Inferring Genome-Wide Gene Regulatory Networks with GPU or CPU Parallel Algorithm

Ming Zheng
Guangxi Colleges and Universities Key Laboratory of Professional Software Technology
Wuzhou University
Wuzhou, China
E-mail: 370505375@qq.com

Shugong Zhang
College of Mathematics
Jilin University
Changchun, China
E-mail: zhangsg@jlu.edu.cn

Mugui Zhuo
Guangxi Colleges and Universities Key Laboratory of Professional Software Technology
Wuzhou University
Wuzhou, China
E-mail: 756456050@qq.com

Guixia Liu*
College of computer science and technology
Jilin University
E-mail: liugx@jlu.edu.cn
*The corresponding author

***Abstract*—Expression of gene block, with the GPU parallel thread structure characteristic calculation, according to the structural characteristics of GPU thread design of double parallel mode, and the use of texture cache memory to achieve high efficiency; on the basis of CPU two level cache capacity of basic blocks further subdivided into sub blocks to improve the cache hit rate, the technology to reduce the number of memory accesses the use of data, reduce the thread migration in the core between the use of thread binding technology; according to the calculated capacity allocation of multi-core CPU and GPU CPU and GPU gene in the mutual information calculation task to calculate the load balance of CPU and GPU; in the design of the new threshold calculation algorithm based on the design and implementation of memory efficient construction of global gene control network CPU /GPU parallel algorithm. The experimental results show that compared with the existing algorithms, this algorithm speed is more obvious, and is able to build more large-scale global gene regulation Control network.**

Keywords-Genome-wide; Gene regulatory network; CPU /GPU cooperative computing; Efficient access cache; Parallel algorithm

I. INTRODUCTION

With the complete genome sequence of the human genome work sketch, multiple model organisms, after genomics genome era main focus from sequencing steering function research[1]. Analysis of gene expression microarray technology makes the establishment of global gene regulatory networks become possible, but the construction of gene regulatory network is very difficult[2]: every eukaryotic organisms have tens of thousands of genes, leading to the gene regulatory network to build a special complex; there is no model of a mature method, from gene expression analysis of gene regulatory relations spectrum map; there are a lot of noise and affect the gene expression significantly, increased the difficulty of constructing gene regulatory networks.

At present the construction of gene regulatory network model are: Bayesian network model[3] and mutual information model[4]. The Bayesian network model into directed acyclic graph model and hidden Markov chain to describe the relationship between Bayesian network variables and interactions, to construct regulatory network models. However, the Bayesian model of exponential time complexity in the construction of large-scale global the efficiency of gene regulatory network is very low[5]. Butte and IKohane proposed the use of mutual information as the detection of gene regulation relationships between complex tools, experiments show that the network model based on mutual information in the construction of regulatory network quality and time complexity and has obvious advantages[6]. The mutual information algorithm for constructing gene regulatory networks based on mostly serial algorithm the construction control network of approximately one thousand genes. These serial algorithms can only eukaryotic Creatures generally consist of tens of thousands of genes. The establishment of global gene regulatory network requires 109 orders of number of mutual information calculation.

This proposed algorithm is a global gene regulatory network platform design and implementation of multi-core CPU/GPU[7] in the parallel collaborative heterogeneous computing, the main contributions are as follows: the parallel construction of gene regulatory network model design, the design and implementation of the new regulatory threshold selection algorithm; design and implementation of the CPU and GPU memory efficient parallel computing gene the mutual information algorithm.

II. CONSTRUCTION OF GENE REGULATORY NETWORK MODEL BASED ON MUTUAL INFORMATION

A. Mutual information estimation

Mutual information measures the correlation between two event sets, and the mutual information of the two events X and Y is defined as[8]:

$$I(X,Y)=H(X)+H(Y)-H(X,Y) \tag{1}$$

H(X,Y) is the joint entropy in Eq. (1). The H(X,Y) can be shown as below:

$$H(X,Y)=-\sum_{x\in X,y\in Y}P(x,y)\log P(x,y) \tag{2}$$

Where P(x, y) is the joint probability of X and Y, mutual information can be expressed as:

$$I(X,Y)=\sum_{x\in X,y\in Y}P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \tag{3}$$

Mutual information I (X, Y) is a function of probability, can be estimated by using mutual information kernel function[9]. X n samples of known variable value, the variable X probability density function f (x) kernel function estimation:

$$\hat{I}(x)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{x-x_i}{h}) \tag{4}$$

K is called the kernel function, h for the window width or smooth parameters. The window width parameter is usually equal to:

$$h\approx[\frac{4}{(d+2)}]^{\frac{1}{d+4}}zn^{-\frac{1}{d+4}} \tag{5}$$

Where d is the dimension of the data set, z is the standard deviation of the sample data. By Eq. (3), I (X, Y) of the estimated value is shown as below:

$$\hat{I}(X,Y)=\frac{1}{n}\sum_{i=1}^{n}\log\frac{\hat{f}(x,y)}{\hat{f}(x_i)\hat{f}(y_i)} \tag{6}$$

The KSTest of the gene expression data in the gene expression profile shows that the kernel function is chosen to be normal distribution, so the kernel function is selected by Gauss function:

$$K(x)=(\frac{1}{\sqrt{2\pi}})^{e^{-\frac{x^2}{2}}} \tag{7}$$

Finally, the estimated I(X,Y) can be obtained as below:

$$\hat{I}(X,Y)=\frac{1}{n}\sum_{i=1}^{n}\log\frac{n\sum_{j=1}^{n}e^{-\frac{n^{2/6}}{2}[\frac{(x_j-x_i)2}{z_1^2}-\frac{(x_j-x_i)(y_j-y_i)}{z_1z_2}+\frac{(y_j-y_i)2}{z_2^2}]}}{\sum_{j=1}^{n}e^{\frac{-(x_j-x_i)2}{2(\frac{4}{3n})^{\frac{2}{5}}z_1^2}}\sum_{j=1}^{n}e^{\frac{-(y_j-y_i)2}{2(\frac{4}{3n})^{\frac{2}{5}}z_2^2}}} \tag{8}$$

B. Mutual information estimation parallel model

The expression is subdivided into basic block gene; secondly, the computing tasks allocated to CPU and GPU cooperative computing; thirdly, the basic block is further subdivided into sub blocks, the multi-core parallel computing effective caching; finally, the design of GPU terminal two layers diagonal parallel computation, to achieve efficient access.

First of all the basic blocks, each GPU parallel diagonal matrix calculation results on all the matrix blocks, the block matrix is the two gene expression profile of basic block calculation, each block corresponding to the two basic block calculation. Then to base because the unit, each GPU parallel computing all the genes in a diagonal line inside the matrix blocks on the value of mutual information calculation, each thread corresponds to a pair of genes. In order to make GPU a parallel matrix block diagonal can be calculated for all genes on the mutual information value, set up the basic block containing gene number is equal to the number of threads in a thread block. The GPU parallel computing strategy agreement CUDA thread structure the characteristics, can increase the utilization rate of hundreds of core processing in GPU.

Each diagonal parallel computing all the genes on the value of mutual information, every expression of the need to calculate the matrix blocks gene basic blocks are different, such as the (i, j), the calculation of matrix blocks to gene expression of No. I and No. J basic block basic block spectrum expression. Two copies of this gene for data storage, you can make each diagonal parallel computing have no access conflict.

C. Determination of the regulation relationship

The threshold is an important parameter to control whether the evaluation of two gene regulation relationship, accurate determination of this parameter is difficult. This paper calculates the mutual information between the 200 gene values, and these values are sorted before 1000 increments (two mutual information and each adjacent difference value plotted) as shown in the figure.

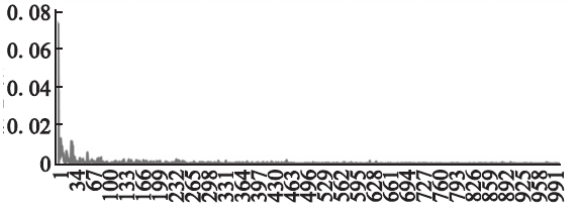

Figure 1. Incremental curve for mutual information

From the picture we can see that the change of mutual information was large and then leveled off, which shows obvious inflection point mutual information curve. At the same time, Fig .1. Curve jitter phenomenon is obvious, and the first change of mutual information is too large. So how to eliminate the chattering phenomenon to accurately identify the inflection point (threshold) is one of the key problems. This paper competition scoring system, using the following

method to eliminate the effects of jitter to accurately find the threshold value: the sort of mutual information of all genes calculated; the minimum mutual information value, calculate the increment between them, get rid of one of the largest and the smallest one for the rest of the average increment, increment value; the increment threshold for alpha times to average, of which 0.001 < alpha < 0.1; if the increment between 10 consecutive mutual information are small In the incremental threshold, the corresponding mutual information value is the desired control threshold.

But the mutual information on all eukaryotic gene values about hundreds of millions, of mutual information values for all the sort of large computational complexity; and the position corresponding to the inflection point threshold should be in the mutual information value that is relatively small. Therefore, this paper calculates the threshold to remove the mutual information minimum mutual information a value of 5%, and then refer to the "two search" threshold selection method from the 5% in the value of mutual information.

III. ANALYSIS OF THE PROPOSED PARALLEL ALGORITHM

The main idea of the algorithm: gene expression profiling is subdivided into basic blocks, with basic blocks on the diagonal parallel computing; distributed computing times for CPU and GPU; according to the two level cache multi-core structure of the capacity of basic blocks further subdivided into sub blocks, and the next time to calculate the required data to prefetch cache GPU; end take double diagonal parallel computing, the use of texture memory bound data.

Global gene regulatory network result matrix is very large, the construction of gene regulation network of the 50 thousand genes the matrix size is about 10GB, a single GPU memory to store the entire result matrix, so take part the result of each GPU storage, then summary results.

Algorithm 1. Constructs a parallel algorithm of CPU and GPU for global gene regulation network

Gene expression profile

Gene regulatory network

Begin

(1) read the gene expression profile, and according to the GPU thread size BlockRowCount, calculation of NumBlock and calculation of basic blocks round ComputeCount;

(2) calculation is assigned to CPU and each GPU round ComputeCountCPU calculation

And ComputeCountGPU;

(3) do steps (3.1), (3.2) in parallel

(3.1) call CPU parallel computing mutual information algorithm (algorithm 2);

(3.2) call multi GPU parallel computing mutual information algorithm (algorithm 3);

(4) summary of the result matrix returned by multiple GPU;

(5) the threshold value calculation algorithm (algorithm 4) is used to calculate the threshold value and the threshold is used to filter the mutual information matrix;

(6) the mutual information matrix of gene was analyzed by DPI, and the control network was further simplified;

End

The algorithm 1 is mainly based on GPU thread structure partition, then according to the calculation ability of CPU and GPU will calculate the corresponding rounds assigned to CPU and GPU, in order to achieve load balance.

Algorithm 2.CPU parallel computing gene pair mutual information algorithm

Input: gene expression profile, ComputeCountCPU, CPU thread number thread-NumCPU

Output: CPU end mutual information calculation result matrix

Begin

(1) calculate the number of lines of the w block, the basic block is further divided into sub blocks, the number of sub blocks are SubNum, and calculate the parameters of k = w /threadNumCPU;

(2) with the instruction prefetch expression basic block prefetch to level three cache memory gene number zeroth, and a copy of this, two pieces of data were recorded as basic blocks A and B;

(3) for I = 0 to do BlockNum1

(3.1) for J = 0 to do ComputeCountCPU1

Do steps (3 1.1) ~ (~ 3) in parallel

(3.1.1) with the three level cache prefetch instruction No. I No. zeroth block gene expression profile in a basic block read to the two level cache, and a copy of this, two blocks are respectively denoted as sub block SA and sb;

(3.1.2) for Si = 0 to do SubNum1

(3.1.2.1) = 0 for SJ to do SubNum1

Do steps (3 1 1.1) ~ (3 1 2 1.2) in parallel ()

(3.1.2.1.1) = 0 to for TID par-do TheadNumCPU1

For SWI = 0 to do W1

For swj = tid* to (TID + 1) * k do K

Begin

The Y (SWI + swj)% w gene Y of the swj gene X and the sub block sb in the sub block SA is read into the primary cache from the two level cache; the mutual information of the X and the;

End

(3.1.2.1.2) if (SJ + 1) < SubNum based prefetch instruction reads the gene number I from the three level cache spectrum basic block in the SJ + 1 chant block to the two level cache replacement block sb expression;

End for

(3.1.2.2) if (Si + 1) < SubNum based prefetch instruction reads the gene number I from the three level cache spectrum basic block in the Si + 1 chant block to the two level cache replacement block SA expression;

End for

(3.1.3) with a prefetch instruction from main memory into the (I + j + 1) expression of basic blocks to level three cache replacement basic block B%BlockNum gene;

End for

(3.2) with prefetch instructions read from main memory I + expression of basic block to level three cache replacement basic block A 1 gene;

End for

End

Algorithm 2 according to the cache capacity of CPU, made a further subdivision of the basic block into several sub

blocks, and then prefetch the basic block, sub block to level three, level two cache, the number of accessing main memory was significantly reduced. The partition can make the three level cache can be transferred to the 4 basic block, divided the sub block can make the two level cache can be transferred to 4 sub blocks, so the three level cache and level two cache can accommodate the next calculation calculation and the data needed to achieve zero loss. At the same time the use of "cache latency hiding" model, computing and memory access overlap, forming multilevel pipeline model, make the calculation the process has been accelerated.

Algorithm 3 multi GPU parallel computing gene pair mutual information algorithm

Input: gene expression profile, ComputeCountGPU, GPU thread block size

Output: the mutual information result matrix for each GPU gene

Begin

For each GPU do in parallel

(1) specify a calculation GPU;

(2) from the memory transfer of gene expression data to GPU, and the use of 2D texture structure of these data is bound to the texture memory;

(3) for I to do in parallel BlockNum1 = 0

For J = 0 to do ComputeCountGPU1

For Ti = 0 to do in parallel BlockRowCount1

For TJ = 0 to do BlockRowCount1

Begin

The number of Ti (Ti + J - 1) gene expression X (I + Tj)% BlockRowCount gene in the basic block of the gene expression profile of I gene was studied. The mutual information between X and Y was calculated by Y;

End

(4) from the GPU memory to memory transfer matrix results;

End for

End

IV. EXPERIMENT

A. Experimental and Experimental Data

The experimental data from the public gene expression database GEO[10], this paper used two groups of gene expression data: contains 32996 genes and each gene has 25 sample data set GSE7431, contains 54675 genes and each gene has 143 sample data sets GSE22148.

The experimental platform for the 2 XEON E5620 2 4GHz 4 core Intel processor and 4 GPU (4 × Nvidia Tesla C2050 3GB) of the multi-core computer, the memory capacity of 12GB, sharing the three level cache capacity of 12MB, the two level cache capacity of each core private 512KB, a cache capacity of 64KB, operation the system is running red Hat Enterprise 5 Linux, OpenMP and CUDA using C language programming.

B. Experimental Results and Analysis

For data set GSE7431, run 1, 2, 3 and 4 GPU, respectively, each thread block in the operation of the thread, Fig .2 gives the algorithm in this paper, the parallel computation of the 3 genes on the time required for mutual information:

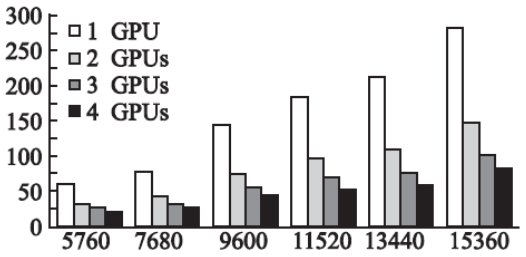

Figure 2. Required time to execute Algorithm 3 running GPUs with different number to compute mutual information

The experimental results show that the more GPU algorithm operation, less calculation is needed for the gene mutual information time; in addition, can also see that the computation time is about running n GPU. single run this shows that GPU can effectively enhance the performance of parallel computing, algorithm, this algorithm is suitable for the operation of 3 in GPU system, with good scalability.

Table 1 gives the data set GSE7431 gene on serial computing mutual information algorithm, this algorithm 2, algorithm 3 run 64 threads running 4 GPU and each thread block has 192 threads, respectively calculate the genes required to mutual information time, speedup and parallel algorithm 2 and 3 obtained.

TABLE I. REQUIRED TIME TO COMPUTE MUTUAL INFORMATION USING SERIAL ALGORITHM, ALGORITHM 2 AND ALGORITHM 3

number	time	Algorithm 2		Algorithm 3	
		time	Speedup	time	Speedup
960	291	257	6	4698	23
2240	1587	1	32	8673	48
3520	3924	81	73	7423	53
4800	7311	93	130	704	55
6080	11724	8	216	521	54
7360	17203	8	308	268	55

From Table 1, the experimental results show that the parallel multi core CPU and GPU parallel computing of genes has accelerated effect on mutual information. For the 960 gene data for smaller, because the parallel overhead of CPU parallel algorithm running time accounted for a larger proportion, so the acceleration effect is not obvious, the speedup is only 23; when the data size increases to a certain extent, multi-core CPU parallel algorithm of acceleration is relatively stable, about 55. 960 genes for small data size, GPU parallel algorithm and computation time than CPU in parallel, it is because the GPU communicates with the CPU time of a larger proportion, influence the performance of the algorithm when; the data size increases to a certain extent, the speedup increases rapidly, the GPU parallel algorithm's advantage is obvious, this shows that the GPU parallel algorithm is suitable for large-scale data gene The calculation of mutual information.

For the GSE7431 data set, each thread block has 192 threads, the next page is shown in Fig and CPU algorithm 3 run 64 threads and GPU threads block has 192 threads parallel time calculation algorithm in this paper 1 gene on the mutual information.

V. CONLUSION

CPU and GPU proposed the parallel computing of the gene mutual information algorithm to build more large-scale global gene regulation networks and significantly shorten the construction of global gene regulatory network is the most complex gene computation time of mutual information, because it is on the gene expression profile of block, according to the structural characteristics of GPU parallel thread computing according to the structural characteristics of GPU, design the double thread parallel mode, and the use of texture cache memory to achieve high efficiency; based on nuclear CPU cache, the basic block further subdivided into sub blocks to ensure cache zero loss, take the technology to reduce the number of memory accesses the data pre, reduce the thread migration in the core between the use of thread binding technology; the task to achieve CPU and GPU load balancing through the rational allocation of the CPU and GPU calculation. The next step will be the reference of community discovery thoughts on global gene Module partition method of control network

ACKNOWLEDGMENT

This work was supported by grants from The National Natural Science Foundation of Chi-na (No. 61502343, No. 61373051, and No. 61402423), China Postdoctoral Science Foun-dation funded(No. 2016M590260), the Guangxi Natural Science Foundation (No. 2015GXNSFBA139262), the Science Research Funds for the Guangxi Universities (No. KY2015ZD122), Guangxi Colleges and Universities Key Laboratory of Professional Software Technology, Wuzhou University.

REFERENCES

[1] Carter, M.Q.: 'Decoding the Ecological Function of Accessory Genome', Trends Microbiol., 2017, 25, (1), pp. 6-8

[2] Fujii, C., Kuwahara, H., Yu, G., Guo, L.L., and Gao, X.: 'Learning gene regulatory networks from gene expression data using weighted consensus', Neurocomputing, 2017, 220, pp. 23-33

[3] Fan, Y., Wang, X., and Peng, Q.K.: 'Inference of Gene Regulatory Networks Using Bayesian Nonparametric Regression and Topology Information', Computational and Mathematical Methods in Medicine, 2017

[4] Chen, C., and Yan, X.F.: 'Optimization of a Multilayer Neural Network by Using Minimal Redundancy Maximal Relevance-Partial Mutual Information Clustering With Least Square Regression', IEEE Trans. Neural Netw. Learn. Syst., 2015, 26, (6), pp. 1177-1187

[5] Thorne, T.: 'NetDiff - Bayesian model selection for differential gene regulatory network inference', Scientific Reports, 2016, 6

[6] Kurt, Z., Aydin, N., and Altay, G.: 'Comprehensive review of association estimators for the inference of gene networks', Turkish Journal of Electrical Engineering and Computer Sciences, 2016, 24, (3), pp. 695-U1401

[7] Wei, R., and Murray, A.T.: 'A parallel algorithm for coverage optimization on multi-core architectures', Int. J. Geogr. Inf. Sci., 2016, 30, (3), pp. 432-450

[8] Zu-yun, F.: 'Information theory: basic theory and application' (2007, 2nd Edition edn. 2007)

[9] Yan, X.Y., Zhang, S.W., and Zhang, S.Y.: 'Prediction of drug-target interaction by label propagation with mutual interaction information derived from heterogeneous network', Molecular Biosystems, 2016, 12, (2), pp. 520-531

[10] Barrett, T., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., Holko, M., Yefanov, A., Lee, H., Zhang, N.G., Robertson, C.L., Serova, N., Davis, S., and Soboleva, A.: 'NCBI GEO: archive for functional genomics data sets-update', Nucleic Acids Res., 2013, 41, (D1), pp. D991-D995