CE4302 – Arquitectura de Computadores II

Multiprocesamiento

PROFESOR: ING. LUIS BARBOZA ARTAVIA

Retos en procesamiento paralelo

- Ancho de banda: cores individuales corren a velocidades mucho más rápidas que la memoria externa.
- Dependencia de hilos e inversión de prioridad: ejecución de un hilo puede atrasarse por un hilo de menor prioridad. Hilos pueden tener dependencias serializadas.
- Contención de cache y *false sharing*: puede haber un overhead de migración.

https://developer.arm.com/documentation/den0013/d/Parallelizing-Software/Performance-issues?lang=en

Retos en procesamiento paralelo

- **Limitación** de paralelismo en programa.
- **Comunicación:** los hilos se pueden requerir comunicarse entre sí.

Limitación de paralelismo

Ley de Amdahl

$$Speedup = \frac{1}{\frac{Fraction_{enhanced}}{Speedup_{enhanced}} + (1 - Fraction_{enhanced})}$$

$Speedup$ = Speedup objetivo
$Fraction_{enhanced}$ = tiempo en modo paralelo.
$Speedup_{enhanced}$ = Speedup debido al **paralelismo** (unidades)

Limitación de paralelismo

Ley de Amdahl

$$Speedup = \frac{1}{\frac{Fraction_{enhanced}}{Speedup_{enhanced}} + (1 - Fraction_{enhanced})}$$

3 casos:

- 1 unidad ejecutando
- N unidades trabajando en paralelo
- Diferentes combinaciones de unidades en diferentes tiempos

Limitación de paralelismo

Queremos llegar a una mejora de 80 con 100 procesadores. ¿Qué fracción debe ser serial del programa original, para llegar a esa mejora?

$$Speedup = \frac{1}{\frac{Fraction_{enhanced}}{Speedup_{enhanced}} + (1 - Fraction_{enhanced})}$$

$Fraction_{enhanced}$ = tiempo en modo paralelo.

Limitación de paralelismo

$$Speedup = \frac{1}{\frac{Fraction_{enhanced}}{Speedup_{enhanced}} + (1 - Fraction_{enhanced})}$$

$$Speedup = 80, Speedup_{enhanced} = 100$$

$$80 = \frac{1}{\frac{Fraction_{parallel}}{100} + \left(1 - Fraction_{parallel}\right)}$$

Limitación de paralelismo

$$0.8 \times Fraction_{parallel} + 80 \times \left(1 - Fraction_{parallel}\right) = 1$$

$$80 + 0.8Fraction_{parallel} - 80Fraction_{parallel} = 1$$

$$79.2Fraction_{parallel} = 79$$

$$Fraction_{parallel} = \frac{79}{79.2}$$

$$Fraction_{parallel} = 0.9975$$

Limitación de paralelismo

$$Fraction_{parallel} = 0.9975$$

$$Fraction_{serial} = 1 - 0.9975 = 0.0025$$

$$0.25\%\ del\ programa\ puede\ ser\ secuencial$$

Limitación de paralelismo

Example Suppose we have an application running on a 100-processor multiprocessor, and assume that application can use 1, 50, or 100 processors. If we assume that 95% of the time we can use all 100 processors, how much of the remaining 5% of the execution time must employ 50 processors if we want a speedup of 80?

$$Speedup = \frac{1}{\frac{Fraction_{enhanced}}{Speedup_{enhanced}} + (1 - Fraction_{enhanced})}$$

$$80 = \frac{1}{\frac{0.95}{100} + \frac{X}{50} + (1 - (0.95 + X))}$$

$$X = 4.79\%$$

Limitación de paralelismo

Amdahl's Law in the Multicore Era

Mark D. Hill and Michael R. Marty

Everyone knows Amdahl's Law, but quickly forgets it.
-Dr. Thomas Puzak, IBM, 2007

Note: These cartoons omit important structures and assume area, not power, is a chip's limiting resource.

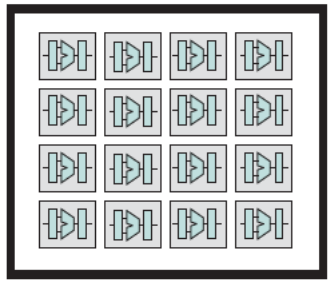


Figure 1. Symmetric
Sixteen 1-BCE cores

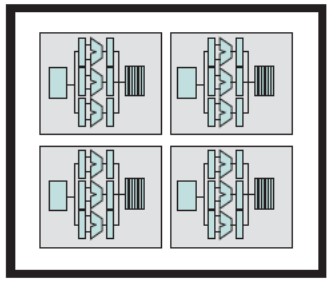


Figure 2. Symmetric
Four 4-BCE cores

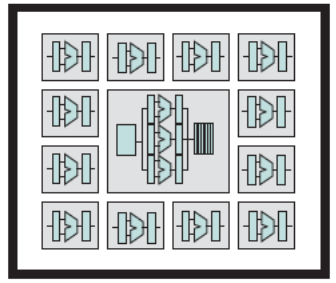


Figure 3. Asymmetric
One 4-BCE core, 12 1-BCE cores

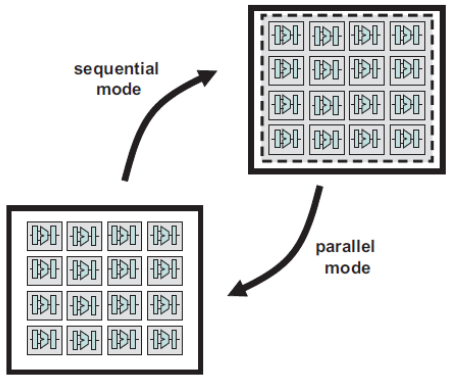


Figure 5. Dynamic
Sixteen 1-BCE cores

$$Speedup_{symmetric}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f \cdot r}{perf(r) \cdot n}}$$

$$Speedup_{asymmetric}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r) + n - r}}$$

$$Speedup_{dynamic}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{n}}$$

Limitación de paralelismo

Amdahl's Law in the Multicore Era

Mark D. Hill and Michael R. Marty

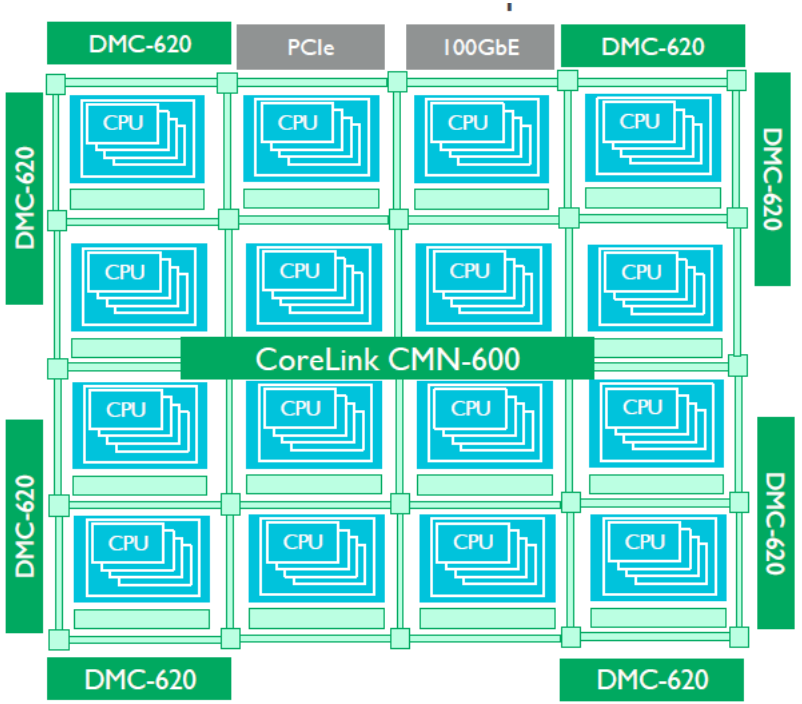
Everyone knows Amdahl's Law, but quickly forgets it.
-Dr. Thomas Puzak, IBM, 2007

Results first reaffirm that seeking greater parallelism is critical to obtaining good speedups. We then show how core designs that are locally inefficient can be globally efficient.

Of course, our model seeks insight by making many simplifying assumptions. The real world is much more complex. Amdahl's simple software model may not hold reasonably for future software. Future mainstream parallel software may also behave differently from today's highly tuned parallel scientific and database codes. Our simple hardware model does not account for the complex tradeoffs between cores, cache capacity, interconnect resources, and off-chip bandwidth. Nevertheless, we find value in the controversy and discussion that drafts of this paper have already stimulated.

Limitación de comunicación

- Latencia entre cores por acceso remoto a datos.
- Depende del mecanismo de interconexión (topologías).
- CPI se ve afectado por un costo de acceso remoto (C_{req}) definido por:
  - Tipo de instrucción / programa
  - Tasa de solicitud de acceso remoto ($Rate_{req}$)
  - Tiempo promedio de acceso remoto ($RemoteAccessCost$)



Limitación de comunicación

Suponga que se tiene un sistema multiprocesador que requiere un tiempo de 200ns para accesos a memoria fuera de chip. Suponga que todos los accesos dentro del chip se toman como aciertos en caché. Al tener un acceso fuera de chip, los procesadores deben esperar el dato, y la frecuencia de los procesadores es de 3GHz. Si el desempeño del computador, en CPI (tomando en cuenta solamente accesos dentro del chip), es de 0.5. ¿Qué tan rápido es tener multiprocesadores sin comunicación entre sí (sin accesos fuera de chip) contra si el 0.2% de las instrucciones requirieran accesos fuera de chip?

Limitación de comunicación

- Calcular los ciclos de reloj por instrucción.

$$CPI = CPI_{base} + Tasa_{accesoAfuera} \times Costo_{accesoRemoto}$$

$$CPI = 0.5 + 0.2\% \times Costo_{accesoRemoto}$$

- Calcular Costo Acceso Remoto

$$\frac{Tiempo_{accesoRemoto}}{Tiempo_{ciclos}} = \frac{200ns}{\frac{1}{3\ GHz}} = 600\ ciclos$$

Limitación de comunicación

- Calcular los ciclos de reloj por instrucción.

$$CPI = 0.5 + 0.2\% \times Costo_{accesoRemoto}$$

- Calcular el CPI

$$CPI = 0.5 + 0.002 \times 600 = 1.7$$

- Comparar con el CPI base.

Sin comunicación: $\frac{1.7}{0.5} = 3.4$ más rápido

Limitación de comunicación

Example Suppose we have an application running on a 32-processor multiprocessor that has a 100 ns delay to handle a reference to a remote memory. For this application, assume that all the references except those involving communication hit in the local memory hierarchy, which is obviously optimistic. Processors are stalled on a remote request, and the processor clock rate is 4 GHz. If the base CPI (assuming that all references hit in the cache) is 0.5, how much faster is the multiprocessor if there is no communication versus if 0.2% of the instructions involve a remote communication reference?

Comunicación entre hilos/cores

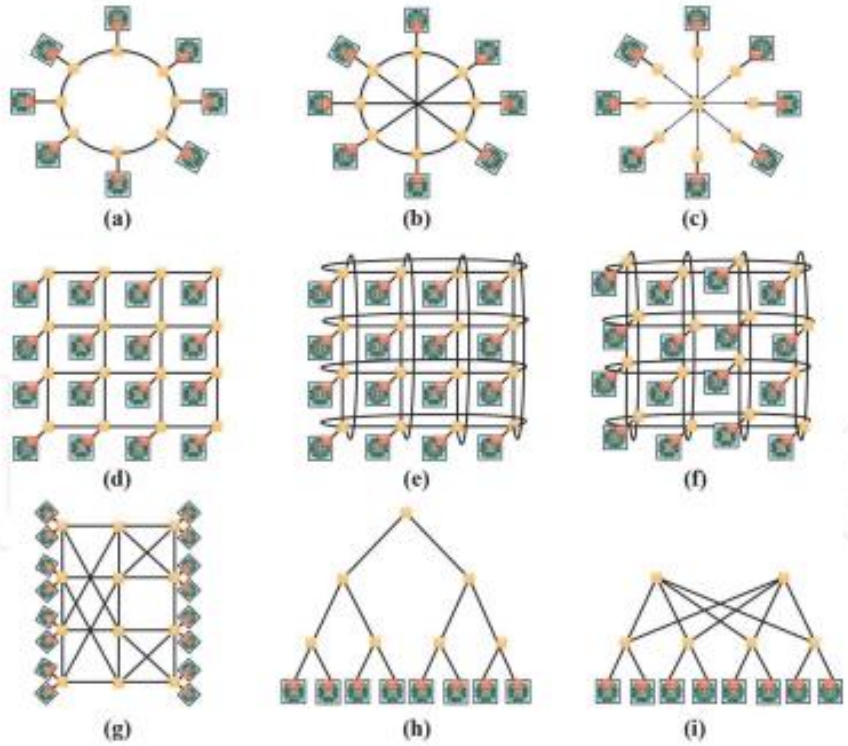


Figure 7.
NoC topologies: (a) ring, (b) octagon (c) star, (d) 4 × 4 mesh, (e) 4 × 4 torus, (f) 4 × 4 folded torus, (g) butterfly, (h) binary tree, (i) fat tree.

Mecanismos para reducir limitaciones

- **Paralelismo:**
  - Algoritmos para mayor paralelismo y uso de FU.
  - Lenguaje o API’s que favorezcan paralelismo.
- **Comunicación:**
  - Entre chips.
  - Uso de multihilo (memoria compartida).
  - Caché compartida para procesadores.
  - Mejoras en caché.

Consideraciones para explorar TLP

- Identificar tareas que pueden hacerse de forma concurrente.
- Identificar maneras en que el trabajo se puede compartir si no hay dependencia de datos.
- Minimizar sincronización y comunicación.
- Superponer la comunicación con la computación

Investigación

- Repasar de Arqui I: caché, políticas de escritura, invalidación y técnicas de implementación.
- ¿Qué instrucciones existen para hacer "*cache maintenance*" en x86, RISC-V, Armv8-9?
- ¿Qué es "*Memory Consistency Model*"?

1. Se tiene un sistema multiprocesador con memoria distribuida para 64 procesadores. Estos se encuentran conectados en una topología de matriz 8x8 sin conexiones en diagonal. La frecuencia de reloj es de 3.3GHz y el CPI en el caso de que no ocurran desaciertos en caché es de 0.5. Asuma que el 98.2% de las instrucciones no necesitan de comunicación remota. En caso de que se requiera una comunicación, el costo viene dado por la función $f(h) = (100 + 10h)$ns donde h es el número de saltos a otros procesadores que debe realizar hasta llegar al procesador que tiene la información. Asuma que todos los enlaces son bidireccionales. Con este escenario responde lo siguiente:

   (a) Calcule el mejor y peor escenario en tiempos de comunicación.

   (b) Calcule el CPI que presenta comunicación para el peor escenario y compárelo contra aquel que no presenta comunicación.

   (c) Compare los resultados obtenidos si se cambia la configuración por una de anillo. ¿Cuál de las dos topologías sería mejor? Justifique la respuesta con los cálculos respectivos.

   (d) ¿Qué modificación haría a cualquiera de las dos topologías, en términos de cantidad de procesadores, para que tengan un mismo CPI en el peor escenario?

Referencias

- Stallings, W. (2003). Computer organization and architecture: designing for performance. Pearson Education India.
- Hennessy, J. L., & Patterson, D. A. (2011). Computer architecture: a quantitative approach. Elsevier.

CE4302 – Arquitectura de Computadores II

Multiprocesamiento

PROFESOR: ING. LUIS BARBOZA ARTAVIA