

一种在线的动态网页分块缓存方法

尤 朝, 周明辉, 林 泊, 曹东刚, 梅 宏

(北京大学信息科学技术学院 软件研究所, 北京 100871)

(北京大学高可信软件技术教育部重点实验室, 北京 100871)

摘 要: 分块缓存技术能够有效提高动态网页的服务质量. 现有的既存系统较少使用分块缓存技术设计, 如何将其应用于这些系统是一个很大的挑战. 本文提出了一种在线的动态网页分块缓存方法, 使原系统演化成基于分块的系统, 为用户服务. 该方法具有三方面优点: (1) 使原系统在线演化, 不影响系统对用户的服务提供; (2) 简化了模板的维护, 使逻辑执行的粒度从页面降低到分块, 减轻了服务器端的压力; (3) 独立于原系统, 有效支持系统的变化和升级. 文章最后对方法进行了实现和评估, 结果说明该方法能够较好实现系统的演化, 提高系统的服务质量.

关键词: 动态网页; 分块缓存; 在线; 标记

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2009) 05-1087-05

An Online Approach for Fragment Based Caching of Dynamic Web Pages

YOU Chao, ZHOU Ming-hui, LIN Bo, CAO Dong-gang, MEI Hong

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

Abstract: Fragment based caching technology can effectively improve the quality of dynamic web pages' services. Now a days, many existing dynamic web pages systems have not been designed by this technology, so how to apply it to these systems is a big challenge. This paper presents an online approach to apply fragment based caching technology to these existing systems. Our approach has three main advantages as follow. Firstly, the original system is evolved into a fragment based system online without service interception. Secondly, the approach simplifies templates maintenance, decreases the logic computing granularity from page to fragment, and greatly reduced the pressure on the server side. Thirdly, the approach is independent of the original system, which effectively supports the change and upgrade of the system. Evaluation results show that this approach can fulfill the system evolution and greatly improve its service quality.

Key words: dynamic web pages; fragment based caching; online; flag

1 引言

随着 Internet 上信息容量的快速增长和用户良好体验的迫切需求, 动态网页的发展已呈快速上升趋势.

然而这种在运行时刻根据用户和网站的状态动态生成网页的方式, 使得服务器端对每一个请求都需要执行计算, 不仅延缓了用户的响应时间, 还给服务器端带来很大的压力.



图1 模板与分块的关系图

收稿日期: 2008-07-17; 修回日期: 2008-11-17

基金项目: 国家 973 重点基础研究规划 (No. 2002CB312003); 国家自然科学基金 (No. 60603038, 60503029); 国家 863 高技术研究发展计划 (No. 2007AA01Z133, No. 2006AA01Z156)

针对上述问题,近年来有很多的工作致力于提高动态网页服务的质量,其中分块缓存技术在动态网页的生成、缓存和交付等方面已经得到了广泛的关注^[1~3].如图1所示,网页中与其它部分在主题、功能和特征上有明显区别的部分称为分块(F1~F5),Web页面维护这些分块的引用,通过组合这些分块展现网页的内容.分块缓存技术通过降低缓存的粒度,不仅提高了数据的命中率,减少了缓存的磁盘使用空间,而且还改善了用户体验,减轻了网络和服务器端的压力.

然而,目前许多既存系统并没有采用分块缓存方法设计.为了将分块缓存技术应用于这些系统,就需要为该系统进行重新设计,定义系统的模板和分块,这极大的增加了开发成本,可行性很低.

本文提出了一种在线的动态网页分块缓存方法.该方法对系统动态生成的HTML页面进行分析,然后根据分析结果对源页面文件进行标记,在线实现系统的逐步分块和平滑迁移.该方法不影响系统对用户的服务提供;使逻辑执行的粒度从页面降低到分块,从而减轻了服务器端的压力;同时还独立于原系统,有效支持了系统的变化和升级.

2 方法和框架

本章描述了该方法的基本思想;围绕该方法建立了一个框架,描述了其结构和功能;阐述了框架中不同方面的策略,并对选择策略定制框架提供了指导意见;同时描述了该方法中系统在线渐进演化的特性.

2.1 基本思想

如图2所示,分块管理系统独立于未分块的原系统,通过对原系统的监控、分析、计划和执行等步骤^[4],把未分块的系统转成分块的系统.分块管理系统在原系统运行时,截取系统返回给用户的HTML页面,并对这些HTML页面进行分块分析,然后根据分块的结果在系统源文件中进行标记.当被标记的页面再次被请求时,系统会识别被标记的缓存分块,从缓存中读取.

为了对方法进行详细的描述,抽取了方法中的关键元素,构成一个可扩展的框架.用户可以灵活选择和定制策略,调整和配置框架,生成一个更贴近系统状态特征和访问需求的框架具体实现,最大化系统的性能和用户的体验.

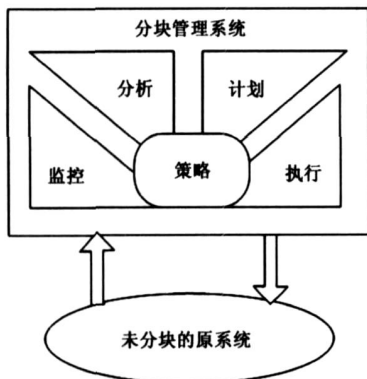


图2 基本结构图

2.2 框架结构

该框架描述了方法中的关键元素及它们之间的联系.框架分为四个部分,如图3所示,包括:

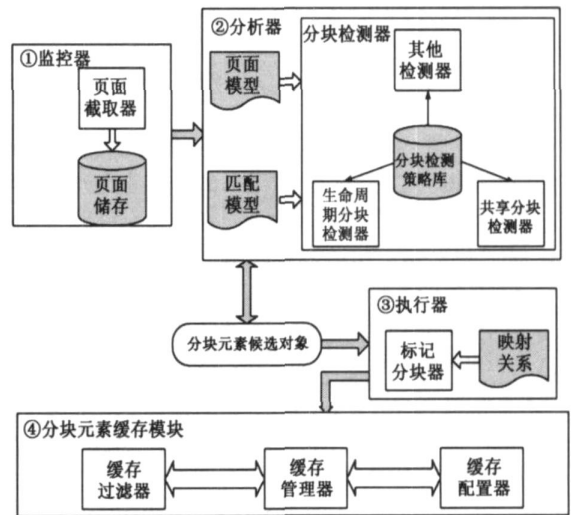


图3 框架结构图

(1) 监控器

此模块在运行时截取服务器动态生成给用户的动态内容(HTML页面),同时将此页面传递给用户而不影响用户的服务.然后将截取的内容存储,提供给分析器来分析页面中的分块.

(2) 分析器

页面模型用于对存储的网页进行建模,高效的遍历和比较网页的内容,简化分析过程的复杂度和节约分析需要的时间.匹配模型用来对不同的网页进行比较.

分块检测器通过使用匹配模型对页面比较,进行分块.根据策略中设定的不同检测器类型,实例化不同的检测器(如共享分块检测器和生命周期分块检测器等),文献[5]中详细描述了这两种检测的算法.同时用户可以对分块检测策略库增加和修改策略.

通过选择的页面模型,匹配模型和选择器生成分块元素对象.同时在运行过程中根据系统的特征分析、经验和测试数据,调整该检测器中的参数,提高检测器的准确性和性能.通过设定各种性能指标KPI(Key Performance Indicator),对比不同的策略给系统带来的影响,最优化系统的配置.

(3) 执行器

执行器根据分析器模块生成的缓存分块,依照页面源码和动态内容(HTML页面)之间的映射关系对动态页面源码(如JSP)进行标记,并重新编译.当下次该页面的请求到达时,在生成动态页面的过程中,如果遇到特定的标记,就根据该分块的关键字从缓存中取出,组合生成页面响应用户.此过程具有在线渐进演化的

特性(见 2.4 节)。

(4) 分块元素缓存模块用于缓存检测得到的分块。

2.3 策略定制

在整个框架中对于每一个模块及其子模块进行配置, 以形成整个基于分块的动态网页系统, 我们称之为策略。主要有以下几个方面:

(1) 在分块检测器中, 通过分析网页的特性和用户的需求, 配置形成不同的分块检测器, 例如由于动态网页中很多页面之间都有共享部分, 通过使用共享分块检测器, 检测出页面之间的共享分块, 从而这些页面共同拥有这一分块的引用, 缓存中也只需缓存这一个分块, 减少了缓存的磁盘使用空间以及缓存多个副本的维护代价。在同一个动态页面内, 不同部分的更新变化频率并不一致, 通过使用生命周期分块检测器, 检测出不同变化频率的分块, 对于变化频率较高的分块不予以缓存, 对变化频率较低的延长该分块的缓存失效时间, 从而减少缓存中数据失效的数量和频率。

(2) 每种分块检测器使用的匹配模型和页面模型都可以根据需求定制, 比如页面模型可以使用 DOM Tree^[6]或 AF Tree^[5]等。匹配模型可以使用文本匹配, Shingles 编码, MD5 编码或者 HiSh 编码^[5]等。

(3) 每种分块检测器会根据用户对分块的准确度和性能需求调整配置因子。例如在共享分块检测器中, 共享因子 SF 表示一个分块至少需要被 SF 个网页共享时, 才能被认为是共享分块; 最小匹配因子 MM 表示两个分块的匹配值大于 MM 时, 便可以认为是同一分块, 如低于 MM 则不是同一分块; 最小分块大小 MS 表示小于 MS 的分块将不予以考虑。在生命周期分块检测中, 子节点变化阈值 CCT 表示, 当一个节点的子节点中值发生变化的比例大于 CCT 是就认为该节点也发生了变化; 页面存储时间差 TS 表示缓存不同版本页面的时间间距, 可以设定 TS 来调整检测到的生命周期分块的失效时间等。

2.4 在线渐进演化

系统的演化是通过多次的标记来实现的。如图 4 所示, 原来没有分块的系统处于状态 1, 当执行器标记了

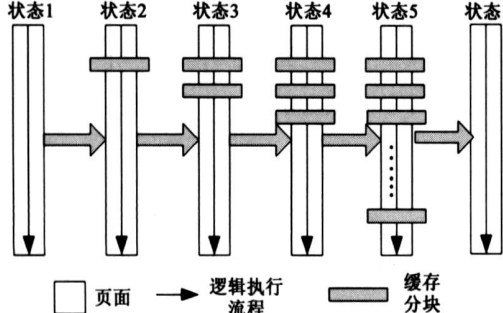


图4 系统状态变迁图

第一个分块时, 页面变成状态 2。在此状态转化过程中并没有间断向用户提供服务。当系统在状态 1 时, 客户的访问使用的是没有分块的页面, 而在状态 2, 客户访问使用的是分块的页面。这样不仅使用户的体验得到了改善, 而且此过程对用户透明。当该页面的设计发生变化的时候, 页面转化成状态 1, 分析器会重新检测该页面中的分块。每次对页面进行标记之后, 系统会重新编译该页面, 从而保证了系统的在线平滑迁移。

3 方法评估

3.1 原型实现

根据该方法, 我们实现了一个原型, 结构如图 5 所示: 监控器模块采用开源 Web 服务器 Tomcat^[7]的 Valve 截取器机制实现, 在 Servlet 响应客户期间截取页面, 并存储到数据库中; 页面模型使用 AF Tree; 匹配模型使用 HiSh 编码; 分块检测使用共享分块检测器和生命周期分块检测器; 缓存模块使用 OSCache^[8]; 标记模块, 使用 OSCache 提供的 JSP 标记库, 在动态页面源文件中标记需要缓存的分块对象。

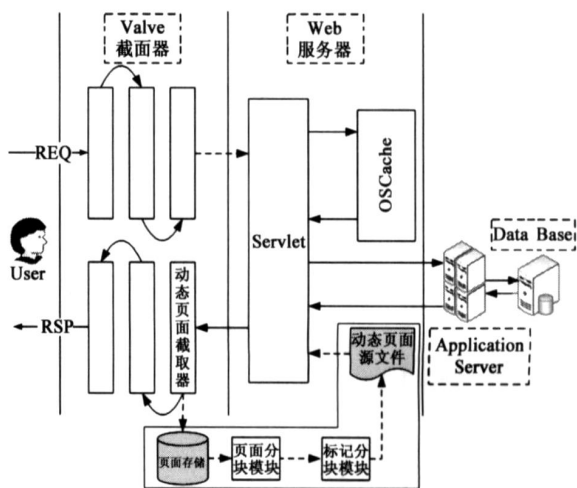


图5 框架的一种实现方式流程

3.2 实验方法

为了对该方法进行评估, 本文使用一个基于 JSP 的动态网站系统做了测试。

实验环境

如图 6 所示, 一台客户机运行负载生成器, 采用 LoadRunner 8.0^[9]; 一台机器运行 Web 服务器, 采用 Tomcat 6.0; 一台机器运行数据库服务器, 采用 MySQL 5.2; 所有机器均运行在 Windows XP 操作系统上, 通过 100Mbps 以太网互联。

客户负载

客户端采用负载生成器, 模拟产生会话类型的负载, 会话过程中发出一系列的 Web 请求, 两个请求之间会等待一定的思考时间, 负载的轻重可以通过改变会

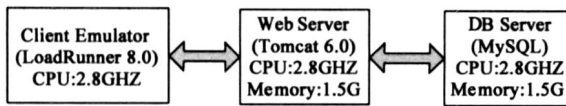


图6 实验环境

话数量来调整。本文模拟了真实世界中两种客户负载情况: (1) RAMP 模式, 300 用户负载同时启动, 维持一段时间, 同时停止; (2) STEP 模式, 用户的负载数量以 50 为阶梯逐步递增, 到达最大值 300 后, 维持一段时间, 然后以 50 为阶梯逐步递减。

3.3 实验结果

使用该方法对系统性能的提高 在 RAMP 和 STEP 两种用户负载模式下, 如图 7 所示, 使用该方法对系统进行分块检测之后, 演化成的新系统的平均响应时间低于原系统的平均响应时间。在 RAMP 模式下系统的平均响应时间从 10.09s 下降到 8.88s, 在 STEP 模式下系统的平均响应时间从 9.13s 下降到 7.04s。

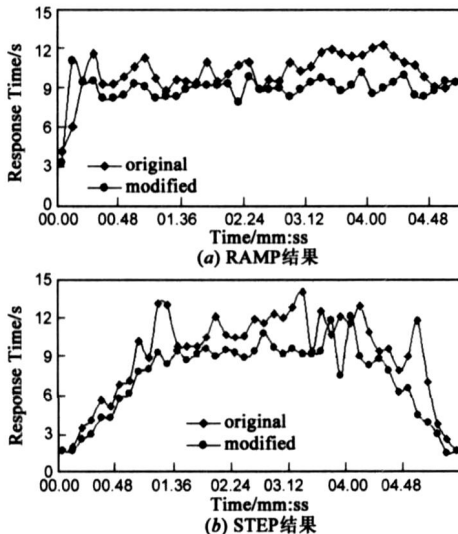


图7 不同用户负载模式下原系统和使用工具分块处理的新系统的响应时间比较

在线渐进演化 为了测试系统在线渐进演化特性, 我们在特定的时间对系统进行检测和标记。图 8 是原系统向分块系统状态迁移的过程。在 02:00 (mm:ss) 时刻, 分块管理系统对原系统进行一次分块处理和源文件修改, 在 02:20 时刻左右, 响应时间有一次跳跃, 是由于修改代码重新编译和替换引起的, 实现了状态的平滑迁移, 没有中断系统提供服务。04:00 时刻在原有基础上又进行一次分块处理, 情况与 02:00 时刻类似。

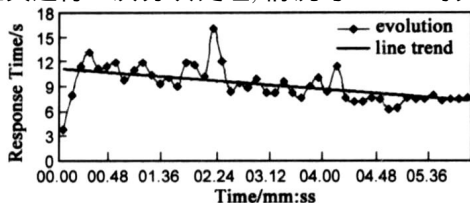


图8 系统状态变迁及趋势图

从线性趋势图来看, 系统的平均响应时间呈下降趋势。

4 相关研究

文献[10]提出一种方法, 对生成动态网页的源代码进行静态分析, 构建一个描述属性和不同脚本之间依赖的语义树, 从而对网页进行分块。一方面这种静态分析的方式从技术上存在一定的困难; 另一方面它对系统的演化不是逐步进行的, 不支持服务不间断的提供。本文对动态生成的 HTML 页面进行分析, 弥补了对静态源码分析困难的不足, 同时本文的方法支持系统的渐进演化。

文献[2]采用基于关键字的对象抽取技术, 生成模板及对应的分块。它对 HTML 页面进行分析, 得到该页面中的对象和模板。首先, 分析出适用于缓存的分块之后, 建立包含这些分块引用信息和位置信息的模板需要很大的工作量, 并且很难反映原来设计的原貌, 不能保证响应内容的强一致性。而本文的方法中采用的是利用生成动态页面的源码本身具有的模板信息, 在其中标记具体的分块, 方法简单, 并且最大限度的保证设计的初衷。其次, 文献[2, 11]更多的依赖于服务器端生成的 HTML 页面, 当页面中的某个分块发生变化时, 整个页面的业务逻辑都需要全部执行一遍, 并未减轻服务器端的压力。而本文中的方法则当某一部分发生变化时, 只执行这一分块的逻辑, 减少了重复计算。

文献[12]谈到了通过发现网页中的 Pagelets 自动检测模板的问题。首先, 该文献中模板的检测在于提高搜索算法的精确度, 而我们的方法则侧重于提高动态网页的服务质量。其次, 该文献只关注于发现模板, 而我们的方法不仅关注于模板, 还关注分块以及两者之间的关系。

5 结束语

在不重新设计系统的前提下, 将分块缓存技术应用与目前的动态网页系统, 会给系统带来很大的好处, 同时大幅减轻维护人员的工作。本文提出的方法, 有效的实现了系统的变迁, 同时还具有不间断系统服务提供, 支持原系统的变化和升级, 将逻辑执行粒度从页面降低到分块等优点。

在实现本文的方法中, 我们也遇到一些挑战, 也是下一步工作中待解决的问题, 如下所述:

(1) 在标记的过程中, 我们需要建立动态页面和源文件之间的映射关系来达到自动化。但是由于动态页面源文件(如 JSP 页面)中存在许多程序逻辑, 所以建立这种直接的映射存在一定的困难, 本实验中采用手工处理。如何通过语义或者标记的方式自动的建立这种映射, 将是一个研究重点。

(2) 不同的策略配置会给系统的性能带来不同的影响, 而我们的目标是找到适合该系统的最优配置. 所以需要借助机器学习, 控制论, 排队论和启发搜索等方法来得到合适的策略, 使该系统达到最优配置, 这也是我们下一步的研究工作之一.

参考文献:

- [1] Brodie D, Gupta A, Shi W S. Accelerating dynamic web content delivery using keyword-based fragment detection[J]. Journal of Web Engineering, 2005, 4(1): 79-100.
- [2] Challenger J, Iyengar A, Dantzig P. A scalable system for consistently caching dynamic Web data[A]. Proc of IEEE Conference on Computer Communications[C]. New York, NY, USA, 1999. 294-303.
- [3] Challenger J, Iyengar A, Witing K, et al. A publishing system for efficiently creating dynamic Web content[A]. Proc of IEEE Conference on Computer Communications[C]. Tel Aviv, Israel, 2000. 844-853.
- [4] Kephart J O, Chess D M. The vision of autonomic computing[J]. IEEE Computer, 2003, 36(1): 41-50.
- [5] Ramaswamy L, Iyengar A, Liu L, et al. Automatic detection of fragments in dynamic web pages and its impact on caching[J]. IEEE Trans, 2005, Knowl. Data Eng 17(6): 859-874.
- [6] DOM [Z/OL]. <http://www.w3.org/DOM>, 2005.
- [7] Apache Tomcat [Z/OL]. <http://tomcat.apache.org>, 2007.
- [8] OSCache [Z/OL]. <http://www.opensymphony.com/oscache>, 2007.
- [9] LoadRunner [Z/OL]. <http://www.mercury.com.cn/products/performance-center/loadrunner>, 2007.
- [10] Chabbouh I, Makpangou M. Caching dynamic content with automatic fragmentation[A]. Proc of the 7th International Conference on Information Integration and Web Based Applications and services[C]. Kuala Lumpur, Malaysia, 2002.
- [11] Bar-Yossef Z, Rajagopalan S. Template detection via data mining and its applications[A]. Proc of the 11th International World Wide Web Conference[C]. New York, NY, USA, 2002. 580-591.
- [12] Ma J, Gu Z. Fragment based delta encoding: A new scheme for efficient delivery of dynamic web pages[A]. Proc of the 6th IEEE International Conference on Computer and information Technology[C]. Washington, DC, USA, 2006. 101.

作者简介:



尤 朝 男, 1983 年 10 月出生于山西运城. 北京大学软件研究所博士研究生. 主要从事软件工程和中间件技术等方面的研究.

E-mail: youchao06@sei.pku.edu.cn



周明辉 女, 1974 年 1 月出生于湖南. 博士, 北京大学软件研究所副教授. 主要研究方向为分布计算, 软件工程和可信计算. 通信作者.

E-mail: zhmh@sei.pku.edu.cn