

并行网络文件系统 PNFS 研究

张 媛^{1,3} 于冠龙² 阚云鹤¹

(65042 部队 沈阳 110001)¹ (海军驻沈阳地区航空军事代表室 沈阳 110031)²
(国防科学技术大学计算机学院 长沙 410073)³

摘 要 从解决高性能计算机 I/O 瓶颈面临的问题着手,首先详细分析并设计了并行网络文件系统——PNFS 的结构、存储机制、管理机制和工作机制,然后对并行网络文件系统 PNFS 的原型系统进行了性能测试,从而验证了并行网络文件系统 PNFS 可以有效地解决传统网络文件系统在可扩展性、可用性和性能上存在的问题,能够为高性能文件系统的建立提供一种行之有效的解决方法。

关键词 并行网络文件系统,部署,网络存储,性能测试

中图法分类号 TP393 **文献标识码** A

Study of Parallel Network File System

ZHANG Yuan^{1,3} YU Guan-long² KAN Yun-he¹

(The 65042nd Unit of PLA, Shenyang 110001, China)¹

(Naval Aeronautic Military Representatives Office in Shenyang, Shenyang 110031, China)²

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)³

Abstract Beginning with the problems of solving the I/O bottleneck of high performance computer. First, this paper mainly analyzed and designed the architecture, access mechanism, management mechanism and operating mechanism of parallel network file system-PNFS. Second, we made the test of PNFS, and the results show that PNFS can resolve the problems of traditional file systems about scalability, availability and performance and provides an effective method for building high performance file system.

Keywords PNFS, Layout, Network-storage, Performance evaluation

1 引言

随着高性能微处理器、高速互连网络和高性能分布式计算的快速发展,以及人们对计算能力需求的日益增长,传统的基于主机的存储架构已成为性能瓶颈,不能满足集群系统的需求。集群的存储系统必须有效解决两个主要问题:(1)提供共享访问数据,便于集群应用程序的编写和存储的负载均衡;(2)提供高性能的网络存储,在 I/O 级和数据吞吐率方面能满足成百上千台规模的 Linux 集群服务器聚合访问的需求。而传统的网络文件共享是通过网络文件系统 NFS(Network File System)^[1]来实现的。虽然 NFS 支持异构系统,并可以在不同硬件平台和操作系统的计算机之间实现文件共享和数据资料的集中管理,但其协议开销高、带宽低、延迟大,不利于在高性能集群中应用,因此迫切需要一种更好的资源共享方式。本文研究并设计实现了一种并行的网络文件系统 PNFS(Parallel Network File System),它是对网络文件系统 NFSv4 的扩展,保持了原有网络文件系统的优点,同时可使元数据服务器与数据服务器相分离,大大降低了由于频繁访问元数据服务器所带来的开销,并将文件存储在多个存储服务器当中,同时可为多个客户端提供并发的访问方式。

本文首先对 PNFS 做了简要介绍;然后,对其原型系统进

行了设计,并分析了其软件结构;最后,在千兆网络环境下使用 Linux 系统命令 dd 对并行网络文件系统 PNFS 进行了相关测试。在此基础上提出了在高性能计算机环境下通过部署 PNFS 可以有效地解决高性能计算机 I/O 的瓶颈问题。

2 PNFS 介绍

2.1 PNFS 体系结构

并行网络文件系统 PNFS 由 3 部分组成,即元数据服务器(MDS, Meta Data Server)、数据服务器(DS, Data Server)、PNFS 客户端(Client)。其结构如图 1 所示。

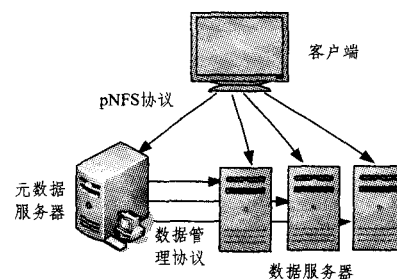


图 1 PNFS 结构

PNFS 是对 NFSv4 协议^[4]的扩展,允许客户端对存储设

张 媛(1982—),女,硕士,工程师,主要研究方向为网络体系结构、高性能网络存储, E-mail: ruby820111@126.com; 于冠龙(1983—),男,工程师; 阚云鹤(1981—),女,硕士,助理工程师。

备直接进行访问以减少由于频繁访问元数据带来的瓶颈问题。当原有单个 NFS 服务器上的数据被存放在多个或者存放在拥有高吞吐量的存储设备上时,这种效果尤其明显,它可以显著地提高文件访问的性能。PNFS 在 NFSv4 的基础上加入了新的操作^[4]来管理数据的定位和信息——layout。layout 与 NFSv4 的代理机制在管理上有一些相似之处(例如,它们均可以回调和废除)。然而,在操作上它们又截然不同。当一个客户拥有了 layout,他就可以通过 layout 中的定位信息对数据服务器进行直接访问。

PNFS^[2]是一个透明的文件系统,将文件数据以条块化的方式存储到多个数据服务器上,提供了比传统 SAN 的“块共享”更为灵活的共享访问方式。元数据服务器通过 layout 指导实际的文件 I/O 请求到数据服务器上,数据服务器管理在基于块设备上的物理存储。客户端可以通过网络透明地读取数据服务器中的数据。采用元数据和存储数据相分离的技术可以充分分离计算和存储资源,使得客户端计算机可以专注于用户和应用程序的请求;数据服务器和元数据服务器专注于读、传输和写数据。存储服务器端的数据备份和存储配置以及存储服务器扩充等操作不会影响到客户端,存储服务器和元数据服务器均不会成为性能瓶颈。

2.2 PNFS 设计目标

并行网络文件系统 PNFS^[2,3,5]的设计目前并没有统一的标准,但要使网络文件系统真正高效、实用,就需要通过多种途径加以解决,使 PNFS 实现以下目标:

- 将数据和控制信息(元数据)分离开,包括标准的元数据协议和针对不同对象的存储访问协议(面向文件、面向块、面向对象等)。
- 将文件分割存放在多个服务器上,实现高聚集的带宽、可以并行访问单个文件的多个数据源、支持多个后端存储协议。
- 支持标准化客户端。标准化客户端在 IETF NFSv4^[4]的最新版中提出,它既可以支持 PNFS 的 NFSv4,又可以支持常规 NFSv4。
- PNFS 的元数据协议采用标准的 NFSv4,数据管理协议并没有标准化的规定。
- 可伸缩性。扩展了客户端的并行性,提供了文件的分布结构。
- 互操作性。客户端要兼容标准的 NFSv4,服务器必须支持标准的 NFSv4 客户端,存储访问协议必须是良定义的。
- 多个客户端可以并发访问数据。

3 PNFS 原型设计

3.1 PNFS 原型体系结构

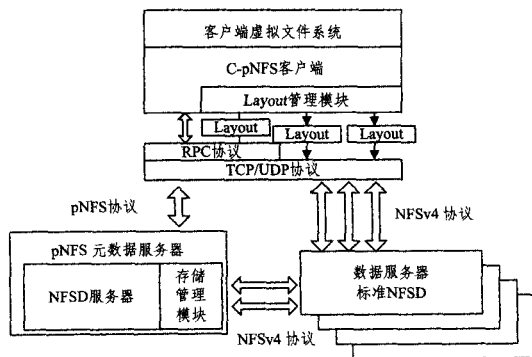


图 2 PNFS 原型体系结构

并行网络文件系统 PNFS 作为传统 NFS 的功能扩展,除了支持 RFC3530^[4]定义的 NFSv4 功能以外,还通过 layout 驱动程序支持并行的高速文件数据存储功能。PNFS 在 Linux 内核中的地位与传统 NFS 完全相同,作为一个独立的内核模块,位于 VFS 的下层,向 VFS 提供网络文件系统的远程访问功能。每个 layout 驱动程序作为一个独立的内核模块,启动时动态地向 PNFS 注册,通过增加新的 layout 驱动程序,使 PNFS 同时支持多种下层存储协议,并可以灵活扩展。原型体系结构如图 2 所示。

PNFS 是用户访问并行网络文件系统的入口,用户通过 mount 命令将元数据服务器输出的目录树挂载在本地文件系统的某个挂载点上。这时 PNFS 作为内核模块被加载到内核空间中,解析用户的挂载请求,向 PNFS 元数据服务器注册,并获取元数据服务器的根级 inode 结点和 dentry 结点信息。用户将网络文件系统挂载点范围内的文件、目录等操作通过 VFS 文件系统抽象层调用到 PNFS 模块中,由 PNFS 转换为对元数据服务器或者数据服务器的操作。其中 PNFS 的主要功能包括:

- 向 VFS 注册各种操作的回调函数表;
- 使用 PNFS 协议与元数据服务器通讯,进行目录、文件属性等信息的访问;
- 使用 NFS 协议与数据服务器通讯,主要进行文件的读、写等数据块操作。

3.2 PFS 原型软件结构

PNFS 软件系统主要由 NFS 协议扩展模块、layout 注册管理模块和 layout 驱动程序模块组成,PNFS 软件结构如图 3 所示。

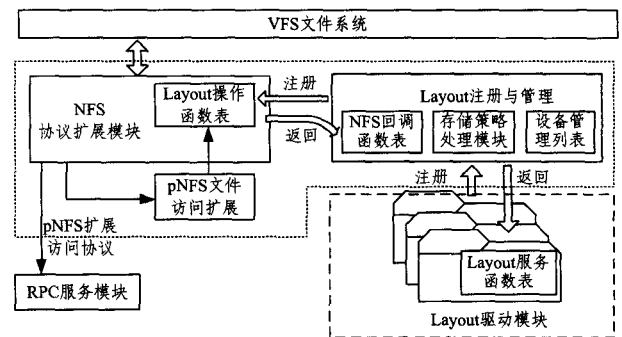


图 3 PNFS 软件结构

其中,每个 layout 驱动程序模块作为独立的内核模块来实现,可以动态地加载或者卸载,使得新增 layout 功能不需要 PNFS 的任何修改。NFS 协议扩展模块和 layout 注册管理模块作为向 VFS 提供网络文件系统功能服务的内核模块存在,也可以动态地加载或者卸载。下面分别介绍各个模块的具体构成及主要功能。

3.2.1 NFS 协议扩展模块

NFS 协议扩展模块是传统 NFSv4 客户端^[4]的功能扩展,向 VFS 文件系统提供对并行网络文件系统的访问。主要的功能扩展有:PNFS 文件访问过程的扩展和 layout 信息访问的扩展。

NFS 客户端在文件操作时,直接调用 RPC 服务模块,将文件操作请求发送到远端的 NFS 文件服务器中。NFS 协议扩展模块修改 NFS 客户端的文件操作过程,将其重定向到对 layout 操作函数表的调用,由 layout 驱动程序来具体执行文件中数据块的读写。layout 驱动程序需要元数据服务器上存

储的当前文件的 layout 信息,来进行读取数据块的偏移量到对应数据服务器的映射操作。因此在 NFS 协议扩展模块与元数据服务器之间的交互协议中,需要扩展对 layout 信息的支持。主要扩展的功能包括:layout 设备列表的获取、文件 layout 信息的获取、文件 layout 分配与提交等。主要数据结构如下:

```

struct FileLayout_devices{
    u32 dev_id; /* 数据服务器标识符 */
    struct sockaddr_in dev_addr; /* 数据服务器的 IP 地址 */
    unsigned int pathlen; /* 数据服务器 NFSD 输出的路径信息长度 */
    char datasrv_export_path[MAXNAMSIZE]; /* 数据服务器 NFSD 输出的路径信息 */
    atomic_t count; /* 数据服务器引用计数 */
    unsigned int datafilenamelen; /* 根目录下的数据文件名称长度 */
    char datafile_name[MAXNAMSIZE]; /* 根目录下的数据文件名称 */
    struct dentry * p_root; /* 数据服务器输出的根目录 */
    struct nfs_fh * fh; /* 数据文件的文件句柄 */
    struct nfs_fattr * fattr; /* 数据文件的文件属性信息 */
    struct nfs_open_context * context; /* 文件打开时的上下文信息 */
    struct rpc_clnt * rpc_clnt; /* RPC 调用的客户端数据结构 */
};

struct FileLayout_devicelist {
    atomic_t count; /* 引用计数 */
    unsigned int num_devs; /* 包含的数据服务器的数量 */
    rwlock_t dev_lock; /* 操作锁 */
    struct FileLayout_devices devs [NFS4_PNFS_DEV_MAXCOUNT]; /* 数据服务器信息列表 */
};

enum nfs4_file_data_extentstate {
    READ_WRITE_STATE=0; /* 数据读/写状态 */
    READ_STATE=1; /* 数据只读状态 */
    INVALID_STATE=2; /* 数据无效状态,新分配或处于修改状态的数据单元 */
    NONE_STATE=3; /* 数据无效状态,无实际分配物理空间 */
};

struct nfs4_file_data_extent {
    u32 dev_id; /* 数据服务器标识符 */
    u64 block_id; /* 数据服务器块标识符 */
    nfs4_file_data_extentstate es; /* 数据块状态 */
};

struct nfs4_filelayout {
    int uncommitted_write;
    loff_t last_commit_size;
    u64 layout_id;
    u64 file_offset; /* 文件偏移量 */
    u64 length;
    u32 iomode;
    u64 file_size; /* 文件大小 */
    u64 stripe_unit; /* 分片大小 */
    striptype4 stripe_type; /* 分片类型 */
    unsigned int num_devs; /* 包含的数据服务器的数量 */
    struct nfs4_file_data_extent * data_info[NFS4_PNFS_MAX_DATA_NUMBER];
};

```

3.2.2 layout 注册管理模块

layout 注册管理模块管理 layout 驱动程序模块的注册和

注销。将这些 layout 驱动程序注册的操作函数表虚拟为一个操作函数表,在向 NFS 协议扩展模块注册时,将该操作函数表提供给 NFS 协议扩展模块,在执行文件操作时进行调用。注册时,NFS 协议扩展模块返回一个回调函数表,向 layout 驱动模块提供与 NFS 操作相关的服务函数。layout 注册管理模块的主要构成包括:

- layout 操作函数表:向 NFS 协议扩展模块提供与 layout 相关的操作函数,如 layout 信息设置,设备信息设置,文件读、写处理过程等;
- NFS 回调函数表:主要向基于标准 NFS 协议的 layout 驱动模块提供 NFS 协议的服务函数、layout 驱动程序操作结束时的通知函数等;
- 存储策略处理模块:提供各种标准存储策略的服务函数,如 RAID 中偏移量到设备的映射计算、RAID 校验和的计算、RAID 信息恢复等;
- 设备管理列表:对应一种类型的 layout 驱动程序,元数据服务器维护一个可用的数据服务器列表。NFS 协议扩展模块从元数据服务器获取使用的设备列表,并设置到 layout 注册管理模块中。layout 驱动程序通过查询该列表,完成设备号到设备地址的映射过程。

3.2.3 layout 驱动程序模块

layout 驱动程序模块向 layout 注册管理模块注册 layout 类型和相应的操作函数表。主要功能包括:解析具体的 layout 信息描述,提供文件数据块的读写处理过程,将文件操作映射为对多个数据服务器的并发操作,提供较高的聚合处理性能。

a) layout 驱动程序注册流程

函数原型:

```

register_layoutdriver(unsigned int layoutclass, char *
drvname, struct layoutdriver_io_operations * ops, struct layoutdriver_policy_operations * policy)

```

执行流程:

- 在驱动程序管理结构中遍历,判断是否存在相同的 layout 类型;
- 如果存在,则返回失败;
- 分配一个未使用的 driver 数据项;
- 对数据结构中各个域进行赋值;
- 返回。

b) layout 驱动程序注销流程

函数原型:

```

PNFS_unregister_layoutdriver(unsigned int layoutclass)

```

执行流程:

- 遍历驱动程序管理结构,判断是否存在相同的 layout 类型;
- 如果不存在,则返回失败;
- 调用驱动程序的 uninitialize_mountpoint 处理函数;
- 将该项各个域清空;
- 返回。

c) Layout 驱动程序初始化流程。

函数原型:

```

struct PNFS_mount_type * (* initialize_mountpoint)
(struct super_block * sb, void * devinfo);

```

参数说明:sb 为超级块对象;Devinfo 为设备列表描述信

息。

执行流程:

- 对 devinfo 设备信息列表进行解码;
- 对每个设备信息,执行如下操作:
 - 1)按照服务器地址,构造服务器地址结构;
 - 2)创建对应的 RPC Client 数据结构并保存;
 - 3)调用 nfsroot_mount,登录数据服务器,获取根文件句柄;
 - 4)调用 lookup 操作函数,获得根目录下的数据文件句柄;
 - 5)获得根目录下的数据文件属性信息;
 - 6)打开并读写数据文件,验证文件的正确性;
 - 7)保存数据文件的访问句柄。
- 初始化完成。

4 性能测试与分析

4.1 测试环境

测试使用设备如表 1 所列,操作系统为 Linux,内核版本为 Linux-2.6.37,测试工具为 Linux 系统命令 dd。

表 1 测试设备列表

元数据服务器 MDS(1 台)	
设备类型	浪潮英信 NF280
CPU	Intel Xeon 3GHz
内存容量	1GB
硬盘类型	Ultra 320 SCSI
存储服务器、客户端(共 6 台)	
设备类型	PC 机
CPU	Intel Pentium 4 2.6GHz
内存容量	512M
硬盘类型	SATA 硬盘
网络设备	
千兆交换机、千兆网卡、光纤	

4.2 性能测试

在测试环境正确配置运行的基础上,进行了如下的性能测试。测试中为了消除系统缓存策略对系统性能测试结果的影响,每组块读/写测试中每个客户端读写文件的长度均大于节点内存大小的两倍,在本次测试中测试文件的大小均为 4G。

4.2.1 磁盘性能测试

磁盘性能测试中,测试了每台 PC 机的硬盘写入速率,如表 1 所列 PC 机为 SATA 硬盘。测试后,得到 6 台 PC 机硬盘写入速率,如表 2 所列。

表 2 PC 机本地硬盘性能测试数据表

测试文件 4G	PC-1	PC-2	PC-3	PC-4	PC-5	PC-6
运行时间 (s)	1m50.8	2m9.4	1m59.1	2m11.3	1m54.6	2m5.3
写入速度 (MB/s)	36.95	31.63	34.4	31.2	35.73	32.68

接下来的测试中,连接存储设备使用的是千兆网络,千兆网络有 100MB/s 以上的通信带宽,而从测试结果来看 PC 机硬盘仅能提供 31.2MB/s~36.95MB/s 的数据传输率,因此可以认为,所测得数据的差异能真实反映文件系统的性能^[6]。

4.2.2 PNFS 读/写性能测试

测试中客户端结点读/写 4G 的文件,每次测试读/写不同大小的数据块,数据块大小分别为:1k、4k、8k、16k、32k、64k、128k、256k、512k、1024k、2048k、4096k、8096k。测试分别针对 1 个数据服务器、2 个数据服务器、3 个数据服务器、4 个

数据服务器、5 个数据服务器 5 种不同的情况进行。测试结果如图 5、图 6 所示。

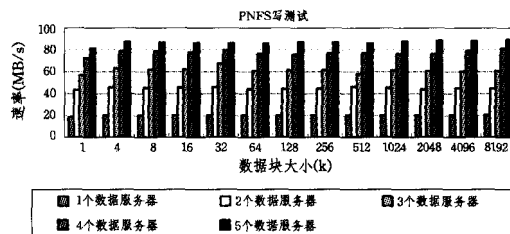


图 5 块写带宽

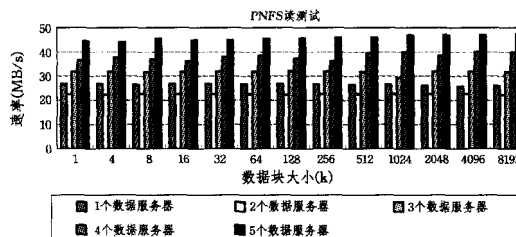


图 6 块读带宽

图 5 和图 6 显示了在读/写块不同、客户端为 1 个而数据服务器的数量从 1 到 5 递增时,系统读/写带宽随着数据服务器数目的增加而变化的柱形图。可以看出,在本测试中当数据服务器的数量达到 5 个时客户端的读/写速率可达到最大值,分别为:读速率 47.556MB/s,写速率 87.689MB/s。

4.2.3 性能测试分析

明显可见,PNFS 的块读速率和块写速率随着数据服务器数量的增长,其聚合带宽一直在上升且几乎呈现线性的增长方式,表现出了其良好的可扩展性。可以预计随着数据服务器数量的增长,块读和块写聚合带宽将持续上升,直到达到网络环境中所能提供的最大带宽。

然而,从图 6 中可以看到,由于磁盘 cache 的影响使得 PNFS 在数据服务器数量为 2 时反而比数据服务器数量为 1 时的读速率慢,由此诸如磁盘 cache 等多种因素都会对并行网络文件系统的性能带来很大的影响,这也是将来需要我们进一步研究的方向。同时,比较 PNFS 块读/写速率不难发现,PNFS 的读速率在存储服务器数量大于 1 时一直低于其块写速率,这是由于在 512M 内存的机器上测试 4G 的文件时,写操作能够部分利用系统缓存,而读操作则无法利用。

PNFS 结构中客户端获得高聚合带宽主要是因为,客户得到元数据以后可以并发访问多个存储设备,同时由于多个存储设备的并行工作使得在存储集群中单个设备的传输时延和磁盘寻道时间被隐藏起来了,因此降低了客户端对整个文件的平均访问时延,提高了系统的整体性能。

由此可见,PNFS 为客户端提供了并行访问文件数据的能力,提高了网络文件系统的性能。PNFS 的提出消除传统网络文件系统局限于单个服务器文件访问的瓶颈问题,实现了用于并行数据访问的标准化客户端接口。PNFS 系统运行于 Linux 文件系统上,基于当前的通信、磁盘的技术水平,充分利用了层次型系统中多处理器强大的并行处理能力,提供了较高的 I/O 服务带宽和较低的访问延迟,能较好地满足并行科学计算应用的 I/O 需求。

结束语 并行网络文件系统 PNFS 的研究是探索未来 P 量级存储技术的基础,也是大规模并行处理系统中解决 I/O 瓶颈的主要途径之一。本文对 PNFS 网络文件系统做了总体的介绍,并且对 PNFS 原型系统的实现进行了描述。下一步

研究工作将从 PNFS 的安全性、实现方法和性能分析等角度着手做更进一步的探究。总之,并行网络文件系统技术是一个方兴未艾的研究领域,该领域中还有许多问题尚未解决,并且随着新技术的出现和新应用的需求,新的问题和难点还会层出不穷。希望本文的工作能对并行网络文件系统的研究有一定的帮助。

参考文献

[1] Sun Microsystems. NFS; Network File System Protocol Specification[Z]. RFC 1094, March 1989

[2] Black D. PNFS Block/VolumeLayout [OL]. ftp://www.ietf.org/internet-drafts/draft-black-PNFS-block-01.txt, 2005-07
 [3] Goodson G. NFSv4 PNFS Extinctions [OL]. ftp://www.ietf.org/internet-drafts/draft-ietf-nfsv4-PNFS-00.txt, 2005-10
 [4] Shepler S, Callaghan B, Robinson D, et al. Network File System (NFS) version 4 Protocol [Z]. RFC 3530, April 2003
 [5] Shepler S. NFSv4 Minor Version 1 [OL]. ftp://www.ietf.org/internet-drafts/draft-ietf-nfsv4-minorversion1-02.txt, 2006-03
 [6] 曹立强,熊劲. Global File System 性能评测与分析 [J]. 高性能计算技术, 2003, 165(12): 2-3

(上接第 373 页)

(6) VirtualMachineImage: 虚拟机镜像表, 保存着虚拟机镜像的大小、操作系统、应用程序等虚拟机镜像信息。

(7) ScheduleDomain: 调度域表, 保存着调度机器的配置、授权、IP 等调度域信息。

(8) SecurityGroup: 安全组表, 保存着调度域、授权、安全级别等安全组信息。

(9) Storage: 存储表, 保存着存储介质的大小、读写速度等存储信息。

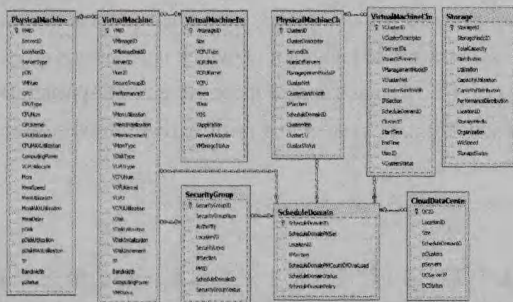


图 2 云资源管理数据库

2.3 云资源管理数据库的使用

通过对云资源管理数据库操作, 可以方便灵活地查询、更新、管理云资源。

1. 调度域查询

Select * From ScheduleDomain;

ScheduleDomainID	ScheduleDomainName	LocationID	IPSection	ScheduleDomainName	ScheduleDomainName	ScheduleDomainName
1	1-30	1	192.168.1.2-15...	0	OK	Loadbalance
2	1-20	2	192.168.2.2-15...	0	OK	Loadbalance
3	1-30	3	192.168.3.2-15...	0	OK	Heatbalance
NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 3 调度域查询

2. 更新物理机信息

Update PhysicalMachine Set LocationID = '2' Where PMID=5;

PMID	ServerID	LocationID	ServerType	OS	VMWare	CPU
1	1	1	Windows	WindowsServer...	5	3.1
2	2	1	Windows	WindowsServer...	5	3.1
3	3	1	Linux	RedhatES	5	3.1
4	4	1	Linux	RedhatES	5	3.1
5	5	2	Windows	WindowsServer...	5	3.1
6	6	2	Linux	RedhatES	5	3.1
7	7	2	Linux	UbuntuServer 11	5	3.1
8	8	2	Windows	WindowsServer...	5	3.1
9	9	3	Linux	UbuntuServer 11	5	3.1
10	10	3	Linux	UbuntuServer 11	5	3.1
11	11	3	Windows	WindowsServer...	5	3.1
12	12	3	Windows	WindowsServer...	5	3.1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 4 更新物理机信息

3. 增加虚拟机

Insert Into VirtualMachine Values('5', '15', '3', '2', '2', '4', ...);

VMID	VMImageID	VMImageSetID	ServerID	UserID	SecureGroupID	Perf
11	5	10	3	2	2	4
12	5	11	3	2	2	4
13	5	12	3	2	2	4
14	5	14	3	2	2	4
15	5	15	3	2	2	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5 增加虚拟机

4. 删除指定虚拟机

Delete From VirtualMachine Where VMID=12.

VMID	VMImageID	VMImageSetID	ServerID	UserID	SecureGroupID	Perf
10	5	10	3	2	2	4
11	5	11	3	2	2	4
13	5	12	3	2	2	4
14	5	14	3	2	2	4
15	5	15	3	2	2	4

图 6 删除指定虚拟机

结束语 云计算可帮助医疗机构实现 IT 资源与应用的整合, 给医疗机构传统的 IT 信息化运维环境带来革命式的变革, 可有效地解决传统环境下的系统分散管理、应用维护复杂、灾难恢复困难、安全访问不足等多方面的问题与挑战。云计算应用到医疗机构中还可以显著减少服务器数量和数据中心的能耗, 为所有的应用提供高可用的架构平台, 用低成本实现无中断的计划内停机与快速恢复的计划外停机, 并大大降低整合的难度, 加速系统的部署。文中介绍的关于采用云资源数据库的云资源管理方式目前已经在部分医疗机构中得到部署, 其可行性已经得到肯定。

参考文献

[1] Danielson, Krissi. Distinguishing Cloud Computing from Utility Computing [Z]. Ebizq.net, 2010-08-22
 [2] Gartner Says Cloud Computing Will Be As Influential As E-business [Z]. Gartner.com, 2010-08-22
 [3] Gruman, Galen. What cloud computing really means [Z]. InfoWorld, 2009-06-02