Multiprocessor Models

- Identical (Homogeneous): All the processors have the same characteristics, i.e., the execution time of a job is independent on the processor it is executed.
- Uniform: Each processor has its own speed, i.e., the execution time of a job on a processor is proportional to the speed of the processor.
  - A faster processor always executes a job faster than slow processors do.
  - For example, multiprocessors with the same instruction set but with different supply voltages/frequencies.
- Unrelated (Heterogeneous): Each job has its own execution time on a specified processor
  - A job might be executed faster on a processor, but other jobs might be slower on that processor.
  - For example, multiprocessors with different instruction sets.

Scheduling Models

- Partitioned Scheduling:
  - Each task is assigned on a dedicated processor.
  - Schedulability is done individually on each processor.
  - It requires no additional on-line overhead.
- Global Scheduling:
  - A job may execute on any processor.
  - The system maintains a global ready queue.
  - Execute the M highest-priority jobs in the ready queue, where M is the number of processors.
  - It requires high on-line overhead.
- Semi-Partitioned Scheduling:
  - Adopt task partitioning first and reserve time slots (bandwidths) for tasks that allow migration.
  - It requires some on-line overhead.

Course Material

- Ronald L. Graham: Bounds for certain multiprocessing anomalies. in Bell System Technical Journal (1966).
- Ronald L. Graham: Bounds on Multiprocessing Timing Anomalies. SIAM Journal of Applied Mathematics 17(2): 416-429 (1969)
- Dorit S. Hochbaum, David B. Shmoys: Using dual approximation algorithms for scheduling problems theoretical and practical results. J. ACM 34(1): 144-162 (1987) (in textbook Approximation Algorithms by Vijay Vazirani, Chapter 10, not covered)
- Sanjoy K. Baruah, Nathan Fisher: The Partitioned Multiprocessor Scheduling of Sporadic Task Systems. RTSS 2005: 321-329
- Jian-Jia Chen: Partitioned Multiprocessor Fixed-Priority Scheduling of Sporadic Real-Time Tasks. ECRTS 2016.

Problem Definition

Partitioned Scheduling

Given a set $\mathbf{T}$ of tasks with implicit deadlines, i.e., $\forall \tau_i \in \mathbf{T}$, $T_i = D_i$, the objective is to decide a feasible task assignment onto M processors such that all the tasks meet their timing constraints, where C_{im} is the execution time of task τ_i on processor m.

- For identical multiprocessors: $C_i = C_{i1} = C_{i2} = \cdots = C_{iM}$.
- For uniform multiprocessors: each processor m has a speed s_m, in which $C_{im}s_m$ is a constant.
- For unrelated multiprocessors: C_{im} is an independent parameter.

Hardness and Approximation of Partitioned Scheduling

$\mathcal{N}P$-complete

Deciding whether there exists a feasible task assignment is $\mathcal{N}P$-complete in the strong sense.

Proof

Reduced from the 3-Partition problem.

Hardness and Approximation of Partitioned Scheduling

$\mathcal{N}P$-complete

Deciding whether there exists a feasible task assignment is $\mathcal{N}P$-complete in the strong sense.

Proof

Reduced from the 3-Partition problem.

- Approximations are possible, but what do we approximate when only binary decisions (Yes or No) have to be made?
  - Deadline relaxation: requires modifications of task specification
  - Period relaxation: requires modifications of task specification
  - Resource augmentation by speeding up: requires a faster platform
  - Resource augmentation by allocating more processors: requires a better platform

Approximation Algorithms

An algorithm $\mathcal{A}$ is called an η-approximation algorithm (for a minimization problem) if it guarantees to derive a feasible solution for any input instance I with at most η times of the objective function of an optimal solution. That is,

$$\mathcal{A}(I) \leq \eta OPT(I),$$

where $OPT(I)$ is the objective function of an optimal solution.

Terminologies Used in Scheduling Theory

Graham's Scheduling Algorithm Classification

- Classification: $a|b|c$
    - a: machine environment (e.g., uniprocessor, multiprocessor, distributed, ...)
    - b: task and resource characteristics (e.g., preemptive, independent, synchronous, ...)
    - c: performance metric and objectives (e.g., $L_{\max}$, sum of finish times, ...)
- Makespan problem:
    - $M||C_{\max}$
    - The course material mainly comes from the traditional makespan scheduling in the context of *minimizing the maximum utilization*. (The paper by Baruah and Fisher in RTSS 2005 is an exception.)
    - Imagine that the goal is to minimize the maximum utilization after task partitioning.

Bin Packing Problem

- Given a bin size b, and a set of items with individual sizes, the objective is to assign each item to a bin without violating the bin size constraint such that the number of allocated bins is minimized.