High-performance computer system based on CPU/GPU isomeric architecture parallel algorithm

Hanyang Jiang
Hengyang Normal University, Hengyang, Hunan, 421002 China.
jianghanyangjhy@163.com

Journal of Digital Information Management

***ABSTRACT**: With the rapid development of computer, central processing unit + graphics processing unit (CPU+GPU) isomeric system is becoming more and more common, which has strong computing power and can meet the requirement of application on highly intensive computation. On the basis of CPU/GPU isomeric architecture, the parallel algorithm based high-performance computer system was deeply analyzed in this study. Main contents included three aspects. Firstly, programming models of three kinds of parallel technologies of CPU/GPU isomeric system were introduced and analyzed, and a hybrid parallel optimizing strategy was put forward. Secondly, the high-performance computer system model based on CPU/GPU isomeric architecture parallel algorithm was constructed and message passing interface + Open Multi-Processing/compute unified device architecture (MPI+OpenMP/CUDA) model was established for further analysis. Finally, RRTM long-wave radiation program was transplanted to the large-scale CPU/GPU isomeric high-performance computer system. Results indicated that, under the premise that long-wave radiation transmitting procedure simulated computational accuracy, accelerated computing of CPU could significantly improve the computational efficiency of computer.*

Subject Categories and Descriptors
I.3.2 [Graphics Systems]: I.3.1 Parallel processing; **B.7.1 [Algorithms Types and Design Styles]**: Algorithms implemented in hardware

General Terms: High Performance System, Parallel Technologies

Keywords: Hazard analysis and critical control point; Workflow technology; Critical control point; Food safety; Workflow model

Received: 8 January 2016, Revised 7 February 2016, Accepted 19 February 2016

1. Introduction

In recent years, with the development of technology and progress of times, high-performance computer system becomes more and more popularized and common, and demands of people on computation and storage capacity are increasing day by day; therefore, it is an inevitable trend to use large-scale parallel computers for parallel computation [1]. With the improvement of floating point arithmetic performance of graphics processing unit (GPU), CPU/GPU isomeric architecture is usually adopted to improve the computing performance of large-scale parallel computers. Central processing unit (CPU) is mainly responsible for serial computation of complex logic and transaction processing, while GPU is responsible for a large amount of parallel computation. Such kind of system construction method takes full advantage of CPU and GPU and is convenient for development of different levels of parallelism [2-3]. Because CPU/GPU isomeric high-performance computer system has unique program execution method and performance features, a new kind of parallel computational model should be constructed to provide more effective technical support for construction of such kind of platform [4-5].

High-performance computer system based on CPU/GPU isomeric architecture parallel algorithm was analyzed and discussed in this study, which provided powerful theoretical and technical support for improvement of computational efficiency of computers [6-7]. Transistor resource can be applied to integration of more computing cores by GPU, thus to increase the integral throughput capacity. Compared with other common programming models, the

message passing interface + Open Multi-Processing/ compute unified device architecture (MPI+OpenMP/CUDA) model constructed by the large-scale CPU/GPU isomeric high-performance computer system can develop the enormous computing power of CPU/GPU isomeric architecture better [8-9]. Transplanting compute-intensive code segment to GPU and reconstructing application program facing GPU architecture [10-11] are two main methods of developing large-scale scientific and engineering computing applications based on GPU platform. Parallel computational models were put forward in this study; main operation center (MOC) and collaborative awareness extensible model were adopted and their correctness and effectiveness were verified.

2. Parallel Programming Model

2.1 Data parallel model

Data parallel model (CUDA) is carried out under CPU/GPU isomeric architecture and its core kernel function is carried out by GPU; serial part of the program is carried out by CPU. Suppose in data parallel model program, both main engine and equipment have independent dynamic random access memory (DRAM), then the program can manage the memory space by calling CUDA runtime application program interface (API) or CUDA driver API.

In data parallel model, High Performance Fortran (HPF) is the most important data parallel language; it has characteristics like array can achieve global access, program has single control flow, degree of parallelism comes from distributed data structure and implicit expression of communication [12]. Although HPF language is of great significance to program debugging, it is not successfully applied by far due to underdeveloped compilation technology and imperfect standards, etc.

2.2 Message passing model

Message passing refers to users explicitly send and receive messages to achieve data exchange, pace coordination and control execution between processors. The message passing program has following characteristics: the parallel program is composed of various processes and each process can execute different codes respectively; each progress executes asynchronously and methods like roadblock or communication block are used for synchronous execution; each progress stays in different address spaces and the interaction of its relevant data variables between progresses relies on message passing; users have to explicitly describe data mapping, load balancing, synchronization and other interactive operation in the parallel program [13]. Message passing interface (MPI) is the standard specification of message passing function library and its frame is shown in figure 1.

MPI supports blocking and non-blocking data transmitting mechanisms; it defines abundant function interfaces as well as binds Fortran and other high-level languages, thus users can realize MPI parallel program by calling these function interfaces.

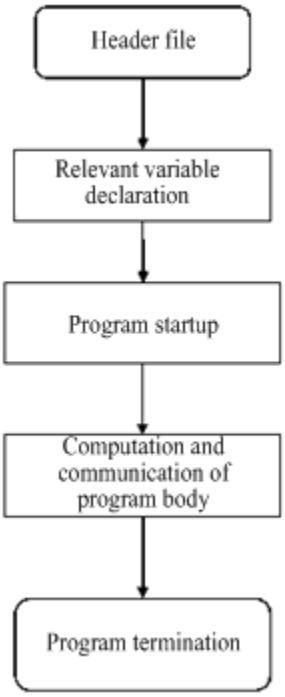

Figure 1. Program frame of MPI

2.3 Genetic algorithm

Open Multi-Processing is the most mainstream parallel design model based on shared memory multiple processor system at present, which is characterized by simple programming, powerful portability and good readability, etc.; the disadvantages of openMP is that it can not be applied to distributed memory system and its expansibility is poor.

Compiling execution of OpenMP program is very convenient so that the number of threads of openMP program startup can be controlled by setting environment variable OMP_NUM_THREADS, then openMP program can be executed directly. OpenMP base provides abundant API functions and users can control the parallel region through these library bases; moreover, openMP defines the environment variable that controls openMP program to some extent.

2.4 Optimizing strategy of hybrid parallel

(1) Study of communication optimization of multi-core cluster. In multi-core nodes constructed cluster, interior communication of nodes is greatly different from the communication among nodes; thus hierarchical communication strategy is adopted to divide communication into intra-node communication and inter-node communication. In data communication, data that need communication in nodes are collected and integrated and then communicate with the first progress of other nodes; after inter-node communication, first progress transfers data through intra-node communication.

(2) Study of communication optimization of GPU cluster. CUDA-Aware MPI can optimize data communication on GPU among different nodes, which avoids the

participation of CPU.

3. High Performance Computer System Model

3.1 Definition of computer system model

Latency overhead gap Processor (LogP) model can evaluate the practical performance of parallel computer; four parameters (L, o, g, P) accurately express the most important performance parameters of parallel computer. LogGP model [14] is generated by extension of LogP model; linear long message mechanism is introduced in communication model, which not only exerts the efficiency of parallel machine, but also makes the actual results more closer to design expectation. While sending long messages, an important point that affects the communication performance of LogGP model is the neglect of synchronization. Long messages are given a length boundary value to make the MPI program more accurate and efficient in cluster environment.

HLog(n)GPM model is the limited extension of LogGP, which emphasizes the importance of memory storage, thus there is the letter M in its name; H refers to that the model is applicable to isomeric high performance computer system; n means the total number of atom communication steps captured by model; L refers to the maximum delay of transmission of single message from source to destination; o is the extra cost of execution of the communication operation by initiator of atom communication; g refers to describing the capacity of processor continuously sending messages; G refers to describing the ability of communication media transmitting long messages, and its reciprocal represents the bandwidth of communication media; P refers to the computing power of processor in the system, which is an important parameter for describing the local computing.

3.2 Application of computer system model

3.2.1Communication operations from point to point

For atom communication i captured by HLog(n)GPM model, suppose its message length has m_i threshold values, then the communication bandwidth is divided into m_i+1 intervals; if every atom communication operation is described by LogGP model, then the point-to-point communication time on GPU cluster can be defined by the following theory.

Suppose the point-to-point communication operation (message length is k) on GPU cluster contains n atom communication, then the predicted communication time based on the model is

$$t_l \le t \le t_u$$

$$t_l = \sum_{i=1}^{n} \{2o + (k-1)G_{iu} + L\},$$

$$t_u = \sum_{i=1}^{n} \{2o + (k-1)G_{il} + L\}.$$

Suppose the atom communication i has m_i threshold values, message length k satisfies $k \in [\lambda_j, \lambda_{j+1}]$ and messages are transmitted according to the bandwidth $1/G_{ij}\ (0 \le j \le m_i)$, then the consumed time of the ith atom communication is:

$$t_i = o + L + (k-1)G_{ij} + o \le 2o + L + (k-1)G_{il}$$

(Equation 1)

Thus the time of overall n atom communication is:

$$t = \sum_{i=1}^{n} t_i \le \sum_{i=1}^{n} (2o + L + (k-1)G_{il}) = t_u$$ (Equation 2)

And because

$$t_i = o + L + (k-1)G_{ij} + o \ge 2o + L + (k-1)G_{iu}$$

(Equation 3)

Thus the following equation can be obtained:

$$t = \sum_{i=1}^{n} t_i \ge \sum_{i=1}^{n} (2o + L + (k-1)G_{iu}) = t_l$$

(Equation 4)

Above equations indicate that, the performance of point-to-point communication on GPU cluster relies on the effective bandwidth of communication media. Therefore, various kinds of factors that restrict bandwidth should be eliminated to thoroughly explore the bandwidth potential of different kinds of communication media.

3.2.2 Memory copy mechanism

Memory copy is one of the important operation steps commonly used in programming [15]. LogGP model can be used for description during memory copy, in which the parameter G is of vital importance and closely related to the size of transmission data. G can be defined as a piecewise function G=f(s), in which s refers to the transmission quantity of data. Because influencing factors of performance of memory copy are few, performance of memory copy can be predicted accurately and copy mechanism can be grasped by measuring the memory copy bandwidth of partial s and then fitting a bandwidth function G=f(s).

3.3 Construction of MPI+OpenMP/CUDA hybrid parallel model

Figure 2 shows that, in GPU cluster, there are multi-core CPU processor resources except the enormous computing power of GPU, which can be fully developed by MPI-openMP+CUDA three-level collaboration and parallel. Progress of every MPI is serial. Thread level parallel can be realized by opening OpenMP in MPI progress and finally each MPI progress can derive mn_c/n_g open MP threads through adjusting runtime library function. Such

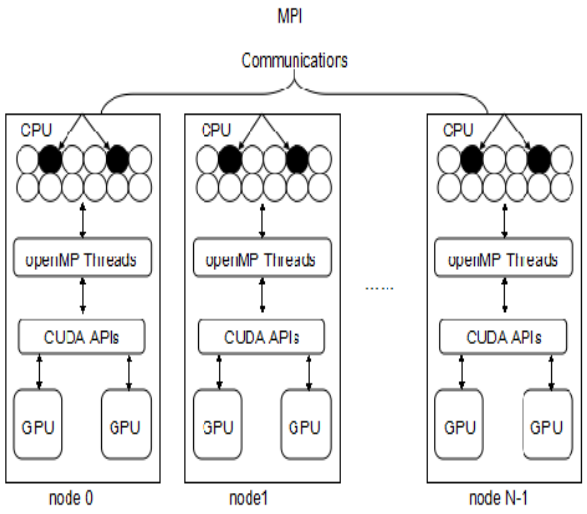

Fig. 2. Sketch map of MPI+OpenMP/CUDA hybrid parallel model

TS/min	CPU			MOC		
	WT/ sec	T2 rmse	GLW rmse	WT/ sec	T2 rmse	GLW rmse
2	5.49	0.04	5.61	2.45	0.04	6.25
4	3.66	0.04	5.47	1.59	0.04	6.20
6	2.38	0.03	5.38	1.05	0.04	6.08
8	1.62	0.03	3.61	0.71	0.02	4.44
10	1.33	-	-	0.56	0.01	1.56

Table. 1. Performance of RRTM at different time steps

model can parallelly process relevant computing tasks by calling CUDA API drive GPU equipment, and each piece of GPU equipment only needs to maintain one program context to eliminate the redundancy allocation problem of memory space exists in MC mode, which greatly improves the utilization efficiency of memory resources.

4. Realization of Long-Wave Radiation Scheme in Large-Scale High Performance Computer System

4.1 RRTM long-wave radiation transmission scheme

4.1.1 Definition of RRTM scheme

RRTM long-wave radiation transmission scheme is used to calculate the radiation flux and cooling rate in any long-wave spectrum band under clear sky, which can effectively improve the computational efficiency of radiation transfer while still maintains high computational accuracy [16].

4.1.2 Characteristics of RRTM program execution

RRTM program traverses all horizontal grid points by two-layer circulation and a series of calculation is carried out along the vertical direction of any grid point. RRTM program contains five subprograms which are INIRAD, MM5ATM, SETCOEF, GASABS and RTRN. INIRAD is responsible for some initialized tasks; MM5ATM is responsible for reading atmospheric outlines, such as temperature and intensity of pressure, etc.; SETCOEF is responsible for calculation of indexes and proportions related to intensity of pressure and temperature difference of a specific atmospheric layer; GASABS is responsible for calculation of optical thickness of 16 long-wave spectral bands; RTRN is the core module of the whole radiation transmission scheme, which is responsible for calculation of up/down flux and heat rate of any cloudless atmosphere.

4.2 Computational efficiency of RRTM hybrid program

Radiation transmission scheme is always one of the most time-consuming physical process parameterization schemes in numerical weather prediction system, and its computing time depends on horizontal resolution, vertical layers and times of calling during radiation transmission process. If radiation transmission at every time step is calculated, the computing time of the whole numerical weather prediction system can be improved significantly [17].

Table 1 shows the performance of RRTM at different time steps; TS represents corresponding time steps of radiation transmission calculation; WT refers to the wall clock time needed by program run; T2 rmse refers to the mean square error of temperature 2 meters off the ground; GLW rmse means the mean square error of terrestrial radiation flux. Table 1 shows that, during the increasing process of radiation transmission time step from 2 min to 10 min, execution time of CPU and MOC is decreasing gradually; moreover, the execution time of MOC is half of that of CPU. Therefore, MOC hybrid RRTM is very sensitive to time resolution of long-wave radiation calculation, which can effectively improve the calculating efficiency of long-wave radiation process simulation and significant shorten the calculating time.

4.3 Strong extendibility analysis of RRTM hybrid program

Table 2 shows the performance of CPU and MOC edition RRTM running in TH-1A subsystem composed of 1024 nodes. Except using CPU computing resource through openMP, MOC edition uses computing power of GPU in virtue of CUDA. With the increase of computational nodes, computing time of CPU edition and MOC edition shows decreasing tendency and the speed-up ratio maintains at around 2, indicating that they have good strong extendibility. Therefore, in consideration of unique architecture characteristics of GPU, it should have sufficient computing quantity to guarantee its powerful computational advantage to CPU.

Computational nodes	RRTM run time		Speed-up ratio
	CPU	MOC	
256	7.15	3.01	2.38
384	5.77	2.60	2.22
512	3.82	1.71	2.24
640	3.27	1.54	2.13
768	2.89	1.47	1.97
892	2.54	1.25	2.04
1024	2.03	0.98	2.08

Table 2. Run time of RRTM in different system scales

5. Discussion and Conclustion

With the improvement of technology and progress of times, high performance computer system based on CPU/GPU isomeric architecture parallel algorithm has become a very important high performance computing platform. By virtue of its powerful floating-point calculation capacity, it is widely applied to calculation of computational fluid mechanics related fields [18].

High performance computer system based on CPU/GPU isomeric architecture parallel algorithm was analyzed and discussed in this study. First of all, three parallel layers which were MPI, openMP and CUDA were basically introduced; optimization methods of each parallel layer were emphatically introduced and optimization strategies under hybrid parallel were studied to lay a technological basis for development and optimization of parallel algorithms. Secondly, construction of high performance computer system model was put forward and parallel programming model MPI+OpenMP/CUDA was constructed; on the basis of large-scale GPU cluster, a hybrid model MOC based on current programming model was constructed [19-20]. In memory copy mechanism, compute-intensive application could usually acquire great speed-up ratio on GPU; however, if most computing load was transplanted onto GPU, MOC hybrid programming model is recommended for calculation and each node only uses one MPI progress. Finally, RRTM hybrid parallel algorithm was designed and long-wave radiation scheme program was transplanted to the large-scale CPU/GPU isomeric high performance computer system, thus to provide effective suggestions for better application of CPU/GPU isomeric system [21]. Numerical experiment indicated that, under the premise of maintaining computing accuracy, GPU accelerated computing could double the computing efficiency of long-wave radiation transmission simulation; in addition, it was also verified that RRTM isomeric hybrid program had good strong extendibility and collaborative awareness extendibility [22-23].

In conclusion, high performance computer system based on CPU/GPU isomeric architecture parallel algorithm has good application prospect and GPU accelerated computing can significantly improve the computing efficiency of computer, which has good acceleration effect. Although the computing model is based on CPU/GPU isomeric architecture, coordination also exists in many other structures, thus collaborative awareness extendibility model should be further improved.

References

[1] Rajamony, R. Arimilli, L. B. Gildea, K.. (2011). PERCS: The IBM POWER7-IH high-performance computing system, *IBM J of Research & Development*, 55 (3) 233-244.

[2] Campa, S., Danelutto, M., Goli, M. (2014). Parallel patterns for heterogeneous CPU/GPU architectures: Structured parallelism from cluster to cloud. *Future Generation Computer Systems*, 37. 354-366.

[3] Richmond, P., Walker, D., Coakley, S. (2010). High performance cellular level agent-based simulation with FLAME for the GPU. *Briefings in Bioinformatics*, 11 (3) 334-47.

[4] Lu, F. , Song, J., Cao, X. (2012). CPU/GPU computing for long-wave radiation physics on large GPU clusters. *Computers & Geosciences*, 41 (2) 47-55.

[5] Pajot, A., Barthe, L., Paulin, M.(2011). Combinatorial Bidirectional Path-Tracing for Efficient Hybrid CPU/GPU Rendering. *Computer Graphics Forum,* 30 (2) 315–324.

[6] Ray, D., Aswal. V. K (2015). A Parallel Computing Model of Agent enabled Mining of Globally Strong Association Rules. *Journal of Nanoparticle Research,* 5 (4) 17-30.

[7] Liao, X. , Yuan, Z., Hu, P. (2014). GPU-assisted energy asynchronous diffusion parallel computing model for soft tissue deformation simulation. *Transactions of the Society for Modeling & Simulation International*, 90 (11) 1199-1208.

[8] Ayres, D. L., Darling, A., Zwickl, D. J.(2012). BEAGLE: An Application Programming Interface and High-Performance Computing Library for Statistical Phylogenetics. *Systematic Biology,* 61(1) 170-173.

[9] Samadi, M., Hormati, A., Lee, J (2014). Leveraging GPUs using cooperative loop speculation. *ACM Transactions on Architecture & Code Optimization,* 11 (1) 13-15.

[10] Bubak, M., Funika, W., Wismüller. R (2015). Visualization-Based Active Learning for the Annotation of SAR Images. *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, 23 (2) 1-1.

[11] Frazão-Teixeira, E., de Oliveira, F. C., Fiuza, V. R. (2011). Compromised nutrition in gerbils infected by Cystoisospora felis detected through an animal performance

analysis tool. Revista brasileira de parasitologia veterinaria, *Brazilian Journal of Veterinary Parasitology:* Orgao Oficial do Colegio Brasileiro de Parasitologia Veterinaria, 20 (3) 242-5.

[12] Wu, S. Sheth, A.Miller, J. (2010). Authorization And Access Control Of Application Data In Worlflow Systems. *J. Intell. Inf.Syst*, 18 (1) 71—94.

[13] Komatitsch, D. D. , Erlebacher, G., Göddeke, D. (2010.) High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster. *Journal of Computational Physics,* 229 (20) 7692-7714.

[14] Zhu, Jun., Lastovetsky, Alexey., Ali, Shoukat. (2013).Communication Models for Resource Constrained Hierarchical Ethernet Networks. Lecture Notes in Computer Science, 8374. 259-269.

[15] Fang, Y. Q., Song, Z. C., Ge, J. W. (2014) Cloud Computing-Oriented Virtual Machine Live Migration Mechanism. *Applied Mechanics & Materials*, 513-517: 1731-1734.

[16] Lábó, E., Geresdi, I. (2016). Study of longwave radiative transfer in stratocumulus clouds by using bin optical properties and bin microphysics scheme. *Atmospheric Research,* 167. 61-76.

[17] Yang, P., Liou, K. N., Bi, L (2015). On the radiative properties of ice clouds: Light scattering, remote sensing, and radiation parameterization. *Advances in Atmospheric Sciences,* 32 (1) 32-63.

[18] Józsa, C. M., Domene, F., Vidal, A. M. (2014). High performance lattice reduction on heterogeneous computing platform. *Journal of Supercomputing*, 70 (2) 1-14.

[19] Salzwedel, A., Wegscheider, K., Herich, L.(2014). Impact of clinical and sociodemographic patient characteristics on the outcome of cardiac rehabilitation in older patients. Aging Clinical & Experimental Research, 27 (3) 315-321.

[20] Thiery, W., Martynov, A., Darchambeau, F (2014). Understanding the performance of the FLake model over two African Great Lakes. Geoscientific Model Development, 7 (1) 317-337.

[21] Playne, D. P.,Hawick, K. A (2012). Comparison of GPU architectures for asynchronous communication with finite-differencing applications. *Concurrency & Computation Practice & Experience,* 24 (1) 73–83.

[22] Mielikainen, J., Huang, B., Huang, H. L.(2013). A GPU acceleration experience with RRTMG long wave radiation model. Proceedings of SPIE - The International Society for Optical Engineering, 8895 (2) 206-209.

[23] Jian, H. W. Lia, G. Y.. Sheng, L (2010). Rapid Parallel Calculation of shell Element Based On GPU. 1252 (1) 755-761.