

讨论课二

1、数据缓存

(1) 分布式缓存

分布式缓存这种东西存在的目的就是为了提供比 RDB 更高的 TPS 和扩展性，同时有帮你承担了数据同步的痛苦。在技术和业务都可以接受的情况下，我们可以尽量把读写压力从数据库转移到缓存上，以保护看似强大，其实却很脆弱的关系型数据库。

优秀的分布式缓存系统有以下几种：

1>Memcached

优点：

- Memcached 开源，其访问协议也都是公开的，所以目前有很多第三方的客户端或扩展，在一定程度上对 Memcached 的集群扩展做了支持，但是大部分都只是做了一个简单 Hash 或者一致性 Hash。
- 由于 Memcached 内部通过固定大小的 chunk 链的方式去管理内存数据，分配和回收效率很高，所以其读写性能也非常高；官方给出的数据，64KB 对象的情况下，单机 QPS 可达到 15w 以上
- Memcached 的多语言支持非常好，目前可支持 C/C++、Java、C#、PHP、Python、Perl、Ruby 等常用语言，也有大量的文档和示例代码可供参考，而且其稳定性也经过了长期的检验，应该说比较适合于中小型系统和初学者使用的缓存系统。

不足：

- Memcached 集群的不同机器之间是相互独立的，没有数据方面的通信，所以也不具备 failover 的能力，在发生数据倾斜的时候也无法自动调整。
- 容量有限制

2>Redis

优点：

- Redis 除了像 Memcached 那样支持普通的<k,v>类型的存储外，还支持 List、Set、Map 等集合类型的存储，这种特性有时候在业务开发中会比较方便
- 就性能来讲，在全部命中缓存时，Redis 的性能接近 memcached，但是一旦使用了持久化之后，性能会迅速下降，甚至会相差一个数量级。
- 就性能来讲，在全部命中缓存时，Redis 的性能接近 memcached，但是一旦使用了持久化之后，性能会迅速下降，甚至会相差一个数量级。
- Redis 支持订阅模式，即一个缓存对象发生变化时，所有订阅的客户端都会收到通知，这个特性在分布式缓存系统中是很少见的。

缺点：

- 在扩展方面，Redis 目前还没有成熟的方案，官方只给出了一个单机多实例部署的替代方案，并通过主备同步的模式进行扩容时的数据迁移，但是还是无法做到持续的线性扩容。

(2) 客户端缓存

主要指基于客户端浏览器的缓存方式，由于浏览器本身的安全限制，web 系统能在能在客户端所做的缓存方式非常有限，主要分为以下几种：

- 浏览器 cookie
- 浏览器本地缓存：很多浏览器都提供了本地缓存的接口，但是各个浏览器的实现有差异，例如 Chrome 的 Google Gear，IE 的 userData，火狐的 sessionStorage
- flash 本地存储：相较于 cookie，flash 缓存基本没有数量和体积的限制，而且由于基于 flash 插件，所以也不存在兼容性问题

对客户端存储的合理使用，原本每天几千万甚至上亿的接口调用，一下就可能降到了每天几百万甚至更少，而且即便是用户更换浏览器，或者缓存丢失需要重新访问服务器，由于随机性比较强，请求分散，给服务器的压力也很小！在此基础上，再加上合理的缓存过期时间，就可以在数据准确和性能上做一个很好的折衷。

2、MQ 高可用性和负载均衡方案 — MQ Cluster

主要研究如何应用 WebSphere MQ 的特性来实现企业级应用的业务需求

(1) MQ Cluster 结构与特点

- 使用 Queue Manager Cluster 技术，可以把安装在不同平台（如 AIX，LINUX，WINDOW，z/OS）上的若干个 Queue Manager 设计为一个集群，每个 Queue Manager 都创建成集群中的一员。集群中有一个或多个 Queue Manager 可以定义成拥有整个集群的对象定义信息，称作 Repository queue manager。
- 当用户在集群中创建一个接收通道或队列时，系统会自动在其他队列管理器中创建相应的发送通道和远程队列定义。不论整个集群中有多少 Queue Manager，每个 Queue Manager 只要建立一个接收通道，和一个指向 Repository queue manager 的发送通道就可以完成消息连通

(2) 部署方式

使用 Cluster 技术提高系统的可用性和实现负载均衡，是通过在 Cluster 内的不同 Queue Manager (QMGR)上建立同名的 Queue（同一个 Queue 的多个实例）来实现的。每个 Queue 的实例都能作为消息的目的地，MQ 能够在依照一定的算法决定实际消息应当传给哪个 QMGR。这样当集群里某个 QMGR 失效时，消息会自动路由到其他活动的 QMGR 管理的实例上去。

3、安全性实现

- MQ 提供对 MQ 资源管理的安全性控制管理，确保 MQ 资源只能被合法的人定义和改动
- MQ 提供对 MQ 资源访问的安全性控制，确保本地 MQ 里的信息只能被合法的人使用
- MQ 提供对通道的安全性控制，确保只有合法的人才能够从外面向本地 MQ 传递信息。在 MQ 的通道安全性机制之上，我们可以进一步使用安全套接字层（SSL）协议来对通信链路上的数据进行保护。安全套接字层协议提供业界标准通道安全性，具有确认通信对象、防窃听、防数据篡改的功能。

4、垃圾消息过滤

- 基本的结构化文本过滤器：归纳已被检测到的垃圾信息的常见发文模式，例如信息中链接的比例，是否含有某种常用的符号编码等
- 统计接受者行为：垃圾信息往往发送给大量的接受者，如果该条信息已经被大多数接收者删除，则系统自动停止该消息的转发
- 应用于垃圾邮件检测的贝叶斯单词分布过滤词也可以适用于垃圾消息过滤

参考资料：

1. http://www.ibm.com/developerworks/cn/websphere/library/techarticles/0805_we nhongt/index.html
2. http://www.ibm.com/developerworks/cn/websphere/library/techarticles/0902_we nhongtao2/index.html
3. <https://www.ibm.com/developerworks/cn/linux/other/l-spamf/>
4. <http://data.qq.com/article?id=2879>