

Grouper cache communication via database

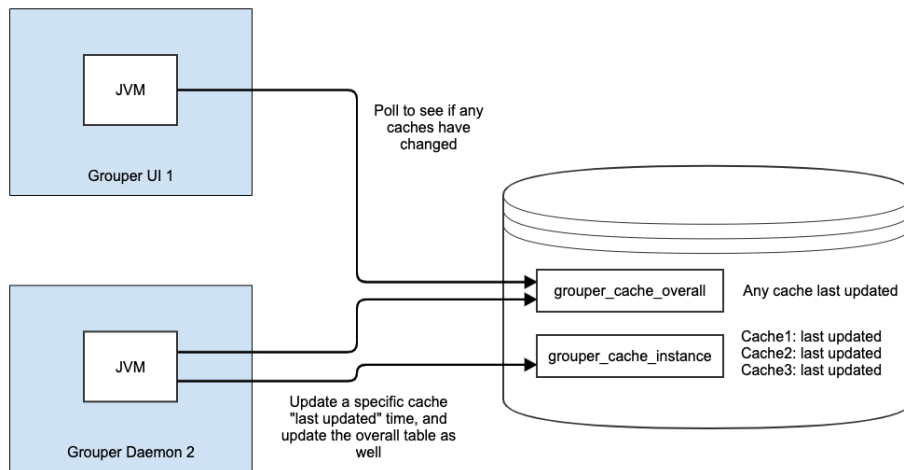
Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

This is an up and coming feature in v2.5.30+

This is intended for caches which do not change often, and when they do change they should clear that cache on all JVM. This is not currently intended to expire individual elements from a cache, but rather the entire cache to be cleared.

Here are some examples:

1. When configuration stored in the database changes
2. If someone changes a password
3. If there is a new "recent membership" group configured
4. If there is a new attribute definition



Expirable cache example

An expirable cache is a simple cache implemented in the grouper client. It does not use ehcache. Items stay in the cache for a max number of minutes

1. Register the cache as expirable in database. Give it a label that will be unique. Probably use the class it is used as prefix

```
ExpirableCache<String, Integer> someCache = new ExpirableCache<String, Integer>(5);

// note: dont register the same cache name twice
someCache.registerDatabaseClearableCache("edu.internet2.middleware.grouper.cache.
GrouperCacheDatabaseTest.myTest");
```

2. Notify that there are changes

```
// you probably will change something in the cache though this isnt necessary
someCache.put("test", 5);

// then tell the database to clear all these caches in other JVMs
someCache.notifyDatabaseOfChanges();
```

3. Once the database is changed, the expirable cache on all JVMs of that name will be automatically cleared (generally in 5 seconds)

Grouper cache example

A Grouper cache is a genericized wrapper around ehcache.

1. Register the cache as expirable in database.

```
GrouperCache<String, Integer> someCache = new GrouperCache<String, Integer>("edu.internet2.middleware.grouper.cache.GrouperCacheDatabaseTest.myTest",  
    1000, false, 3600, 3600, false);  
someCache.registerDatabaseClearableCache();
```

2. Notify that there are changes

```
// you probably will change something in the cache though this isnt necessary  
someCache.put("test", 5);  
  
// then tell the database to clear all these caches in other JVMs  
someCache.notifyDatabaseOfChanges();
```

3. Once the database is changed, the grouper cache on all JVMs of that name will be automatically cleared

Ehcache example

Ehcachees are used directly or from hibernate

1. Register the cache as expirable in database.

```
Cache someCache = EhcacheController.ehcacheController().getCache("edu.internet2.middleware.grouper.privs.CachingAccessResolver.HasPrivilege");  
GrouperCacheDatabase.ehcacheRegisterDatabaseClearableCache(someCache.getName());
```

2. Notify that there are changes

```
// you probably will change something in the cache though this isnt necessary  
someCache.put(new Element("test2", 6));  
  
// this will tell the database to do an update  
GrouperCacheDatabase.ehcacheNotifyDatabaseOfChanges(someCache.getName());
```

3. Once the database is changed, the ehcache on all JVMs of that name will be automatically cleared

Custom cache

If you want to clear something that is not in one of the above caches, you can register a callback which will be called when another JVM notifies about changes

1. Register the cache as expirable in database.

```
final Map<String, Integer> someCache = new HashMap<String, Integer>();  
  
GrouperCacheDatabase.customRegisterDatabaseClearable(  
    "edu.internet2.middleware.grouper.cache.GrouperCacheDatabaseTest.myTest", new  
    GrouperCacheDatabaseClear() {  
        @Override  
        public void clear() {  
            someCache.clear();  
        }  
    });
```

2. Notify that there are changes

```

someCache.put("test", 5);

// this will tell the database to do an update
GrouperCacheDatabase.customNotifyDatabaseOfChanges(
    "edu.internet2.middleware.grouper.cache.GrouperCacheDatabaseTest.myTest");

```

3. Once the database is changed, the custom cache callback on all JVMs of that name will be automatically called

Note, if you put 4 underscores in your cache name, then you can register to listen for that cache on the prefix of that, or the full name.

Data elements

grouper_cache_overall

Column	Example value	Description
overall_cache	0	There is one row in this table, this row has "0" for this col
nanos_since_1970	1592063170661123456	nanos since 1970 UTC. This is the time the most recent cache was updated

grouper_cache_instance

Column	Example value	Description
cache_name	ehcache__edu.internet2.middleware.grouper.internal.dao.hib3. Hib3AttributeAssignDAO.FindByld expirableCache__myExpirableCache custom__someRandomString	cache name, if there are two underscores, split and the first part is cache, and second part is instance
nanos_sinc e_1970	1592063170661123456	nanos since 1970 UTC. This is the time this cache was last updated

Race conditions

It will not update the nanos_since_1970 to the same value it was before. If there is a conflict, it will wait a few millis and try again.

It will use `System.currentTimeMillis * 1000000 + Math.random()*1000000`. So if something happens in the same milli from different JVMs then there is a one in a million chance the random will be the same. And it is the same, it is ok, it will detect it.

If there is an update less than what the JVM thinks in the latest, it will get all changes in the last 10 seconds from the instance table, so it doesnt miss anything.

This assumes there is NTP on all servers.

Configuration

grouper.properties

```

#####
## Grouper cache database
#####

# If we should run a thread to check the database to see if a cache changed in another JVM
# {valueType: "boolean", defaultValue: "true"}
grouper.cache.database.use = true

# How much time to sleep between checks in seconds
# {valueType: "integer", defaultValue: "5"}
grouper.cache.database.checkIncrementalAfterSeconds = 5

# How much time in between full checks (select all from cache instance table)
# {valueType: "integer", defaultValue: "3600"}
grouper.cache.database.checkFullAfterSeconds = 3600

```

Logging

Set this in `log4j.properties` to see debug logging on this feature

```
log4j.logger.edu.internet2.middleware.grouper.cache.GrouperCacheDatabase = DEBUG
```

Sample logs

```
2020-06-15 00:39:12,263: [Thread-9] DEBUG GrouperCacheDatabase.retrieveFull(279) - - method: retrieveFull,
startup: true, rows: 4, cacheClears: 0, tookMillis: 28
2020-06-15 00:39:22,297: [Thread-9] DEBUG GrouperCacheDatabase.retrieveIncremental(438) - - method:
retrieveIncremental, cacheOverallLastUpdatedNanosFromDb: 1592195958919739392, cacheChange: false, tookMillis: 1
2020-06-15 00:39:32,335: [Thread-9] DEBUG GrouperCacheDatabase.retrieveIncremental(438) - - method:
retrieveIncremental, cacheOverallLastUpdatedNanosFromDb: 1592195958919739392, cacheChange: false, tookMillis: 1
2020-06-15 00:39:42,371: [Thread-9] DEBUG GrouperCacheDatabase.retrieveIncremental(438) - - method:
retrieveIncremental, cacheOverallLastUpdatedNanosFromDb: 1592195978930000000, cacheChange: true, rows: 1,
ehcache__edu.internet2.middleware.grouper.privs.CachingAccessResolver.HasPrivilege: mismatchCleared,
cacheClears: 1, tookMillis: 2
2020-06-15 00:39:52,405: [Thread-9] DEBUG GrouperCacheDatabase.retrieveIncremental(438) - - method:
retrieveIncremental, cacheOverallLastUpdatedNanosFromDb: 1592195978930000000, cacheChange: false, tookMillis: 1
```

Notes

This table can be cleared anytime and will function correctly. It is not exported or imported.