



J2Cache 两级缓存框架

@红薯

<http://my.oschina.net/javayou>

内 容

- 关于开源中国的一些数据
- OSChina 的几种缓存策略
- Ehcache 缓存框架
- 实际运行的问题
- 缓存系统选型时的考量
- J2Cache —— 两级缓存框架

开源中国的一些统计数据

- 每天 IP > 50w
- 每天 PV > 300w
- 每天处理动态请求数 2000w
- 高峰期 NginxStatus > 15000
- 系统负载 50%



互动性强！

主服务器：

DELL SC1435 → 12G 内存 / AMD 2G*2 / SATA * 2

DELL R720 → 32G 内存 / Xeon E5520 * 2 / SAS / RAID 1+0

CDN 节点：

4 个自建的 CDN 节点（电信、联通、移动、阿里云）

OSChina 的几种缓存策略

- 对象缓存 `id → object`
- 列表缓存 `List<id>` 而不是 `List<Bean>`
- 页面片段缓存
- 页面缓存 CDN (Squid、Varnish、Traffic Server)

缓存的清除策略

- Ehcache自动清除 `timeToLiveSeconds=3600`
- 程序清除
- 手工清除

Ehcache 缓存框架

Ehcache 是内存中的 Java 缓存框架，特点：

- 内存中，速度快
- 两级缓存（内存 + 磁盘）
- 多区域 (region) 的缓存数据结构
- 缓存数据侦听接口
- 支持多种集群部署方式 (JGroups/RMI/Ehcache Server)

<http://www.oschina.net/p/ehcache>

<https://www.ibm.com/developerworks/cn/java/j-lo-ehcache/>

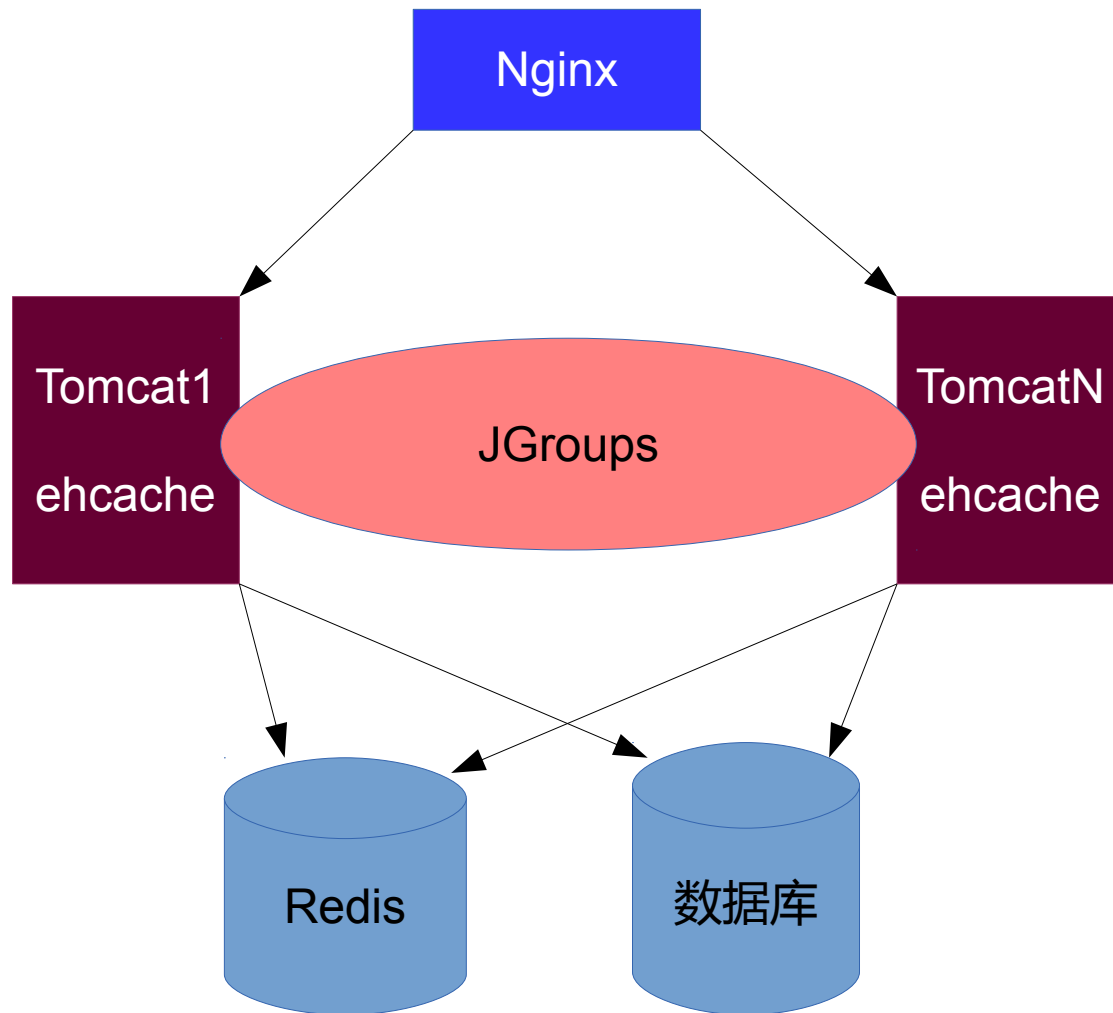
实际运行的问题

- 高峰期重启导致的缓存雪崩
- 单节点对突发的攻击应付不足
- 多节点运行时缓存数据不同步

缓存系统选型

- Ehcache 集群方案的不足
- 集中式缓存的不足 (redis, memcached)
- 是否可以取长补短 ???

J2Cache 的设计思路

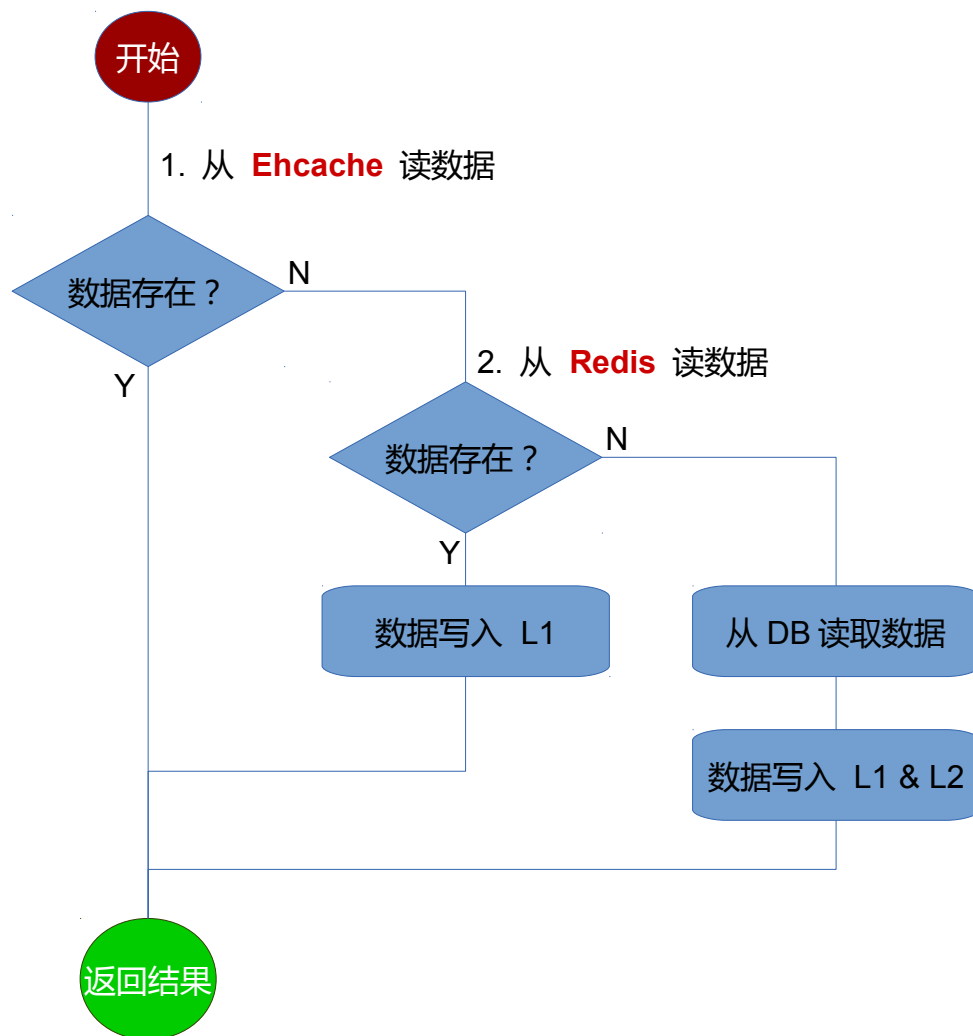


缓存：

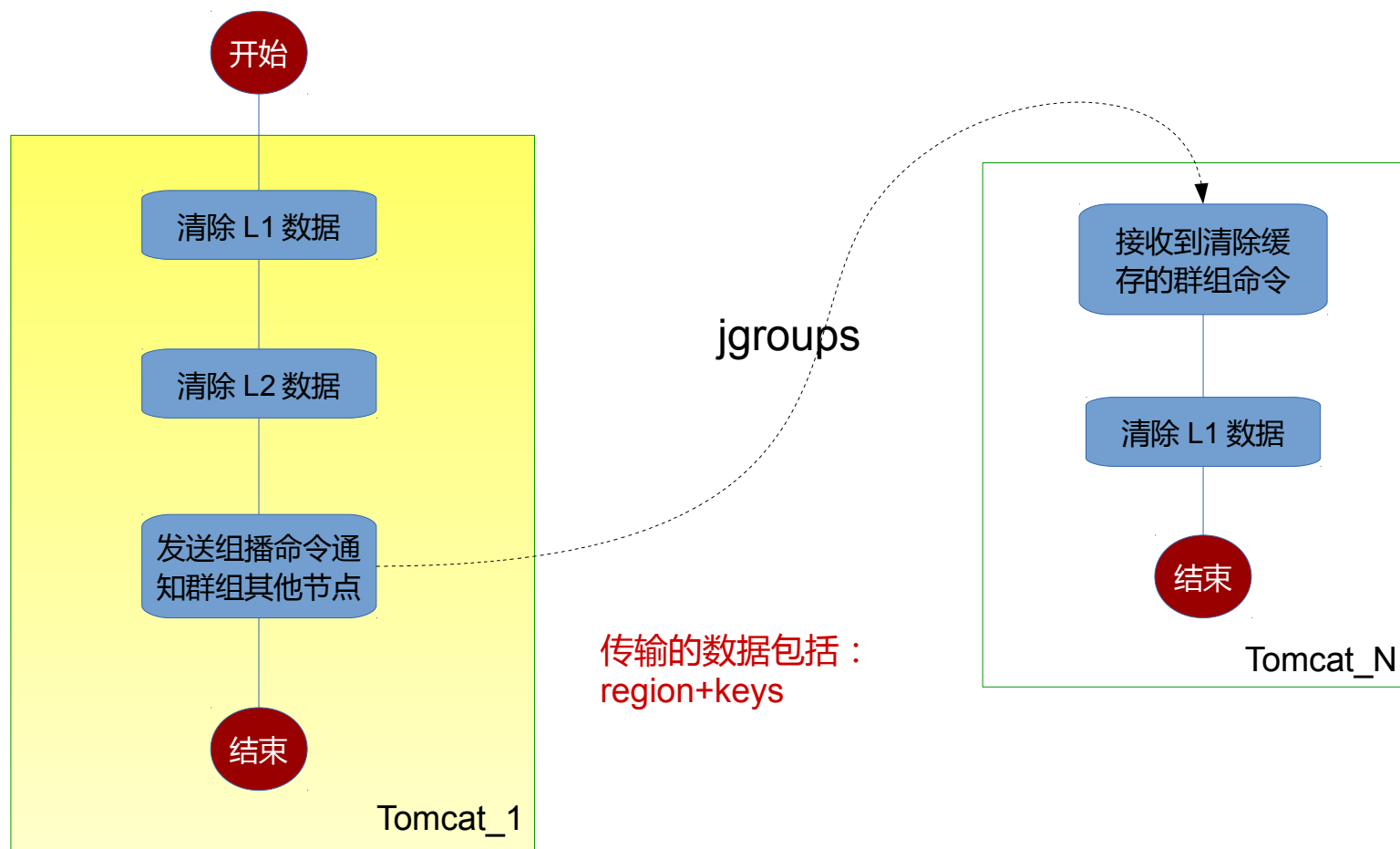
L1：Ehcache

L2：Redis

J2Cache 数据读取流程



J2Cache 数据更新流程



细节 1 : Ehcache 缓存自动失效

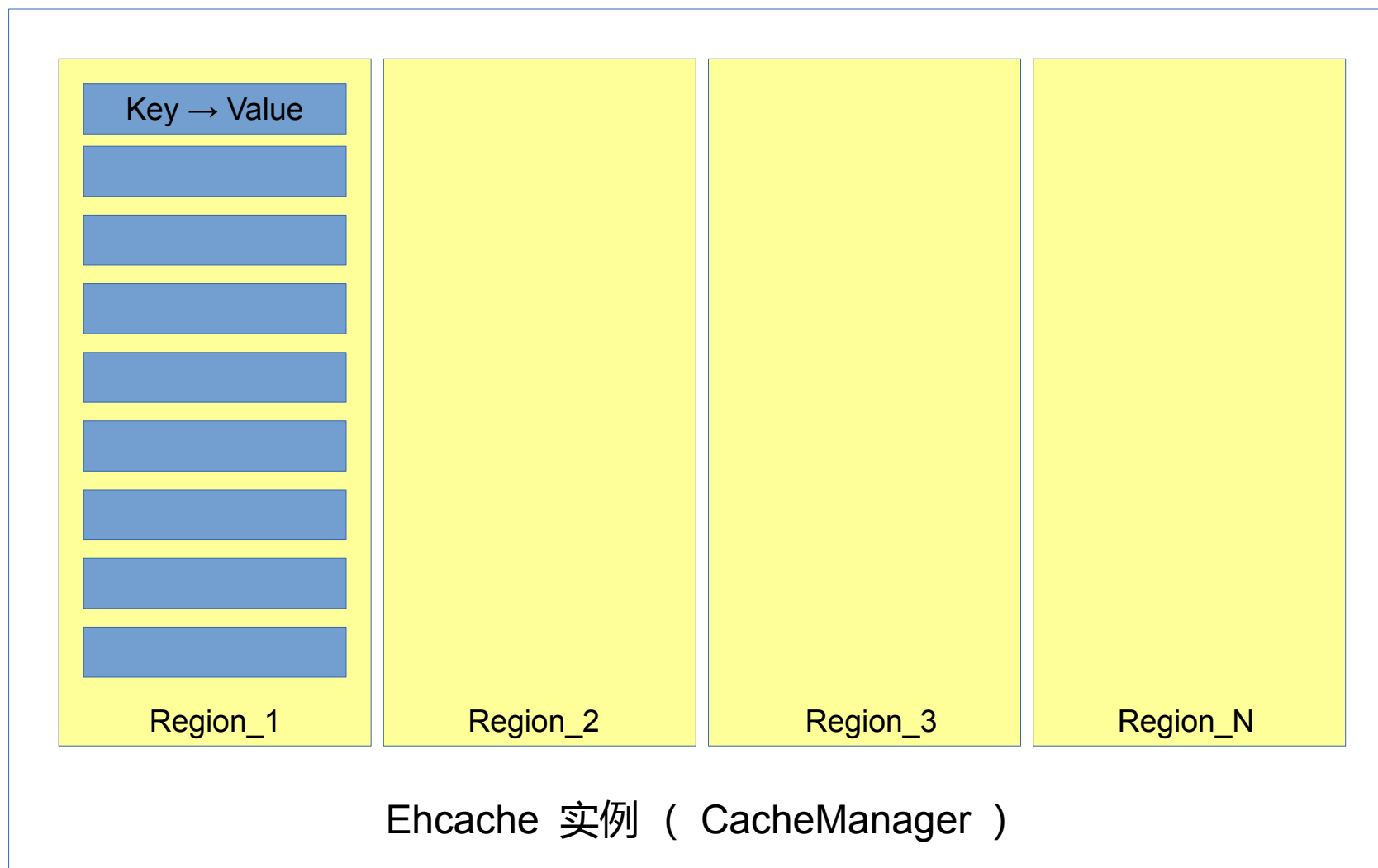
```
public class EhCache implements Cache, CacheEventListener {

    private net.sf.ehcache.Cache cache;
    private CacheExpiredListener listener;

    /**
     * Creates a new pluggable cache based on a cache name.
     *
     * @param cache The underlying EhCache instance to use.
     */
    public EhCache(net.sf.ehcache.Cache cache, CacheExpiredListener listener) {
        this.cache = cache;
        this.cache.getCacheEventNotificationService().registerListener(this);
        this.listener = listener;
    }

    @Override
    public void notifyElementExpired(Ehcache cache, Element elem) {
        if(listener != null){
            listener.notifyElementExpired(cache.getName(), elem.getObjectKey());
        }
    }
}
```

细节 2-1 : 多 region 的处理



细节 2-2 : Redis 处理 region

```
/**
 * 生成缓存的 key
 * @param key
 * @return
 */
@SuppressWarnings("rawtypes")
private String getName(Object key) {

    if(key instanceof Number)
        return region + ":I:" + key;
    else{
        Class keyClass = key.getClass();
        if(String.class.equals(keyClass) || StringBuffer.class.equals(keyClass) ||
StringBulder.class.equals(keyClass))
            return region + ":S:" + key;
    }
    return region + ":O:" + key;
}
```

细节 3：序列化库的选择

	优点	缺点
Java 序列化	兼容性最佳，无侵入性	速度慢，体积大
Kryo	速度快	侵入性强
FST	速度快，无侵入性	不同平台序列化的结果不同（新版本已解决）

J2Cache 配置

#J2Cache configuration

```
#####  
# Level 1&2 provider  
# values:  
# none -> disable this level cache  
# ehcache -> use ehcache as level 1 cache  
# redis -> use redis as level 2 cache  
# [classname] -> use custom provider  
#####
```

```
cache.L1.provider_class=ehcache  
cache.L2.provider_class=redis
```

```
## connection  
redis.host = localhost  
redis.port = 6379  
redis.timeout = 2000  
##redis.password =  
redis.database = 1
```


J2Cache 使用的一些技巧

- Ehcache的磁盘存储建立在内存中 (/dev/shm/xxxx)
- 尽量设置缓存永久性，通过程序进行更新

```
<cache name="pages"  
    maxElementsInMemory="500"  
    eternal="true"  
    overflowToDisk="false"  
/>
```

J2Cache 下一步计划

- 各级缓存的统计 (HITS/MISSE 等)
- 使用 HGET/HSET 替代现有大哈希表的方式
暂时不考虑 field 批量删除性能低下
- 支持其他的缓存框架和服务

哪里获取 J2Cache

发现帮助建议搜索

建议使用HTTPS方式管理您的项目，而不是SSH，这可能导致速度较慢。

红薯 / J2Cache Java 荐

Unwatch 30 Star 63 Fork 54

代码项目网络Pull Requests 1Issues 1Wiki附件 0管理

这是一个 Java 的开发包，可以让应用支持二级缓存框架 ehcache + redis (memcached)。避免完全使用独立缓存系统所带来的网络IO开销问题。—— 查看更多

<http://www.oschina.net/p/j2cache>

下载 zip 挂件 HTTP SSH <http://git.oschina.net/ld/J2Cache.git> 解除推荐 删除

master 文件历史分支 1 标签 0 统计

J2Cache <http://git.oschina.net/ld/J2Cache> 30 commits

增加组成员变化列表显示

 Winter Lau 最后提交于 6天前 [f026ba54642](#)

 .settings	2个月前	 廖凯 ignore和readme修改
 lib	15天前	 Winter Lau 升级 jgroups 到 3.4.3 版本
 src	6天前	 Winter Lau 增加组成员变化列表显示

中国源



1. 使用 Team@OSC 进行团队协作开发
2. 将代码托管到 Git@OSC
3. 使用 Sonar@OSC 进行代码质量分析
4. 将代码部署到 PaaS@OSC 演示平台

TODO : 测试 + 代码评审



Questions