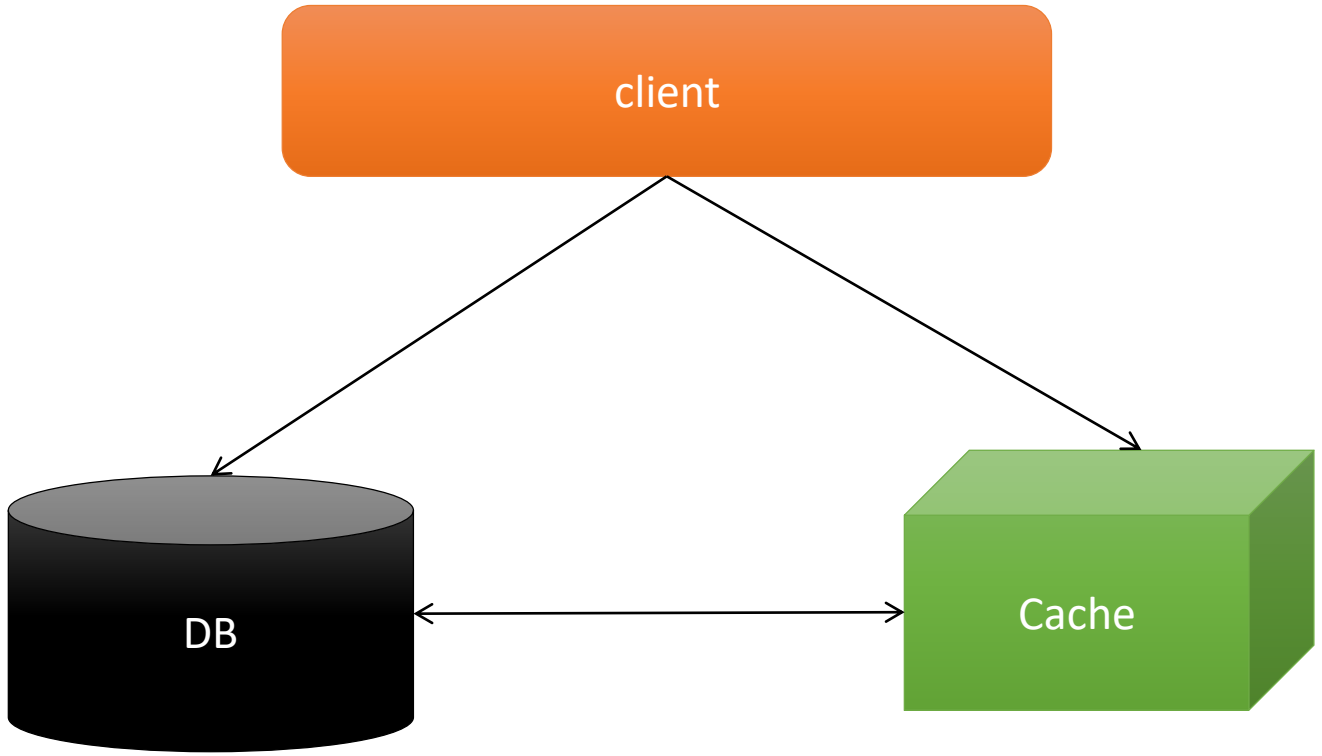
Web性能优化-缓存

凯盛软件

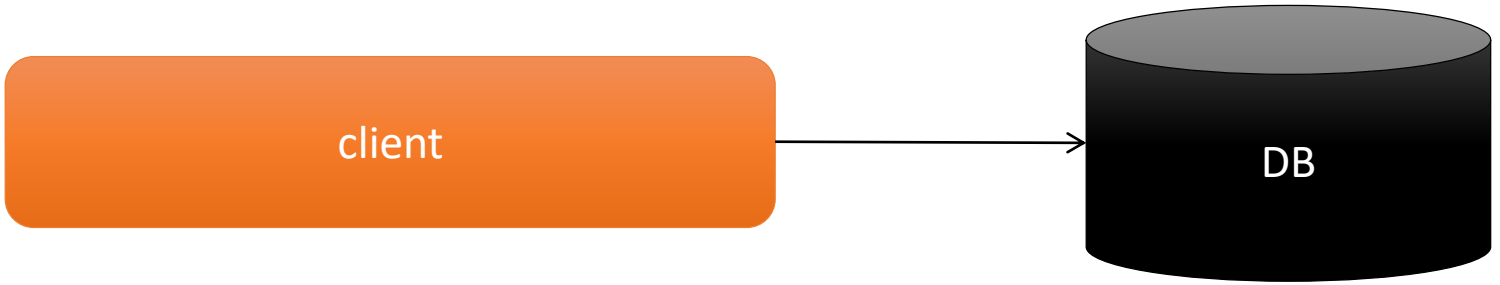
缓存



当前的代码

凯盛软件

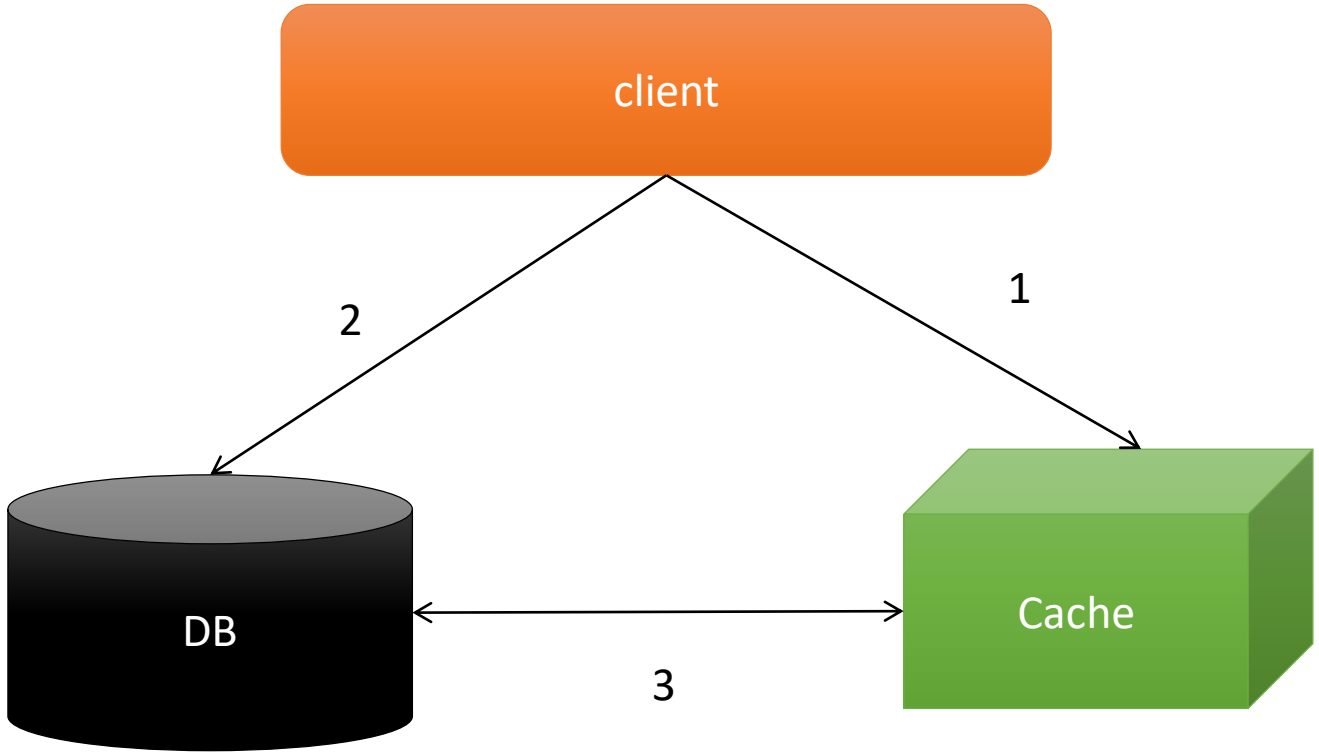
```
public User findById(int id) {
        String sql = "select * from t_user where id = ?";
        return DBHelp.executeQueryForObject(User.class, sql, id);
}
```



加入缓存

凯盛软件

```
public User findById(int id) {
        User user = cache.get(" user" +id);
        if(user == null) {
                String sql = "select * from t_user where id = ?";
                user = DBHelp.executeQueryForObject(User.class, sql, id);
                cache.set(" user" +id,user);
        }
        return user ;
}
```


client
2
1
DB
3
Cache

Ehcache

凯盛软件

http://ehcache.org/

特性：

1. 快速.
2. 简单.
3. 多种缓存策略
4. 缓存数据有两级：内存和磁盘，因此无需担心容量问题
5. 缓存数据会在虚拟机重启的过程中写入磁盘
6. 可以通过RMI、可插入API等方式进行分布式缓存
7. 具有缓存和缓存管理器的侦听接口
8. 支持多缓存管理器实例，以及一个实例的多个缓存区域
9. 提供Hibernate的缓存实现

ehcache.xml

凯盛软件

```
<ehcache>

        <diskStore path="java.io.tmpdir"/>

        <cache name="users"
        maxElementsInMemory="10000"
        eternal="false" //如果为true 导致下面两个时间设置无效
        timeToIdleSeconds="30" //闲置时间 （最后访问时间-当前时间）
        timeToLiveSeconds="30" //存活时间（当前时间-创建时间）
        overflowToDisk="true"
       />
</ehcache>
```

Ehcache的使用

凯盛软件

```
CacheManager cacheManager = new CacheManager();
Ehcache cache = cacheManager.getEhcache("users");
Element element = new Element("name","tom");
cache.put(element);


Element e = cache.get("name");
System.out.println(e.getValue());
cache.remove("name");
```

Guava Cache

凯盛软件

http://ifeve.com/google-guava-cachesexplained/

https://github.com/google/guava/wiki/CachesExplained