Análisis de “Using FPGA Devices to Accelerate Biomolecular Simulations”

Lic. Eduardo A. Sanchez

Problema generalizado a atacar

- Aplicación compleja:
  - Mucho poder de cómputo.
  - Manejando un gran volumen de datos.
  - Diseñada, implementada, “testeada” y ya probada “empíricamente” en lenguaje de alto nivel.
  - Concebida desde su origen para correr sobre procesadores de propósito general.

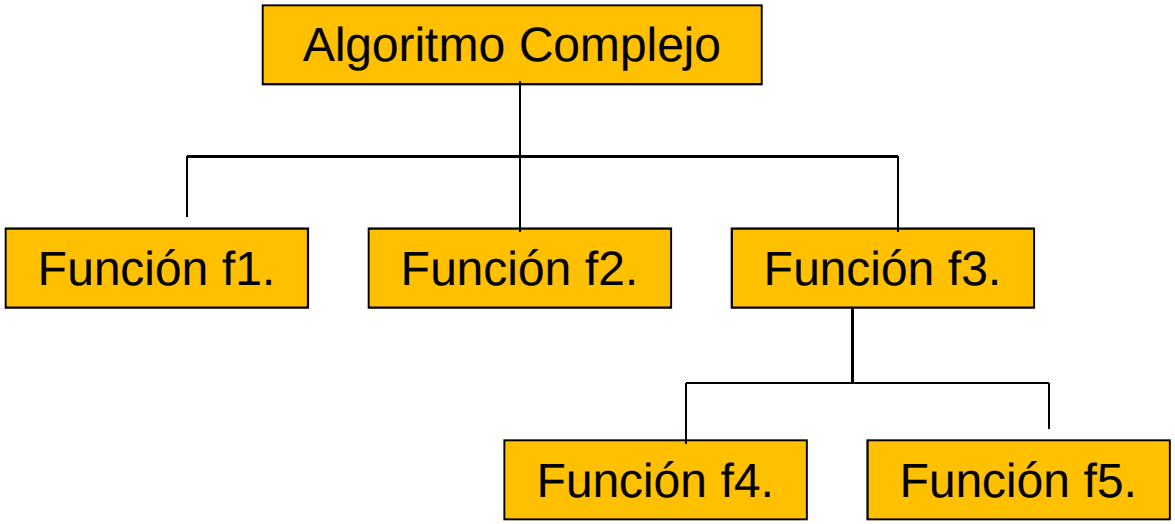
Técnica Propuesta

- Estudiar el algoritmo conocido en HLL.
- Analizar su composición interna.
  - Que partes consumen más tiempo.
  - Que partes consumen más recursos.
  - Que partes son secuenciales.
  - Que partes pueden paralelizarse.
  - Se pueden utilizar técnicas de encauzamiento?

Técnica Propuesta (cont.)

- Implementar las distintas partes en FPGAs e interconectarlas, dejando una como líder.
- Conectar la computadora con la FPGA líder para que le envíe los datos “crudos” y reciba los resultados.
- Hacerle creer al algoritmo que corre en la computadora, que aún lo sigue haciendo.
  - Tanto para el envío de datos desde/hacia FPGA.
  - Como para las llamadas a sus funciones.

El algoritmo Biomolecular

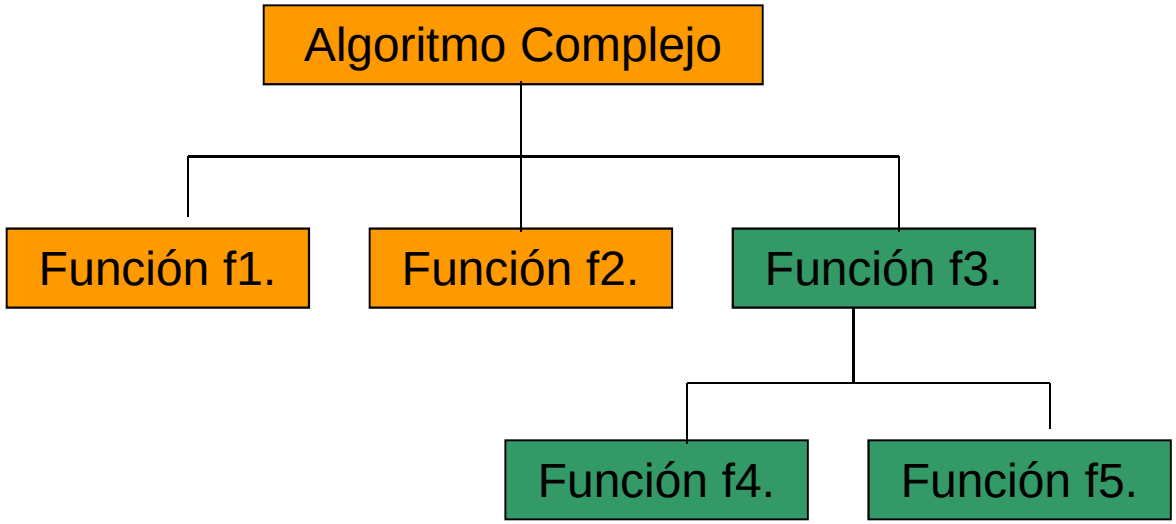


En cada paso, se llama a f1, f2 y f3. La cantidad de llamadas a f4 depende del Número de átomos. f5, se ejecuta el doble de veces que f4.

El algoritmo Biomolecular (cont.)

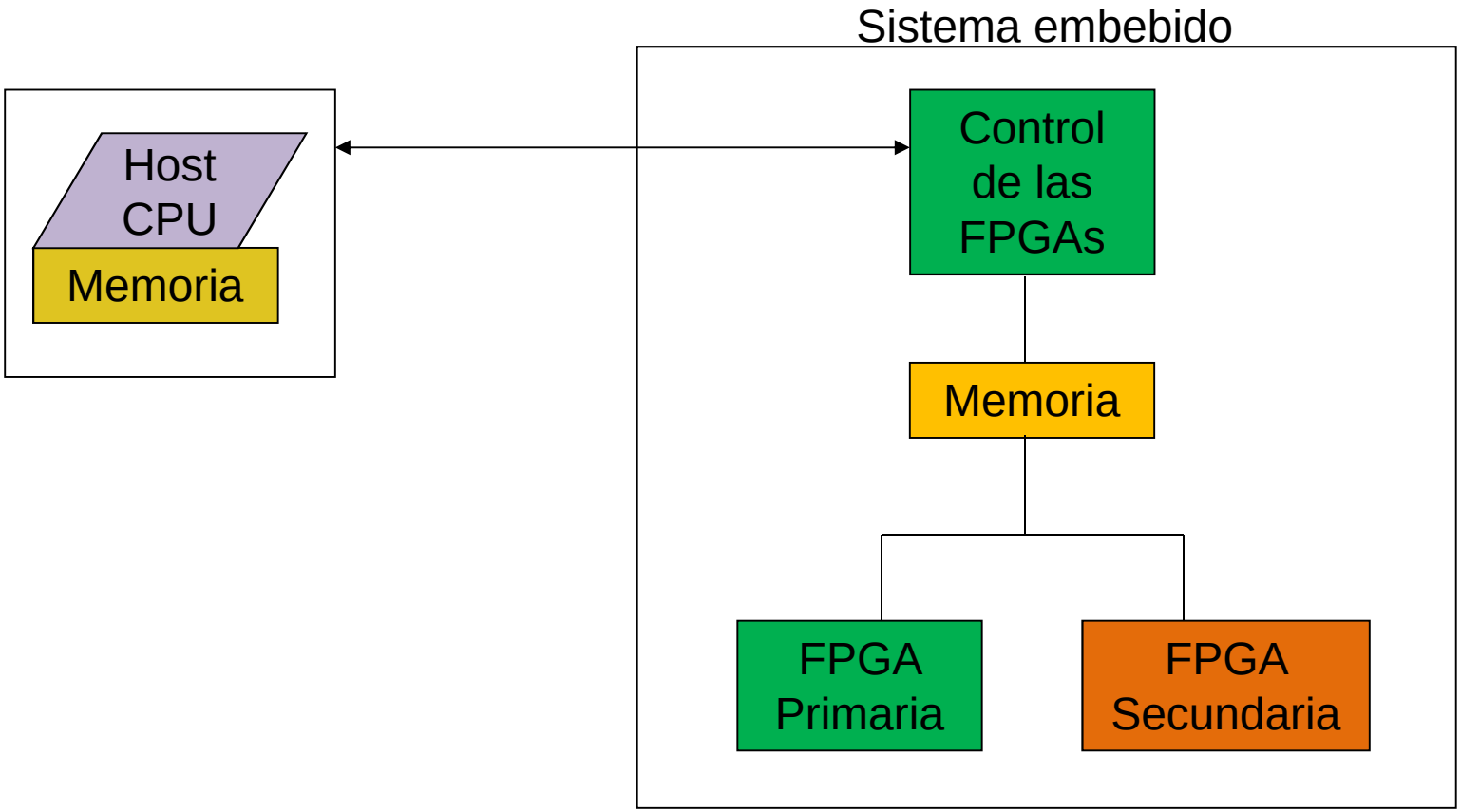
- Inicialmente decidieron portar a la FPGA solo la parte que consumiera más tiempo.
- Utilizando *gprof (herramienta de profiling utiliza instrumentación y muestreo),* calcularon las contribuciones de f1 a f5, identificando f4 como la función más cara.
- Conociendo cuantas veces se invocan las funciones y su costo, les permitió optimizar la distribución del código en la FPGA.

El algoritmo Biomolecular (cont.)



Decidieron mapear las funciones f3, f4 y f5 directamente a una FPGA auxiliar.

El algoritmo Biomolecular (cont.)



Algunas técnicas utilizadas para mejorar aún más la performance

- Ventajas principales de utilizar FPGAs:
  - Encauzamientos de varios ordenes.
  - Ejecuciones concurrentes. (ya no hay que esperar que se termine f5 para volver a pedir datos).
  - Data streaming. La FPGA primaria utiliza DMA y controla el movimiento de datos con la secundaria.

Análisis de la performance

- Contra el primer algoritmo, logran aumentar la velocidad de 3 a 4X.
- Luego de un estudio más detenido, se dan cuenta que el cuello de botella esta en el movimiento de datos.
- Aplican algunas técnicas generales: prefetch-poststore, OpenMP, etc.

Análisis de la performance (cont.)

- Por último, mejoran el algoritmo:
- Caracterizando/mejorando el manejo de *arrays*:
- Inicializar una sola vez.
- Si no se modifica, no hace falta DMA al host y se puede paralelizar aún más.

Análisis de la performance (cont.)

- Al aplicar todo lo anterior logran bajar el costo de transferir datos del 70% del tiempo de ejecución total a solo el 5%!
- En concreto, utilizando el algoritmo en un sistema con dos microprocesadores 2.8GHz Xeon, toma 10 días. Utilizando solo FPGA para acelerar, toma 5 días. Utilizando FPGAs y Host, toma solo 3 días.

Conclusiones

- El análisis del algoritmo con *gprof* es bueno.
- La utilización HW dedicado para acelerar el procesamiento también esta bueno.
- Si bien esta idea no es nueva, si lo es el utilizar la PC como host e interface al exterior.

Conclusiones (cont.)

- Algo que no mencionan es el bajo consumo.
- Muchas veces para mejorar la performance se aumenta el consumo (overclocking) y ellos no lo hacen.
- Por el contrario, lo bajan, utilizando menos tiempo de procesamiento, integrando todo en uno o dos chips de bajo consumo y de gran nivel de integración.

Preguntas?

Referencias

- “Using FPGA Devices to Accelerate Biomolecular Simulations”, Sadaf Alam, Pratul Agarwal, Melissa Smith, Jeffrey Vetter, David Caliga.
- “Accelerate system performance with hibrid multiprocessing and FPGAs”, Dan Isaacs, Ed Trexel and Bruce Karsten.
- DSP/FPGA co-processing demos 20x acceleration using SW-to-HW desing flow.