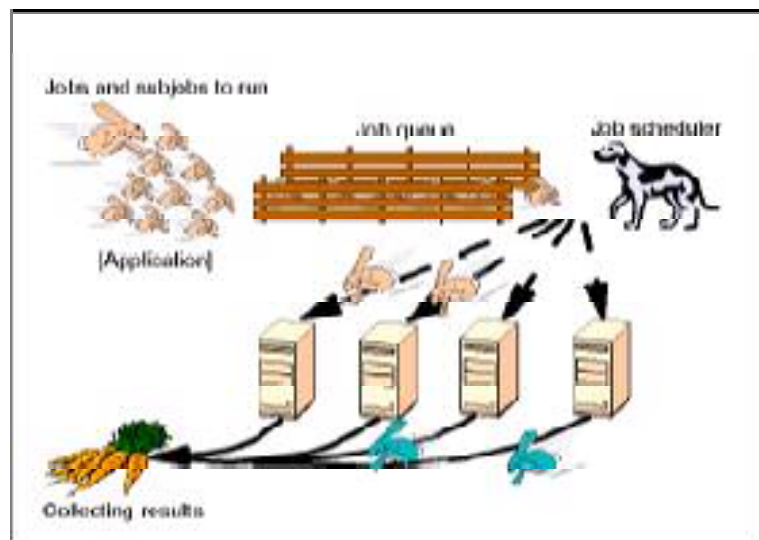




Universidad Nacional del Nordeste  
Facultad de Ciencias Exactas, Naturales y Agrimensura

Monografía de Adscripción a Sistemas Operativos

## Grid Computing



Verónica Vanessa Barrios

Licenciatura en Sistemas de Información  
Corrientes - Argentina

2005



# Índice General

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Concepto de Grid Computing . . . . .	1
1.1.1	Beneficios que Ofrece el Grid Computing . . . . .	1
1.2	Introducción a Grid Computing . . . . .	2
1.3	¿Qué es y Para Qué Sirve el Grid Computing? . . . . .	4
1.4	¿Qué es GLOBUS? . . . . .	7
1.5	Arquitectura del Grid . . . . .	8
1.6	Aplicaciones y Servicios en Grid . . . . .	11
1.6.1	Supercomputación . . . . .	11
1.6.2	Proceso Intensivo de Datos . . . . .	12
<b>2</b>	<b>Principios de Grid Computing</b>	<b>15</b>
2.1	Lo Que el Grid Computing Puede Hacer . . . . .	15
2.2	Aprovechar los Recursos Que No Siempre se Usan . . . . .	15
2.3	La Capacidad de CPU Paralela . . . . .	17
2.4	Las Aplicaciones . . . . .	18
2.5	Los Recursos y Organizaciones Virtuales . . . . .	19
2.6	El Acceso a los Recursos Adicionales . . . . .	20
2.7	Balanceo de Recursos . . . . .	21
2.8	Confiabilidad . . . . .	22
2.9	Administración . . . . .	23
2.10	Los Conceptos y Componentes del Grid . . . . .	24
2.10.1	Los Tipos de Recursos . . . . .	25
2.10.2	Computación . . . . .	25
2.10.3	Almacenamiento . . . . .	26
2.10.4	Las Comunicaciones . . . . .	28
2.10.5	El Software y las Licencias . . . . .	28
2.10.6	El Equipo Especial, Capacidades, Arquitecturas, y Políticas . . . . .	29

2.10.7	Los Trabajos y las Aplicaciones . . . . .	30
2.11	Scheduling, Reservación, y Barrido . . . . .	31
2.12	Intragrid a Intergrid . . . . .	32
2.13	Construcción del Grid . . . . .	35
2.14	Planificación del despliegue . . . . .	35
2.14.1	Seguridad . . . . .	35
2.14.2	Organización . . . . .	36
2.15	Componentes del Software Grid . . . . .	36
2.15.1	Componentes de administración: . . . . .	37
2.15.2	Software Servidor . . . . .	37
2.15.3	Software de Sumisión . . . . .	39
2.15.4	Administración del Grid Distribuido . . . . .	39
2.15.5	Schedulers . . . . .	40
2.15.6	Las Comunicaciones . . . . .	41
2.16	Observación, Dirección, y Medición . . . . .	41
2.17	Usar un Grid: Perspectivas de Usuario . . . . .	42
2.17.1	Conectar e instalar el software de Grid . . . . .	42
2.18	Registrarse en el Grid . . . . .	43
2.19	Solicitar y Realizar Trabajos. . . . .	44
2.20	Configuración de Datos . . . . .	46
2.21	Monitoreo del Progreso y Recuperación . . . . .	47
2.22	Reservar recursos . . . . .	48
2.23	Usar un Grid: La Perspectiva de un Administrador . . . . .	49
2.23.1	Planeación . . . . .	49
2.23.2	Instalación . . . . .	49
2.24	Matriculación de Dirección . . . . .	50
2.25	Certificado de Autoridad . . . . .	51
2.26	Administración de Recursos . . . . .	53
2.27	Compartir Datos . . . . .	54
2.28	Usar un Grid:Una Perspectiva del Diseñador de la Aplicación . . . . .	54
2.29	El Presente y el Futuro del Grid . . . . .	55
2.30	¿Qué No Puede Hacer el Grid Computing? . . . . .	56
<b>3</b>	<b>Estándares Abiertos</b> . . . . .	<b>57</b>
3.1	Web Service: Servicios Web . . . . .	57
3.2	Grid Service: Servicios Grid . . . . .	58
3.3	Open Grid Service Architecture . . . . .	59
3.4	Open Grid Services Infrastructure . . . . .	59
3.5	¿Cuáles son los objetivos de OGSA? . . . . .	60
3.5.1	Arquitectura: . . . . .	61

<i>ÍNDICE GENERAL</i>	v
3.6 ¿Qué plataformas? . . . . .	67
3.7 Conclusión. . . . .	73
<b>Bibliografía</b>	<b>75</b>
<b>Índice de Materias</b>	<b>77</b>



# Índice de Figuras

1.1	Origen y Desarrollo del Grid Computing. . . . .	3
1.2	Aplicaciones de la Computación en Grid. . . . .	5
1.3	Características de un Grid Computacional. . . . .	7
1.4	Arquitectura del Grid. . . . .	10
1.5	Arquitectura Actual de los Sistemas de Información. . . . .	13
1.6	Arquitectura “Grid Computing” para Sistemas de Información. . . . .	13
2.1	Los Recursos Virtuales y Organizaciones Virtuales Para Colaboración. . . . .	20
2.2	Balaceo de Recursos. . . . .	22
2.3	Cofiabilidad en los Sistemas de Grid. . . . .	23
2.4	Administración en los Sistemas de Grid. . . . .	24
2.5	Almacenamiento en el Grid. . . . .	27
2.6	Trabajos y Aplicaciones del Grid. . . . .	30
2.7	Un Grid más Simple. . . . .	34
2.8	Un Intergrid más Complejo. . . . .	34
3.1	La Arquitectura de OGSA. . . . .	61
3.2	La Estructura de OGSA. . . . .	63
3.3	Componentes de OGS. . . . .	64
3.4	El OGS y Web service. . . . .	66
3.5	El OGS y el Hosting de Web service. . . . .	66
3.6	La Estructura de la Arquitectura de servicio de OGSA. . . . .	68
3.7	Servicio de núcleo de Grid. . . . .	69
3.8	Ejecución de Programas de Grid y Data Service. . . . .	71
3.9	La Ejecución de Programas de Grid y Data Service Hosting. . . . .	72



# Capítulo 1

## Introducción

### 1.1 Concepto de Grid Computing

El concepto principal de Grid Computing es el de compartir potencia computacional [13].

Desde el momento en el que los primeros ordenadores comenzaron a conectarse a Internet, surgió la idea de unir la potencia inutilizada de cada uno para abordar problemas a los que sólo podían enfrentarse las supercomputadoras pertenecientes a organizaciones gubernamentales, universidades o grandes multinacionales.

La tecnología que hace esto posible se llama Grid Computing (malla de ordenadores) y se basa en el aprovechamiento de los ciclos de procesamiento no utilizados por los millones de ordenadores conectados a la Red. De esta forma se consigue que puedan resolver tareas que son demasiado intensivas para ser resueltas por una única máquina.

#### 1.1.1 Beneficios que Ofrece el Grid Computing

Los principales beneficios son:

- Ofrecer flexibilidad para llenar las necesidades cambiantes del negocio.
- Brindar alta calidad a menor costo.

- Facilitar el pronto retorno de las inversiones.
- No necesitar de toda una nueva infraestructura para que funcione.
- Facilitar poder de computación / precio muy barato.
- Brindar el poder de un supercomputador.
- Utilizar software gratuito y usar código fuente abierto.
- No precisar hardware adicional, para posibilitar el incremento de la potencia de cómputo.
- Brindar transparencia para el usuario que participa en el Grid.

## 1.2 Introducción a Grid Computing

La idea del Grid está enfocada fundamentalmente en el acceso remoto a recursos computacionales y pretende ser un paradigma de desarrollo no centrado en una tecnología concreta [9].

La evolución de grid computing se refleja en el avance de la estandarización de esta tecnología (el estándar de Globus Project es el estándar de facto) donde se encuentra definida la arquitectura del grid, los niveles de acceso, los requisitos, los servicios, etc.

El grid computing se enmarca dentro de la tecnología de computación distribuida englobando conceptos como sistemas operativos distribuidos, programación multiprocesador, redes de ordenadores, computación paralela, redes de computadoras, seguridad, bases de datos, etc. De alguna manera el concepto de grid computing da una unidad conceptual a estos problemas de manera que todos ellos puedan verse desde una perspectiva grid.

El Grid Computing es más que una idea ambiciosa, ya que no sólo se trata de compartir ciclos de CPU para realizar cálculos complejos sino que se busca la creación de una infraestructura distribuida. Esta ardua tarea involucra labores de definición de la arquitectura general, de interconexión de diferentes redes, de definición de estándares, de desarrollo de procedimientos para la construcción de aplicaciones, etc. [11].

El Grid es una idea que promete revolucionar el mundo de la computación y el cómo se desarrollan las aplicaciones actualmente [13].

Parece claro que Grid Computing no es una moda, y queda, todavía, hay un largo camino por recorrer desde los conceptos hasta las aplicaciones reales. Aunque existen muchos mini-grids para el desarrollo de investigaciones no parece cercano el día en que todos los ordenadores del mundo formen un Grid Mundial a modo de gigantesco sistema de distribución eléctrico (paradigma del grid computing en casi todas las publicaciones sobre el tema), donde los usuarios se conecten y tengan acceso a la capacidad de cómputo y de almacenamiento que precisen sin preocuparse de donde se genera (ver fig. 1.1 de la pág. 3) [13].



Figura 1.1: Origen y Desarrollo del Grid Computing.

### 1.3 ¿Qué es y Para Qué Sirve el Grid Computing?

El concepto de Grid Computing da idea de una gran potencia de cálculo y almacenamiento, y parece un gran avance en las ciencias de la computación.

Existen además una multitud de aplicaciones reales que hacen uso de mini-grids, casi todas ellas están centradas en el campo de la investigación en el terreno de las ciencias físicas, médicas y del tratamiento de la información.

Se ha hablado mucho de lo que ofrece y de las posibilidades del grid computing pero todavía no se ha definido de manera concreta el término.

Un grid computacional es una infraestructura hardware y software que suministra al que la utiliza acceso seguro (dependable), consistente (consistent), penetrante (pervasive) y barato (inexpensive), a unas elevadas capacidades computacionales.

El concepto infraestructura se utiliza porque un grid es un conjunto de recursos (ciclos de CPU, datos, sensores, etc.), y todos esos recursos necesitan una interconexión hardware y un control software para que estén ensambladas en un grid. Esta infraestructura debe proporcionar a los usuarios un servicio seguro a todos los niveles: capacidad de cómputo, de integridad de datos, de seguridad de acceso, etc.

El servicio debe ser consistente, basado en estándares, y de esta manera el acceso y las operaciones sobre el grid estarán definidas por dichos estándares evitando la heterogeneidad. La idea de penetración no es tanto la posibilidad de acceder a cualquier recurso del grid como que el grid llega a cualquier sitio, de esta manera se asegura que una vez conectado desde cualquier punto puede extraer del grid toda la potencia que requiera. Por último el acceso y uso del grid debe tener un coste económico que le haga atractivo para que su utilización se universalice.

Para terminar de definir el concepto se desgana cada uno de los posibles campos de aplicación de la tecnología grid. (ver fig. 1.2 de la pág. 5).

El análisis lleva a definir cinco grandes áreas de trabajo determinadas por las necesidades de cálculo, espacio para el almacenamiento de los datos y tiempo de respuesta. Las áreas son:

#### **Supercomputación distribuida**



Figura 1.2: Aplicaciones de la Computación en Grid.

Dentro de esta área se encuentran aquellas aplicaciones cuyas necesidades es imposible satisfacer en un único nodo. Estas necesidades se producen en instantes de tiempo determinados y consumen muchos recursos, por lo que se dice que son puntuales e intensivas. Ejemplo de este tipo de aplicaciones son las simulaciones, las herramientas de cálculo numérico, los procesos de análisis de datos, la extracción de conocimiento de almacenes de datos, etc.

### **Sistemas distribuidos en tiempo real**

En este tipo de aplicaciones se consideran aquellas que generan un flujo de datos a alta velocidad que debe ser analizado y procesado en tiempo real. Ejemplo de este tipo de aplicaciones son los experimentos de física de alta energía, control remoto de equipos médicos de alta precisión y precio, todos los procesos de la denominada e-Medicine, el tratamiento de imágenes para la visión artificial, etc.

### **Proceso intensivo de datos**

Esta área se centra en aquellas aplicaciones que hacen un uso intensivo del espacio de almacenamiento. Las necesidades de almacenamiento de este tipo de aplicaciones desbordan la capacidad de almacenamiento de un único nodo y los datos son distribuidos por todo el grid. Además de los beneficios por el incremento de espacio, la distribución de los datos a lo largo del grid permite el acceso a los mismos de forma distribuida. Ejemplos de este tipo de aplicaciones son todos los sistemas gestores de bases de datos distribuidas.

### **Servicios puntuales**

En esta área, se olvida el concepto de potencia de cálculo y capacidad de almacenamiento, para centrarse en recursos que una organización puede considerar como no necesarios. De esta manera el grid ofrece a la organización esos recursos sin que la organización deba desarrollarlos por si misma. Ejemplos de este tipo de aplicaciones son aquellas que permiten acceder a hardware muy específico (equipos costosos de medida o de análisis de muestras) para la realización de labores a distancia.

### **Entornos virtuales de colaboración**

Esta área está relacionada directamente con el concepto de Teleinmersión, de manera que se utilizan los enormes recursos computacionales del grid y su naturaleza distribuida para generar entornos virtuales 3D distribuidos (ver fig. 1.3 de la pág. 7).

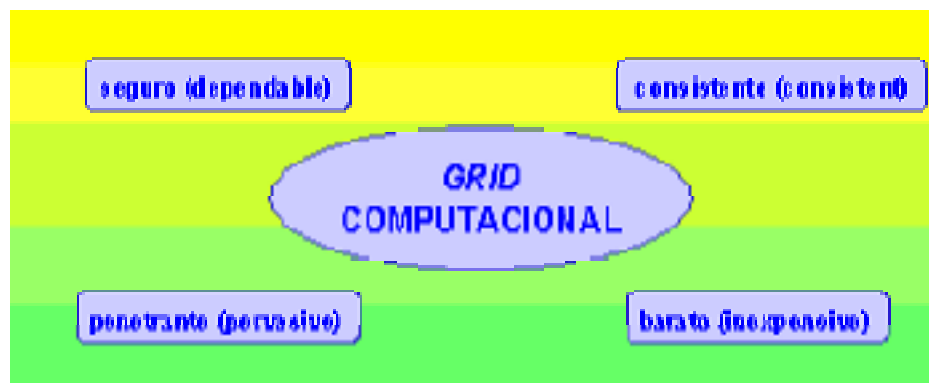


Figura 1.3: Características de un Grid Computacional.

## 1.4 ¿Qué es GLOBUS?

Globus es un proyecto de investigación y desarrollo enfocado a permitir la aplicación de los conceptos de Grid al campo científico.

El proyecto Globus desarrolla software capaz de resolver las dificultades técnicas que aparecen al tratar de implementar Grid Computing.

Por ejemplo administracion de recursos y de datos, recursos de informacion, seguridad o desarrollo de aplicaciones. Este esfuerzo a nivel de software ha dado como resultado el Globus Toolkit, un conjunto de servicios y librerías de software capaz de soportar aplicaciones tipo Grid. El Toolkit incluye software relacionado con seguridad, informacion y manejo de recursos e intercambio de datos.

El software de Globus es libre y está soportado por los sistemas operativos Linux, Solaris, IRIX, AIX, HPUX, True64,...

**La arquitectura del Globus Toolkit v3 se estructura en varias capas.**

La capa inferior es el núcleo donde se encuentran las factorías de recursos, el servicio de notificaciones, el servicio de persistencia y el servicio de ciclo de vida. En la segunda capa se encuentran los servicios de seguridad (GSI Grid Security Infraestructure). En la tercera capa se encuentran los servicios básicos como la gestión de trabajos, los servicios de directorio y monitorización y los

de transferencia de ficheros. En el nivel más alto se encuentran los servicios de gestión de grandes cantidades de datos y aquellos servicios que no son del GT3 pero que se basan en esta arquitectura.

En esta versión se introduce el concepto de grid service que es una ampliación de los Web Services .La idea es buscar una tecnología de objetos distribuidos que se adaptase a las necesidades de una aplicación grid, y para ello se utilizaron los Web Services, aunque estos presentan algunas limitaciones que se superaron:

- Los Web Services no mantienen el estado de una invocación a otra, los grid services si.
- Los Web services no son transientes, es decir no se pueden crear varias instancias de un mismo servicio según se necesita y destruirlas cuando ya no son necesarias, en los grid services sí se puede.
- Los Web Services no incluyen servicios de apoyo que han sido incluidos en los grid services como son las notificaciones, el servicio de persistencia, la gestión del ciclo de vida, etc.

Los grid services utilizan un enfoque de Factorías de Objetos de manera que en lugar de tener un único servicio compartido por todos los usuarios (como el Web Service) se tiene un servicio-factoría que crea instancias individuales del servicio. Cuando se invoca a una operación del servicio se accede a la instancia y no a la factoría. Además crea una instancia por cliente, o varias por cliente o una para varios clientes. Por último la destrucción de la instancia puede correr a cargo del cliente o de la factoría.

El Globus Project (el proyecto iniciador) es el que ha proporcionado una implementación estándar que es utilizada por una multitud de empresas; permitiendo que los sistemas que hacen uso del grid se integren como sustento de aplicaciones empresariales.

## 1.5 Arquitectura del Grid

Es una nueva arquitectura de computación designada para direccionar las necesidades de utilidad de computación, lo que significa que la arquitectura formada por Grid permite utilizar los recursos de procesamiento propios de

la compañía sin tener que invertir más en adquirir capacidad extra de procesamiento. Es como colocar los procesadores existentes en la empresa en el “centro” y enviarles a ellos los datos para procesar, esto permite que los recursos se utilicen a medida que se necesiten.

Esta arquitectura se ha utilizado para resolver grandes problemas de cómputo, como procesar datos de investigaciones, simulaciones científicas, procesamientos de datos estadísticos y simulaciones técnicas, a diferencia del Grid empresarial donde se ejecutan aplicaciones reales en tiempo real y se optimizan recursos informáticos (es hacia donde está apuntando Oracle con Grid Computing).

Las aplicaciones, los toolkits, las APIs, los SDK, etc. que se definen para la computación en Grid deben de cumplir una arquitectura general. Esta arquitectura general se articula en cinco niveles: la infraestructura, la conectividad, la gestión del recurso, la gestión de varios recursos y el nivel de aplicación, según se muestra en la fig. 1.4 de la pág. 10.

Principalmente, la arquitectura propuesta es una arquitectura de protocolos que definen los mecanismos básicos que permiten a los usuarios y a los recursos negociar, establecer, gestionar y explotar la compartición de recursos. Una arquitectura abierta basada en un estándar facilita la extensibilidad, la interoperabilidad, la portabilidad y la compartición de código.

De esta manera la estandarización de los protocolos permitira estandarizar los servicios y mejorar las capacidades del grid.

En el nivel de infraestructura se encuentran los recursos computacionales, como son los ordenadores, los clusters, los supercomputadores, los sistemas de almacenamiento en red, las bases de datos, etc. También se incluyen en este nivel la infraestructura de la red y sus mecanismos de gestión y control. En la terminología Grid la infraestructura se denomina la Fábrica y suministra los componentes que serán compartidos.

El nivel de conectividad incluye los protocolos de comunicación y seguridad que permiten a los recursos computacionales comunicarse. Entre estos protocolos se encuentran: la pila de protocolos TCP/IP, el protocolo SSL, Certificados X.509. También se incluyen los nuevos protocolos que se encuentran en fase de estudio y que permitirán mejorar el rendimiento en las redes de alta velocidad. La seguridad es un punto muy importante de la computación en grid por su propia naturaleza distribuida ya que se comparten recursos entre distintas organizaciones que pueden tener distintas políticas de seguridad.

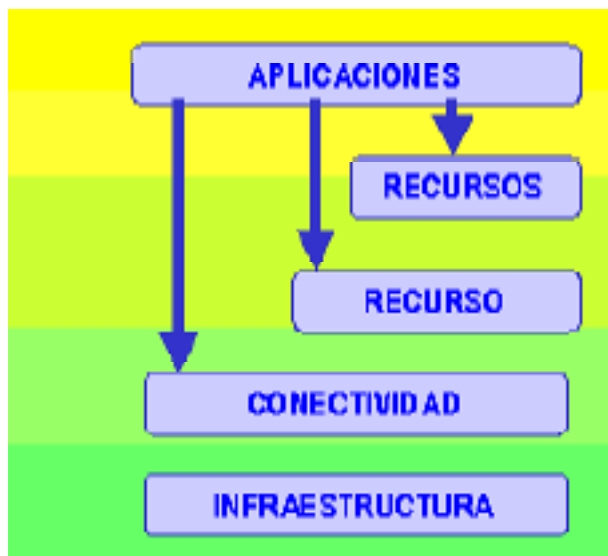


Figura 1.4: Arquitectura del Grid.

El nivel de recurso se centra en la gestión de un único recurso y permite tener información y control sobre el mismo. En este nivel se encuentran los protocolos que permiten obtener la información de un recurso: las características técnicas, la carga actual, el precio, etc. También se encuentran los protocolos que permiten el control del recurso: el acceso al mismo, el arranque de procesos, la gestión, la parada, la monitorización, la contabilidad de uso y la auditoría del recurso.

La capa de recursos engloba todos los servicios que permiten gestionar un conjunto de recursos. Se encuentran los servicios de directorio, que permiten localizar los recursos que son de nuestro interés; los schedulers distribuidos, que permiten asignar las tareas a cada recurso; la monitorización y diagnóstico de la ejecución de las distintas tareas en que se distribuyen la ejecución de una aplicación; la contabilidad, que permite calcular el coste de la utilización de varios recursos heterogéneos, y, el acceso a datos distribuidos, que gestiona la replicación de datos.

El servicio de scheduler distribuido es una de las aplicaciones más complejas de un desarrollo grid ya que existen tres scheduler distintos: el planificador de trabajos (job scheduler) que intenta maximizar la cantidad de trabajo realizado (trabajos por unidad de tiempo), el planificador de recursos que intenta

maximizar el uso de los recursos, y, el planificador de la aplicación que divide la aplicación en tareas, asigna los recursos para su ejecución y vigila el desarrollo de los mismos.

Los dos primeros priman la eficiencia del sistema grid, mientras que el tercero prima la eficiencia de la aplicación.

El último nivel, el de aplicación, se centra en la definición de protocolos que permitan a las aplicaciones el acceso a la infraestructura del grid a través de las distintas capas. Según el tipo de aplicación puede ser necesario conectarse a las distintas capas o acceder directamente a una de ellas, incluso directamente a la infraestructura.

## 1.6 Aplicaciones y Servicios en Grid

El avance del grid computing se observa en la aplicabilidad de sus ideas en productos comerciales y los desarrollos de compañías de software basados en el estándar de grid.

### 1.6.1 Supercomputación

Uno de los campos donde se han centrado más los esfuerzos de compañías comerciales es en el de la supercomputación. Empresas con un amplio abanico de productos, ofrecen desarrollos para la instalación de grids con altas capacidades de cómputo.

Por ejemplo, HP conectó al grid del DOE (DOE Science Grid) una máquina Linux de 9,2 teraflops . Este grid se usa para llevar a cabo simulaciones, analizar datos y coordinar experimentos. Sobre la máquina se instala el software de Globus para permitir la gestión de recursos, el movimiento de datos y el control de la seguridad entre los grupos de investigación que hagan uso de él. Los usuarios se identificarán en el grid mediante las credenciales de autenticación del GSI (Grid Security Infrastructure), el estándar de seguridad de Globus. El supercomputador de HP será puesto a disposición del grid a través del Globus MDS (Monitoring and Discovery Service), un catálogo seguro de todos los recursos del grid.

En las instalaciones de Sun se llevan a cabo las simulaciones sobre aerodinámica. Su versión comercial es costosa.

GridSystems ha desarrollado el productor InnerGrid para la aplicación de la tecnología grid en entornos empresariales, académicos y de investigación. Este software divide datos y procesos en pequeñas unidades que se ajustan dinámicamente a los recursos conectados a la red. El sistema es multiplataforma (Windows y todos los sistemas UNIX).

Esta empresa lo que hace es aprovechar los ciclos libres de CPU aumentando la capacidad de cómputo global con un bajo coste. De esta manera los ciclos libres son utilizados por aplicaciones que consumen muchos recursos y necesitarían de un supercomputador o por trabajos puntuales e intensivos en una fracción de tiempo.

La instalación del producto en una empresa permite la utilización de máquinas que han quedado obsoletas y la instalación de nuevas máquinas sin que se interfiera en el trabajo de la compañía, ya que el software es fácilmente escalable.

### 1.6.2 Proceso Intensivo de Datos

Otra de las grandes áreas de aplicación es el almacenamiento masivo de datos. Dentro de esta área las nuevas versiones se centran en la aplicación de las ideas de Grid Computing al almacenamiento de datos. La nueva versión parte de la arquitectura actual en la que se encuentran la mayoría de los sistemas de información (ver fig. 1.5 de la pág. 13).

Cada una de las capas es de uso dedicado a cada sistema de información. Evidentemente, el fallo de alguno de los componentes de los tres niveles (Application Server, Database y Storage) supondría una pérdida del servicio (Oracle: Oracle Database 10g, Oracle Application Server 10g, Oracle Enterprise Manager 10g y Oracle 10g Development Tools). Esta nueva arquitectura basada en grid computing pretende crear "granjas/factorías de almacenamiento" y servidores estándares y modulares, a las que se puede añadir servidores rápidamente. En la fig. 1.6 en la pág. 13 se detalla a grandes rasgos la arquitectura propuesta (versión Oracle 10g.).

Las características de esta arquitectura serían:

- Capacidad de balanceo de sistemas.
- Alta disponibilidad.

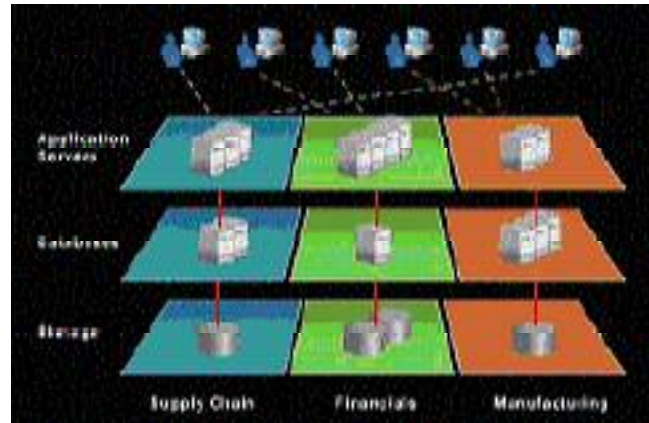


Figura 1.5: Arquitectura Actual de los Sistemas de Información.

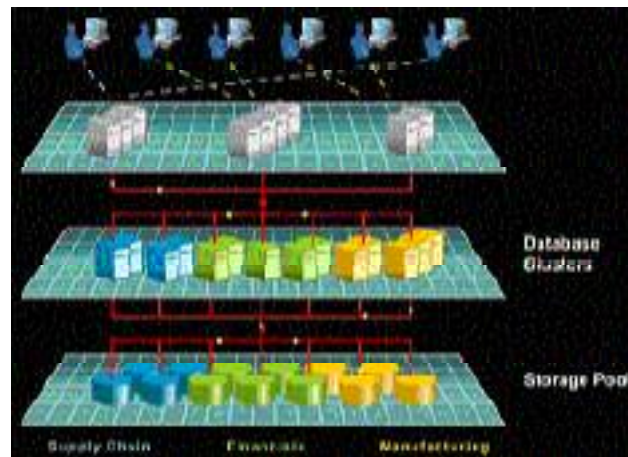


Figura 1.6: Arquitectura "Grid Computing" para Sistemas de Información.

- Reducción de costes.
- Facilidades de administración.

La utilización de esta versión permitira a una empresa mejorar sus prestaciones sin renunciar al comportamiento tradicional de los Sistemas de Información .

## Capítulo 2

# Principios de Grid Computing

### 2.1 Lo Que el Grid Computing Puede Hacer

Cuando se despliega un Grid, se podrá encontrar con un conjunto de requerimientos del cliente.

Para que las habilidades del Grid Computing se adapten mejor a esos requerimientos, es útil tener presente las razones para las que se usa el Grid Computing.

A continuación se describen las capacidades más importantes del Grid Computing.

### 2.2 Aprovechar los Recursos Que No Siempre se Usan

El uso más fácil del Grid Computing es ejecutar una aplicación existente en una máquina diferente [1] [11] [12] [5].

La máquina en que la aplicación normalmente se ejecuta podría estar inusualmente ocupada debido a un pico inusual de actividad.

El trabajo en cuestión podría ejecutarse en otra parte en una máquina ociosa en el Grid.

Hay dos requisitos previos a considerar en este apartado.

Primero, la aplicación debe ser ejecutable remotamente.

Segundo, la máquina remota debe encontrar cualquier hardware especial, software, o requerimientos de recursos impuestos por la aplicación.

Por ejemplo, un trabajo en lotes (batch) que consume una cantidad significativa de tiempo procesando un conjunto de datos de entrada para producir un conjunto de resultados, es quizás el uso más ideal y simple para un Grid.

Si las cantidades de entradas y salidas son grandes, más análisis y planeación podrían requerirse para usar eficazmente el Grid para tal trabajo.

Normalmente no tendría sentido usar un procesador de palabra remotamente en un Grid, porque habría retrasos mayores y más puntos potenciales de fracaso.

En la mayoría de las organizaciones, hay grandes cantidades de recursos no siempre utilizados.

La mayoría de las máquinas del escritorio están ocupadas menos del 5% del tiempo.

En algunas organizaciones, incluso las máquinas servidoras pueden estar a menudo relativamente ociosas.

El Grid Computing provee una estructura para explorar estos recursos y así tener la posibilidad de aumentar la eficacia de su uso.

Los recursos del procesamiento no son los únicos que pueden estar subutilizados.

A menudo, las máquinas pueden tener la enorme capacidad de disco no usadas.

El Grid Computing, más específicamente, un “Grid de datos”, puede usarse para agregar este almacenamiento no utilizado a un almacén de datos virtuales más grandes, posiblemente configurado para lograr mejorar la calidad y confiabilidad de cualquier máquina.

Si un trabajo batch necesita leer una cantidad grande de datos, éstos datos podrían reproducirse automáticamente en varios puntos estratégicos en el Grid. Así, si el trabajo debe ejecutarse en una máquina remota en el Grid, los

datos ya están allí y no necesitan ser movidos a ese punto remoto. Esto ofrece claros beneficios de calidad.

También, tales copias de datos pueden usarse como backups cuando las primeras copias se dañan o no están disponibles.

Otra función del Grid es equilibrar mejor la utilización del recurso.

Una organización puede tener picos inesperados ocasionales de actividad que exigen más recursos.

Si las aplicaciones en el Grid están habilitadas, ellos pueden moverse a las máquinas poco utilizadas durante tales picos.

De hecho, algunas aplicaciones del Grid pueden migrar los trabajos parcialmente terminados.

En general, un Grid puede proporcionar una manera consistente de equilibrar las cargas en una coalición de recursos más amplia.

Esto incluye la CPU, almacenamiento, y muchos otros tipos de recursos que pueden estar disponibles un Grid.

La administración puede usar un Grid para ver bien los modelos de uso en una organización más grande, permitiendo una mejor planificación al actualizar los sistemas, incrementar la capacidad, o retirar recursos de computing que ya no se necesitan.

## **2.3 La Capacidad de CPU Paralela**

El potencial para la capacidad de CPU paralela masiva es uno de los rasgos más atractivos de un Grid. Además de necesidades científicas puras, tal poder de computing está conduciendo a una nueva evolución en las industrias como el campo bio-médico, planeación financiera, exploración petrolera, etc.

El atributo común entre tales usos es que las aplicaciones se han escrito para usar algoritmos que pueden dividirse independientemente en partes de ejecución.

Una aplicación de Grid intensiva de CPU puede pensarse como muchos sub-trabajos más pequeños, cada uno ejecutándose en una máquina diferente en el Grid.

Si éstos sub-trabajos no necesitan comunicarse con el otro, la aplicación se vuelve más “escalable”.

Una aplicación absolutamente escalable, terminará diez veces más rápido si usa diez veces el número de procesadores.

A menudo existen barreras para perfeccionar la escalabilidad.

La primer barrera depende de los algoritmos usados para dividir la aplicación entre muchas CPUs.

Si el algoritmo sólo puede dividirse en un número limitado de partes independientes de ejecución, entonces eso forma una barrera de escalabilidad.

La segunda barrera aparece si las partes no son completamente independientes; esto puede causar contención que puede limitar la escalabilidad.

Por ejemplo, si todos los sub-trabajos necesitan leer y escribir desde un archivo común o base de datos, el acceso a ese archivo o base de datos se vuelve un factor limitante en la escalabilidad de la aplicación.

Otras fuentes de contención de inter-trabajo en una aplicación de Grid paralela incluye latencias de comunicaciones de mensajes entre los trabajos, la red, las capacidades de comunicación, los protocolos de sincronización, el ancho de banda de entrada-salida, los dispositivos de almacenamiento, y latencias que interfieren con los requerimientos de tiempo real.

## 2.4 Las Aplicaciones

Hay muchos factores para considerar en una aplicación de habilitación de Grid.

Se debe entender que no todas las aplicaciones pueden transformarse para ejecutarse en paralelo con un Grid y lograr escalabilidad.

Además se dispone de herramientas prácticas, que los diseñadores de aplicación pueden usar para escribir una aplicación paralela.

Sin embargo, una transformación automática de aplicaciones es una ciencia nueva.

Éste puede ser un trabajo difícil y a menudo puede requerir de matemática avanzada y de talentos de programación, si incluso es posible en una situación

dada.

Nuevas aplicaciones intensivas de computación se están diseñando para la ejecución paralela y éstas serán fácilmente habilitadas para su ejecución en Grid.

## **2.5 Los Recursos Virtuales y las Organizaciones Virtuales Para la Colaboración**

Otra importante contribución del grid es habilitar y simplificar la colaboración entre un grupo más amplio de usuarios.

Usar, computación distribuida prometía esta colaboración y la logró en alguna magnitud.

El Grid toma estas capacidades para un público aún más amplio, mientras ofrece estándares importantes que permiten a los sistemas muy heterogéneos trabajar juntos para formar la imagen de un sistema de Grid virtual grande, que ofrece una variedad de recursos virtuales.

Los usuarios del Grid pueden organizarse dinámicamente en un número de organizaciones virtuales (VO), cada una con diferentes requerimientos de política.

Estas VO pueden compartir sus recursos colectivamente como un Grid más grande.

Compartir, comienza con datos en forma de archivos o bases de datos.

Un “Grid de datos” puede extender las capacidades de varias maneras.

Primero, los archivos o bases de datos pueden ampliar muchos sistemas y lograr mayores capacidades que en cualquier sistema simple. Tales ampliaciones pueden mejorar las tasas de transferencia de datos a través del uso de las técnicas de striping.

Los datos pueden estar duplicados a través del grid para servir como backups y pueden organizarse en o cerca de las máquinas que probablemente necesiten los datos, junto con las técnicas de planificación avanzadas.

Compartir no se limita a los archivos, también incluye muchos otros recur-

sos, como equipos, software, servicios, licencias, y otros. Estos recursos son los “virtualizados” para darles una interoperabilidad más uniforme entre los participantes heterogéneos del Grid.

Los participantes y usuarios de Grid pueden ser miembros de varias organizaciones reales y virtuales.

El Grid puede ayudar a potenciar las reglas de seguridad entre ellas e implementar políticas que puedan resolver prioridades para recursos y usuarios.

Como se observa en la fig. 2.1 de la pág. 20, donde se virtualizan recursos heterogéneos y geográficamente dispersos para cada organización virtual que presenta una vista más simple.

Figura 2.1: Los Recursos Virtuales y Organizaciones Virtuales Para Colaboración.

## 2.6 El Acceso a los Recursos Adicionales

Los recursos adicionales pueden proporcionarse en número y capacidad variable.

Por ejemplo, si un usuario necesita aumentar su ancho de banda total a Internet para implementar un mecanismo de búsqueda de minería de datos, el trabajo podría dividirse entre máquinas del Grid que tienen conexiones independientes a Internet.

De esta manera, la capacidad total de búsqueda se multiplica, desde que cada máquina tiene una conexión separada a Internet.

Si las máquinas hubieran compartido la conexión en Internet, no habría habido un aumento eficaz en el ancho de banda.

Algunas máquinas pueden tener instalado el software bajo licencia costoso que los usuarios requieran.

Su trabajo puede enviarse a tales máquinas aprovechando las licencias del software.

## **2.7 Balanceo de Recursos**

Para aplicaciones habilitadas, el grid puede ofrecer un efectivo balanceo de recursos mediante la planificación de trabajos de grid en máquinas con poca utilización, como se ilustra en fig. 2.2 de la pág. 22.

Esta facilidad puede mejorar invaluablemente el manejo de picos de carga de actividad en sectores de una organización más grande. Esto puede pasar de dos maneras:

- Un pico inesperado puede ser conducido a máquinas relativamente ociosas en el Grid.
- Si el grid ya se utiliza totalmente, el trabajo de prioridad más baja que se realiza en el grid debe ser suspendido temporalmente o incluso cancelado y realizado posteriormente para dejar lugar a un trabajo de prioridad mayor.

Sin una infraestructura de Grid, tales decisiones de equilibrio serían difíciles de priorizar y ejecutar.

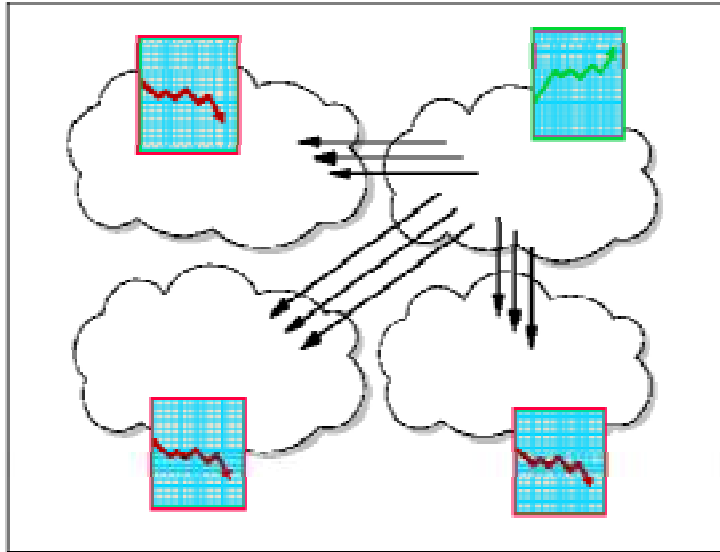


Figura 2.2: Balanceo de Recursos.

## 2.8 Confiabilidad

Los sistemas informáticos convencionales high-end usan hardware caro para aumentar la fiabilidad.

Se construyen usando chips con circuitos redundantes que brindan resultados y contienen mucha lógica para lograr una buena recuperación ante fallas de hardware.

Las máquinas también usan procesadores dobles con pluggability (conexión en caliente) para que cuando uno de ellos falla, se lo pueda reemplazar por el otro sin apagar el equipo.

Los sistemas de suministros de corriente eléctrica y de ventilación están duplicados. Los sistemas operan con fuentes especiales de energía que pueden encender los generadores si la corriente utilizada se interrumpe. Todo esto construye un sistema fiable, pero a un gran costo, debido a la duplicación de componentes de alta fiabilidad.

Los sistemas en un Grid pueden ser relativamente baratos y geográficamente dispersos.

Así, si hay algún tipo de falla, no es probable que las otras partes del Grid sean afectadas (ver fig. 2.3 en la pág. 23).

El software de gestión del Grid puede automáticamente reenviar trabajos a otras máquinas del Grid, cuando en una se descubre una falla.

En situaciones críticas de tiempo real, copias múltiples de trabajos importantes pueden ejecutarse en diferentes máquinas a través del Grid.

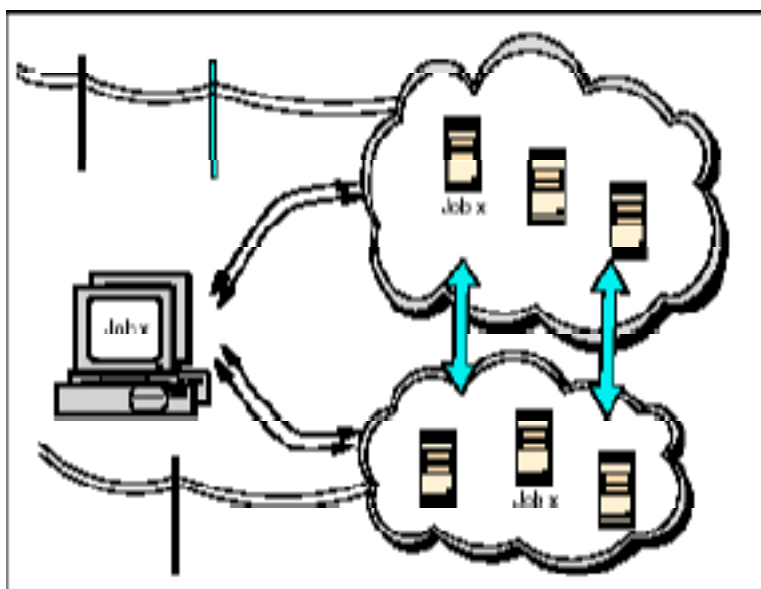


Figura 2.3: Confiabilidad en los Sistemas de Grid.

## 2.9 Administración

La meta de virtualizar los recursos en el Grid y administrar más uniformemente sistemas heterogéneos crearán nuevas oportunidades para gestionar mejor una infraestructura de IT más grande, y más dispersa.

Será más fácil visualizar la capacidad y utilización, haciéndolo más fácil para los departamentos de IT y así controlar los gastos para recursos de computación en una organización más grande.

El Grid ofrece administración de prioridades entre diferentes proyectos.

En el pasado, cada proyecto podría haber sido responsable de su propio hardware de recurso de IT y los gastos asociados con él.

A menudo este hardware podría no estar siempre utilizado mientras otro proyecto se encontrara en problemas, necesitando más recursos debido a eventos inesperados.

Con una visión mayor, un Grid puede ofrecer controlar y manejar más fácilmente tales situaciones. Como se ilustra en la fig. 2.4 de la pág. 24, los administradores pueden cambiar gran número de políticas que afectan en cómo las diferentes organizaciones pueden compartir o competir por los recursos.



Figura 2.4: Administración en los Sistemas de Grid.

## 2.10 Los Conceptos y Componentes del Grid

A continuación se introducen varios conceptos del Grid, componentes, y términos más detallados.

### 2.10.1 Los Tipos de Recursos

Un Grid es una colección de máquinas, a veces llamado “nodos”, “recursos”, “miembros”, “servidores”, “clientes”, “organizadores”, y muchos otros.

Todos ellos aportan grupos de recursos al Grid. Algunos recursos pueden ser usados por todos los usuarios del Grid, mientras que otros pueden tener restricciones específicas.

### 2.10.2 Computación

El recurso más común lo constituyen los ciclos de cómputo proporcionados por los procesadores de las máquinas en el Grid.

Los procesadores pueden variar en velocidad, arquitectura, plataforma de software, y otros factores asociados, como la memoria, almacenamiento, y conexión.

Hay tres formas primarias para aprovechar los recursos de cómputo de un Grid.

El primero y más simple es usarlo para ejecutar una aplicación existente en una máquina disponible en el grid, en lugar de localmente.

El segundo es para usar una aplicación diseñada para dividir su trabajo de tal manera que las partes separadas pueden ejecutarse en paralelo en los diferentes procesadores.

El tercero es ejecutar una aplicación que precise ser ejecutada muchas veces en muchas máquinas diferentes en el Grid. “Escalabilidad” es una medida de cómo se usan eficazmente los múltiples procesadores en un Grid.

Si los procesadores realizan dos veces una aplicación completa en la mitad del tiempo, entonces se dice que es absolutamente escalable.

Sin embargo, puede haber límites en la escalabilidad cuando las aplicaciones sólo pueden ser divididas en un número limitado de partes ejecutables separadamente, o si esas partes experimentan alguna otra contención para recursos de algún tipo de almacenamiento.

### 2.10.3 Almacenamiento

El segundo recurso más común en un Grid es el almacenamiento de los datos. Un Grid que proporciona una visión integrada de almacenamiento de datos es a veces llamado un “Grid de datos”.

Cada máquina en el Grid normalmente mantiene alguna cantidad de almacenamiento para uso del grid, aun temporalmente.

El almacenamiento puede ser memoria unida al procesador o almacenamiento secundario, usando el disco rígido u otros medios de almacenamiento permanente.

La memoria unida a un procesador normalmente tiene un muy rápido acceso pero es volátil.

Sería mejor usarlo para guardar datos y servir como un almacenamiento temporal para las aplicaciones ejecutadas.

El almacenamiento secundario en un Grid puede usarse de manera interesante para aumentar la capacidad, la calidad, el compartir, y la fiabilidad de los datos.

Muchos sistemas de Grid usan sistemas de archivo de red montables tales como, Andrew File System (AFS), Network File System (NFS), Distributed File System (DFS), o el General Parallel File System (GPFS). Estos ofrecen diferentes grados de calidad, seguridad y fiabilidad.

La capacidad puede ser incrementada usando el almacenamiento en múltiples máquinas con un sistema de archivo unificado.

Cualquier archivo individual o base de datos pueden expandirse a varios dispositivos de almacenamiento y máquinas, eliminando las restricciones de máxima capacidad, a menudo impuestas por los sistemas de archivo de los sistemas operativos.

Un sistema de archivo unificado también puede mantener un sólo espacio uniforme para el almacenamiento del Grid.

Esto facilita a los usuarios referenciar los datos que residen en el Grid, sin considerar su ubicación exacta.

De manera similar, el software especial de base de datos puede “federar”

un conjunto de bases de datos individuales y archivos para formar bases de datos más amplias y más comprensivas.

Como se observa en la fig. 2.5 de la pág. 27 el **Data Striping** significa escribir o leer sucesivos datos desde o para diferentes dispositivos físicos, solapando el acceso para un mejor rendimiento; asimismo, con técnicas adicionales se aumenta la fiabilidad.

Los sistemas de archivo más avanzados en un Grid pueden duplicar automáticamente conjuntos de datos, para proporcionar redundancia para mayor fiabilidad y aumento del rendimiento.

Un scheduler de grid inteligente puede ayudar a seleccionar los dispositivos de almacenamiento apropiados para guardar datos, basados en patrones de uso.

Los trabajos pueden ser programados más cerca de los datos, preferentemente en las máquinas directamente conectadas a los dispositivos del almacenamiento que guardan los datos requeridos.

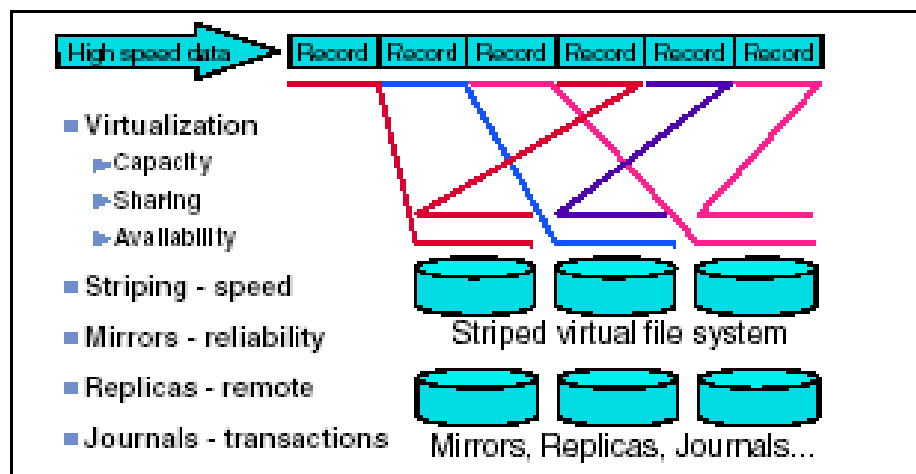


Figura 2.5: Almacenamiento en el Grid.

#### 2.10.4 Las Comunicaciones

El crecimiento rápido en la capacidad de comunicación entre las máquinas hace que el Grid computing sea práctico, comparado con el limitado ancho de banda disponible cuando la computación distribuída emergió por primera vez.

Por consiguiente, no debe ser una sorpresa que otro recurso importante de un Grid sea la capacidad de comunicación de datos. Esto incluye las comunicaciones dentro del Grid y fuera de él.

Las comunicaciones dentro del Grid son importantes para enviar trabajos y sus datos requeridos hacia puntos dentro del Grid.

Algunos trabajos exigen procesar una cantidad grande de datos y no siempre los mismos pueden residir en la máquina que ejecuta el trabajo.

El ancho de banda disponible para tales comunicaciones frecuentemente puede ser un recurso crítico que puede limitar la utilización del Grid.

El acceso de comunicación externo a Internet, por ejemplo, puede ser valioso al construir motores de búsqueda.

Las máquinas en el Grid pueden tener conexiones a Internet externas, en adición a las conexiones entre las máquinas del Grid. Cuando estas conexiones no comparten la misma ruta de comunicación, se agregan al ancho de banda disponible, para acceder a Internet.

A veces se necesitan rutas de comunicación redundantes para manejar mejor fallas potenciales en la red y el excesivo tráfico de datos.

En algunos casos se deben utilizar redes de altas prestaciones para satisfacer demandas de trabajos que transfieren cantidades muy grandes de datos.

Un sistema de administración de Grid puede mostrar mejor la topología del Grid y puede resaltar los cuellos de botella de comunicación. Esta información puede usarse para planificar las actualizaciones del hardware.

#### 2.10.5 El Software y las Licencias

El grid puede tener el software instalado que puede ser demasiado caro para instalar en cada máquina de éste.

Usando un Grid, los trabajos que requieren este software son enviados a máquinas particulares en las cuales este software está instalado.

Cuando las tarifas de licencia son significantes, esta aproximación puede ahorrar gastos importantes para una organización.

Algún arreglo de licencia de software permite instalarlo en todas las máquinas de un grid pero puede limitar el número de instalaciones que pueden usarse simultáneamente en cualquier momento.

El software de administración de licencias registra cuántas copias coexistentes de éste están usándose y previene que se ejecute un número mayor en un tiempo dado .

Los schedulers de trabajo de Grid pueden configurarse para tener en cuenta las licencias de software, opcionalmente balanceándolas contra otras prioridades o políticas.

### **2.10.6 El Equipo Especial, Capacidades, Arquitecturas, y Políticas**

Las plataformas en el Grid tendrán a menudo diferentes arquitecturas, sistemas operativos, dispositivos, capacidades, y equipos.

Cada uno de estos ítems representa un tipo diferente de recurso que el Grid puede usar con criterio para asignar trabajos a las máquinas.

Mientras algún software puede estar disponible en varias arquitecturas, por ejemplo PowerPC y x86, tal software es a menudo diseñado para ejecutar sólo un tipo particular de hardware y sistema operativo. Tales atributos deben ser considerados al asignar recursos en el Grid.

En algunos casos, el administrador de un Grid puede crear un nuevo tipo de recurso artificial que será usado por el schedulers para asignar el trabajo según el tipo de política u otras restricciones. Por ejemplo, algunas máquinas pueden diseñarse para sólo ser usadas para investigación médica y otras para sólo participar en el Grid si no se usan para propósitos militares.

### 2.10.7 Los Trabajos y las Aplicaciones

Aunque varias clases de recursos en el Grid pueden compartirse y usarse, normalmente son accedidos vía la ejecución de una “aplicación” o “trabajo”.

Normalmente se usa el término “aplicación” como el nivel más alto de un porción de trabajo en el Grid. Sin embargo, a veces el término “trabajo” se usa equivalentemente.

Las aplicaciones pueden dividirse en cualquier número de trabajos individuales, como se ilustra en la fig. 2.6 de la pág. 30; aquéllas, a su vez, pueden dividirse en sub-trabajos.

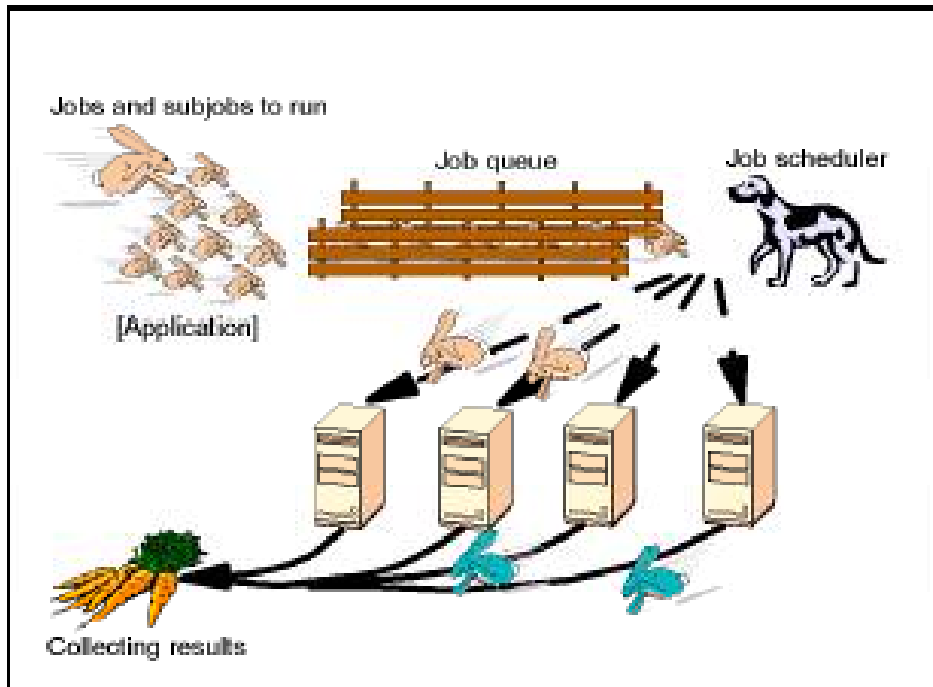


Figura 2.6: Trabajos y Aplicaciones del Grid.

## 2.11 Scheduling, Reservación, y Barrido

El sistema del Grid es responsable de enviar un trabajo a una máquina dada para ser ejecutado.

En el más simple sistema de Grid, el usuario puede seleccionar una máquina conveniente para realizar su trabajo y luego ejecutar una orden de Grid, que envía el trabajo a la máquina seleccionada.

Los sistemas de Grid más avanzados incluirían un trabajo “scheduler” de algún tipo que, automáticamente, encuentre la máquina más apropiada en la cual ejecutar cualquier trabajo dado, el cual estaría esperando ser ejecutado.

Los schedulers reaccionan a la disponibilidad actual de recursos en el Grid.

El término “scheduling” no debe ser confundido con la “reservación” de recursos para mejorar la calidad de servicio.

En un “scavenging” (barrido) de un sistema Grid, cualquier máquina que se vuelve ociosa (sin trabajo local que ejecutar) informaría su estado al nodo de administración del Grid. Este nodo de administración asignaría a esta máquina el próximo trabajo que se satisfecería por los recursos de la misma.

El “scavenging” normalmente es llevado a cabo de manera que no obstruya al usuario.

Si la máquina se pone en estado de “ocupada” con el trabajo local no-grid, el trabajo del grid normalmente se suspende o se demora. Esta situación crea de alguna manera tiempos de terminación imprevisibles para los trabajos del Grid, aunque no molesta a las máquinas donantes de recursos.

Para crear una conducta más predecible, las máquinas del Grid se dedican a menudo al Grid y no pueden ser retiradas del mismo para realizar trabajos externos al Grid. Esto permite al scheduler computar el tiempo aproximado de realización para un conjunto de trabajos, cuando sus características de ejecución son conocidas.

En el próximo paso, los recursos del Grid pueden “reservarse” por adelantado para un conjunto de trabajos designados. Esto se hace para reunir fechas tope y garantizar la calidad de servicio.

Cuando las políticas lo permitan, los recursos reservados de antemano podrían ser recogidos para ejecutar trabajos de menor prioridad, cuando no están

ocupados durante un período de reservación.

Así, varias combinaciones de scheduler, reservación y barrido, pueden ser usados para utilizar un Grid más completamente.

El scheduling y reservación es bastante directo cuando sólo un tipo de recurso, normalmente, CPU, está involucrado. Sin embargo, las optimizaciones de Grid adicionales pueden ser logradas considerando más recursos en proceso de reservación y planificación.

Por ejemplo, sería deseable asignar trabajos de ejecución a las máquinas más cercanas a los datos que estos trabajos requieren.

Esto reduciría el tráfico de la red y posiblemente reduciría los límites de escalabilidad.

La planificación óptima, considera múltiples recursos, es por ello que se la considera un problema matemático difícil. Por consiguiente, tales schedulers pueden usar heurísticas. Estas heurísticas son reglas que se diseñan para mejorar la probabilidad de encontrar la mejor combinación de schedulers de trabajo y reservaciones para perfeccionar el rendimiento o cualquier otra métrica.

## 2.12 Intragrid a Intergrid

Como se presenta en fig. 2.7 de la pág. 34, el grid más simple consiste en sólo unas pocas máquinas, todas de la misma arquitectura de hardware y el mismo sistema operativo, conectadas en una red local. Este tipo de Grid usa sistemas homogéneos, así hay menos consideraciones y pueden ser usados sólo para experimentar con el software de Grid.

Las máquinas usualmente están en un departamento de una organización, y sus usos como un Grid pueden no requerir una política especial de seguridad. Debido a que las máquinas tienen la misma arquitectura y sistema operativo, elegir software de aplicación para dichas máquinas es generalmente sencillo. Esto podría ser llamarlo una aplicación de “cluster” en lugar de “Grid”.

El siguiente avance sería incluir las máquinas heterogéneas. En esta configuración, hay más tipos de recursos disponibles. El sistema Grid puede incluir algunos componentes de scheduling. También puede lograrse el compartimiento de archivos, usando los sistemas de archivo de red.

Las máquinas que participan en el grid pueden incluir uno de los departamentos de la misma organización. Tal modelo de grid también será llamado “Intragrid”.

Cuando el Grid se extiende a muchos departamentos, las políticas pueden requerirse para indicar cómo el Grid debe usarse. Por ejemplo puede haber políticas, para qué tipos de trabajo permite el Grid y cuántas veces.

Puede haber también una priorización por departamento o por tipos de aplicaciones que deben tener acceso a los recursos del Grid.

Los datos sensibles en un sólo departamento pueden necesitar ser protegido contra el acceso de trabajos que se ejecutan para otros departamentos.

Las máquinas del Grid especializadas pueden ser agregadas para aumentar la calidad de servicio, en lugar de depender completamente de los recursos de “barrido”.

El Grid puede crecer geográficamente en una organización que tiene los medios en diferentes ciudades.

Las conexiones de comunicaciones dedicadas pueden usarse entre estas facilidades y el Grid.

En algunos casos, VPN tunneling u otras tecnologías pueden usar Internet para conectar las diferentes componentes de la organización. La seguridad aumenta una vez que los límites de cualquier facilidad se superan.

El Grid puede crecer para ser jerárquicamente organizado para reducir la contención implicada por el control central, aumentando la escalabilidad.

Un Grid puede crecer, cruzar los límites de la organización y puede usarse para colaborar en los proyectos de interés común. Esto se conoce como un “Intergrid” (como se ve en la fig. 2.8 de la pág. 2.8).

Habitualmente se requieren los niveles más altos de seguridad en esta configuración para prevenir posibles ataques y espionajes.

El Intragrid ofrece la posibilidad de comerciar recursos a un público más amplio.

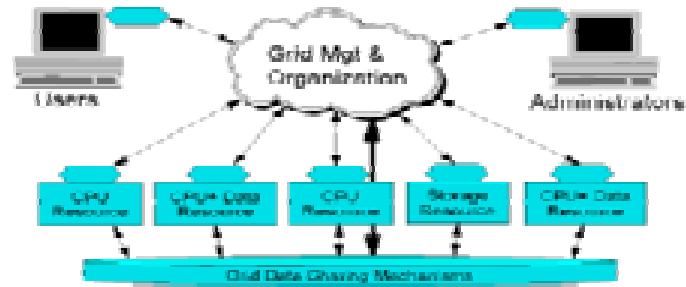


Figura 2.7: Un Grid más Simple.

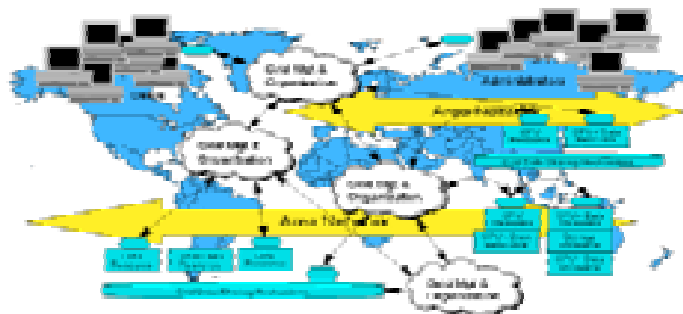


Figura 2.8: Un Intergrid más Complejo.

## 2.13 Construcción del Grid

Un Grid ad hoc puede instalarse por unos programadores en su tiempo libre, pero como el Grid crece, y los usuarios se vuelven más dependientes de él para el trabajo misión-crítico, un grado de planeamiento es esencial.

Es mejor entender los requerimientos de la organización y escoger las tecnologías del Grid que mejor se adapten a estos requerimientos.

Esta sección se discutió algunas consideraciones de planificación y componentes del Grid que cumplen con los requerimientos.

## 2.14 Planificación del despliegue

El uso de un Grid nace a menudo de una necesidad de incrementar recursos de algún tipo. Por ello una de las primeras consideraciones son el hardware disponible y cómo se conecta vía una LAN o WAN.

Luego, una organización puede querer agregar el hardware adicional para aumentar las capacidades del Grid. Es importante entender las aplicaciones a ser usadas en él.

Sus características pueden afectar las decisiones de cómo escoger y configurar el hardware y sus datos conectados.

### 2.14.1 Seguridad

La seguridad es un factor mucho muy importante; planear y mantener un grid computing convencional distribuido dónde el compartimiento de datos reduce el volumen de la actividad.

En un Grid, las máquinas miembro se configuran para ejecutar los programas en lugar de sólo mover datos. Esto es lo que hace un Grid potencialmente fértil para virus y programas de Caballo de Troya.

Por esta razón, es importante entender qué componentes exactamente del Grid deben afianzarse para detener rigurosamente cualquier clase de ataque.

Además, es importante comprender los problemas involucrados al autenti-

car usuarios y ejecutar adecuadamente las responsabilidades que esto implica.

### 2.14.2 Organización

Las consideraciones tecnológicas son importantes al desplegar un Grid. Sin embargo, los problemas comerciales y de organización pueden ser igualmente importantes.

Es primordial entender cómo los departamentos en una organización interactúan, operan y contribuyen al todo.

Frecuentemente hay barreras construidas entre los departamentos y proyectos para proteger sus recursos, en un esfuerzo de aumentar la probabilidad de éxito.

Sin embargo, volviendo a preocuparse por algunas de estas relaciones, se puede encontrar, que más compartimiento de recursos a veces puede beneficiar a la organización entera.

Por el ejemplo, un proyecto que se encuentra detrás del scheduler y encima del **budget** no puede admitirse el lujo de conseguir los recursos requeridos para resolver el problema.

Un Grid agregaría una medida de seguridad, proporcionando un margen extra de capacidad de recurso, necesitado para finalizar el proyecto. De manera similar, un proyecto en sus fases tempranas cuando los recursos de computadora no están utilizándose totalmente, puede donarlos a otros proyectos que los necesitan.

Un Grid también ofrece la habilidad para la administración de organización, y ver así el cuadro de prioridad más grande y reaccionar más rápidamente cambiando la utilización del recurso, prioridades, y políticas.

## 2.15 Componentes del Software Grid

En esta sección se presenta algunos de los componentes importantes que deben discutirse antes de diseñar una arquitectura de Grid Computing.

### 2.15.1 Componentes de administración:

Cualquier sistema de Grid tiene algunos componentes de administración.

Primero hay un componente que guarda , la muestra de los recursos disponibles para el Grid, y qué usuarios son miembros de él. Esta información se usa principalmente para decidir dónde deben asignarse los trabajos del Grid.

Segundo, hay componentes de medida que determinan, las capacidades de los nodos y su proporción de utilización actual en cualquier momento dado.

Tal información se usa también para determinar la robustez del Grid, advirtiéndolo al personal sobre los problemas como, los paros, congestión, u **over commitment**.

Esta información también es usada para determinar modelos de uso globales y estadísticas, así como para registrar y responder por el uso de recursos del Grid.

Tercero, el software de administración del Grid avanzado puede manejar muchos aspectos automáticamente.

Esto se conoce como “autonomic computing”, o “computing de recuperación” orientada. Este software se recuperaría automáticamente de diversos tipos de fracasos del Grid y paros, encontrando maneras alternativas de conseguir el trabajo procesado.

### 2.15.2 Software Servidor

Cada máquina que contribuye con recursos típicamente necesita conectarse como un miembro del Grid e instalar algún software que maneje el uso de los recursos del Grid. Habitualmente, alguna clase de proceso de identificación y autenticación deberá realizarse antes de que una máquina pueda unirse al Grid.

Un certificado de mando puede usarse para establecer la identidad de la máquina donante, también como los usuarios y el Grid mismo.

Algunos sistemas de Grid proporcionan su propio login, mientras otros dependen de los sistemas operativos procedentes para la autenticación del usuario.

En el último caso, un sistema de mapeo “ID de usuario” puede ser necesario para unir adecuadamente los derechos del usuario a las diferentes máquinas. Esto es típicamente mantenido de forma manual por un administrador del Grid.

Él determina qué usuario ID puede poseer qué máquina del Grid y qué base de datos o registros.

De esta manera, cuando los trabajos del Grid se presentan a diferentes máquinas para un usuario, donde el ID del usuario de la máquina local es usado para determinar sus derechos.

En algunos sistemas de Grid, es posible unir el mismo sin ninguna autenticación especial. Y en otros, es posible para cualquier usuario presentar trabajos al Grid.

Tales sistemas pueden ser convenientes para set up pero no para despliegues mas grandes, debido a problemas de seguridad serios que ellos ocasionarían.

El sistema de Grid hace que la información sobre los recursos recientemente agregados se encuentre disponibles a lo largo de él.

La máquina donante normalmente tendrá alguna clase de monitor que determine o mida cuan ocupada está y la proporción o cantidad de recursos utilizados.

Esta información es “bubbled up” al software de dirección del Grid y usada para scheduler de acuerdo con esos recursos.

En un sistema barrido de basura, esta información le dice al software de dirección del Grid, cuándo la máquina está ociosa y disponible para el trabajo.

Es importante que el software instalado en una máquina dada pueda aceptar un trabajo ejecutable desde un sistema de administración de Grid y ejecutarlo.

El software de gestión de Grid debe ser capaz de recibir el archivo ejecutable o seleccionar el propio, desde copias preinstaladas en la máquina donante. Éste lo ejecuta y el resultado es enviado devuelta al cliente.

Implementaciones mas avanzadas pueden dinamicamente ajustar la prioridad de un trabajo en ejecución suspenderlo y continuar después, o controlarlo con la posibilidad de continuar su ejecución en una máquina diferente.

Estos tipos de acciones pueden ser necesarios para responder a problemas de nivelación de carga, prioridad o cambios de política en el Grid.

### 2.15.3 Software de Sumisión

Normalmente cualquier máquina miembro del Grid puede usarse para realizar los trabajos e iniciar sus requerimientos.

Sin embargo, en algunos sistemas de Grid, esta función se lleva a cabo como un componente separado, instalado en “nodos de sumisión” o “clientes de sumisión”. Cuando un Grid se construye usando recursos especializados en lugar de “recursos scavenged” el software de sumisión separado es normalmente instalado en la estación de trabajo.

### 2.15.4 Administración del Grid Distribuido

Los grid más grandes normalmente pueden tener una topología organizacional o jerárquica uniendo las conexiones topológicas. Es decir, que las máquinas localmente conectadas con una LAN forman un cluster de máquinas. Éste puede organizarse en una jerarquía que consiste en cluster de cluster.

El trabajo involucrado en la administración del Grid se distribuye para aumentar su escalabilidad.

Programa un trabajo subordinado directamente a la máquina para ejecutarlo.

En cambio el trabajo se envía a un programador de menor nivel que ocupa un conjunto de máquinas. Este se ocupa de la asignación de máquinas específicas.

De la misma forma se distribuye la colección de información estadística.

Los cluster de menor nivel reciben la información de actividad, la agregan y las envían desde las máquinas individuales a los nodos de dirección de niveles más altos en la jerarquía.

### 2.15.5 Schedulers

La mayoría de los sistemas de Grid incluyen alguna clase de software de programación de trabajo.

Este software localiza una máquina en la cual ejecutar un trabajo de grid que ha sido presentado por un usuario.

En los más simples de los casos, puede asignar ciegamente los trabajos en forma round-robin a la próxima máquina cumpliendo con los requerimientos del recurso. Sin embargo hay ventajas al usar un scheduler más avanzado.

Algunos schedulers llevan a cabo un sistema de prioridad de trabajo. Esto a veces se hace usando varias colas de trabajo, cada uno con una prioridad diferente. Cuando las máquinas del Grid están disponibles para ejecutar los trabajos, éstos son tomados en orden de prioridad.

Varias clases de políticas son llevadas a cabo usando los programadores.

Las políticas pueden incluir los varios tipos de restricciones en los trabajos, los usuarios, y recursos. Puede haber una política de que restrinja la ejecución de trabajos de Grid en ciertos momentos del día.

Los schedulers normalmente reaccionan a la carga inmediata del Grid. Ellos usan la información de medida, sobre la utilización actual de máquinas para determinar cuáles no están ocupadas, antes realizar un trabajo.

Los schedulers pueden organizarse en una jerarquía.

Por ejemplo, un meta-scheduler puede enviar un trabajo a un scheduler del cluster o a otros scheduler de menor nivel en lugar de a una máquina individual.

Los schedulers más avanzados supervisarán el progreso de trabajos programados conduciendo el flujo de trabajo global. Si los trabajos se pierden debido al sistema o paros de la red, un buen scheduler reenviaría automáticamente el trabajo a otra parte.

Sin embargo, si un trabajo parece estar en un loop infinito y alcanza una interrupción máxima, entonces no deberan reprogramarse tales trabajos.

Típicamente, los trabajos tienen diferentes tipos de códigos de realización, algunos de los cuáles son convenientes para el reenvío y otros no.

Reservar los recursos en el Grid de antemano se realiza mediante un “sistema reservación”.

Este es primero, un sistema basado en un calendario para reservar los recursos con períodos de tiempo específico y así evitar que otros reserven el mismo recurso al mismo tiempo.

Este también debe poder quitar o suspender trabajos que podrían estar ejecutándose en cualquier máquina o recurso, cuando el período de la reservación es alcanzado.

### 2.15.6 Las Comunicaciones

Un sistema de Grid puede incluir un software para ayudar a los trabajos a comunicarse entre sí.

Por ejemplo, una aplicación puede dividirse en un gran número de sub-trabajos. Cada uno de estos es un trabajo separado en el Grid. Sin embargo, la aplicación puede ser llevada a cabo por un algoritmo que requiere que los sub-trabajos se comuniquen alguna información entre ellos.

Los sub-trabajos necesitan ser capaces de localizar otros sub-trabajos específicos, establecer una conexión de comunicaciones con ellos, y enviar los datos apropiados.

El estándar abierto Message Passing Interface (MPI) y cualquier otra variación es a menudo incluida como parte del sistema de Grid sólo para este tipo de comunicación.

## 2.16 Observación, Dirección, y Medición

Se mencionó que los schedulers reaccionan ante las cargas actuales en el Grid.

Habitualmente, el software servidor incluirá algunas herramientas que midan la carga actual y la actividad en una máquina dada usando, las facilidades de un sistema operativo o por medición directa.

Este software es a veces llamado “sensor de carga”.

Algunos sistemas de Grid proporcionan los medios para implementar sen-

sores de carga personalizada en lugar de CPU o recursos del almacenamiento.

Tal información de medición es útil no sólo para programar sino también para descubrir el uso de patrones globales en el Grid.

Las estadísticas pueden mostrar tendencias que pueden señalar la necesidad de un hardware adicional. También, la información de medición sobre trabajos específicos pueden ser recolectados y usarse para predecir mejor los requerimientos de recurso del trabajo que se ejecutara la próxima vez.

Cuanto mejor sea la predicción más eficiente será el trabajo del Grid.

La información de medición también puede ser guardada con propósitos de contabilidad, para formar la base, la ejecución de un recurso de Grid , o para manejar las prioridades de forma más justa.

La información también puede ser mostrada en varios formatos para visualizar mejor actividad y utilización del Grid.

## **2.17 Usar un Grid: Perspectivas de Usuario**

### **2.17.1 Conectar e instalar el software de Grid**

Un usuario se conecta primero como un usuario de Grid, e instala el software en su propia máquina.

Él también puede conectar su máquina como un servidor en Grid.

Conectarse en él puede requerir la autenticación con propósitos de seguridad.

El usuario efectivamente establece su identidad con un certificado de autoridad. Esto no debe hacerse solamente vía Internet.

La autoridad hace que un certificado este disponible para las necesidades del software para asegurar así la identidad de un usuario de Grid y sus requerimientos.

Pasos similares pueden ser requeridos para identificar la máquina donante.

El usuario tiene la responsabilidad de guardar y proteger sus credenciales de grid.

Una vez el usuario y/o la máquina se autentican, el software del Grid se proporciona al usuario para instalar en su máquina con propósitos de usar el Grid, así como servir a éste.

Este software puede ser automáticamente pre-configurado por el sistema de administración del Grid para conocer la dirección de comunicación, los nodos de dirección en el Grid e información de identificación del usuario o máquina.

De esta manera, la instalación puede realizarse haciendo un click con un mínimo de interacción requerida por parte del usuario.

En las instalaciones del Grid menos automatizadas, puede pedirse al usuario que identifique el nodo de dirección del mismo y posiblemente otra información de configuración.

Él puede escoger limitar los recursos servidos al Grid, las veces que su máquina es utilizada por dicho Grid, y otras restricciones relacionadas con la política del sistema.

El usuario también puede que necesite informar al administrador del Grid que IDs de usuario son suyos en otras máquinas que existen en el Grid.

## 2.18 Registrarse en el Grid

Para usar el Grid la mayoría de los sistemas exigen al usuario que se registre en un sistema usando el ID de usuario que se matricula en el Grid.

Otros sistemas de Grid pueden tener su propio login de Grid ID separado del sistema operativo.

Un registro del Grid es usualmente más conveniente para los usuarios de Grid. Este elimina los problemas de conexión de ID entre las diferentes máquinas.

También hace que el Grid se parezca más a una gran computadora virtual en lugar de una conjunto de máquinas individuales.

Por ejemplo, Globus lleva a cabo un modelo de login que mantiene al usuario registrado por una cantidad específica de tiempo, aun cuando él se desconecta y vuelve al sistema operativo o cuando la máquina es reanudada.

Una vez registrado el usuario puede solicitar al Grid y realizar los trabajos.

Algunas aplicaciones de del Grid permiten algunas funciones de solicitud si el usuario no está conectado en él o aun cuando el usuario no este registrado (**enrolled**) en el Grid.

## 2.19 Solicitar y Realizar Trabajos.

El usuario normalmente realizará algunas preguntas para verificar y para ver cuan ocupado está el Grid, para ver cómo sus trabajos realizados están progresando, y para buscar los recursos en él.

Los sistemas del Grid habitualmente proporcionan las herramientas de línea de orden así, como interfaces gráficas de usuario (GUIs) para las preguntas.

Las herramientas de línea de orden son especialmente útiles cuando el usuario quiere escribir una documento que automatiza una sucesión de acciones.

Por ejemplo, el usuario podría escribir un documento para que busque un recurso disponible, realice un trabajo para él, mire el progreso del trabajo, y presente los resultados cuando el trabajo ha terminado.

La realización del trabajo normalmente consiste en tres partes, aun cuando hay sólo una orden requerida.

Primero, algunos datos de entrada y posiblemente programas ejecutables o la ejecución de un archivo de documento, se debe enviar a la máquina para ejecutar el trabajo; enviar la entrada se llama “escenario de los datos de entrada”.

Alternativamente, pueden pre-instalarse los datos y archivos del programa en las máquinas del Grid o hacerlas accesibles vía un montable sistema de archivo de red de computadoras.

Cuando el Grid consiste en máquinas heterogéneas, puede haber archivos de programa ejecutables múltiples, cada uno compilado en diferentes plataformas de máquinas en el Grid.

Un rasgo bueno proporcionado por algunos sistemas Grid es registrar estas versiones múltiples del programa, para que el sistema de Grid pueda escoger automáticamente una versión que se conecte correctamente a la máquina de él, la cual ejecutará el programa.

Algunas tecnologías de Grid requieren que el programa y los datos de la entrada primero se procesen o incluyan (“**wrapped**”) de alguna manera por el sistema de Grid. Esto puede hacerse para agregar controles de ejecución protectores alrededor de la aplicación o simplemente recolectar todos los archivos de datos en uno.

Segundo, el trabajo se ejecuta en la máquina del Grid.

El software de Grid que se ejecuta en la máquina donante, ejecuta el programa en un proceso a favor del usuario.

Puede usar a un ID de usuario común en la máquina o puede usar ID de usuario propio del usuario , dependiendo de que tecnología de Grid se usa.

Algunos sistemas de Grid llevan a cabo un “sandbox” protector alrededor del programa, para que este último no cause ninguna daño a la máquina servidora si encuentra un problema durante la ejecución. Pueden restringirse derechos para acceder archivos y otros recursos en la máquina del Grid.

Tercero, se envían de vuelta los resultados del trabajo al cliente.

En algunas aplicaciones, los resultados intermedios pueden ser vistos por el usuario que realizó el trabajo.

En algunas tecnologías de Grid que no envían de vuelta automáticamente los datos del resultado al usuario, deben ser explícitamente enviados al usuario, posiblemente usando un sistema de archivo conectado a una red de computadoras.

Los documentos también son útiles para realizar una serie de trabajos, para una aplicación espacial de parámetro, por ejemplo.

Algunos problemas del cómputo consisten en una búsqueda de resultados deseados basados en algunos parámetros de entrada. La meta es encontrar los parámetros de entrada que producen los mejores resultados deseados.

Para cada parámetro de entrada, un trabajo separado se ejecuta para encontrar el resultado a ese valor.

La aplicación entera consiste en muchos trabajos como estos, los cuales exploran los resultados un gran número de valores de parámetro de entrada.

Los documentos normalmente se usan para enviar los sub-trabajos, cada uno, recibiendo sus propios valores de parámetro particular.

Las entradas de parámetro a veces pueden ser más complejas, que solo un número. A veces un conjunto de datos de entrada diferente representa el parámetro “input”.

Los documentos ayudan a automatizar la gran variedad de problemas del estudio espacial de parámetros.

Para entradas espaciales de parámetro más simple, algunos productos de grid proporcionan un GUI para realizar una serie de sub-trabajos, cada uno con valores de parámetro de entrada diferentes.

Cuando hay un gran número de sub-trabajos, el trabajo requerido para recolectar los resultados y producir el resultado final es normalmente realizado por un sólo programa, mientras se ejecuta en la máquina en el momento de realización del trabajo.

Si hay un gran número de sub-trabajos requeridos para una aplicación, el trabajo de compilar los resultados también podría ser distribuido.

Por ejemplo, el sub-trabajo que envíe más sub-trabajos al Grid sería responsable de recolectar y agregar los resultados que estos obtuvieron.

## 2.20 Configuración de Datos

Los datos obtenidos por los trabajos del Grid simplemente pueden organizarse adentro y a fuera del sistema de Grid.

Sin embargo, dependiendo de su tamaño y el número de trabajos, este puede potencialmente agregarse a una gran cantidad de tráfico de los datos. Por esta razón, algunas ideas se dan acerca de cómo obtener el mínimo movimiento de los datos en el Grid.

Por ejemplo, si va a haber un número muy grande de sub-trabajos ejecutándose en la mayoría los sistemas del Grid para una aplicación que se ejecutara repetidamente, los datos que ellos usan pueden copiarse para cada máquina y residir hasta la próxima vez que se ejecute una aplicación. Esto es mejor que usar un sistema del archivo conectado a una red de computadoras para compartir estos datos, porque en tal sistema de archivo, los datos serían efectivamente movidos desde una ubicación central cada vez que la aplicación se ejecute.

Esto es cierto a menos que el sistema de archivo lleve a cabo una exclusiva copia o duplique los datos automáticamente.

Hay muchas consideraciones en la planificación eficiente de la distribución y compartimiento de datos en un Grid.

Este tipo de análisis es necesario para grandes trabajos para así poder utilizar bien el grid y no crear cuellos de botella innecesarios.

## 2.21 Monitoreo del Progreso y Recuperación

El usuario puede solicitar el sistema de Grid, para ver cómo su aplicación y sus sub-trabajos están progresando.

Cuando el número de sub-trabajos crece, se vuelve mas difícil de listarlos a todos en una ventana gráfica. En cambio puede haber simplemente un sólo gráfico grande de barra que puede mostrarlos.

Es más difícil para el usuario decir si algún sub-trabajo particular no se ejecuta apropiadamente.

Un sistema de Grid, junto con su scheduler de trabajo, proporciona a menudo algún grado de recuperación para sub-trabajos que fallan.

Un trabajo puede fallar debido a un:

- **Error de Programación:** El trabajo se detiene alguna parte con alguna falla de programa.
- **Fallo de Hardware :** La máquina o dispositivos que se usan dejan de trabajar de alguna manera.
- **Interrupción en las Comunicaciones:** Un camino de comunicación a la máquina ha fallado o ha sido cargado excesivamente con otro tráfico de datos.
- **Lentitud Excesiva:** El trabajo podría estar en un loop infinito o el progreso de trabajo normal puede ser limitado por otro proceso que se ejecuta con una prioridad mayor.

No siempre es posible determinar automáticamente si la razón del fracaso

de un trabajo es debido a un problema con el diseño de la aplicación o si es debido a los fracasos de varios tipos en la infraestructura del sistema del Grid.

Los schedulers se diseñan a menudo para categorizar los fracasos del trabajo de alguna manera y automáticamente rehacer éstos para que ellos tengan éxito, mientras se ejecuten en otra parte del Grid.

En algunos sistemas, el usuario está informado sobre cualquier fracaso del trabajo y debe decidir si emitir una orden para intentar reejecutar los trabajos fallados.

Las aplicaciones de Grid pueden diseñarse para automatizar el monitoreo y recuperación de sus propios sub-trabajos usando las funciones proporcionadas por las interfaces de programación (APIs) de aplicación del software de sistema de Grid.

## 2.22 Reservar recursos

Para mejorar la calidad de un servicio, el usuario puede acordar reservar un conjunto de recursos por adelantado para su uso exclusivo o de mayor prioridad.

Una analogía del sistema de calendario puede usarse aquí. Tal sistema de reservación también puede usarse junto con el hardware planeado o eventos de mantenimiento de software, cuando el recurso afectado no están disponible para el uso del Grid.

En un Grid de recolección de residuos, no es posible reservar las máquinas específicas de antemano. En cambio, los sistemas de dirección de Grid pueden asignar un fragmento más grande de su capacidad para una reservación dada para permitir la posibilidad de que algunos de los recursos se vuelvan no disponibles.

Esto debe hacerse junto con herramientas que han perfilado la capacidad de trabajo de Grid para tener las estadísticas fiables sobre la habilidad de dicho Grid y así dar la reservación.

## 2.23 Usar un Grid: La Perspectiva de un Administrador

### 2.23.1 Planeación

El administrador debe entender los requerimientos de la organización del Grid, para elegir mejor las tecnologías del Grid, que satisfagan esos requerimientos. Las siguientes secciones describen brevemente los pasos que el administrador debe tomar para manejar el Grid.

Se sugiere que se comience por desplegar un pequeño Grid, para aprender sobre su instalación y dirección, antes de tener que confrontar los problemas más complicados involucrados con un Grid grande.

### 2.23.2 Instalación

Primero, el sistema de Grid seleccionado debe instalarse un conjunto de máquinas apropiadamente configuradas. Estas máquinas usando redes con amplitud suficiente para otras máquinas en el Grid.

Es importante entender los escenarios de fracaso para el sistema de Grid dado, de manera que este continúe operando aun cuando cualquiera de las máquinas de dirección falle de alguna manera.

Las máquinas deben configurarse y conectarse para facilitar los escenarios de recuperación.

Cualquier base de datos crítica u otros datos esenciales para guardar la muestra de los trabajos, los miembros del Grid, y las máquinas, éstos deben tener posibilidad de backups.

Además, los certificados claves públicos deben tener una copia de seguridad y las claves privadas deben guardarse en un lugar seguro inaccesible por otros.

Después de la instalación, el software del Grid puede necesitar ser configurado para la dirección de la red local e IDs.

El administrador normalmente requerirá el acceso base a las máquinas conductoras en el grid.

En algunos sistemas de Grid, él necesitará también el acceso **base** a las

máquinas servidoras requeridas para instalar el software también en aquellas. El software a ser instalado en las máquinas servidoras puede necesitar ser personalizado que para pueda encontrar automáticamente las máquinas de dirección (management) del Grid e incluir las claves públicas pre-instaladas para el Grid.

Este software puede proporcionarse para servidores potenciales en un FTP o el servidor equivalente o sea disponible en los medios de comunicación físicos.

Una vez, que el Grid es operacional, puede haber software de aplicación y datos que deben ser también instalados en las máquinas servidoras. Este software puede tener restricciones específicas de autorización, las cuales deben entenderse y adherirse. Algunos sistemas de Grid incluyen herramientas para ayudar con la dirección de la autorización de un Grid amplio. Este puede ayudar seguir las reglas de autorizaciones y explotar eficazmente esas autorizaciones.

## 2.24 Matriculación de Dirección de Servidores y Usuarios

Una tarea continua para el administrador del grid es dirigir a los miembros del Grid, a las máquinas servidoras de recursos y los usuarios.

Los usuarios pueden organizarse como grupos de proyecto.

El administrador es responsable para controlar los derechos de los usuarios en el Grid.

Las máquinas donantes pueden tener derechos de acceso que requieren también la dirección.

La ejecución de trabajos de Grid en las máquinas donantes puede hacerse bajo un especial ID usuario de Grid a beneficio de los usuarios que realizan los trabajos.

Los derechos de éstos IDs de usuario de grid deben ser apropiadamente puestos para que los trabajos no permitan el acceso a las partes de la máquina donante en las cuales el usuario no esta registrado.

Cuando los usuarios se conectan al Grid, su identidad debe establecerse positivamente y debe entrar en el certificado de autoridad.

El usuario y sus credenciales de certificado deben ser agregadas a la lista del usuario usando el software apropiado para el sistema de Grid desplegado.

En algunos casos, el administrador debe propagar la información del usuario a varias o todas las máquinas.

También, cuando el sistema de Grid depende principalmente del sistema operativo, para el login del usuario, el administrador puede necesitar para agregar entradas para perfilar al usuario del Grid específicos sistema operativo de IDs de usuario en las máquinas donantes.

Una actividad de registro similar normalmente se exige para registrar máquinas donantes en el Grid.

La identidad de máquina se establece y registra con el certificado la autoridad.

El administrador del Grid debe estar un acuerdo con el administrador de la máquina donante sobre el IDs de usuario, software, derechos de acceso, y cualquier restricción de política.

El administrador debe introducir las credenciales de identificación de la máquina, direcciones, y características de recurso usando el software apropiado para registrar la máquina servidora del Grid.

En algunos casos, el administrador puede necesitar propagar esta información manualmente a otras máquinas en el Grid.

Los procedimientos correspondientes para quitar usuarios y máquinas deben ejecutarse por el administrador.

## **2.25 Certificado de Autoridad**

Es crítico asegurar los más altos niveles de seguridad en un Grid ya que éste se diseña para ejecutar un código y no sólo para compartir datos. Así, este puede ser fértil para virus, los Trojan horses, y otros ataques, si el sistema de Grid se compone de alguna forma.

El certificado de autoridad para mantener una fuerte seguridad del Grid.

Una organización puede escoger usar un externo certificado de autoridad u operar uno ella misma.

Usted debe poder confiar en el certificado de autoridad para adherirse estrictamente a sus responsabilidades.

Las responsabilidades primarias de un certificado de autoridad son:

- Identificar positivamente las entidades que piden los certificados.
- Emitir, quitar, y archivar los certificados.
- Proteger al servidor del certificado de autoridad.
- Mantener un namespace de nombres únicos para los dueños del certificado.
- El servidor firma los certificados a aquéllos que necesiten autenticar entidades.
- Actividad de Registración.

Brevemente, un certificado de autoridad se basa en un sistema de encriptación de clave pública.

En este sistema, se generan las claves de a pares, una clave pública y una privada.

Cualquiera de las dos puede usarse para encriptar algunos datos, tal que la otra se necesita descifrarlo.

La clave privada es guardada por el dueño y nunca revelada a nadie, y la pública es dada a cualquiera.

Un certificado de autoridad es usado para mantener estas claves públicas y garantizar a quienes ellas pertenecen.

Cuando un usuario usa su clave privada para encriptar algo, el receptor usa la correspondiente clave pública para descifrarlo.

El receptor sabe que solo la clave pública puede descifrar el mensaje correctamente.

Sin embargo, cualquiera podría interceptar este mensaje y podría descifrarlo porque cualquiera puede conseguir la clave pública original.

Si el creador encripta doblemente el mensaje con su clave privada y la clave pública del destinatario, una segura conexión de comunicación se forma.

El receptor usa su clave privada para descifrar el mensaje y luego usa la clave pública del remitente para la segunda decriptación.

Ahora el destinatario sabe que si el mensaje se descifra apropiadamente, sólo el remitente lo podría haber enviado y además, este sabe que sólo el receptor al que se dirigió podrá descifrarlo.

Lo bueno de todo esto es que nadie tiene que llevar en forma segura una clave de encriptación del remitente al receptor, como debe hacerse para los sistemas de encriptación convencionales, y cualquier alteración en la comunicación se revela.

Así, este puede ser fértil para virus y, los Trojan horses, y otros ataques si el sistema de grid se compone de alguna forma.

Un intercambio similar se usa para obtener la clave pública de alguien desde el certificado de autoridad, para que el usuario sepa que él ha recibido una clave pública inalterada para el usuario deseado.

## 2.26 Administración de Recursos

Otra responsabilidad del administrador es manejar los recursos del Grid.

Esto incluye establecer los permisos para los usuarios del Grid para usar los recursos, así como ruteo del uso del recurso y llevar a cabo un correspondiente sistema de contabilidad o registro.

El uso de estadísticas es útil en la identificación de direcciones en una organización que puede requerir la adquisición hardware adicional, la reducción de hardware en exceso para reducir los costos, y ajustes de prioridades y políticas para lograr una utilización más justa o lograr mejor las metas globales de una organización.

Algunos componentes del Grid, normalmente, los schedulers de trabajo, tienen provisiones para reforzar prioridades y políticas de varios tipos.

Es responsabilidad del administrador configurarlos para el mejor logro de los objetivos de la organización global.

Los administradores de licencia de software pueden usarse en un Grid para controlar la utilización apropiada. Éstos pueden configurarse para trabajar

con schedulers de trabajo y para priorizar el uso de las licencias limitadas.

## 2.27 Compartir Datos

Para Grid pequeños, el compartir datos puede ser bastante fácil, usando los sistemas de archivo existentes, conectados a una red de computadoras, a bases de datos, o protocolos estándar de transferencia de datos. Cuando un Grid crece y los usuarios se vuelven dependientes de cualquiera de los almacenes de almacenamiento de datos, el administrador debe considerar los procedimientos para mantener copias de seguridad y réplicas, para mejorar así la realización.

Todos lo que se refiere a la dirección de recursos se aplican a los datos en el Grid.

## 2.28 Usar un Grid:Una Perspectiva del Diseñador de la Aplicación

Las aplicaciones de Grid pueden categorisarse en una de las siguiente tres categorías:

- Aplicaciones que no se habilitan por usar los procesadores múltiples pero pueden ejecutarse en diferentes maquinas.
- Aplicaciones que ya son diseñadas para usar los procesadores múltiples de un conjunto del Grid.
- Aplicaciones que necesitan ser modificadas o reescritas para aprovecharse mejor un Grid.

La última categoría es de interés para diseñadores de aplicación de Grid. Ellos encontrarán una necesidad por las herramientas para depurar y medir la conducta de aplicaciones del grid. Tales herramientas basadas son aun nuevas.

Puede ser útil para diseñadores configurar un Grid pequeño para poder usar los depuradores en cada máquina, para así controlar y mirar los funcionamientos detallados de las aplicaciones. Ya que el proceso de depuración

puede desviar ciertas precauciones de seguridad, puede no siempre ser eficaz para permitir tal depuración en un Grid de producción.

Globus es más un equipo de herramientas del diseñador (toolkit) para construir componentes de grid, en lugar de un el sistema de Grid comprensivo. Este posee los componentes básicos necesarios para construir nuevos medios para administrar funcionamientos del Grid, medición, reparación, y depuración de aplicaciones.

Las herramientas que conforman las interfaces de la Arquitectura de Servicios del Grid Abierta surgiendo (OGSA)serán utilizables en varios sistemas de Grid del verndedor.

## **2.29 El Presente y el Futuro del Grid**

El toolkit de Globus es un juego de herramientas útil para construir un Grid. Su fuerte es un modelo de alta seguridad, con una provisión para jerárquicamente colectar datos sobre el Grid, así como facilidades básicas para implementar un simple, world-spanning grid.

El Globus crecerá con el tiempo promete extenderse a través del trabajo de muchas organizaciones que están desarrollando sus capacidades.

La mayoría de los sistemas de Grid incluyen algún schedulers de trabajo, pero como áreas más amplias de medición de Grid, habrá una necesidad para más meta-schedulers que puedan manejar colecciones de clusters diversamente configurados y los grid más pequeños.

Estos schedulers evolucionarán para programar bien los trabajos, considerando múltiples recursos en lugar de sólo utilización de CPU.

Ellos también extenderán su alcance para llevar a cabo una mejor calidad de servicio, usando reservaciones, redundancia, e historia de perfiles de trabajos y la calidad de un Grid.

Hoy, los sistemas de Grid todavía están en las fases tempranas para proporcionar un almacenamiento y compartir datos virtuales de manera fiable, bien realizada, y automáticamente recuperables.

Se verán productos que colocan esta tarea en un conjunto de Grid, creando datos de todo tipo, y logrando una mejor calidad, integración con la progra-

mación, fiabilidad, y capacidad.

El computing autónomo tiene el objetivo de hacer el trabajo del administrador más fácil, automatizando varias tareas complicadas involucradas en la administración de un Grid. Éstos incluyen identificación de problemas en tiempo real y una rápida iniciación de acciones correctivas, antes de que ellos dañen seriamente el Grid.

La Arquitectura de Servicios de Grid Abierta (OGSA) es una norma abierta a la base de todas éstas mejoras de los Grid futuros.

OGSA estandarizará las interfaces del grid que serán usadas por los nuevos schedulers, agentes de computing átomos y cualquier otro servicio a ser desarrollado por el Grid. Este lo hará más fácil para reunir los mejores productos de varios vendedores, aumentando el valor del Grid computing global.

### **2.30 ¿Qué No Puede Hacer el Grid Computing?**

El Grid no es una bala color de plata que puede tomar cualquier aplicación y ejecutarla mil veces más rápido sin la necesidad de comprar más máquinas o software.

No toda aplicación es conveniente o capaz de ejecutarse en un Grid.

Algunos tipos de aplicaciones simplemente no pueden ser puestas en forma paralela.

En otros casos, puede tomar una gran cantidad de trabajo para modificarlos y así lograr un movimiento más rápido.

La configuración de un Grid puede que afecte la calidad, fiabilidad, y seguridad de una infraestructura de organización de computing.

Por todas estas razones, es importante entender cómo ha evolucionado el Grid hoy y qué características tendrá en un futuro distante.

## Capítulo 3

# Estándares Abiertos

Para entender el rol desempeñado por el Grid Toolbox de IBM, primeramente se tienen que entender ciertos factores y discutir sobre algunos de los componentes fundamentales de los que el producto depende.

### 3.1 Web Service: Servicios Web

Un Web service puede ser usado para construir una aplicación identificada por una dirección Uniform Resource Locator : Localizador Uniforme de Recursos (URL). Las interfaces y enlaces de los Web services pueden ser definidas, descritas y descubiertas por componentes Extensible Markup Language : Lenguaje Extensible de Marcas (XML) y pueden soportar interacciones directas con otras aplicaciones de software usando mensajes basados en XML vía protocolos basados en Internet. En términos simples, un Web service es una aplicación que se llama usando una dirección de Web, pasando los parámetros en formato XML.

Al usar XML, el Web Services Description Language : Lenguaje de Descripción de Servicios Web (WSDL) describe una red de servicios como una colección de puntos finales que operan por medio de mensajes que contienen información ya sea orientada al proceso u orientada al documento. Para definir un punto final, se describen abstractamente operaciones y mensajes y posteriormente se limitan a un protocolo de red establecido.

Análogamente los puntos finales descritos son agrupados en puntos fina-

les abstractos, normalmente llamado “servicios”. La funcionalidad clave de WSDL es permitir la descripción de productos finales y sus mensajes sin tener en cuenta los formatos de los mensajes o los protocolos de comunicación utilizados.

## 3.2 Grid Service: Servicios Grid

La tecnología de los Grid service está basada en la Service Oriented Architecture: Arquitectura Orientada a Servicios (SOA) que define una arquitectura donde una aplicación se constituye de componentes independientes y cooperadores llamados “servicios”. Esos servicios construyen los bloques que utiliza un modelo de objeto para crear sistemas distribuidos abiertos y habilitar a las compañías e individuos para que creen rápidamente y en forma global sus aplicaciones disponibles para la red. [1, Aguilar]

Los mecanismos adicionales para crear y administrar Servicios Grid son habilitados al desarrollar un servicio nuevo que será desplegado dentro de un sistema OGSA. Esos mecanismos son:

**Factory:** Fábrica: Es una clase especial para crear dinámicamente instancias de Servicios Grid, código de Servicios Grid ejecutables y esperar por requerimientos.

**Registry:** Registro: Es la interfaz que habilita un conjunto de instancias de Servicios Grid para registrar el Grid Service Handle: Manejador de Servicio Grid (GSH) dentro de un servicio de registro, que permita la identificación de servicios en ese conjunto.

**Discovery:** Descubrimiento: Es la interfaz que permite a los clientes del Servicio Grid obtener información acerca de los servicios proporcionados.

**Life cycle:** Ciclo de vida: Se refiere a los estados de las instancias de Servicios Grid entre su creación y destrucción.

**Service data:** Datos del servicio: Es la colección estructurada de información que se asocia con una instancia de Servicios Grid.

**Notification:** Notificación: Mecanismo por el cual una parte envía (origen de notificación) información de un cambio de estado a la parte (destino de notificación) que ha pedido ser notificada.

Reliable invocation: Invocación fiable: Técnicas que aseguran la fiabilidad de invocación de métodos en caso de que hayan sido creadas múltiples instancias con Servicios Grid redundantes en el espacio.

Lo importante a tener en cuenta es que el único contacto entre los Servicios Grid y sus usuarios es la interfaz de servicios. Esas interfaces de servicios son definidas por el Lenguaje de Descripción de Servicios web (WSDL) existente. Varias mejoras a WSDL han sido identificadas para requerimientos de OGSi y actualmente están siendo agregadas al estándar WSDL.

### **3.3 Arquitectura de Servicios de Grid Estándar (OGSA)**

El Foro Global de Grid fue formado para manejar las estandarizaciones en un Grid Computing.

La Open Grid Services Architecture: Arquitectura de Servicios de Grid Estándar (OGSA) del Foro Global de Grid representa una evolución hacia una arquitectura de sistemas basada en conceptos y tecnologías de Servicios Web.

Es importante destacar que OGSA es una arquitectura basada en los estándares existentes de Web service, y que también se utiliza para definir muchos estándares de grid.

Los estándares de Web service incluyen: XML, SOAP y WSDL.

### **3.4 Infraestructura de Servicios de Grid Estándar (OGSI)**

El Foro Global de Grid promueve el desarrollo de estándares para la infraestructura de un Grid Computing.

OGSI se refiere a la infraestructura base sobre la cual es construida la OGSA. En su núcleo se encuentran las especificaciones de Servicios Grid, que definen la interfaz estándar y conductas de un Servicio Grid, armando una base de Web service.

OGSI provee una estructura sobre la que se definen y construyen los estándares OGSA. Proporciona especificaciones técnicas para la implementación de cada componente de OGSA, usando Servicios Grid para definir cada interfaz. La especificación se basa en un grupo de Servicios Web estándar, con ciertas extensiones para WSDL y XML necesarias para los Servicios Grid.

OGSI define detalles tales como estabilidad de Servicios Web, la herencia de interfaces de Servicios Web, notificación asíncrona, referencias a instancias de servicios, colección de instancias de servicios y datos de estados de servicios.

El mundo de los Web services ha reconocido las mejoras significativas logradas para OGSA OGSI y el trabajo se encamina para incluir algunas de esas mejoras en los Web services mismos.

### 3.5 ¿Cuáles son los objetivos de OGSA?

#### Objetivos:

- Manejar recursos a través de plataformas heterogéneas distribuidas.
- Brindar quality of service: calidad de servicio (QoS). La topología de Grid es a menudo compleja. La interacción de recursos del Grid es normalmente dinámica. Es importante que el Grid proporcione servicios robustos ocultos, tales como autorización, control de acceso, y delegación.
- Proveer una base común para soluciones de administración autónomas. Un Grid puede contener muchos recursos, con numerosas combinaciones de configuraciones, interacciones, y estado cambiante y modos de fracaso.
- Definir las interfaces abiertas publicadas. La OGSA es una norma abierta manejada por el cuerpo de normas de GGF (Global Grid Forum). Para la interoperabilidad de diversos recursos, los Grid deben construirse en interfaces standard y protocolos.
- Aprovechar las industrias de tecnologías de integración standad. [17, Unger]

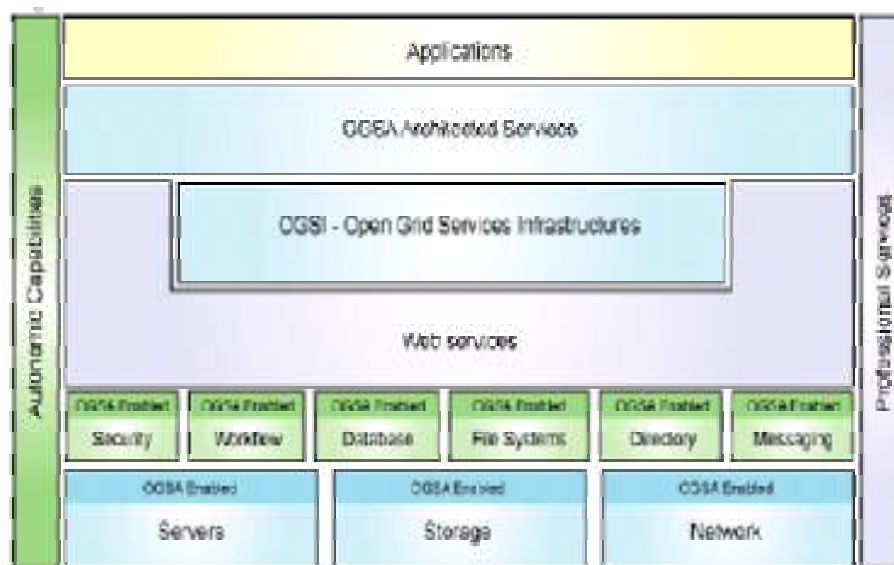


Figura 3.1: La Arquitectura de OGSA.

### 3.5.1 Arquitectura:

Cuatro capas principales comprenden la arquitectura de OGSA: Vea fig.3.1 de la pág.61.

Empezando del fondo, ellos son:

- Recursos—los recursos físicos y los recursos lógicos Web services más las extensiones de OGSI que definen servicios Grid.
- Servicios de arquitectura de OGSA
- Aplicaciones Grid.

Se debe observar estas capas, una a la vez.

#### Capa de los recursos físicos y lógicos..

El concepto de recursos es central para OGSA y Grid Computing en general. Los recursos comprenden las capacidades del Grid, y no son limitados a los procesadores.

Los recursos físicos incluyen servidores, almacenamiento, y red.

Sobre los recursos físicos están los recursos lógicos. Ellos proporcionan función adicional virtualizando y agregando los recursos en la capa física. El propósito general, del software intermedio tales como sistemas de archivos, gestores de bases de datos directorios, y gestores de flujos de trabajo (workflow) proporcionan estos servicios abstractos sobre el Grid físico.

#### **Capa de Web services.**

La segunda capa en la arquitectura de OGSA es la de Web services.

Aquí hay un importante principio de OGSA: Todos los recursos del Grid (lógico y físico) son modelados como servicios. La especificación Abierta de Infraestructura de Servicios de Grid (OGSI) define servicios de grid y construye sobre las tecnologías de Web services standad. OGSI aprovecha los mecanismos de Web services como XML y WSDL para especificar interfaces standads, conductas e interacción para todos los recursos del grid. OGSI extiende la definición de Web services para proporcionar capacidades para una Web services dinámica, estable y manejable que se exige para modelar los recursos del Grid.

#### **Capa Servicios de Grid de la Arquitectura de OGSA.**

La capa de Web services, con sus extensiones de OGSI, provee una infraestructura base a la próxima capa de arquitectura de Grid services. El Global Grid Forum está actualmente trabajando para definir muchos de estos servicios de Grid de arquitectura en áreas como la ejecución de programas, servicios de datos, y servicios centrales. Algunos ya se han definido, y algunas aplicaciones ya han aparecido. En tanto las aplicaciones de éstos servicios nuevos de arquitectura empiecen a aparecer, OGSA se volverá una arquitectura orientada al servicio más útil (SOA).

#### **Capa de Aplicaciones de Grid.**

Con el tiempo, en tanto una cantidad importante de servicios de arquitectura de Grid continúe desarrollándose, las nuevas aplicaciones del Grid que utilicen uno o más servicios de arquitectura de éste, aparecerán. Estas aplicaciones comprenden la cuarta capa principal de la arquitectura de OGSA.

Se puede observar más de cerca a los dos componentes lógicos principales de OGSA (los Web services) más capa de OGSI y la capa se servicios de arquitectura de OGSA. Ver fig.3.2 de la pág.63.



Figura 3.2: La Estructura de OGSA.

El grupo de trabajo de OGSA creyó que era necesario aumentar la funcionalidad de los Web services centrales para dirigir los requerimientos de servicios de Grid. El OGSI se extiende los Web services introduciendo interfaces y convenciones en dos áreas principales.

- Primero hay una naturaleza de servicios en un Grid dinámica y potencialmente transitoria. En un Grid, instancias particulares de servicios pueden venir y pueden ir en relación al trabajo que se envía, a los recursos que son configurados y provisionados, y a los cambios de estado de sistema. Por consiguiente, los servicios del Grid necesitan interfaces para manejar su creación, destrucción, y administración de ciclo de vida.
- Segundo, hay un estado. Los servicios del Grid pueden tener atributos y datos asociados con ellos. Esto es similar en concepto a la estructura tradicional de programación orientada a objetos. Los objetos tienen conducta y datos. Igualmente, los Web services tuvieron ser extendidos para soportar datos de estados asociados con servicios del Grid.

El OGSI presenta un modelo de interacción para los servicios del Grid. OGSI provee una manera uniforme a diseñadores de software para planear e interactuar con servicios del Grid al proveer interfaces para descubrimiento,



Figura 3.3: Componentes de OGISI.

ciclo de vida, estado, administración, creación y destrucción, notificación de evento, y administración de la referencia. Éstos se describen en la fig.3.3 de la pág.64. Si un diseñador de software está desarrollando un servicio de Grid o una aplicación, el modelo de programación de OGISI, provee un camino consistente para que el software del Grid interactúe.

**Infraestructura:** servicios de Grid que llevan a cabo esta interfase proporcionan una manera de crear nuevos servicios del Grid. La infraestructura puede crear casos temporales de función limitada, como un scheduler que crea un servicio para representar la ejecución de un trabajo particular, o también pueden crear servicios de más larga vida tales como una réplica local de un conjunto de datos frecuentemente usados. No todos los servicios de Grid se crean dinámicamente. Por ejemplo, algunos podrían ser creados como el resultado de una instancia de un recurso físico en el Grid tales como procesador, almacenamiento, o dispositivo de la red.

**Ciclo de vida:** porque los servicios del Grid pueden ser transitorios, las instancias de servicio de éste se crean con un tiempo de vida específico.

La vida de cualquier instancia de servicio particular puede negociarse y puede extenderse como requieran los componentes que son dependientes o manejan ese servicio. El mecanismo de ciclo de vida se construye para prevenir que servicios de Grid consuman recursos indefinidamente sin requerir en gran escala servicios distribuidos de "recolección de residuo".

**Administración de Estado:** los Grid services pueden tener estado. OGISI especifica un marco para representar este estado llamado Services Data y un mecanismo para inspeccionar o modificar lo que el estado llama Services Data Set Find.

Además, el OGSi requiere una mínima cantidad de estado en Service Data Elements que cada Grid service debe soportar, y requiere que todos los servicios implementen el Find/SetService Data portType.

**Grupos de servicios:** son grupos de servicios de Grid que son incluidos, usando el Service Data, para algún propósito en particular. Por ejemplo, podrían ser usados para recolectar todos los servicios que representan los recursos en un nodo-cluster particular dentro del Grid.

**Notificación:** la información de estado (Service Data) que se modela para servicios Grid cambia cuando el sistema se ejecuta. Muchas interacciones entre los servicios Grid requieren un monitoreo dinámico de estado cambiante. La notificación aplica un paradigma tradicional publish/subscribe para este monitoreo.

Los servicios Grid soportan una interfase (NotificacionSource) para permitir que otros servicios Grid (NotificacionSink) se suscriban a los cambios.

**Mapeo:** cuando las infraestructuras se usan para manejar una nueva instancia de servicios Grid, la infraestructura devuelve la identidad de los servicios instanciados recientemente. Esta identidad se compone de dos partes, un Grid Service Handle (GSH) y un Grid Service Reference (GSR). Se garantiza un GSH para referenciar el servicio Grid indefinidamente, mientras un GSR puede cambiar dentro del tiempo de vida del servicio Grid. La interfase de mapeo provee una manera de obtener un GSR dado un GSH. Esto podría parecer simple pero, hay un grupo de problemas asociados con tal consulta.

Como se observó, la arquitectura del OGSA permite a los Web Services acomodar los requerimientos del Grid de mejor manera.

Estas mejoras son especificadas en el OGSi. Cuando las especificaciones del OGSi finalicen y las implementaciones comiencen a aparecer, algunas organizaciones de estándares comenzaran a interesarse en incorporar la mayoría de las funcionalidades bosquejadas en el OGSi dentro de un apropiados Web Services standard. Esto es razonable. Un grupo de aquello que el OGSi trata en algún aspecto, no es propio del Grid computing, pero se requiere para construir arquitecturas robustas orientadas al servicio. Con el tiempo se espera que muchas de las funcionalidades del OGSi se incorporen a los Web Services standard. Esto se muestra en la fig.3.4 de la pág.66, la cual lista varios Web Services standards emergentes que podrían incorporar el OGSi. En la misma figura se hace referencia a las mejoras de Web Services como extensiones de Web services.



Figura 3.4: El OGSi y Web service.

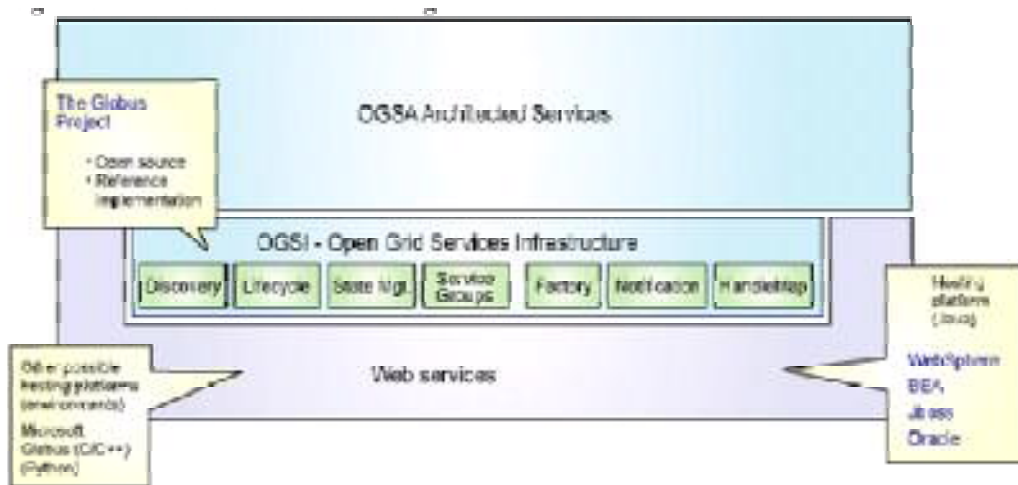


Figura 3.5: El OGSi y el Hosting de Web service.

El Globus Toolkit 3 es la primer implementación en gran escala del standard OGSi. El toolkit fue desarrollado por el proyecto Globus, una investigación y un proyecto de desarrollo focalizado en permitir la aplicación de conceptos de Grid computing a ingeniería, ciencia y comercio.

El toolkit fue escrito en lenguaje Java usando el entorno J2EE. Como el centro de la arquitectura de Grid services el OGSi necesita ser alojado en una plataforma de entrega que soporte a los Web services.

Aunque el OGSi fue hecho en código Java y alojado en un ambiente de tiempo de ejecución J2EE, nada impide que el OGSi sea implementado en otro lenguaje de programación y alojado en otros ambientes. De hecho, para que crezca la aceptación del OGSa el OGSi deberá ser habilitado en plataforma de alojamiento múltiple.

### 3.6 ¿Qué plataformas?

En la fig.3.5de la pág.66, se observa que una implementación Java de OGSi puede ser potencialmente alojada en cualquiera de los ambientes J2EE tales como JBOSS, WebSphere, o BEA Weblogic. Esta es una de las diferentes ventajas de implementar el OGSi (y para esa cuestión cualquier software) en la tecnología Java.

Sin embargo, plataformas alternativas como un entorno tradicional C, C++, o C# y Microsoft .Net son factibles de alojar ambientes para el OGSi. Ya hay implementaciones iniciales para la ejecución del OGSi en otros ambientes, incluyendo C#/.Net y Python.

Se espera que muchas de las implementaciones del OGSi sean enviadas vía modelo de desarrollo de fuente abierta y que las implementaciones de referencia existentes (el Globus Toolkit 3) sean usadas sin modificarse en entornos apropiados de alojamiento. Idealmente, un pequeño número de implementaciones centrales del OGSi (una por plataforma hosting) será conjuntamente desarrollada con la industria y usada en muchos productos. [7, Haynos]

El OGSi es ciertamente un paso importante en el desarrollo de una arquitectura orientada al servicio para Grid.

Sin embargo, para que aplicaciones útiles sean desarrolladas, un importante grupo de servicios Grid (los servicios de arquitectura de OGSa) necesitará

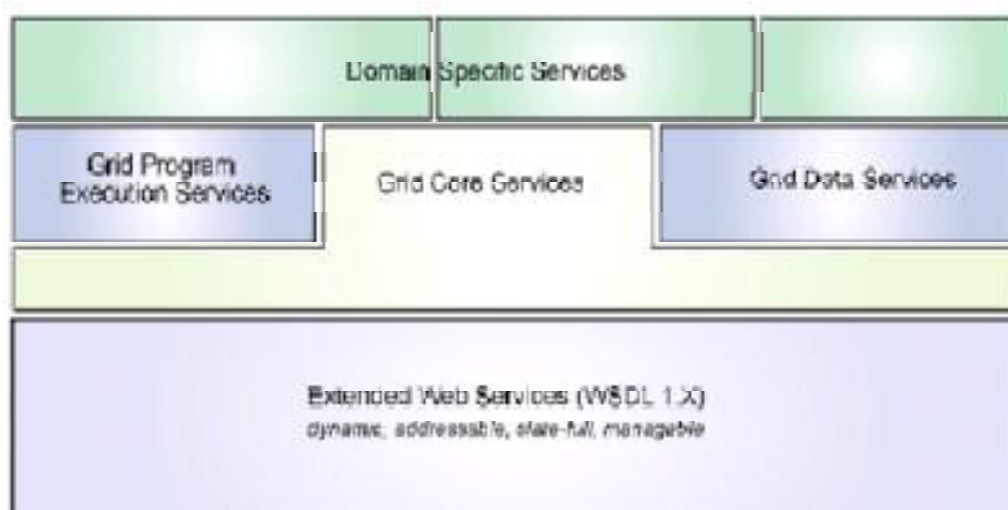


Figura 3.6: La Estructura de la Arquitectura de servicio de OGSA.

ser implementado y distribuido por iniciativas de fuente abierta como el proyecto Globus y por compañías de software de middleware. En este sentido, el OGSI y las extensiones que él provee para Web Services son necesarias pero insuficientes para la maduración de la arquitectura orientada al servicio.

En la fig.3.6 de la pág.68, además se dividen los servicios de arquitectura de Grid en cuatro categorías:

- Servicios centrales de Grid.
- Servicios de ejecución de programa de Grid.
- Servicios de datos de Grid.
- Servicios específicos de dominio.

Las primeras tres categorías representan áreas de trabajo activo por investigación de GGF o grupos de trabajo.

Con el tiempo, cuando estos servicios maduren, podrán ser especificados servicios de dominio, los cuales harán uso de la funcionalidad que éstos servicios provean. Hoy es importante que los grupos de trabajo GGF se concentren

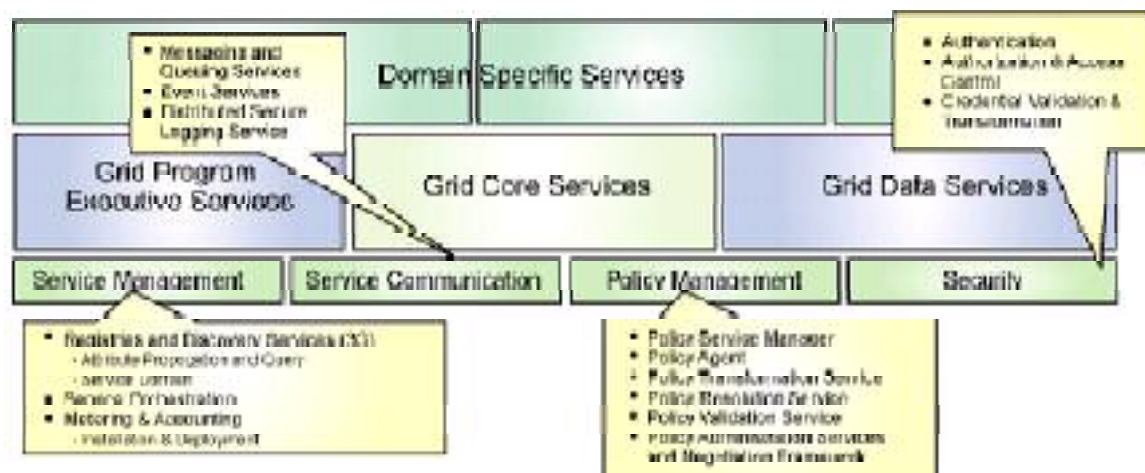


Figura 3.7: Servicio de núcleo de Grid.

en especificar un amplio conjunto de servicios de Grid útiles que los vendedores de software y diseñadores podrán luego comenzar a implementar.

Los servicios centrales de Grid se componen de cuatro tipos de servicios principales. Ver fig.3.7 de la pág.69.

- Administración de Servicio.
- Comunicación de Servicio.
- Administración de Políticas.
- Servicio de Seguridad.

A diferencia de funciones del OGIS que están ampliamente implementadas como extensiones para los protocolos de Web services básicos y un modelo de interacción, estos servicios centrales son realmente implementados como servicios Grid (en la base del OGIS).

Estos servicios se consideran centrales primeramente porque se espera que sean ampliamente explotados por la mayoría (sino todos) de los servicios de alto nivel implementados además para soportar la ejecución de un programa, o acceso de datos, o como servicios de dominio específico.

**Administración de Servicio:** provee funciones que administran los servicios de despliegue en el Grid distribuido. Éste automatiza y asiste con una variedad de tareas de instalación, mantenimiento, monitoreo y de búsqueda de error dentro de un sistema de Grid. Éste incluye funciones para provisionar y desplegar los componentes del sistema. También incluye funciones para recolectar e intercambiar datos acerca de la operación del Grid.

Estos datos son útiles para operaciones de manejo online y offline, e incluyen información acerca de errores, eventos, determinación de un problema, auditoría, medición, contabilidad y pago.

**Comunicación de servicio:** incluye un grupo de funciones que soportan los métodos básicos para que se comuniquen los Grid services. Soportan varios modelos de comunicación que pueden ser combinados para permitir una efectiva comunicación entre servicios, incluyendo mensajes encolados, notificación de eventos de suscripción y publicación, notificación y registro (logging) de distribución confiable.

**Servicios de Política:** crea un marco general para creación, administración y conducción de políticas y acuerdos para operación de sistemas. Estos incluyen seguridad de gobierno de políticas, asignación de recursos y performance, también una infraestructura para servicios de “consideración de política” para usar políticas para gobernar su operación. Documentos de política y acuerdo proveen un mecanismo para la representación y negociación de términos entre los proveedores de servicios y sus clientes (ya sea consulta de usuarios u otros servicios). Estos términos incluyen especificaciones, requerimientos y objetivos para función, performance, y calidad que los proveedores y consumidores intercambian y que luego usan para influenciar sus interacciones.

**Servicios de Seguridad:** soportan, integran y unifican modelos de seguridad populares, mecanismos, protocolos y una variedad de tecnologías de una manera que posibilita a una variedad de sistemas interoperar de forma segura. Estos servicios de seguridad capacitan y extienden enlaces y protocolos de seguridad de Web services central y proveen mecanismos orientados al servicio para autenticación, autorización, refuerzo de confianza de la política, transformación de credencial y cosas como éstas.

**Muestra dos importantes clases de Grid services.** Vea la fig.3.8de la pág.71.

- Servicios de ejecución de programas de Grid.

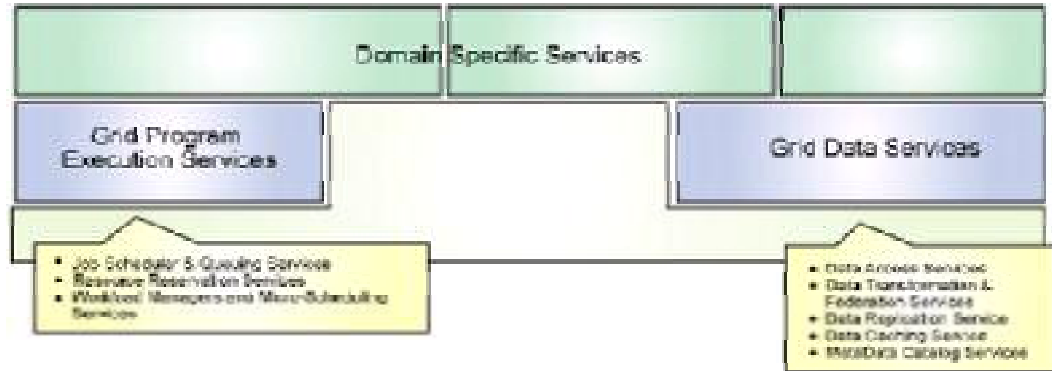


Figura 3.8: Ejecución de Programas de Grid y Data Service.

- Grid Data services.

**Servicios de ejecución de programa de grid:** mientras el OGSi y los servicios centrales de Grid son generalmente aplicables a cualquier sistema de computación distribuido, la clase de ejecución de programas de Grid es única, en el modelo de Grid de ejecución de tareas distribuidas, que soporta la computación de alta performance, paralelismo, y colaboración distribuida. Los principios de programación de trabajos y de administración de carga implementados a esta clase de servicios y la habilidad para virtualizar recursos en proceso, son centrales al Grid computing.

**Grid data services:** para complementar las convenciones de virtualización de cómputo especificada por los servicios de ejecución de programas, existen los servicios que forman grid data services.

Estas interfases soportan el concepto de virtualización de datos y provee mecanismos relacionados con el acceso distribuido a la información de todo tipo incluyendo bases de datos, archivos, documentos, almacén de contenidos y corrientes de datos generados por las aplicaciones. Grid data services explotará y virtualizará datos usando métodos de ubicación, replicación de datos, caching y movimientos de datos de alta performance para brindar aplicaciones que requieran Calidad de Servicio (QoS) y acceso a través del Grid distribuido. Métodos para asociar múltiples tipos de datos dispares, como también recursos de datos distribuidos, pueden proveer la integración de datos almacenados bajo diferentes esquemas tales como archivos y bases de datos relacionales.

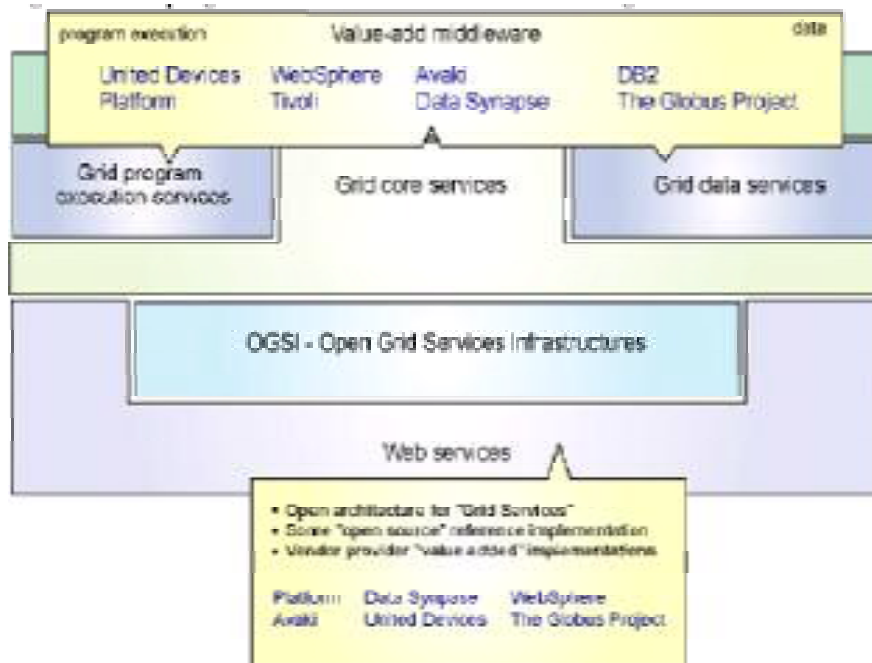


Figura 3.9: La Ejecución de Programas de Grid y Data Service Hosting.

En esta categoría es evidente que la OGSA localiza a los recursos de datos de manera equivalente a los recursos de cómputo.

Los vendedores probablemente no competirán al ofrecer un amplio rango de implementaciones de OGSI. En lugar de eso como parte del “**desarrollo**” de las implementaciones de Web services, los vendedores, quienes ofrecerán estas implementaciones, también usarán directamente las implementaciones de fuente abierta existentes provistas por organizaciones como Globos, o integrarán las implementaciones con sus productos de plataforma hosting como WebSphere, WebLogig, Apache o Net.

Sin embargo, los servicios de arquitectura de Grid proveen algunas áreas naturales preparadas oportunamente para vendedores y organizaciones, para competir y diferenciarse entre ellos. Esta competencia creará una “economía” de proveedores de software de Grid cuya innovación ayudará a orientar la aceptación de standards como OGSI/OGSA, y esto permitirá a los clientes construir sistemas fuera de los componentes interoperables. Además, las áreas

de funcionalidad en la ejecución de programas de Grid y servicios de datos requerirá innovación y nuevos planteamientos, y esto acelerará la aceptación del mercado de soluciones de Grid y proveerá oportunidades de mercado a los vendedores.

En la fig.3.9 de la pág.72 se observa que los servicios centrales del Grid pueden ver una mezcla de implementaciones de referencia de fuente abierta e implementaciones de "valor agregado" provistas por el vendedor (proveedor). Muchas tecnologías en esta área pueden ser comoditizadas, pero áreas como política y seguridad proveerán a los vendedores una oportunidad para diferenciarse entre ellos.

Implementaciones en ambos, ejecución de programas de Grid y data services, se espera que formen parte de implementaciones de valor agregado de diferentes compañías. Estas áreas representan oportunidades importantes para integrar ofertas de middleware líderes dentro del marco de OGSA y permitirán un buen "ecosistema" para desarrollar soluciones de Grid.

### **3.7 Conclusión.**

Se han presentado y descrito los componentes de la estructura de OGSA. Con la entrega de las implementaciones iniciales de OGSI, OGSA se prepara para acelerar su entrada en la principal corriente comercial de computación. Si las iniciativas se refieren a organic computing (computación orgánica), on-demand computing (computación bajo demanda) o adaptive computing (computación adaptativa), es necesario un standard comprensivo y abierto como el OGSA (construido en tecnología standard) para realizar computación heterogénea distribuida en el mundo comercial.



# Bibliografía

- [1] L. Joyanes Aguilar. *Cibersociedad*. Mac Graw-Hill, 1997.
- [2] Viktors Berstis. *Fundamentals of Grid Computing*. IBM, USA, 2002.
- [3] James A. O'Brien. *Sistemas de Información Gerencial*. Editorial Nomos S.A., Argentina, 2003.
- [4] Arun K. Iyengar Marcos Novaes Li Zhang Catherine H. Crawford, Daniel M. Dias. *Commercial Applications of Grid Computing*. IBM, USA, 2003.
- [5] H.Ñimrod D. Abramson, R. Giddy Sasic. *The Grid: Blueprint for a New Computing Infrastructure*. IEEE, USA, 1995.
- [6] A. González del Alba Baraja; V. Yague Galaup; L. Joyanes Aguilar. *Impacto de las Tecnologías en la Gestión de los Sistemas de Información en II Congreso Internacional de Sociedad de la Información y del Conocimiento*. McGraw Hill, Madrid-España, 2003.
- [7] Matt Haynos. *The Physiology of the Grid: A visual of Open Grid Services Architecture*. IBM, 2003.
- [8] C. Kesselman I. Foster. *The Grid: Blueprint for a New Computing Infrastructure*. 1998.
- [9] J.Ñick S. Tuecke I. Foster, C. Kesselman. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Global Grid Forum, USA, 2002.
- [10] R. Gutierrez I. Foster B. Ginsburg O. Larsson I. Lopez, G. Follen. *Using CORBA and Globus to Coordinate Multidisciplinary Aerospace Applications*. NASA, USA, 2000.
- [11] T. Durniak P. Herman J. Karuturi C. Woods C. Gilman J. Barry, M. Aparicio. *Enterprise Distributed Computing Workshop*. IEEE, USA, 1998.

- [12] H. Bivens S. Humphreys R. Rhea J. Beiriger, W. Johnson. *Constructing the ASCI Grid*. IEEE, USA, 2000.
- [13] Q. Peng D.P. Schissel M. Thompson I. Foster M. Greenwald D. McCune K. Keahey, T. Fredian. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. 2002.
- [14] MIT. *Technology Review*. MIT, USA, 2004.
- [15] A. Keahey K. Kohn S. McInnes S. Parker R. Armstrong, D. Geist Gannon. *The Grid: Blueprint for a New Computing Infrastructure*. IEEE, USA, 1999.
- [16] I. Foster C. Kesselman S. Tuecke J. Volmer V. Welch R. Butle, D. Engert. *Design and Deployment of a National - Scale Authentication Infrastructure*. IEEE, USA, 2000.
- [17] Jay Unger. *The Physiology of the Grid:A visual of Open Grid Services Architecture*. IBM, 2003.
- [18] A. Skjellum W. Gropp, E. Lusk. *Portable Parallel Programming with the Message Passing Interface*. IEEE, USA, 1994.
- [19] Colin J. White. *IBM Enterprise Analytics for the Intelligent e-Business*. IBM Press, USA, 2001.

# Índice de Materias

- AFS, 26
- Aplicaciones del Grid Computing
  - Entornos Virtuales, 6
  - Proceso Intensivo de Datos, 6
  - Servicios Puntuales, 6
  - Sistemas Distribuidos en Tiempo Real, 6
  - Supercomputación Distribuida, 4
- Aplicaciones y Servicios, 11
  - GridSystems, 12
  - HP, 11
  - Sun, 11
- bubbled up, 38
- certificado de autoridad, 51
- clases de grid services, 70
  - grid data service, 71
  - servicios de ejecución de programas de grid, 71
- concepos y componentes
  - trabajos y aplicaciones, 30
- conceptos y componentes, 24
  - almacenamientos, 26
  - Comunicaciones, 28
  - comunicaciones
    - externa, 28
    - interna, 28
  - equipos, capacidades, arquitecturas y políticas, 29
  - software y licencias, 28
  - tipos, 25
    - computación, 25
  - configuración de datos, 46
  - cpu paralela, 17
    - algoritmos, 17
    - primer barrera, 18
    - segunda barrera, 18
- Data Striping, 27
- DFS, 26
- escalabilidad, 25
- estándares, 57
  - ciclo de vida, 58
  - descubrimiento, 58
  - fábrica, 58
  - Grid Services, 58
  - invocación fiable, 59
  - notificación, 58
  - OGSA, 59
  - OGSI, 59
  - registro, 58
  - Web Services, 57
- find/setservice data porttype, 65
- FTP, 50
- Globus Project, 2
- GPFS, 26
- Grid Computing, 1
  - Arquitectura, 8
  - Conectividad, 9
  - Infraestructura, 9

- Recurso, 10
- Recursos, 11
- Beneficios, 1
- concepto, 1
- grid computing
  - componentes, 36
    - administración del grid distribuido, 39
    - comunicaciones, 41
    - de administración, 37
    - observación, dirección y medición, 41
    - schedulers, 40
    - software de sumisión, 39
    - software servidor, 37
  - construcción, 35
- grid de datos
  - capacidades, 19
- grid service handle, 65
- grid service reference, 65
- GUI, 44
- heurísticas, 32
- intragrid a intergrid, 32
  - compatimiento de archivos, 32
- IT, 23
- j2ee, 67
- monitoreo del progreso y recuperación, 47
- MPI, 41
- NFS, 26
- OGSA, 55, 58
- ogsa, 59
  - arquitectura
    - capa de web services, 62
    - arquitectura, 61
    - capa de aplicaciones, 62
    - capa de recursos lógicos y físicos, 61
    - capa de servicios, 62
    - objetivos, 60
- ogsi, 59
  - interfase
    - administración, 64
    - ciclo de vida, 64
    - grupo de servicio, 65
    - infraestructura, 64
    - mapeo, 65
    - notificación, 65
  - interfases, 63
- organizaciones y recursos virtuales para colaboración, 19
- perspectiva del administrador, 49
- tilde nador, 54
- perspectivas de usuario, 42
- perspectivas del administrador
  - instalación, 49
  - planeación, 49
- planificación del despliegue, 35
  - organización, 36
  - seguridad, 35
- puggability, 22
- recursos, 25
- recursos adicionales, 20
- registrarse en el grid, 43
- reservación, 31
- sandbox, 45
- scavening, 31
- servicios centrales del grid, 69
  - administración, 70
  - comunicación, 70
  - política, 70
  - seguridad, 70
- servise data set find, 64

VPN tunneling, 33

WSDL, 58

XML, 57

