基于集群服务器的容灾系统的副本管理研究*

武 鲁，李伟华，李钟华
(西北工业大学 计算机学院，陕西 西安 710072)

摘 要：提出一种基于集群服务器的容灾系统副本管理方案，提出多个副本的一致性维护和副本选择的算法以及副本数量和分布方式的数学模型。通过容灾系统的性能测试实验，证明它能够实现数据的快速自动恢复，有效地管理副本，并保持副本可靠性和集群服务器性能之间的平衡。

关键词：容灾系统；数据恢复；副本一致性；副本选择
中图法分类号：TP393 文献标识码：A 文章编号：1001-3695(2006)06-0076-03

Research on Replica Management of Disaster Recover System Based on Cluster Server

WU Lu, LI Wei-hua, LI Zhong-hua
(College of Computer Science, Northwestern Polytechnical University, Xi'an Shanxi 710072, China)

Abstract: A disaster recover system replica management scheme based on cluster servers is introduced in this paper, in which a consistency maintenance algorithm of multiple replicas and a replica selection algorithm are put forward and a mathematical model of quantity and distribution of replicas is set up. Fast auto recovery of data, efficient copy management and a balance dependability of replicas and performance of cluster server can be achieved by this system scheme, which is proven by performance test of the disaster recover system.

Key words: Disaster Recover System; Data Recovery; Consistency of Replicas; Selection of Replicas

1 引言

随着集群服务器的广泛应用，它的安全问题越来越引起人们的关注。虽然集群服务器本身可以通过服务转移、负载平衡等技术实现高可用性的网络服务，但当集群中的节点遭到破坏时，要使这个节点恢复服务必须人工干预，这样既浪费人力，又在一定时间内影响了集群服务器的性能。因此，开发一种能够在关键数据和服务遭到破坏时对其进行实时自动恢复的容灾系统是十分必要的[1,2]。

为了保证容灾系统能够快速有效地恢复遭到破坏的数据，必须在集群服务器的多个节点上保存数据副本。多个副本保存在多个节点中，可有效提高遭破坏节点至少获得一个数据拷贝的机会，提高数据的可靠性[3]。但是，副本的增加也会带来新的问题：如何管理众多的副本；如何在发生事故时迅速选择一个最优的副本进行恢复；如何在数据更新时保持多个副本的一致性。这些都会给系统带来额外的开销，势必影响到集群服务器的性能。针对上述问题，本文提出了一个有效管理副本、保持集群服务器可靠性和性能平衡的方案。

2 容灾系统体系结构

容灾系统由监测模块、控制模块、文件信息模块和通信代理四部分组成，如图 1 所示。监测模块通过 FAM(File Alteration Monitor)[4] 来监测关键文件列表，并传递事件消息给控制模块。控制模块的核心是传输控制进程，它用来控制系统中各种消息的传输、各个列表的更新维护以及数字签名和散列值的计算。文件信息模块包括关键文件和副本文件及其文件列表。关键文件库由监测模块进行检测，副本文件库存放着来自其他节点关键文件的备份。通信代理完成集群服务器各节点上的容灾系统之间的副本维护、文件恢复等过程的安全通信，将多个节点组成一个容灾系统联盟。

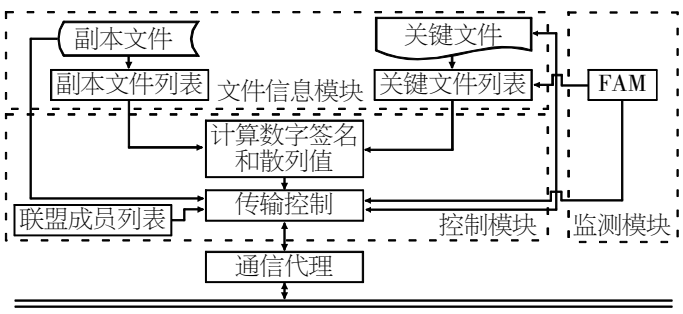

图 1 容灾系统体系结构

3 容灾系统联盟副本的管理

3.1 相关符号定义

容灾系统中的各种符号定义如下：设集群服务器中的容灾系统联盟有 N 个节点，其中每一个成员节点 M_i 定义为一个四元组：

$$M_i = \{ < Id, L, F_{O\_LIST}, F_{B\_LIST} > | i \in N \}$$

容灾系统联盟定义为

$$A_R = \{ M_i | M_i = < Id, L, F_{O\_LIST}, F_{B\_LIST} >, i \in N \}$$

其中 $M_i.Id$ 表示节点的标志信息；$M_i.L$ 为联盟成员列表，定义为一个二元组：$L = \{ < M_j.Id, T_j > | j \in N, j \neq i \}$，$T_j$ 为具有标 $M_j.Id$ 的节点 M_j 相对于 M_i 的负载，这里的负载通过两个成员

收稿日期：2005-04-23；修返日期：2005-06-15
基金项目：国家“863”计划资助项目(2003AA142060)

点对点通信的所用时间计算; F_{O_LIST}, F_{B_LIST}分别是原文件表和副本文件列表, 它们都是文件信息表 F_{LIST}, 文件定义为一个四元组, 文件信息表定义为

$$F_{LIST}=\{K_m \mid K_m = <F, C, H, M.Id>, m\in M\}$$

其中 F 为文件信息, 包括文件名、版本等信息; C 为存放该文件的路径; H为文件的散列函数值, $M.Id$ 为文件来源标志, M 为列表中文件的个数。

3.2　副本维护算法

容灾系统联盟维护包括联盟成员列表的维护和副本一致性的维护。联盟中的成员节点通过定期发送经过加密的消息来完成对其他成员节点的“心跳”检测, 维护其联盟成员列表。成员节点间的探测过程是相互的, 通过探测还可以得到成员节点之间通信的网络负载情况。与此同时, 向副本节点发送原文件的散列函数值, 在监测模块没有发现原文件变更的情况下, 一旦副本文件与原文件的散列值不一致, 立即对该副本进行维护。容灾系统联盟中副本的维护使用如下算法:

(1) 节点 M_i 生成随机数 R_{ij}, 用于标志一次副本维护行为; 对于 $M_i.L$ 中每一个副本保存节点 M_j, M_i 向 M_j 传递 R_{ij}以及 $M_i.Id$ 的数字签名; 保存发送时间 $t_{i,0}$。

(2) M_j 收到来自 M_i 的消息后, 解码得到 R_{ij}和 $M_i.Id$; 查询 $M_i.L$, 如果 $M_i.Id$ 在列表中, 则确认此信息来自 M_i; 将解码得到的 R_{ij}和 $M_j.Id$的数字签名发送到 M_i, 并记录发送时间 $t_{j,0}$。

(3) M_i 收到来自 M_j 的消息, 解码得到 R_{ij}和 $M_j.Id$, 查询 $M_i.L$, 如果 $M_j.Id$在其中, 确认 M_j 处于存活状态; 记录接收时间 $t_{i,1}$, 将 $M_i.L.T_j$ 更新为 $t_{i,1}-t_{i,0}$; 将 R_{ij}和 F_{O_LIST}中所有文件的信息依次进行数字签名并发送给 M_j。

(4) M_j 收到来自 M_i 的消息后, 解码得到 R_{ij}和 $M_i.F_{O\_LIST}$; 记录接收时间 $t_{j,1}$, 将 $M_j.L.T_i$ 更新为 $t_{j,1}-t_{j,0}$; 查询其 $M_j.F_{B\_LIST}$, 将其中 $M.Id$ 为 $M_i.Id$ 的文件和 $M_i.F_{O_LIST}$中的文件信息按文件名依次比较散列函数值。如果两者不一致, 则 M_j 向 M_i 发送不一致的文件信息 K_m 和 $M_j.Id$, 请求维护; 否则发送 $M_j.Id$, 完成来自 M_i 的副本的维护。

(5) M_i 收到 M_j 的消息后, 如果发现其中有文件信息 K_m, 则向 M_j 传输该文件; 否则完成 M_j 中副本的维护。

容灾系统联盟中的节点每隔一定时间相互进行一次副本维护, 可维持一个递减的计数器来计时。计数器初始值可以为定值, 也可以是一个随机产生的数。节点上的原文件和副本文件列表也需要定期的更新维护, 但是频繁计算大量文件的散列值会带来很大的开销, 严重影响集群服务器各节点的处理能力, 因此不能与副本维护同步进行, 可以单独为其设立一个计时器。

3.3　副本选择算法

当节点 M_i 的关键文件遭到破坏时, 监控模块首先发现文件不正常的改变, 然后通知控制模块, 控制模块在可达的副本节点上发起一次“选举”, 比较各节点副本文件的散列值, 用少数服从多数原则选出一个可信、最优的节点。如果“选举”产生的副本和原文件的散列值一致, 则可以对原文件进行恢复; 否则, 节点 M_i 可以“一票否决”选举产生的副本。此时认为容灾系统联盟遭到大规模的破坏, 必须请求管理员人为介入。当某个节点的关键文件请求恢复时, 为防止恢复操作和一致性维护操作的冲突, 将阻塞一切副本维护进程。算法如下:

(1) M_i 阻塞本机的副本维护进程, 向 $M_i.L$ 中每一个副本保存节点 M_j 发送控制信息 D_{ij}和 $M_i.Id$ 的数字签名。

(2) M_j 收到来自 M_i 的信息后, 确定节点 M_i 发生事故, 阻塞目前进行的副本维护进程; 向 M_i 发送就绪信息 D_{ji}和 $M_j.Id$ 的数字签名, 通知 M_i 该副本节点已经就绪。

(3) M_i 收到来自 M_j 的消息, 确定 M_j 已经就绪, 根据监测模块的信息, 在 F_{O_LIST}中查询需要恢复的文件信息 K_m, 并向 M_j 发送 $K_m.F$ 的数字签名, 通知 M_j 进行表决。

(4) M_j 收到来自 M_i 的消息, 查询 $M_j.F_{B_LIST}$得到文件 K_m 的副本信息 $M_j.F_{B\_LIST}.K_m$; 向 M_i 发送 $M_j.F_{B\_LIST}.K_m$ 和 $M_j.Id$ 数字签名, 进行表决。

(5) M_i 将收到的 $M_j.F_{B\_LIST}.K_m.H$ 相互比较, 找到一个多数一致的散列值, 然后将其与 $M_i.F_{O\_LIST}.K_m.H$ 比较。如果两者一致, 则查询 $M_i.L$, 找到 M_i 与多数一致的节点间网络负载最小的一个节点 M_l; 否则, 发出报警信号通知管理员人为介入。

(6) M_i 向 M_l 发送恢复控制信号 R 的数字签名;

(7) M_l 收到来自 M_i 的消息, 根据(4) 中得到的 $M_l.F_{B\_LIST}.K_m$ 向节点 M_i 传送文件 K_m。

(8) M_i 收到文件 K_m, 计算出散列值 H, 并将其与 $M_i.F_{O\_LIST}.K_m.H$ 比较, 如果一致, 则向 $M_i.L$ 中所有节点发送启动副本维护进程控制信号 RS 的数字签名, 同时启动本机副本维护进程; 否则, 重复步骤(6)。

(9) M_j 收到来自 M_i 的消息, 确认 M_i 中的文件 K_m 已经恢复, 重新启动副本维护进程。

4　副本数量和分布方式

在联盟中只保存一个副本是危险的。集群服务器中任何一个节点都要面临可靠性问题, 节点的故障、网络的延迟、节点的退出都将造成该点不可到达。如果一个不可达的节点保存着其他节点的唯一副本, 当原文件遭到破坏时将无法恢复, 影响了集群服务器的高可用性。

如果副本分布按照“多到一”的原则, 在 N 个节点的集群服务器设置一个副本服务器, 用来存储其他各节点的副本。这就需要副本服务器节点有很大的存储容量, 为了达到与其他节点相同的可靠性等性能要求, 机器价格势必更加昂贵, 整个集群服务器的成本也会提升。如果采用与普通节点相同的机器, 副本服务器节点的可靠性和性能就必然低于普通节点, 从而成为系统的“最短的一块桶板”[5]。即使采取折中办法, 在集群服务器中设置多个副本服务器, 仍然不能改变副本服务器可靠性低和性能下降的现状。

由此可见, 集群中所有节点都应该等同对待。所以我们采取“分组对等互备”的副本分布方式: 对于 N 个节点的集群服务器, 将其分成 N/m 个联盟小组(设 m 可以整除 N), 小组成员之间相互保存副本。这样每个数据都有 m 个副本且分布在联盟小组的 m 个节点上, 每个节点恰好保持来自 m 个联盟小组成员的副本。如果单个节点正常工作的概率为 p, 则 N 个节点的容灾系统联盟可以正常提供副本数据的概率 R_n 为

$$R_n = 1 - (1 - p)^{m \quad N/m} \qquad (1)$$

显然,副本越多,数据的可靠性就越高。

虽然副本的增加能有效提高数据的可靠性,但是不得不考虑为此付出的代价,如磁盘空间、增加副本过程中引起的网络传输开销、各节点之间频繁进行心跳检测通信的开销等。当 $m = N$,即联盟成员之间相互保存副本时,虽然可靠性得到了提高,它带来的开销却难以忍受。因此,需要有一个数学模型来确定副本可靠性、网络开销和副本个数之间的关系,从而确定满足可靠性需求情况下最小的网络开销。

由于关键文件被破坏是一个小概率事件,数据恢复的开销是暂时的;在一个联盟节点上增加副本的网络开销也是瞬间的。在系统运行的全过程中,它们都可以忽略不计。但是,容灾系统联盟的成员列表维护和副本的一致性维护带来的网络开销是长期的,因此本文认为:容灾系统联盟网络开销等于成员列表维护的网络开销和副本维护的网络开销之和。为了方便计算,设置各变量如下:

B_{total} 为所消耗的网络带宽;B_{permit} 为可以接受的网络带宽消耗;R_{expect} 为期望得到的可靠性;f 为文件一致性维护的频率;N 为集群服务器中节点个数;m 为联盟小组节点个数;M 为每个节点备份到其他节点文件个数;p_{ij} 为主机 i 的第 j 个文件需要维护的概率;K_{ij} 为主机 i 的第 j 个文件的大小;令节点间传递消息的格式一致,大小都是 k。

根据前面描述的算法,由于每两个节点之间都要相互探测,故联盟小组中的探测次数为 P_m^2 次,成员列表维护需要在两节点之间发送两次大小为 k 的消息,副本维护需要某个节点向其他节点首先发送 M 个大小为 k 的消息,然后按副本的情况决定是否对其进行更新。因此,维护整个容灾系统联盟成员列表维护的网络开销为

$$B_{m\_\text{list}} = 2k \times P_m^2 \times \frac{N}{m} \times f \qquad (2)$$

副本一致性维护的网络开销为

$$B_{m\_\text{replica}} = (M+1) \times k \times P_m^2 \times \frac{N}{m} \times f + \sum_{i=1}^{N}\sum_{j=1}^{M}(p_{ij} \times K_{ij}) \quad \times f \qquad (3)$$

如果在一定时间内需要更新的副本文件的大小为一个常量,则副本维护过程中文件传递的网络开销也是一个常量,因此可以认为容灾系统联盟的网络开销为

$$B_{\text{total}} = (M+3) \times k \times N \times f \times m + [\chi - (M+3) \times k \times N] \times f \qquad (4)$$

由于副本的个数只与 m 有关而与 N 无关,因此通过式(4)可以发现,当 f 一定的情况下,容灾系统联盟网络开销的总量和联盟小组的节点个数,即副本文件的个数 m 呈线性关系。随着 m 的增大,网络开销随之增大,而网络下层提供的网络带宽是有限的,不能无限地增加副本。综合式(1)和式(4),得到副本个数取值范围的数学模型为

$$\begin{cases} [1 - (1-p)^m]^{N/m} \geq R_{\text{expect}} \\ (M+3) \times k \times N \times f \times m + [\chi - (M+3) \times k \times N] \times f \leq B_{\text{permit}} \end{cases} \qquad (5)$$

5 性能测试

在测试实验中,我们使用了一个由 16 台服务器组成的集群服务器。根据式(1)得到的结论,表 1 给出了在不同的节点正常工作概率下保存不同数量的副本,整个容灾系统联盟可以有效提供副本文件的概率。

表 1 容灾系统联盟能有效提供副本文件概率的比较

p \ m	m = 1	m = 2	m = 4	m = 8	m = 16
p = 0.8	0.028	0.721	0.993	0.999	0.999
p = 0.9	0.185	0.923	0.999	0.999	0.999
p = 0.99	0.851	0.999	0.999	0.999	0.999

从表 1 中可以看出,只存在一个副本时,即使单个节点的正常工作的概率非常高,整个系统可以正常提供副本文件的概率仍然比较低;当副本增加到四个时,正常提供副本的概率基本上可以达到高可用性的要求。

在集群服务器的各个节点上配置容灾系统,备份不同数量的关键文件,并设置 1 ~16 个副本,然后删除一个节点上的关键文件,让容灾系统对其进行恢复。在这个过程中,监控此节点上的副本维护进程和恢复进程,测试副本维护和自动恢复时所消耗的时间,得到图 2 和图 3。

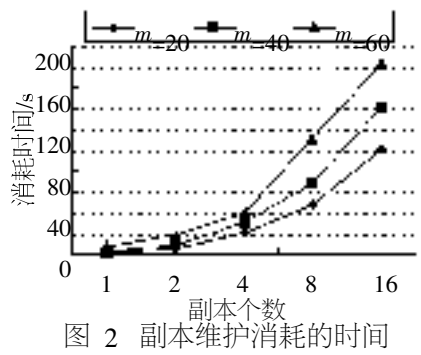

图 2 副本维护消耗的时间随副本数量的变化情况

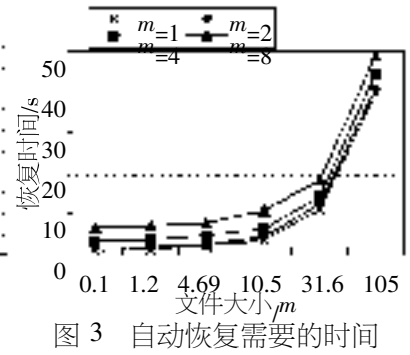

图 3 自动恢复需要的时间随文件大小的变化情况

图 2 和图 3 中的数据可以说明,副本维护和自动恢复所消耗的时间随着副本数量的增长而快速增长,大量副本的维护所需要的时间是十分惊人的,这不仅会消耗服务器宝贵的资源,还占用各节点之间的网络带宽,因此必须把副本数量控制在适当的范围之内。不难看出,如果保存四个副本,对于不同数目的关键文件,一次副本维护的时间在 20s ~40s,数据恢复时间比无须进行副本比较时($m = 1$)平均多 3.26s,因此无论是可靠性还是容灾系统总的开销方面都是可以接受的。

6 结束语

容灾系统中的副本管理是一个很重要的问题,本文提出的基于集群服务器的容灾系统中副本的维护和选择算法,以及副本数量和分布的模型,形成一个容灾系统副本管理方案。经实验证明,它能有效管理副本,实现数据快速恢复,并将副本数量控制在一个适当的范围内,保持副本可靠性和集群服务器性能之间的平衡,具有很高的应用价值。

参考文献:

[1] 何聚厚,李由,李伟华. 基于联盟备份的自动恢复安全模型研究[J]. 西北工业大学学报,2004,22(2):176-179.

[2] 黄遵国,任剑勇,胡光明. 一种信息安全事故的快速响应与恢复(r-RR)框架研究[J]. 计算机工程与科学,2001,23(6):43-46.

[3] 魏青松,卢显良,侯孟书. AdpReplica:自适应副本管理机制[J]. 计算机科学,2004,31(12):34-36.

[4] SGI Developer Open Source. Fam Overview[EB/OL]. http://oss.sgi.com/projects/fam/,2005-04-10.

[5] 沈海华,陈世敏,等. WWW 集群服务器数据副本分布方式研究[J]. 软件学报,2001,12(3):367-371.

作者简介:

武鲁(1982-),男,硕士研究生,主要研究方向为计算机网络安全;李伟华(1951-),男,教授,博士生导师,主要研究方向为计算机网络安全、多媒体通信技术、智能决策支持系统;李钟华(1976-),男,博士研究生,主要研究方向为计算机网络安全。