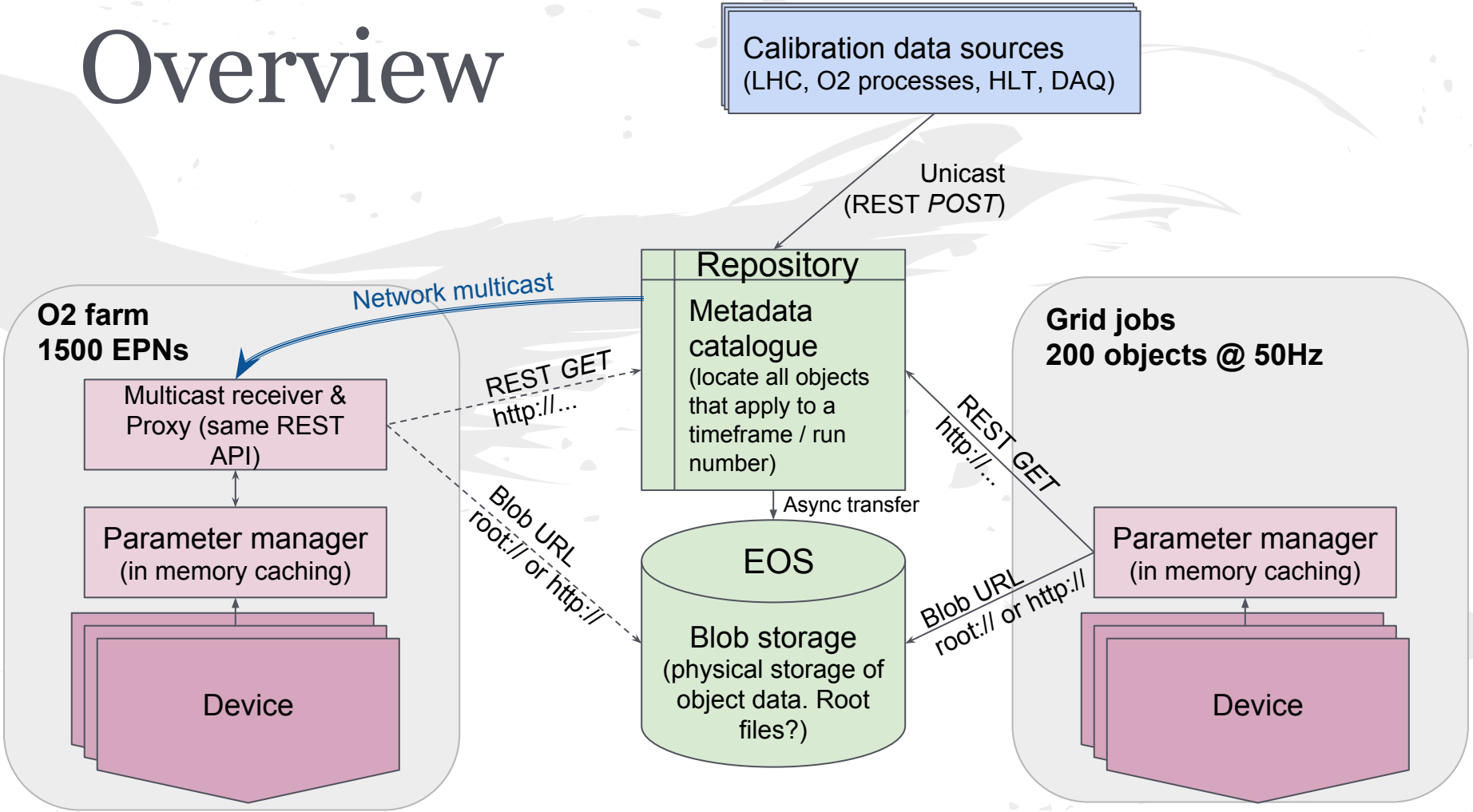
CCDB
Conditions DB for Run 3

costin.grigoras@cern.ch

Overview


Calibration data sources
(LHC, O2 processes, HLT, DAQ)
Unicast
(REST *POST*)
Repository
Metadata catalogue
(locate all objects that apply to a timeframe / run number)
Network multicast
O2 farm
1500 EPNs
Multicast receiver & Proxy (same REST API)
Parameter manager
(in memory caching)
Device
REST GET
http://...
Blob URL
root:// or http://
Async transfer
EOS
Blob storage
(physical storage of object data. Root files?)
Grid jobs
200 objects @ 50Hz
REST GET
http://...
Blob URL
root:// or http://
Parameter manager
(in memory caching)
Device

CCDB repository

- Implemented as a HTTP REST web server using an embedded Tomcat engine
- Actual metadata backend is completely hidden from the client
- The client has to follow any redirects to access the binary blob of serialized calibration data

SQL backend status

Server side available now at

http://ccdb-test.cern.ch:8080/

DNS alias for a decent machine

HP DL360 Gen9, 256GB RAM, 56 HT cores

One 1.8TB SSD

REST methods overview

/

- POST - upload new objects
- PUT - update existing objects
- GET - retrieve the content (query by path / time)
- HEAD - only the headers (metadata)
- DELETE - obvious

/browse

- GET - listing of all matching objects and subfolders, with regexp matching for the namespace

/latest

- GET - like the above but only the most recent object in each folder is returned == current calibration data
  equivalent to today's `***find***` in AliEn

Query format

…/Task/Detector/[more/]/*tStart*[/tEnd][/UUID][/key=value]

Folder structure is between 1 and 10 levels deep

For most requests the reference time is mandatory

Additional HTTP headers:

- *If-None-Match* : client cached object to validate
- *If-Not-After* : snapshot timestamp to limit queries by

POST /

Uploads the blob of a new object to a path

```
$ curl -i \
    -F blob=@/etc/passwd \
    ccdb-test:8080/Task/Detector/Subdetector/1/100000/quality=2
```

```
HTTP/1.1 201
Date: Fri, 27 Apr 2018 07:50:26 GMT
Valid-From: 1
Valid-Until: 100000
Created: 1524815426777
ETag: "a61d9c90-49ef-11e8-8098-200114580202"
Last-Modified: Fri, 27 Apr 2018 07:50:26 GMT
quality: 2
Location: http://ccdb-test.cern.ch:8080/download/a61d9c90-49ef-11e8-8098-200114580202
Content-Length: 0
```

PUT /

Modify end of validity and/or set metadata

```
$ curl -i -X PUT \
'ccdb-test:8080/Task/Detector/1/1400000/quality=2?quality=1&checked=true'
```

New validity

New metadata

```
HTTP/1.1 204
Date: Fri, 27 Apr 2018 08:59:38 GMT
Valid-From: 1
Valid-Until: 1400000
InitialValidityLimit: 100000000
Created: 1524815426754
ETag: "a61a1a20-49ef-11e8-8098-200114580202"
Last-Modified: Fri, 27 Apr 2018 07:50:26 GMT
UpdatedFrom: 2001:1458:202:56:0:0:101:b2e7
checked: true
quality: 1
Location: http://ccdb-test.cern.ch:8080/download/a61a1a20-49ef-11e8-8098-200114580202
```

PUT / (using IDs)

Same as before, but making sure that exactly this version of the object is modified

```
$ curl -i -X PUT \
'ccdb-test:8080/Task/Detector/1/123/a61a1a20-49ef-11e8-8098-200114580202'
```

GET /

It might **return** the content or **redirect** to it!

$ curl -i -L ccdb-test:8080/Task/Detector/Subdetector/60000/quality=2

```
HTTP/1.1 200
Content-Location: /download/a61d9c90-49ef-11e8-8098-200114580202
Content-Location: root://eosalice.cern.ch:1094//04/19198/a61d9c90-49ef-11e8-8098-200114580202.ccdb
<...other alternative locations if available>
Date: Fri, 27 Apr 2018 09:04:33 GMT
Valid-From: 1
Valid-Until: 100000
Created: 1524815426777
ETag: "a61d9c90-49ef-11e8-8098-200114580202"
Last-Modified: Fri, 27 Apr 2018 07:50:26 GMT
quality: 2
Accept-Ranges: bytes
Content-Disposition: inline;filename="passwd"
Content-MD5: 160aed2f4622e9c99a082e52a0450e49
Content-Type: application/octet-stream
Content-Length: 3060

...
```

All replicas as alternative locations of this content

GET / (with client **cached** obj)

```
$ curl -i \
      -H 'If-None-Match: "a61d9c90-49ef-11e8-8098-200114580202"' \
      ccdb-test:8080/Task/Detector/Subdetector/91234
```

```
HTTP/1.1 304
Date: Fri, 27 Apr 2018 09:08:20 GMT
```

The client is free to **reuse** the object it has since it is still valid for the next timestamp that is being processed (authoritative answer)

GET / (with **snapshot**)

$ curl -i -H '**If-Not-After**: 1524815426777' \
ccdb-test:8080/Task/Detector/Subdetector/70000
HTTP/1.1 200
ETag: "a61d9c90-49ef-11e8-8098-200114580202"
Last-Modified: Fri, 27 Apr 2018 **07:50:26 GMT**

Current version

$ curl -i -H '**If-Not-After**: 1524815426776' \
ccdb-test:8080/Task/Detector/Subdetector/70000
HTTP/1.1 200
ETag: "fb6c8f40-49ee-11e8-8098-200114580202"
Last-Modified: Fri, 27 Apr 2018 **07:45:40 GMT**

Previous version

$ curl -i -H '**If-Not-After**: 100000000000' \
ccdb-test:8080/Task/Detector/Subdetector/70000

HTTP/1.1 **404**

No data back then

DELETE /

Same logic, the latest matching version is **deleted**
or it can be restricted to an object ID

Remove the most recent object version

```
$ curl -i -X DELETE \
ccdb-test:8080/Task/Detector/60000

$ curl -i -X DELETE \
ccdb-test:8080/Task/Detector/1/fb6c8f40-49ee-11e8-8098-200114580202
```

```
HTTP/1.1 204
ETag: "fb6c8f40-49ee-11e8-8098-200114580202"
```

GET /**browse**/ and /**latest**/

Human or machine browsable content of the repository

Reference timestamp is optional (default *now*) but any constraints passed will be applied (metadata filtering, ID, snapshot time etc)

Output format controllable by the **Accept** header or URL parameter

GET /**browse**/ and /**latest**/

Human interface to investigate / debug

http://ccdb-test.cern.ch:8080/browse/Task/Detector/

ID	Valid from	Valid until	Initial validity limit	Created at	Last modified	MD5	File name	Content type	Size	Path	Metadata	Replicas
a54a1b10-6fae-11e8-b409-7f0000015566	1 01 Jan 1970 01:00	100000000 02 Jan 1970 04:46	100000000 02 Jan 1970 04:46	1528965652289 14 Jun 2018 10:40	1528965652289 14 Jun 2018 10:40	bb374f437ec48dfc708508874c5a2dd6	u2.sh	application/octet-stream	468	Task/Detector	partName blob	• 0 • 370 • 363 • 332 • 222 • 287

Subfolder
Task/Detector/Subdetector

Other formats

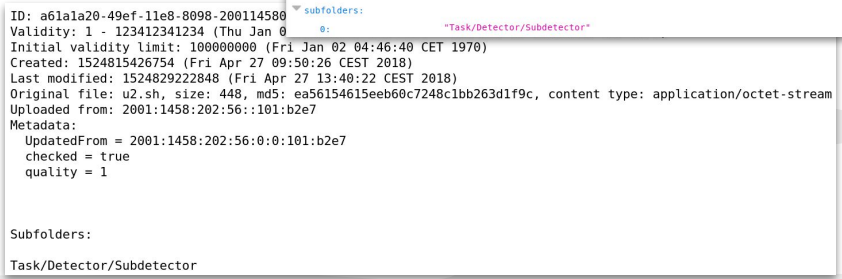
Accept=**text/xml**

```
- <document>
  - <objects>
    - <object id="a54a1b10-6fae-11e8-b409-7f0000015566" validFrom="1" validUntil="100000000" initialValidity="100000000"
      created="1528965652289" lastModified="1528965652289" md5="bb374f437ec48dfc708508874c5a2dd6" fileName="u2.sh"
      contentType="application/octet-stream" size="468" path="Task/Detector">
        <metadata key="partName" value="blob"/>
        <replica id="0" addr="http://pcalice79:8080/download/a54a1b10-6fae-11e8-b409-7f0000015566"/>
        <replica id="370" addr="root://alicemgm0.lbl.gov:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"/>
        <replica id="363" addr="root://lxaliafrd.gsi.de:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"/>
        <replica id="332" addr="root://eosalice.cern.ch:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"/>
        <replica id="222" addr="root://srm.grid.sara.nl:1094//pnfs/grid.sara.nl/data/alice/disktup//04/19198/a54a1b10-6fae-11e8-b409-7f00
        <replica id="287" addr="root://alice-xrd.to.infn.it:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"/>
      </object>
    </objects>
  - <folders>
      <path name="Task/Detector/Subdetector"/>
    </folders>
</document>
```

Accept=**application/json**

```
objects:
  0:
    id:                "a54a1b10-6fae-11e8-b409-7f0000015566"
    validFrom:         "1"
    validUntil:        "100000000"
    initialValidity:   "100000000"
    createTime:        "1528965652289"
    lastModified:      "1528965652289"
    MD5:               "bb374f437ec48dfc708508874c5a2dd6"
    fileName:          "u2.sh"
    contentType:       "application/octet-stream"
    size:              "468"
    path:              "Task/Detector"
    partName:          "blob"
    replica0:          "http://pcalice79:8080/download/a54a1b10-6fae-11e8-b409-7f0000015566"
    replica370:        "root://alicemgm0.lbl.gov:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"
    replica363:        "root://lxaliafrd.gsi.de:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"
    replica332:        "root://eosalice.cern.ch:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"
    replica222:        "root://srm.grid.sara.nl:1094//pnfs/grid.sara.nl/data/alice/disktup//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"
    replica287:        "root://alice-xrd.to.infn.it:1094//04/19198/a54a1b10-6fae-11e8-b409-7f0000015566.ccdb"
subfolders:
  0:                   "Task/Detector/Subdetector"
```

Accept=**text/plain**



```
ID: a61a1a20-49ef-11e8-8098-200114580
Validity: 1 - 123412341234 (Thu Jan 0
Initial validity limit: 100000000 (Fri Jan 02 04:46:40 CET 1970)
Created: 1524815426754 (Fri Apr 27 09:50:26 CEST 2018)
Last modified: 1524829222848 (Fri Apr 27 13:40:22 CEST 2018)
Original file: u2.sh, size: 448, md5: ea56154615eeb60c7248c1bb263d1f9c, content type: application/octet-stream
Uploaded from: 2001:1458:202:56::101:b2e7
Metadata:
  UpdatedFrom = 2001:1458:202:56:0:0:101:b2e7
  checked = true
  quality = 1



Subfolders:

Task/Detector/Subdetector
```

HTTP performance

Test the HTTP calls overhead with a ping RTT=0.305ms and 1000 random DELETE queries (no body, just headers)

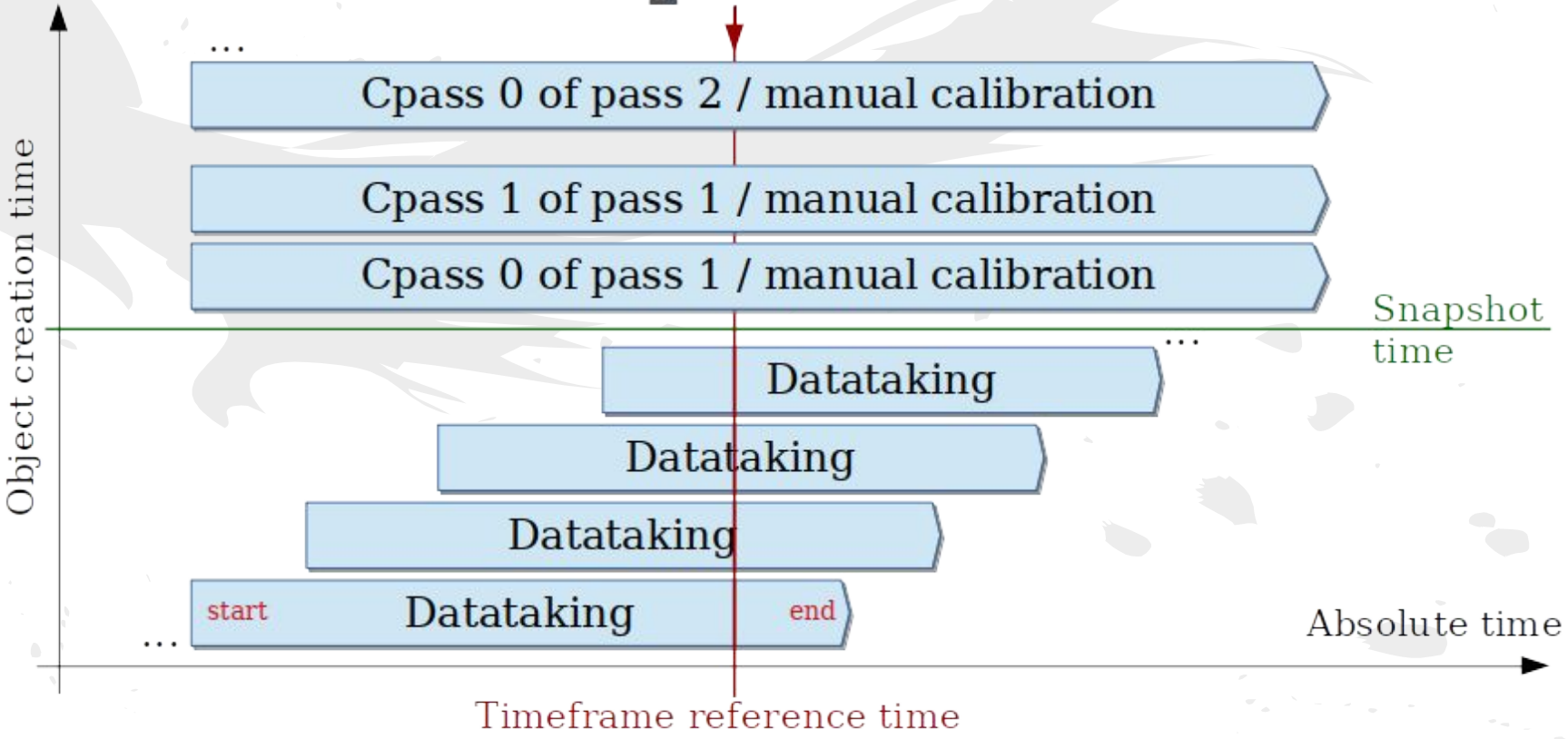
Main qry	Servlet run	Client time	Overhead	Scenario
7.757 ms	8.201 ms	9.123 ms	**0.922 ms**	1 req / conn
7.609 ms	8.061 ms	8.491 ms	**0.43 ms**	10 req / conn
7.533 ms	7.932 ms	8.331 ms	**0.399 ms**	100 req / conn
7.392 ms	7.786 ms	8.179 ms	**0.393 ms**	1000 req / conn
7.425 ms	7.8 ms	7.849 ms	**0.049 ms**	1000 req / conn

+ keep-alive

+ pipelining

Tomcat 9.0 supports HTTP/2, request multiplexing with it should have similar performance to pipelining with HTTP/1.1 - to be tested

Metadata queries



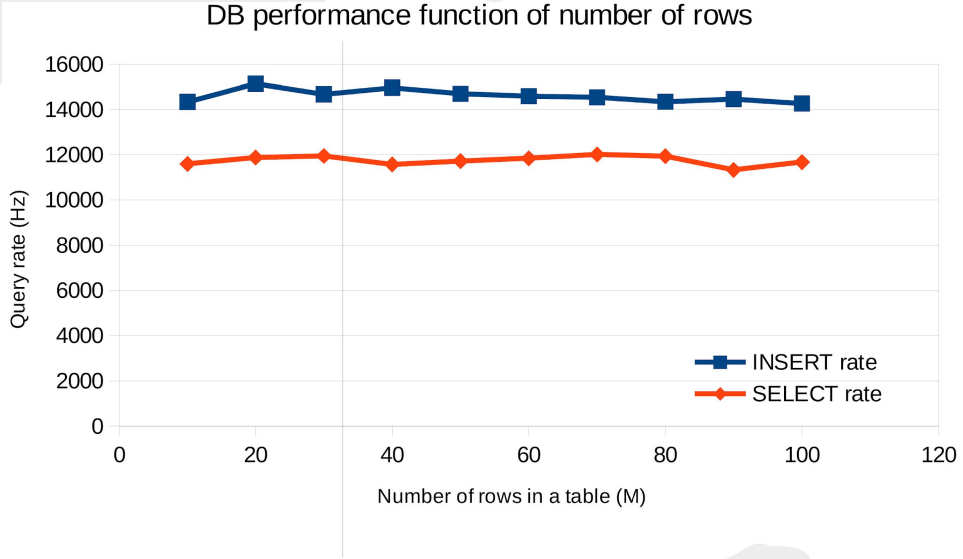
PostgreSQL backend

- Using GiST indexes for time range intervals
- Direct mapping of calibration object time validity ranges to a tsrange structure (timestamp without time zone)
- 8 bytes storage, 1 microsecond resolution for the interval boundaries
- More indexes for direct object lookup and parameter constraints

```
      Table "public.ccdb"
      Column       |   Type
-------------------+----------
 id                | uuid
 pathid            | integer
 validity          | tsrange
 createtime        | bigint
 replicas          | integer[]
 size              | bigint
 md5               | uuid
 filename          | text
 contenttype       | integer
 uploadedfrom      | inet
 initialvalidity   | bigint
 metadata          | hstore
 lastmodified      | bigint
Indexes:
      "ccdb_pkey" PRIMARY KEY, btree (id)
      "ccdb_pathid_idx" btree (pathid)
      "ccdb_validity_idx" gist (validity)
```

PostgreSQL performance

- 15kHz INSERT
- 12kHz SELECT
- Well suitable for CCDB target rates
- And then we can shard by year or namespace



No. of objects at end of Run 4 for a single parameter

Summary

A ready to use, full fledged central metadata & blob repository

http://ccdb-test.cern.ch:8080/

For development purposes a local service can be run on users' machines

In development is the multicast-based distribution within the EPN cluster

Open questions

Who will generate the calibration objects?

- No Shuttle service equivalent
- Each detector has to implement its handling of calibration data
- Common API for writing / reading objects ? (starting from Barth's QC code)

Monitoring

How do we know what object should be published by the detectors?

- What is the expected frequency?
- When to raise alarms if they are not seen?
- Explicit failure codes?

Namespaces

The hierarchy is flexible (folder-like structure)

No need to go through the year/run number lookups

But how do we organize the IDEAL / RESIDUAL namespaces?

- One extra level or root folders?
- Do we allow arbitrary new custom storages?

How to authenticate ops?

Accept all updates from the O2 nodes? Yes / no

Open read access from Grid jobs? Yes / no

Write access from Grid jobs? Yes / no

Which authentication mechanism? Open / AliEn

What else are we overlooking?