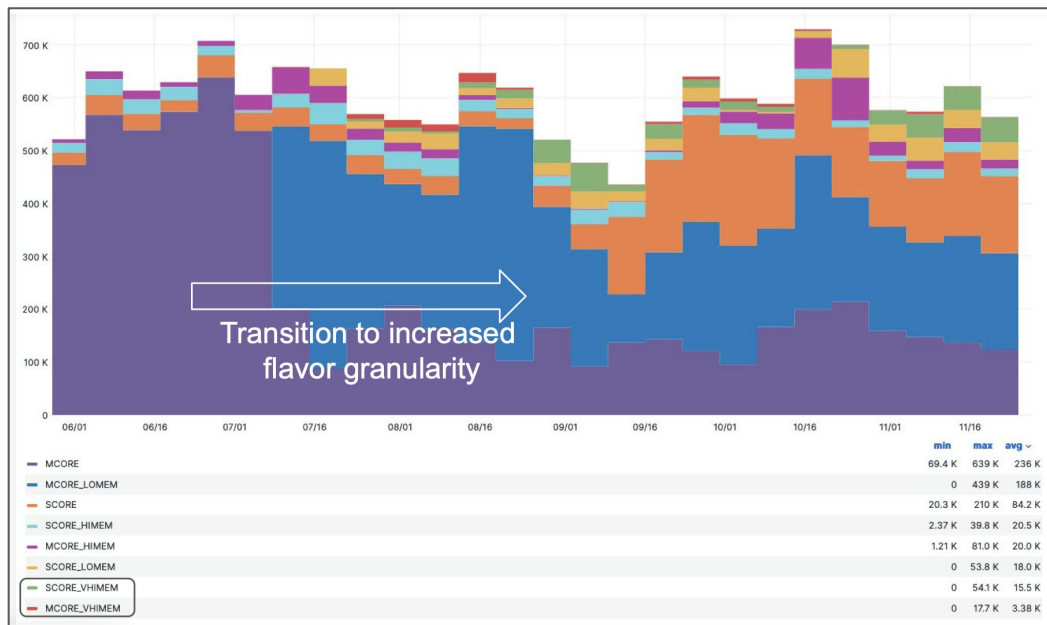
Experiments reports: ATLAS

- ATLAS employs a mix of pilots modes (push, pull, pilot streaming)
- Observed by sites as jobs of diverse core count and memory requirements, scheduling relying on the LRMS

Recently increased granularity in ATLAS job flavors:

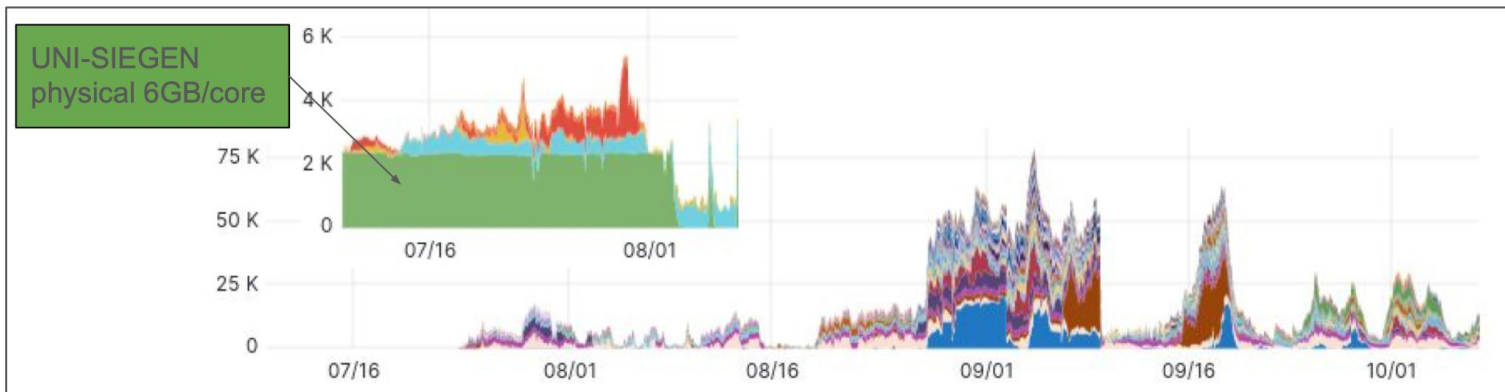
*[single core, multi core] x
[low memory, standard, high memory,
very high memory]*

- *Low memory: 1GB/core*
- *Standard: 2GB/core*
- *High memory: 3GB/core*
- *Very high memory: the maximum defined by the queue*



Experiments reports: ATLAS

- Need for a increased fraction of high-memory slots: use diverse Panda queues to throttle the average use of memory/core at sites.
 - Workloads in ATLAS generally dominated by low memory simulation tasks, at 1 GB
 - Room for high/very-high mem. jobs with low impact on total unused cores, even on regular mem/core slots
 - Control by site by ATLAS, but local scheduling (WN) left to the site admins.
 - Some idle cores per WN unavoidable, impact depending on the site admins expertise.
 - This needs to be accounted for.
 - Overall **success of the strategy**, up to 50k cores in total for very high mem slots (5 GB/core), more sites are now available to very high mem/core workloads



Experiments reports: ATLAS

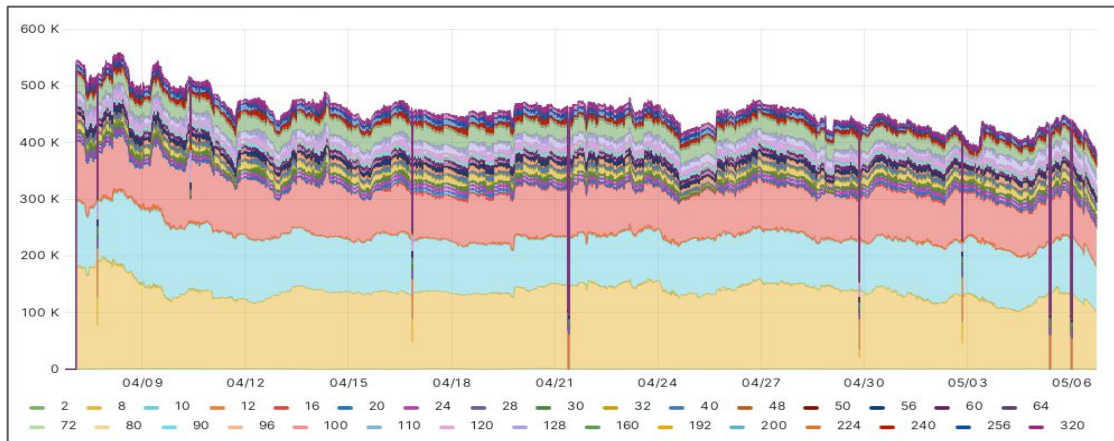
- **Interest in moving to 16 cores as new default for ATLAS?**
 - Trade-off, as advantages are expected (less scheduling at the PANDA queues for the same workload), but potentially some effect on CPU Eff scaling, and throughput, depending on job type
 - Larger jobs need also be longer (e.g. double events/job) to try and conserve CPU efficiency
- In any case, **moving to 16 cores does not look too bad**, so ATLAS could be in favor in shared sites in order keep compatibility with CMS / ALICE.
 - But keeping also 8 core slots for those jobs that need that size (overlay reconstruction).
 - And single core as well (evgen, merge, analysis...)
 - **Can 1/8/16 cores live together, efficiently?**
- **Runtime:** 96h already standard walltime supported for ATLAS resources
 - Jobs target more like 12h, although some do run for 4 days
- **Whole-node** scheduling is a requirement when using most HPCs, but not all payloads can scale
 - **No clear benefit on the grid**

Experiments reports: CMS

CMS model based **entirely on multicore pilots**, based on HTCondor partitionable slots, suitable for **8/16/other** fixed size slots, and **whole-nodes**, both in the Grid and HPCs.

- **Internal partitioning** deals with diversity of payloads jobs, transparent to the LRMS
- No need for dedicated himem slots
- **Whole-nodes slots already in use** in multiple sites
 - Beneficial for CMS due to providing increased flexibility
 - Efficient use possibly linked to extended pilot lifetime
 - A solution for the larger HPCs granularity

CMS Global Pool resources as a function of the slot size



Experiments reports: CMS

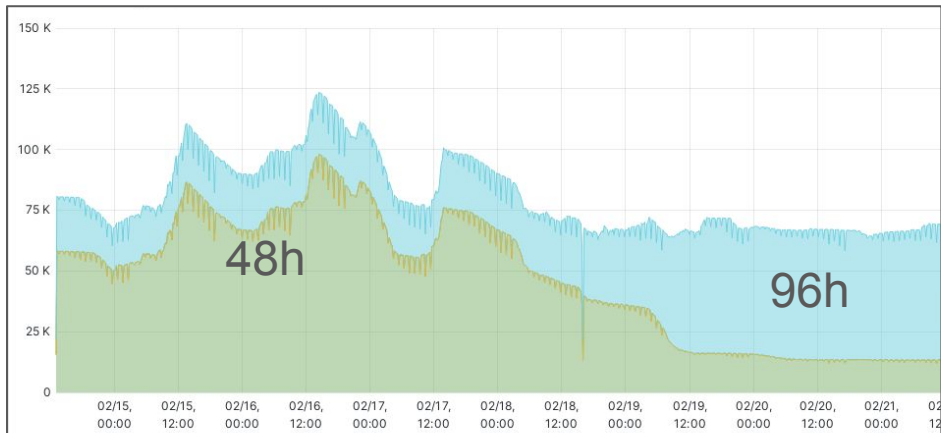
- Ongoing campaign to switch 8-core slots at CERN and European Tier-1s into larger slots: 16-core and also whole-node were available
 - Better overall flexibility for internal scheduling
 - An immediate use case: Tier-0 PromptReco jobs requirements evolving, with 16-cores being tested. Jobs executed at CERN and also overflow into European Tier-1s



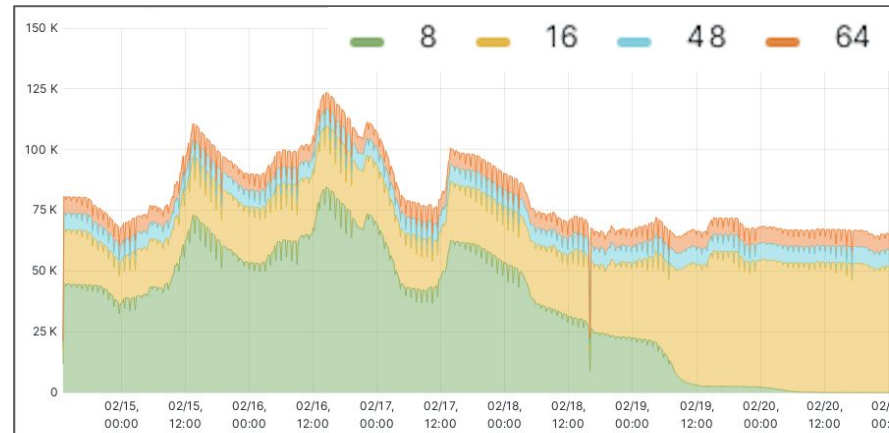
Experiments reports: CMS

- Transition started at CERN mid February in preparation for the new LHC run
- Already using longer and larger slots
 - Whole-node slots in the dedicated CERN partition
 - Switched from 8 to 16 cores in the shared farm
 - Longer jobs supported, now in use

Slot runtime



Slot CPUs

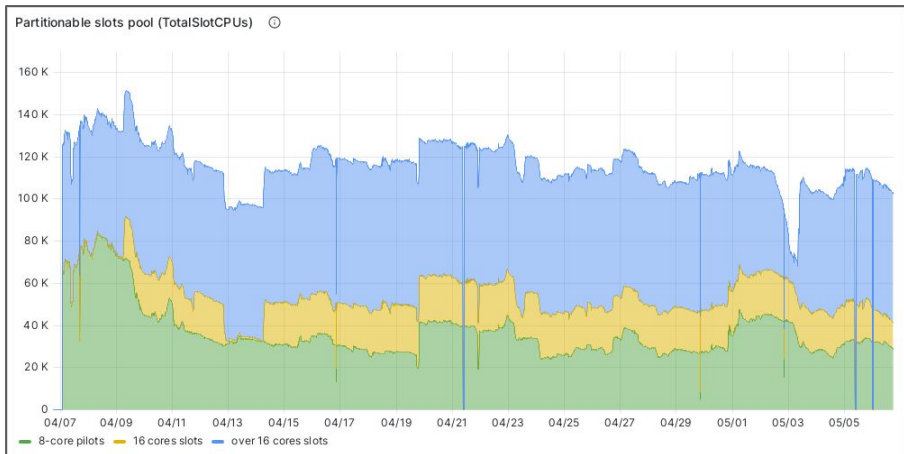


Experiments reports: CMS

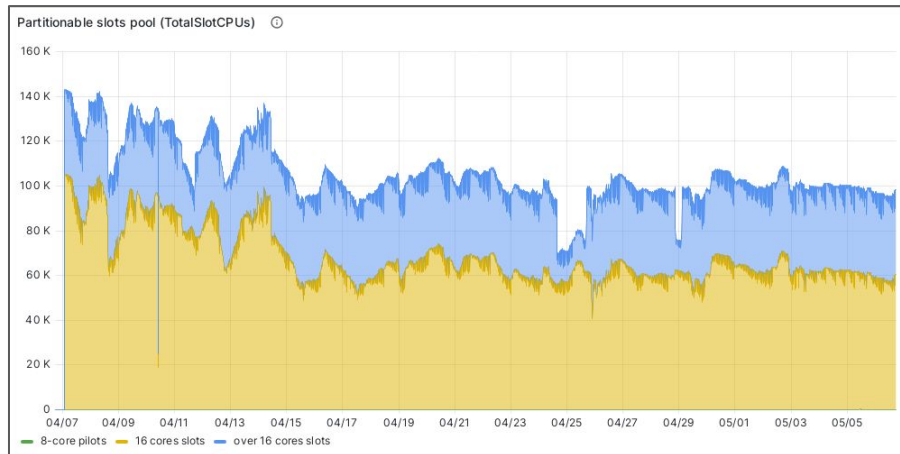
Current distribution of slots at CERN pool (CERN and P5 farm) as well as Tier-1s:

- European Tiers-1s evolving from 8-core to 16-cores slots,
- Whole-node slots at FNAL
- CERN pool entirely on 16-core slots and larger

Tier-1s



CERN pool



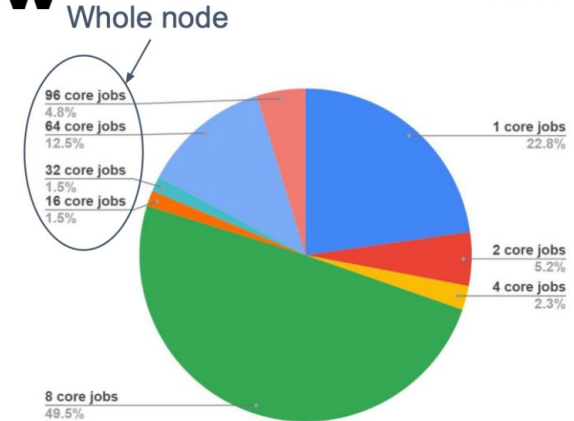
Experiments reports: ALICE

Aligned with CMS strategy, built with different technologies but **similar principles**:

- Local voboxes query the WM queues, launching pilot jobs based on the compatibility of the workflows to the sites
 - Pilots acquire resources from the local batch systems and pull payloads from the queues.
 - Voboxes are easy to maintain and update, based on standard images shared via repo.
- Java ALICE Environment (**JAlEn**) middleware launches **multicore pilot jobs** capable of executing multiple payloads in parallel.
 - **Combination** of multi-sized, multi-user payloads to **fit the mcore slot**
 - Payloads run in nested containers, for software availability for all payloads, then sub-containers for process isolation.
 - Resource usage fine-tuning possible where cgroups v2 are supported. Preferred, as singularity containers do not enforce the strict utilization of allocated resources.
 - Slot overcommit applied dynamically to help regain CPU efficiency. It is only applied on whole-nodes slots.

ALICE Grid workload overview

- ALICE jobs with different resource requirements
 - Including CPU cores, memory, disk and accelerators
- All Grid sites are running **multicore queues**
 - 14 whole-node queues
 - All others 8+ CPU core queues
- JALiEn adapts to the different allocations
 - Automatically discover and scale to whole node resources where available
 - If fixed-size slot, limits inferred from cgroups v2 or batch queue settings
 - Bounding CPU and memory usage
 - Further partition slot resources within the job mix





Optimising job placement

- **Job-to-core mapping** algorithm starts by jobs with larger allocations
 - Simulation jobs (8-core) are placed first and analysis (1,2,4-core) as backfill
 - Transparently balancing resource loads
- NUMA-aware pinning leads to improved **execution efficiency**¹
- Predictable execution time fostering better **scheduling decisions**



¹ [CPU-level resource allocation for optimal execution of multi-process physics code](#)



Experiments reports: ALICE

- **Multiple benefits from using large multi-core slots identified:**
 - Flexibility in mixing workloads
 - GPU brokering
 - Oversubscription for improved CPU efficiency
- In conclusion, ALICE's model is very **suitable, ready, and in favor** of moving to **larger slots** (16-core) and even **whole-node** (preferred) where supported
- **Analogous** position to CMS



Experiments reports: LHCb

- LHCb **workloads** up to now **dominated** by **single-core jobs**, which cover their needs
 - Hi-mem and multicore jobs represent a **very small fraction** and there is no expectation for this to change in the short to mid term
 - LHCb executes mostly MC GEN in the grid, which even if technically capable of being multi-core, is very efficient in the use of memory, so kept under control within the standard 2 GB envelope
- The LHCb computing team has been busy with deploying DiracX in production
 - DIRAC(x) is equipped for exploiting multi and large-core nodes, also whole-node allocations, however in practice missing the payloads
 - LHCb software version fully enabling exploitation of multiprocessor jobs is not yet used for large productions
- For LHCb, provisioning CPUs in 1, 8 or 16 cores slots would not make much difference, as they would be partitioned for single core payloads anyway

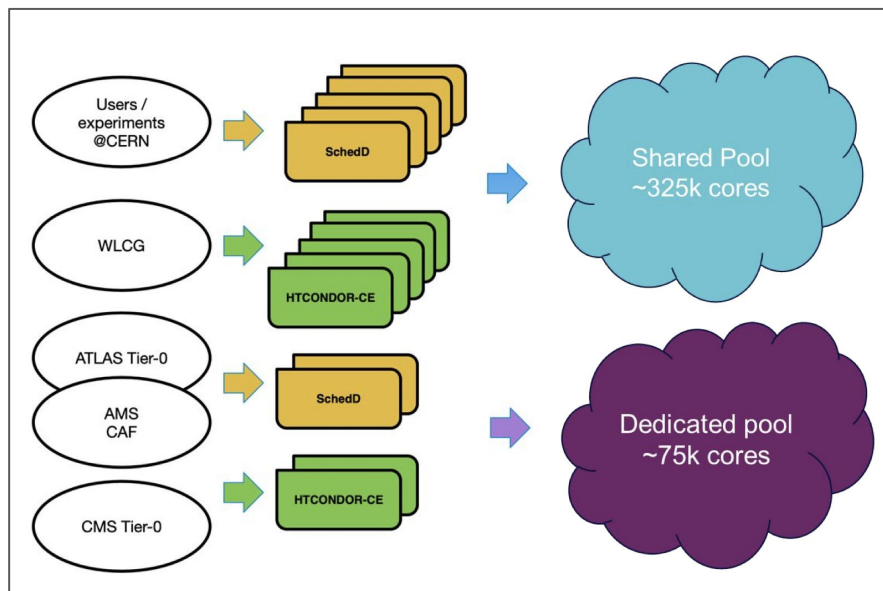


Feedback from sites

- WG specially interested in input from key big multi-VO shared sites
- So far requested and got contributions from CERN and a number of European Tier-1s
- I will present a summary of their reports in relation to the main questions being discussed within the WG

Feedback from sites: CERN

- Large shared pool of resources for single and multicore slots, plus dedicated pool dominated by ATLAS, where some whole-node slots can be provisioned (e.g. CMS)
- HTCondor: a key element in managing the complexity
 - **Quantization** of the slots, at 3 GB/core, mainly for accounting purposes and simplified scheduling
 - **Minimal impact** on cluster occupancy from defrag_daemon draining effect to allow 16-core slots (CMS).
 - WNs are much larger than the requested slots



Feedback from sites: CERN

LHC VO jobs at CERN

VO	Dedicated?	Pilot Size	Late Binding?
ALICE	Not currently, but interested in whole node	8 core, can do whole node	Yes
ATLAS	Large dedicated tier-0 pool, backfilled with grid	Mix of 8 core and single core	Less so: jobs are classified by type
CMS	Some dedicated, but both share + dedicated in global pool	16 core or whole node on dedicated	Yes
LHCb	No	Single core	Yes

Feedback from sites: CERN

Fair-share with large surplus utilization!

Pledge vs running for WLCG VO's

ATLAS



ALICE



LHCb



CMS



Feedback from sites: Tier-1s

GRIDKA:

- Large scale of resources (Tier-1 cluster at 55k CPU cores) allows for great flexibility.
 - 97% of CPUs typically allocated.
- High-mem requests are allowed: RAM is at 70% allocated, schedule on cpu cores only.
- Whole-node slots provisioning for ALICE
 - Could be also extended to CMS
- Scheduling principle: avoid maximal fragmentation
 - On the last 8 cores of a WN, only one further job will be assigned
 - No use of usual HTCondor tools (defrag daemon, etc), but no need for further micro managing
- Runtime up to 7d supported, but very much discouraged.
 - But change from 2d to 4d for example should not a problem
- Slots at HPCs via T3 allows access to more whole node slots

PIC:

- Cluster occupancy typically around 92%, which is not bad considering smaller cluster (12k CPU cores) and the mix of
 - Resources (older, smaller machines)
 - Multitude of users, via grid (T1+T2) and local (T3), and their requirements.
- Max. runtime of 100h for grid jobs allowed
- No whole-node slots supported
- HTCondor tools experience: not trivial, but great support from the devs and community of users
- Defrag daemon useful in providing slots for 8-core jobs, now also 16-core for CMS.
 - Impact on overall cluster occupancy negligible
 - but parameter tuning once the situation is stable

Feedback from sites: Tier-1s

RAL:

- HTCondor batch with ARC CEs, running jobs in containers, dominated (80%) by single core jobs (LHCb)
- Defrag daemon to provide multi-core slots (2 and 8), defrag to 16 cores only enabled in the largest WNs.
- Opportunistic use of overpledge allowed,
 - e.g. CMS using up to 5x its pledge frequently
- Procurement of WNs is generic enough to satisfy all request from VOs, e.g. 4 GB/core
- As total HS23 only marginally increases when running into HT cores, no point in trying to saturate number of logical cores.
- 16-core pilot are not a problem, acceptable.
- 48h runtime limit: in principle, longer jobs are not allowed
 - If VOs require them, longer jobs could be accommodated
 - BUT it would severely impact and reduce their ability to schedule jobs opportunistically
 - Again, CMS does benefit significantly from opportunistic running.

NDGFT1:

- Federation of 7 data centers, supporting ALICE and ATLAS as part of WLCG
 - Use of ARC-CEs with data prestaging for ATLAS, no pilot jobs, no "late binding".
 - ALICE running pilots in dedicated partition, 16 core slots, with some whole-node slots at sub-sites
- Slurm batch system with classical fair-share
 - ATLAS jobs coexist with non-WLCG jobs
 - Cgroups used to enforce resource limits, e.g. memory request typically underestimated
- Nordic T1 enables VOs access to HPCs, some big like Vega
 - Whole-node or partial allocations possible at all sites
 - Unclear however if some HPCs (Vega) have a minimal granularity at whole-node allocations.
 - "No limit" in execution time at HPCs
- In general, variety of job "sizes" favors flexibility, and "fill the gaps" improving overall usage.



Feedback from sites

In general, sites remark that they will **comply with VO requirements according to pledges**, but present a **warning** for the VOs:

The harder to schedule jobs become, the more unlikely they will be able to use the resources optimally, also potentially leading to severe reduction of overpledged opportunistic usage, from which some VOs are greatly benefiting.

Deployment of resources in **dedicated partitions should be avoided**, even if they can be supported at sites with very large clusters (e.g. CERN)

Conclusions

Personal impression: we are very close to agreement between sites and VOs:

- **16-core slots** seem not to be a major concern for sites,
 - Especially if more than 1 VO request them at a given site.
 - Simple switch from 8 to 16 cores, not significantly more difficult or wasteful
 - Much larger WNs today than 10 years ago in relation to slot size (e.g. 16/192 vs old 8/24)
- **Runtime** can be extended beyond 48h
 - Sites in many cases already support this...
 - ...but discourage VOs from very long requests, as it reduces flexibility in use of the resources, turnaround between users, and support for policies enabling surplus and backfilling
- **Memory:**
 - Sites already provide in general more than the nominal 2 GB/core
 - In general, memory is therefore abundant, and scheduling is based on CPUs, not memory, saturation
 - Cgroups allows for a control of the actual memory consumption, which is often not correctly predicted
 - ATLAS mechanism to control overall average memory works as intended, while ALICE and CMS take care of memory management within their pilots
- **Whole-node slots:**
 - Often require dedicated VO partitions, so only some big sites agree in providing them
 - Reduces the ability for sites to satisfy all their customers
 - Goes against “fair-share” policies benefits



Next steps

- Discussion on next steps should involve the community represented here
 - Are the conclusions expressed in the previous slide shared?
 - Can they be safely extrapolated to the rest of sites?
- Producing a draft for a document with recommendations to sites and VOs, as defined in the mandate.



Backup slides



WG Mandate

- **Mandate:** “The goal of the WG is to define the most efficient strategy for job allocation and handling, namely, the most efficient **core count** for batch job allocation and a more efficient handling of different **job classes** with **diverse requirements**. This includes **analysis of the experience accumulated so far** for running multi-core jobs, whole job scheduling, different approach for handling jobs with specific requirements, as for example, high-memory jobs. Based on these studies, the **recommendations** for the sites and LHC VOs should be developed and **presented to the community**. The WG includes experts from the WLCG sites and LHC VOs.”

Discussion in DESY (2024 WS)

Proposals for discussion

- Allow some jobs to request memory over physical amount per core
 - up to site configured max RSS
 - unlimited number of cores if mix means no idle cores, or capped at low(10%) level?
 - VO responsible for this. Monitored and enforced by the site - how? Trust but verify?
 - accounting for any idle cores important for pledges
 - important enough to develop memory dimension or rescaling in APEL?
 - HS23 does not scale linearly with HT cores. Accounting to reflect this.
- Move to 16 core as new standard(currently 8) where a VO requests it,
 - if then CMS would want 16 everywhere. ATLAS ok with mix, especially when HT-off.
 - major/all VOs send 16 core jobs as standard, to ease slot re-use
 - must allow 1 core jobs
 - ALICE want then 72hr walltime. CMS also prefer longer walltimes for more cores.
 - to do with draining inefficiency at the end

Discussion in DESY (2024 WS)

Whole node scheduling

- Wholenode only where advantageous, e.g. GPU, numa pinning, many-core job(non-parallel)
 - oversubscription of cpu can improve efficiency when some jobs not cpu-bound
 - useful for packing gaps and draining, i.e. better slot efficiency
 - should be allowed also in MCORE if cgroup contained
 - otherwise let the Batch System schedule
- Pledged resources must be able to continue with S/MCORE jobs
 - combining whole nodes with S/MCORE jobs is problematic for many sites
 - If (CMS & ALICE & expert BS admin & volunteer) then do it
 - need at least 2 VOs with reliable whole node jobs, to keep slots
 - for this subset ATLAS would submit some wholenode(but also S/MCORE)

~~Relaunch Multicore TF to address this between exps, sites and WLCG~~