第 35 卷 第 8 期 计 算 机 工 程 2009 年 4 月
Vol.35 No.8 Computer Engineering April 2009
·软件技术与数据库· 文章编号：1000—3428(2009)08—0091—03 文献标识码：A 中图分类号：N945.12

一种面向服务的 P2P 存储系统模型

宋 玮[1,2]，赵跃龙[1,3]，曾文英[1]，王文丰[1]
(1. 华南理工大学计算机科学与工程学院，广州 510006；2. 广东工业大学计算机学院，广州 510006；
3. 中南大学信息科学与工程学院，长沙 410083)

摘 要：P2P 存储系统以功能对等的方式组成存储网络，面向服务的体系结构为存储资源的有效管理以及按需服务的实现提供了一种思路。提出的 P2P 存储系统模型采用分层思想，建立在结构化覆盖网络之上，将异构节点存储资源封装成服务块，以用户需求为出发点，通过服务的动态选取和组合，形成可定制的个人存储视图，并给出一种对等节点的功能部署结构。模型达到分散控制，具有良好的可扩展性。
关键词：P2P 存储系统；面向服务；动态选取与组合；按需服务

Service-oriented P2P Storage System Model

SONG Wei[1,2], ZHAO Yue-long[1,3], ZENG Wen-ying[1], WANG Wen-feng[1]
(1. School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006; 2. Faculty of Computer, Guangdong University of Technology, Guangzhou 510006; 3. School of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】P2P storage system is organized in a peer-to-peer way. Service-oriented architecture is an idea to provide effective storage resources management and service on demand. The P2P storage system model presented is constructed on structured overlay network using layer mechanism. Storage resources on heterogeneous nodes are encapsulated in service blocks. Individual storage view can be customized from users' point by dynamic choice and combination of services. In addition, deployment of layers in peer is also described. This model is decentralized controlled and has good scalability.
【Key words】P2P storage system; service-oriented; dynamic choice and combination; service on demand

1 概述

随着信息量的增长，用户对存储的需求越来越大。购买大容量的存储设备是一个很好的解决办法，但是先进而海量的存储设备却是一般用户所不能支付的，同时随着应用的变化，需求也会发生变化。借鉴 P2P 系统中利用对等节点空闲计算力的思想，采取聚集对等节点空闲或者自愿提供的存储力的方式来扩大用户的存储能力。本文采用文献[1]中所定义的 P2P 存储系统，即指存储节点以一种功能对等的方式组成的一个存储网络。P2P 存储系统既可以完全由服务器节点以对等方式组成，又可以完全由用户桌面机组成，也可以是 2 类节点共同以对等的方式组成的存储系统。这样，空闲或自愿提供的存储力既可来源于专业的大型存储服务，也可以来自闲散的桌面机资源。

不同用户对存储资源的需求不一样，同一用户在不同的时期对存储资源的需求也不一样。因此，如何搭建合理的存储资源管理基础架构以及怎样管理复杂的资源系统，使得资源能被有效管理和按需服务[2]，就自然成为 P2P 存储系统待解决的基础问题。面向服务的体系结构(SOA)为实现按需服务提供了一种思路。它是一个组件模型，将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来，接口采用中立的方式进行定义，独立于实现服务的硬件平台、操作系统和编程语言，使得构建在各种这样的系统中的服务可以用一种统一和通用的方式进行交互[3]。

为充分利用 SOA 提供按需分配的能力，提出一种面向服务的 P2P 存储系统模型。模型将对等节点提供的存储能力封装成存储服务，其他对等节点可根据自己的需求定制合适的存储服务，模型在一定程度上保证该存储服务的长期有效性。

2 相关研究

目前已有很多 P2P 的存储项目。CFS[4]是一个只读的文件系统，它提供文件系统的语义，只允许信息发布者更新内容，不支持同步更新的语义。底层覆盖网络使用 MIT 提出的 Chord；DHash 负责块级别的数据存取，并维护数据冗余；客户端的文件系统层负责提供文件系统接口与数据块之间的转换。OceanStore[5-6]系统构建在较为稳定的由服务商提供的节点集合上，使用 Berkeley 提出的 Tapestry 覆盖网络，系统中的数据是可以共享和全局可访问的，既保证数据私密性又保证其完整性；系统提供一定的数据一致性保证。BitVault[7]是一个面向较少更新的参考数据的存储系统，它由较为稳定的机房内部存储节点构成，建立在微软亚洲研究院提出的 Xring 覆盖网之上。Granary[8]系统的设计目标是能够自适应地支持高动态系统和稳定系统，并提供面向对象的存储。

上述系统分别在工作环境(稳定或动态)、存储层次(块级或文件级)、采用的覆盖网络上各有不同，这也是由于各自的应用背景不同所导致。本文所提出的 P2P 存储系统模型，希望能给出一个通用的存储系统构建平台，将对等点提供的存

基金项目：国家自然科学基金资助项目(60573145)；湖南省自然科学基金资助项目(05JJ30120)；广州市科技计划基金资助项目(2007J1-C0401)
作者简介：宋 玮(1978－)，女，讲师、博士研究生，主研方向：P2P 系统，网络存储技术；赵跃龙，教授、博士；曾文英，副教授、博士研究生；王文丰，博士研究生
收稿日期：2008-07-27 **E-mail：**color_unsw@126.com

储力封装成具有标准定义的存储服务，目的是为不同需求展现一个可定制的存储视图。这在实际的应用中有一定的必要性。首先，对于普通用户，他们进入系统的随机性很大，对存储资源的需求量小，需求不具有长期性。从有效利用资源的角度考虑，应该为用户提供可选的存储服务，建立各种用户的局部数据存储视图。其次，资源的使用并不总是无偿的，从有偿使用角度看，用户只能够使用他所能够支付的数据存储服务。

3 面向服务的 P2P 存储系统模型设计原理

每个对等节点是服务的提供者也是使用者。每个对等点提供一个或多个存储服务，服务的使用者可以根据自己的需求定制一定的存储空间，系统选择合适的存储服务组合后提供给使用者。

3.1 模型工作流程

图 1 以申请者的角度描述模型简要工作流程，仅涉及主要工作模块和主要交互关系。存储服务提供者向服务注册模块中注册和发布服务；申请者使用可视化界面，提出对存储服务需求的描述；动态选取模块依据需求描述在注册的服务中进行动态选取；服务组合对选取出满足要求的服务进行组合；监控模块对组合的服务以及单个存储服务进行监控；服务评价模块根据监控情况对服务进行评价，评价的结果被记录到服务注册中心，用以修正服务提供者发布的信息。

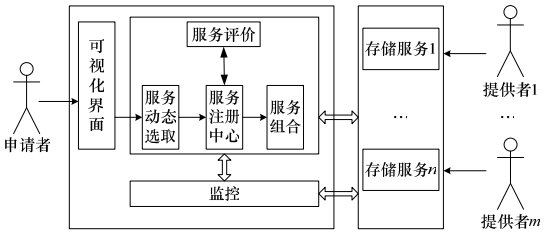

图 1 模型工作流程

3.2 模型的层次划分

模型可分为 6 层，如图 2 所示。

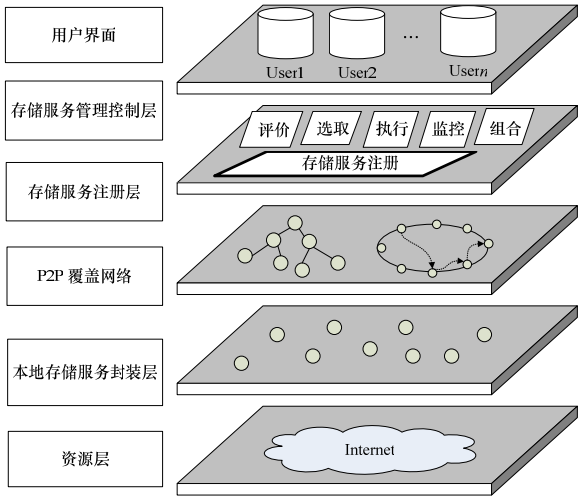

图 2 模型层次结构

各层次功能如下：

(1)资源层：对等节点按照一定的互联协议进行连接，提供存储资源。存储资源既可来源于专业的大型存储服务，也可以来自闲散的桌面机。

(2)本地存储服务封装层：每一个对等节点会根据自身的情况提供一定的存储资源。封装层采用统一的描述语言把存储资源抽象为服务，这样有利于通过统一的标准接口来管理和共享 P2P 系统中异构的存储资源。在抽象为服务时要考虑 2 个问题：1)存储资源的封装粒度。可以将每一个对等节点所提供的存储资源封装为一个存储服务，也可以设置固定大小的存储资源封装块，这样每一个对等节点将提供不同份额的存储块服务。前者封装粒度大，管理简单，系统开销小；后者封装粒度小，管理较复杂，使用灵活。2)提供标准化访问机制。如对于文件级别数据对象，可通过定义一个标准操作集合(如 create, open, close, unlink, read, write, seek, sync, stat, fstat, mkdir, rmdir, chmod, opendir, closedir, readdir)，以跨越多个文档、层次资源管理器、文件系统和数据库。

(3)P2P 覆盖网络层：本文提出的模型建立在结构化 P2P 之上，它按照一定的逻辑拓扑结构将系统中的节点互连起来，并通过路由消息使得系统中任意 2 个节点可以互相通信，一般是基于 DHT 技术[9]。模型可采用任意一种覆盖网络结构，如 chord、Xring、虚拟二叉查找树等，来建立对等节点之间的逻辑关系。在覆盖网络提供对等节点的加入、路由、查找、退出等基本功能之上，可以实现不同的应用服务，如数据管理。当然，分布式应用服务可以直接建立在物理层来管理他们的资源，但是使用覆盖网络可以更好地支持语义路由并提供信任，授权管理等。因此，覆盖网络层是整个 P2P 存储系统运行的基础。

(4)存储服务注册层：提供存储服务的注册、查找、发现。建立在覆盖网络所构建的对等节点的关系之上，该层和覆盖网络层分别实现了服务的注册与节点的注册，这两者往往可以同时进行。如，在某一种覆盖网络之上，对等点 A 采用相应的加入算法进入到 P2P 系统中。按照结构化 P2P 的思想，意味 A 在某一或某些对等点，假设为 B 的路由表中注册自己的信息。这时可在这个注册的基础上，将 A 的存储服务同时也注册在 B 的服务注册库中。采取这种方式，充分地利用了 P2P 的分散管理的方式，摆脱了一般面向服务结构中注册中心集中管理的缺陷。

(5)存储服务管理控制层：该层提供了丰富的管理和控制功能，用以在一定程度上保证存储服务的长期有效性，包括服务动态选取、服务组合、监控、评价、安全机制等。

服务动态选取是一个资源调度的问题，即存储服务的调度。根据用户的需求和现有的资源状态动态选择多个存储服务，包括首次动态选取和后续动态维护。First_Dynamic_Select()给出基于存储服务块的首次动态选取的伪代码描述。后期动态维护体现在 2 个方面：1)用户需求的变化。用户当前的存储要求可能会和前一次不同，系统必须能保证在满足当前要求的同时，又不影响前一次对系统的使用。这要求系统能动态调整对存储服务的选择，又要达到较小的调整代价。2)资源状态的变化与负载均衡。在 P2P 这种极度分散的系统中，要实现全局负载均衡是极其困难的。这里只要求达到局部负载均衡，即在已知的注册节点中选择当前负载最好的节点所提供的存储服务。但随着资源状态的变化，均衡会被破坏。如服务提供者会因为自身负载加大或是网络故障等原因导致服务质量下降，或者对等节点负载降低，资源过剩而服务质量得以提高。系统要能根据实际情况动态调整对存储服务的选择，以达到对资源的有效利用。

存储服务组合是一个任务分配问题，即根据动态选取的结果分配存储任务，将实际的存储请求交由合适的存储服务执行。为了提高系统的可用性和可靠性，需采用一定的冗余

分配方案。这里采用多粒度冗余方案，即完全副本方式和纠删码 2 种方案的结合。Multi_Granularity_Redundancy()描述其基本思想，关于首次动态选取及多粒度冗余方案的代码如下所示：

```
First_Dynamic_Select(){
  //Register_Servivce_Set:某节点所知的已注册服务集合
  //Selected_Service_Set:已选择服务集合
  //Add(x,y):向 x 集合中添加元素 y
  //N:存储服务块代表的存储块大小
  //Requestsize:请求的存储空间大小
  //Redundant:冗余系数，指每个数据对象的副本数，体现为对
//存储空间的冗余供给
  //select_suitable_service(x) :从 x 服务块集合中，根据综合参数
//选出最合适的服务块
  //因为综合参数的设定和匹配选择是一个可被独立研究的问
//题，此处并未给出具体的策略
  S:=0;
  While(S<Requestsize*Redundant)
  {
     current:=select_suitable_service(Register_Servivce_Set);
     S:=S+N;
     add(Selected_Service_Set, current);
   }
   return Selected_Service_Set
  }
Multi_Granularity_Redundancy(){
  //DataObject:数据对象
  //Size(x):x 数据对象的大小
  //allocate(x,y,z):将数据对象 x 以 y 的冗余度放置在 z 服务块集
//合中
  //N:存储服务块代表的存储块大小
  //Erasure_code(x,m,n):对数据对象 x 使用(m,n)的纠删码，即将
//x 分成 n 块，仅需>=m 块可恢复到原 x，该函数返回一个数据对象
//数组 dataobj[n]
  If (Size(DataObject)<N) then allocate(DataObject,R, Selected_
Service_Set)
  else {
        Dataobj[n]=Erasure_code(DataObject,m,n)
        for(i=0;i<n; i++)
           allocate(Dataobj[i],R,Selected_Service_Set)
     }
}
```

监控用于监视组合服务以及单个存储服务。监视结果用于动态选取以及修正注册信息。

评价除了对监控情况进行分析，作为动态选取和修正注册信息的依据，还可作为一个扩展的功能，用于 P2P 系统的信用机制中。

除此之外，还需要提供一定的安全机制、事务管理、审计和日志等。

(6)用户界面层：向用户提供一个可视化的图形界面，方便用户的使用。

4 模型层次功能在对等节点的部署

每一个对等点都是一个相对独立的实体，它可以根据自己的当前状态和意愿来决定是否对外部的服务请求进行响应。为了获得或实施指定的存储服务，每个对等点实体必须完成如下任务：搜索目标对等点实体，响应从其他对等点发来的服务请求，动态选取，组合，监控，评价，安全认证等。而这些任务正是模型层次结构图中所描述的，因此，图 2 中的各层次功能应该存在于所有的对等点实体中。这样一个对等节点扮演了多个角色，主要体现为系统管理者、服务提供者和服务使用者。图 3 给出了一个对等节点内部的组织方式，采用 3 层结构设计。

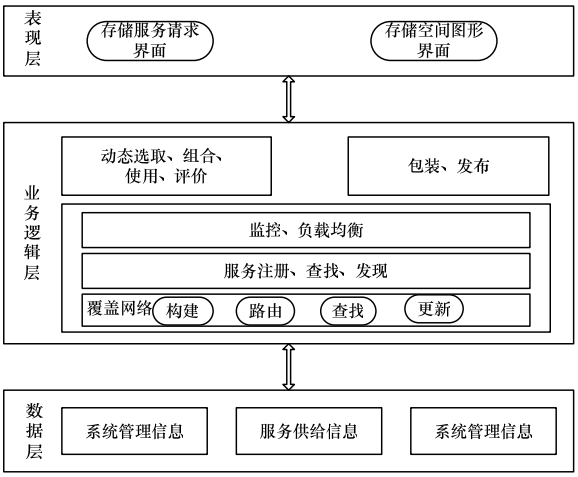

图 3　对等节点结构

数据层记录对等点以 3 个不同角色所使用的信息：系统管理信息，服务供给信息以及服务使用信息。系统管理信息体现为覆盖网络路由信息和存储服务注册库。在构建覆盖网络中形成路由信息，记录了各对等节点之间的关系，具体格式依赖于特定的覆盖网络，是整个系统得以正常运作的基础。存储服务注册库即服务的注册中心，其注册的服务与覆盖网络路由信息具有对应关系，如采用服务块的封装方式，则每一个节点的路由信息和其注册的服务信息是一对多的关系。服务供给信息体现为监控时产生的自身状态信息、存储日志和存储状态表，用以作为服务的提供者记录自身所提供的存储情况，并依据路由关系以某种更新方式反馈到对应的对等节点，以更新其管理信息。服务使用信息包括监控时产生单个及组合存储服务日志和存储服务评价信息。对等节点之间数据层的关系如图 4 所示。

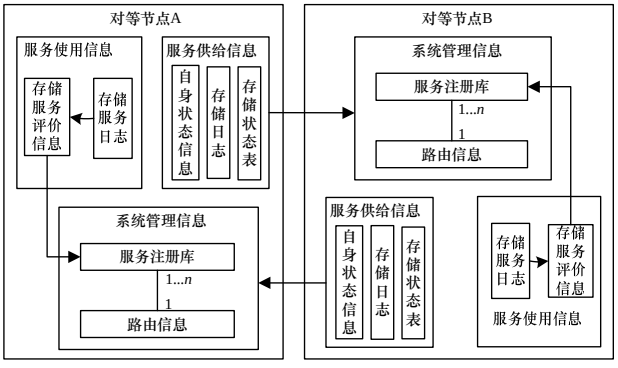

图 4　对等节点数据层关系

业务逻辑层主要实现层次结构中的各功能，同样提供 3 大类功能。作为管理角色，维护覆盖网络和服务注册库，监控系统状态，动态反映到数据层，并以此作为负载均衡的依据。作为服务提供者主要进行自身存储服务的包装，依据路由关系发布到相应的对等节点中。作为服务使用者，进行服务的选取、组合和最终的使用。

表现层针对于用户，为用户提供一个良好的图形环境，屏蔽掉底层的操作细节，以更好地使用系统。

(下转第 96 页)