

FRS 模块设计及其协作关系

任 堃, 李战怀, 王 超, 姜 楠

(西北工业大学计算机学院, 西安 710129)

摘 要: 容灾系统的基础是实现数据在线复制。论述 FreeBSD 平台数据复制系统(FRS)的结构, 给出其模块设计和模块间的协作关系。设计并实现一种 FRS, 其中嵌入了通用复制设备驱动层。该系统不依赖具体的存储设备或逻辑卷管理器, 具有很强的独立性、灵活性和通用性, 能实现数据块级别的复制。

关键词: 数据复制系统; 通用复制设备驱动层; FreeBSD 平台数据复制系统

FRS Module Design and Its Collaborative Relationship

REN Kun, LI Zhan-huai, WANG Chao, JIANG Nan

(School of Computer Science and Technology, Northwestern Polytechnic University, Xi'an 710129)

【Abstract】 Online data replication is the basic part of disaster-tolerate system. This paper discusses the system architecture and the module architecture of the FreeBSD platform data Replication System(FRS), and presents how these modules cooperate. It designs and implements a FRS which embeds general replication device driver layer. The system has a strong independence, flexibility and versatility, and can achieve replication on data block level while maintain independent on specific storage devices and logical volume manager.

【Key words】 data replication system; generic replication device driver layer; FreeBSD platform data Replication System(FRS)

1 概述

在信息化浪潮的推动下, 越来越多企业和组织开始构建自己的信息化系统, 产生了大量数据。数据资源是不可替换的, 数据的损坏或丢失会造成不可弥补的损失。各种不确定因素导致本地容灾系统已不能满足对计算机系统容灾的需求, 因此, 通常需要把数据备份到异地, 以保证在各种情况下数据都不会丢失。远程复制^[1]是实现远程数据容灾的主要手段。目前已有众多知名研究机构、高校和 IT 公司在 Linux 和 Windows 平台上开展了相关研究, 成果显著, 但基于 Unix 平台的研究较少。FreeBSD 是 Unix 家族中最稳定和优秀的操作系统之一, 也是应用互联网服务器的最佳操作系统, 很适合作为 FTP 服务器、WWW 服务器、电子邮件服务器等网络服务器系统。

远程复制的实现可分为阵列级和主机级。基于阵列级的复制通常是将数据以磁盘块级别从一个磁盘阵列拷贝到另一个完全相同的阵列, 阵列之间需要专用的通道和设备进行连接, 具有通用性强、对应用程序影响小的特点, 但系统构建的成本很昂贵。基于主机的存储复制通过软件实现, 主要分为 2 类: (1) 采用文件复制和应用程序对象复制, 实现起来相对经济, 但通用性较差; (2) 独立于上层应用(文件系统、数据库系统和其他应用)的数据块级别的远程复制, 通用性强。目前, 基于数据块的数据复制受到研究者的关注, 主要代表产品有 VVR(Veritas Volume Replicator)^[2]、LVR(Linux Volume Replicator)^[3]等。

VVR 能提供面向 Unix、Windows、Solaris 操作系统的服务, 但它需要和自带的卷管理器配套, 价格较高, 是面向高端的容灾产品。LVR 仅提供 Linux 平台上的复制容灾, 且在一定程度上依赖 Linux 的 LVM(Logical Volume Manager)逻辑卷系统, 其独立性和通用性不强。目前几乎没有基于 FreeBSD

平台的数据复制系统, 鉴于此, 本文在这方面开展研究。

2 FRS 通用复制设备层

在现有物理存储设备的基础上虚拟出逻辑设备层, 对逻辑设备进行划分的基本单位称为卷或逻辑卷。与物理卷相比, 逻辑卷提供更高的可用性且更易于管理, 对上层应用来说, 数据存放的具体物理位置是透明的。Vinum^[4]是 FreeBSD 平台下的一款优秀的逻辑卷管理器, 它增加了系统可靠性和性能, 并提供了更丰富的管理功能。

FreeBSD 平台数据复制系统(FreeBSD platform data Replication System, FRS)通过创建和管理 FRS 逻辑卷来管理复制对象, 其通用复制设备层的层次关系如图 1 所示。FRS 通用复制设备层(即 FRS 逻辑卷)本质上是一个虚拟设备驱动程序, 它在物理存储设备或已有的逻辑卷基础上产生一个抽象层, 既可以管理物理存储设备, 也可以管理已有的逻辑卷, 如 Vinum 的逻辑卷。它能在 FRS 逻辑卷上创建文件系统、数据库系统以及构建其他应用。FRS 利用了逻辑卷的上述优点, 又不依赖某种逻辑卷管理器, 具有更强的独立性、扩展性和通用性。

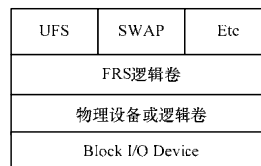


图1 通用复制设备层的层次关系

基金项目: 国家自然科学基金资助项目(60720106001)

作者简介: 任 堃(1986 -), 男, 硕士研究生, 主研方向: 数据存储, 数据容灾, 数据管理; 李战怀, 教授、博士、博士生导师; 王 超、姜 楠, 硕士研究生

收稿日期: 2010-02-16 **E-mail:** renkun.nwpu@gmail.com

3 FRS 的基本原理

本文提出的 FreeBSD 平台远程复制系统是嵌入了通用复制设备驱动层的复制系统,使用通用设备 FRS 逻辑卷进行复制,该层在文件系统以下、在所有设备之上,屏蔽了下层存储设备细节,不依赖特定的逻辑卷,实现了任意块设备上的实时复制,具有更强的独立性和通用性。数据复制的单元是上层应用通过操作系统提交的磁盘块,独立于具体的数据类型,系统的基本原理如图 2 所示。

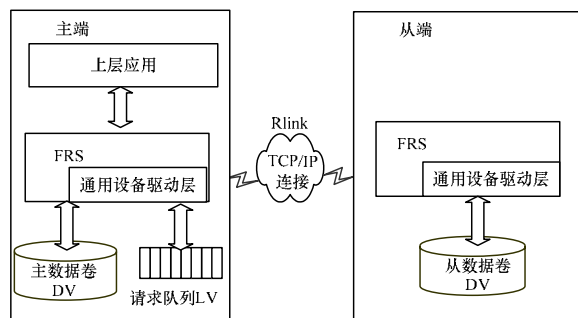


图 2 FRS 系统基本原理

在 FRS 中,复制是以复制卷组(Replication Volume Group, RVG)为单位进行的。RVG 由一个日志卷(Log Volume, LV)和若干个数据卷(Data Volume, DV)组成。LV 按照 RVG 中数据产生的顺序对其进行记录,因此,出现网络阻塞或故障时也不会丢失数据。DV 用来存放应用的数据。Rlink 负责主从端之间的数据传输和通信,确保数据复制的完成。

FRS 的基本原理如下:先截获主端上层应用的写请求,然后对请求进行分析,如果是作用在需要进行复制的卷上的写请求,则按请求次序写入日志卷及数据卷中,再按照请求的顺序把请求发送到从端,从端接收数据并写入从端的数据卷。

为确保从端数据可恢复,必须保证数据一致性。为达到该目的,FRS 严格执行写顺序的一致性,即数据在从端与在主端的更新顺序相同,以主端请求到达 RVG 的顺序将数据保存到 LV 中,并按此顺序将数据发送到从端,保证在从端按照相同顺序写入磁盘。数据传输通过 TCP/IP 协议实现。TCP/IP 协议是目前使用最广泛的协议,不需要专用的网络和设备。

4 FRS 内核模块的设计与实现

根据以上基本原理,对相似功能进行归并,可以归纳出系统主要模块包括主端请求截获模块、主端数据处理模块、从端数据处理模块、主端网络传输模块和从端网络传输模块。

4.1 各个模块的功能

每个模块都以多个线程的形式存在,下文给出每个模块的功能。

4.1.1 请求截获模块

请求截获模块被设计成过滤驱动程序,是 FRS 的基础,因为它决定了哪些数据应该被复制,哪些不用。通过截获上层应用请求,与复制无关的卷上的写以及所有读操作均交给 FRS 卷的下一层驱动进行处理,若是复制卷上的写操作,则将请求加入写请求队列,让 FRS 来处理。

4.1.2 主端数据处理模块

主端数据处理模块负责主端的磁盘 I/O 部分。在严格维持上层请求提交顺序的情况下,将请求队列中的请求分别写入日志卷和数据卷,并跟踪请求的完成情况,释放请求队列中已完成的请求,向上层应用返回成功。本模块由 PWT (Primary Write Thread)和 PCT(Primary Complete Thread)2 个

线程来实现。

4.1.3 从端数据处理模块

与主端数据处理模块相对应,从端数据处理模块主要负责从端的磁盘 I/O 部分,严格按照接收请求的顺序将其直接写入 DV,并跟踪从端请求的完成情况,通知网络传输模块发送给主端 ACK 确认信息。本模块由 SWT(Secondary Write Thread)和 SCT(Secondary Complete Thread)2 个线程来实现。

4.1.4 主端网络传输模块

主端网络传输模块负责主端数据传输和通信。按照主端接收请求的顺序读取日志卷中的数据发往从端,有时还需要发送一些控制报文通知从端进行相应的操作,如请求建立连接、判断从端是否存在、更改工作模式等,并负责接收从端发过来的 ACK 报文,确认数据在从端发送和写成功。本模块由 PRT(Primary Rlink Thread)和 PMT(Primary Message Thread) 2 个线程来实现,一个线程发送数据,另一个线程循环接收 ACK。

4.1.5 从端网络传输模块

负责监听主端的连接请求,接收主端发送过来的数据和控制信息,并解析报文,同时需要对主端进行确认,如确认从端数据 I/O 完成、状态更改完成、连接正常等。本模块由 LT(Listen Thread)、SRT(Secondary Receive Thread)、SPT(Secondary Parse Thread)、SAT (Secondary Ack Thread) 4 个线程协作实现。

4.2 模块间的协作

请求截获模块与主端数据处理模块的协作关系见图 3。

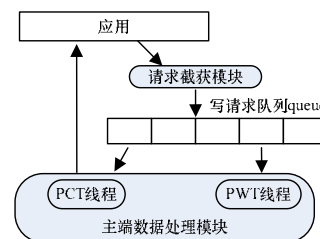


图 3 请求截获模块与主端数据处理模块的协作关系

有请求到来时,请求截获模块先对其进行判断。若需要复制,就先将它封装到一个 request node 结构中,此结构中包含了该请求的描述信息,然后将该 request node 添加写入请求队列 queue 中,主端数据处理模块中的 PWT 线程在严格维持上层请求提交顺序的情况下,从请求队列 queue 中取出数据,将数据分别写入日志卷和数据卷。主端数据处理模块的 PCT 线程扫描并释放请求队列中已完成的请求,以提供空间容纳后续的写请求,向上层应用通知请求完成。

主端数据处理模块与主端网络传输模块的协作关系见图 4。

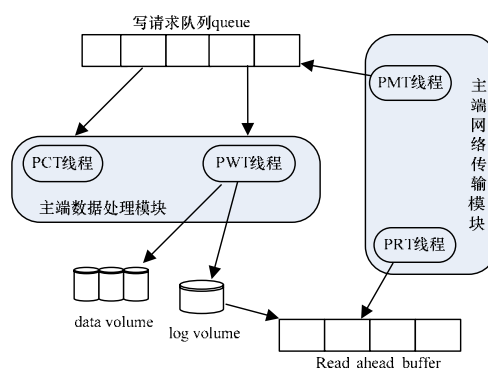


图 4 主端数据处理模块与主端网络传输模块的协作关系

在主端把请求数据顺序写入日志卷之后,主端网络传输模块的PRT线程按照主端接收请求的顺序读取日志卷中的数据并发往从端。为了减少主端读日志卷的次数、提高FRS性能,创建了发送缓冲机制,即设计read_ahead_buffer。它采用预先读策略,一次性地从日志卷中读取大量数据块,以减小从日志卷的读取次数。PMT线程接收从端发过来的报文的ACK,确认数据在从端发送和写成功。

从端网络传输模块与数据处理模块的协作关系见图5。

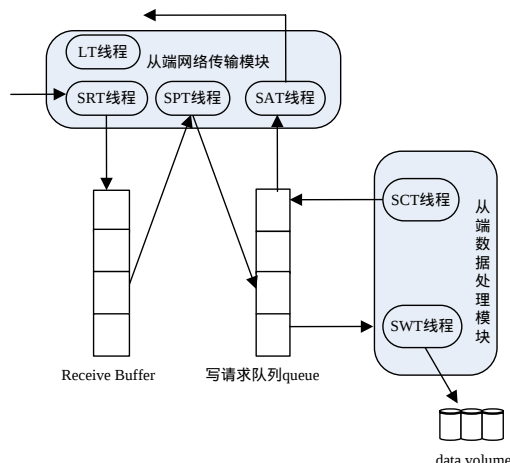


图5 从端网络传输模块与数据处理模块的协作关系

LT线程监听主端的连接请求,并对主端进行身份验证,如果身份认证通过,就派生出其他3个线程。

从端网络传输模块的SRT线程定长接收主端发过来的数据,而不关心具体的报文类型及内容,并将其插入接收缓存Receive Buffer中,SPT线程负责解析报文,过滤掉控制报文,并把要写入数据卷的报文插入到从端写请求队列queue中,通知从端数据处理模块中的SWT线程。SWT按照queue中请求的顺序依次写入数据卷。SCT循环检查从端写请求队列queue中的请求完成情况,并把要回复ACK的请求头交给SAT。SAT线程根据SCT返回的报文信息组装ACK报文并发送给主端。主端收到从端的写确认后,将日志卷中的该请求去除。

4.3 FRS 内核模块的实现

FRS的具体实现严格遵循本文阐述的模块划分。FRS的

(上接第50页)

可以看出,本文的设计方法与现有利用文件系统读取多媒体数据方法相比,数据读取速度提高显著,有效地满足了互动游戏的实时性要求。

4 结束语

本文根据互动游戏的特点和发展现状,研究了互动式游戏数据库的数据存储与管理,构建的互动游戏数据库具备商用数据库的优势,且更加经济实用。针对存储在外存的大容量多媒体数据,采用二级Hash方法实现数据的存储和管理;利用基于角色的关联规则提前预判并读取所需数据,提高了内外存数据的交换效率。

参考文献

- [1] Bohannon P, Lieuwen D, Rastogi R. The Architecture of the Dali Main-memory Storage Manager[J]. Journal of Multimedia Tools and Applications, 1997, 4(2): 115-151.
- [2] Baulier J, Bohannon P, Gogate S, et al. DataBlitz Storage Manager:

通用复制设备驱动层是通过分析FreeBSD的设备驱动机制和操作系统I/O请求分发流程设计出来的简易逻辑卷管理器。每个模块均以多个线程的形式存在,增强了模块内的并行性、模块间的并行处理和多个线程的协同工作能力,能充分发挥SMP的优势,提高系统性能。FreeBSD平台支持动态内核链接机制的特点,整个FRS以模块的形式提供给用户,可以方便无缝地在原有系统上提供复制功能。根据FreeBSD提供的通信机制,线程间的通信通过信号处理函数和事件等待唤醒机制实现。

5 评估与实验

为了验证实验设计的正确性,本文进行了原型实验。实验环境为跨过一台路由器的2台安装FreeBSD 5.5的PC机,分别作为主从端。实验测试了不同上层应用的数据,如文件系统和数据库系统等,并测试了物理存储设备和逻辑卷设备,分别作为通用复制设备层的下一层。实验结果证明,FRS能独立于上层应用进行复制,不依赖具体的存储设备,可靠地实现了复制功能,说明本文所述模块设计具有科学性和高效性。

6 结束语

FRS通过将数据完整地复制到若干个从端,对上层应用提供高可用性。本设计方案根据请求数据的系统处理流程,对整个复制过程进行模块划分,各个模块相对独立、协同工作。因为对数据的处理做到了线性化,即请求到达的顺序、主端I/O的顺序、网络发送的顺序和从端I/O的顺序保持一致,所以有效保证了数据的写顺序一致性,从而实现了从端数据的一致性。

参考文献

- [1] Patterson H, Manley S, Federwisch M, et al. SnapMirror: File-system-based Asynchronous Mirroring for Disaster Recovery[C]//Proc. of the 1st USENIX Conference on File and Storage Technologies. Monterey, CA, USA: [s. n.], 2002: 117-129.
- [2] VERITAS Volume Replicator Whitepaper[Z]. VERITAS Corporation, 2002
- [3] 董欢庆,李战怀,林伟,等. Linux平台远程逻辑卷复制系统的设计[J]. 计算机工程与应用, 2004, 40(18): 109-112.
- [4] Lehey G. The Complete FreeBSD[M]. [S. l.]: O'Reilly Publishing, 2006.

编辑 陈 晖

Main-memory Database Performance for Critical Applications[J]. ACM SIGMOD Record, 1999, 28(2): 519-520.

- [3] Francik E, Rudman S E, Cooper D, et al. Putting Innovation to Work: Adoption Strategies for Multimedia Communication Systems[J]. Communications of the ACM, 1991, 34(12): 52-63.
- [4] Vakali A, Terzi E. Multimedia Data Storage and Representation Issues on Tertiary Storage Subsystems: An Overview[J]. ACM SIGOPS Operating Systems Review, 2001, 35(2): 61-77.
- [5] Hirzalla N, Falchuk B, Karmouch A. A Temporal Model for Interactive Multimedia Scenarios[J]. IEEE MultiMedia, 1995, 2(3): 24-31.
- [6] Bertino E, Ferrari E. Temporal Synchronization Models for Multimedia Data[J]. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(4): 612-631.

编辑 顾姣健