

DOI: 10.7667/PSPC161586

大电网潮流修正方程并行求解实现方法

张国芳¹, 罗雅迪², 李静², 戴旭¹, 代宇涵¹

(1. 国网四川省电力公司, 四川 成都 610041; 2. 中国电力科学研究院, 北京 100192)

摘要: 针对智能电网调度控制系统对大规模潮流快速计算的需求, 充分利用现有系统计算资源, 提出了一种适用于共享内存编程模型的潮流修正方程多路多核并行实现方法。利用 C++ 标准容器, 简化了稀疏矩阵的存储和遍历, 并基于图论和共享内存编程模型, 对因子分解过程进行并行化改造, 实现了潮流修正线性方程的并行求解。最后, 对比智能电网调度控制系统调度员潮流软件, 进行了分析测试。测试结果表明, 随着计算规模的增大, 所提出的实现方法计算效率越高, 验证了方法的实用性。

关键词: 线性方程; 图论; 因子分解; 潮流计算; OpenMP; 并行计算

Parallel solution method of power flow correction equation for large-scale power grid

ZHANG Guofang¹, LUO Yadi², LI Jing², DAI Xu¹, DAI Yuhan¹

(1. State Grid Sichuan Electric Power Company, Chengdu 610041, China;

2. China Electric Power Research Institute, Beijing 100192, China)

Abstract: According to the needs as rapid computing of large-scale power flow calculation in the smart grid dispatching and control systems, and making full use of the computing resources of existing systems, this paper presents a multi-CPU and multi-core parallel method for power flow correction equation, which is suitable for shared-memory programming model. The storage and traversal of sparse matrix is simplified by C++ standard containers, and the factor decomposition process is transformed in parallel based on the graph theory and the shared memory programming model, and the parallel solution for power flow linear correction equation is realized. Finally, the comparison between the current dispatcher power flow software of smart grid dispatching and control system and the above parallel power flow software is down, and the test results show that, the computational efficiency proposed by this paper is better with the increasing of computing scale, which verifies the feasibility of above program.

Key words: linear equation; graph theory; factor decomposition; power flow; OpenMP; parallel computing

0 引言

随着特高压交直流互联大电网的快速建设, 以及风电等间歇性新能源的大规模接入, 各级调控中心智能电网调度控制系统在线分析软件的计算规模不断扩大, 且实时调控对其计算实时性提出了更高的要求。潮流计算是调度员潮流、静态安全分析、自动电压控制、暂态稳定分析、电压稳定分析和最优潮流等在线分析应用的核心计算模块^[1-4], 其计算效率的提升具有重要意义。

目前, 用于潮流并行计算的方法主要包括并行

迭代法、网络分块和因子路径并行等^[5-6], 这些并行计算方法可利用多路多核 CPU^[7-8]或协处理器 GPU^[9-10]实现加速, 提升潮流修正方程的求解速率。迭代法较易实现并行化, 但可能存在收敛性和加速效果的问题^[11]。网络分块并行算法将电网分解成若干子网, 根据子网和子网间联络线进行节点排序, 形成对角加边的系数矩阵形式, 从而分块并行求解^[12-15]; 基于因子路径的并行算法利用潮流修正方程的稀疏性, 可将消去路径树分层或按路径进行并行计算, 这种方法起始加速效果好, 后期由于矩阵稀疏性发生变化, 并行度降低^[16-17]。文献[17]提出了一种适用于 GPU 的基于道路树分层的稀疏矩阵直接分解算法, 取得了不错的加速效果, 但该方法的使用需要在服务器上安装 GPU, 且 CUDA 并行编程需要

基金项目: 国家电网公司科技项目“四川省地调度控制系统一体化模型管理平台及电网分析应用关键技术研究”

相应版本 Linux 操作系统支持^[18], 难以在现有智能电网调度控制系统中得到实际应用, 其推广应用存在一定难度。近年来, 关于潮流并行计算方法方面的研究较多, 都具有一定的加速比, 然而在一定程度上编程模型和程序实现方法的优劣决定了潮流并行计算的效率。

本文提出了一种适用于共享内存编程模型的并行因子分解方法。该方法使用 C++ 标准容器实现因子表的存储和遍历, 改进并简化了传统稀疏矩阵的存储方法, 同时给出了并行因子分解的具体程序实现方案, 在现有智能电网调度控制系统上, 基于共享内存编程模型 OpenMP 实现了并行潮流计算, 提高了系统潮流计算的实时性。

1 因子分解

1.1 潮流修正方程

常用的潮流计算方法主要有牛顿-拉夫逊法和 P-Q 分解法, 其修正方程分别如式(1)和式(2)所示^[19]。

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = - \begin{bmatrix} H & N \\ K & L \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \frac{\Delta V}{V} \end{bmatrix} \quad (1)$$

$$\begin{aligned} \Delta P/V &= B' \Delta \theta \\ \Delta Q/V &= B'' \Delta V \end{aligned} \quad (2)$$

可将其表示为以下形式:

$$Ax = b \quad (3)$$

采用因子分解法求解上述线性方程。

牛顿-拉夫逊法修正方程的系数矩阵 A 为非对称, 可将 A 分解为单位下三角矩阵 L 、对角线矩阵 D 和单位上三角矩阵 U 的乘积形式, 其中 L 和 U 的对角元素均为 1, 可表示为 $A = LDU$ 。引入中间矢量 z 和 y 后, 可以用式(4)进行前代回代求解 x 。

$$\begin{cases} Lz = b \\ Dy = z \\ Ux = y \end{cases} \quad (4)$$

P-Q 分解法修正方程的系数矩阵 A 为对称矩阵, 通常只稀疏存储其上三角(或下三角)部分, 以存储矩阵上三角部分为例, 因子分解后可得矩阵 U , 而下三角矩阵 L 可在前代时按式(5)还原其元素, 可将矩阵 A 的因子分解记为 $U^T D U$ (或 LDL^T) 的形式。

$$l_{ji} = u_{ij} \times d_{ii} \quad (j > i) \quad (5)$$

1.2 因子分解

以牛顿-拉夫逊潮流计算为例, 其系数矩阵 A 的因子分解过程可分为规格化和消去两部分运算^[12]。矩阵 A 的第 i 行的规格化只需要对上三角部分中的第 i 行非零元进行计算, 若第 i 行 j 列为非零元, 则

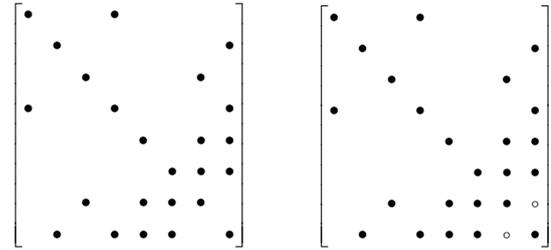
$$a_{ij} = \frac{a_{ij}}{a_{ii}} \quad (j > i) \quad (6)$$

消去下三角第 l 行 k 列非零元时, l 行消去运算为

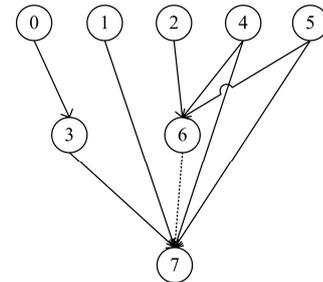
$$a_{lp} = a_{lp} - a_{lk} \times a_{kp} \quad (p > k) \quad (7)$$

当 $a_{lp} = 0$, $a_{kp} \neq 0$ 时, 则 l 行消去运算后 a_{lp} 为注入元。

上述因子分解过程可用有向图 $G(A) = (V, E)$ 进行描述, 如图 1(c)所示。图 1(a)和图 1(b)分别是矩阵 A (采用静态优化法对其进行节点排序) 及其因子表 A' , 图 $G(A)$ 的每个顶点 v_i 对应矩阵 A 的第 i 行, 图 $G(A)$ 的边 e_{ij} 对应矩阵 A 第 i 行消去第 j 行, 图 1(c)虚线表示对注入元的消去, 顶点 v_i 入度 $d_G^-(v_i)$ 为 i 行被消去次数, 其出度 $d_G^+(v_i)$ 为 i 行消去其他行的次数。



(a) 矩阵 A (b) 因子表 A'



(c) 因子分解过程图 $G(A)$

图 1 矩阵 A 及其因子分解过程图

Fig. 1 Diagram matrix A and its factor decomposition

2 因子分解的并行化

若因子分解对应有向图 $G(A)$ 顶点 v_i 的入度 $d_G^-(v_i) = 0$ 且出度 $d_G^+(v_i) > 0$, 则该顶点对应的行不会被其他行消去, 可直接进行规格化运算; 若 $d_G^-(v_i) = n (n > 0)$, 则该行会被消去 n 次, 然后可进行规格化运算; 顶点每被消去一次, 则入度 $d_G^-(v_i)$ 减 1, 每消去其他行一次则 $d_G^+(v_i)$ 减 1; 在因子分解过程中, 若 $d_G^-(v_i) = 0$ 且 $d_G^+(v_i) = 0$, 则该顶点完成所有计算, 可从有向图 $G(A)$ 中删除。定义规格化集合:

$$S^{(m)} = \{v_i \mid d_G^-(v_i) = 0\} \quad (8)$$

即因子分解过程中入度为零的顶点集合, 其中定义 m 为因子分解的深度。 $S^{(m)}$ 中顶点所指向的顶点集合为

$$B^{(m)} = \{f \mid f: S^{(m)} \rightarrow B^{(m)}\} \quad (9)$$

定义为消去集合, 显然 $S^{(m)}$ 集合中的顶点对应的行可分组同时进行规格化运算, 而集合 $B^{(m)}$ 中的顶点则对应被集合 $S^{(m)}$ 消去的行, 也可分组进行消去运算。

以图 1(a)矩阵 A 为例, 当 $m=1$ 时, 集合 $S^{(1)}$ 元素为 0、1、2、4 和 5 号顶点, 集合 $B^{(1)}$ 的元素为 3、6 和 7 号顶点; 当 $m=2$ 时, 集合 $S^{(2)}$ 元素为: 3 和 6 号顶点, 集合 $B^{(2)}$ 的元素为 7 号顶点。集合 $S^{(1)}$ 元素对应行分多个线程可同时进行规格化运算, 而对集合 $B^{(1)}$ 元素进行消去时, 1 和 5 号顶点不能同时消去 7 号顶点, 2、4 和 5 号顶点不能同时消去 6 号顶点, 为避免不必要的锁竞争, 需要根据式(9)再对集合 $B^{(m)}$ 中的顶点进行合理分组。前代回代和因子分解过程相似^[12], 可用上述方法进行并行化求解。

在实际电网潮流计算修正方程中, 初始入度 $d_G^-(v_i) = 0$ 的顶点对应系统中发电机组所连节点或辐射网末端负荷所连节点, 而实际系统中该类节点较多, 因此能够起到较好的加速效果。

3 基于共享内存编程模型的程序实现

3.1 共享内存编程模型

共享内存模型是指一个并行程序由多个共享内存的并行任务组成, 数据的交换通过隐含地使用共享数据来完成。OpenMP 是一种标准的 fork/join 式并行模式, 其基本思想是, 程序开始时只有一个主线程, 程序中的串行部分都由主线程执行, 并行的部分是通过派生其他线程来执行, 待派生线程执行完成后回到主线程, 通过编译指导语将程序分为多个线程并行执行, 由编译器自动完成线程派生等功能^[20]。

目前智能电网调度控制系统一般使用高性能的应用服务器, 一般配置两路以上 CPU, CPU 核心数超过 8 个, 支持 32 个以上并行线程, 内存配置大于 32 G, 而潮流计算规模在几百节点至几万节点不等, 因此可利用系统现有服务器计算资源提升潮流计算模块性能。

3.2 基于容器的因子表存储

利用 C++ 抽象数据类型标准库提供的顺序容器

vector 和关联容器 map 实现潮流计算修正方程系数矩阵的存储。定义对象 `vector<map<int,double>> A` 用于保存系数矩阵 A 。其中 `map<int,double>` 用于保存一行元素, 列号由 int 型的键索引表示, 稀疏矩阵非零元由 double 型的关联值表示。利用迭代器 `map<int,double>::iterator iter` 即可实现对行非零元素的遍历, 其中 `iter->first` 表示列号, `iter->second` 表示非零元的值。

遍历系数矩阵元素的算法伪代码如下:

```
for i ← 0 to n
    iter ← B[i].begin()
    while(iter != B[i].end()){
        iter->second
        iter++
    }
```

相比传统稀疏矩阵存储方法, 本文方法简化了矩阵存储和遍历, 便于程序实现且执行效率高。

3.3 并行程序实现

首先遍历系数矩阵 A 的非零元素, 统计每行元素被消去次数, 同时在 `map<int,double>> A` 中增加注入元数位置, 将最大被消去次数记为 `max`。定义变量 `vector<int> muster` 和 `vector<vector<int>> vec_muster(muster,max)` 保存集合 $S^{(m)}$ 中的元素, `vec_muster[i]` 中保存入度 $d_G^-(v_i) = l$ 的行号。根据并行进程个数将 `vec_muster[0]` 中的行号分成若干组, 利用编译指导语句 “`#pragma omp parallel for num_threads(N)`” 将主进程分成 N 个线程, 用多个线程进行并行规格化运算和消去运算, 被消去行被消去次数减 1, 收集被消去次数变成 0 的行号, 并保存在 `vec_muster[1]` 中。再将 `vec_muster[1]` 分组进行多线程并行规格化和消去运算, 收集被消去次数变成 0 的行号, 程序流程如图 2 所示。依此类推, 直至完成因子表生成。前代的消去过程和生成因子表的消去过程一致, 可以利用 `vector<vector<int>> vec_muster` 中的行分组信息进行并行前代运算。规格化运算可以用编译指导语句 “`#pragma omp parallel for`” 完成 for 循环的并行执行。

在并行消去过程中需要注意多个线程不能对同一个非零元进行消去运算, 因此, 可以利用 OpenMP 提供的锁操作函数, 防止并行程序对同一内存的同时操作。由于各个线程不一定会同时执行到调用锁函数的行, 且数据分组时考虑集合 $S^{(m)}$ 和 $B^{(m)}$ 关联性, 因此, 采用锁操作不会大幅度降低并行程序执行效率, 但也不能消去其对并行计算性能的影响。随着集合 $S^{(m)}$ 基数的减小, 派生线程开销增大, 程序转换为串行计算, 可进一步提升计算效率。

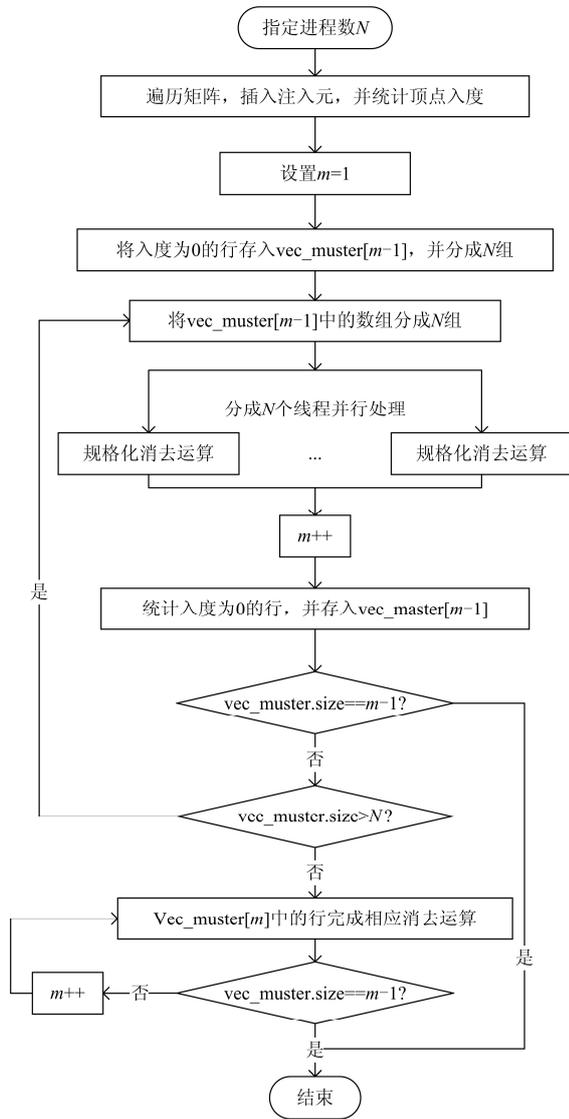


图 2 并行因子表解法程序流程

Fig. 2 Program flow of parallel factor table solution

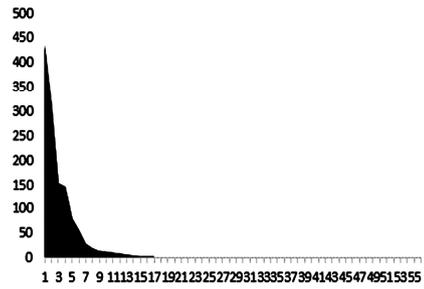
4 算例分析

分析不同规模实际电网牛顿拉夫逊法潮流计算修正方程的系数矩阵，得到集合 $S^{(1)}$ 的基数和潮流计算规模关系如表 1 所示，因子分解过程中集合 $S^{(m)}$ 的基数随 m 变化情况(即并行因子分解过程中有向图 $G(A)$ 顶点的分布情况)如图 3 所示。

表 1 电网矩阵的集合 $S^{(1)}$ 基数及 m 最大值

Table 1 Cardinal number of set $S^{(1)}$ and max m for grid matrix

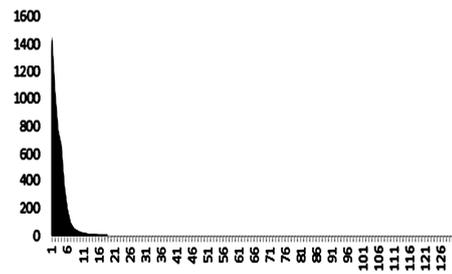
电网编号	节点数	支路数	矩阵维数	$ S^{(1)} $	$\max(m)$
1	781	1170	1328	435	55
2	1396	2041	2459	756	75
3	2647	4668	4935	1461	129
4	6251	9195	11 271	3479	105



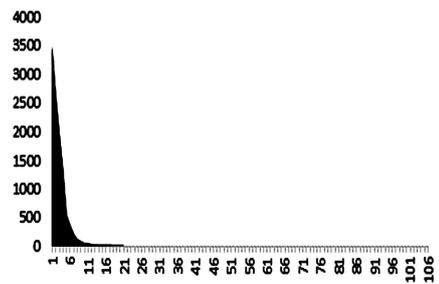
(a) 电网 1



(b) 电网 2



(c) 电网 3



(d) 电网 4

图 3 电网矩阵的集合 $S^{(m)}$ 基数

Fig. 3 Cardinal number of set $S^{(m)}$ for grid matrix

分析图 3 数据不难发现，顶点大部分都分布在第 1 层到第 10 层中。当 m 小于 10 时，入度为 0 的可并行顶点占总数的 90% 以上，具有较好的并行度，随着因子分解的推进，入度为 0 的顶点数逐渐减少，

并行度下降。当 m 大于 20 时, 入度为 0 的顶点较少, 并行度十分低, 当入度为 0 的顶点仅有 1 个时, 已经是完全的串行计算, 而本文采用 OpenMP 编程模型 fork/join 的并行模式, 能够很好地适应因子分解过程中并行度的变化, 根据 $|S^{(m)}|$ 值的变化, 及时完成并行和串行的转换, 具有较好的加速效果。表 1 数据中, m 值随着矩阵维数的增大而增大, 但电网 3 的矩阵规模小于电网 4 的, 而支路数和节点数的比值大于电网 4 的, 其 m 值反而大于电网 4 的, 可见因子分解的深度不仅和矩阵规模有关, 还和电网支路数有关, 支路越多, 矩阵越稠密, 其并行因子分解的整理加速效果降低。

在智能电网调度控制系统典型配置服务器上对本程序进行耗时测试, 其中串行耗时为智能电网调度控制系统潮流计算所需时间, 并行耗时为本程序 20 线程并行计算所需时间, 潮流算法均采用牛顿-拉夫逊法。测试服务器配置 2 块 Intel Xeon E5-2680CPU, 主频 2.5 GHz, 共 24 核心。测试软件平台为凝思 Linux 4.2, 安装智能电网调度控制系统测试环境。计算设定收敛精度为 0.001, 耗时为 30 次计算的平均时间, 单位 ms, 加速比公式为

$$\text{加速比} = \frac{\text{串行耗时}}{\text{并行耗时}} \quad (9)$$

由表 2 测试结果可知, 本程序从 1396 节点算例开始, 呈现加速效果, 随着计算规模的增大, 加速比增大, 具有较好的加速效果。相比文献[17]提出的基于 GPU 加速的潮流并行算法, 计算规模在 12 000 节点以下本文方法加速比具有优势, 计算规模在 20 000 节点以上时文献[17]加速比具有一定优势。

表 2 算例计算耗时测试结果

Table 2 Test results of computing time consuming

电网编号	节点数	迭代次数	串行耗时	并行耗时	加速比
1	781	4	7	8	0.875
2	1396	6	16	15	1.066 667
3	2647	9	67	49	1.367 347
4	6251	12	124	69	1.797 101
5	12 754	15	514	243	2.115 226
6	24 147	16	1546	614	2.517 912

5 结论

针对智能电网调度控制系统现状, 提出了适用于共享内存编程模型的潮流修正方程并行求解方法, 利用服务器多路多核 CPU 对潮流计算进行加速, 简化了稀疏矩阵存储方法, 提升了程序执行效率。由于目前智能电网调度控制系统服务器一般配

置多路多核 CPU, 而未配置 GPU, 现阶段本文方法具备推广和实用价值。若未来系统能够配置 GPU 用于计算提速, 则可采用不同计算规模使用不同加速方法来提升各级调控系统的潮流计算效率。

参考文献

- [1] 于尔铿, 刘广一, 周京阳, 等. 能量管理系统(EMS) [M]. 北京: 科学出版社, 1998: 168-174.
- [2] 殷自力, 陈宇星. 基于功率传输转移分布因子的简化电网潮流计算方法[J]. 电力系统保护与控制, 2016, 44(17): 25-30.
YAN Zili, CHEN Yuxing. A power flow computation method for reduction grid based on power transfer distribution factor[J]. Power System Protection and Control, 2016, 44(17): 25-30.
- [3] 蒋智化, 刘连光, 刘自发, 等. 网格式中压直流配电网潮流计算与稳态功率分布特性研究[J]. 电网与清洁能源, 2016, 32(3): 1-7.
JIANG Zhihua, LIU Lianguang, LIU Zifa, et al. Research on power flow calculation and steady state power distribution in meshed medium voltage DC distribution network[J]. Power System and Clean Energy, 2016, 32(3): 1-7.
- [4] 肖盛, 王斐, 唐玮, 等. 智能电网在线安全稳定预警系统研究及应用[J]. 电网与清洁能源, 2016, 32(8): 23-28.
XIAO Sheng, WANG Fei, TANG Wei, et al. Research and application of online security early warning system for smart grid[J]. Power System and Clean Energy, 2016, 32(8): 23-28.
- [5] 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 Xingquan, WANG Chengshan. A comparative study on parallel processing applied in power system[J]. Power System Technology, 2003, 27(4): 22-26.
- [6] 薛巍, 舒继武, 王心丰, 等. 电力系统潮流并行算法的研究进展[J]. 清华大学学报(自然科学版), 2002, 42(9): 1192-1199.
XUE Wei, SHU Jiwu, WANG Xinfeng, et al. Advance of parallel algorithms for power flow simulation[J]. Journal of Tsinghua University (Science and Technology), 2002, 42(9): 1192-1199.
- [7] 傅志生, 白晓清, 李佩杰, 等. 一种快速求解大规模安全约束最优潮流的多核并行方法[J]. 电力系统保护与控制, 2015, 43(3): 29-37.
FU Zhisheng, BAI Xiaqing, LI Peijie, et al. A high-speed multi-core parallel method for solving large-scale

- security constrained OPF[J]. Power System Protection and Control, 2015, 43(3): 29-37.
- [8] 孟晓丽, 唐巍, 刘永梅, 等. 大规模复杂配电网三相不平衡潮流并行计算方法[J]. 电力系统保护与控制, 2015, 43(13): 45-51.
MENG Xiaoli, TANG Wei, LIU Yongmei, et al. Parallel computing of three-phase unbalanced power flow in large-scale complex distribution network[J]. Power System Protection and Control, 2015, 43(13): 45-51.
- [9] 夏俊峰, 杨帆, 李静, 等. 基于 GPU 的电力系统并行潮流计算的实现[J]. 电力系统保护与控制, 2010, 38(18): 100-110.
XIA Junfeng, YANG Fan, LI Jing, et al. Implementation of parallel power flow calculation based on GPU[J]. Power System Protection and Control, 2010, 38(18): 100-110.
- [10] 周挺辉, 赵文恺, 严正, 等. 基于图形处理器的电力系统稀疏线性方程组求解方法[J]. 电力系统自动化, 2015, 39(2): 74-80.
ZHOU Tinghui, ZHAO Wenkai, YAN Zheng, et al. A method for solving sparse linear equations of power systems based on GPU[J]. Automation of Electric Power Systems, 2015, 39(2): 74-80.
- [11] 谷同祥. 大型稀疏线性代数方程组的并行非定常迭代方法[D]. 北京: 中国工程物理研究院, 2001.
GU Tongxiang. Parallel nonstationary iterative methods for solving large sparse linear systems[D]. Beijing: China Academy of Engineering Physics, 2001.
- [12] 张伯明, 陈寿孙, 严正. 高等电力网络分析[M]. 北京: 清华大学出版社, 2007: 58-87.
- [13] 李传栋, 房大中, 杨金刚, 等. 大规模电网并行潮流算法[J]. 电网技术, 2008, 32(7): 34-39.
LI Chuandong, FANG Dazhong, YANG Jingang, et al. New research on parallel power-flow calculation for large-scale power system[J]. Power System Technology, 2008, 32(7): 34-39.
- [14] 黄彦全, 肖建, 刘兰, 等. 基于支路分割方法的电力系统潮流并行协调算法[J]. 电网技术, 2006, 30(4): 21-25.
HUANG Yanquan, XIAO Jian, LIU Lan, et al. A Coordinational parallel algorithm for power flow calculation based on branch cutting[J]. Power System Technology, 2006, 30(4): 21-25.
- [15] 孙新民, 毛承雄, 陆继明, 等. 对角加边模型的并行潮流计算[J]. 电网技术, 2002, 26(1): 22-25.
SUN Xinmin, MAO Chengxiong, LU Jiming, et al. Parallel load flow calculation of block bordered model[J]. Power System Technology, 2002, 26(1): 22-25.
- [16] 洪潮, 沈俊明. 求解大型稀疏线性方程组的一种并行算法及其在并行潮流计算中的应用[J]. 武汉水利电力大学学报, 2000, 33(4): 29-34.
HONG Chao, SHEN Junming. A parallel algorithm for solving large sparse matrix equations and its application to parallel power flow calculation[J]. Engineering Journal of Wuhan University, 2000, 33(4): 29-34.
- [17] 陈德扬, 李亚楼, 江涵, 等. 基于道路树分层的大电网潮流并行算法及其 GPU 优化实现[J]. 电力系统自动化, 2014, 38(22): 63-69.
CHEN Deyang, LI Yalou, JIANG Han, et al. A parallel power flow algorithm for large-scale grid based on stratified path trees and its implementation on GPU[J]. Automation of Electric Power Systems, 2014, 38(22): 63-69.
- [18] 孙统华, 李东, 李松泽, 等. CUDA 并行程序设计 GPU 编程指南[M]. 北京: 机械工业出版社, 2014: 52-55.
- [19] 邱晓燕, 刘天琪, 黄媛. 电力系统分析的计算机算法[M]. 北京: 中国电力出版社, 2015: 36-64.
- [20] 邓倩妮. 并行程序设计导论[M]. 北京: 机械工业出版社, 2012: 139-143.

收稿日期: 2016-09-22; 修回日期: 2016-12-06

作者简介:

张国芳(1985—), 女, 硕士研究生, 研究方向为电力市场、电力系统稳定控制; E-mail: 364783936@qq.com

罗雅迪(1986—), 男, 通信作者, 学士, 研究方向为电力调度自动化及其高级应用。E-mail: luoyadi_cau@163.com

(编辑 魏小丽)