Total Bandwidth Server

Adijat Ajoke Sulaimon
Embedded System
Hochschule Hamm Lippstadt
Email: Adijat-ajoke.sulaimon@stud.hshl.de

***Abstract*—As an essential commodity, time is an integral part of real-time systems. A real-time system consists of both periodic and aperiodic tasks in modern times. Periodic tasks have regular arrival times and hard deadlines; tasks are issued at methodical intervals, which means that there is the fixed time at which tasks are released. On the other hand, aperiodic tasks have irregular arrival times, jobs are released at irregular times, and have either soft or hard deadlines. Therefore, it is essential to use a server algorithm that will improve response time for both hard and soft deadlines. This paper focuses on a type of server algorithm that improves response time for periodic and aperiodic tasks. A total bandwidth server (TBS) is a server used for scheduling time for periodic and aperiodic tasks in real-time embedded systems. This paper aims to provide detailed insight and enumerate the importance of using TBS as a means of real-time scheduling systems. The research further provides the differences between TBS and other types of scheduling algorithms.**

I. Introduction

A Real Time System (RTS) deals with both periodic and aperiodic tasks where tasks must be executed at a specific time. To guarantee the schedulability of tasks to meet both it's hard and soft tasks, to prioritize hard real time tasks, it is crucial to use scheduling algorithm that supports both hard and soft deadlines. There are many of such algorithms but this study is focused on Total bandwidth server(TBS). The Total Bandwidth Server is a dynamic priority server that improves responsiveness by using a server to maintain background time unused by the periodic tasks. It has been merited to have utilization of up to 100 percent while also maintaining schedulability.

The rest of this paper is organized as follows: The prior research recounted in the next section. The methods and simulations are enumerated in Section 3. Section 4 contains Data and results. In Section 5, the methods, simulation and data gathered is discussed. This paper is concluded in Section 6 with several directions for the future work.

"A total bandwidth server only consumes budget when it Replenishment Rule: Initially, e s = 0 and d = 0 When an aperiodic job with execution time e arrives at time t to an empty aperiodic job queue Set d = max(d, t) + e/ũ s and e s = e

When the server completes the current aperiodic job, the job is removed from the queue then If the server is backlogged, set d = d + e/ũ s and e s = e

If the server is idle, do nothing Total bandwidth server is always ready for execution when backlogged and assigns at least fraction ũ s of the processor to a task for execution.

For example: ũ s = 0.25 T1 = (3, 0.5), T2 = (4, 1), T3 = (19, 4.5)"

II. Literature Review

Kiyofumi Tanaka in 2013 worked on using Total bandwidth server algorithm to shorten response time of soft tasks. He evaluated different dynamic-priority servers and decided on TBS because of it's good response time with moderate implementation complexity. He was able to prove by simulation, that the use of predictive execution times can shorten response times of aperiodic tasks.

Tanaka in another research conducted in 2017 used Adaptive Total Bandwidth Server(ATBS) and Enhanced Virtual release advancing(EVRA) to improve the responsiveness of TBS. the research proposes using predictive excution times and not worst case execution times so as to get shorter deadlines and reduce response times of both periodic and aperiodic tasks. He compared seven different methods by simulation and concluded that the combination of ATBS and EVRA reduced the run-time overhead by up to 60 percent.

S. Kato and N. Yamasaki, "Scheduling Aperiodic Tasks Using Total Bandwidth Server on Multiprocessors," 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008, pp. 82-89, doi: 10.1109/EUC.2008.28. This research addressed the (i) the scheduling of aperiodic tasks that can be dispatched to any processors when they arrive, and (ii) the scheduling of aperiodic tasks that must be executed on particular processors on which they arrive. The relevance of this paper is it's talk about aperiodic tasks, their arrival times and execution. It proposed ways to reduce processor time of aperiodic tasks in multiprocessor systems.

Scheduling Aperiodic Tasks in Dynamic Priority Systems by Marco Spuri and Giorgio Buttazzo Scuola Superiore SAnna presented five different new online algorithms, including Total Bandwidth Server, for servicing soft aperiodic requests in realtime systems and their scheduling using Earliest Deadline First (EDF) algorithm. It conclused that DPE, DSS, TBS, EDL, and IPE algorithms are the simplest algorithms possible.

Sprunt, B., Sha, L. Lehoczky, J. talked about scheduling aperiodic tasks with examples while mentioning different algorithm that uses EDF. They enumerate different theorem with proofs. They concluded that In terms of schedulability, a sporadic server can be regarded as a periodic task with the same execution time and period. It was also mentioned that the SS algorithm by using a deadline monotonic priority assignment instead of a rate monotonic priority assignment can guarantee tehir deadlines.

G. Fohler, T. Lennvall and G. Buttazzo, "Improved handling of soft aperiodic tasks in offline scheduled real-time systems using total bandwidth server proposed a method of using offline scheduling to handle complex constraints and reduce their complexity with Earliest Deadline First scheduling system.

Another research carried out in 2010 compared the Multiple total bandwidth server to Total bandwidth server and concluded that Total bandwidth server is superior in comparison to the former.

III. System Model and Total Bandwidth Server

Periodic tasks are tasks with regular tasks arrival time while aperiodic tasks have irregular task arrival time. Periodic tasks is commonly used to process sensor data and update the current state of the real-time system at regular intervals. Periodic tasks have hard deadlines and are generally used in signal processing and control. Aperiodic tasks are typically have soft deadlines and are used to handle the processing requirements of random events such as operator requests. Aperiodic tasks with hard deadlines are referred to as Sporadic tasks. Periodic tasks therefore have both hard and soft deadlines. A single inability to meet the deadline in a hard-real time system may lead to a complete system failure while in a soft real time system, is only referred to as performance degraded.

	Execution Time	Period	Priority
Task A	4	10	High
Task B	8	20	Low

Figure 1. A periodic task

The figure 1 above is an example of a periodic task set. Notice how task A has higher priority than task B.

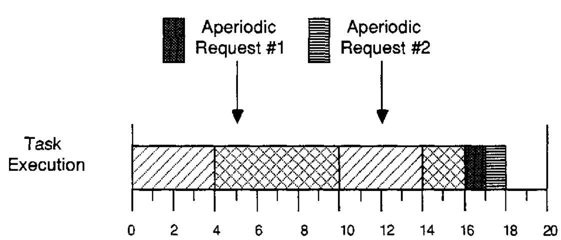

Figure 2. Example of an aperiodic task

Figure 2 is an example of an aperiodic task execution. This figure shows the task execution time from time 0 to time 20. There are two aperiodic task requests, one at time 5 and the other at time 12. Background service only occur when resource is idle in figure 2, therefore aperiodic task execution cannot begin until time 16, this means that the response time performance is poor. Even though the requests only need a unit of time each to complete, their requests are time units 12 and 6.

A. Dynamic-Priority Servers

The dynamic priority server is based on Earliest Deadline First (EDF). The scheduling is this type of server has higher bounds with utilization rate of 100%. Apart from Total Bandwidth server (TBS), there are other dynamic-priority servers. Examples are: Dynamic Priority Exchange, Dynamic Sporadic server, Constant Bandwidth Server and Earliest Deadline Late Server. The essence of these algorithms is to shorten the response times of aperiodic tasks while maintaining their effectiveness. These are properties of dynamic-priority server

- The tasks are schedulable if and only if $U_p + U_s \leq 1$.
- All aperiodic tasks J_i: i=1,...,m do not have deadlines.
- Each periodic tasks τ_i, has a period T_i, a computation time C_i, and a relative deadline D_i, equal to it's period.
- All periodic tasks τ_i: i=1,...,n have hard deadlines.
- All periodic tasks are activated concurrently at a time t=0 which means that the first instance of each periodic task has a request time of $r_i(0) = 0$
- Aperiodic tasks have regular arrival times and this makes their arrival time unknown.
- The worst case execution time of each aperiodic task is considered to be known at its arrival

B. Total Bandwidth Server

The Total Bandwidth Server (TBS) is a dynamic Scheduling algorithm used in real-time to execute aperiodic task. The total bandwidth of the server is applied toan aperiodic request as soon as possible when it enters the system, that is where the name Total Bandwidth Server is gotten from. It is designed primarily for Earliest deadline first (EDF) based systems to prioritize hard tasks.Total Bandwidth server has low overhead, has good response time and it's implementation is simple because it only requires a simple computation. Total bandwidth server improves the responsiveness of a real-time system by utilizing the unused background time by periodic tasks. As soon as the server backlogged, the server replenishes the server budget. An aperiodic task request r_k (Arrival time) usually comes with a deadline d_k

$dk = max(r_k, d_{k1}) + \frac{C_k}{U_s}$

The term, max$(rk, dk1)$, prevents bandwidths of successive aperiodic instances from overlapping with each other. k means k^{th} instance of the aperiodic tasks, $dk1$ is the absolute deadline of the k1th (previous) instance, Ck is required(worst) computation time of the k^{th} instance, and U_s is the CPU utilization by the server. It was proven that a task set is schedulable if and only if $U_p + U_s \leq 1$

TBS assigns the absolute deadline to an incoming job such that the utilization for this server is at most U_s.

In Figure 3 above, the arrival of the first aperiodic task is r_1 which is 6 and has a scheduled deadline of $d_1 = r_1 + \frac{C_k}{U_s} = 6 + \frac{1}{0.25} = 10$ This means that 10 is the earliest deadline in the system, the aperiodic task is carried out immediately. Just like the first task, the second task received deadline $d_2 = r_2 + \frac{C_k}{U_s} = 8 + \frac{2}{0.25} = 21$. This second task however did not service immediately because there is a task active at time T=13

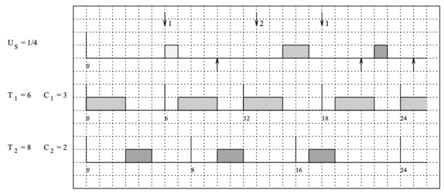

Figure 3. An example of a Total Bandwidth Server

that has a shorter deadline of 18. The last aperiodic request arrived at time T=18 with deadline $d_3 = max(r_3, d_2) + \frac{C_k}{U_s} = 21 + \frac{1}{0.25} = 25$ with a service time of t=22. There is no change for the third task while the response time of the first two tasks improved.

TBS Implementation is simple, deadlines just have to be assigned new request by monitoring the deadline assigned to the last aperiodic request $(dk - 1)$. The request will then be positioned into the queue that is ready and implemented as other periodic task.

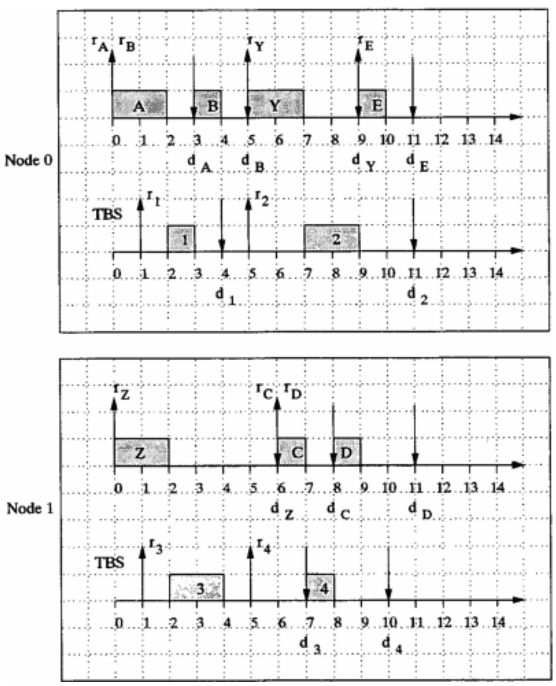

Figure 4. Schedule produce by EDF

In the integrated schedule produce by EDF shown in Figure 4, the offline algorithm reserve the available bandwidth $U_s = 1/3$. To efficiently handle the online aperiodic request, the Total Bandwidth Server is used. as shown in figure 5, an aperiodic task request J_1 with computation time C1=1 with an arrival time of $r_t = 1$ at Node 0. The TBS allocates a deadline $d_1 = r_1 + \frac{C_1}{U_s} = 4$. As a result of this, Task J1 is executed before task B, since d1¡dB.

Another request J_2 with a computation time C_2 and arrival time $r_t = 5$ on Node 0. The request has a deadline $d_2 = r_2 + \frac{C_2}{U_s} = 11$. The deadline collided with that of task E but because server has more priority, J_2 executes before task E.

A request J_3 with an arrival time $r_3 = 1$ and a computation time $C_3 = 2$ on Node 1. It has a deadline $d_3 = r_3 + \frac{C_3}{U_s} = 7$. Another request j_4 with an arrival time $r_4 = 5$ arrives before deadline d_3. In this scenario, the TBS rule applies. The rule states that the deadline has to be $d_4 = max(r_t, d_3) + \frac{C_2}{U_s} = 10$ because the bandwidth U_s has been assigned to J_3 till time d_3. This means that deadline d_4 is less than d_D and therefore will executes before task D.

C. Total Bandwidth Server VS Constant Utilization Server

IV. Data and Results

I have carefully considered the response times of aperiodic tasks and checked what happens to soft tasks when there is increase in offline load. TBS is the only one running when offline load is 0.

V. Discussion and Conclusion

In this paper, I have produced insight to dynamic priority scheduling algorithms with real time systems, especially Total Bandwidth Server. All scheduling algorithm mentioned use Earliest Deadline First. They all deal with both periodic and aperiodic tasks. I considered efficiency, cost,real world constraints like distribution and pressing deadlines. The importance and efficiency of TBS has been talked about in details with examples. I have shown how the total bandwidth server algorithm can be flexible, efficient, and handle overload.

REFERENCES

[1] G. Fohler, T. Lennvall and G. Buttazzo, "Improved handling of soft aperiodic tasks in offline scheduled real-time systems using total bandwidth server," ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597), 2001, pp. 151-157 vol.1, doi: 10.1109/ETFA.2001.996364.

[2] D. Duy and K. Tanaka, "An effective approach for improving responsiveness of Total Bandwidth Server," 2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), 2017, pp. 1-6, doi: 10.1109/ICTEmSys.2017.7958777.

[3] S. Kato and N. Yamasaki, "Scheduling Aperiodic Tasks Using Total Bandwidth Server on Multiprocessors," 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008, pp. 82-89, doi: 10.1109/EUC.2008.28.

[4] Tanaka, K. (2013). Adaptive Total Bandwidth Server: Using Predictive Execution Time. In: Schirner, G., Götz, M., Rettberg, A., Zanella, M.C., Rammig, F.J. (eds) Embedded Systems: Design, Analysis and Verification. IESS 2013. IFIP Advances in Information and Communication Technology, vol 403. Springer, Berlin, Heidelberg.

[5] Rajiv Bikram 2022. Total Bandwidth Server. Retrieved from https://benchpartner.com/total-bandwidth-server

[6] Scheduling Aperiodic Tasks in Dynamic Priority Systems Marco Spuri and Giorgio Buttazzo Scuola Superiore SAnna via Carducci 40, 56100 Pisa Italy Created: 07/05/2012, 10:24:40 Modified: 07/05/2012, 10:26:25

[7] Sprunt, B., Sha, L. and Lehoczky, J. Aperiodic task scheduling for Hard-Real-Time systems. Real-Time Syst 1, 27–60 (1989). https://doi.org/10.1007/BF02341920