

A Novel Method of Parallel Computation for the Whole Scene Test in Power System

Shi Jing*, Qi Huang, Jianbo Yi

*Sichuan Provincial Key Lab of Power System Wide-area Measurement and Control
University of Electronic Science and Technology of China, Chengdu 611731, China*

Abstract

In this paper, design and implementation of a new parallel computing method based on CUDA (Compute Unified Device Architecture) platform is described in detail. The method includes algorithm of matrix fraction and partial LU decomposition that are used to support parallel computing for simulation of the whole scene test in power system. The paper describes all the steps of algorithm implementation for deploying the software with CUDA technology. And performances are evaluated by test, in which two Jacobi matrix equations of power flow calculation with different dimension are selected as the test case. The results are compared with that of serial computing and MPI (Message Passing Interface). The results show that the method proposed in this paper has better performance of acceleration on CUDA platform, which can support the simulation of the whole scene test on wide-area power grid.

Keywords: Power system, the whole scene test, CUDA platform parallel computing, partial LU decomposition

1. Introduction

The whole scene test method for intelligent substation is a new technology, which can identify correctness of actions of protection devices and configures for information network. Especially, because of remaining wiring under working condition, it can implement the test with all or partial of protection devices in once experiment, which is benefit for interaction examination [1]. By synchronization function of GPS, the test method can apply to wide-area power grid. Fig. 1 is structure of the test system.

The whole scene test system drives secondary system of substation to work, which takes place of electrical transducer by numerical simulation as input source. With simulating stationary-state or transient-state status, we can acquire test data for kinds of working condition.

For the simulation, there are scores of line nodes and electrical devices in intelligent substation. And if the test is extended to wide-area power grid, the number of them is hundred of times above the level in one substation. There is a problem about computing time in the simulation, which can lead to inefficiency of the test process, or even fail.

Parallel computation is a good way to settle the problem. More researches in power system have paid attention to it [2]. A new upsurge in cloud-computation technology is in the making. The platform of cloud-computation can implement parallel operation for power system by using resources of computation in wide-area network. But it does not fit the whole scene test system, because the experiment site does not readily set up cloud network.

The most suitable scheme for the whole scene test is to establish platform of parallel computation in one PC (PC, personal computer). CUDA (Compute Unified Device Architecture) technology makes it actual because of using resources of GPU (Graphic Processing Unit) on view card to accelerate process of computation [3].

* Manuscript received July 14, 2012; revised August 15, 2012.

Corresponding author. Tel.: +86-13658008738; E-mail address: elitejs@163.com.

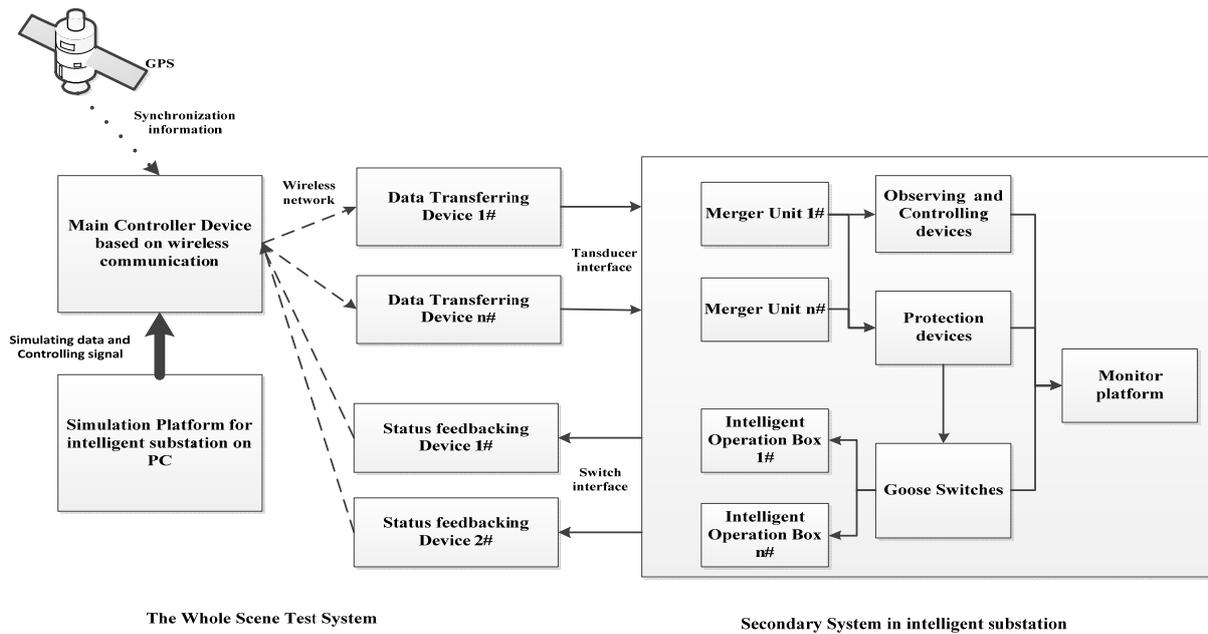


Fig.1. Structure of the whole scene test system working in intelligent substation.

This paper describes a method of parallel computation based on CUDA technology, which has been used in simulation for the whole scene test system.

2. CUDA Technology Based on GPU

CUDA technology is a programmed model based on GPU which is released by NVIDIA Corporation. With CUDA model, resources of CPU and GPU can cooperative work to accelerate process of computation.

Thread is a basic element in the model. And one thread can be distributed to one ALU (arithmetic logical unit) of GPU. In all the ALUs, processes of computation are mutual independence. This feature can be used for parallel computation.

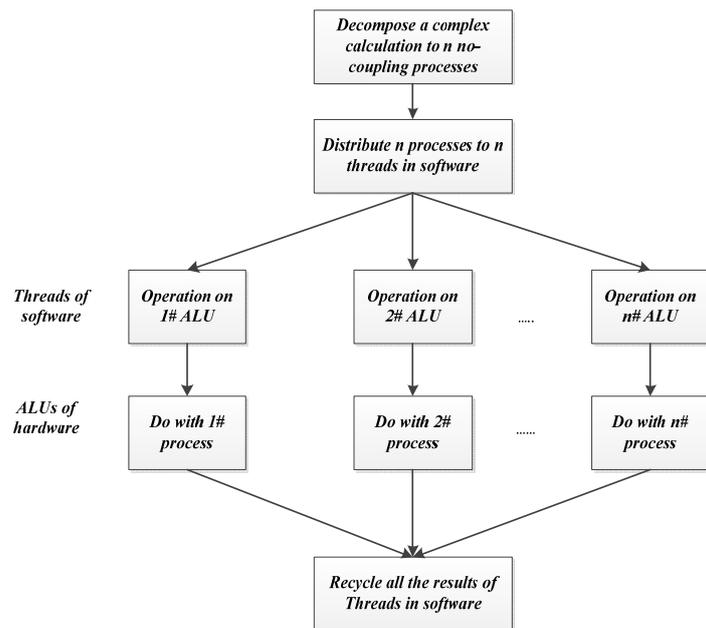


Fig.2. Parallel scheme for CUDA platform.

We can decompose a complex calculation into several no-coupling slave processes. By distributing them to independent ALUs, parallel operation can be gained. There are hundreds of ALUs in presented GPU, whose resources are so enough as to achieve better performance for parallel computation in power system. Fig. 2 is a scheme for CUDA platform to finish parallel computation

3. Algorithm for Parallel Computation

It is a basic question to solve equation as $Ax=b$ for stationary-state or transient-state computation in power system. We design an algorithm of parallel computation on it, which is adequate for CUDA platform. There are two steps in the process.

Step1:

Matrix A is put into Block-Bordered Diagonal Form (BBDF) by fraction algorithm, such as A in (1).

$$\begin{bmatrix} A_{11} & & & & A_{1n} \\ & A_{22} & & & A_{2n} \\ & & \ddots & & \vdots \\ & & & A_{n-1,n-1} & A_{n-1,n} \\ A_{n1} & A_{n2} & \cdots & A_{n,n-1} & A_{n,n} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n-1} \\ B_n \end{bmatrix} \quad (1)$$

where A_{ij} is partitioned matrix. And $A_{11} \sim A_{n-1,n-1}$ are called diagonal blocks. $A_{n,n}$ is called border block.

Step 2:

By using partial LU decomposition, the solution of (1) is parallel calculated. And Step1 is linear transformation of matrix. The properties of matrix are not undermined. So the positive and soft-diagonal definiteness of matrix A can be used in parallel process of solution.

3.1. Algorithm of matrix fraction

The method named as Path Tree of Factor Table has been used in some parallel computation and achieved good results, which can disclose parallel properties of solution process for matrix equation. Minimum Degree-Minimum Number of Predecessors (MD-MNP) is a better algorithm for power system in it. Existing research has shown that quantity of arterial elements of path tree in MD-MNP is fewer than production of the other algorithm [4].

Node admittance matrix (NAA) which describes electrical connections of buses in power system is the object for matrix fraction algorithm. We define a matrix T , which is a topology matrix of NAA. The nonzero elements of NAA are set one in consistent places of T , the others are zero. And T' is a BBDF of T .

The flow chart of MD-MNP for applying to software is shown in Fig.3. Some descriptions of symbol in it are listed here:

- $D(i)$ is original degree of the i th node in T . It describes the number of nodes connected with the i th one.
- $P(i)$ is present degree of the i th node in path tree. Its value is one plus the number of predecessors connected with the i th one. And initial value is 1.
- $C[i][j]$ is a relation table of branches and leaves in path tree. It is a matrix of the same dimension with T . If $C[i][j]$ is 1, it is shown that the j th node is connected with the i th node and the j th one is a predecessors of the i th one.

$P(i)$ and $C[i][j]$ are output values of MD-MNP. We can make a path tree described by matrix form with the benefit of them. First, we select arterial nodes with maximum value in $P(i)$. Then, according to $C[i][j]$, we select branch nodes and leaf nodes connected with arterial nodes. By contrast with (1), we adjust position of the entire nodes. Arterial nodes are placed on A_{nn} , and the other nodes in the same branch are placed on one of $A_{11} \sim A_{n-1,n-1}$. With presented process, T is transformed to T' .

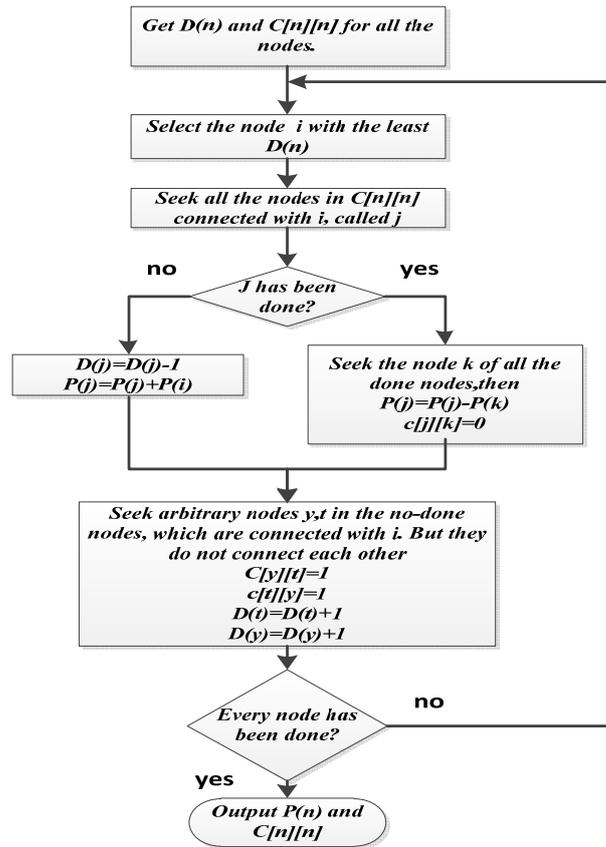


Fig.3 Flow chart for MD-MNP in software.

However, T' or T dose not directly help solution process of $Ax=b$ in power system. In fact, partial matrices of A in $Ax=b$ equation match along with connections of every bus node (such as Jacobi matrix).If they were regarded as matrix elements, A is the same form with T . So the partial matrices of A can be set to BBDF according to new sequence of nodes in T' .

3.2. Algorithm of partial LU decomposition

LU decomposition algorithm is a common way for the solution of linear equation in serial computation. Reference [5] has popularized it to parallel computation, called partial LU decomposition method. Because A of $Ax=b$ is nonsingular and positive matrix in stationary-state and transient computation for power system, the method is numerical stability. With BBDF achieved, we can design a partial LU decomposition algorithm for CUDA platform.

We define two types of threads in CUDA model. One is main thread. It is responsible for distributing computation tasks for diagonal blocks and implementing computation for border block. The other is slave thread, which finish computation for border block. The number of slave threads depends on quantity of diagonal blocks.

In equation (1), we separate $A_{n,n}$ and B_n into sum form of $n-1$ item in main thread.

$$\sum_{i=1}^{n-1} A_{nii} = A_{n,n} \tag{2}$$

$$\sum_{i=1}^{n-1} B_{ni} = B_n \tag{3}$$

where A_{nii} and B_{ni} are items distributed to the i th slave thread. Main thread transfers $A_{ii}, A_{iv}, A_{ni}, A_{nii}, B_i, B_{ni}$ to the i th thread. So a slave node equation is established in this GPU, as:

$$\begin{bmatrix} A_{ii} & A_{in} \\ A_{ni} & A_{nii} \end{bmatrix} \begin{bmatrix} X_i \\ X_{ni} \end{bmatrix} = \begin{bmatrix} B_i \\ B_{ni} \end{bmatrix} \quad (4)$$

If there are $n-1$ slave threads, we can get $n-1$ equation as (4). Then, LU decomposition is implemented in every slave threads. The decomposition result of A matrix in (4) can be regarded as Fig. 5.

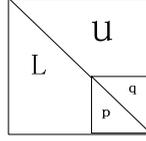


Fig.5. Structure of A matrix of slave nodes after LU fraction.

Following LU decomposition, p and q are elements blocks in position of A_{nii} in (4).

Set $Y=UX$, we can get (5)

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 \\ l_{1,0} & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ l_{k,0} & l_{k,1} & \vdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{m,0} & l_{m,1} & \cdots & l_{m,k} & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \\ \vdots \\ b_m \end{bmatrix} \quad (5)$$

And elements of A_{nii} are from the k th to the m th row. Solving (5) by forward calculating, the results before k row are such as

$$\begin{cases} y_0 = b_0 \\ y_t = b_t - \sum_{r=0}^{t-1} l_{t,r} y_r, \quad t = 0 \sim k-1 \end{cases} \quad (6)$$

The values of y are not needed after k row. We needs revised values for b . So we can get

$$b_t = b_t - \sum_{r=0}^{k-1} l_{t,r} y_r, \quad t = k \sim m \quad (7)$$

With calculating in (7), new elements in B_{ni} of (4) are achieved. And new elements in A_{nii} can be calculated.

We get new $(n-1)$ A_{nii} and B_{ni} from $n-1$ slave threads. Returning them to main thread, we accumulate them with negative process for (2) and (3). In main thread, a linear equation about X_n corresponding to border block is established.

$$A_{n,n} X_n = B_n \quad (8)$$

Because of fewer dimension of (8), serial way can be used to solve it.

With gaining X_n , we distribute it to $n-1$ slave threads. And we get (9) in them

$$\begin{bmatrix} u_{0,0} & u_{1,0} & \cdots & u_{k,0} & \cdots & u_{m,0} \\ & u_{1,1} & \cdots & u_{k,1} & \cdots & u_{m,1} \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & u_{k,k} & \cdots & u_{m,k} \\ & & & & \ddots & \vdots \\ & & & & & u_{m,m} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_k \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \\ \vdots \\ y_m \end{bmatrix} \quad (9)$$

x_k, \dots, x_m are elements in vector X_n . By back calculating from the $(k-1)$ th row to the 0 th row, we can get

$$x_t = \left(y_t - \sum_{r=m-1}^{t+1} u_{t,r} y_r \right) / u_{t,t}, \quad t=k-1 \sim 0 \quad (10)$$

By (11), elements of X_i corresponding to diagonal block can be calculated in its distributed threads. By collecting X_i in every slave thread and getting X_n in (8), the solutions of $Ax=b$ have been got completely.

Obviously, with CUDA platform, the processes of LU decomposition on partial matrices, forward and back solution for X_1, \dots, X_{n-1} are parallel executed in presented algorithm for solving $Ax=b$. Only X_n is got by serial calculation.

4. Performance test

In this paper, we make Jacobi matrix equation in power-flow computation as test object. Test sources come from IEEE-300 bus data and 1342 bus data combined by several IEEE bus data.

With fraction of MD-MNP, 300 bus nodes are divided into six blocks. Average dimension of diagonal blocks is 54. The dimension of border block is 31. 1342 bus nodes are divided into twelve blocks. Average dimension of diagonal blocks is 120. The dimension of border block is 24.

The hardware platform of test is listed at Table 1

We adapt performance of method of serial computation and MPI compared with it of parallel method of this paper. Newton algorithm is in serial computation. And algorithm in MPI is the same with the method in this paper. Table 2 is the result of test.

In simulation of the whole scene test system, the solution process of $Ax=b$ can be hundred of times, which depends on time step and total time of simulation. Following improvement on the number of nodes, the advantage of CUDA platform on performance of acceleration is more obvious.

Table 1. Configure of hardware platform

CPU	I7-2620M, 2.7GHZ
MEMORY	2GB DDR3
GPU	NVIDIA QUADRO 3000
OS	32-bit WINDOWS XP

Table 2. Result of test with solution of Jacobi matrix

Node	Serial computation on 1 PC	MPI		CUDA
		1 PC	4 PC	Muti-threads on 1 PC
300	2.39s	1.00s	0.65s	0.68s
1342	147.77s	1.84s	0.97s	0.88s

5. Conclusions

This paper presents a novel method of parallel computation based on CUDA platform. The accelerate performance of it in one PC is about the same with MPI on Muti-PC, which improves efficiency of simulation. Taking advantage of it, the simulation for transient-state can support the on-loop experiment for power system, whose time step is only in hundred of μ s. So the whole scene test system with CUDA technology can be popularized to test for wide-area power grid.

Acknowledgements

This work was supported by “The Fundamental Research Funds for the Central Universities” under Grand No. ZYGX2009X014.

References

- [1] Wu J, Huang Q, Jing S, *et al.* Design on hardware platform of whole-scene testing apparatus for smart substation based on FPGA and ARM. *Sichuan Electric Power Technology*, 2012; 35(2):15-18.

- [2] Jing S. Design and Implementation of Parallel Operation Components for Power System on Grid Computation Platform. PhD dissertation. University of Electric Science and Technology of China. Chengdu, China; 2006.
- [3] Zhang NY, Gao S, Zhao X. A fine granularity parallel algorithm for electromechanical transient stability simulation based on graphic processing unit. *Automation of Electric Power Systems*, 2012; 36(9):54-60.
- [4] Ji XQ, Wang CS. A comparative study on parallel processing applied in power system. *Power System Technology*, 2003; 27(4):22-26.
- [5] Hong C, Shen JM. A parallel algorithm for solving large sparse matrix equations and its application to parallel power flow calculation. *Journal of Wuhan University of Hydraulic and Electric Engineering*, 2000; 33(4):29-34.
- [6] Lau K, Tylavsky DJ. Coarse grain scheduling in parallel triangular factorization and solution of power system matrices. *IEEE Trans. on Power Systems*, 1991; 6(2):708-714.
- [7] Qiu JJ, Lo KL. Parallel algorithm studies in power systems-the big branch method based on the sparse vector technique and comparisons with other methods. *Power System Technology*, 1995; 19(7):708-714.
- [8] Gomez A, Franquelo LG. An efficient ordering algorithm to improve sparse vector methods. *IEEE Trans. on Power Systems*, 1988; 3(4):1538-1544.
- [9] Zhou XX, Han ZX, Tian F, et al. Concept and mechanism on full-process dynamic real-time simulation of power system with parallel-in-time-space. In: *Proc. of 2010 International Conference on Power System Technology*, 2010:1-7.