

Web Cache 在 IPTV 系统中的应用

张建标, 林 涛

(北京工业大学计算机学院, 北京 100022)

摘 要 电信级 IPTV 系统向用户提供丰富的节目的同时也要考虑提高系统自身的性能。该文描述了 IPTV 系统架构与现有的 Web Cache 技术, 通过对测试数据的分析找到影响 EPG 性能的瓶颈。分析了 IPTV 系统的特点, 提出了一种适合 IPTV 系统的 Web cache 架构 EPG_Cache, EPG_Cache 通过提供缓存部分数据的方法提高 EPG 的响应速度。

关键词: Web cache; IPTV; EPG; 性能

Application of Web Cache in IPTV System

ZHANG Jianbiao, LIN Tao

(College of Computer Science and Technology, Beijing University of Technology, Beijing 100022)

【Abstract】 A national large-scale IPTV system needs to consider how to make its performance optimized when providing user with various multi-media services. This paper describes the architecture of IPTV system and Web cache methods. It finds the bottleneck of EPG performance after analyzing test data, and designs a cache architecture named EPG_Cache to fit IPTV system. EPG_Cache can improve EPG performance by caching part of valuable data.

【Key words】 Web cache; IPTV; EPG; Performance

IPTV 系统是近几年发展起来的一种基于流媒体技术的网络多媒体服务, 使得运营商可以利用宽带 IP 网向用户提供在线观看影视节目的业务, 而 IPTV 用户通过与机顶盒进行交互即可观看到数量众多、画质清晰的电影和电视节目。1999 年英国的 Video Networks 公司首先向用户提供 IPTV 服务, 随后 IPTV 在世界范围内受到电信运营商的重视。随着 Internet 的普及与技术发展, IPTV 业务也在迅速发展, 截至到 2004 年底全球已有超过 50 家电信运营商推出了 IPTV 业务, IPTV 用户超过 100 万, 中国国家广电总局也于 2005 年 5 月向上海文广新闻传媒集团颁发了我国首张 IPTV 牌照。

我国拥有巨大的 IPTV 市场潜力, 有着数量庞大的潜在用户群, 如何尽可能地提高 IPTV 服务性能是我们重点解决的问题之一。提高性能主要可从两方面考虑: 一方面是采用高性能的硬件环境; 另一方面在于设计高效的系统软件架构充分发挥硬件的性能。EPG(Electronic Program Guide)是运营商根据业务需求定义的一组用户与机顶盒(Set Top Box, STB)交互界面的集合, 目前很多 IPTV 工程的 EPG 采用 B/S 架构实现, 所有的界面都是 Web 服务器提供的动态 HTML 页面, 机顶盒充当浏览器的角色。EPG 的响应速度直接影响着用户对系统的满意程度, 提高 Web 服务器的性能是提高 EPG 性能的重要手段, 而采用 Web Cache 又是提高 Web 服务器性能的有效途径。本文描述了在提高 EPG 性能方面所获得的一些经验。

1 Web Cache 概述

针对提高 Web 服务性能的方法有很多种, 如动态负载均衡、Proxy、Pre-Fetching、Cache、AJAX 架构等, 目前 Cache 技术是应用最广泛的解决方案, 其基本思想是利用时间局部性原理, 将客户访问过的内容保存起来, 当这项内容再次被访问时, Cache 提供部分或全部的所需数据, 提高访问效率。

Web Cache 根据放置位置不同有不同的实现方式^[1]:

(1) Browser Caches: 用户本机的浏览器将所访问的页面内容(包括文本和图片)缓存在本地供将来再次访问时使用;

(2) Caching Proxies: 设置于学校、公司等机构的网络出口处, 可以同时服务多个用户。

(3) Surrogates: 在 RFC 3040 中定义的一种近服务器的 Cache 结构, 它处于 Web 服务器的外层, 为 Web 服务器分担静态 HTML 资源的服务任务。

前两种结构只能服务于部分用户, 无法有效地提高服务器的整体性能, 于是许多门户网站采用 Surrogates 来提高网站的整体性能。IPTV 不同于传统的网络服务, 传统的网络服务可以没有用户的概念, 任何用户访问相同的 URL 地址会看到相同的结果; IPTV 是一种有用户概念的服务, 不同的用户进入相同的 EPG 界面后可能会看到不同的效果, IPTV 工程中采用 Surrogates 可以在缓存静态资源方面起到一定作用, 但这远远不够。

EPG 的特点是界面元素简洁、清晰, 流程结构不会经常变动, 但是界面所展示的信息会根据不同时间、不同用户和不同操作而变化, 例如: 电视节目单(图 1)左侧为 8 个电视频道的列表, 它们在一个时间段内的节目列表对应显示在中间主体部分。当前用户没有购买 05 频道的观看权, 在他所看到的电视节目单中“环法自行车赛”节目呈灰色显示, 03 频道的“辛普森一家 13”节目因为未到播出时间也呈灰色显示。虽然不同的用户在不同时间看到的这个界面可能不相同, 但 EPG 在构造这个页面时从数据库中读到的电视频道、节目信息的数据完全相同。EPG 原型系统在构造电视节目单这

作者简介: 张建标(1969 -), 男, 博士、副教授, 主研方向: 网络技术及应用; 林 涛, 硕士生

收稿日期: 2006-05-25 **E-mail**: lintao@emails.bjut.edu.cn

个页面时会向数据库发起 9 次查询，尽管访问数据库是通过进程间通信的方式，但频繁的数据库操作也会影响 EPG 的性能，本文将从此处着手进行分析，找到适合 IPTV 工程的 EPG 性能提升方案。



图 1 EPG 页面示例

2 EPG 性能瓶颈分析

在工程初期首先实现了 EPG 的原型系统，EPG、Web 服务器与 STB 的系统架构简略如图 2 所示。

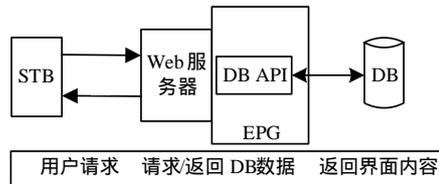


图 2 原型系统中 EPG 架构

系统主要由 STB、EPG(Web 服务器)和 DB 3 部分组成，其中 STB 负责接收遥控器的信息获取用户的操作信息，Web 服务器通过应用程序接口调用 EPG 模块生成 EPG 界面，DB 存储 EPG 使用到的数据。

一般来说在用户请求内容的长度相差不大的情况下，用户请求静态文件时服务器响应时间最短，用户请求的内容由应用程序动态生成时服务器响应时间会有所增长^[2]，频繁的外部 IO 操作也会影响服务器的性能。测试中逐步引入应用程序接口调用、EPG 代码、数据库访问等因素，测试出对性能影响较大的因素。

Web 服务器性能由平均每秒钟处理请求的数量、请求平均响应时间、请求最长响应时间等数据体现出来。请求平均响应时间是 Web 服务器性能的一个重要指标，也是性能优劣的最终表现，下面的测试结果分析中用请求平均响应时间来衡量 Web 服务器性能。

按照上面的分析以电视节目单为测试界面构造 4 项测试(测试工具为 Web 服务器自带的测试程序)：

测试 A 将测试界面保存为静态 HTML 文件，测试 Web 服务器处理静态资源时的性能。

测试 B 测试仅包含一条简单语句的 Web 应用程序，测试 Web 服务器与 Web 应用程序之间接口的性能。

测试 C 将测试界面中进行数据库操作的部分替换为预定义的测试数据，测试未引入数据库操作时 Web 应用程序生成测试界面的性能。

测试 D 测试原测试界面，测试引入数据操作后原型系统的性能。

测试结果如表 1 所示。

表 1 并发请求=1 测试时长=60s 时的测试结果

	页面大小(B)	每秒处理请求数	请求平均响应时间(ms)	相对于原型系统测试 D 平均响应时间下降率
测试 A	24 972	1 220	0.83	98.19%
测试 B	581	532	1.87	95.93%
测试 C	24 838	294	3.4	92.59%
测试 D	24 838	22	45.9	0

以测试 D 的测试结果为参照数据，逐步去掉数据库访问、EPG 代码、应用程序接口调用等因素后请求平均响应时间均有下降，下降率根据式(1)进行计算。表 2 为去掉应用程序接口调用、EPG 代码、数据库访问等因素后平均响应时间下降率。

表 2 平均响应时间下降率

	响应时间下降率	备注
数据库访问	92.6%	测试 C 相对于测试 D
EPG 代码	45.0%	测试 B 相对于测试 C
应用程序接口调用	55.6%	测试 A 相对于测试 B

设：M=请求平均响应时间

$$\text{测试 X 相对于测试 Y 平均响应时间下降率} = 1 - (M_x / M_y) \quad (1)$$

可以看出，Web 程序接口调用与 EPG 代码都消耗了较多的服务器性能，相比之下访问数据库的操作对服务器性能有更大的影响，这是因为用户打开相同的界面时 EPG 要反复执行相同的数据库查询而造成性能损失，为此，我们设计了一种 Cache 处在 Web 服务器之后的 Cache 架构——EPG_Cache。

3 EPG 架构

3.1 改进后的 EPG 架构

图 3 是增加 EPG Cache 后的结构，EPG 增加了与 EPG_Cache 的两个接口。EPG 内部获取数据的流程改变为：首先 EPG 根据当前状态与用户操作信息向 EPG_Cache 查询所需数据，如果 EPG_Cache 返回有效的缓存数据，EPG 绕过数据库查询操作直接构造页面，否则 EPG 查询数据库，然后将所得的数据保存到 EPG_Cache 中，再构造页面。

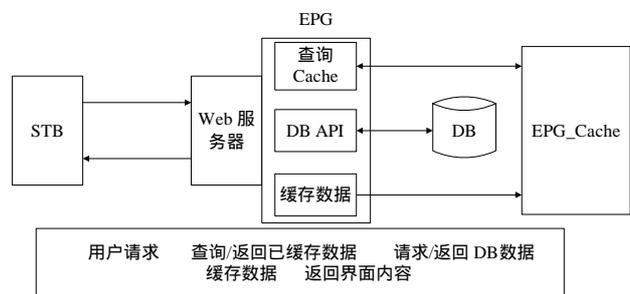


图 3 改进后的 EPG 架构

3.2 EPG_Cache 的结构

EPG_Cache 模块是一个用 C 语言写的系统后台进程，工作流程如图 4 所示。EPG_Cache 主要实现了 3 项功能：

- (1) 接收到数据库同步模块更新数据库的消息后使全部已缓存数据失效；
- (2) 提供给管理程序的查询 Cache 使用情况的接口；
- (3) 提供给 EPG 模块的缓存数据、提取数据的服务。

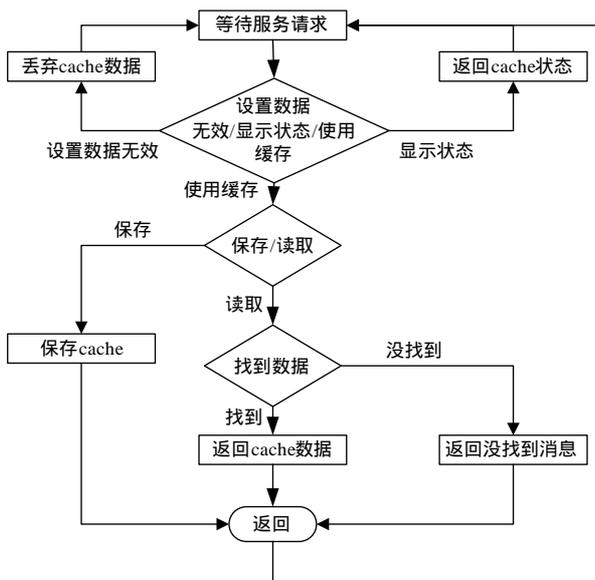


图4 EPG_Cache 流程

EPG_Cache 采用 Hash 表来提高匹配数据的速度。EPG 把界面接收到的关键参数组合生成关键字串，再将对关键字串进行 CRC32 运算所得的 32 位 CRC32 值与界面唯一编号、函数编号共同组成 Hash 键值：

$$hash_key = CRC32(key_string) + page_id + cache_func_id \quad (2)$$

EPG_Cache 使用一段连续的内存空间存放被缓存的数据，避免了频繁分配、释放内存块而产生系统内存碎片，这对需要连续运行很长时间的 EPG_Cache 来说很重要，分配的内存空间大小由程序启动时的参数指定。EPG_Cache 设置两条双向链表管理内存空间：一条存放已分配空间的缓存数据 (Used Block List, UBL)；另一条链表存放未被使用的内存空间块 (Free Block List, FBL)，FBL 链表节点按块大小递增有序。当新增加一条需要缓存的数据时 EPG_Cache 首先从 FBL 中搜索大小合适的内存块分配给新数据，如果 FBL 没有可用空间，则按照 LRU 算法丢弃 UBL 链表中的部分数据直到有足够的空间为止。同时 EPG_Cache 也会定时整理 UBL 与 FBL 两个链表，尽量将空闲空间集中到一个连续区域中。

4 测试结果

在加载 EPG_Cache 模块后重新测试 Web 服务器性能，表 3 和表 4 分别是并发用户为 1 和 100 时的 Web 服务性能数据，测试时长 60s，Cache 命中率由 EPG_Cache 模块进行模拟，其中响应时间提升率均为相对于未加载 EPG_Cache 模块时的性能，命中率由 EPG_Cache 模块使用随机数模拟，CPU 使用情况的状态记录由 Linux 系统的 vmstat 程序提供。

表 3 并发请求为 1 时 Web 性能测试结果

命中率	处理单个请求平均时间(ms)	请求失败次数	每秒处理请求次数	请求最长时间	相对无 cache 时响应时间下降率	用户态程序 CPU 使用率	内核程序 CPU 使用率
99%	5.9	0	169.0	57	87.2%	45%	30%
75%	16.6	0	60.1	100	63.8%	77%	15%
50%	27.6	0	36.3	101	39.9%	82%	13%
25%	39.9	0	25.1	87	13.1%	86%	12%
0%	51.5	0	19.4	205	-12.2%	86%	12%
无 cache	45.9	0	21.8	96	N/A	90%	8%

表 4 并发请求为 100 时 web 性能测试结果

命中率	处理单个请求平均时间(ms)	请求失败次数	每秒处理请求次数	请求最长时间	相对无 cache 时响应时间下降率	用户态程序 CPU 使用率	内核程序 CPU 使用率
99%	6.5	0	153.5	2326	85.3%	39%	52%
75%	17.7	0	56.4	6635	60.0%	74%	26%
50%	29.1	0	34.3	9212	34.3%	79%	19%
25%	43.0	0	23.3	12409	2.9%	77%	23%
0%	48.6	148	20.6	14724	-9.7%	81%	17%
无 cache	44.3	165	22.6	17935	N/A	84%	16%

测试结果显示 EPG_Cache 在命中率不低于 25% 的情况下能不同程度地提高 EPG 模块的性能，原型系统在并发用户过多时会出现无法响应请求的情况，在采用 EPG_Cache 后有效降低了这种情况的发生，同时，CPU 使用时间也有所降低。

由于向 EPG_Cache 中缓存数据的动作是由 EPG 模块主动发起的，实现 EPG 模块功能时我们可以针对具体的界面进行分析是否需要缓存这个界面中的数据。如果 EPG 在一段较长时间内为构造某个界面所做的数据库查询完全相同，我们称这个界面是稳定的，反之称这个界面是不稳定的。在构造稳定性较高的界面时，使用 EPG_Cache 能够在很大程度上保证 Cache 的高命中率，从而使 Web 服务器获得较高的性能提升。

5 结论

本文讨论了在实施 IPTV 工程过程中分析性能瓶颈的方案，针对 IPTV 系统的特点设计出一种适合 IPTV 系统的 Cache 结构，通过在模拟现场系统中的充分测试和性能分析，证明改进后的 EPG 系统的性能有了较大的提升。目前 EPG_Cache 在实际工程中仅被用来缓存从数据库中获取的数据，但这种由 EPG 主动缓存数据的模式十分灵活，EPG 在需要时可以缓存任何有价值、值得缓存的数据，甚至整个动态页面。EPG_Cache 仍然存在进一步优化的空间，接下来的工作主要从两个方面继续研究：(1)尽量减少 Cache 使用的内存数量可以提高 Web 服务器的性能，需要研究一种合适的内存使用量计算方法；(2)各 EPG 界面被访问的频度不同，各界面的稳定度也不同，接下来需要总结 IPTV 界面数据元素特点后设计适合的缓存替换算法。

参考文献

- 1 Wessels D. Web Caching[M]. 1st ed. O'Reilly, 2001-06.
- 2 Iyengar A, Challenger J. Improving Web Server Performance by Caching Dynamic Data[C]//Proc. of USENIX Symposium on Internet Technologies and Systems. 1997.
- 3 Iyengar A, MacNair E, Nguyen T. An Analysis of Web Server Performance[C]//Proceedings of the IEEE Global Telecommunications Conference, Phoenix, AZ, USA. 1997: 1943-1947.
- 4 Williams S, Abrams M, Standbridge C R, et al. Removal Policies in Network Caches for World Wide Web Documents[C]//Proceedings of the ACM Sigcomm'96. 1996-08.
- 5 Cardenas L G, Gil J A. Performance Comparison of a Web Cache Simulation Framework[C]//Proc. of the 19th International Conference on Advanced Information Networking and Applications. 2005: 28.