一种暂态稳定并行仿真的改进算法及其加速比分析

王成山,杨建林,张家安,陈光远,杨晓东
(天津大学 电气与自动化工程学院,天津 300072)

摘要:提出了一种基于因子路径树网络划分的暂态稳定空间并行仿真的改进算法。采用新的网络划分性能评价指标、任务划分后续调整策略进行任务划分,使各处理器间计算负荷的分配更为合理;同时,还提出了该算法在理想状况下加速比的计算公式,分析了影响并行仿真计算效率的各个因素,并以某 3 872 节点系统为例介绍了一种估计实现最大加速比所需处理器数目的方法。在 Cluster 1 350 集群系统上的算例表明,所提改进算法提高了仿真效率。

关键词:暂态稳定;因子路径树;前代回代;加速比;Cluster 1 350
中图分类号:TM 712;TM 744 **文献标识码**:A **文章编号**:1006-6047(2006)05-0001-04

0 引言

随着电力系统互联规模的不断扩大,电力工程界对在线动态安全分析的实用需求越来越迫切,而现有的串行计算无法满足大规模电力系统暂态过程的实时仿真。自 20 世纪 80 年代以来,人们把并行处理引入电力系统仿真计算中并提出了不少优秀的并行算法。总体而言,电力系统暂态稳定并行算法可分为 3 类[1-2],即空间并行算法、时间并行算法以及时空并行算法。对空间并行算法而言,并行任务的划分往往对应着电力网络的划分,已有的划分方法有基于地理位置的划分、基于主节点的划分[3]、基于模拟退火法的划分[4]以及基于因子路径树的划分[5-6]。其中,基于因子路径树的划分方法由于考虑了稀疏线性方程组前代回代的并行性,其计算负荷分配较其他划分方案更加合理。

本文从任务划分和算法实施的角度对基于因子路径树的暂态稳定并行算法[5]进行了改进,并给出了理想状况下的并行加速比估算公式,运用此公式可根据该改进算法结果估计出某系统进行并行处理实现最大加速比时所需的处理器数目。在 Cluster 1 350 集群系统上的仿真验证表明,与原有算法相比,改进算法能使各处理器间的计算负荷分配更为合理,实现更高的加速比,提高了并行仿真的效率。

1 暂态稳定空间并行仿真算法

用于暂态稳定分析的电力系统数学模型可由如下非线性微分代数方程组表示。

$$\dot{\boldsymbol{X}}=\boldsymbol{A}\boldsymbol{X}+\boldsymbol{B}\boldsymbol{u}(\boldsymbol{X},\boldsymbol{U}) \quad (1)$$

$$\boldsymbol{I}(\boldsymbol{X},\boldsymbol{U})=\boldsymbol{Y}(\boldsymbol{X})\boldsymbol{U} \quad (2)$$

式(1)中的微分方程描述了电力系统中相关元件的动态特性,$\boldsymbol{X}$ 为动态元件的状态变量,输入 $\boldsymbol{u}$ 是状态变量 $\boldsymbol{X}$ 与电压 $\boldsymbol{U}$ 的函数;式(2)代表系统的网络方程,$\boldsymbol{Y}(\boldsymbol{X})$ 是与 $\boldsymbol{X}$ 相关的系统导纳矩阵。

空间并行一般采用分块的思想对系统进行区域划分,不同区域在不同的处理器上计算,实现对子区域系统的并行计算。网络划分是整个暂态稳定空间并行仿真的关键,一般,需将网络方程变换为式(3)所示对角加边形式 (BBDF)[7] 并行求解。动态元件微分方程的求解具有明显的空间并行性,可将其放在对应节点所在分区中计算。

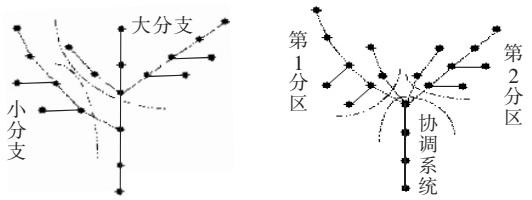
$$\begin{bmatrix} Y_{11} & & & & Y_{1t} \\ & Y_{22} & & & Y_{2t} \\ & & \ddots & & \vdots \\ & & & Y_{kk} & Y_{kt} \\ Y_{t1} & Y_{t2} & \cdots & Y_{tk} & Y_{tt} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_k \\ U_t \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_k \\ I_t \end{bmatrix} \quad (3)$$

2 算法改进

2.1 改进的并行任务划分算法

2.1.1 基于因子路径树的任务划分

1995 年,K.W.CHAN 通过移动因子路径树分支及合并其分支的方法[5]进行网络划分。以某简单 20 节点系统为例,图 1(a)为其因子路径树,图 1(b)为上移其小分支并通过合并分支得到 2 个分区的示意图。



(a) 因子路径树 (b) 分区示意

图 1 20 节点系统因子路径树及其简单划分

Fig.1 The factorization path tree and its simple partitioning scheme for a 20-bus system

收稿日期:2005-09-20
基金项目:国家自然科学基金项目(50595412)

这种任务划分方法仅考虑了网络方程的前代回代计算量,没有考虑各发电机节点对应微分方程的计算量,因此可能会引起总任务划分的不均;同时随着各分区计算量不平衡度的减少,协调系统(由连接不同分支的协调节点组成)的规模会不断增加,如何在不平衡度和协调系统规模间寻找一个较好的结合点也是一个尚未解决的问题。

为解决以上问题,本文提出了新的网络划分性能评价指标、任务划分的后续调整策略,得到了改进的因子路径树网络划分算法。

2.1.2 衡量划分结果的性能指标

为衡量划分结果的优劣,本文提出了如下性能指标:

$$J(p)=\max_{i=1,\cdots,p}(C_{\mathrm{LU}i}+C_{\mathrm{Gen}i})+C_{\mathrm{Com}}+C_{\mathrm{T}} \quad (4)$$

式(4)右端项的值越小表明划分结果越理想。式(4)中 p 为分区个数;$C_{\mathrm{LU}i}$ 为第 i 个分区中所有分支所对应的前代回代次数之和;$C_{\mathrm{Gen}i}$ 为第 i 分区中发电机等动态元件计算量,$C_{\mathrm{Gen}i}=k_1 G_{\mathrm{num}i}$,$G_{\mathrm{num}i}$ 为 i 分区中发电机的总数;k_1 为发电机节点权值,反映了与发电机节点对应动态元件计算量等价的前代回代的次数,根据仿真程序中所用动态元件模型及大量的实验测量,取 k_1 为 13;C_{Com} 代表协调系统节点前代回代计算量;C_{T} 近似代表了与并行过程中的通信损耗等价的前代回代次数,由于通信时间随着分区个数以及协调系统节点个数的增加而增加,取 $C_{\mathrm{T}}=k_2 p N_{\mathrm{Coord}}$,其中 N_{Coord} 表示协调系统的节点个数,k_2 的取值决定于集群系统中单机计算性能以及网络通信性能,根据本文所用 Cluster1350 集群系统的性能特点以及大量的实验测量,本文近似取 $k_2=2$。

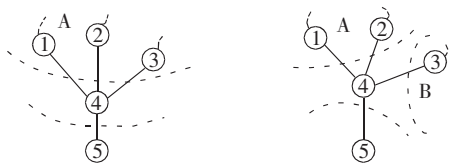
基于因子路径树的网络划分通过合并路径树分支的方法进行,最优划分方案的确定过程就是通过移动合并路径树分支寻找 $J(p)$ 最小值的过程。在改进算法中,如果因子路径树分支连续上移 n 次(本文取 $n=10$)后,$J(p)$ 的值没有变化,则认为已经没有更优的合并方案,即 $J(p)$ 已经达到了最小值。

2.1.3 任务划分的后续调整策略

当分区数较多的时候,网络划分所得的协调节点数目仍然较多,不利于并行仿真效率的提高。本文根据协调节点所连分支的不同特点提出了如下后续调整策略以进一步减小协调系统规模。

a. 当某协调节点所连分支属于同一分区 i 时,将该协调节点归入 i 分区。如图 2(a)所示,协调节点 4 与 3 个同属于 A 分区的分支以及协调节点 5 相连,则协调节点 4 被划入分区 A。

b. 当某协调节点所连分支不属于同一分区时,假设该协调节点所连分支中属于 i 分区的各分支权值之和最大,如果所有非 i 分区各分支权值之和小于某常数(本文中取 30)且 i 分区不是总体计算量最大的分区(尽量不影响划分不平衡度),则将该协调节点及非 i 分区的各分支归入 i 分区。如图 2(b)所示,协调节点 4 连有 3 个属于不同分区的分支以及协调节点 5,其中 1 和 2 分支属于分区 A,3 分支属于分区 B 且 1 和 2 分支权值之和大于 3 分支权值,当分区 A 不是总体计算量最大的分区且 3 分支权值小于 30 时,就将协调节点 4 以及 3 分支归入到分区 A 中。



(a) 分支属于同一分区 (b) 分支属于不同分区

图 2 后续调整过程图例

Fig.2 Illustration of post adjusting process

2.1.4 算法步骤

改进后并行任务划分方法的具体步骤如下。

步骤 1 利用最小度最小长度(MDML)排序方法进行预排序,以生成高度矮分支多且非零注入元少的因子路径树。

步骤 2 假设 $J(p)$ 最小值的初值为某一极大值,从因子路径树的树根出发,计算分支节点所连各分支的权值(即为分支内部前代回代次数及发电机等动态元件所对应的计算权值之和)。

a. 当分支节点所连分支数目小于分区数 p 时,把权值较小的分支沿着权值最大的分支方向移动;

b. 当分支节点所连分支的数目等于分区数 p 时,用式(4)计算此时的划分权值,并以此为起点继续移动分支直至寻找到 $J(p)$ 的最小值;

c. 当分支节点所连分支的数目大于分区数 p 时,依据权值把各分支排序,合并排在最后的两分支为一组,然后再排序,再合并直至剩下 p 个组为止,以每个组中节点为一个分区,用式(4)计算此时的划分权值,以此为起点继续移动分支直至寻找到 $J(p)$ 的最小值。

步骤 3 采用上述介绍的后续调整策略对步骤 2 的划分结果进行后续调整。

2.2 并行实现过程中的计算和通信优化

在并行实现的过程中,本文采用了部分前代回代策略来提高网络方程的计算效率,并对通信过程进行了优化。

常规算法在各分区中根据最小度(MD)原则进行节点排序[5],本文在算法实施过程中对网络方程的求解采用了部分前代回代策略。为实现该策略,首先需对各分区内的节点排序进行调整,即首先人为增加具有非零注入电流节点的出度,然后再根据 MD 原则进行节点排序。这样处理使各分区网络方程的前代回代过程只需要从第 1 个具有非零注入电流节点位置开始即可,而不必对所有节点进行前代回代操作。实践证明,这样处理可以有效地减少网络方程的前代回代计算时间。

此外,本文还对数据通信过程进行了优化:即各分区对应过程仅发送和收集与该分区相邻的协调节点信息,这样就比传输所有协调节点信息减少一半以上的数据传输量,进一步减少了通信损耗。

3　并行加速比分析

并行加速比定义为串行处理时间除以并行处理时间,是衡量并行计算性能的重要指标。在实际对某系统进行并行仿真过程中,需要解决的一个问题是应该选用多少台并行处理器,才能达到最大的并行处理加速比。

3.1　加速比的估算公式

首先,假设在理想状况下并行化后的总体计算量仍等于原串行程序的计算量,在此基础上,本文给出如下加速比 sp 估算公式:

$$sp=\frac{t_{ps}+\sum t_{fi}+t_{celse}}{t_{ps}+(\mathrm{Max}\ t_{fi})/l+t_c+t_{belse}} \tag{5}$$

式中　t_{ps} 表示并行程序中串行部分的计算时间;t_{fi} 表示第 i 分区并行处理部分在原串行程序中所需的处理时间,$i=1,\cdots,p$;l 表示考虑并行系统存储性能优势[8]的比例系数(一般 $l>1$);t_c 是通信时间;t_{celse} 及 t_{belse} 分别为串行及并行程序中的非计算时间,一般相对较小。

由式(5)可知,影响 sp 的因素很多,随着处理器 p 的增多,t_{ps} 和 t_c 增大,而 $\mathrm{Max}\,t_{fi}$ 减小。当 $\mathrm{Max}\,t_{fi}$ 减小时间等于 t_{ps} 及 t_c 增加时间之和时,sp 达到最大,如何找到这些影响因素的最佳组合,使之产生最佳的并行性能需要进一步研究。

3.2　实现最大加速比所需处理器数目估算方法

对任何系统如果通过测量能估计算出 t_{ps},t_c 等参数,即可根据式(5)求出不同处理器数目情况下 sp 的值,进而进行比较选优。

以某 3 872 节点系统为例,对其进行 5 s 暂态过程串行仿真所需时间为 8.35 s。根据单机的计算性能(即前代回代次数与计算时间之间的关系)及本文所提改进算法划分结果中的协调系统前代回代的计算量,可估计出使用 6,7 和 8 台处理器时的 t_{ps} 分别为 0.1,0.12 和 0.16 s;根据集群系统的通信性能(即通信时间与处理器个数及数据通信量之间的关系)、数据传输的规模以及通信次数(串行求解中线性方程组的求解次数),可估计出使用 6,7 和 8 台处理器时的 t_c 分别为 0.45,0.55 和 0.70 s;忽略非计算时间,为了简化问题取 $l=1$;此外,考虑到本文所提改进算法对任务的划分比较均匀,可假定各分区并行部分处理时间 t_{fi} 均相等,于是可得到估计结果如表 1 所示。

表 1　加速比估算结果

Tab.1 Estimation results of speedup

p	sp
6	4.34
7	4.52
8	4.43

由表 1 可见,当处理器的数目为 7 时,系统可能达到最大加速比。需要说明的是上述很多参数的取值是在 Cluster 1 350 集群系统上经过大量实验测量统计出来的。对硬件参数的测量和评估是所有并行处理问题中必不可少的环节。同样该方法也可适用其他并行计算平台下的加速比分析。

4　仿真测试

为了验证上述改进算法正确性和有效性,本文在 Cluster 1 350 集群系统(Network 为 1 000 M,Memory 为 1.5 GB,CPU 为 Xeon 2.0)上对本文所提改进算法和原算法的仿真结果进行了比较。

仿真中采用某 3 872 节点系统作为测试系统,该系统有 4 788 条支路,990 台发电机,发电机采用 4 阶模型,并配有励磁器调速器以及 PSS 模型,负荷采用恒阻抗模型。本文选择 5 s 的仿真过程,仿真步长为 0.01 s,测试故障为母线 7 266 在 0.1 s 时发生三相短路故障,0.2 s 时切除该故障。表 2 列出了计算中采用的处理器个数 p,仿真时间 t 及加速比 sp。

表 2　仿真性能对比

Tab.2 Comparison of simulation performance

p	原算法		改进算法	
	t/s	sp	t/s	sp
1	8.35	1.00	8.35	1.00
2	5.35	1.56	4.91	1.70
3	3.33	2.52	3.35	2.49
4	3.11	2.68	2.70	3.09
5	2.62	3.20	2.13	3.92
6	2.40	3.48	1.91	4.37
7	2.21	3.79	1.56	5.35
8	2.43	3.44	1.62	5.15

由表 2 可见,在不同分区数目情况下,改进算法的并行仿真速度较原算法均有不同程度提高,尤其在 7 分区时更是达到了最高加速比,与 3.2 节估计结果一致。此外,还发现仿真得到的 sp 值均大于表 1 的估算结果,这主要是因为在上节估算过程中,将计算性能参数 l 值取为 1,而实际集群系统的 l 要大于 1。

为了进一步说明影响 sp 的各个因素,在图 3,4 中分别列出了采用 7 台及 8 台处理器时第 i 台处理器处理并行部分的时间 t_{pi} ($i=1,2,\cdots,p$)、并行程序中处理串行部分的时间 t_{ps} 以及通信时间 t_c。

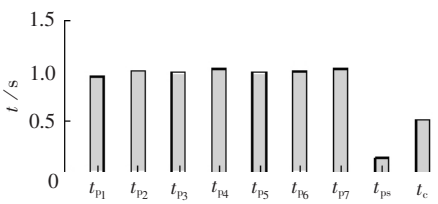


图 3　采用 7 台处理器时各部分的耗时情况

Fig.3 The time consumptions when 7 processors are used

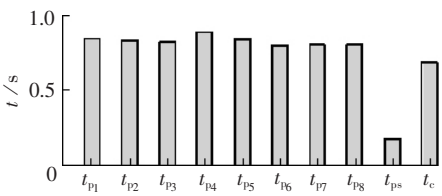


图 4　采用 8 台处理器时各部分的耗时情况

Fig.4 The time consumptions when 8 processors are used

可以看出,尽管8台处理器处理并行部分处理时间较7台处理器时有所减小,但其串行部分处理时间和通信时间的增加却更大,导致其 *sp* 反而下降,其中通信时间的增加是制约 *sp* 进一步提高的主要瓶颈[9]。另外,由图3,4还可看出,各个处理器对并行部分的处理时间相差很少,也从另一侧面说明本文对并行任务的划分是比较合理的。

5 结语

本文提出了一种基于因子路径树网络划分的暂态稳定空间并行仿真的改进算法。仿真测试表明,改进后的算法使各个处理器间计算负荷的分配更为合理,能较显著地提高暂稳计算的并行仿真速度。另外,本文还给出了理想状况下加速比 *sp* 的计算公式,分析了影响加速比的各种因素,并以某测试系统为例,介绍了一种估计实现最大加速比所需处理器数目的方法。

参考文献:

[1] 吉兴全,王成山. 电力系统并行计算方法比较研究[J]. 电网技术,2003,27(4):22-26.
JI Xing-quan,WANG Cheng-shan. A comparative study on parallel processing applied in power system[J]. **Power System Technology**,2003,27(4):22-26.

[2] 李亚楼,周孝信,吴中习. 基于PC机群的电力系统机电暂态仿真并行算法[J]. 电网技术,2003,27(11):6-12.
LI Ya-lou,ZHOU Xiao-xin,WU Zhong-xi. Personal computer cluster based parallel algorithms for power system electromechanical transient stability simulation[J]. **Power System Technology**,2003,27(11):6-12.

[3] VALE M H M,FALCAO D M,KASZKUREWICZ E. Electrical power network decomposition for parallel computations[C]// **Proceeding of the 1992 IEEE International Symposium on Circuits and Systems.** New York:IEEE,1992:2761-2764.

[4] IRVING M R,STERLING M J H. Optimal network tearing using simulated annealing[J]. **IEE Proceedigs — Generation,Transmission and Distribution**,1990,137(1):69-72.

[5] CHAN K W,DUNN R W,DANIELS A R. Efficient heuristic partitioning algorithm for parallel processing of large power systems network equations[J]. **IEE Proceedings — Generation Transmission and Distribution**,1995,142(6):625-630.

[6] 洪潮,单巍. 在IBM-SP2上实现电力系统暂态稳定计算的一种并行算法[J]. 电力系统及其自动化学报,2001,13(1):18-22.
HONG Chao,SHAN Wei. A parallel algorithm for power system transient stability simulation on an IBM-SP2 parallel computer[J]. **Proceedings of the EPSA**,2001,13(1):18-22.

[7] HAPP H H. 分块法及其在电力系统中的应用[M]. 丘昌涛,译. 北京:科学出版社,1987.

[8] BUYYA R. 高性能集群计算[M]. 郑炜民,石威,汪东升,译. 北京:电子工业出版社,2001.

[9] CHAI Jian-sheng,BOSE A. Bottlenecks parallel algorithms for power system stability analysis[J]. **IEEE Transactions on Power Systems**,1993,8(1):9-15.

(**责任编辑**:李育燕)

作者简介:

王成山(1962-),男,天津宝坻人,长江学者特聘教授,博士研究生导师,主要从事电力系统安全性分析、城市电网规划和配电系统自动化等方面的研究工作;

杨建林(1980-),男,山西太原人,硕士研究生,研究方向为电力系统并行仿真(**E-mail**:yjlscuttju@yahoo.com.cn);

张家安(1975-),男,河北衡水人,博士,主要研究方向为电力系统仿真及其分布式计算。

Improved parallel algorithm for transient stability simulation and analysis of its speedup

WANG Cheng-shan,YANG Jian-lin,ZHANG Jia-an,
CHEN Guang-yuan,YANG Xiao-dong
(Tianjin University,Tianjin 300072,China)

Abstract: An improved parallel-in-space algorithm based on factorization path tree partitioning is proposed for transient stability simulation. To make the distribution of computing load among processors more reasonable,a new performance index for assessing the partitioning scheme and a post-adjusting strategy are adopted for the task assignment in this algorithm. A formula,which can be used to estimate the speedup under the ideal circumstance for this new algorithm is given,and the factors which influence the efficiency of the parallel simulation is analyzed. A method to evaluate the quantity of processors needed to achieve the maximum speedup is presented and is applied to a 3872-bus power system for test. The simulation results obtained on the Cluster 1 350 for the test power system validates that the improved algorithm has higher efficiency than the conventional one.

This project is supported by National Natural Science Fund of China(50595412).

Key words: transient stability; factorization path tree; forward and backward substitution; speedup; Cluster 1 350