Balanceo de Carga

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Problema del balanceo de carga

Tenemos:

- Un Set de m maquinas M_1, M_2, M_m
- Un Set de n tareas
- Cada tarea j requiere T_j de tiempo de procesamiento.

Objetivo: Asignar las tareas a las maquinas de tal forma que la carga quede balanceada

(el tiempo asignado a cada maquina sea lo más parejo posible)

Cómo medir el balanceo?

Si llamamos A(i) al conjunto de tareas asignadas a la maquina i

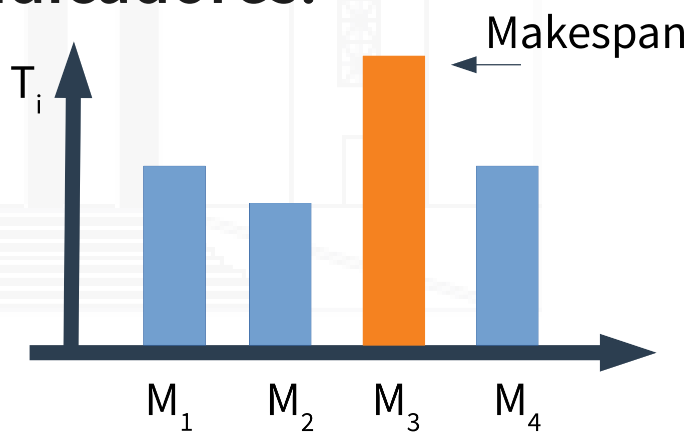
Podemos calcular la carga de la maquina i como:

$$T_i = \sum_{j \in A(i)} t_j$$

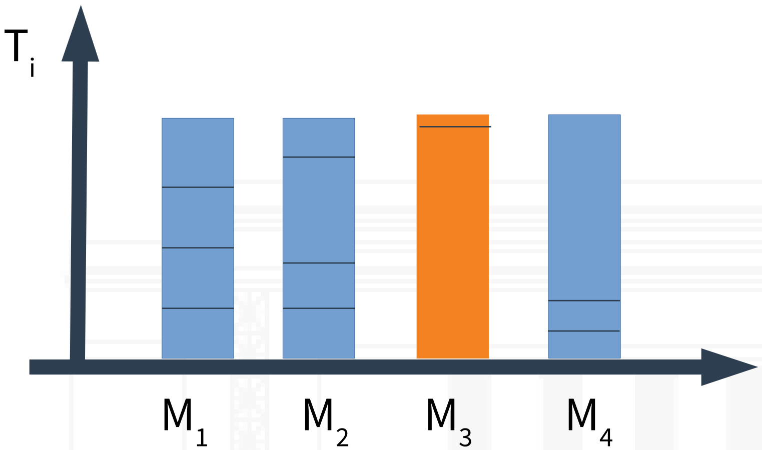
Podemos medir el balanceo por diferentes indicadores.

Usaremos:

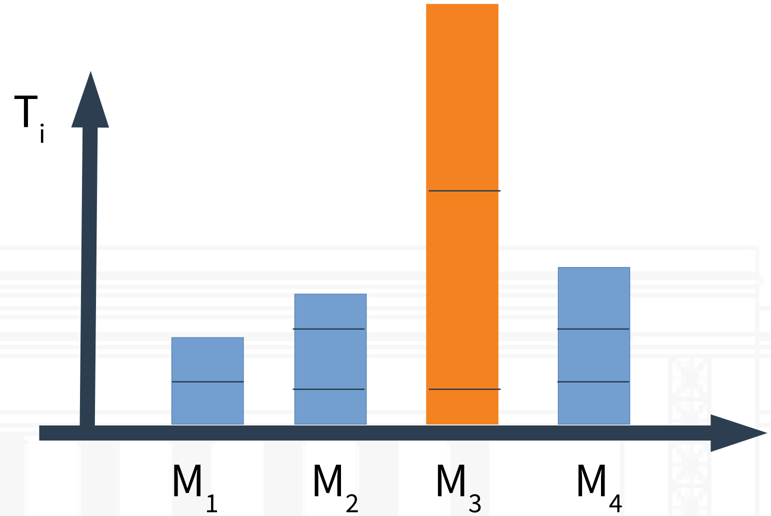
Makespan : max (Ti) para todas las maquinas



Asignación de trabajos



Ideal



No deseado

El método para seleccionar la asignación y la naturaleza de los trabajos determinará la programación final de las tareas

- Es un problema NP-HARD

Un primer método greedy

Para cada tarea i, asignarla a la maquina j con menor carga en el momento.

Comenzar sin trabajos asignados

Definir $T_i = 0$ y $A(i)=\emptyset$ para todos las maquinas M_i

Desde $j = 1$ a n

  Sea M_i la maquina con menor T_k ($k=1$ a m)

  Asignar Tarea j a maquina M_i

  Establecer $A(i) \leftarrow A(i) \cup \{j\}$

  Establecer $T_i \leftarrow T_i + t_j$

Análisis del algoritmo

Para determinar cuanto se aleja la solución obtenida de la optima (T*), debemos compararlas.

Pero … no tenemos la solución optima

Sin embargo, podemos acotarla:

$$T^* \geqslant \frac{1}{m} \sum_j t_j$$

El optimo es mayor o igual al tiempo promedio total

$$T^* \geqslant max_j \, t_j$$

El optimo es mayor o igual al tiempo del trabajo mas largo

Análisis del algoritmo (cont.)

A.1 El algoritmo asigna los trabajos a las máquinas con un makespan T ≤ 2T*.

El ultimo trabajo es asignado a máquina M_i con mínima carga

Antes de la asignación tendrá $T_i - t_j$ de carga.

Sabemos que: $\sum_k T_k \geq m(T_i - t_j)$

La carga en todas las maquinas

Acotamos:

$$T^* \geqslant \frac{1}{m}\sum_j t_j$$

$$T^* \geqslant max_j t_j$$

$$\left.\begin{array}{l} (T_i - t_j) \leq \frac{1}{m}\sum_k T_k \leq T^* \\ t_j \leq T^* \\ T_i = (T_i - t_j) + t_j \end{array}\right\} \quad T_i \leq 2T^*$$

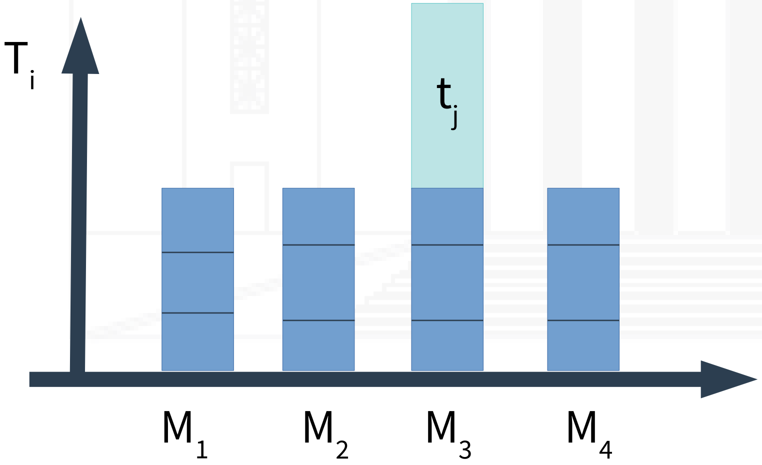
Podemos mejorar nuestro algoritmo?

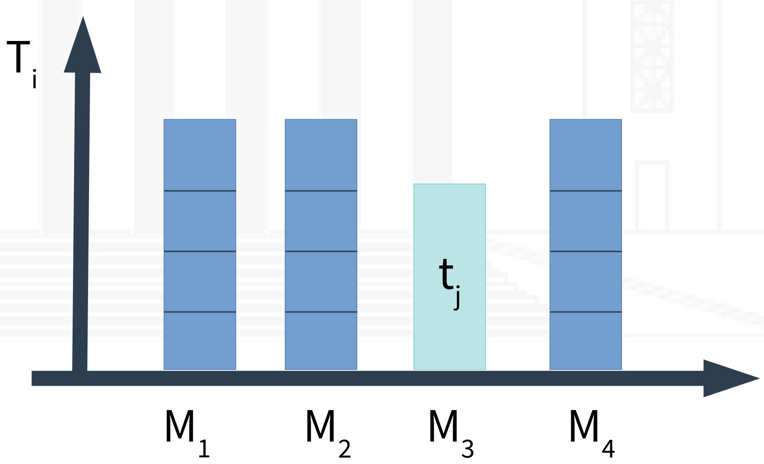
Cuando ocurre el peor caso?

El algoritmo intenta mantener siempre el mayor balance posible.

Si la ultima tarea coincide con aquella de longitud mas grande quedará peor balanceado

Ej: Las j-1 tareas de $t_x=a$ y $t_j>>a$





Algoritmo de aproximación mejorado

Procesar primero las tareas mas extensas

Ordenar las tareas.

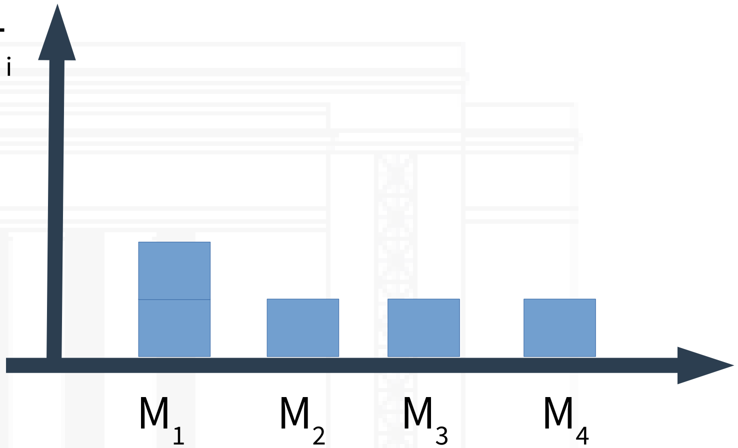
```
Comenzar sin trabajos asignados
Ordenar las tareas de mayor a menor duración
Definir Ti = 0 y A(i)=Ø para todos las maquinas Mi
Desde j = 1 a n
  Sea Mi la maquina con menor Tk (k=1 a m)
  Asignar Tarea j a maquina Mi
  Establecer A(i) ← A(i) ∪ {j}
  Establecer Ti←Ti + tj
```

Análisis del algoritmo mejorado

Si hay m o menos tareas

La solución es optima.

(A.2) Si hay más de m tareas, entonces $T^* \geq 2t_{m+1}$



Tomemos las primeros m+1 tareas ordenadas por tiempo descendiente.

Hay m máquinas, por lo tanto solo 1 recibe 2 tareas

En el peor de los casos las primeras m+1 tareas tienen la misma duración.

Por lo tanto en el makespan el menos $2t_{m+1}$

Análisis del algoritmo mejorado (cont.)

(A.3) El algoritmo asigna los trabajos a las máquinas con un makespan $T \leq 3/2T^*$.

Sea la maquina M_i que tiene al menos 2 trabajos.

Sea T_j el ultimo trabajo asignado a M_i $(j \geq m+1)$

$t_j \leq t_{m+1} \leq ½\, T^*$ (Utilizando A.2)

$T_i - t_j \leq T^*$

$$(T_i - t_j) + T_j \leq ½\, T^* + T^*$$

$$(T_i - t_j) \leq \frac{1}{m}\sum_k T_k \leq T^*$$

Por lo tanto: $\mathbf{T_i \leq 3/2\, T^*}$

Presentación realizada en Julio de 2020