



UNIVERSIDAD  
DE BURGOS

# Medida del rendimiento

## Benchmarks

V 1.0

José M. Cámara  
([checam@ubu.es](mailto:checam@ubu.es))



# Motivación

- ❑ El rendimiento debe ser evaluado para:
  - Valorar el comportamiento del sistema.
  - Comparar varios sistemas.
  - Optimizar la utilización.
  - Eliminar cuellos de botella.
- ❑ Cuantificar el rendimiento es una tarea abrumadora:
  - Los sistemas son distintos entre sí.
  - Son muy complejos.
  - Manejan una amplia variedad de aplicaciones y datos.

# Métricas

- ❑ Latencia: tiempo para completar una acción: enviar un mensaje, ejecutar un programa, responder a una petición... En función de la situación se puede expresar también como tiempo de ejecución o como tiempo de respuesta.
- ❑ Throughput: tareas completadas por unidad de tiempo: instrucciones, mensajes, consultas...
- ❑ Throughput =  $1 / \text{latencia}$  solo cuando no se produce solapamiento (pipe-line de instrucciones o de mensajes). Si no  $\text{throughput} > 1 / \text{latencia}$ .

# Benchmarks

- ❑ Concepto: programa de aplicación empleado para cuantificar el rendimiento de un computador.
- ❑ Objetivo: los resultados deben ser numéricos, objetivos y equitativos.
- ❑ Tipos:
  - Programas reales.
  - Sintéticos.
  - Kernels.
  - Juguetes.
  - Suites.

# Benchmarks

- Los programas reales parecen a priori la opción más objetiva, pero a menudo sus resultados son difíciles de interpretar y poco extrapolables ya que muchos sistemas se ven afectados por ellos de forma incierta y variable.
- Los benchmarks sintéticos se diseñan para reflejar el rendimiento de ciertos subsistemas.
- Para evaluar todos los subsistemas lo adecuado es emplear una suite de programas sintéticos.
- Los kernels son similares a programas reales. Eliminan lo que no es relevante, como la interfaz de usuario, los resultados del cálculo, etc.
- Los juguetes son programas cortos que producen resultados ya conocidos por el usuario.

# Evaluación de resultados

- ❑ Las suites de benchmarks están compuestas por programas diferentes. El rendimiento de los sistemas puede ser diferente bajo cada uno de ellos.
- ❑ Una comparación directa no es posible en este caso. Se utilizan métricas más elaboradas:
  - Media aritmética:  $AM = \frac{\sum_i^N r_i}{N}$ . No es acertada cuando las pruebas no están relacionadas. La más larga tiende a marcar la tendencia.
  - Media geométrica:  $GM = \sqrt[N]{\prod_i^N r_i}$ . Su utilidad no está muy clara.
  - Media armónica:  $HM = \frac{N}{\sum_i^N \frac{1}{r_i}}$
- ❑ Los resultados  $r_i$  pueden ser tiempos de ejecución absolutos (o sus inversos) o incrementos (referidos a un sistema concreto).

# Comparación de medias

- ❑ Pensemos en un conjunto de pruebas en las que los tiempos de ejecución son: 1s, 4s y 10s.
- ❑ La media aritmética es 5s. Es correcto pero, teniendo en cuenta que los programas cortos se pueden ejecutar un número mayor de veces que los largos, quizá deberían pesar más en la media.
- ❑ La media armónica es  $3/1,35 = 2,22s$ . Esto podría reflejar mejor el rendimiento global del sistema.

# Selección de benchmarks

- Sistemas distintos requieren benchmarks diferentes.
- Vamos a considerar dos tipos de servidores: supercomputadores y centros de datos.
- En los supercomputadores, independientemente del subsistema que queramos evaluar, lo que importa es el tiempo de ejecución (el throughput suele ser también relevante). Los resultados se proporcionan habitualmente como FLOPS.
- En los centros de datos el tiempo de respuesta es lo que el usuario percibe como “rendimiento”. El throughput no tiene mucho sentido en este ámbito. Los resultados se pueden proporcionar como relación SLO/SLA (service level objectives/agreements). Por ejemplo: 99% de respuestas en menos de 100ms.

# Evaluación de supercomputadores

- ❑ De largo, el benchmark más popular para supercomputadores es el Linpack. Proporciona el rendimiento en FLOPS bajo la ejecución de un kernel basado en la resolución de sistemas de ecuaciones lineales. Es el banco de pruebas para el ranking Top500.
- ❑ NAS NPB: es una suite de kernels desarrollada por la NASA. Los kernels intentan reflejar el núcleo de cálculo de aplicaciones típicas en mecánica de fluidos junto con otras aplicaciones habituales relacionadas con su actividad. Todos los benchmarks llevan asociados cálculos pesados; difieren en el problema que resuelven y en el volumen de tráfico de comunicación que generan:
  - **EP: *Embarrassingly Parallel***. Conlleva cálculo pero poca comunicación entre procesos.
  - **MG: *Multigrid***. A diferencia del anterior provoca comunicación entre procesadores próximos y lejanos.
  - **CG: *Conjugate Gradient***. Provoca un bajo volumen de comunicación muy esparcida entre nodos próximos y lejanos.
  - **FT: *Fourier Transform***. Patrón de comunicación abundante y uniformemente distribuida.
  - **IS: *Integer Sort***. La comunicación es pesada y uniforme, pero no tan pesada ni uniforme como en el caso anterior.
  - **LU, SP & BT**: abordan el mismo problema matemático usando algoritmos diferentes. BT es “menos paralela” que el resto.

# Linpack

- La primera versión (Linpack 100) data de 1977.
  - Empleaba matrices de rango  $n=100$ .
  - No permitía modificaciones de código.
  - Solo permitía optimizaciones a nivel de compilador.
  - Utilizaba la librería BLAS (Basic Linear Algebra Subprograms) de nivel 1.
- La siguiente versión (Linpack 1000) data de 1986.
  - Empleaba matrices de rango  $n=1000$ .
  - Permitía optimizaciones manuales del código.
  - Utilizaba la librería BLAS de nivel 3.
- La versión que da lugar al ranking top500 es denominada HPL (high performance Linpack) y data de 1991.
  - Las matrices pueden ser de cualquier tamaño.
  - Permite optimizaciones manuales.
  - Se trata de una implementación paralela de paso de mensajes.
  - Para las operaciones matemáticas emplea BLAS o VISPL.
  - Mediante el uso de versiones derivadas de estas librerías, HPL puede extenderse a sistemas heterogéneos. P. ej. cuBLAS para sistemas CPU + GPU.
  - Para el paso de mensajes emplea MPI.

# Evaluación de centros de datos

- ❑ Evaluar centros de datos no es sencillo. Hay dos formas posibles:
  - Desde el punto de vista del usuario, el tiempo de respuesta a las consultas, búsquedas, etc, es lo que importa.
  - Para proporcionar tiempos de respuesta reducidos los administradores tienen que probar muchos subsistemas.
- ❑ La evaluación de centros de datos es un reto reciente, a pesar de lo cual existe ya un cierto consenso en lo que un buen benchmark debe cumplir.
  - Incluir cargas de trabajo diversas.
  - Las cargas de trabajo y las pilas de software deben ser representativas.
  - Abordar tecnologías actuales.
  - Debe ser “usable”, es decir, fácil de implantar, configurar y ejecutar.

# Evaluación de centros de datos

Los sitios de Internet más relevantes en la actualidad son:

(<http://www.alex.com/topsites/global;0>)

- Motores de búsqueda: 40%. Manejan principalmente texto en consultas.
- Redes sociales: 25%. Manejan grafos de datos.
- Comercio electrónico: 15%. Datos tabulados.

Algoritmos habituales empleados por estos sitios:

- Sort.
- Scan.
- Classification.
- Graph mining.
- Segmentation.

# Ejemplo

- Un cluster de 6 nodos y 4 núcleos por nodo completa una multiplicación de matrices de tamaño  $5000 \times 5000$  en 91,07 segundos.
  - Determinar el rendimiento en FLOPS de este cluster.
  - Encontrar su posición en el histórico del Top500.

# Referencias

- ❑ Advanced Computer Architecture: Introduction to quantitative analysis. Prof. Sherief Reda. School of Engineering. Brown University.
- ❑ BigDataBench: a Big Data Benchmark Suite from Internet Services. Lei Wang et al. The 20th IEEE International Symposium On High Performance Computer Architecture (HPCA-2014), February 15-19, 2014, Orlando, Florida, USA.
- ❑ DCBench: a Benchmark Suite for Data Center Workloads. Zhen Jia. HVC tutorial in conjunction with The 19th IEEE International Symposium on High Performance Computer Architecture (HPCA 2013).
- ❑ Notes on Calculating Computer Performance. Bruce Jacob and Trevor Mudge. Advanced Computer Architecture Lab. EECS Department, University of Michigan.
- ❑ The LINPACK Benchmark: past, present and future. Jack J. Dongarra, Piotr Luszczek and Antoine Petit. CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE. *Concurrency Computat.: Pract. Exper.* 2003; **15**:803–820 (DOI: 10.1002/cpe.728).
- ❑ The NAS parallel benchmarks. D.H. Bailey et al. The International Journal of Supercomputer Applications (1991).