Dynamic Scheduling

Slide 1

- We have looked at computing schedules if we know all the jobs in advance.
- Dynamic scheduling does not assume that we know the jobs in advance (although dynamic scheduling can be used for static schedules).
- We will look at Earliest Deadline First Scheduling which is optimal for a single processor system.

Running Example

Slide 2

Two tasks A and B

- A has period 2 and worst case execution time 0.9
- B has period 5 and worst case execution time 2.3

Liu and Layland test gives:

$$\frac{0.9}{2} + \frac{2.3}{5} = 0.45 + 0.46 = 0.91$$

So it fails the Liu and Layland test.

Exact Analysis

Slide 3

A has the highest priority this gives $R_A = 0.9$. B is interrupted by A so

$$R_B = 2.3 + \lceil \frac{R_B}{2} \rceil 0.9$$

Solve this by iteration, start with $R_B = 2.3$.

$$2.3 + \lceil \frac{2.3}{2} \rceil 0.9 = 2.3 + 2 * 0.9 = 4.1$$

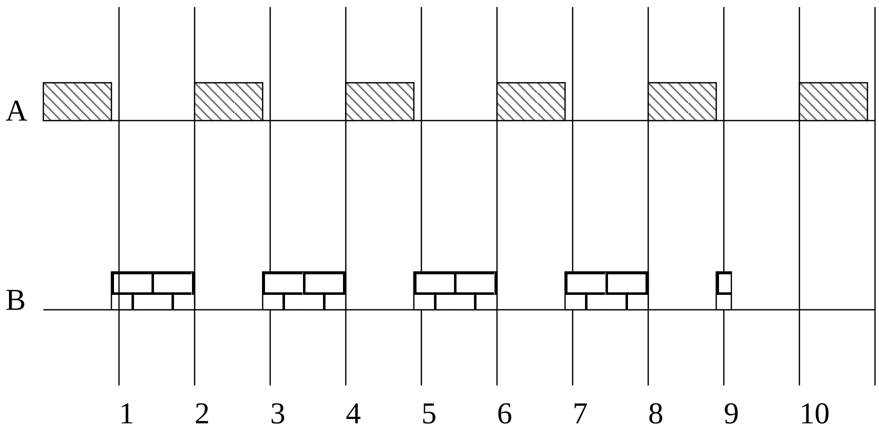
$$2.3 + \lceil \frac{4.1}{2} \rceil 0.9 = 2.3 + 3 * 0.9 = 5$$

$$2.3 + \lceil \frac{5}{2} \rceil 0.9 = 2.3 + 3 * 0.9 = 5$$

So B only just makes its deadline.

Schedule

Slide 4



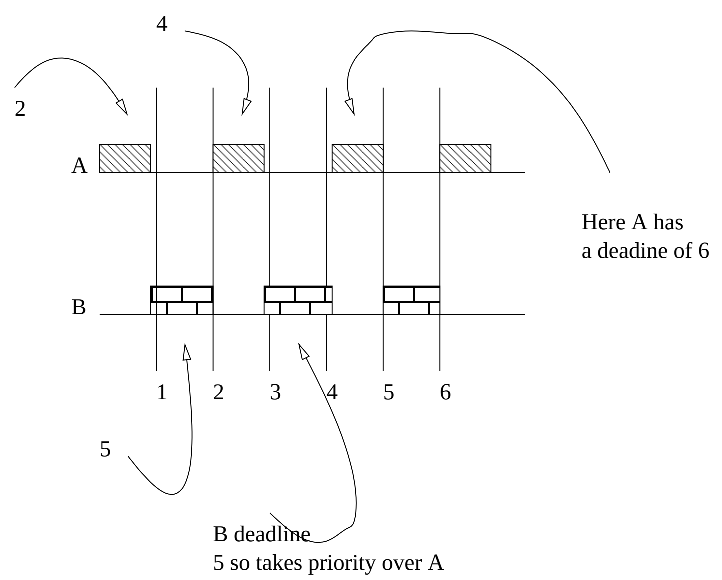
Earliest Deadline First - Dynamic

Slide 5

- The EDF algorithm can be implemented as a priority based schedule, except that priority of processes can change dynamically.
- EDF, the runnable task with the earliest deadline is always executed.

To illustrate the algorithm use the same task set as before, assume that the tasks are periodic, deadlines are equal to their periods.

Slide 6



EDF Schedules

Slide 7

- EDF schedules can be different from the Rate Monotonic schedule
- They are normally used when the task set is not known in advance.

Question:

- When can we use EDF schedules?

EDF Schedules are optimal

Slide 8

- If there exists a schedule such that all the tasks can be preemptively scheduled where the tasks are independent then there is an EDF schedule. (Only applies to single processors)

The idea of the proof is not that complicated, given a schedule you need to show that it can be transformed into an EDF schedule. This can be done by swapping around tasks that are not in an EDF relation. You have to show that such swaps still give a valid schedule.

When do EDF schedules work?

Slide 9

Section 6.3.1. Theorem 6.1

- A system of T independent preemptable tasks with relative deadlines equal to their periods can be feasibly scheduled on one processor if and only if its total utilization equal to or less than 1.

General test for EDF

Slide 10

- The idea is to keep the utilization below 1.

Suppose we have a set of tasks $T_i = (A_i, C_i, D_i)$ where A_i is the arrival time, C_i is the computation time and D_i is the deadline relative to the arrival time.

Then if at all i, we order the tasks $T_1, \ldots, T_i$ by earliest deadline then the tasks $1 \ldots i$ can be scheduled if $C_1 + \cdots + C_i \leq D_i$.

If for all i it the i tasks can be scheduled then the whole set can be scheduled.

EDF Example

Slide 11

Task set (Arrival,Computation,Relative Deadline)

- $A = (1, 5, 11)$, $B = (2, 1, 3)$ and $C = (3, 4, 8)$.

Then at each time we get:

- At time 1, $A = (1, 5, 11)$ and $5 \leq 11$.
- At time 2, $B = (2, 1, 3)$, $A = (1, 4, 10)$ (A has executed for one unit of time) and $1 + 4 \leq 10$
- At time 3, $B = (2, 0, 2)$, $C = (3, 4, 8)$, $A = (1, 4, 9)$ and $4 + 4 \leq 9$.

EDF extensions

Slide 12

- Not optimal for multiprocessor scheduling.
- EDF is not optimal for non-preemptive tasks

Part II

Slide 13

- Issues and topics in Multiprocessor Scheduling:
  - Models of distributed systems.
  - Task assignment.
  - Task assignment and communication overhead.

Multiprocessor Systems

Slide 14

- Many modern systems are designed with a number of processors. For example:
  - Mercedes S-class has 63 microprocessors;
  - a 1999 BMW 7-series has 65.

The Volvo S70 has not one, but two CAN buses running through it, connecting the microprocessors in the mirrors with those in the doors with those in the transmission. The mirrors talk to the transmission so that they can tilt down and inwards when you put the car in reverse. The radio talks with the anti-lock brakes so that the volume can go up and down with road speed (the ABS has the most accurate speed information). [a]

[a] Source `http://www.embedded.com/1999/9905/9905turley.htm+`

The Problem: How do tasks get assigned to processors?

Slide 15

It would obviously a waste of resources if for every task we had a single processor. We can do better. But there are some issues:

- Different task with different scheduling constraints and processor utilization. How do we assign tasks to different processors?
- There is a communication cost (network) if two tasks need to communicate and they are on different processors. If the tasks are on the same processor then there is less of a communication cost (shared memory). How do you allocate the tasks onto processors?

Task Assignment with No Communication Overhead.

Slide 16

- The simplest model for task assignment is where we simply look at the Execution time and deadline requirements.
- We ignore any possible communication overhead.

Even simple task assignment is NP-complete.

Bin Packing formulation

Slide 17

Remember with EDF schedules is the utilization is less than one then the tasks will meet their deadlines.

Model, assume a number of identical processors and a number of periodic tasks (deadlines = Periods).

Utilization

$$U_i = C_i/P_i$$

C_i computation time, P_i period.

Example

Slide 18

- 5 Tasks $U_A = 0.5$,$U_B = 0.2$,$U_C = 0.3$,$U_D = 0.75$,$U_E = 0.8$.
- Three processors.

It is possible to schedule as follows:

	$Processor1$	$Processor2$	$Processor3$
	A C	D B	E
$Total$	0.7	0.95	0.8

Slide 19

But there is no schedule staring:

	$Processor1$	$Processor2$	$Processor3$
	A	D	C
	B		
$Total$	0.7	0.75	0.3

There is no where to put E.

As the number of bins grow the number of possible solutions grow.

Slide 20

Heuristics - First Fit

Assign a random order to the tasks, then assign tasks to the first

processor that has enough space.

On the previous slide we saw an example that failed with first fit.

Meaningful question. When does the fail-first principle work. Oh and Baker have shown with m identical processors each scheduled with fixed-priority scheduleding first-fit can always find a feasible schedule when the total utilization is less than:

$$m(\sqrt{2} - 1)$$

More Complicated Algorithm

Slide 21

- Assume Period = Deadline.
- The test $m(\sqrt{2}-1)$ is not the best we can do.

Remember Liu & Layland test. If the utilization is less than:

$$U_{RM}(n) = n(2^{1/n} - 1)$$

then it is possible schedule the tasks (remember not a complete test).

Bin pack according to the utilization bound.

RMFF

Slide 22

- Sort tasks in nondecreasing order according to their periods.
- Assign next task to a processor (with x tasks) if the new utilization would be equal or less than $U_{RM}(x+1)$.

RMFF Example

Rate Monotonic First Fit example 6 tasks two processors.

Slide 23

Task	Period	WCET	Utilisation
1	2	1	0.5
2	2.5	0.1	0.04
3	3	1	0.33
4	4	1	0.25
5	4.5	0.1	0.022
6	5	1	0.2

Processor 1:

1 (0.5) 2 (0.54) ~~3~~ ~~4~~ 5 (0.56) 6 (0.76)

Processor 2:

3 (0.33) 4 (0.58)

$U_{RM}(1) = 1$, $U_{RM}(2) = 0.828$, $U_{RM}(3) = 0.787$, $U_{RM}(4) = 0.757$, $U_{RM}(5) = 0.743$.