

PADREANDO-EL-BP4.pdf



BlackTyson



Arquitectura de Computadores



2º Grado en Ingeniería Informática



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

GENERALES:

- Lo importante a optimizar son los cuellos de botella sobre todo.
- Se puede optimizar sin ir a ensamblador.
- Hay optimizaciones a alto nivel/ensamblador, y para procesadores específicos/cualquiera.
- Hay diferentes opciones de optimización en un compilador.

EJECUCIÓN:

- Evitar las divisiones: sustituirlas por desplazamientos o multiplicaciones.
- Desenrollado de bucles: poner mas de una iteración en la misma iteración si son independientes.
 - o Ventajas: Reduce los saltos, puede encontrar más instrucciones independientes, facilita las instrucciones que ocultan latencias.
 - o Aumenta el tamaño de código.
- Código ambiguo: hay que evitar punteros ya que no permiten ciertas optimizaciones.
 - o Usar variables locales o globales, poner instrucciones de almacenamiento después/mucho antes de las de carga.

ACCESO A MEMORIA:

- Alineamiento de datos: Si un dato ocupa más de una línea de cache se aumenta el tiempo de acceso.
- Localidad de accesos:
 - o Crear estructuras que hagan que nuestros accesos sean locales.
 - o Acceder primero a las columnas en una matriz ya que se guardan fila a fila (así que accediendo a todos los elementos de cada fila es más óptimo)
- Atascos: un load adelanta a un store. Se produce cuando datos de mismo/distinto tamaño se almacenan y luego se cargan de direcciones solapadas pero no alineadas.
 - o Solución: usar datos del mismo tamaño en direcciones alineadas y los loads lo más lejos posible de los stores.
- Precaptación:
 - o `_MM_HINT_NTA`: en buffer no temporal.
 - o `_MM_HINT_T0`: en toda la caché.
 - o `_MM_HINT_T1`: en L2 y L3.
 - o `_MM_HINT_T2`: en L3.

SALTOS:

- Evaluar todas las condiciones a la vez si se puede, ya que cada una es un salto.
- Colocar primero las alternativas que más se ejecutan en un switch si se implementa como una búsqueda binaria.
- Instrucciones condicionales: hacen la instrucción si se cumple la condición.