第25卷 第3期 计算机工程与设计 2004年3月
Vol. 25 No. 3 COMPUTER ENGINEERING AND DESIGN Mar. 2004

文章编号：1000-7024(2004)03-0334-03 中图分类号：TP302 文献标识码：A

一种新的文件系统元数据的检查点容错策略

秦 航，徐 婕

(华中科技大学 计算机学院，湖北 武汉 430074)

摘 要：针对目前在集群文件系统中出现的元数据的故障问题，在PVFS的基础上提出了一种新的元数据检查点的日志管理策略。该策略在Linux环境下实现，解决了文件系统中较慢的元数据管理这一瓶颈问题，并且具备了较强的容错功能。该方法采用磁盘日志和内存日志的结构，通过对事务的管理，能满足集群文件系统中元数据高可用性的要求。

关键词：元数据；PVFS；检查点；日志恢复

Novel checkpoint fault-tolerance scheme of metadata in file system

QIN Hang, XU Jie

(Computer Institute of Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Based on the failure problems in the metadata of cluster file system, a novel metadata checkpoint journal system on PVFS was researched and developed. This system was fully implemented in the Linux environment and could solve the bottleneck in metadata manage of file system, so it was provided with fault-tolerance function. This method used the structure of disk journal and memory journal, and adopted a mechanism of transaction, so it could meet the request of high availability in file system metadata.

Key words: metadata; PVFS; checkpoint; journal recovery

1 引 言

PVFS(Parallel Virtual File System)[1]是由美国Clemson大学开发的一种基于Linux集群的并行文件系统,它针对并行计算,具备客户机/服务器结构。由于集群文件系统建立在集群上,所以结点机都会有失效的几率,即存储在元数据管理器(metadata manager, MGR)上的元数据在系统运转的过程中随时都有损坏或丢失信息的可能。研究发现PVFS中元数据是以文件形式存放,只有一个MGR节点,故在规模扩大后势必成为系统的一个瓶颈,并且会成为单一失效点。在此情况下,持续保证元数据的可用性,自动检测MGR的错误并进行灾难抢救,都是元数据高可用性和容错性必须考虑的问题。

当一个结点崩溃时,磁盘上的最后的一些关于元数据的操作会处于不一致状态,在重启系统时必须找回这些操作来纠正这些不一致性。考虑到现有的容错方法,在传统的那些没有日志管理的文件系统中,系统无法确定最后哪里发生变化,必须扫描所有MGR上的元数据结构来恢复一致性。这种扫描在当存储系统膨胀时代价极高。同时当操作系统先写入文件内容,然后等到有空的时候才写入文件的元数据时,如在写入文件内容之后但又在写入文件的元数据之前突然断电,文件系统仍然会产生不一致。本文提出的方案,在PVFS的基础上克服了上述磁盘扫描的缺点。它是通过一种日志的检查点策略来管理元数据,提高了集群系统[2]的容错能力。

2 元数据中的CRR技术

检查点设置和回卷恢复(Checkpointing and Rollback Recovery, CRR)技术是实现系统后向恢复的重要手段。CRR技术通过在系统正常运行过程中设置检查点(Checkpoint)[2]来保存系统当时的一致性状态,并对各个进程进行相关性跟踪和记录。系统中元数据发生故障后,将相关进程回卷(rollback)到故障前系统一致性状态,经过状态恢复后从该检查点处重新执行(而不是从程序开始执行),由此实现对系统故障的恢复。另外基于检查点的后向故障恢复机制可避免由于从头开始执行程序而引起计

基金项目：国家“863”高技术计划基金项目(2002AA1Z2102)。 **收稿日期**：2003-09-19；**修订日期**：2003-12-18。

作者简介：秦航(1980-)，男，湖北沙市人，硕士生，主要研究方向为集群技术、文件系统、元数据容错；徐婕，女，博士生，主要研究方向为集群于网格技术、并行与分布式处理、文件系统。

算上的大量浪费，充分提高集群系统的可用性。

在元数据管理中发生故障时，使用检查点可使受影响的进程从最后一次保存的检查点(状态)而不是进程开始重新运行[3]。此技术特别适合于保护长时间运行集群程序中出现短暂错误的情况。长时间运行的应用程序通常是运行数天或数月的程序，它对于这种程序重新开始恢复由错误造成的偶然损坏是不可接受的。

3 元数据检查点设计

3.1 基于日志的元数据可靠性问题

为达到系统的高可用性，提出了一种通过日志的方法来达到容错目的的检查点实现方案。因为使用日志用独立的日志文件跟踪磁盘内容的变化，故它比传统的文件系统安全。就像关系型数据库[4]一样，日志文件系统可用事务处理的方式来提交或撤消文件系统的变化。

在 PVFS 文件系统中，因为只有一个元数据管理器(metadata manager, MGR)节点，规模扩大之后势必成为系统的一个瓶颈，且会成为单一失效点；所以为了具备元数据的容错性，改进的设计的是文件系统中有多个 MGR。当一个元数据在一个 MGR 中被修改时，它必须在能够被另一个MGR使用前先被写到磁盘日志中，然后再被写到永久磁盘中。然而如果它能够在被写到磁盘日志后和在被写到永久磁盘前的这段时间内被其它的MGR使用，这里就最少可以省略一次磁盘访问。因为比较而言，网络传输时间比磁盘访问时间小，故整个事务处理时间能被减少。我们把这里的事务看做是一组形成一个稳定的状态转变的操作。

3.2 元数据日志的系统结构

PVFS中结点分为运行应用的计算结点(CN)、处理元数据的元数据管理结点(MGR)和存储文件数据的存储结点(IOD)。PVFS的客户端、元数据管理器端和存储服务器端都是在用户级实现的。考虑到它本身没有文件锁[5]，只是在客户端、元数据管理器端和存储服务器端维护了打开文件结构，因此设计了一个元数据锁机制。并且在PVFS管理器上，所有的元数据以本地文件系统的方式存放，其较慢的元数据管理器由此成为系统瓶颈。本地文件系统驻留在一个计算机内，无法被其它的文件系统访问，故对元数据的处理速度比较慢，而这里改进的MGR之间的协作恰恰能解决这一瓶颈问题。

同时要规定的一个日志事务是由在一个原子操作(atomic operation)中改变的元数据块所构成。每一个日志项有一个或更多的锁(lock)和它相关联，它通过特别的锁机制来与所保护的元数据相一致，如一个create操作能够容纳目录、索引节点和分配位图的锁。这里我们列出了一个元数据检查点和恢复机制，元数据在它被写到磁盘日志之后，立即能被其它结点共享。首先是检查点的设计，每一个MGR在磁盘上有它自己的一个日志空间，在其它的MGR之间它使用上述的锁策略从而得到保护，图1表示出了这个磁盘日志设计的例子，并假定系统有3个MGR。每个日志空间应该能够在一个MGR失败后被其它的MGR访问。

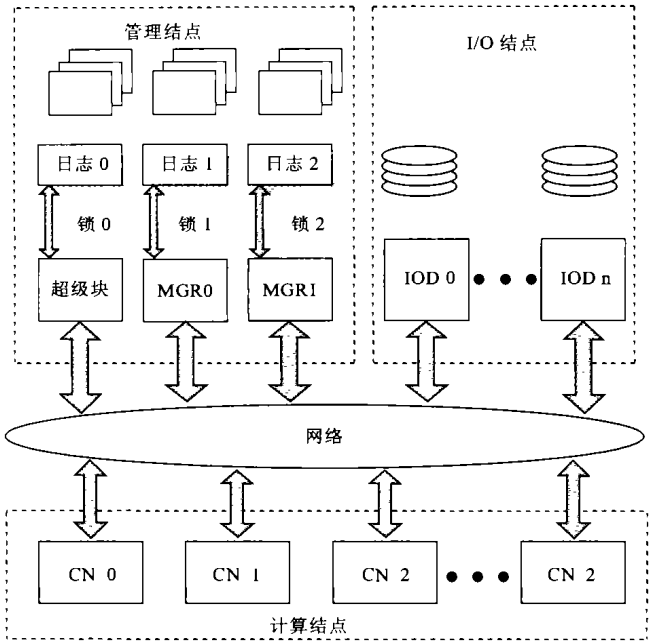

图 1 PVFS 中元数据检查点的结构

3.3 磁盘日志结构

其次讨论的是检查点数据结构，为了实现日志中的检查点，我们需要两种数据结构，一种是磁盘日志结构、另一种是内存日志结构。一方面，磁盘上的数据结构由一个元数据更新表、一个日志表头和一个元数据日志表所组成，如图2所示。日志表头被用来保存检查点的位置信息，即磁盘上的开始地址和结束地址。元数据更新表存储元数据信息，如元数据、永久磁盘地址、锁标志和更新号。这里的更新号是被用来在恢复时检测磁盘上的元数据是否被失效的MGR修改了时使用的。而且，元数据的锁标志保存已被修改的元数据。同时PVFS中文件

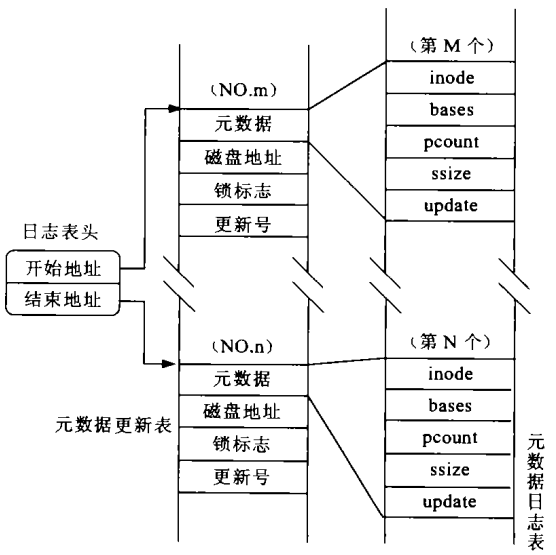

图 2 磁盘上的日志结构

的分布由3个元数据参数来表示:初始结点号,结点数,和分组的大小。

3.4 内存日志结构

另一方面,内存日志数据结构由一个元数据日志表、一个日志事务描述表和一个内存元数据更新表组成,如图3所示。数据日志表被用来保存内存上的日志项。日志事务描述表保存内存上的检查点开始和结束以及事务标志。与上述磁盘元数据更新表相似,内存元数据更新表保存元数据、元数据的磁盘地址、锁标志和更新号。同时这里还有一个cache日志表,它由等待MGR的队列、关闭(bolt)标记、延迟计数器(retard counter)、MGR标志、日志页号、锁标志、磁盘页号和缓存页号所构成。

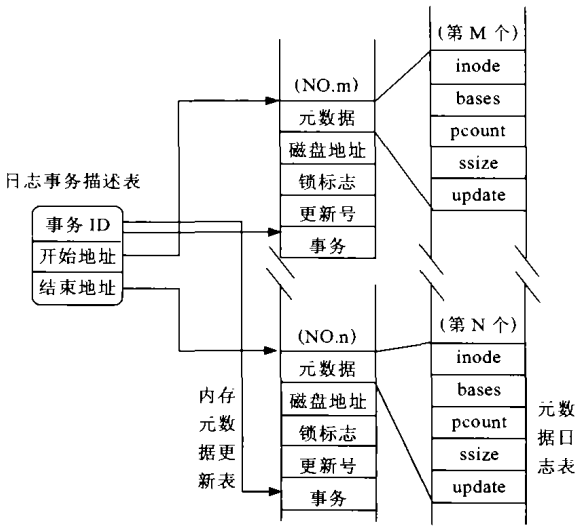

图3 内存上的日志结构

其中的关闭(bolt)标记表示页是否处于关闭位置,即页在被一个MGR使用时不应被写到永久磁盘上。延迟计数器(retard counter)表示一个MGR已经把元数据送到其它的MGR并且这个MGR已经开始计数。MGR等待一个消息,这个消息是来自其它MGR的已经收到元数据直到计数器终止(vanish)时的消息。正是因为MGR收到了消息,元数据已经被其它的MGR修改了和做下检查点了,它就不需要把修改的元数据写到永久的磁盘上,所以我们可以省略掉不必要的磁盘访问。最后的等待MGR队列保存仅仅是等待元数据的MGR标志。

4 检查点与恢复过程

4.1 事务管理和日志管理

为了能从日志中找回信息和能管理磁盘上的自由空间,来使磁盘上大范围的自由空间能够被用来写新的元数据,再来看看检查点日志,检查点是通过改变文件系统状态来进行操作的。这些操作因为是原子的,所以文件系统能从一个一致的磁盘状态转移到另一个一致的磁盘状态。这些操作与PVFS的操作如creat, mkdir, write, unlink等相一致。通过这些处理,文件系统元数据能很快地返回到一致状态。这里有两个管理器被用在检查点中,一个是管理事务的事务管理,另一个是事实上处理检查点操作的日志管理。日志管理区别于事务管理,它容纳来自处理模块要被写的元数据并把它写到磁盘上。有关元数据的所有的变化、添加和改变需要被记录到一个文件系统中日志的那部分中去。每隔一段时间,文件系统在检查点回卷更新在磁盘上的索引结点,并释放文件中用不到的那些元数据旧块。

4.2 设置检查点算法

在日志管理中的把元数据记录到磁盘中的设置检查点算法描述如下:

```
{
从事务管理器中得到事务;
搜集异步的日志事务并进行磁盘检查点设置;
if(元数据已被从其它的MGR上收到){
  if(元数据不是从磁盘上读到的)
    发送一日志消息;
    }
if(如果有一个MGR等待元数据)
    把元数据送到这个MGR;
  else{
      设置延迟标记并且开始计数;
      if(如果一个日志消息从一个已发送元数据的MGR里收到)
          释放内存缓冲并且终止事务;
      }
  打开(unbolt)元数据缓冲,使之能被写到磁盘上;
  通过缓冲管理器提交给永久磁盘上;
  }
```

4.3 回卷恢复的算法

在数据的处理过程中,先是开始一个事务,并且获得所需的锁来完成事务,然后关闭(bolt)内存中的元数据缓冲,以防它被写到永久磁盘上,并且在这里修改元数据,最后把事务传到日志管理器上去,这就是一个事务的生成过程。最后是恢复的过程,恢复管理(recovery management)的作用是在它获得失败的MGR id和日志锁后再开始恢复。回卷恢复的算法描述如下:

```
{
  找到日志表头的开始地址(begin);
  找到日志表头的结束地址(end);
  for( ;对于每一个日志项; )
    找到日志项所有的相关的锁(lock);
compare(磁盘元数据的更新号,日志元数据的更新号);
if(磁盘元数据的更新号 > 日志元数据的更新号)
  进行恢复操作;
```

(下转第373页)

系统的运行参数和各部件的结构参数通过计算得到以上的各特征频率。以滚动轴承的特征频率计算为例，程序界面如图4所示。

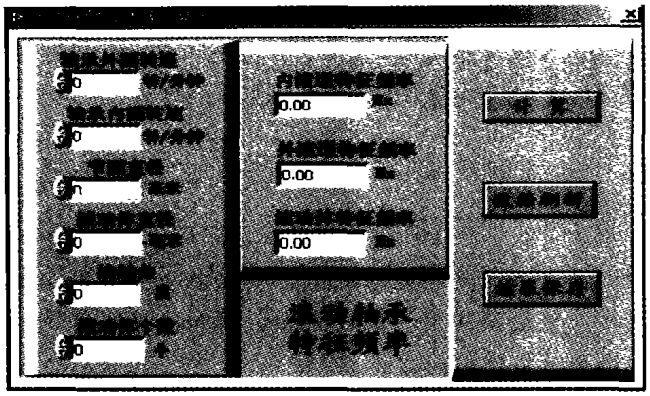

图4 滚动轴承特征频率计算程序界面

在“轴承外圈转速”、“轴承内圈转速”、“节圆直径”、“滚动体直径”、“接触角”、“滚动体个数”输入相应的零件参数，点击“计算”，在“滚动轴承特征频率”框中将显示各元件的特征频率值。点击“数据刷新”可以将前一次的参数刷新，为下一次计算做准备。点击“结果保存”可以将计算结果以文本“*.txt”的格式进行保存。

故障特征频率计算子程序可以快速、准确地计算出齿轮箱各部件特征频率，并进行保存，从而为今后的故障诊断提供依据。

4.2 齿轮箱振动信号分析程序的设计

本系统软件将信号的采集和分析一体化，信号采集完毕后程序会自动地进行数据处理和分析，得到振动信号的功率谱图，并将它和信号的时域图一起在前界面中显示，如图3所示。测试人员在完成振动信号的采集分析之后，可以根据时域和频域图中信号展示的信息，通过和故障特征频率的比较来得到相应的诊断结果，为维修提供理论依据。为了方便操作者的使用，设计了功率谱寻峰功能，可以快速地通过使用图中峰值寻找“+”坐标来找到功率峰值所在的频率值。

信号的分析程序包括信号的滤波和功率谱变换。程序框图如图2所示。

“Snd Read Waveform”是声音信号采集功能模块，信号通过一个“Scaled Time Domain Window”时域加窗模块对信号进行处理，而后在“Auto Power Spectrum”自功率谱模块中得到信号的频域图。

从程序框图中可以看到，LabVIEW是基于G语言的编译程序，它没有繁杂的代码语句，程序搭建只需要选择合适的功能模块并将它们适当地组合和连接，因而非常适合中小软件系统的程序设计。

5 小 结

本文介绍了一种利用普通笔记本计算机声卡作为振动信号采集工具的便携式齿轮箱故障诊断系统的开发过程。利用声卡完成信号采集工作，在保证信号采集完整性和信号分析准确性的同时，降低了系统硬件的开发成本。利用LabVIEW搭建了故障特征频率计算程序和信号采集、处理和分析程序，为故障测试人员提供了较为准确的判断依据。这套系统在中小企业的设备检测和维修中可以发挥重要的作用，有较好的应用前景。

参 考 文 献:

[1] 李国华,张永忠. 机械故障诊断[M].北京:化学工业出版社, 1999.

[2] 严志伟. 滚动轴承的故障诊断[J]. 轴承.1999, (1):31-33.

[3] National Instrument.LabVIEW user manual [CP].2001.

[4] Wang Changting. A virtual instrumentation system for integrated bearing condition monitoring[J]. IEEE Transactions on Instrumentation and Measurement, 2000,49 (2):325-332.

(上接第336页)

```
        else{保持更新号的先后顺序的一致性；
            一直等到所有在前的元数据新号已被恢复；
            }
    }
```

5 总 结

本文研究了一种在解决集群文件系统中出现的元数据故障问题的方案，并给出了在linux下的实现。它在现有的PVFS环境下为元数据在磁盘和内存中扩充了日志功能，从而具备了容错能力。本系统实现了故障恢复技术，为集群文件系统提供了高可用性。

参 考 文 献:

[1] Philip H Carns, Walter B Ligon. PVFS: A parallel file system for linux clusters[C]. 4th Annual Linux Showcase and Conference, 2000.

[2] Buyya R.高性能集群计算:结构与系统(第1卷)[M]. 北京:电子工业出版社, 2001. 334-337.

[3] Zambonelli F. Distributed checkpoint algorithms to avoid roll-back propagation[C]. Euromicro Conference, 1998 Proceedings 24th, 1998.403-410.

[4] 武剑锋,戈弋,李三立. 基于数据库的机群检查点的研究与实现[J].小型微型计算机系统, 2002, 23(3):257-261.

[5] Yong Kyu Lee, Shin Woo Kim, Gyoung Bae Kim,et al Metadata management of the SANtopia file system[C]. Parallel and Distributed Systems, 2001 ICPADS 2001 Proceedings Eighth International Conference, 2001.492-499.