基于三层混合编程模型的Petri网并行算法研究

周 杰[1] 李文敬[2,3]
(广西师范学院计算机与信息工程学院 南宁 530023)[1]
(广西师范学院科学计算与智能信息处理高校重点实验室 南宁 530023)[2]
(广西师范学院物流管理与工程学院 南宁 530299)[3]

摘 要 为解决多核机群Petri网并行化过程中,运用MPI+OPenMP混合编程实现同步会出现死锁的问题,提出了基于三层混合编程模型的Petri网并行算法。首先,根据事务内存的同步优势,在多核机群环境下构建MPI+OPenMP+STM的三层编程模型;然后,对Petri网的几何模型与代数模型的并行化进行分析,建立MPI+OPenMP+STM三层结构的Petri网并行模型,并对三层混合编程模型的Petri网并行算法进行设计与分析;最后,通过示例进行编程验证,该算法的运行效率明显优于其他编程模式,而且Petri网的规模越大,其并行计算的效果就越明显。因此,该算法是多核机群环境下模拟Petri网并行运行的一种高效且可行的算法。

关键词 Petri网,MPI+OPenMP+STM编程,Petri网并行化,Petri网并行模型,并行算法
中图法分类号 TP301.1 **文献标识码** A

Research on Parallel Algorithm of Petri Net Based on Three-layer Mixed Programming Model

ZHOU Jie[1] LI Wen-jing[2,3]
(College of Computer and Information Engineering, Guangxi Teachers Education University, Nanning 530023, China)[1]
(Key Laboratory of Scientific Computing and Intelligent Information Processing, Guangxi Teachers Education University, Nanning 530023, China)[2]
(School of Logistics Management and Engineering, Guangxi Teachers Education University, Nanning 530299, China)[3]

Abstract In order to solve the deadlocks in the synchronization realized by using the MPI+OPenMP mixed programming during the parallelization of Petri nets based on the multi core cluster, the paper proposed the Petri net parallel algorithm based on a three layer mixed programming model. Firstly, it builds a three layer programming model of MPI+OPenMP+STM in the multi-core cluster environment according to the synchronous advantage of the transactional memory. Then, it analyzes the parallelization of the geometric model and the algebraic model of the Petri net. It also builds the Petri net parallel model with a three-layer structure of MPI+OPenMP+STM as well as designing and analyzing the Petri net parallel algorithm of the three-layer programming model. Finally, the paper validates the programming through examples and proves that the operating efficiency of this algorithm is much better than those of other programming modes. In addition, the larger the size of the Petri net is, the better effect of the parallel computing it has. Therefore, the algorithm is an efficient and applicable algorithm for the simulated parallel operation of the Petri net in the multi-core cluster environment.

Keywords Petri net, MPI+OPenMP+STM programming, Petri net parallelization, Petri net parallel model, Parallel algorithm

Petri网是一种系统的数学、图形建模和分析工具。它特别适用于具有同步、并发、冲突的离散事件系统的建模,并被广泛应用于分布式并行处理、离散事件、柔性制造等复杂系统的设计和分析[1]。Petri网划分并行子网可以帮助找出实际问题可并行的工序,以及影响生产进程的关键生产步骤,从而更好地指导和改进生产过程,提高生产的效率。因此,对基于三层混合编程模型的Petri网并行算法的研究具有重要的科学意义和解决实际问题的价值。

1 相关研究

1.1 MPI+OpenMP+STM三层并行模型

Rajkumar Sharma等人[2]通过使用MPI与混合MPI+

本文受国家自然科学基金(61363037,61363074,61163012),广西自然科学基金(2016GXNSFAA380243),广西自然科学基金重点项目(2014GXNSFDA118037)资助。

周 杰(1992-),男,硕士生,主要研究方向为并行计算、Petri网;李文敬(1964-),男,教授,硕士生导师,主要研究方向为并行计算、Petri网。

OpenMP 编程方法比较多线程粗细粒度计算，并评估多核心集群的适用性；Rolf Rabenseifner 等人将拓扑感知构建到混合 MPI＋OpenMP 应用程序中，可以使混合动力编程更容易[3]；赵永华等[4-5]对 SMP 集群上构建 MPI＋OpenMP 混合编程模型的有效性进行了研究，提高了集群的执行速度；陈辉等[6-7]设计了并行算法，将并行编程模型应用到 FDTD 等实际问题，使多进程多线程混合编程获得了实际的应用。但是，MPI＋OpenMP 并行编程模型主要通过锁机制来实现同步，而同步容易造成死锁。

1.2　Petri 网的并行化

Petri 网的并行化主要采用 P-不变量技术对 Petri 网进行功能划分的方法[1]。李文敬教授详细地介绍了 Petri 网进行功能划分的方法[8]，对高级 Petri 网的并行化算法进行了研究[9]。柯琦等人运用桶分法实现了基于多核机群环境下高效的任务调度和分配策略[10]。本文在相关研究基础上，提出基于 MPI＋OpenMP＋STM 三层混合编程模型的 Petri 网并行算法，对 Petri 网并行化进行分析，划分成多线程共享存储器协同完成任务。

2　事务内存与 MPI＋OPenMP＋STM 模型的构建

2.1　事务内存的优势

事务内存相较锁机制具有 4 种优势[11]：1）锁可能导致死锁，事务则不会死锁；2）锁强制操作的顺序执行，同步线程相互等待，事务内存能够检测出并发读与并发写，允许事务并发执行；3）粗粒度的锁限制并发性，细粒度的锁可能造成死锁，事务对粗细粒度的并发性具有更好的调节功能；4）锁不能很好地复合或嵌套，事务的复合或嵌套能更好地实现。因此，基于事务内存比基于锁机制的多线程并行程序能更好地发挥多核 PC 的潜能，事务内存的并行编程模型可以避免 MPI＋OpenMP 并行模型锁机制存在的死锁、优先级反转和同步造成速度慢的问题[12]。

2.2　MPI＋OPenMP＋STM 模型

在应用多核机群解决实际问题时，首先对问题进行任务分解，每一个任务分配到机群的每一个节点，节点就是一个进程，进程之间的并行与消息传递由 MPI 实现；然后对每个进程的任务再划分成若干个子任务，分配给节点内的多核处理器或多线程来处理，节点内的若干子任务则由 OpenMP 创建的一组线程进行计算；最后计算每个子任务，其共享变量或临界区等同步问题，用 STM 模型替代锁机制，并由节点内的多处理器或多线程并发执行 STM 模块程序[12]。

3　Petri 网的并行化分析

3.1　Petri 网并行化技术

Petri 网的并行化主要是基于 P-不变量技术。P-不变量技术在求解 Petri 网的过程中需要求解的关键问题为求解 Petri 网的关联矩阵，以及 P-不变量的支集。

除此之外，在进行 Petri 网并行进程求解之前需要将非 P/T 网转化为 P/T 网，并对 P/T 网结构进行检测，查看是否有共享库所存在，并等效改变网结构消除共享库所。因此，完整的 Petri 网并行化过程流程图如图 1 所示。

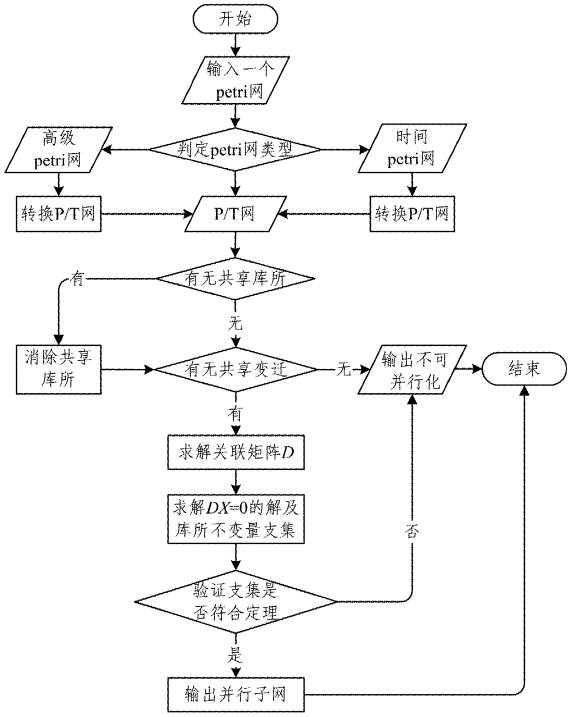

图 1　Petri 网并行化流程图

3.2　Petri 网几何模型的并行划分

划分 Petri 网的目的是为了分析实际问题的并行性，以提高生产效率。因此，对于一个规模较大的 Petri 网，在使用 P-不变量进行划分后，得到的子网依然有一定的规模，对这些划分后的子网可以使用一次并行划分，以得到更精细的划分结果。因此可以先对各子网进行检测，查看网内是否仍存在共享变迁。若存在，则再一次进行划分，若不存在则程序终止。

因此，若要 Petri 网完全划分，上述 Petri 网并行化的流程图如图 2 所示。

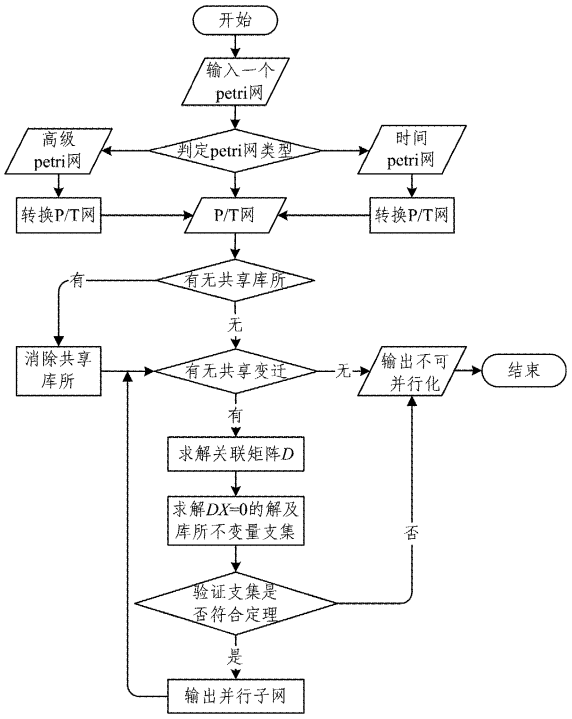

图 2　Petri 网深层并行化流程图

将 Petri 网并行化的各步骤封装成模块，则可得到 Petri 网的并行化图形化模型，如图 3 所示。

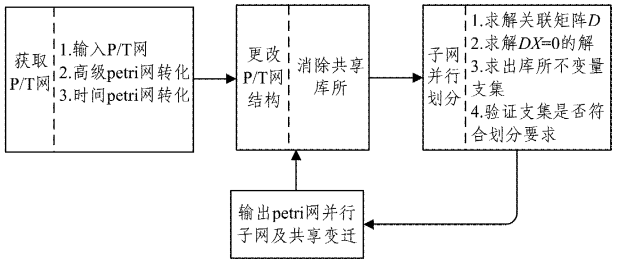

图 3 Petri 网并行化模型

3.3 Petri 网代数模型的并行划分

Petri 网的代数模型主要包括 Petri 网预处理代数模型和 Petri 网求库所不变量的代数模型，以及库所不变量支集的判定代数模型。

Petri 网预处理代数模型如下。

Petri 网的预处理包括非 P/T 网转化为 P/T 网，以及共享库所消除。非 P/T 网转化为 P/T 网包括高级 Petri 网的转化和时间 Petri 网的转化。这里以颜色 Petri 网为例进行高级 Petri 网预处理代数模型的研究。

(1)高级 Petri 网的转化代数模型

P/T 网的代数表示为 $N=(P,T;F,K,W,M)$，颜色 Petri 网的代数表示为 $N=(P,T;F,C,K,W,M)$。对于颜色 Petri 网，有：

$$\begin{cases}\forall p_i\in P,\forall t_j\in T,\forall(p_i\times t_j)\in F,\forall(t_j\times p_i)\in F\\ C=(c_1,c_2,\cdots,c_n)\\ W(p_i\times t_j)=(w_1,w_2,\cdots,w_n)\\ W(p_i\times t_j)=(u_1,u_2,\cdots,u_n)\end{cases}\tag{1}$$

根据颜色 Petri 网转化成 P/T 网的方法，得到转化模型 $(P,T;F,C,K,W,M)\rightarrow(P,T;F,K,W,M)$，其中：

$$\begin{cases}p_i\rightarrow(p_{i_1},p_{i_2},\cdots,p_{i_m},\cdots,p_{i_n})(m\leqslant n,c_m\neq 0\text{ 则 }p_{i_m}\text{存在})\\ M(p_{i_m})=c_m\\ \forall(p_{i_m}\times t_j)\in F(c_m\neq 0,w_m\neq 0)\\ \forall(t_j\times p_{i_m})\in F(c_m\neq 0,u_m\neq 0)\\ W(p_{i_m}\times t_j)=w_m,W(t_j\times p_{i_m})=u_m\end{cases}\tag{2}$$

(2)时间 Petri 网的转化代数模型

时间 Petri 网代数的定义为 $N=(P,T;F,M,I)$，对于时间 Petri 网，有：$\forall p_i\in P,\forall t_j\in T,I(t_j)=\alpha,\forall(t_j\times p_i)\in F$。

根据时间 Petri 网转化为 P/T 网的方法，得到代数转化模型 $(P,T;F,M,I)\rightarrow(P,T;F,K,W,M)$，其中：

$$\begin{cases}t_j\rightarrow(t_{j_1},t_{j_2},\cdots,t_{j_{(\alpha-1)}})(\alpha>1)\\ p_{i_1},\cdots,p_{i_{(\alpha-1)}}\in P\\ \{(t_j\times p_{i_1}),(p_{i_1}\times t_j),\cdots,(p_{i_{(\alpha-1)}}\times t_{j_{(\alpha-1)}}),\\ (t_{j_{(\alpha-1)}}\times p_i)\}\subset F\end{cases}\tag{3}$$

(3)共享库所的消除的代数模型

若一个库所为共享库所，需对其进行消除，依据 3 种共享库所类型的不同情况，有不同的消除方法。

第一种情况：$\forall p_i\in P,\forall t_j\in T,\forall t_i\in T,{}^{\cdot}t_j\cap{}^{\cdot}t_i\in p_i$。则 P_i 为共享库所，消除其代数模型为：

$$\begin{cases}\{p_{i_1},p_{i_2}\}\subset P,t_{j_1}\in T\\ \{(p_i\times t_{j_1}),(t_{j_1}\times p_{i_1}),(t_{j_1}\times p_{i_2}),(p_{i_1}\times t_j),\\ (p_{i_2}\times t_i)\}\subset F\\ (p_i\times t_j)\notin F,(p_i\times t_i)\notin F\end{cases}\tag{4}$$

第二种情况：$\forall p_i\in P,\forall t_j\in T,\forall t_i\in T,t_j^{\cdot}\cap t_i^{\cdot}\in p_i$。则 P_i 为共享库所，消除其代数模型为：

$$\begin{cases}\{p_{i_1},p_{i_2}\}\subset P,t_{j_1}\in T\\ \{(t_{j_1}\times p_i),(p_{i_1}\times t_{j_1}),(p_{i_2}\times t_{j_1}),(t_j\times p_{i_1}),\\ (t_i\times p_{i_2})\}\subset F\\ (t_j\times p_i)\notin F,(t_i\times p_i)\notin F\end{cases}\tag{5}$$

第三种情况为第一种情况和第二种情况的结合。

求 P-不变量的代数模型。

求 P-不变量要求 P/T 网的关联矩阵 D，及 $DX=0$ 的解。因此，其代数模型为：

$$\begin{cases}DX=0\\ D=D^+-D^-;\\ D^+=[d_{ij}^+],D^-=[d_{ij}^-]\end{cases}\tag{6}$$

$$d_{ij}^+=\begin{cases}1, & \text{if }(t^i,p^j)\in T\times P\\ 0, & \text{otherwise}\end{cases}\tag{7}$$

$$d_{ij}^-=\begin{cases}1, & \text{if}(p^j,t^i)\in P\times T\\ 0, & \text{otherwise}\end{cases}\tag{8}$$

(4)库所不变量支集的判定代数模型

库所不变量支集是否符合条件的判定主要依据以下 4 个公式：

1) $\forall p\in\{p_i\in P|x(i)>0\},\forall p\in(t,t')\in p^{\cdot}\cap{}^{\cdot}p$

2) $(w(p,t)=w(t,p)=1)\wedge(\sum_{p\in\|X\|}M(p)\geqslant 1)\vee(\sum_{p\in\|X\|}M(p)=0\wedge(M({}^{\cdot}(p^{\cdot}))\geqslant 1))$

3) $\forall s_1,s_2\in\Gamma_p,\|s_1\|\cap\|s_2\|=\varnothing$

4) $U_{i=1}^{\Gamma_p}(U_{j\in|s_i|}(p_j^{\cdot}\cup{}^{\cdot}p_j))=T$

4 MPI＋OpenMP＋STM 三层结构的 Petri 网并行模型

Petri 网并行化中存在矩阵的计算、网的代数模式的转化等，这些步骤计算量大，串行计算耗费时间长，不利于实际的操作，因此有必要对程序进行并行化。

在 Petri 网的并行化的串行过程中首先进行 Petri 网类型的判定，Petri 网类型判定要考察的因素较多，耗费的资源较大，即使进行并行判定其计算时间也较长，且此阶段的判定只属于 Petri 网并行化的准备阶段。因此，在输入 Petri 网信息时标注 Petri 网的类型，只需进行名称的匹配，这就节省了大量的时间。因此，这一步骤可以采取串行方法。

若判定出 Petri 网的类型不是 P/T 网，则需将其转化为 P/T 网。在对非 P/T 网进行转化时，需要检测每个库所、变迁、弧及库所包含的托肯值或变迁上的时间函数，针对托肯和时间函数进行判断，针对网结构运用 Petri 网模型转化算法决策转化方案，最后实施网结构的改变以得到新的网。假设一个网结构有 m 个库所，n 个变迁和 q 条弧，若采用深度遍历，则一次网检索过程的时间复杂度为 $O((n+m)^2)$，对托肯或时间函数进行判断的时间复杂度为 $O(m)$或 $O(n)$，网结构的改变所需的时间复杂度依据网结构的托肯数或时间函数而

定。因此,在网结构的转化方案的过程中,无论是高级 Petri 网还是时间 Petri 网,花费的时间都较长,有必要考虑将过程并行化,以进行时间优化。

对 Petri 网的转化过程进行如下的并行化。首先,针对复杂 Petri 网的检索过程,可将此任务划分成 3 个进程,每个进程负责不同的任务,需要完成的工作为依次获得 Petri 网的库所集、Petri 网的变迁集和 Petri 网的弧集。之后将获得信息汇总给主进程,主进程再次分配任务给各进程。若是高级 Petri 网,则将检测库所托肯值的任务均分给各进程;若是时间 Petri 网,则将变迁时间函数集均分给各进程,并在检测后改变相应的网结构,执行完毕后将结果汇总。在检测 P/T 网有无共享库所和共享变迁的步骤中,同样可将对 Petri 网的关系集(即弧集的检测)划分成多个并行进程同时进行,且各自根据自己检测出的共享库所的类型进行相应的转化以消除共享库所,将处理后的结果汇总,得到最终的结果。

求解关联矩阵的步骤同样可以并行,将输入矩阵和输出矩阵切割成小模块,并行地进行相加然后组合。Petri 网并行进程求解的关键步骤为求解 $DX=0$ 的解集,即求出库所不变量的支集。在此求解过程中将用到矩阵初等变换齐次线性方程组基础解系迭代解法、正交化列处理法等算法。因此这一过程是 Petri 网并行化过程中计算量最大的阶段。针对此过程实施并行运算,将大大缩短求解的时间。从而求出并行子网需要验证的 4 个公式,每个子网都要逐个验证,此时,可将要验证的子网中的各个子网并行地进行验证,最后汇总结果。在求得子网后,还需重新判断这些子网是否可再分,并进行并行划分操作,此时可使各子网并行执行接下来的操作,这将大大缩短运行的时间。

多核机群的 Petri 网并行化模型如图 4 所示。

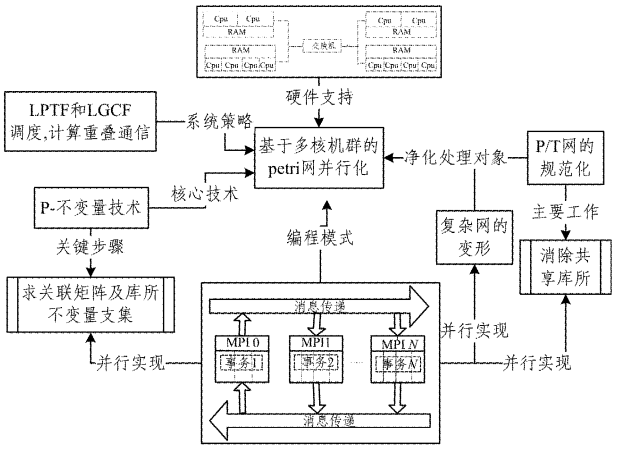

图 4　多核机群的 Petri 网并行化模型

5　MPI＋OPenMP＋MPI 的三层编程模型的 Petri 网并行算法设计

5.1　petri 网并行划分

petri 网并行划分流程图如图 5 所示。

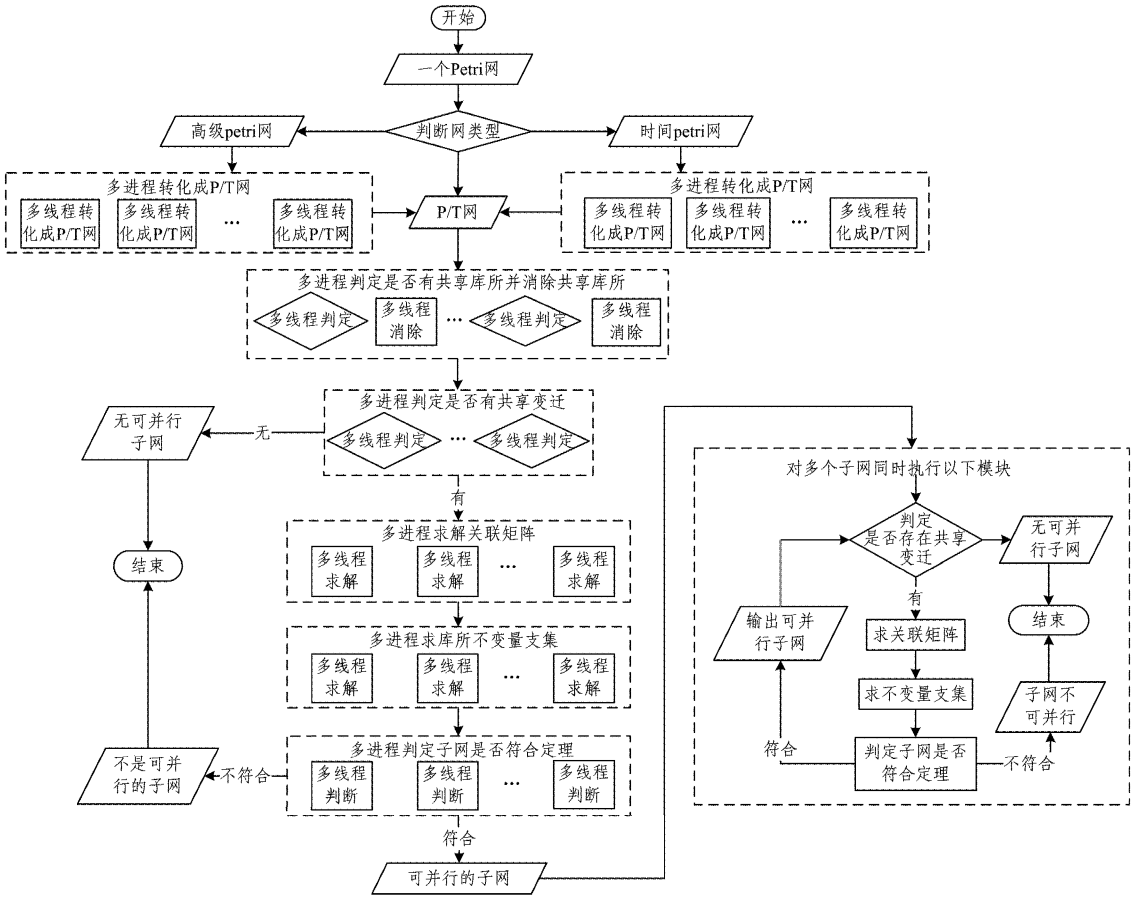

图 5　多核机群的 Petri 网并行划分流程图

5.2　并行算法设计

基于多核集群的 Petri 网并行化算法的基本步骤如下:

Begin:

输入:Petri 网

输出:并行子网

Step1　检测 Petri 网类型。

Step2 根据判断出的 Petri 网类型对输入的 Petri 网进行相应的操作。若是 P/T 网转入下一步，否则将非 P/T 网进行转化操作。操作过程并行进行，将该操作过程中检索 Petri 网获得库所集变迁集及弧集的过程、检索判断托肯集和时间函数集的过程以及修改网结构的过程均并行化。同时将并行的进程内部细分为若干并行的线程。最后将运行结果汇总。

Step3 并行检测 P/T 网的网结构，查找是否存在共享库所，并判断共享库所的类型。根据第 3 节提出的 3 种情况，对 P/T 网结构进行相应的改进。在并行检测是否存在共享库所时可根据 P/T 网结构的弧集来确定，将弧集按序分成若干任务并指派给各进程，各进程内部再将其分成若干线程来并行执行。各并行的进程再检测出共享库所后分别对其进行改进。

Step4 检测改进后的 P/T 网结构，并行查看是否有共享变迁，查看方法类似共享库所的查看方法，并将共享变迁作为关键变迁进行标记。

Step5 并行求出 P/T 网的输出矩阵和输入矩阵，并算出其关联矩阵 D。

Step6 根据 $DX=0$，并行计算出 X 的解集，并得出 X 的支集的集合。

Step7 根据支集中的元素划分子网。

Step8 根据定理 1、定理 2 并行判断划分后的子网是否符合要求。将各子网分配给不同的进程同时进行验证。

Step9 将 Step8 中符合要求的子网输出，即为 Petri 网的并行进程。

Step10 对输出的各并行子网分别进行检测，查看网中是否存在共享变迁。若存在则跳转至 Step5；若不存在则终止程序。该步骤采用并行完成，每个子网的验证都分配给不同的进程，并行进行下去，直至程序结束。

6 实验结果与分析

6.1 实例分析

选取 4 个 Petri 网模型，作为求解并行子网的实例分析。

例 1 图 6 给出了一个 P/T 网，分析该网的结构可知：该网是一个存在共享变迁但不存在共享库所的 Petri 网，若要对该网求并行子网，无需改变其网结构，且该网存在共享变迁，则有存在并行子网的可能性。

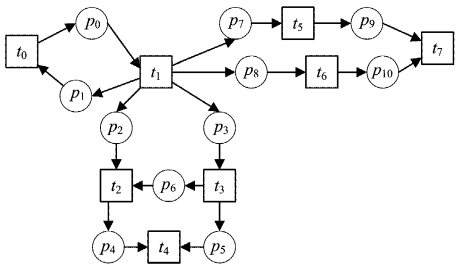

图 6 网∑1

首先，求网∑1 的关联矩阵得到的并行子网为[P_0，P_1]，[P_2，P_3，P_4，P_5]，[P_7，P_8，P_9，P_{10}]。虽然已求得并行子网，但仍需验证子网是否可再分。若检测到 3 个子网中均有共享变迁，则均有再分的可能，子网可再分为[P_7，P_9]和[P_8，P_{10}]两个子网。

例 2 图 7 给出了网∑2 的结构，检测该结构可发现，该网中存在共享库所，因此存在并行子网的可能性，但该网中同时存在共享变迁，且共享变迁的 3 种情况均存在，故需要做消除共享库所的操作。

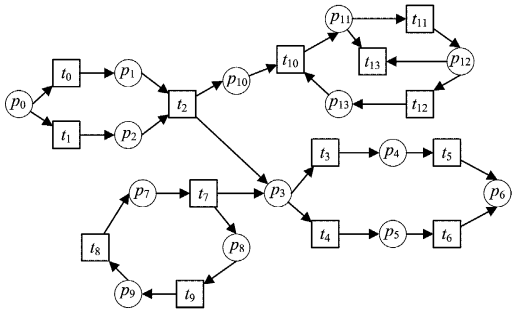

图 7 网∑2

对网∑2 进行消除库所操作，可得到不含共享库所的等效 Petri 网结构，如图 8 所示。

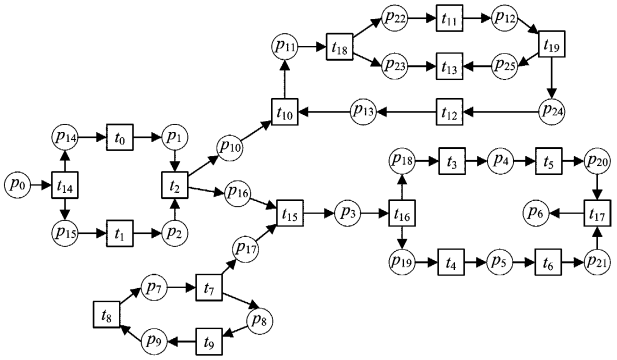

图 8 消除共享库所后的网∑2

如图 8 所示，在原共享库所 P_0 后增加变迁 t_{14}，将 P_0 复制为 P_{14} 和 P_{15} 两份，在共享库所 P_3 前后分别增加变迁 t_{15} 和 t_{16}，并增加 4 个扩所为 P_{16}，P_{17}，P_{18}，P_{19}，在共享库所 P_6 前增加变迁 t_{17}，增加库所 P_{20} 和 P_{21}，在共享库所 P_{11} 后增加变迁 t_{18}，增加库所 P_{22} 和 P_{23}，在共享库所 P_{12} 后增加变迁 t_{19} 和库所 P_{24} 和 P_{25}。此时，网∑2 的关联矩阵得出可并行的子网有 6 个，分别为：[P_2，P_{15}]和[P_1，P_{14}]可行，[P_4，P_{18}，P_{20}]，[P_5，P_{19}，P_{21}]，[P_7，P_8，P_9]，[P_{11}，P_{12}，P_{13}，P_{22}，P_{24}，P_{25}]可并行，观察子网内部无可再分的共享变迁。因此无需进一步划分。

例 3 图 9 给出了网∑3 的结构，网∑3 为颜色 Petri 网，且颜色种类为 3 类，对其并行化时需将其转化为 P/T 网。

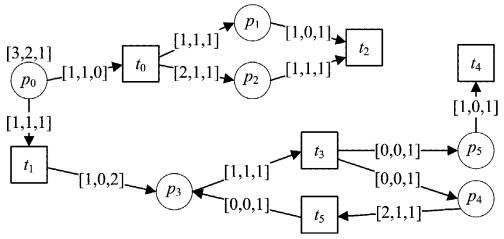

图 9 颜色 Petri 网∑3

图 10 给出了网∑3 转化成的 P/T 网。

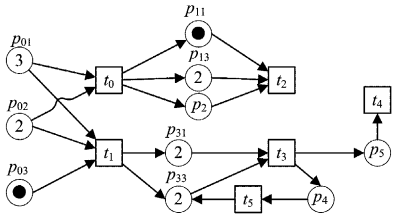

图 10 转化后的∑3

图 10 将颜色 Petri 网中的 P_0，P_1，P_3 按照托肯拓展为库所 3 和库所 2，其分别为 P_{01}，P_{02}，P_{03}，P_{11}，P_{13}，P_{31}，P_{33}，P_2，P_4，P_5，因为托肯值为零，所以保持不变，增加后的库所根据边上的权值连接，权值为 0 的则不连接。

检查转化后的网∑3，此网中存在共享变迁，存在并行子网的可能性，但同时也存在并行库所，需要消除。消除共享库所后的∑3 的网结构如图 11 所示。

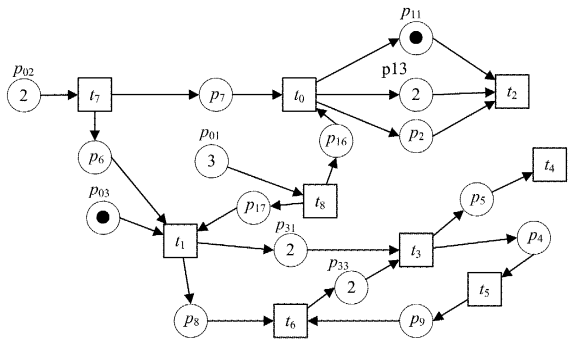

图 11　消除共享库所后的网∑3

此时，网∑3 的关联矩阵可并行的子网为[P_{02}，P_7，P_6]和[P_{01}，P_{17}，P_{16}]，[P_1，P_{13}，P_2]和[P_{31}，P_4，P_5，P_8，P_9]。

例 4　如图 12 为一个时间 Petri 网∑4。网∑4 做并行处理，需将其转化成 P/T 网。

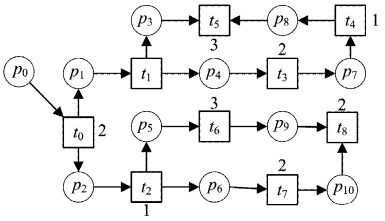

图 12　一个时间 Petri 网∑4

转化后的网∑4 如图 13 所示。

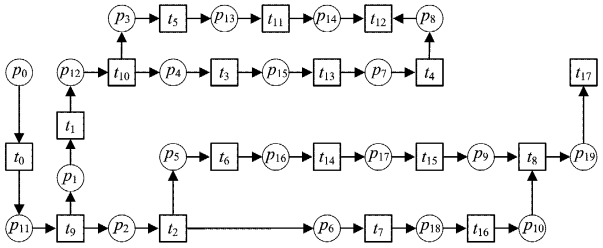

图 13　P/T 转化后的∑4

如图 13 可知，转化后的∑4 依据原时间 Petri 网各变迁的时间函数增加了相应的变迁和库所。检测网中无共享库所可直接求解。此时，∑4 的关联矩阵可并行的子网为[P_1，P_{12}，P_3，P_{13}，P_{14}，P_4，P_{15}，P_7，P_8]，[P_2，P_5，P_{16}，P_{17}，P_6，P_{18}，P_{10}，P_9，P_{19}]。检查这两个子网，运用 P-不变量划分技术还可分为[P_3，P_{13}，P_{14}]，[P_4，P_{15}，P_7，P_8]，[P_5，P_{16}，P_{17}，P_9]，[P_6，P_{18}，P_{10}]。

6.2　实验结果分析

应用本文算法对以上 4 个实例进行实验，设置 3 组对比实验分别为：串行、MPI、OPenMP。实验硬件环境为几台主机通过网络相连。本实验选用 4 台同构的 PC 机，PC 机的配置如表 1 所列。网络选 100Mb/s 的以太网作为本次实验的网络条件。本次实验选用的软件环境为 windows7 操作系统，在 visual studio2008 编程环境下采用 C 语言编程。

表 1　PC 机的各项参数

主要部件	参数说明
CPU	Intel (R) Xeon (R) E3-1225 V2 3.2GHz 三级缓存
内存	16GB
硬盘	1T 7200r/min
网卡	Intel (R)82579LM

对 4 个实验对象进行对比实验，实验结果如表 2 所列。

表 2　实验对比结果

实验对象 / 编辑模型	准确率/%				实验所用时间/s				加速比				平均加速比
	∑1	∑2	∑3	∑4	∑1	∑2	∑3	∑4	∑1	∑2	∑3	∑4	
串行	100	100	100	100	6.14	7.86	7.83	7.77					
MPI	100	100	100	100	4.23	5.59	5.73	5.65	1.45	1.41	1.37	1.38	1.40
OPenMP	100	100	100	100	4.18	5.36	5.28	5.14	1.47	1.47	1.48	1.51	1.48
MPI+ OPenMP+STM	100	100	100	100	2.11	2.98	3.19	3.07	2.91	2.64	2.45	2.53	2.63

4 个网结构在 4 组对比实验下的结果有相似处，也有各自的特点。从实验的运行时间来看，4 个网运行的时间均是 MPI+OPenMP+STM<OPenMP< MPI<串行。首先串行所用时间最多，源于 Petri 网并行化处理的步骤多，且有关矩阵的运算多，运算量较大，因此耗费时间较多。MPI 和 OPenMP 均属于单层并行，一个是基于消息传递的，一个是基于共享存储的，从实验数据来看，共享存储优于消息传递，原因在于共享存储节省了通信的时间，但共享存储也存在争用资源和同步的问题，若处理不好也会出现盲等待，从而耗费时间。本文所用的三层编程模型是用时最短的，且优化程度远高于 MPI 和 OPenMP，其原因在于三层编程模型集合了二者的优点，并将计算任务充分地并行，使程序的并行化率大大提高，且三层模型加上事务内存机制，有效地解决了同步问题和资源争用的问题，使程序的运行效率大大提高。

对比 4 个网的加速比可知，MPI 模式下加速比的大小关系为：∑1>∑2>∑4>∑3，网 ∑3 的加速比最小的原因在于其进行 Petri 网的并行化时经过的步骤繁多，在 MPI 模式下频繁的通信会造成通信时间增多，进程间的负载不平衡以及主进程的空闲等待和从进程的盲等待等问题累计，这些都会影响程序的执行效率，造成时间消耗大，因此对需要频繁通信的网 ∑3 在 MPI 模式下不能较好地发挥并行作用。OPenMP 模式下，4 个网的加速比的大小关系为：∑1＝∑2＜∑3＜∑4。此时网 ∑3、网 ∑4 的加速比最大，其原因在于 OpenMP 是基于共享存储模式的，因此颜色 Petri 网和时间 Petri 网在进行并行处理前的一系列转化步骤的运行效率均得到了提高，这源于并行系统共享内存的关系。在 MPI+OPenMP+STM 模式下，4 个网结构的加速比的大小关系为：∑1>∑2>∑4>∑3。这种关系的产生综合了以上两种情况，但在数值上远大于以上两种情况又证明了该模式下并行效率的高效性。且虽然网 ∑2 的处理步骤简单但网规模大，加速比较高说明了并行计算对处理大矩阵运算的高效性。

（下转第 595 页）

结束语　本文针对现有对 GIS 管道的检测方法不能同时兼顾异物清理的问题，设计并实现了一种基于 Inter Edison 平台的机器人系统。利用 Inter Edison 平台下的 Yocto Project Linux 系统，实现了 GIS 腔体信息的实时采集、处理以及外部设备的快速有效驱动。测试结果表明，设计的机器人系统在实现对 GIS 腔体可视化检测的同时还能清除腔体底部的异物杂质，提高了 GIS 设备运行的安全可靠性。但该系统仍存在需要改进的地方，这主要表现在机器人机械臂轨迹的自主规划、人机监控画面信息的实时传输等方面。

参 考 文 献

[1] 丁登伟，唐诚，高文胜，等. GIS 中典型局部放电的频谱特征及传播特性[J]. 高电压技术，2014，40(10)：3243-3251.

[2] 孙曙光，陆俭国，俞慧忠，等. 基于超高频法的典型 GIS 局部放电检测[J]. 高压电器，2012，48(4)：7-12.

[3] 齐波，李成榕，耿弼博，等. GIS 设备绝缘子高压电机故障局部放电严重程度的诊断与评估[J]. 高电压技术，2011，37(7)：1719-1727.

[4] 齐波，李成榕，骆立实，等. GIS 中局部放电与气体分解产物关系的试验[J]. 高电压技术，2010，36(4)：957-963.

[5] 骆立实，姚文军，王军，等. 用于 GIS 局部放电诊断的 SF6 分解气体研究[J]. 电网技术，2010，34(5)：225-230.

[6] 王卫东，赵现平，王达达，等. GIS 局部放电检测方法的分析研究[J]. 高压电器，2012，48(8)：13-17.

[7] 曾雄杰，刘旭明. GIS 设备局放类型图谱以及现场局放测试诊断图谱的应用[J]. 高压电器，2013，49(11)：31-36.

[8] 孙强，董明，任重，等. 现场用 GIS 冲击耐压试验及局部放电检测装置设计[J]. 高电压技术，2012，38(3)：639-644.

[9] 乔胜亚，周文俊，唐念，等. 不同吸附剂对 GIS 局部放电特征气体变化规律的影响[J]. 电工技术学报，2016，31(3)：113-120.

[10] 闫斌，何喜梅，吴童生，等. GIS 设备 X 射线可视化检测技术[J]. 中国电力，2010，43(7)：44-47.

[11] 吴章勤，魏杰，况华，等. GIS 的 X 射线数字成像检测[J]. 无损检测，2013，35(1)：11-12.

[12] 姜芸，付庄. 一种小型电缆隧道检测机器人设计[J]. 华东电力，2009，37(1)：95-97.

[13] 曹建树，林立，李杨，等. 油气管道机器人技术研发进展[J]. 油气储运，2013，32(1)：1-7.

(上接第 591 页)

通过实验对比可以得出以下结论：本文的算法解决 Petri 网的并行化是可行的且在运行效率上明显优于其他编程模式。对于判定过程复杂的 Petri 网，在做并行处理时，要处理好程序的通信问题，尽量减少进程的多线程。对于网规模越大的 Petri 网实施并行计算的效果越显著。

结束语　本文提出了基于三层混合编程模型的 Petri 网并行算法，并对 Petri 网并行划分子网的过程进行了实例分析，最后对算法的可行性进行了实验验证，采用多个 Petri 网和多组实验，进行了纵向和横向的双向比较。在已验证本文算法的有效性和高效性的前提下验证了不同网结构情况下算法的性能特征，为今后该算法的研究提供了研究方向。

参 考 文 献

[1] 李文敬，王汝凉，廖伟志. 基于 P-不变量的 Petri 网并行化方法的研究[J]. 计算机工程与设计，2009，30(16)：3758-3770.

[2] SHARMA R，KANUNGO P. Performance evaluation of MPI and hybrid MPI＋OpenMP programming paradigms on multi-core processors cluster[C]//International Conference on Recent Trends in Information Systems. 2011：137-140.

[3] RABENSEIFNER R，HAGER G，JOST G. Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes [C]// Euromicro International Conference on Parallel. 2009：427-436.

[4] 赵永华，迟学斌. 基于 SMP 集群的 MPI＋OpenMP 混合编程模型及有效实现[J]. 微电子学与计算机，2005，22(10)：7-11.

[5] 潘卫，陈燎原，张锦华，等. 基于 SMP 集群的 MPI＋OpenMP 混合编程模型研究[J]. 计算机应用研究，2009，26(12)：4592-4594.

[6] 陈辉，孙雷鸣，李录明，等. 基于 MPI＋OpenMP 的多层次并行偏移算法研究[J]. 成都理工大学学报(自然科学版)，2010，37(5)：528-534.

[7] 吴长莉. 基于 MPI 和 OpenMP 的三维 FDTD 并行算法的研究[D]. 武汉：华中科技大学，2009：10-12.

[8] 李文敬，廖伟志，王汝凉. Petri 网系统的功能划分及其并行算法[J]. 计算机工程，2009，35(21)：48-50，53.

[9] 李文敬，廖伟志，元昌安，等. 高级 Petri 网并行化预处理方法的研究[J]. 广西大学学报(自然科学版)，2013(5)：1100-1107.

[10] 柯琦，钟诚，陈清媛，等. 多核机群上通信高效的整数序列并行排序方法[J]. 计算机应用，2013，33(3)：821-824.

[11] LIN C，SNYDER L. Princilples of Parallel Programming[M]. Pearson Education，Inc.，publishing as Addison Wesley. 2009：180-220.

[12] LI W J，LI S，YUAN C A，et al. Research on Soft-ware Transactional Memory Parallel Programming Model Based on Multi-core Cluster[J]. Journal of Chinese Computer Systems，2014，35(8)：1732-1737.

[13] Gilbert Fridgen，Christian Stepanek，Thomas Wolf. Investigation of exogenous shocks in complex supply networks-a modular Petri Netapproach [J]. International Journal of Production Research，2015，53(5)：1387-1408.

[14] SOURAVLAS S I，ROUMELIOTIS M. Petri net modeling and simulation of pipelined redistributions for a deadlock-free system [J]. Cogent Engineering，2015，2(1)：1057427.

[15] WANG R Z，AGARWAL S，DAGLI C H. Executable System of Systems Architecture Using OPM in Conjunction with Colored Petri Net：A Module for Flexible Intelligent and Learning Architect-ures for System of systems[J]. Incose International symposium，2014，24(s1)：581-596.