

Conceptos y Métodos de la Computación Paralela	Código: CMCP
	Créditos ECTS: 4
<i>Profesores:</i> Antonio M. Vidal	

Objetivos

- Conocer los distintos modelos computacionales paralelos, tanto físicos como formales, y ser capaces de utilizarlos en la especificación, resolución, evaluación y diseño de algoritmos.
- Conocer las técnicas de evaluación de los algoritmos paralelos y su utilización en fase de diseño.
- Conocer las metodologías de diseño de algoritmos paralelos y ser capaz de utilizarlas en problemas y casos de estudio concretos.

Pre/Co-requisitos recomendados

Conocimientos de programación.

Programa resumido

1. Arquitecturas paralelas.
2. Modelos computacionales paralelos.
3. Complejidad de los algoritmos paralelos.
4. Evaluación y diseño de algoritmos paralelos.
5. Actualizaciones y casos de estudio en el ámbito de la computación paralela.
6. Entornos de programación paralela.

Metodología

La asignatura trata de desarrollar los aspectos básicos de la computación paralela a la vez que pretende que el alumno sea capaz de ir utilizándolos en el desarrollo de aplicaciones concretas. Por ello, la metodología queda condicionada por estos dos aspectos. Por una parte, se irán introduciendo los conceptos básicos a través de clases teóricas, complementándose con lecturas de artículos, textos, específicos, etc. Por otra parte se irán proponiendo una serie de Casos de Estudio que el alumno debe resolver, como si fueran pequeños proyectos. Estos Casos de Estudio se eligen convenientemente, de forma que su desarrollo obligue al alumno a alcanzar los objetivos previstos. Su dificultad se irá graduando en orden creciente con el fin de que el aprendizaje sea paulatino. Cada Caso de Estudio se desarrolla típicamente en tres etapas: Información teórica, implementación y documentación.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
30	4	-	14	6	26	20	

Evaluación

La evaluación de la asignatura estará basada en el análisis de los Casos de Estudio, desarrollados por el alumno. La evaluación de cada Caso de Estudio se realiza en

función de las actitudes mostradas, por el trabajo realizado, el cual debe quedar reflejado en una memoria, y por la defensa pública de la misma.

Bibliografía más relevante

- [1] I.Foster. "Designing and Building Parallel Programs", (Online), Addison-Wesley. <http://www-unix.mcs.anl.gov/dbpp/>
- [2] V.Kumar, A.Grama et al., "Introduction to Parallel Computing. Design and Analysis of Algorithms". 2ª Edición. Addison-Wesley (2003)
- [3] M.Snir, S.Otto et al., "MPI. The complete reference", J. Kowalik, Eds., Scientific and Engineering Computation. The MIT Press, Cambridge (1996)
- [4] J.L. Pérez y A.M. Vidal. "Introducción a la Programación en MPI". Libro de apuntes editado por el servicio de publicaciones de la UPV. Ref. 902.
- [5] B.Wilkinson y M.Allen. "Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers." Ed. Prentice-Hall. (1999)

Herramientas de Computación de Altas Prestaciones	Código: HCAP
	Créditos ECTS: 4
<i>Profesores: Victor M. García</i>	

Objetivos

Dar al alumno una formación básica en conceptos, compiladores y librerías orientadas a la computación científica y matricial de altas prestaciones.

Pre/Co-requisitos recomendados

Conocimientos de programación en algún lenguaje estructurado (C, Pascal, Fortran) y de algebra lineal básica. Recomendable haber trabajado con MATLAB o algún entorno similar.

Programa resumido

1. Conceptos básicos de Computación de altas prestaciones y Computación Matricial.
2. Lenguajes de programación para computación de altas prestaciones.
3. Jerarquía de memorias; su influencia y aprovechamiento; algoritmos escalares y a bloques.
4. Técnicas de Almacenamiento de Matrices (densas, estructuradas, dispersas)
5. Núcleos computacionales: BLAS, LAPACK
6. Optimización de código.

Metodología

Clases magistrales combinadas con clases de prácticas, en las que se implementarán y analizarán los algoritmos descritos en las clases magistrales.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
20	-	-	20	10	25	25	

Evaluación

La evaluación tendrá dos componentes

- 1) Evaluación continua en las clases de prácticas.
- 2) Realización de trabajos por parte del alumno, complementando y ampliando los propuestos en las clases de prácticas.

Bibliografía más relevante

- [1] Gene H. Golub, Charles F. van Loan, "Matrix computations", Johns Hopkins University Press, 1996.
- [2] E. Anderson et al., "LAPACK users' guide", SIAM, 1995.
- [3] Michael Metcalf, John Reid "Fortran 90/95 explained", Oxford University Press, 1998.

Conceptos de la Computación en Grid	Código: CCG
	Créditos ECTS: 4
<i>Profesores:</i> Vicente Hernández, Ignacio Blanquer	

Objetivos

Esta asignatura pretende proporcionar una visión completa de las tecnologías Grid utilizadas para compartir recursos computacionales ubicados en dominios de administración diferentes y geográficamente dispersos. El objetivo principal es resolver problemas, de computación intensiva, de interés para la comunidad que comparte dichos recursos. Para ello se persiguen los siguientes objetivos:

- Dar a conocer los conceptos básicos relativos a los entornos de computación en Grid.
- Describir los componentes de las arquitecturas software de los entornos Grid actuales.
- Proporcionar una visión del estado del arte en el ámbito de las tecnologías Grid.
- Dar a conocer y ejercitarse en los comandos básicos para la creación y ejecución de trabajos en Grid.

Pre/Co-requisitos recomendados

Conocimiento de sistemas UNIX/Linux a nivel de usuario, y de Sistemas Operativos en general. Conceptos básicos de la computación paralela. Conocimiento de programación sobre shell-scripts.

Programa resumido

1. Evolución de las Tecnologías de Computación Intensiva.
2. Infraestructuras Grid Actuales: Taxonomías y Recursos.
3. Componentes de los Sistemas Grid.
 - a. Arquitectura General de un Entorno Grid.
 - b. Infraestructura de Seguridad en el Grid.
 - c. Virtualización de Recursos: Gestores de Colas.
 - d. Estructura, Ejecución y Monitorización de Trabajos en Grid.
 - e. Descubrimiento de Recursos.
 - f. Gestión de Trabajos en Grid.
 - g. Acceso a Datos en Entornos Grid.
4. Arquitecturas Grid Avanzadas (WSRF, GT4, LCG, gLite).

Metodología

La asignatura ofrece una formación teórica y práctica, aunque todos los conceptos son en sí eminentemente prácticos. En lo que respecta a la formación teórica, se presentará una introducción de las tecnologías Grid y su diferenciación con otras tecnologías similares del ámbito de la computación paralela y distribuida. Se analizarán las tecnologías básicas que fundamentan el Grid y se detallarán los servicios específicos de estos entornos. Se describirán también las nuevas tendencias en este campo.

En cuanto a la formación práctica, se propone experimentar con los servicios Grid fundamentales, comenzando por la seguridad, el descubrimiento de recursos, la gestión de trabajos y el acceso a datos, con el objetivo de completar la formación

necesaria para la migración e implementación de programas que utilicen componentes básicos del Grid.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
22	-	-	14	4	14	46	

Evaluación

La evaluación se realizará mediante la presentación de trabajos teórico/prácticos que serán defendidos por el alumno en una sesión oral ante profesores del master.

Bibliografía más relevante

- [1] Foster I., Kesselman C., “The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann Publishers, Inc., 1998
- [2] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002, www.globus.org/research/papers/ogsa.pdf.
- [3] Expert Group Report, “Next Generation Grid(s)”, European Commission. <http://www.cordis.lu/ist/Grids/index.htm>
- [4] Berman F., Fox G., Hey T., “Grid Computing : Making the Global Infrastructure a Reality”, West Sussex : John Wiley & Sons, cop. 2003 .
- [5] Globus alliance Home Page. “Relevant documents”, <http://www.globus.org>
- [6] Enabling Grids in E-science (EGEE), <http://www.eu-egee.org>

Fundamentos de las Aplicaciones Distribuidas	Código: FAD
	Créditos ECTS: 4
<i>Profesores:</i> Pablo Galdámez	

Objetivos

Proporcionar a los alumnos los fundamentos teóricos más importantes de los sistemas distribuidos, que les permitan diseñar algoritmos y aplicaciones distribuidas con rigor, ayudándose para ello de herramientas formales y de una primera visión a la problemática tratada en los problemas distribuidos más relevantes de comunicación y sincronización.

Pre/Co-requisitos recomendados

Pre-requisito: Sistemas Distribuidos y conocimientos de informática teórica.

Programa resumido

1. Introducción a los Sistemas Distribuidos.
2. Modelos formales: autómatas de entrada/salida.
3. Algoritmos distribuidos.
4. Aplicaciones.

Metodología

La asignatura es fundamentalmente teórica y se impartirá en el aula haciendo uso de transparencias y proyector. Para la parte correspondiente a algoritmos distribuidos se hará mayor uso de la pizarra, pues parece una forma más adecuada para involucrar al alumno en el diseño activo de los algoritmos que se plantean en clase.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
25	-	15	-	2	28	30	

Evaluación

La evaluación consiste en la entrega de un trabajo que propondrá el profesor, donde el alumno deberá diseñar un algoritmo adecuado para algún tipo de aplicación distribuida, aplicando para ello las herramientas y metodologías propuestas en clase. Para aquellos alumnos que no entreguen el trabajo, se les planteará un examen teórico sobre el contenido de la asignatura.

Bibliografía más relevante

- [1] Nancy Lynch. "Distributed Algorithms." Morgan Kaufmann Publishers, San Mateo, CA, 1996.
- [2] Sape Mullender. "Distributed Systems." Addison-Wesley, 2nd edition. 1993.
- [3] Andrew S. Tanenbaum, Maarten van Oteem. "Distributed Systems. Principles and Paradigms". Prentice-Hall. 2002.

Algoritmos Paralelos Matriciales en Ingeniería	Código: APMI
	Créditos ECTS: 4
<i>Profesores:</i> Antonio M. Vidal	

Objetivos

La resolución de muchos problemas en Ingeniería acaba convirtiéndose en la implementación de un conjunto típico de algoritmos matriciales. Este tipo de algoritmos precisa de la utilización de herramientas computacionales paralelas potentes como las Librerías de Altas Prestaciones. El objetivo fundamental de esta asignatura es el conocimiento de este tipo de algoritmos y la adquisición de habilidades en el manejo de las Librerías de Altas Prestaciones que permiten su aplicación en la resolución práctica de problemas de Ingeniería. Los objetivos específicos de la asignatura son los siguientes:

- Conocer los algoritmos matriciales más utilizados en Ingeniería.
- Conocer las Librerías numéricas de Altas prestaciones presentes en el mercado actualmente.
- Adquisición de habilidades en el manejo avanzado de Librerías de Altas Prestaciones.
- Resolución de pequeños problemas de Ingeniería utilizando Librerías de Altas Prestaciones.

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela, Herramientas de Computación de Altas Prestaciones

Programa resumido

1. Modelado de problemas en Ingeniería. Problemas lineales y no lineales.
2. Computación secuencial y paralela de descomposiciones matriciales: descomposiciones finitas (LU, QR), cálculo de valores propios y singulares.
3. Resolución de sistemas no lineales y problemas de optimización.
4. Manejo de librerías específicas de altas prestaciones.
5. Diseño de algoritmos paralelos utilizando librerías de altas prestaciones.
6. Casos de estudio en Ingeniería.

Metodología

En una parte de la asignatura se desarrollarán los algoritmos matriciales más utilizados en Ingeniería a través de clases teóricas, complementándose con lecturas de artículos, textos, específicos, etc. La adquisición de habilidades en el manejo de librerías requiere, en cambio, una metodología más práctica, proponiéndose a este fin una parte importante de experimentación a través de clases y desarrollos prácticos. Las clases prácticas se organizan en clases de aprendizaje, donde el alumno va realizando ejercicios de dificultad creciente que garanticen el conocimiento y cierta habilidad en el manejo de librerías. Se propone también el desarrollo de Casos de Estudio que el alumno debe resolver, con la ayuda de las herramientas estudiadas, como si fueran pequeños proyectos de Ingeniería.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
20	2	2	20	6	30	20	

Evaluación

La evaluación de la asignatura estará basada en el seguimiento de las prácticas y en el análisis de los Casos de Estudio desarrollados por el alumno. El trabajo realizado debe quedar reflejado en una memoria.

Bibliografía más relevante

- [1] J.Wilkinson. "The Algebraic Eigenvalue Problem". Oxford Univ. Press. 1965
- [2] G.H.Golub and C.F. Van Loan. "Matrix Computations". The Johns Hopkins Univ. Press, 3rd Ed. Baltimore, 1996.
- [3] J.J. Dongarra et al., "Numerical Linear Algebra for High-Performance Computers". SIAM, Philadelphia, 1998.
- [4] C.T.Kelley "Iterative Methods for Linear and Nonlinear Equations" SIAM, 1995
- [5] E.Anderson et al. "LAPACK Users' Guide Third Edition". SIAM 1999.
- [6] ScaLAPACK <http://www.netlib.org/scalapack/html>
- [7] BLACS. <http://www.netlib.org/blacs/index.html>
- [8] PBLAS. http://www.netlib.org/scalapack/html/pblas_qref.html

Tecnología de la Programación Paralela	Código: TPP
	Créditos ECTS: 4
<i>Profesores:</i> José E. Román	

Objetivos

Conocer a nivel práctico las distintas posibilidades existentes para desarrollar programas paralelos. Adquirir destrezas en el desarrollo de este tipo de programas utilizando los paradigmas más extendidos.

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela.

Programa resumido

1. Revisión de tecnologías de programación paralela.
2. Paradigma de memoria compartida: OpenMP.
3. Paradigma de paso de mensajes: MPI.
4. Herramientas de desarrollo de software paralelo.
5. Esquemas de programación paralela.

Metodología

En las sesiones de teoría se presentan los aspectos teóricos básicos con ejemplos de aplicación. En las sesiones prácticas se desarrollarán programas paralelos para diversos ejercicios propuestos. Adicionalmente, el alumno aplicará los conocimientos adquiridos al desarrollo de un proyecto de mayor complejidad.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
12	-	-	22	6	40	20	

Evaluación

La evaluación consta de dos partes. La parte de teoría (30 %) se evalúa mediante un examen. La parte de prácticas se evalúa mediante un portafolio (20 %) que comprende las soluciones comentadas a los ejercicios propuestos en prácticas y un proyecto (50 %) consistente en el desarrollo de un programa paralelo más completo. Los proyectos podrán realizarse individualmente o en grupos de dos alumnos, dependiendo de su complejidad.

Bibliografía más relevante

- [1] A. Grama, A. Gupta, G. Karypis, V. Kumar., "Introduction to Parallel Computing", 2nd ed., Addison-Wesley, 2003.
- [2] R. Chandra et al., "Parallel Programming in OpenMP", Morgan-Kaufmann, 2001.
- [3] S. Akhter, J. Roberts, "Multi-core Programming", Intel Press, 2006.
- [4] W. Gropp, E. Lusk, A. Skjellum, "Using MPI: Portable Parallel Programming with the Message Passing Interface", 2nd ed., The MIT Press, 1999.
- [5] B. Wilkinson, M. Allen, "Parallel Programming", Prentice-Hall, 1998.

Programación Avanzada en Entornos Grid tipo Batch	Código: PAEGB
	Créditos ECTS: 3
<i>Profesores:</i> Vicente Hernández, Ignacio Blanquer	

Objetivos

Esta asignatura tiene un carácter eminentemente práctico y se orienta a las técnicas de programación en los entornos Grid orientados a la ejecución masiva de trabajos. Las prácticas se realizarán sobre el entorno pre-Web Services de Globus Toolkit (Globus Toolkit 2 o GT2), que constituye el estándar de facto para el despliegue de entornos de este tipo. Se considerará tanto la programación basada en scripts de línea de comandos como la basada en APIs de bibliotecas software. Para ello se persiguen los siguientes objetivos:

- Desarrollar en el alumno las habilidades para diseñar e implementar programas Grid eficientes en entornos orientados a trabajos.
- Ser capaz de ejecutar y evaluar programas Grid en entornos Grid tipo Batch.
- Conocer los detalles específicos de los entornos Grid tipo Batch actuales (especialmente GT2).

Pre/Co-requisitos recomendados

Conocimientos de sistemas UNIX/Linux a nivel de usuario y de Sistemas Operativos en general. Conceptos básicos de la computación paralela. Conocimientos de programación estructurada en “C” y Java.

Asignaturas del máster: Conceptos de la Computación en Grid.

Programa resumido

1. Estructura de Entornos Grid tipo Batch: El Entorno Globus Toolkit 2 (GT2).
 - a. Infraestructura de Seguridad en Grid.
 - b. Estructura de una Entidad Certificadora.
 - c. Creación de Proxies.
 - d. Servidores de Claves.
2. Gestión de Trabajos.
 - a. Ciclo de Vida Avanzado.
 - b. Estructura Básica de un Trabajo.
 - c. Integración con las Políticas Administrativas Locales.
 - d. Trabajos Paralelos.
 - e. Especificación de Trabajos (RSL). Prioridades, Restricciones y Relanzamiento.
3. Acceso a Datos.
 - a. Protocolos de Acceso a Datos.
 - b. Gestión de Catálogos y Manejo de Réplicas.
4. Sistema de Información en Grid.
 - a. Estructura Jerárquica de los Sistemas de Información.
 - b. Descubrimiento de Recursos.
5. Otros Entornos tipo Batch.

Metodología

La docencia se impartirá directamente en los laboratorios, y cada unidad temática estará precedida de una breve introducción que explique los casos prácticos en los que se enmarca el contenido de la misma.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
7	-	-	20	3	25	20	

Evaluación

La evaluación se realizará mediante la presentación de trabajos teórico/prácticos que serán defendidos por el alumno en una sesión oral ante profesores del master.

Bibliografía más relevante

- [1] Globus alliance Home Page. “Relevant documents”, <http://www.globus.org>
- [2] Enabling Grids in E-science (EGEE), <http://www.eu-egee.org>
- [3] Foster I., Kesselman C., “The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann Publishers, Inc., 1998

Seguridad en Sistemas Distribuidos	Código: SSD
	Créditos ECTS: 3
<i>Profesores: Pablo Galdámez</i>	

Objetivos

Proporcionar a los alumnos los conceptos más importantes de la seguridad informática aplicada a sistemas distribuidos, poniendo especial énfasis en los algoritmos distribuidos necesarios para lograr seguridad y mostrar los problemas más relevantes aplicados a las variantes más actuales de sistemas distribuidos, como son los sistemas para alta disponibilidad, los sistemas grid, las redes ad-hoc o los entornos peer-to-peer.

Pre/Co-requisitos recomendados

Pre-requisito: Sistemas Distribuidos, Fundamentos de las Aplicaciones Distribuidas.

Co-requisito: Diseño de Aplicaciones Distribuidas Robustas, Sistemas peer-to-peer, Sistemas móviles.

Programa resumido

1. Conceptos fundamentales de seguridad informática.
2. Introducción a la seguridad en sistemas distribuidos.
3. Mecanismos de seguridad y algoritmos distribuidos.
4. Seguridad en sistemas distribuidos actuales.

Metodología

La asignatura es fundamentalmente teórica y se impartirá en el aula haciendo uso de transparencias y proyector. Para la parte correspondiente a algoritmos distribuidos seguros, se hará mayor uso de la pizarra, pues parece una forma más adecuada para involucrar al alumno en el diseño activo de los algoritmos que se plantean en clase. En la última parte de la asignatura, se distribuyen al alumno una serie de artículos de investigación seminales, que se discutirán en el aula.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
25	-	5	-	2	-	43	

Evaluación

La evaluación consiste en la entrega de un trabajo que propondrá el profesor, donde el alumno deberá analizar los mecanismos de seguridad de cierto sistema distribuido, desde un punto de vista crítico y basándose en los mecanismos y algoritmos estudiados en clase.

Bibliografía más relevante

- [1] William Stallings. "Cryptography and Network Security. Principles and Practices." Prentice-Hall, 3rd edition. 2003.

- [2] Andrew S. Tanenbaum, Maarten van Oteem. “Distributed Systems. Principles and Paradigms”. Prentice-Hall. 2002.
- [3] Colección de artículos distribuidos en clase.

Tecnologías de las Aplicaciones Distribuidas	Código: TAD
	Créditos ECTS: 3
<i>Profesores:</i> Jordi Bataller, Luis Irún	

Objetivos

Con esta asignatura se pretende que el alumno conozca y pueda utilizar los mecanismos más utilizados en la programación de sistemas distribuidos. Estos tienen que ver fundamentalmente con las comunicaciones entre procesos y con la sincronización de los mismos.

Pre/Co-requisitos recomendados

Conviene que alumno tenga soltura en la programación con el lenguaje Java y conozca los fundamentos del lenguaje de modelado UML.

Asignaturas de nivelación: Sistemas distribuidos.

Asignaturas del máster: Fundamentos de las aplicaciones distribuidas.

Programa resumido

1. Repaso de los fundamentos sobre procesos y threads, de su comunicación (paso de mensajes / memoria compartida) y de su sincronización.
2. Sockets.
3. RMI.
4. Corba.
5. Sincronización de threads.

Nota: las bibliotecas estudiadas para los puntos 2 a 5 son las correspondientes al lenguaje Java.

Metodología

La forma de trabajo será muy práctica. Una vez presentados los fundamentos y un programa ejemplo sobre el punto objeto de estudio, el alumno, partiendo del ejemplo, lo ampliará inmediatamente. Adicionalmente se propondrán trabajos (programas) individuales y colectivos que serán realizados de forma autónoma aunque con la ayuda del profesor si fuera necesaria. La revisión de esta colección de programas será la que determine la calificación en la asignatura.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
14	-	-	20	3	25	13	

Evaluación

La colección de programas realizados en prácticas, revisados presencialmente (profesor-autor) será la que determine la nota final.

Bibliografía más relevante

[1] El lenguaje unificado de modelado / Grady Booch, James Rumbaugh, Ivar Jacobson

[2] Thinking in Java / Bruce Eckel

[3] Client-server programming with Java and CORBA / Robert Orfali, Dan Harkey

[4] Client/server and open systems : Technologies and the tools that make them work / Rand Dixon

Diseño de Aplicaciones Distribuidas Robustas	Código: DADR
	Créditos ECTS: 4
<i>Profesores:</i> Francesc D. Muñoz, Luis Irún	

Objetivos

Presentar los conceptos básicos para entender y diseñar aplicaciones distribuidas altamente disponibles.

Pre/Co-requisitos recomendados

Asignaturas de nivelación: Sistemas distribuidos, Redes de área local e interconexión de redes, Diseño de sistemas operativos.

Asignaturas del máster: Fundamentos de las aplicaciones distribuidas, Tecnologías de las aplicaciones distribuidas.

Programa resumido

1. Concepto y modelos de fallo. Sistemas confiables.
2. Componentes de una arquitectura altamente disponible.
3. Gestión de la pertenencia. Detectores de fallos.
4. Sistemas de comunicación a grupo.
5. Replicación de componentes. Consistencia. Modelos de replicación.
6. Soporte transaccional.
7. Estrategias de recuperación.

Metodología

Se utilizarán clases teóricas en las que se propondrán problemas a resolver en la propia aula y otras actividades de tipo práctico que el alumno deberá resolver por su cuenta, entregando la correspondiente memoria de actividad.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
30	-	15	-	5	25	25	

Evaluación

Se evaluarán las memorias de actividad entregadas por cada alumno. También se realizará una prueba escrita final. La calificación del alumno será la media de estas dos calificaciones (actividades: 50%, examen: 50%).

Bibliografía más relevante

- [1] Sape J. Mullender: "Distributed Systems", 2ª ed., Addison-Wesley, ISBN 0-201-62427-3, 1993.
- [2] Andrew S. Tanenbaum, Maarten van Steen: "Distributed Systems: Principles and Paradigms", Prentice Hall, ISBN 0-13-088893-1, 2002.
- [3] Kenneth P. Birman: "Reliable Distributed Systems: Technologies, Web Services and Applications", Springer, ISBN 0-387-21509-3, 2004.

Programación en Entornos Grid Orientados a Servicios	Código: PEGOS
	Créditos ECTS: 3
<i>Profesores:</i> Vicente Hernández, Ignacio Blanquer	

Objetivos

Esta asignatura, eminentemente práctica, pretende formar al alumno en la migración y programación de aplicaciones sobre entornos Grid orientados a servicios, contemplando su integración con otras tecnologías y estándares del W3C.

Los objetivos concretos que se persiguen son:

- Desarrollar en el alumno las habilidades para diseñar e implementar Servicios Grid.
- Ser capaz de ejecutar y evaluar Programas Grid en entornos de producción avanzados.
- Conocer los detalles específicos de los Entornos Grid más destacados en la actualidad (especialmente GT4).

Pre/Co-requisitos recomendados

Conocimientos de sistemas UNIX/Linux a nivel de usuario y de Sistemas Operativos en general. Conceptos básicos de la computación paralela. Conocimientos de programación estructurada en “C” y Java.

Asignaturas del máster: Conceptos de la Computación en Grid.

Programa resumido

1. Estructura de Entornos Grid Orientados a Servicios.
 - a. La Arquitectura de Servicios Grid Abierta (OGSA).
 - b. El Estándar WSRF (Web Service Resource Framework).
 - c. El Entorno Globus Toolkit 4 (GT4).
2. Conceptos de Servicios Grid.
 - a. Diferencias entre Servicios Web y Servicios Grid.
 - b. Implementación de Servicios Grid Básicos.
3. Recursos Únicos.
 - a. Recursos y Servicios.
 - b. Lenguajes de Especificación.
 - c. Implementación de Servicios Grid Avanzados.
4. Recursos Múltiples.
 - a. Factorías, Instancias y Recursos.
 - b. La especificación WS-Resource Factory.
 - c. Ciclo de Vida de Recursos
 - d. Notificaciones.
5. Infraestructura de Seguridad en Arquitecturas Orientadas a Servicios (SOA).
 - a. Servicios Grid Relativos a Seguridad.
 - b. Gestores de Organizaciones Virtuales (VOMS)
 - c. Gestores de Listas de Control de Acceso.

Metodología

La docencia se impartirá directamente en los laboratorios, y cada unidad temática estará precedida de una breve introducción que explique los casos prácticos en los que se enmarca el contenido de la misma.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
7	-	-	20	3	25	20	

Evaluación

La evaluación se realizará mediante la presentación de trabajos teórico/prácticos que serán defendidos por el alumno en una sesión oral ante profesores del master.

Bibliografía más relevante

- [1] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, 2002, www.globus.org/research/papers/ogsa.pdf.
- [2] Expert Group Report, “Next Generation Grid(s)”, European Commission. <http://www.cordis.lu/ist/Grids/index.htm>
- [3] Berman F., Fox G., Hey T., “Grid Computing : Making the Global Infrastructure a Reality”, West Sussex : John Wiley & Sons, cop. 2003 .
- [4] Globus alliance Home Page. “Relevant documents”, <http://www.globus.org>
- [5] gLite, a Lightweight Middleware for Grid Computing, Enabling Grids in E-science (EGEE), <http://www.eu-egee.org/glite>

Librerías de Altas Prestaciones para Problemas Algebraicos Dispersos	Código: LAPPAD
	Créditos ECTS: 4
<i>Profesores:</i> Vicente Vidal	

Objetivos

- Conocer las técnicas computacionales más utilizadas para la resolución de problemas de ingeniería, a través del análisis de casos de estudio cercanos a problemas reales.
- Adquirir habilidades y destrezas en la implementación secuencial y paralela de algoritmos numéricos, sobre computadores secuenciales y paralelos, empleando técnicas de computación de altas prestaciones.
- Utilizar librerías específicas

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela, Herramientas de Computación de Altas Prestaciones.

Programa resumido

1. Modelado de Aplicaciones de Ingeniería. Estructuras de almacenamiento. Matrices estructuradas y no estructuradas.
2. Resolución de aplicaciones de ingeniería deducidas de problemas dinámicos y estáticos de gran dimensión dispersos. Métodos de Krylov.
3. Librerías basadas en métodos de Krylov.
4. Diseño de algoritmos secuenciales y paralelos.
5. Aplicaciones y casos de estudio de problemas de ingeniería.

Metodología

La asignatura trata de analizar los aspectos básicos de cómo abordar un problema real de ingeniería. Se pretende que el alumno sea capaz de ir utilizando las herramientas de computación de altas prestaciones más adecuadas en el desarrollo de aplicaciones concretas. Para lograr este propósito utilizaremos la siguiente metodología. Por una parte, se irán introduciendo los conceptos básicos a través de clases teóricas, complementándose con lecturas de artículos, textos, específicos, etc. Por otra parte se le asignará a cada alumno un problema real de ingeniería, que deberá ir resolviendo como pequeños proyectos durante el transcurso de las explicaciones de las materias de la asignatura. Los pequeños proyectos se elegirán convenientemente, de forma que su desarrollo obligue al alumno a alcanzar los objetivos previstos.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
25	5	-	18	6	26	20	

Evaluación

La evaluación de la asignatura estará basada en el análisis de los Casos de Estudio, desarrollados por el alumno. La evaluación de cada Caso de Estudio se realiza en

función de las actitudes mostradas, por el trabajo realizado, el cual debe quedar reflejado en una memoria, y por la defensa pública de la misma.

Bibliografía más relevante

- [1] B.N. Datta, "Numerical Linear Algebra and Applications", Brooks/Cole Publishing Company, Boston, 1994.
- [2] J.W. Demmel, "Applied Numerical Linear Algebra", SIAM, Philadelphia, 1997.
- [3] G.H. Golub and C.F. Van Loan, "Matrix Computations", The Johns Hopkins Univ. Press, 3rd Ed., Baltimore, 1996.
- [4] V. Kumar et al., "Introduction to Parallel Computing: Design and Analysis of Algorithms", The Benjamin/Cummings Publishing Company, Redwood City, 1994.
- [5] Y. Saad, "Numerical Methods for Large Eigenvalue Problems", Halstead Press, New York, 1992.
- [6] Y. Saad, "Iterative Methods for Sparse Linear Systems", 2nd Ed., SIAM, Philadelphia, 2003.
- [7] C.F. Van Loan, "Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB", Prentice Hall, New Jersey, 1997.
- [8] Henk van der Vorst, "Iterative Krylov Methods for Large Linear systems", Cambridge University Press, Cambridge, April 2003.

Computación Paralela Heterogénea	Código: CPH
	Créditos ECTS: 2
<i>Profesores:</i> Pedro Alonso	

Objetivos

La implementación de algoritmos eficientes en entornos heterogéneos de computadores constituye un reto particular y bien diferenciado de la programación paralela en entornos homogéneos consistente en obtener el máximo rendimiento de la potencia de computadores de distinta naturaleza trabajando juntos en la solución de un determinado problema científico caracterizado éste casi siempre por requerir una gran potencia de cómputo. En esta asignatura se pretende realizar una aproximación a este problema y su solución mediante la adquisición de destreza en el manejo de ciertas herramientas existentes para la programación de estos entornos. Se tratará su aplicabilidad tanto en el campo de la industria así como en el campo de la investigación donde las redes heterogéneas pueden resolver las necesidades computacionales requeridas por ciertos problemas científicos.

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela, Herramientas de Computación de Altas Prestaciones.

Programa resumido

1. Evolución de la computación paralela. Lugar histórico de las Redes Heterogéneas de Computadores (RHC).
2. Desafíos de la programación de RHC.
3. Modelos de medida de prestaciones: rendimiento y productividad.
4. Modelos de particionado y distribución de datos en RHC.
5. Herramientas de programación en entornos de RHC.
6. Diseño de “Modelos de Prestaciones” y programación de RHC con HeteroMPI.
7. Aplicaciones de la Computación Paralela Heterogénea.

Metodología

Aunque la metodología de la asignatura se basa en clases teóricas (puntos 1-5 del temario) y las prácticas se basan en el aprendizaje y utilización de la herramienta HeteroMPI se realizará especial hincapié en las aplicaciones y utilidades de los entornos heterogéneos. También se plantearán a discusión diversos temas que hoy día permanecen abiertos a la investigación para mejorar de la facilidad de uso y aprovechamiento de estos entornos.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
16	-	4	8	4	12	6	

Evaluación

La asignatura se evalúa mediante la presentación de un trabajo práctico y otro de investigación/documentación de algún concepto específico del campo que el alumno trabajará y pondrá en común con el resto.

Bibliografía más relevante

[1] Alexey L. Lastovetsky, Parallel Computing on Heterogeneous Networks, Jhon Wiley and Sons, Inc..

Artículos y Manuales que serán proporcionados al alumno en el curso.

Gestión y Visualización de Datos en Computación Científica	Código: GVDCC
	Créditos ECTS: 2
<i>Profesores:</i> José E. Román	

Objetivos

Conocer las diferentes estructuras de datos que se utilizan habitualmente en aplicaciones de computación científica, especialmente en el caso de aplicaciones paralelas que requieren gestionar grandes volúmenes de datos. Saber cómo se pueden gestionar y almacenar estos datos, así como conocer las principales técnicas de visualización de los resultados.

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela.

Programa resumido

1. Estructuras de datos para computación científica.
2. Gestión de datos en aplicaciones paralelas.
3. Entrada-salida en aplicaciones paralelas.
4. Entornos y técnicas de visualización.

Metodología

En las clases teóricas se expondrán los conceptos fundamentales, haciendo referencia a diverso software disponible para cada uno de los aspectos tratados. En las sesiones prácticas se trabajará con varias de estas librerías, incidiendo en una de ellas para el trabajo de la asignatura.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
7	-	-	10	3	20	10	

Evaluación

Los alumnos deberán realizar un trabajo práctico, el cual puede ser propuesto por el profesor o por el alumno. Los trabajos serán individuales o en grupos de dos alumnos, dependiendo de su complejidad. La evaluación del trabajo se hará en base a la memoria y la presentación realizada.

Bibliografía más relevante

- [1] C. D. Hansen, C. R. Johnson (eds.), "The Visualization Handbook", Elsevier, 2005.
- [2] A. Grama, A. Gupta, G. Karypis, V. Kumar, "Introduction to Parallel Computing", (2nd ed.), Addison-Wesley, 2003.

Sistemas Peer-to-Peer	Código: SP2P
	Créditos ECTS: 2
<i>Profesores:</i> Francesc D. Muñoz, Adolfo Ferré	

Objetivos

Identificar las ventajas e inconvenientes de los sistemas distribuidos descentralizados (o "sistemas peer-to-peer"), así como su aplicabilidad a la hora de desarrollar ciertos tipos de aplicaciones distribuidas.

Pre/Co-requisitos recomendados

Conocimientos de sistemas distribuidos, redes de área local e interconexión de redes, administración de sistemas operativos,...

Asignaturas del máster: Fundamentos de las aplicaciones distribuidas.

Programa resumido

1. El modelo "peer-to-peer".
2. Encaminamiento. Localización de recursos.
3. Seguridad en entornos descentralizados.
4. Gestión del contenido: replicación, migración y "caching".
5. Gestión de recursos.
6. Ejemplos de aplicaciones.

Metodología

Se alternarán clases magistrales (donde el profesor explicará los conceptos fundamentales de cada tema y propondrá el estudio, en grupo, de artículos complementarios), con clases de seminario (donde cada grupo expondrá las ventajas e inconvenientes de la solución estudiada, respondiendo a las cuestiones que planteen otros grupos y el profesor). Deberá entregarse una memoria sobre cada uno de los seminarios donde cada alumno resuma los principales resultados de cada una de estas sesiones.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
7	10	-	-	5	8	20	

Evaluación

La evaluación se realizará en función de la participación de cada alumno en los seminarios, así como de la calidad de las memorias entregadas. No se realizará ninguna prueba final.

Bibliografía más relevante

- [1] Ian J. Taylor: "From P2P to Web Services and Grids: Peers in a Client/Server World", Springer, ISBN 1-85253-869-5, 2004.
- [2] S. Androutsellis-Theotokis, D. Spinellis: "A Survey of Peer-to-Peer Content Distribution Technologies", ACM Computing Surveys, vol. 36, nº 4, pgs 335-371, diciembre 2004.

Algoritmos Paralelos en Procesamiento de la Señal	Código: APPS
	Créditos ECTS: 2
<i>Profesores:</i> Antonio M. Vidal, Víctor M. García	

Objetivos

Este es un campo donde se pone de manifiesto la necesidad de resolver problemas de complejidad alta en tiempo real o al menos con tiempos de respuesta pequeños. La asignatura ofrece una visión computacional de los principales métodos numéricos utilizados en Procesamiento de la Señal y de la implementación de algoritmos paralelos eficientes en ese campo. Es una asignatura muy específica pero que abre al alumno un amplio campo de posibilidades donde encontrar aplicaciones de la computación paralela.

- Proporcionar al alumno una visión computacional de los problemas más usuales en Procesamiento de la Señal y analizar las herramientas existentes para su resolución en el marco de la Computación de Altas Prestaciones.
- Estudiar los métodos numéricos más usados actualmente en Procesamiento de la Señal y analizar una paralelización eficiente de los algoritmos resultantes en función de las especificaciones dadas del problema.
- Estudiar problemas propios del procesado digital de señales y analizar los beneficios que proporciona la Computación de Altas Prestaciones en su resolución.

Pre/Co-requisitos recomendados

Asignaturas del máster: Conceptos y Métodos de la Computación Paralela, Herramientas de Computación de Altas Prestaciones

Programa resumido

1. Conceptos computacionales básicos en Procesamiento de la Señal
2. Los problemas de mínimos cuadrados. Algoritmos secuenciales y paralelos. Aplicaciones en procesado de señal.
3. Marco computacional secuencial y paralelo para la Transformada rápida de Fourier y otras transformadas útiles en procesado de señales digitales (Transformada Wavelet,...)
4. Procesamiento paralelo de matrices estructuradas.
5. Aplicaciones. Casos de estudio

Metodología

La metodología propuesta para este curso se adapta perfectamente al carácter muy específico de la asignatura. En este curso se aborda cada tema como un caso de estudio. El alumno debe implementar una solución paralela atendiendo a unas especificaciones dadas (tiempo real, gran tamaño, algoritmos on-line, etc.) y compararla, caso de que sea posible, con la solución que ofrecen otros algoritmos implementados en librerías especializadas.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
10	-	-	10	6	10	14	

Evaluación

Por tratarse de una asignatura muy específica se evalúa fundamentalmente la calidad de los trabajos desarrollados por el alumno, su capacidad de generar soluciones eficientes y su presentación escrita y oral.

Bibliografía más relevante

- [1] T.K. Moon, W.C. Stirling. “Mathematical methods and algorithms for signal processing”. Prentice-Hall, (1999)
- [2] C. Van Loan. “Computational Frameworks for the Fast Fourier Transform”. SIAM (1992)
- [3] I. Daubechies. “Ten Lectures on Wavelets”. SIAM (1992)
- [4] G.G.Walter & X.Shen. “Wavelets and other Orthogonal Systems”. Chapman and HALL/CRC (2000)
- [5] T. Kailath y A. H. Sayed, “Displacement structure: Theory and applications”, SIAM Review, 37(3):297-386, September (1995).
- [6] A. H. Sayed, T. Kailath Eds, “Fast reliable algorithms for matrices with structures”, SIAM Pub. (2000).

Sistemas Móviles	Código: SM
	Créditos ECTS: 2
<i>Profesores:</i> Jordi Bataller, Adolfo Ferré	

Objetivos

Los dispositivos móviles “programables” (teléfonos móviles y PDAs p.ej.) son hoy en día ubicuos y sus capacidades van en aumento. Con este seminario se pretende principalmente que alumno conozca los fundamentos de su programación y qué tipo de aplicaciones distribuidas pueden hoy en día desarrollarse con ellos.

Pre/Co-requisitos recomendados

Asignaturas del máster: Fundamentos de las Aplicaciones Distribuidas.

Programa resumido

1. Principales dispositivos móviles programables: presentación y descripción.
2. Programación de teléfonos móviles con Java.
3. Programación de PDAs con Java y C++.
4. Fundamentos de encaminamiento en redes de dispositivos móviles
5. Pertenencia a grupos dinámicos de dispositivos móviles

Metodología

La forma de trabajo será muy práctica. Una vez presentados los fundamentos y un programa ejemplo sobre el punto objeto de estudio, el alumno, partiendo del ejemplo, lo ampliará inmediatamente. Adicionalmente se propondrán trabajos (programas) individuales y colectivos que serán realizados de forma autónoma aunque con la ayuda del profesor si fuera necesaria. La revisión de esta colección de programas será la que determine la calificación en la asignatura.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
6	-	-	16	3	20	5	

Evaluación

La colección de programas realizados en prácticas, revisados presencialmente (profesor-autor) será la que determine la nota final.

Bibliografía más relevante

- [1] Professional Java Mobile programming / Ronald Ashri
- [2] Professional Symbian programming : mobile solutions on the EPOC platform / Martin Tasker, Jonathan Allin
- [3] Java development on PDAs : building applications for pocketPC and Palm devices / Daryl Wilding-McBride
- [4] Protocolos de pertenencia a grupos para entornos dinámicos / M^a Carmen Bañuls Polo

Instalación y Configuración de Middlewares Grid: GT2 y GT4	Código: ICMGT
	Créditos ECTS: 2
<i>Profesores:</i> Vicente Hernández, Ignacio Blanquer	

Objetivos

Globus Toolkit representa actualmente el estándar de facto para la Computación en Grid, siendo el punto de partida de los más importantes despliegues de Infraestructuras Grid en producción existentes actualmente.

El objetivo de este seminario es aprender a instalar y configurar adecuadamente Globus Toolkit, tanto en su versión pre-WS (GT2), como en la orientada a WSRF (GT4). El objetivo principal consiste en adquirir conocimientos suficientes acerca de los diferentes servicios de los que constan estos middlewares, la interacción entre ellos, su combinación para realizar tareas más sofisticadas como la ejecución remota de tareas o la monitorización de recursos. Así mismo, el alumno aprenderá a identificar los componentes de estos middlewares con la arquitectura Grid.

Pre/Co-requisitos recomendados

Conocimiento de sistemas UNIX/Linux a nivel de usuario y de sistemas operativos.

Asignaturas del máster: Conceptos de la Computación en Grid.

Programa resumido

1. Introducción a Globus Toolkit.
 - a. Servicios Principales
 - b. Método de Instalación
2. Componentes de Gestión de Recursos Locales.
 - a. Gestores de Colas
3. Instalación de Globus Toolkit preWS.
 - a. Configuración de Globus Toolkit preWS
 - i. Seguridad y Certificados de Máquina
 - ii. Servicio GridFTP
 - iii. Servicio preWS GRAM
 - iv. Servicio MDS2 (Monitoring and Discovery Service)
 - b. Pruebas y Validación de la Instalación.
4. Instalación de Globus Toolkit WSRF.
 - a. Configuración de Globus Toolkit WSRF
 - i. Seguridad y Certificados de Máquina y Contenedor
 - ii. Servicios RFT (Reliable File Transfer)
 - iii. Servicio WS-GRAM
 - iv. Servicio WS-MDS (MDS4)
 - b. Pruebas y Validación de la Instalación.

Metodología

La docencia se impartirá directamente en los laboratorios, y cada unidad temática estará precedida de una breve introducción que explique los casos prácticos en los que se enmarca el contenido de la misma.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
6	-	-	12	2	20	10	

Evaluación

La evaluación se realizará atendiendo al trabajo desarrollado en las sesiones de prácticas y evaluando las habilidades y destrezas que el alumno haya adquirido.

Bibliografía más relevante

- [1] Globus alliance Home Page. “Relevant documents”, <http://www.globus.org>
- [2] Gobus Toolkit: <http://www.globus.org/toolkit/>
- [3] Luis Ferreira, Viktors Berstis, Jonathan Armstrong, Mike Kendzierski, Andreas Neukoetter, Masanobu Takagi, Richard Bing-Wo, Adeeb Amir, Ryo Murakawa, Olegario Hernandez, James Magowan, Norbert Bieberstein, “Introduction to Grid Computing with Globus”, IBM Redbooks, ISBN 0738499889

Diseños de Recuperación de Información para Entornos Cooperativos	Código: DRIEC
	Créditos ECTS: 2
<i>Profesores: Vicente Vidal</i>	

Objetivos

- Conocer las técnicas computacionales de altas prestaciones que se utilizan en el diseño de los Sistemas de Recuperación de Información basados en modelos Algebraicos.
- Adquirir habilidades y destrezas en la creación de un sistema de recuperación de información simplificado.
- Utilizar librerías específicas.

Pre/Co-requisitos recomendados

Ninguno.

Programa resumido

1. Modelos Algebraicos de Sistemas de Recuperación de Información.
2. Preprocesado de documentos. Formatos de Almacenamiento.
3. Modelos basados en la Descomposición en Valores singulares.
4. Modelos que utilizan métodos de Clustering.
5. Diseños Paralelos de S.R.I. Benchmarks.

Metodología

La asignatura trata de analizar los aspectos básicos de cómo construir un sistema de recuperación de Información eficiente y preciso con respecto al tiempo computacional y a las soluciones obtenidas. Se pretende que el alumno sea capaz de ir utilizando las herramientas de computación de altas prestaciones más adecuadas en el desarrollo de un sistema de recuperación de información algebraico. Para lograr este propósito utilizaremos la siguiente metodología. Por una parte, se irán introduciendo los conceptos básicos a través de clases teóricas, complementándose con lecturas de artículos, textos, específicos, etc. Por otra parte se irán realizando prácticas de laboratorio donde se emplearán librerías de altas prestaciones para la implementación de las diferentes partes de un sistema de recuperación de información.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
12	-	-	8	6	20	5	

Evaluación

La evaluación de la asignatura estará basada en la presentación oral y por escrito de un trabajo que consistirá en la implementación de parte de un sistema de recuperación simplificado.

Bibliografía más relevante

- [1] Kolda, T.G.: Limited-memory matrix methods with applications. Technical Report CS-TR-3806 (1997)
- [2] Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
- [3] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* 31 (1999) 264-323.
- [4] Berry, M.W., Browne, M.: Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM (1999)
- [5] Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, vector spaces, and information retrieval. *SIAM* 41 (1999) 335-362
- [6] Dhillon, I.S., Fan, J., Guan, Y.: Efficient clustering of very large document collections. In R. Grossman, C. Kamath, V.K., Namburu, R., eds.: *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers (2001) Invited Book Chapter.
- [7] Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: a cluster-based approach to browsing large document collections. In: *SIGIR'92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press (1992) 318-29
- [8] Shankar, S., Karypis, G.: Weight adjustment schemes for a centroid based classifier. Technical report, University of Minnesota, Department of Computer Science (2000)
- [9] Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. *SIAM Rev.* 37 (1995) 573-595
- [10] Karypis, G., Han, E.H.: Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Technical report tr-00-0016, University of Minnesota (2000)
- [11] Letsche, T.A., Berry, M.W.: Large-scale information retrieval with latent semantic indexing. *Information Sciences* 100 (1997) 105-137
- [12] King, O., Kobayashi, M.: Information retrieval and ranking on the web: Benchmarking studies ii. (1999)
- [13] Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Technical report, IBM (2000)
- [14] Modha, D., Spangler, S.: Feature weighting in k-means clustering. *Machine Learning* 52 (2003).

Grid y Computación Paralela	Código: GCP
	Créditos ECTS: 2
<i>Profesores:</i> Vicente Hernández, Ignacio Blanquer	

Objetivos

Las tecnologías Grid y la Computación Paralela son dos tecnologías cercanas y compatibles que permiten explotar diferentes niveles de paralelismo. Sin embargo su integración no es trivial y plantea numerosas dificultades.

El objetivo de este seminario es comprender y conocer los problemas y sinergias de ambas tecnologías y ser capaz de utilizarlas de forma adecuada en la programación paralela. En la parte práctica se ganará experiencia en entornos como MPI-G2, la interacción de MPI y Grids, el balance de carga, la gestión de memoria, y el uso de planificadores.

Pre/Co-requisitos recomendados

Conocimiento de sistemas UNIX e nivel de usuario y de sistemas operativos. Conceptos básicos de la computación paralela. Conocimientos de programación paralela sobre MPI. Conocimiento de programación estructurada en “C” y Java.

Asignaturas del máster: Conceptos de la Computación en Grid.

Programa resumido

1. Particularidades de los Entornos Grid Multicluster.
 - a. Diferencias con Respecto a los Entornos Grid no Paralelos.
 - b. Granularidad y Eficiencia.
2. Ejecución de Trabajos Paralelos sobre Entornos Grid.
 - a. Requerimientos de Configuración.
 - b. Especificaciones de Trabajos Paralelos.
 - c. Casos de Estudio
3. Ejecución de Trabajos MPI entre Nodos.
 - a. Requerimientos de Configuración.
 - b. MPI-G2.
 - c. Casos de Estudio.

Metodología

La docencia se impartirá directamente en los laboratorios, y cada unidad temática estará precedida de una breve introducción que explique los casos prácticos en los que se enmarca el contenido de la misma.

Actividades y dedicación (horas)

Actividades presenciales					A. no presenciales		Otras
Teoría	Seminarios	Problemas	Prácticas	Tutorías	Prácticas	Estudio	(*)
6	-	-	12	2	20	10	

Evaluación

La evaluación se realizará atendiendo al trabajo desarrollado en las sesiones de prácticas y evaluando las habilidades y destrezas que el alumno haya adquirido.

Bibliografía más relevante

- [1] Globus alliance Home Page. “Relevant documents”, <http://www.globus.org>
- [2] Enabling Grids in E-scienceE (EGEE), <http://www.eu-egee.org>