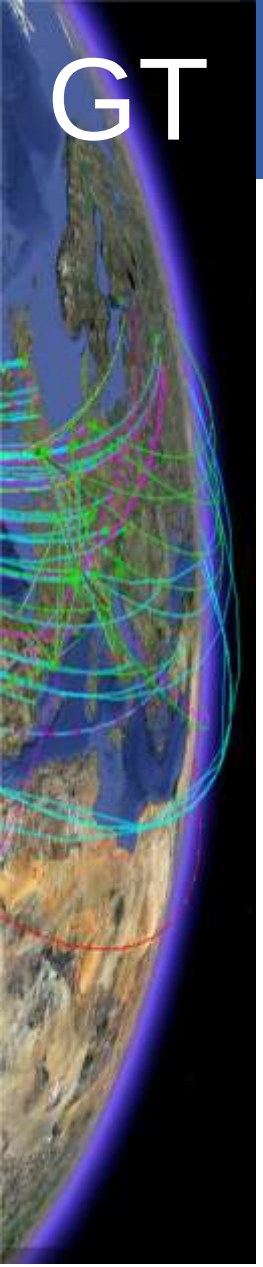
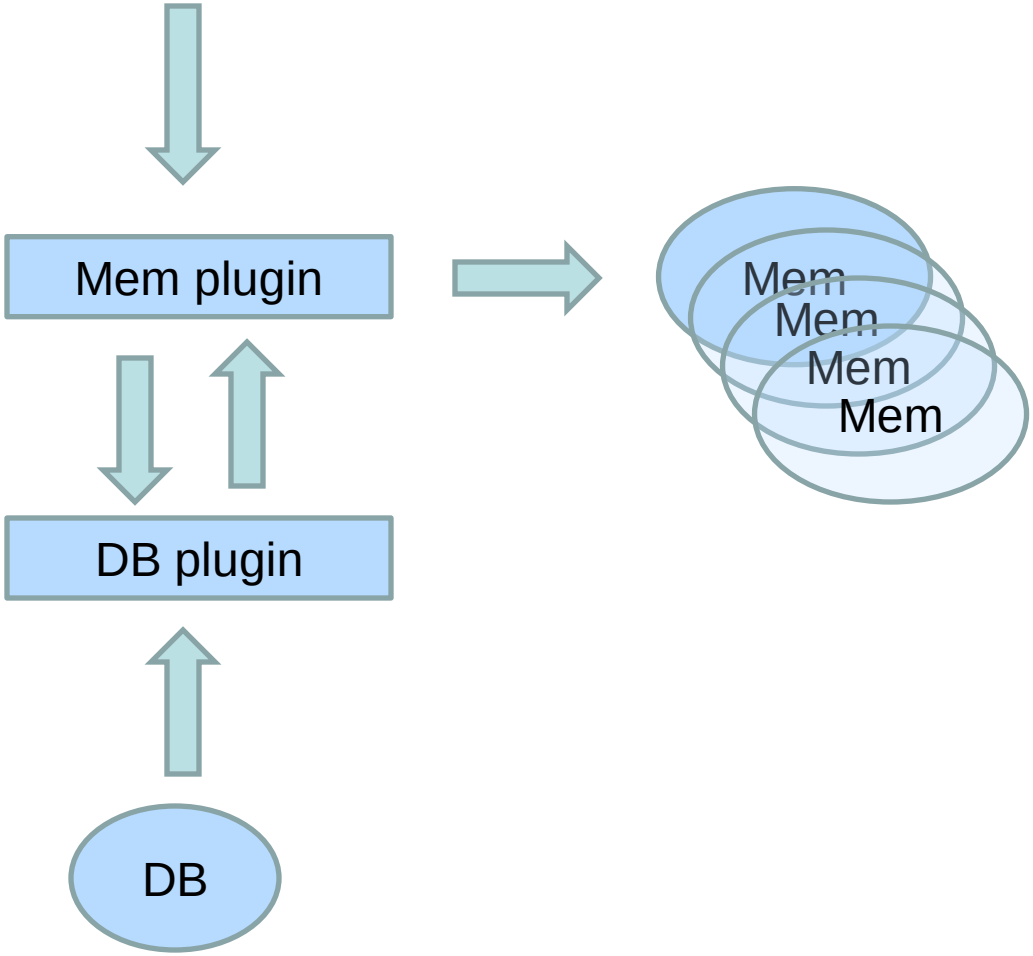


Memcached for dmlite

CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it

Memcached in the stack



API functions affected

- extendedStat | readLink
  - create, unlink, makeDir, removeDir, symlink, chmod, chown, setACL, utime
- get | getReplicas
  - addReplica, deleteReplica, replicaSet*
- openDir | readDir | closeDir
- getComment
  - setComment

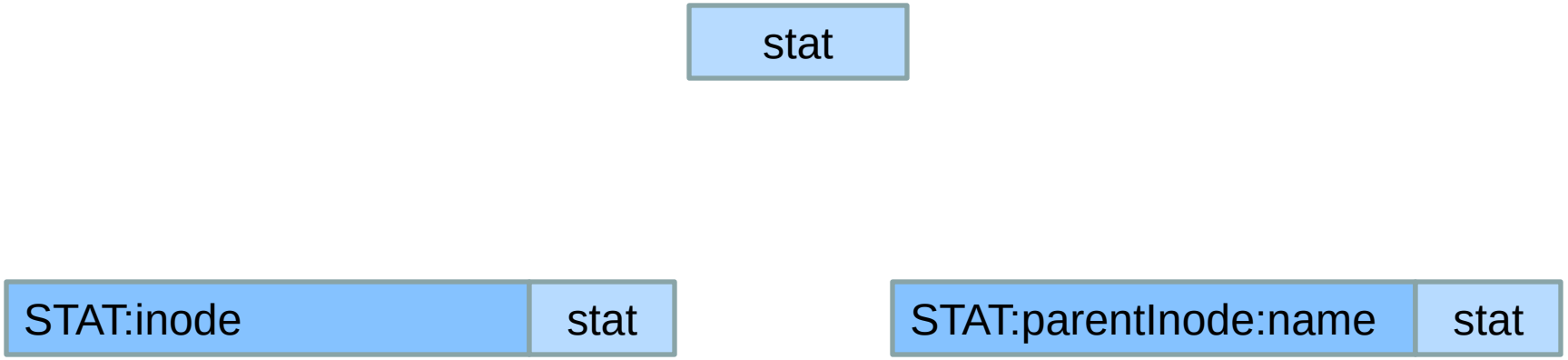
extendedStat /1

- extendedStat(path) ← unchanged
- extendedStat(inode)
  extendedStat(parentInode, name):

```
memcKey = keyFromAny(STAT, inode)
memcVal = memcached_get(memcKey)
if (memcVal)
        stat = deserialize(memcVal)
else
        DELEGATE_ASSIGN(stat, extendedStat, inode)
        memcVal = serialize(stat)
        memcached_set(memcKey, memcVal)

return stat
```

extendedStat /2

- Memcached Representation

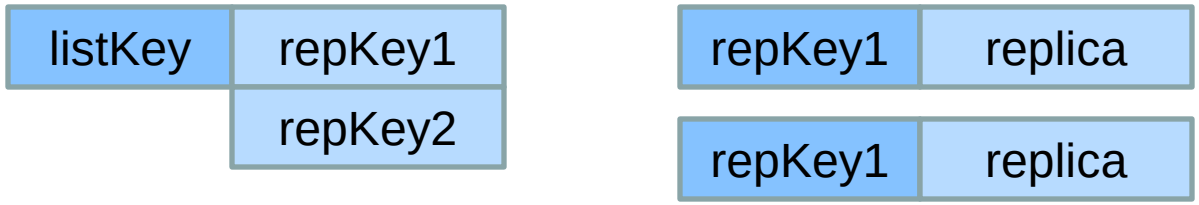


extendedStat /3

- Side effects:
  - create, unlink, … → st_nlink changes of parent
- Solution:
  - memcached_del(inode)
    memcached_del(parentOf(inode))
- In practice:
  - inode = extendedStat(path)
    key = keyFromAny(STAT, inode)
    memcached_del(key)
    pInode = getParent(inode)
    key = keyFromAny(STAT, pInode)
    memcached_del(key)

getReplicas /1

- Memcached List Implementation:



Get the list:
listKey = keyFromAny(DIR, inode)
list = memcached_get(listKey)
if (list)
  replicas = memcached_multiget(list)

memcached_multiget(keyList):
memcached_mget(keyList)
while (val = memcached_fetch())
  list.append(val)

Make the list:
listKey = keyFromAny(DIR, inode)
memcached_set(listKey, NULL)

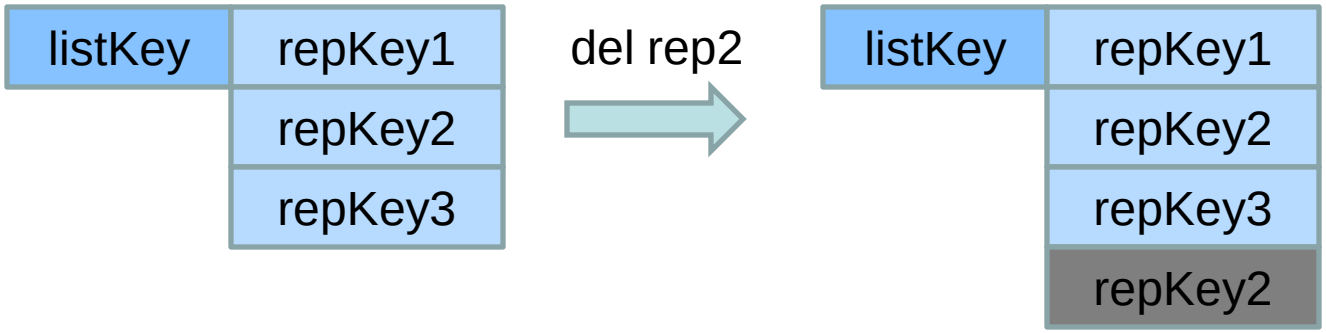
(…)

repKey = keyFromAny(REPLICA, uri)
memcached_set(repKey, replica)
memcached_append(listKey, repKey)

No List Download

getReplicas /2

- Side effects:
  - addReplica, delReplica
- Solution: blacklists

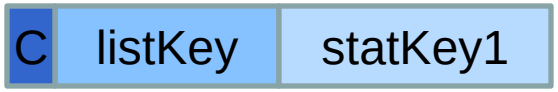


- In Practice:
  - lazy add: only create list item

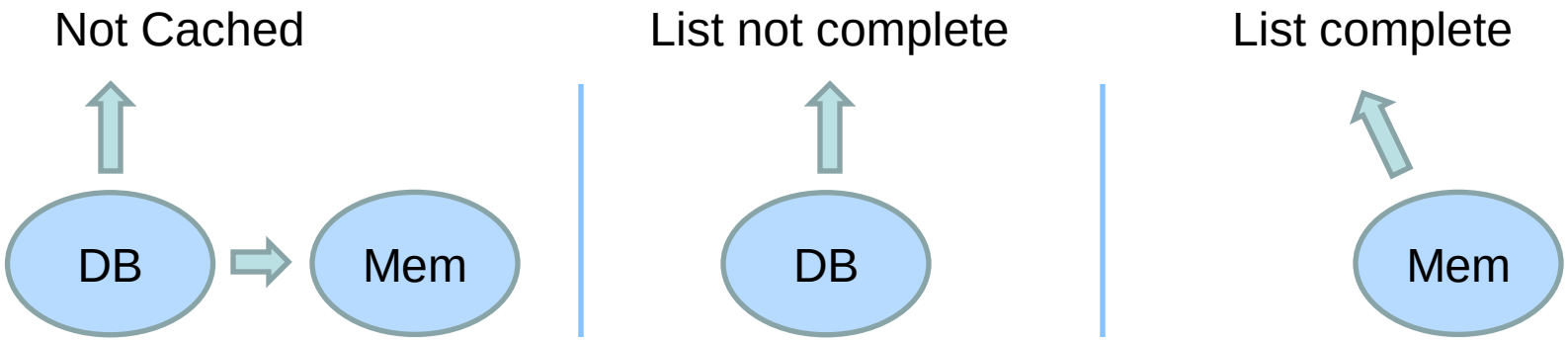
openDir | readdir /1

- cached as list:

- List extension: isComplete bit



- 3 cases:

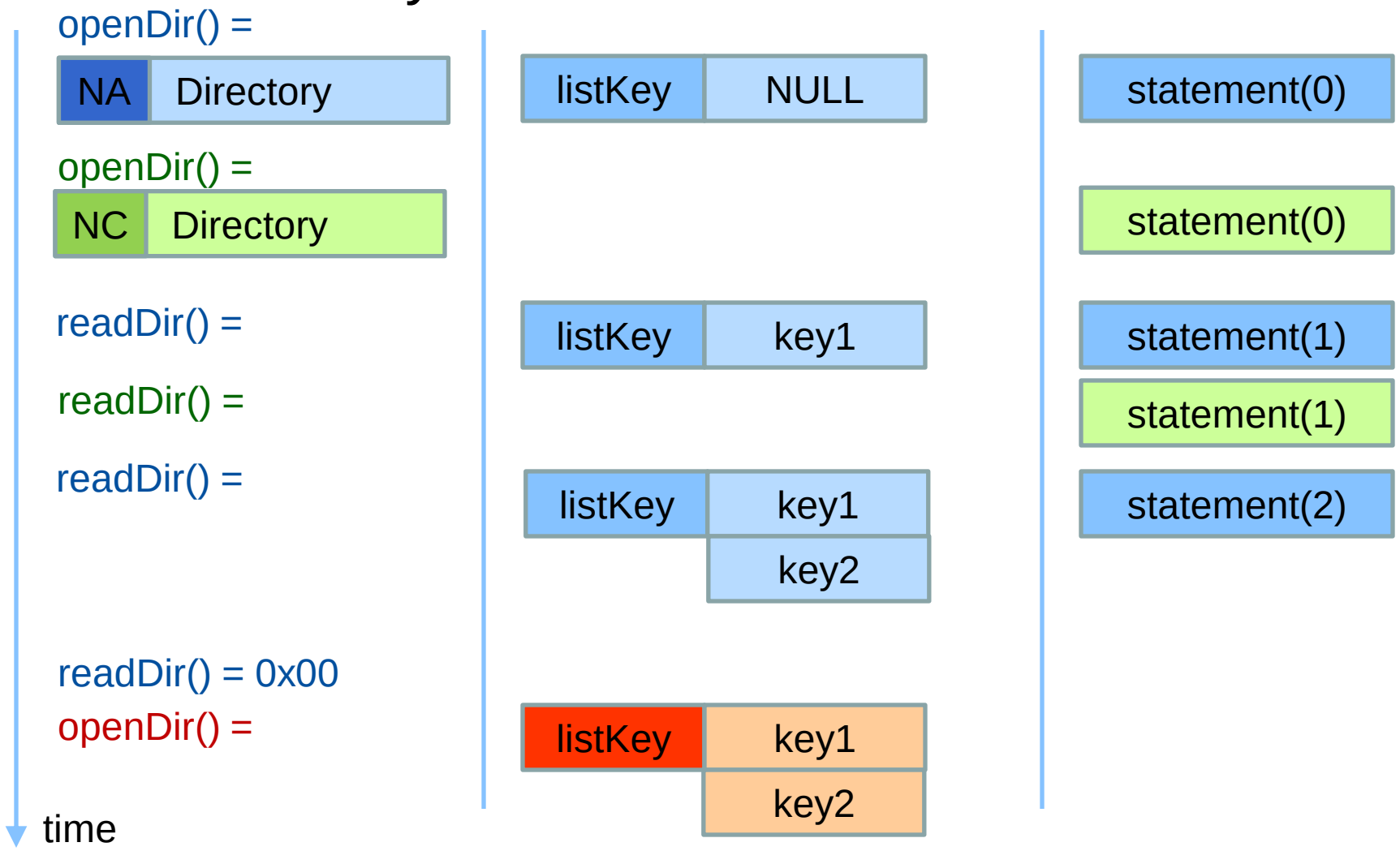


openDir | readdir /2

- Make a List:
- openDir():
  - memcached_set(listKey, NULL, complete = False)
- readDir():
  - memcached_set(stat)
    memcached_append(listKey, stat)
- readDir() == 0x00:
  - mark_list_complete(listKey, complete = True)

openDir | readdir /3

- Concurrency:



Limitations

- memcached 1M limit:
  - Num Replicas per File ~7M
  - Num Files per Dir ~40M
- MemcacheDir hack:
  - Num inodes ~1 Billion
- readDir issue:
  - No lazy file-fetching
  - Memory limited

Google Protobuf

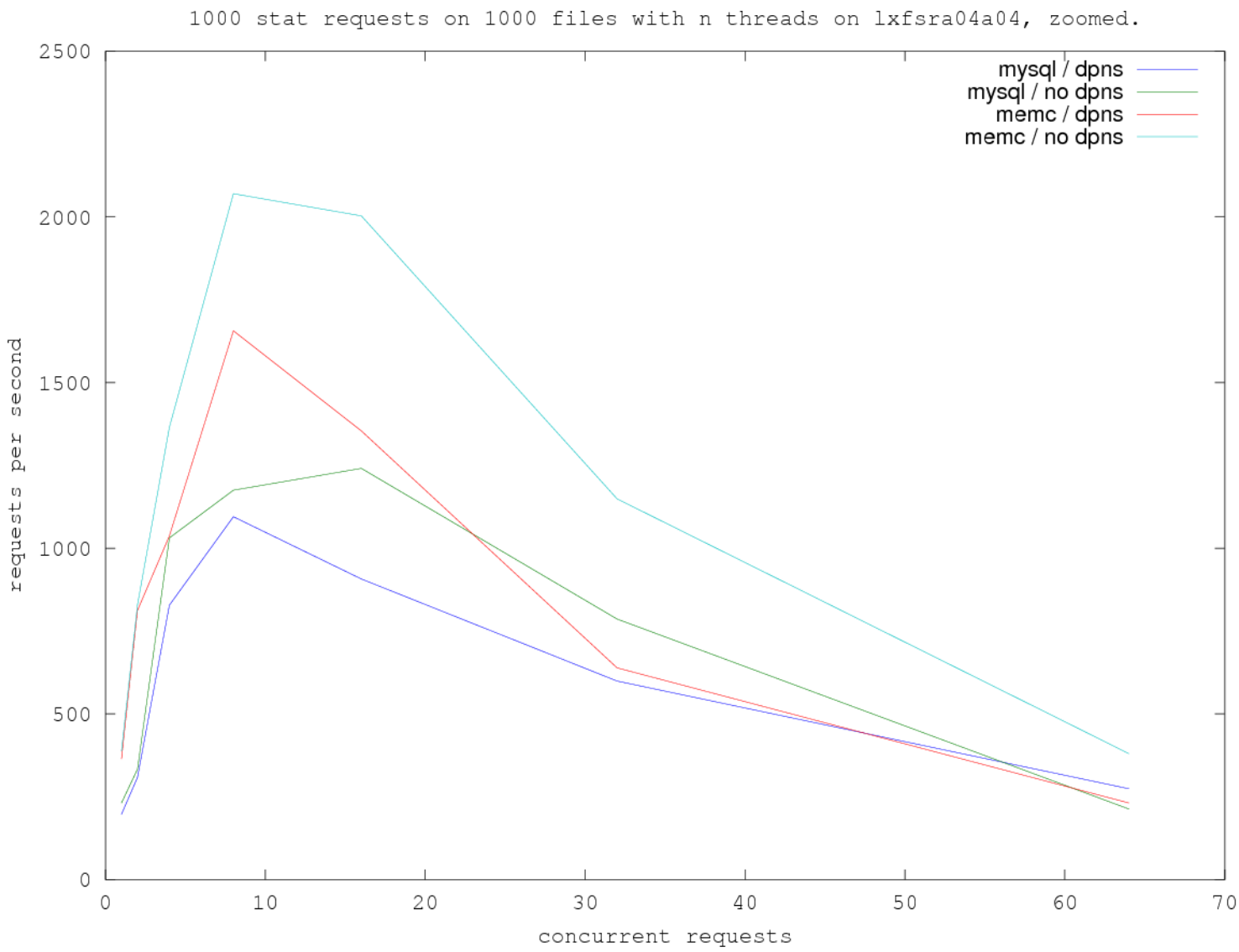
- Example:

.proto file:

```
message SerialKeyList {
  required bool iscomplete = 1;
  repeated SerialKey key = 2;
}

message SerialKey {
  required string key = 1;
  required bool white = 2 [default = true];
}
```

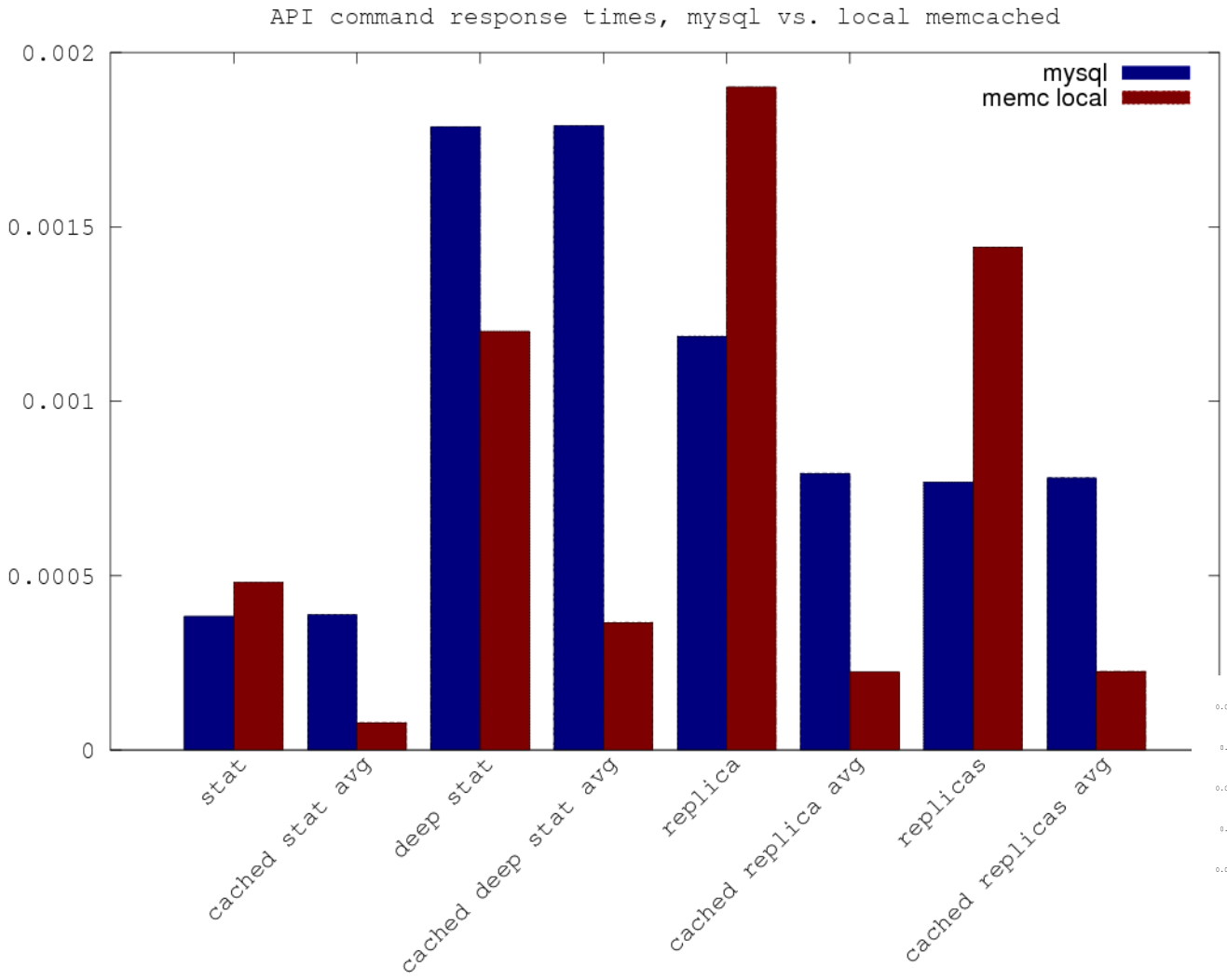
```
def serialize(list = [k1, k2, k3 ,k4])

  SerialKeyList skList;
  string serializedList

  for key in list
    skList.add_key(key)

  serializedList =
            skList.SerializeAsString()

  return serializedList
```

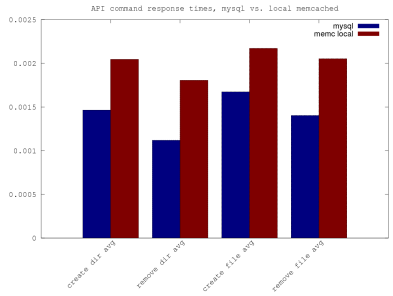
webDAV Performance /1



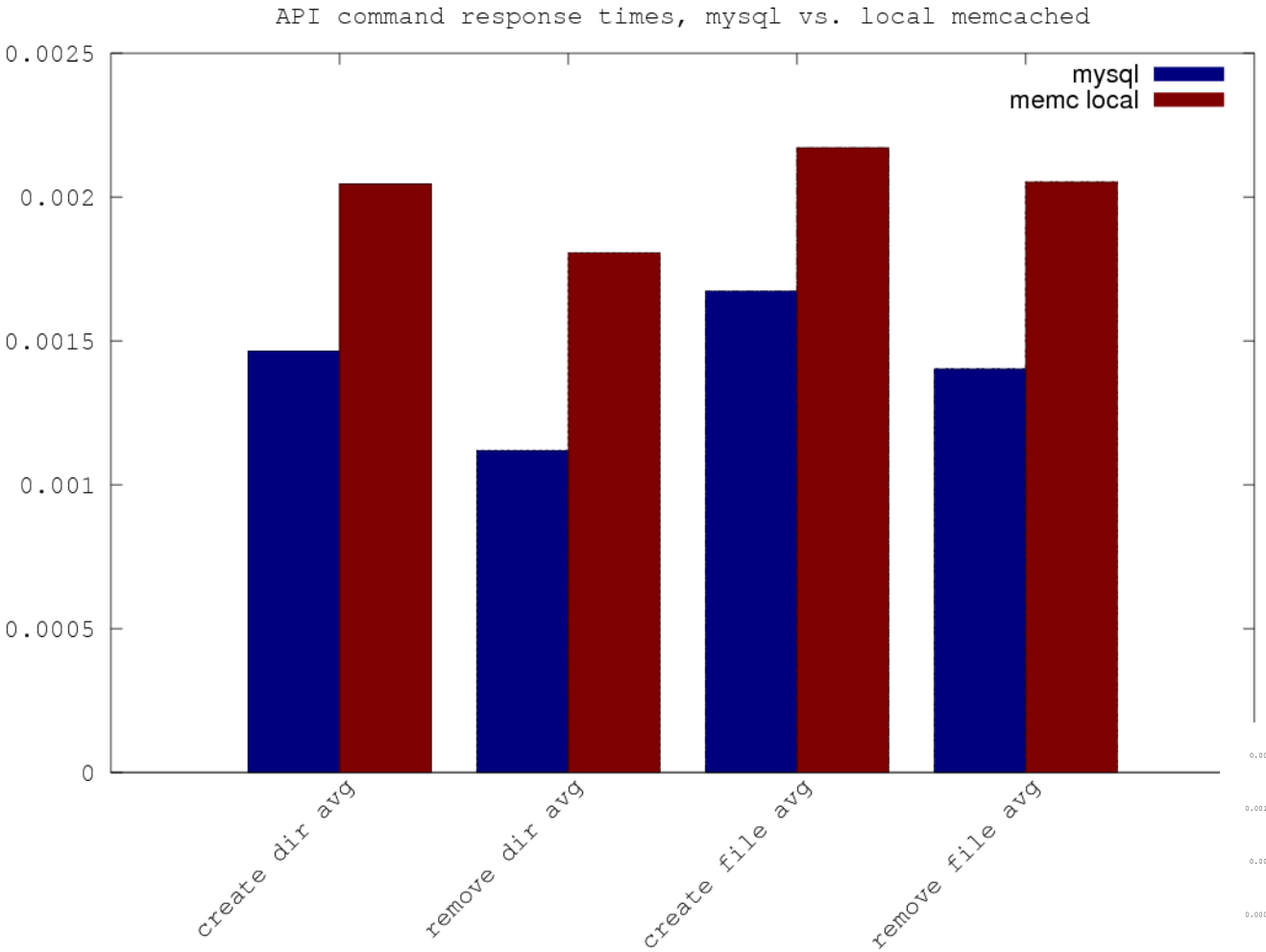
webDAV Performance /2

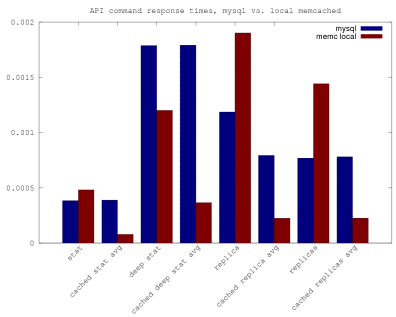
dmlite Performance /1



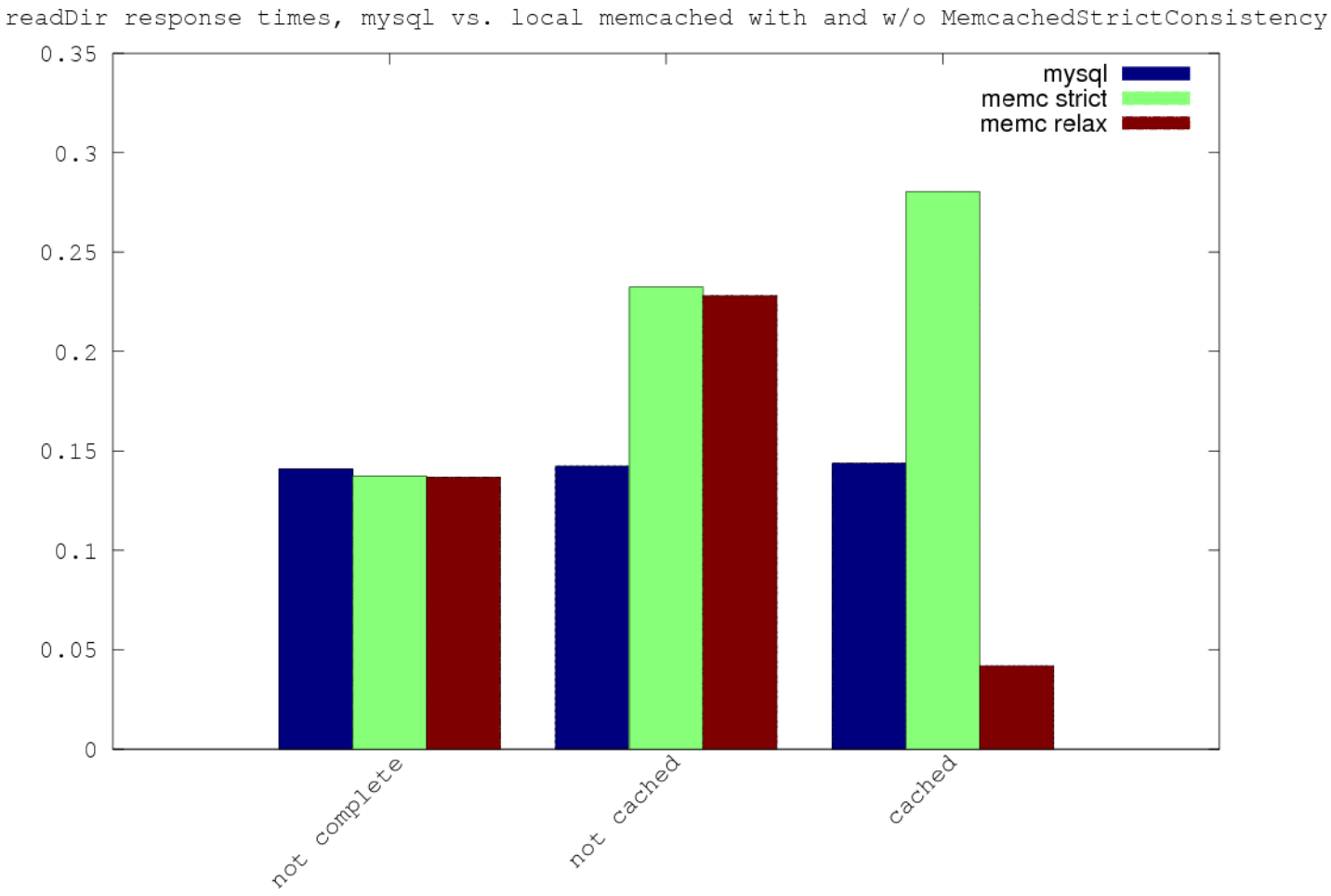


dmlite Performance /2



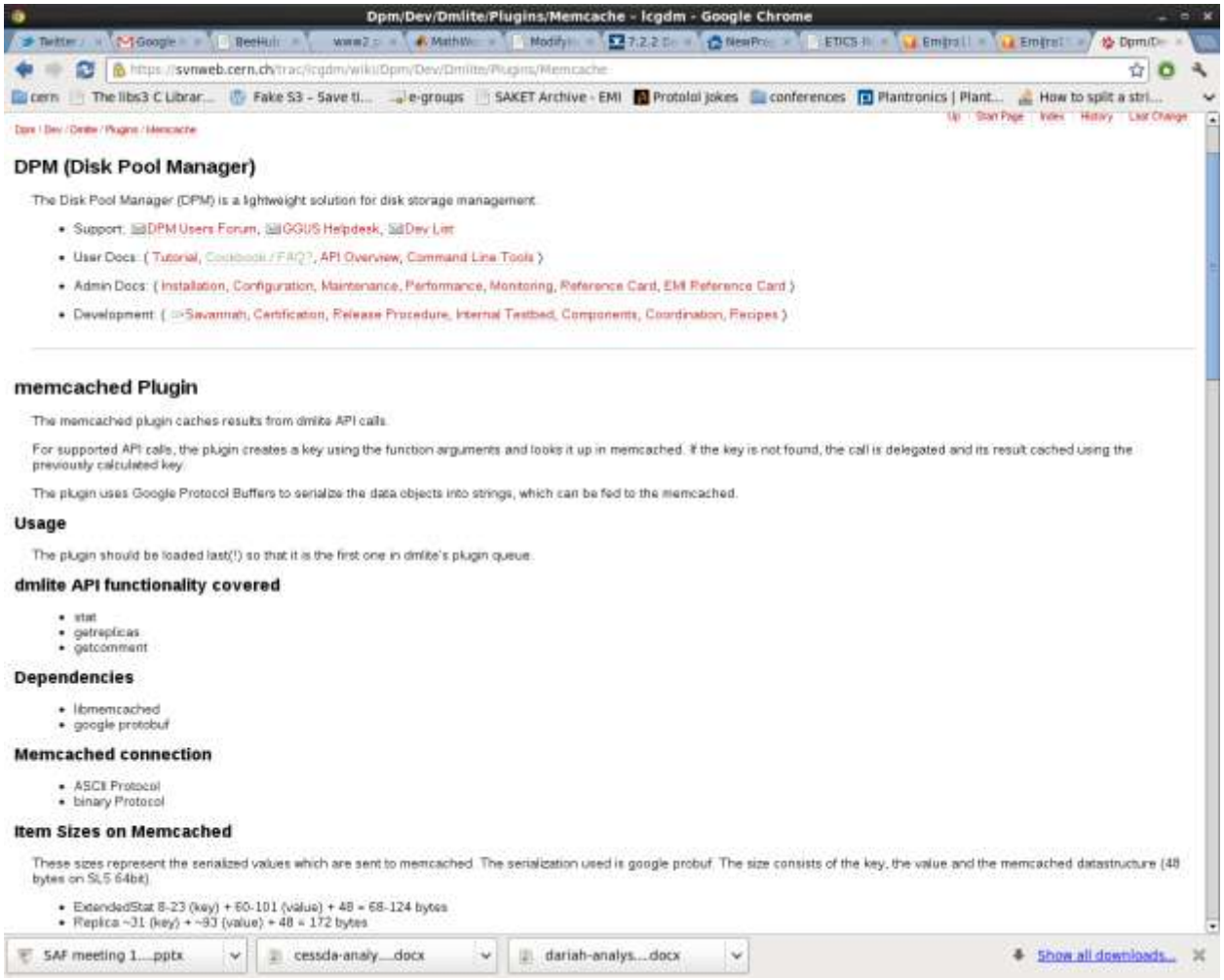


dmlite Performance /3



Plugin documentation

Components > Dmlite > memcached



https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite/Plugins/Memcache

Further info

- memcached.org
- libmemcached.org
- code.google.com/p/protobuf

Performance Improvements?

- save stat by path:
  - extendedStat can fetch the tree all at once
- invalidate the cache asynchronously
  - deletes and utime() to DB
- serialized values from delegate
  - mysql → struct → string

End

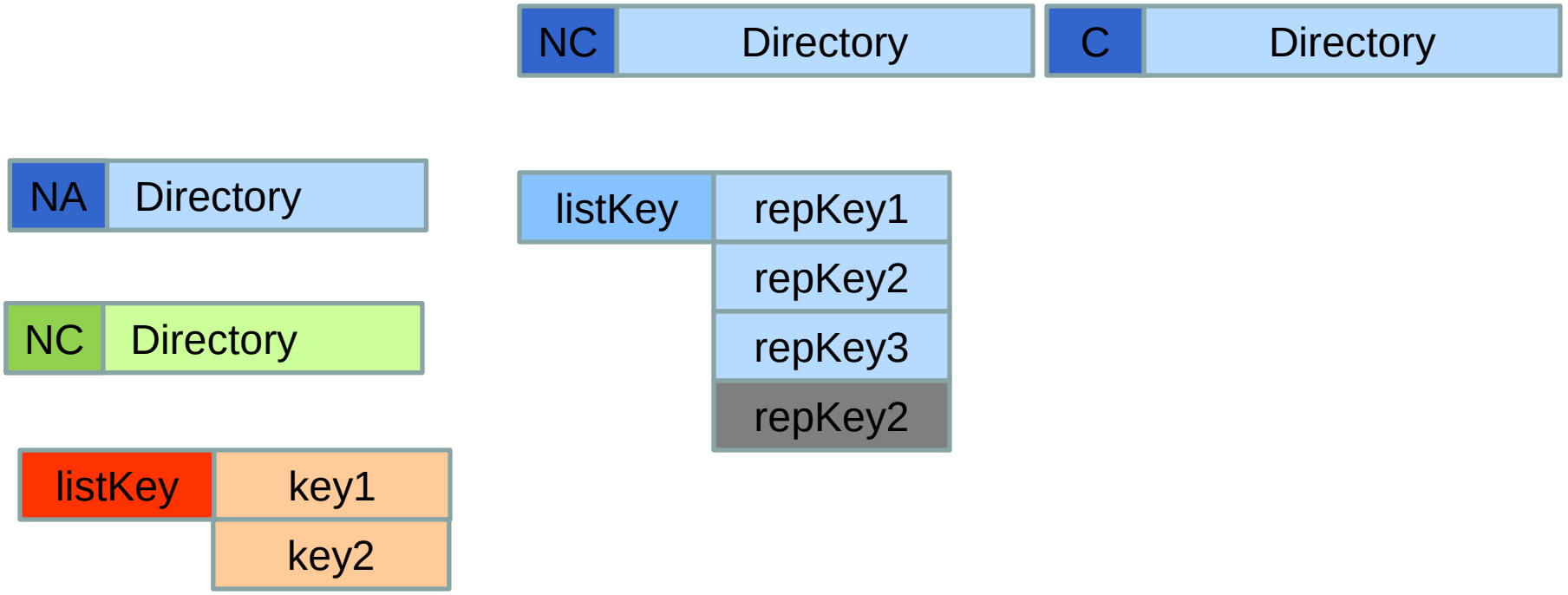
Thank you for your attention!

openDir | readdir /2

- openDir():
- dir = MemcacheDir()
  list = memcached_get(listKey)

  if not list
    DELEGATE_ASSIGN(dir, openDir, path)
    dir.isCached = notcached
  elif list != complete
    dir.isCached = notcomplete
  else
    dir.isCached = complete

Halde


NC
Directory
C
Directory
NA
Directory
listKey
repKey1
repKey2
repKey3
repKey2
NC
Directory
listKey
key1
key2