Improving the performance of the LFC

Lana Abadie, CERN, IT-GD

JRA1 , 24-26 Oct 2007, CERN

gLite Lightweight Middleware for Grid Computing

www.eu-egee.org

Information Society and Media

What parameters ?

- **LFC too slow ? What parameters have an impact?**
  - Hardware quality
  - Oracle usually more scalable and faster for large tables
  - Load on the LFC server
  - Time needed to process and execute the job on the server: some requests are more time-consuming
  - RTT : the further away the LFC is, the slower it will be
  - Security issue (authentication) : the more secure, the slower
- **What can be done to tune the LFC performance**
  - Avoiding the time spent for authentication -> **session**
  - Avoiding the RTT -> **bulk methods**

Session

- **Authentication done once, at the beginning of the session**
  - TCP connection kept alive
  - All the LFC operations performed in a session will use the same TCP connection
  - 8 RTT during authentication !
- **A session is automatically dropped by the server: the session must be idle for less than 60 seconds**
- **Usable from the LFC C and python API**

```
rescode = lfc_startsess (lfc_host, "Starting adding replicas");
// adding replicas to the LFC
(void) lfc_endsess ();
```

- **Performance : x10-15 times faster**

Bulk methods

- **Reduce the number of RT**
  - Very useful when the catalog is remote
- **ATLAS request : bulk methods**
  - Getting the list of replicas
  - Passing a list of guids to be deleted
  - Passing a list of lfns with some filtering options to be deleted
- **Performance * 15-20 times faster**

LFC bulk methods

```
//deleting replicas (if force=1) and LFNs if mapped to one of guid in the list
   of guids
```

- `int DLL_DECL lfc_delfilesbyguid(int nbguids, const char **guids, int force, int *nbstatuses, int **statuses)`

```
//deleting replicas (if force=1) and LFNs (specified by the **paths)
```

- `int DLL_DECL lfc_delfilesbyname(int nbfiles, const char **paths, int force, int *nbstatuses, int **statuses)`

```
//deleting replicas (if force=1) and LFNs if LFN starts with
   concat(path,pattern)
```

- `int DLL_DECL lfc_delfilesbypattern(const char *path, const char *pattern, int force, int *nbstatuses, struct Cns_filestatus **statuses)`

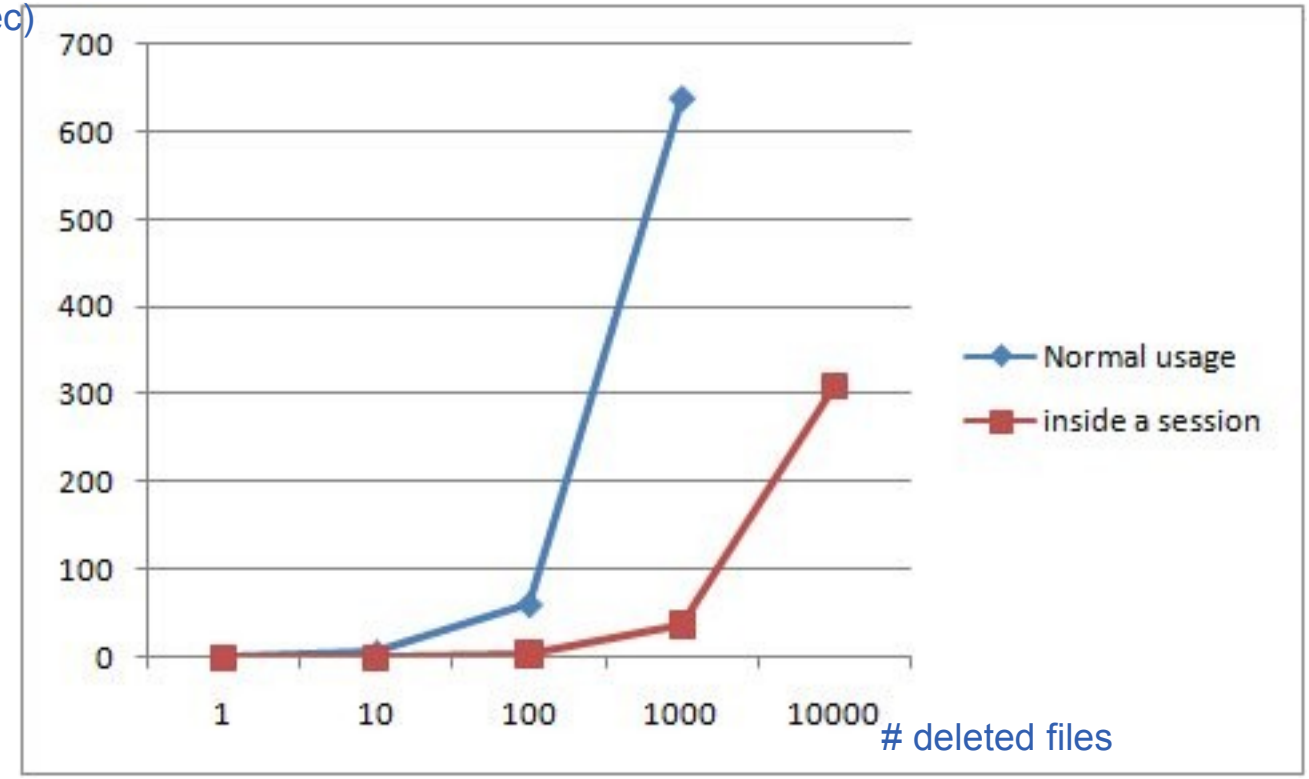
```
//deleting replicas stored on the given SE and associated with
one guid provided by the list of guids
```

- `int DLL_DECL lfc_delreplicas(int nbguids, const char **guids, char *se, int *nbstatuses, int **statuses)`

```
//list the content of the directory + replica info if matches the
pattern
```

- `struct lfc_direnrep DLL_DECL * lfc_readdirxp(Cns_DIR *dirp, char *pattern, char *se)`

Performance study 1/3



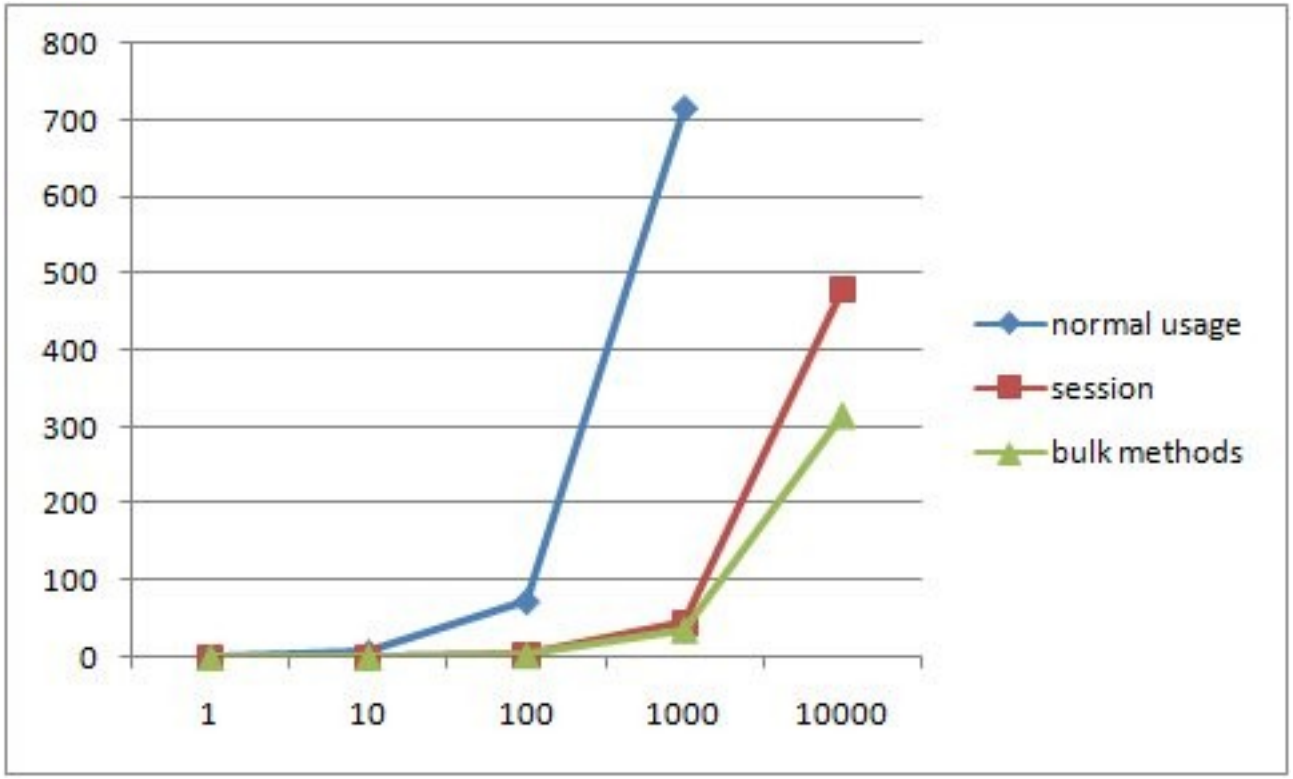
LFC : prod-lfc-shared-central.cern.ch, oracle

Ping : 0.493 ms

Session : up to 20 times faster

Authentication /operation : ≈ 0.3 sec

Performance study 2/3



LFC : lxb0986v2.cern.ch : VM, no load, MySQL

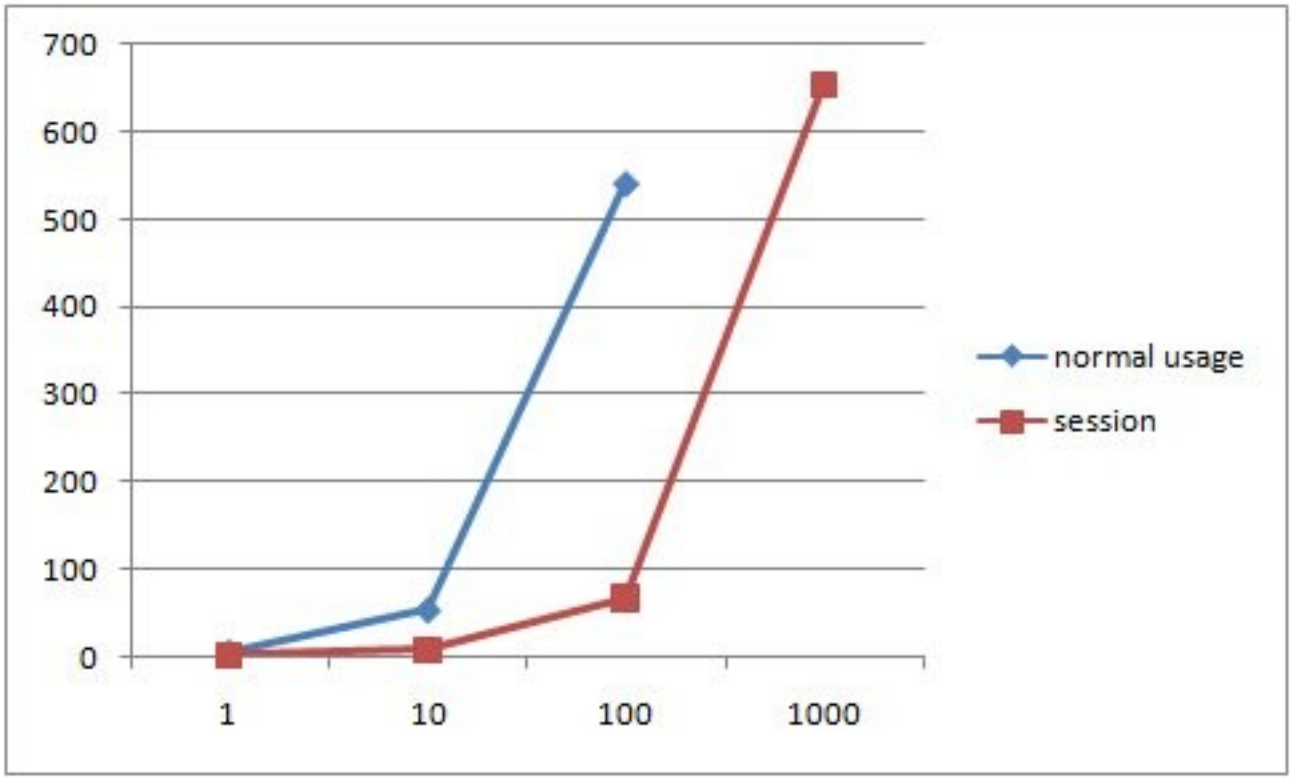
Ping : 0.380 ms

Session : up to 20 times faster

Authentication /operation : ≈ 0.3 sec

The LFC production is faster -> hw quality, VM

Performance study 3/3



LFC : lfc.grid.sinica.edu.tw : production, MySQL

Ping : 312 ms

Session : up to 10 times faster

Authentication /operation : ≈ 3 sec

Impact of the RTT

Conclusion

- **Usage of session and bulk methods**
  - Performance is far better
  - Satisfaction from the LHC experiments
- **Needs to make more tests for the bulk methods**
  - Measurements with a remote LFC
  - Influence of the nb of guids given