容灾系统中数据的多源快速备份和恢复*)

于 忠[1] 李钟华[2] 武 鲁[1] 李伟华[1]
(西北工业大学计算机学院 西安 710072)[1] (江西财经大学信息管理学院 南昌 330013)[2]

摘 要 通过对容灾系统多副本备份方式的分析,为有效解决数据的快速备份和恢复问题,利用P2P资源共享的思路,提出了数据文件副本在网络中多点之间传输与共享的算法思想,并进行了仿真测试。试验结果表明,多源快速备份和恢复算法(MSB&RT)极大提高了数据传输的效率,为关键数据的恢复提供了更为可靠的保证。

关键词 网络安全,容灾,备份与恢复

The Multi-source Backup and Recovery of Data in the Disaster Recovery System

YU Zhong[1] LI Zhong-Hua[2] WU Lu[1] LI Wei-Hua[1]
(School of Computer Science, Northwestern Polytechnic University, Xi'an 710072)[1]
(School of Information Technology, Jiangxi University of Finance & Economics, Nanchang 330013)[2]

Abstract Through analyzing the way of multi-transcriptions backup in the Disaster Recovery system, and effectively solves the question of the data fast backup and restoration, a algorithm thought which to transmits and shares the data file transcription among network multi-nodes is proposed. Here we use the resource sharing idea of P2P and carried on a simulation test, the result indicate that the Multi-Source Backup and Recovery quick Transmission Algorithm (MSB&RT) extremely enhance the efficiency of data transmission and reliably guarantee the essential data recovery.

Keywords Network security, Disaster recovery, Backup and recovery

当前容灾系统所用的备份传输方法中,一般采用单播或多播方式。前者由于所需的时间为所有节点的传输时间之和,因此效率极低。采用多播方式[1~3],虽然传输时间大为减小,传输效率有很大提高,但此时所有上载开销都在源数据节点上,其上行带宽将成为并行传送的瓶颈。在数据恢复传输阶段,当原数据节点出现故障需要进行恢复时,只能从副本节点中的一个传输备份数据,而其他副本节点得到的数据仅提供了增加副本可用性的功能,造成很大的浪费。P2P[4,5]的最初目的是为了解决文件服务器带宽不能满足大量用户对文件下载需求的问题。现在P2P有很大的发展,已成为因特网中资源共享的最佳方法。正是利用这种思路,我们提出了数据文件副本在网络中多点之间传输与共享的思想。

多源快速备份与恢复传输算法(Multi-Source Backup and Recovery quick Transmission Algorithm, MSB&RT),充分利用了联盟中各点使用LECC编码[6]节省下来的反馈信道带宽。备份过程中,源数据节点并行地给各个副本节点发送不同的编码包。各副本节点在接收源数据节点编码包的同时,也相互发送相异的编码包,从而大大降低了源数据节点的工作量,提高备份传输的效率。当原数据节点需要数据恢复时,则控制多个副本节点并行地给它传送相异的编码包,共同完成数据的恢复传输。特别是对海量数据的备份与恢复,MSB&RT算法具有很大的优势和极好的效率。

1 MSB&RT的基本思想及描述

LECC编码传输的编码包相互独立,各副本节点只要接收到数量足够、以任意顺序到达的编码包,就可以合成原始数据,无需反馈应答,为一个副本节点向其他副本节点传输数据提供了上行带宽;副本节点不必接收全部编码包就可以取得源数据节点同等的地位,有效地减小了源数据节点备份传输的负载。MSB&RT算法是根据LECC编码传输的这些特性而设计的,它可以实现海量数据的多点协同快速备份与恢复。为了便于叙述,给出如下两个定义:

定义1(源数据节点和副本节点) 在备份传输的过程中,若将节点A的数据文件备份到节点B、C、D,则节点A处于传输的"源头"位置,称之为源数据节点或源节点(Source Node);节点B、C、D保存A的备份文件,称之为副本节点(Replica Node)。

定义2(原数据节点) 在恢复过程中,若节点A的数据遭到破坏,就需要节点B、C、D将其数据恢复到原来的状态,此时称节点A为原数据节点(Original Data Node)。

1.1 主要组成部分及其描述

控制中心:控制中心(Control Center)负责MSB&RT传输的控制与调度,记录参与备份恢复传输的各个节点的信息及当前拥有编码包的信息。实时维护更新两个信息表:节点信息表和编码包信息表。

源节点(Source Node):节点信息用0号节点及其地址信息A_0进行标识:$SN=\langle 0,A_0\rangle$。

副本节点(Replica Node):控制中心对多个副本节点进行顺序编号,用其编号i标识。i节点信息用编号i及其地址信息A_i进行标识:$RN_i=\langle i,A_i\rangle$。

编码包(Encode Package):MSB&RT系统在网络上传输的数据,由LECC编码产生。控制中心对编码包进行顺序编号,用其编号j标识。编码包j用编号j及其散列值H_j进行描述:$EP_j=\langle j,H_j\rangle$。

*)国家863高科技研究发展计划项目资助(2003AA142060);陕西省自然科学研究项目(2004F13)。于 忠 博士生,主要研究方向:计算机网络安全、多媒体通信技术;李伟华 教授,博导,主要研究方向:计算机网络安全、多媒体通信技术、智能决策支持系统。

下载和上传：进行数据交换的节点中，提供数据传输称为上传(Upload)，得到数据的传输称为下载(Download)。

注册(Register)：某节点拥有一个正确的编码包后，向控制中心报告拥有该包并准备为其他节点提供下载的过程。注册过程包括节点的注册请求和控制中心的注册应答。

注册请求：$R_{request}$::=〈结点信息，编码包信息〉；

注册应答：R_{answer}::=〈编码包信息，Registered〉。

1.2 下载选择策略

MSB&RT算法为避免备份传输失败(源节点故障)，需要采取一定的副本选择策略，用以完成各个备份节点及传输标码包的优化选择，使得数据在各副本节点间合理分布，以最少的时间完成备份传输。

最少者优先原则：对一个下载节点来说，在选择一个编码包下载时，此编码包应是整个联盟这一时间内最"稀有"的包，也就是所谓的"最少者优先"。这种策略能确保每个节点都能拥有其他副本节点最希望得到的编码包，使那些比较"普及"的编码包下载顺序靠后，从而使所有副本节点编码包的并集更趋于完整，备份传输也就趋向于一个更优的状态。

最快者优先原则：在备份传输进行了一段时间以后，经过"最少者优先"原则选择出来编码包，可能已经分布于多个副本节点中。在没有该编码包的副本节点下载它时，就需要另外的策略来选择下载位置。为体现MSB&RT算法的快速高效备份特性，实现备份恢复传输的负载均衡，采用"最快优先"的策略来选择下载点，即选择当前网络负载最小、连接代价最小的节点来下载选中的编码包。

1.3 MSB&RT传输系统结构

MSB&RT传输系统由控制中心、源节点及副本节点组成。图1为一个具有5个节点的容灾系统联盟示意图。所有节点都与源节点0的控制中心(CC)不断交换信息，源节点0与副本节点1、2、3、4都建立了连接，同时2与3、4与3、3与1之间也都有数据传输，通过箭头来表示编码包传输的方向。联盟中的所有节点之间就形成了一个网状的结构。

控制中心拥有所有联盟节点信息以及编码包的分布状况，所有节点都与控制中心联系，在它的控制下根据上述的"最小者优先"和"最快者优先"两个原则进行下载与上传。由此可见，控制中心在MSB&RT传输系统中的地位是至关重要的，它的失效会造成整个备份恢复传输的失败。因此，我们将MSB&RT控制中心纳入容灾系统的控制中心内。容灾系统联盟中每个节点上都存在一个MSB&RT控制中心，谁作为源节点备份数据，就启用该节点上的MSB&RT控制中心。这就将MSB&RT控制中心的可靠性和节点的可靠性等价起来，避免了控制中心失效后整个联盟的MSB&RT传输系统都失效的问题。

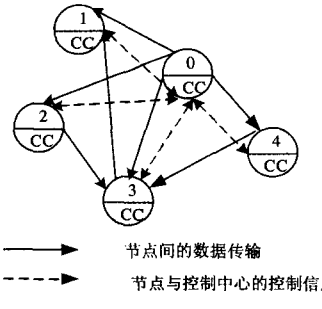

图1 事故恢复系统联盟中的MSB&RT转输结构

控制中心需要实时维护2个表：节点信息列表和编码包信息列表。节点信息表中，容灾系统联盟的各节点可用一个五元组来定义：

$\{P_i \mid P_i=\langle\langle i,A_i\rangle,L_i,F_i,O_{ij},U_{i,j}\rangle, i\in[0,r], j\in[1,k]\}$

其中$\langle i,A_i\rangle$为节点标识符，i为节点编号，源节点$i=0$，副本节点$i=1,2,\cdots,r$(r为备份副本总数)。A_i为节点的地址信息，包括IP地址、端口号。L_i为某节点最近一次与控制中心通讯时的负载情况。F_i为某节点完成备份的标志。源节点在备份阶段$F_0=1$，在恢复阶段$F_0=0$；备份节点i未成功备份时$F_i=0$，备份完成时$F_i=1$。$O_{i,j}$表示节点当前拥有的编码包列表，其值为编码包编号列表。$U_{i,j}$表示节点当前未拥有的编码包列表，其值为编码包编号列表。k为LECC编码生成的k个编码包。

编码包信息表包含当前拥有某一编码包的节点数量及所处的位置信息，其中所示的编码包可用一个三元组来定义：

$\{EP_j \mid EP_j=\langle\langle j,H_j\rangle,N_j,M_{L_{i,j}}\rangle, i\in[0,r], j\in[1,k]\}$

其中$\langle j,H_j\rangle$为编码包标识，j为编码包编号，在生成编码包时按顺序标识；H_j为该编码包的$hash$散列值。N_j为当前拥有此编码包的节点数量，初始值为0。$M_{L_{i,j}}$为当前拥有此编码包的节点位置列表，其值为节点编号列表。

源节点及副本节点除了与控制中心交换控制信息外，节点之间还进行数据交换。每个节点在本地也维护一个可供下载的编码包列表。其描述为

$\{LP_j \mid LP_j=\langle j,H_j\rangle, j\in[1,k]\}$

2 MSB&RT算法流程

MSB&RT算法流程具有各节点对控制中心的编码包注册、控制中心为节点选择下载编码包以及节点间的编码包传输三个阶段。其中编码包传输分为备份传输和恢复传输。下面对这三个阶段分别进行描述。

2.1 注册

MSB&RT传输的所有操作都是在控制中心统筹下进行，控制中心维护所有节点的信息及所有编码包的信息。在此基础上，各副本节点进行下载选择，通过控制中心的指挥进行传输。任何节点拥有的编码包，如果没有向控制中心正确注册，将不能参与上传。

1)源节点可用编码包注册

(1)源节点将需要备份的文件经过LECC编码生成k个编码包，并计算其散列值H_j，发送信息$\langle 0,L_0,H_j\rangle$给控制中心，运行操作请求注册：$R_{request}=\langle\langle 0,L_0\rangle,H_j\rangle$。

(2)控制中心接收到源节点的注册请求后，对编码包进行顺序编号，依次赋予一个ID，将散列值对应写入编码包信息表的H_j中，将N_j加1，源节点编号0及其刚分配的编码包ID添加到$M_{L_{0,j}}$中。同时填写节点信息表中节点0的信息：更新L_0，将编码包ID写入$O_{0,j},U_{1,j},U_{2,j},\cdots,U_{r,j}$列表中。完成这些注册操作后，给源节点返回注册信息$\langle j,H_j\rangle$。

(3)源节点收到注册应答$R_{answer}=\langle\langle j,H_j\rangle,Registered\rangle$后，把信息$\langle j,H_j\rangle$写入可供下载的编码包列表中。

2)副本节点的编码包注册

(1)副本节点m成功拥有一个编码包j后，将$\langle\langle m,L_m\rangle,\langle j,M_{h_j}\rangle\rangle$发送给控制中心请求注册。其操作为：$R_{request}=\langle\langle m,L_m\rangle,\langle j,H_j\rangle\rangle$。

(2)控制中心接收到副本节点的注册请求后，将N_j加1，m添加到$M_{L_{i,j}}$列表中，更新L_m，将j从$U_{m,j}$列表中移除并添

加到$O_{m,j}$列表中，并返回注册信息$\langle j, H_j \rangle$。

(3)副本节点收到注册应答$R_{answer}=\langle\langle j, H_j\rangle, Registered\rangle$后，把信息$\langle j, H_j\rangle$写入可供下载的编码包列表中。

2.2 文件备份

当第一个编码包完成注册后，控制中心就启动下载选择算法，开始进行数据传输，传输的过程与注册的过程可以同时进行。数据的备份过程中，数据不仅从源节点传输到副本节点，副本节点之间也相互传输数据。备份过程的数据传输形式如图3所示。

在备份开始阶段的主要任务是尽快把源节点的编码包分发出去。此时控制中心采用“最少者优先”的编码包选择策略，比较编码包信息表中的N_j项，选出N_j中最小的非零值N_i，其对应的编码包i即被选定传输的编码包。比较所有不在表$M_{L_{i,j}}$中的节点，选择出当前负载最轻的节点L作为下载节点。控制中心给L节点发送备份数据下载信息$\langle A_0, i, H_i\rangle$，控制副本节点$L$与源节点建立连接传输编码包$\langle i, H_i\rangle$。

当所有副本节点拥有的编码包的并集足够解码得出原始数据时，即便源节点出现故障而退出，备份传输也可以完成，随即就可以对其进行恢复。这一状态称之为MSB&RT传输拥有完全的健康度，它可以根据编码包信息表中的N_j项判断出，即N_j中所有的值都大于2。接下来把提高传输效率为主要任务，控制中心开始转为采用“最快者优先”的节点选择策略，控制中心并行地为F_i标志为0的各个副本节点选择下载编码包。对于副本节点m，控制中心从其尚未得到的编码包中选出分布最少的编码包i进行下载传输。

然后从$M_{L_{i,j}}$中选出当前负载最轻的节点M，作为编码包i的上传节点。控制中心给节点m发送备份下载信息$\langle A_M, i, H_i\rangle$，命令节点$m$与$M$建立连接下载编码包$\langle i, H_i\rangle$。副本节点$m$接收到控制中心的下载控制信号后，根据$A_M$中的节点地址信息开始从节点$M$下载该编码包。

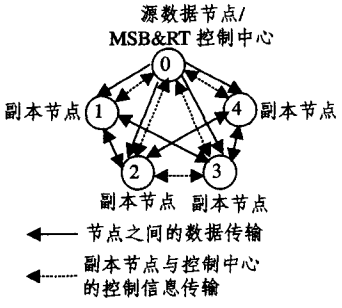

图2 事故恢复系统联盟使用*MSB&RT*算法进行备份的过程

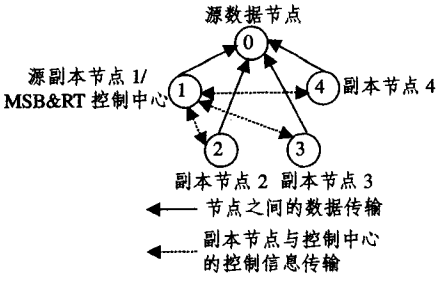

图3 事故恢复系统联盟使用*MSB&RT*算法进行恢复的过程

当副本节点m接收到足够多的编码包解码得到原文件时，给控制中心发送消息$\langle m, F_m\rangle$，告知控制中心副本节点m已经完成备份。同时该节点开始行使等同于源节点的功能，向控制中心发送消息$R_{request}=\langle\langle m, L_m\rangle, H_m\rangle$，作为另一个“源节点”请求编码包注册。

当控制中心接收到副本节点m的备份完成消息后，将F_m标记置为1，并对其新生成编码包进行注册。如果所有节点F值都为1，则表示备份结束，控制中心发送备份成功信息，所有节点结束数据传输。

2.3 数据恢复

MSB&RT的恢复过程相当于多对一的数据传输过程。容灾系统首先进行副本节点的选择，选出s个较优的节点参与数据恢复的传输过程，其中最优的一个副本节点作为MSB&RT控制中心。与备份过程相反，选出的s个副本节点作为编码包的“源头”，原数据节点是下载点。各个副本节点用对副本文件进行LECC编码，产生k个编码包，并计算其散列值，在控制中心进行编码包注册。控制中心将所有编码包信息发送给原数据节点，但并不给原数据节点下载策略。原数据节点并发s个线程和s个副本节点建立连接，所有副本节点同时向原数据节点传输编码包。备份过程中的数据传输形式如图4所示。

当原数据节点接收到足够多的编码包，解码生成数据文件，并计算其散列值。如果其散列值和原文件散列值不一致，就向控制中心发送信号，清除$O_{0,j}$列表中所有标记，请求副本节点重发所有编码包；如果散列值一致，给控制中心发送信息$\langle 0, F\rangle$，通知控制中心恢复过程完成。当控制中心接收到原数据节点的传输完成消息后，同样将此消息转发给s个副本节点，所有副本节点停止数据传输。

3 系统仿真实验

3.1 仿真实验环境

为了分析仿真环境下容灾系统各模块的性能，现由五台主机构建一个容灾系统联盟试验环境，主机1及主机2安装100M网卡，主机3、主机4和主机5安装1000M网卡。其中主机1、2、3、4分别位于局域网中，局域网为千兆以太网。另外，为了验证MSB&RT传输系统在广域网上的性能，主机5通过ADSL宽带接入电信网。主机1为源数据节点，其他主机为其副本节点。

3.2 MSB&RT性能测试

为了验证MSB&RT算法在容灾系统联盟的备份和恢复传输中的性能，在试验环境中利用逐点传输、并行传输及MSB&RT传输分别对主机1上的512M数据文件向其他副本节点进行备份传输，并比较它们之间的性能差别。

表1给出逐点传输、并行传输及MSB&RT传输三种备份方式下各副本节点完成备份的时间，其中总完成时间为完成4个副本节点数据备份所需的时间。

表1 三种传输方式完成时间比较

	主机2	主机3	主机4	主机5	总完成时间
逐点传输/s	125.60	113.64	118.95	5051.66	5409.85
并行传输/s	144.72	126.86	135.54	5116.45	5156.45
MSB&RT传输/s	105.20	93.08	81.39	1186.62	1186.62

MSB&RT传输过程中，各个副本节点皆分担了一部分负载(表2所示)。完成所有副本节点的数据备份后，源数据节点的总传输数据量(仅上传)为878.28M，仅为逐点传输和并行传输的传输数据量2048M的43%，有效地减小了源数据

(下转第117页)

可以返回部分展品的虚拟展览，如表 2 所示。

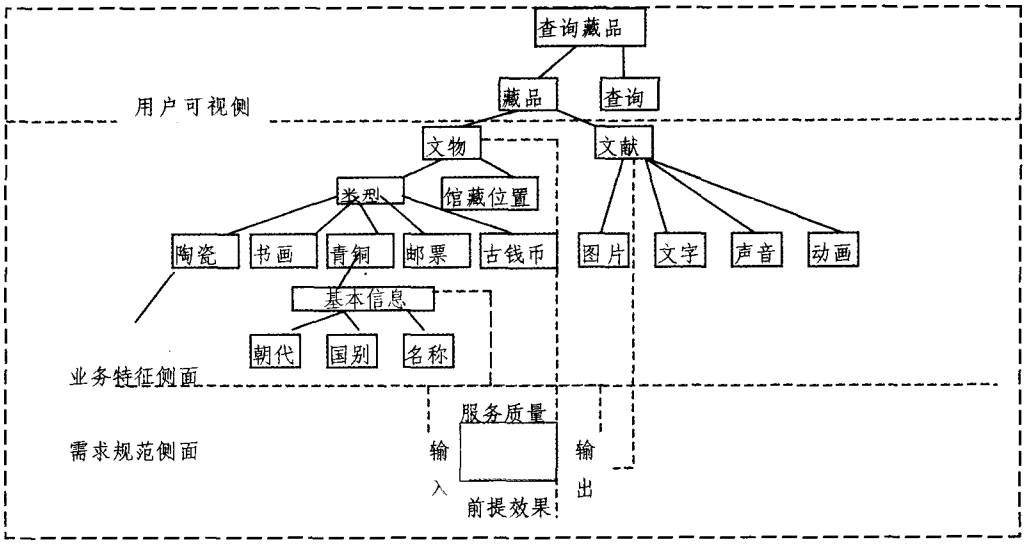

图 5　查询藏品业务服务

表 2　服务列表

Web 服务	输入	输出	提供者
#查询藏品	#藏品名称 Or#藏品分类 or#藏品朝代	#藏品文字信息 #藏品图像 #藏品相关论文目录	通用平台
#资料检索	#文著名称	#文著内容	首都博物院
#虚拟展览	#藏品名称	#三维动画	上海博物馆

表 3　藏品出入库服务

Name	#藏品入库	Desc	藏品信息的入库
Input	#藏品名称或#藏品类别	Ouput	#藏品入库情况

若用户希望使用这个平台得到藏品的专著资料，以及该藏品的三维效果图，领域专家进行分析后发现独立的查询都满足不了用户的需求。因此可以在采用虚拟化机制，将三个 Web 服务进行组合转换，返回给用户一个包含所需信息的界面。

通用平台中的藏品出入库服务，其描述如表 3。藏品入库时首先应该先检索库中是否已有该物品。若没有，则藏品入库，否则按情况修改。需要调用查询藏品的服务。因此可以使用服务虚拟化的方法将二者组合起来，提供给用户藏品入库的接口，屏蔽处理细节。

结束语　服务虚拟化机制的目的是利用已有的 Web 服务实现业务服务功能，保证业务服务能透明落实到 Web 服务，从而支持最终用户编程。本文通过研究虚拟化的运作机制，并对服务虚拟化相关方法进行尝试，结果表明，采用服务虚拟化可以有效降低最终用户构建应用的时间和难度。后续工作将深入研究基于虚拟化方法建立关联时的效率问题。

参 考 文 献

1　赵卓峰，韩燕波，喻坚，等. 一种支持业务用户编程的服务虚拟化技术-VINCA 聚合服务机制. 计算机研究与发展，2004，41(12)：2224～2230
2　Fan W, Wu Z, Yang J. Approximate Common Structures in XML Schema Matching. In: Pr-ceedings of Web-Age Information Management Conference. Hangzhou, China, 2005
3　顾宁，刘家茂，柴晓路，等. Web Services 原理与研发实践. 北京：机械工业出版社，2006
4　林海略，刘晨，王建武，等. 面向业务领域的服务建模方法及支持框架. 信息技术快报，2006，4(3)：10～17

(上接第 106 页)

节点的上传负载。

表 2　MSB&RT 算法备份性能测试结果

	主机 1	主机 2	主机 3	主机 4	主机 5
下载平均速度(kB/s)	——	4983.72	5632.96	6441.38	441.83
上传平均速度(kB/s)	757.91	243.85	339.03	406.95	220.67
总上传量/M	878.28	282.58	392.87	471.60	22.67

实验仅备份 512M 数据，在广域网环境中就花去 1 个半小时。如若网络条件更差的话，对于海量数据的消耗时间将是不可容忍的，异地多点数据备份将成为不可能完成的任务。使用 MSB&RT 算法，在现有网络环境下，即可以快速完成网络异地多点备份，而无需花费高昂的代价来建立专线网络。

结论　MSB&RT 算法实现了快速的网络异地多点备份和恢复。在备份过程中，源数据节点向各副本节点并行传送编码包，与此同时各副本节点之间也互相传送互补的编码包；在数据恢复传输过程中，多副本节点同时向原数据节点传送相异编码包，提高数据恢复的速度及各个副本节点的利用率。MSB&RT 传输方法结合 LECC 编码，使得容灾系统在现有条件下，既能实现快速的海量数据备份和恢复，又能有效地提高数据的安全性和可用性，具有很高的实用价值。

参 考 文 献

1　Bhattacharyya S, Kurose J F, et al. Efficient rate-controlled bulk data transfer using multiple multicast groups [J]. In: Proc. IEEE INFOCOM, 1998. 1172～1179
2　Byers J, Luby M, Mitzenmacher M. A Digital Fountain Approach to Asynchronous Reliable Multicast [J]. IEEE Journal on Selected Areas in Communications, 2002, 20(8): 1528～1540
3　Byers J, Luby M, Mitzenmacher M. Accessing multiple mirror sites in parallel: Using Tornado codes to speed up downloads [J]. Proc IEEE INFOCOM, 1999, 1: 275～283
4　万淑超，金蓓弘，黄宇. P2P 平台的关键技术[J]. 计算机科学，2005，32(6)：21～24
5　张铁军，张玉清，战守义. Peer-to-Peer 典型应用安全需求分析[J]. 计算机工程，2005，31(20)：56～58
6　李钟华，李伟华，武鲁. 网络灾备的传输补偿技术研究[J]. 计算机科学，2005，32(12)：58～60