

Tema 4 AC2

Planificación de instrucciones

Bloque básico estático BB

Algoritmo de planificación de instrucciones

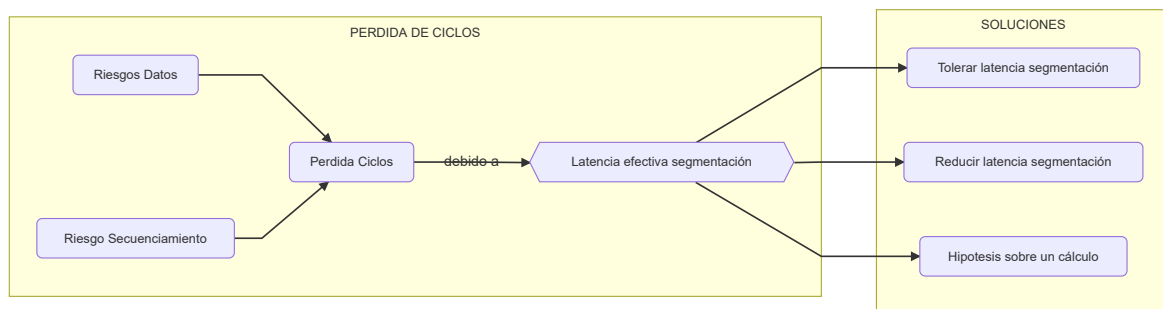
Cortocircuitos

Reducción de la penalización del secuenciamiento

Predicción fija del sentido

Tema 4 AC2

Con la segmentación del tema 3 tenemos ciertos momentos de riesgo donde perdemos ciclos, en este tema se exponen varias soluciones.



Planificación de instrucciones

Sobre un conjunto de instrucciones, puede existir una ordenación que no afecte en el resultado final y tolere la latencia de segmentación. Entre una instrucción productora y una consumidora se colocan otras instrucciones que no afecten el resultado y que 'den tiempo' a la productora a escribir el resultado.

Bloque básico estático BB

Los bloques de instrucciones a ordenar a través de un algoritmo de planificación se llaman bloques básicos estáticos.

- Empezará en un *líder* que se identifica:
 - En el principio del código,
 - Una instrucción destino de secuenciamiento,
 - Una instrucción de secuenciamiento, acaba en el próximo líder.
- Acabará en el próximo líder, esta última instrucción tendrá una dependencia de control. Deberá interpretarse siempre en último lugar.

Cualquier ordenación del grafo del BB es una planificación correcta. El objetivo es perder el menor número de ciclos posible.

Algoritmo de planificación de instrucciones

- Una planificación para reducir los ciclos perdidos debe tener en cuenta el retardo/latencia productor-uso.

- Los compiladores son los encargados de aplicar los algoritmos (medio transparente al programador).
- Un algoritmo de planificación, por lista tendrá las siguientes características:
 1. Partir del grafo de dependencias, añadir etiqueta con el retardo productor-uso -1.
 2. Antidependencias y dependencias de salida etiqueta 0. Arcos adicionales para la última instrucción(BR).
 3. Lista elegibles:
 1. Raíces del árbol o nodo sin predecesor
 2. Tiene predecesor y $t \geq TMC$ (se calcula $\rightarrow TMC = t + tag$)
 4. Euristicas de selección: Se eligirá de la lista de elegibles el que cumpla
 1. Camino más largo
 2. Si hay empate, el que apareciera primero en el orden del programa

```
//PSEUDOCODIGO:
int t = 0;
List<ints> ret; Graph<inst> dep; List<inst> ele;
while(!dep.isEmpty()){
    ele = seleccionar(dep);
    inst n = heuristica(ele);
    del_graph_list(n);
    append(ret, n);
    ++t;
    for_each(node s: n->sucesors)
        s.tag = max(s.tag, t+s.arco);
}
return ret;
```

Cortocircuitos

Reducción de la penalización del secuenciamiento

Predicción fija del sentido
