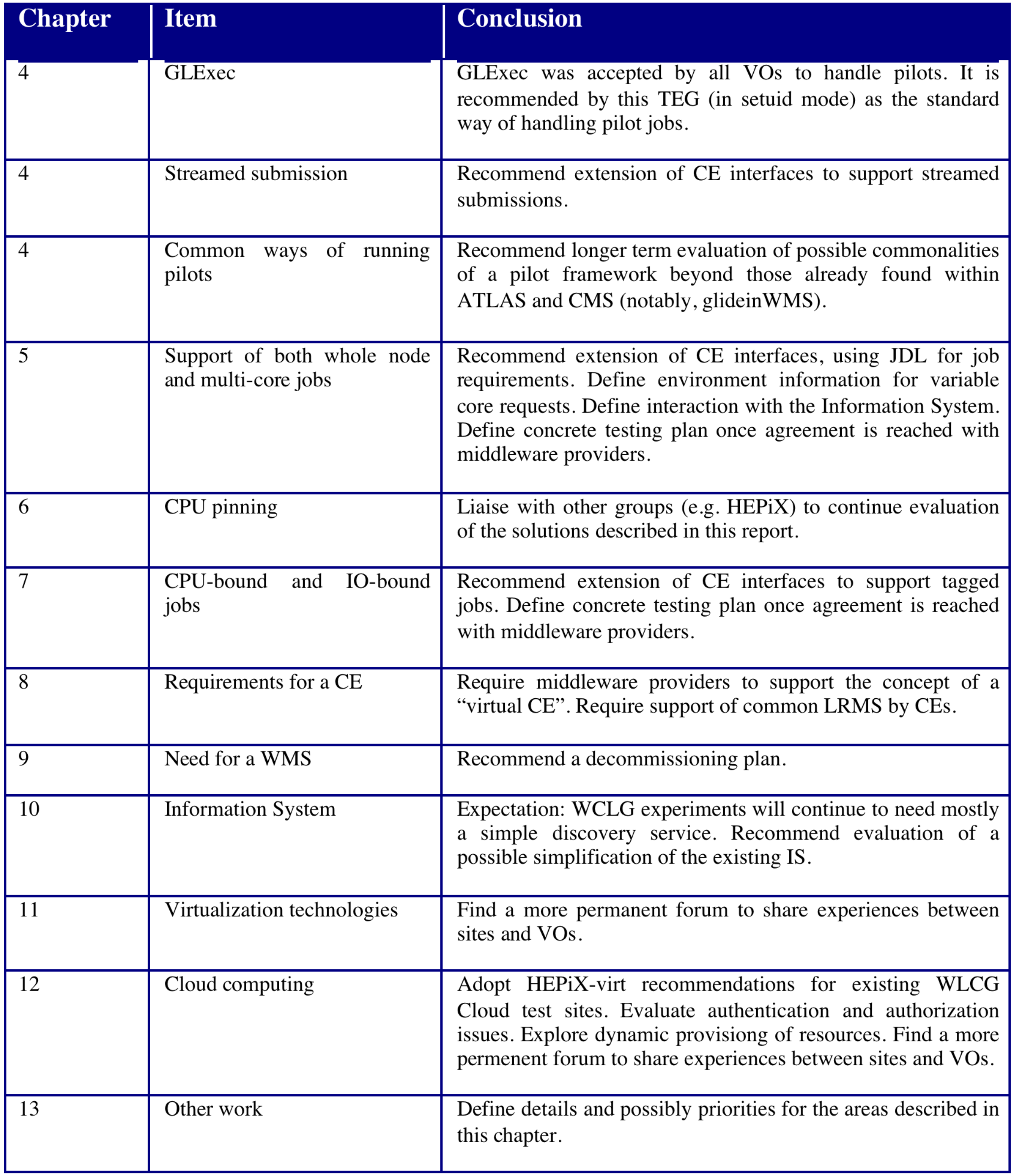
WM TEG
Some points for discussion

Davide Salomoni, INFN
WLCG workshop

NYC, 20 May 2012



Chapter	Item	Conclusion
4	GLExec	GLExec was accepted by all VOs to handle pilots. It is recommended by this TEG (in setuid mode) as the standard way of handling pilot jobs.
4	Streamed submission	Recommend extension of CE interfaces to support streamed submissions.
4	Common ways of running pilots	Recommend longer term evaluation of possible commonalities of a pilot framework beyond those already found within ATLAS and CMS (notably, glideinWMS).
5	Support of both whole node and multi-core jobs	Recommend extension of CE interfaces, using JDL for job requirements. Define environment information for variable core requests. Define interaction with the Information System. Define concrete testing plan once agreement is reached with middleware providers.
6	CPU pinning	Liaise with other groups (e.g. HEPiX) to continue evaluation of the solutions described in this report.
7	CPU-bound and IO-bound jobs	Recommend extension of CE interfaces to support tagged jobs. Define concrete testing plan once agreement is reached with middleware providers.
8	Requirements for a CE	Require middleware providers to support the concept of a “virtual CE”. Require support of common LRMS by CEs.
9	Need for a WMS	Recommend a decommissioning plan.
10	Information System	Expectation: WCLG experiments will continue to need mostly a simple discovery service. Recommend evaluation of a possible simplification of the existing IS.
11	Virtualization technologies	Find a more permanent forum to share experiences between sites and VOs.
12	Cloud computing	Adopt HEPiX-virt recommendations for existing WLCG Cloud test sites. Evaluate authentication and authorization issues. Explore dynamic provisiong of resources. Find a more permenent forum to share experiences between sites and VOs.
13	Other work	Define details and possibly priorities for the areas described in this chapter.

From the WM TEG Report

- On the following slides: some points in need of proper
  - Prioritization
  - Discussion
  - Implementation plans
- Not discussing here security-related topics
  - See the Security TEG and e.g. the next pre-GDB day on AAI on WNs (June 12, CERN)

Higher Priority (GDB, Feb/Apr '12)

- Streamed submission
  - Effort: moderate-significant
  - CREAM considered this feasible, but need to define details on how this should be done (e.g. to protect sites). Need to discuss implementation plans for other CEs as well.
- Full multi-core / whole-node jobs JDL support
  - Effort: moderate (possibly significant for "variable core" support, or if some LRMS' do not natively support all requested capabilities)
  - Middleware providers should exactly specify which capabilities they are going to offer on which LRMS (a preliminary evaluation was done in the TEG; this needs to be tested in the field)
- Virtual CE support
  - Effort: moderate (planned for CREAM already)
  - Key to reduce downtimes and dependencies on single CE instances
- Support of CPU-bound vs. IO-bound jobs
  - Effort: moderate
- Extended environmental info (e.g. HS06, job life time)
  - Effort: moderate for HS06, possibly significant for job life time. Related to work needed for multi-core support

– Full glexec deployment → operational issue, timescale?

Lower Priority (GDB, Feb/Apr '12)

- WMS decommissioning plan → timescale? Set up a decommissioning group?
- Support for additional LRMS by CEs
- Evaluate (more) common ways of running pilots
  - → glideinWMS for CMS and ATLAS
  - → direct submission to the CREAM CE for Alice, LHCb
- Virtualization, cloud computing and volunteer computing
  - Many interesting (and dispersed) work, from both sites and VOs → HEPiX but also regular status / development reports at GDBs?
- Evolution of information system support
  - → WLCG clearly does not need complexity here – define targets
- CPU pinning
- Standard signaling for job termination

New CE features

- Background: several sites need/want to support
  - Not only WLCG… (shared facilities, multiple communities)
  - … with a clear, validated entry point into site resources…
  - … with an efficient combination of 1) job management and 2) resource allocation
    - Hence resource requirement extensions → one could argue this IS CaaS
    - And the convergence in the TEG toward e.g. “site-based pilot factories”, vs. “WLCG- (or VO-) specific pilot factories”
- A point discussed over and over again in the TEG: a “single CE implementation” won’t likely fit all of WLCG
  - → functional definition (using CREAM and GRAM interfaces)
- Unknowns
  - Support for existing CEs beyond 2013/2014
  - Implementation / negotiation plans w/ CE providers (ARC, gLite, OSG)
    - → “CE Upgrade” working group? (management/technical)
  - OTOH
    - these are incremental changes to what we already have and use in production
    - compare to a “new” solution, wrt 1) development, 2) adoption by sites, 3) adoption by experiments

Optimizations

- Site-level optimizations
  - Distinction between I/O vs. CPU bound jobs
    - Much requested feature by sites, but which will eventually benefit (potentially to a significant extent) experiments
    - Should be much simpler to implement than e.g. multi-core support
    - Simplify set up at sites (e.g. avoid complex and unreliable heuristics to mitigate bottlenecks issues)
- Pinning, SMT, etc.
  - Links w/ HEPiX
  - Also related to deployment of SL6 on WNs (e.g. lxc, cgroups)
- Operations
  - Redundant/virtual CE → should have been a standard feature / requirement implemented long ago