Arquitectura e Ingeniería de Computadores - 3er Curso. Sol. del Examen 1er parcial 25/10/2019
Responde cada pregunta en una hoja distinta. Tiempo disponible: 2h30m

1. (2,5 **puntos)** Se dispone de una aplicación que realiza el proceso de inferencia de una red neuronal entrenada para clasificar tipos de defectos de fabricación de piezas industriales para el sector agrícola. Dicha aplicación, una vez caracterizada, realiza las siguientes operaciones por cada pieza analizada:

   - Multiplicación de matrices: 80 % del tiempo de ejecución.
   - Acceso a disco: 15 % del tiempo de ejecución.
   - Resto de operaciones: 5 % del tiempo de ejecución.

   Se desea utilizar la aplicación en un contexto industrial donde el tiempo máximo de ejecución de la aplicación debe ser de 1 segundo, tiempo que corresponde al proceso de fabricación de una pieza.

   El sistema actual donde se ejecuta la aplicación consta de un computador con un procesador y un disco duro con tiempo medio de acceso de 10ms. El computador tiene un coste de 1000 euros.

   Con este computador la aplicación no cumple la limitación del tiempo de ejecución ya que utiliza 1,2 segundos para analizar una pieza. Por este motivo. se plantea uno de los siguientes tres cambios en la configuración del computador:

   - Añadir una GPU de gama alta para el cálculo de matrices. La GPU permite mejorar el cálculo de multiplicación de matrices en un factor 2.5 comparado con la CPU. El coste de la GPU es de 500€.
   - Añadir un disco de gama alta con tiempo de acceso medio de 2ms. El coste del disco de gama alta es de 250€.
   - Añadir una GPU de gama media con factor de mejora en multiplicaciones de 1.2 y disco duro de gama media con tiempo de acceso medio de 5ms. El coste de la solución (GPU + disco) es de 400€.

   Calcular:

   a) La aceleración global que se producirá si se opta por cada una de las tres mejoras en el computador por separado.
   b) El tiempo de ejecución de la aplicación en el computador una vez incorporadas cada una de las mejoras por separado.
   c) Indicar qué opción es mejor desde el punto de vista combinado de coste y prestaciones y que cumple con el tiempo máximo de ejecución de 1 segundo.

   Solución:

   a) Aceleración global con cada una de las mejoras en el computador.
   - GPU gama alta. La aceleración local es de 2.5 ($S = 2{,}5$) y la fracción de uso es de 0.8 (F). A partir de estos dos valores podemos aplicar la ley de Amdhal y obtener la aceleración global (S').
     $S' = \frac{1}{(1-F)+\frac{F}{S}} = \frac{1}{0{,}2+0{,}32} = \frac{1}{0{,}52} = 1{,}92$
   - Disco gama alta. La aceleración local (S) es la relación entre los tiempos de acceso ($\frac{10ms}{2ms} = 5$). La fracción de uso es 0.15 (F). Podemos aplicar la ley de Amdhal para obtener la aceleración global en este caso:
     $S' = \frac{1}{(1-F)+\frac{F}{S}} = \frac{1}{0{,}85+0{,}03} = \frac{1}{0{,}88} = 1{,}136$

- GPU y disco gama media. La aceleración local de cada uno viene definida por el enunciado ($S_{gpu} = 1{,}2$) o la relación de tiempos de acceso ($S_{disco} = \frac{10ms}{5ms} = 2$). Las fracciones de uso vienen definidas del enunciado: $F_{computo} = 0{,}8$, $F_{disco} = 0{,}15$. Podemos aplicar la ley de Amdhal con más de una mejora:
  $S' = \frac{1}{1-F_{computo}-F_{disco}+\frac{F_{computo}}{S_{gpu}}+\frac{F_{disco}}{S_{disco}}}$
  $S' = \frac{1}{0{,}05+\frac{0{,}8}{1{,}2}+\frac{0{,}15}{2}} = \frac{1}{0{,}05+0{,}667+0{,}075} = \frac{1}{0{,}792} = 1{,}263$

b) Tiempo de ejecución con las nuevas mejoras
Podemos obtener el nuevo tiempo de ejecución a partir de la ecuación de la formula de la aceleración
$S = \frac{T_{ejec\_base}}{T_{ejec\_mejora}} \rightarrow T_{ejec\_mejora} = \frac{T_{ejec\_base}}{S}$
$T_{ejec\_gpu} = \frac{1{,}2s}{1{,}92} = 0{,}625s$
$T_{ejec\_disco} = \frac{1{,}2s}{1{,}136} = 1{,}056s$
$T_{ejec\_gpu\_disco} = \frac{1{,}2s}{1{,}263} = 0{,}950s$

c) Coste/Prestaciones
La opción de GPU de gama alta y la opción de gama media de GPU y disco cumplen con la restricción de tiempo de ejecución menor de 1 segundo por pieza. Ahora bien, debemos ver la relación coste/prestaciones para seleccionar la mejor opción.
El coste relativo de cada opción en porcentaje lo podemos obtener de la fracción entre la suma del coste del computador y la opción con el coste del computador:
$Coste_{relativo} = \frac{Coste_{computador}+Coste_{opcion}}{Coste_{computador}}$
$Coste_{relativo\_gpu} = 1{,}5$
$Coste_{relativo\_disco} = 1{,}25$
$Coste_{relativo\_gpu\_disco} = 1{,}4$
La relación Prestaciones/Coste es la siguiente:
$\frac{S_{gpu}}{Coste_{relativo\_gpu}} = \frac{1{,}92}{1{,}5} = 1{,}28$
$\frac{S_{disco}}{Coste_{relativo\_disco}} = \frac{1{,}136}{1{,}25} = 0{,}91$
$\frac{S_{gpu\_disco}}{Coste_{relativo\_gpu\_disco}} = \frac{1{,}263}{1{,}4} = 0{,}90$
Por tanto, la opción que es mejor desde el punto de vista del incremento de prestaciones en relación al incremento del coste, es la opción de una GPU de gama alta. La opción de gama media, aunque cumple el requisito del tiempo de ejecución menor de 1 segundo, no resulta adecuada desde el punto de vista del coste/prestaciones.

□

2. (2,5 **puntos)** Sea un procesador MIPS segmentado en cinco etapas utilizado en un sistema empotrado. El procesador no incorpora cortocircuitos e inserta ciclos de parada para resolver todos los posibles conflictos, tanto de datos, como de control. El procesador tiene una frecuencia de reloj de 125 MHz. El procesador se ha evaluado con una aplicación típica, obteniendo un tiempo de ejecución de 12 ms. La aplicación contiene 1.2 millones de instrucciones.

   Calcular:

   a) CPI del procesador al ejecutar la aplicación.

   b) El nuevo tiempo de ejecución si el procesador fuera capaz de eliminar todos los ciclos de parada.

   c) Aceleración obtenida si sustituimos el procesador inicial por otro que no tiene lógica para detectar conflictos de datos ni de control, utilizando un compilador que reorganiza el código, el cual inserta un 20 % más de instrucciones al código inicial de la aplicación para resolver todos los conflictos de datos y de control.

   d) Aceleración respecto al procesador original si se utilizara la técnica *predict-not taken* para resolver los conflictos de control y cortocircuitos para resolver los conflictos de datos. Las instrucciones de carga insertan un ciclo de parada en caso de conflicto de datos con la instrucción siguiente, lo que

representa el 2 % de todas las instrucciones ejecutadas. El 12 % de las instrucciones son de salto condicional, donde el 60 % son efectivos. El PC se actualiza en la 2ª etapa de segmentación.

Solución:

a) CPI del procesador al ejecutar la aplicación
$T_{eje} = I \times CPI \times T$, donde $I = 1{,}2M$, $T = \frac{1}{125MHz} = 8ns$, $T_{eje} = 12\ ms$
$CPI = \frac{T_{eje}}{I \times T} = \frac{12ms}{1{,}2M \times 8ns} = \frac{12ms}{9{,}6ms} = 1{,}25$

b) El nuevo tiempo de ejecución si el procesador fuera capaz de eliminar todos los ciclos de parada.
El $CPI = 1$ en el caso de resolverse todos los conflictos sin insertar ciclos de parada. En ese caso:
$T_{eje} = I \times CPI \times T = 1{,}2M \times 1 \times 8ns = 9,6ms$

c) Aceleración obtenida si sustituimos el procesador inicial por otro que no tiene lógica para detectar conflictos de datos ni de control, utilizando un compilador que reorganiza el código, el cual inserta un 20 % más de instrucciones al código inicial de la aplicación para resolver todos los conflictos de datos y de control.
$I = 1{,}2M \times 1{,}2 = 1{,}44M$
$CPI = 1$
$T = 8ns$
$T_{eje} = 1{,}44M \times 1 \times 8ns = 11,52ms$
$S = \frac{12s}{11{,}52s} = 1,042 \rightarrow 4{,}2\,\%$

d) Aceleración respecto al procesador original si se utilizara la técnica *predict-not taken* para resolver los conflictos de control y cortocircuitos para resolver los conflictos de datos. Las instrucciones de carga insertan un ciclo de parada en caso de conflicto de datos con la instrucción siguiente, lo que representa el 2 % de las instrucciones ejecutadas. El 12 % de las instrucciones son de salto condicional, donde el 60 % de los saltos son efectivos. El PC se actualiza en la 2ª etapa de segmentación.
$T = 8ns$
$CPI = 1 + 0,02 \times 1 + 0,12 \times 0,6 \times 1 = 1,02 + 0,072 = 1,092$
$I = 1{,}2M$
$T_{eje} = 1,2M \times 1,092 \times 8ns = 10,48ms$
$S = \frac{12s}{10{,}48s} = 1,14$

□

3. (3 **puntos)** El siguiente diagrama instrucciones–tiempo corresponde a la ejecución de cierto programa P sobre un procesador MIPS64 segmentado:

```
PC      Instruccion       1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
loop    l.s f0,0(r10)     IF ID EX ME WB
20      l.s f1,4(r10)        IF ID EX ME WB
24      add.s f0,f0,f10         IF ID A1 A2 A3 WB
28      add.s f1,f1,f10            IF ID A1 A2 A3 WB
32      s.s f0,0(r11)                 IF ID EX ME
36      s.s f1,4(r11)                    IF ID EX ME
40      dadd r10,r10,#8                     IF ID EX ME WB
44      dadd r11,r11,#8                        IF ID EX ME WB
48      bne r10,r20,loop                          IF ID EX ME WB
52      sigte                                        IF ID -
.etext  sigte                                           IF -
loop    l.s f0,0(r10)                                      IF ID EX ME WB
```

Se pide:

a) Identifica una dependencia de datos, una antidependencia, una dependencia de salida y una dependencia de control en el código anterior.

b) Calcula los CPI obtenidos y el tiempo de ejecución del programa en función del número de iteraciones n.

c) Indica los cortocircuitos que se han aplicado para resolver los riesgos de datos. Utiliza la notación: "Ciclo Z. cortocircuito de X-a-Y" donde X e Y representan etapas de segmentación del procesador.

d) Indica qué técnica se ha empleado para resolver los riesgos de control. Justifica la respuesta.

Para mejorar las prestaciones, se indica al compilador que utilice instrucciones SIMD que permiten operar simultáneamente sobre las dos porciones de 32 bits de cada registro de coma flotante de 64 bits. Estas instrucciones tienen el sufijo ".ps" y el código queda como sigue:

```
loop: l.ps f0,0(r10)
      add.ps f0,f0,f10
      s.ps f0,0(r11)
      dadd r10,r10,#8
      dadd r11,r11,#8
      bne r10,r20,loop
```

e) Dibuja el diagrama instrucciones–tiempo resultante de su ejecución.

f) Calcula los CPI obtenidos y el tiempo de ejecución del programa en función del número de iteraciones n.

Solución:

a) Dependencias: De datos: l.f f0,0(r10) $\rightarrow$ add.f f0,f0,f10.
Antidependencia: s.f f0,0(r11) $\leftarrow$ dadd r11,r11,#8.
Dep. salida: l.f f0,0(r10) $\leftrightarrow$ add.f f0,f0,f10.
Dep. control: bne r10,r20,loop $\rightarrow$ l.f f0,0(r10).

b) CPI y tiempo de ejecución:
Cada iteración del bucle tiene 9 instrucciones y contribuye con 11 ciclos. Por lo tanto, para n iteraciones:
$T_{ejecucion} = 11n$ ciclos y CPI=$\frac{11}{9} = 1{,}22$.

c) Cortocircuitos.
Ciclos 5 y 6. WB-a-A1 Ciclos 8 y 9. WB-a-M Ciclo 11. WB-a-EX (condición se calcula en EX) o de MEM-a-ID (condición se calcula en ID).

d) Técnica para resolver los riesgos de control:
Se cancelan instrucciones cuando los saltos son efectivos. Por lo tanto, se utiliza *predict-not-taken* o predicción dinámica con fallo de predicción.

e) Diagrama instrucciones–tiempo con instrucciones SIMD:

PC	Instruccion	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
loop	l.ps f0,0(r10)	IF	ID	EX	ME	WB										
20	add.ps f0,f0,f10		IF	id	ID	A1	A2	A3	WB							
24	s.ps f0,0(r11)			if	IF	id	ID	EX	ME	WB						
28	dadd r10,r10,#8					if	IF	ID	EX	ME	WB					
32	dadd r11,r11,#8							IF	ID	EX	ME	WB				
36	bne r10,r20,loop								IF	ID	EX	ME	WB			
40	sgte									IF	ID	–				
44	sgte										IF	–				
loop	l.ps f0,0(r10)											IF	ID	EX	ME	WB

f) Cada iteración del bucle tiene 6 instrucciones y contribuye con 10 ciclos. Por lo tanto, para n iteraciones:
$T_{ejecucion} = 10n$ ciclos y CPI=$\frac{10}{6} = 1{,}67$.

□

4. (2 **puntos)** Durante la ejecución de una aplicación en un procesador con un predictor dinámico de saltos del tipo BTB con 1 bit para la predicción se obtienen las siguientes estadísticas:

   a) Los saltos son efectivos en un $70{,}00\,\%$ de los casos.

   b) Los saltos se encuentran en la tabla del predictor en un $90{,}00\,\%$ de los casos.

   c) El predictor acierta la predicción en un $85{,}00\,\%$ de los saltos que se encuentran en la tabla.

   El procesador está segmentado en 7 etapas y la escritura del PC por parte de las instrucciones de salto se realiza en la quinta etapa. El predictor BTB ofrece la predicción al final de la primera etapa.

   Se solicita, justificando las respuestas, lo siguiente:

   a) Rellenar la tabla adjunta (ver hoja de respuestas) indicando para cada posible transición de la máquina de estados del predictor, la condición de salto, si se produce o no penalización en el salto y la frecuencia con que dicha transición se produce. Supóngase que la probabilidad de acertar o fallar es independiente del estado actual del predictor.

   b) Mostrar en la tabla adjunta (ver hoja de respuestas) el comportamiento de la BTB cuando predice que **salta** y finalmente el salto **no salta**, indicando qué instrucciones son las que se ejecutan en los ciclos siguientes al salto. Indicar además la penalización que sufre la instrucción de salto en este caso.

   Las etapas de cada instrucción se mostrarán como `E1`, `E2`, `E3`, ... Las instrucciones canceladas se representarán con una `X` en el ciclo correspondiente. Las instrucciones siguientes al salto se representarán como `pc+1`, `pc+2`, ... y las instrucciones de destino del salto como `dest`, `dest+1`, etc.

   c) Si en ausencia de fallos las instrucciones de salto tienen un CPI de 1, indicar el CPI real de las instrucciones de salto dadas las estadisticas obtenidas.

   d) Se pretende sustituir el predictor actual por un predictor de dos niveles que utilice 1 bit para la predicción y 1 bit de historia global, es decir, un predictor del tipo *(1: global, 1: local)*. La siguiente tabla muestra las estadísticas obtenidas con dicho predictor. Teniendo en cuenta que la probabilidad de que el salto previo sea efectivo se mantiene en un $70{,}00\,\%$, ¿sería interesante la sustitución? ¿Cuál sería el nuevo CPI de las instrucciones de salto?

Estado Origen		Estado Destino	Frecuencia (%) Último salto: No salta	Salta
$\emptyset$	$\rightarrow$	No Salta	3.00 %	
$\emptyset$	$\rightarrow$	Salta	7.00 %	
No Salta	$\rightarrow$	No Salta	21.60 %	24.30 %
No Salta	$\rightarrow$	Salta	12.60 %	6.30 %
Salta	$\rightarrow$	No Salta	5.40 %	2.70 %
Salta	$\rightarrow$	Salta	50.40 %	56.70 %

$\emptyset \rightarrow$ Indica que la instrucción de salto no está en la tabla.

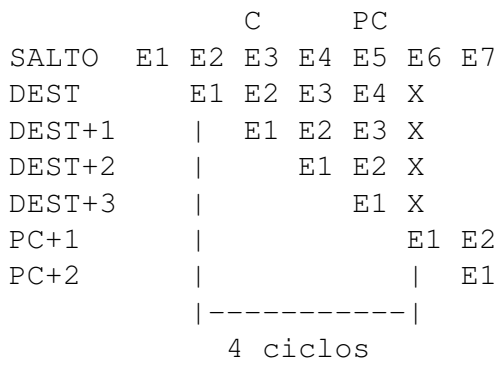
Solución:

a) Rellenar la tabla adjunta indicando para cada posible transición de la máquina de estados del predictor, la condición de salto, si se produce o no penalización en el salto y la frecuencia con que dicha transición se produce. Supóngase que la probabilidad de acertar o fallar es independientemente del estado actual del predictor.

Estado Origen		Estado Destino	Condición (S / NS)	¿Hay penalización? (S)í / (N)o	Frecuencia (%)
$\emptyset$	$\rightarrow$	No Salta	NS	N	$0{,}1 \times 0{,}3 = 3{,}00\,\%$
$\emptyset$	$\rightarrow$	Salta	S	S	$0{,}1 \times 0{,}7 = 7{,}00\,\%$
No Salta	$\rightarrow$	No Salta	NS	N	$0{,}9 \times 0{,}3 \times 0{,}85 = 22{,}95\,\%$
No Salta	$\rightarrow$	Salta	S	S	$0{,}9 \times 0{,}7 \times 0{,}15 = 9{,}45\,\%$
Salta	$\rightarrow$	No Salta	NS	S	$0{,}9 \times 0{,}3 \times 0{,}15 = 4{,}05\,\%$
Salta	$\rightarrow$	Salta	S	N	$0{,}9 \times 0{,}7 \times 0{,}85 = 53{,}55\,\%$

$\emptyset \rightarrow$ Indica que la instrucción de salto no está en la tabla.

b) Mostrar en la tabla adjunta (ver hoja de respuestas) el comportamiento de la BTB cuando predice que **salta** y finalmente el salto **no salta**, indicando qué instrucciones son las que se ejecutan en los ciclos siguientes al salto. Indicar además la penalización que sufre la instrucción de salto en este caso.



c) Si en ausencia de fallos las instrucciones de salto tienen un CPI de 1, indicar el CPI real de las instrucciones de salto dadas las estadisticas obtenidas.

$$\text{CPI} = 1 + (0{,}1 \times 0{,}7 + 0{,}9 \times 0{,}7 \times 0{,}15 + 0{,}9 \times 0{,}3 \times 0{,}15) \times 4$$

$$\text{CPI} = 1 + (0{,}07 + 0{,}0945 + 0{,}0405) \times 4$$

$$\text{CPI} = 1 + 0{,}205 \times 4 = 1 + 0{,}82 = 1{,}82$$

d) Se pretende sustituir el predictor actual por un predictor que utilice 1 bit para la predicción y 1 bit de historia global. La siguiente tabla muestra las estadísticas obtenidas con dicho predictor. Teniendo en cuenta que la probabilidad de que el salto previo sea efectivo se mantiene en un $70{,}00\,\%$, ¿sería interesante la sustitución? ¿Cuál sería el nuevo CPI de las instrucciones de salto?

$$\text{CPI} = 1 + 0{,}3 \times (0{,}07 + 0{,}126 + 0{,}054) \times 4 + 0{,}7 \times (0{,}07 + 0{,}063 + 0{,}027) \times 4$$

$$\text{CPI} = 1 + 0{,}3 \times 0{,}25 \times 4 + 0{,}7 \times 0{,}16 \times 4$$

$$\text{CPI} = 1 + 0{,}3 + 0{,}448 = 1{,}748$$

□