Summary: CPU Scheduling and Memory Management for Interactive Real-Time Applications
Shinpei Kato â Yutaka Ishikawa â Ragunathan

I. Overview

This paper investigates into how Constant Based Server (CBS) algorithm with EDF can be used to increase the performance of soft real time applications on multiprocessor systems. To achieve this, the paper improves on CBS and EDF-WM (Earliest deadline first with windows constraint migration) algorithms by proposing and implementing FCBS (Flexible CBS) and EDF-WMR (with reservation) algorithm.

EDF-WM algorithm schedule tasks in a task set using the normal EDF algorithm and it assigns task i to a CPU that has enough free capacity to run the task, if such CPU is not found, EDF-WM splits the task i relative deadline and its execution cost based on the number of CPUs in the system and assigns each part to different CPUs. It also allows the divided task to migrate between the different CPUs. CBS algorithm reserves CPU bandwidth for arriving sporadic task by specifying a server budget and a period P.

Flexible CBS is proposed as an improvement to CBS by reclaiming unused reserved bandwidth and adding it to the reserved bandwidth of tasks that are running late. To improve the EDF WMR algorithms, the flexible CBS is combined with EDF-WM algorithm. Instead of running the task set using the relative deadline, a certain amount of CPU bandwidth is reserved for the task and then scheduled on the CPU based on either allocating and available CPU splitting the server budget among the CPU. Unused server budgets is also reclaimed and allocated to tasks that are running late.

The paper also discusses memory management in real time systems. To improve real time application performance and unnecessary blocking time, memory pages used should not be swapped out to disk.

As discussed in the paper, most real time applications reserve all the memory pages needed to run the application at the beginning. These reserved memory pages are not swapped out to disk. However, memory reservation techniques end up over reserving the memory and therefore wasting memory.

To improve on this memory reservation technique, Private-Shared-Anonymous Paging (PSAP) algorithm is proposed and implemented. This algorithm reserves a limited amount of private memory needed by the first few jobs of the task. As the task run more memory is provisioned from a larger pool. This allows dynamic use of memory at run time. Memory pages that has not been used for a while is reclaimed and reallocated to task that needs it.

To benchmark these improvements, a video application was used. Video application exhibit characteristics of Soft real time system where hard real time guarantees are not needed. The Quality of Service (QoS) of the video application is used as a measure of the improved performance of the system.

II. Contribution

By benchmarking the algorithms proposed and implemented using a video application, the paper shows that the QoS of the video application in terms of stability of frames per second is high compared to ordinary EDF without CPU bandwidth reservation and reclamation.

PSAP memory reservation also outperforms the reserving all the memory needed by the real time application. This is due to dynamic use of memory reservation as it is needed by the task. The reservation technique reduces page swap to disk therefore enhancing application performance. This demonstrated in experiment as the QoS in terms of maintaining better frames per second video playback.

III. Limitation

Both decode and display task is grouped as one

The paper does not distinguish between the decoding and the display task component of the video application. The two functions is bundled as one and treated as a Soft real time application. This may not guarantee proper execution of display task that need hard deadlines to maintain quality of service.

I/O Blocking

The effect of I/O blocking on the task is not considered. Access to disk is an important aspect of realtime multimedia applications. Blocking time affects the response and QoS of such real time applications.

No consideration for task dependency

As discussed above, there is no separation between the decoding and the display task. In multimedia applications, it is necessary that the dependency between the decode and the display task be guaranteed to ensure the QoS is correctly evaluated.