

Minimum and Maximum Utilization Bounds for Multiprocessor RM Scheduling

J.M. López, J.L. Díaz, D.F. García
University of Oviedo
Department of Computer Science
Campus de Viesques, 33204 Gijón, Spain
{chechu,jdiaz,daniel}@atc.uniovi.es

Abstract

This paper deals with the problem of finding utilization bounds for multiprocessor Rate Monotonic scheduling with partitioning. The minimum and maximum utilization bounds among all the reasonable allocation algorithms are calculated. We prove that the utilization bound associated with the reasonable allocation heuristic Worst Fit (WF) is equal to that minimum. In addition, we prove that the utilization bound associated with the heuristics First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) is equal to the maximum, of value $(n + 1)(2^{1/2} - 1)$, where n is the number of processors.

1. Introduction

Multiprocessor scheduling is a challenging problem in the real-time systems theory. There are basically two strategies when dealing with this problem: partitioning strategies and global strategies [11].

In a partitioning strategy, once a task is allocated to a processor, it is executed exclusively on that processor. In a global strategy, any instance of a task can be executed on any processor, or even be pre-empted and moved to a different processor, before it is completed.

From a theoretical point of view, global strategies provide in general higher schedulability than partitioning strategies. However, partitioning strategies have several advantages over global strategies. Firstly, the scheduling overhead associated with partitioning strategies is lower than the overhead associated with global strategies. Secondly, partitioning strategies allow us to apply well known uniprocessor scheduling algorithms to each processor. Furthermore, Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduling, which are optimal uniprocessor scheduling algorithms [7], perform poorly when extended to global multiprocessor scheduling. The utilization bound associated

with global RM or EDF multiprocessor scheduling is not higher than one, for any number of processors [2].

In this paper, we follow the partitioning strategy. Tasks are allocated to processors, and are pre-emptively scheduled on each processor according to fixed priorities chosen following the RM criterion. Thus, the allocation algorithm is the only degree of freedom of the system.

Finding the optimal allocation algorithm is not practical, as the problem is NP-hard in the strong sense [5].

In the literature, one can find simple allocation heuristics [1, 2, 5] and complex heuristics based on branch-and-bound [12], or simulated annealing [14] techniques. In this paper, we focus on simple allocation heuristics.

Two different approaches were followed in the literature to establish the schedulability associated with a given allocation heuristic: simulation approaches, and theoretical approaches.

In the simulation approach, task sets are randomly generated. Next, the average number of processors required to allocate task sets of given total utilization is obtained. Uniprocessor exact tests [13], or uniprocessor sufficient tests [6] are commonly used to decide whether a given task set fits into the processors. Nevertheless, simulation results should be considered carefully, since randomly generated task sets may not be representative of those that appear in practice.

The traditional theoretical approach focuses on the calculation of the metric (N_{AA}/N_{OPT}) , for pairs of *uniprocessor scheduling algorithm-allocation algorithm* [1, 2, 3, 4, 5, 11]. This metric gives the relationship between the number of processors required to schedule a task set using a given allocation heuristic AA, and the number of processors required using the optimal allocation algorithm. This metric is useful in order to compare different allocation algorithms, but not to perform schedulability tests. There are several reasons for this. Firstly, N_{OPT} can not be calculated in polynomial time. Secondly, even if N_{OPT} were known, the utilization bound derived from the metric would be too pessimistic [10].

A new theoretical approach consists of calculating the

utilization bounds associated with pairs of *scheduling algorithm-allocation algorithm*, analogous to those known for uniprocessors [7]. This approach has several interesting features. Firstly, it allows us to carry out fast schedulability tests. Secondly, it allows us to quantify the influence of certain parameters, such as the number of processors, on schedulability. The major disadvantage of this approach is the sufficient character of the associated schedulability tests. This approach was followed in [10] to obtain the upper and lower limits, given by (1), on the utilization bound U_{wc}^{RM-FF} , for multiprocessor RM scheduling with FF allocation.

$$n(2^{1/2} - 1) < U_{wc}^{RM-FF}(n) \leq (n + 1)/(1 + 2^{1/(n+1)}) \quad (1)$$

where n is the number of processors. The tight utilization bound for multiprocessor RM-FF scheduling, given by (2), is presented in [9].

$$U_{wc}^{RM-FF}(m, n, \alpha) = (n - 1)(2^{1/(\beta_{RM} + 1)} - 1)\beta_{RM} + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) \quad (2)$$

where α is the maximum reachable utilization factor, and

$$\beta_{RM} = \lfloor 1/\log_2(\alpha + 1) \rfloor$$

An analogous utilization bound for EDF-FF scheduling, given by (3), was presented in [8]

$$U_{wc}^{EDF-FF}(n, \alpha) = \frac{n\beta_{EDF} + 1}{\beta_{EDF} + 1} \quad (3)$$

where $\beta_{EDF} = \lfloor 1/\alpha \rfloor$.

Our paper makes the following theoretical contributions to the multiprocessor schedulability analysis:

- The calculation of the minimum and maximum utilization bounds among all the reasonable allocation algorithms for multiprocessor RM scheduling.
- The proof that no allocation algorithm exists, reasonable or not, with an associated utilization bound higher than the previous maximum.
- The expression of the common utilization bound associated with the simple heuristics FFD and BFD [5], which coincides with the previous maximum. Thus, not even the optimal allocation algorithm can guarantee a higher utilization bound than that associated with FFD or BFD allocation.

Reasonable allocation algorithms are those which fail to allocate a task only when there is no processor in the system with sufficient free capacity to hold the task [8]. The idea of restricting the study to reasonable allocation algorithms is

to exclude theoretically possible, but impractical allocation algorithms.

The rest of the paper is organized as follows. Section 2 defines the computational system we deal with. The minimum and maximum utilization bounds for multiprocessor RM scheduling with reasonable allocation are provided in Section 3. Sections 4 and 5 provide the expressions of the utilization bounds using WF and Reasonable Allocation Decreasing (RAD) heuristics respectively. Both expressions are analyzed in Section 6. Finally, Section 7 presents our conclusions.

2. System definition

The task set consists of m independent periodic tasks $\{\tau_1, \dots, \tau_m\}$, of computation times $\{C_1, \dots, C_m\}$, periods $\{T_1, \dots, T_m\}$, and hard deadlines equal to the task periods. The utilization factor, u_i , of any task, τ_i , defined as $u_i = C_i/T_i$, is assumed to be $0 < u_i \leq \alpha \leq 1$, where α is the maximum reachable utilization factor.

Tasks are allocated to an array of n identical processors $\{P_1, \dots, P_n\}$. Once a task is allocated to a processor it is executed only on that processor. Within each processor tasks are pre-emptively scheduled using fixed priorities assigned according to the RM criterion. Allocation is carried out using reasonable allocation algorithms [8], represented by RA. A reasonable allocation algorithm is one which fails to allocate a task only when there is no processor in the system which can hold the task.

Whether a task fits into a processor depends on the uniprocessor scheduling algorithm and the schedulability condition. In this paper, we use the schedulability condition based on utilizations proposed in [7] for RM scheduling. Thus, a task of utilization factor u_i fits into processor P_j if

$$(m_j + 1)(2^{1/(m_j + 1)} - 1) - U_j \geq u_i$$

where U_j is the total utilization of the tasks previously allocated to processor P_j , and m_j is the number of these tasks.

Using the same schedulability condition, a reasonable allocation algorithm is one which fails to allocate a task of utilization factor u_i to a multiprocessor made up of n processors, only when

$$(m_j + 1)(2^{1/(m_j + 1)} - 1) - U_j < u_i \quad (4)$$

for all $j = 1, \dots, n$.

A class within the reasonable allocation algorithms is also considered in this paper. This class, termed Reasonable Allocation Decreasing (RAD), is made up of all the reasonable allocation algorithms which order the tasks by decreasing utilization factors before performing a sequential allocation. After the ordering,

$$u_1 \geq u_2 \geq \dots \geq u_m$$

This corresponds to the intuitive idea of allocating the “biggest” tasks first. After the ordering, task τ_1 is allocated first, next task τ_2 , and so on until task τ_m .

For example, the algorithms FFD and BFD, obtained through the ordering of the tasks by decreasing utilization factors before performing the FF and BF allocation respectively, belong to the class RAD.

3. Minimum and maximum utilization bounds

The utilization bound U_{wc}^{RM-RA} , associated with any reasonable allocation algorithm RA and multiprocessor RM scheduling, is in the interval $[L_{RM}, H_{RM}]$. This interval is defined as follows:

$$L_{RM} = \min_{RA} U_{wc}^{RM-RA}$$

$$H_{RM} = \max_{RA} U_{wc}^{RM-RA}$$

The calculation of the interval will allow us to know beforehand what is the worst and the best utilization bound we can expect from any reasonable allocation algorithm.

Before calculating the expressions of L_{RM} and H_{RM} , it is necessary to introduce the parameter β_{RM} . This parameter is analogous to the parameter β_{EDF} defined in [8] for EDF multiprocessor scheduling.

β_{RM} is the maximum number of tasks of utilization factor α which fit into one processor. β_{RM} can be expressed as a function of α , as proved in [9].

$$\beta_{RM} = \lfloor 1/\log_2(\alpha + 1) \rfloor$$

Any multiprocessor made up of n processors can allocate at least $n\beta_{RM}$ tasks of arbitrary utilization factor (less than or equal to α). Thus, any task set fulfilling $m \leq n\beta_{RM}$ is schedulable using RM scheduling together with any reasonable allocation algorithm. Henceforth, we will assume $m > n\beta_{RM}$, as otherwise there would be no point in obtaining the utilization bounds.

Theorem 1 provides a lower limit on the utilization bound associated with any reasonable allocation algorithm and RM scheduling. Section 4 provides an upper limit on the utilization bound for the reasonable allocation algorithm WF, which is equal to the lower limit, and therefore is also equal to L_{RM} .

Theorem 2 provides an upper limit on the utilization bound associated with any allocation algorithm, reasonable or not, and RM scheduling. Section 5 provides a lower limit on the utilization bound associated with all the reasonable allocation algorithms in the class RAD, which is equal to the upper limit, and therefore is also equal to H_{RM} .

Next, Lemma 1 is proved. Lemma 1 is necessary to prove Theorem 1.

Lemma 1 *Let $\{r_1, \dots, r_n\}$ be a set of n positive integers such that $\sum_{j=1}^n r_j = M$. Let $g(r_1, \dots, r_n) = \sum_{j=1}^n r_j(2^{1/r_j} - 1)$. It follows that*

$$g(r_1, \dots, r_n) \geq (M - \lfloor M/n \rfloor n) \lceil M/n \rceil \left(2^{\lceil M/n \rceil} - 1\right) \\ + (n - M + \lfloor M/n \rfloor n) \lfloor M/n \rfloor \left(2^{\lfloor M/n \rfloor} - 1\right)$$

Proof:

We will prove that the minimum of function g is obtained when M is quasi-equitably divided among $\{r_1, \dots, r_n\}$. When M is not a multiple of n , the quasi-equitable distribution of M produces $(M - \lfloor M/n \rfloor n)$ terms r_j of value $\lceil M/n \rceil$, and $(n - M + \lfloor M/n \rfloor n)$ terms r_j of value $\lfloor M/n \rfloor$, i.e, one unit less. The quasi-equitable distribution is strictly equitable when m is a multiple of n . Thus, a distribution is quasi-equitable if, and only if, $|\hat{r}_j - \hat{r}_k| \leq 1$ for all j, k in $1, \dots, n$.

We will prove by contradiction that the vector $\{\hat{r}_1, \dots, \hat{r}_n\}$, giving the minimum of function g , fulfils $|\hat{r}_j - \hat{r}_k| \leq 1$ for all j, k in $1, \dots, n$.

Let us suppose that two terms \hat{r}_j and \hat{r}_k exist, such that $|\hat{r}_j - \hat{r}_k| > 1$ with $\hat{r}_j > \hat{r}_k$, i.e, $(\hat{r}_j - \hat{r}_k) > 1$. Terms j^{th} and k^{th} within the expression of g accumulate the value

$$\hat{r}_j(2^{1/\hat{r}_j} - 1) + \hat{r}_k(2^{1/\hat{r}_k} - 1)$$

Decrementing \hat{r}_j one unit and incrementing \hat{r}_k this unit, the restriction $\sum_{j=1}^n \hat{r}_j = M$ is fulfilled, and the j^{th} and k^{th} terms in the expression of g change to accumulate the value

$$(\hat{r}_j - 1)(2^{1/(\hat{r}_j - 1)} - 1) + (\hat{r}_k + 1)(2^{1/(\hat{r}_k + 1)} - 1)$$

Let $f(x) = x(2^{1/x} - 1)$. For $x > 0$ $f'(x) < 0$, and $f''(x) > 0$. Thus, function $f(x) = x(2^{1/x} - 1)$ decreases more and more slowly as x increases. Applying this result to our problem gives

$$\hat{r}_k(2^{1/\hat{r}_k} - 1) - (\hat{r}_k + 1)(2^{1/(\hat{r}_k + 1)} - 1) > \\ (\hat{r}_j - 1)(2^{1/(\hat{r}_j - 1)} - 1) - \hat{r}_j(2^{1/\hat{r}_j} - 1)$$

and therefore,

$$(\hat{r}_j - 1)(2^{1/(\hat{r}_j - 1)} - 1) + (\hat{r}_k + 1)(2^{1/(\hat{r}_k + 1)} - 1) < \\ \hat{r}_j(2^{1/\hat{r}_j} - 1) + \hat{r}_k(2^{1/\hat{r}_k} - 1)$$

which is a contradiction, because changing \hat{r}_j and \hat{r}_k into $(\hat{r}_j - 1)$ and $(\hat{r}_k + 1)$ gives a value of g less than the minimum. Therefore,

$$g(r_1, \dots, r_n) \geq (M - \lfloor M/n \rfloor n) \lceil M/n \rceil \left(2^{\lceil M/n \rceil} - 1\right) \\ + (n - M + \lfloor M/n \rfloor n) \lfloor M/n \rfloor \left(2^{\lfloor M/n \rfloor} - 1\right)$$

■

Theorem 1 Let RA be a reasonable allocation algorithm. If $m > n\beta_{RM}$ then

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_a U_a + n_b U_b - (n-1)\alpha$$

where

$$\begin{aligned} n_a &= m + n - 1 - \left\lfloor \frac{m+n-1}{n} \right\rfloor n \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Proof:

Let $\{\tau_1, \dots, \tau_m\}$ be a set of n tasks which does not fit into the multiprocessor. There are tasks of the set which are allocated to processors, and tasks which are not allocated. Let us change the indexes in the set, so that the tasks which were not allocated have the last indexes in the set. Let τ_k be the first task in the set (after the change of indexes) which was not allocated to any processor. Since the allocation algorithm is reasonable, from (4) we get

$$(m_j + 1)(2^{1/(m_j+1)} - 1) - U_j < u_k \quad (5)$$

for all $j = 1, \dots, n$, where U_j is the total utilization of the tasks previously allocated to processor P_j , m_j is the number of these tasks, and u_k is the utilization factor of task τ_k .

The total utilization of the whole set, U , fulfils

$$U = \sum_{i=1}^m u_i \geq \sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (6)$$

From (5) we get

$$\begin{aligned} \sum_{j=1}^n U_j &> \sum_{j=1}^n \left((m_j + 1)(2^{1/(m_j+1)} - 1) - u_k \right) = \\ &= \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - nu_k \end{aligned}$$

Substituting this inequality into (6)

$$U > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n-1)u_k$$

From the system definition, all the utilization factors are less than or equal to α , so $u_k \leq \alpha$ and

$$U > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n-1)\alpha$$

One constraint of the m_j values is that $\sum_{j=1}^n m_j = (k-1)$. This constraint is totally equivalent to the constraint $\sum_{j=1}^n (m_j + 1) = (k+n-1)$. Bearing this last constraint in mind, defining $r_j = (m_j + 1)$, $M = (k+n-1)$, $g(r_1, \dots, r_n) = \sum_{j=1}^n r_j (2^{1/r_j} - 1)$ and applying Lemma 1

$$\begin{aligned} U &> \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n-1)\alpha \geq \\ &= \left(k+n-1 - \left\lfloor \frac{k+n-1}{n} \right\rfloor n \right) \left\lfloor \frac{k+n-1}{n} \right\rfloor \left(2^{\frac{1}{\lceil \frac{k+n-1}{n} \rceil}} - 1 \right) \\ &+ \left(1-k + \left\lfloor \frac{k+n-1}{n} \right\rfloor n \right) \left\lfloor \frac{k+n-1}{n} \right\rfloor \left(2^{\frac{1}{\lfloor \frac{k+n-1}{n} \rfloor}} - 1 \right) \\ &- (n-1)\alpha \end{aligned} \quad (7)$$

The right-hand term in (7) decreases as k increases, because any increment of k raises the value of $M = (k+n-1)$ to distribute quasi-equitably among all the processors. Index k is in the interval $[1, m]$, since τ_k is a task of the set of m tasks. Therefore, for $k = m$ we obtain the minimum of the right-hand term in (7). Hence, from (7) and considering the definitions of n_a, n_b, U_a and U_b we get

$$U > n_a U_a + n_b U_b - (n-1)\alpha$$

Any task set which does not fit into the processors fulfils the previous expression. Consequently, any task set of total utilization less than or equal to $n_a U_a + n_b U_b - (n-1)\alpha$ fits into the processors, and $U_{wc}^{RM-RA}(m, n, \alpha) \geq n_a U_a + n_b U_b - (n-1)\alpha$ ■

We will prove that $U_{wc}^{RM-WF}(m, n, \alpha \leq \ln 2) \leq n_a U_a + n_b U_b - (n-1)\alpha$ in section 4. Since WF is a reasonable allocation algorithm we can state that

$$L_{RM}(m, n, \alpha \leq \ln 2) = n_a U_a + n_b U_b - (n-1)\alpha$$

Next, we provide an intuitive idea about what n_a, n_b, U_a and U_b represent. This will be useful in the proof of Theorem 3 in Section 4. After dividing $(m-1)$ tasks quasi-equitably among n processors, there are n_a processors with $\lceil (m-1)/n \rceil$ tasks, and $n_b = n - n_a$ processors with one less task, i.e., $\lfloor (m-1)/n \rfloor$ tasks. U_a is the uniprocessor utilization bound for each of the n_a processors after receiving one more task. Likewise, U_b is the uniprocessor utilization bound for each of the n_b processors after receiving one more task.

Theorem 2 provides an upper limit on the utilization bound associated with any allocation algorithm, reasonable or not, and RM scheduling. The proof is analogous to that shown in [8] for multiprocessor EDF scheduling.

Theorem 2 Let AA be an arbitrary allocation algorithm. If $m > n\beta_{RM}$ then

$$U_{wc}^{RM-AA} \leq (n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

We will prove that a set of m tasks $\{\tau_1, \dots, \tau_m\}$ exists, with utilization factors $0 < u_i \leq \alpha$ for all $i = 1, \dots, m$, and total utilization $(n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$, with $\epsilon \rightarrow 0^+$, which does not fit into n processors using any allocation algorithm and RM scheduling on each processor.

We will construct this set of m tasks composed of two subsets: a first subset with $(m - n\beta_{RM} - 1)$ tasks, and a second subset with $(n\beta_{RM} + 1)$ tasks.

All the tasks of the first subset have the same utilization factor of value

$$u_i = \frac{\epsilon}{m}$$

where $i = 1, \dots, (m - n\beta_{RM} - 1)$.

All the tasks of the second subset have the same utilization factor of value

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m}$$

where $i = (m - n\beta_{RM}), \dots, m$.

It can be seen that the total utilization of the whole task set is $(n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$.

Firstly, it is necessary to prove that the utilization factors of both subsets are valid, i.e, $0 < u_i \leq \alpha$ for all $i = 1, \dots, m$.

Check of the utilization factors of the first subset. By choosing a small enough value for ϵ , we obtain $0 < u_i = \frac{\epsilon}{m} \leq \alpha$.

Check of the utilization factors of the second subset. By definition of β_{RM} , $(\beta_{RM} + 1)$ tasks of utilization factor α do not fit into one processor, therefore $(\beta_{RM} + 1)\alpha > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$, and

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1) \quad (8)$$

It is always possible to find one real number between two real numbers. Hence, a positive value ϵ exists such that

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m} = u_i$$

which proves that the utilization factors of the second subset are less than α when $\epsilon \rightarrow 0^+$. In addition, the utilization factors of the second subset are obviously greater than zero.

From the above results, we conclude that the proposed task set is valid. Below, we prove that it does not fit into n processors, using RM scheduling and any allocation algorithm.

There are $(n\beta_{RM} + 1)$ tasks in the second subset. Hence, at least one processor of the n available should allocate

$(\beta_{RM} + 1)$ or more of these tasks. However, no processor can allocate $(\beta_{RM} + 1)$ or more tasks of the second subset, since $(\beta_{RM} + 1)$ of these tasks together have a utilization over Liu & Layland's bound for RM [7].

$$(\beta_{RM} + 1) \left((2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m} \right) > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

We conclude that the proposed task set of total utilization $U = (n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$ does not fit into n processors when $\epsilon \rightarrow 0^+$, so the utilization bound U_{wc}^{RM-AA} must be less than or equal to $(n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$.

NOTE: the tasks of the first subset are necessary in the proof only to fulfil the restriction of having m tasks. ■

4. Utilization bound for Worst Fit allocation

This section shows that the allocation algorithm termed Worst Fit (WF) is the worst reasonable allocation algorithm in terms of utilization bound for multiprocessor RM scheduling.

The WF algorithm allocates each task to the processor with the highest remaining capacity among all the processors with sufficient capacity to hold the task. Tasks are allocated one by one following the sequence $\{\tau_1, \dots, \tau_k\}$. Using Liu & Layland's schedulability condition for RM scheduling, the remaining capacity of processor P_j is given by the expression $(m_j + 1)(2^{1/(m_j+1)} - 1) - U_j$.

Next, Theorem 3 gives an upper limit on the utilization bound for WF allocation and RM scheduling. This upper limit coincides with the lower limit provided by Theorem 1 for any reasonable allocation algorithm, and therefore with the utilization bound for WF allocation, given by Corollary 1.

The terms n_a , n_b , U_a and U_b in the statement of Theorem 3 have been defined in Theorem 3. The reader should refer to the intuitive description of these parameters given after the proof of Theorem 1, in order to better understand the proof of Theorem 3. .

Theorem 3 If $m > n\beta_{RM}$

$$U_{wc}^{RM-WF}(m, n, \alpha \leq \ln 2) \leq n_a U_a + n_b U_b - (n - 1)\alpha$$

Proof:

We will prove the existence of a set of m tasks, $\{\tau_1, \dots, \tau_m\}$, of utilization factors less than or equal to α , and total utilization

$$n_a U_a + n_b U_b - (n - 1)\alpha + \epsilon$$

with $\epsilon \rightarrow 0^+$, which does not fit into the processors using the allocation algorithm WF. The set of m tasks is

built as follows, strictly in the order indicated. There are $\lfloor (m-1)/n \rfloor$ subsets of $n = (n_a + n_b)$ tasks each. All these subsets are made up of n_a tasks of utilization factor

$$u_a = \frac{U_a - \alpha}{\lceil (m-1)/n \rceil} + \frac{\epsilon}{m-1}$$

followed by n_b tasks of utilization factor

$$u_b = \frac{U_b - \alpha}{\lfloor (m-1)/n \rfloor} + \frac{\epsilon}{m-1}$$

Following the previous $\lfloor (m-1)/n \rfloor$ subsets, there are n_a tasks of utilization factor u_a . Finally, there is the last task, τ_m , of utilization factor α .

It can be seen that the whole task set is made up of m tasks, and the total utilization is

$$n_a U_a + n_b U_b - (n-1)\alpha + \epsilon$$

Firstly, it is necessary to prove that the utilization factors of all the tasks are valid, i.e, $0 < u_i \leq \alpha$ for $i = 1, \dots, n$.

For $x > 0$, function $x(2^{1/x} - 1)$ decreases as x increases. Therefore, $U_b \geq U_a > \ln 2 \geq \alpha$, and the utilization factor of all the tasks is higher than zero. The utilization factor of the last task is α , and therefore it is less than or equal to α . We have to prove that $u_a \leq \alpha$ and $u_b \leq \alpha$ in order to prove that the proposed task set is valid. It is sufficient to prove that $u_b \leq \alpha$, as $u_a \leq u_b$.

Substituting the value of U_b in the definition of u_b

$$u_b = \frac{\lfloor \frac{m+n-1}{n} \rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - \alpha}{\lfloor (m-1)/n \rfloor} + \frac{\epsilon}{m-1} \quad (9)$$

From the hypothesis of the Theorem $m > n\beta_{RM}$. Since β_{RM} is an integer, $(m-1) \geq n\beta_{RM}$, $\lfloor (m-1)/n \rfloor \geq \beta_{RM}$, and $\lfloor (m+n-1)/n \rfloor \geq (\beta_{RM} + 1)$. In addition, $\alpha > (2^{1/(\beta_{RM}+1)} - 1)$ (see (8)). For $x > 0$, function $x(2^{1/x} - 1)$ decreases as x increases. Hence,

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \leq (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

and

$$u_b < (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m-1}$$

As was indicated previously, $(2^{1/(\beta_{RM}+1)} - 1) < \alpha$, so by making ϵ close to zero we get $u_b < \alpha$.

Next, we will prove that the task set does not fit into the multiprocessor. The first $(m-1)$ tasks are allocated by the WF heuristic as indicated in Figure 1. Numbers within parenthesis in Figure 1 represent task indexes. The proof of the allocation pattern shown in Figure 1 is too large to be included in this paper. As a result of the allocation of the

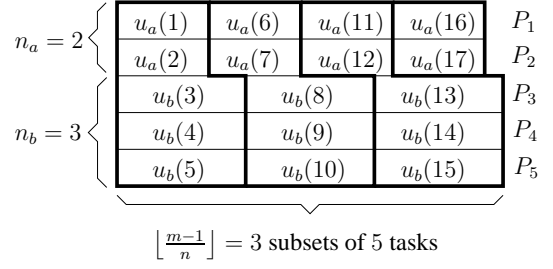


Figure 1. Example of allocation of the first $(m-1)$ tasks in Theorem 3, for $m = 18$ and $n = 5$.

first $(m-1)$ tasks, the first n_a processors hold $\lceil (m-1)/n \rceil$ tasks of utilization factors u_a . These processors may hold one additional task of utilization factor

$$U_a - \lceil (m-1)/n \rceil u_a = \alpha - \epsilon \lceil (m-1)/n \rceil / (m-1)$$

Therefore, the last task of utilization factor α does not fit into any of these processors. Nor can the remaining n_b processors hold the last task because at most they can hold one additional task of utilization factor

$$U_b - \lfloor (m-1)/n \rfloor u_b = \alpha - \epsilon \lfloor (m-1)/n \rfloor / (m-1)$$

We conclude that the proposed task set of total utilization $n_a U_a + n_b U_b - (n-1)\alpha$ does not fit into n processors when $\epsilon \rightarrow 0^+$, so the utilization bound $U_{wc}^{RM-WF}(m, n, \alpha)$ must be less than or equal to $n_a U_a + n_b U_b - (n-1)\alpha$. ■

Corollary 1 provides the utilization bound for WF allocation and RM multiprocessor scheduling.

Corollary 1 If $m > n\beta_{RM}$

$$U_{wc}^{RM-WF}(m, n, \alpha \leq \ln 2) = n_a U_a + n_b U_b - (n-1)\alpha$$

Proof:

The proof is direct from Theorem 1 and Theorem 3. ■

5. Utilization bound for RAD allocation

The Reasonable Allocation Decreasing (RAD) algorithms are reasonable allocation algorithms fulfilling the following conditions:

- Tasks are ordered by decreasing utilization factors before making the allocation, i.e, $u_1 \geq u_2 \geq \dots \geq u_m$.
- Tasks are allocated sequentially, That is, task τ_1 is allocated first, next task τ_2 , and so on until task τ_m .

The heuristics FFD and BFD, belong to this class.

Theorem 4 provides a lower limit on the utilization bound associated with the class of RAD allocation algorithms under RM scheduling. This lower limit coincides with the upper limit on the utilization bound associated with any allocation algorithm under RM scheduling. Therefore, both bounds also coincide with the utilization bound associated with any RAD allocation algorithm and RM scheduling, as given by Corollary 2. Furthermore, RAD allocation algorithms are optimal from the point of view of the utilization bound, since no allocation algorithm exists guaranteeing a higher utilization bound. Thus, the utilization bound associated with the optimal allocation algorithm coincides with the utilization bound associated with the RAD algorithms.

Theorem 4 *If $m > n\beta_{RM}$ then*

$$U_{wc}^{RM-RAD}(m, n > 1, \alpha) \geq (n\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Proof:

Let $\{\tau_1, \dots, \tau_m\}$ be a set of n tasks which does not fit into the multiprocessor. Let τ_k be the first task in the set which does not fit into the multiprocessor. Since RAD allocation algorithms are reasonable, from (4) we get

$$(m_j + 1)(2^{1/(m_j+1)} - 1) - U_j < u_k \quad (10)$$

for all $j = 1, \dots, n$, where U_j is the total utilization of the tasks allocated to processor P_j , m_j is the number of these tasks, and u_k is the utilization factor of task τ_k . The total utilization of the first k tasks fulfils

$$\sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (11)$$

From (10) and (11) we get

$$\sum_{i=1}^k u_i > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n-1)u_k \quad (12)$$

Tasks were ordered in decreasing utilization factors before carrying out the allocation, so

$$u_k \leq \frac{\sum_{i=1}^k u_i}{k}$$

Substituting this inequality into (12) and finding $\sum_{i=1}^k u_i$

$$\sum_{i=1}^k u_i > \frac{k}{k+n-1} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1)$$

The total utilization of the first k tasks is less than or equal to the total utilization of the whole task set. Thus,

$$U > \frac{k}{k+n-1} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) \quad (13)$$

Variables k and $\{m_1, \dots, m_n\}$ can not take any value. In particular, $m_j \geq \beta_{RM}$ for all j in $1, \dots, n$. Otherwise, task τ_k would fit into the processors, because each processor can allocate at least β_{RM} tasks, and this would contradict the hypothesis which states that τ_k does not fit into the processors. In addition,

$$k = 1 + \sum_{j=1}^n m_j$$

and so $k \geq (n\beta_{RM} + 1)$. Let $g(k, n)$ be the minimum of function $\sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1)$ under the constraints $k = 1 + \sum_{j=1}^n m_j$ and $m_j > 0$, which can be obtained from Lemma 1.

$$U > \frac{k}{k+n-1} g(k, n)$$

If $n > 1$ the right term of the inequality is minimized¹ for $k = (\beta_{RM}n + 1)$ and $m_j = \beta_{RM}$ for all $j = 1, \dots, n$. The proof is not provided here for the sake of brevity. Moving these values to (13)

$$\begin{aligned} U &> \frac{n\beta_{RM} + 1}{n\beta_{RM} + 1 + n - 1} \sum_{j=1}^n (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1) \\ &= (n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1) \end{aligned}$$

A necessary condition to be fulfilled by the total utilization of any task set which does not fit into the n processors is

$$U > (n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1)$$

In other words, any task set of total utilization less than or equal to

$$(n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1)$$

fits into the n processors. Thus, we finally conclude

$$U_{wc}^{RM-RAD}(m, n > 1, \alpha) \geq (n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1) \quad \blacksquare$$

Corollary 2 provides the utilization bound for RAD allocation and RM multiprocessor scheduling.

Corollary 2 *If $m > n\beta_{RM}$*

$$U_{wc}^{RM-RAD}(m, n, \alpha) = \begin{cases} m(2^{1/m} - 1) & \text{if } n = 1 \\ (n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1) & \text{if } n > 1 \end{cases}$$

Proof:

The proof is direct from Theorem 2, Theorem 4, and Liu & Layland's bound for uniprocessor RM scheduling. \blacksquare

¹If $n = 1$ it is minimized for $k = m$, and we will finally obtain the utilization bound $m(2^{1/m} - 1)$.

6. Analysis of the theoretical results

In this section, we analyze the functions

$$L_{RM}(m, n, \alpha \leq \ln 2) = n_a U_a + n_b U_b - (n-1)\alpha$$

and

$$H_{RM}(m, n, \alpha) = \begin{cases} m(2^{1/m} - 1) & \text{if } n = 1 \\ (n\beta_{RM} + 1)(2^{1/(n\beta_{RM}+1)} - 1) & \text{if } n > 1 \end{cases}$$

$L_{RM}(m, n, \alpha)$ is the minimum utilization bound for multiprocessor RM scheduling evaluated among all the reasonable allocation algorithms. This minimum coincides with the utilization bound for RM-WF.

It can be seen that for the uniprocessor case $L_{RM}(m, n=1, \alpha \leq \ln 2) = m(2^{1/m} - 1)$, and so it coincides with Liu & Layland's bound.

The expression of L_{RM} provided in the paper has one theoretical limitation, α can not be higher than $\ln 2$. However, this is not a practical limitation. The utilization bound can be obtained by subtracting $\epsilon \rightarrow 0^+$ from the minimum utilization evaluated among all the task sets which do not fit into the multiprocessor. Thus, if $\alpha > \ln 2$, all the task sets fulfilling $u_i \leq \ln 2$ also fulfill $u_i \leq \alpha$, and therefore $L_{RM}(m, n, \alpha > \ln 2) \leq L_{RM}(m, n, \ln 2)$. From Figure 2, we can extrapolate the value of $L_{RM}(m, n, \ln 2)$ to be low, which makes in this case the utilization bound of little importance.

One of the difficulties of dealing with the function $L_{RM}(m, n, \alpha \leq \ln 2)$ is its complexity. Nevertheless, for $n > 1$ and $m > n$ we have found the relation

$$0 \leq \frac{n_a U_a + n_b U_b - (m+n-1)(2^{n/(m+n-1)} - 1)}{n} \leq 0'0054$$

Therefore, from a practical point of view

$$L_{RM}(m, n, \alpha \leq \ln 2) \approx (m+n-1)(2^{n/(m+n-1)} - 1) - (n-1)\alpha$$

In addition, it can be seen that for the uniprocessor case this expression gives Liu & Layland's bound.

Figure 2 depicts the function $L_{RM}(m, n, \alpha \leq \ln 2)$ as a function of the numbers of processors, for different values of α . In spite of n being an integer, it is represented as a continuous function with the aim of improving its visualization. The representation has been normalized by dividing L_{RM} by the number of processors, in order to show the average degree of utilization of the processors. For each value of α two different curves have been plotted. The top curve is associated with the minimum number of tasks, i.e.,

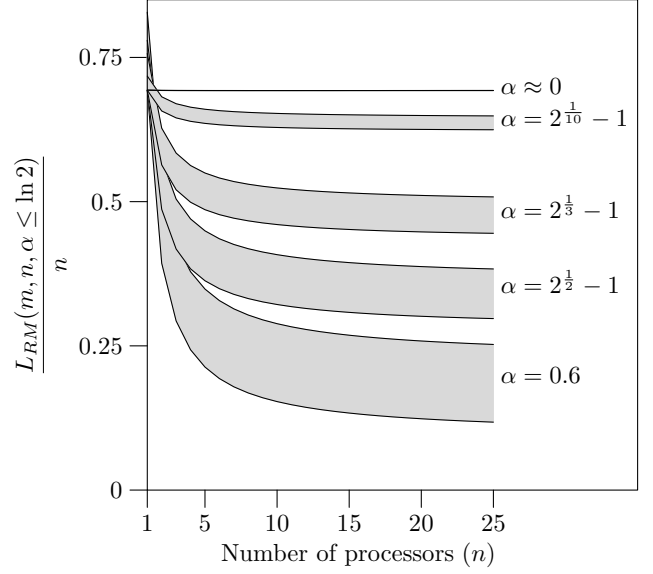


Figure 2. Plot of $L_{RM}(m, n, \alpha \leq \ln 2)$.

$m = (n\beta_{RM} + 1)$. The bottom curve is associated with the maximum number of tasks, i.e., $m \rightarrow \infty$. The shaded area between the top and bottom curves corresponds to values of m in $(n\beta_{RM} + 1, \infty)$.

For high values of α the utilization bound L_{RM} is too small. However, as α nears 0, the utilization bound becomes close to $n \ln 2$. In this case, the multiprocessor behaves approximately like a uniprocessor n times faster.

$H_{RM}(m, n, \alpha)$ is the maximum of the utilization bounds for multiprocessor RM scheduling evaluated among all the reasonable allocation algorithms. This maximum coincides with the utilization bound for RM-RAD. The class of RAD allocation algorithms include allocation algorithms such as FFD and BFD. All of them have the same utilization bound. In addition, no allocation algorithm exists which guarantees a utilization bound higher than $H_{RM}(m, n, \alpha)$. From this point of view, RAD allocation algorithms are optimal. Figure 3 depicts the function $H_{RM}(m, n > 1, \alpha)$ as a function of the numbers of processors for different values of α . In spite of n being an integer, here as well it is represented as a continuous function to improve its visualization. This function has not been represented for $n = 1$, since in this case the utilization bound coincides with the well-known utilization bound $m(2^{1/m} - 1)$. For $n > 1$ it does not depend on the number of tasks. The representation has been normalized by dividing H_{RM} by the number of processors. Each curve in Figure corresponds to a different value of β_{RM} , and therefore to a different value of α .

For $\alpha > (2^{1/2} - 1)$ we obtain $\beta_{RM} = 1$, and $H_{RM}(m, n > 1, \alpha) = (n+1)(2^{1/2} - 1)$. The addition of one processor increments the value of H_{RM} by

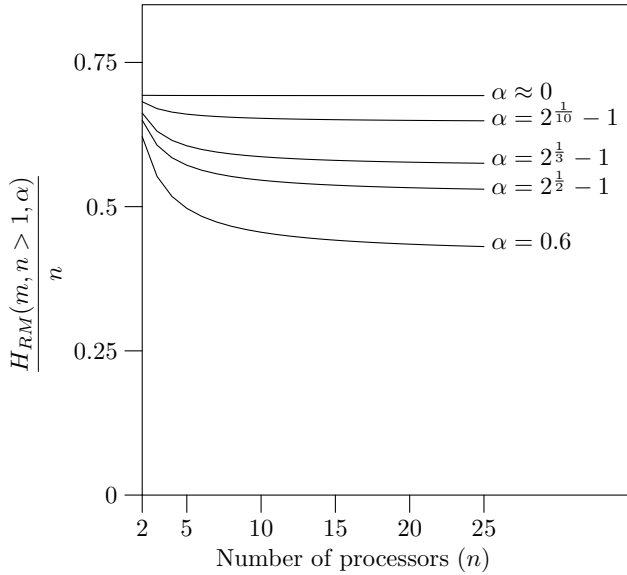


Figure 3. Plot of $H_{RM}(m, n > 1, \alpha)$.

$(2^{1/2} - 1) \approx 0.41$. When $\alpha \rightarrow 0$ then $\beta_{RM} \rightarrow \infty$ and $H(m, n > 1, \alpha \rightarrow 0) = n \ln 2$. That is, the multiprocessor behaves like an ideal uniprocessor n times faster.

For example, the utilization bound associated with RM scheduling and FFD allocation in a multiprocessor made up of two processors is $3(2^{1/2} - 1) \approx 1.24$, about 0.62 per processor. If tasks have utilization factors less than or equal to $(2^{1/2} - 1) \approx 0.41$ then $\beta_{RM} = 2$. In this case, the utilization bound for FFD allocation takes the value $5(2^{1/3} - 1) \approx 1.30$, about 0.65 per processor, close to the ideal $\ln 2 \approx 0.69$.

7. Conclusions

We have obtained the tight interval in which the utilization bound associated with any reasonable allocation algorithm is found. Since practical allocation algorithms are reasonable, the interval obtained is of wide applicability.

The WF algorithm was proved to be the worst reasonable allocation algorithm in terms of the utilization bound for multiprocessor RM scheduling. Its utilization bound is a function of the number of processors, n , the number of tasks, m , and a parameter α that takes the “task size” into account. For high values of α the bound is less than or equal to $2(2^{1/2} - 1)$, while for low values of α it is close to $n \ln 2$.

In addition, algorithms such as FFD and BFD were proved to be optimal in terms of the utilization bound. The utilization bound associated with these heuristics is close to the ideal $n \ln 2$ when the multiprocessor is made up two processors, or when the utilization factors of the tasks are

small. The utilization bound associated to these algorithms does not depend on the number of tasks.

The task set model of the paper considered periodic and independent tasks. Nevertheless, it is also possible to analyze the schedulability of task sets including also sporadic and aperiodic tasks, whenever the aperiodic tasks are served by aperiodic servers whose worst-case behaviour can be assimilated to periodic tasks. This is the case of the Polling Server and the Sporadic Server.

References

- [1] A. Burchard, J. Liebeherr, Y. Oh, and S. Son. New strategies for assigning real-time tasks to multiprocessor systems. *IEEE Transactions on Computers*, 44(12), 1995.
- [2] S. Dall and C. Liu. On a real-time scheduling problem. *Operations Research*, 6(1):127–140, 1978.
- [3] S. Davari and S. Dhall. On a periodic real-time task allocation problem. In *Annual international Conference on Systems Sciences*, pages 133–141, 1986.
- [4] S. Davari and S. Dhall. An on line algorithm for real time tasks allocation. In *Proceedings of the IEEE Real-Time Systems Symposium*, pages 194–200, 1986.
- [5] M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freeman, New York, 1979.
- [6] S. Lauzac, R. Melhem, and D. Mossé. An efficient rms admission control and its application to multiprocessor scheduling. In *Proceedings of the International Parallel Processing Symposium*, pages 511–518, 1998.
- [7] C. L. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.
- [8] J. López, J. Díaz, M. García, and D. García. Worst-case utilization bound for edf scheduling on real-time multiprocessor systems. In *Proceedings of the Euromicro Workshop on Real-Time Systems*, pages 25–33, 2000.
- [9] J. López, J. Díaz, M. García, and D. García. Utilization bounds for multiprocessor rm scheduling. *To appear in the Real-Time Systems Journal*, 2001.
- [10] D. Oh and T. Baker. Utilization bounds for n-processor rate monotone scheduling with static processor assignment. *Real-Time Systems*, 15(2):183–193, 1998.
- [11] Y. Oh and S. Son. Allocating fixed-priority periodic tasks on multiprocessor systems. *Real-Time Systems*, 9(3):207–239, 1995.
- [12] D. Peng, K. Shin, and T. Abdelzaher. Assignment and scheduling communicating periodic tasks in distributed real-time systems. *Transactions on Software Engineering*, 23(12):745–758, 1997.
- [13] S. Sáez, J. Vila, and A. Crespo. Using exact feasibility tests for allocating real-time tasks in multiprocessor systems. In *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, pages 53–60, 1998.
- [14] K. Tindell, A. Burns, and A. Wellings. Allocating hard real-time tasks (an np-hard problem made easy). *Real-Time Systems*, 4(2):145–165, 1992.