(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104657358 A
(43)申请公布日 2015.05.27

(21)申请号 201310576768.9

(22)申请日 2013.11.15

(71)申请人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72)发明人 于小军

(74)专利代理机构 广州华进联合专利商标代理有限公司 44224
代理人 何平 邓云鹏

(51)Int.Cl.
G06F 17/30(2006.01)

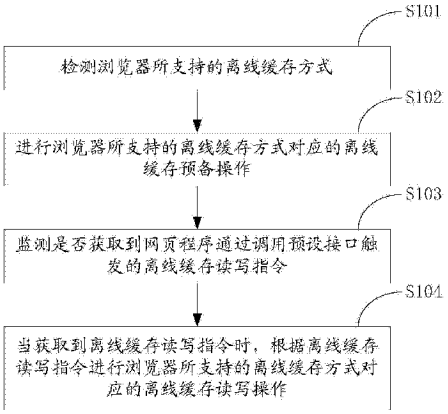
权利要求书2页 说明书10页 附图3页

(54)发明名称

实现网页程序离线缓存的方法和系统

(57)摘要

一种实现网页程序离线缓存的方法，包括：检测浏览器所支持的离线缓存方式；进行所述离线缓存方式对应的离线缓存预备操作；监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令；当获取到所述离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。上述方法在网页程序进行离线缓存之前，检测浏览器所支持的离线缓存方式，并进行对应的离线缓存预备操作，网页程序只需要调用预设接口以触发离线缓存读写指令，即可实现离线缓存读写操作，不需要网页程序兼容各种离线缓存方式，从而降低了网页程序的代码复杂度、提高了网页程序的运行效率。此外，还提供一种实现网页程序离线缓存的系统。



CN 104657358 A

1. 一种实现网页程序离线缓存的方法，包括以下步骤：

检测浏览器所支持的离线缓存方式；

进行浏览器所支持的离线缓存方式对应的离线缓存预备操作；

监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令；

当获取到所述离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

2. 根据权利要求 1 所述的实现网页程序离线缓存的方法，其特征在于，检测浏览器所支持的离线缓存方式的步骤包括以下步骤：

按照预设顺序依次检测浏览器是否支持 Web SQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式，当检测到浏览器支持的离线缓存方式时，则停止检测，并设置浏览器所支持的离线缓存方式为当前检测的离线缓存方式，若未检测到浏览器支持的离线缓存方式，则设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。

3. 根据权利要求 1 所述的实现网页程序离线缓存的方法，其特征在于，进行浏览器所支持的离线缓存方式对应的离线缓存预备操作包括以下步骤：

当浏览器所支持的离线缓存方式为 File System 离线缓存方式时，判断用于缓存的文件是否存在，若不存在，则生成预设数量的用于缓存的文件，并打开用于缓存的文件。

4. 根据权利要求 1 所述的实现网页程序离线缓存的方法，其特征在于，所述预设接口的传入参数包括网页程序将要进行离线缓存读写操作的数据的关键字，通过调用预设接口触发的离线缓存读写指令中包含从预设接口传入的关键字。

5. 根据权利要求 4 所述的实现网页程序离线缓存的方法，其特征在于，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作包括以下步骤：

当浏览器所支持的离线缓存方式为 File System 离线缓存方式时，根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在该文件中进行与离线缓存读写指令相对应的操作。

6. 一种实现网页程序离线缓存的系统，其特征在于，包括：

检测模块，用于检测浏览器所支持的离线缓存方式；

初始化模块，用于进行浏览器所支持的离线缓存方式对应的离线缓存预备操作；

指令接收模块，用于监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令；

缓存控制模块，用于当获取到所述离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

7. 根据权利要求 6 所述的实现网页程序离线缓存的系统，其特征在于，所述检测模块用于按照预设顺序依次检测浏览器是否支持 Web SQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式，当检测到浏览器支持的离线缓存方式时，则停止检测，并设置浏览器所支持的离线缓存方式为当前检测的离线缓存方式，若未检测到浏览器支持的离线缓存方式，则设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。

8. 根据权利要求 6 所述的实现网页程序离线缓存的系统，其特征在于，所述初始化模

块用于当浏览器所支持的离线缓存方式为File System离线缓存方式时，判断用于缓存的文件是否存在，若不存在，则生成预设数量的用于缓存的文件，并打开用于缓存的文件。

9. 根据权利要求6所述的实现网页程序离线缓存的系统，其特征在于，所述预设接口的传入参数包括网页程序将要进行离线缓存读写操作的数据的关键字，通过调用预设接口触发的离线缓存读写指令中包含从预设接口传入的关键字。

10. 根据权利要求9所述的实现网页程序离线缓存的系统，其特征在于，所述缓存控制模块用于当浏览器所支持的离线缓存方式为File System离线缓存方式时，根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在该文件中进行与离线缓存读写指令相对应的操作。

实现网页程序离线缓存的方法和系统

【技术领域】

[0001] 本发明涉及互联网技术领域，特别涉及一种实现网页程序离线缓存的方法和系统。

【背景技术】

[0002] HTML5 是 HTML（Hypertext Markup Language，超文本标记语言）的一个标准版本，现在仍处于发展阶段。广义的 HTML5 指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合。

[0003] HTML5 的一个特点是支持离线缓存。离线缓存就是让网页应用程序即使在断网的情况下依然可以正常的运行。

[0004] HTML5 支持的离线缓存一般采用本地存储库（localstorage）、网页关系型数据库（Web SQL Database）、索引数据库（indexedDB）或文件系统（File System）等进行本地存储，对应上述四种存储方式的离线缓存可分别称之为 localstorage 离线缓存方式、Web SQL Database 离线缓存方式、indexedDB 离线缓存方式和 File System 离线缓存方式。

[0005] 然而，除了 localstorage 离线缓存方式被所有的浏览器支持之外，其它的离线缓存方式并不被所有的浏览器支持。例如，对于 Web SQL Database 离线缓存方式、indexedDB 离线缓存方式和 File System 离线缓存方式，只有 chrome 浏览器支持全部三种离线缓存方式，而大部分浏览器只支持其中的一种或两种。

[0006] 因此，网页程序要想在各个浏览器上运行并支持离线缓存，则需要兼容各个浏览器所支持的离线缓存方式，从而导致网页程序代码复杂度高，进一步导致网页程序运行效率降低。

【发明内容】

[0007] 基于此，有必要提供一种能提高网页程序运行效率的实现网页程序离线缓存的方法。

[0008] 一种实现网页程序离线缓存的方法，包括以下步骤：

[0009] 检测浏览器所支持的离线缓存方式；

[0010] 进行浏览器所支持的离线缓存方式对应的离线缓存预备操作；

[0011] 监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令；

[0012] 当获取到所述离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

[0013] 此外，还有必要提供一种能提高网页程序运行效率的实现网页程序离线缓存的系统。

[0014] 一种实现网页程序离线缓存的系统，包括：

[0015] 检测模块，用于检测浏览器所支持的离线缓存方式；

[0016] 初始化模块，用于进行浏览器所支持的离线缓存方式对应的离线缓存预备操作；

[0017] 指令接收模块，用于监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令；

[0018] 缓存控制模块，用于当获取到所述离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

[0019] 上述实现网页程序离线缓存的方法和系统，在网页程序进行离线缓存之前，检测浏览器所支持的离线缓存方式，并进行对应的离线缓存预备操作，网页程序只需要调用预设接口以触发离线缓存读写指令，即可实现离线缓存读写操作，不需要网页程序兼容各种离线缓存方式，从而降低了网页程序的代码复杂度、提高了网页程序的运行效率。

【附图说明】

[0020] 图 1 为一个实施例中的实现网页程序离线缓存的方法的流程示意图；

[0021] 图 2 为另一实施例中的实现网页程序离线缓存的方法的流程示意图；

[0022] 图 3 为一个实施例中的实现网页程序离线缓存的系统的结构示意图。

【具体实施方式】

[0023] 如图 1 所示，在一个实施例中，一种实现网页程序离线缓存的方法，包括以下步骤：

[0024] 步骤 S101，检测浏览器所支持的离线缓存方式。

[0025] 浏览器打开网页后，步骤 S101 在与浏览器进行交互的后台执行。

[0026] 在一个实施例中，步骤 S101 可按照预设顺序依次检测浏览器是否支持 WebSQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式，当检测到浏览器支持的离线缓存方式时，则停止检测，并设置浏览器所支持的离线缓存方式为当前检测的离线缓存方式，若未检测到浏览器支持的离线缓存方式，则设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。

[0027] 在一个实施例中，步骤 S101 可依次按照 Web SQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式的先后排列顺序检测浏览器所支持的离线缓存方式。

[0028] 本实施例中，优先检测浏览器是否支持 Web SQL Database 离线缓存方式，当浏览器不支持 Web SQL Database 离线缓存方式时，再检测浏览器是否支持 FileSystem 离线缓存方式，当浏览器不支持 File System 离线缓存方式时，再检测浏览器是否支持 indexedDB 离线缓存方式，当浏览器不支持 indexedDB 离线缓存方式时，再设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。由于 Web SQL Database 离线缓存方式目前是浏览器支持最为广泛的技术，且其效率较高、稳定性较好；而 File System 离线缓存方式的效率虽比 Web SQL Database 低，但其可供使用空间最大，另外 indexedDB 离线缓存方式的可供使用空间比 localstorage 离线缓存方式大，localstorage 离线缓存方式则是浏览器支持最为广泛的离线缓存方式。因此本实施例可使得网页程序优先使用效率高、稳定性好、可使用空间大的离线缓存方式进行缓存，从而提高网页程序进行离线缓存的效率、增加其稳定性以及增大其可使用空间。

[0029] 其中，在一个实施例中，检测浏览器是否支持 Web SQL Database 离线缓存方式

的步骤包括：判断与浏览器进行交互的后台中用于打开数据库的系统对象(即 window.openDatabase 对象)是否存在，若存在，则判定浏览器支持 Web SQLDatabase 离线缓存方式，若不存在，则判定浏览器不支持 Web SQL Database 离线缓存方式。在另一个实施例中，在上一实施例的基础上，当 window.openDatabase 对象存在时，进一步的，可调用该 window.openDatabase 对象，若调用成功，则判定浏览器支持 Web SQL Database 离线缓存方式，否则，判定浏览器不支持 Web SQL Database 离线缓存方式。本实施例中，在 window.openDatabase 对象存在时，还要进一步调用该 window.openDatabase 对象，调用成功才判定浏览器支持 Web SQL Database 离线缓存方式，可避免调用 window.openDatabase 对象抛掷异常的情况下网页程序进行离线缓存失败的问题。

[0030] 在一个实施例中，检测浏览器是否支持 File System 离线缓存方式的步骤包括：判断与浏览器进行交互的后台中用于响应文件系统的系统对象(即 window.requestFileSystem 对象)是否存在，若存在，则判定浏览器支持 File System 离线缓存方式，若不存在，则判定浏览器不支持 File System 离线缓存方式。在另一个实施例中，在上一实施例的基础上，当 window.requestFileSystem 对象存在时，进一步的，可调用该 window.requestFileSystem 对象，若调用成功，则判定浏览器支持 File System 离线缓存方式，否则，判定浏览器不支持 File System 离线缓存方式。本实施例中，在 window.requestFileSystem 对象存在时，还要进一步调用该 window.requestFileSystem 对象，调用成功才判定浏览器支持 FileSystem 离线缓存方式，可避免调用 window.requestFileSystem 对象抛掷异常的情况下网页程序进行离线缓存失败的问题。

[0031] 在一个实施例中，检测浏览器是否支持 indexDB 离线缓存方式的步骤包括：判断与浏览器进行交互的后台中的索引数据库系统对象(即 window.Indexeddb 对象)是否存在，若存在，则判定浏览器支持 indexDB 离线缓存方式，若不存在，则判定浏览器不支持 indexDB 离线缓存方式。

[0032] 步骤 S102，进行浏览器所支持的离线缓存方式对应的离线缓存预备操作。在一个实施例中，步骤 S102 包括以下步骤：

[0033] 当浏览器所支持的离线缓存方式为 Web SQL Database 离线缓存方式时，根据浏览器打开的网页对应的域名生成数据库名和表名，检测与浏览器交互的后台中该数据库名指定的 SQL 类型数据库和该表名指定的表是否存在，若数据库不存在，则根据数据库名创建 SQL 类型数据库，若表不存在，则根据表名创建表。进一步的，打开 SQL 类型数据库。

[0034] 当浏览器所支持的离线缓存方式为 File System 离线缓存方式时，判断用于缓存的文件是否存在，若不存在则生成预设数量的用于缓存的文件；进一步的，打开用于缓存的文件，读取用于缓存的文件到内存中。在一个实施例中，判断用于缓存的文件是否存在的步骤包括：根据浏览器打开的网页对应的域名获取用于缓存的文件的文件名和文件路径，并根据文件名和文件路径判断文件是否存在。生成预设数量的用于缓存的文件的步骤包括：根据域名获取预设数量的用于缓存的文件的文件路径和文件名，根据文件路径和文件名创建文件。其中，预设数量大于 1。本实施例生成多个用于缓存的文件，网页程序在进行离线缓存时可将数据存储于多个文件中，从而可降低每个文件的数据存储量，提高在文件中的读写速度。而且，本实施例读取用于缓存的文件到内存中，离线缓存时网页程序可直接在内存中读写数据，从而大大提高了数据读写效率。

[0035] 当浏览器所支持的离线缓存方式为 indexDB 离线缓存方式时，则根据浏览器打开的网页对应的域名生成用于缓存的容器的名称，并调用 indexDB 的相关函数打开用于缓存的容器。

[0036] 一般的浏览器都支持 localstorage 离线缓存方式，浏览器系统在浏览器打开后都将进行 localstorage 离线缓存方式对应的相关的离线缓存预备操作，因此，当步骤 S101 设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式时，步骤 S102 可直接转入到步骤 S103。

[0037] 步骤 S103，监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令。

[0038] 在一个实施例中，预设接口的传入参数包括网页程序将要进行离线缓存读写操作的数据的关键字，通过调用预设接口触发的离线缓存读写指令中包含从预设接口传入的关键字。

[0039] 在一个实施例中，预设接口包括存储缓存数据接口、读取缓存数据接口、删除缓存数据接口和清除所有缓存记录接口。在一个实施例中，存储缓存数据接口传入的参数包括待缓存数据以及待缓存数据的关键字，存储缓存数据接口被设置为 cache.setItem(key,value,suc,err)，其中，key 为待缓存数据的关键字，其数据类型为字符串类型，value 为待缓存数据，其数据类型也为字符串类型，suc 为存储成功的回调函数，err 为存储失败的回调函数。读取缓存数据接口传入的参数包括待读取数据的关键字，读取缓存数据接口被设置为 cache.getItem(key,suc,err)，其中，key 为需要读取的缓存数据的关键字，其数据类型为字符串类型，suc 为读取成功的回调函数，err 为读取失败的回调函数。删除缓存数据接口传入的参数包括待删除数据的关键字，删除缓存数据接口被设置为 cache.removeItem(key,suc,err)，其中，key 为需要删除的缓存数据的关键字，其数据类型为字符串类型，suc 为删除成功的回调函数，err 为删除失败的回调函数。清除所有缓存记录接口为 cache.clear(suc,err)，其中，suc 为清除成功的回调函数，err 为清除失败的回调函数。

[0040] 本实施例将缓存数据操作预先设置为统一的接口，网页应用程序需要进行缓存数据操作时，只需要调用统一的预设接口即可。

[0041] 步骤 S104，当获取到离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

[0042] 根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作包括以下步骤：

[0043] 当浏览器所支持的离线缓存方式为 Web SQL Database 离线缓存方式时，若网页程序调用存储缓存数据接口，则将存储缓存数据接口传入的待缓存数据写入到 Web SQL Database 中，该 Web SQL Database 与浏览器打开的网页的域名对应，在 Web SQL Database 中的写入位置与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则从 Web SQL Database 中读取数据，读取的数据在 Web SQL Database 中的位置与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则从 Web SQL Database 中删除数据，删除的数据在 Web SQL Database 中的位置与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则清空 Web SQL Database 中的所

有缓存数据。

[0044] 当浏览器所支持的离线缓存方式为 File System 离线缓存方式时，根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在文件中进行与离线缓存读写指令相对应的操作。具体的，若网页程序调用存储缓存数据接口，则根据存储缓存数据接口传入的关键字进行哈希映射，得到对应的文件，进一步在对应的文件中写入从存储缓存数据接口传入的待缓存数据，并将待缓存数据写入到内存中该文件对应位置。若网页程序调用读取缓存数据接口，根据读取缓存数据接口传入的关键字进行哈希映射，得到对应的文件，在内存中该文件对应位置读取数据。若网页程序调用删除缓存数据接口，则根据删除缓存数据接口传入的关键字进行哈希映射，得到对应的文件，进一步删除该文件中的数据，并在内存中该文件对应位置删除内存文件中的数据。若网页程序调用清除所有缓存记录接口，则清空文件系统中用于缓存的所有文件中的数据，并清空内存中所有的缓存数据。

[0045] 本实施例根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在文件中进行与离线缓存读写指令相对应的操作，可快速的获取到需要进行操作的文件，而且本实施例中，同时将待缓存数据写入内存，读取数据时直接在内存中读取，从而可极大地提高数据读取效率。

[0046] 当浏览器所支持的离线缓存方式为 indexedDB 离线缓存方式时，若网页程序调用存储缓存数据接口，则将存储缓存数据接口传入的待缓存数据写入到 indexedDB 的容器中，该 indexedDB 与浏览器打开的网页的域名对应，在写入数据的容器与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则从 indexedDB 的容器中读取数据，读取数据的容器与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则从 indexedDB 的容器中删除数据，删除数据的容器与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则清空 indexedDB 的容器中所有缓存数据。

[0047] 当浏览器所支持的离线缓存方式为 localstorage 离线缓存方式时，若网页程序调用存储缓存数据接口，则将存储缓存数据接口传入的待缓存数据写入到浏览器默认的存储空间中，写入数据的位置与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则从浏览器默认的存储空间中读取数据，读取数据的位置与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则从浏览器默认的存储空间中删除数据，删除的数据的位置与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则清空浏览器默认存储空间中所有缓存数据。

[0048] 图 2 为上文所述实现网页程序离线缓存的方法的部分实施例组合而成的一个实施例的流程示意图。如图 2 所示，在一个实施例中，一种实现网页程序离线缓存的方法包括以下步骤：

[0049] 步骤 S201，检测浏览器是否支持 Web SQL Database 离线缓存方式，若是，则执行步骤 S202，若否，则执行步骤 S203。

[0050] 步骤 S202，设置浏览器所支持的离线缓存方式为 Web SQL Database 离线缓存方式；并根据浏览器打开的网页对应的域名生成数据库名和表名，检测与浏览器交互的后台中该数据库名指定的 SQL 类型数据库和该表名指定的表是否存在，若数据库不存在，则根

据数据库名创建 SQL 类型数据库，若表不存在，则根据表名创建表，进一步打开 SQL 类型数据库。

[0051] 步骤 S203，检测浏览器是否支持 File System 离线缓存方式，若是，则执行步骤 S204，若否，则执行步骤 S205。

[0052] 步骤 S204，设置浏览器所支持的离线缓存方式为 File System 离线缓存方式；并判断用于缓存的文件是否存在，若不存在则生成预设数量的用于缓存的文件；进一步的，打开用于缓存的文件，读取用于缓存的文件到内存中。

[0053] 步骤 S205，检测浏览器是否支持 indexedDB 离线缓存方式，若是，则执行步骤 S206，若否，则执行步骤 S207。

[0054] 步骤 S206，设置浏览器所支持的离线缓存方式为 indexedDB 离线缓存方式；并根据浏览器打开的网页对应的域名生成用于缓存的容器的名称，并调用 indexDB 的相关函数打开用于缓存的容器。

[0055] 步骤 S207，设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。

[0056] 步骤 S208，监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令。

[0057] 步骤 S209，当获取到离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

[0058] 如图 3 所示，在一个实施例中，一种实现网页程序离线缓存的系统，包括检测模块 10、初始化模块 20、指令接收模块 30 和缓存控制模块 40，其中：

[0059] 检测模块 10 用于检测浏览器所支持的离线缓存方式。

[0060] 浏览器打开网页后，实现网页程序离线缓存的系统在与浏览器进行交互的后台运行。

[0061] 在一个实施例中，检测模块 10 用于按照预设顺序依次检测浏览器是否支持 Web SQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式，当检测到浏览器支持的离线缓存方式时，则停止检测，并设置浏览器所支持的离线缓存方式为当前检测的离线缓存方式，若未检测到浏览器支持的离线缓存方式，则设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。

[0062] 在一个实施例中，检测模块 10 用于依次按照 Web SQL Database 离线缓存方式、File System 离线缓存方式和 indexedDB 离线缓存方式的先后排列顺序检测浏览器所支持的离线缓存方式。

[0063] 本实施例中，优先检测浏览器是否支持 Web SQL Database 离线缓存方式，当浏览器不支持 Web SQL Database 离线缓存方式时，再检测浏览器是否支持 FileSystem 离线缓存方式，当浏览器不支持 File System 离线缓存方式时，再检测浏览器是否支持 indexedDB 离线缓存方式，当浏览器不支持 indexedDB 离线缓存方式时，再设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式。由于 Web SQL Database 离线缓存方式目前是浏览器支持最为广泛的技术，且其效率较高、稳定性较好；而 File System 离线缓存方式的效率虽比 Web SQL Database 低，但其可供使用空间最大，另外 indexedDB 离线缓存方式的可供使用空间比 localstorage 离线缓存方式大，localstorage 离线缓存方式则是浏览器支持最为广泛的离线缓存方式。因此本实施例可使得网页程序优先使用效率高、稳定性好、可

使用空间大的离线缓存方式进行缓存，从而提高网页程序进行离线缓存的效率、增加其稳定性以及增大其可使用空间。

[0064] 其中，在一个实施例中，检测模块 10 检测浏览器是否支持 Web SQL Database 离线缓存方式的过程为：判断与浏览器进行交互的后台中用于打开数据库的系统对象(即 window.openDatabase 对象)是否存在，若存在，则判定浏览器支持 Web SQL Database 离线缓存方式，若不存在，则判定浏览器不支持 Web SQLDatabase 离线缓存方式。在另一个实施例中，在上一实施例的基础上，当 window.openDatabase 对象存在时，进一步的，检测模块 10 调用该 window.openDatabase 对象，若调用成功，则判定浏览器支持 Web SQL Database 离线缓存方式，否则，判定浏览器不支持 Web SQL Database 离线缓存方式。本实施例中，在 window.openDatabase 对象存在时，还要进一步调用该 window.openDatabase 对象，调用成功才判定浏览器支持 Web SQL Database 离线缓存方式，可避免调用 window.openDatabase 对象抛掷异常的情况下网页程序进行离线缓存失败的问题。

[0065] 在一个实施例中，检测模块 10 检测浏览器是否支持 File System 离线缓存方式的过程为：判断与浏览器进行交互的后台中用于响应文件系统的系统对象(即 window.requestFileSystem 对象) 是否存在，若存在，则判定浏览器支持 File System 离线缓存方式，若不存在，则判定浏览器不支持 File System 离线缓存方式。在另一个实施例中，在上一实施例的基础上，当 window.requestFileSystem 对象存在时，进一步的，检测模块 10 调用该 window.requestFileSystem 对象，若调用成功，则判定浏览器支持 File System 离线缓存方式，否则，判定浏览器不支持 FileSystem 离线缓存方式。本实施例中，在 window.requestFileSystem 对象存在时，还要进一步调用该 window.requestFileSystem 对象，调用成功才判定浏览器支持 File System 离线缓存方式，可避免调用 window.requestFileSystem 对象抛掷异常的情况下网页程序进行离线缓存失败的问题。

[0066] 在一个实施例中，检测模块 10 检测浏览器是否支持 indexDB 离线缓存方式的过程为：判断与浏览器进行交互的后台中的索引数据库系统对象(即 window.Indexeddb 对象)是否存在，若存在，则判定浏览器支持 indexDB 离线缓存方式，若不存在，则判定浏览器不支持 indexDB 离线缓存方式。

[0067] 初始化模块 20 用于进行浏览器所支持的离线缓存方式对应的离线缓存预备操作。

[0068] 在一个实施例中，初始化模块 20 用于当浏览器所支持的离线缓存方式为 Web SQL Database 离线缓存方式时，根据浏览器打开的网页对应的域名生成数据库名和表名，检测与浏览器交互的后台中该数据库名指定的 SQL 类型数据库和该表名指定的表是否存在，若数据库不存在，则根据数据库名创建 SQL 类型数据库，若表不存在，则根据表名创建表。进一步的，初始化模块 20 用于打开 SQL 类型数据库。

[0069] 初始化模块 20 还用于当浏览器所支持的离线缓存方式为 File System 离线缓存方式时，判断用于缓存的文件是否存在，若不存在则生成预设数量的用于缓存的文件；进一步的，打开用于缓存的文件，读取用于缓存的文件到内存中。在一个实施例中，初始化模块 20 判断用于缓存的文件是否存在的过程为：根据浏览器打开的网页对应的域名获取用于缓存的文件的文件名和文件路径，并根据文件名和文件路径判断文件是否存在。初始化模块 20 生成预设数量的用于缓存的文件的过程为：根据域名获取预设数量的用于缓存的文

件的文件路径和文件名，根据文件路径和文件名创建文件。其中，预设数量大于 1。本实施例生成多个用于缓存的文件，网页程序在进行离线缓存时可将数据存储于多个文件中，从而可降低每个文件的数据存储量，提高在文件中的读写速度。而且，本实施例读取用于缓存的文件到内存中，离线缓存时网页程序可直接在内存中读写数据，从而大大提高了数据读写效率。

[0070] 初始化模块 20 还用于当浏览器所支持的离线缓存方式为 indexDB 离线缓存方式时，则根据浏览器打开的网页对应的域名生成用于缓存的容器的名称，并调用 indexDB 的相关函数打开用于缓存的容器。

[0071] 一般的浏览器都支持 localstorage 离线缓存方式，浏览器系统在浏览器打开后都将进行 localstorage 离线缓存方式对应的相关的离线缓存预备操作，因此，当检测模块 10 设置浏览器所支持的离线缓存方式为 localstorage 离线缓存方式时，初始化模块 20 可直接调用指令接收模块 30。

[0072] 指令接收模块 30 用于监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令。

[0073] 在一个实施例中，预设接口的传入参数包括网页程序将要进行离线缓存读写操作的数据的关键字，通过调用预设接口触发的离线缓存读写指令中包含从预设接口传入的关键字。

[0074] 在一个实施例中，预设接口包括存储缓存数据接口、读取缓存数据接口、删除缓存数据接口和清除所有缓存记录接口。在一个实施例中，存储缓存数据接口传入的参数包括待缓存数据以及待缓存数据的关键字，存储缓存数据接口被设置为 cache.setItem(key,value,suc,err)，其中，key 为待缓存数据的关键字，其数据类型为字符串类型，value 为待缓存数据，其数据类型也为字符串类型，suc 为存储成功的回调函数，err 为存储失败的回调函数。读取缓存数据接口传入的参数包括待读取数据的关键字，读取缓存数据接口被设置为 cache.getItem(key,suc,err)，其中，key 为需要读取的缓存数据的关键字，其数据类型为字符串类型，suc 为读取成功的回调函数，err 为读取失败的回调函数。删除缓存数据接口传入的参数包括待删除数据的关键字，删除缓存数据接口被设置为 cache.removeItem(key,suc,err)，其中，key 为需要删除的缓存数据的关键字，其数据类型为字符串类型，suc 为删除成功的回调函数，err 为删除失败的回调函数。清除所有缓存记录接口为 cache.clear(suc,err)，其中，suc 为清除成功的回调函数，err 为清除失败的回调函数。

[0075] 本实施例将缓存数据操作预先设置为统一的接口，网页应用程序需要进行缓存数据操作时，只需要调用统一的预设接口即可。

[0076] 缓存控制模块 40 用于当获取到离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作。

[0077] 缓存控制模块 40 根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作的过程包括：

[0078] 当浏览器所支持的离线缓存方式为 Web SQL Database 离线缓存方式时，若网页程序调用存储缓存数据接口，则缓存控制模块 40 将存储缓存数据接口传入的待缓存数据写入到 Web SQL Database 中，该 Web SQL Database 与浏览器打开的网页的域名对应，在

Web SQL Database中的写入位置与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则缓存控制模块40从Web SQL Database中读取数据，读取的数据在Web SQL Database中的位置与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则缓存控制模块40从Web SQL Database中删除数据，删除的数据在Web SQLDatabase中的位置与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则缓存控制模块40清空Web SQL Database中的所有缓存数据。

[0079] 当浏览器所支持的离线缓存方式为File System离线缓存方式时，缓存控制模块40根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在文件中进行与离线缓存读写指令相对应的操作。具体的，若网页程序调用存储缓存数据接口，则缓存控制模块40根据存储缓存数据接口传入的关键字进行哈希映射，得到对应的文件，进一步在对应的文件中写入从存储缓存数据接口传入的待缓存数据，并将待缓存数据写入到内存中该文件对应位置。若网页程序调用读取缓存数据接口，则缓存控制模块40根据读取缓存数据接口传入的关键字进行哈希映射，得到对应的文件，在内存中该文件对应位置读取数据。若网页程序调用删除缓存数据接口，则缓存控制模块40根据删除缓存数据接口传入的关键字进行哈希映射，得到对应的文件，进一步删除该文件中的数据，并在内存中该文件对应位置删除内存文件中的数据。若网页程序调用清除所有缓存记录接口，则缓存控制模块40清空文件系统中用于缓存的所有文件中的数据，并清空内存中所有的缓存数据。

[0080] 本实施例根据从预设接口传入的关键字进行哈希映射，得到关键字对应的用于缓存的文件，并在文件中进行与离线缓存读写指令相对应的操作，可快速的获取到需要进行操作的文件，而且本实施例中，同时将待缓存数据写入内存，读取数据时直接在内存中读取，从而可极大地提高数据读取效率。

[0081] 当浏览器所支持的离线缓存方式为indexedDB离线缓存方式时，若网页程序调用存储缓存数据接口，则缓存控制模块40将存储缓存数据接口传入的待缓存数据写入到indexedDB的容器中，该indexedDB与浏览器打开的网页的域名对应，在写入数据的容器与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则缓存控制模块40从indexedDB的容器中读取数据，读取数据的容器与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则缓存控制模块40从indexedDB的容器中删除数据，删除数据的容器与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则缓存控制模块40清空indexedDB的容器中所有缓存数据。

[0082] 当浏览器所支持的离线缓存方式为localstorage离线缓存方式时，若网页程序调用存储缓存数据接口，则缓存控制模块40将存储缓存数据接口传入的待缓存数据写入到浏览器默认的存储空间中，写入数据的位置与从存储缓存数据接口传入的关键字对应；若网页程序调用读取缓存数据接口，则缓存控制模块40从浏览器默认的存储空间中读取数据，读取数据的位置与读取缓存数据接口传入的关键字对应；若网页程序调用删除缓存数据接口，则缓存控制模块40从浏览器默认的存储空间中删除数据，删除的数据的位置与删除缓存数据接口传入的关键字对应；若网页程序调用清除所有缓存记录接口，则缓存控制模块40清空浏览器默认存储空间中所有缓存数据。

[0083] 上述实现网页程序离线缓存的方法和系统，在网页程序进行离线缓存之前，检测

浏览器所支持的离线缓存方式，并进行浏览器所支持的离线缓存方式对应的离线缓存预备操作，网页程序只需要调用预设接口以触发离线缓存读写指令，即可实现离线缓存读写操作，不需要网页程序兼容各种离线缓存方式，从而降低了网页程序的代码复杂度、提高了网页程序的运行效率。

[0084] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序控制相关的硬件来完成的，所述的程序可存储于一计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，所述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory，ROM）或随机存储记忆体（Random Access Memory，RAM）等。

[0085] 以上所述实施例仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明专利的保护范围应以所附权利要求为准。

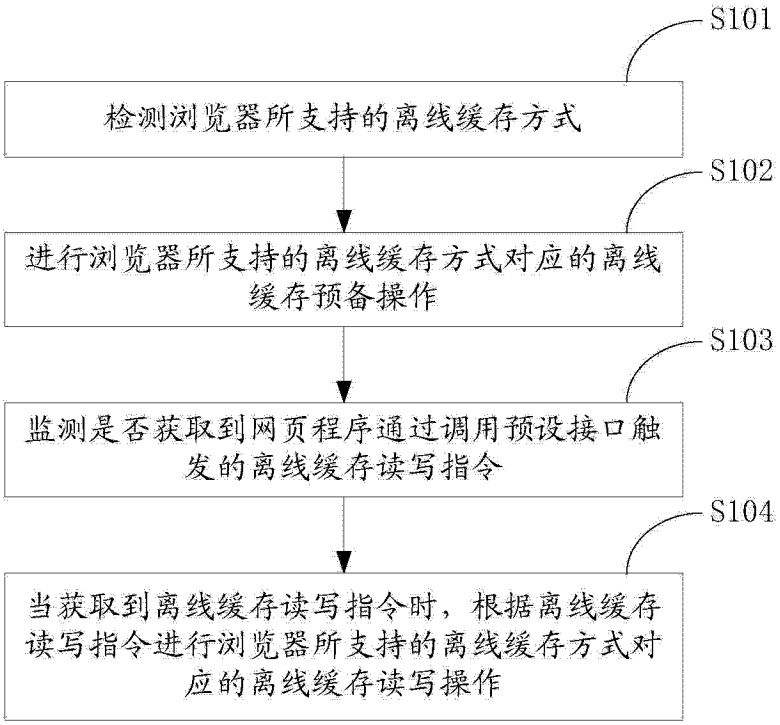

S101
检测浏览器所支持的离线缓存方式
S102
进行浏览器所支持的离线缓存方式对应的离线缓存预备操作
S103
监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令
S104
当获取到离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作

图 1

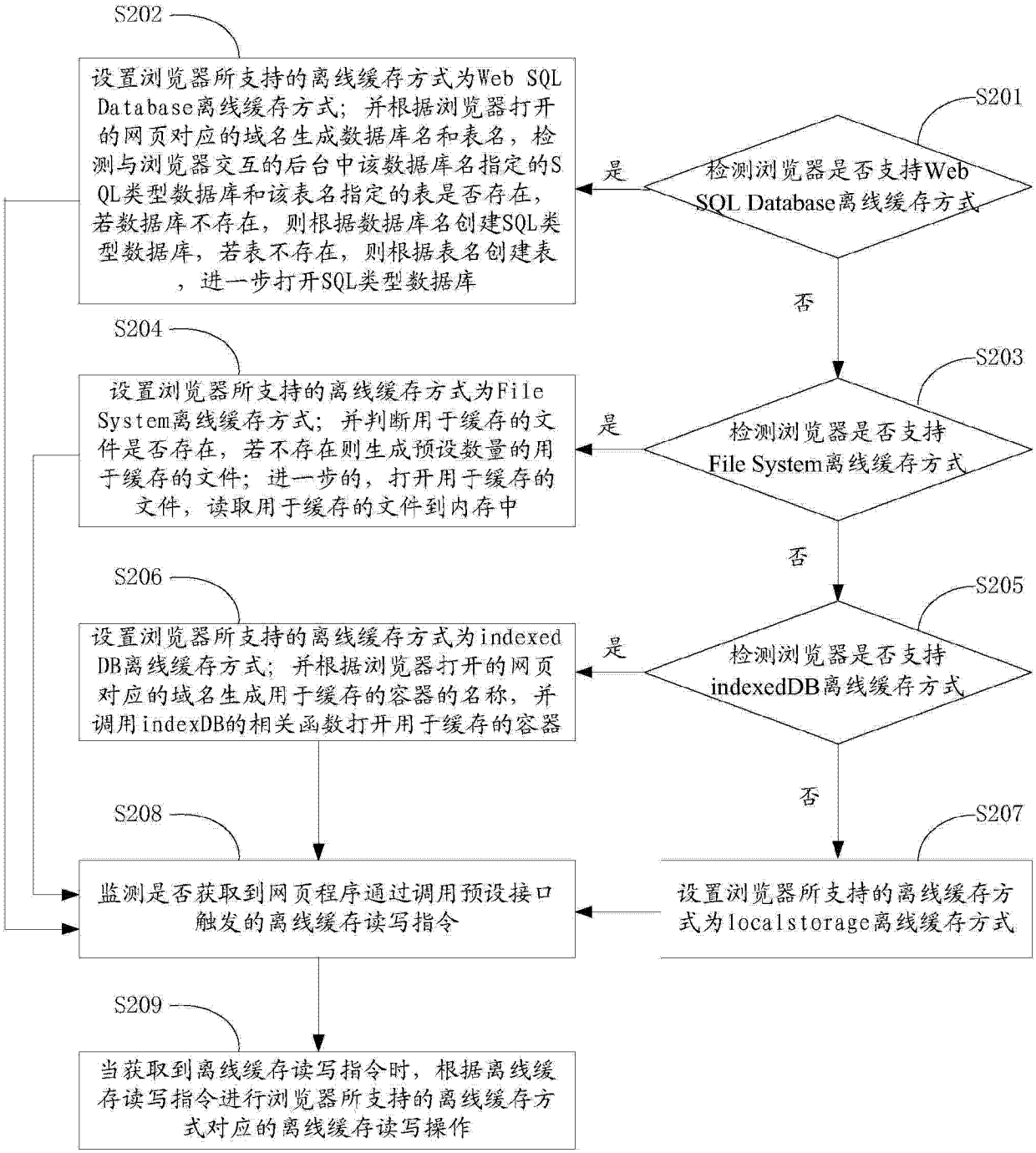

S201
检测浏览器是否支持Web SQL Database离线缓存方式
是
否
S202
设置浏览器所支持的离线缓存方式为Web SQL Database离线缓存方式；并根据浏览器打开的网页对应的域名生成数据库名和表名，检测与浏览器交互的后台中该数据库名指定的SQL类型数据库和该表名指定的表是否存在，若数据库不存在，则根据数据库名创建SQL类型数据库，若表不存在，则根据表名创建表，进一步打开SQL类型数据库
S203
检测浏览器是否支持File System离线缓存方式
是
否
S204
设置浏览器所支持的离线缓存方式为File System离线缓存方式；并判断用于缓存的文件是否存在，若不存在则生成预设数量的用于缓存的文件；进一步的，打开用于缓存的文件，读取用于缓存的文件到内存中
S205
检测浏览器是否支持indexedDB离线缓存方式
是
否
S206
设置浏览器所支持的离线缓存方式为indexed DB离线缓存方式；并根据浏览器打开的网页对应的域名生成用于缓存的容器的名称，并调用indexDB的相关函数打开用于缓存的容器
S207
设置浏览器所支持的离线缓存方式为localstorage离线缓存方式
S208
监测是否获取到网页程序通过调用预设接口触发的离线缓存读写指令
S209
当获取到离线缓存读写指令时，根据离线缓存读写指令进行浏览器所支持的离线缓存方式对应的离线缓存读写操作

图 2

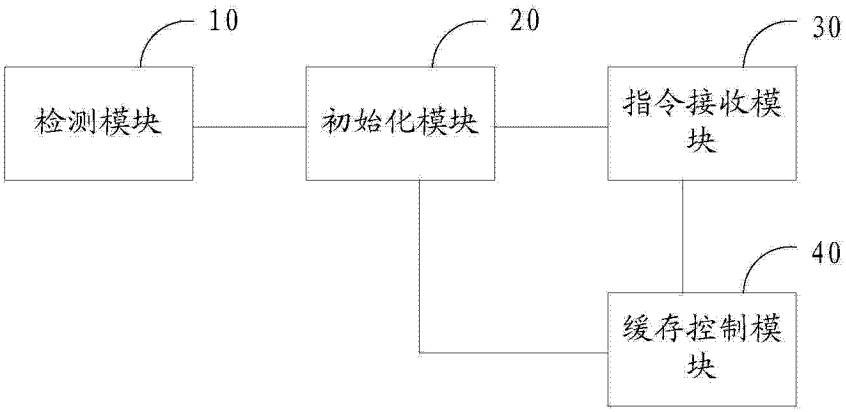


图 3