多节点分级网络RAID存储结构研究*)

A Research on Multi-Level Networked RAID Based Cluster Architecture

刘晓光 王 刚 曾昭智 刘 璟
(南开大学信息技术学院 天津300071)

Abstract Storage network is popular solution to constraint of server in storage field. As the Gibson's standards, the performance of multi-level networked RAID based Cluster is almost same to the improved 2D-parity. Compared with other structure, it's easy to realize.

Keywords RAID, parity, MTTDL, MTTF

一、介绍

在计算机系统中,CPU和内存芯片的性能是决定指令的处理速度的主要因素。它们基本上遵循摩尔定律增长:

$$Transistors/Chip = 2^{year-1964} \quad (1)$$

随着计算机技术的深入发展,各类I/O密集型应用逐渐取代传统的计算密集型应用,占据越来越显著的地位。I/O系统逐渐成为计算机系统的核心部分之一。作为I/O系统最主要部分的硬盘,由于受到各种因素的制约,其性能的提高要滞后于芯片技术的发展。硬盘的性能指标包括两个部分:存储密度(与数据传输率相关);寻道和反转延迟时间。存储密度遵循"第一硬盘密度法则"[1]增长,即

$$Bits/inch^2 = 10^{(year-1971)/10} \quad (2)$$

但是,受到工艺条件的限制,寻道和反转延迟的性能增长缓慢(约7%/年),限制了硬盘整体性能的提高,使I/O系统成为整个计算机系统的重要瓶颈之一。

为了解决这一问题,Patterson等人提出了RAID的概念[2]:利用大量硬盘的并行操作,提高存储系统的性能;同时,通过存储空间的冗余,解决大系统带来的可靠性问题。目前,常用的RAID结构包括0,1,3,4,5级。但随着网络技术的深入发展,各类应用对存储系统的要求进一步提高,包括大容量、高传输率,低响应延迟等方面。在现有的结构设计中,所有的存储设备都是连接到同一个Server上的。由于Server的I/O性能不可能无限制提高,使得Server本身成为系统的瓶颈。在这种情况下,基于网络的存储系统(Storage Networks)逐渐成为一种必然的选择。目前在这一领域的研究包括SAN、NAS[3]和基于Cluster结构的多结点(node)网络存储系统等。其中,一个node包括CPU、内存、总线、硬盘、电源等多个部分。任一部分的失败都可能导致整个node的失败。因此单个node的MTTF(平均无故障时间)要低于单个硬盘的MTTF。为多硬盘系统设计的可容单错的RAID方案,并不能完全满足多node网络存储系统的要求,必须加以改进,以能够容许多个node失败。

二、分级网络RAID的组织结构

目前国内外对容多个错的磁盘阵列组织结构的研究包括分组RAID、2D-parity、3D-parity[4]、Reed-Solomon编码等。分级网络RAID是结合了分组RAID和2D-parity的思想,针对Cluster结构的特点提出的一种新型组织结构。

下面以一个16个node的二级网络RAID为例进行说明。如图1所示。所有的node被分为4组,每组4个node;每个node带有若干硬盘,为了描述的方便,下文中若不说明,都认为每个node只带有一个硬盘(实际上,多个硬盘通过RAID可以组成一个逻辑上的"硬盘",从而成为一个三级RAID结构);同组的四个node通过一个二级子网连接,全部4组node通过一级内部网连接。在二级子网内,所有node的硬盘通过网络RAID 5组成一个可以容单个node错的网络存储系统。在一级子网内,4组网络存储系统同样利用网络RAID 5组成一个可以容单个二级子网失败的大规模网络存储系统。

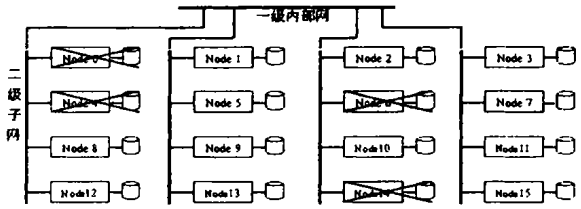

图1 二级网络RAID存储系统结构

三、结构分析

3.1 评价标准

为了评价磁盘阵列组织结构的优劣,Gibson提出了四个标准[4]:

·MTTDL (mean time to data loss) 是衡量磁盘阵性能的最重要标准,表示一个阵列系统的平均无故障时间。在实际应用中,一般使用蒙特卡洛模拟计算具体值。

·Check Disk Overhead 是阵列系统中校验空间和数据空间的比例。它表明了磁盘阵列系统存储空间的冗余度,即存储空间利用率。

·Update Penalty 表示数据更新的代价:当一个给定的数据单元的内容被改变后,需要更新的校验单元的数目。如果一种阵列系统的设计方案要求大于1个硬盘的校验单元被更新,则说明这种设计降低了阵列系统的并行性。而并行性正是

*)课题研究得到国家"863"计划资助,编号:863-306-ZD01-02-6.刘晓光 博士研究生,主要研究方向为并行与分布式系统等.王 刚 博士研究生,主要研究方向为并行与分布式系统等.曾昭智 硕士研究生.刘 璟 教授,博士生导师,主要研究方向为并行与分布式系统、算法分析、VLSI等.

磁盘阵列系统的最重要的优点之一。

·Group Size 是指当一个硬盘失败后，重构该盘的数据所需要访问的硬盘数目。当 group size 变大时，重构所需的时间也会相应地线性增加。这将增加 MTTR(mean time to recovery)，从而减少系统的 MTTDL。

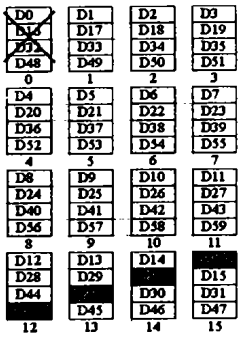

图2 RAID5

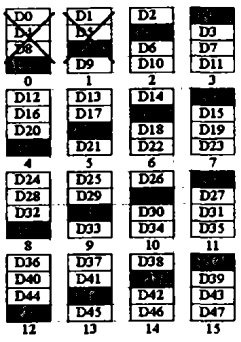

图3 分组 RAID 5

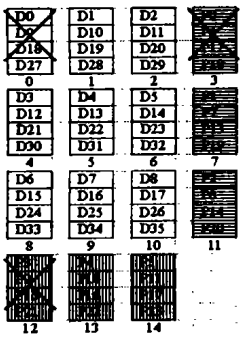

图4 2D-Parity

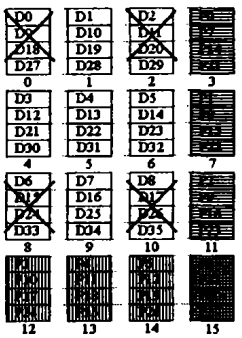

图5 2D-Parity 的改进

基于上述几个标准，本文以二级网络 RAID 为例，对比分级网络 RAID 和其它几种磁盘阵列组织结构(RAID 5、分组 RAID 5、2D-parity、2D-parity 的改进)的不同。图2、3、4、5分别是这几种组织结构的示意图。

3.2 比较分析

· 容错性 为了更形象地描述 MTTDL，本文引入了容错性概念。容错性是指造成一个系统产生不可恢复失败时的最低条件。

RAID 5容许所有单个错，当任意两个硬盘同时失效时，系统失败。

分组 RAID 5容许所有单个错，也能容许部分多个错情况。最低失效条件：当两个属于同一组的硬盘同时失效时，系统失败。如图3中硬盘0 、1同时失效的情况。2D-Parity 容许所有双错误和部分多错误情况。最低失效条件：当一个失效硬盘所属的横向和纵向条纹组的两个校验盘也同时失效，系统失败，即在数据布局图中，以失效数据盘为顶点，与两个校验盘构成一个直角三角形。例如，在图4中，硬盘0、3、12同时失效的情况。其中，P0、P1、P2是横向校验，P3、P4、P5是纵向校验。

2D-Parity 的改进容许所有三错误和部分多错误情况。最低失效条件：同时失效的四个盘，在横向和纵向上，都在相同的两个条纹组中，即在数据布局上四个盘构成一个矩形。例如，在图5中硬盘0、2、8、10同时失效的情况。其中，P0、P1、P2是横向校验，P3、P4、P5是纵向校验，P6是校验的校验。

对分级网络 RAID 而言，级数越大，容错性越好。为了比较的方便，仍然以二级网络 RAID 为例。二级网络 RAID 也可以容所有三错误。最低失效条件是：两个二级子网中同时出现两个 Node 失败，则系统失败，即在数据布局图中，四个失效 Node 以条纹方向为平行边，构成一个梯形。例如：在图1中 Node 0、4、6、14同时失效的情况。同时，在网络环境中，网络设备也是系统的一个重要的单一失败点，例如，交换机的失效，会导致连接在该交换机上的所有设备的失效。此外，在灾难情况下(如火灾等)，往往会造成某一物理区域内所有设备的失效。但以往的设计一般都不考虑这些情况，回避了网络设备故障和灾难情况下系统的可用性问题。二级网络 RAID 可以容单个二级子网的失败。对二级子网而言，保证了网络设备故障或灾难情况下系统的可用性。同传统设计相比，系统可用性进一步提高。存在的不足是：图1中的一级内部网仍然是一个单一失败点。

·MTTDL 假设硬盘的 MTTF 和 MTTR 是外生变量，失败事件是独立的。用 N 表示一个条纹组中数据单元的个数，G 表示划分的组数，对 RAID 5有：

$$P_{(\text{在修复期间至少一个盘坏})}=1-e^{-MTTR\cdot N/MTTF} \tag{3}$$

因为 MTTR 远小于 MTTF/N，所以公式(3)可以简化为：

$$P=\frac{MTTR_{disk}N}{MTTF_{disk}} \tag{4}$$

$$MTTDL_{RAID5}=\frac{E_{(\text{出错时间间隔})}}{P_{(\text{在修复期间至少一个错})}}=\frac{MTTF_{disk}}{(N+1)P}$$

$$=\frac{MTTF^2_{disk}}{N(N+1)MTTR_{disk}} \tag{5}$$

$$MTTDL_{\text{分组}RAID5}=\frac{MTTF^2_{disk}}{GN(N+1)MTTR_{disk}} \tag{6}$$

$$MTTDL_{2D\text{-}Parity}=\frac{MTTF^3_{disk}}{GN(N-1)^2MTTR^2_{disk}} \tag{7}$$

$$MTTDL_{2D\text{-}parity\text{改进}}=\frac{4MTTF^4_{disk}}{GN(N-1)^2MTTR^3_{disk}} \tag{8}$$

$$MTTDL_{\text{二级网络}RAID}=\frac{8MTTF^4_{node}}{G(G-1)N^2(N-1)^2MTTR^2_{node}} \tag{9}$$

假设 Disk 和 Node 的 MTTF、MTTR 相同。从公式(7)、(8)、(9)可以得到：

$$MTTTDL_{\text{二级网络}RAID}=\frac{8MTTF}{(G-1)N}MTTDL_{2D\text{-}Parity}$$

$$=\frac{2}{(G-1)NMTTR}MTTDL_{2D\text{-}Parity\text{改进}} \tag{10}$$

一般情况下，MTTF 都在数万小时以上，而 MTTR 都在数小时以内。因此二级网络 RAID 的 MTTDL 要远大于2D-Parity 而要小于2D-Parity 的改进。而且，二级网络 RAID 比 2D-Parity 要高一个数量级，与2D-Parity 的改进在同一数量级上，只存在系数之差。

·Check Disk Overhead、Update Penalty 和 Group Size

表1 五种结构的对比(16个 disk/node)

组织结构	Check Disk Overhead	Update Penalty	Group Size
RAID 5	1/15	1	16
分组 RAID 5	4/12	1	4
2D-Parity	6/9(共15个盘)	2	4
2D parity 改进	7/9	3	4
二级网络 RAID	7/9	3	4

表1列出了在16个盘或节点条件下，五种组织结构在 Check Disk Overhead、Update Penalty 和 Group Size 方面的差异。可以看出，2D-Parity 的改进和二级网络 RAID 的代价

(下转第97页)

```
    Btmp=R;
    Seed=R 所对应的 F;
    L=0;
  End;
  if(Btmp≥Bth)
    Return R 所对应的 F;
    Break;
  End;
  if(循环次数 L=T)Break;
  L=L+1;
End;
```

群体遗传算法的一个主要特点是在子代染色体发生变异进化时，初始群体也在同时发生变异进化。一旦得到了较好的物化视图集，就以其为新的种子，开始新一轮的进化。这些特点充分体现了遗传算法的核心思想。

在群体遗传算法中，“种子”(Seed)的使用，使初始群体能够从一开始就有较好的基因构成，种子的基因被认为是较优良的基因，初始群体中所有个体开始都包含种子的基因，我们称这样的基因为“保守基因”，这样从初始群体产生的所有后代都能尽可能地包含保守基因，能够保证后代在早期的进化过程中有较好的质量，同时主要进行局部搜索。随着进化时间的增加，初始群体中保守基因的含量逐渐下降，这说明原来的局部进化空间在逐渐缩小，新的基因正在逐渐被引入，总的进化空间在逐渐增大，也就是算法正逐渐在越来越大的进化基础上、越来越多的进化方向上开始搜索更优解，即进行全局搜索。当得到了较好的物化视图集并将此作为新的种子时，新种子的优良基因成为新形成的初始群体中的保守基因，然后开始新一轮进化，搜索重心又集中于新的局部。这样，交替的局部、全局、局部、全局的搜索方式能够取得较理想的搜索效果。群体遗传算法的缺点是算法较复杂，运行时间较长。

算法中 Pe 的取值直接影响初始群体中个体的多样性，相对较大的取值更有利于取得较好的优化效果，但是如果 Pe 取值过大，将严重影响算法的运行效率。具体的适应度函数的选择和几个算子的实现，限于篇幅，不再在这里详细介绍。

总结 本文以传统关系数据库为基础，提出了对数字图书馆中 XML 元数据进行存储的一个可行的解决方案。本文的最大贡献不在于解决了 XML 半结构化数据在传统关系数据库中的存储问题，而是针对实际的查询操作使用了物化视图技术和遗传算法对已有的存储方法进行了优化，并对传统的遗传算法进行了改进，使这一经典算法在数字图书馆研究领域又得到了新的应用。我们已经在 DB2上实现了本文中提到的主要功能的原型系统，并将在以后的工作中对具体存储模式和算法作进一步的研究和改进。

参 考 文 献

1 Deutsch, Fernandex M, Suciu D. Storing Semistructured Data with STORED. ACM SIGMOD Conf. 1999. 431～442

2 Shanmugasundaram J, Tufte K, Gang H, Chun Z. Relational Databases for Querying XML Documents: Limitations and Opportunites. In: Proc. of the 25th VLDB Conf. Edinburgh, Scotland, 1999

3 Tian Feng, DeWitt D J, Chen Jianjun, Zhang Chun. The Design and Performance Evaluation of Alternative XML Storage Strategies. Unpublished manuscript, 2001

4 周明，孙树栋编著. 遗传算法原理及应用. 北京：国防工业出版社，1999

5 邵军力，张景，魏长华. 人工智能基础. 北京：电子工业出版社，2000

(上接第23页)

相同，而且是最大的。这也是容错性提高所付出的代价。

3.3 系统实现

目前，常用的 RAID 0到5的实现技术已经非常成熟，包括硬件 RAID 实现和软件 RAID 实现等。与之相比，其它数据布局方案(如2D-parity)的实现需要重新设计实现映射函数，代价非常大。分级网络 RAID 可以充分利用现有的 RAID 5实现，不需要对映射函数作任何修改，实现代价很小。同时，它还具有良好的可扩展性。当系统规模扩大时，通过增加组数和级数，分级网络 RAID 可以方便地进行系统的扩展。实际上，利用网络软 RAID 技术，基于100M 以太网，我们已经实现了一个简单的二级网络 RAID 系统[5]。图6是该系统和单一 SCSI 硬盘(IBM DDYS-T36950M)、光纤通道 RAID 系统(3S FC3012)的对比。实验系统采用 ZD-NET 的 NetBench 7.02。

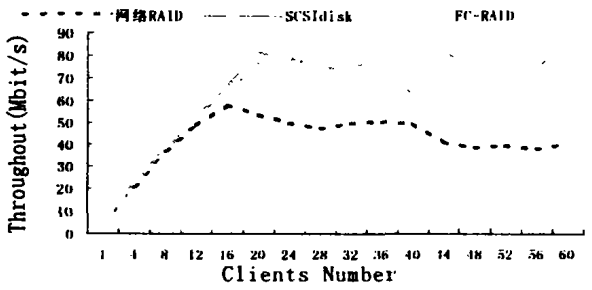

图6 NetBench 测试结果

结论 分级网络 RAID 是一种基于 Cluster 结构的网络存储系统组织结构。基于 Gibson 提出的四个评价标准，可以看出，分级网络 RAID 是一种容错性高而数据恢复代价小的优选设计。同时，与其它设计相比，分级网络 RAID 可以充分利用现有的 RAID 实现，易于系统实现。此外，它还具有可宽展性高、配置灵活的特点。

现有的二级网络 RAID 设计存在的不足在于：一级内部网络本身和一级网络 RAID 的组织 Node 是整个系统的单一失败点。目前只能使用高可用系统来提高这些设备的 MTTF。今后，将力图通过设计新的数据布局方案等途径解决这一问题。

参 考 文 献

1 Frank P D. Advances in Head Technology, Presentation at Challenges in Disk Technology Sgort Course. Institute for Information Storage Technology, Santa Clara University, Santa Clara, California, Dec. 1987. 15～17

2 Patterson D A, Gibson G, Katz R H. A case for redundant arrays of inexpensive disks (RAID). In: Proc of 1988 ACM SIGMOD Int'l Conf on Management of Data, New York: ACM Press, 1988. 109～116

3 Gibson G A, Nagle D F. NASD scalable storage systems. In: USENIX99, Extreme Linux Workshop, Monterey CA, 1999

4 Gibson G, et al. Coding Techniques for Handling Failures in Large Disk Arrays: [Technical Report UCB/CSD 88/477]. Computer Science Division, University of California, July, 1988

5 王刚，刘晓光，刘璟. 网络软 RAID 的设计与实现. 计算机研究与发展，2000，37(10·增刊)：81～83