Multi Core Real Time Task Allocation Algorithm for the Resource Sharing Gravitation in Peer to Peer Network

Hua Huang
Modern Education Technology Center, Information Teaching Applied Technology Extension Center, Chongqing Vocational Institute of Engineering, Chongqing, 402260, China

Qing Wan
Information Teaching Applied Technology Extension Center, Chongqing Vocational Institute of Engineering, Chongqing, 402260, China

Lin Liu
Information Teaching Applied Technology Extension Center, Chongqing Vocational Institute of Engineering, Chongqing, 402260, China

Abstract
Due to the increasingly large size and function of the multi core real time task allocation, the limitations of the traditional design methods become more and more obvious. To solve this problem, a kind of multi core real time task allocation algorithm based on the resource sharing gravitation in the peer to peer network is put forward in this paper. The peer to peer network is composed of 4 parts, that is, a cell library, the probe points, the optimization convergence algorithm and the kernel unit. Among them, the cell library is designed to help the designers easily build the multi core real time task allocations for various kinds of topological structures and map the complex algorithms to the multi core system. The probe points are embedded in the established multi core, which can perform evaluation on the time delay, power consumption throughput and other system performance when the multi core system executes the algorithm. The optimization convergence algorithm relies on the information obtained by the probe points to identify the topological structure and the algorithm mapping that optimizes the system performance. And the task of the kernel unit is to carry out simulation for the configured multi core system and obtain the corresponding data resources and optimization results. The method proposed in the paper can be used to analyze and optimize the complex real time task distribution in the early stage of the design, which can shorten the development cycle and improve the design performance.

Keywords: Peer to Peer Network, Real Time Task Allocation, Multi Core, Resource Sharing.

1. INTRODUCTION
With the increasingly wide application of the multi core real time task allocation (Wang, Bian, Wu, and Xiong, 2006), how to map a complex application to a multi core real time task distribution of a certain topological structure and meet the corresponding requirements for performance and overhead has become one of the important topics in the studies of the real time task allocation (Keutzer, Newton, Rabaey, and Sangiovanni-Vincentelli, 2000; Grötker and Martin, 2002; Balarin, Watanabe, Hsieh, Lavagno, Passerone, and Sangiovanni-Vincentelli, 2003). As the size and complexity of the multi core system is becoming greater and greater (Chakrabarty, 2005), the problems such as the hardware and software partitioning, the task allocation, and establishment of the on-chip internet and so on, need to be evaluation properly in the early stage of the design (Posadas, Herrera, Sanchez, Villar, and Blasco, 2004; Bergamas, Nair, and Dittmann, 2007; Kempf., Doerper, and Leupers, 2005), so as to realize the optimization of the overall performance. The multi core system design method can only describe and analyze these problems in the early stage of the design, make performance evaluation and then put forward the performance optimization strategy on this basis (Kienhuis, 1999; Bolic, DjuricP, and Hong, 2005).

An excellent multi core system design method should be able to meet the requirements of the space exploration for a specific application, so as to identify a relatively superior task allocation method to improve the performance of the multi core systems. Therefore, a kind of multi core real time task allocation algorithm with the resource sharing gravitation in the peer to peer network is put forward in this paper. This method makes use of the multi core system modeling language to carry out modeling flexibly to various types of multi core systems. By introducing the concept of the probing points, evaluation can be conducted for the performance parameters of the multi core system such as the power consumption, the time delay, the throughput and so on. The most important is that, the method put forward in this paper has improved the algorithm using the resource sharing gravitation to obtain the extreme with a constraint, and applied it to the method proposed in this paper, which can conduct optimization convergence for the power consumption, time delay, throughput and other
performance index of the multi core system, so as to implement the multi core real time task allocation. Compared with the traditional optimization convergence algorithms that are based on the exhaustivity, the method put forward in this paper has greatly reduced the complexity of the algorithm.

2. DESIGN METHOD FOR THE MULTI CORE REAL TIME TASK ALLOCATION

2.1. Prediction Accuracy of the Multi Core System

For the real time task allocation design, the traditional Y-diagram design method divides the design of the multi core system into two independent aspects: the system architecture and the application function. On the basis of the mapping method, the computing and the communication, the behavior and the structure are combined organically, to obtain the multi core system performance description that can be executed.

On the basis of the Y diagram design method, a kind of multi core real time task allocation design method based on the peer to peer network is put forward in this paper. By integrating the cell library, the probe point and the optimization convergence algorithm into the design, to meet the requirements for the common ability, the evaluation ability and the optimization ability in the design of the three multi core systems.

1) Cell library. The cell library is composed of the processor cell library, the internet cell library, the time sequence cell library and the task partition library, in which, the processor cell library keeps the high level behavior model of different processors. The internet cell library keeps the high level behavior model that connections each of the processors and the bus of the peripheral equipment or the on-chip internet. The time sequence unit library keeps the processing time delay of all the tasks that are contained in the applications as well as the dependency relation between the tasks. The task partition unit library keeps the specific information on how the various tasks are allocated to the multi core system.

The processor unit library and the internet unit library are subordinate to the architecture modules, through which the high level modeling of the multi core system architecture can be realized. The sequential cell library and the task partitioning unit library are subordinate to the application function modules, through which the description of the time sequence relation with the tasks and the distribution mode can be realized.

As the cell libraries have covered the various components of the real time task allocation, they can carry out modeling for a wide range of different applications and topologies very well to meet the requirements for the common ability.

2) Probe point: The probe point is embedded in the multi core architecture module and the application function module at the same time, through which the power consumption, time delay, throughput and other information of the multi core system is collected. In addition, calculation and evaluation is carried out on the corresponding multi core system performance, which has met the requirements for the evaluation ability.

3) Optimization convergence algorithm: The optimization convergence algorithm provides the design space exploration for the multi core system. In this paper, the idea of the resource sharing gravitation for the solving of the extremum with a constraint is introduced into the design space exploration. And the iteration configuration of the unit library is used to make the multi core system performance index provided by the probe points reach the extreme point, so as to meet the requirements for the optimization ability.

2.2. Multi Core Real Time Task Allocation Shared Resources

One of the keys to the design of the multi core system is how to describe the distribution, execution and transmission of the application in the multi core system in a high level language. In the method put forward in this paper, this part of work is managed, scheduled and executed by the kernel of the peer to peer network. The kernel of the peer to peer network includes three kinds of main data structures, that is, the request queue, the transmission queue and the bus traffic. And the specific working principle is shown in Figure 1.

![Figure 1. Kernel bus information of the peer to peer network](image-url)
First of all, the request queue accepts the data loading and storage requests from each processor. If the request queue of the processor is not full, the request will be added to the corresponding request queue; otherwise, the processor will enter into a waiting state until the request queue can accept a new data transfer request. Then, the transmission queues store the transfer requests that are issued by each request queue and determine whether the transfer request is allowed to access the bus resources through an arbitration strategy that is embedded in the transmit queue. When the bus is busy, the transmission queue is waiting there and no longer accepts the transmission request issued by the request queue. Finally, the bus traffic determines whether the transmission queue requests are allowed to access the corresponding bus resources in accordance with the occupancy of the bus resources.

3. DESIGN OF THE MULTI CORE REAL TIME TASK ALLOCATION SPACE RETRIEVAL

The design space exploration is a very important part in the multi core real time task allocation design. The design space exploration can be divided into the exploration of topological structure and the exploration of task partition. In the exploration of the topological structure, different situations of the specific application under different bus architectures, arbitration methods, internet and other hardware configuration conditions are taken into consideration, so as to identify the relatively good architecture for the implementation of this application. On the other hand, the exploration of the task partitioning focuses on the division of the specific application to the existing hardware architecture. And different methods of partitioning will indicate different multi core system performance such as power consumption, time latency, throughput and so on; hence the method of task partitioning that can exert the advantages of the hardware architecture to the maximum will be identified.

3.1. Overall Architecture

The overall architecture of the method put forward in this paper to design the space exploration is shown in Figure 2. After the exploration of the design space begins, the various cell libraries are first configured, and the application is executed. Then through the probe point, the power consumption, time delay, throughput and other performance overhead incurred from the execution is obtained from the architecture module. Finally, it is judged whether the optimization is completed. If optimization is not completed, the cell library is reconfigured by the optimization convergence algorithm. The process is running in cycles until the optimization is completed.

Figure 2. Overall architecture of the design space exploration
3.2. Topological Structure and Exploration of the Task Partitioning

By changing the depth of the request queue of each processing unit in the kernel of the peer to peer network, the arbitration mode of the bus, as well as the internet cell library, it is possible to realize the design space exploration of the topological structure.

Table 1 shows the settings of the modeling parameters for the simple bus that is based on the bus arbiter, the resource sharing of the peer to peer network of the IBMCELL processor, as well as the on-chip network based on the two-dimensional grid structure.

<table>
<thead>
<tr>
<th>Table 1. Parameters setting of different internets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth of the request queue</td>
</tr>
<tr>
<td>Depth of the request queue</td>
</tr>
<tr>
<td>Bus arbitration mode</td>
</tr>
<tr>
<td>Internet cell library</td>
</tr>
<tr>
<td>Internet cell library</td>
</tr>
</tbody>
</table>

By configuring the task partitioning unit library, it is possible to achieve the design space exploration of the task partitioning. And the different configuration of the cell libraries corresponds to the different mapping modes of different tasks to the multi core system.

3.3. Shared Resource Optimization Algorithm

In this paper, the fixed topological structure is adopted to carry out optimization convergence to the partition of the shared resources in the peer to peer network. The performance evaluation function $f(X)$ is taken into consideration, in which $X = x_1, x_2, x_3, \ldots, x_N$, where $N$ different sub tasks are assigned to $N$ different processing units for operation, and the sub task $i$ is running on the $x_i$-th processing unit, then there is the constraint condition as the following

\[ C : x_i \in [1, N] \& x_i \neq x_j (i \neq j). \] (1)

Therefore, the design space exploration problem of the partition of the tasks has become the problem of solving the extremum of the performance evaluation function $f(X)$, $X = x_1, x_2, x_3, \ldots, x_N$ under the condition of the constraint $C : x_i \in [1, N] \& x_i \neq x_j (i \neq j)$.

Due to the presence of the full permutation mode of the $P_N^N = N!$ in the $X = x_1, x_2, x_3, \ldots, x_N$. When the number of the processing units in the multi core system is relatively large, it may be excessively huge in the computational complexity to obtain the performance evaluation function $f(X)$ through the exhaustive traversal. Therefore, on the basis of the method for the solution of the extremum with a constraint using the resource sharing gravitation, the optimization convergence algorithm with low algorithm complexity is put forward in this paper, so that the performance evaluation function $f(X)$ can be quickly converged to the local extremum point. The specific algorithm is as the following:

When the partition of the sub task to the processing unit is changed, the variation quantity of the performance evaluation function is as the following

\[ \Delta f(X) = f(X + \Delta X) - f(X), \] (2)

The variation distance is as the following

\[ \rho = \sqrt{(\Delta x_1)^2 + (\Delta x_2)^2 + (\Delta x_3)^2 + \cdots + (\Delta x_N)^2}. \] (3)

Therefore, the resource sharing gravitation of the performance evaluation function $f(X)$ is as the following
\[ \text{grad} f(X) = \max \left\{ \frac{\partial f}{\partial l} \right\} \quad (4) \]

In which, the directional derivative is as the following:
\[ \frac{\partial f}{\partial l} = \lim_{\rho \to 0} \frac{\Delta f(X)}{\rho}. \]

Considering the constraint condition \( C : x_i, x_j \in [1, N] \& x_i \neq x_j \). When and only when the values of \( x_i \) and \( x_j \) are exchanged, there is the minimum variation distance \( \rho_{\min} = \sqrt{(x_i - x_j)^2 + (x_j - x_i)^2} = \sqrt{2} |x_i - x_j| \); then the extrema of \( f(X) \) is \( f_{\text{extreme}}(X) = \sum_{i=0}^{\infty} (\text{grad} f(X_i) \times \rho_{\min}) \).

When \( \text{grad} f(X_i) = 0 \), the function is converged to the extreme point.

However, compared with the traditional methods of the design space exploration based on exhaustivity, the algorithm based on the resource sharing gravitation can only achieve the convergence of the shared resources to the local extremum rather than the extreme point of the shared resources. As shown in Figure 3, Point1 is the starting point of the convergence algorithm. By adopting the algorithm to solve the extremum with a constraint based on the resource sharing gravitation, the shared resources will be gradually converged to Point2, that is, the local extreme point. However, the method that is based on exhaustivity can perform search in all the possible ranges, which makes it possible to eventually converge in Point3, that is, the extreme point of the shared resources.

![Performance function](image_url)

**Figure 3.** Schematic diagram of the convergence situation of the shared resources

4. EXPERIMENT AND SIMULATION

4.1. Architecture and Algorithm

According to the design flow as shown in Figure 2, the 32-bit RISC processing unit is selected from the topological structure module in this paper as the processing element (PE) and the central unit (CU), and the peer network resource sharing is adopted as the PE connection method to construct the multi core system topological structure as shown in Figure 4. In the task module, the improved distributed particle filter algorithm is selected in this paper to be applied in the image target tracking to carry out the algorithm mapping and the design space exploration and analysis. Through the setting of the probe points, the multi core system power consumption and the time delay information is collected. In Figure 4, the multi core system is composed of 1 CU and 12 PEs. The internet is 4 buses with the width of 128B, in which, there are 2 clockwise and 2 counterclockwise ones each. Each bus can support 3 transmission requests at the same time.
4.2. Design Space Exploration and Optimization Convergence

The design space exploration and the improved distributed particle filter algorithm can be processed in parallel, that is to say, when 12 PEs are performing the particle filter algorithm for the current frame, the CU can compute how to partition the particle filter task in the next frame, so that the power consumption, time delay or throughput of the multi core system can achieve the minimum.

In order to ensure that the computational overhead of the design space exploration does not affect the multi core system, the time for the CU to complete the design space exploration time must be less than the time for the PE to execute the improved distributed particle filter algorithm, that is, to meet the time constraint condition as the following

$$T_{CU} \leq T_{PE}; \quad (5)$$

In which, $T_{CU}$ stands for the time of the space exploration for the CU operation, and $T_{PE}$ stands for the time of the improved distributed particle filter for the operation of the PE.

In fact, due to the very large computational complexity of the particle filter algorithm, the time it takes for the PE to run the improved distributed particle filter is much higher than the time it takes for the CU to explore the design space. And the time constraint is met. Therefore, the dynamic design space exploration can be realized. When the PE1~PE12 are implementing the image target tracking, the CU is calculating the next design space exploration in the next frame at the same time, as shown in Figure 5.

$PE_{1} \sim PE_{12}$ calculate the image target tracking
CU calculates the next frame in the design space exploration

In the design space exploration, it is necessary to conduct optimization for certain performance. And in this paper, the power consumption is selected as the goal of the performance optimization.
1) Establish the performance evaluation function $F_{\text{power}}(X)$. Since the different task allocation modes only affect the power consumption when the PEs are communicating with each other, there is no need to consider the power consumption calculated within each PE.

The specific process for the establishment of the performance evaluation function $F_{\text{power}}(X)$ is as the following: The probe points of the monitoring bus traffic in the peer to peer network are used, with four peer to peer network resource sharing. There are a total of 12 PEs on each bus, and 48 probe points are set accordingly. When there is transmission request on the bus, the corresponding probe point records that the bus is busy, and that there is the power consumption overhead; while when there is no transmission request on the corresponding bus, the probe point records that the bus is idle, and that there is no power consumption overhead. The performance evaluation function $F_{\text{power}}(X)$ monitors these 48 probe points at each clock cycle, accumulate the total number of the probe points where the bus is recorded as busy, and finally obtain the power consumption that is required for processing one frame of image.

2) Optimization convergence put forward in Section 3 is carried out for different particle distribution modes, and the extreme value of the performance evaluation function is obtained through the iteration.

The specific process of the iteration is as the following: In the target algorithm, a total of 480 particles are used. First of all, the computing tasks of 480 particles are randomly allocated to 12 Pes, so that each PE has the computing task of 40 particles, which is taken as a sub task. In the process of each iteration, 2 out of the 12 PEs are selected for the sub task exchange (a total of $C_2^2$ exchange modes), and the corresponding $F_{\text{power}}(X)$ function is calculated. The corresponding exchange $S[i, j]$ (PE$_i$ and PE$_j$ exchange the sub tasks) is recorded. From these $F_{\text{power}}(X)$ functions, the one with the minimum function value is identified to carry out the exchange, and then enter into next iteration, which is conducted in cycles, until the performance evaluation function $F_{\text{power}}(X)$ reaches the extreme value.

4.3. Experimental Data and Analysis

Table 2 shows the optimization convergence process of three consecutive frames, in which, T stands for the number of convergence cycles of the design space exploration algorithm, Si stands for the sub task exchange process of the i-th frame image, $F(X)$ stands for the function value of the performance evaluation function of the i-th frame image, which corresponds to the power consumption information herein.

<table>
<thead>
<tr>
<th>T</th>
<th>S1</th>
<th>$F_1(X)$</th>
<th>S2</th>
<th>$F_2(X)$</th>
<th>S3</th>
<th>$F_3(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>3328</td>
<td></td>
<td>2848</td>
<td></td>
<td>3360</td>
</tr>
<tr>
<td>3</td>
<td>[2,4]</td>
<td>2112</td>
<td>[0,1]</td>
<td>2128</td>
<td>[5,6]</td>
<td>2064</td>
</tr>
</tbody>
</table>

What is shown in Figure 6 can indicate more intuitively the convergence trend shown in Table 2. In Figure 6, the optimization convergence of each frame of image is completed within 5 convergence cycles, and in each cycle, the sub task of 2 PEs is exchanged, so that the reduction in the performance evaluation function $F_{\text{power}}(X)$ in the cycle is the maximum. After 5 cycles, it is eventually converged to the minimum value of the performance evaluation function $F_{\text{power}}(X)$.

![Figure 6. Design space exploration for 3 consecutive frames of image](image-url)
Figure 7 shows the proportion of the degree of optimization of the performance evaluation function to the overall degree of optimization in each convergence. And the mean value of the degree of optimization for the 3 frames in each convergence is 51%, 26%, 12%, 8% and 3%, respectively. In Figure 7, frame 1, frame 2 and frame 3 are the optimization cases of the 3 frames, respectively. The mean is the mean value of these 3 frames. Therefore, it can be seen that the optimization degree of the first convergence is the largest, which has reached 51%; while the cumulative degree of optimization in the first 3 frames is up to 89%. Therefore, in the environment that has strict time requirements for the time on the design space exploration, it is acceptable to take only the first few times of convergence into consideration. And the specific number of convergence shall be determined in accordance with the demand.

The possible differences between the optimization convergence algorithm based on the resource sharing gravitation and the traditional optimization convergence algorithm based on the exhaustivity are discussed above, that is, the former can only converge on the local extreme, while the latter can converge at the extreme point. From the experimental results, it can be seen that the optimization convergence algorithm that solves the extremum with a constraint through the resource sharing gravitation can converge the performance evaluation function to the extreme point, while the complexity of the algorithm is much lower than that of the traditional optimization convergence algorithm based on exhaustivity.

In this paper, the power is selected in the experiment as the objective function of the design space exploration, and the throughput or time delay can also be taken as the performance evaluation function to conduct the design space exploration through the method put forward in this paper.

5. CONCLUSIONS

In this paper, a kind of multi core real time task allocation algorithm for the resource sharing gravitation in the peer to peer network is put forward. Through the cell library, the probe point and the optimization convergence algorithm, the requirements for the common ability, assessment ability and optimization ability that are faced by the multi core system design can be solved. The method put forward in this paper can design the space search under the multi core real time task allocation in the two dimensions of the topological structure and the task division simultaneously, which can perform the optimization and convergence for the power consumption, time delay or throughput and other multi core system performance while implementing the design space exploration at the same time. This kind of optimization convergence algorithm that is on the basis of the resource sharing gravitation to solve the extremum with a constraint can significantly reduce the complexity of the algorithm compared with the traditional optimization convergence algorithm that is on the basis of exhaustivity. The experimental results show that the algorithm put forward in the paper can allow the multi core system to perform dynamic space exploration while allocating a large number of real time tasks in the shared gravitation of the shared resources in the peer to peer network.

Acknowledgements

This work was supported by New Technology Promotion Project of Higher Vocational and Technical Colleges in Chongqing "Development and Application of Smart Campus New Technology Based on the Mobile Internet and Big Data Support" (No.: GZTG2016009); Science and Technology Research Project of Chongqing Municipal Education Commission "University Information Innovation Teaching Integrated Platform Research and Practice Based on Cloud Technology" (No.: KJ1732434); and Higher Education Teaching Reform
Research Project in Chongqing "Teaching Resource Library Platform Construction Research and Practice Based on Cloud Technology" (No. 151026).

REFERENCES


