Documento anónimo

Examen de problemas Grupo B 2013 resuelto.pdf

Exámenes Resueltos (teoría y Prácticas)

2º Arquitectura de Computadores

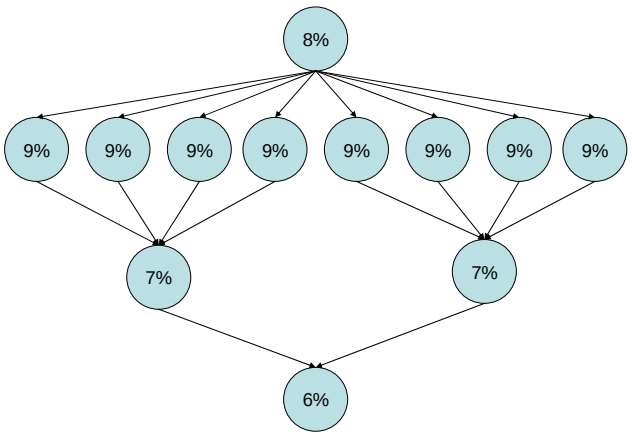
Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

ARQUITECTURA DE COMPUTADORES (Informática y Matemáticas)
Benchmark 2013

1. En la Figura se muestra el grafo de dependencia entre tareas para una aplicación, indicando en cada nodo la fracción del tiempo de ejecución secuencial que la aplicación tarda en ejecutar cada grupo de tareas del grafo. Suponiendo un tiempo de ejecución secuencial de 200 s, que las tareas no se pueden dividir en tareas de menor granularidad y un tiempo de comunicación despreciable:

- ¿Cuál es el valor de f en la ley de Amdahl?
- ¿Para qué número de procesadores se obtiene la máxima ganancia de velocidad?
- ¿Cuál es esa ganancia?
- Si dispone de un multiprocesador UMA con tantos procesadores como necesite, conectados a través de un bus compartido, y la comunicación entre dos tareas, a través de la memoria compartida, tarda 10 segundos (si las tareas que se comunican están asignadas a procesadores diferentes), ¿cuál es la máxima ganancia de velocidad que se obtiene?



SOLUCIÓN:

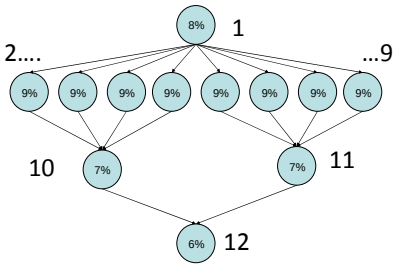
- El valor de la f de la ley de Amdahl se puede obtener teniendo en cuenta que es la parte del tiempo secuencial que sólo puede ejecutarla un procesador. Teniendo en cuenta la figura, esta sería el 8% de la primera tarea más el 6% de la segunda. Por lo tanto, el total sería el 14% del código secuencial, y por lo tanto, f=0.14.
- El número de procesadores que pueden trabajar en paralelo en la aplicación es 8, dado que no hay en el grafo más de 8 tareas que puedan ejecutarse en paralelo (primero solo una, luego 8, luego 2, y finalmente otra).
- La ganancia de velocida que se tendría sería:

$$T_{paralelo}=(0.08+0.09+0.07+0.06)*T_{secuencial} = 0.30*T_{secuencial}$$

Por lo tanto: $S = T_{secuencial}/T_{paralelo}=1/0.30= 3.334$

- Si tuviésemos que tener en cuenta el coste de comunicación de 10 segundos por comunicación entre tarea y tarea asignadas a procesadores distintos, conectados a la memoria compartida través de un bus compartido, habría que considerar la asignación de nodos a procesadores que se hace.
  Una posible asignación es ejecutar nodos {1,2,10,12} (según la numeración de la figura) en el procesador P1, los nodos 3, 4,…., 9 respectivamente en los procesadores P2, P3, ….., P8,

y el 11 en el P8 (o en cualquiera de los procesadores P5, P6, P7, P8 que ejecutan los nodos 6, 7, 8, y 9 que deben comunicarse con el procesador al que se asigne el nodo 11)



Así, el número de comunicaciones entre procesadores diferentes es 7 + 6 + 1 = 14, y por lo tanto, el tiempo de comunicación sería Tcom = 10*14 = 140 y el tiempo paralelo:

$$T_{paralelo}(8 \text{ procesadores}) = (0.08+0.09+0.07+0.06)*T_{secuencial}+T_{com}= 0.30*200+140 = 200 \text{ s}$$

Por lo tanto no se ganaría nada.

Si se desea, se podría analizar qué pasa con menos procesadores. Así, con dos procesadores se podría obtener una pequeña ganancia de velocidad, si se hace la asignación de los nodos {1,2,3,4,5,10,12} al procesador P1, y de los nodos {6,7,8,9,11} al procesador 2:

$$T_{paralelo}(2 \text{ procesadores}) = (0.08+0.36+0.07+0.06)*T_{secuencial}+ 10*(4+1)=164 \text{ s}$$

Por lo tanto, se tiene una ganancia de 200/164= 1.22

Con cuatro procesadores, si asignamos {1,2,3,10,12} a P1, {4,5} a P2, {6,7} a P3, y {8,9,11} a P4 se tiene:

$$T_{paralelo}(4 \text{ procesadores}) = (0.08+0.18+0.07+0.06)*T_{secuencial}+ 10*(6+4+1)=188 \text{ s}$$

Empeora con respecto a 2 procesadores. Dado que hay bastante carga de comunicación, a medida que crecen los procesadores se reduce la ganancia de velocidad.

2. En un multiprocesador SMP con 4 procesadores o nodos (N0-N3) basado en un bus, que implementa el protocolo MESI de *espionaje* para mantener la coherencia de cache, cada procesador dispone de una cache de datos de 512 Kbytes con marcos de bloque (también llamados líneas) de 32 bytes.
En el multiprocesador se están ejecutando en paralelo cuatro hebras que acceden a los elementos de dos arrays X[] e Y[] de 16 elementos de 32 bits, que se encuentran almacenados en posiciones consecutivas de la memoria principal (primero los de X[] y luego los de Y[]) a partir de un inicio de bloque.

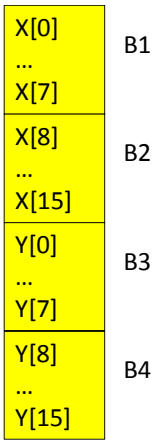
Indique los estados de los bloques en las caches, ante la siguiente secuencia de eventos:
- Lectura generada por el procesador 1 a X[0]
- Escritura generada por el procesador 2 a Y[0]
- Escritura generada por el procesador 3 a X[4]
- Lectura generada por el procesador 4 a Y[2]

NOTA: Suponga que bloques distintos se almacenan en la cache de cada procesador en marcos de bloque (líneas) diferentes.

SOLUCIÓN:

Teniendo en cuenta que los datos son de 32 bits, es decir, 4 bytes, en cada bloque (32 bytes) se encuentran 8 datos (32 bytes por bloque / 4 bytes por dato)

Por lo tanto tendríamos los datos de los dos arrays ubicados en cuatro bloques que denominamos B1, B2, B3, y B4, según la figura:



Por lo tanto, los accesos que se realizan son:

R1(B1)	(lectura desde el procesador 1 accede a X[0] que está en el bloque 1)
W2(B3)	(escritura desde el procesador 2 a Y[0] que está en el bloque 3)
W3(B1)	(escritura desde el procesador 3 a X[4] que está en el bloque 1)
R4(B3)	(lectura desde el procesador 4 a Y[2] que está en el bloque 3)

Por lo tanto se realizan accesos a dos bloque diferentes, B1 y B3, y solo habría problemas de coherencia entre los accesos que van al mismo bloque:

En el caso del bloque B1:

R1(B1)	hace que el bloque B1 pase a la **cache de P1 a estado E** (no hay otra copia en ninguna cache)
W3(B1)	hace que el bloque **B1 pase en la cache de P1 a estado I** y en la **cache de P3 a estado M**

En el caso del bloque B3:

W2(B3)	hace que el bloque B3 pase a la **cache de P2 a estado M**
R4(B3)	hace que el bloque B3 se actualice en memoria principal y pase a estado **S en la cache de P2** y pase a la **cache de P4 a estado S**