



WLCG Accounting of EOS and smart files

József Makai, CERN

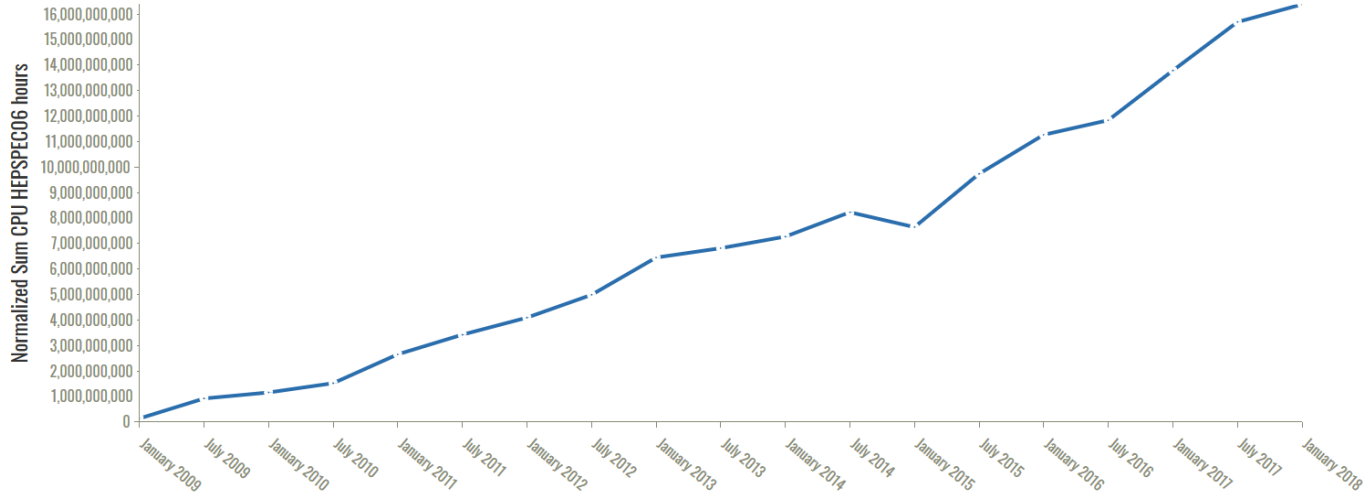


What is WLCG accounting?

- WLCG: Worldwide LHC Computing Grid
- Accounting
 - Shows resource allocation/usage by site and experiment
 - Produced each month
 - Consists of CPU, disk storage and tape storage capacities, consumption
- A working group is working on generating the reports for sites
- Reports for T1 and T2 sites

EGI Portal

EGI Normalized Sum CPU HEPSPEC06 hours in 6 months periods



Normalized CPU HEPSPEC06 hours in last 6 months period

16,382,427,169

Number of jobs in last 6 months period

304,644,086

CPU Time hours in last 6 months period

1,520,515,155

EOS Accounting interface

- Where does EOS come to the picture?
 - EOS is the disk storage system for the experiments
 - Extract the disk storage information through an easy to use interface designed for this purpose
 - ... and the eos accounting command was born
- Accounting information from EOS
 - Quota nodes
 - used/available space
 - Number of files stored
 - Path
 - Update timestamp
 - Custom information
 - Instance level information
 - Used/available space
 - Version
 - Custom information

Custom information syntax – part 1

- `sys.accounting` attributes
 - On quota nodes for quota nodes
 - On `/eos/<instance>/proc` for instance level
- Defined mapping between attributes and place in JSON output
 - a name after a `'.'` starts a new object
 - a number (from 0 on) after a `'.'` starts a new array
 - Last name in attribute path will be leaf level field with a value
 - Last number in attribute path will be leaf level array element with value
 - N attributes for N values

Custom information syntax – part 2

- Whaaat??
- Let's see an example...

```
EOS Console [root://localhost] [/eos/dev/replica/> attr ls /eos/dev/proc/
sys.accounting.capabilities.0="data.access.file"
sys.accounting.capabilities.1="data.access.streaming"
sys.accounting.capabilities.2="data.management.storage"
sys.accounting.implementation="DPM"
sys.accounting.servicetype="org.wlwg.se"
sys.accounting.storageendpoints.0.capabilities.0="data.management.transfer"
sys.accounting.storageendpoints.0.capabilities.1="data.management.storage"
sys.accounting.storageendpoints.0.interfacetype="srm"
sys.accounting.storageendpoints.0.name="atlassrm"
sys.accounting.storageendpoints.1.capabilities.0="data.management.transfer"
sys.accounting.storageendpoints.1.interfacetype="gsiftp"
sys.accounting.storageendpoints.1.name="atlasgsiftp"
sys.s3.bucket.test="test"
sys.s3.id.test="0123456789"
sys.s3.path.test="/eos/dev/test/instancetest/s3.dir"
```



```
EOS Console [root://localhost] [/eos/dev/replica/> accounting report
{
  "storageservice" : {
    "capabilities" : [
      "data.access.file",
      "data.access.streaming",
      "data.management.storage"
    ],
    "implementation" : "DPM",
    "implementationversion" : "4.2.12-20180202120141git3855655",
    "latestupdate" : 1517768303,
    "name" : "eosdev",
    "servicetype" : "org.wlwg.se",
    "storagecapacity" : {
      "online" : {
        "totalsize" : 747512791040,
        "usedsize" : 69408686080
      }
    },
    "storageendpoints" : [
      {
        "capabilities" : [ "data.management.transfer", "data.management.storage" ],
        "interfacetype" : "srm",
        "name" : "atlassrm"
      },
      {
        "capabilities" : [ "data.management.transfer" ],
        "interfacetype" : "gsiftp",
        "name" : "atlasgsiftp"
      }
    ]
  },
}
```

Caching for accounting

- Accounting supports rich features for caching the information
 - Information doesn't need to be completely recent
 - Can take some time to compute result
 - `accounting config -e <expiry_time>` → after that served from cache and asynchronous update triggered in background
 - `accounting config -i <invalidity_time>` → after that no longer served from cache, synchronous update, client waits for result, cache updated with result
 - `accounting report -f` for forced synchronous update if need be
- Own cache implementation designed for this purpose in background
 - Decouples requests and concrete executions (computing the result)
 - Only 1 execution can be at a time, others wait for the response of it if they need to be synchronous, otherwise they can get it from cache if present

Accounting with smart files

- Accounting introduced an other small feature called smart files
- `sys.proc="mgm.cmd=accounting&mgm.subcmd=report&mgm.format=fuse"` attribute on file
- Opening the file will execute the command instead
- Get output of command as it was the content of file
- Easy to run it from a browser (HTTP) and get the JSON

State of accounting

- Tested on PPS for a while by WLCG ops people
- Hervé started to setup the production instances for accounting
 - EOSUSER
 - EOSATLAS → they want a different older format
 - EOSCMS

Thank you! Questions?

József Makai, CERN

