Computación de Altas Prestaciones, una herramienta en ayuda de la ciencia

Domingo Giménez

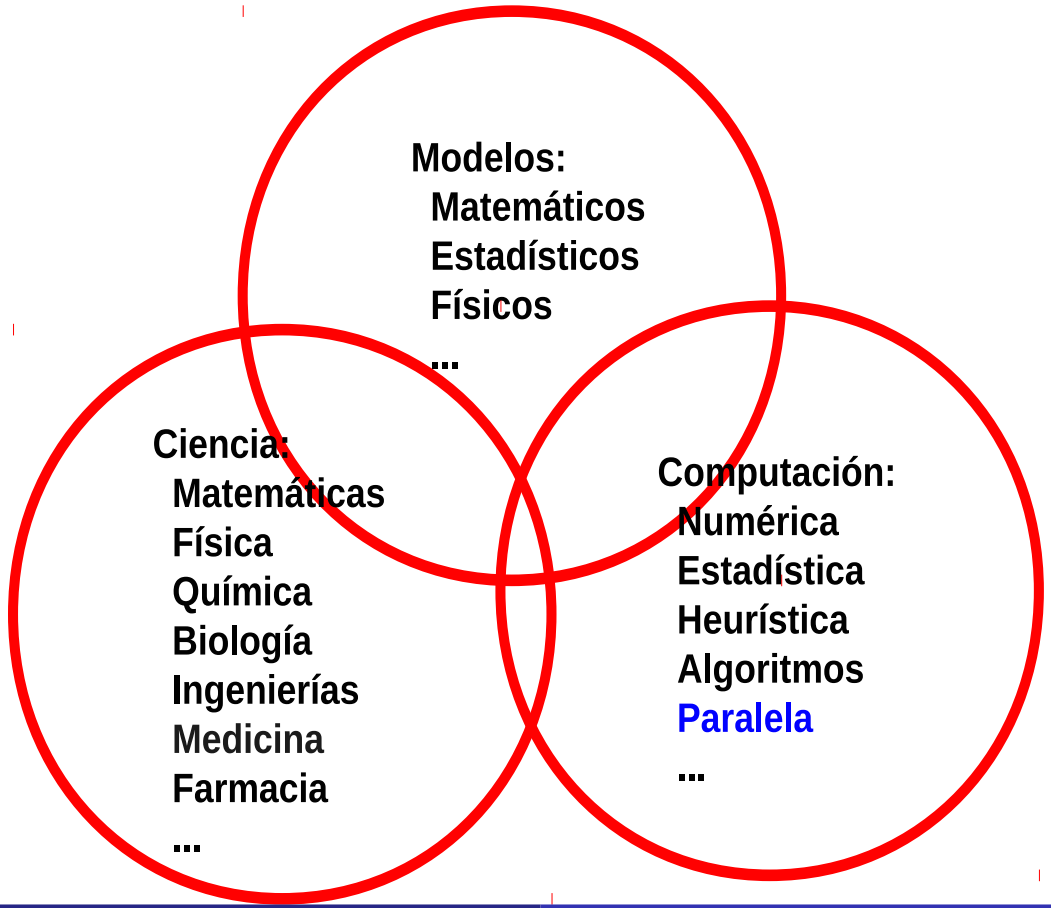
http://dis.um.es/~domingo

Departamento de Informática y Sistemas
Grupo de Computación Científica y Programación Paralela
http://luna.inf.um.es/grupo_investigacion

San Alberto, 14 noviembre 2014

Computación Científica

Integración de modelos con técnicas computacionales para solución de problemas complejos,
con grandes volúmenes de datos (*big data*) y/o necesidades de computación (*High Performance Computing*)



¿Título?

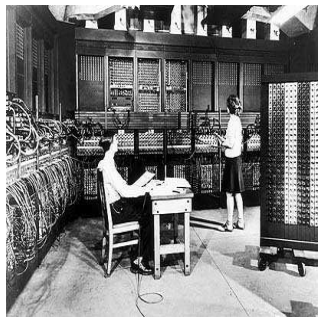
- Supercomputación
- Computación de Altas Prestaciones
- Computación Paralela

¿Título?

- Supercomputación

¿Título?

- Supercomputación

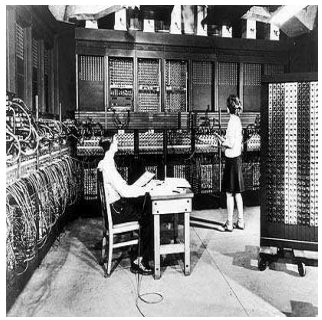


- Computación Paralela

¿Título?

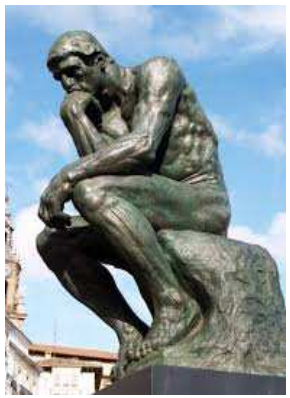
- Supercomputación

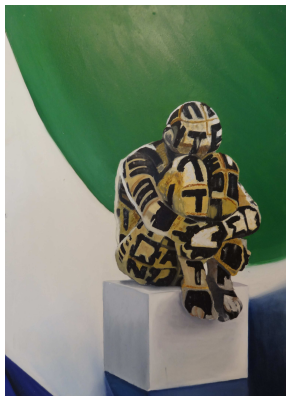
- Computación Paralela



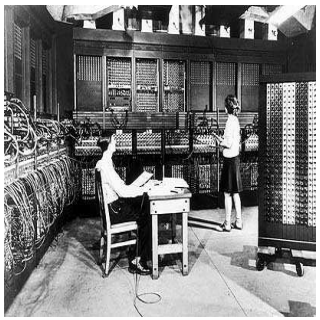
¿Título?

- Supercomputación





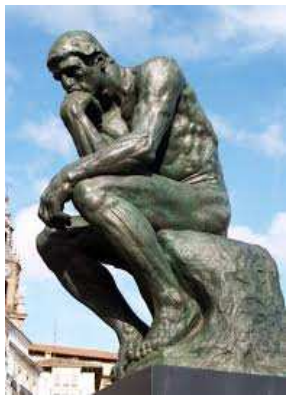
- Computación de Altas Prestaciones

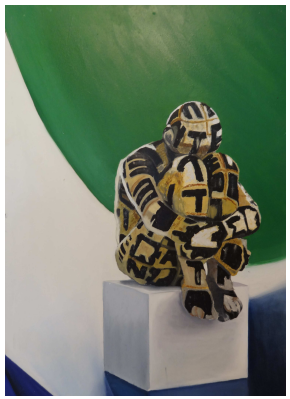


- Computación Paralela

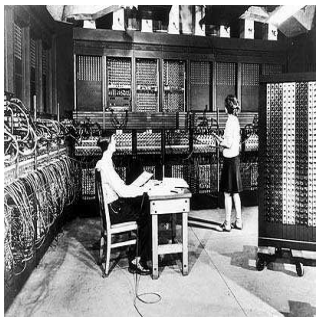
¿Título?

- Supercomputación





- Computación de Altas Prestaciones



- Computación Paralela

¿Título?

- Supercomputación
  la que se realiza en los computadores más potentes para resolver los problemas científicos con mayores necesidades computacionales

- Computación Paralela
  Utiliza sistemas computacionales paralelos. En la actualidad todos
  ¿Cuántos núcleos tiene nuesto móvil, tablet, portátil, sobremesa...?

¿Título?

- Supercomputación
  la que se realiza en los computadores más potentes para resolver los problemas científicos con mayores necesidades computacionales
- Computación de Altas Prestaciones
  se realiza intentando obtener las máximas prestaciones del sistema computacional con el que se trabaja
- Computación Paralela
  Utiliza sistemas computacionales paralelos. En la actualidad todos ¿Cuántos núcleos tiene nuesto móvil, tablet, portátil, sobremesa...?

¿Título?

- Supercomputación
  la que se realiza en los computadores más potentes para resolver los problemas científicos con mayores necesidades computacionales
- Computación de Altas Prestaciones
  se realiza intentando obtener las máximas prestaciones del sistema computacional con el que se trabaja
- Computación Paralela
  Utiliza sistemas computacionales paralelos. En la actualidad todos ¿Cuántos núcleos tiene nuesto móvil, tablet, portátil, sobremesa...?

Un ejemplo histórico

4.

Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen *).

(Von Herrn Professor Dr. *C. G. J. Jacobi.*)

1.

In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen geführt, in welchem die Coëfficienten der verschiedenen Unbekannten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coëfficienten noch dieselbe Gröfse $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen mufs. Für jeden Werth von x, welcher diese Bedingungsgleichung erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Gröfsen x und der ihnen entsprechenden Systeme der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält. Diese Methode überhebt der beschwerlichen Bildung und Auflösung der Gleichung, deren Wurzeln die Werthe von x sind, indem man das gegebne System Gleichungen so transformirt, dafs man für die Gröfsen x starke Annäherungen erhält, worauf für jedes x ein schnell convergirendes Näherungsverfahren zugleich dessen *genauen* Werth und die entsprechenden Werthe der Unbekannten und zwar diese letztern viel leichter als durch die gewöhnlichen Eliminationen ergiebt. Zur Erläuterung dieser Methode habe ich die numerische Auflösung derjenigen Gleichungen gewählt, von welchen die Säcularstörungen der Excentricitäten und der Längen der Perihelien der Pla-

*) Die sorgfältige Ausführung der in diesem Aufsatze vorkommenden numerischen Rechnungen verdanke ich der Gefälligkeit eines meiner Schüler, des Herrn *Ludwig Seidel* in München.

Un ejemplo histórico

4.

Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen *).

(Von Herrn Professor Dr. *C. G. J. Jacobi.*)

1.

In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen geführt, in welchem die Coëfficienten der verschiedenen Unbekannten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coëfficienten noch dieselbe Gröfse $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen mufs. Für jeden Werth von x, welcher diese Bedingungsgleichung erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Gröfsen x und der ihnen entsprechenden Systeme der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält. Diese Methode überhebt der beschwerlichen Bildung und Auflösung der Gleichung, deren Wurzeln die Werthe von x sind, indem man das gegebne System Gleichungen so transformirt, dafs man für die Gröfsen x starke Annäherungen erhält, worauf für jedes x ein schnell convergirendes Näherungsverfahren zugleich dessen *genauen* Werth und die entsprechenden Werthe der Unbekannten und zwar diese letztern viel leichter als durch die gewöhnlichen Eliminationen ergiebt. Zur Erläuterung dieser Methode habe ich die numerische Auflösung derjenigen Gleichungen gewählt, von welchen die Säcularstörungen der Excentricitäten und der Längen der Perihelien der Pla-

*) Die sorgfältige Ausführung der in diesem Aufsatze vorkommenden numerischen Rechnungen verdanke ich der Gefälligkeit eines meiner Schüler, des Herrn *Ludwig Seidel* in München.

- ¿Supercomputación?
- ¿Computación Paralela?
- ¿Computación de Altas Prestaciones?

Un ejemplo histórico

4.

Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen *).

(Von Herrn Professor Dr. *C. G. J. Jacobi.*)

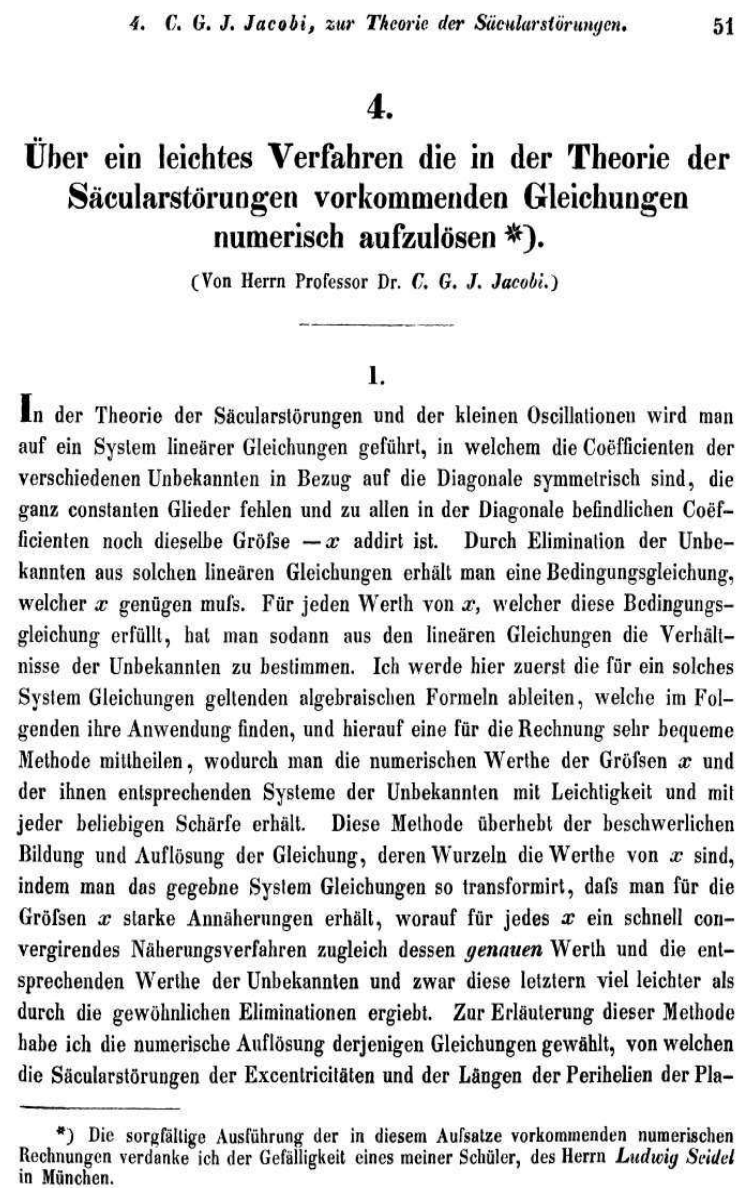
1.

In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen geführt, in welchem die Coëfficienten der verschiedenen Unbekannten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coëfficienten noch dieselbe Gröfse $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen mufs. Für jeden Werth von x, welcher diese Bedingungsgleichung erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Gröfsen x und der ihnen entsprechenden Systeme der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält. Diese Methode überhebt der beschwerlichen Bildung und Auflösung der Gleichung, deren Wurzeln die Werthe von x sind, indem man das gegebne System Gleichungen so transformirt, dafs man für die Gröfsen x starke Annäherungen erhält, worauf für jedes x ein schnell convergirendes Näherungsverfahren zugleich dessen *genauen* Werth und die entsprechenden Werthe der Unbekannten und zwar diese letztern viel leichter als durch die gewöhnlichen Eliminationen ergiebt. Zur Erläuterung dieser Methode habe ich die numerische Auflösung derjenigen Gleichungen gewählt, von welchen die Säcularstörungen der Excentricitäten und der Längen der Perihelien der Pla-

*) Die sorgfältige Ausführung der in diesem Aufsatze vorkommenden numerischen Rechnungen verdanke ich der Gefälligkeit eines meiner Schüler, des Herrn *Ludwig Seidel* in München.

- ¿Supercomputación?
- ¿Computación Paralela?
- ¿Computación de Altas Prestaciones?

Un ejemplo histórico



4. C. G. J. Jacobi, zur Theorie der Säcularstörungen. 51

4.

Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen *).

(Von Herrn Professor Dr. *C. G. J. Jacobi.*)

1.

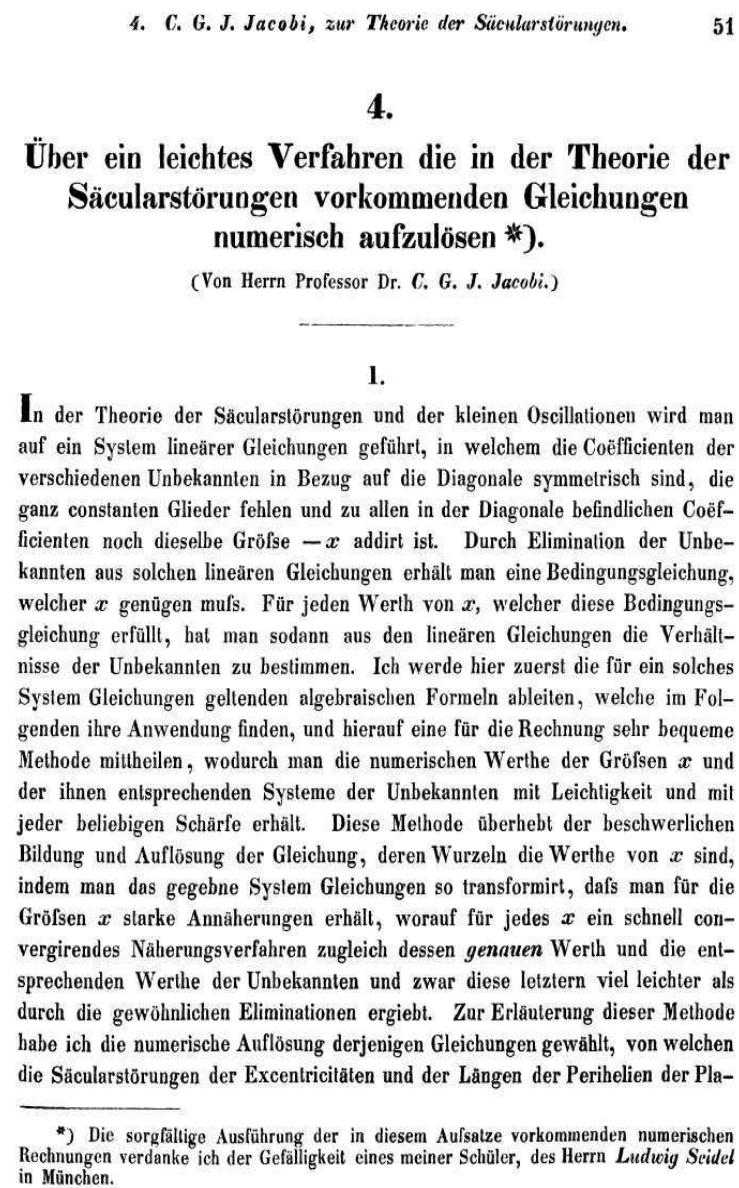
In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen geführt, in welchem die Coëfficienten der verschiedenen Unbekannten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coëfficienten noch dieselbe Gröfse $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen mufs. Für jeden Werth von x, welcher diese Bedingungsgleichung erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Gröfsen x und der ihnen entsprechenden Systeme der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält. Diese Methode überhebt der beschwerlichen Bildung und Auflösung der Gleichung, deren Wurzeln die Werthe von x sind, indem man das gegebne System Gleichungen so transformirt, dafs man für die Gröfsen x starke Annäherungen erhält, worauf für jedes x ein schnell convergirendes Näherungsverfahren zugleich dessen *genauen* Werth und die entsprechenden Werthe der Unbekannten und zwar diese letztern viel leichter als durch die gewöhnlichen Eliminationen ergiebt. Zur Erläuterung dieser Methode habe ich die numerische Auflösung derjenigen Gleichungen gewählt, von welchen die Säcularstörungen der Excentricitäten und der Längen der Perihelien der Pla-

*) Die sorgfältige Ausführung der in diesem Aufsatze vorkommenden numerischen Rechnungen verdanke ich der Gefälligkeit eines meiner Schüler, des Herrn *Ludwig Seidel* in München.

- ¿Supercomputación?
- ¿Computación Paralela?
- ¿Computación de Altas Prestaciones?

- 1846: tamaño siete
- 1995: unos pocos miles
- 2014: más de un millón

Un ejemplo histórico



4. C. G. J. Jacobi, zur Theorie der Säcularstörungen. 51

4.

Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen *).

(Von Herrn Professor Dr. *C. G. J. Jacobi.*)

1.

In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen geführt, in welchem die Coëfficienten der verschiedenen Unbekannten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coëfficienten noch dieselbe Gröfse $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen mufs. Für jeden Werth von x, welcher diese Bedingungsgleichung erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Gröfsen x und der ihnen entsprechenden Systeme der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält. Diese Methode überhebt der beschwerlichen Bildung und Auflösung der Gleichung, deren Wurzeln die Werthe von x sind, indem man das gegebne System Gleichungen so transformirt, dafs man für die Gröfsen x starke Annäherungen erhält, worauf für jedes x ein schnell convergirendes Näherungsverfahren zugleich dessen *genauen* Werth und die entsprechenden Werthe der Unbekannten und zwar diese letztern viel leichter als durch die gewöhnlichen Eliminationen ergiebt. Zur Erläuterung dieser Methode habe ich die numerische Auflösung derjenigen Gleichungen gewählt, von welchen die Säcularstörungen der Excentricitäten und der Längen der Perihelien der Pla-

*) Die sorgfältige Ausführung der in diesem Aufsatze vorkommenden numerischen Rechnungen verdanke ich der Gefälligkeit eines meiner Schüler, des Herrn *Ludwig Seidel* in München.

- ¿Supercomputación?
- ¿Computación Paralela?
- ¿Computación de Altas Prestaciones?

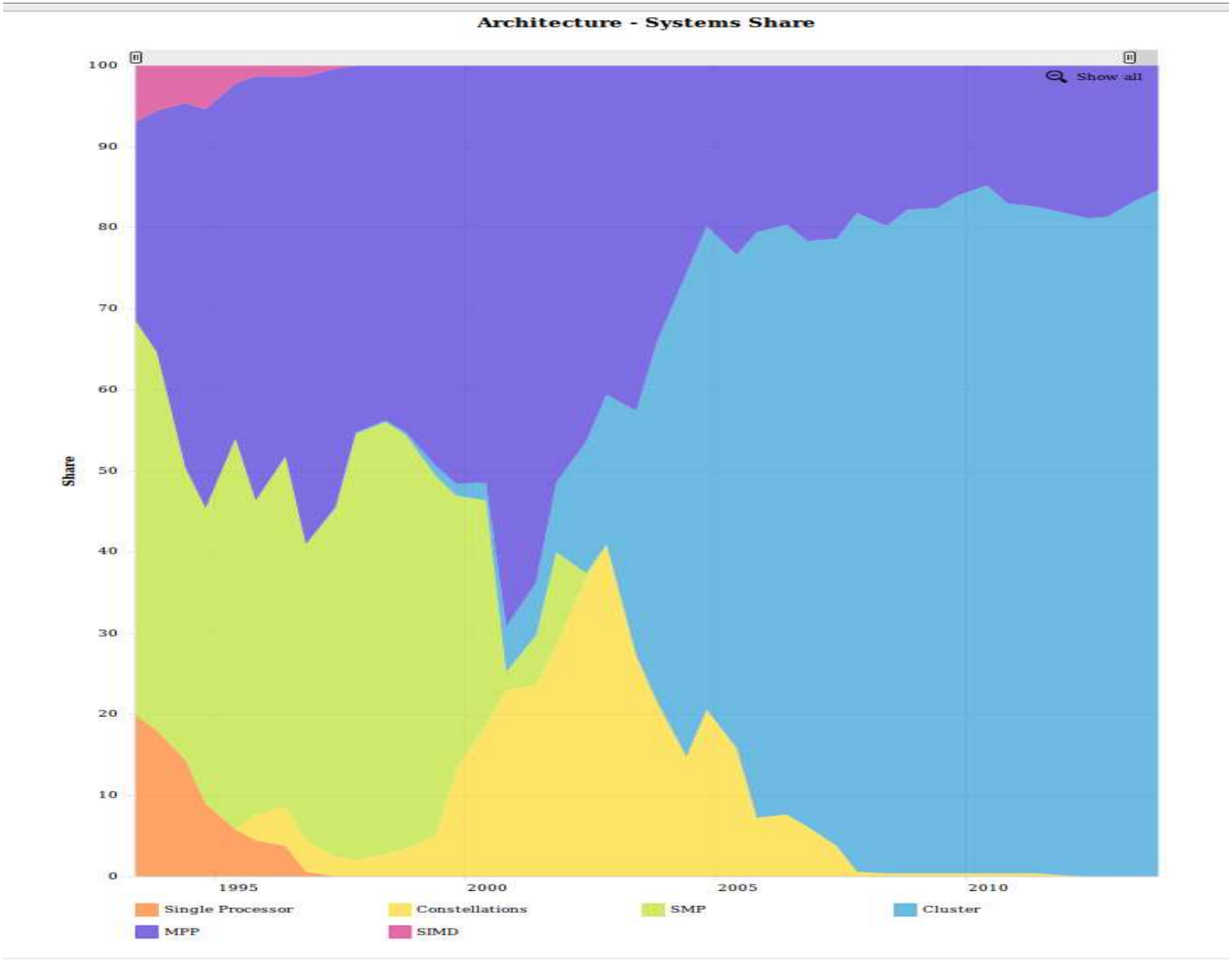
- 1846: tamaño siete
- 1995: unos pocos miles
- 2014: más de un millón

Temas a tratar

1. Sistemas
2. Aplicaciones
3. Programación
4. Conclusión

Temas a tratar

Evolución de arquitecturas en TOP500

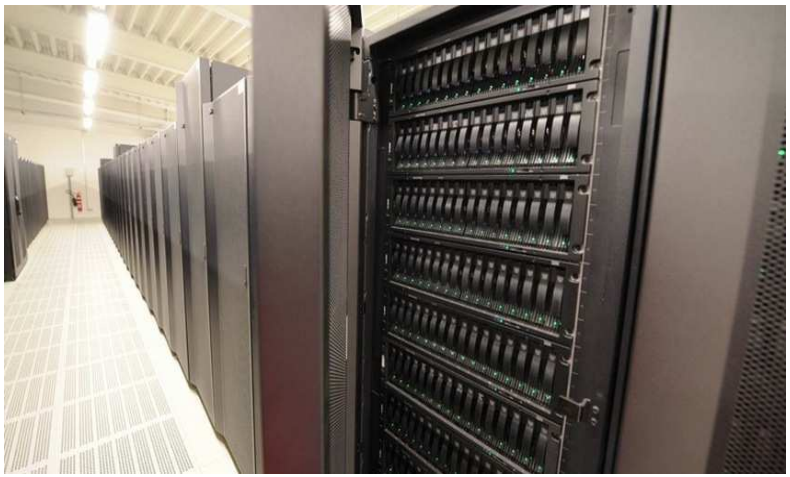


Los más rápidos - junio 1993 a junio 1996

CM-5, 1024 proc. 59.7 GFlops
Los Alamos National Lab
junio 1993

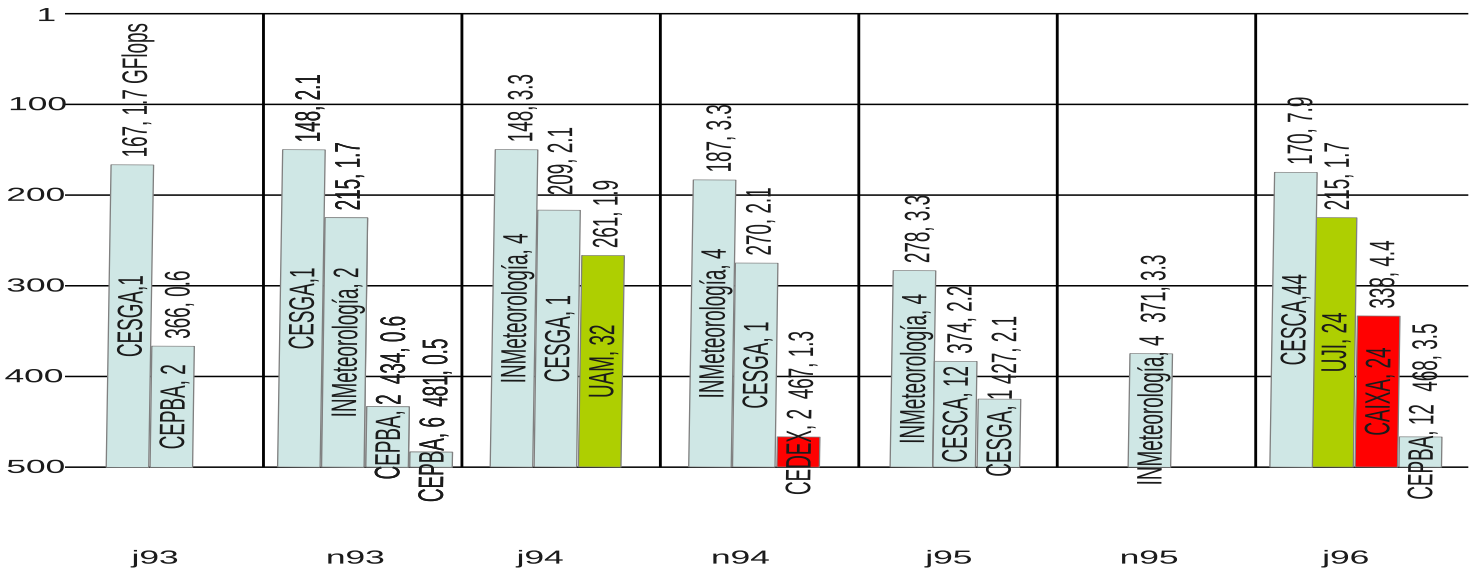
Numerical Wind Tunnel, 167 proc. 170 GFlops
National Aerospace Laboratory of Japan
nov. 1993 y nov. 1994 a dic. 1995

Intel XP/S 140 Paragon, 3680 proc. 143.4 GFlops
Sandia National Labs
junio 1994



Hitachi SR2201, 1024 proc. 232.4 GFlops
University of Tokyo
junio 1996

... y más cerca



Universidad de Murcia:

- Cluster de VAX (Digital) 8 nodos.
- 3 IBM con AIX Parallel Environment (Cartagena).
- Cluster de HP Apollo 700.
- Placas de Transputers, en array y con conexiones programables.

Los más rápidos - noviembre 1996 a junio 2004

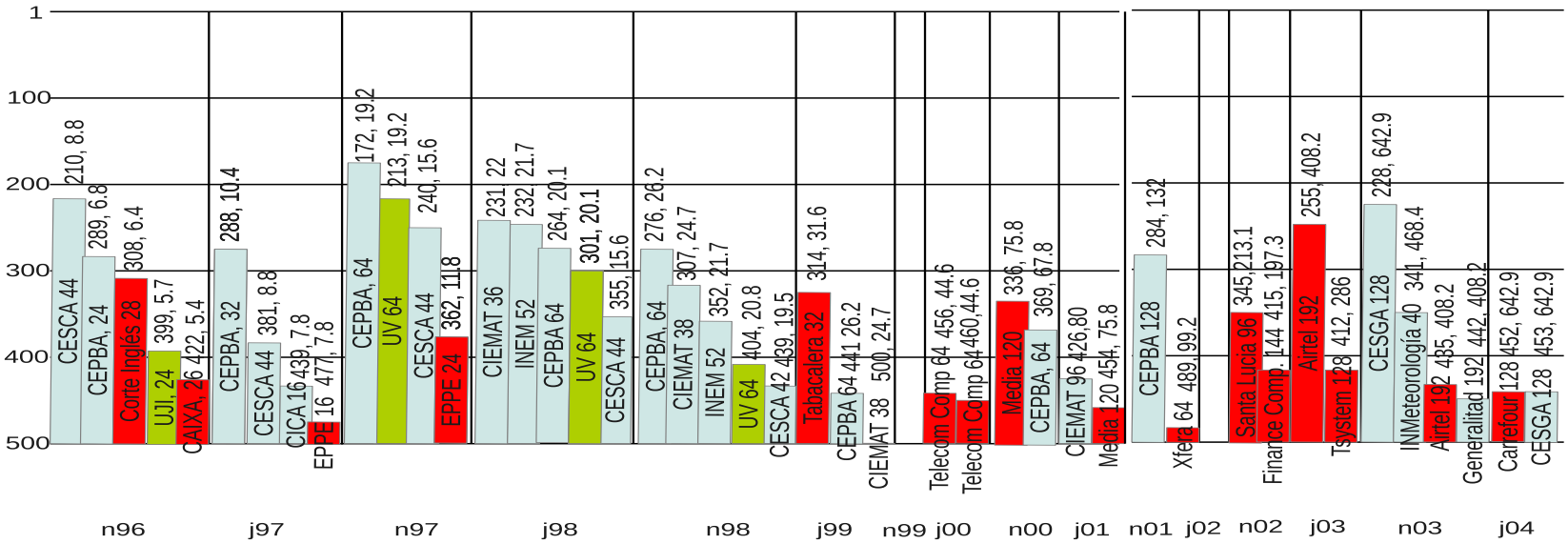
CP-PACS, 2048 proc. 368.2 GFlops
University of Tsukuba
noviembre 1996

ASCI Red, 7264 proc., 1212 GFlops
Sandia National Laboratory
junio 1997 a junio 2000

ASCI White, 512 nodos×16 proc., 7.2 Teraflops
Lawrence Livermore National Laboratory
noviembre 2000 a noviembre 2001

The Earth Simulator, 640 nodos×8 proc. vect., 35.86 TFlops
Earth Simulator Center
junio 2002 a junio 2004

... y más cerca



Universidad de Murcia:

- Sistema de Memoria Compartida SGI con 6 procesadores.
- Cluster de 4 nodos HP AlphaServer quad, 16 núcleos (Cartagena).
- Facultad de Informática y Grupos de Investigación:
  Clusters de 5 SUN Ultra 1 + 1 SUN Ultra 5, de 7 SUN Sparcstation, de 13 PC 486, de 6 Pentiums...
  y combinaciones heterogéneas.

Los más rápidos - noviembre 2004 a noviembre 2010

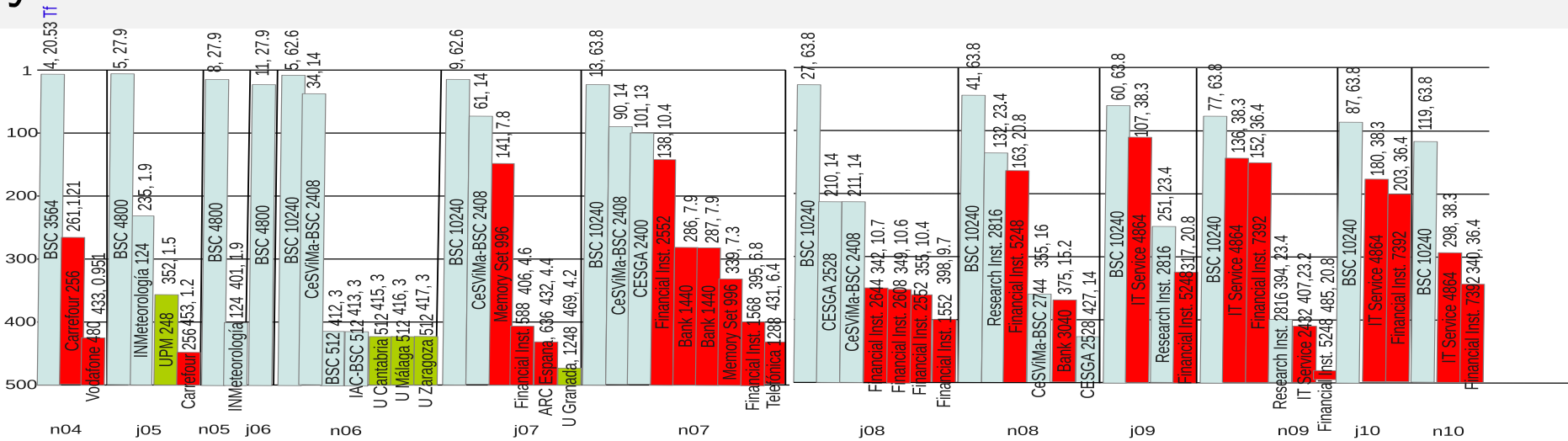
BlueGene/L, 106496 nodos duales, 478.2 TFlops
Lawrence Livermore National Laboratoryy
noviembre 2004 a noviembre 2007

Roadrunner, 116640 núcleos, 1.456 Petaflops
Los Alamos National Laboratory
junio 2008 a junio 2009

Jaguar, 200000 núcleos, 1.759 PFlops
Oak ridge National Laboratory
noviembre 2009 a junio 2010

Tianhe-1A, 14336 Intel Xeon+7168 NVIDIA Tesla GPU, 2.57 PFlops
National Supercomputing Center in Tianjin
noviembre 2010

... y más cerca



Murcia:

- Universidad de Murcia:
  Cluster HP con 8 nodos Itanium 2, 64 núcleos, después Cluster Core 2 Quad con 20 nodos, 160 núcleos.
- Politécnica de Cartagena:
  Cluster de 16 HP AlphaServer; después Cluster de 40 procesadores con 152 núcleos.
- Centro de Supercomputación de Murcia (2008 a 2013), Ben (128 núcleos de memoria compartida) + Arabi (cluster de 102 nodos de 8 núcleos), total 944 núcleos.
- Grupos de investigación UMU:
  Clusters de nodos quad, sobre 64 núcleos.
  Al final del periodo: sistemas de Memoria Compartida con 24 núcleos, clusters de multicore+GPUs.

Los más rápidos - desde junio 2011

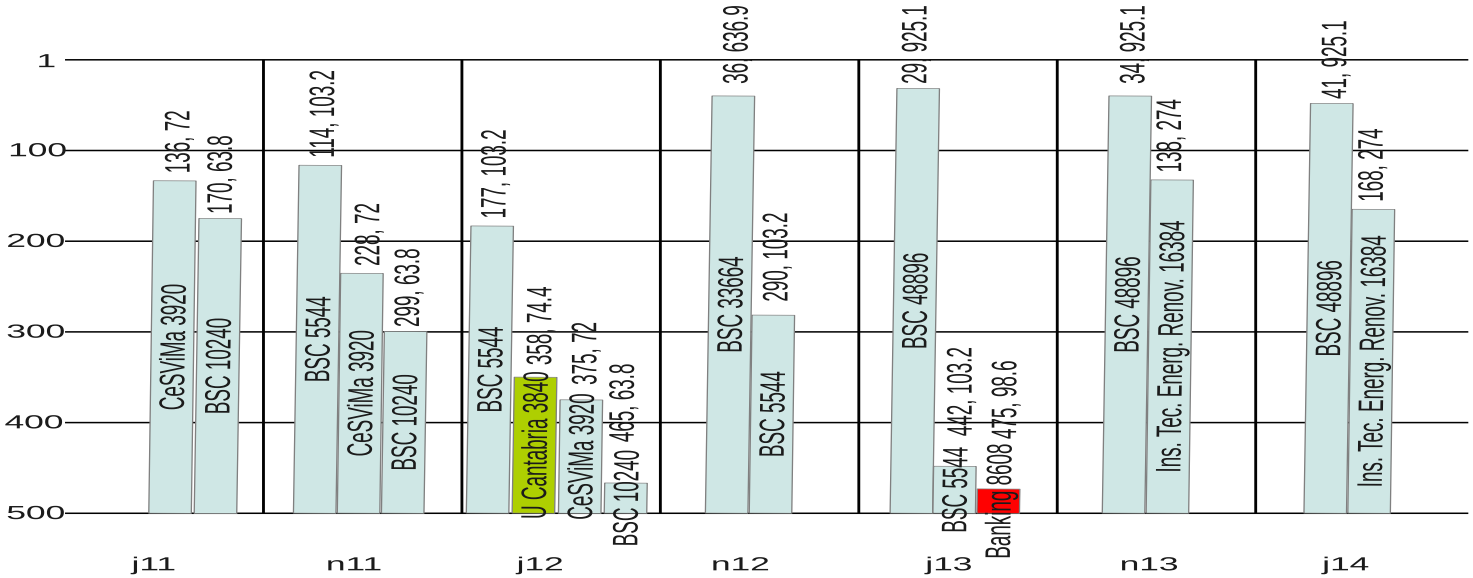
K Computer, 705024 núcleos, 10 PFlops
RIKEN Advanced Institute for Computational Science
junio 2011 a noviembre 2011

Sequoia, 1572864 núcleos 16.32 PFlops
Lawrence Livermore National Laboratory
junio 2012

Titan, 18688×(Tesla K20+16 núcleos), 27 PFlops
Oak ridge National Laboratory
noviembre 2012

MilkyWay-2, 3120000 núcleos (Ivy Bridge+Xeon Phi), 33.86 PFs
National Super Computer Center in Guangzhou
desde junio 2013

... y más cerca



Universidad de Murcia:

- Cluster Intel Xeon, 8 con 16 núcleos + 8 con 32 núcleos = 384 núcleos.
- Grupos de investigación:

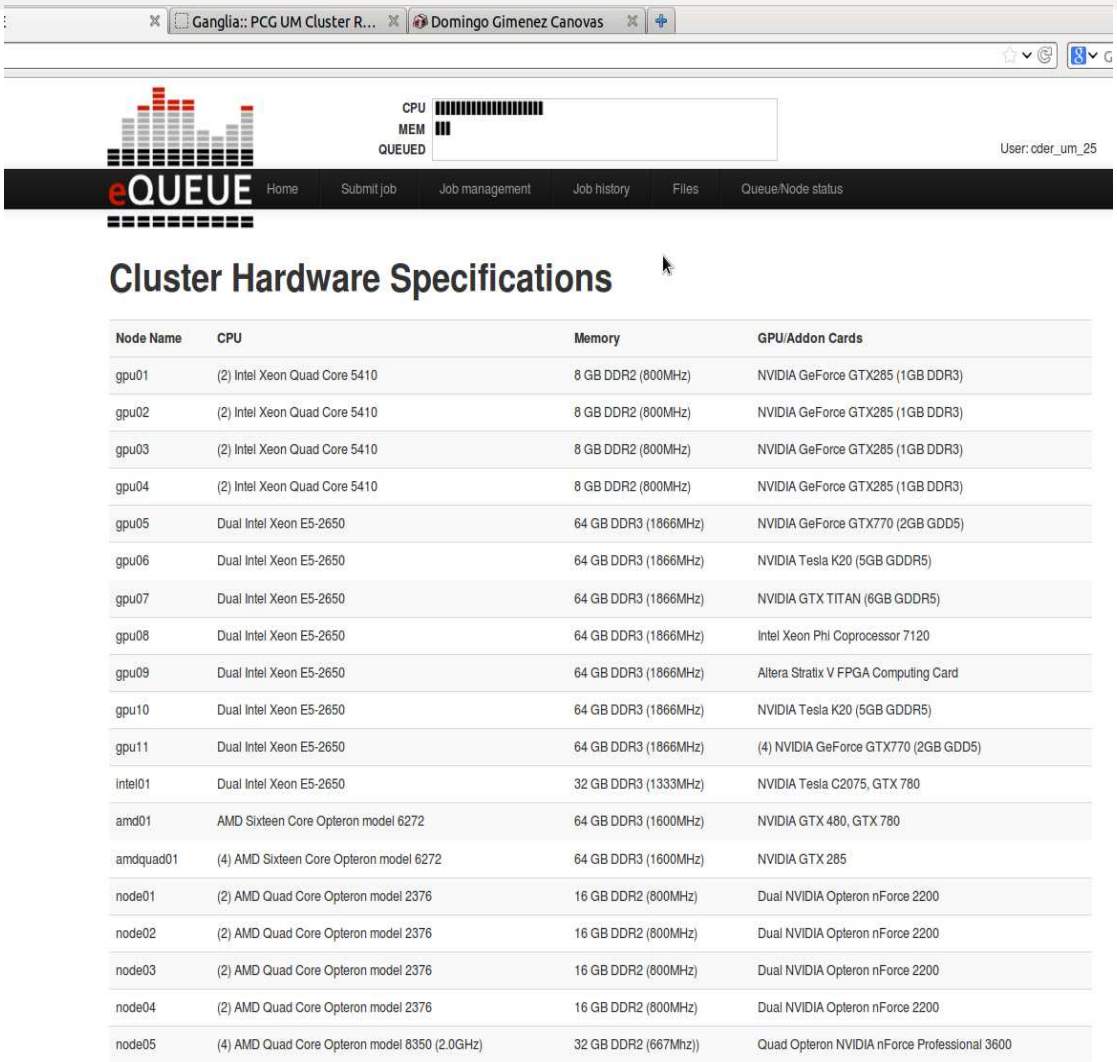
  Clusters, en algunos casos heterogéneos con nodos con una o varias GPUs de tipos distintos.

  Algunos grupos con Intel Xeon Phi.

Aproximadamente entrarían en el TOP500 en 2006, en el puesto del primer español en 2004, y el primero de la lista en 2000.

Un PC actual entraría en 2000 y sería el primero de 1993.

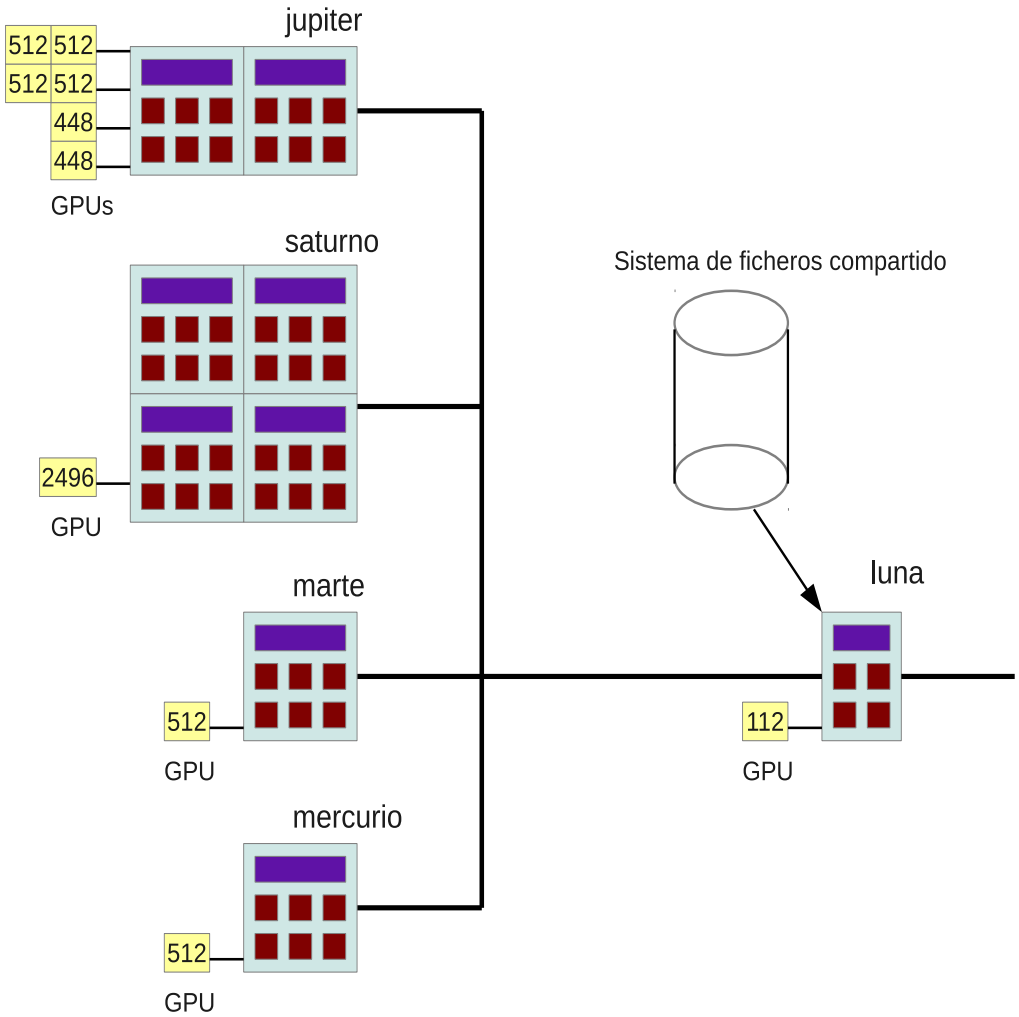
Por ejemplo, NSF-supported Center for Parallel and Distributed Computing Curriculum Development and Educational Resources



`http://www.cs.gsu.edu/~tcpp/`

6 colas,
19 nodos,
250 núcleos,
28 GPU,
1 XPhi,
1 FPGA

Por ejemplo, Laboratorio de Computación Científica y Programación Paralela

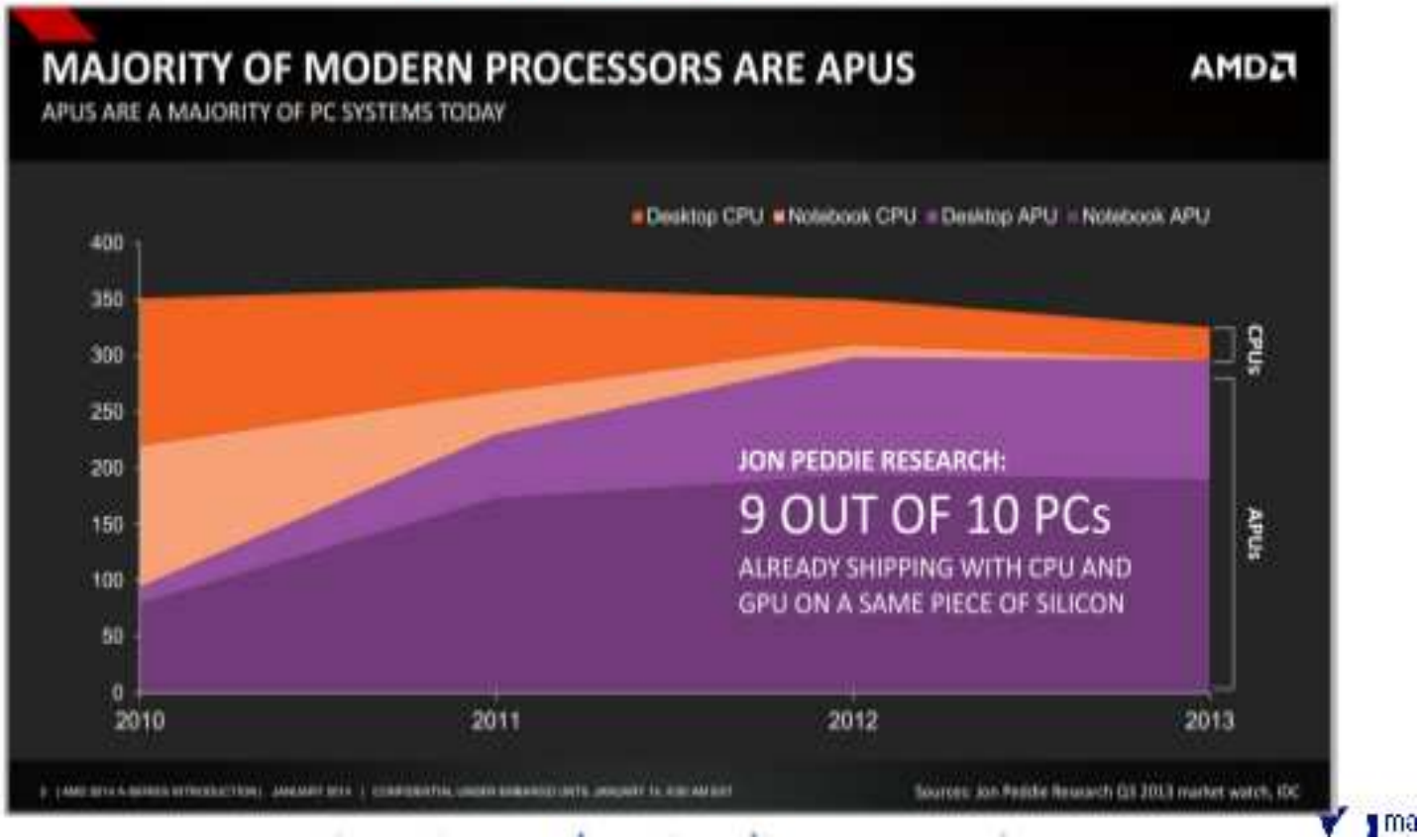


http://luna.inf.um.es/grupo_investigacion

¿Futuro? - heterogeneidad

Cada vez más coprocesadores (GPU, Xeon Phi, FPGA...) integrados y con configuración heterogénea.

– Integrated GPUs on more than 90% of shipped processors

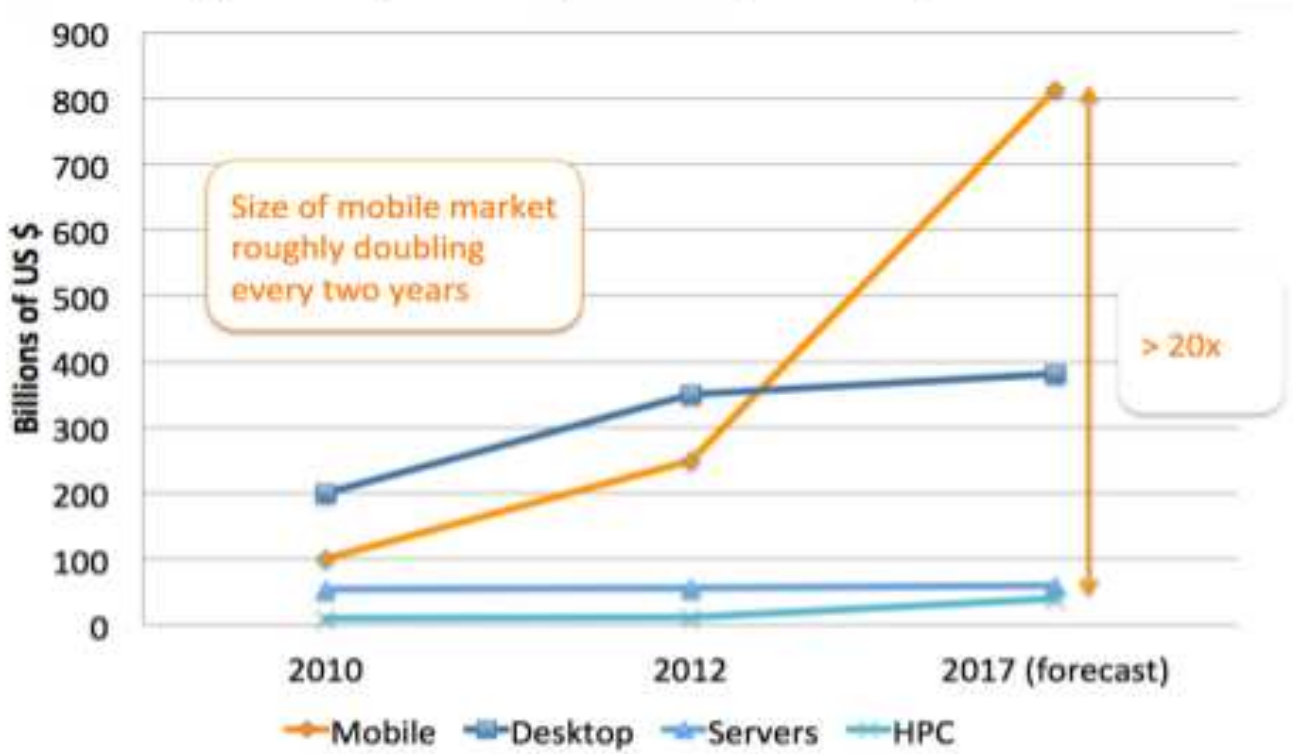


Rafael Asenjo, en Jornadas de Paralelismo, Valladolid, Septiembre 2014.

¿Futuro? - móviles

Cada vez más importancia de sistemas móviles.

- There is (parallel) live beyond supercomputers:

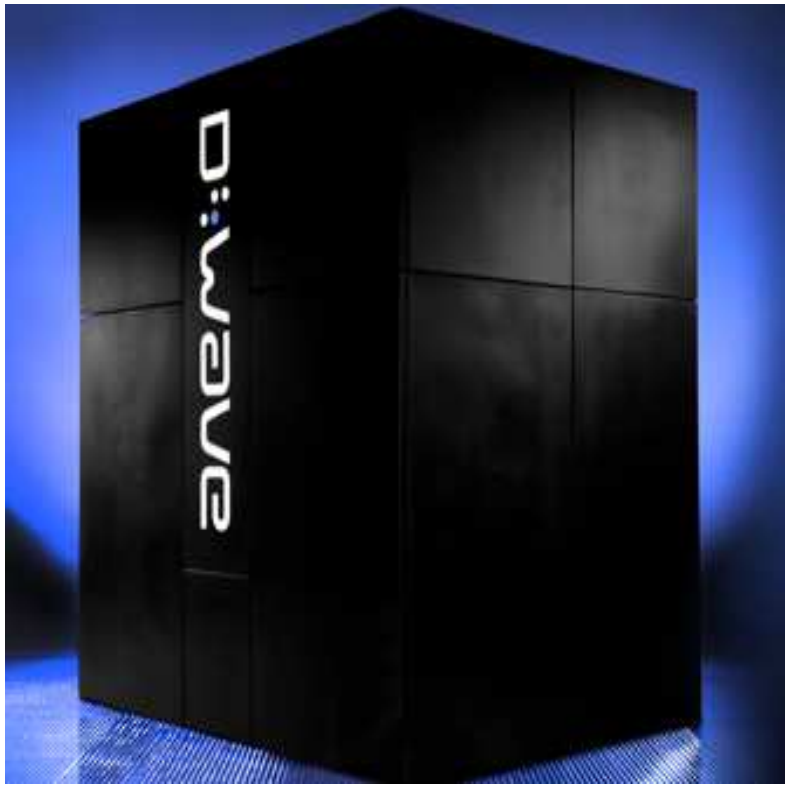


Rafael Asenjo, en Jornadas de Paralelismo, Valladolid, Septiembre 2014.

Proyecto Montblanc del BSC para desarrollar supercomputador con procesadores de tablet.

¿Futuro? - computación no convencional

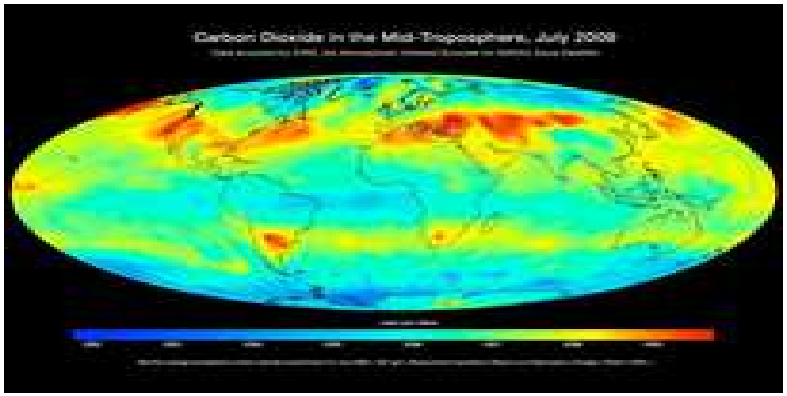
¿Para cuándo Computación Cuántica de propósito general y al alcance de todos?

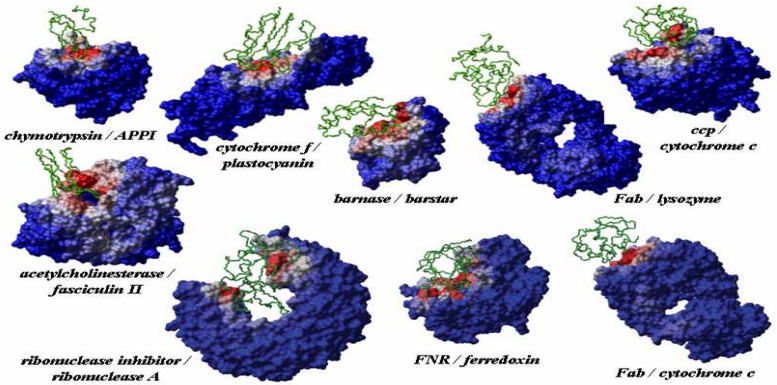


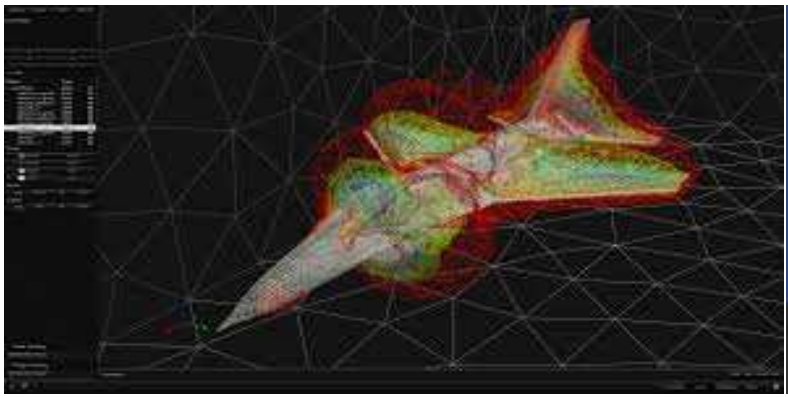
El primer computador cuántico comercial en 2011, vendido a compañía aeroespacial y de defensa.
Unas 4000 veces más rápido que Intel Xeon en 2013.

Temas a tratar

Aplicaciones







... y también

LUNES 14.7.14 **LA VERDAD**

Lea y Recicle

LA VERDAD UTILIZA PAPEL 100% RECICLADO

vocento

LA VERDAD.ES

oferplan.laverdad.es

Antonio Orozco en Los Alcázares, el 9 de agosto, a 16 euros (ahorro de nueve euros)

NUEVE DE NUEVE DE MARC MÁRQUEZ

El piloto de Honda tampoco halla rival en Alemania

MANUEL ALCÁNTARA

EL DETECTIVE FRANCISCO

El chef es un superordenador

'Watson' es un programa de IBM que ha sorprendido por su originalidad en crear nuevas recetas, como burrito tex-mex con chocolate y melocotón

:: TERESA BOUZA

AUSTIN. 'Watson' es un superordenador de IBM que se hizo famoso en 2011 en EE UU tras derrotar a los campeones del popular concurso televisivo 'Jeopardy', una especie de trivial con todo tipo de preguntas. Ni Bud Rutter ni Ken Jennings –poseedores de los récord de permanencia y ganancias en el programa–, pudieron con él. Ahora, a 'Watson' le han reprogramado para que utilice su inteligencia en la cocina.

El cerebro cibernético, instalado en un colorido camión de comida, ha mostrado sus habilidades recientemente en Austin (Texas), durante la celebración del multitudinario festival de tecnología, música y cine South by Southwest (SXSW). 'Watson' tiene de pinche a Michael Laiskonis, director del Instituto de Educación Culinaria de Nueva York. El chef, experto en pastelería, ha trabajado con la supermáquina durante dos años, ayudando a los programadores de IBM para hacerle memorizar decenas de miles de recetas.

«Lo entrenamos con unas 35.000 recetas y aprendió qué ingredientes son comunes en ciertas cocinas, qué ingredientes funcionan bien juntos y qué combinaciones hay que hacer para obtener un cierto plato». La cosa fuenciona así: los cocineros sugieren un ingrediente principal y 'Watson' comprueba su enorme base de datos, estudia la composición química de los ingredientes y la percepción humana de los mismos. Luego pronostica qué combinaciones resultarán más sabrosas y sorprendentes.

El chef Laiskonis y Florien Pinel, un ingeniero de IBM que ha cursado también estudios superiores de cocina, invitan estos días a los asistentes al SXSW a que elijan un plato para que Watson «lo prepare». Por votación, el público pidió burritos. A partir de ahí, 'Watson' se puso manos a la obra y sorprendió a todos con un «burrito austriaco con chocolate». «El público votó por un burrito, un plato tex-mex típico de Austin y quisimos desafiar al programa y a nosotros mismos al combinar eso con parámetros que creímos nos darían resultados sorprendentes, como cocina austriaca y chocolate», explicó Laiskonis. Los ingredientes complementarios que ofreció 'Watson' incluyeron la carne de ternera picada, tortillas de maíz, queso y habas de soja, pero también sabores interesantes como el del melocotón, la canela, la raspadura de naranja y la vainilla.

«Resulta poco probable que yo hubiese llegado a asociar todos esos ingredientes», admitió el chef neoyorquino, quien señaló que una vez que el ordenador hace su sugerencia, depende del cocinero el interpretar la receta. «Creo que los ingredientes del burrito austriaco con chocolate funcionan muy bien juntos y tengo una gran curiosidad por ver cómo seguirá evolucionando 'Watson'», explicó Laiskonis.

El ingeniero informático Florien Pinel subrayó al dar las explicaciones técnicas de su criatura que el objetivo del programa es ayudar a los cocineros a ser «más creativos en sus platos». Lo difícil será plantarle cara a 'Watson' cuando alguna de sus insólitas recetas se haga indigerible.

Las recetas de 'Watson', instalado en un camión de 'comida cognitiva', ha triunfado en la feria tecnológica de Austin. :: EFE

Ya sólo falta que al Padre de la Iglesia le encarguen vigilar a los hijos de Jordi Pujol. La misa es mucha, pero la basura acumulada es todavía mayor y ahora al buen cura, no menos ecuménico que argentino, se le están acumulando los trabajos de desescombro. O le faltan manos o le sobran escobas. Menos mal que ahora Su Santidad son dos, cosa siempre más fácil de entender que la Santísima Trinidad sea uno solo. Por si no fuese bastante luchar contra los pederastas habituales, que son los que llevan hábito, Francisco ha excomulgado a los mafiosos y se ha echado encima a la mafia calabresa, que también tiene sus escalas, desde los monaguillos hasta el alto clero.

Quizá sea el Pontífice el cargo más difícil de desempeñar entre todos, entre otras cosas porque ya no hay puentes y sin embargo abundan las personas que viven bajo ellos. A pesar de ser un cargo envidiable, no ha sido envidiado en los últimos tiempos. Hace ya muchos años, cuando se rumoreaba, en la época del nacional-catolicismo, que la paloma del Espíritu Santo podía posarse sobre la cabeza del cardenal Segura, dijo Foxá, que era gordo, católico y sentimental, que si sucedía eso él se haría socio del Tiro al Pichón. Ahora habría que llamar a Saviano, que es el que más arriesgadamente sabe de eso, para descifrar las conexiones. ¿Por qué los presos de la 'Ndrángheta se niegan a ir a misa? Los pobres seres humanos nos consolamos con lo que tenemos más cerca, pero sobre todo con quienes nos ofrecen consuelo.

El combate de Bergoglio con los 'padrinos' será largo, pero hay que tener en cuenta que desde ayer Su Santidad, que es argentino, ha estado pendiente, al igual que su predecesor, que es alemán, de Campeonato Mundial que se celebra en Brasil. Desde hoy, todo pertenece al pasado. Ya lo sabemos todo, porque no es cierto que el futuro esté en las rodillas de los dioses, sino en el menisco de los futbolistas. El tiempo sigue siendo sucesivo, pero hay noches que parece plano. Desde hoy sólo tendrán que esperar ustedes cuatro años.

LA VERDAD MEJORA LA OFERTA DE REVISTAS FEMENINAS DEL FIN DE SEMANA

GRATIS los Sábados

mujer hoy

¡La revista femenina más leída de España!

NUEVA MUJER HOY

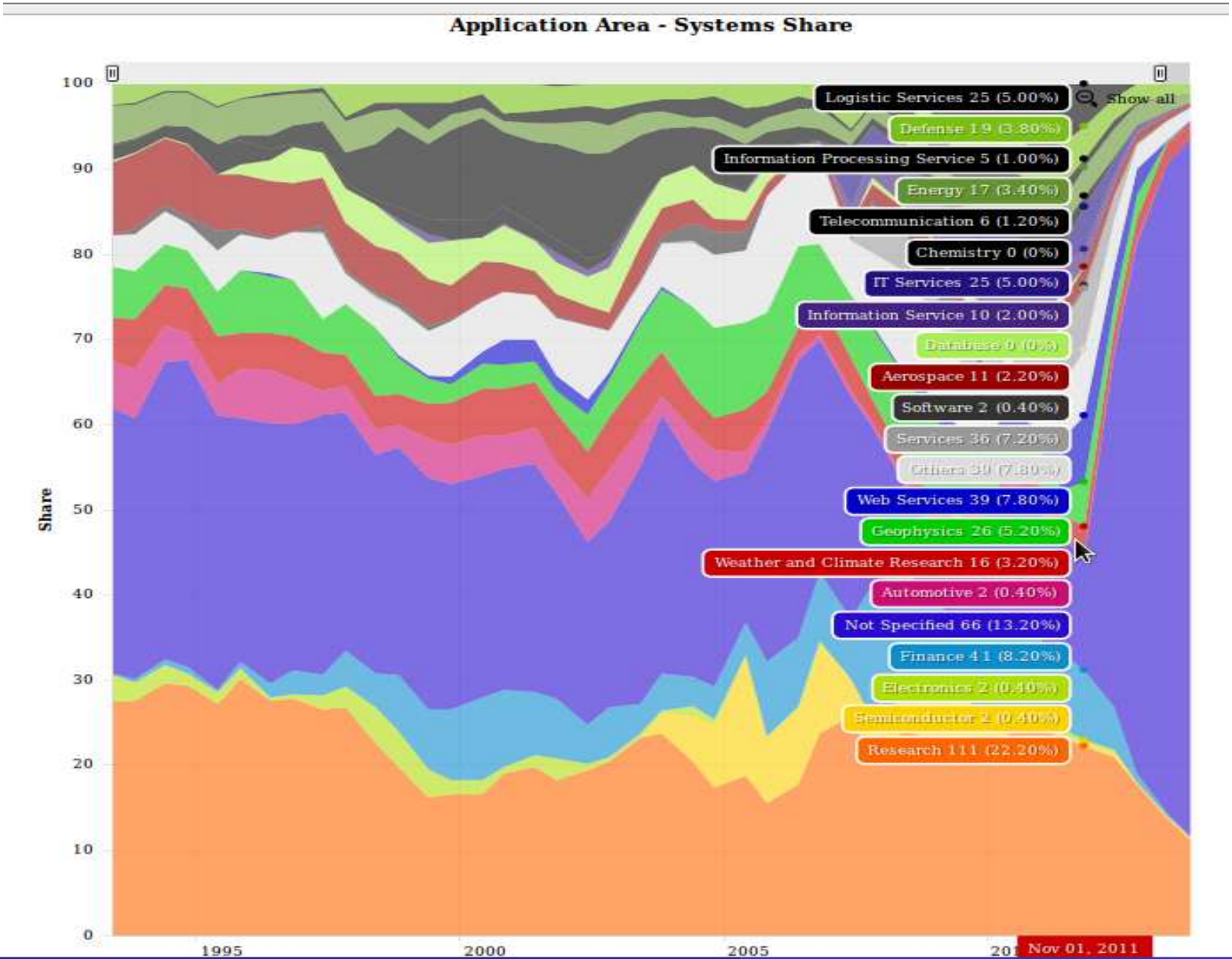
¡MÁS PÁGINAS! ¡MÁS FIRMAS! ¡NUEVAS SECCIONES!

Por sólo 0,50 € los Domingos

Hoy Corazón

LA VERDAD

Evolución de aplicaciones en TOP500



Tipos de problemas

- De gran desafío
  Alto coste computacional $\theta\left(a^n\right)$
  problemas NP, optimización combinatoria...
- De gran dimensión
  Coste moderado $\theta\left(n^3\right)$, $\theta\left(n^4\right)$...
  pero gran dimensión, por ejemplo $n = 1.000.000$ (8 Teras)
- De tiempo real
  Coste bajo pero necesidad de respuesta inmediata
  juegos, control, atención médica...

	Clima	Bio	Diseño	Simul	Juego	Imagen	recetas
G. Desafío		X	X				X
G. Dimensión	X	X	X	X		X	
Tiempo Real	X	X		X	X	X	X

Problemas de Gran Desafío

Alto coste computacional $\theta(a^n)$

	Clima	Bio	Diseño	Simul	Juego	Imagen	recetas
G. Desafío		X	X				X
G. Dimensión	X	X	X	X		X	
Tiempo Real	X	X		X	X	X	X

- Problemas NP, optimización combinatoria:
  Planificación, Logística,
  Asignación de recursos,
  Estudio de eficiencia de organizaciones
- Biología, Medicina:
  Búsquedas en DNA,
  Diseño de fármacos
- Diseño
  Cuando se trata de determinar componentes (ejemplos, filtros de señal, diseño de puentes con restricciones...)
- Recetas?
  si varios componentes e intentar combinarlos para satisfacer distintas restricciones y maximizando algunos aspectos

Problemas de Gran Dimensión

Coste moderado ($\theta(n^p)$) pero gran tamaño

	Clima	Bio	Diseño	Simul	Juego	Imagen	recetas
G. Desafío		X	X				X
G. Dimensión	X	X	X	X		X	
Tiempo Real	X	X		X	X	X	X

- En algunos casos los problemas de memoria son mayores que los de computación
- Clima:

  Simulaciones con mallado muy fino

  Tiempo cuadrático (superficie) o cúbico (espacio) que aumenta de orden con la evolución temporal
- Bioinformática:

  Problemas de búsqueda en cadenas, con bases de datos muy grandes
- Diseño y Simulación:

  Puede necesitarse mallado muy fino

  El coste aumenta por el número de simulaciones a realizar
- Imagen:

  Generación de imágenes para películas,

  muchas imágenes y gran precisión

Problemas de Tiempo Real

Requieren respuesta en un plazo de tiempo "corto"

	Clima	Bio	Diseño	Simul	Juego	Imagen	recetas
G. Desafío		X	X				X
G. Dimensión	X	X	X	X		X	
Tiempo Real	X	X		X	X	X	X

- Clima:

  Si la predicción es para el día siguiente

- Medicina:

  Asistencia inmediata

- Simulación:

  Si se realiza para controlar un sistema, expansión de un incendio...

- Juego e Imagen:

  Generación de 24 imágenes por segundo

- Recetas:

  Si se proporcionan los requerimientos o datos personales en el mismo momento

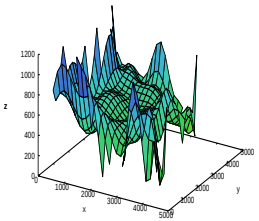
Grupos en la UMU

- Dedicados al paralelismo:
  - Arquitectura de Computadores y Sistemas Paralelos
  - Arquitectura y Computación Paralela
  - Computación Científica y Programación Paralela
- Uso de paralelismo:
  - Computación Móvil y Visión Artificial
  - Informática Industrial
  - Sistemas Inteligentes y Telemática
  - Grupo de Modelización Atmosférica Regional
  - Láseres, Espectroscopía Molecular y Química Cuántica
  - Materia Condensada
  - Polímeros
  - Anillos
  - Investigación Operativa
  - Optimización de Recursos y Teoría de Juegos
  - Sistemas Dinámicos y Aplicaciones

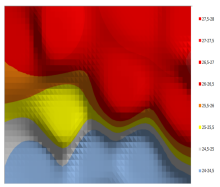
Un ejemplo

Modelos Paralelos para la Resolución de Problemas de Ingeniería Agrícola, tesis de Murilo do Carmo Boratto, en la UPV

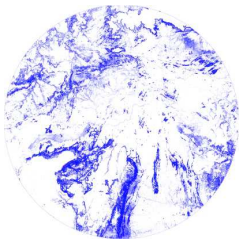
Tres problemas en la zona del Rio São Francisco, en Brasil

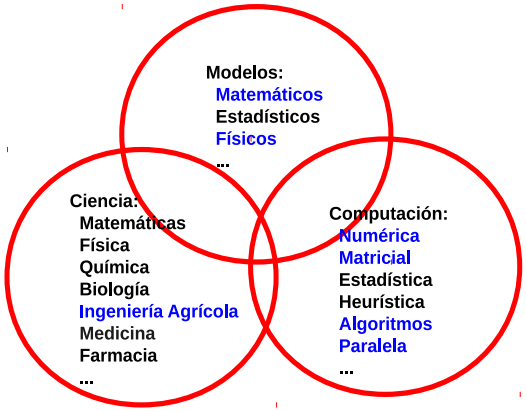


- Representación del relieve
- Interpolación de variables meteorológicas



- Modelado de corrientes de ríos





- Modelos
  - Mínimos cuadrados
  - Representación matricial
  - Ecuaciones diferenciales
- Métodos
  - Algoritmos matriciales
  - Computación eficiente en sistemas heterogéneos multicore+multiGPU
  - Técnicas de autooptimización de software

Un ejemplo - Modelado de corriente de ríos

- Modelo hídrico

$$b\frac{\partial y}{\partial t} + A\frac{\partial v}{\partial x} + v \cdot b\frac{\partial y}{\partial x} = 0$$
$$\frac{1}{g}\frac{\partial v}{\partial t} + \frac{v}{g}\frac{\partial v}{\partial x} + \frac{\partial y}{\partial x} = (S_o - S_f)$$

- A partir de él el problema matricial

$$\begin{pmatrix} A_1 & B_1 & C_1 & D_1 & & & & & & & & \\ A_2 & B_2 & C_2 & D_2 & & & & & & & & \\ & & A_3 & B_3 & C_3 & D_3 & & & & & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & & & & & \\ & & & & & & A_{n-1} & B_{n-1} & C_{n-1} & D_{n-1} \\ & & & & & & A_{n-2} & B_{n-2} & C_{n-2} & D_{n-2} \end{pmatrix} \begin{pmatrix} Q_1 \\ y_1 \\ y_2 \\ \cdots \\ Q_n \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ E_1 \\ E_2 \\ \cdots \\ E_n \\ 0 \end{pmatrix}$$

- Y resolver con multicore+multiGPU (determinar el volumen de trabajo para cada componente computacional) la formación de las matrices a partir de los datos experimentales y la resolución del sistema por bloques.

Temas a tratar

Entornos de programación paralela

Multitud de tipos y entornos de programación:

- C/C++ con fork-join
- Clases de paralelismo en Java
- Pthreads
- OpenMP, estándar para Memoria Compartida
- MPI, estándar de Paso de Mensajes
- CUDA
- OpenCL, estándar para GPU e híbridos
- ...
- Versiones paralelas de entornos científicos (Matlab, R...)
- Y programación paralela híbrida

Librerías

Pero seguramente habrá disponible para el problema con el que estamos trabajando librerías paralelas optimizadas:

- Álgebra lineal densa: BLAS, LAPACK, MAGMA, PLASMA.
- Álgebra lineal dispersa: ARPACK, SPARSE, SPARSE-BLAS, ITPACK, SVDPACK, SuperLU, Trilinos, PETSc.
- Optimización: HeO, ParadisEO, MALLBA, GitHub.
- Transformada de Fourier: FFTPACK, P3DFFT, FFTW.
- Científicas: MOOSE, COOLFluid, OpenFVM, PyClaw, PetIGA...

Información de librerías en
`http://www.netlib.org/liblist.html`
`http://www.netlib.org/utk/people/JackDongarra/la-sw.html`

Librerías - Ventajas

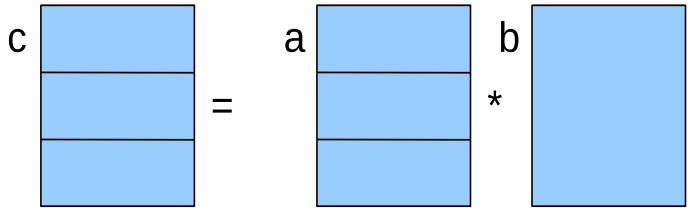
- Facilitan el desarrollo de software
- El software obtenido es portable
- y su eficiencia se basa en la implementación eficiente de las rutinas básicas.

Multiplicación de matices 1000×1000 en sistema con 12 núcleos, tiempo en segundos

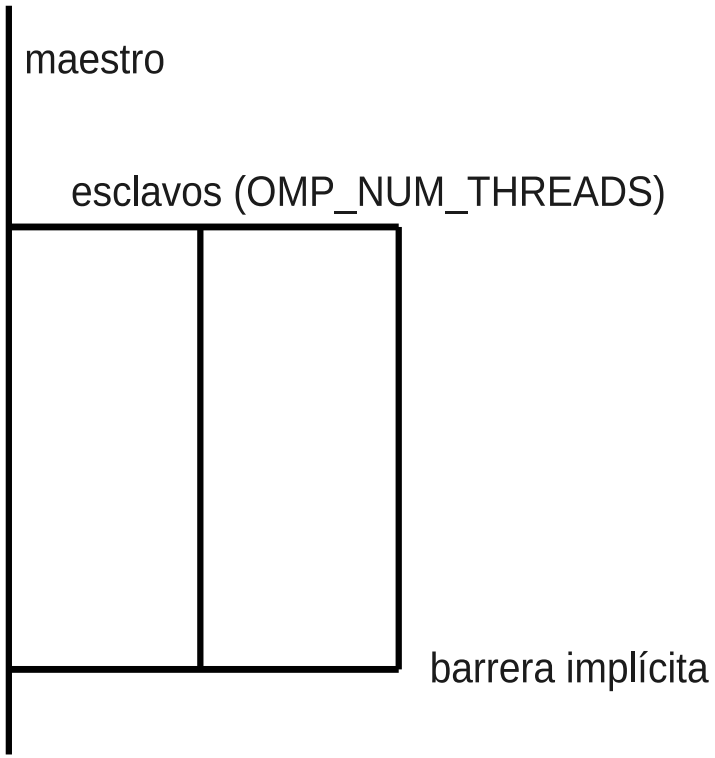
núcleos:	1	2	4	8	12
BLAS 1	11.15	10.66	10.64	10.63	10.63
BLAS 2	0.61	0.30	0.17	0.099	0.11
BLAS 3	0.14	0.072	0.044	**0.030**	0.055

de 0.25 Gflops, con uso de librería 20 Gflops, con paralelismo implícito a 100 Gflops

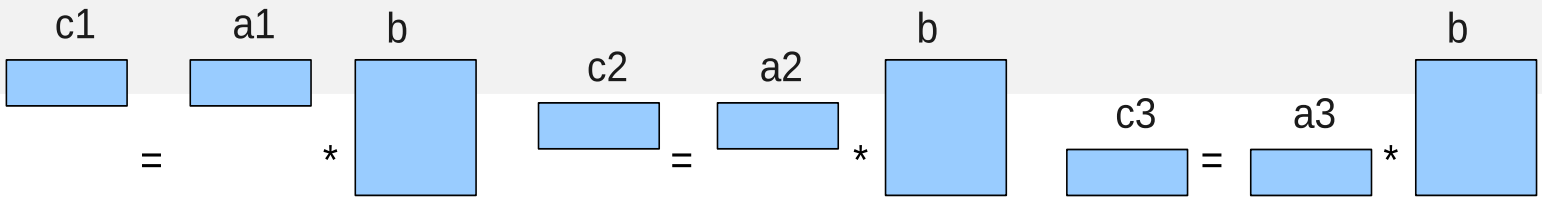
OpenMP

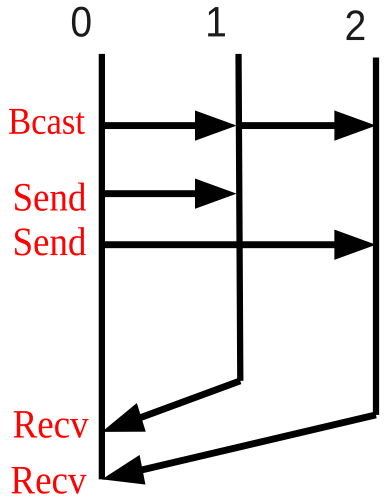



```
void multmat(double *a,double *b,double *c,int t)
{
  int i,j,k;
  double s;

  #pragma omp parallel for private(i,j,k,s)
    for(i=0;i<t;i++)
      for(j=0;j<t;j++)
      {
        s=0.;
        for(k=0;k<t;k++)
          s+=a[i*t+k]*b[k*t+j];
        c[i*t+j]=s;
      }
}
```



MPI

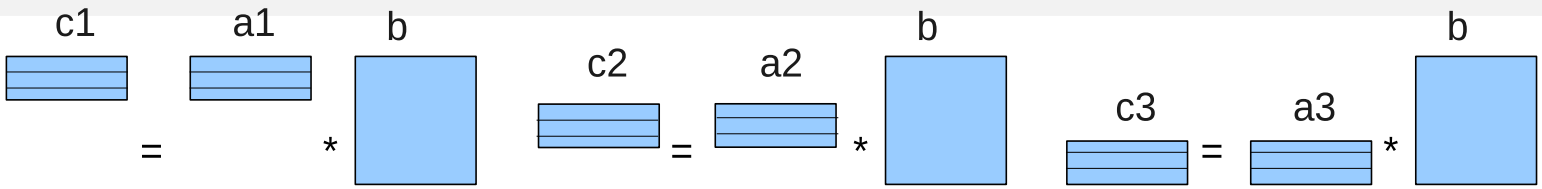


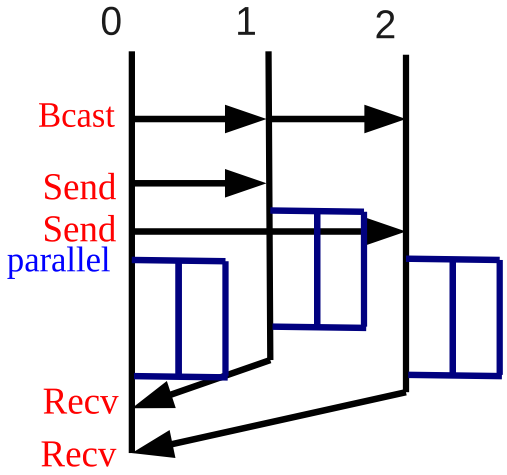



```
int main(int argc,char *argv[]) {
  MPI_Status estado;
  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD,&np);
  MPI_Comm_rank(MPI_COMM_WORLD,&nodo);
  if(nodo==0)  N=atoi(argv[1]);
  MPI_Bcast(&N,1,MPI_INT,0,MPI_COMM_WORLD);
  MPI_Barrier(MPI_COMM_WORLD);
  ti=MPI_Wtime();
  mm(a,fa,ca,lda,b,fb,cb,ldb,c,fc,cc,ldc,nodo,np);
  MPI_Barrier(MPI_COMM_WORLD);
  tf=MPI_Wtime();
  ...
  MPI_Finalize();
}
```

```
void mm(...) {
  MPI_Status *estado=NULL;
  if(nodo==0)
    for(i=1;i<np;i++)
      MPI_Send(&a[i*lda*fa/np],fa/np*ca,
        MPI_DOUBLE,i,20,MPI_COMM_WORLD);
    MPI_Bcast(b,fb*cb,MPI_DOUBLE,0,
      MPI_COMM_WORLD);
  }
  else {
    MPI_Recv(a,fa/np*ca,MPI_DOUBLE,0,20,
      MPI_COMM_WORLD,estado);
    MPI_Bcast(b,fb*cb,MPI_DOUBLE,0,
      MPI_COMM_WORLD);
  }
  mms(a,fa/np,ca,lda,b,fb,cb,ldb,c,fc/np,cc,ldc);
  if(nodo==0)
    for(i=1;i<np;i++)
      MPI_Recv(&c[i*ldc*fc/np],fc/np*cc,MPI_DOUBLE,
        i,30,MPI_COMM_WORLD,estado);
  else  MPI_Send(c,fc/np*cc,MPI_DOUBLE,0,30,
    MPI_COMM_WORLD);
}
```

MPI+OpenMP



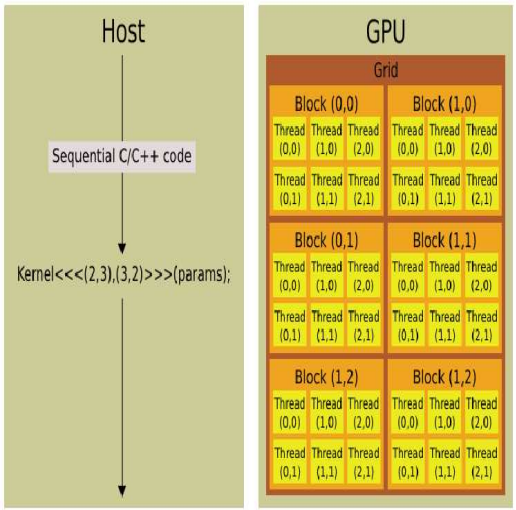



```
void mm(...) {
  MPI_Status *estado=NULL;
  if(nodo==0)
    for(i=1;i<np;i++)
      MPI_Send(&a[i*lda*fa/np],fa/np*ca,
        MPI_DOUBLE,i,20,MPI_COMM_WORLD);
    MPI_Bcast(b,fb*cb,MPI_DOUBLE,0,
      MPI_COMM_WORLD);
  }
  else {
    MPI_Recv(a,fa/np*ca,MPI_DOUBLE,0,20,
      MPI_COMM_WORLD,estado);
    MPI_Bcast(b,fb*cb,MPI_DOUBLE,0,
      MPI_COMM_WORLD);
  }
  mms(a,fa/np,ca,lda,b,fb,cb,ldb,c,fc/np,cc,ldc);
  if(nodo==0)
    for(i=1;i<np;i++)
      MPI_Recv(&c[i*ldc*fc/np],fc/np*cc,MPI_DOUBLE,
        i,30,MPI_COMM_WORLD,estado);
  else  MPI_Send(c,fc/np*cc,MPI_DOUBLE,0,30,
    MPI_COMM_WORLD);
}
```

```
void mms(...) {
  int i, j, k;
  double s;
  #pragma omp parallel for private(i,j,k,s)
  for (i=0;i<fa;i++)
    for(j=0;j<cb;j++) {
      s=0.;
      for (k=0;k<ca;k++)
        s+=a[i*lda+k]*b[k*ldb+j];
      c[i*ldc+j]=s;
    }
}
```

CUDA

```
double  *a_GPU,*b_GPU,*c_GPU;
unsigned int  nPos, nBytes;
dim3  ThreadsPerBlock, BlocksPerGrid;
nPos=t*t;
nBytes=nPos*sizeof(double);

ThreadsPerBlock.x = 4;
ThreadsPerBlock.y = 4;
ThreadsPerBlock.z = 1;

BlocksPerGrid.x = (t + ThreadsPerBlock.x - 1) / ThreadsPerBlock.x;
BlocksPerGrid.y = (t + ThreadsPerBlock.y - 1) / ThreadsPerBlock.y;

cudaMalloc((void **)&a_GPU, nBytes);
cudaMalloc((void **)&b_GPU, nBytes);
cudaMalloc((void **)&c_GPU, nBytes);

cudaMemcpy(a_GPU, a, nBytes, cudaMemcpyHostToDevice);
cudaMemcpy(b_GPU, b, nBytes, cudaMemcpyHostToDevice);

kernel_mm<<<BlocksPerGrid, ThreadsPerBlock>>>(t,
  a_GPU, b_GPU, c_GPU);

cudaThreadSynchronize();

cudaMemcpy(c, c_GPU, nBytes, cudaMemcpyDeviceToHost);

cudaFree((void *) a_GPU);
cudaFree((void *) b_GPU);
cudaFree((void *) c_GPU);

cudaThreadExit();
```




```
__global__ void kernel_mm(int t,
  double *a_GPU,double *b_GPU,double *r_GPU)
{
  unsigned int k,
    j = blockDim.x * blockIdx.x + threadIdx.x,
    i = blockDim.y * blockIdx.y + threadIdx.y;
  double aux;

  if (i<t && j<t) {
    aux=0.;
    for(k=0;k<t;k++)
      aux+=a_GPU[i*t+k]*b_GPU[k*t+j];
    r_GPU[i*t+j]=aux;
  }
  __syncthreads();
}
```

Intel Xeon Phi

- Coprocesador, en general más barato que GPU pero también con prestaciones más bajas.
- Ventaja: programación más cercana a la estándar (OpenMP y MPI).
- Inconveniente: tecnología en sus inicios.

```
double omp_reduction(double *data, int size)
{
    double ret = 0.0;
    #pragma offload target(mic) in(data:length(size))
    {
        #pragma omp parallel for reduction(+:ret)
            for (int i = 0; i < size; i++) {
                ret += data[i];
            }
    }
    return ret;
}
```

Programación Híbrida y Heterogénea

Para obtener el máximo provecho de sistemas computacionales actuales:

- Híbridos: combinan componentes de distintas características
- Heterogéneos: con distintas capacidades de almacenamiento y velocidad de computación y comunicación
- Jerárquicos: y organizados por niveles

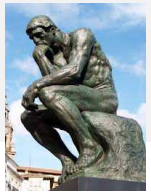
En cluster de multicore+multiGPU+multiXPhi:

Varios procesos, cada uno:

```
omp_set_num_threads(#Cores+#GPU+#XPhi)
#pragma omp parallel for
    for (int i = 0; i < #Cores+#GPU+#XPhi; i++) {
        if(i<#Cores) {
            llamar a rutina en multicore
        } else if(i<#Cores+#GPU) {
            llamar a rutina que lanza kernel en GPU
        } else {
            llamar a rutina que lanza trabajo en XPhi
        }
    }
```

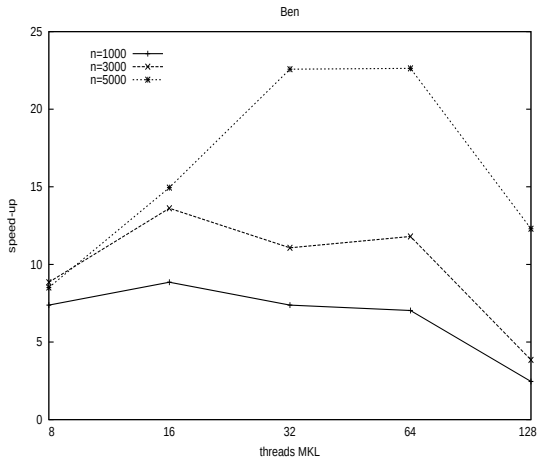
Problema de balanceo de la carga

Resultados cc-NUMA



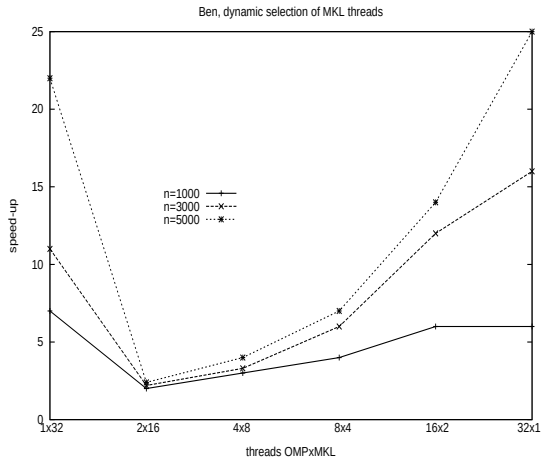
Multiplicación de matrices en Ben (128 cores), ganancia de velocidad respecto a secuencial:

llamando a MKL



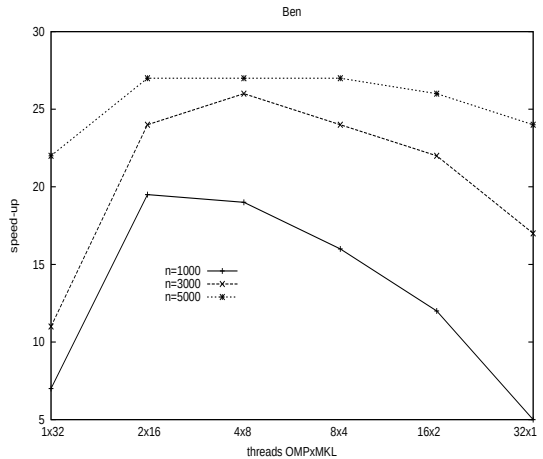
lejos de las prestaciones máximas

OpenMP+MKL dinámico



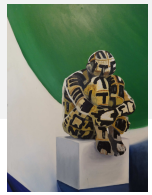
la selección dinámica de MKL no funciona bien con OpenMP

OpenMP+MKL fijando hilos



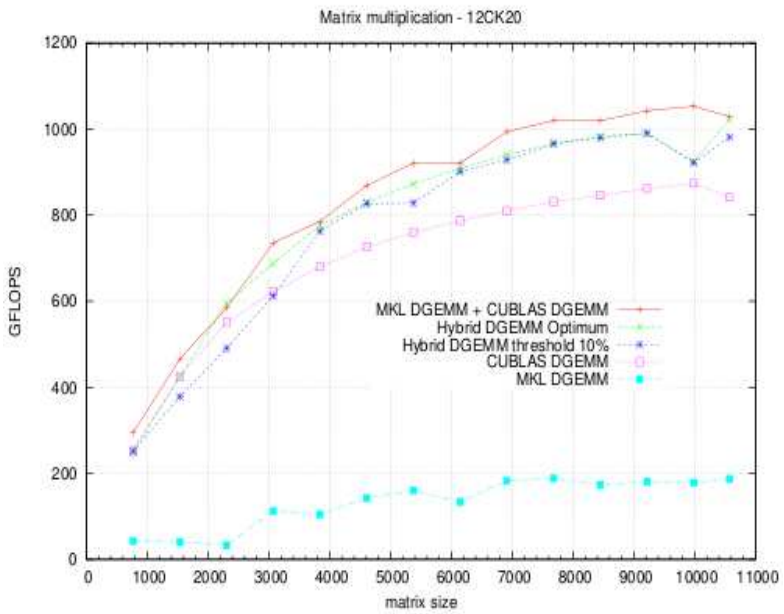
con paralelismo de dos niveles se aumentan las prestaciones

Resultados coprocesadores



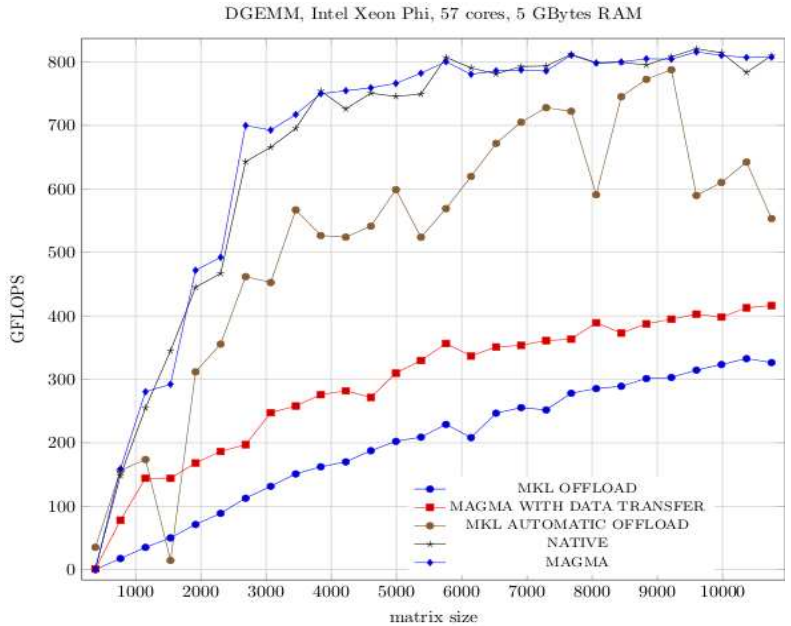
Multiplicación de matrices híbrida multicore+coprocesador, prestaciones obtenidas:

multicore+GPU



el híbrido mejora prestaciones de la librería, y la estrategia de autotuning da resultados cercanos al óptimo

multicore+XPhi



necesaria selección de librería y determinación automática de volumen de trabajo en host y coprocesador

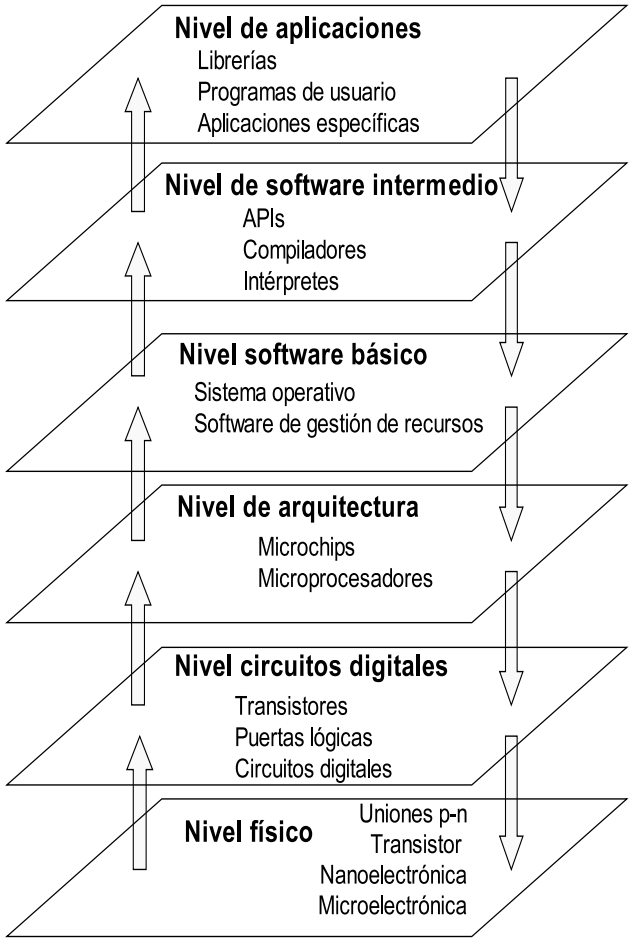
Referencias de Programación Paralela

- Grama, Karypis, Kumar, Gupta, Introduction to Parallel Computing, BenjaminCummings, 2003. (el clásico)
- Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2003.
- Almeida, Giménez, Mantas, Vidal, Introducción a la Programación Paralela, Paraninfo, 2008. (ejemplos en OpenMP y MPI)
- Material en la Facultad de Informática de la UMU:
  - Metodología de la Programación Paralela (`http://dis.um.es/~domingo/app.html`)
    Conceptos básicos de paralelismo, OpenMP, MPI, algoritmia paralela.
  - Programación Paralela y Computación de Altas Prestaciones (`http://dis.um.es/~domingo/cap.html`)
    Librerías paralelas, algoritmos matriciales secuenciales por bloques, out-of-core, paralelos, en coprocesadores.
  - Concurso Español de Programación Paralela (`http://luna.inf.um.es`)
    Problemas en OpenMP, MPI y CUDA. Tabla de records con los códigos.

Temas a tratar

Conclusión

- En la Computación Paralela se utiliza un enfoque ingenieril, con una organización por niveles,
- y es necesario trabajo en optimización de software y librerías para aumentar las prestaciones,
- de forma que con HPC se puedan resolver problemas de gran dimensión y alta complejidad,
- facilitando la resolución de problemas científicos cada vez de mayor interés,
- ... pero siempre quedan aspectos que no se pueden abordar.



Créditos

- Pedro Alonso, codirector de la tesis de Murilo
- Rafael Asenjo, transparencias de presentación en Jornadas de Paralelismo
- Barcelona Supercomputing Center, información e imágenes
- Murilo do Carmo Boratto, por información de su tesis
- Center for Parallel and Distributed Computing Curriculum Development and Educational Resources de la University of Georgia, información y acceso al cluster
- Javier Cuenca, por transparencias, códigos y resultados
- Luis Pedro García, información de sistemas en la UPCT, figura, resultados y códigos
- Google, acceso a información en la web
- Grupos de Investigación de la UMU, por su información en la web de investigación de la UMU
- María José Majado, por su óleo sobre una escultura de Jaume Plensa
- La Verdad, por su artículo del chef es el supercomputador
- Arturo Sólvez, información de sistemas en la UMU
- TOP500, imágenes e información
- Antonio M. Vidal, por material suyo del libro de Introducción a la Programación Paralela

Cuestiones

- ¿Hay grupos de la UMU que saquen provecho de la computación paralela?
- ¿Qué recursos computacionales utilizan?
- ¿Son suficientes los recursos computacionales y los conocimientos de paralelismo?
- ¿Sería conveniente una coordinación mayor para mejorar o poner en común el uso de los recursos,
- o para mejorar la formación en computación paralela?

¿Otras cuestiones?