

ADHSNT 自发现层次存储网络拓扑模型设计

刘庆江

(哈尔滨师范大学阿城学院计算机与信息系 哈尔滨 150301)

摘要 当存储网络的数据节点达到数以百计的时候,手工设计所产生的结果往往不能很好地适合需求。自动生成存储网络的拓扑设计,可以处理这些问题。因此本文在研究了广域网络存储虚拟化形成的拓扑结构的基础上,指出现有简单层次模型的不足,提出了自动发现层次式存储网络拓扑模型 ADHSNT。结合存储网络的实际特点提出 IP 路径合并算法,并进行拓扑结构规整。

关键词 广域网,网络存储,层次式存储网络拓扑

Design of Auto-Discover Hierarchical Storage Network Topology

LIU Qing-Jiang

(Department of Computer and information, Acheng College, Haerbin Normal University, Haerbin 150301)

Abstract The result of manual design will not always adapt the requirements well when the data nodes of the network storage reach hundreds in amount. Topology design which can build the network storage automatically will solve these problems. After research the Topology structure formed by the Virtual Storage of Wide Area Network, this paper not only detects the deficiencies of existing simple layer module, but also points out the ADHSNT (Automatic Discover Hierarchical Storage Network Topology). Combining the actual traits of the network storage, IP path coalition arithmetic has been brought forward, and adjusted the Topology structure.

Keywords Wide area network, Network storage, Hierarchical storage network topology

1 引言

网络存储已经被普遍认为是继个人计算机、网络之后的“数字技术的第三次浪潮”。存储系统大致可以分为三种形式^[1]:直接附接存储(Direct Attached Storage, DAS)^[2],网络附接存储(Network Attached Storage, NAS)^[3]和存储区域网络(Storage Area Network, SAN)^[4],其中 SAN 从 FC SAN 向 IP SAN 的发展则基本上代表了网络存储的广域化方向。同时随着网络计算研究的深入,基于广域虚拟化的存储网络被认为是存储网络下一步发展目标,也被引起广泛关注。计算网络所需的存储 I/O 规模要比现今大多数存储系统的规模大得多。大型强子对撞机(LHC)计算网络将 87 个计算中心的大型集群和存储系统连接,用于量子物理研究。而 TeraGrid 则将 9 个大型超级计算中心的电脑连接起来用于科学研究。

拓扑结构是网络全局性能优化的重要依据。在网络环境中,许多全局优化算法效果不佳的一个重要原因是优化路径时没有充分借助网络拓扑结构。对于网络存储环境,更是有着数据量巨大、实时性和效率要求高等特点,拓扑结构对性能优化方面的影响尤为重要。本文从广域 IP 存储网络整体到 FC SAN fabric 局部,着眼于存储网络层次拓扑结构,针对简单的多层结构不能满足有效全局网络性能优化的要求,在结合实际网络存储特点的基础上提出层次网络拓扑,并进行拓扑结构规整。

2 传统层次拓扑结构模型的不足

对于大多数广域网络环境,尽管拓扑结构在某些局部是不规则的,但在总体上都呈现出一定的层次特征,它们在拓扑结构上的共性是层次性,适合于用层次结构来描述。许多全

局网络性能优化系统,例如 MAGPIE^[5]、ECO^[6]等,在优化时都采用了层次拓扑结构。而其他拓扑结构,例如 LOOP、MESH 等,只是突出了某个局部的特殊性,不能描述层次式连接的共性。因此在广域拓扑结构模型方面的工作围绕层次结构展开,而在局部性方面的工作则充分地考虑到 SAN 的特殊性。

MAGPIE 采用的拓扑结构模型是自顶向下的两层结构。它把整个网络分成两层(不包括终端节点):第一层为广域网,连接各个园区网;第二层为园区网,连接终端节点。采用这种模型,图 1(a)所示的环境就描述成图 1(b)。在图 1(b)中,第二层直接和终端节点连接,没有把图 1(a)中的第二、第三层连接信息区分开。这种模型适用于第一层的延时非常大、带宽非常低的环境,在这种环境中,第一层的延时、带宽与其他各层相比差距大到可以忽略不计其他各层,从而没有必要再在中间进一步划分。对于不满足这样条件的环境,这种模型是不完备的,具有很大局限性。

ECO 采用的拓扑结构模型是自底向上的两层结构。它也把整个网络分成两层:第二层为子网,连接终端节点;第一层为子网互连;采用这种模型,图 1(a)所示的环境就描述成图 1(c)。在图 1(c)中,第二层子网直接互连,没有把图 1(a)中的第一、第二层连接信息区分开。这种模型适用于子网之上各层的延时、带宽差距不大的环境。对于不满足这样条件的环境,这种模型也是不完备的。

算法^[7]采用的拓扑结构模型为多级的层次结构,这种结构由人工描述获得,并且在每一层定义该层的主节点。采用这种模型,图 1(a)所示的环境就描述成图 1(d)。对于这种简单的多层结构,主节点的存在限制了优化时的节点选择性,不能满足有效全局网络性能优化的要求。

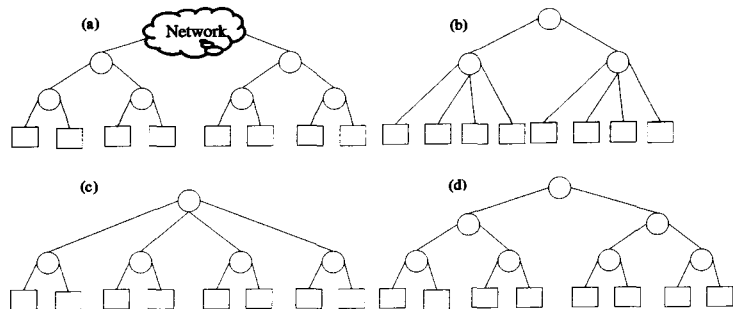


图1 相关拓扑结构模型

3 ADHSNT 型层次拓扑结构模型

在 MAGPIE 中,网络的层次结构依靠手工生成,用起来不方便;在 ECO 中,层次结构自动生成:首先测试终端节点间的消息传递延时,然后根据延时把节点划分为不同的子集群,从而形成两层的拓扑结构;这种方法不足之处在于获得的拓扑结构只有两层。此外,有许多拓扑结构发现算法^[8,9],它们的侧重点在于测试方法,例如 DNS、SNMP、Ping、Traceroute 等,而没有涉及如何抽象出层次特征并生成层次拓扑结构。因此我们提出了一种自动发现层次拓扑结构的算法,包含以下四个过程:IP 路径测试、平均延时计算、IP 路径合并以及拓扑结构树生成。

3.1 IP 路径测试

定义 1(IP 路径) 指 IP 报文从信源 N_i 发送到信宿 N_j 过程中所经过的节点序列,记为 P_{ij} 。例如图 2 中从终端节点 N_0 到 N_6 所经过的网络设备节点依次为 S_1 和 S_6 ,记为 $P_{06}(N_0, R_1, R_6, N_6)$ 。

定义 2(边延时) 指 IP 路径中相邻节点之间 IP 报文的 Ping-Pong 延时。例如在上述 IP 路径 P_{06} 中,边 (S_0, S_2) 的延

时为 IP 报文从节点 S_0 到 S_2 ,再从 S_2 返回到 S_0 的延时,记为 $L_{S_0S_2}$ 。

已知 n 个终端节点 $\{N_i, 0 \leq i \leq n-1\}$,首先在各个终端节点测试该节点到其他终端节点之间的 IP 路径以及路径中各边的延时。为方便起见,假设数据流量往返路径是一致的,仅测试 $n \times (n+1)/2$ 条 IP 路径,如图 3 所示。

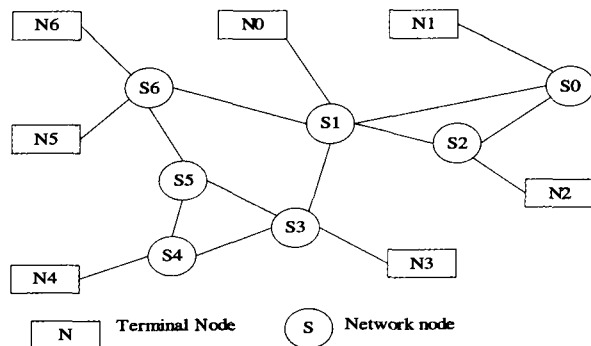


图2 广域存储网络拓扑

	N_0	N_1	N_2	N_i	N_{n-2}	N_{n-1}	Path count
N_0		P_{01}	P_{02}	P_{0i}	$P_{0(n-2)}$	$P_{0(n-1)}$	$n-1$
N_1			P_{12}	P_{1i}	$P_{1(n-2)}$	$P_{1(n-1)}$	$n-2$
N_2				P_{2i}	$P_{2(n-2)}$	$P_{2(n-1)}$	$n-3$
N_i					$P_{i(n-2)}$	$P_{i(n-1)}$	$n-i-1$
N_{n-2}						$P_{(n-2)(n-1)}$	1
N_{n-1}							0

图3 终端节点的测试路径

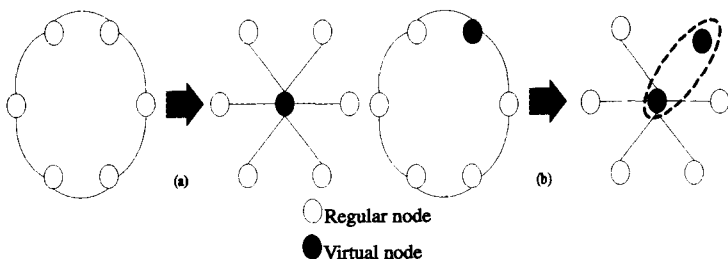


图4 IP 路径合并的环处理

3.2 平均延时计算

定义 3(节点延时) 在一条 IP 路径中,与某个节点相邻边的延时之和称为该节点的延时。例如在 IP 路径 P_{06} 中,节点 S_1 的相邻边为 (N_0, S_1) 和 (S_1, S_6) ,因此节点 S_1 的延时 $L_{s1} = L_{N_0s1} + L_{s1s6}$ 。

定义 4(节点平均延时) 一个节点可能在多条 IP 路径

中出现,相应的就有多个节点延时,它们的平均值称为该节点的平均延时。

计算各个节点的平均延时,目的是在后面的步骤中确定平均延时最大的节点为拓扑结构树的根节点。

3.3 IP 路径合并

合并上述 $n \times (n+1)/2$ 条 IP 路径,生成无环拓扑图。为

此设置 $G_1 = \{V_1, E_1\}$ 、 $G_2 = \{V_2, E_2\}$ 分别存储拓扑图和无环拓扑图, 其中 V_1, V_2 为节点集合, E_1, E_2 为边集合。首先将 V_1, V_2, E_1, E_2 初始化为空, 然后依次处理每一条 IP 路径的每一条边, 把边合并到 G_1, G_2 中; 如果边不在 E_1 中, 则把边并入 E_1, E_2 ; 如果边的节点不在 V_1 中, 则把节点并入 V_1, V_2 。

此外, 在合并时需对 G_2 进行额外处理, 避免在 G_2 中出现环。判断环的出现以 G_1 为依据, 如果一条边不在 E_1 中, 但该边的两个节点均在 V_1 中, 那么把这条边并入 E_2 后就会出现环。对环的处理在 G_2 中进行, 生成虚拟节点, 并以该虚拟节点为中心把环转变为星形, 如图 4(a) 所示。如果环上已有虚拟节点, 把环上的虚拟节点和新生成的虚拟节点合并, 如图 4(b) 所示。

完整的 IP 路径合并算法描述如下:

算法: IPR_Merger

输入: IP 路径

输出: 拓扑图 G_1 , 无环拓扑图 G_2

设置图 $G_1 = \{V_1, E_1\}$, 图 $G_2 = \{V_2, E_2\}$, 初始化 $V_1 = E_1 = V_2 = E_2 = \emptyset$;

对于每一条 IP 路径 R_{ij} ($0 \leq i \leq P-2, i < j \leq P-1$,

Do//Do 1 begin/no end

对于每一条边 $(r_x, r_y) \in R_{ij}$, Do//Do 2 begin

{

if $(r_x \notin V_1)$ then { //if 1 begin

$V_1 = V_1 + \{r_x\}$;

$V_2 = V_2 + \{r_x\}$;

} //if 1 end

if $(r_y \notin V_1)$ then { //if 2 begin

$V_1 = V_1 + \{r_y\}$;

$E_1 = E_1 + \{(r_x, r_y)\}$;

$V_2 = V_2 + \{r_y\}$;

$E_2 = E_2 + \{(r_x, r_y)\}$;

} //if 2 end

if $(r_x, r_y) \notin E_1$ then { //if 3 begin

$E_1 = E_1 + \{(r_x, r_y)\}$;

建立新虚拟节点 $v_r, L_{vr} = 0$;

在图 G_2 中搜索从 r_x 到 r_y 的路径, 记为 R ;

对于 R 中的每一条边 (r_m, r_n) , Do//Do 3 begin

{

$L_{vr} = L_{vr} + L_{mn}$; $E_2 = E_2 - \{(r_m, r_n)\}$;

If $(r_m$ 不是虚拟节点) then

$E_2 = E_2 + \{(r_m, v_r)\}$;

else

对于所有与 r_m 相邻的边 (r_m, r_k) , Do

{

$E_2 = E_2 - \{(r_m, r_k)\}$;

$E_2 = E_2 + \{(v_r, r_k)\}$;

} //Do 3 end

$E_2 = E_2 + \{(r_y, v_r)\}$;

$L_{vr} = L_{vr} + L_{ry}$;

} //if 3 end

} //Do 2 end

}

4 层次拓扑结构的优化

我们将层次拓扑结构变换为层次树, 再将层次树变换为变换树, 目的是使广播源节点靠近树的根节点, 以利于通讯路径优化。层次树的变换根据具体的源节点进行, 需要在层次树中选择靠近源节点的网络节点作为根节点, 并改变层次树相应的拓扑结构。

以平均延时最大的节点为根节点, 从根节点开始遍历无环图 G_2 生成拓扑结构树 T 。如果 V_1 的平均延时最大, 那么从 V_1 开始遍历, 可以得到图 5 所示的以 V_1 为根节点的拓扑结构树。

利用无环拓扑图 G_2 生成拓扑结构树 T 的算法如下:

算法: G2T

输入: 无环拓扑图 G_2

输出: 拓扑结构树 T

```
{
  Visited( $r_x$ ) = 1;
  对于  $G_2$  中  $r_x$  的每一条邻边  $(r_x, r_y)$ , Do;
  如果 Visited( $r_y$ ) = 0, 那么
  {
    在树  $T$  中加入节点  $r_y$ , 并且使  $r_y$  为  $r_x$  的儿子节点;
```

G2T(r_y); //递归嵌套调用

}

得到的 ADHSNT 拓扑结构模型虽然忽略了局部连接的特殊性, 但是它要求尽量体现节点间的层次连接关系, 而不是两层结构那样把许多层次信息也忽略了, 更为强调层次结构上的完备性。

在模型中, 通过层次树的变换使得全局通讯的根节点靠近层次拓扑结构的根节点, 这样有利于全局通讯的优化。层次结构只是描述了节点间的静态的层次连接关系, 要在优化过程中加以利用, 还必须随着优化过程动态地描述。简单的层次结构用主节点来描述动态层次性, 对优化不利。ADHSNT 的规整过程充分体现动态的层次性。

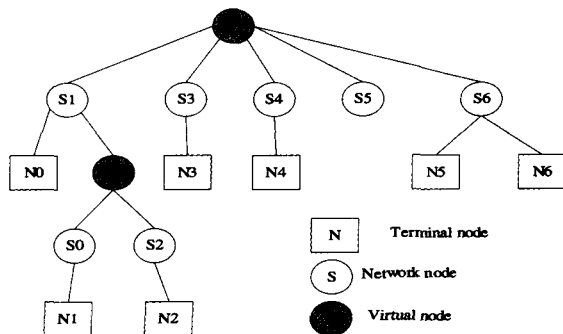


图 5 无环图生成拓扑变换树

结束语 拓扑结构模型是网络全局性能优化的重要依据, 尤其对存储环境而言, 不规则的网络拓扑结构对优化效果影响更大。本文研究了广域网络存储虚拟化形成的拓扑结构, 指出有简单层次模型的不足, 提出了一种增强的动态多层次结构模型——自动发现层次式存储网络拓扑 (Auto-Discover Hierarchical Storage Network Topology, ADHSNT), 既考虑了网络拓扑结构本身的特点, 又考虑了全局网络性能优化的需求。模型的层次性适合于描述存储网络环境, 并且动态规整过程使得它更加适用于全局网络性能优化研究。

参考文献

- Clark T. Design Storage Area Networks: A Practical Reference for Implementing Fibre Channel. Addison Wesley Professional, 2001
- Ruwart T M. OSD: A Tutorial on Object Storage Devices. In: the Tenth NASA Goddard Conference on Mass Storage Systems and Technologies Nineteenth IEEE Symposium on Mass Storage System College Park, Maryland, USA, April 2002
- Nagle D, Ganger G, Butler J, et al. Network Support for Network-Attached Storage. In: Proc. of Hot Interconnects, Stanford, CA, August 1999
- Yokota H. Performance and reliability of secondary storage systems. In: Proc. of World Multi conference on Systemic, Cybernetics and Informatics, July 2000. 668~673
- Kielmann T, Hofman R F H. MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems. In: Proceedings of the Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Atlanta, GA, May 1999
- Loweckamp B B, Beguelin A. ECO: Efficient Collective Operations for Communication on Heterogeneous Networks. In: International Parallel Processing Symposium, Honolulu, HI, 1996. 399~405
- Karonis N T, De Supinski B R. Exploiting Hierarchy in Parallel Computer Networks to optimize Collective Performance. In: Proceedings of Fourteenth International Parallel and Distributed Processing Symposium (IPDPS '00), Cancun, Mexico, May, 2000. 377~384
- Lin Hwa-Chun, Lai Shou-Chuan, Chen Ping-Wen. An algorithm for Automatic Topology Discovery of IP Networks. IEEE 0-7803-4788-9/98, 1998
- Miller N, Steenkiste P. Collecting Network Status Information for Network-Aware Applications. Infocom, 2000