Multiprocessor Scheduling

Jardines Mendoza César Eduardo

Universidad Nacional Autónoma de México

7 de Mayo del 2020

Multiprocessor Scheduling

Scheduling

Process Scheduling es la actividad del administrador de procesos que maneja la eliminación de algun proceso en ejecución del CPU y la selección de otro proceso en función de una estrategia particular

Multiprocessor Scheduling

- Es un problema de optimización NP-hard

Multiprocessor Scheduling

- Es un problema de optimización NP-hard
- Es una generalización de la versión de optimización del problema de number partitioning problem, que considera el caso de particionar un conjunto de números (trabajos) en dos conjuntos iguales (procesadores).

Multiprocessor Scheduling

- Multiprocessor Scheduling es más complejo a comparación con Processor Scheduling

Multiprocessor Scheduling

- Multiprocessor Scheduling es más complejo a comparación con Processor Scheduling
- Hay dos tipos de Multiprocessor Scheduling:
  - Asymmetric Scheduling
  - Symmetric Scheduling

Problema

Dado un conjunto j de trabajos donde el trabajo j_i tiene una longitud l_i y varios procesadores m, ¿Cuál es el tiempo mínimo posible requerido para generar todos los trabajos en los procesadores j en m de modo que ninguno se superponga?"

Multiprocessor Scheduling

Algoritmo

- Longest Processing Time (LPT)

Algoritmo

- Longest Processing Time (LPT)
- **input:** n trabajos con tiempo de procesamiento $p_1, p_2, ..., p_n$ y m procesadores.

Algoritmo

- Longest Processing Time (LPT)
- **input:** n trabajos con tiempo de procesamiento $p_1, p_2, ..., p_n$ y m procesadores.
- **output:** La asignación de trabajos a los procesadores.

Algoritmo

1. Sort $p_i \geq p_{i+1}$ for i=1,...,n-1
2. Cada vez que una máquina está disponible, el trabajo con el índice más bajo (entre los trabajos no asignados) se asigna a esa máquina.

Algoritmo

Teorema LPT es 4/3-approximación.

Multiprocessor Scheduling

Demostración

Dem La duración del programa LPT es como máximo $4/3$ veces la duración optima, esto es cierto si la instancia solo tiene un trabajo. Ahora supongamos que se cumple para cualquier instancia con a lo más $n' < n$ trabajos. Ahora consideremos un ejemplar I de n trabajos y deje que σ sea scheduling LPT y denote su longitud por C_{max}. Ahora sea l un trabajo (n) que se complete al final, consideraremos dos casos.

Demostración

- **Caso 1:** l < n
  Eliminar el trabajo n del Schedule σ. Denotamos el horario restante con σ'. Entonces, la longitud de σ' es igual a la longitud de σ, ya que el trabajo n fue agregado en último lugar, pero no es el trabajo que completa el último. Ahora sea I' el ejemplar que consiste en los trabajos 1,...,n-1 (Tengamos en cuenta que σ' es exactamente el schedule que se optuvo aplicando LPT a I') así que por inducción, la C_{max} es a los más $4/3$ veces la longitud óptima de I', que a su vez es a lo más $4/3$ veces la longitud óptima de I.

Demostración

- **Caso 2:** l = n
  Sea s_n el tiempo de inicio del trabajo n. Del análisis de la listas Scheduling sabemos que:

$$C_{\max} \quad = \quad S_n + p_n \leq \frac{1}{m}(\sum_j p_j) + p_n \leqslant C^*_{\max} + p_n.$$

  Si $p_n < C^*_{max}/3$ entonces habremos terminado, ahora asumemos que $p_n > C^*_{max}/3$ entonces $C^*_{max} < 3p_n$ y dado que p_n es el tiempo de procesamiento más pequeño, esto implica que el programa óptimo tiene como máximo 2 trabajos por máquina i.e $C_{max} = C^*_{max}\square$