High Performance Computing and Architectures Group

http://www.hpca.uji.es
Universidad Jaime I de Castellón

ANACAP, noviembre de 2008

Generalidades

- Creado en 1991, al mismo tiempo que la Universidad Jaime I

- Investigación en el uso de técnicas de computación de altas prestaciones en aplicaciones de ciencia e ingeniería:
  - Optimización en procesadores de propósito general y hardware específico (FPGAs, GPUs y otros aceleradores)
  - Paralelización en sistemas con memoria compartida (SMP, NUMA y multicore) y distribuida (clusters)

- 11 PDI doctores, 6 PDI no doctor
- Línea de investigación relacionada con GPUs: 6 PDI

Proyectos en curso

- “Diseño y desarrollo de una biblioteca de cálculo sobre GPUs”. Fundación Caixa-Castelló/Bancaja y UJI, 2008-2009
- “Desarrollo de una librería de cálculo matricial para procesadores gráficos”. Generalidad Valenciana, 2008

- “COMPARHE: Computación en paralelo y sistemas heterogéneos”. CICYT, 2005-2008
- “COPABIB: Construcción y optimización automáticas de bibliotecas de computación científica”. CICYT, 2009-2011
- “CAPAP-H: Red de computación de altas prestaciones sobre arquitecturas paralelas heterogéneas”. MEC, 2008-2009

- 2008 NVIDIA Professor Partnership Award

Plataformas de cálculo disponibles

- http://www.hpca.uji.es/?q=node/4
- http://www.hpca.uji.es/gpgpu/machines.html

- A finales de 2008:
  - PCs con 1 GPUs de NVIDIA: GTX280, 8800 Ultra, T10P
  - PCs con múltiples GPUs de NVIDIA: 4x9800GX2, Tesla S870, Tesla S1070
  - Cluster de 5 PCs: Tesla C1060

Líneas de investigación

- Computación matricial densa (CMD)
- Procesamiento de imágenes biomédicas (PIB)

- Motivación
- Estado
- Líneas de desarrollo y resultados

CMD: Motivación

- Los problemas de CMD aparecen en casi todas las aplicaciones científicas y de ingeniería con base numérica:
  - Resolución de sistemas de ecuaciones lineales
  - Problemas de mínimos cuadrados
  - Cálculo de valores propios y singulares de matrices

- El coste de resolver de manera fiable estos problemas es cúbico con la dimensión del problema. Cuando éste último es de tamaño considerable, resulta necesario utilizar técnicas de computación de altas prestaciones
  - O(100-1,000) es habitual
  - O(10,000-100,000) también se da en algunas pocas aplicaciones

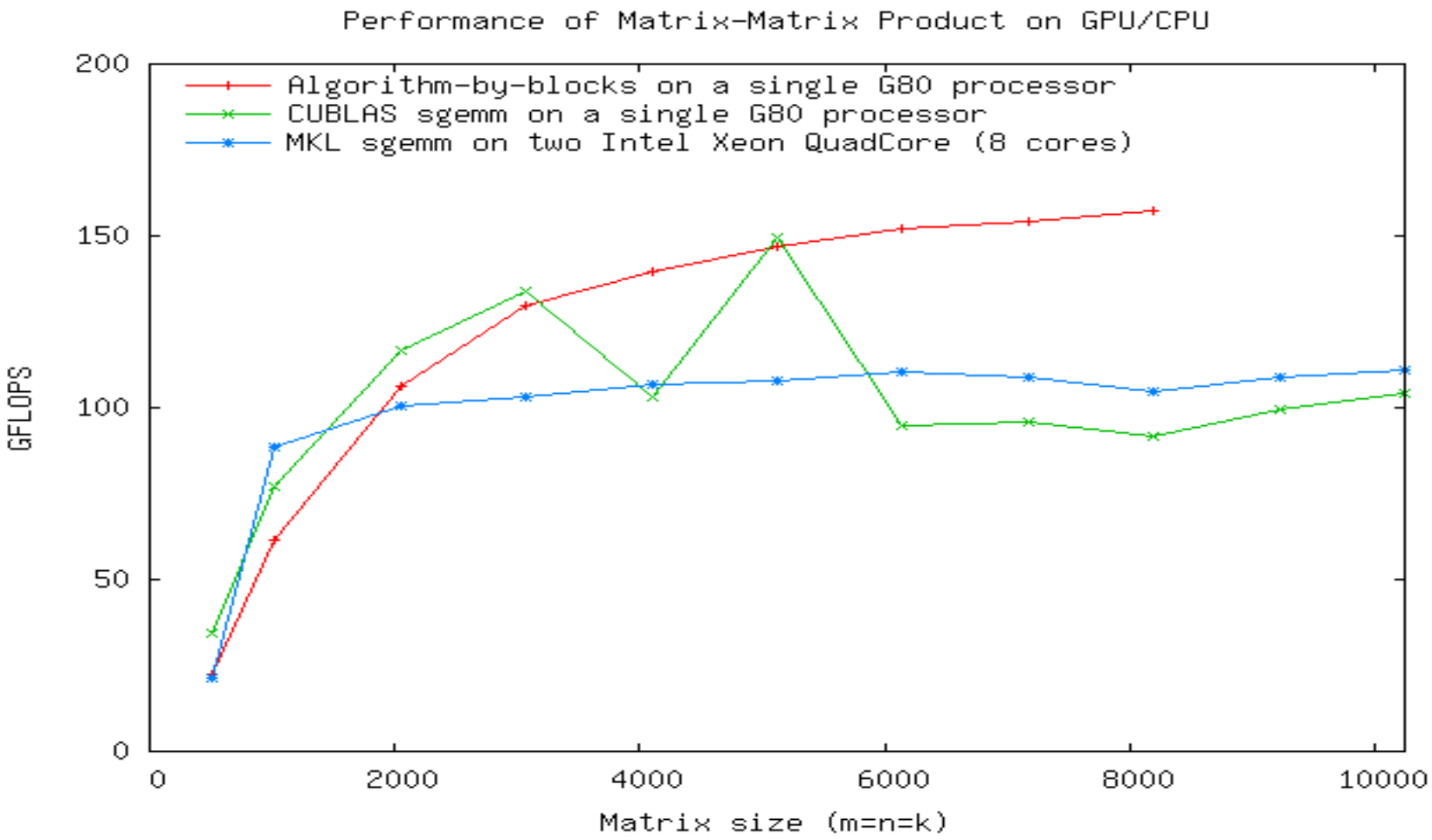
CMD: Estado

- Existen bibliotecas para computación matricial sobre plataformas “tradicionales”:
  - BLAS específico para procesadores de Intel, AMD, IBM, etc.
  - Algunas de estas implementaciones son multihebra → procesadores multinúcleo y multiprocesadores con memoria compartida (MMC)
  - LAPACK/FLAME para operaciones más avanzadas, haciendo uso de BLAS
  - ScaLAPACK/PLAPACK para clusters de computadores

- Algunas implementaciones de BLAS ya disponibles para aceleradores gráficos: NVIDIA, AMD, ClearSpeed.
  - No hay bibliotecas para operaciones avanzadas (tipo LAPACK)
  - No hay bibliotecas para arquitecturas multi-GPU
  - No hay bibliotecas para clusters de computadores equipados con GPUs

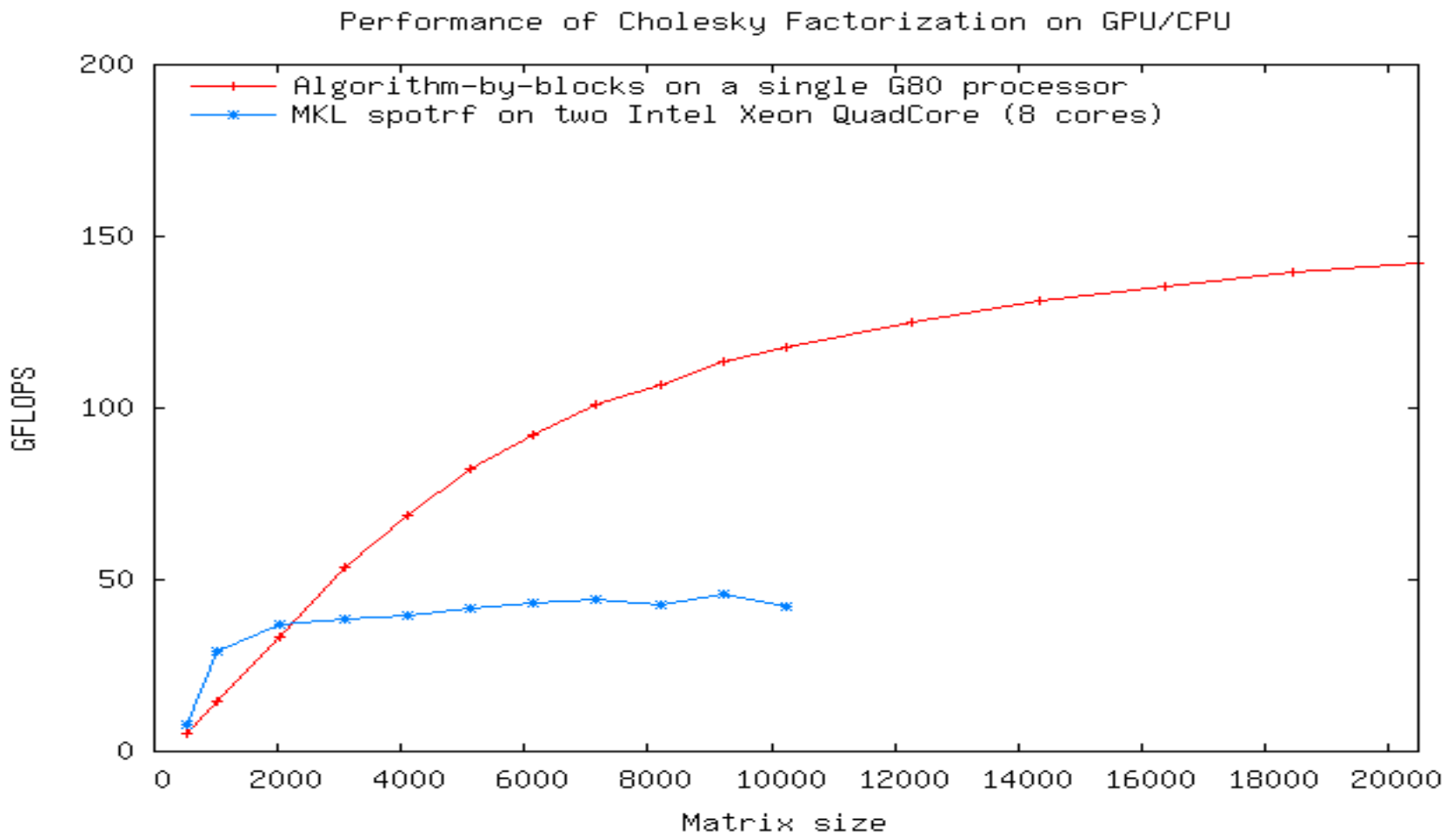
CMD: Líneas de desarrollo y resultados

- Optimización de rutinas de CUBLAS



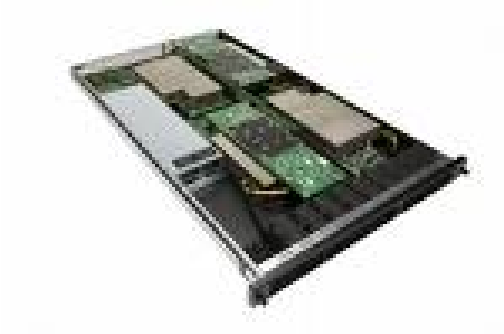
CMD: Líneas de desarrollo y resultados

- Resolución de sistemas de ecuaciones lineales



CMD: Líneas de desarrollo y resultados

- Desarrollo de bibliotecas para plataformas multi-GPU (Tesla S870, S1070, etc.).
  En colaboración con *The University of Texas at Austin* (proyecto FLAME)
  - Memoria en GPUs distribuida
  - Sin coherencia hardware
  - Transferencias a través de la RAM



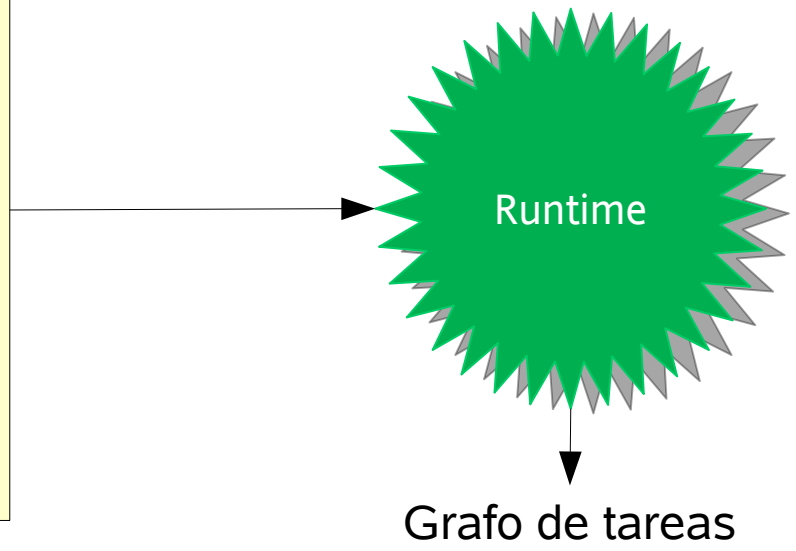
  - ¿Programación como una arquitectura con memoria distribuida (paso de mensajes)?
  - La facilidad de programación es la clave
  - Algoritmo independiente de la arquitectura
  - Run-time encargado de extraer el paralelismo

CMD: Líneas de desarrollo y resultados

- Desarrollo de bibliotecas para plataformas multi-GPU

  Primera etapa: Ejecución simbólica del código

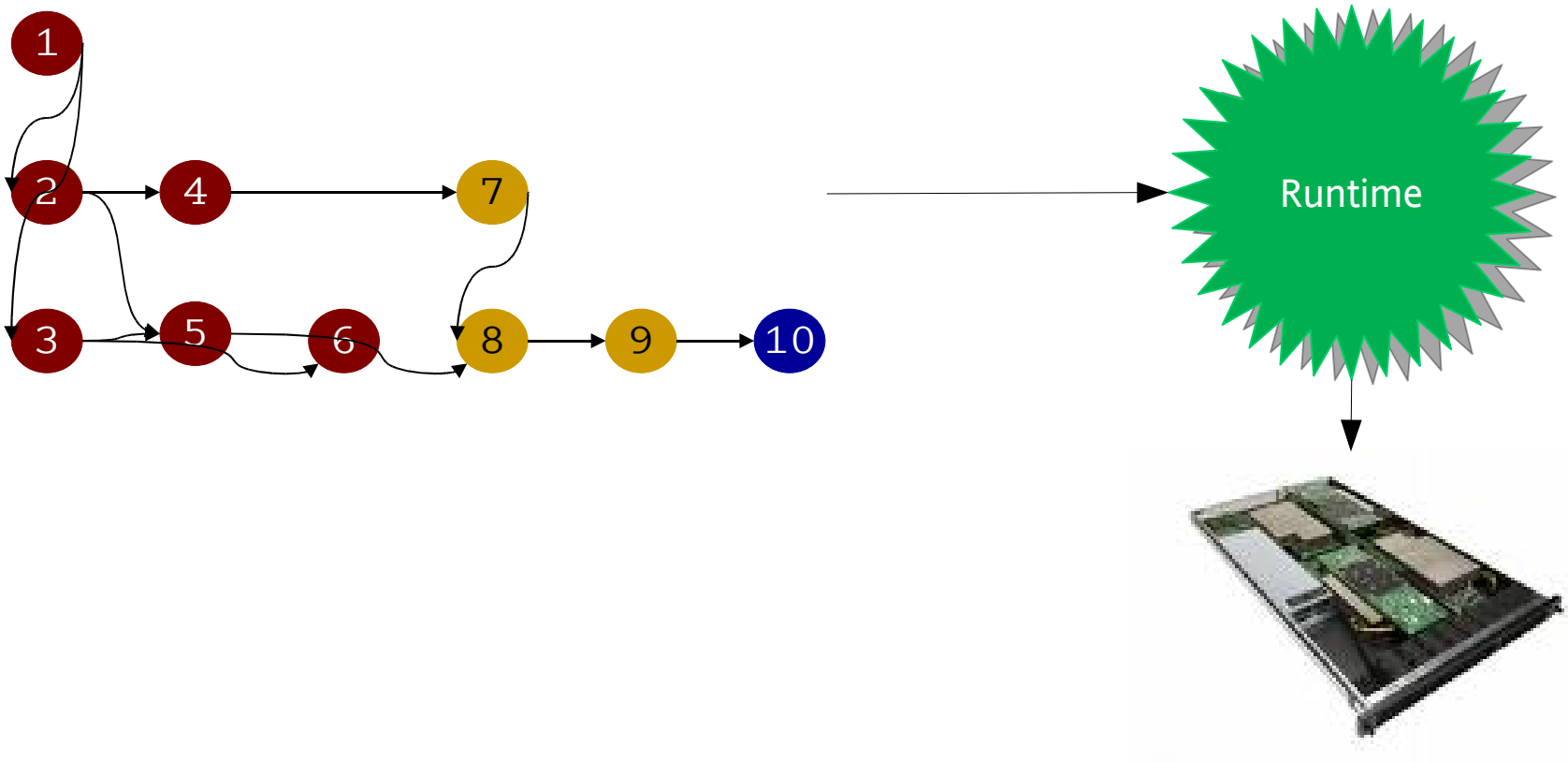
```
while ( /*...*/ ) { /*...*/
    FLAG_Chol_unb_var3( A11 );
    FLAG_Trsm( /*...*/
               FLA_ONE, A11, A21 );
    FLAG_Syrk( /*...*/
               FLA_MINUS_ONE, A21,
               FLA_ONE,       A22 );
/*...*/ }
```



CMD: Líneas de desarrollo y resultados

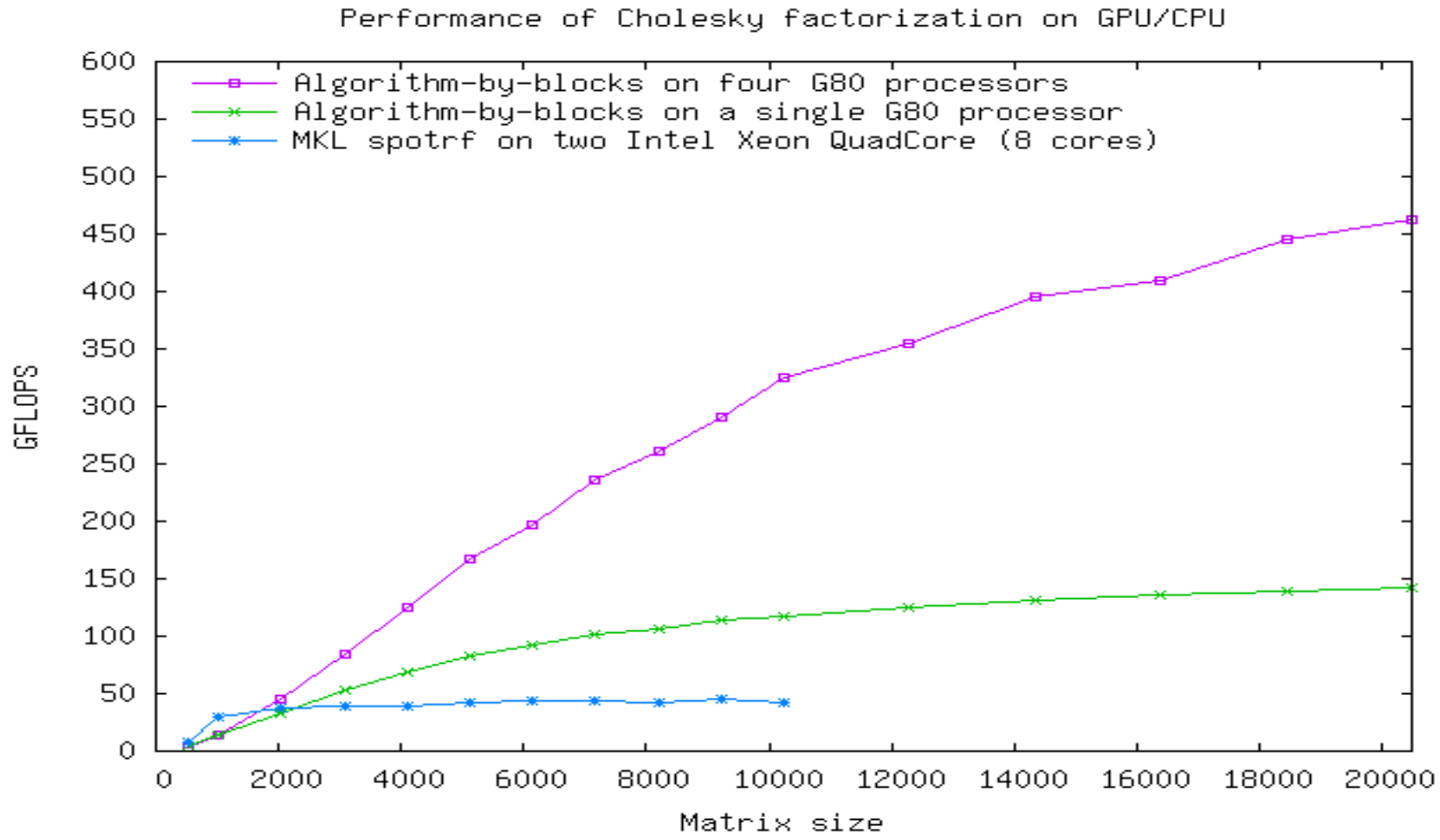
- Desarrollo de bibliotecas para plataformas multi-GPU

  Segunda etapa: Ejecución real del grafo de tareas (planificación y asignación)



CMD: Líneas de desarrollo y resultados

- Desarrollo de bibliotecas para plataformas multi-GPU



Speed-up de 3.25 respecto al algoritmo en un G80

CMD: Líneas de desarrollo y resultados

- Paralelización automática de códigos para plataformas con múltiples aceleradores, posiblemente heterogéneos

  En colaboración con el *Barcelona Supercomputing Center*

  - Extensión de StarSs con GPUSs:

```
#pragma css task input(A[b][b], B[b][b]) inout(C[b][b]) device CUDA
__global__ void matmul(float *A, float *B, float *C) {
    /* Código CUDA */
    /*...*/
}
```

CMD: Líneas de desarrollo y resultados

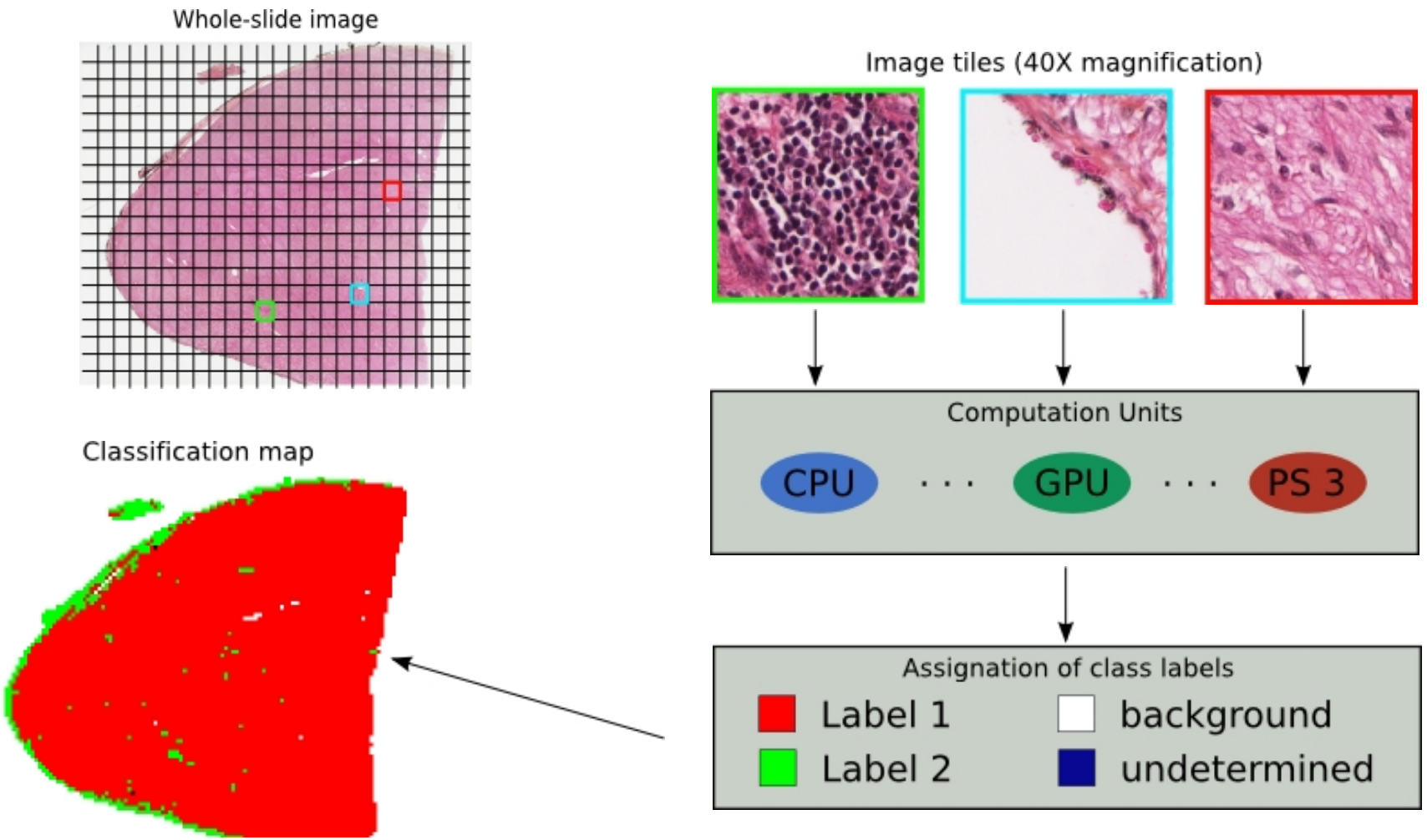
- Otros:
  - Uso de aceleradores en la resolución de problemas CMD en disco
  - Adaptación de bibliotecas CMD para clusters de computadores equipados con aceleradores

PIB: Motivación

- Campo de aplicación: detección temprana de tejido canceroso
- Justificación: muestras digitalizadas de tejido de enormes dimensiones
  - Muestras típicas: 120K x 120K píxeles
  - Más de 40 Gbytes por imagen
  - Múltiples muestras por paciente
- Problemática: el tiempo de procesamiento es vital para el diagnóstico
  - Implementaciones actuales en Matlab o C/C++ sobre multicores
  - El procesamiento puede llevar días, incluso semanas por paciente
- Solución: aplicación de técnicas GPGPU
  - Reducción a minutos en clusters de CPU/GPU
- En colaboración con la *Universidad de Málaga* y *Ohio State University*

PIB: Estado

- Proceso de clasificación



PIB: Estado

- **BIPGPU**: biblioteca para procesamiento de imágenes biomédicas sobre GPUs
  - Biblioteca desarrollada en CUDA
  - Comprende todas las fases del proceso:
    - Conversión de color: RGB, XYZ, LA*B*, ...
    - Operadores para extracción de características:
      - Operador LBP
      - Matrices de coocurrencia
      - Momentos de Zernike
    - Clasificadores
  - Kernels optimizados atendiendo a la arquitectura CUDA
  - Líneas de trabajo futuras:
    - Procesamiento en el dominio de la frecuencia
    - Wavelets, filtros de Gabor, ...

PIB: Resultados

- El tipo de operador determina la ganancia a la hora de utilizar GPUs

- Operadores totalmente orientados a flujos de datos:
  - Conversiones de color: **speedup 200x**

- Operadores de naturaleza recursiva:
  - Momentos de Zernike: **speedup 2x**

- Operadores con baja/media carga computacional:
  - LBP: **speedup 50x**
  - Matrices de coocurrencia: **speedup 80x**

High Performance Computing and Architectures Group

http://www.hpca.uji.es
Universidad Jaime I de Castellón

Contacto: Enrique S. Quintana (quintana@icc.uji.es)