

Herramientas para el estudio de prestaciones en clusters de computación científica, aplicación en el Laboratorio de Computación Paralela

Virginio García López

Ingeniería en Informática

Febrero 2009

Motivación de este proyecto

- ▶ Interesa conocer las prestaciones de clusters
 - ▶ De manera automática
 - ▶ Tanto de computación paralela como secuencial (álgebra lineal)
 - ▶ Interés en el cluster SOL
- ▶ Material útil para el Grupo de Computación Paralela.
 - ▶ Manuales, herramientas, mediciones, gráficos, etc.
- ▶ Utilidad de este tipo de proyectos: proyectos anteriores...
 - ▶ Software para Computación Matricial paralela: Librerías de Paso de Mensajes, 1996 (Seminario MPICH/MPI).
 - ▶ Software para Computación Matricial Paralela: Librerías de Computación Matricial, 1997 (Seminario BLAS).
 - ▶ Estudio de las Prestaciones de Librerías de Paso de Mensajes y de Computación Matricial, 1998.

Objetivos de la herramienta

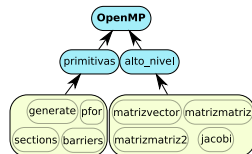
- ▶ Conocer las prestaciones de un cluster.
- ▶ Comparar distintos clusters de computación científica.
- ▶ Automatizar un conjunto de experimentos.
- ▶ Facilitar el análisis y la interpretación mediante gráficos.
- ▶ Facilitar el cálculo de megaflops, *speedup*, eficiencia, coste, y su documentación.
- ▶ Herramienta fácilmente configurable, ampliable, se puede lanzar sólo la parte que interese.

Software científico: paralelo y secuencial

- ▶ OpenMP
 - ▶ Estándar para la programación en memoria compartida.
 - ▶ Basado en directivas: el programador indica las partes paralelizables (`fork-join`).
- ▶ MPI
 - ▶ Estándar para la programación en memoria distribuida.
 - ▶ Basado en paso de mensajes.
 - ▶ Implementación LAM/MPI.
- ▶ BLAS (Basic Linear Algebra Subprograms)
 - ▶ Biblioteca de funciones que llevan a cabo operaciones básicas con matrices y vectores.
 - ▶ Gran eficiencia y difusión. Estándar de facto.
- ▶ LAPACK (Linear Algebra PACKage)
 - ▶ Biblioteca que, basándose en BLAS, resuelve problemas de más alto nivel.

Mediciones con OpenMP

- ▶ ¿Cuál es el rendimiento de OpenMP?
- ▶ ¿Qué compilador ofrece mejor rendimiento?
- ▶ ¿Con qué optimizaciones?



OpenMP

▶ primitivas

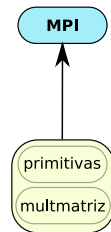
- ▶ **generate**: Primitivas de generación de hilos
- ▶ **pfor**: Primitiva para paralelizar bucles for
- ▶ **sections**: Primitiva sections y section
- ▶ **barriers**: Primitiva barrier

▶ alto_nivel

- ▶ **matrizvector**: Multiplicación matriz por vector
- ▶ **matrizmatriz**: Multiplicación matrices almacenadas por filas
- ▶ **matrizmatriz2**: Multiplicación de matrices $A \times B$, con B por columnas
- ▶ **jacobi**: Relajación de Jacobi

Mediciones con MPI

- ▶ ¿Cuál es el rendimiento de MPI?
- ▶ ¿Qué rendimiento ofrecen las arquitecturas heterogéneas?.
- ▶ Permite observar el rendimiento de la red y trabajar con arquitecturas heterogéneas en MPI

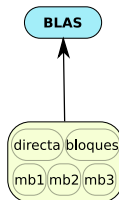


MPI

- ▶ **primitivas:** Primitivas de MPI
(MPI_Send / MPI_Recv, MPI_Isend / MPI_Irecv, MPI_Bcast, MPI_Gather, MPI_Scatter)
- ▶ **multmatriz:** Multiplicación de matrices

Mediciones con BLAS

- ▶ ¿Cuál es el rendimiento de BLAS?
- ▶ ¿Cuál es el comportamiento de cada nivel?
- ▶ ¿Cuál es el comportamiento de cada implementación?



Implementaciones: Reference, Goto, ATLAS, Threaded-ATLAS

Niveles: 1 (vectores), 2 (matriz-vector), 3 (matriz-matriz)

Blas

- ▶ **directa:** Multiplicación directa de matrices (sin BLAS)
- ▶ **bloques:** Multiplicación de matrices por bloques (sin BLAS)
- ▶ **mb1:** Multiplicación de matrices llamando a BLAS de nivel 1
- ▶ **mb2:** Multiplicación de matrices llamando a BLAS de nivel 2
- ▶ **mb3:** Multiplicación de matrices llamando a BLAS de nivel 3

Mediciones con LAPACK

- ▶ ¿Cuál es el rendimiento de LAPACK?
- ▶ ¿Cuál de las distintas implementaciones ofrece el mejor rendimiento?
- ▶ ¿Son las mismas que en BLAS?
- ▶ ¿Cuál es el rendimiento con y sin bloques de la factorización LU?

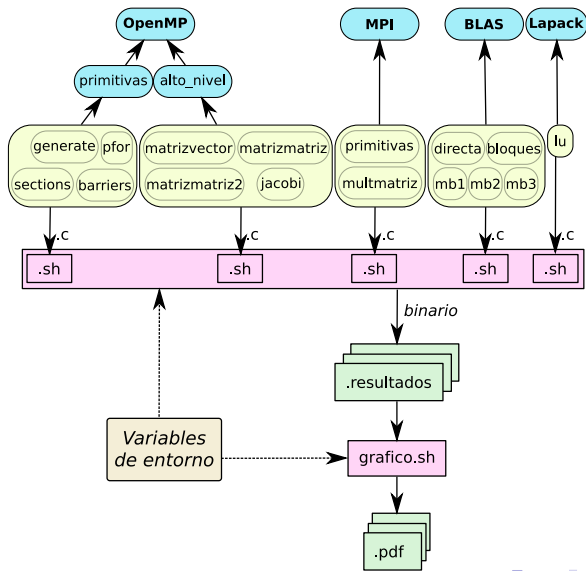


Implementaciones: Reference, ATLAS

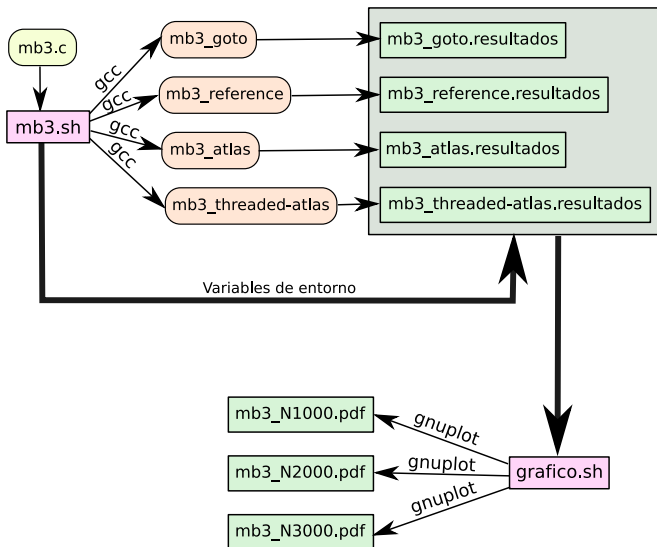
Lapack

- ▶ **lu**: Factorización LU (dgetf2, dgetrf)

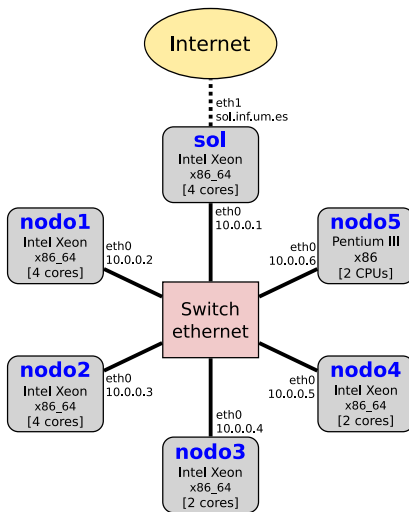
Diagrama de estructura



Ejemplo BLAS-3

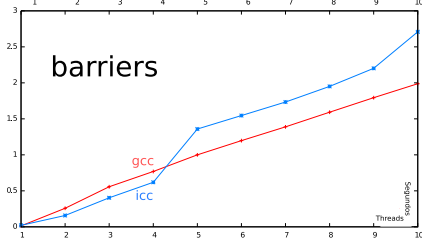
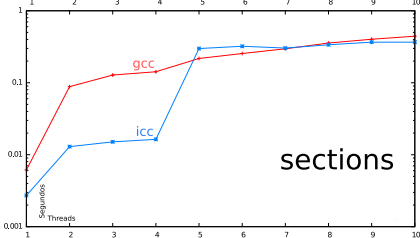
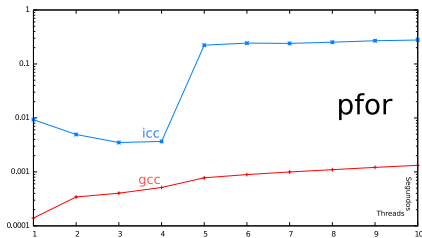
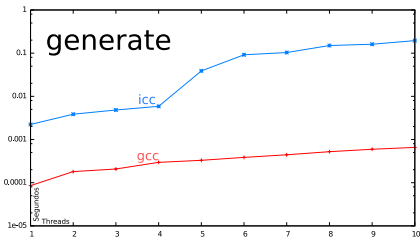


El cluster SOL



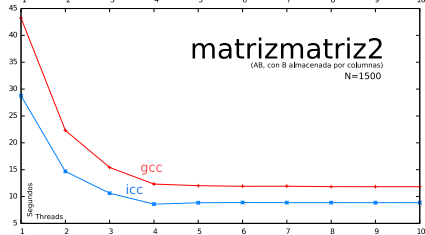
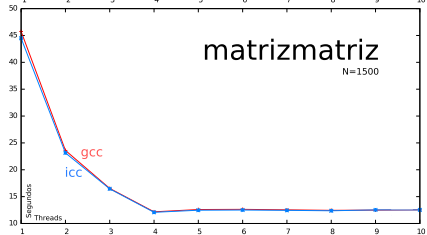
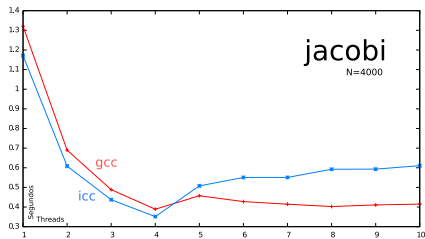
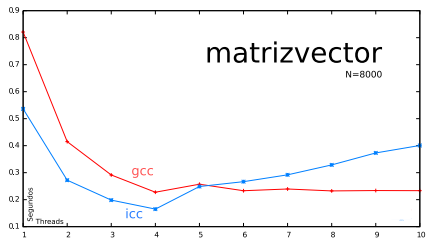
Prestaciones de OpenMP

Primitivas: generate, pfor, sections, barriers ($N=5000$)



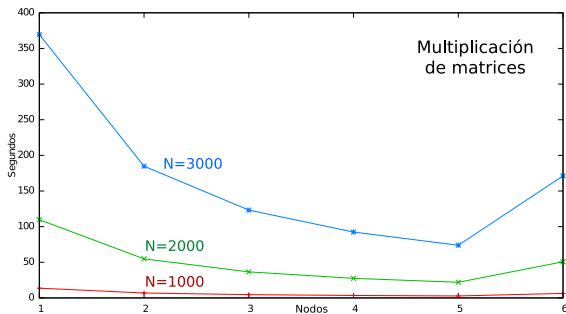
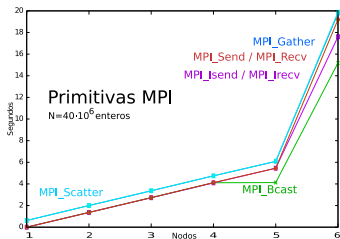
Prestaciones de OpenMP

Rutinas de alto nivel: matrizvector, jacobi, multmatriz, multmatriz2



Prestaciones de MPI

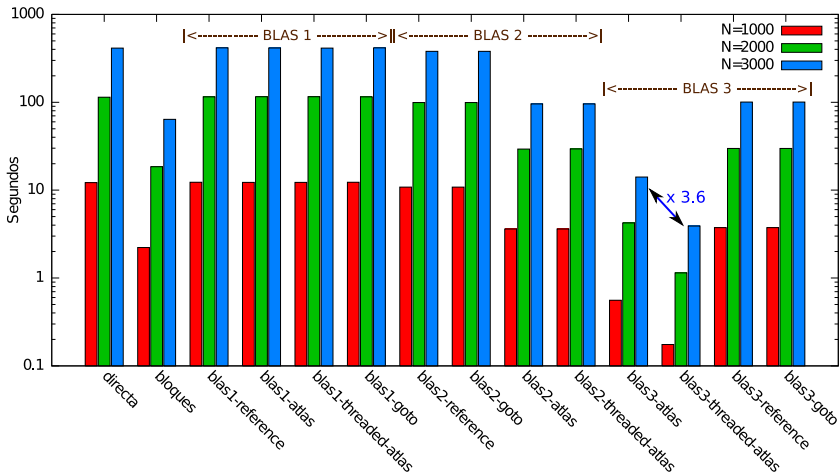
Primitivas y multiplicación de matrices



Prestaciones de BLAS

Multiplicación de matrices: directa, bloques, BLAS-1, BLAS-2, BLAS-3

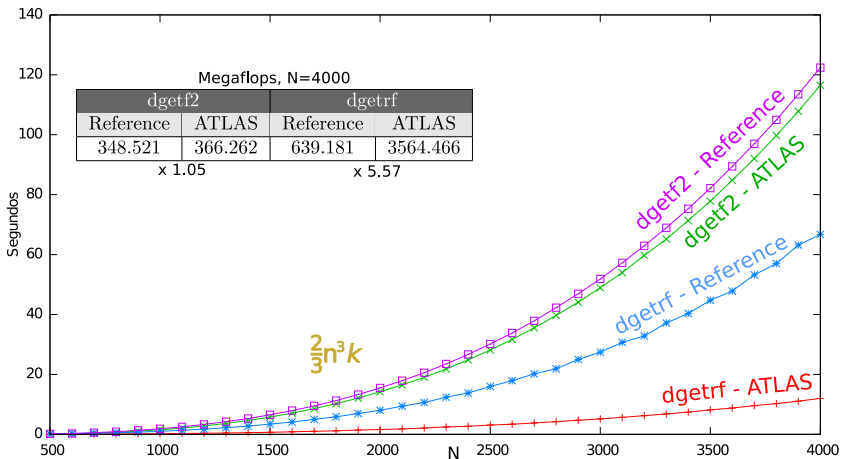
Implementaciones: Reference, ATLAS, Threaded-ATLAS, Goto



Prestaciones de LAPACK

Factorización LU: Sin bloques (dgetf2) y con bloques (dgetrf)

Implementaciones: Reference, ATLAS



Conclusiones

▶ Herramienta

- ▶ **Herramienta**: Útil para medir prestaciones y comparar clusters. Parte gráfica y otros scripts de ayuda.
- ▶ **OpenMP**: Soporte reciente en gcc.
- ▶ **MPI**: Actualiza otros PFC.
Arquitecturas heterogéneas.
- ▶ **BLAS y LAPACK**: Comparación de implementaciones.

▶ Cluster SOL

- ▶ **OpenMP**: Compilador icc cuando hasta un hilo por core, gcc en otro caso.
- ▶ **MPI**: Excelente *speedup* con nodos homogéneos.
No conviene usar nodo antiguo.
- ▶ **BLAS**: Mejor Threaded-ATLAS.
- ▶ **LAPACK**: Mejor ATLAS.

Vías futuras

- ▶ Aplicar las herramientas a otros clusters.
- ▶ Ya se está aplicando en el cluster **Hipatia**.
- ▶ Ampliación de la herramienta: ScaLAPACK
- ▶ Mejora y adaptación de las herramientas.
- ▶ Mejoras en la automatización, en la generación de documentación, ...
- ▶ Añadir otros algoritmos.



Herramientas para el estudio de prestaciones en clusters de computación científica, aplicación en el Laboratorio de Computación Paralela

http://www.um.es/pcgum/PFCs_y_TMs/