

# THE ART OF HIBERNATE OPTIMIZATION

Bryce Penberthy  
Daugherty Business Solutions



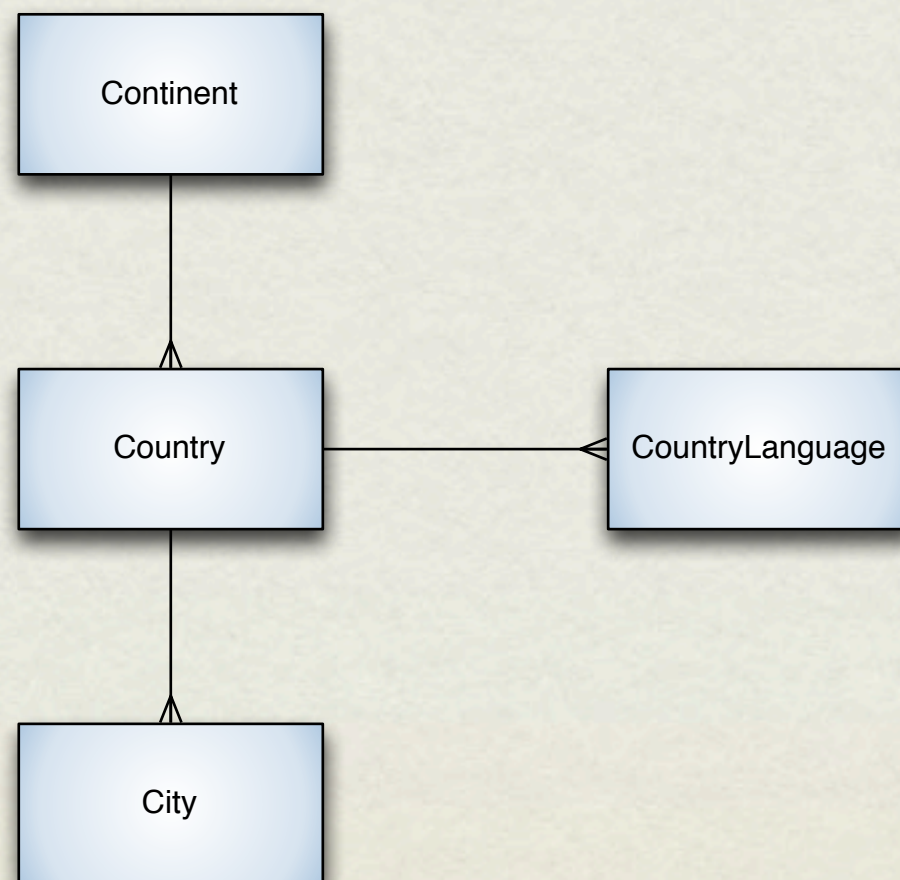
# PREPARATION

- Install / Set up your favorite IDE
- If you want to use Eclipse, install eGit & m2e plugins
- Example code can be found on GitHub: <https://github.com/brycep/world-example>



# EXAMPLE DATABASE

- MySQL world database example / Converted to HSQL  
<http://dev.mysql.com/doc/world-setup/en/world-setup.html>





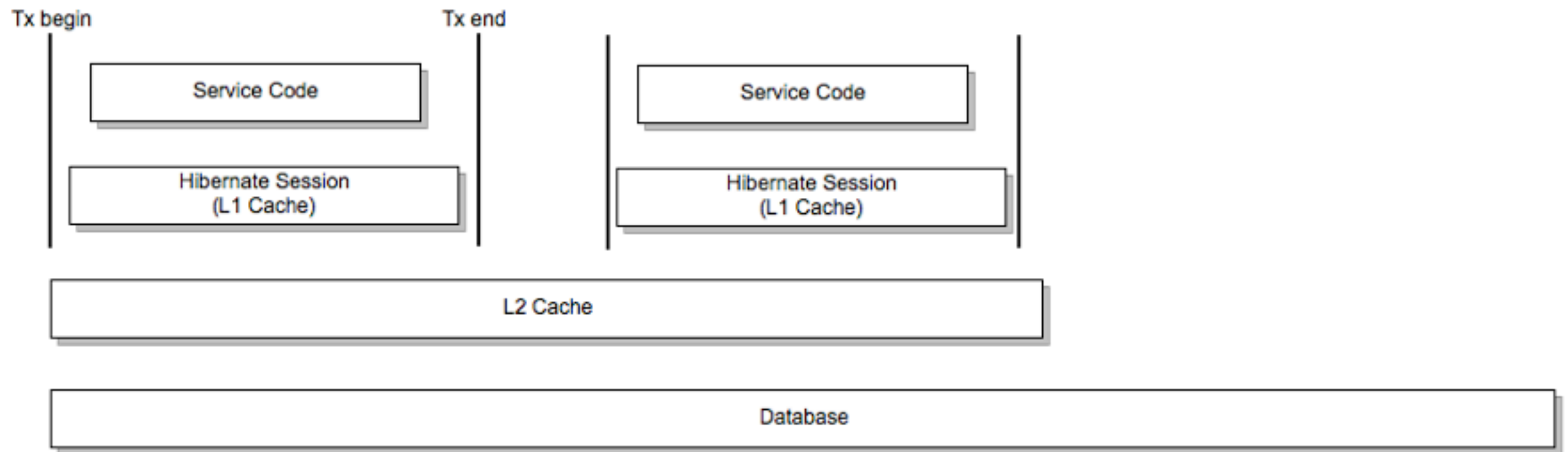
# FIRST STEPS

HOLD ON TO YOUR DATA WITH BOTH HANDS

- Turn on SQL output
- Hibernate statistics are your friend
- Be careful your data doesn't get too stale!



# OBJECT CACHE



- Hibernate Session cache remembers objects in the same transaction
- L2 Cache remembers objects across requests
- Database remembers objects indefinitely



# L2 CACHE IMPLEMENTATIONS

Cache	Type	Cluster Safe	Query Cache Supported
EHCache	Memory, Disk		Yes
OSCache	Memory, Disk		Yes
SwarmCache	Clustered	Yes	
JBoss Cache 1.x	Clustered	Yes (Replication)	Yes
JBoss Cache 2	Clustered	Yes (Replication or invalidation)	Yes

\*Hibernate Documentation Table 19.1

<http://docs.jboss.org/hibernate/core/3.3/reference/en/html/performance.html#performance-cache>



# OPTIMIZING FOR THE CACHE

- Rely on lazy loading for cached objects
- HQL Queries don't utilize the L2 cache
- Use Query Caching carefully



# SETTING UP L2 CACHE

Persistence.xml (or Hibernate config)

```
<property name="hibernate.cache.region.factory_class"
          value="net.sf.ehcache.hibernate.EhCacheRegionFactory" />
<property name="hibernate.cache.use_second_level_cache" value="true" />
<property name="hibernate.generate_statistics" value="true" />
<property name="hibernate.cache.use_query_cache" value="true" />
```



# EHCACHE CONFIGURATION

```
<ehcache>

  <defaultCache
    maxEntriesLocalHeap="10000"
    eternal="false"
    timeToIdleSeconds="120"
    timeToLiveSeconds="120"
    overflowToDisk="true"
    maxEntriesLocalDisk="1000000"
    diskPersistent="false"
    diskExpiryThreadIntervalSeconds="120"
    memoryStoreEvictionPolicy="LRU"
  />

  <cache name="com.servolabs.world.domain.Continent"
    maxEntriesLocalHeap="1000"
    eternal="false"
    timeToIdleSeconds="300"
    timeToLiveSeconds="600"
    overflowToDisk="false"
  />

  ...
</ehcache>
```



# CACHABLE OBJECTS

```
@Cacheable  
@Cache(usage=CacheConcurrencyStrategy.READ_ONLY )  
public class Continent {
```

- Mappings can give your application hints to Hibernate about how to cache domain objects.
- This is especially useful for objects that encapsulate reference data or data that doesn't change much



# EXAMPLE

- CountryDaoImplTest:  
findAllCountriesInContinentLoadsTheCorrectContinent



# USING QUERY CACHE

ehcache.xml

```
<!-- This is an example of a named query cache -->
<cache name="countryQueryCache"
       maxEntriesLocalHeap="100"
       eternal="false"
       timeToLiveSeconds="86400"
       overflowToDisk="true"
/>
```

```
Query query = entityManager
    .createQuery("from Country country where country.region = :regionName")
    .setParameter("regionName", regionName);
// If you're using the Hibernate API directly, use setCacheable(true) and
// setCacheRegion("countryQueryCache") calls
query.setHint("org.hibernate.cacheable", true);
query.setHint("org.hibernate.cacheRegion", "countryQueryCache");
return query.getResultList();
```



- CountryDaoImplTest:  
findCountriesForRegion\_QueryCacheTest



# THAT'S ALL FOR NOW!

- Get my sample code from <https://github.com/brycep/world-example>. Feel free to play around with it, use it for your own apps, presentations or anything you want. Have fun!
- Any Questions?