

栗 欣 (国家安全生产监督管理局
通信信息中心 100713)

分布式文件传输系统的构建及实施

Build and Implement of Distributed File Transfer System

摘 要: 本文介绍了基于 Lotus Domino 平台, 构建 B/S 模式的文件管理系统, 建立一种分布式的、用户可定制的、灵活的文件流转、文件传输系统。文中说明了分布式文件传输系统的构建以及实施原理, 并对系统的体系结构与功能特性进行了描述。

关键词: 办公自动化 分布式 公文流转 文件传输

1 引言

系统首要目的就是要解决在广域网和分布式环境下的公文收发。每一个应用点(单位)都具有相对独立性, 而通过在应用层统一、规范、安全地规划, 使这些独立的应用点得以整合形成一个有机的整体应用体系, 从而实现广泛的、无瓶颈的应用互连及信息交换和共享。

2 系统的结构与功能

2.1 体系结构

2.1.1 系统构架

系统设置了收发中心、流转中心、利用中心、监控中心等四个中心以满足公文应用的需要(如图1)。

这四个中心既是各自独立地运作以履行其职能, 又因为相互之间的信息传递产生联系, 从而构成一个有机的整体。

2.1.2 各种业务平台

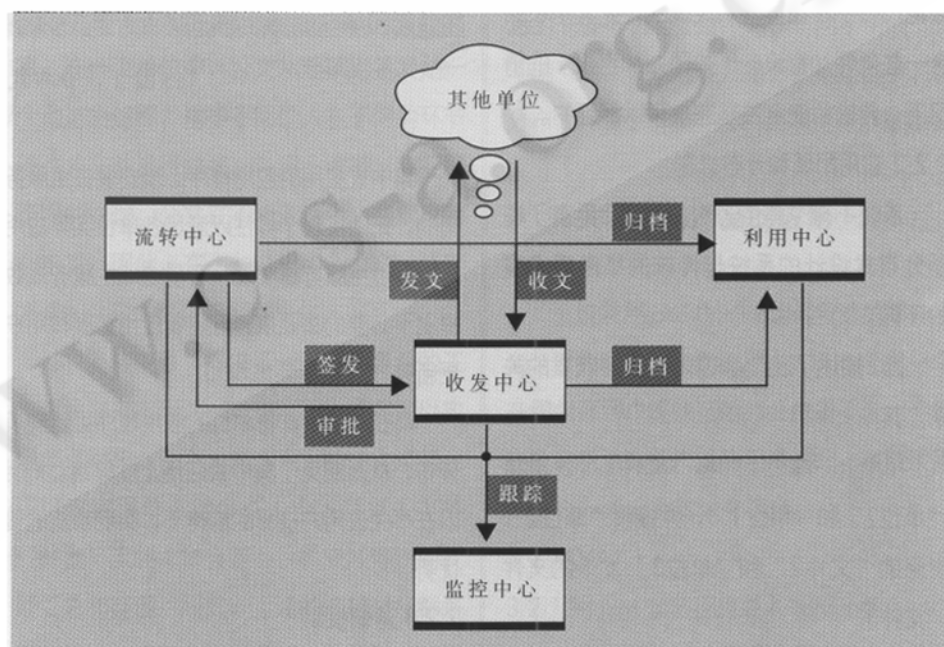


图1 体系结构示意图

(1) 网络应用支持平台。尽管目前在广域网和局域网中广为流行TCP/IP网络协议, 但多种网络协议并存的局面仍将继续存在。综合采用多种技术实施网络的管理、监控、控制对确保整个系统运行的高效安全是十分必要的。

(2) 信息资源管理平台。信息资源管理平台应对文本信息、数字信息、图象信息、声

音信息进行管理。通过基于超文本链的资源定位系统及开放的体系结构使数据库与WEB协同工作，以满足信息服务和发布的需要。

(3) 消息传递与工作流管理平台。消息传递和追踪工作流的管理平台用以发布信息并及时掌握信息具体流向和反馈，提供有效的文件跟踪能力以提高办公效率。

(4) 业务处理平台。为各级领导和部门提供日常办公及管理服务。

2.2 功能

2.2.1 传输

在每一个网员单位，我们设置一个消息收发“适配器”，以实现公文的收发和伴随的回执、反馈、跟踪等信息的传递，使得“收发”功能就象一个零件一样可以被安装到应用系统中。这实际上是对公文收发过程的一种包装，对于完整地描述和体现这一过程在应用上具有重要的意义。

公文在传输中的安全性主要是通过网络数据加密手段在传输层实现的，尤其是在结合一些硬件加密设备（如国办加密机）的情况下，传输的安全性是可以保证的。

2.2.2 应用配属和分布机制

系统不是“集中办公”。图2示意了基于分布式设计的系统与传统的集中式方案的不同。

图2描述了三个单位之间文件收发的关系。其中：

(1) “单位1”将“文件1”发送给“单位2”和“单位3”，并接收“单位2”发来的“文件2”和“单位3”发来的文件“文件3”；

(2) “单位2”将“文件2”发送给“单位1”和“单位3”，并接收“单位1”发来的“文件1”和“单位3”发来的文件“文件3”；

(3) “单位3”将“文件3”发送给“单位1”和“单位2”，并接收“单位1”

发来的“文件1”和“单位2”发来的文件“文件2”。

基于集中式设计，一般需要设置一个“文件交换中心”用作所有文件的存储中心（一般位于专网上的主服务器上）。所谓发送，就是各发文单位将所发送的文件存放到“文件交换中心”；而接收，则是各单位按授权直接访问“文件交换中心”中的文件。整个收发过程并未发生实际的文件流动（如图2）。

而本系统由于采用分布式设计则不需要所谓“文件交换中心”。发文是直接发送给收文单位。各单位之间的文件收发是直接点对点实现的，而无需中间存储或转存环节的存在（如图2），从而避免了文件在中间环节（如文件交换中心）的停留，提高效率，同时消除主服务器端文件存储访问的瓶颈。

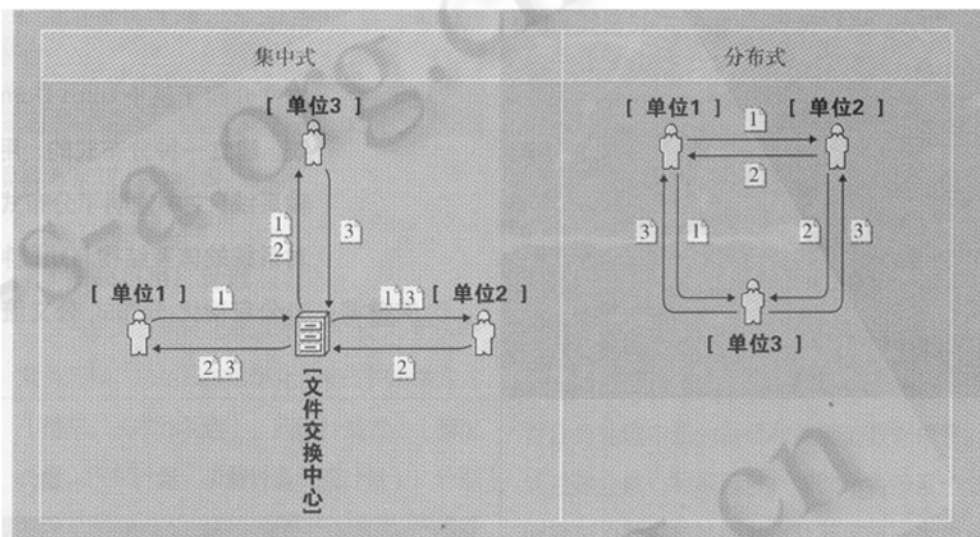


图2 分布式与集中式设计的差别

集中式应用的应用程序必须安装在主服务器上，这意味着所有接入用户所使用的程序是相同的，无法实现个性化以满足各单位实际应用需求的差异；而更为致命的是，所有的文件交换必须经过主服务器实现，容易造成主服务器数据计算和数据通信的瓶颈，势必对服务器的性能提出不必要的高要求，且随着应用扩展和用户群的增加，这种要求会越来越高，使得服务器的更新换代和花费频繁增加。

而在系统中，各单位的应用程序可以安装在不同的服务器上，实现了数据的分布和应用的分布，从而避免了集中式应用上述弊端。而且系统采用B/S结构，用户端都是浏览器界面，可以在不失去用户接受的前提下，实现功能的普遍适用性，具有适用面广、使用简单、跨平台的优势。

2.2.3 通信机制

公文的发送、接收过程在后台是通过邮件收发来实现的。但是，公文的收发不象日常的收发邮件那样简单，还需要考虑监听、回执、反馈、跟踪等一系列伴随或相关处理。这样看来，如果我们不说传输和接收的是公文，而是消息或报文，那么，公文的收发过程可以描述成一个适用于消息中间件实现的功能（如图3）。

2.2.4 信息共享机制

从概念上说，没有信息共享，一个个各自分离、隔绝的信息孤岛——单位/个人是谈不上

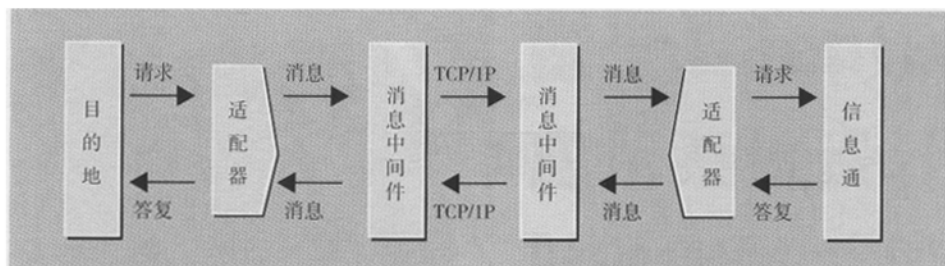


图3 公文收发过程

“联合办公”的。因此，信息共享是实现“联合办公”的基础。

需要特别说明的是，在「联合办公」中，共享的主要对象是公文。“公文”并非象在文件系统中那样只是一个孤立的特定文件。本文中所提到的“公文”都是由“正文”、“条目”（如发文的发文稿纸）、“附件”、“处理意见”、“处理状态”等内容整合而成的一个复合体，我们称之为“复合文档”。

“联合办公”对信息共享的要求可以分成两个层次：

(1) 单位和单位之间的信息共享。

(2) 单位内部的信息共享。据此，系统设置了“收发中心”、“流转中心”、“利用中心”等三个中心以满足这些要求。

“收发中心”用于实现单位和单位之间的信息传递（在这里特指公文收发，其实现机制在前文中已经描述），从而跨越单位之间的信息壁垒。公文的收发使得信息不必孤立在一个单位的内部，而是可以有目的地流向其他单位。在图2中，单位1所发送的文件1，在单位1是一个发文，而单位2和单位3签收后，就变成了他们的收文。这样，同一个文件“文件1”，三个单位都可以看到，并分别进行相关的处理。这样，实际上就解决了单位和单位之间的信息共享问题。同时，对这三个单位来说，文件1都是其各自单位内部的文件，因此其共享就成了单位内部信息共享的问题。

单位内部的信息共享包括两方面的内容：动态的信息共享和静态的信息共享。下表1描述了两种信息共享方式的不同。

	动态信息共享	静态信息共享
公文	变化	不变化
处理方式	变化	不变化
处理人	变化	变化

“流转中心”用于实现动态的信息共享。在一个单位中，公文从起草到签发是一个动态的过程，需要不同的人在不同的环节修改公文并进行核稿、办理、审批、会签等不同的处理。公文在不断流动中，得到改善和丰富。因此，在这一过程中，公文、处理方式和处理人都可能发生变化。

“流转中心”提供了一套方便灵活的工作流机制来实现这一动态流程。不仅可以实现单点对单点（个人对个人）的公文流转，还可以实现单点对多点（个人对多人）的“分发”流转（如会签），并完成伴随的反馈汇总和整理。而且，上述的流转过程都是双向的，“流转中心”通过“收回”功能取消已执行的流转操作。

2.2.5 全程跟踪机制

政府公文因其严肃性和重要性，无论是内部的流转处理，还是对外的收发，都要严格地确保安全。

安全性的保障一方面要求事前的预防，

这方面在前文中的“通信机制”和“信息共享机制”两节中已经进行了介绍；另一方面则要求事后的跟踪以追查和落实责任，这一点对于应对各网员单位内部可能的恶意操作和误操作尤为重要，这就是系统提供全程跟踪机制的意义。

首先，系统实现了公文的收、发及相应的回执、反馈过程的全程跟踪。不仅跟踪公文、回执和反馈信息的流向，同时跟踪记录发送这些信息的人员、单位、发件机IP地址和Mac号等相关信息，从而确保信息传递的安全并可将来责任落实到人/机。

其次，公文在单位内部的处理也按照其处理环节的划分，对其处理细节予以跟踪记录。针对每一件公文，“什么人在什么时候干了什么”，都一一记录在案，并以“处理明细”的形式，作为公文的一个特定属性存在于公文内部以随时备查。

“全程跟踪”就是系统的监控中心。在这个“监控中心”的监控之下，不会存在信息的来去不明、处理不明、访问不明的“三不明”现象，从而在规范分工的前提下，使分布式环境下群集办公的透明度得到提高。

3 有关技术与产品的应用

3.1 通信

系统采用Lotus Domino为其首选的邮件平台，以实现文件的收发。Lotus Domino是一个在国内得到广泛认可的办公应用平台，尤其在政府机关有着广泛的用户基础。系统采用Lotus Sametime为其消息平台。

3.2 JAVA

3.2.1 用JVM加载程序装入servlet类

servlet管理器类加载程序不会装入使用自身代码、创建定制类加载程序或执行其他某些受限制的操作的类。如果servlet所要求的类不能通过servlet管理器装入，可以试用Domino JVM类加载程序进行装入。JVM加载程序通常用于与Domino一起安装的Java归档系统（特别是java.*和lotus.*软件包）

中装入类。通过将 servlet 从 Servlet 管理者类路径移到 JVM 类路径, 可以强制使用 JVM 加载程序而不使用 servlet 管理器加载程序来装入服务器小程序。JVM 类路由 NOTES.INI 变量 JavaUserClasses 指定。

Servlet 是为响应 Web 浏览器的请求而运行的 Java 程序。与代理不同, 当 Web 服务器启动且驻留在服务器上时 Servlet 才装载。它们通常用于动态生成和更新 Web 页以及在不同的应用程序之间交换数据。可以扩展 Servlet 的功能, 以使用 XML 作为应用程序之间的通用语言在应用程序之间架起一座桥梁。Java servlet 不仅能产生 XML 标记并传递到服务器以进行处理, Servlet 还可以与 LotusXSL 处理器进行交互作用以格式化 XML 标记描述的数据。XML 与 XSL 配合使用, 将为您提供一种定制数据传输的强大工具。

3.2.2 XML Servlet 应用程序的样例

作为 Servlet 如何促进 XML 传送定制数据包的样例, 可考虑一个组织, 其中的域销售代表使用不同的设备 (如 Notes 客户机、浏览器或 PDA) 从 Domino 数据库下载信息。销售代表可向 Domino 应用程序请求所有与特定客户有关的信息。Servlet 将来自不同数据源的信息集合到一起并使用合适的 XML 标记将数据打包。Servlet 还可以使用 LotusXSL processor 将样式应用于使用 XML 标记的数据并以最适合于连接设备的格式传送数据。这种情况下, 通过窄带电话线与 PDA 连接的销售代表比通过宽带网络连接与 Notes 客户机连接的销售代表获取的信息要少。

图4显示连接设备、运行 Servlet 的 Domino 应用程序以及后端数据库之间的关系:

3.2.3 创建和使用 Servlet

可以从头开始创建 Servlet, 通过 URL 引用现有的 Servlet, 或使用预置 Servlet。每次修改 Servlet 时, 必须重新启动服务器以查看所做的改动。运行 Servlet 涉及到编写 Servlet、在 Domino 中启用 Servlet 支持以及

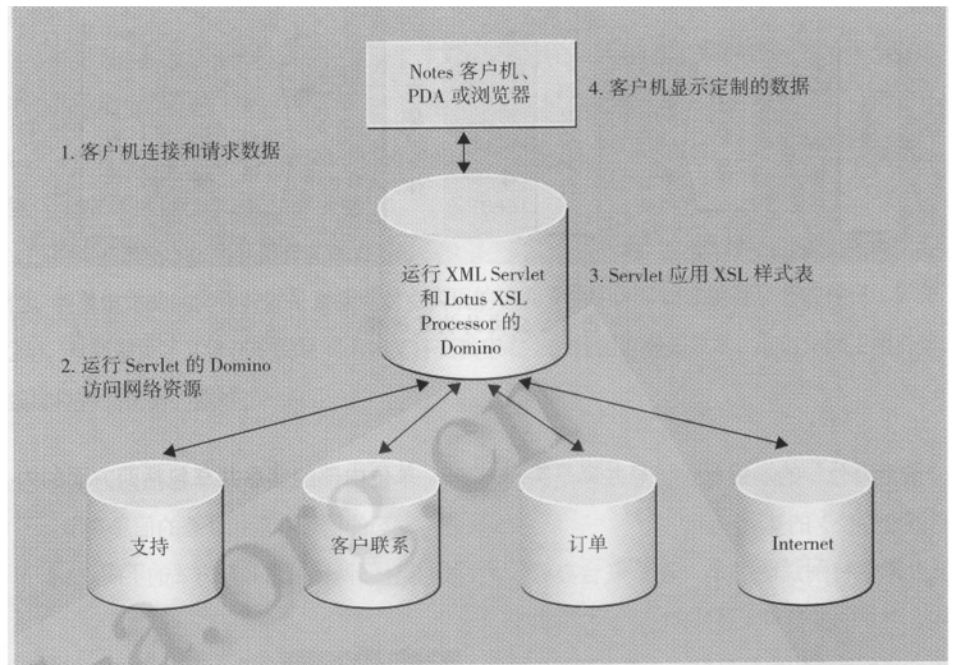


图 4 连接设备、运行 Servlet 的 Domino 应用程序及后端数据库关系图

必要时设置 Servlet 属性等方面的内容。

3.2.4 代理、servlet 和 CGI 程序的比较

代理、servlet 和 CGI 程序都允许对 Domino Web 应用程序的功能进行扩展。代理可以通过表单 WebQueryOpen 和 WebQuerySave 事件与 Web 应用程序紧密集成。Servlet 通过 Servlet API 类可以获得一些特殊功能, 如会话和 Cookie 管理。由于 Java 日益受到人们的青睐, 现今使用 servlet 而不是 CGI 程序来开发新产品已成为大势所趋。但是, 许多现有的 CGI 程序仍在使用。

如果正在编写自己的应用程序并需要为服务器上的某些功能编程, 则需对使用哪种类型的程序的选项作出选择。每种类型的程序均有各自的优点, 而且可能成为特定情形下的最佳选择。表2是对每种程序类型的使用建议:

程序类型	最佳用途
代理	读取或邮寄文档时执行 Domino 操作的程序。 需要按预定时间或当数据库操作发生 (如新邮件到来) 时运行的程序。
Servlet	使用标准 Java 接口 (如 JDBC) 的程序。 使用 HTTP 会话维护或 Cookie 的程序。 复杂或资源密集型 Java 程序。
CGI 程序	需要对系统资源进行低级别访问的程序。 通过非 Java API 与其他产品连接的程序。