Load Balancing Algorithm of GPU Based on Genetic Algorithm

Zhang Xiangyang*, Feng Chaomin, Zhao Shugui. Wen Ling. Li Changchun
PetroChina Research Institute of Petroleum Exploration &Development-Northwest,
Lanzhou, Gansu Province 730020, China
*Corresponding author, e-mail: zxy_petro@163.com

Abstract

As the development of GPU/CPU parallel computing in recent years, load balance of GPU server has been more and more important, so we promote a genetic algorithm-based load balancing algorithm for GPU RTM. The algorithm takes the server status and job assignment into account, and design a coding mechanism and genetic manipulation, as well as fitness function. The experiments show that,the algorithm can reach a better effect of efficiency and load-balancing. It can hidden data transmission in the parallel computing, and during server downtime, it can prevent the idle of other computing resources.

Keywords: *reverse-time migration, genetic algorithm, load balancing, GPU/CPU heterogeneous parallel computing, GPU Server*

1. Introduction

With the increasing difficulty of oil exploration, reverse time migration, full waveform inversion and other new processing technologies as well as GPU and other equipments have been applied in seismic data processing. Relative to the CPU server, GPU server greatly improved the parallelism and computational efficiency of reverse time migration, and has a good effect of reverse time migration. However, due to the lack of load balancing strategy or just through simple method to achieve load balancing of the reverse-time migration algorithm based on GPU server, there is not suitable load balancing algorithm for the algorithms based on GPU server. In this paper, a Genetic Algorithm based load balancing algorithm of reverse-time migration has been designed to adapt to the characteristics of GPU servers, and make different batches and different models of GPU servers dynamically assign tasks depending on the load of servers, so that all servers can be fully utilized.

Genetic algorithm (GA Genetic Algorithm) is a class of self-organizing and adaptive artificial intelligence technique by simulating evolution and mechanisms of natural biological to solve the problem [1-5]. By the encoding, fitness function and genetic manipulation algorithms, GA can be used to obtain environmental information to search and adjust the search direction. Through this self-organizing, adaptive characteristics, GA can automatically discover the laws of the environment, so that it is very suitable for real-time status changing environment of application and servers, according to the current state of the server decide task allocation. In summary, this paper presents a Genetic Algorithm based dynamic load balancing algorithm of reverse-time migration that takes the each server status and server type into consideration together, it can flexibly adjust task for each server. On the one hand, based on different types of servers, it allocate different amounts of task, on the other hand, adjusts assignments based on server load state, so that, it can make full use of servers and achieve the effect of load-balance, thereby increasing the load capacity of servers [6-11].

2. Parallel Algorithm for Reverse-time Migration

Reverse time migration is the seismic data migration method that using two-way wave equation, the basic formula is as follows:

$$\frac{1}{v^2}\frac{1}{\rho}\frac{\partial^2 P}{\partial t^2}=\frac{\partial}{\partial x}(\frac{1}{\rho}\frac{\partial P}{\partial x})+\frac{\partial}{\partial y}(\frac{1}{\rho}\frac{\partial P}{\partial y})\frac{1}{v^2}\frac{1}{\rho}\frac{\partial^2 P}{\partial t^2}=\frac{\partial}{\partial x}(\frac{1}{\rho}\frac{\partial P}{\partial x})+\frac{\partial}{\partial y}(\frac{1}{\rho}\frac{\partial P}{\partial y}) \tag{1}$$

Received October 1, 2013; Revised January 11, 2014; Accepted February 3, 2014

Where, $P = P(x,y,z,t)$ is the pressure field of medium, $\rho = \rho(x,y,z)$ is the medium density, $v = v(x,y,z)$ is the velocity field, $s(x,y,z,t)$ is the source term. With the high order difference or compact difference scheme is used to solve the Equation (1), it can be used for the numerical simulation of wave propagation. Dablain(1986) already discussed in detail the higher order finite difference solution of the three-dimensional two-way wave equation. Here just list the basic calculation formula of the forward and inverse extrapolation [12-15].

The 3d high order finite difference wave equation, with truncation error is $O\left(\Delta x^M, \Delta y^M, \Delta z^M, \Delta t^2\right)$, is as follows:

$$\begin{aligned}
u_{i,j,k}^{n\pm1} &= 2u_{i,j,k}^{n} - u_{i,j,k}^{n\mp1} + \frac{1}{2}\left(\frac{v\Delta t}{\Delta x}\right)^2 \left[\omega_0 u_{i,j,k}^{n} + \sum_{m=1}^{\frac{M}{2}} \omega_m \left(u_{i+m,j,k}^{n} + u_{i-m,j,k}^{n}\right)\right] \\
&+ \frac{1}{2}\left(\frac{v\Delta t}{\Delta y}\right)^2 \left[\omega_0 u_{i,j,k}^{n} + \sum_{m=1}^{\frac{M}{2}} \omega_m \left(u_{i,j+m,k}^{n} + u_{i,j-m,k}^{n}\right)\right] \\
&+ \frac{1}{2}\left(\frac{v\Delta t}{\Delta z}\right)^2 \left[\omega_0 u_{i,j,k}^{n} + \sum_{m=1}^{\frac{M}{2}} \omega_m \left(u_{i,j,k+m}^{n} + u_{i,j,k-m}^{n}\right)\right]
\end{aligned} \tag{2}$$

The implementation of reverse-time migration with no load-balance is shown in Figure 1, the source data has been totally distributed at the beginning, with no consideration of the server type and server load state, resulting in the servers with faster processing speed waiting the servers with slower processing speed for a long time. And once a server failure, job processing time will be longer again; all the other servers must be waiting for the fault server to finish its task.

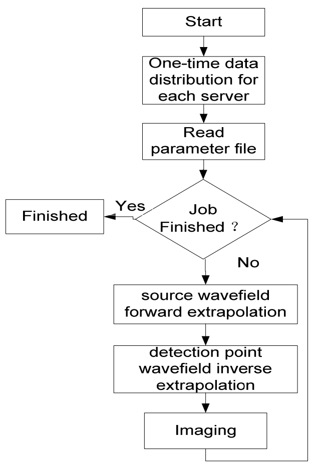
Start
One-time data distribution for each server
Read parameter file
Job Finished ?
Yes
Finished
No
source wavefield forward extrapolation
detection point wavefield inverse extrapolation
Imaging

Figure 1. Parallel Algorithm of Reverse-time Migration with on Load Balance

Because of the problems of reverse-time migration with no load-balance, through the optimization of software, design the polling algorithm based load balancing algorithm for reverse-time migration, as shown in Figure 2. This algorithm can assign task during the seismic data processing, and avoid that all the other servers wait for the fault server, but because of the algorithm without considering the server load state and the type of server, job completion time difference is bigger, and still cannot make full use of the servers.

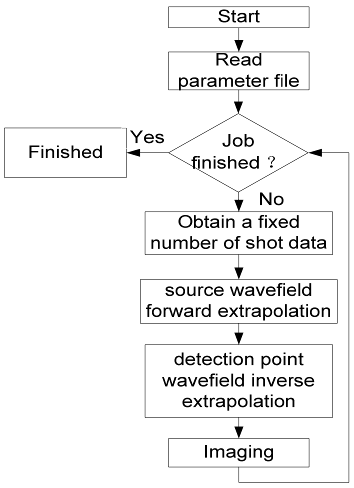
Start
Read parameter file
Job finished ?
Yes
Finished
No
Obtain a fixed number of shot data
source wavefield forward extrapolation
detection point wavefield inverse extrapolation
Imaging

Figure 2. The Polling Algorithm Based Load Balancing Algorithm for Reverse-time Migration

The above algorithm still cannot solve the problem of load balancing very well, so this article designed a genetic algorithm based load balancing algorithm of reverse-time migration. The algorithm considers the different processing capacity of each GPU server type, for different GPU servers Sets in different service matching degree. The algorithm flow chart is as follows:

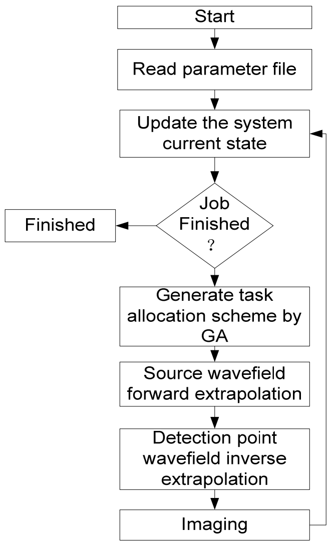
Start
Read parameter file
Update the system current state
Job Finished ?
Finished
Generate task allocation scheme by GA
Source wavefield forward extrapolation
Detection point wavefield inverse extrapolation
Imaging

Figure 3. The Genetic Algorithm Based Load Balancing Algorithm of Reverse-time Migration

3. Algorithm Design for the Genetic Algorithm Based Load Balancing Algorithm of Reverse-time Migration

In the process of solving practical problems, genetic algorithms cannot deal directly with the data in the problem space, it can handle only data expressed in the form of genome chromosome, and therefore to use genetic algorithm to solve the problem, first of all is converting the solution of problem into the organization form of chromosome, namely coding [16-19].

3.1. Encoding Mechanism

Each task in the form of a "n (l) p" description, respectively task number, and the size of the task, task matching degree. The number of shot data is used to describe the size of the task, according to the type of server setups task matching degree, due to processing capacity different type of task of GPU server matching degree varies. In the algorithms, each gene represents a task to be distributed; a group to be assigned tasks that make up a chromosome, each chromosome represents a scheme. So coding problem is the primary problem occurring in the use of genetic algorithm. The algorithm uses the decimal encoding, such as a cluster of N servers, including GPU S2090 servers, GPU S1070 servers and GPU k10 servers. Due to the different processing capacity, different server types have different service matching degree, service matching degree is described by using p, If a task named M, each task information < n, l, p > information contains three information, n for task number, l for task size, and p on behalf of the task matching degree. Obviously, n, l and p element are decimal number, a chromosome encoded as shown:

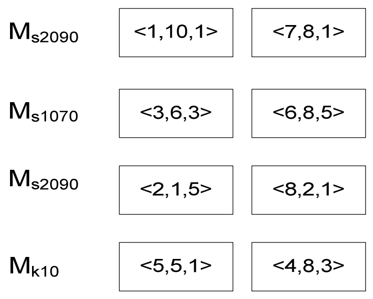

Figure 4. Chromosome Encoding

3.2. Fitness Function

Server's status is the important factors influencing the load balancing. First, assume the total time for each server completes tasks as T_{sum} , its maximum as $T_{\max}$, the minimum value as $T_{\min}$,average as $\overline{T}$, the minimum difference of Δs ,The smaller of Δs, means that the task allocation more balanced; Secondly, the load error rate reflects the overall distribution of the load of GPU servers, the higher utilization of GPU server's and smaller load error rate, the better performance of the server, a server fitness function for f ,set the current server's load for CL_i, new load for NL_i, server's utilization rate of GPU for GP_i, with mean $\overline{GP}$, there are:

$$T_i = CL_i + NL_i = \frac{(\sum_{j=0}^{N_1} n_p \cdot l + \sum_{k=0}^{N_2} n_q \cdot l)}{p} \tag{1}$$

$$\overline{T} = \frac{\sum_{i=0}^{N} T_i}{N} \tag{2}$$

$$\Delta s = T_{\max} - T_{\min} \tag{3}$$

$$GP_i = \frac{T_i}{T_{\max}} \tag{4}$$

$$\overline{GP} = \frac{\sum_{i=0}^{N} GP_i}{N} \tag{5}$$

The fitness function of algorithm as follow:

$$f = (1 - \frac{\Delta s}{\overline{T}}) \times (1 - \sqrt{\left(GP_i - \overline{GP}\right)^2})_i \tag{6}$$

3.3. Genetic Operations

(1) Select operation. In order to make the individuals with larger fitness degree can be directly retained to the next generation of group, select " determine the type of sampling to choose" method, specific steps to see literature [3],setting threshold, keep the server which GP_i is bigger than the threshold directly to the next generation of groups.

(2) The crossover operation. For convenient of crossover operation and mutation operation, so the above chromosomes recombine, combined into one dimensional encoded string. As shown in figure 5 is one-dimensional coding genes used in the algorithm.

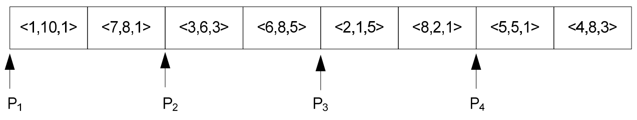

Figure 5. One-dimensional Coding Genes of Chromosomes

Select two parent individuals from a population, make crossover operation, and meet that intersection has the same gene location. Randomly exchange genes which are on the left of the intersection and are not the same, after that to get two new individual chromosomes. When a crossover operation is completed, the change of gene location will lead to the change of task allocation, so the p_i values should change correspondingly.

(3) Mutation. Due to the state of the server is a real-time change, so values of p_i will also change. When a server failure, p_i to 0; If the server replacement values of p_i is also changing. Accordingly, after the completion of a mutation, values of p_i also need to be modified.

4.Experiment and Analysis

4.1. Test Case

Article takes an actual reverse-time migration task as the example; equipment includes 24 servers of S2090 GPU, 24 servers of GPU K10, and 12 servers of S1070 GPU, operation parameters as follows: Contrast diagram of the tasks total time as follows, through the actual test result, we can see that the effect of the algorithm obviously is better than the other two kinds of algorithms, especially for the presence of server failure, the load-balance performance of algorithm effect is better, and with the increase of Work Load, the effect of the algorithm will be more obvious.

Table 1. Parameter List of Reverse-time Migration Task

Area（Km^2）	300
Data amount (Gb)	487
Sampling interval（ms）	2
FLOD	120
Trace length（ms）	6000
Bin size（m）	25x25
Shot number	29328
Continuation depth	5000
Time continuation	0.4ms
Main frequency of wavelet	F=20

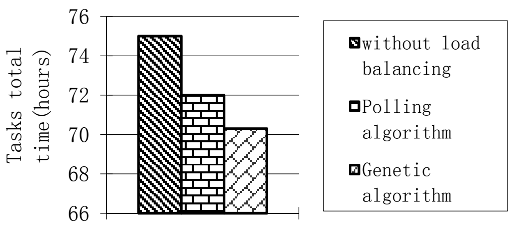

Figure 6. Comparison Chart of Algorithm's Effect in Trouble-Free Condition

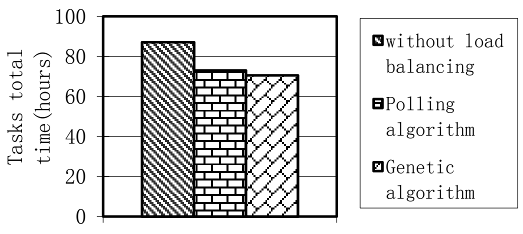

Figure 7. Comparison chart of Algorithm'S Effect with One Server Failure for 12 Hours

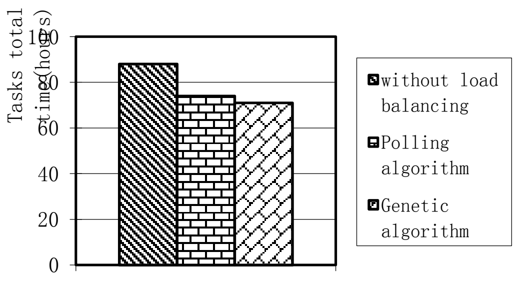

Figure 8. Comparison Chart of Algorithm'S Effect with Two Servers Failure for 12 Hours

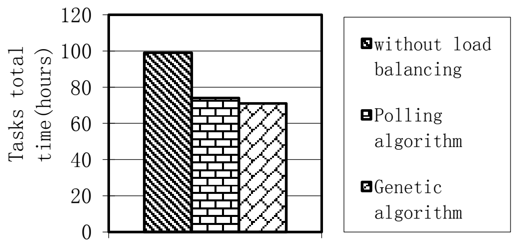

Figure 9. Comparison Chart of Algorithm'S Effect with One Server Failure for 24 Hours

4.2. Performance Comparison

Relative to the other two algorithms, genetic algorithm based load balancing algorithm of reverse-time migration can have better running effect, the algorithm can generate reasonable distribution scheme by server load state and the type of server; The polling algorithm just assigned a fixed number of task, and can't adjust according to the server status, so that it's effect of load balancing is not as good as genetic algorithm; without of load balancing algorithm, seismic data is distributed in one-time, and the algorithm completely does not consider the server state.When server failure, the genetic algorithm based algorithm can have the effect of load-balance, and compared with the polling algorithm, it has better effect of load-balance, and as for the algorithm without load balancing, once a server failure, all the other servers must wait until the server is normal.

5. Conclusion

The algorithm fully considered the type of server, server load state, and the average processing time of each server, this algorithm can distribute tasks according to the status of different servers, make full use of the servers, and avoid the influence of server failure, so it has a good application effect.

References

[1] Zhang Wenxiu, Liang Yi. The mathematical basis of genetic algorithm. Xi 'an: xi 'an jiao tong university press. 2000.
[2] Dai Wenhua. Research on text classification and servering based on genetic algorithm. Beijing: science press. 2008.
[3] Yang Ping, Zheng Jinhua. Comparison of genetic selection operator and study. *Computer engineering and application*. 2007; 43(15): 59-62.
[4] Zhu Li, Shen Weiming, Pan Shaoming, et al. *A Dynamic Load Balancing Method for Spatial Data Network Service.* The 5th International Conference on Wireless Communications. Networking and Mobile Computing. Beijing. 2009.

[5] Pico CAG, Wainwright RL. Dynamic Scheduling of Computer Tasks Using Genetic Algorithms. *IEEE World Congress Computational Intelligence.* 1994; (2): 829-833.
[6] Zomaya AY, Yee-Hwei T. Observations on Using Genetic Algorithms for Dynamic Load-Balancing. *Parallel and Distributed Systems.* 2001; 12(9): 899-911.
[7] GB Zheng. Achieving High Performance on Extremely Large Parallel Machines: Performance Prediction and Load Balancing. Urbana: UIUC. 2005
[8] JM Bahi, et al. Dynamic load balancing and efficient load estimators for asynchronous iterative algorithms. *IEEE T Parall Distrib.* 2005; 16(4): 289-299.
[9] YIN Wen, YIN Xing-yao, ZHANG Fan-chang. A study on seismic attribute optimization based on parallel genetic algorithm. *Journal of Jilin University (Earth Science Edition).* 2005; 35(5): 672-676.
[10] Yu Lei, Lin Zong-Kai, Guo Yu-Chai, Lin Shuo-Xun. Load balancing and fault-tolerant services in multi-server system. *Journal of System Simulation*. 2001; 13(3): 325-328 (in Chinese)
[11] Li Shuang-Qing, Gu Ping, Cheng Dai-Jie. Analysis and research on load balancing strategy in Web server system. *Journal of Computer Engineering and Application*. 2002; (19): 40-43 (in chinese)
[12] Liu Min, Wu Cheng, Yin Wenjun. Solving identical parallel machine production line scheduling problem with special procedure constraint by genetic algorithm. *Acta Automatica Sinica*. 2001; 27(3): 381-386 (in Chinese)
[13] Liu Min, Hao Jinghua, Wu Cheng. A genetic algorithm for parallel machine scheduling problems with procedure constraint and its applications. *Chinese Journal of Electronics.* 2006; 15(3): 463-466.
[14] Ramamritham KJ, Stankovic A, Shiah PF. Efficient scheduling algorithms for real-time multiprocessor systems. *IEEE Trans on Parallel and Distributed Systems*. 1990; 1(2): 184-194.
[15] Manimaran G, Murthy CSR. An efficient dynamic scheduling algorithm for multiprocessor real-time systems. *IEEE Trans on Parallel and Distributed Systems.* 1998; 9(3): 312-319.
[16] Xuesong Yan, Weighted K-Nearest Neighbor Classification Algorithm Based on Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(10): 6173-6178.
[17] Yanbei Li, Lei Yan, Hua Qian. A Gait Recognition System using GA-based C-SVC and Plantar Pressure. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(10): 6135-6142.
[18] YU Zhijun. RBF Neural Networks Optimization Algorithm and Application on Tax Forecasting. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(7): 3491-3497.
[19] Jingsong Yang. A personification heuristic Genetic Algorithm for Digital Microfluidics-based Biochips Placement. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(6): 3187-3193.