

可靠的分布式系统生存性保障模型

耿 技, 陈 非*, 聂 鹏, 陈 伟, 秦志光

(电子科技大学, 计算机科学与工程学院, 成都 610054)

(*通信作者电子邮箱 flikecn@126.com)

摘 要:基于检查点的协同式回滚恢复机制是一种针对分布式系统生存性保障的有效机制, 现有分布式系统中基于检查点的回滚恢复机制以分布式信道可靠作为假设前提, 而实际应用场景中, 该假设并不总是成立。针对分布式系统实际的应用环境, 提出了适用于信道不可靠的分布式计算环境的协同式系统生存性保障模型。该模型在保留检查点回滚恢复机制优点的基础上, 通过建立冗余通信链路和进程迁移来保障不可靠通信信道环境下分布式系统的生存性。

关键词:检查点; 分布式系统; 软件生存; 回滚恢复; 进程迁移

中图分类号: TP338 **文献标志码:** A

Reliable assurance model for distributed system survivability

GENG Ji, CHEN Fei*, NIE Peng, CHEN Wei, QIN Zhi-guang

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu Sichuan 610054, China)

Abstract: The cooperative rollback recovery mechanism based on checkpointing is an effective mechanism for the survivability of distributed system. The existing cooperative rollback recovery mechanism based on checkpointing presumes that the communication channel is reliable. However, this assumption is not always true in actual application scenarios. For the actual application scenarios of distributed system, a reliable assurance model for the survivability of distributed system was proposed, based on the checkpointing-based rollback recovery mechanism. Through the creation of redundant communication channel and process migration mechanism, the proposed model assures the survivability of distributed system in actual application scenarios where the communication channel is not reliable.

Key words: checkpoint; distributed system; software survivability; rollback recovery; process migration

0 引言

航空航天、天气预报、石油勘探等领域, 对软件的计算能力提出了很高的要求。大规模分布式计算能很好地满足上述领域对计算能力的需求。这些分布式系统软件通常需要进行长时间的运行, 这导致了单个计算节点中的工作进程崩溃的可能性增加。因此检查点设置及恢复机制被提出, 并作为一种分布式计算软件系统生存保障机制。

通过周期性采集进程运行状态, 并将进程状态保存为检查点信息, 当进程崩溃时, 只需利用该进程最新的检查点恢复这个进程, 而无需重新运行该进程^[1-4]。该类方案减少了系统出错带来的损失, 节省了系统恢复时间。并行分布式系统中需要获得各个子系统检查点状态, 以获得整个系统的“全局一致性状态”^[5-8], 这样才能保证系统中进程的正确恢复执行。在并行分布式系统中检查点设置算法一般可以分为协同式检查点和非协同式检查点算法^[9-11]。非协同式检查点算法具有较好的独立性, 各个进程独立设置检查点信息, 而无需进程间进行通信协调。但该算法也存在明显缺点, 即非协同式检查点算法受到多米诺效应的影响, 从而导致难以找到可用的检查点进而带来较大的计算损失^[12]。协同式检查点算

法则可以避免多米诺效应, 很好地控制计算损失。进程在相同时间间隔内统一设置检查点信息, 当系统中某个进程崩溃时, 可以选择该进程最近的检查点进行恢复^[9, 13]。协同式检查点算法回滚距离短、实现简单、无需进程执行分段确定性假设, 并具备容忍多个进程故障的优点^[14-15]。本文讨论的模型——可靠的分布式系统生存性保障模型 (Reliable Assurance Model for the Distributed System, RAMS) 属于协同式检查点设置算法, 中心进程通过可靠通信协议向其他子系统进程发送控制指令, 对各个子系统中检查点进行同步和设置, 并接收子系统进程的检查点信息; 当子系统或子系统间通信发生异常时, RAMS 模型可自适应地对系统进行恢复, 确保系统的生存性。

RAMS 重点讨论了进程通信的信道异常和子系统服务异常问题。传统模型假设通信信道完全可靠, 但通信信道在实际系统运行中是存在一定故障率的; 确保分布式软件系统在非可靠信道和子系统服务异常时的整体生存性是 RAMS 的研究重点。

1 可靠的分布式软件系统生存性保障模型

在协同式检查点设置算法中, 为了保证检查点的全局一

收稿日期: 2012-04-18; 修回日期: 2012-05-30。

基金项目: 国家自然科学基金资助项目 (60973118); 中央高校基本科研业务项目 (ZYGX2011J072)。

作者简介: 耿技 (1963-), 男, 安徽合肥人, 教授, 博士研究生, 主要研究方向: 系统软件、软件确保、信息安全; 陈非 (1988-), 男, 江西樟树人, 硕士研究生, 主要研究方向: 软件确保、信息安全; 聂鹏 (1977-), 男, 陕西汉中, 博士研究生, 主要研究方向: 软件确保、软件测试软件可靠性; 陈伟 (1978-), 男, 四川温江人, 讲师, 博士研究生, 主要研究方向: 无线网络路由、网络安全; 秦志光 (1965-), 男, 四川隆昌人, 教授, 博士生导师, 主要研究方向: 开放系统、中间件、信息安全。

致性,中心进程 P_c 需要定期获取整个系统中各个子进程的 checkpoints 信息。中心进程 P_c 保存子系统进程 P_i 的 checkpoints 信息,当进程 P_i 崩溃时, RAMS 无需重新运行进程 P_i ,而仅使用保存的 checkpoints 信息对进程 P_i 进行恢复,可节省系统计算资源。该方法的优点是相关进程的恢复协议比较简单;但由于每次 checkpoints 设置过程需要和所有的进程 P_i 进行交互,因此协同开销比较大。

在通信信道不可靠的情况下, RAMS 中心进程定时地向各个子系统发起设置 checkpoints 信息的指令,子系统进程收到指令后将其 checkpoints 信息发送给中心进程。其中 checkpoints 信息指的是进程堆栈内容、进程硬件上下文、进程相关的文件描述符表等信息。当子系统进程崩溃时,子系统向中心进程发送请求回滚的消息,中心进程将该进程的最后一次 checkpoints 信息发送给子系统,对子系统进程进行回滚操作。

当通信信道出现故障时, RAMS 尝试在中心进程 P_c 与子系统进程 P_i 之间建立新的通信信道;当新信道建立失败时, RAMS 则将子系统进程迁移到其他子系统服务器中。在子进程崩溃或通信信道故障的情况下,中心进程 P_c 都可以及时发现并修复系统故障,从而提高了分布式系统的生存性。整个 RAMS 的进程间的信息交互如图 1 所示。

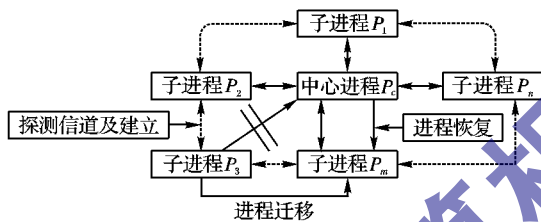


图1 协同式检查点算法进程通信模型

如图1所示,分布式系统中存在 n 个子进程和一个中心进程,子进程 P_i 运行在整个分布式系统的子系统 S_i 中,子进程 P_i 用于获取检测点信息并向中心进程发送 checkpoints 消息,同时也在相应的子系统 S_i 中为用户提供服务。中心进程 P_c 运行在中心系统中,是中心系统中的一个进程。它向各个子系统进程 P_i 发送获取 checkpoints 指令,并负责接收子系统进程 P_i 返回的 checkpoints 消息,中心进程 P_c 负责获取子进程的 checkpoints 信息、修复信道、回滚故障进程等任务并不向用户提供具体的服务。在图1中,实线表示子进程 P_i 与中心进程 P_c 之间通过可靠的通信协议进行数据传输;虚线表示子进程间可能存在通信链路。

当子进程 P_m 崩溃而 P_m 与中心进程 P_c 间存在可用链路时, RAMS 中心进程 P_c 使用最近的历史 checkpoints 信息对 P_m 进行恢复。当子进程 P_3 与中心进程 P_c 之间无可用链路时,中心进程 P_c 则首先寻找并试图建立新的通信信道。若其他子进程都不能和 P_3 建立通信信道,则需要将子进程 P_3 择优迁移到其他子系统中。

2 检查点生存性保障算法

在不可靠通信信道下, RAMS 获取 checkpoints 信息与故障恢复操作可以分为两个主要部分:1) 通信信道正常情况下,中心进程 P_c 获取各个子系统进程 P_i 的 checkpoints 信息。当进程 P_i 出现服务异常时, RAMS 利用 checkpoints 信息对进程 P_i 进行恢复操作,使进程 P_i 能够继续为用户提供服务。2) 当 P_c 和 P_i 间通信信道出现故障时,中心进程 P_c 首先试图建立 P_c 和 P_i 间新的通信信道,在不能建立新信道的情况下, P_c 将 P_i 迁移到

其他子系统中,确保整体系统的生存性。图2为改进后 checkpoints 获取算法流程。

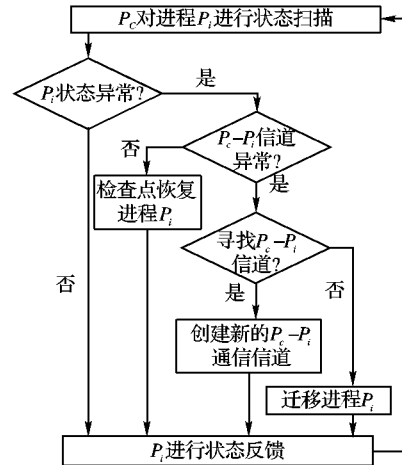


图2 改进后检查点获取算法流程

P_c 定期地对 P_i 进行状态扫描,扫描的目的是为了及时地发现子进程本身出现故障或子进程与中心进程之间的信道出现故障。当 P_i 状态正常时,则整个工作流程结束,并等待下一次工作流程开始。但中心进程扫描到 P_i 的服务异常时,则需要进一步区分是 P_i 本身出现故障还是其与中心进程之间的信道出现故障。如果是 P_i 本身出现故障,则中心进程利用已保存的 checkpoints 信息将 P_i 恢复,并向中心进程反馈恢复成功的信息,整个工作流程至此结束,等待下一次工作流程开始。如果是 P_i 与 P_c 之间的信道出现故障,则中心进程首先寻找并建立它与 P_i 之间的新可用信道,如果操作成功,整个工作流程至此结束,等待下一次工作流程开始。但如果未能寻找新的信道时,中心进程将择优地将 P_i 迁移到其他子系统中,整个工作流程至此结束,等待下一次工作流程开始。RAMS 生存性算法如算法1所示。

算法1 中心进程算法。

```

1)  foreach  $P_i \in P$  do
2)      if (  $P_c \text{ scan } P_i$  ) == statuserror then
3)          if ( the channel disconnect ) then
4)              if ( 寻找信道成功 ) then
5)                   $P_c$  与  $P_i$  之间建立新的信道;
6)              else
7)                  将进程迁移到其他子系统;
8)              end if
9)          else
10)             choose checkpoint to resume process;
11)          end if
12)      end if
13)      report status of  $P_i$  to  $P_c$  ;
14)  end foreach

```

2.1 寻找可用通信信道

当 P_i 与 P_c 无法使用原信道进行通信时, P_c 将负责探测出一条与 P_i 进行通信的新信道。通过 RAMS 可使中心进程 P_c 与子进程 P_i 在新的信道上通信,增强了系统的生存性。如果 RAMS 无法通过其他子进程建立新的通信信道,则 P_i 需要进行进程迁移。通过将进程 P_i 迁移到其他子系统中,使整个分布式系统可提供持续可靠的服务。整个通信信道的探测如图3所示。

中心进程 P_c 与除进程 P_i 之外的子进程协商建立新信道。首先中心进程 P_c 与其他非信道故障子进程建立连接,连接建

立成功之后,中心进程 P_c 广播存在信道故障的子进程 P_i 所在系统 S_i 的信息。各个子进程收到信息后,利用收到的信息与子系统 S_i 建立连接,并将建立连接是否成功的结果发送给中心进程 P_c 。若存在子进程 $P_j (i \neq j)$ 能与 S_i 建立连接,则使用子进程 P_j 所在的子系统 S_j 作为中转子系统,重新构建新的通信信道。若不存在子进程 P_j 可与 S_i 建立连接,则考虑对进程 P_i 进行迁移。

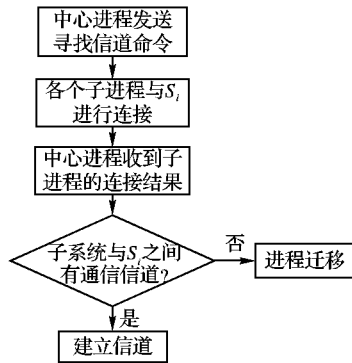


图3 探测通信信道流程

2.2 建立通信信道

如果中心进程 P_c 收到的结果是有多个子进程能与子系统 S_j 进行通信,则建立通信信道时需择优选择一个子系统作为中转子系统,来转发中心进程 P_c 和子进程 P_i 间的通信消息。整个建立通信信道的主要过程如图4所示。

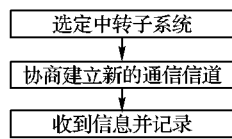


图4 建立通信信道流程

中心进程 P_c 从子进程返回的结果中,根据具体策略择优选择一个子系统作为中转子系统,假定为子系统 S_j ,中心进程 P_c 将消息 $MSG = \langle LOCAL_SYSTEM_IP, PORT, BUILD_CHANNEL_TAG \rangle$ 发送给子进程 P_j , 其中: $LOCAL_SYSTEM_IP$ 表示中心进程所在系统的 IP 地址, $PORT$ 表示中转端口号, $BUILD_CHANNEL_TAG$ 表示建立新信道标志。子进程 P_j 收到消息后,子进程 P_j 对 MSG 消息进行记录并发送确认信息。同时向子系统 S_i 发送 MSG 消息。子系统 S_i 根据标识信息对通信信道建立状态进行识别,并将中心系统的 IP 地址和端口号替换为中转子系统 S_j 的 IP 地址和中转端口。中心进程 P_c 收到中转子系统 S_j 中子进程 P_j 发送的确认信息后,将发送消息时使用的子系统 S_i 的 IP 地址和端口号更改为中转子系统 S_j 的 IP 地址和中转端口号。至此,中心进程 P_c 和子进程 P_i 之间的新通信信道建立成功,它们之间的通信将基于新的通信信道进行交互。

2.3 进程迁移

为了保证整个分布式系统中全局一致性检查点的特性,必须将子系统 S_i 所提供的服务迁移到其他子系统中^[12, 14]。进程迁移具体步骤如下,进程迁移流程如图5所示。

首先,中心进程 P_c 根据接收的子系统消息,从中择优选择一个子系统 S_k 作为迁移目标子系统。进程 P_c 将子进程 P_i 最近一次检查点信息、迁移端口号和特定的标识信息 $MIGRATION_TAG$ 发送给子系统 S_k ,其中 $MIGRATION_TAG$ 是进程迁移消息标识。然后子系统 S_k 根据检查点信息恢复子进程 P_i 。

最后,中心进程 P_c 将发送消息时使用的子系统 S_i 的 IP 地址和端口号更改成子系统 S_k 的 IP 地址和迁移端口号。子系统通过迁移端口将迁移服务的检查点信息发送给中心进程。进程迁移的主要流程如图5所示。

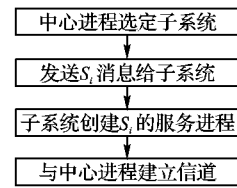


图5 进程迁移流程

2.4 模型分析评估

为了说明本文提出的 RAMS 生存性模型的有效性,本节对分别采用 RAMS 模型和现有的没有考虑信道不可靠性的生存性模型的分布式系统的可生存性进行了数学推理对比分析。

如图1所示,假设一个分布式系统由 $m (m > 1)$ 个子系统(进程)构成,各个子进程与中心进程间的通信信道可靠性概率为 $n (0 < n < 1)$ 。

在通信信道正常的情况下,当一个或多个子进程出现故障时,中心进程可根据之前保存的相应的子进程检查点信息恢复故障子进程,甚至将不可恢复的子进程迁移到其他子系统。

本文将分布式系统的生存性定义为一个或多个子系统出现故障时被恢复或迁移的概率。

现有的分布式系统生存性模型没有考虑子进程与中心进程通信信道的不可靠性。因此在通信信道出现故障时,中心进程无法掌握子进程的状态,因而不能确定故障子进程并对其进行恢复,从而在通信信道不可靠时不能保证分布式系统的生存性。所以基于现有生存性模型的分布式系统的生存性 P_1 等于分布式系统的全部子进程(m 个)与中心进程通信信道可靠的概率,即 $P_1 = n^m$ 。然而在本文提出的 RAMS 模型中,在子进程与中心进程通信信道故障时,由于可以建立新的通信信道,并在子进程不能通过检查点信息恢复时对其进行迁移。因而,只有在全部的子进程与中心进程都不能正常通信的情况下,才能保证整个分布式系统的生存性。所以,采用 RAMS 模型的分布式系统的生存性为 $P_2 = 1 - (1 - n)^m$ 。

为了对比两个模型的生存性,计算:

$$P_2 - P_1 = 1 - (1 - n)^m - n^m$$

通过计算可得:由不等式: $a^k + b^k < (a + b)^k (a, b > 0, 1 < k)$ 可知: $(1 - n)^m + n^m < (1 - n + n)^m = 1$, 由此可知: $P_2 > P_1$ 。

从上述数学推理过程可知,由于 RAMS 模型考虑了子进程与中心进程通信信道的不可靠性,在通信信道不可靠时通过新建通信信道以及故障子进程迁移,从而提高了整个分布式系统的生存性。

3 结语

针对检查点设置与获取过程中通信信道不可靠的情况,提出了一种可提高分布式系统生存性的模型 RAMS。该模型基于中心进程与子系统进程之间的通信信道不可靠的假设,采用检查点恢复与重建故障通信信道的机制,确保分布式系统在工作环境异常的情况下,可通过检查点机制和进程迁移

方式自适应地对系统进行恢复,实现整个系统在不可靠通信信道下生存性的提高。

参考文献:

- [1] GUPTA B, RAHIMI S, YANG Y. A novel roll-back mechanism for performance enhancement of asynchronous checkpointing and recovery[J]. *Informatica*, 2007, 3(11): 1-13.
- [2] ELNOZAHY E N, ALVISI L, WANG Y M, *et al.* A survey of roll-back-recovery protocols in message-passing systems[J]. *ACM Computing Surveys*, 2002, 34(3): 375-408.
- [3] WANG Y M, CHUNG P Y, LIN I J, *et al.* Checkpoint space reclamation for uncoordinated checkpointing in message-passing systems[J]. *IEEE Transactions on Parallel and Distributed Systems*, 1995, 6(5): 546-554.
- [4] RUSCIO J F, HEFFNER M A, VARADARJAN S. DejaVu: Transparent user-level checkpointing, migration, and recovery for distributed systems[C]// SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. New York: ACM, 2006: 158.
- [5] MALONEY A, GOSCINSKI A. A survey and review of the current state of rollback-recovery for cluster systems[J]. *Concurrency and Computation: Practice and Experience*, 2009, 21(12): 1632-1666.
- [6] TRIPATHY M, TRIPATHY C R. A new coordinated checkpointing and rollback recovery scheme for distributed shared memory clusters[J]. *International Journal of Distributed and Parallel Systems*, 2011, 2(1): 49-58.
- [7] PRIYA S B, RAVICHANDRAN T. Fault tolerance and recovery for grid application reliability using check pointing mechanism[J]. *International Journal of Computer Applications*, 2011, 26(5): 32-37.
- [8] BOUTELLER A, HERAULT T, BOSILCA G, *et al.* Correlated set coordination in fault tolerant message logging protocols[C]// Euro-Par'11: Proceedings of the 17th International Conference on Parallel Processing. Berlin: Springer-Verlag, 2011: 51-64.
- [9] CHANDY K M, LAMPORT L. Distributed snapshots: Determining global states of distributed systems[J]. *ACM Transactions on Computer Systems*, 1985, 3(1): 63-75.
- [10] ELNOZAHY E N, JOHNSON D B, ZWAENEPOEL W. The performance of consistent checkpointing[C]// Proceedings of the 11th Symposium on Reliable Distributed Systems. [S. l.]: IEEE, 1992: 39-47.
- [11] 魏晓辉,鞠九滨. 分布式系统中的检查点算法[J]. *计算机学报*, 1998, 21(4): 367-375.
- [12] 慈铁为,张展,左德承,等. 可扩展的多周期检查点设置[J]. *软件学报*, 2010, 21(2): 218-230.
- [13] RANA M, PANGHAL A, PANGHAL S. Checkpointing based roll-back recovery in distributed systems[J]. *Journal of Current Computer Science and Technology*, 2011, 1(6): 45-49.
- [14] 汪东升,沈美明,郑伟民,等. 一种基于检查点的卷回恢复与进程迁移系统[J]. *软件学报*, 1999, 10(1): 68-73.
- [15] 汪东升,邵明琰. 具有 $O(n)$ 消息复杂度的协调检查点设置算法[J]. *软件学报*, 2003, 14(1): 43-48.

(上接第2747页)

3 结语

本文针对二维标量双曲守恒律方程的数值求解,构造出CWENO型-熵相容数值求解格式,通过对两个数值算例的分析与讨论,所得结论如下:

- 1) 该数值求解格式能准确捕捉激波,对稀疏波计算结果与准确解吻合很好,且能有效避免非物理振荡的产生;
- 2) 该数值求解格式具有强稳定性,CFL条件数可以取为0.6,所得结果依旧可靠。

本文所构造的数值求解格式是针对二维标量双曲型方程设计的,但文中所采用的CWENO格式以及优化TVD Runge-Kutta格式均易于向高维和方程组推广。本文作者正在将该高性能算法向高维双曲型方程及方程组情形推广。

参考文献:

- [1] JIANG GUANG-SHAN, WANG SHUN CHI. Efficient implementation of weighted ENO schemes[J]. *Journal of Computational Physics*, 1996, 126(1): 202-228.
- [2] ZHANG XIANGXIONG, LIU YUANYUAN, SHU CHI-WANG. Maximum-principle-satisfying high order finite volume weighted essentially nonoscillatory schemes for convection-diffusion equations[J]. *SIAM Journal on Scientific Computing*, 2012, 34(2): 627-658.
- [3] LEVY D, PUPPO G, RUSSO G. A third order central WENO scheme for 2D conservation laws[J]. *Applied Numerical Mathematics*, 2000, 33(1/2/3/4): 415-421.
- [4] KURGANOV A, LEVY D. A third-order semi-discrete central scheme for conservation laws and convection diffusion equations[J]. *SIAM Journal on Scientific Computing*, 2000, 22(4): 1461-1488.
- [5] TADMOR E. Numerical viscosity and the entropy condition for conservative difference schemes[J]. *Mathematics of Computation*, 1984, 43(168): 369-381.
- [6] TADMOR E. The numerical viscosity of entropy stable schemes for systems of conservation laws[J]. *Mathematics of Computation*, 1987, 49(179): 91-103.
- [7] LEFLOCH P-G, ROHDE C H. High-order schemes, entropy inequalities and nonclassical shocks[J]. *SIAM Journal on Numerical Analysis*, 2000, 37(6): 2023-2060.
- [8] LUKÁČOVÁ M, TADMOR E. On the entropy stability of Roe-type finite volume methods[EB/OL]. [2012-01-01]. <http://www.docin.com/p-75822789.html>.
- [9] ISMAIL F, ROE P L. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks[J]. *Journal of Computational Physics*, 2009, 228(15): 5410-5436.
- [10] FJORDHOLM U-S, MISHRA S, TADMOR E. Entropy stable ENO scheme[EB/OL]. [2012-02-01]. <ftp://ftp.sam.math.ethz.ch/pub/sam-reports/.../2011-05.pdf>.
- [11] FJORDHOLM U-S, MISHRA S, TADMOR E. Arbitrarily high-order accurate entropy stable essentially non-oscillatory schemes for systems of conservation laws[J]. *SIAM Journal on Numerical Analysis*, 2012, 50(2): 544-573.
- [12] 李红霞. 一维守恒型方程(组)的熵耗散格式[D]. 上海: 上海大学, 2000.
- [13] 陈荣三. 双曲守恒型方程若干数值方法研究[D]. 上海: 上海大学, 2009.
- [14] 罗力,封建湖,唐小娟,等. 求解双曲守恒律方程的高分辨率熵稳定格式[J]. *计算物理*, 2010, 27(5): 671-678.
- [15] LEVY D, PUPPO G, RUSSO G. A fourth-order central WENO scheme for multi-dimensional hyperbolic systems of conservation laws[J]. *SIAM Journal on Scientific Computing*, 2002, 24(2): 480-506.