Caching in Hibernate

Hibernate caching improves the performance of the application by pooling the object in the cache.

There are mainly two types of caching: first level cache and second level cache.

#### First Level Cache

Session object holds the first level cache data. It is enabled by default. The first level cache data will not be available to entire application. An application can use many session object.

#### Second Level Cache

SessionFactory object holds the second level cache data. The data stored in the second level cache will be available to entire application. But we need to enable it explicitely.

Second Level Cache implementations are provided by different vendors such as:

* EH (Easy Hibernate) Cache
* Swarm Cache
* OS Cache
* JBoss Cache

Hibernate Second Level Cache

**Hibernate second level cache** uses *a common cache for all the session object of a session factory*. It is useful if you have multiple session objects from a session factory.

**SessionFactory** holds the second level cache data. It is global for all the session objects and not enabled by default.

Different vendors have provided the implementation of Second Level Cache.

1. EH Cache
2. OS Cache
3. Swarm Cache
4. JBoss Cache

Each implementation provides different cache usage functionality. There are four ways to use second level cache.

1. **read-only:** caching will work for read only operation.
2. **nonstrict-read-write:** caching will work for read and write but one at a time.
3. **read-write:** caching will work for read and write, can be used simultaneously.
4. **transactional:** caching will work for transaction.

The cache-usage property can be applied to class or collection level in hbm.xml file. The example to define cache usage is given below:

1. **<cache** usage="read-only" **/>**