



[12] 发明专利说明书

专利号 ZL 200410061895.6

[45] 授权公告日 2009 年 4 月 22 日

[11] 授权公告号 CN 100480999C

[22] 申请日 2004.6.23

[21] 申请号 200410061895.6

[30] 优先权

[32] 2003.6.23 [33] US [31] 10/602, 122

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 A·W·史密斯 R·M·海华德
P·Y·-K·吴

[56] 参考文献

US6026413A 2000.2.15

US5897634A 1999.4.27

审查员 詹芊芊

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 谢喜堂

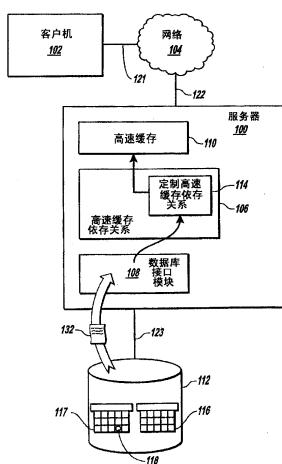
权利要求书 6 页 说明书 13 页 附图 4 页

[54] 发明名称

无效化高速缓存条目的一般依存关系模型

[57] 摘要

说明了用于衍生自定义高速缓存依存关系的系统、方法和计算机产品。揭示了一种包括可扩展高速缓存依存关系基类的框架，所述可扩展高速缓存依存关系基类可用来衍生用于在某些自定义条件下将高速缓存条目无效化的自定义高速缓存依存关系类。在一些实施例中，web 页服务器包括高速缓存以及包括可扩展高速缓存依存关系基类的框架。披露了用于引发高速缓存条目依赖于定制高速缓存依存关系并用于基于定制高速缓存依存关系将高速缓存条目无效化的方法。例如，高速缓存条目可以依赖于数据库表和 web 服务，并基于其依存关系被清除。



1. 在向请求客户计算机系统提供 web 页的服务器计算机系统中，所述 web 页可能包括来自所述服务器计算机系统外部的位置的内容，所述服务器计算机系统包括一高速缓存，其储存先前从所述服务器计算机系统外部的位置接收的高速缓存的内容的部分，一种用于使高速缓存条目依赖于定制依存关系的方法，其特征在于，所述方法包括以下步骤：

访问可扩展高速缓存依存关系基类的步骤；

从所述可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类的步骤，所述定制高速缓存依存关系类用于实现定制依存关系；

访问将要向客户计算机系统传送的内容的一部分的步骤；

创建与定制高速缓存依存关系关联的高速缓存条目的步骤；以及

将所述高速缓存条目插入所述高速缓存以使所述高速缓存条目的有效性依赖于所述定制依存关系的步骤。

2. 如权利要求 1 所述的方法，其特征在于，访问可扩展高速缓存依存关系基类的所述步骤包括：

访问包括通知依存关系改变方法的高速缓存依存关系基类的步骤，所述步骤使从所述高速缓存依存关系基类衍生的类能够通过调用所述通知依存关系改变方法来实现所述高速缓存依存关系基类的清除功能。

3. 如权利要求 1 所述的方法，其特征在于，访问可扩展高速依存关系基类的所述步骤包括访问包括依存关系处置方法的高速缓存依存关系基类的步骤，所述步骤使从所述高速缓存依存关系基类衍生的类能够通过调用所述依存关系处置方法来释放由高速缓存条目占用的资源。

4. 如权利要求 1 所述的方法，其特征在于，从所述可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类的所述步骤包括衍生能够实现来引发高速缓存条目依赖于数据库表的定制高速缓存依存关系类的步骤。

5. 如权利要求 1 所述的方法，其特征在于，从所述可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类的步骤包括衍生能够实现以使高速缓存条目依赖于 web 服务的定制高速缓存依存关系类的步骤。

6. 如权利要求 1 所述的方法，其特征在于，访问将要向客户计算机系统传送的内容的一部分的所述步骤包括访问将要包括在 web 页内的内容的一部分的步骤。

7. 如权利要求 1 所述的方法，其特征在于，访问将要向客户计算机系统传送的内容的一部分的所述步骤包括访问来自数据库表的内容的一部分的步骤。

8. 如权利要求 1 所述的方法，其特征在于，访问将要向客户计算机系统传送的内容的一部分的所述步骤包括访问来自 web 服务的内容的一部分的步骤。

9. 如权利要求 1 所述的方法，其特征在于，创建将所述定制依存关系与所访问的内容的一部分相关联的高速缓存条目的所述步骤包括创建使来自数据库表的内容依赖于所述数据库表的高速缓存条目的步骤。

10. 如权利要求 1 所述的方法，其特征在于，创建将所述定制依存关系与所访问的内容的一部分相关联的高速缓存条目的所述步骤包括创建使来自数据库表的内容依赖于聚集依存关系的高速缓存条目的步骤。

11. 如权利要求 1 所述的方法，其特征在于，创建将所述定制依存关系与所访问的内容的一部分相关联的高速缓存条目的所述步骤包括创建使来自 web 服务的内容依赖于所述 web 服务的高速缓存条目的步骤。

12. 如权利要求 1 所述的方法，其特征在于，将所述高速缓存条目插入高速缓存以使所述高速缓存条目的有效性依赖于所述定制依存关系的所述步骤包括插入依赖于数据库表的高速缓存条目的步骤。

13. 如权利要求 1 所述的方法，其特征在于，将所述高速缓存条目插入高速缓存以使所述高速缓存条目的有效性依赖于所述定制依存关系的所述步骤包括插

入依赖于聚集依存关系的高速缓存条目的步骤。

14. 如权利要求 1 所述的方法，其特征在于，不需要指定开始时间属性来调用所述可扩展高速缓存依存关系基类的功能。

15. 在向请求客户计算机系统提供 web 页的服务器计算机系统中，所述 web 页可能包括来自所述服务器计算机系统外部的位置的内容，所述服务器计算机系统包括储存先前从所述服务器计算机系统外部接收到的高速缓存的内容的部分的高速缓存，一种用于清除高速缓存条目的方法，其特征在于，所述方法包括以下步骤：

监控与定制高速缓存依存关系的实例相关联的一个或多个自定义依存关系条件的步骤；

确定是否满足所述一个或多个自定义依存关系条件的步骤；

接收满足所述一个或多个自定义依存关系条件的指示的步骤；以及

作为对接收所述指示的响应，在所述服务器计算机系统上清除高速缓存条目的步骤。

16. 如权利要求 15 所述的方法，其特征在于，监控与定制高速缓存依存关系的实例相关联的一个或多个自定义依存关系条件的所述步骤包括监控数据库表依存关系的步骤。

17. 如权利要求 15 所述的方法，其特征在于，监控与定制高速缓存依存关系的实例相关联的一个或多个自定义依存关系条件的所述步骤包括监控 web 服务依存关系的步骤。

18. 如权利要求 15 所述的方法，其特征在于，监控与定制高速缓存依存关系的实例相关联的一个或多个自定义依存关系条件的所述步骤包括监控聚集依存关系的步骤。

19. 如权利要求 15 所述的方法，其特征在于，确定是否满足所述一个或多个自定义依存关系条件的所述步骤包括确定标记条目是否增加的步骤。

20. 如权利要求 15 所述的方法，其特征在于，确定是否满足所述一个或多个自定义依存关系条件的所述步骤包括确定是否满足与聚集依存关系关联的依存关系条件的步骤。

21. 如权利要求 15 所述的方法，其特征在于，确定是否满足所述一个或多个自定义依存关系条件的所述步骤包括在发生由定制高速缓存依存关系类的实例指定的某一事件时调用通知依存关系改变方法。

22. 如权利要求 15 所述的方法，其特征在于，接收满足所述一个或多个自定义依存关系条件的指示的所述步骤包括从通知依存关系改变方法接收满足所述一个或多个自定义依存关系条件的指示的步骤。

23. 如权利要求 15 所述的方法，其特征在于，作为对接收所述指示的响应在所述服务器计算机系统上清除高速缓存条目的所述步骤包括清除依赖于数据库表的高速缓存条目的步骤。

24. 如权利要求 15 所述的方法，其特征在于，作为对接收所述指示的响应在所述服务器计算机系统上清除高速缓存条目的所述步骤包括清除依赖于 web 服务的高速缓存条目的步骤。

25. 如权利要求 15 所述的方法，其特征在于，作为对接收所述指示的响应在服务器计算机系统上清除高速缓存条目的所述步骤包括清除依赖于聚集依存关系的高速缓存条目的步骤。

26. 如权利要求 15 所述的方法，其特征在于，它还包括：
释放为了保留所述清除的高速缓存条目而被消耗的资源的步骤。

27. 如权利要求 26 所述的方法，其特征在于，释放为了保留所述清除的高速缓存条目而被消耗的资源的所述步骤包括调用与定制高速缓存依存关系相关联的依存关系处置方法，所述定制高速缓存依存关系从可扩展高速缓存依存关系基类衍生。

28. 一种在服务器计算机系统中使用的装置，所述服务器计算机系统向请求客户计算机系统提供 web 页，所述 web 页可能包括来自所述服务器计算机系统外部的位置的内容，所述服务器计算机系统包括储存先前从所述服务器计算机系统外部的位置接收的高速缓存的内容的部分的高速缓存，其特征在于，所述装置用于使高速缓存条目依赖于定制依存关系，该装置包括：

访问可扩展高速缓存依存关系基类的装置；

从所述可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类的装置，所述定制高速缓存依存关系类用于实现所述定制依存关系；

访问将要向客户计算机系统传送的内容的一部分的装置；

创建将所述定制依存关系与所述访问的内容的部分相关联的高速缓存条目的装置；以及

将所述高速缓存条目插入高速缓存使所述高速缓存条目的有效性依赖于所述定制依存关系的装置。

29. 一种在服务器计算机系统中使用的装置，所述服务器计算机系统向请求客户计算机系统提供 web 页，所述 web 页可能包括来自所述服务器计算机系统外部的位置的内容，所述服务器计算机系统包括储存先前从所述服务器计算机系统外部的位置接收的高速缓存的内容的部分的高速缓存，其特征在于，所述装置用于清除高速缓存条目，所述装置包括：

监控与定制高速缓存的实例相关联的一个或多个自定义依存关系条件的装置；

确定是否满足所述一个或多个自定义依存关系条件的装置；

接收满足所述一个或多个自定义依存关系条件的指示的装置；以及

作为对接收所述指示的响应在服务器计算机系统上清除高速缓存条目的装置。

30. 一种 web 页服务器，其特征在于，它包括：

处理器，所述处理器包括：

访问可扩展高速缓存依存关系基类的装置；

从所述可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类的装

置，所述定制高速缓存依存关系类用于实现所述定制依存关系；

访问将要向客户计算机系统传送的内容的一部分的装置；

创建将所述定制依存关系与所述访问的内容的部分相关联的高速缓存条目的装置；以及

将所述高速缓存条目插入高速缓存使所述高速缓存条目的有效性依赖于所述定制依存关系的装置；以及

一个或多个计算机可读媒质，其上储存以下内容：

高速缓存；

框架，其中，所述框架包括：

用于衍生自定义依存关系类的可扩展高速缓存依存关系基类，其中，所述自定义依存关系类用来定义至少一个事件，所述事件的出现使得所述高速缓存中包含的高速缓存条目被清除。

无效化高速缓存条目的一般依存关系模型

技术领域

本发明一般涉及对高速缓存的内容的管理，尤其涉及扩展高速缓存依存关系(dependency)类从而使开发者能够创建自定义高速缓存依存关系。

背景技术

在现代计算机系统和网络系统中，数据高速缓存是一种加快向请求数据一方的数据传递的有效工具。一般而言，数据高速缓存涉及在数据能够被快速检取之处存储哪些被频繁地请求的数据。在一个示例中，存储在计算机系统的物理存储器中的数据能够比存储在计算机系统的硬盘上的数据更快地被检取。这部分地是由于与物理存储器相比，硬盘通常具有更大的数据检取等待时间。同样，储存在服务器计算机系统的硬盘或存储器中的数据能够比储存在远程数据库中的数据更有效地被访问。此外，使用缓存的数据能够减少生成传递到请求客户计算机系统的内容所需要的计算量。

现代网络已经方便了与各种位置广泛传输数字信息的能力。一种令数据可用于传输的方法是通过创建一个静态文件，如 HTML 文件，该文件能够从服务器下载至客户计算机系统以供查看。静态文件能够高速缓存到多种位置，包括服务器计算机系统、区域数据库或者甚至在客户计算机系统本地。当计算机系统静态文件储存在客户计算机系统中时，客户计算机系统为所有实际目的的检取和显示都是瞬时的。即使当静态文件直接储存在连接至客户计算机系统（通常称为输出高速缓存）的服务器计算机系统中时，对高速缓存的内容的检取可以相对快速地出现。此外，由于其静态特性，静态文件能够长时间地高速缓存，本质上是无限的。

不幸的是，静态文件无法包括实时或自定义内容。经常期望呈现周期性的变化的内容或从其它内容的计算或操作而形成的内容。尽管这类内容仍然可以被高速缓存，然而高速缓存的内容的值仅在有限的持续时间内有效。由此，必须刷新高速缓存来确保数据有效。

一种协助 web 开发者创建 web 页和 web 站点的技术是使用运行在 web 页服务

器，如因特网信息服务器上的计算机可执行代码的脚本或模块化片断。脚本可以嵌入其它 HTML 标签之中来创建动态且可定制的 web 页。正如对象可以在诸如写在 Visual Basic、JavaScript 和 C#中的程序中被调用一样，同样的对象也可以在基于这些或其它语言的脚本中被调用。

类是 ASP.NET 框架等编程框架中的对象。ASP.NET 框架包括上千种 web 页开发者可以用来开发 web 页的类。以与通过目录路径访问文件相类似的方式，能够通过引用特定名称空间来访问类。例如，对名称空间 System.WinForms.Form 的引用提供了特定类的位置的指示。类可以与属性、时间和方法相关联。属性可以认为是相应类中的一些可变赋值或设定（如， $x = 5$ ）。事件是当事件出现时即升起的标志。其它类或其它应用可以读取事件来引发采取一些步骤。方法提供了调用相应类的功能来执行步骤（如，向显示设备输出数据）的方式。方法定义如何向外部组件提供功能。

为方便内容的高速缓存并因此提高 web 页服务器的性能，编程框架可包括用于从各种位置，如文件、数据库和 web 服务器中检取数据的类，以及用于通过各种操作来处理数据的类。在一些环境中，具有对这些类的接入的 web 页服务器计算机系统（如 ASP 服务器）可创建并储存基于实时内容的 HTML 代码，该代码在输出高速缓存的情况下被发送到客户计算机系统。在其它环境中，web 页服务器计算机系统可在 web 页服务器计算机系统上本地地高速缓存内容，如 XML 文件，使得不需要在每次请求包括该内容的 web 页时检取或重新计算内容。

一些编程框架允许 web 开发者创建指令，当执行该指令时，促使内容被储存 在高速缓存中。例如，为在高速缓存中插入一条目，可执行以下指令：`Cache.Insert([key],[value],[dependency])`，其中 key 是高速缓存条目的引用标识符， value 是放置在高速缓存条目中的值（或内容），dependency 是可任选自变量，定义了应当何时从高速缓存中清除 key 的值。依存关系是类的一个实例，如高速缓存依存关系（CacheDependency）类，可以用来使高速缓存值依赖于特定的事件的出现。

一些编程框架支持高速缓存依存关系，导致高速缓存的值依赖于时间、文件或关键字中的任一个。时间高速缓存依存关系使与特定依存关系关联的条目在特定时间段过去之后被清除。例如，`Cache.Insert` 指令可以如下实现，来使数据依赖于时间：

```
Cache.Insert(strCacheKey, dataset, nothing, DateTime.Now.AddMinutes(60),
```

TimeSpan.Zero, CacheItemPriority.High)

更明确地, Date.Time.Now.AddMinutes(60)指令使高速缓存中的条目在 60 分钟之后从高速缓存中清除。使用设置得太短的时间高速缓存依存关系将导致太频繁地清除高速缓存并且增加了频繁从数据库获取数据的成本或计算要储存在高速缓存中的数据的成本。如果将时间值设置得太长, 则有当储存在高速缓存中的数据被传送到请求客户计算机系统中时变为无效的风险。

Cache.Insert 指令也可以用来实现文件高速缓存依存关系, 使如果在特定文件中有变化时清除高速缓存条目。此外, Cache.Insert 指令可以用来实现关键字依存关系, 使当一些储存在存储器中的其它高速缓存值变化时清除一个或多个高速缓存条目。使用关键字依存关系, 一个高速缓存值的变化可以使许多高速缓存值从高速缓存中清除。

尽管上述三种高速缓存依存关系: 时间、文件和关键字在保持储存在高速缓存中的值最新方面是有用的, 然而实现这些依存关系的类通常是密封的。从而, web 开发者通常无法使高速缓存条目依赖于其它类型的事件。这妨碍了各种其它类型的高速缓存依存关系, 而这些其它类型的高速缓存依存关系在使用时能够令高速缓存的管理变为更有效。因此, 用于扩展可用来无效化高速缓存条目的依存关系的机制是非常有利的。

发明内容

上述相关技术的问题由本发明的原理来克服, 本发明针对用于扩展高速缓存依存关系类, 使开发者能够创建自定义高速缓存依存关系的方法、系统和计算机程序产品。服务器计算机系统 (如, 包括 web 页服务器) 访问可扩展高速缓存依存关系基类, 并从可扩展高速缓存依存关系基类来衍生定制的高速缓存依存关系。定制的高速缓存依存关系能够例如导致高速缓存条目依赖于数据库表 (如 SQL 表) 或 web 服务的变化。

定制的高速缓存依存关系可包括聚集依存关系 (aggregate dependency), 从而使高速缓存条目取决于许多事件。在部分实施例中, 定制的高速缓存依存关系的功能覆盖了可扩展高速缓存依存关系基类的功能。可扩展高速缓存依存关系基类可包括通知依存关系改变方法, 监控事件 (如, 数据库表中的变化) 并当出现事件时通知服务器计算机系统。可扩展高速缓存依存关系基类也可以包括依存关系处置方法, 当高速缓存条目从高速缓存中移除时释放资源。

服务器计算机系统从例如数据库或 web 服务访问将要传送到客户计算机系统的内容。服务器计算机系统创建将定制高速缓存依存关系与访问的内容相关联的高速缓存条目。服务器计算机系统将高速缓存条目插入到高速缓存中，使高速缓存条目的有效性（同时也是所访问的内容的有效性）依赖于定制的依存关系。服务器计算机系统能够使用 Cache.Insert 方法将高速缓存条目插入到高速缓存中，Cache.Insert 方法指定了为定制的高速缓存依存关系的实例的依存关系。

可在定制高速缓存依存关系的基础上将高速缓存条目无效化。服务器计算机系统（如包括 web 页服务器）接收出现监控的事件的通知。通知可以发自依存关系改变方法，该方法作为创建定制高速缓存的依存关系的结果而执行，并且监控与定制高速缓存的依存关系关联的事件。作为对通知的响应，服务器计算机系统将高速缓存条目无效化（并移除）。服务器计算机系统调用依存关系处置方法以释放该高速缓存条目使用的任何资源。

以下描述将详细阐明本发明的另外的特征和优点，阅读以下描述能够清除这些特征和优点，或者通过对本发明的实践而部分地学到。本发明的特征和优点可以通过所附权利要求书中特别指出的仪器和组合来实现和获取。

附图说明

为以能获取本发明的上述和其它优点和特征的方式来描述，参考附图中说明的具体实施例来特别描述上文概括描述的本发明。应当理解，这些附图仅描述了本发明的典型实施例，并且因此不是对其范围的局限，将通过使用附图描述本发明的另外的特征和细节，附图中：

图 1 说明了实现服务器端脚本和自定义高速缓存依存关系的 web 页服务器的示例。

图 2 所示是用于使高速缓存条目依赖于自定义依存关系的方法的一个实施例的示例流程图。

图 3 所示是用于基于自定义高速缓存依存关系来清除高速缓存条目的方法的一个实施例的示例流程图。

图 4 是适合在其中实践本发明的各方面的环境的图解表示。

具体实施方式

本发明延及用于扩展高速缓存依存关系类，以使开发者可以创建自定义高

速缓存依存关系的系统、方法和计算机程序产品。本发明的实施例将高速缓存依存关系扩展到一种可扩展形式，使开发者能够开发他们自己的自定义高速缓存依存关系。服务器计算机系统（例如包括 web 页服务器）访问其它定制的高速缓存依存关系（CacheDependency）类可从其衍生的可扩展高速缓存依存关系基类。定制的高速缓存依存关系类尤其能够利用可扩展高速缓存依存关系基类的高速缓存管理特征。通过创建从可扩展高速缓存依存关系基类衍生出的高速缓存依存关系类，开发者只需要很少的（如果有的话）如何实现可扩展高速缓存依存关系基类的知识来利用可扩展高速缓存依存关系基类的有利特征。可扩展高速缓存依存关系基类抽象了来自开发者的实现细节。

而且，开发者还可以访问可扩展高速缓存依存关系基类的方法，该方法能够处理高速缓存同步和高速缓存资源管理。另外，可扩展高速缓存依存关系基类可以向后兼容，使现存的 API 可以使用可扩展高速缓存基类及其衍生的类。这类 API 的示例包括 Cache.Insert、Cache.Add 以及 BasePartialCachingControl.Dependency。此外，从可扩展高速缓存基类衍生的自定义高速缓存的实现令开发者无需指定 StartTime（开始时间）属性以发信号通知应当何时开始高速缓存的监控。尽管开发者可以自由实现这一属性，它是不需要的。

可扩展高速缓存依存关系基类包括另外两个公共可用的方法，使定制高速缓存依存关系类的开发者能够调用可扩展高速缓存依存关系基类的清除能力。第一个方法是通知依存关系改变（NotifyDependencyChanged）方法。该方法由从可扩展高速缓存依存关系基类衍生的类调用，来告知可扩展高速缓存依存关系类依存关系目已改变。第二个新方法是依存关系处置（DependencyDispose）方法。该方法是一个虚方法，提供了对衍生的类特定的位置清除。更详细来说，该方法提供了指令的定位，该指令若非定制高速缓存依存关系，通常被置于可扩展高速缓存依存关系基类的 Dispose 方法中。由此，可以释放定制高速缓存依存关系所使用的资源，由需要访问该资源的其它模块使用。

通知依存关系改变（NotifyDependencyChanged）方法可由从可扩展高速缓存依存关系（CacheDependency）基类依赖的定制高速缓存依存关系调用。由此，定制高速缓存依存关系类能够使高速缓存条目依赖于数据库中、web 服务的数据上或鉴于任一其它原因的变化。从而，高速缓存能够识别无效的高速缓存条目关键字，并从高速缓存中移除该关键字（以及相应内容）。

在一个说明性示例中，开发者创建从可扩展高速缓存依存关系基类衍生的自定义类“SqlCacheDependency”，来轮询 SQL 数据库以确定是否对数据库作出了改变。SqlCacheDependency 从编程框架内的可扩展高速缓存依存关系基类衍生。因此，类 SqlCacheDependency 能够调用通知依存关系改变方法，令服务器计算机系统清除相关高速缓存条目。

使用本发明的一个实施例，使用以下指令能够实现定制高速缓存依存关系的实例：

```
10  If(Cache["Produce"]==null{  
20      SqlConnection Connection=new SqlConnection([connection  
details]);  
30      SqlDataAdapter ds=new SqlDataAdapter("Product_GetProducts",  
connection);  
...  
40      SqlCaheDependency dep=new SqlCacheDependency("NorthWind",  
"Products");  
50      Cache.Insert("Product", ds, dep);  
}
```

上述代码的行 10 检查高速缓存中是否已经有条目“Product”。如果该条目不在高速缓存中，则执行“if”语句中的函数。行 20 建立到要检取内容的数据库的连接。行 30 使用“Product_GetProducts”方法从数据库检取数据并将该数据储存在变量“ds”中。在行 40，变量“dep”被赋值为“NorthWind”数据库中“Products”表上的依存关系。这一依存关系是从可扩展高速缓存依存关系基类衍生的定制高速缓存依存关系。行 50 然后使变量“ds”中的数据插入到由关键字“Product”引用的高速缓存条目中，使该高速缓存条目取决于“dep”。

在该示例中，SqlCacheDependency 是从可扩展高速缓存依存关系基类衍生的定制高速缓存依存关系。当从可扩展高速缓存依存关系衍生的类用作依存关系时，服务器计算机系统将该类作为可扩展高速缓存依存关系的一个实例来处理。因此，服务器计算机系统能够响应由可扩展高速缓存依存关系基类展示的类似行为，而在内部，衍生的类可以实现其自己的特有功能，如当数据库查询改变时被通知。定制高速缓存依存关系，本例中即 SqlCacheDependency，覆盖

了可扩展高速缓存依存关系基类的任何方法，来获得所期望的功能。在本示例中，当向 SqlCacheDependency 通知数据库查询变化时，将调用 base.Notify.DependencyChanged 方法，使服务器计算机系统移除由关键字“Product”引用的高速缓存条目。

尽管本示例按照数据库变化依存关系来描述，重要的是要记住，可扩展高速缓存依存关系基类的可扩展性允许虚拟地创建 web 开发者可构想的任一高速缓存依存关系。采用可扩展模型，web 开发者能够建立任一类自定义依存关系，包括基于两个或更多依存关系的聚集依存关系。例如，高速缓存条目可能同时依赖于文件和 SQL 表。以下指令示出了聚集依存关系（aggregate dependency）的实现。

```
70  if(Cache["Product-List"]==null{  
80      SqlConnection Connection=new SqlConnection([connection details]);  
90      SqlDataAdapter ds=new SqlDataAdapter("Product_GetProducts",  
connection);  
100     SqlCacheDependency sqlDep=new SqlCacheDependency("NorthWind",  
"Products");  
120     CacheDependency dep=new  
AggregateCacheDependency(Server.MapPath("sales.xml"), sqlDep);  
130     Cache.Insert("Product-List, ds, dep);  
}
```

这一代码除行 120 外类似于上述标号为行 10-50 的代码。聚集高速缓存依存关系由高速缓存依存关系的衍生类，即 AggregateCacheDependency 实现。这可以用来使高速缓存中的条目同时依赖于文件（sales.xml）和从可扩展高速缓存依存关系基类衍生的定制 SqlCacheDependency。从可扩展高速缓存依存关系基类衍生的类也可以提供另外的定制功能。即，可以开发衍生类来处理用于实现定制功能的任意事件、方法和属性。例如，可以将衍生类开发为开始时间（例如，指定高速缓存依存关系何时应当开始监控变化的时间）、支持依存关系的聚集的能力、支持回叫机制等等。因此，开发者不局限于可扩展高速缓存依存关系基类的固有事件、方法和属性。

现在参考图 1，示出了适合在其中实践本发明的各方面的环境。图 1 描述服务器 100 和客户机 102，分别通过相应的链路 121 和 122 连接到网络。服务

器 100 和客户机 102 能够使用包括 HTTP 在内的多种协议通过网络 104 来通信。服务器 100 可以是 ASP.NET 服务器，能够向客户机 102 以动态 web 页的形式传送动态内容。服务器包括各种类和模块，包括可扩展高速缓存依存关系类 106 和定制高速缓存依存关系 114。服务器 100 也包括高速缓存 110，可能位于服务器 100 的物理存储器中或其它任一方便的位置。

服务器 100 通过链路 123 耦合至数据库 112。数据库 112 可以是任一合适的数据库，例如 SQL 数据库或 Oracle 数据库。在本发明的其它实施例中，数据库 112 可以表示 XML 服务或任一其它数据源。

用作说明，客户机 102 向服务器 100 请求特定的内容。例如，客户机 102 可能访问向服务器 100 请求 web 页的 URL。服务器 100 检查高速缓存 110 来确定所请求的内容（如，web 页中的数据）是否在高速缓存 110 中。如果该内容不在高速缓存 110 中，则数据库接口模块 108 可以向数据库 112 请求该内容。如果合适，内容，如内容 132 被检取并返回到数据库接口模块 108。

可选地，所请求的内容可能是从某一数据库，如数据库 112 中检取的内容的计算值或某种处理。数据库接口模块 108 可能将内容计算或处理为适当的。然后将内容发送至客户 102，如以一个或多个 HTTP 消息的形式。也可以使用 Cache.Insert 方法将该内容添加到高速缓存 110 中。Cache.Insert 方法指定由从可扩展高速缓存依存关系基类 106 衍生的定制高速缓存依存关系类 114 定义的自定义依存关系。

由于内容保留在高速缓存 110 中，它可以更有效地传送到任一请求该内容的客户机。该内容保留在高速缓存 110 中，直到定制高速缓存依存关系类 114 调用方法，如通知依存关系改变方法。在调用通知依存关系改变方法时，从高速缓存中清除该内容。客户机随后对该内容的请求需要从数据库 112 中重新获取该内容和/或由数据库接口模块 108 重新计算该内容。

定制高速缓存依存关系类 114 可以在数据库 112 中有变化时清除高速缓存 110。在本示例中，尽管从数据库 112 获取数据可能是资源密集型的，然而另一方面，向数据库 112 轮询变化使用较少的资源。数据库 112 可包括储存数据库 112 中的信息的数据库表，如数据库表 116。另一数据库表 117 还可包括标记条目 118，指示是否向数据库表 116 中的信息作出了改变。数据库表 117 可包含数据库中具有自定义依存关系的其它表的标记。在一个实施例中，标记条目 118 可以是递增的，令数据库表 116 中的每一变化使标记条目 118 递增地增

加其值。可以设计定制高速缓存依存关系 114 以当检测到标记条目 118 中的增加时移除高速缓存条目。

现在参考图 2，示出了用于插入依存关系条目的方法的流程图。图 2 的方法包括访问可扩展高速缓存依存关系基类的步骤（步骤 202）。步骤 202 可包括服务器计算机系统访问可扩展高速缓存依存关系基类。例如，服务器 100 可以访问可扩展高速缓存依存关系类 106。服务器 100 可访问高速缓存依存关系类 106 作为执行由 web 页开发者、其它软件开发者开发的或作为一些程序模块的一部分的指令的结果。

图 2 的方法包括从可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类（步骤 204）。步骤 204 包括服务器计算机系统从可扩展高速缓存依存关系基类衍生定制高速缓存依存关系类。例如，服务器 100 可以从可扩展高速缓存依存关系基类 106 衍生定制高速缓存依存关系 114。服务器 100 可衍生定制高速缓存依存关系 114，作为执行由 web 开发者、软件开发者开发的或者作为某一程序模块的一部分的指令的结果，来创建定制高速缓存依存关系。服务器 100 也可以从许多源衍生定制高速缓存依存关系 114，源包括但不限于：连接到计算机的外部传感器，如温度计，只要条件变化就触发事件；运动得分板程序，只要比赛中的得分变化就触发事件；网络管理程序，只要网络配置变化（如计算机添加到网络或从其中移除）就触发事件；以及目录程序，只要目录变化（如添加或删除人员）就触发事件。

图 2 的方法包括访问内容（步骤 206）。步骤 206 可包括服务器计算机系统访问将要包括在 web 页内用于向客户计算机系统传送的内容。例如，服务器 100 能够访问将要包括在 web 页内用于向客户机 102 传送的内容。访问内容可包括创建或处理内容。内容可以从多种不同的位置访问，如，从数据库和 web 服务。例如，服务器可以访问储存在数据库 112 上的数据。数据库接口模块 108 或其它合适的模块可以适当地格式化数据，用于包括在 web 页中。

图 2 的方法包括创建与定制高速缓存依存关系关联的高速缓存条目的步骤（步骤 208）。步骤 208 可包括服务器计算机系统创建与定制高速缓存依存关系关联的高速缓存条目。例如，服务器 100 能够创建与定制高速缓存依存关系 114 关联的高速缓存条目。在一个实施例中，高速缓存条目由服务器 100 作出的可变定义语句创建（如，类似于上述指令的行 50）。

图 2 的方法包括将高速缓存条目插入到高速缓存的步骤（步骤 210）。步

骤 210 可包括服务器计算机系统将高速缓存条目插入到服务器计算机系统的高速缓存位置。例如，服务器 100 可以将高速缓存条目插入到高速缓存 110 中。可以将高速缓存的条目插入到高速缓存中，使高速缓存的条目的有效性（并因此也使高速缓存的内容的有效性）依赖于定制依存关系。例如，服务器 100 可以将高速缓存条目插入到依赖于由定制高速依存关系 114 产生的依存关系的高速缓存 110 中。

现在参考图 3，示出了用于使用定制高速缓存依存关系来将高速缓存条目无效化的方法。图 3 的方法包括监控由定制高速依存关系的实例设置的自定义依存关系条件的步骤（步骤 302）。步骤 302 可包括服务器计算机系统监控由在服务器计算机系统上实现的定制高速缓存依存关系类的实例设置的自定义依存关系条件。例如，服务器 100 能够监控由定制高速缓存依存关系类 114 的实例设置的自定义依存关系条件。定制依存关系条件可以是事件的出现，如数据库的内容或 web 服务提供的内容中的变化。例如，服务器 100 能够监控标记条目 118 来监控数据库表 16 的内容的变化。当定制高速缓存依存关系类 114 与聚集依存关系关联时，可以监控多个自定义依存关系条件。

图 3 的方法包括确定是否满足了自定义依存关系条件的步骤（步骤 304）。步骤 304 可包括服务器计算机系统确定是否满足了与定制高速缓存的依存关系关联的自定义依存关系条件。例如，服务器 100 能确定是否满足与定制高速缓存依存关系类 114 关联的自定义依存关系条件。当定制高速缓存依存关系类 114 与聚集依存关系关联时，服务器 100 能确定是否满足多个自定义依存关系条件中的每一自定义依存关系条件。

在一些实施例中，服务器 100 通过将储存的标记条目（flag entry）与当前标记条目相比较来确定是否满足了自定义依存关系条件。例如，服务器 100 能将标记条目 118 的先前检取并储存的值与标记条目 118 的当前值相比较。当标记条目 118 是递增的时，定制高速缓存依存关系类 114 能将当前在数据库表 117 中的标记条目 118 与在某一之前的时刻在数据库表 117 中存在的标记条目相比较。如果当前在表中的标记条目 118 大于某一之前时刻在数据库表 117 中的标记条目，则满足自定义依存关系条件（数据库表 116 的内容变化）。

当不满足自定义依存关系条件时（步骤 304 之否），服务器 100 能够再一次监控自定义依存关系条件。可以以指定的时间间隔（如每 5 秒）重复监控自定义依存关系条件，直到满足一个或多个自定义依存关系条件，如，出现了监

控的事件（步骤 304 之是）。

图 3 的方法包括指示满足自定义依存关系条件的步骤（步骤 306）。步骤 306 可包括向服务器计算机系统指示满足自定义依存关系条件的通知依存关系改变方法。例如，定制高速缓存依存关系类 114 的通知依存关系改变方法能够在数据库 116 的内容变化时通知服务器 100。满足自定义依存关系条件的通知能够向服务器 100 指示要清除高速缓存条目。

图 3 的方法包括清除高速缓存（步骤 308）。步骤 308 可包括服务器计算机系统清除服务器计算机系统上的高速缓存条目。例如，服务器 100 能从高速缓存 110 中清除高速缓存条目。可以清除高速缓存的条目作为对接受满足与定制高速缓存依存关系关联的自定义依存关系条件的通知的响应。例如，服务器 100 可清除依赖于数据库表 116 的高速缓存的条目，作为对接受标记条目 118 增加的通知的响应。当清除高速缓存条目后，计算机系统 100 能够释放被消耗来保留高速缓存条目的资源（如系统存储器）。例如，服务器计算机系统 100 能够调用与定制高速缓存依存关系类 114 关联的依存关系处置方法来释放储存来自数据库表 116 的内容的高速缓存条目的资源。

本发明的实施例可包括专用或通用计算机，包括各种计算机硬件，如下文所详细讨论的。处于本发明的范围之内的实施例也包括用于携带或在其上储存计算机可执行指令或数据结构的计算机可读媒质。这类计算机可读媒质可以是可由通用或专用计算机访问的任何可用媒质。作为实例而非局限，这类计算机可读媒质包括 RAM、ROM、EEPROM、CD-ROM 或其它光盘存储、磁盘存储或其它磁存储设备、或可以用来以计算机可执行执行或数据结构携带或储存期望的程序代码方法并可由通用或专用计算机访问的任何其它媒质。当通过网络或另一通信连接（任一硬布线、无线或硬布线和无线的组合）向计算机传输或提供信息时，计算机适当地将该连接视为计算机可读媒质。因此，任一这类连接适当地称为计算机可读媒质。上述的组合也应当包括在计算机可读媒质的范围内。计算机可执行指令包括，如，使通用计算机、专用计算机或专用处理设备执行特定的函数或函数组的指令和数据。

图 4 及以下讨论提供了适合在其中实现本发明的计算环境的简要一般描述。尽管并非所需要，本发明在计算机可执行指令的一般语境下描述，计算机可执行指令如程序模块，由网络环境中的计算机执行。一般而言，程序模块包括例程、程序、对象、组件、数据结构等等，执行特定的任务或实现特定的抽

象数据类型。计算机可执行指令、相关数据结构以及程序模块代表了用于执行这里披露的方法的步骤的程序代码方法。这类可执行指令或相关数据结构的特定顺序代表用于实现这类步骤中描述的功能的相应动作的示例。

本领域的技术人员可以理解，本发明可以在具有多种类型的计算机系统配置的网络计算环境中实践，计算机系统配置包括个人计算机、手持式设备、多处理器系统、微处理器系统或可编程消费者电子产品、网络 PC、小型机、大型机等等。本发明也可以在分布式计算环境中实践，其中，任务由通过通信网络连接（或者通过硬布线链路、或者通过无线链路、或者通过硬布线和无线链路的组合）的本地和远程处理设备执行。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

参考图 4，用于实现本发明的示例系统包括以常规计算机形式 420 的通用计算装置，包括处理单元 421、系统存储器 422 以及将各类系统组件包括系统存储器 422 耦合至处理单元 421 的系统总线 423。系统总线 423 可以是若干种总线结构类型的任一种，包括存储器总线或存储器控制器、外围总线以及使用各类总线结构的本地总线。系统存储器包括只读存储器（ROM）424 和随机存取存储器（RAM）425。基本输入/输出系统（BIOS）426，包含如在启动时协助在计算机 420 内的元件之间传输信息的基本例程，可储存在 ROM 424 中。

计算机 420 也可包括用于对磁硬盘 439 进行读写的磁硬盘驱动器 427、用于对可移动磁盘 429 进行读写的磁盘驱动器 428 以及用于对可移动光盘 431 如 CD-ROM 或其它光媒质进行读写的光盘驱动器 430。磁硬盘驱动器 427、磁盘驱动器 428 以及光盘驱动器 430 分别通过硬盘驱动器接口 432、磁盘驱动器接口 433 和光盘驱动器接口 434 连接至系统总线 423。驱动器及其相关的计算机可读媒质为计算机 420 提供了计算机可执行指令、数据结构、程序模块和其它数据的非易失存储。尽管这里描述的示例环境采用了磁硬盘 439、可移动磁盘 429 以及可移动光盘 431，然而也可以使用用于储存数据的其它类型的计算机可读媒质，包括盒式磁带、闪存卡、数字多功能盘、Bernoulli 盒式磁盘、RAM、ROM 等等。

包括一个或多个程序模块的程序代码方法可储存在硬盘 439、磁盘 429、光盘 431、ROM 424 或 RAM 425 中，包括操作系统 435、一个或多个应用程序 436、其它程序模块 437 以及程序数据 438。用户可以通过键盘 440、指向设备 442 或其它输入设备（未示出），如麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等向计算机 420 输入命令和信息。这些和其它输入设备通常通过耦合至系统总线 423

的串行端口接口 446 连接到处理单元 421。可选地，输入设备也可以通过其它接口连接，如并行端口、游戏端口或通用串行总线（USB）。监视器 447 或另一显示设备也通过接口，如视频适配器 448 连接到系统总线 423。除监视器之外，个人计算机通常包括其它外围输出设备（未示出），如扬声器和打印机。

计算机系统 420 可连接到网络，如办公室范围或企业范围计算机网络、内联网和/或因特网。计算机系统 420 能够通过这类网络与外部源，如远程计算机系统、远程应用和/或远程数据库交换数据。

计算机系统 420 包括网络接口 453，计算机系统 420 通过该接口从外部源接收数据和/或向外部源发送数据。如图 4 所述，网络接口 453 方便了通过链路 451 与远程计算机系统 483 的数据交换。链路 451 表示网络的一部分，远程计算机系统 483 表示网络的一个节点。例如，远程计算机 483 可以维护提供用于在计算机系统 420 上构造的 web 页的内容的数据。另一方面，计算机系统 483 可以是向计算机系统 420 维护的数据库请求内容的服务器计算机。

类似地，计算机系统 420 包括串行端口接口 446，计算机系统 420 通过该接口从外部源接收数据和/或向外部源发送数据。串行端口接口 446 耦合至调制解调器 454，计算机系统 420 通过调制解调器从外部源接收数据和/或向其发送数据。可选地，调制解调器 454 可以是通过适当的接口连接到计算机系统 420 的有线数据业务接口技术规范（“DOCSIS”）调制解调器或数字用户线（“DSL”）调制解调器。然而，如图 4 所述，串行端口接口 446 和调制解调器 454 方便了通过链路 452 与远程计算机系统 493 的数据交换。链路 452 表示网络的一部分，远程计算机系统 493 表示网络的一个节点。例如，远程计算机系统 493 可以是向计算机系统 420 请求 web 页的客户计算机系统。另一方面，计算机系统 493 可以是向计算机系统 420 提供 web 页的服务器计算机系统。

本发明可以在不脱离其精神或本质特征的情况下以其它特定的形式来实施。所描述的实施例在所有方面仅为说明性的，而非限制性的。因此，本发明的范围由所附权利要求书而非上文的描述来指出。所有处于权利要求的等效技术方案的意思和范围之内的变化都包含在其范围之内。

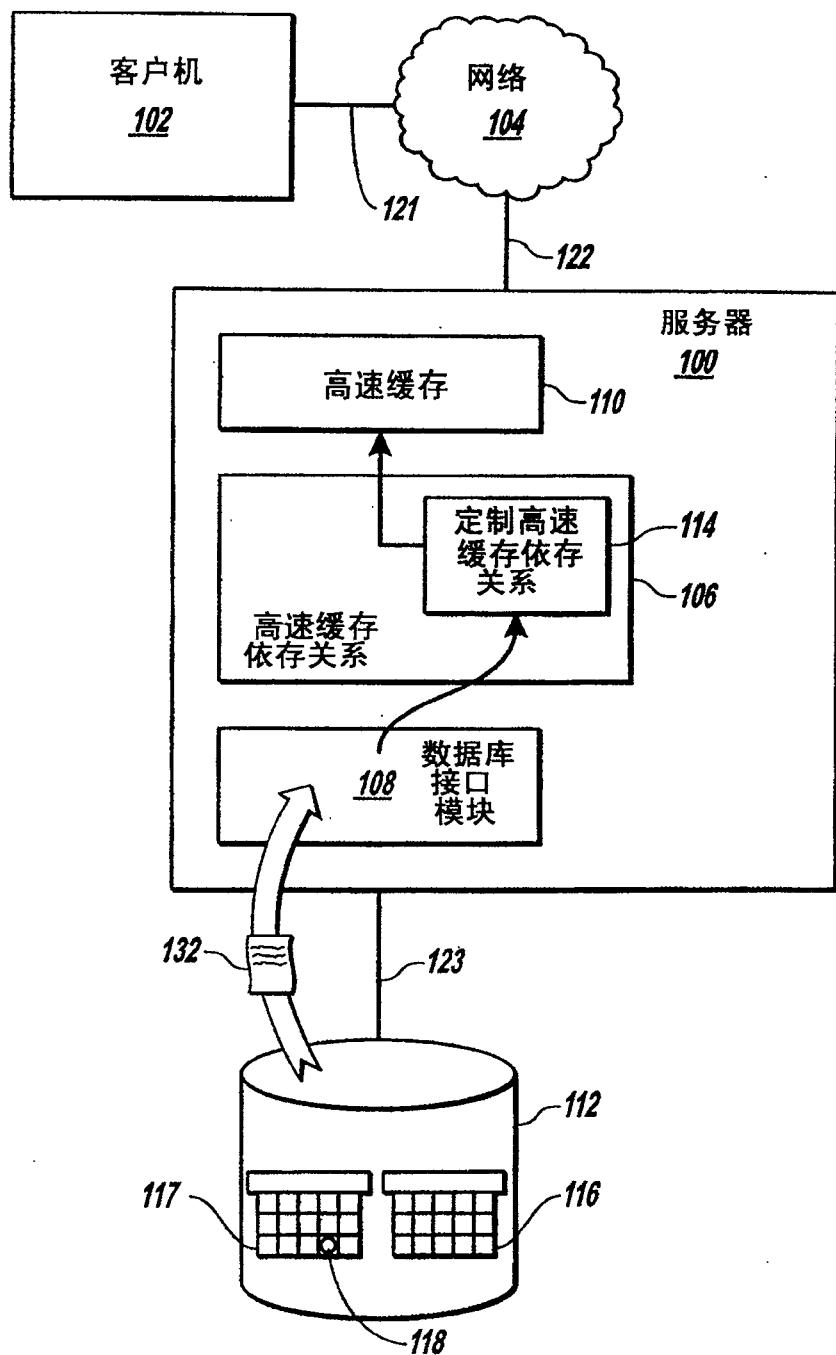


图 1

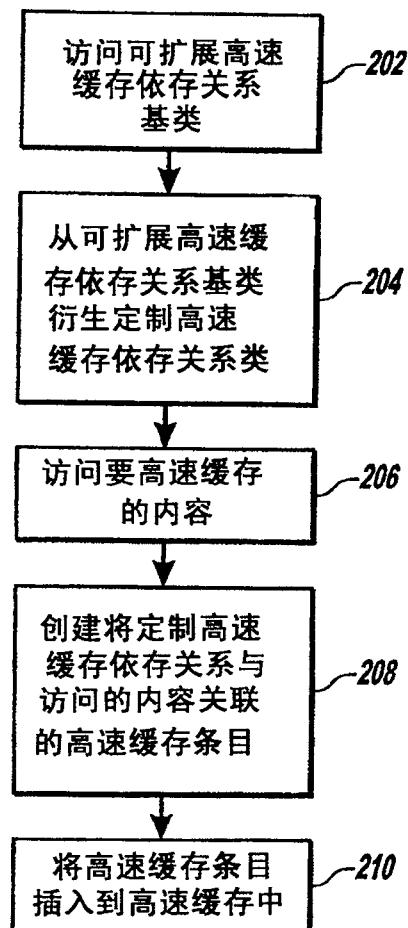


图 2

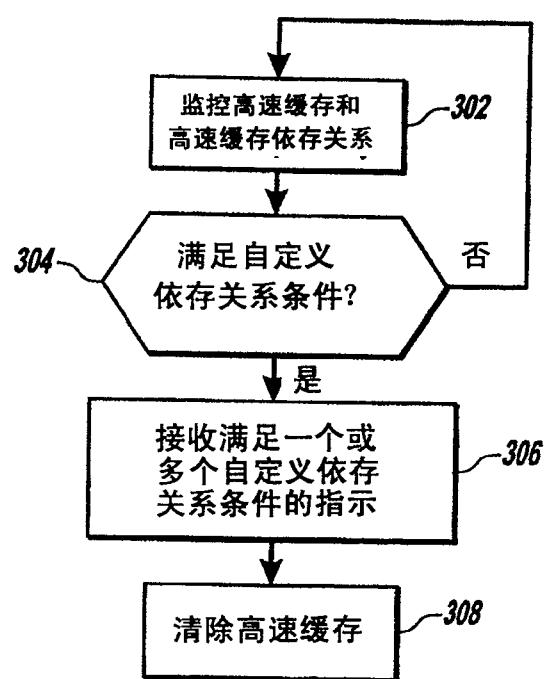


图 3

