Local Network Parallelization Computing System for Computer Game Programs

Shi-Jim Yen[†1] Tsan-Cheng Su [†2]
Shun-Chin Hsu [†3] Jr-Chang Chen[†4]

Abstract: This paper proposed the Local Network Parallelization Computing (LNPC) system, which is suitable to excuse computer games in parallel. The LNPC system is a job-level system. The experimental results show that the LNPC system can decrease the communication time of the computer game program and increase the performance. It is useful in solving specific games and computer game competitions.

Keywords: Local Network, Computer Game Program, Parallelization.

1. Introduction

Development of computer game programs and game solvers are popular research topics. [1][2][5][6] The development is often needed to parallelize the program on many computing units. When the system is run on a mass of computer connected by a network, the network latency will affect the performance. This paper proposed a method to reduce the network latency information. This method is novel and efficient for computer game applications.

2. Architecture of the LNPC system

The LNPC system contains special network setting. The Maximum Transmission Unit (MTU) of every interface is set to 9K, and each computing unit must be installed with a Gigabit Ethernet network interface card. The computing unit master is also installed with a Gigabit Ethernet network interface card, and it must has other one Gigabit Ethernet network interface card to computing unit of the user program. The Gigabit Ethernet switch communications between all computing units and computing unit master on a private Gigabit Ethernet LAN.

Computer game programs usually used standard Transmission Control Protocol (TCP) communication for communication in Internet. These papers [2][5] used TCP protocol to build a TCP parallel computing system. The protocol of the TCP computing system is as in Figure 3. Note that the TCP communication in private LAN is inefficient.

The architecture of the LNPC system is presented in this section. The whole architecture is divided into three parts: (1) the User Program (2) the Computing unit master (3) the computing unit. Figure 1 and Figure 2 show how the LNPC system works.

We divide the game programs into two parts: the User Program and the computing unit. The computing unit is a bridge between these two parts. There are six steps per round in a solved:

(1) The User program encodes the game state and sends it to the computing unit master;
(2) The computing unit master dispatches jobs from the User program to computing units;
(3) The computing unit receives, decodes, and executes the task;
(4) after finishing the task, the computing

†1 Department of Computer Science & Information Engineering, National Dong Hwa University, Hualien, Taiwan. sjyen@mail.ndhu.edu.tw
†2 Department of Computer Science & Information Engineering, National Dong Hwa University, Hualien, Taiwan. d9821009@ems.ndhu.edu.tw
†3 Department of Information Management, Chang Jung Christian University, Tainan, Taiwan.
schsu@mail.cjcu.edu.tw
†4 Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan, Taiwan.
jcchen@cycu.edu.tw

unit encodes the results and sends them to the computing unit master;

(5) The computing unit master passes the results from the computing units to the User program;

(6) The User program receives, decodes, and handles the results, as Figure 5 shows.

At User program and Computing unit, we used encode at a simulation status to transmission. The method can add checksum to check the solved state is correct. At user program also add some information in code to send. At receive decode can also be sure that the integrity of the received data. In the LNPC system used the method of encode and decode can check the state surefire.

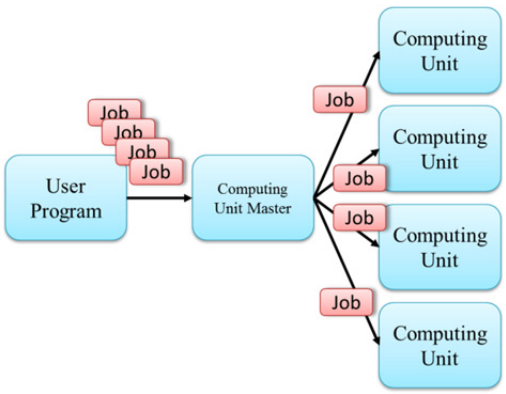

Figure 1. Distribute jobs in the LNPC system

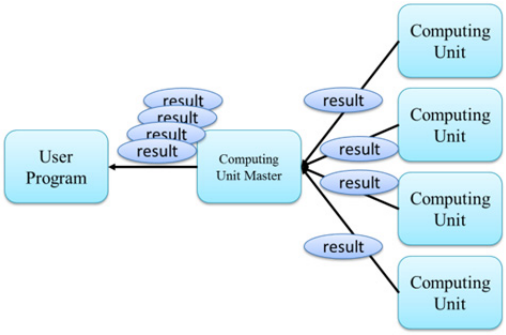

Figure 2. Integrate results in the LNPC system

The communication protocol of the LNPC system is described in Figure 4. For a successful request, it needs 8 operations to communicate information among the user program, the computing master unit, and the computing unit. If there are many jobs in the user program, we should make one operation, which is to repeat Step 3. Figure 4 presents the protocol of the LNPC system.

In this paper, the experiment of MTU size is set to 9K, and used the same LNPC system of computing units. Table 1 compares our method and the standard TCP computing system.

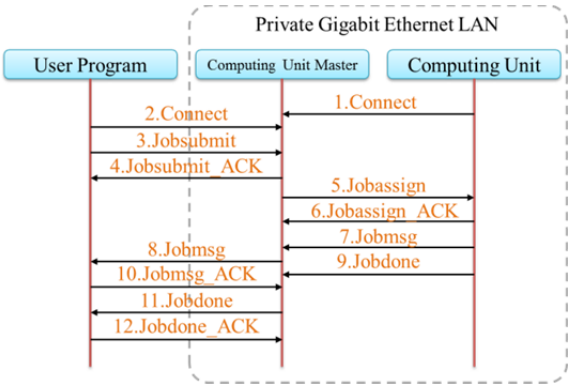

Figure 3. Protocol of standard TCP communication system

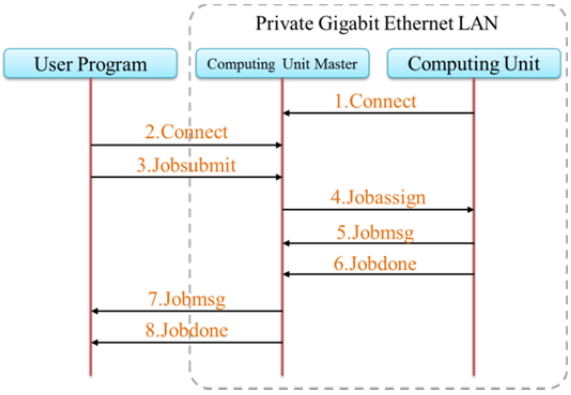

Figure 4. Protocol of the LNPC system

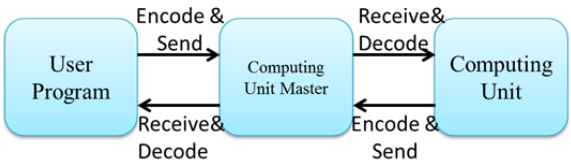

Figure 5.The Model for Paralleling Games

3. Experimental and Discussions

3.1 Experimental Setup

The experiments were performed on several computing units which have the same specification. Each computing unit has an AMD Bulldozer FX-8150 CPU operated at 4.5GHz and has 16 Gigabytes of main memory. These computing units are running Microsoft windows 7 SP1. We selected Othello Solver as the experimenter. At computing unit master has an Intel Q9550 CPU operated at 3.3GHz and has 8 Gigabytes of main memory. The computing unit master is running Microsoft Windows 2008 enterprise R2.

These computing units and computer unit master are used Marvell gigabit Ethernet controller. And we used gigabit switch in the communication between computing units and computing unit master.

The all computing units, computing unit master and user program of computer should be set to the network card of MTU size set at 9K, we also checked the computing units and computing unit master must in the same private local network LAN.

Table 1. Comparison of our method and the TCP computing system

	Packet type	**MTU size**	**latency**	**Packet header**
Our method	UDP	9K	few	short
Normal commun-ication	TCP	1500	larger	long

3.2 Result and Discussions

The subject program is a 7x7 Othello solver. The solver sends 10000 Othello board states to the LNPC system, and then the system generates all possible legal moves of each board state, and returns the result to the solver. Table 2 shows that our method is more efficient than the method of the standard TCP communication. When the number of cores is 4, the LNPC can speed up 200%. When the number of cores is 16, then the LNPC system performance decreases due to the limitation of bandwidth.

Table 2. Compare with our method and standard TCP communication

Number of cores	**1 core**	**4 cores**	**16 cores**
LNPC system	80408~ 81510ms	10251~ 22414ms	1521~ 5212ms
standard TCP communi-cation	81875~ 82151ms	12213~ 48478ms	1595~ 8870ms

Reference

1) T. Cazenave and N. Jouandeau, “On the parallelization of UCT,” in Proceedings of the Computer Games workshop2007 (CGW 2007), ser. MICC Technical Report Series, J. H.van den Herik, J. Uiterwijk, M. Winands, and M. Schadd,Eds., no. 07-06. Universiteit Maastricht, June 2007, pp.93–101.
2) C. H. Chen, S. S. Lin and M. H. Huang, “Volunteer Computing System Applied to Computer Games” TCGA workshop 2012 (TCGA2012), June 2012, Hualien, Taiwan, pp.25-28.
3) BOINC, available at http://boinc.berkeley.edu/
4) XtremWeb, available at http://www.xtremweb.net/.
5) H. Y. Liou, I. C. Wu, H. H. Kang, J. H. Guo and T. F. Liao, “General framework development on board game” TCGA workshop 2012 (TCGA2012), June 2012, Hualien, Taiwan, pp.19-24.

6) H. Kato and I. Takeuchi, “Parallel monte-carlo tree search with simulation servers”, The 13th Game Programming Workshop (GPW 2008), November 2008, Hakone, Japan.

Acknowledgments

This work is supported by the National Science Council of Taiwan under the grant NSC 99-2221-E-259-009-MY3.