基于内存功能划分的并行程序检查点策略研究

薛瑞尼 陈文光 郑纬民

(清华大学 计算机科学与技术系,北京 100084)

摘要:目前采用检查点设置技术的并行程序容错系统存在着不能透明处理通信环境变量的缺点,需要在设置检查点之前关闭进程间通信套接字,在恢复之后重新构建,为此提出了基于内存功能划分的通信隔离策略,分离计算模块和通信模块,避免对通信套接字的直接操作,完成了透明的容错功能.实验结果显示此策略对并行检查点系统性能有一定程度的改善,可以降低系统实现的复杂度,提高卷回恢复的可靠性,而且独立于并行系统,具有良好的移植性.

关 键 词:容错;检查点设置;卷回恢复;内存排除

中图分类号:TP302.8 **文献标识码**:A **文章编号**:1671-4512(2005)S1-0107-04

Checkpointing of parallel applications through differential memory functions

Xue Ruini Chen Wenguang Zheng Weimin

Abstract: As high-performance computing systems continue to grow in size and popularity, issues of fault tolerance and reliability turn into limiting factors on application scalability and system availability. Current fault tolerance systems for parallel applications through checkpoint/restart cannot handle the communication environment transparently. Sockets would be closed before checkpointing and reestablished after recovery, which is difficult to implement and prone to errors. "Communication exclusion" based on differential memory function is proposed to separate the communication and computation modules in order to avoid dealing with sockets directly. Experimental results indicate a little improvement on checkpointing performance. The strategy is helpful on reducing implementation complexity and improving recovery reliability, and is easy to be ported due to its independency to any parallel system.

Key words: fault tolerance; checkpointing; rollback recovery; memory exclusion

Xue Ruini Doctoral Candidate; Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

1 检查点设置技术

一个系统若不能达到其功能要求,则称为系统失效.容错技术即是保证程序在系统发生失效之后能够继续运行,并给出正确的输出结果[1].检查点设置与卷回恢复(CRR)技术是一种被广泛应用的、基于时间冗余的后向容错技术.概括来说,CRR技术包括以下两方面:

a. 检查点设置在系统正常运行过程中,由程序员或操作系统指定,在适当的时刻设置检查点,保存系统当时的一致性状态;

b. 卷回恢复如果系统发生故障,相关进程将被卷回到检查点中保存的故障前系统的一致性状态,并从此处继续运行,实现故障恢复.

对于并行系统,有如下定义:系统可靠性由t_{MTTF}(Mean Time To Failure)决定,即系统失效

收稿日期:2005-08-24.

作者简介:薛瑞尼(1981-),男,博士研究生;北京,清华大学计算机科学与技术系(100084).

E-mail: xueruini@gmail.com

基金项目:国家高技术研究发展计划资助项目(2002AA1Z2103).

前的平均正常运行时间;服务能力由 t_{MTTR}(Mean Time To Repair)决定,即失败发生后修复错误重新运转所需的平均时间;可用性 C 则定义为:

$$C = t_{MTTF}/(t_{MTTF} + t_{MTTR}).$$

常见的容错技术强调系统无故障运行,提高系统可靠性,重点在于延长 t_{MTTF}. 如前文所述,大规模并行系统的累计失效率会限制 t_{MTTF}的增加.检查点技术则注重提高服务能力,即缩短 t_{MTTR},尽量缩短系统的修复时间.两种策略的最终效果都是提高系统可用性.

2 通信隔离策略

检查点文件需要保存进程的所有特征数据,主要包括 CPU 信息和虚拟地址空间. CPU 信息一般采用系统调用 Setjmp 和 longjum 来保存和恢复,用户基本上无法干预.虚拟地址空间留给用户操作余地比较大,内存排除技术就是指将进程内存空间的堆(Heap)中的某些区域加以标注,在设置检查点的时候这些区域不被保存,从而在进程恢复的时候这些区域也不会被旧数据覆盖.这种技术主要用来处理检查点过程中可能遇到的较消耗资源的临时变量(比如矩阵,数组等),通过减小检查点文件来提高检查点性能[2].

现有并行程序容错系统的模型见图 1.各进程被视为独立的整体,所有进程协作形成并行运行环境中的一个并行程序,即并行程序 = ∑并行进程.所以,在设置检查点的时候,各进程的虚拟地址空间需要全部保存.由于单进程检查点工具本身的不足,其中的套接字就必须事先关闭,否则会导致前面所述的后果.

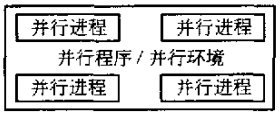

图 1 传统模型

基于内存功能划分的"通信隔离"策略系统模型见图 2.通过分析进程的特征数据,对内存空间按照功能划分, 每个并行进程由两个独立的部分组成:计算模块和通信模块.计算模块完成并行程序的核心计算任务,主要包括计算用到的变量;通信模块专门负责进程间通信(如发送接受消息),主要包括套接字句柄、进程分布信息等.如果对并行系统的层次进行抽象,一个并行程序的并行运行环境就是各个进程通信模块的联合,即:

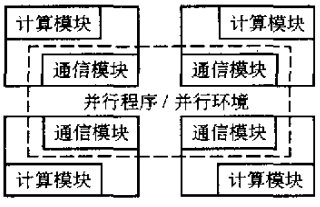

图 2 通信隔离模型

$$\text{并行程序} = \sum_{i=0}^{N-1} \text{并行程序}_i = \sum_{i=0}^{N-1} \text{通信模块}_i + \sum_{i=0}^{N-1} \text{计算模块}_i = \text{并行环境} + \text{计算环境}.$$

这个虚拟的并行运行环境正是并行系统区别于裸机集合的关键,但是它对检查点技术而言没有任何用处.检查点技术的核心是充分利用已完成的计算结果,避免重新计算.也就是说,计算模块是检查点真正应该保留的部分.通信模块只是维持计算模块能不断向前推进的辅助设施,它对卷回恢复没有意义.

通信模块和计算模块在程序中反映为不同的内存区域,利用内存排除技术在设置检查点之前将通信模块对应的内存区域排除在外.那么在进程卷回的时候,只有必要的计算模块部分被恢复,现有的通信环境不会受到影响.这种策略避免了目前容错系统所采用的关闭连接重建立的过程,可以有效地降低并行检查点技术实现难度,提高系统的可用性.

在并行程序中,消息队列是常见的消息缓存,它们属于计算模块而非通信模块.因为这些消息是进程间交换计算信息的载体,而传递消息的环节则属于通信部分.

下面给出采用协调式检查点设置协议,基于内存功能划分的"通信隔离"策略完成检查点设置以及恢复过程的详细步骤.检查点设置过程:

a. 启动并行程序,建立检查点设置策略(定时或者人工);

b. 设置检查点之前,根据协调方式清空通信通道,保证系统进入全局一致性状态;

c. 将各进程的通信模块加以标记,检查点文件只保存计算模块的数据;

d. 同步所有进程.

检查点设置完毕需要同步所有进程以保证继续运行的时候所有进程都已经完成检查点.如果缺少这一步则可能出现如下错误:A 进程正在进行检查点保存自身状态,B 进程已经完成检查点设置,并向 A 发送消息 m.A 被中断接收 m,导致被保存状态不再是同步时的一致性状态,因为 m

表现为孤儿消息(已经接收,但未发送).

如果并行程序中的某个进程发生故障无法继续运行的时候(可能由于节点失效、网络断开或者其他原因),必须进行卷回恢复:

a. 立即停止所有其他并行进程;

b. 按照完全相同的命令行重新启动故障程序;

c. 向所有进程发出卷回恢复信号;

d. 各进程暂停计算,读取检查点文件,进行状态恢复;

e. 同步所有进程.

发生故障的并行程序不能继续向前执行,一种策略是在故障状态下只恢复故障进程(记为P).当P被重新启动时,必须通过一个主控进程与其他进程交换信息.这个过程需要用户独立控制,属于典型的重新建立并行运行环境.本文采用的策略是将所有未发生故障的并行进程也全部停止,然后重新启动此并行程序,让其自动建立通信连接形成新的并行运行环境.此时所有进程开始卷回恢复,只有计算模块数据被替换,现有通信模块不会被覆盖,从而避免了通信环境重建.

同设置检查点一样,卷回恢复也需要二次同步,否则会出现如下情况.进程A已经恢复完毕,而进程B尚在恢复之中.此时A欲向B发送消息m,B接收之后继续恢复,可能导致m被覆盖,已表现为丢失消息(已经发送但没有接收),形成非一致性全局状态.

通过建立检查点服务器(保存所有进程的检查点文件),"通信隔离"策略可以实现对永久故障的容忍和进程迁移,即将失效节点上的进程(或欲迁移进程)的检查点文件从检查点文件服务器传输至正常节点(或目的节点),修改并行运行环境配置文件,指定失效进程(或迁移进程)被派生在正常节点(或目的节点)上,然后再执行前面所述恢复过程.

3　性能评测

基于Thckpt[①]实现了内存排除功能,将其作为ChaRM[②]的后台单进程检查点工具实现对MPI并行程序的容错功能.测试环境为8个节点的集群系统,每个节点的配置为:CPU 4×Xeon 700 MHz, RAM 1 Gbyte, Swap 1 Gbyte, Linux-2.4.20,千兆以太网.测试程序采用NASA NPB中的CG、MG和EP,在A数据集下分别运行2,4,8个进程.

"通信隔离"策略可以降低系统实现的复杂度,提高系统恢复时的成功率.这两个方面很难用确切的数量去评价,所以本文采用检查点和恢复时间作为主要衡量指标.检查点时间是指从准备设置检查点开始到完成同步结束为止,其中包括两次同步时间、清空信道时间以及单纯的写检查点文件时间.恢复时间则是从准备恢复开始到所有进程同步完成为止,其中包括读取检查点文件时间和同步时间.图3给出两个指标在"通信隔离"策略和常规策略下的比较结果,其中C_1和C_2分别为常规策略和"通信隔离"策略的检查点时间;R_1和R_2分别为常规策略和"通信隔离"策略的卷回恢复时间(横轴为测试进程数目,纵轴为对应开销).

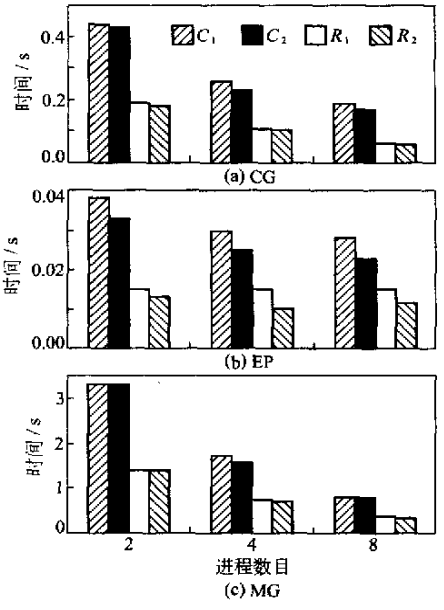

图3　系统性能

从图3中可以看出,采用"通信隔离"策略的系统开销普遍要小于常规策略.这是因为检查点设置时间和恢复时间主要取决于检查点文件的大小,进程同步和清空信道的时间相对可以忽略[3].通过内存排除标注通信模块可以减小检查点文件大小,因此可以降低系统开销.同时,由于

① Xue R N, Zhang Y H, Chen W G, et al. Thckpt: Transparent Checkpointing of UNIX Processes under IA64. Proc of Parallel and Distributed Processing Techniques and Applications(PDPTA'05), Las Vegas, 2005. 325—332

② Wang D, Deng X, Zheng W. ChaRM: A Checkpoint-based Rollback Recovery and Process Migration System for Cluster of Workstations. Proc of 2000 IEEE 4th International Conference on Algorithms and Parallel Processing(ICA3PP), Hong Kong, 2000. 708—709

通信模块相对于计算模块而言很小,排除通信模块所带来的性能提升也就比较微小.

参 考 文 献

[1] Elnozahy E N, Alvisi L, Wang Y M, et al. A survey of rollback-recovery protocols in message passing systems [J]. ACM Computing Surveys, 2002, 34(3): 375—408

[2] Plank J S, Chen Y, Li K, et al. Memory exclusion: optimizing the performance of checkpointing systems [J]. Software Practice and Experience, 1999, 29(2): 12—142

[3] 薛瑞尼. 面向集群系统的MPI并行程序容错技术研究[D]. 北京: 清华大学计算机系, 2005.

(上接第106页)

当 $t_{RTT}=1$ 时, $v=(9+125)/125=1.1$(实际测试在2.0~5.0)

当 $t_{RTT}=100$ 时, $v=(50\times9+125)/125=4.6$;

当 $t_{RTT}=200$ 时, $v=(100\times9+125)/125=8.2$;

当 $t_{RTT}=400$ 时, $v=(200\times9+125)/125=15.4$;

当 $t_{RTT}=800$ 时, $v=(400\times9+125)/125=29.8$.

图7显示在不同网络环境下传输小文件(1 byte~1 Mbyte)的加速比,其值是根据上面计算公计算得的近似值,反映了在网络环境比较差的环境下采用Socket复用技术对小文件传输起到比较大的作用.

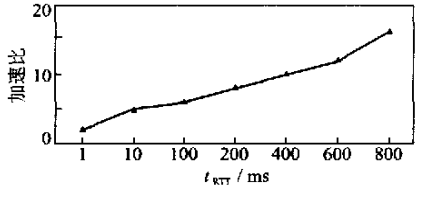

图7 数据传输时间对比图

未来的工作主要是加强G2NFS的系统监控能力的研究,影响系统的性能有很多因素,如网络状况、服务器负载等.其次进一步加强G2NFS的安全,一方面要加强身份认证的管理,另一方面要采用安全有效的访问控制机制.最后,进一步提高系统的负载均衡能力.

参 考 文 献

[1] Chervenak A, Foster I, Kesselman C, et al. The data grid: towards an architecture for the distributed management and analysis of large scientific datasets [J]. Journal of Network and Computer Applications, 2001, 23(2): 187—200

[2] 庞丽萍,王志平,吴 松等. 广域存储虚拟化的访问控制模型研究[J]. 华中科技大学学报(自然科学版), 2004, 32(1): 38—40

[3] 张文松,金士尧,吴泉渊. 一个虚拟Internet服务器的设计与实现[J]. 软件学报, 2000, 21(1): 122—125

基于内存功能划分的并行程序检查点策略研究

作者： 薛瑞尼， 陈文光， 郑纬民， Xue Ruini， Chen Wenguang， Zheng Weimin
作者单位： 清华大学,计算机科学与技术系,北京,100084
刊名： 华中科技大学学报（自然科学版） ISTIC EI PKU
英文刊名： JOURNAL OF HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY(NATURE SCIENCE)
年，卷(期)： 2005,33(z1)

参考文献(5条)

1. Wang D;Deng X;Zheng W ChaRM:A Checkpoint-based Rollback Recovery and Process Migration System for Cluster of Workstations 2000

2. Xue R N;Zhang Y H;Chen W G Thckpt:Transparent Checkpointing of UNIX Processes under IA64 2005

3. 薛瑞尼 面向集群系统的MPI并行程序容错技术研究 2005

4. Plank J S;Chen Y;Li K Memory exclusion:optimizing the performance of checkpointing systems 1999(02)

5. Elnozahy E N;Alvisi L;Wang Y M A survey of rollback-recovery protocols in message passing systems [外文期刊] 2002(03)