Impacto de la red de interconexión en la planificación de trabajos en Grids

Impact of the Interconnection Network on Grid Computing Scheduling

A. Fuentes, E. Huedo, R. S. Montero e I. M. Llorente

Resumen

Los entornos Grid son por naturaleza altamente dinámicos y heterogéneos, y éste es especialmente el caso del rendimiento de la red de interconexión entre los recursos de un Grid. Por lo tanto, la fase de selección de recursos durante la planificación de trabajos debe considerar la calidad de la red de interconexión para reducir el coste de la transferencia de ficheros. En este artículo revisaremos brevemente los aspectos más relevantes de la computación en Grid y propondremos un algoritmo de planificación que considere dichos aspectos, en concreto la calidad de la red de interconexión que se evalúa dinámicamente. El planificador propuesto selecciona el mejor recurso en el Grid atendiendo a la información sobre el estado de los recursos y la red de interconexión en términos de latencia y ancho de banda.

Palabras clave: Computación en Grid, planificación adaptativa, red de interconexión

Summary

By nature, Grids are highly dynamic and heterogeneous environments, and specially in the case of the interconnection network performance of Grid resources. Therefore, this selection should take into account network interconnection quality between computational resources in order to reduce the cost of file staging. Main aspects of Grid computing are briefly described in this paper and a new job scheduling algorithm that takes into account network interconnection quality dynamically evaluated is proposed. It chooses the better available resource in the Grid considering the information on resource status and network interconnection in terms of bandwidth and latency for job submission.

Keywords: Grid Computing, Adaptive Scheduling, Interconnection Network

La investigación en computación de alto rendimiento ha estado orientada durante las últimas décadas al diseño de arquitecturas avanzadas que permitan resolver los problemas de gran desafío

1.- Introducción

La investigación en computación de alto rendimiento ha estado orientada durante las últimas décadas al diseño de arquitecturas avanzadas que permitan resolver los problemas de gran desafío (intensivos en uso de procesador o en manejo de datos); al desarrollo de sistemas operativos, lenguajes de programación o extensiones de lenguajes existentes que permitan explotar estos sistemas; y a la optimización de aplicaciones y algoritmos para que aprovechen de forma más eficiente estas arquitecturas.

Tradicionalmente la investigación realizada ha seguido siempre un modelo "centralizado" orientado a un único sistema que presta servicios de alto rendimiento. Sin embargo, a mediados de los años 90 comenzaron a popularizarse otras alternativas "distribuidas" qu e consiguen rendimientos comparables a los proporcionados por las arquitecturas más avanzadas a un precio más razonable. Estas tendencias de computación en red (Network Computing) consisten básicamente en interconectar sistemas distribuidos para aprovechar sus recursos, principalmente potencia de cálculo. Ejemplos de estas alternativas son:

- Cluster Computing: Diseño de un cluster de equipos estándar co mo alternativa a la adquisición de un sistema multiprocesador. Su ventaja fundamental es la mejor relación coste/rendimiento y sus inconvenientes, la dificultad de programación y de mantenimiento.
- Intranet Computing: Unión de la potencia computacional desapro vechada en los recursos hardware distribuidos en una red de área local (la mayoría de los PCs y estaciones de trabajo están frecuentemente ociosos). Su principal ventaja es que, para un determinado tipo de aplicaciones, puede proporcionar rendimientos semejantes a los ofrecidos por los sistemas de alto rendimiento con un coste económico casi nulo.

- Internet Computing: Aprovechamiento de la potencia de los recursos distribuidos por Internet siguiendo el modelo cliente/servidor. Actualmente casi todas estas herramientas se limitan a ejecución paramétrica. Su ventaja es el gran rendimiento que se puede obtener y sus inconvenientes se deben al bajo ancho de banda y a la escasa seguridad en Internet.
- Computing Portal: Desarrollo de un portal que proporcione un único punto de acceso seguro basado en tecnología Web para invocar servicios de ejecución de aplicaciones en plataformas de alto rendimiento. Su principal ventaja es la transparencia y facilidad de acceso a plataformas de alto rendimiento.

El uso de las tecnologías descritas anteriormente posibilita el aprovechamiento eficiente de los recursos dentro de una misma organización. Algunas de ellas permiten incluso unir diferentes departamentos u organizaciones pero con la condición de que sea su software el que gestione los recursos internos. Sin embargo, ninguna de estas tecnologías permite unir dominios de administración diferentes manteniendo la política de seguridad de cada centro y las herramientas de planificación ya en uso. Por otro lado, los interfaces y protocolos básicos que utilizan las herramientas anteriores no están basados en estándares abiertos, condición imprescindible para que la tecnología Grid se extienda, y en pocos años sea tan habitual como actualmente es la tecnología Web.

La necesidad de aprovechar los recursos disponibles en los sistemas informáticos conectados a Internet y simplificar su utilización ha dado lugar a una nueva forma de tecnología de la información conocida como Grid Computing. Esta tecnología es análoga a las redes de suministro eléctrico: la idea es ofrecer un único punto de acceso a un conjunto de recursos distribuidos geográficamente (supercomputadores, clusters, almacenamiento, fuentes de información, instrumentos, personal,...). De este modo, los sistemas distribuidos se pueden emplear como un único sistema virtual en aplicaciones intensivas en datos o con gran demanda computacional.

La necesidad de aprovechar los recursos conectados a Internet y simplificar su utilización ha dado lugar a una nueva forma de tecnología de la información conocida como Grid Computing

Desde su creación en el 2001, el software Globus[3] (www.globus.org) se ha convertido paulatinamente en el estándar de facto para la computación distribuida en Grids. Sus componentes ofrecen la infraestructura básica necesaria para el desarrollo y ejecución de aplicaciones distribuidas, en concreto estos servicios son: Grid Security Infrastructure (GSI), Globus Resource Allocation Manager (GRAM), Global Access to Secondary Storage (GASS) y Monitoring and Discovery Service (MDS).

Los componentes anteriores, ya sea de forma independiente o conjunta, facilitan el acceso transparente y seguro a recursos distribuidos geográficamente en múltiples dominios de administración, además de servir como soporte básico para implementar las fases de la planificación de trabajos en Grids, a saber: descubrimiento y selección de recursos; y preparación, envío, monitorización, migración y finalización de trabajos[7]. Sin embargo, la explotación de forma agregada de recursos distribuidos geográficamente está lejos de ser sencilla ya que, a pesar de la relativa madurez de Globus, el usuario es responsable de realizar manualmente las acciones de planificación de trabajos mencionadas anteriormente. Además, Globus no ofrece ningún soporte para la migración de trabajos, y por lo tanto para la ejecución adaptativa necesaria en Grids dinámicos.

El problema más desafiante al que ha de enfrentarse la comunidad científica en computación en Grid, es que estos entornos presentan condiciones altamente variables de forma impredecible, que podemos clasificar del siguiente modo:

- Alta tasa de fallos: Los recursos compartidos en un Grid no pertenecen al mismo dominio de administración, de forma que una vez enviado el trabajo su propietario pierde el control del mismo en favor del administrador del recurso, quien libremente puede cancelar o suspender su ejecución. Además en un Grid los fallos, tanto de recursos como de red, son la regla más que la excepción.
- Disponibilidad, carga y coste dinámico de los recursos: Los recursos de un Grid son explotados por usuarios internos además de por otros usuarios del Grid, de forma que los recursos que

inicialmente se mostraban ociosos pueden saturarse durante la ejecución del trabajo, y viceversa; recursos con una alta carga de trabajo inicial pueden quedar desocupados[2].

Los Grids son por naturaleza entornos altamente dinámicos y heterogéneos, y éste es especialmente el caso del rendimiento de los enlaces de interconexión entre los recursos del Grid[1]. Por tanto, la selección de recursos en el Grid debe tener en cuenta la proximidad de los recursos computacionales a los datos necesarios para reducir el coste de transferencia de los ficheros. Este hecho es especialmente importante en el caso de ejecución adaptativa de trabajos, ya que la migración requiere la transferencia de grandes ficheros de reinicio entre los recursos de ejecución.

Así, nuestro objetivo en este artículo es demostrar que, al igual que en sistemas multiprocesador, la red de interconexión en un entorno Grid (Internet) es un factor clave a la hora de obtener ejecuciones eficientes en este nuevo entorno de computación. Con el fin de conseguir este objetivo a continuación haremos una introducción a la arquitectura GridWay[5 y 6], un software de planificación para Grids que usaremos como base en nuestro banco de pruebas. Incorporaremos a GridWay, sección 3, un nuevo algoritmo de selección de recursos que considera dinámicamente el ancho de banda y la latencia entre los diferentes recursos en el Grid. En la sección 4 se describe el banco de pruebas construido usando los servicios ofrecidos por la infraestructura IRISGrid. Los beneficios de usar esta nueva política de selección se muestran en la sección 5. Finalmente, describiremos el trabajo futuro y proporcionaremos algunas conclusiones sobre la estrategia presentada en este artículo.

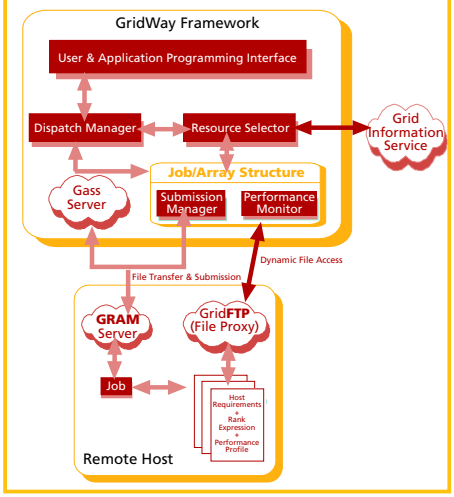
Los Grids son por naturaleza entornos altamente dinámicos y heterogéneos

2.- El Prototipo GridWay

El núcleo de la herramienta GridWay es un agente de envío que realiza automáticamente todas las fases de la planificación de un trabajo y vela por que su ejecución sea correcta y eficiente:

- La ejecución adaptativa del trabajo se realiza mediante una pl anificación dinámica. Una vez que el trabajo es asignado inicialmente a un recurso, se re-planifica periódicamente para descubrir recursos más idóneos, cuándo se detecta un deterioro en su rendimiento o se produce un fallo.
- El rendimiento real de la aplicación se evalúa periódicamente comprobando el tiempo de suspensión acumulado y mediante un programa externo (performance evaluator) que examina un perfil de rendimiento generado por la aplicación.
- La selección y descubrimiento de recursos se realiza mediante otro programa (resource selector) que construye una lista de recursos factibles, según los requisitos del trabajo, ordenados atendiendo a las preferencias de la aplicación.

FIG. 1: ARQUITECTURA DEL PROTOTIPO GRIDWAY



La arquitectura del agente de envío se muestra en la Figura 1. El usuario interactúa con la herramienta a través de un interfaz de usuario, que maneja sus peticiones (submit, kill, stop, resume...) y las reenvía al dispatch manager. El dispatch manager se despierta periódicamente en cada intervalo de planificación e intenta enviar los trabajos pendientes a recursos del Grid y es también responsable de decidir si la migración de los

trabajos re-planificados es factible y merece la pena. Una vez que un trabajo es asignado a un recurso, se arranca un submission manager y un performance monitor para vigilar su ejecución correcta y eficiente.

La flexibilidad de este entorno se garantiza por un interfaz de programación (Application Program Interface, API) bien definido para cada componente del agente de envío y por un diseño modular que permite la extensión y mejora de sus funcionalidades de forma sencilla.

3.- Algoritmo de selección de recursos considerando la red

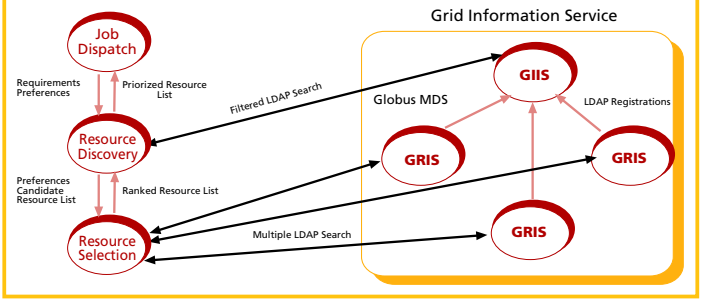
Debido a la naturaleza dinámica y heterogénea del Grid, los usuarios finales establecen los requisitos que deben satisfacer los recursos candidatos y su criterio de clasificación. Los atributos necesarios en este proceso se obtienen del servicio de información del Grid, Globus Monitoring and Discovery Service (MDS). Actualmente el descubrimiento de recursos se basa en información estática (sistema operativo, arquitectura,...) recogida del Grid Information Index Service (GIIS), mientras que su selección lo hace en atributos dinámicos (memoria libre, carga,...) obtenidos del Grid Resource Information Service (GRIS).

Tanto el ancho de banda como la latencia entre los recursos se consideran en el proceso de selección

Tanto el ancho de banda como la latencia entre los recursos se consideran en el proceso de selección. Hay varias estrategias para obtener los atributos de rendimiento de la red. En nuestras pruebas, el MDS se ha configurado para proporcionar tales parámetros accediendo a la información proporcionada por las herramientas Network Weather Service (NWS) [8] e Iperf [4].

El esquema de descubrimiento y selección de recursos de la arquitectura GridWay se muestra en la Figura 2. Inicialmente, los recursos de computación disponibles se descubren accediendo al servidor GIIS, y aquellos recursos que no cumplen las precondiciones mínimas son automáticamente descartados. En este paso, se realiza una prueba de autorización en cada recurso de computación descubierto para garantizar que el usuario tiene acceso a éste. Entonces se obtienen los atributos dinámicos de cada elemento de computación y los parámetros dinámicos de la red de interconexión accediendo a su GRIS local.

FIG. 2: DESCUBRIMIENTO Y SELECCION DE RECURSOS DEL GRIDWAY



La herramienta GridWay se ha extendido para considerar también la proximidad dinámica de los recursos en el proceso de selección. El modelo de rendimiento usado en el nuevo proceso considera tanto el rendimiento dinámico como la proximidad dinámica de los recursos computacionales. En concreto se consideran los siguientes términos en la etapa de selección de recursos:

- El rendimiento computacional del recurso candidato para reducir el tiempo de ejecución.
- La proximidad entre el recurso candidato y el cliente, para reducir el tiempo de envío y monitorización del trabajo, y de transferencia de ficheros.

Con el fin de reflejar todas las circunstancias descritas previamente, cada recurso candidato h_n se clasifica usando el tiempo estimado de ejecución total (cuanto menos mejor) cuando el trabajo se lanza a un recurso en un instante dado t_n. Por tanto, podemos asumir que el tiempo total de ejecución puede separarse en:

$$T_{exe} = T_{cpu}(h_n, t_n) + T_{xfr}(h_n, t_n)$$

donde $T_{cpu}(h_n, t_n)$ es el tiempo computacional estimado, considerando el rendimiento del recurso y la carga computacional de la aplicación, y $T_{xfr}(h_n, t_n)$ es el tiempo de transferencia estimado, considerando el ancho de banda, la latencia y el tamaño de los ficheros.

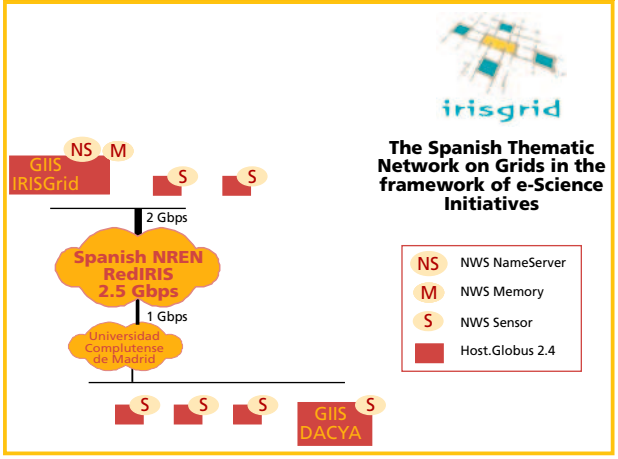
4.- Banco de pruebas de investigación

La arquitectura del banco de pruebas engloba los recursos computacionales distribuidos en dos organizaciones virtuales

Con el fin de analizar el comportamiento del algoritmo de selección de recursos previamente descrito, hemos construido un entorno de pruebas usando los servicios ofrecidos por la infraestructura de IRISGrid: la red temática nacional sobre Grid en el marco de las iniciativas de e-Ciencia.

La arquitectura del banco de pruebas mostrada en la figura 3, engloba los recursos computacionales distribuidos en dos organizaciones virtuales, RedIRIS y el Dpto. de Arquitectura de Computadores y Automática de la UCM, ambos conectados a la Red Nacional de I+D+I española, RedIRIS. Las máquinas de cada organización están conectadas por una red local de 100 Mbps, con una línea de comunicaciones entre ambas organizaciones de un 1 Gbps no dedicado.

FIG. 3: RED DE INTERCONEXION DE LOS RECURSOS GRID DEL TESTBED



Todos los recursos se han configurado para registrarse en el servidor GIIS de IRISGrid. Además, el servicio MDS se ha configurado para proporcionar el ancho de banda dinámico y la latencia entre dichos recursos, usando NWS e Iperf, descritos en los puntos 4.1 y 4.2 respectivamente. Las características de las máquinas se describen en la Tabla 1.

4.1.- Network Weather Service (NWS)

Network Weather Service proporciona una estimación precisa de los cambios dinámicos de rendimiento entre diferentes recursos computacionales y realiza estimaciones de parámetros de red y computacionales. Esta herramienta se basa en cuatro procesos para obtener y ofrecer la información:

1.- Persistent State: almacena la información recogida de los recursos computacionales.
2.- Name Server: implementa el soporte para el resto de procesos.
3.- Sensor: recoge las medidas de rendimiento y distribuye la información.
4.- Forecast: se encarga de calcular los valores esperables.

En nuestro entorno de pruebas, todas las máquinas cuentan con un proceso sensor, mientras que los procesos Persistent State y Name Server se sitúan en aristoteles.

RECURSO	MODELO	VELOCIDAD	S.O.	MEMORIA	DOMINIO
aquila	Pentium III	700 Mhz	Linux 2.4	128 MB	dacya.ucm.es
cygnus	Pentium IV	2.5 Ghz	Linux 2.4	512MB	dacya.ucm.es
hydrus	Pentium IV	2.5 Ghz	Linux 2.4	512MB	dacya.ucm.es
cepheus	Pentium III	600 Mhz	Linux 2.4	256MB	dacya.ucm.es
aristoteles	Pentium III	1.4 Ghz	Linux 2.4	1GB	rediris.es
platon	Pentium III	1.4 Ghz	Linux 2.4	1GB	rediris.es
heraclito	Celeron	700 Mhz	Linux 2.4	256MB	rediris.es

Tabla 1: Recursos computacionales usados en el banco de pruebas

4.2.- Iperf

Es una herramienta para medir el ancho de banda TCP. Es una aplicación peer-to-peer (P2P) que permite conocer el ancho de banda existente entre dos recursos computacionales de forma bidireccional. Se usa ampliamente para realizar pruebas de rendimiento de las pilas TCP y UDP.

Iperf es una herramienta para medir el ancho de banda TCP

5.- Experimentos

El comportamiento de la estrategia de selección de recursos previamente descrita se muestra con la ejecución de varias cargas artificiales de uso intensivo de CPU. En particular, estas cargas satisfacen la siguiente relación:

$$W_i = T_{xfr}(h_{min}) \mid T_{cpu}(h_{min}) = 0.01$$

donde $T_{xfr}(h_{min})$ es el tiempo de transferencia al recurso computacional más cercano y $T_{cpu}(h_{min})$ es el tiempo de computación sobre el recurso más rápido.

FIG. 4: TIEMPO DE EJECUCION SIN CONSIDERAR EL ANCHO DE BANDA

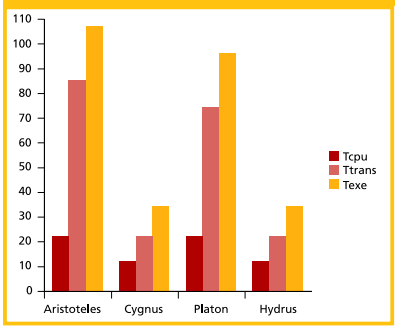
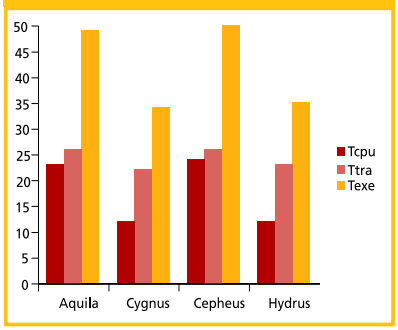


FIG. 5: TIEMPO DE EJECUCION CONSIDERANDO EL ANCHO DE BANDA



En el siguiente experimento, la máquina cygnus almacena el ejecutable y los ficheros de entrada, y recibe los ficheros de salida generados como resultado de la ejecución distribuida. Un dato importante es la limitación impuesta entre los recursos de ambas organizaciones a 10 Mbps. Si observamos las Figuras 4 y 5, la eficiencia en la ejecución de los trabajos es mayor cuando el algoritmo de planificación considera el ancho de banda entre los recursos. En este caso, el algoritmo de planificación decide asignar trabajos a recursos más cercanos aunque sus prestaciones computacionales sean menores y, como consecuencia, el tiempo total de ejecución se reduce en un 54%.

6.- Conclusiones y trabajo futuro

En este trabajo, nuestro objetivo ha sido ofrecer una breve introducción a la computación en Grid, denotando la importancia de la red de interconexión entre los recursos computacionales involucrados. La calidad de la red de interconexión supone un factor relevante en el proceso de planificación de trabajos en el Grid. Por tanto, considerar las capacidades de la red, en términos de ancho de banda y latencia, es tan importante como considerar las características computacionales de los recursos.

Es importante resaltar que la naturaleza descentralizada y modular de la arquitectura GridWay garantiza la escalabilidad de la estrategia de planificación propuesta. Actualmente, estamos aplicando estas ideas en el desarrollo de un agente de recursos de red que integre reserva dinámica de ancho de banda y asignación de prioridades al tráfico, dependiendo de los requisitos dinámicos de los trabajos pendientes de ejecución.

La calidad de la red de interconexión supone un factor relevante en el proceso de planificación de trabajos en el Grid

Agradecimientos

Esta investigación ha sido financiada por el MCyT, mediante el proyecto TIC2003-01321 y la acción especial TIC2002-11109-E, y por el Instituto Nacional de Técnica Aeroespacial "Esteban Terradas".

Referencias

[1] W. Allcock, A. Chervenak, I. Foster, L. Pearlman, V. Welch y M. Wilde. *Globus Toolkit Support for Distributed Data-Intensive Science*. En Proceedings of Computing in High Energy Physics, 2001.

[2] R. Buyya, D.Abramson y J. Giddy. *A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker*. Future Generation Computer Systems, 2002.

[3] I. Foster y C. Kesselman. Globus: *A Metacomputing Infrastructure Toolkit*. Intl. J. Supercomputer Applications, 11(2): 115-128, 1997

[4] Chung-Hsing Hsu y Ulrich Kremer. IPERF: *A Framework for Automatic Construction of Performance Prediction Models*. En Workshop profile and Feedback-directed compilation (PFDC), 1998.

[5] E. Huedo, R. S. Montero e I. M. Llorente. *An Experimental Framework For Executing Applications in Dynamic Grid Environments*. Technical Report 2002-43,ICASE NASA Langley, 2002.

[6] E. Huedo, R. S. Montero e I. M. Llorente. *A Framework for Adaptive Execution on Grids*. International Journal of Software: Practice and Experience, 2004. In press.

[7] J.M.Schopf. *Ten Actions when Superscheduling*. Technical Report GFD-I.4, Scheduling Working Group. The Global Grid Forum, 2001.

[8] R. Wolski, N. Spring y J. Hayes. *The Network Weather Service: A Distributed Resource Performance Forecasting Service for MetaComputing*. Future Generation Computing Systems, 15,1999.

Antonio Fuentes
(antonio.fuentes@rediris.es)
Red. es / RedIRIS
Eduardo Huedo
(huedoce@inta.es)
Laboratorio de Computación Avanzada,
Centro de Astrobiología (CSIC-INTA),
Rubén S. Montero, Ignacio M. Llorente
(rubensm@dacya.ucm.es), (llorente@dacya.ucm.es)
Dpto. Arquitect. de Computadores y Automática,
UCM