International Conference on Applied Mathematics, Simulation and Modelling (AMSM 2016)

Application of Dynamic Allocation Task in Multi Processor Platform

Zhiwen Yang and Pingzhen Zhou
Jiangxi institute of fashion technology, China

Abstract—**Traditional multi GPU processor cluster task scheduling and allocation method uses the time gap allocation method does not consider each task execution requirements and each computer node of actual operation ability, the overall efficiency of the implementation. Proposed a hybrid scheduling multi GPU processor platform dynamic task allocation method, using multi processor simulation of multiple computing nodes in a computer system, for different tasks, task computation demand and allocation according to the computer's actual operation ability, the hybrid task scheduling, the system has good robustness and experimental tests were conducted using a set of practical tasks. The results show that using mixed task, and realize the dynamic task allocation method, task execution time is greatly reduced, with good efficiency improvement results.**

Keywords-dynamic task allocation; multi processor cluster; hybrid scheduling

I. INTRODUCTION

At present, the rapid development of computer technology, we in the operation of the computer also to computer operation speed and level also put forward higher requirements, such as 3dsmax graphics rendering, real-time task scheduling, computer system has very strong operation ability. Traditional GPU computing mode in daily documents processing and data operation can very good adaptation, but for the amount of data is very large, the graphics computing, real-time multidimensional operations, the computation on the GPU is [1,2] powerless. Compared to the GPU system, the biggest advantage of the multi processor system lies in its ability to super. Multiprocessor cluster is the guarantee of the efficient operation of multi processor, and it has become the first choice of the modern multi processor processing intensive computer task [3.]

II. MULTIPLE GPU PROCESSOR CLUSTER TASK SCHEDULING

A. *Analysis of Multi GPU Processor Cluster Scheduling Model*

Currently, cluster of multiprocessors scheduling method is divided into two categories. The first is between each processor scheduling can directly communicate with each other. The advantage of this scheduling method is inter processor can achieve information interaction and data communication is very convenient and complete, but due to the lack of unified management mechanism system, so the system is unstable, sometimes, multiple processors also to the same processor communication or data, resulting in a sharp increase in the dynamic load of the processor, resulting in [4] of disorder in the system. The second is mutual independence between each processor and direct communication can not be achieved, the communication between processors must rely on a certain management mechanism to complete, this bottleneck is architecture design of the network structure of the system, due to the system need to be unified scheduling processor to complete the system of management, so for high processor scheduling requirements. On the other hand, if the scheduling processor problems, then the entire system will not work [4]. Computer task scheduling model based on the above analysis is shown in Figure 1.

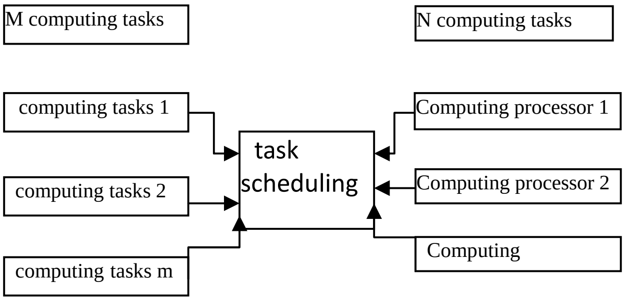

FIGURE I. FLOW CHART OF MULTIPROCESSOR TASK SCHEDULING.

B. *Multi GPU Processor Cluster Scheduling*

The function of multi GPU processor cluster task scheduling is mainly to complete the task scheduling of the whole system, mainly including the following aspects:

(1) Multi processor cluster task scheduling based on the task of computing the intensity of all input system tasks, the task of extracting the appropriate nodes from the computing task to add to the scheduling system.

(2) Multi processor cluster task scheduling based on the computing power of the computer system, the different tasks assigned to different computer nodes.

(3) According to the real-time operation of each task, the task scheduling of multi processor cluster is evaluated according to the real-time operation of each task.

(4) According to the characteristics of each processor, the corresponding data format conversion is carried out according to the characteristics of each processor, which makes the system data compatible, and has a certain error data redundancy processing ability.

III. INTRODUCTION OF DYNAMIC ALLOCATION TASK SCHEDULING

A. *Dynamic Assignment Method for Hybrid Tasks*

Dynamic task allocation method is a hybrid task technology embedded quartz multiprocessor data task allocation scheduling method based on the adaptive ability strong hybrid technology task, the task of data on each processor realize scheduling, in order to achieve the purposes of overall and efficient implementation of the.

A multiprocessor system scheduling task is defined as:

$$\boldsymbol{T}_{n+1} = F(T_n) \tag{1}$$

Among them:

$\boldsymbol{T}_n$ Digital sequence of hybrid scheduling for embedded Quartz;

$F(\bullet)$ The management and scheduling functions of embedded Quartz hybrid scheduling system;

$\boldsymbol{T}_{n+1}$ - the progressive of multiprocessor task scheduling.

The multi processor system scheduling task is represented by a discrete system, and the differential variance expression for scheduling management is:

$$\frac{\partial \boldsymbol{z}(t)}{\partial t \partial (t-1)} = f(\boldsymbol{z}) \tag{2}$$

For a set of actual task scheduling the time series is represented as an output:

$$x_n = f(\boldsymbol{z}_n) + f(\omega_n) \tag{3}$$

Among them: ω_n Embedded Quartz mixed task migration error parameter; x_n Compensation output of hybrid scheduling sequence of embedded Quartz.

Of course, for a continuous task scheduling system, which is:

$$x_n = x(t_0 + n\Delta t) = h[\boldsymbol{z}(t_0 + n\Delta t)] + \omega_n \tag{4}$$

Form in: $h(.)$ Embedded Quartz hybrid scheduling response function; ω_n Hybrid scheduling error of embedded Quartz.

Let M be the D dimension of the task scheduling and distribution system, F is expressed as a vector, with a smooth, for: Φ : $M \to R^{2d+1}$

$$\Phi(z) = (h(z), h(\varphi_1(z)), \cdots, h(\varphi_{2d}(z)))^T \tag{5}$$

When $\omega_n = 0$ the task scheduling sequence is studied, $\{x_n\}_{n=1}^N$ the new mapping is formed by the following vectors in the scheduling task:

$$x_{\mathbf{n}} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} (x_n, x_{n-\tau}, \cdots, x_{n-(m-1)\tau}) \tag{6}$$

Time series for task scheduling: $\{x(t_0 + i\Delta t)\}$, $i = 0,1,\cdots, N-1$, and its management scheduling method is:

$$\boldsymbol{X} = [\boldsymbol{x}(t_0), \boldsymbol{x}(t_0 + \Delta t), \cdots, \boldsymbol{x}(t_0 + (K-1)\Delta t)] \tag{7}$$

$\boldsymbol{x}(t)$ In the above formula,m a set of space state vector in embedded management scheduling Δt is embedded dimension, $K = N-(m-1)J$, τ which is the time interval of sampling,. $\tau = J\Delta t$ 。 Time window:

$$\tau_w = (m-1)\tau = (m-1)J\Delta t \tag{8}$$

Through the above embedded quartz hybrid scheduling method, can very good realization of multiprocessor task scheduling and management, collaborative scheduling, task execution, and ultimately achieve optimization scheduling system to perform output structure.

B. *Multi GPU Processor Cluster Task Scheduling Based on Embedded Quartz Dynamic Allocation Task*

The characteristics of multi processor cluster task scheduling is a huge task, and based on the embedded quartz dynamic task allocation method is characterized in high volume task multiprocessor scheduling performance good, so very suitable to multi processor cluster task scheduling.

Suppose a multiprocessor cluster task scheduling set S, is represented as follows:

$$S = \left(U, A, V, f\right) \tag{9}$$

$$A = P \cup Q \tag{10}$$

$$P \cap Q = f \tag{11}$$

Among them:

U - multiprocessor cluster task scheduling domain;

Non empty set of A - multi processor cluster task scheduling attributes;

P - a set of defining conditions for task scheduling in multiprocessor clusters;

Q - multi processor cluster task scheduling decision set.

To assign a task scheduling function to a multi processor cluster:

$$U \times A \longrightarrow V \tag{12}$$

For the multi processor cluster task scheduling set S, the attribute relation is represented as the limit strip and the multi processor cluster task scheduling:

$$P \subseteq A \tag{13}$$

Can get expression:

$$ind(P) = \left\{ \begin{matrix} (x, y) \in U^2 \mid a(x) = a(y), \\ \forall a \in P \end{matrix} \right\} \tag{14}$$

The classification set of mixed tasks is a kind of equivalent relation, $ind(P)$ Is equivalent to a set of conditions U/P for the conditional expression, U/Q and the decision expression is.

When $X \subseteq U$, $R \subseteq A$ the condition is established, the maximum scheduling solution for the whole embedded Quartz hybrid scheduling problem is the minimum scheduling solution:

$$R_\beta X = U\left\{E \in U/R \mid c(E,X) \le \beta\right\} \tag{15}$$

$$R_\beta X = U\left\{E \in U/R \mid c(E,X) \le 1-\beta\right\} \tag{16}$$

$$bnr_\beta(X) = R_\beta X - R_\beta X_1 \tag{17}$$

Among them, R is a set of mixed scheduling solution for embedded Quartz, which can determine the boundary position of the whole embedded Quartz hybrid scheduling algorithm by the set of Quartz Mixed Scheduling object solution.

Due to the addition of the management factor in the embedded Quartz hybrid scheduling management method, the management factor is:

$$Y(P,Q,\beta) = Y\left[red(P,Q,\beta),Q,\beta\right] \tag{18}$$

By dynamic task allocation technology in the multi processor cluster rendering platform task scheduling, can achieve a better scheduling performance, making the system computing power to maximize the application.

IV. EXPERIMENT AND ANALYSIS

A. *Description of The Experimental Environment*

In order to test the performance of multi GPU processor cluster task scheduling based on embedded Quartz hybrid scheduling, the actual data of multi processor cluster is used to test the data. The detailed properties of the 10 tasks are shown in table 1.

TABLEI. DESCRIPTION OF EXPERIMENTAL ENVIRONMENT

Task number	**Task computing strength**	**Task priority**
1	0.1	0.44
2	0.3	0.51
3	0.57	0.67
4	0.7	0.65
5	0.35	0.56
6	0.53	0.86
7	0.12	0.34
8	0.98	0.57
9	0.67	0.28
10	0.34	0.76

The computing power of the computer 5 processors is shown in table 2.

TABLEII. FIVE COMPUTER PROCESSORS PROPERTIES

Computer number	**Computing power**	**Communication ability**
1	0.8	0.67
2	0.9	0.89
3	0.6	0.45
4	0.7	0.68
5	0.8	0.85

B. *Experimental Results Analysis*

Task scheduling, time window for the task of scheduling and management, scheduling, as shown in figure 2.

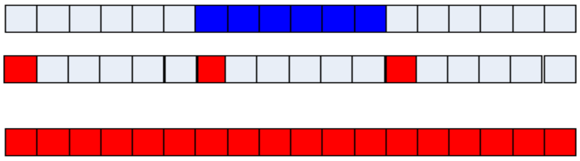

FIGURE II. FIGURE OF TASK SCHEDULING.

After the traditional method is used to schedule tasks, the execution of tasks over time is shown in Figure 3.

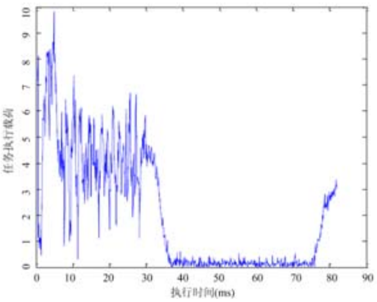

FIGURE III. RESULT WITH TRADITIONAL METHOD

Using the dynamic allocation task of multi GPU processor scheduling method, the task with the time of the implementation of the situation as shown in figure 4.

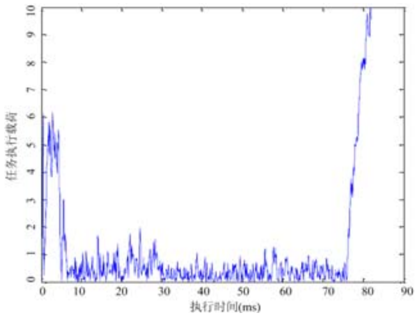

FIGURE IV. RESULT WITH METHOD IN THIS PAPER

From Figure 3, we can see that and Figure 4 can be seen that performs the same task, traditional method used 36ms completes execution, and the methods only 8ms execution completed, so for the same task scheduling, based on embedded quartz mixed task multi GPU processor scheduling technology has greatly improved.

V. CONCLUDING REMARKS

A design method of dynamic allocation processor platform task is studied. In task scheduling and allocation by multi processor simulation of multiple computing nodes in a computer system, for different tasks, task computation demand and allocation according to the computer's actual operation ability, to achieve dynamic task allocation. Finally, using the actual task to test the experiment, the results show that this method can greatly reduce the execution time of the task, and has a good application value.

REFERENCE

[1]. Wang Haifeng, Chen Qingkui, a summary of the research on the key technology of the general calculation of the graphics processor [J] Journal of computers, 2013, 36 (4): 757-772

[2]. Han Jungang, Liu Youyao, Zhang Xiao, history status and development trend of graphics processor [J]. Journal of Xi'an University of Posts and telecommunications, 2011, 16 (3): 61-64

[3]. Tian Bing, Wu Ning, Tan Xianqiang - embedded graphics processor cutting engine design [J], "computer and digital engineering" - 2011 –

[4]. Garcia V, Debreuve E, Barlaud M. Fast K Nearest Neighbor Search using GPU. Proceedings of the CVPR Workshop on Computer Vision on GPU. 200