

海量电子文档的分布式存储策略

陈雪玲 高福祥 董 玲

(东北大学信息科学与工程学院 沈阳 110004)

摘 要 针对现代企业文档管理的现状及需要,提出了一种基于 Web 的企业级文档管理系统中海量电子文档的分布式存储策略。详细描述了这一策略的物理架构、设计思路及实现。在文档管理系统中的成功应用,表明这一存储策略能够很好地满足海量电子文档的存储需求。

关键词 电子文档,文档管理,分布式存储

1 引言

随着计算机技术的发展、计算机应用的普及,以及企业内部业务管理系统的逐步建立,企业累积了大量的信息资源,科学地管理和有效地利用这些信息资源已成为提高企业竞争力的关键。IBM 研究部门调查发现,在这些海量信息资源中,只有 15% 是传统关系数据库管理系统能够处理的结构化数据,剩余的 85% 的信息都是非结构化的,包括纸上

的文件、报告、视频和音频文件、照片、传真件、信件等^[1]。如何同时对结构化和非结构化的信息进行统一的管理,最大限度地挖掘有价值的信息已经被越来越多的企业和开发商所关注^[2]。

2 分布式海量存储系统的设计与实现

2.1 系统的基本架构

在一个基于 Web 的文档管理系统中,分布式海量存储系统的结构如图 1 所示。

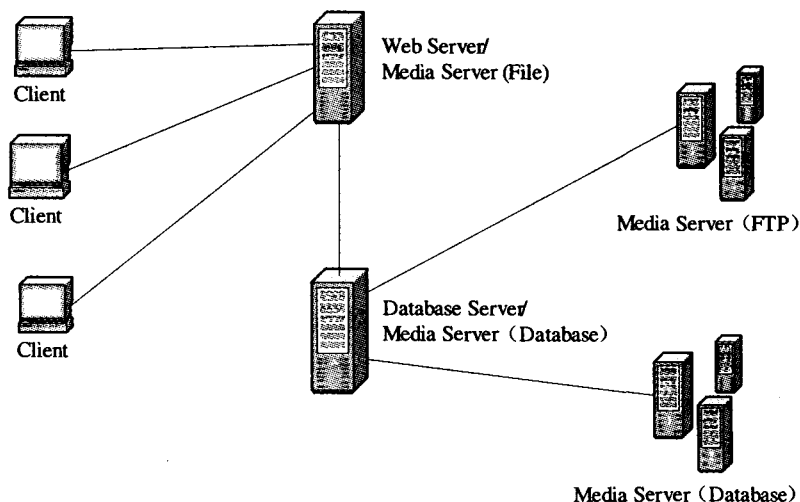


图 1 系统结构

图 1 所示的系统主要由四大部分构成:Client 为客户端浏览器;Web Server 为 Web 应用服务器,系统的应用程序包就在这里;Database Server 为数据库服务器,存放系统的所有结构化数据,包括目录信息、文档记录信息、权限控制信息、介质配置信息等;Media Server 为存储介质服务器,用于存放系统中的所有非结构化数据,也就是电子文档,这样的介质服务器在系统中可以同时存在多个。

对于 Media Server 来说,在系统中可以由三种服务器来充当,分别是:本地文件系统、远程 FTP 服务器和数据库服务器。

(1) 本地文件系统

将电子文档存放在系统当前 Web Server 的本地目录中,因此 Web Server 本身也可当作系统的 Media Server 介质服务器之一。

(2) 远程 FTP 服务器

将电子文档存放在远程 FTP 服务器的目录中。

(3) 数据库服务器

将电子文档存放在当前数据库服务器或远程数据库服务器中,作为大字段存放。

在系统中,电子文档的逻辑属性信息集中存放在 Database 服务器中,而实际的电子文档分散地存

放在各个 Media Server 上,这就是一个基于 Web 的文档管理系统中,分布式海量存储设计的基本架构。

2.2 设计思路

在系统中,所有 Media Server 都由系统管理员创建并进行管理维护,对于系统前端处理具体文档业务的人员来说,为自己的文档目录在系统提供的可用的 Media Server 中选择一个即可,不需要关心这个 Media Server 实际的物理位置。

(1) Media Server 的配置与使用

这些 Media Server 的相关配置信息都以 URL 的字符串形式统一存放在 Database 服务器中,URL 字符串的语法形式完全按照 RFC2396 定制,如:文件系统类型介质的 URL 形式为 file://path;FTP 类型介质的 URL 形式为 ftp://user:pwd@host-name:port/path;数据库类型介质的 URL 形式为 db://user:pwd@drivername/jdbcurl 和 db://。(表示使用当前数据库服务器)。如果文档目录使用了某一介质,使用时通过该介质的 URL 动态构造相应的介质解析类,URL 解析成功后,即可进行相应文档操作了,比如获取、写入、删除介质中的电子文档。针对不同类型的 Media Server 进行相应文档操作时,需使用不同的 URL。

(2) 电子文档在 Media Server 上的压缩存储

在文档管理系统中的文档目录下,上传电子文档时,系统会先登录该目录的相应 Media Server(通过 URL 配置),再将电子文档进行压缩后,存放到 Media Server 中。系统在获取电子文档时,从 Media Server 上获得文件后还需要进行相应的解压缩。这样大大节省了 Media Server 的物理空间,提高了使用率。目前可使用的压缩方式包括 GZIP 和标准的 WINZIP,具体使用哪种压缩方式取决于系统安装时的初始配置。

(3) 电子文档在 Media Server 上的命名策略

电子文档在 Media Server 上保存时,压缩后是不以原始文件名形式保存的。系统在 Media Server 上采取的文件命名策略是:电子文档在上传后,系统即为其分配一个 32 位长的数字唯一标识,那么在 Media Server 上保存时,先将这 32 位数字按 8 位进行分解,前三个 8 位分别作为三层目录的目录名,最后一个 8 位作为文件名,如电子文档标识为 2713756096,那么它在介质服务器中的存放位置就为:./161/192/161/192。这种命名策略仅适用于文件系统类型和 FTP 类型的 Media Server,对于数据库类型的 Media Server,在数据库中,直接以其唯一标识为名存入相应大字段类型字段即可。这样的存储方式更有益于 Media Server 上对系统电子文档的统一管理和维护,也增加了数据的安全性和保密性。

(4) Media Server 的有效性合法性校验

一旦某个本地目录,或远程 FTP 目录,或远程数据库被指定为当前系统的 Media Server,那么它将无法再为其他系统提供存储介质服务,因为当前系统在相应目录或数据库中设置了介质标记。当系统管理员在创建 Media Server 时,系统会首先检查该 Media Server 配置的合法性,如实际物理位置是否存在,是否已存在介质标记等。这样,在保证 Media Server 介质服务器的合法性的同时,也保证了其上的电子文档数据的安全性。

(5) Media Server 的预警安全

在系统中,系统管理员统一对多个不同类型的 Media Server 介质服务器进行维护,他能够查看每个 Media Server 的使用情况,如果达到或快要达到设定容量时,系统会通过消息管理模块发出介质预警信息,通知相关人员介质将满,尽快采取解决措施。这无疑增加了 Media Server 的可靠性。

目前的预警模式有两种,一种是剩余容量方式(如剩余 10M 时发出预警),一种是利用率方式(如利用率达 95%时发出预警)。

消息管理模块是文档管理系统中另一功能模块,与分布式存储系统相对独立,目前支持的消息发送方式有三种:用户登录即时消息、电子邮件、手机短信息。

2.3 系统扩展性设计

为实现系统中存储介质的统一管理与调用,本系统设计了 IMediaHandle 接口,开发人员通过 CMediaHandleFactory 工厂类获取相应的 IMediaHandle 实例。IMediaHandle 接口将系统存储介质读写电子文档的功能封装成独立的方法,包括获取文档,写入文档,删除文档等。其中三个 write 方法分别提供了以文件名(filename),文件(File)以及输入流(InputStream)三种写入介质的方式。CFileHandle, CFtpHandle, CDbHandle 分别是 IMediaHandle 针对三种不同存储介质的分别实现。

CMediaHandleFactory 这个工厂类屏蔽了 IMediaHandle 接口实例的创建,外部程序只需调用 CMediaHandleFactory.getMediaHandle()方法即可动态获取 IMediaHandle 相应的实例,然后再调用相应的读写方法,即可完成向相应的存储介质中读取文档,写入文档,删除文档的操作,而不需要关心这个存储介质是什么类型的,物理位置在哪里。

如果系统以后增加了新的存储方式的实现,比如 webdav 等,可以直接增加一个针对新存储方式的 IMediaHandle 的实现,对于外部程序来说,无需修改任何相关代码,就可以很好地完成一个系统功能的扩展,完全符合了面向对象设计的开闭原理。

2.4 具体实现

分布式存储系统的实现是基于 Web 应用框架 Struts 进行开发的。Struts 是一个开放源码项目,具有高可配置性,有一个前端控制组件,一系列动作类,动作映射,处理 XML 的实用工具类,服务器端 Java Bean 的自动填充,支持验证的 Web 表单,国际化支持,生成 HTML,实现表现逻辑和模板等^[3]。Struts 底层是基于 MVC 设计模式的。本系统的实现基于 Struts1.1 版本。

系统的实现共分为 4 大部分:介质管理的相关实现,介质调用接口的实现,介质存储类的实现,介质预警的相关实现。介质管理主要实现介质管理的界面以及数据库服务器中 Media Server 记录信息的维护与管理;介质调用接口主要实现介质调用接口的定义(即从 Media Server 读写电子文档的方法定义),以及目前系统支持的三种 Media Server 的调用接口的实现;介质存储类相关实现包括实际介质目录的创建与移除,以及介质目录合法性的校验等;介质预警的实现主要依赖于文档管理系统中的消息管理模块和系统守护进程管理模块,系统通过一个介质预警检查的守护进程(该进程已注册到系统守护进程队列中,定期执行,且时间周期在系统内可配置),获得需要发送预警信息的 Media Server 后,组织好预警信息,将预警信息提交给消息管理模块即可。

结论 据 Jupiter Research 的报告显示,企业所产生的文档数量每 6~8 个月便以双倍或更快的速率急增,如此大的文档量对电子文档管理系统的

存储能力提出了新的挑战。这个分布式海量存储设计方案与其他存储方案相比较有很多优势^[4,5]。

(1)同时支持多种类型存储介质,包括本地文件系统、远程 FTP 服务器、本地或远程数据库服务器以及其他;

(2)存储介质调用接口与系统松耦合,更易于扩展;

(3)采用多种方式压缩存储,更节省存储介质空间;

(4)采用文档加密技术,使文档数据更安全;

(5)易于同其他大型信息软件系统,如 HIS, PACS, ERP 等集成。

目前这种分布式海量存储方案已成功应用于基于 J2EE 的文档管理系统中,系统运行一段时间以来状态良好,完全能够满足企业、政府等的文档管理和存储要求。

参考文献

- 1 付铮. 内容管理峰回路转市场需求日益攀高[J]. 电信工程技术与标准化, 2004(5): 73~75
- 2 邵志举. 海量数据高效存储与管理研究: 一种高效海量数据存储、管理与实时发布方案[J]. 现代软科学, 2006(3): 127~129
- 3 陈志凌, 程耀东, 刘爱贵等. 海量存储文件系统 CastorFS 的设计与实现[J]. 计算机工程与应用, 2006, 42(4): 108~110
- 4 周志华, 何萍, 尹建伟等. 基于 Web 的海量存储柔性分布式文件服务器设计[J]. 计算机应用研究, 2002(11): 14~16
- 5 杨本杰. 企业电子文档的安全存储和访问问题的研究[J]. 工业控制计算机, 2003, 16(11): 3~7

(上接第 32 页)

5)易于运营:控制流和业务流完全分离,易于实现多业务运营,少量改造传统网络即可升级成运营级网络。

结论 建设可管理、可运营的高校网络正成为 IT 业的热点之一,是互联网进一步发展的潜在市场。校园网络要做到学校要求的可运营、可管理和用网公平性,身份验证、授权和计费是必不可少的。利用 802.1X 认证方式,为高校运营提供了一个认证、授权和计费的整体解决方案。

802.1X 认证计费管理方案特色体现:与安全交换机结合,可以实现对用户账号、IP、MAC、交换机 IP、交换机端口、VLAN ID 六元素任意绑定的严格认证,实现包月、预付款、计时长、计流量、实时扣费等计费原型,更支持用户自定义的计费策略,并且支

持限制用户固定 IP 地址,或限制用户只能是 DHCP 动态获取 IP,或限制用户只能是静态设备 IP 地址功能,完全可以满足学校对认证计费的综合要求。

参考文献

- 1 Mamakos L, Lidl K, Evarts J, Carrel D, Simone D, Wheeler R. A Method for Transmitting PPP over Ethernet (PPPoE), February 1999
- 2 罗娟. 网络计费系统的关键技术. 网络世界, 2001(7)
- 3 钟健松. 宽带网认证计费方式比较. 北京朗新信息系统有限公司, 2002(10)
- 4 杜宾. 宽带城域网的建设与以太网的发展. 网络世界, 2002(3)
- 5 周卫东, 等. RADIUS 协议安全分析. 计算机应用与软件[J], 2002(11)
- 6 易正强, 王耀军. 基于 Proxy 的网络计费系统的设计与实现[J]. 计算机工程, 2001, 27(2): 166~168