Constant Bandwidth Server

A presentation on Real-Time Systems

Lochana Abhayawardana
lochana.abhayawardana@stud.hshl.de

Real Time systems and Constant Bandwidth Server

- What is the **constant Bandwidth Server (CBS)**

  A scheduling algorithm used in **real-time systems.**

- **Real-time systems** : A system that the respond should be guaranteed within a **specified timing constraint** or the system should meet the **specified deadline**.[1]

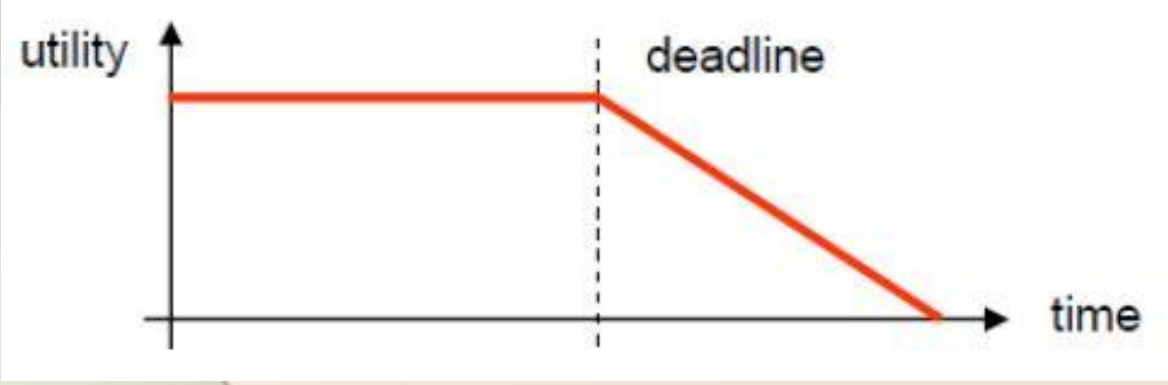
Difference between Soft and Hard real time systems

Soft RTS:

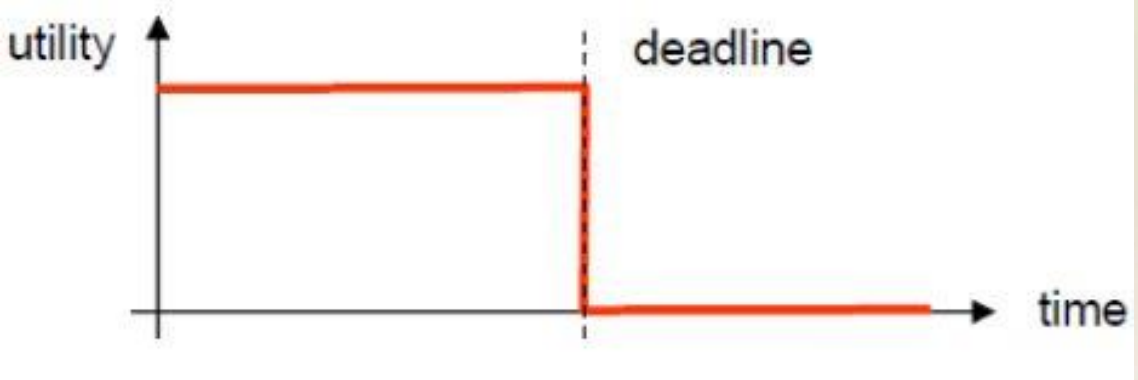
A failure to meet a specified deadline reduces the utility of the result, but **does not lead to a significant financial loss** (E.g., letter sorting machine) [2]

Hard RTS:

A failure to meet a specified deadline **can lead to catastrophic consequences** (e.g., a computer system [2], Flight control systems).



[2]



[2]

What is Constant Bandwidth Server

- Task of **CBS**:

To ensure **temporal isolation** and **effective resource allocation among tasks**.

- Explanation to Temporal Isolation:

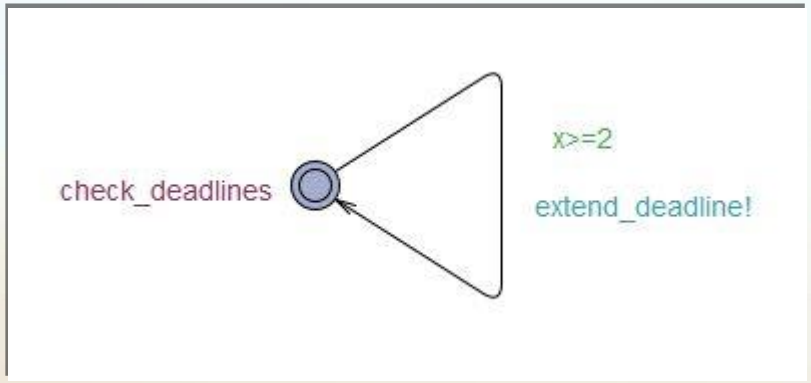
Temporal isolation in real-time systems allows the execution of software processes isolated from one another in the temporal domain[3]

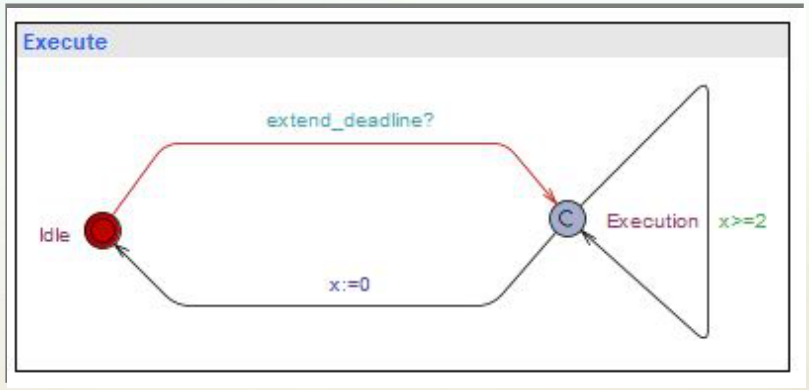
Or the ability to ensure different tasks of a system operates independently to each other in terms of timing.

RESOURCE RESERVATION IN DYNAMIC REAL-TIME SYSTEMS

Resource reservation:

- to assure that the task is executed within its allocated time limits.
- allocates a fraction of the processor bandwidth to each task considering each timing requirement

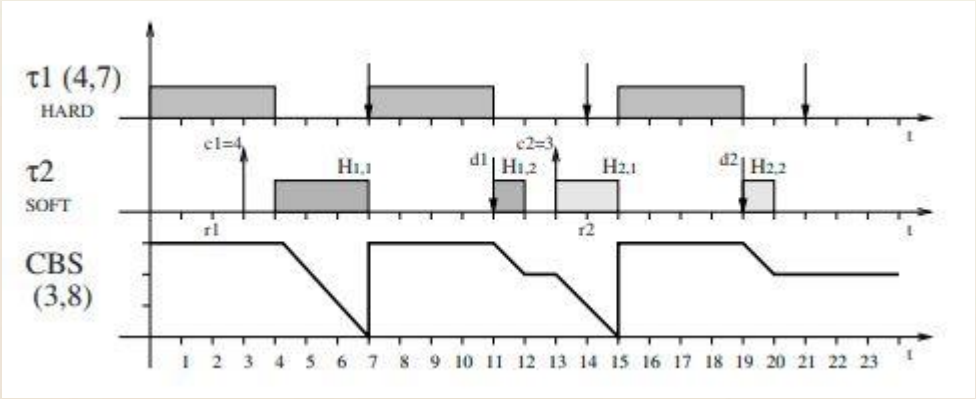




- A way to implement resource reservation is **to reserve a task-specific amount of CPU time at every interval**. If a task is assigned a pair (Qi and Pi), Where **Qi is the amount of CPU time allocated to a task** and **Pi is the interval** in which the task is executed. The task can execute for the Qi unit(Qi = 2 In figure) for **each Pi time, But if the task is not finished, It assigns another time Qi at the beginning of the next time period** and schedules it as a real-time task and the budget expires.
- **But the kernel then prevents each task from consuming more than its allocated bandwidth** to protect other tasks in the system.

EFFICIENT RECLAIMING IN RESERVATION-BASED REAL-TIME SYSTEMS WITH VARIABLE EXECUTION TIMES

- **Real Time systems with periodic and aperiodic tasks** efficiently utilizing **the spare time** is ensured using the DPS server to **reclaim spare time** from the periodic task and add it to the **corresponding aperiodic capacity**.
- Which allows for **immediate service of aperiodic requests**.



[4]

THE CBS ALGORITHM AND ITS IMPLEMENTATION

- When describing the CBS algorithm, let a_k **and** d_k **be the release time** and the **deadline of the** k^{th} chunk generated by the server
- Let **c and n be the actual server budget number of the pending requests in the server queue**.
- Variables are initialized as follows:

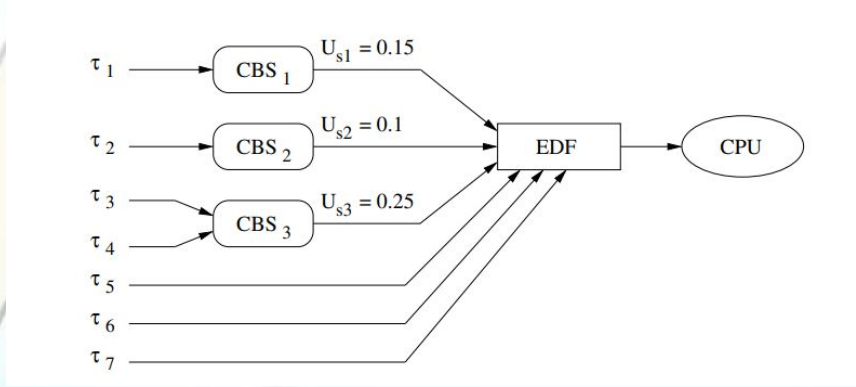
  d_0 = 0 , c = 0, n = 0, k = 0.
- The server behavior can be described by the algorithm shown below in Figure.

```
When job J_j arrives at time r_j
      enqueue the request in the server queue;
      n = n + 1;
      if (n == 1) /* (the server is idle) */
        if (r_j + (c / Q_s) * T_s >= d_k)
          /*---------------Rule 1---------------*/
          k = k + 1;
          a_k = r_j;
          d_k = a_k + T_s;
          c = Q_s;
        else
          /*---------------Rule 2---------------*/
          k = k + 1;
          a_k = r_j;
          d_k = d_{k-1};
          /* c remains unchanged */
When job J_j terminates
      dequeue J_j from the server queue;
      n = n - 1;
      if (n != 0) serve the next job in the queue with deadline d_k;
When job J_j executes for a time unit
      c = c - 1;
When (c == 0)
      /*---------------Rule 3---------------*/
      k = k + 1;
      a_k = actual_time();
      d_k = d_{k-1} + T_s;
      c = Q_s;
```

[4]

HANDLING OVERRUNS AND IMPLEMENTING RESOURCE RESERVATIONS WITH CBS

- Overrun conditions are caused **by tasks that execute more than expected or are activated more frequently than expected**.
- because the **task parameter is incorrectly estimated and calculated**
- If the overruns are not properly handled, **task overruns can cause critical issues** in real-time systems.
- resource reservations are used to handle this overruns.
- To implement temporal protection, each task T_i with variable computation time should be handled by a dedicated CBS with bandwidth U_{si} , that can not interfere with the rest of the tasks for more than U_{si} .



[4]

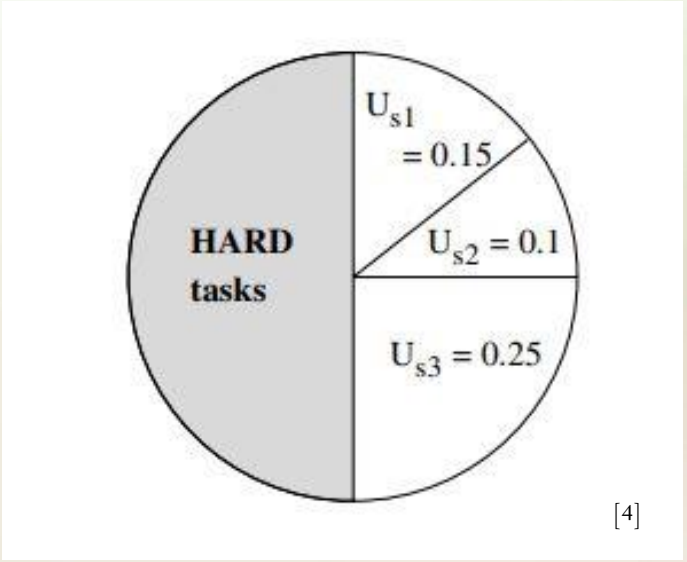
- Above Figure shows an example of two tasks (T_1 and) T_2 served by the two dedicated 1CBSs with a bandwidth of U_{s1} = 0.15 and U_{s2} = 0.1, a group of two tasks (T_3 , T_4) is handled by a single CBS with the bandwidth U_{s3} = 0.25, and three hard periodic tasks (T_5 , T_6 ,T_7) are directly scheduled by the EDF, without server intercession, because their execution times are not subject to larger variations. The example the total processor bandwidth is shared among the task as shown in Figure below.

- The properties of the CBS guarantee that the set of hard periodic tasks (with utilization Up) are schedulable by EDF only if,

$$U_p + U_{s1} + U_{s2} + U_{s3} \leq 1$$

If the above condition holds the set of **hard periodic tasks always use fifty percent of the processor** independent of the execution time of the other tasks. [4]

- The CBS can be **updated to enforce hard reservations by postponing the budget replenishment to the server deadline.**



[4]

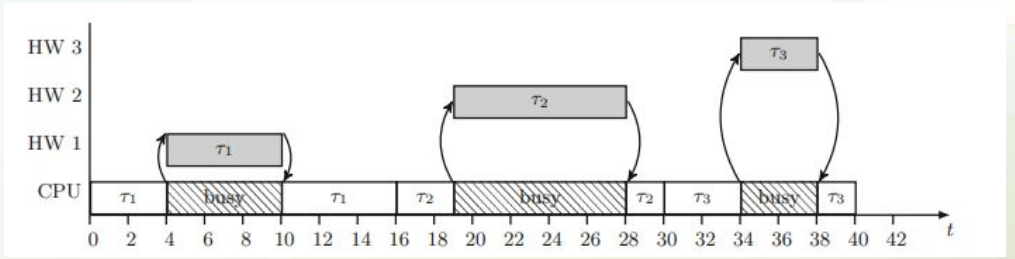
THE USE OF CBS FOR SERVING SELF-SUSPENDING TASKS

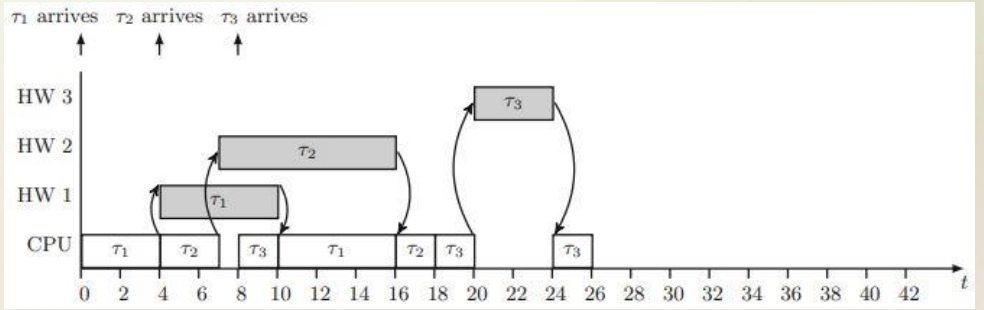
- **voluntarily suspend their execution** during their execution time due to **waiting for some activity to complete**, e.g., an accelerator to return data. However, this situation can cause substantial performance/schedulability degradation.

Examples of the self-suspending task system:

- **Hardware acceleration by using co-processors and computation offloading**.

1. **Busy Waiting**: The **software task does not give up its privileges on the processor** and must wait by spinning on the processor until the co-processor finishes the requested work
2. self-suspending task: This **allows freeing the processor so the processor can be used by the other ready tasks**. Therefore, even **single CPU systems can execute tasks simultaneously**. This arrangement is **called limited parallelism**, which **improves the performance of the CPU** by effectively utilizing the processor and the co-processors





H-CBS algorithm definition

Hard Constant Bandwidth Server (H-CBS)

- Rule 1: Each server is Idle with q = 0 and d = 0.

- Rule 2: When the server is Idle or a job receives at time t, a replenishment time is computed as

$$t_r = d(t) - q(t)/\alpha$$

If t is smaller than t_r , the server becomes Suspended and it remains suspended until time t_r , at **time t_r , the sever becomes Ready**, the budget is replenished to Q and

$d \leftarrow t_r + P$.

Otherwise, if the server becomes Ready, the budget is immediately replenished to Q and $d \leftarrow t + P$.

- Rule 3: **When q = 0, the server becomes suspended and suspended until time d**. At time d, the server becomes Ready, the budget becomes Q and the deadline is postponed to $d \leftarrow d + P$.
- Rule 4: **When the server does not have any pending work it turns to the Idle state, holding the current values of both budget q and deadline d.**

- Theorem 01: $\sum_{i=1}^{n} \frac{Q_i}{P_i} \leq 1$

The **H-CBS server can be used to achieve temporal isolation among a set of real-time tasks**. [5]

- Theorem 02:

Given sets of n **non-self-suspending tasks having an implicit deadline,** associates each task to an H-CBS server S_i. For each H-CBS server S_i, define $Q_i = C_i$ and $P_i = T_i = D_i$. This set of tasks is executed each upon a dedicated H-CBS and is schedulable with EDF if and only if the test of the previous Equation holds. [5]

- The advantage of associating each task to an H-CBS server is **enabling the protection from over-run tasks of the system**. In this way, **when each task is executed upon a reservation server, all the overruns are protected by the budget exhaustion mechanism that stops the execution of the task**.
- This way, only the task that is experiencing an overrun is affected in terms of stimulability, guaranteeing the deadlines of other tasks. [5]
- The advantage of associating each task to an H-CBS server is **enabling the protection from over-run tasks of the system**.
- In this way, when each task is executed upon a reservation server, **all the overruns are protected by the budget exhaustion mechanism that stops the execution of the task**. This way, only the task that is experiencing an overrun is affected in terms of stimulability, guaranteeing the deadlines of other tasks. [5]

COMPARISON OF CBS WITH OTHER SERVICE MECHANISMS

- Variable bandwidth Server (VBS)

VBS is an **extension of a constant-bandwidth server** where the **throughput and latency of process execution cannot be controlled to remain constant across different workloads** but also vary in time as long as the resulting bandwidth stays below the given bandwidth cap. [6]

1) **Period Adjustment**: VBS provides **dynamic period adjustment** while the **CBS provides constant bandwidth for each process**. In comparison, **VBS provides better resource utilization and adaptation to dynamic workload demands**.
2) **Responsiveness**: VBS provides responsiveness to the changing resource demands and allocates bandwidth dynamically. **This process allows to obtain more resources when needed and release resources when they are not required.** While CBS is using a constant bandwidth allocation and may not be responsive to sudden changes in process requirements or workloads.
3) **Flexibility: VBS provides great flexibility when it comes to resource allocation by allowing dynamic bandwidth allocation** in changing workload requirements. While CBS provides deterministic and predictable resource allocation.

- Best-Effort Bandwidth Server (BEBS)

The BEBS server is **an aperiodic server** that can be integrated into the real-time system. The **algorithm adjusts its period dynamically based on the runtime of tasks** and it **recognizes each best-effort task is not needed equal responsiveness and adapts its scheduling accordingly**. [7]

1) **Period Adjustment**: BEBS **dynamically adjusts its period based on the behavior of the assigned task** while CBS allocates a fixed bandwidth to each best-effort task and maintains it over time.
2) **Responsiveness: BEBS aims to provide better responsiveness for best-effort tasks by combining a timeshare strategy with an aperiodic server**. While CBS treats all best effort tasks quality and does not consider the levels of responsiveness.
3) **Flexibility: BEBS algorithm is designed to be integrated into a real-time scheduler allowing the simultaneous operation of soft real-time, hard real-time and best effort tasks**. While the CBS is more focused on providing CPU bandwidth reservations continues media applications and is less flexible when it comes to mixed time-constrained workloads.

- Total Bandwidth Server (TBS)

1)Period Adjustment: TBS allows for dynamic period adjustments based on workload demands and available resources.
2)Responsiveness: TBS dynamically allocates bandwidth based on the workload and requirements and releases unused resources improving the system's responsiveness and workload variations.
3)Flexibility: TBS allows dynamic bandwidth allocation based on the workload demands improving the flexibility of the system.

ADVANTAGES AND DISADVANTAGES OF CONSTANT BANDWIDTH SERVER

- **Advantages**:

1) CBS allows a **guaranteed bandwidth regardless of the system load and other tasks**.

2) CBS **allocates a fixed priority to each task** from higher priority to lower priority ensuring that each critical task's deadline is met.

3) CBS **provides precise predictions**. **The timing** and **execution are predicted more accurately** allowing strict timing requirements.

4) CBS allows **temporal isolation for tasks**. This behavior prevents tasks from exceeding allocated bandwidth. [5] [6]

- **Disadvantages:**

1. **Limited flexibility**, CBS is **not designed to dynamically changing environments**. It is designed for real-time systems where the timing is critical.
2. CBS **is not originally designed to cope with overruns**. This can affect system performance.
3. **CBS lacks adaptability**, which means once the bandwidth allocation is set any actual requirements spike will not be considered.

CONCLUSION

- CBS is a Scheduling algorithm used in real-time systems all over the world.
- CBS ensures temporal isolation and **maintains system integrity while ensuring that each task is executed within its allocated time frame**.
- This means CBS **ensures reliable operations of critical tasks guaranteeing the quality of service**.
- CBS is an essential part of real-time systems across multiple industries.

References

- [1]Dahiya, P., Real Time Systems, Geeks for Geeks, https://www.geeksforgeeks.org/real-time-systems/ visited on 11th June 2023
- [2] Henkler, S., Lecture Slides, RTS_introduction ,Hochschule Hamm-Lippstadt.
- [3] Silviu S., Christoph M. , Harald K.,Sokolova A., 'Temporal Isolation in Real-Time Systems', VBS-STTT.pdf (uni-salzburg.at) , visited on 11th June 2023
- [4] G. Buttazzo, Hard Real-Time Computing Systems: PredictableScheduling Algorithms and Applications - Second Edition, Springer2005, 2011
- [5] . M. M. A. Biondi, "Resource reservation for real-time selfsuspending tasks: Theory and practice," 2015
- [6] C. M. K. H. P. H. R. a. A. S. S. s. Craciunas, "Programmable temporal isolation"
- [7] S. Banachowski, T. B. Brandt, and S. A. Anderson, "Integrating best-effort scheduling into a real-time system," 2006