(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 104765786 A
(43)申请公布日 2015.07.08

(21)申请号 201510128329.0

(22)申请日 2015.03.24

(66)本国优先权数据
201420573239.3 2014.09.30 CN

(71)申请人 贵阳朗玛信息技术股份有限公司
地址 550022 贵州省贵阳市高新区金阳科技产业园创业大厦130室

(72)发明人 国兴旺 段凌云

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

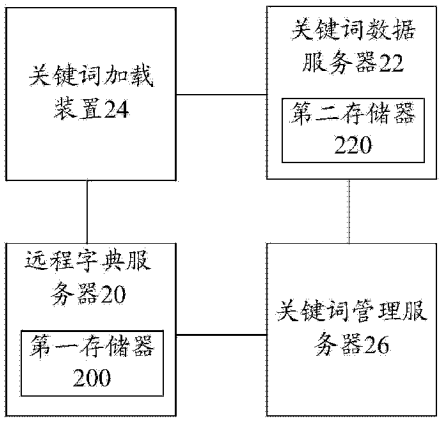
权利要求书2页 说明书6页 附图5页

(54)发明名称

关键词过滤系统及其应用方法

(57)摘要

本发明公开一种关键词过滤系统及其应用方法，上述系统包括：远程字典服务器内部设置有存储关键词算法库的第一存储器；关键词数据服务器，内部设置有存储关键词数据库的第二存储器；关键词加载装置，分别与所述远程字典服务器和所述关键词数据服务器相耦合；以及关键词管理服务器，分别与所述远程字典服务器和所述关键词数据服务器相耦合。根据本发明提供的技术方案，解决了相关技术中基于软件内存进行关键词过滤时，当内存管理不当会造成内存泄漏，从而导致一定风险的问题，从而有效提高了可靠性。



CN 104765786 A

1. 一种关键词过滤系统，其特征在于，包括：

远程字典服务器，内部设置有存储关键词算法库的第一存储器；

关键词数据服务器，内部设置有存储关键词数据库的第二存储器；

关键词加载装置，分别与所述远程字典服务器和所述关键词数据服务器相耦合；以及

关键词管理服务器，分别与所述远程字典服务器和所述关键词数据服务器相耦合。

2. 根据权利要求 1 所述的关键词过滤系统，其特征在于，所述远程字典服务器和所述关键词加载装置设置有一组或多组。

3. 根据权利要求 1 所述的关键词过滤系统，其特征在于，

所述远程字典服务器与所述关键词加载装置通过 unix 域协议或者传输控制协议耦合；

所述关键词数据服务器和所述关键词加载装置通过分布式结构化查询语言 mysql 数据库使用的协议耦合；

所述关键词管理服务器与所述远程字典服务器通过传输控制协议耦合，所述关键词管理服务器与所述关键词数据服务器通过分布式 mysql 数据库使用的协议耦合。

4. 根据权利要求 1 所述的关键词过滤系统，其特征在于，所述关键词过滤系统还包括：应用服务器和 / 或网页服务器，其中，所述应用服务器和 / 或网页服务器，与所述远程字典服务器相耦合。

5. 根据权利要求 4 所述的关键词过滤系统，其特征在于，所述关键词过滤系统包括应用服务器时，所述应用服务器与所述远程字典服务器通过 unix 域协议或者传输控制协议耦合。

6. 根据权利要求 4 所述的关键词过滤系统，其特征在于，所述关键词过滤系统包括网页服务器时，所述网页服务器与所述远程字典服务器通过传输控制协议耦合。

7. 一种如权利要求 1 至 6 中任一项所述关键词过滤系统的应用方法，其特征在于，包括：

远程字典服务器启动后运行关键词加载装置，以使的所述关键词加载装置将关键词数据服务器中的初始关键词信息以循环的方式通过命令加载到所述远程字典服务器的内存中；

所述远程字典服务器接收来自于应用服务器或者网页服务器的需要过滤的内容；

所述远程字典服务器执行过滤算法，对需要过滤的内容进行过滤并对于预定时间内的过滤内容进行统计，形成统计数据。

8. 根据权利要求 7 所述的应用方法，其特征在于，还包括：

所述关键词管理服务器在维护所述关键词数据服务器的同时，将所述关键词数据服务器中的数据通过命令实时同步到所述远程字典服务器中。

9. 根据权利要求 7 所述的应用方法，其特征在于，在所述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还包括：

所述远程字典服务器将过滤后的结果反馈给所述应用服务器或者所述网页服务器。

10. 根据权利要求 7 所述的应用方法，其特征在于，在所述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还包括：

所述关键词管理服务器通过命令向所述远程字典服务器查询统计数据；

所述关键词管理服务器接收来自于所述远程字典服务器的统计数据；

所述关键词管理服务器获取所述统计数据后形成统计报表，以便于定期维护和重新设定关键词。

关键词过滤系统及其应用方法

技术领域

[0001] 本发明涉及通信领域，具体而言，涉及一种关键词过滤系统及其应用方法。

背景技术

[0002] 随着移动智能终端设备的标准化和普及，移动互联网得到了迅猛的发展，同时在移动互联网中出现了大量的应用软件，例如，腾讯微信，新浪微博等。在以内容为中心的软件中，存在大量的审核需求，只靠人工审核这些海量内容显然无法满足业务发展的需要，所以一般的软件运营商会通过系统自动过滤再进行人工确认的方式进行二次审核，运营人员会搜集一些敏感词或高危词（以下统称为关键词）形成词库并更新到业务系统中，发布的内容中如果遇到这些词或相似的词就会触发审核警告。

[0003] 通常会将关键词库更新到具体的软件内存中，这样有利于提高过滤效率，但是大多软件系统后台属于一种分布式结构，有很多服务器或集群组成，具体可以参见图 1。图 1 示出了关键词过滤算法的传统部署结构，可以部署多组网页服务器 (WebServer) 和应用服务器 (AppServer)，图 1 示出了一组 WebServer 和 AppServer，其中 Libcensor 关键词算法库需要基于 WebServer 和 AppServer 实现两套结构，发布关键词时需要同步到 WebServer 和 AppServer。

[0004] 将这些关键词同时同步到多台服务器本身就存在一定的风险，在服务器性能和可维护性上很好能达到一个相对的平衡。具体体现在如下几点：

[0005] 1、关键词同步到多个系统环境存在可靠性问题，不能满足统一发布关键词的需求；

[0006] 2、过滤算法基于内存进行可以提高工作效率，但会由于内存管理不当造成内存泄漏；

[0007] 3、可能会涉及到服务器进程的重新启动，影响正常业务；

[0008] 4、一般需要将关键词算法实现为多种开发语言，多产品与多语言支持存在问题，不利于横向扩展。

发明内容

[0009] 本发明的主要目的在于公开了一种关键词过滤系统及其应用方法，以至少解决相关技术中基于软件内存进行关键词过滤时，当内存管理不当会造成内存泄漏，从而导致一定风险的问题。

[0010] 根据本发明的一个方面，提供了一种关键词过滤系统。

[0011] 根据本发明的关键词过滤系统包括：远程字典服务器 (remote dictonary server，简称为 Redis) 内部设置有存储关键词算法库的第一存储器；关键词数据服务器，内部设置有存储关键词数据库的第二存储器；关键词加载装置，分别与上述远程字典服务器和上述关键词数据服务器相耦合；以及关键词管理服务器，分别与上述远程字典服务器和上述关键词数据服务器相耦合。

[0012] 优选地，上述远程字典服务器和上述关键词加载装置设置有一组或多组。

[0013] 优选地，上述远程字典服务器与上述关键词加载装置通过 unix 域协议或者传输控制协议耦合；上述关键词数据服务器和上述关键词加载装置通过分布式结构化查询语言 mysql 数据库使用的协议耦合；上述关键词管理服务器与上述远程字典服务器通过传输控制协议耦合，上述关键词管理服务器与上述关键词数据服务器通过分布式 mysql 数据库使用的协议耦合。

[0014] 优选地，上述关键词过滤系统还包括：应用服务器和 / 或网页服务器，其中，上述应用服务器和 / 或网页服务器，与上述远程字典服务器相耦合。

[0015] 优选地，上述关键词过滤系统包括应用服务器时，上述应用服务器与上述远程字典服务器通过 unix 域协议或者传输控制协议耦合。

[0016] 优选地，上述关键词过滤系统包括网页服务器时，上述网页服务器与上述远程字典服务器通过传输控制协议耦合。

[0017] 根据本发明的另一方面，提供了一种关键词过滤系统的应用方法。

[0018] 根据本发明的关键词过滤系统的应用方法包括：远程字典服务器启动后运行关键词加载装置，以使的所述关键词加载装置将关键词数据服务器中的初始关键词信息以循环的方式通过命令加载到所述远程字典服务器的内存中；所述远程字典服务器接收来自于应用服务器或者网页服务器的需要过滤的内容；所述远程字典服务器执行过滤算法，对需要过滤的内容进行过滤并对于预定时间内的过滤内容进行统计，形成统计数据。

[0019] 优选地，上述方法还包括：所述关键词管理服务器在维护所述关键词数据服务器的同时，将所述关键词数据服务器中的数据通过命令实时同步到所述远程字典服务器中。

[0020] 优选地，在所述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还包括：所述远程字典服务器将过滤后的结果反馈给所述应用服务器或者所述网页服务器。

[0021] 优选地，在所述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还包括：所述关键词管理服务器通过命令向所述远程字典服务器查询统计数据；所述关键词管理服务器接收来自于所述远程字典服务器的统计数据；所述关键词管理服务器获取所述统计数据后形成统计报表，以便于定期维护和重新设定关键词。

[0022] 与现有技术相比，本发明实施例至少具有以下优点：使用集成有关键词算法库的远程字典服务器（以下称为 Redis-censor-server）构建关键词过滤系统，Redis 的协议属于文本协议，该协议本身不支持异步 Pipeline（异步 Pipeline 一般通过消息的 SeqID 实现），解决了相关技术中基于软件内存进行关键词过滤时，当内存管理不当会造成内存泄漏，从而导致一定风险的问题，从而有效提高了可靠性。

附图说明

[0023] 图 1 是相关技术中关键词过滤系统的结构示意图；

[0024] 图 2 是根据本发明实施例的关键词过滤系统的结构框图；

[0025] 图 3 是根据本发明优选实施例的关键词过滤系统的结构示意图；

[0026] 图 4 是根据本发明优选实施例的关键词过滤系统的应用方法的流程图。以及

[0027] 图 5 是根据本发明优选实施例的关键词过滤系统的应用方法的信息交互流程图。

具体实施方式

[0028] 下面结合说明书附图对本发明的具体实现方式做一详细描述。

[0029] 图 2 是根据本发明实施例的关键词过滤系统的结构框图。如图 2 所示，该关键词过滤系统包括：远程字典服务器 20(Redis-censor-server)，内部设置有存储关键词算法库的第一存储器 200；关键词数据服务器 (Censor-database)22，内部设置有存储关键词数据库的第二存储器 220；关键词加载装置 (Censor-loader)24，分别与上述远程字典服务器和上述关键词数据服务器 22 相耦合；以及关键词管理服务器 (Censor-admin)26，分别与上述远程字典服务器 20 和上述关键词数据服务器 22 相耦合。

[0030] 在图 2 所示的系统中，使用集成有关键词算法库 (Libcensor) 的远程字典服务器 (以下称为 Redis-censor-server) 构建关键词过滤系统，Redis 的协议属于文本协议，该协议本身不支持异步 Pipeline(异步 Pipeline 一般通过消息的 SeqID 实现)，解决了相关技术中基于软件内存进行关键词过滤时，内存管理不当造成内存泄漏，导致一定风险的问题，从而有效提高了可靠性。

[0031] 优选地，上述关键词过滤系统中，上述远程字典服务器和上述关键词加载装置可以设置有一组或多组。图 3 示出了部署了 1 组远程字典服务器和关键词加载装置。在具体实施过程中，上述远程字典服务器和上述关键词加载装置可以部署在同一台计算机设备上。

[0032] 在优选实施过程中，上述 Censor-loader 可以是开源的 Redis 服务器的 Java 语言客户端，Censor-loader 在 Redis-censor-server 启动后会将关键词从数据库 Censor-database 逐条读取并同步到 Redis-censor-server 的内存中。

[0033] 上述 Censor-admin 可以维护多组 Redis-censor-server 的词库同步；运营人员如果需要删除或添加新的关键词则通过 Censor-admin 进行 Redis-censor-server 和 Censor-database 的同步。

[0034] 优选地，如图 3 所示，上述关键词过滤系统还可以包括：应用服务器 (App Server) 和 / 或网页服务器 (Web Server)，其中，上述应用服务器和 / 或网页服务器，与上述远程字典服务器相耦合。

[0035] 在优选实施过程中，可以在 App Server 上设置 Java 语言客户端，在 Web Server 上设置 PHP 语言客户端，用户通过应用服务器 (App Server) 和 / 或网页服务器 (Web Server) 上设置的客户端组件库对需要过滤的内容采用扩展的命令 (wdetect) 进行过滤。

[0036] 在上述关键词过滤系统中，上述远程字典服务器与上述关键词加载装置通过 unix 域协议 (Unix Domain Socket, 简称为 UDS) 或者传输控制协议 (Transmission Control Protocol，简称为 TCP) 耦合；上述关键词数据服务器和上述关键词加载装置通过分布式 (microsoft structure quest language, 简称为 mysql) 数据库使用的协议耦合；上述关键词管理服务器与上述远程字典服务器通过 TCP 耦合，上述关键词管理服务器与上述关键词数据服务器通过分布式 mysql 数据库使用的协议耦合。

[0037] 上述关键词过滤系统包括 App Server 时，上述 App Server 与上述远程字典服务器通过 unix 域协议 (UDS) 或者传输控制协议耦合。其中，App Server 与 Redis-censor-server 采用 UDS 通信时，需要将 Redis-censor-server 与 App Server 部署

在一台计算机上。

[0038] 上述关键词过滤系统包括 Web Server 时，上述 Web Server 与上述远程字典服务器通过传输控制协议 (TCP) 耦合。

[0039] 以下结合图 3 进一步描述上述优选实施方式。基于现有开源的 Redis 服务器，通过二次开发扩展其命令将关键词算法库 (Libcensor) 集成到 Redis 服务器内部，得到图 3 中所示的 Redis-censor-server。通过 Censor-loader 将关键词数据库 Censor-database 中的数据加载到 Redis-censor-server 服务器中，部署多台 Redis-censor-server 服务器后客户端可通过扩展命令后的 Phpredis，Jedis 和 Hiredis 等常用开发语言客户端进行统一访问。此外可以通过扩展 Redis 命令进行关键词库的删除与增加进行关键词的实时发布，此部分可以通过 Censor-admin 来完成。以下分别对上述各组件进行描述：

[0040] Libcensor：使用标准 c 语言实现的算法库，Libcensor 属于抽象层面的软件库，可以采用相关技术中实现基本过滤功能的算法库充当 Libcensor，此组件也可以由具体的第三方提供，具有可替换性；

[0041] Redis-censor-server：通过二次开发扩展开源的 Redis 服务器而成为本系统的核心服务器，内部集成了关键词过滤算法 Libcensor；

[0042] Censor-database：基于 mysql 的关键词数据库，可以是一个或多个关键词数据表，该关键词数据表的内容具体可以参见表 1。

[0043] 表 1

[0044]

GroupID	WordText	WordType	IsPY
20140722	测试 1	1	0
20140722	测试 2	0	0
20140723	ceshi3	1	1
…	…	…	…

[0045] 在表 1 中，GroupID 为关键词分组号，WordText 为关键词，WordType 标识出关键词的类型 (0- 敏感词，1- 高危词)，IsPY 指示关键词内容是否是拼音。

[0046] Censor-loader：使用 Java 开发语言实现，主要作用为从 Censor-database 加载关键词来初始化 Redis-censor-server；

[0047] Censor-admin：使用 PHP 语言实现的管理站点，用于关键词库的维护与关键词命中情况的统计；

[0048] Phpredis：开源的 Redis 服务器的 PHP 语言客户端，需要扩展命令；

[0049] Jedis：开源的 Redis 服务器的 Java 语言客户端，需要扩展命令；

[0050] Hiredis：开源的 Redis 服务器的 C/C++ 语言客户端，需要扩展命令；

[0051] 在具体实施过程中，可以根据系统规模部署多台高性能的 Linux 服务器作为关键词过滤的算法核心服务器使用，通过扩展 Redis 的各个客户端来提供多产品 / 多语言的支持。

[0052] 对于 Redis 系列组件扩展的主要命令有：

[0053] wadd :用于增加关键词；

[0054] wremove :用于移除某个关键词；

[0055] wclear :全部移除关键词库；

[0056] wdetect :用于对文本进行关键词过滤；

[0057] wstat :用于统计关键词命中情况，以便运营人员实时设定词库。

[0058] 图 4 是根据本发明优选实施例的关键词过滤系统的应用方法的流程图。如图 4 所示，该流程主要包括以下处理：

[0059] 步骤 S401 :远程字典服务器启动后运行关键词加载装置，以使的上述关键词加载装置将关键词数据服务器中的初始关键词信息以循环的方式通过命令加载到上述远程字典服务器的内存中；

[0060] 步骤 S403 :上述远程字典服务器接收来自于应用服务器或者网页服务器的需要过滤的内容；

[0061] 步骤 S405 :上述远程字典服务器执行过滤算法，对需要过滤的内容进行过滤并对于预定时间内的过滤内容进行统计，形成统计数据。

[0062] 在使用集成有关键词算法库 (Libcensor) 的远程字典服务器（以下称为 Redis-censor-server) 构建关键词过滤系统中，远程字典服务器执行关键词过滤，Redis 的协议属于文本协议，该协议本身不支持异步 Pipeline（异步 Pipeline 一般通过消息的 SeqID 实现），解决了相关技术中基于软件内存进行关键词过滤时，内存管理不当造成内存泄漏，导致一定风险的问题，从而有效提高了可靠性。

[0063] 优选地，上述方法还可以包括以下处理 :上述关键词管理服务器在维护上述关键词数据服务器的同时，将上述关键词数据服务器中的数据通过命令实时同步到上述远程字典服务器中。

[0064] 优选地，在上述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还可以包括以下处理 :上述远程字典服务器将过滤后的结果反馈给上述应用服务器或者上述网页服务器。

[0065] 优选地，在上述远程字典服务器对于预定时间内的过滤内容进行统计，形成统计数据之后，还可以包括以下处理 :上述关键词管理服务器通过命令向上述远程字典服务器查询统计数据 ;上述关键词管理服务器接收来自于上述远程字典服务器的统计数据 ;上述关键词管理服务器获取上述统计数据后形成统计报表，以便于定期维护和重新设定关键词。

[0066] 以下结合图 5 进一步描述上述优选实施过程。

[0067] 图 5 是根据本发明优选实施例的关键词过滤系统的应用方法的信息交互流程图。如图 5 所示，该信息交互流程主要包括以下处理：

[0068] 步骤 S501 :关键词过滤系统中的核心服务器远程字典服务器 Redis-censor-server 通过脚本启动后会运行关键词加载装置 Censor-loader，Censor-loader 将关键词数据服务器 Censor-database 中的初始关键词信息以循环的方式通过命令（例如 :wadd 命令）加载到 Redis-censor-server 的内存中。

[0069] 步骤 S503 :运营人员通过 Censor-admin 管理界面增加 / 删除关键词，或清空关键

词，Censor-admin 在维护 Censor-database 的同时通过 wadd，wremove，wclear 等命令实时同步到 Redis-censor-server。

[0070] 步骤 S505：用户通过 WebServer 或者 AppServer 中扩展后的 Redis 客户端组件库将需要过滤的内容发送至 Redis-censor-server。

[0071] 其中，WebServer 与 AppServer 完全对等，只是工程部署角色不同。

[0072] 步骤 S507：核心服务器 Redis-censor-server 执行过滤算法，对需要过滤的内容进行过滤并对于一定时间内的过滤内容进行统计，包括某个关键词命中次数与命中的具体情况，形成统计数据。

[0073] 步骤 S509：核心服务器 Redis-censor-server 将过滤后的结果反馈给 WebServer 或者 AppServer。

[0074] 步骤 S511：Censor-admin 通过命令（例如，wstat 命令）向核心服务器 Redis-censor-server 查询统计数据。

[0075] 步骤 S513：核心服务器 Redis-censor-server 将统计数据发送给 Censor-admin，Censor-admin 获取此统计数据后形成统计报表，以便于运营人员根据统计报表定期维护和重新设定关键词。

[0076] 需要说明的是，步骤 S501 是系统启动步骤，步骤 S503 是系统维护步骤，步骤 S505 至步骤 S509 是系统内容过滤步骤，步骤 S511 至步骤 S513 是系统统计分析步骤。步骤 S503 与其他步骤没有时间执行先后关系，即步骤 S503 可以在其他步骤执行前执行，也可以在其他步骤执行后执行。

[0077] 由此可见，上述系统可为业务系统提供高性能的关键词服务，实现成本较低，便于系统人员或运营人员维护和管理。此外，上述系统还支持多语言和多产品。

[0078] 综上所述，借助本发明提供的上述实施例，使用集成有关键词算法库 (Libcensor) 的远程字典服务器 (Redis-censor-server) 构建关键词过滤系统，Redis 的协议属于文本协议，该协议本身不支持异步 Pipeline（异步 Pipeline 一般通过消息的 SeqID 实现），解决了相关技术中基于软件内存进行关键词过滤时，内存管理不当造成内存泄漏，导致一定风险的问题，从而有效提高了可靠性。在本系统中，即使 Redis-censor-server 宕机，在跳过关键词检查的情形下，业务应用服务器也不会受到影响，因而不会涉及服务器进程的重新启动，保证了业务的正常进行。此外，上述系统还支持多语言和多产品，有利于系统的横向扩展。

[0079] 以上所述仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

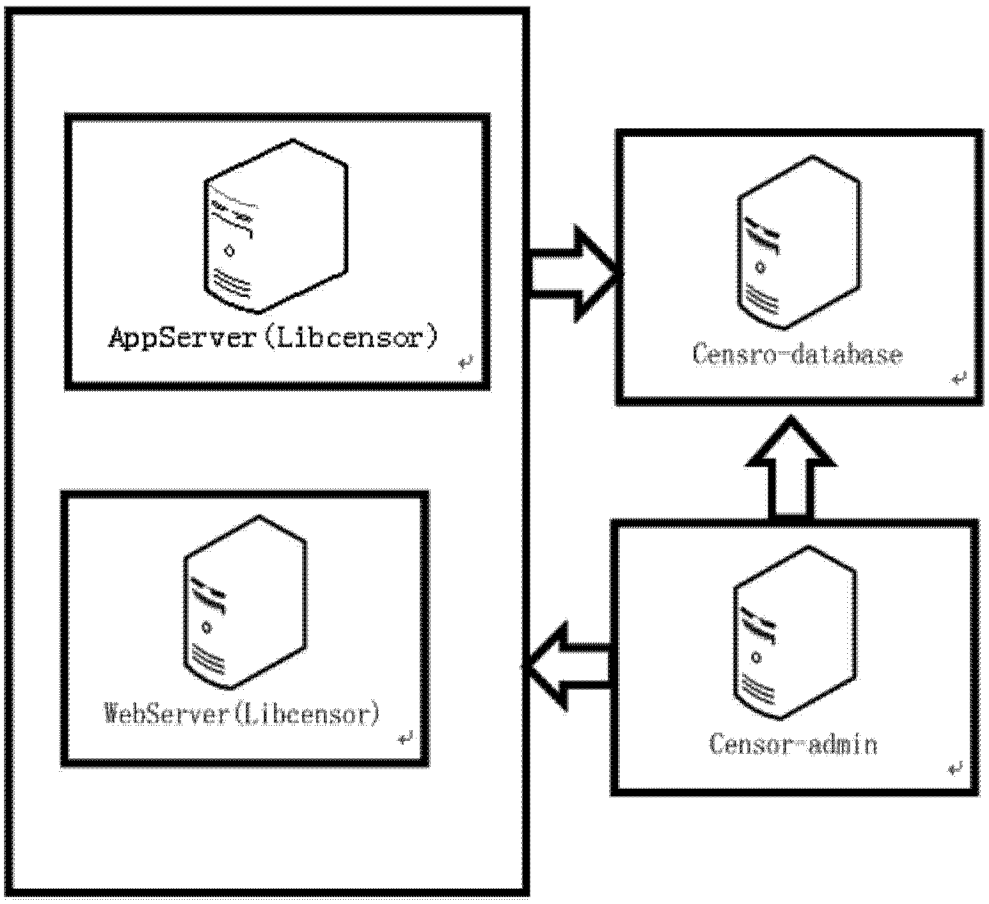

AppServer(Libcensor)
Censro-database
WebServer(Libcensor)
Censor-admin

图 1

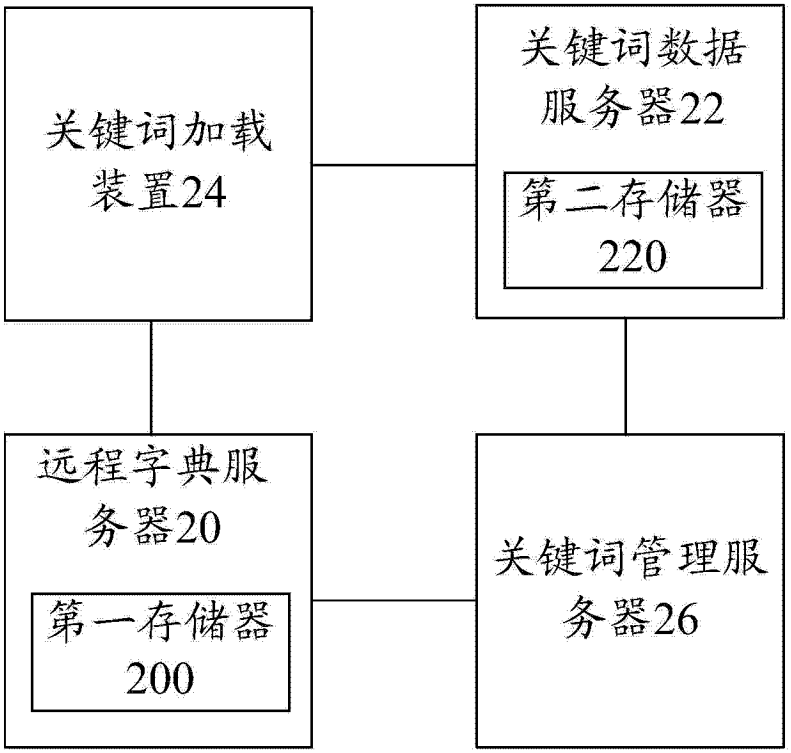

关键词加载装置24
关键词数据服务器22
第二存储器220
远程字典服务器20
第一存储器200
关键词管理服务器26

图 2

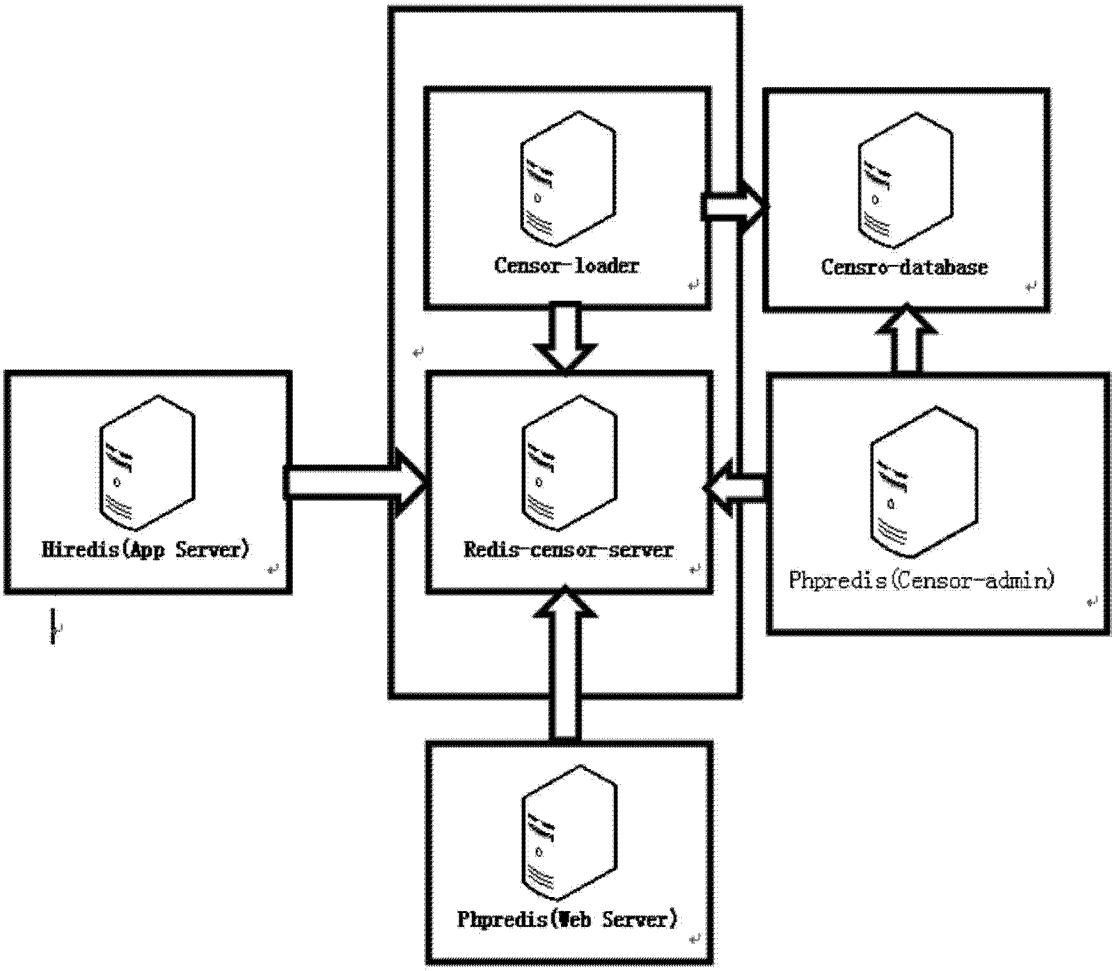

Censor-loader
Censro-database
Hiredis(App Server)
Redis-censor-server
Phpredis(Censor-admin)
Phpredis(Web Server)

图 3

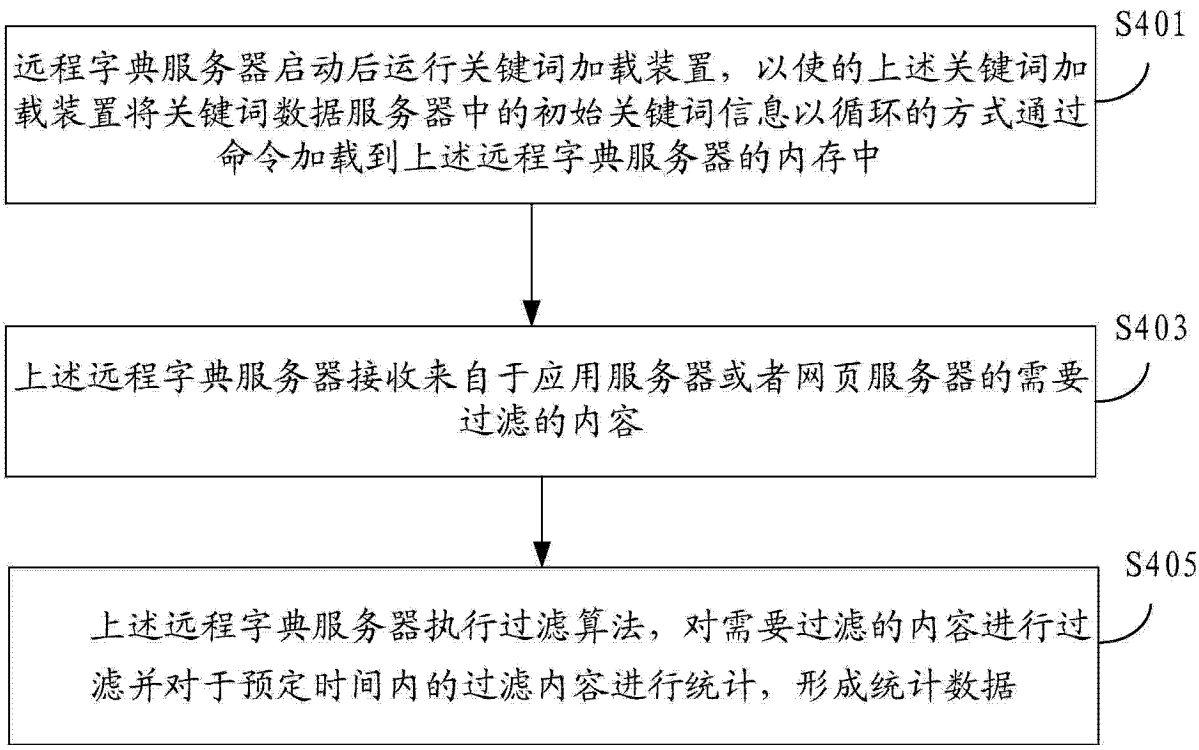

S401
远程字典服务器启动后运行关键词加载装置，以使的上述关键词加载装置将关键词数据服务器中的初始关键词信息以循环的方式通过命令加载到上述远程字典服务器的内存中
S403
上述远程字典服务器接收来自于应用服务器或者网页服务器的需要过滤的内容
S405
上述远程字典服务器执行过滤算法，对需要过滤的内容进行过滤并对于预定时间内的过滤内容进行统计，形成统计数据

图 4

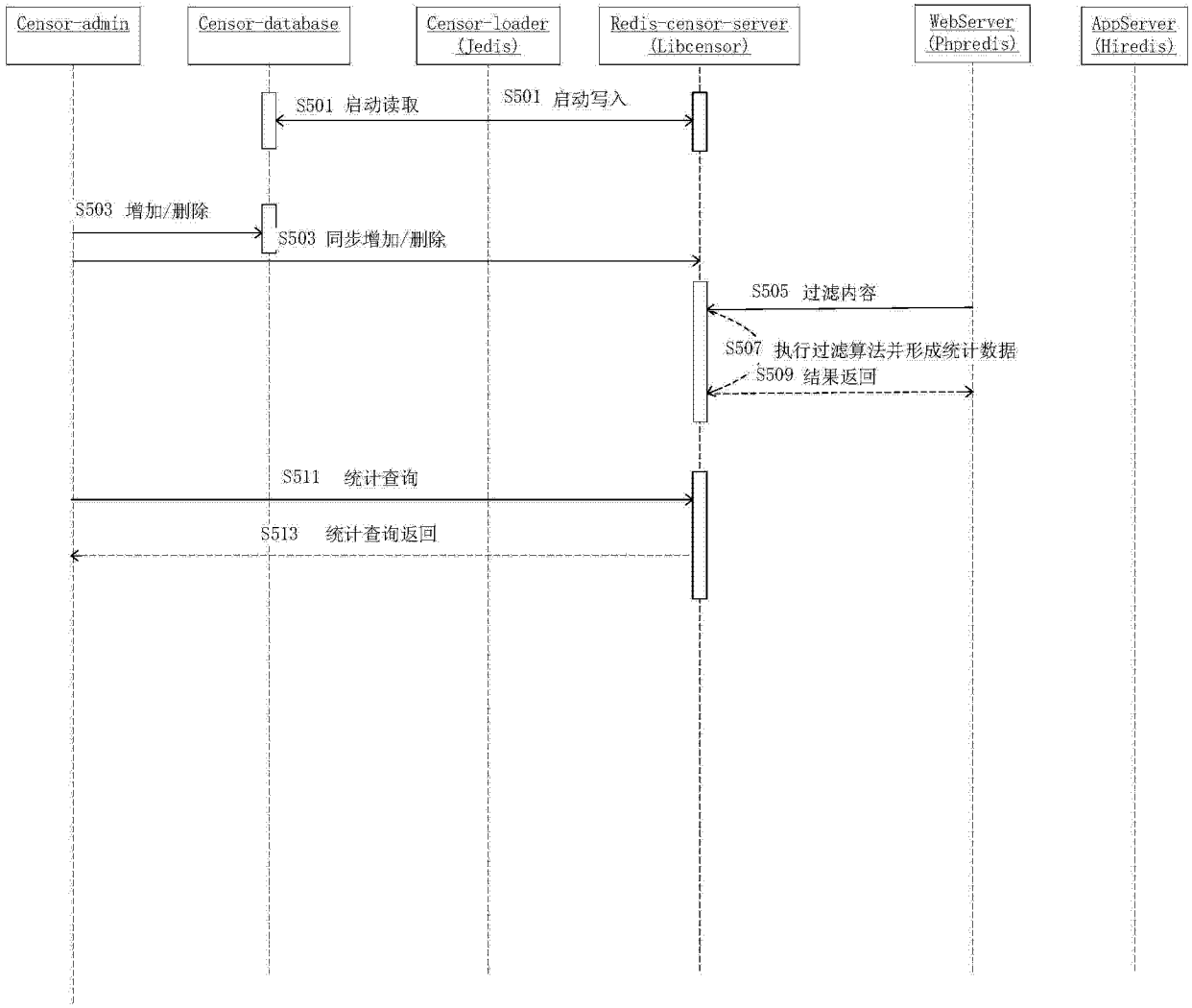

Censor-admin
Censor-database
Censor-loader (Jedis)
Redis-censor-server (Libcensor)
WebServer (Phpredis)
AppServer (Hiredis)
S501 启动读取
S501 启动写入
S503 增加/删除
S503 同步增加/删除
S505 过滤内容
S507 执行过滤算法并形成统计数据
S509 结果返回
S511 统计查询
S513 统计查询返回

图 5