Scheduling Algorithm and Analysis

Aperiodic Server

(Module 36)

Yann-Hang Lee
Arizona State University
yhlee@asu.edu
(480) 727-7507

Summer 2014
Scheduling Aperiodic/Sporadic Tasks

- **Assumptions:**
  - Preemptive, priority-driven algorithms
  - Jobs independent of one another with arbitrary interrelease times

- **Periodic Jobs**
  - Parameters and priority driven algorithm given
  - On their own, periodic jobs meet all deadlines

- **Aperiodic Jobs**
  - Parameters not necessarily known on release

- **Sporadic**
  - Parameters known on release
  - Variable execution time
  - Arbitrary deadline
Aperiodic, Sporadic scheduling algorithms:
- all periodic tasks meet their deadlines
- Sporadic jobs: on arrival, undergo acceptance test. Must not affect periodic jobs and already accepted sporadic jobs.
- Aperiodic jobs: Optimize response time (average) without affecting periodic and accepted sporadic jobs.

Diagram:
- Periodic Jobs
- Aperiodic Jobs
- Sporadic Jobs
- Acceptance Test
- Dispatcher
- Processor
- Priority Queues
- Accept
- Reject
Approaches: Aperiodic

- **Background**: scheduled when processor is idle
- **Interrupt-driven**: scheduled on arrival
- **Periodic server**: defined by $(p_s, e_s)$. Budget replenished at $p_s$ intervals. If scheduled and queue empty then budget set to 0.
- **Bandwidth-preserving server**: Improves on the periodic server by preserving budget (bandwidth) when aperiodic queue is empty:
  - Deferrable servers
  - Sporadic Server
  - Constant utilization and Total bandwidth servers
Example of a Polling Server

- To prove it works
  - the polling server is periodic and has a WCET of $e_s$
- When the polling server is eligible and there is no aperiodic task
  - the budget is lost
- Combine with a background server
Aperiodic Servers

- **A service thread waiting for the external trigger(s)**
  - fixed execution budget
  - replenishment interval (period)

- **Can be compared to periodic tasks**
  - if it is ready, run according to priority scheduling scheme

- **Priority adjusted to meet requirements**

- **Issues:**
  - How to reserve the bandwidth when no aperiodic task exists
  - how to replenish the budget.
  - Example: Polling server
    - no bandwidth preserving
    - fixed replenishment time
Deferrable Server

- **A periodic server task is created.**
  - When the server is invoked with no outstanding aperiodic tasks, the server does not execute but defers its assigned time slot.
  - When an aperiodic task arrives, the server is invoked to execute aperiodic tasks and maintains its priority.

- **Unlike the priority exchange policy, the server’s time is preserved at its initial priority.**

- **The computation time allowance for the server is replenished at the start of its period.**

- **Provides better response time for aperiodic tasks than Polling server**
Deferrable Server (DS)

- **Periodic task** \((p_s, e_s)\) model with rules:
  - budget consumed only when executing
  - budget replenished at \(kp_s\), budget = \(e_s\) at \(kp_s\)
Sporadic Servers

- The deferrable server has this one additional preemption and reduces the schedulability of periodic tasks.

- Vary the points at which the computation time of the server is replenished, rather than merely at the start of each period.
  - Allows to enhance the average response time for aperiodic tasks without degrading the utilization bound for periodic tasks.
  - Any spare capacity (i.e., not being used by periodic tasks) is available for an aperiodic task on its arrival.

- Sporadic server \((p_s, e_s)\) does not demand more processor time than a periodic task with the same parameters.
Supplementary Slides