Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A GENERALIZED MULTIFRAME REAL-TIME TASK MODEL UPON HETEROGENEOUS MULTIPROCESSORS SYSTEM

YAN LIU, CHENG XU, LIDONG WANG
College of Information Science and Engineering, Hunan University, Changsha 410082, Hunan, China

ABSTRACT

The task scheduling problem is one of the basic research areas in computer science especially in real-time systems. Current task scheduling based on heterogeneous multiprocessors system rarely consider the multiframe character of real-time tasks, which assumes a worst-case execution time bound for every task and maybe too pessimistic if the worst-case execution time of task is much longer than the average. Given in this paper is a generalized multiframe real-time task model upon heterogeneous multiprocessors systems based on period task model. It is proved that this model's schedulability is better than period task model. We investigate the schedulability problem for this model for the preemptive fixed priority scheduling policy using genetic algorithm. The scheduling test is given as utilization based and exact worst case response time. The experiments results show that the using response time analysis is better than the utilization based method, and the analysis is applicable to different system model.

Keywords: *Real-time System, Task Model, Response Time*

1. INTRODUCTION

With the development of embedded systems, the single processor computing platform has been difficult to meet the demands of the applications. Heterogeneous multiprocessor computing platform has been become the mainstream due to the pressure from product cost, performance, time-to-market In the field of consumer electronics. Heterogeneous multiprocessor computing platform generally consists of a wide variety of computing units, and each computing unit can be used to execute specific types of computational tasks efficiently. Based on this platform to develop applications need to consider the execution time on different types of computing units. This paper discusses the realization of real time system task scheduling problems based on above platform, proposes a generalized multiprocessors and multiframe task model, which is used to describe the real-time tasks with significant frame characteristics (such as MPEG streaming media task) and their implementations on heterogeneous multiprocessor platform. Multiprocessor task scheduling is NP problem. The existing heterogeneous multiprocessor task allocation method only considering the worst-case execution time, but actually the execution time of many tasks is changing with period (i.e., multiframe properties). Based on the analysis of existing related work, this paper studies the multiframe task model and its schedulability problem upon multiprocessor platform. Theoretical analysis and simulation results show that, considering the difference between task frames can increase the upper bound of processor utilization using fixed priority scheduling strategy, and can improve the success rate of task scheduling compared with the only considering worst-case execution time.

Task scheduling is the core problem of computer science. In 1967, Manacher first presented the concept of task [1]. Liu.C and Layland.J proposed classical periodic task set [2], a task can be described by a two-tuples which consists of the task period and the execution time. They also proposed RM and EDF scheduling algorithm based on single processor, and processor utilization upper bound used for schedulability test. However, treating the task execution time as fixed is not suitable for all situations. Aloysius K. Mok proposed multiframe task model based on periodic task model, to present the changeable tasks in practical implementations [3, 4]. Because the periodic task model proposed by L&L pessimistically using worst-case execution time of task scheduling to analyze the task schedulability, makes some scheduling models can't be scheduled. The research results about multiframe periodic task set is very rich. Literature [5] provided a new judgment method about multiframe task schedulability. Literature [6] discused more exact judgment conditions about task schedulability through adding multiframe character

Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

based on traditional algorithems. In 1999, Aloysius K. Mok further generalized the multiframe task, which made not only the task execution time is variable, task duration and the period are also variable in [7]. In recent years, multi-frame task research also had new progress [8, 9], where A.Zuhily found the necessary and sufficient conditions of multiframe task scheduling by response time. Sanjoy Baruah has been concerned with the research of task allocation on heterogeneous multiprocessor, and published numerous papers. Literature [10] expanded the classical L&L periodic task model on heterogeneous computing system, make use of the processor utilization parameter to reflect the capability of different processors, and provided scheduling algorithm. Literature [11] reviewed recent research in related fields. The group of Dai and Wang in Institute of software, Chinese Academy of Sciences[12-14] published a series of articles, to study the real-time dynamic scheduling algorithm for multiprocessor system. In the multiframe task scheduling, Huang discussed the fixed priority scheduling algorithm of periodic multiframe task[15] [16]. Literature [17] started from the response time to analyze the schedulability of periodic multiframe real-time task. Above works both based on uniprocessor platform.

This paper is organized as follows. Section 2 gives the definition of multiframe task model on multiprocessor, and the description of task allocation problem. Task schedulability analysis described in section 3. Scheduling algorithm design and experimental results are given in section 4 and section 5. Section 6 is the conclusion.

2. MULTIFRAME TASK SCHEDULING ON MULTIPROCESSORS

2.1 Task Scheduling Model

Definition 1. Multiframe task scheduling model on multiprocessor is a model used to describing the periodic task working in the computing system with n processors and m tasks.

$$\tau_i = (\vec{E}, T, D) = ((\eta_{ikj})_{n\times m}, t_i, d_i) \qquad (0 \le i \le n-1) \tag{1}$$

Where $0 \le k \le n_i - 1$, $0 \le j \le m-1$, η_{ikj} denotes the processor utilization rate of the k-th frame t_i of task on processor , t_i and d_i are task period and task deadline respectively. Processor utilization rate represents the processor utilization degree of task, and can be computed by the ratio of task executing time and period [10]. For task τ_i , we adopt e_{ikj} and p_{ikj} to describe the executing time and period of t_i on respectively, then $\eta_{ikj} = e_{ikj} / p_{ikj}$ 。

Example 1. A system with 2 heterogeneous processors P=(p_0 , p_1), there are 3 multiframe periodic tasks need to be executed, that is n=2, m=3. Supposing the period is equal to the deadline for every task, are 3, 6 and 5 respectively. The multiframe type of 3 tasks are 2, 1 and 1 respectively, that is n_0=2, n_1=1, n_2=1. Then,

$$\tau_0 = \left(\vec{E}_0\right)_{2\times 2} = \begin{pmatrix} 3/3 & 3/3 \\ 1/3 & 2/3 \end{pmatrix} \text{ and}$$

$$\tau_1 = \left(\vec{E}_1\right) = \begin{pmatrix} 5/6 & 3/6 \\ 5/6 & 3/6 \end{pmatrix}, \text{ and}$$

$$\tau_2 = \left(\vec{E}_2\right) = \begin{pmatrix} 1/5 & 4/5 \\ 1/5 & 4/5 \end{pmatrix} 。$$

Multiframe task model on multiprocessor is the generalized expand of periodic task model. Through the following case we can see that, the common multiprocessor periodic task, multiframe real-time task and periodic task model are all the special cases of multiframe task model on multiprocessor.

2.1.1 multiprocessor periodic task

Multi frame task model on multiprocessor can be described as a multiprocessor periodic task model, as long as without considering the difference between task frames, but only considering the processor utilization rate of task under worst case [10]. Then example 1 can be modified as:

$\tau_0 = (3/3 \quad 3/3)$,

$\tau_1 = (5/6 \quad 3/6)$,

$\tau_2 = (1/5 \quad 4/5)$ 。

2.1.2 multiframe task model

Multiframe task model on multiprocessor can be described as a multiframe task model, if only considering the multiframe task in single processor system [3, 4]. Then example 1 can be modified as:

Processor p_0:

$$\tau_0 = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix}, \tau_1 = \begin{pmatrix} 5/6 \\ 5/6 \end{pmatrix}, \tau_2 = \begin{pmatrix} 1/5 \\ 1/5 \end{pmatrix} 。$$

Processor p_1:

$$\tau_0 = \begin{pmatrix} 3/3 \\ 2/3 \end{pmatrix}, \tau_1 = \begin{pmatrix} 3/6 \\ 3/6 \end{pmatrix}, \tau_2 = \begin{pmatrix} 4/5 \\ 4/5 \end{pmatrix} 。$$

2.1.3 L&L periodic task model

Without considering the multiframe character, and if the task executing on single processor, then

Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

our model in this paper is the same with classical periodic task model [2].

Processor p0：
$\tau_0 = (3/3)$, $\tau_1 = (5/6)$, $\tau_2 = (1/5)$ 。

Processor p_1：
$\tau_0 = (3/3)$, $\tau_1 = (3/6)$, $\tau_2 = (4/5)$ 。

2.2 Description of Scheduling Problem

In this paper, we pay attention to task's local scheduling method between multiprocessors, that is, if a task is assigned to execute on a certain processor, it will be executed always on this processor. In this way, we can adopt preemptive fixed priority scheduling policy to optimize scheduling on single processor. In addition, the communication between tasks is not considered in this paper.

The task scheduling problem on heterogeneous multiprocessors system can be described as follows: given a multiprocessor computation platform composed by various kinds of computing unit, and a group of real-time tasks executing on it, we need to find the way to assign these tasks on each computing unit reasonably, and satisfy the demand of task timing requirements and each processors' computing capability simultaneously.

Theorem 1. The task allocation problem of heterogeneous multiframe system on multiprocessors is a NP-complete problem.

Proof: we use an n*m Boolean matrix x_{ij} to present the task allocation. Then if x_{ij} =1, it denotes allocating task τ_i on processor p_j to execute, otherwise, it denotes task τ_i is not allocated. Denote the problem as $\Gamma(n,m)$.

The classical 3-Partition problem has been proved an NP-complete problem. For any one instance Π of 3-Partition, we can transform it as a heterogeneous multiframe task allocation problem Π' on multiprocessors through the following method. Assume the system is composed by k processors (corresponding k subsets) and 3k tasks (corresponding 3k elements).

If Π is solvable，we can get the solution of Π' through the following method。For the solution of Π，we can get subsets L0,L1,…,Lk-1。Set every element in the subset representative task will map to corresponding processor, and every processor's utilization rate not exceeding 1. If Π' can be solved, we can compose a subset with tasks running on the k processors. Therefore, problem Π is solvable, if an only if problem Π' , that is the multiprocessor scheduling allocation successful. □

Formalizing the task allocation problem of heterogeneous multiframe system on multiprocessors, is equal to find the allocation matrix of $X_{n\times m} = (x_{ij})$, and to ensure the rational allocation of each task to the processor, and make the loads between processors is relatively balanced. It can be described by nonlinear programming as follows, denotes as $\Gamma(n,m)-M$, and in which abs represents the absolute value.

$$\min \sum_{j_1=0, j_2=0, j_1\neq j_2}^{m-1} abs\left(\sum_{i=0}^{n-1} x_{ij}\eta_{ij_1}^m - \sum_{i=0}^{n-1} x_{ij_2}\eta_{ij_2}^m\right)$$

s.t

$$\begin{cases} x_{ij} = 1 \quad or \quad 0 \qquad 0 \le i \le n-1, 0 \le j \le m-1 \\ \sum_{j=0}^{m-1} x_{ij} = 1 \qquad 0 \le i \le n-1 \\ K = \sum_{i=0}^{n-1} x_{ij} \\ r = \min_{i=0}^{n-1}(x_{ij} r_{ij}) \\ \sum_{i=0}^{n-1} x_{ij}\eta_{ij\max} \le r \cdot K \cdot ((\frac{r+1}{r})^{1/K} - 1) \qquad 0 \le j \le m-1 \end{cases} \quad (2)$$

The system includes n multiframe tasks on multiprocessor, and m processors. Using $\eta_{ij}^0 = \max_{k=0}^{n_i-1}(\eta_{ikj})$ to denote the largest frame of task τ_i , r_{ij} denotes the difference between frames when task τ_i is executing on processor p_j . In which, n_i is the number of the task's frame types, $0 \le i \le n-1$, $0 \le j \le m-1$. For every processor p_j , make $r = \min_{i=0}^{l_j}(r_{ij})$, where l_j is the number of tasks executing on processor p_j .

Denote heterogeneous multiprocessor task allocation in lecture [10] as $\Gamma(n,m)-N$, where allocated task sets is $A(n,m)$, and in this paper, denote the task allocation which the tasks are added multiframe character as $\Gamma(n,m)-M$, where allocated task sets is $B(n,m)$.

3. SCHEDULABILITY ANALYSIS

Referring lectures [4] and [9], we will give the accmulatively monotonic (AM) character of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

multirame task on multiprocessor defined in this paper.

Definition 2. For a multiframe task on multiprocessors $\tau_i = (\vec{E}, T, D) = ((\eta_{ikj})_{n\times m}, t_i, d_i)$, its utilization rate array on processor p_j is $\overline{\eta}_j = (\eta_j^0, \eta_j^1, ..., \eta_j^{n-1})$, if $\forall j, \exists m \in [0, n-1]$, and make $\forall s,t \in [0, n-1]$, has $\sum_{k=m}^{m+s} \eta_j^{k \bmod n} \geq \sum_{k=t}^{t+s} \eta_j^{k \bmod n}$, then such task is called accmulatively monotonic multiframe task on multiprocessors.

Where η_j^m is the largest frame of the task on processor p_j, we can suppose $\eta_j^m = \eta_j^0$.

Definition 3. For a classical multiframe task on multiprocessors τ_i, its difference between frames on processor p_j is:

$$r_j = \begin{cases} \eta^0 / \eta^1 & n \geq 2 \\ 1 & n = 1 \end{cases}$$

Theorem 2. *For n periodic multiframe tasks with AM fixed priority, its upper bound of utilization rate on processor p_j is* $r \cdot n \cdot ((\frac{r+1}{r})^{1/n} - 1)$, *where* $r = \min_{i=0}^{n-1} r_i$.

Because on single processor, the multiframe task on multiprocessors fit the characters of simple multiframe task, then, the proof process of theorem 2 can refer to lecture [4].

Theorem 3. $B(n,m) \supset A(n,m)$。

Proof:

（a） $\exists \Gamma(\tau_0, \tau_1, ..., \tau_n) \in B(n,m)$, but $\Gamma(\tau_0, \tau_1, ..., \tau_n) \notin A(n,m)$。

Considering $\Gamma(3,2) - M$, in which the tasks set $\Gamma(\tau_0, \tau_1, \tau_2)$ is:

$$\tau_0 = \begin{pmatrix} 2/3 & 1 \\ 1/27 & 3/4 \end{pmatrix}, \tau_1 = \begin{pmatrix} 2/7 & 6/7 \\ 2/63 & 1 \end{pmatrix},$$

$$\tau_2 = \begin{pmatrix} 9/10 & 9/10 \\ 1/10 & 9/10 \end{pmatrix}。$$

Allocate task τ_0, τ_1 on processor p_0 to execute, we can get $r = \min(r_{00}, r_{10}) = 9$, then the upper bound of processor's utilization rate is $U_{B-M} = 2 \times 9 \times ((\frac{10}{9})^{1/2} - 1) \approx 0.974$, that is, they can be scheduled on processor p_0 to execute. Similarly, task τ_2 also can be scheduled on processor p_1. Then exist $\Gamma(\tau_0, \tau_1, \tau_2) \in B(3,2)$.

The tasks set described by $\Gamma(n,m) - N$ has the following former:

$\tau_0 = (2/3 \quad 1)$, $\tau_1 = (2/7 \quad 1)$, $\tau_2 = (9/10 \quad 9/10)$。

A processor must be allocated two tasks, then the upper bound of processor's utilization rate to scheduling two tasks is $U_{B-N} = 2 \times (2^{1/2} - 1) \approx 0.828$. The task allocation plan satisfied above restrain can't be found, that is $\Gamma(\tau_0, \tau_1, \tau_2) \notin A(n,m)$.

（b）Using the distributable tasks set described by $\Gamma(n,m) - N$, and transform it as $\Gamma(n,m) - M$, then all tasks can be allocated. That is $\forall \Gamma(\tau_0, \tau_1, ..., \tau_n) \in A(n,m)$, then $\Gamma(\tau_0, \tau_1, ..., \tau_n) \in B(n,m)$。Transforming as discussing the function's monotonous showed by $f(r) = r \cdot K \cdot ((\frac{r+1}{r})^{1/K} - 1)$ $(r \geq 1)$, we only need to prove $f(r) \geq f(1)$。

Combine $K \cdot r^{1/K} \cdot (r+1)^{1-1/K} > 0$ with the derivation of function $f(r)$, we can get $f'(0) > 0$ $(r \geq 1)$, which is monotone increasing function, then exists $f(r) \geq f(1)$. □

In the scheduling of fixed priority, considering the difference between frames improves the upper bound of processor's utilization rate. The upper bound is larger than the case of not considering the difference, that is the case of only considering worst-case execution time.

Definition 4. Given a multiframe task τ_i on multiprocessor satisfying AM characters, its cumulative function is $\xi_{ij}^s(k) = \sum_{l=s}^{s+k-1} e_{ij}^{l \bmod n}$, represents the cumulative executing time from frame f_s to f_k of task τ_i on processor p_j. If the task is not a multiframe task, then is $k \cdot e_{ij}^{\max}$。

Definition 5. Given a multiframe task τ_i on multiprocessor satisfying AM characters, its waiting time is $W_{ij}(R_{ij}) = \sum_{s=0}^{i-1} \xi_{sj}^0 \left(\left\lceil \frac{R_{ij}}{t_s} \right\rceil \right)$, task priority is decreasing with 0-*K*. The above formula represents the waiting time generated by task τ_i on

Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processor p_j, and considering each task with higher priority executing $\left\lceil \frac{R_{ij}}{t_s} \right\rceil$ times in its response time.

Definition 6. Given a multiframe task τ_i on multiprocessor satisfying AM characters, its blocking time B_{ij} represents the blocking time of task τ_i on processor p_j. This blocking time is generated because the tasks which have lower priority than τ_i need to execute and then block the executing of task τ_i.

Definition 7. Given a multiframe task τ_i on multiprocessor satisfying AM characters, its respond time is:

$$R_{ij} = C_{ij}^0 + B_{ij} + W_{ij}(R_{ij}) \qquad (3)$$

The respond time of task τ_i on processor p_j is the maximum time between its arrival and completed. It is the sum of the worst dealing time, blocking time and waiting time.

$$R_{ij} = C_{ij}^0 + W_{ij}(R_{ij}) = C_{ij}^0 + \sum_{s=0}^{i-1} \xi_{sj}^0 \left(\left\lceil \frac{R_{ij}}{t_s} \right\rceil \right) \qquad (4)$$

Based on above description, the multiframe task on multiprocessor can be scheduled on p_j, if and only if $R_{ij} \le D_i$, in which $0 \le i \le K-1$, that is , all the k tasks on p_j satisfy the restrain of respond time not greater than the deadline.

4. ALGORITHM DESING

4.1 Genetic Algorithm Design

In above, we proved the superiority of the multiframe task on multiprocessors model provided in this paper through theoretical analysis, under the condition of fully considering the multiframe character of real-time task, the scheduling can achieve higher success rate, compared with the condition of only considering the worst-case execution time. For the NP-hard problem, this paper designs multiframe task scheduling algorithm on multiprocessors based on genetic algorithm.

4.1.1 Key elements of genetic algorithm

We use real-coded mode in this paper, for example, chromosome <0,1,2,1,2,3,2,3,0,1> represents allocate tasks 0-9 to processors 0、1、2、1、2、3、2、3、0、1 respectively.

In this paper we chiefly consider the load balancing problem between multiprocessors, which make the worst utilization rates between allocated processors trend to balance, in order to avoid the case that some processors are too busy but others are idle. Assume the size of array utilition is equal to the number of processors, we use utilition to record the sum of utilization rate of processor j under responding chromosome, and the calculation method is:

$$utilition\ [j] = \sum_{i=0}^{t-1} \eta_{ij}^0 \qquad (5)$$

The original fitness function SA_utilition is the sum of utilization rate absolute difference between processors.

$$SA\_utilition = \sum_{\substack{j_1, j_2 = 0 \\ j_1 \ne j_2}}^{m} abs(utilition\ [j_1] - utilition\ [j_2]) \qquad (6)$$

We make fitness= PN/SA_utilition. In order to standardization fitness function, PN denotes the number of processors. Then the fitness is bigger, showing the responding chromosome is more superior, and the allocation is more balance.

Selection operator uses the choice mode based on sort, sorts the population according to their contribution to the sum of fitness, and combines the roulette method to select.

4.1.2 Flow of genetic algorithm

Algorithm 1.

Input the worst utilization rate matrix and frame difference matrix of tasks set, output the optimal chromosome if allocating successful or the unsuccessful information.

Step1: Original population setting. For every individual in population, allocates processor for every gene on chromosome randomly according to certain task order. If the j gene of individual i is allocated failed, then deallocation individual i, the number of attempts is 3 times of processor numbers. If the time out, it indicates that this task set is can't allocated, algorithm over, output unsuccessful allocation information.

Step2: Whether or not to meet the end condition, "yes", algorithm over, and output the optimal chromosome, "no", go to Step3.

Step3: Selection. Divide the population into some intervals.

Setp4：Calculate the frame difference vector of all individuals of the selected population. $R'_{n\times m} = (r_{ij}')_{n\times m} = R \otimes X$ ， in which $\otimes$ denotes the multiplication of corresponding position. $\vec{R}_m = (r_j)_m$ ， in which $r_j = \min_{i=0}^{n-1}(r'_{ij})$ 。

Step5：Cross。Execute according to the cross operator designed in 3.2.1.

Journal of Theoretical and Applied Information Technology
10th January 2013. Vol. 47 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step6： Mutation. Execute according to the mutation operator designed in 3.2.1. Go to Step2.

4.2 Iterative Algorithm

From the task schedulability analysis in section 3 we can see that, the calculation of task respond time need using iterative algorithm, reference lecture [9], we give the following iterative algorithm.

Algorithm 2.

Step1： $r_{ij}^0 = C_{ij}^0$；

Step2： $r_{ij}^{l+1} = C_{ij}^0 + W_{ij}(r_{ij}^l)$；

Step3：Judge whether the formula $r_{ij}^{l+1} = C_{ij}^l$ is true, end; Else, go toStep4；

Step4： $r_{ij}^{l+1} = C_{ij}^{l+1}$， go to Step2。

When the utilization rate of single processor is lower than 100%, then the algorithm above is convergence [9]. Using algorithm 1, we can calculate the performance of heterogeneous multiprocessors task model, with considering the multiframe character of real-time task. In which, theorem 2 can be used as the judgment condition to judge whether the utilization rate of processor is exceed the upper bound, this is the sufficient condition of task schedulability judgment. If in the judging process of algorithm 1, if definition 7 is used as judgment condition, this is the necessary and sufficient condition of task schedulability judgment.

5. SIMULATION EXPERIMENT

5.1 Experimental Schemes

In genetic algorithm, we adopt random distribution method to initialize the individual, and use the maximum genetic algebra method to set the end condition.

The experiment is divided into two parts.

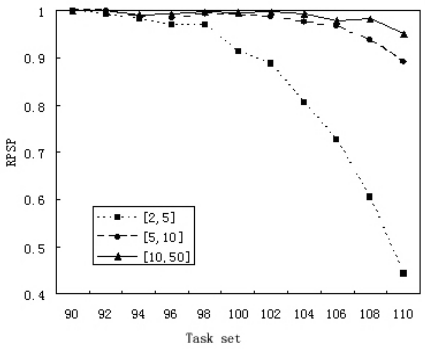
Part 1. Only considering the worst utilization rate in algorithm and record the number of task sets that cannot be allocated. Then, in the case of considering multiframe sufficient condition, calculate the number of task sets that can be scheduled. Compare the relative allocation successful rate RPSP under different parameters (the schedule successful rate of failed task sets using multiframe method).

Part 2. Compare the task allocating successful rate under sufficient condition and necessary and sufficient condition, thus indicate the superiority of the necessary and sufficient condition. For the task sets that can't be allocated by sufficient condition, use necessary and sufficient condition to judge again. In which RDTT denotes the difference between the first frame and the non-secondary big frame.

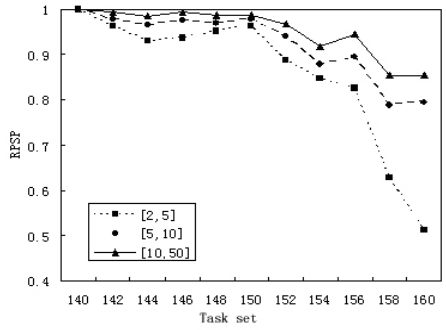
5.2 Experimental Result Analysis

Figure 1 is 2 groups of the experimental results with 20, 40 processors respectively, and the number of tasks in 40-160.

In the first two groups of the experiment, the worst utilization rate range generated are [0.00595, 0.995] and [0.0089, 0.996] respectively. In every group, compare the change of RPSP under three interval [5, 10], [10, 50] and [50,100] of frame difference range. In the last two groups of experiment, the worst utilization rate range generated are [0.00895, 0.995]. In every group, compare the change of RPSP under three interval [2, 5], [5, 10] and [10, 50] of frame difference range.



A) 20 Processors



B) 40 Processors

Figure 1: Task Scheduling Successful Rate

Comparing with the task model with not considering multiframe character, the 4 groups of experimental result in figure 1 show that, 3 curves in every group all improved the task allocation effect, it indicates that, the new model enhances task allocation successful rate through adding the frame difference, which proves the correctness of theoretical analysis. From the curves distribution in the figure we can see that, the value of the frame difference interval is bigger, the relative allocation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

successful rate is higher. This indicates that, with the increasing of system frame difference, the number of task sets can be allocated is increasing too, which further proves the correctness of theoretical analysis.

In order to compare the task allocating successful rate under sufficient condition and necessary and sufficient condition, this paper do experiment based on the following conditions: the number of processors is 10, the number of tasks is in 40-60, the first frame generate range is [490,990], the period generate range is [1000,9000], frame difference range is [5,10]. The experimental results are in table 1.

Table 1: The Relative Task Allocating Successful Rate Under Necessary And Sufficient Condition

Num of tasks	Suff-condition failed	Necessary and sufficient condition succeed (RDTT)		
		1	100	10000
40	159	21	22	24
42	213	29	30	33
44	251	41	44	45
46	263	24	26	29
48	323	58	61	62
50	341	47	48	49
52	443	46	49	53
54	449	43	48	51
56	554	50	55	59
58	626	96	99	105
60	629	65	66	77

From the results in table 1 we can see that, the schedule judgment necessary and sufficient condition of multiframe task model on multiprocessors can further schedule the tasks which failed in part 1 of the experiment, thus can allocate more tasks; at the meanwhile, with the increasing of RDTT(difference between frames), the number of task sets can be allocated is increasing too. Moreover, the necessary and sufficient condition make up the shortage of only using sufficient condition to judge schedulability from two aspects: firstly, for the same task, only use the ratio of biggest frame and secondary biggest frame as the measure of frame difference, in fact, the other frames of some tasks is smaller than the second frame; secondly, for the frame difference on processor, only considering the smallest frame difference is relatively pessimism.

6. CONCLUSION

In view of the present multiprocessor task allocation algorithm is pessimistic about only considering the worst execution time of tasks，this paper provides a general multiframe real-time task model on multiprocessor, to increase the model expression ability. We first introduces the basic definition of multiframe task model on multiprocessors, proves its allocation is an NP-complete problem, and gives formalize definition of the models with AM character. Theoretical analysis shows that, the new model can represent the models more generally, and enhance the task allocation successful rate on multiprocessors calculation platform. We design task scheduling algorithm based on genetic algorithm, and do contrast experiment from two aspects of processor utilization rate bounder and task respond time analysis. Experimental results demonstrate that, the consideration of multiframe character in multiprocessor real-time task scheduling, can enhance the task allocation successful rate. In the process of schedulability judgment, using the task respond time analysis method(necessary and sufficient condition) as judgment condition can get higher scheduling successful rate than using the processor utilization rate upper bound(sufficient condition).

ACKNOWLEDGEMENTS

This work was supported by Hunan Provincial Natural Science Foundation of China under Grant 12JJ4057, and Hunan Province Science and Technology Project under Grant 2012SK3186 and by the Fundamental Research Funds for the Central Universities, and by Department of Public Security of Hunan Province Project.

REFERENCES:

[1] G. K. Manacher, “Production and Stabilization of Real-Time Task Schedules”, *Journal of Association for Computing Machinery*, Vol. 4, No. 3, 1967, pp. 439-465.

[2] LIU C, LAYLAND J, “Scheduling algorithms for multi-programming in hard real-time environment”, *Journal of ACM*, Vol. 20, No. 1, 1973, pp. 46-61.

[3] Aloysius L. Mok, Deji Chen. “A multiframe model for real-time task”, Proceedings of the 17th Real-Time System Symposium, IEEE Computer Society Press, Dec 4-6, 1996, pp. 22-29.

[4] Aloysius L. Mok, Deji Chen. “A multiframe model for real-time task”, *IEEE Transaction on Software Engineering*, Vol. 23, No. 10, 1997, pp. 635-645.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[5] Hiroaki Takada, Ken Sakamura. "Schedulability of Generalized Multiframe Task Sets under Static Priority Assignment", Proceedings of the 4th International Workshop on Real-Time Computing Systems and Application, IEEE Computer Society Press, Oct 1-3, 1997, pp. 80-86.

[6] Ching-Chih Jason Han. "A Better Polynomial-Time Schedulability Test for Real-Time Multiframe Tasks", Proceedings of the IEEE Real-Time Systems Symposium, IEEE Computer Society Press, Dec 2-4, 1998, pp. 104-113.

[7] S. K. Baruah, D. Chen, S. Gorinsky, A. "Mok. Generalized multiframe tasks", *Real-Time Systems*, Vol. 17, No. 1, 1999, pp. 5-22.

[8] Wan-Chen Lu, Kwei-Jay Lin, Hsin-Wen Wei et al. "New Schedulability Conditions for Real-Time Multiframe Tasks", Proceedings of the 19th Euromicro Conference on Real-Time Systems, IEEE Computer Society Press, July 4-6, 2007, pp. 39-50.

[9] A. Zuhily, A. Burns. "Exact Scheduling Analysis of Accumulatively Monotonic Multiframe Tasks Subjected to Release Jitter and Arbitrary Deadlines", Proceedings of 13th IEEE International Conference on Emerging Technologies and Factory Automation, IEEE Computer Society, Sept 15-18, 2008, pp. 600-607.

[10] S. K. Baruah. "Partitioning real-time tasks among heterogeneous multiprocessors", Proceedings of the International Conference on Parallel Processing, IEEE Computer Society Press, Aug 15-18, 2004, pp. 467-474.

[11] Renfa Li, Yan Liu, Cheng Xu. "A Survey of Task Scheduling Research Progress on Multiprocessor System-on-Chip", *Journal of Computer Research and Development*, Vol. 45, No. 9, 2008, pp. 1620-1629.

[12] QIAO Ying, WANG Hong-an, DAI Guo-zhong. "Developing a New Dynamic Scheduling Algorithm for Real-Time Multiprocessor", *Journal of Software*, Vol. 13, No. 1, 2002, pp. 51-58.

[13] WANG Kun, QIAO Ying, WANG Hong-an, FANG Ting, ZOU Bing, AND DAI Guo-zhong. "Study of a dynamic scheduling algorithm for real-time heterogeneous system", *Journal of Computer Research and Development*, Vol. 39, No. 6, 2002, pp. 725-732.

[14] Qiao Ying, Zou Bing. "Design and Evaluation of an Algorithm for Integrated Dynamic Scheduling in Real-time Heterogeneous System", *Journal of Software*, Vol. 13, No. 12, 2002, pp. 2251-2258.

[15] HU ANG Wen-Guang and YU Shi-Qi. "Feasibility analysis of periodic multiframe tasks", *Journal of Computer Research and Development*, Vol. 38, No. 2, 2001, pp. 240-245.

[16] HU ANG Wen-Guang. "An improved real-time task model-periodic multiframe model", *Journal of Computer Research and Development*, Vol. 38, No. 2, 2001, pp. 234-239.

[17] BIN Xue-lian, JIN Shi-yao, YANG Yu-mei. "Analysis of the Response Time of a Periodic Multi-Frame Task Model", *Computer Engineering & Science*, Vol. 25, No. 6, 2003, pp. 104-107.