基于消息传递并行进程迁移技术的研究与实现

刘天田 杨升春 欧中红 袁由光
(武汉数字工程研究所 武汉 430074)

摘 要 高可用在并行计算环境中的地位日益突出。实现LAM/Migration扩展了LAM/MPI的进程迁移功能,可实现MPI整体任务在节点之间的自由迁移,其迁移功能对应用程序透明、智能化程度高,并可应用于集群节点容错与负载均衡,有效提高集群的可用性。

关键词 MPI,高可用,检查点,卷回,进程迁移
中图法分类号 TP338.8 **文献标识码** A

Research on Implementation of Message Passing Based Parallel Process Migration

LIU Tian-tian YANG Sheng-chun OU Zhong-hong YUAN You-guang
(Wuhan Digital Engineering Institute, Wuhan 430074, China)

Abstract High availability plays more and more important role in paralleled computing. This paper developed a new computing platform named LAM/Migration, which introduced process migration ability into LAM/MPI. By migrating whole MPI application among cluster nodes freely, the process migration function is intelligentized and transparent for application. LAM/Migration can implement node fault tolerance and load balance of cluster and improve the availability of cluster effectively.

Keywords MPI, High availability, Checkpoint, Rollback, Process migration

1 引言

MPI是并行程序设计的工业标准,因良好的可移植性被广泛应用于集群计算中。MPI任务中的进程与计算节点的映射关系是固定的,随着集群规模不断增加,进程崩溃、异常关机、节点硬件故障以及电磁干扰引起的节点间歇故障等,将造成系统可用度成指数下降。当某一个进程失效时,整个任务将面临崩溃。大型工程计算任务执行时间长,一旦发生上述事件,整个任务将重新执行,导致计算资源与时间的浪费。所以要求MPI具备容错能力来保证系统在故障期间仍能为用户提供正确的、不间断的服务[1]。

检查点与卷回恢复技术(Checkpoint and Rollback Recovery)是分布式计算常用的容错方法。MPI应用检查点记录任务正常运行过程中的一致性状态,并存到可靠存储介质上。当系统发生故障后,根据检查点将任务卷回到故障前的一致状态继续执行,避免整个任务的重新执行。该方法可以解决操作系统崩溃、应用程序出错等软件方面的问题,缺点是MPI进程只能在原节点上卷回,如果原节点因硬件故障不可再使用时,该进程就被丢失,整个任务也将崩溃。因此,为了对节点的硬件故障进行容错,必须将故障节点上的MPI进程迁移到其它正常节点上运行,这是LAM/Migration所解决的主要问题。

2 研究现状

目前国外已开发出若干具有容错功能的MPI算平台,如应用CoCheck检查点软件实现容错的tuMPI[2],Starfish[3],MPICH-V[4]等,但都没有解决MPI进程迁移这一难题。Jiannong Cao[5]提出的MPI进程迁移方案是通过修改检查点文件中的数据实现的,当检查点文件的内容或数量较大时,将产生巨大开销,且灵活性差。Chao Wang[6]提出的Job Pause Service机制由于不能迁移任务主进程mpirun,因此不能真正实现整体任务迁移,当mpirun进程所在节点发生故障时,仍无法避免整体任务的崩溃。

本文基于MPI-1规范,以扩展MPI运行环境的容错能力为目标,选用Linux 2.6与LAM/MPI-7.3作为开发平台,突破了MPI进程迁移这一关键技术,成功开发出LAM/Migration原型系统,实现MPI进程在任意节点之间的自由迁移,有效提高了系统可用性。其特点如下:

①MPI容错功能扩展:通过将检查点软件BLCR(Berkeley Lab's Checkpoint/Restart)与LAM/MPI相结合,实现MPI整体任务状态的保存与任务的卷回恢复。

②进程迁移:MPI整体任务,包括任务主进程mpirun与任意MPI作业进程均可在集群的任意节点之间迁移,实现对集群中节点故障的容错。

③透明的容错功能:对应用程序透明,使程序员在设计、编写程序的过程中无需考虑任何与容错相关的问题。

④对硬件故障的容错:LAM/Migration突破了原MPI平台只能在本地进行检查点保存与恢复的束缚,通过将MPI进程迁移到其它节点,可有效对节点硬件的故障进行容错。

到稿日期:2008-10-10 本文受"十一五"国防预研项目(513160201)资助。
刘天田(1978-),男,硕士,研究方向为分布式计算与容错技术,E-mail:tiantianl@gmial.com。

3 LAM/Migration 整体设计

将进程从某节点转移至另一节点执行的行为称为进程迁移。LAM/Migration 将进程的检查点文件传输到其它节点进行异地卷回来实现进程迁移，其整体设计思想体现在 LAM 层和 MPI 层，如图 1 所示。

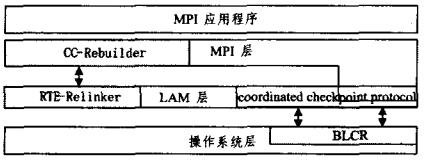

图 1 LAM/Migration 软件体系结构

1. 在 LAM 层添加 LAM 环境重链接模块(Run-Time-Environment Relinker，RTE-Relinker)：使 MPI 进程在异地卷回恢复过程中，自动获取新节点的 LAM 环境信息并与其进行重链接；

2. 在 MPI 层添加通讯通道重建模块(Communication Channel Rebuilder，CC-Rebuilder)：MPI 进程迁移后，更新 MPI 进程间的通讯路由，重建进程间的消息通道。

LAM 环境重链接模块与通讯通道重建模块是实现进程迁移的两大支撑，具体实现将在第 4 节阐述。LAM/Migration 的 MPI 层与 LAM 层共同实现同步检查点算法(coordinated checkpoint protocol)，并配合内核检查点模块 BLCR 实现 MPI 任务检查点的保存工作。因为 CC-Rebuilder 与 RTE-Relinker 分别在 MPI 层和 LAM 层实现，所以 LAM/Migration 的进程迁移功能对应用程序完全透明，而且 MPI 进程在 CC-Rebuilder 与 RTE-Relinker 的协助下，可自主完成 LAM 环境重链接、通讯路由更新、消息通道重建，无需任何外界或人工干预，充分体现出 LAM/Migration 的智能化。

4 进程迁移的实现

LAM/Migration 进程迁移的实现分为检查点保存与异地卷回两个部分，其中 LAM 环境重链接与通讯通道重建模块发挥了关键作用。

4.1 MPI 任务的检查点机制

LAM/Migration 应用同步检查点协议来保证记录 MPI 任务正常运行期间的一致状态。检查点文件保存在集群可靠共享存储设备 NAS 上。检查点操作入口是 mpirun 进程，具体流程如下(如图 2 所示)。

1. mpirun 接收到用户或系统管理程序的检查点命令之后，向所有节点的 MPI 服务进程 lamd 广播检查点请求信号；

2. 各节点上的 lamd 将该检查点信号转发给本地 MPI 进程；

3. 为了记录 MPI 任务一致状态，各 MPI 进程在自身检查点产生之前必须相互协调地做一些准备工作，包括清空 MPI 进程之间的在途消息(on-fly -message)，释放共享内存、文件、信号量、套接字等 BLCR 不能保存的资源。在所有 MPI 进程完成准备工作之后生成检查点；

4. 在所有 MPI 进程成功保存检查点之后，mpirun 再保存自身检查点。

图 2 展示了一个由 mpirun，Foo1，Foo2 组成的 MPI 任务，分别运行在 Node1，Node2，Node3 上。检查点成功保存后，任务继续运行。

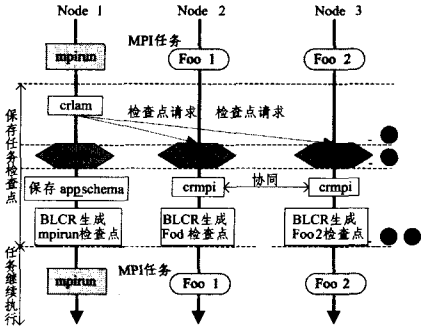

图 2 MPI 任务检查点操作流程

4.2 创建任务拓扑

LAM/MPI 在检查点操作过程中，mpirun 将任务拓扑保存在 app_schema 中，其反映了各 MPI 进程与所在节点的映射关系。进程迁移改变了原始任务拓扑，所以 LAM/Migration 在进程迁移操作中，mpirun 将根据迁移状况创建新的任务拓扑 mig_schema 来对应新的映射关系。

4.3 LAM 环境重链接模块 RTE-Relinker

MPI 进程在运行之初(被 BLCR 恢复时)，是普通 linux 进程，只有同 LAM 环境链接之后才进化为 MPI 进程。对进程迁移而言，检查点文件中所记录的链接信息已无意义。RTE-Relinker 根据新节点的位置信息，协助迁移进程与 LAM 环境的重链接。

4.4 消息通道重建模块 CC-Rebuilder

MPI 进程消息通道的重建需要_gps 结构的支持，该结构包含描述 MPI 进程的地址信息，具体成员如图 3 所示。每个 MPI 进程都保存一个_gps 队列，通过该队列可同任意 MPI 进程建立消息通道进行消息传递。进程迁移使进程的地址信息发生改变，检查点文件中保存的_gps 信息已不可使用。CC-Rebuilder 迁移进程完成 LAM 环境重链接之后，将该进程的新_gps 结构广播给其它 MPI 进程，并对各进程的_gps 队列进行更新，如 gps_node 成员更新为新节点的节点号。随后各进程将各自全新的_gps 结构发送给 mpirun 进程，由 mpirun 组成全新_gps 队列回送给各 MPI 进程，这样全体 MPI 进程的_gps 结构队列都得到了更新。

```
struct_gps{
    int4 gps_node;     //MPI 进程所在节点编号
    int4 gps_pid;      //MPI 进程的 PID 号
    int4 gps_idx;      //对应本地 lamd 保存的索引号
    int4 gps_granks;   //在整个通讯环境中的 rank 号
};
```

图 3 进程地址信息_gps 结构

4.5 mpirun 进程的迁移

同 MPI 进程相似，在 mpirun 进程迁移时，同样必须同新节点上的 lamd 重链接并重建消息通道，这需要 RTE-Relinker 和 CC-Rebuilder 共同支持。值得注意的是，各 MPI 进程在将各自_gps 结构发送给 mpirun 之前，必须感知 mpirun 迁移的新地址，即各 MPI 进程必须更新与 mpirun 之间的路由。而在 mpirun 启动中，只有 lamd 服务知道其地址信息，LAM/Migration 应用 lamd 与 MPI 进程间的通讯将 mpirun 的地址

信息传送给 MPI 进程，实现路由更新。

图 4 展示了较极端的迁移情况。MPI 任务由 mpirun 进程与 2 个 MPI 进程 Foo1，Foo2 组成，分别运行在 Node 1，Node2，Node3 上，并已保存了的检查点文件。假设 Node 1 发生故障，将 mpirun，Foo1，Foo2 分别迁移至 Node2，Node3，Node4 上，具体步骤如下：

1. mpirun 由 BLCR 恢复之后与同 Node2 上的 lamd 重链接；

2. 根据用户或管理程序的要求来创建全新 MPI 任务拓扑 mig_schema；

3. mpirun 根据 mig_schema 中的 MPI 任务拓扑向各节点上的 lamd 服务发出恢复请求；

4. lamd 接收到恢复请求后，从本地或其它节点获取检查点文件之后，调用 BLCR 命令 cr_restart 恢复进程，并与 lamd 重链接；

5. Lamd 服务以进程间通讯方式将 mpirun 新的_gps 结构传送给 MPI 进程，MPI 进程应用 mpirun 新的_gps 结构来更新与 mpirun 的路由，随后 MPI 进程更新各自的_gps 结构；

6. mpirun 将所接收的所有_gps 结构组成全新队列回送给各 MPI 进程；

7. MPI 进程应用全新的_gps 结构队列重建原 MPI 进程间的通讯通道。

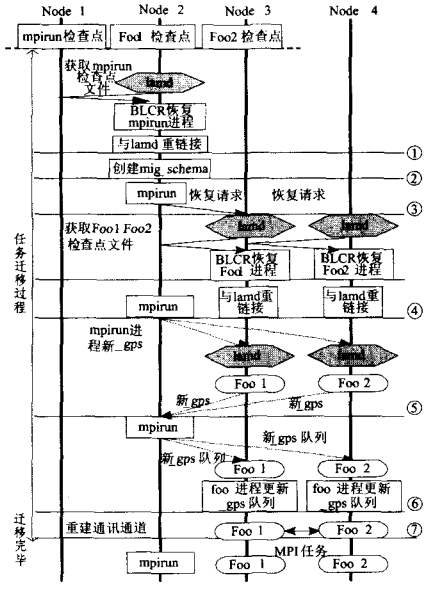

图 4　MPI 进程迁移流程

5　测试

本文针对 LAM/Migration 迁移功能，在由 10 个节点组成的小型集群上进行了测试。各节点配备 P4 处理器（频率为 3GHZ）、1G 内存、千兆网卡，集群内部采用千兆交换机，NAS 作为整个集群的可靠存储设备。各节点安装 Fedora2 操作系统，采用 HPL benchmark 进行测试。

由于 LAM/Migration 检查点同原始的 LAM 完全一样，因此检查点的开销情况同 LAM 基本一致。图 5 所示的实验结果表明，对 MPI 任务生成检查点时间的开销同进程的个数近似线性关系，开销主要来自同步算法以及进程所占用虚拟空间的大小，BLCR 官方报告[7]表明，BLCR 检查点时间同进程虚拟空间大小成线性关系，但当进程虚拟空间大于物理内存时，由于页面频繁的换入/换出将使时间成指数增长。

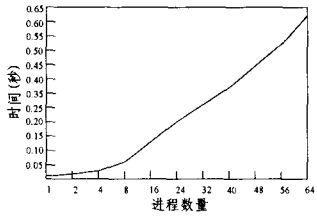

图 5　检查点时间开销

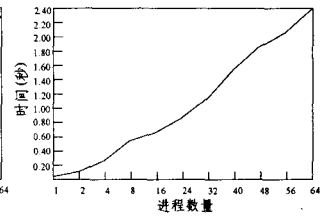

图 6　恢复时间开销

本文对 LAM/Migration 进程迁移与 LAM 本地进程恢复的时间开销比较结果如图 6 所示。LAM/Migration 进程迁移部分重现 LAM 任务启动过程中的初始化工作，其主要时间开销来自对检查点文件的获取，因为节点会根据迁移请求从 NAS 上获取检查点文件。特别在多个进程同时发生迁移时，多节点同时从 NAS 获取检查点势必造成一定的网络拥塞。解决办法可以在 MPI 任务运行期间，由各节点的管理程序在后台将 NAS 上的全体检查点文件复制到本地，可以完全避免在进程迁移过程中检查点文件的传输开销，但将带来节点的本地存储开销。

结束语　LAM/Migration 的核心功能是进程迁移，有效提高了系统的可用性。由于检查点协议与进程迁移机制都在 LAM 与 MPI 层实现，进程迁移对应用程序是完全透明的。通过检查点文件就可根据用户需求在集群任何指定的节点间实现进程迁移，系统可操控性强、智能化程度高。从实验结果来看，在能提供充足的节点本地磁盘空间的前提下，LAM/Migration 进程迁移的时间开销同 LAM 的恢复开销相近。未来的工作将重点研究基于 MPI-2 规范的进程迁移机制。

参 考 文 献

[1] Towards cluster survivability. Chokchai Leangsuksun (1), Anand Tikotekar(1), Stephan L. Scott(2), Makan Pourzandi(3), and Ibrahim Haddad(4). Louisiana Tech University(1), Oak Ridge National Laboratory(2), Open Systems Lab, Ericsson Research Canada(3), Open Source Development Labs(4)

[2] Stellner G. CoCheck: Checkpointing and Process Migration for MPI// Proceedings of the 10th International Parallel Processing Symposium. 1996: 526-531

[3] Agbaria A M, Friedman R. Starfish: Fault-Tolerant Dynamic MPI Programs on Clusters of Workstations// Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing. 1999, 31

[4] MPICH-V Introduction. http://www.lri.fr/~gk/MPICH-V

[5] Process Migration for MPI Applications based on Coordinated Checkpoint. Jiannong Cao(1), Yinghao Li(1), Minyi Guo(2). Department of Computing The Hong Kong Polytechnic University Kowloon, Hung Hom Hong Kong, China PR(1), Department of Computer Software The University of Aizu Aizu-Wakamatsu City, Fukushima 965-8580, Japan(2)

[6] A Job Pause Service under LAM/MPI+BLCR for Transparent Fault Tolerance. Chao Wang (1), Frank Mueller(1), Christian Engelmann(2), Stephen L. Scott(2). North Carolina State University (1), Oak Ridge National Laboratory (2)

[7] Elnozahy E N, Johnson D B, Zwaenepoel W. The performance of consistent checkpointing. //Proceedings of the 11th Symposium on Reliable Distributed Systems. Oct. 1992: 39-47