



www.egi.eu



@EGI_eInfra

Accounting Portal *Roadmap & Status*

Iván Díaz Álvarez

CESGA (Galician Supercomputer Center)



The work of the EGI Foundation
is partly funded by the European Commission
under H2020 Framework Programme

- Python 3 code migration (done)
- PHP 8 code migration (done)
- Improve performance and responsiveness of the Portal (WIP)
- Accounting data anomaly detection
- Storage Accounting
- WLCG new benchmark adaptation
- Stress and security testing
- Automated testing using Selenium framework

- Implement memcached functionality
- Automated regression testing
- Improve virtualisation layer
- Evaluate other Web server possibilities
- GPU Accounting

Python 3 Code Migration (done)

- The Python language had structural changes in version 3 that caused changes in the syntax and other basic aspects, causing incompatibilities.
- Python 2.x has been deprecated after a grace period.
- The Accounting Portal depended on Python components like Django, that took some time to migrate.
- Also, those components depend on libraries that have the same problem.
- Finally, we did the migration of the Python code on the controller, this also needed changes in the Django code, that affected basic URL pattern matching and context manipulation.
- The changes are implemented in the *accounting-pre* instance.

PHP 8 Code Migration (done)

- We also updated PHP from version 5 to 8.
- It doesn't break compatibility as hard as Python, but many changes had to be made to remove excessive warnings in the logs. (e.g. when referencing possibly uninitialized array values).
- Also some older code still had to be modularized, paying technical debt.
- PHP 8 improves performance significantly with JIT compilation techniques.
- Also the language has better constructs, but we still haven't modified the old code to use these.
- The changes are implemented also in *accounting-pre*.

Improve performance and responsiveness of the Portal (WIP)

- Some of the changes done in pre help in that direction:
 - Migration of Apache 2.2 to 2.4
 - Migration from MySQL to MariaDB
 - Python 3 migration (improvement variable depending on the case)
 - PHP 8 migration (JIT)
- We are also working on the JavaScript layer, since Dojo allows many optimizations on minification and consolidation of the JS libraries needed.
- If those steps are not enough to guarantee a good performance, the portal can be splitted into several different virtual machines, so that backend operations don't impact frontend performance.

Accounting Data Anomaly Detection

- In the past, there have been many problems with malformed data sent up from Sites that don't get detected until they show up in the portal.
- We have proposed that importing scripts, specially Cloud ones, have some basic sanity checks (no negative numbers, no numbers that exceed $>100x$ the last period) that at least point out to Site Admins they should check their publication.
- The Portal can also implement some similar measures, but there are some problems:
 - The data has still reached the portal, and can't be treated atomically (no site differentiation, no old data removal, summarization can lose data).
 - The importing software is provided by APEL, so we can't control the import process itself).

- Storage has to implementation in *accounting-devel* and *accounting-devel-next*.
- Summarization is done in the portal, but the data to be processed has grown and the process makes the machine unresponsive.
- Usage in production depends on the summarization to be done by APEL, as with the Grid and Cloud data.
- For GPU accounting we would need schema documentation and sample data, as always.

WLCG new benchmark adaptation

- WLCG is working on the replacement benchmark for HEPSPC06.
- This benchmark is crucial for accounting, since it is the value used for normalisation of sites, so that hours in faster sites count more.
- As of now, the normalisation is done by APEL and the final normalised and unnormalised values are sent to the portal.
- It is expected that for the portal, the changes would amount to changes in the labelling.
- There is a meeting pending with WLCG to discuss more details about this change.

Stress and security testing

- There is already some external security testing in the form of SHODAN.
- Internal one can be done with techniques such as fuzzing, but we have to be careful since our network is in RedIRIS, which tends to block suspicious machines automatically (this includes things like port scanning).
- Some automated stress testing would be also interesting.

- Evaluate new HTTP server
- Improve virtualization layer
- Implement memcached



www.egi.eu



@EGI_eInfra

Thank you
for your attention.

Questions?



This work by the EGI Foundation
is licensed under a Creative Commons
Attribution 4.0 International License.



The work of the EGI Foundation
is partly funded by the European Commission
under H2020 Framework Programme