International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 6, Number 1 (2013), pp. 145-151
© International Research Publication House
http://www.irphouse.com

# Traffic Pattern Analysis in Multiprocessor System

**Kedilaya Shreeganesh B.[1] and Dr. G.M. Patil[2]**

*[1]Professor, Srinivas School of Engg., Mukka, Mangalore, India.*
*E-mail: bskedilaya@yahoo.com*
*[2]Principal, Basava Kalyan Engg. College, Bidar, India.*
*E-mail: gurulingappampatil@yahoo.com*

**Abstract**

It is very important to manage resources in multiprocessor system used in real time systems so that deadline violation is minimized. Fluid traffic flow conditions results in disruptive scheduling. In this paper we enumerate the why's and how's of disruptions in scheduling and present a traffic centric scheduling mechanism that will account for shift to traffic centric system level design paradigm [1]. Despite a plethora of studies and commercial solutions proposed, scheduling is still considered as one of the scientific areas where substantial improvements can be gained by the development and application of new research approaches [2]. Most of the existing scheduling algorithms used in multiprocessor systems do not consider the availability of resources imposed by multiclass applications. To overcome this, in this paper, we analyze the scheduling problem for multiclass applications running in multiprocessor systems. In an effort to explore this issue, each class of task is monitored at schedule queue. Whether a task can be executed depends on satisfying delay performance requirements for both existing and new task flows. Acceptable task flow rate is a threshold for good delay performance and thus it is very important to accurately estimate the acceptable traffic pattern/task flow rate.

**Index Terms**: Scheduling, Resource allocation, Multiclass applications, Multiprocessor scheduling.

## Introduction

It is very important to determine the acceptable task flow rate in multiprocessor systems to determine the delay performance and QOS (Quality Of Service) for various types of task flows [3]. Task scheduling on multiprocessor system has been a source of challenging problems for researchers. Multiprocessor systems are used in parallel computations and effective scheduling is required for parallel program execution to get high QOS. The scheduling has to be done in a way that can minimize total time of program execution with regard to task execution time and also maximize processor utilization [4]. Since this is NP-complete, many researchers have proposed different heuristic algorithms [5] [6] [7].

For a homogeneous multiprocessor platform where all processors are identical, assigning tasks to the processors requires solving Bin packing problem 8].

There are two important goals in traffic pattern analysis. The first one is to ensure of estimated QOS for multiclass flows and the other one is to achieve high processor utilization. To achieve this, we propose a new scheme. Delay is considered as a Quality parameter because real time flows are more sensitive to delay than loss. If multiclass tasks flows are allowed, without considering the present traffic rate, QOS may be degraded.

In order to estimate the acceptable task flow rate which is a threshold for tasks control, decision is made for each class of flow by comparing the peak rate of flow with the acceptable flow. We derive a equation for acceptable task flow by sending a Test Task (TT) and measuring number of Test Tasks executed over a period of time.

This scheme statistically satisfies the delay bound of controlled task flow while maintaining high resource utilization. Conservative resource allocation may satisfy delay bound but high resource utilization cannot be achieved. Satisfying these two goals is still a very challenging issue.

## Model Description and Problem formulation

Figure 1.System Model of Task Flow and Execution

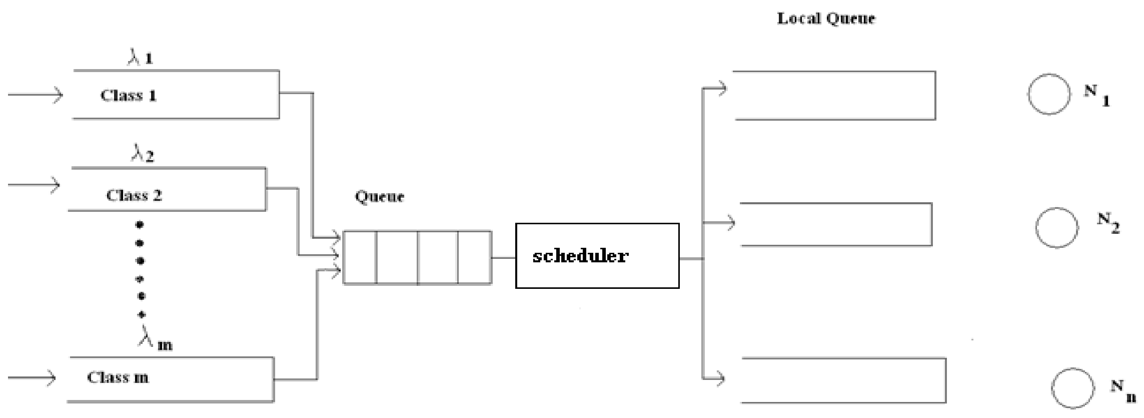


**Figure 1**: System Model of Task Flow and Execution.

Consider a system shown in Figure1. All arriving traffic with same QOS requirement is treated as the same class. Acceptable task flow is managed according to classes. Here we consider only delay violation probability as QOS. Let $T_{Rj}^{1}$ denote acceptable traffic of the jth class. Let $d_j$ be the delay bound $T_j$ be the threshold for delay violation probability. $D_j(0)$ is a random variable representing current delay and $D_j(T_R)$ is a random variable representing delay which the total traffic of the class j experiences after accepting with a rate $T_R$. Then the acceptable task flow rate $T_{Rj}^{1}$ is

$T_{Rj}^{1}=\max\{T_R{:}P(D_j(T_R)>d_j)\leq T_j)\}$ (1)

$T_{Rj}^{1}$ is the maximum task flow rate that can be accepted additionally satisfying delay constraint.

In order to support QOS for new task flow while maintaining QOS for the existing tasks, a controller should be introduced which determines whether new task flow can be allowed or not. The processor cannot determine the amount of traffic involved in new flow. We assume that the admission decision shall be taken on the peak traffic flow rate $r_p$ of a flow. Peak rate $r_p$ is the only parameter used in the controller. Each traffic flow is monitored so that instantaneous traffic rate can be maintained less than or equal to rate $r_p$. If a request from new flow with peak rate of $r_p$ arrives, controller can accept the flow as jth class if the following condition is satisfied.

$r_p<T_{Rj}^{1}$ (2)

Then the delay constraint can be satisfied for both the existing and new traffic. The controller can quickly decide whether to admit new flow or not since the method adopted is very simple.

In this scheme, controller need not calculate the acceptable task flow whenever a new task arrives. The controller sends TT to the multiprocessor queue and calculates the acceptable task flow rate $T_{Rj}^{1}$ in advance.

Consider a queuing system with First Come First Serve (FCFS) policy. The service rate is $\mu$ and the arrival rate of TT is $\lambda$. Let $\bar{L}$ denote the average length of each task. For the Queuing system acceptable task rate $\mu_a=\mu\,(1-\rho)$ where $\rho = \frac{\lambda}{c}\bar{L}$.

This acceptable rate is the maximum spare service that processor can provide.

To estimate the acceptable task flow rate, test task is sent at regular intervals.

Assume that only one TT exists in the system. If a new TT is sent into the system, just at the departure of previous TT, then there exists only one TT at any time. Let $\bar{Y}$ [s,t] be the amount of TT served by the processor in the interval [s,t] when only one TT exists at any time. If the size of TT is L, then for G/G/1 queuing system.

$\underset{t\to\infty}{Lim\,E}\left[\left|\frac{\bar{Y}[s,t]}{t-s}-c(1-\rho)\right|\right]^{q}=0 \quad 0<q<\infty$ (3)

The service rate of TT is measure of acceptable traffic flow. This implies that service rate of TT can be used to estimate the acceptable task flow rate.

Let $w_t$, $s_t$ and $g_t$ denote waiting time, service time and propagation delay of the task.

Total delay $w_t+s_t+g_t$ (4)

The propagation delay $g_t$ is constant in (4) then the remaining term is queuing delay $w_t+s_t$. Then we can frame a path model consisting of delay component ($D_f$) & processors as shown in figure 2.

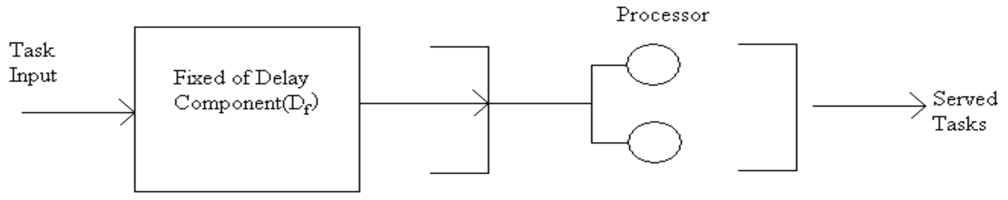


**Figure 2**: Multiprocessor model to estimate delay.

Suppose that testing task TT arrives at time $a_{tt}$ and departs from processor at $d_p$. Then TT arrives at processor at time $a_{tt}^P$

$a_{tt}^P=a_{tt+}D_{f.}$. When execution is complete it departs from the Queue and processor. The processor is continuously backlogged for k (k≥2) Test Task transmissions from j^th test task in the interval.

$[a_j^P, d_{j+k-1}]$ if $d_{j+m\geq}a_j{}^P{}_{+m+1}$ for all $0 \leq m \leq k–2$.

## Estimation of Acceptable Task Flow

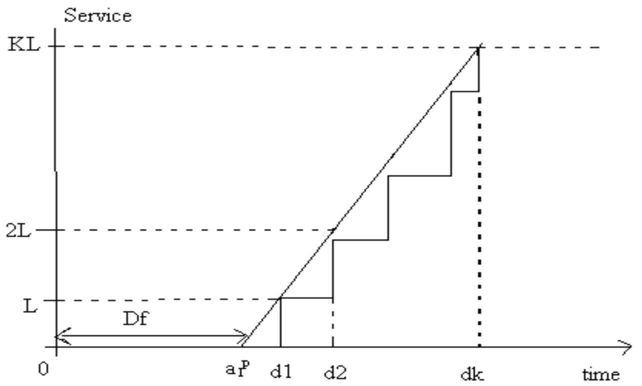
We send a estimated number of Test Tasks into multiprocessor Queue and monitor the execution of these Test Tasks in the interval $[a_1, d_N]$ monitoring period. If N Test Tasks are sent the acceptable flow rate in the interval $[a_1^P, d_N]$ can be estimated by $\frac{NL}{(d_N-a_1^p)}$

Where $a_1^P=a_1+D_f$ is the fixed delay for the current monitoring period.

Due to increased Task flow, let there be K Test tasks be present in the queue. The Measured Test Task Rate (MTTR) is given by

$$\text{MTTR}=\frac{\text{KL}}{\text{d}_\text{k}-\text{a}_1^\text{s}} = \frac{\text{KL}}{(\text{d}_\text{k}-\text{a}_1)-\text{D}_\text{f}}$$

$D_f$ is estimated during busy period.



Rate Increase Ratio (α).

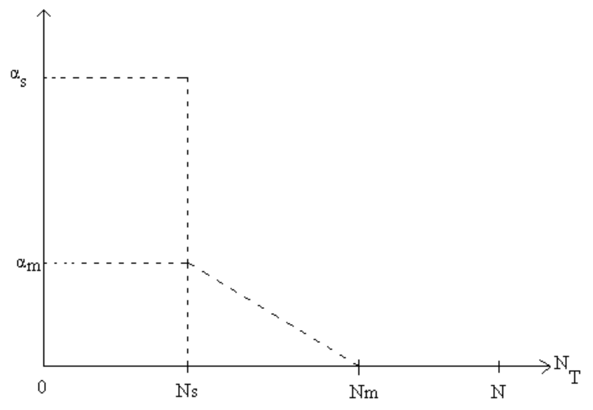


**Figure 3**: Example of Service Curve Figure 4.Rate Increase ratio curve.

Let $N_T$=max $N_T$ (i) where $N_T$ (i) is the number of Test Tasks belonging to the ith period. $N_T$ should be maintained within a reasonable range lower value of $N_T$ is due to less traffic and higher value of $N_T$ is due to increased traffic. Thus MTTR is a reliable estimate of acceptable task flow rate.

If $N_T>N_m$, then MTTR is large and task flow rate is fixed to MTTR.

If $N_T \leq N_s$, MTTR for this period may be inaccurate. Current flow rate is estimated by the flow rate when $N_T>N_s$.

If $N_s<N_T \leq N_m$ then MTTR is a reliable estimate of acceptable task rate.

## Numerical Results and Analysis

In this secton,we present simulation result implemented in C. Here peak traffic rate $r_p$ is estimated based on the number of Test Tasks executed. Simulation is done for schedule lengths varying from 1000 microSecs to 50000 microsecs. When $r_p$ is 0.5, average 50% of tasks offered will be executed satisfying delay QOS. When $r_p$ is 0.7, average 69 % of tasks will be executed satisfying delay QOS. When $r_p$ is 0.9, average 90 % of tasks will be executed satisfying Delay QOS as shown in Figure.5.Remaining tasks will not get processsor time to execute and will be lost. Figure 6. Shows the average waiting time for the tasks when $r_p$ is 0.5,0.7 &0.9.It is observed that waiting time increases when the peak traffic rate increases.

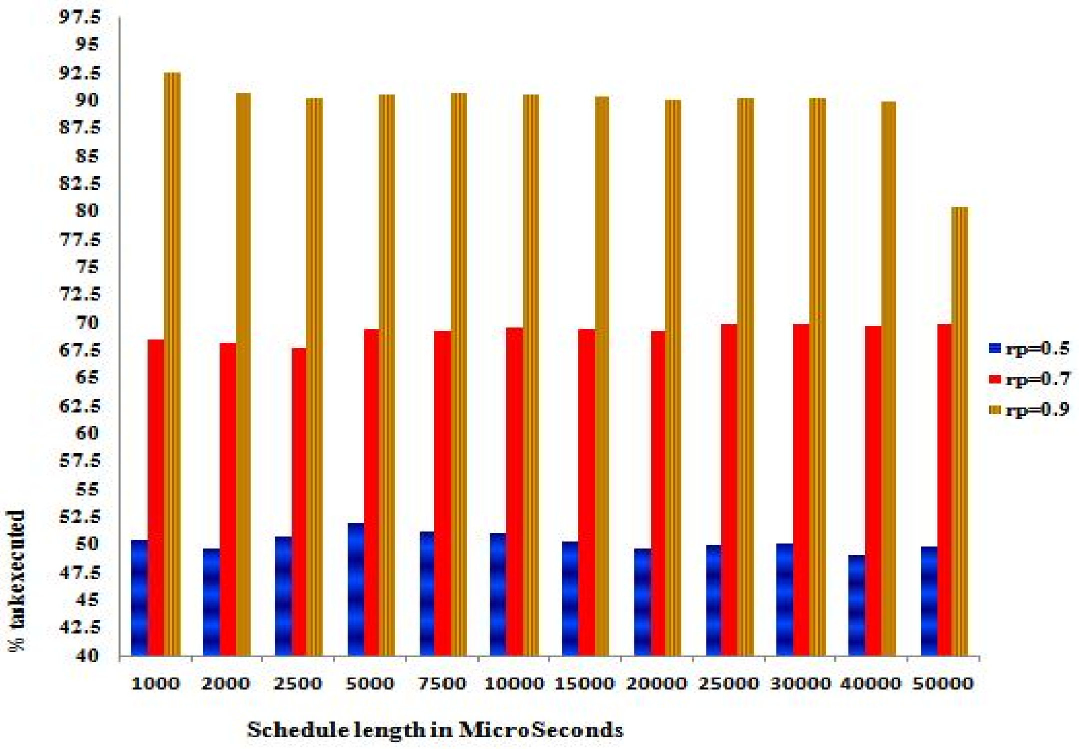


**Figure 5**: Task execution rate for varying peak traffic.

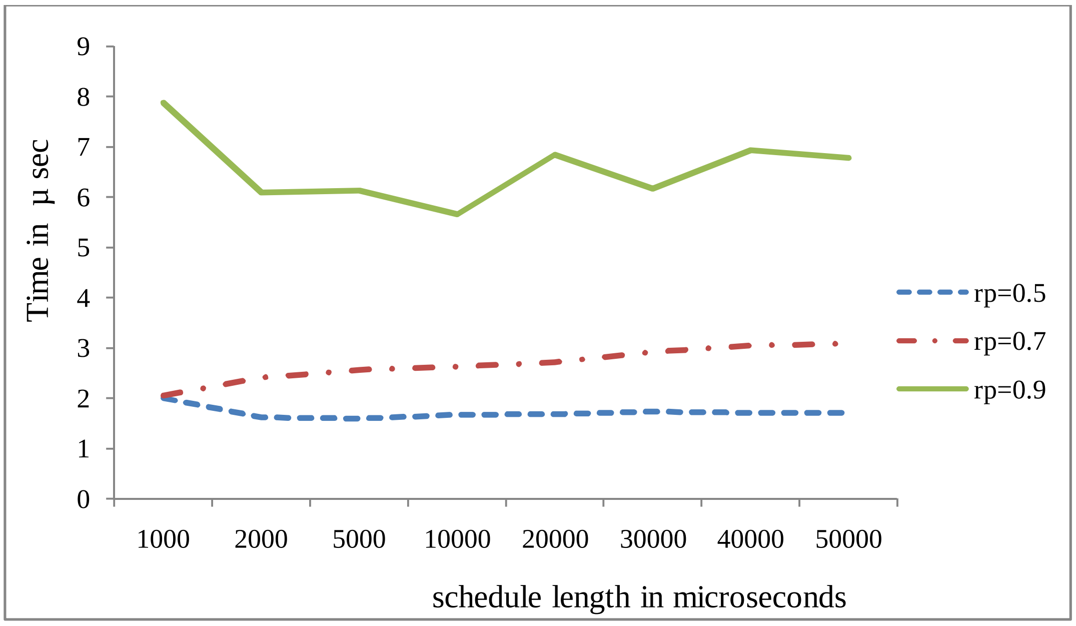


**Figure 6**: Average waiting time for varying peak traffic.

## Conclusion

The effect of traffic flow rate on scheduling is analysed. Waiting time of tasks increases with peak traffic rate. Above $r_p$=0.9, all additional tasks will be rejected. To accommodate these tasks a new traffic cenrtic scheduling scheme is to be framed .It is also possible to adopt any scheduling scheme on multiprocessor based on peak traffic rate.

## References

[1] Kai Nagel "Traffic Networks" ETH Zurich CH-8092 Zurich Switzerland 2002.

[2] A. Burchard, J. Liebeherr, Y. Oh. and S.H. Son "New Stragies for Assigning Real Time tasks to Multiprocessor Systems", IEEE Trans. on Computers, Vol. 44, No.12 Dec 1995.

[3] K. Kuchcinski, "Embeded system synthesis by Timing constraint solving" in in Proc. Int Symp. Syst. Synthesis, Sep 1997, pp. 50–57.

[4] N. Shenoy, A. Choudhary and P. Banergy, "An algorithm for synthesis of large time-Constrained heterogenious adaptive sytems" ACM Trans. Des. Autom. Electron. Syst., Vol. 6, No. 2, pp 207–225, Apr. 2001

[5] M. Woodside and G.G. Monforton, "Fast allocation of processes in distributed and parallel Systems," IEEE Trans. Parallel Distrib. Systems, Vol. 4, No. 2, pp.164–174, Feb. 1993.

[6] S. Lauzac, R. Melhem and D. Mosse, "An Improved Rate Monotonic Admission Control and its application", IEEE Trans. On computers. Vol.52, No. 3, March 2003.

[7] I. Rippol, A. Crespo, and A. Gracia-Fornes, "An optimal Algorithm for Scheduling Real Time Tasks in Dynamic Priority Preemptive systems"IEEE Trans. Software Eng., Vol. 23, No. 6, pp. 388–400, June 1995.

[8] Kedilaya Shreeganesh .B. and Dr. G.M. Patil, "Feasibility Analysis of RM Scheduling Under Fluid trafffic Flow Conditions" International Journal of Engineering Research and Technology (IJERT) ISSN:2278-0181, Vol. 2, Issue 1, pp. 1–5, Jan 2013.