

概述

识别并高速缓存动态 Web 应用：一款灵活的性能解决方案

在企业中部署动态 web 应用可带来诸多显著优势，但用户可能会遇到一些难以预料的性能问题。导致性能下降的一个原因是：应用使用率上升、内容复杂性增加，这导致服务器出现过载。F5 Networks 公司的 WebAccelerator 借助其动态高速缓存技术成功地解决了这一关键问题。动态高速缓存的独特之处在于，它可为用户提供高度动态的 web 支持型应用，在 web 基础设施成本显著降低的同时，速度提升了五至十倍。这种特性尤其适用于受到可扩充性限制以及需要支持更多用户的应用，并且，对于由传统设计和其它问题而导致速度缓慢的 web 应用，这种特性也彰显其独特之处。

挑战

所面临的挑战主要包括：提高最终用户应用的性能、以及动态 web 应用与内容的服务器卸载问题。通常，静态高速缓存只能用于 30% 的 HTTP 请求——该百分比值通常不包括具有更大价值的动态数据（查询响应、XML 数据等）。

解决方案

动态高速缓存

动态高速缓存彻底改变了高速缓存模式，能够对更广泛的内容进行高速缓存，包括高度动态 web 页面、查询响应以及 XML 对象。该项专利技术为 F5 Networks 公司独家拥有，仅可从 F5 Networks 公司获得。

动态高速缓存注重应用逻辑与行为，而不仅仅是个人 web 对象。通过了解某项应用的高级逻辑（可高速缓存和不可高速缓存的内容、可引发无效的事件等），WebAccelerator 可排除对复杂 web 请求的重复处理。动态高速缓存支持 WebAccelerator 系统决定何时使对象无效及如何识别可复用的内容块。这样，通过将预定义的应用加速策略、直观的用户界面、基于 XML 功能强大的 API (ESI) 及基于 HTTP 请求的触发装置相结合，便可提供使内容生效或无效的全面控制。

现有的高速缓存解决方案如果不具备 WebAccelerator 和高速缓存，则只能将对象的有效期作为参考。动态高速缓存支持高速缓存查看 HTTP 请求中的任意内容，包括 URL、cookies、查询参数，以及其它标头等，这有助于生成更加“智能”的无效信息和高速缓存密钥。此外，它还支持 WebAccelerator 系统决定何时使对象无效及如何识别单个内容块。

借助动态高速缓存，WebAccelerator 可直接响应高达 80% 的用户请求（此类请求计算成本颇高），且不涉及站点基础设施的其它部分。此外，WebAccelerator 不受应用语义的困扰，不会将高速缓存中的无效内容发送出来。

静态高速缓存

作为动态高速缓存能力的扩展，WebAccelerator 还可提供静态高速缓存功能。静态高速缓存仅缓存图像、javascript，以及

级联样式表等未到期的对象。即使静态高速缓存可能已存在于应用的计算基础设施中，**WebAccelerator** 也支持静态对象从原有应用中删除其它高开销操作。

内容识别

要想对应用进行高速缓存，有必要对每个单项内容进行准确识别。**WebAccelerator** 首次收到特定内容项请求时，会以代理形式将该请求发送至源服务器，并为用户取回该内容。当 **WebAccelerator** 检索页面时，在发送响应至客户端之前，**WebAccelerator** 会把页面副本转换为一条经过编译的内部请求。利用这些经过编译的响应，**WebAccelerator** 可重建一个页面，作为对 HTTP 请求之响应。

经过编译的响应由页面内部请求（从源站点获得）和指令（描述如何从内部请求重建页面）构成，其中包括用于更新（以随机或循环内容更新）页面的信息。

根据请求中的 **URI**、查询参数等元素，**WebAccelerator** 还能为经过编译的响应指定一个唯一内容标识符 (**UCI**)，**UCI** 可用于请求以及经过编译的响应，旨在为请求提供服务。如果收到的下一个请求包含相同元素且生成相同的 **UCI**，则 **WebAccelerator** 会找到已经缓存的响应，并用之以响应请求。

将这些元素作为标识符的一部分能够使未来请求与高速缓存中的相应内容相匹配。不影响页面内容的 HTTP 请求元素将被忽略且不可在 **UCI** 中使用，因此，这些元素的设定值不可用于识别高速缓存内容的特定实例。

但并非请求中的所有元素都代表一个特有的响应。例如，两个具有相同 **URI**、方法及查询参数的请求，其 **cookie** 不同时，您的源服务器仍可能生成相同的响应。在缺省状态下，**WebAccelerator** 假定某些元素可导致内容变化而其它元素不会。此外，该信息还可用于创建一个 **UCI**，可通过 **WebAccelerator** 的变更策略进行配置。

实例

让我们看看帮助中心应用，这类应用管理一套持续进行变化的琐碎的记录。参考每项记录可能需要多种方式，每条记录代表一个将要被高速缓存的单独对象。如要识别单独的记录，应用需要以其数据库生成的特定的 **ID** 为依据。不同的用户可浏览相同的数据。可按如下方式请求：

```
GET /helpdeskapp/viewticket.asp?ticketid=121& parentname=Login HTTP/1.1 Host: helpdesk.mycompany.com Cookie:
userid=323; sessionid=3xx3s
```

这样，通过以下密钥，即可对 **WebAccelerator** 进行配置，以识别记录并进行高速缓存。

```
[/helpdeskapp/viewticket.asp, ticketid=121].
```

现在，将其与完全依赖于用户身份生成内容的其它应用相比较，如电子邮件客户端。当对收件箱进行高速缓存时，请求如下声明：

```
GET /mymail/inbox.asp?page=1& parentname=Login HTTP/1.1 Host: mymail.mycompany.com Cookie: userid=323;
```

sessionId=3xx3s

然后，利用以下密钥对 WebAccelerator 进行配置以识别该对象：

[/mymail/inbox.asp, page=1, userid=323].

如果希望高效地高速缓存动态内容，则有必要保持识别内容时的灵活性。为了保持功能性，系统必须始终将不同的内容块作为单独的对象，对其进行高速缓存，并且，真实对象和高速缓存密钥之间必须存在一一对应的关系。

保持应用的真实性

由 web 应用生成的内容不断发生变化。导致变化发生的事件广义上分为三类：时间、用户事件和应用事件。在每种事件中，动态高速缓存（能够用于描述何时发生变化以及哪种高速缓存对象受到了影响）内的 WebAccelerator 都有其各自的机制。基于如下两项因素，WebAccelerator 可高速缓存无效信息：

☆ 有效时间，对于高速缓存静态或接近静态的内容，该项非常实用。例如，生成每周财务报告的应用。之前的报告无法确定高速缓存的有效时间，题为“当前”报告的高速缓存有效时间为一个星期。当“过期”时，WebAccelerator 使高速缓存的对象无效。

☆ 用户事件，用户与应用进行交互活动时发生该事件。WebAccelerator 监控符合特定已知标准（可变更应用的状态）的 HTTP 请求。WebAccelerator 查找到这种请求时，会执行一条指令，使高速缓存的相关扇区失效。以留言板（Message boards）为例。单个线索的读取频率远远超过写入频率，因此在高速缓存当中，该应用具有极高的价值。如果希望部署动态高速缓存，在用户提交特定线索时，需对 WebAccelerator 进行配置以识别线索（请参见如下图 1）。WebAccelerator 可高速缓存线程文本，初次使用时，需直接从应用中调用，之后仅调用高速缓存即可。当用户提交线索时，WebAccelerator 首先对其进行识别，之后使与该线索有关的被缓存对象无效。当下一个用户请求线索时，WebAccelerator 从应用中取回该线索，重新送至高速缓存。

☆ 应用事件，该事件来自用户与应用交互活动的外部。当应用事件发生时，WebAccelerator 在接受来自应用、基于标准的 XML 无效消息 (ESI) 后，对应用事件进行处理。

以新消息的接收为例，该实例是一个面向电子邮件应用的外部事件（请参见以下图 2）。当没有新电子邮件时，

WebAccelerator 直接从高速缓存中响应。当新消息到达时，通知从电子邮件应用传递至 WebAccelerator，指示用户哪一个收件箱发生了变化。然后，当用户再次访问收件箱时，WebAccelerator 将请求发送至电子邮件应用，收件箱列表更新。

另一个实例是电子商务站点上的产品价格变更。电子商务站点上的价格频繁变更，也许每天有百分之一的页面发生改变。当价格发生变更时，WebAccelerator 能够删除以前高速缓存的页面。而其它厂商的静态高速缓存则无法做到这一点。

结论

通过采用以下两种密钥能力，WebAccelerator 解决了长期以来对动态内容进行高速缓存的难题：

☆ 可将符合条件的用户请求与高速缓存的内容连接起来的一套完善的匹配算法

☆ 可由应用和用户事件触发的高速缓存无效信息的机制

动态高速缓存可将服务器负载和延迟降低 80%、提高用户性能、降低基础设施成本、为意外或计划外站点故障提供标准性能，此外，无需更改站点架构或代码，即可精确地控制内容的精度。此外，动态高速缓存还可与应用无缝集成，支持各种软件和站点类型，而不依赖于特定的基础设施部件。