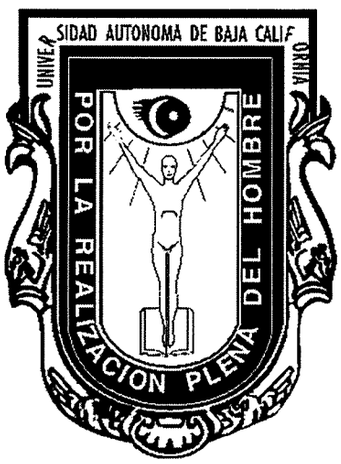
UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA



IMPLEMENTACIÓN DE LA TECNOLOGÍA GRID EN LA RED-CICESE

Tesis Profesional

Que para obtener el Título de:

Ingeniero en Computación

Presenta:

JOSE ELENO LOZANO RIZK

Ensenada, Baja California **Noviembre del 2003**

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE LA TECNOLOGÍA GRID EN LA RED-CICESE

T E S I S P R O F E S I O N A L

QUE PRESENTA:

JOSE ELENO LOZANO RIZK

APROBADO POR:

Ing. Salvador Castañeda Ávila

M.C. Raúl Tamayo Fernández

M.C. Mabel Vazquez Briseño

DEDICATORIA

A mis padres Jose Eleno Lozano y Armida Rizk por todo el apoyo que me han dado en todas mis decisiones.

A mis hermanos Arturo y Jorge por todo el cariño que les tengo. A mis abuelos Eleno y Guadalupe por todos sus buenos consejos.

A mi novia Mayred por todo su amor, cariño y apoyo que me ha brindado.

AGRADECIMIENTOS

El autor desea agradecer de manera muy especial al L.C.C. Julián Delgado Jiménez, por todo su tiempo invertido, su paciencia, orientación y el apoyo brindado en la realización de ésta tesis.

A mi asesor de tesis, Ing. Salvador Castañeda, por haberme dado la oportunidad de trabajar con él en este proyecto, además de sus aportaciones y consejos en la realización de ésta.

A mi asesor de tesis, M.C. Raúl Tamayo, por sus grandes aportaciones en el desarrollo de ésta.

A mi sinodal M.C. Mabel Vázquez Briseño por su ayuda brindada en la finalización de este trabajo de investigación.

Al personal del Departamento de Cómputo de la Dirección de Telemática del CICESE agradezco el apoyo brindado en la realización de este trabajo.

A mis grandes amigos Agustín Pérez y Orthon Reynoso por todo el apoyo brindado a lo largo de la carrera.

CONTENIDO

CONTENIDO (continuación)

LISTA DE FIGURAS

LISTA DE FIGURAS (continuación)

LISTA DE TABLAS

I. INTRODUCCIÓN

I.1 Antecedentes.

En estos tiempos, donde las aplicaciones del mundo real son más sofisticadas y los recursos computacionales son más exigidos para llevar a cabo complejas aplicaciones, es necesario disponer de una gran capacidad de cómputo como la que ofrece una supercomputadora o un cluster (racimo de computadoras), sin embargo en muchos de los casos esta capacidad no es suficiente, por tanto, es indispensable aprovechar los recursos computacionales accesibles que se encuentren en otras partes del mundo.

Conforme las tecnologías de redes y conectividad se hacen más robustas, más eficientes y más rápidas, la globalización de estas capacidades de cómputo es posible. Este hecho de globalización de computadoras de una forma u otra implica una topología de redes de computadoras interconectadas en sitios dispersos.

Cuando hace apenas unos años era sorprendente el poder realizar pequeños cálculos en la PC (contabilidad doméstica, hojas de cálculo, etc.), ahora parece insuficiente aun si las más sofisticadas aplicaciones no ofrecen sus resultados en cuestión de segundos. Este cambio en las exigencias tecnológicas ha surgido de lado al desarrollo mismo de la tecnología. Sectores académico, industriales y de negocios necesitan una potencia de cálculo cada día mayor para poder manejar el volumen de información que generan con la suficiente rapidez para ofrecer un servicio eficiente, rápido y de calidad, y por tanto para muchos de los casos vitales para la toma óptima de decisiones.

Actualmente, las soluciones tecnológicas convencionales se basan en el uso del cómputo de alto rendimiento (supercomputadoras), si bien es cierto, ésta solución es capaz de satisfacer en mayor o menor medida la necesidad de potencia de cálculo, no es menos

cierto que presenta una serie de problemas: una costosa inversión inicial del equipo, difícil y costoso mantenimiento, contratos ventajosos por parte de los proveedores, necesidad de contratación de personal especializado, y la poca o nula tolerancia a fallos, etc.

Pero de entre todas estas desventajas, la falta de escalabilidad es la que mayores problemas ha ocasionado a las entidades, que a la larga, cuando llegan a un punto de bloqueo tecnológico, se ven irremediablemente obligadas a un cambio total en la infraestructura informática de sus departamentos, es decir, a realizar cambios estructurales.

Recientemente, la creación del cluster tipo Beowulf ha sido parte de la solución, pero ello no está exento de la misma problemática que lo anterior. La aparición a principios del año 2000 de la **tecnología GRID** está cambiando esta situación, ya que en base a su concepción, permite ver un horizonte más preciso del poder cuando se habla de cúmulo en cómputo.

La posibilidad de unir todos los recursos informáticos de la red corporativa de un centro de investigación para emplearlos como una supercomputadora virtual, está permitiendo suplir las deficiencias de las supercomputadoras tradicionales.

El presente trabajo, esta basado en una investigación y estudio de la tecnología GRID y su implementación en el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) en su red de computación conocida como Red-CICESE, dando comienzo a futuras investigaciones sobre el tema.

I.2 Objetivos.

I.2.1 Objetivo general.

El objetivo principal de esta tesis es realizar una propuesta e implementación de la tecnología GRID en la Red-CICESE.

I.2.2 Objetivos específicos:

- Realizar un análisis y estudio de cómputo en redes globales.
- Establecer un modelo ideal de la tecnología GRID que se ajuste a las necesidades y perspectivas de la Red-CICESE.
- Implementar un ambiente GRID para la Red-CICESE.
- Adquirir experiencia con la instalación, programación y configuración del sistema GRID en la supercomputadora Sun Fire 4800 "calafia" y el cluster de estaciones de trabajo Sun Blade 1000 "tribus".
- Establecer las bases de futuras investigaciones y aplicaciones en un contexto global, donde instituciones que hayan adoptado esta tecnología, permitan conjuntar los recursos de cómputo y así mejorar el desempeño del sistema GRID.

I.3 Alcances y limitaciones.

En esta investigación, se da a conocer al lector, que hoy en día se cuenta con una gran opción para la unificación de recursos computacionales y de esta manera aprovechar al máximo los recursos con los que se cuente.

La tecnología Grid puede ser implementada para unificar los recursos computacionales de empresas que se encuentren en diferentes partes del mundo, así como los recursos computacionales que se encuentren en los diferentes departamentos de una misma empresa y de igual forma los recursos computacionales que se encuentren dentro de una misma área. En esta tesis, se describe el proceso de implementación de un Grid departamental, unificando los recursos de supercómputo que se encuentran en el Departamento de Cómputo de la Dirección de Telemática del CICESE.

En esta tesis, no se abordará a fondo la evolución que ha tenido el supercómputo a través del tiempo, ni tampoco que equipo de supercómputo que existe en el mercado es mejor o barato, incluyendo el software que emplean, sino que se da más énfasis al estudio de la tecnología Grid, a su arquitectura, su implementación, sus ventajas, y los retos que conllevan a esta nueva estrategia que parece ser la nueva tendencia para el cómputo de alto desempeño (HPC) a nivel mundial.

I.4 Organización de la tesis.

En el capítulo II se describen las diferentes tecnologías de cómputo de alto rendimiento que existen hoy en día, así como la historia del supercómputo, su arquitectura, algunos ejemplos de supercomputadoras, clusters y las tendencias del cómputo de alto rendimiento a nivel mundial.

En el capítulo III se realiza un estudio de la tecnología grid. Este estudio comprende conceptos, beneficios, arquitectura, topologías, software y aplicaciones de esta tecnología.

En el capítulo IV se da una breve historia de lo que ha sido el supercómputo en el CICESE, la situación actual del equipo de supercómputo, así como la propuesta e implementación de la tecnología grid en la Red-CICESE. Además, se describe el software que se utilizó para la unificación de los recursos computacionales.

El capitulo V da a conocer el resultado de las pruebas que fueron utilizadas para verificar el funcionamiento del Grid-CICESE.

En el capitulo VI se tienen las conclusiones referentes a esta tesis.

II. TECNOLOGÍAS DE CÓMPUTO DE ALTO RENDIMIENTO

II.1 Fundamentos.

El supercómputo es un término genérico que se entiende como el empleo de computadoras con una arquitectura sustancialmente diferente de la de los sistemas convencionales [Tecnova 99]. La finalidad de esta tecnología es servir como una herramienta de propósito general para apoyar a un sin número de sectores con el propósito de realizar la investigación científica en cálculos numéricos complejos, simulaciones avanzadas, en el manejo de un gran volumen de datos, y la necesidad de respuesta inmediata de resultados, que en equipo convencional no se podría llevar a cabo.

Las aplicaciones del supercómputo se pueden encontrar en una amplia gama de disciplinas, tales como la predicción del clima, modelado de la biósfera, exploración petrolera, procesamiento de imágenes, fusión nuclear, modelado de océanos, y muchas otras, es decir, existe un segmento importante del mercado mundial que demanda el uso de estos equipos de alto desempeño.

El actor principal clásico del supercómputo es la supercomputadora cuyo término es frecuentemente asociado con alta velocidad y potencia, alto costo y con cómputo paralelo. Conforme ideas nuevas e investigaciones en las Tecnologías de la Información han estado arribando nuevas concepciones han aparecido.

Una definición simple y sencilla sería una computadora más potente y más rápida que está disponible en cierto tiempo. Esta definición es dependiente del tiempo y puede provocar confusión con las actuales computadoras personales que son mas veloces que las supercomputadoras de años atrás [Leru 93]. Pero en si, el término y la finalidad de la supercomputadora, es independiente del tiempo. Y algo que es inherente a ésta, es la

capacidad de ellas para implementar paralelismo en una manera u otra para incrementar el desempeño de una aplicación.

Según la Real Academia Española una computadora se define como:

1. Adj. Que computa (|| calcula). U. t. c. s. 2. m. calculador (|| aparato que obtiene el resultado de cálculos matemáticos). 4. f. calculadora (|| aparato que obtiene el resultado de cálculos matemáticos). 5. f. computadora electrónica, máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la utilización automática de programas informáticos. Computadora personal, computadora electrónica de dimensiones reducidas, con limitaciones de capacidad de memoria y velocidad, pero con total autonomía.

El término de supercomputadora según el Diccionario Académico de Ciencia y Tecnología, se define:

1. Alguna categoría de computadora extremadamente poderosa y de larga capacidad para manipular cantidades masivas de datos extremadamente en un corto periodo de tiempo.

2. Alguna computadora que es una de las más grandes, rápidas y poderosas que existe en un tiempo dado [Morr 92].

II.1.1 Historia.

El supercómputo tuvo sus orígenes en la década de los 70's, asociado básicamente a dos nombres de empresas, Control Data Corporation (CDC) y Cray Research, y al nombre de una personalidad única: Seymour Cray, brillante arquitecto de supercomputadoras, principal fundador de ambas compañías, CDC en 1957 y Cray Research en 1972 [Tecnova 99].

Las primeras computadoras fueron diseñadas con tecnología de tubo de vacío (por Ej: Colossus, computadora británica usada en la 2a. Guerra Mundial para descifrar código alemán), cuya limitación consistía en la lenta velocidad de procesamiento de los relays electromecánicos y la pobre disipación de calor de los amplificadores. Los tubos fueron reemplazados por transistores los cuales eran más rápidos, más pequeños, además producían menos calor, y sirvieron para construir computadoras más confiables y de propósito general [Earl 03]. La figura de Seymour Cray surge en el hecho del interés propio de construir la computadora mas potente de esa época usando transistores, triunfo que consiguió con su equipo de 30 colaboradores en el año 1964, con el CDC 6600, considerada por muchos como la primera supercomputadora comercial que batió ampliamente en capacidad de cálculo y en costo a la computadora más potente de que disponía IBM en aquella época. La CDC 6600 fue capaz de ejecutar aproximadamente 9 Mflops (millones de operaciones de punto flotante por segundo). Posteriormente en 1969, CDC ofreció la CDC 7600, la cual ejecutaba aproximadamente 40 Mflops.

Muchos problemas en la ciencia y la ingeniería se basan en resolver largas ecuaciones matemáticas. Debido esa necesidad de resolución de grandes problemas de cálculo numérico, surge en la mitad de los 70s, supercomputadoras con procesamiento vectorial con las cuales se obtenía una importante ganancia de velocidad sobre las arquitecturas que eran basadas solo en procesamiento en línea de instrucciones (instruction pipelining –

empipado-), esta característica se conservó en los procesadores vectoriales pero se agregó la capacidad de ejecutar instrucciones en el cual los operándos pueden ser arreglos y no únicamente un escalar.

De nuevo la figura de Seymour Cray aparece en la historia de supercómputo, con la aportación en el año de 1976 de la primera supercomputadora vectorial comercial: la Cray-1 con un desempeño de 133 Mflops. Así, las supercomputadoras con procesadores vectoriales se convierten en una arquitectura típica durante los 20 años próximos al surgimiento de la Cray-1 [Earl 03].

En 1982, se diseña la primera supercomputadora con múltiples procesadores, la Cray X-MP que además tenía la característica de poseer memoria compartida y recursos de entrada/salida; por lo que, a mediados de los 80's, Cray controlaba el 70% del mercado del supercómputo [Tecnova 99]. En 1988, Cray lanza la supercomputadora vectorial Y-MP con 8 procesadores y con capacidad total de 2.6 Gflops, la primera en superar la línea de 1 Gflop entre todos sus procesadores. En 1992, la Cray C90 se convierte en la primera con procesadores vectoriales que alcanzan individualmente la velocidad de 1 Gflops. Sin embargo, debido a la poca escalabilidad de las supercomputadoras vectoriales en el número de CPUs (hasta 32 procesadores como la T90), al elemento cuello de botella, al precio y el pobre desempeño en el procesamiento de instrucciones complejas, surge a partir de los años 90, supercomputadoras escalables con procesadores escalares y de memoria distribuida conocidas como sistemas de procesamiento masivamente paralelo (MPP - Massively Parallel Processing) y su contraparte, sistemas con multiprocesamiento simetrico (SMP - Symmetric Multi-Processing) de memoria compartida, aunado al refinamiento y creación de instrucciones especiales como ocurre en los procesadores RISC.

En 1993, Cray lanza su primera supercomputadora MPP llamada T3D liderando el mercado de esta arquitectura que era dominado por Thinking Machine y MasPar [Cray 03].

Basados en esta idea de MPP, surgen los clusters o racimos de PC's para efectuar cómputo paralelo como una alternativa al alto costo de las supercomputadoras (Proyecto Beowulf 1994). [Beowulf 03]

En 1996 Silicon Graphics adquiere parte de Cray Research y lanza al mercado uno de los productos mas destacables: La Origin 2000, la cual presenta una innovadora arquitectura de memoria lógicamente compartida y físicamente distribuida que facilitó la programación que en otras supercomputadoras MPP como la T3E, era un verdadero problema [Origin 97]. Otra parte de esta compañía fue adquirida por Sun Microsystems cuyos planes era, en corto plazo, entrar al mercado de cómputo de alto desempeño. En 1997 ASCI Red desarrollada por Intel Corporation y Sandia National Laboratories se convierte en la primera supercomputadora MPP en alcanzar un Teraflops (Tflops) usando procesadores Pentium Pro que incorporaban ya aplicaciones para procesamiento de imágenes. En 1999 la ASCI Blue-Pacific alcanza los 3 Tflops.

En la historia del supercómputo se puede observar que hay un diseño dinámico de supercomputadoras adoptando lo mejor de cada arquitectura y buscando la manera de superar a la predecesora en capacidad de procesamiento de operaciones de punto flotante.

En México parte de la historia del supercómputo lleva por parte a la SHCP (Secretaria de Hacienda y Crédito Publico) donde disponían de las supercomputadoras de IBM en los 60's, y no es sino hasta 1992 que la nación adquiere la primera supercomputadora CRAY, para llevar a cabo supercómputo para fines de investigación y de desarrollo tecnológico, así mismo, a partir de 1997 instituciones como el CICESE, la UNAM, entre otros, adquieren las primera máquinas de supercómputo de SGI, las Origin 2000.

Para tener una mejor referencia y entendimiento de las supercomputadoras en base a su arquitectura se mencionará a continuación la taxonomía de Flynn.

II.I.2 Arquitectura.

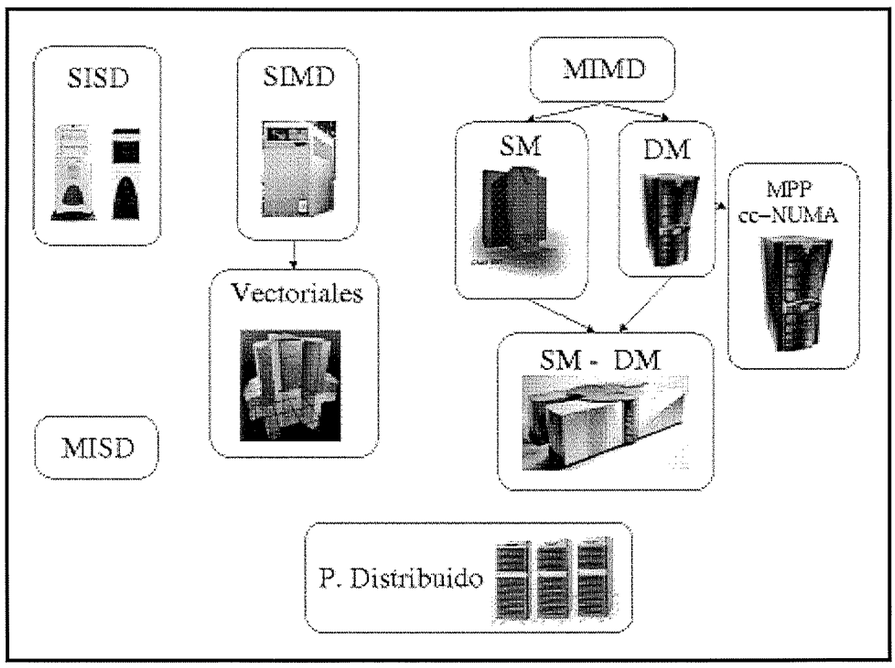


Figura 1. Diferentes arquitecturas de computadoras.

La arquitectura de las supercomputadoras es un término que comprende al hardware así como el software de alto nivel, lenguajes y algoritmos de características selectivas y distintivas que se adapten a los elementos físicos de la supercomputadora [Arqui 03].

Para clasificar las supercomputadoras en base a su arquitectura o estructura de hardware es útil la taxonomía de Flynn (desarrollada en 1966), que hace una distinción entre las computadoras en la manera en que manipulan las instrucciones y los datos que fluyen en el procesador. Son 4 categorías que ha continuación se mencionan:

SISD (Single Instruction Single Data, Una Instrucción Un Dato).

Este tipo de máquina se compone de un CPU que procesa las instrucciones secuencialmente (en serie). Las supercomputadoras actuales tienen más de un CPU, pero si éstos trabajan de manera independiente con las instrucciones y con los datos, hablamos de una máquina SISD.

Ejemplos: Estaciones de trabajo IBM, HP, Sun.

SIMD (Single Instruction Multiple Data, Una Instrucción Múltiples Datos).

Estas máquinas contienen varios CPUs los cuales trabajan en paralelo ejecutando las mismas instrucciones sobre conjuntos diferentes de datos.

Ejemplos: CPP DAP Gamma II, Quadrics Apemille.

Una subclase de las máquinas SIMD son las de procesadores vectoriales; éstas trabajan sobre un arreglo de datos de manera paralela, ejecutando la misma instrucción sobre cada conjunto de datos (que se denomina vector) en uno o varios CPUs.

Ejemplos: Hitachi S3600, Cray YMP, NEC SX, Japonesa, Cray YMP Americana.

MISD (Multiple Instructions Single Data, Múltiples Instrucciones Un Dato).

Esta clase de máquina teóricamente procesa múltiples instrucciones sobre un mismo conjunto de datos. Sin embargo, no existe ninguna máquina que corresponda a dicha clasificación.

MIMD (Multiple Instructions Multiple Data, Múltiples Instrucciones Múltiples Datos).

Estas máquinas contienen una mayor cantidad de CPUs, los cuales al ejecutar un proceso pueden repartir tanto los datos como las instrucciones en los diferentes procesadores. Difiere de una máquina de multiprocesadores tipo SISD en que las diferentes instrucciones pueden estar relacionadas entre si.

Las máquinas MIMD pueden dividirse en dos tipos de acuerdo al manejo de la memoria:

Memoria Compartida (MC)

Múltiples CPU's comparten el mismo espacio físico de memoria. Todos los procesadores accesan a la memoria de la misma forma. Poseen un número de procesadores pequeño porque son difícilmente escalables, ya que al aumentarlos, el tráfico en el ducto de interconexión que comunica a los procesadores y a la memoria compartida aumenta y por lo tanto baja su eficiencia. Para disminuir esto, se ha recurrido al uso y al incremento de memoria de amortiguamiento (cache) y la implementación de diferentes topologías de interconexión ya sea en forma de malla (crossbar), de red o un ducto central.

Las computadoras MIMD con memoria compartida son sistemas conocidos como de multiprocesamiento simétrico (SMP) donde múltiples procesadores tienen igual acceso a memoria y a los dispositivos de E/S y un mismo sistema operativo. Otro término con que se le conoce es: Máquinas firmemente juntas o de multiprocesadores. Ejemplos son: SGI/Cray Power Challenge, IBM R50, HP Alpha Server, NEC SX-5, Sun Fire 4800, entre otras. [Arqui 03]

Memoria Distribuida (MD)

Cada CPU tiene conectada su propia memoria (a lo cual se le conoce como nodo), sin embargo están conectados por algún tipo de red, de manera que pueden intercambiar datos entre ellos, por medio de algún mecanismo de coordinación que generalmente es del envío de mensajes. Su escalabilidad es mayor que en las de memoria compartida, sin embargo la velocidad de comunicación entre los procesadores es menor (dependiente de la eficiencia en el envío de mensajes) y este parámetro es muy importante en estas máquinas. Para algunas aplicaciones este tipo de arquitectura es sin lugar a dudas malo, pero para otras puede ser sustancialmente ventajoso.

Las computadoras MIMD de memoria distribuida son conocidas como sistemas de procesamiento en paralelo masivo (MPP) donde múltiples procesadores trabajan en diferentes partes de un programa, usando su propio sistema operativo y memoria. Además se les llama multicomputadoras, máquinas libremente juntas. Algunos ejemplos de este tipo de máquinas son IBM SP2 y SGI/Cray T3D/T3E.

Algunas máquinas utilizan ambos esquemas (MC MIMD y MD MIMD) ya que dentro de un nodo de varios procesadores utilizan memoria compartida, pero con otros nodos tienen memoria distribuida. Se conocen como sistemas de memoria compartida distribuida.

Ejemplos: NEC SX-5, HP ServerSC, SGI Origin.

II.1.3 Ejemplos.

NEC SX-4 y SX-5.

Características generales.

Las supercomputadoras NEC serie SX son Vectoriales - Paralelas escalables. Son procesadores de memoria compartida (hasta 16 procesadores).

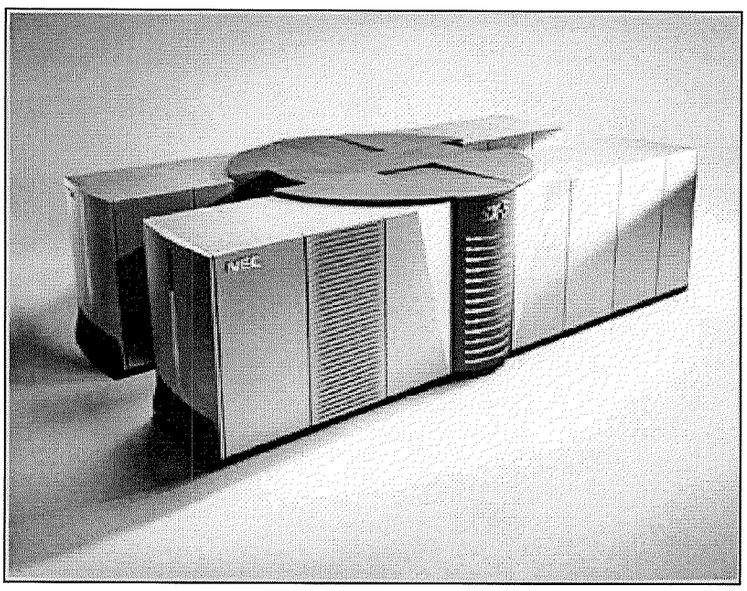


Figura 2. Supercomputadora Nec SX4 – SX5.

Procesadores.

Los procesadores SX-4 tienen un rendimiento de hasta 2 GFlops (el procesador SX-5 hasta 10 GFlops).

El sistema completo puede tener hasta 16 procesadores de memoria compartida y ejecutar

instrucciones escalares con 128 registros de hasta 64 bits (en doble precisión hasta 128 bits), y vectoriales de hasta 16 registros para la SX-4 y 72 para la SX-5, y usa memoria cache vectorial.

Puede ejecutar 4 instrucciones por ciclo de reloj, ya que permite ejecuciones fuera de orden y predicción.

Memoria.

En cada nodo, una conexión de cross-bar proporciona acceso a la memoria compartida a cada procesador de manera uniforme y a alta velocidad, hasta de 16 GB/s por procesador (hasta 80GB/s para la SX-5).

Cuando se conectan dos o más nodos, el sistema se convierte en un sistema de memoria compartida-distribuida con memoria de acceso no uniforme (NUMA) conectadas por otro crossbar entre nodos. [Arqui 03]

Cray SV1.

Características generales.

El sistema Cray Vectorial Escalable 1 (SV1) tiene de 8 a 32 procesadores de 300 Mhz. La memoria compartida puede ser de hasta 32 GB. [Arqui 03]

Esta supercomputadora salió al mercado en 1999.

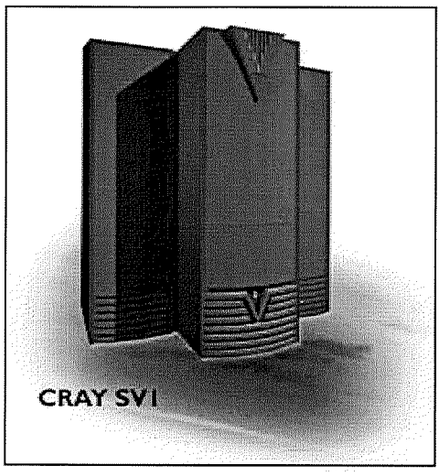


Figura 3. Supercomputadora Cray SVI.

Procesadores.

Cada procesador puede realizar operaciones escalares y vectoriales (de 8 entradas) y en conjunto realizar procesos en paralelo. Cada procesador tiene un rendimiento de 1.2 GFlops (los SV1e y SV1ex hasta de 2 Gflops) operando en modo vectorial, con precisión sencilla o doble. Los procesadores tienen una cache vectorial de 256 KB.

Los procesadores están agrupados en conjuntos MSP (Multistreaming Processors) los cuales tienen 4 procesadores sincronizados conectados a la memoria con 4 interfaces de memoria independientes. Cada uno de ellos contiene registros de 2 entradas para vector, pero sincronizados con los demás procesadores pueden realizar operaciones vectoriales de 8 entradas.

Memoria.

La memoria de la máquina es compartida hasta 32 procesadores. El ancho de banda de procesador a memoria es de 51.2 Gbits . [Arqui 03]

Cray Y-MP 4/464.

Características generales.

Tiene 4 procesadores vectoriales de 167 MHz. Tiene una memoria compartida de 512MB.

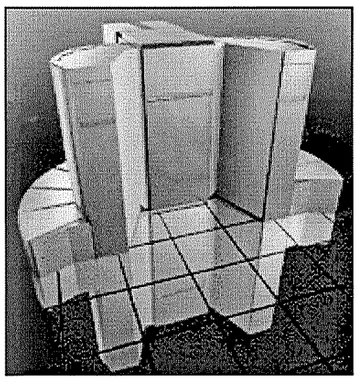


Figura 4. Supercomputadora Cray Y-MP 4/464.

Fue la primera supercomputadora en obtener un rendimiento sostenido de más de 1 Gflops y fue desarrollada en 1988.

Procesadores.

Los 4 procesadores de la máquina Y-MP, diseñados por Cray Research Inc., y pueden trabajar en paralelo y realizar operaciones matemáticas escalares y vectoriales (de 8 entradas). Cada procesador tiene un rendimiento de 166 millones de instrucciones por segundo (MIPS) y en operación vectorial puede realizar 333 Mflops con precisión sencilla o doble (64 o 128 bits).

Memoria.

La memoria de la máquina es compartida, y los procesadores son capaces de direccionar en su totalidad a la memoria principal. Los procesadores tienen 4 puertos de memoria paralela que permiten diferentes tipos de transferencias de manera simultánea.

Unidades funcionales.

Las unidades funcionales son las que realizan las operaciones matemáticas. Son independientes entre si, por lo que las operaciones (como sumas o restas) pueden ejecutarse en paralelo.

Dependiendo de la complejidad de la operación, las unidades funcionales tienen diferente número de segmentos.

Los operandos entran a la unidad funcional uno cada ciclo de reloj. Los resultados son completados uno por ciclo de reloj, después de atravesar todos los segmentos de la unidad.

Las unidades funcionales de la máquina Cray son capaces de encadenarse.

Esto sucede cuando el registro vectorial resultante se vuelve un registro vectorial operando para un segundo cálculo. Esto se debe a que las unidades funcionales operan paralelamente.

Almacenamiento.

La Cray Y-MP tiene un subsistema de entrada/salida que distribuye los datos en una variedad de redes que comunican a 3 procesadores especializados con los discos, los ruteadores y el robot de cartuchos de almacenamiento con capacidad de 160 GB, así como

con el SSD. El SSD (Dispositivo de almacenamiento de estado sólido) es un banco de memoria auxiliar RAM de 1GB.

Sistema de Refrigeración.

La Cray Y-MP cuenta con un sistema de refrigeración que tiene como finalidad mantener a temperatura adecuada los procesadores y las unidades de disco. Estos son enfriados por un líquido llamado Fluorinert. [Arqui 03]

II.2 Tecnología de racimos (clusters).

La adopción de clusters, colección de estaciones de trabajo o computadoras personales conectados por una red local, ha sido ampliamente explotado desde la introducción del primer cluster en 1994 llamado Beowulf por Donald Becker y Thomas Sterling que consistía de 16 nodos 486-DX4 conectados a través de una red Ethernet. Su atracción reside potencialmente en el bajo costo de hardware y software [Steen 02].

Los clusters surgen como alternativa a las costosas supercomputadoras vectoriales y escalares y operan generalmente sobre Linux (alguna minoría usa Microsoft Windows), y redes de alta velocidad. La figura 5 muestra un cluster del tipo Beowulf [Ava 03].

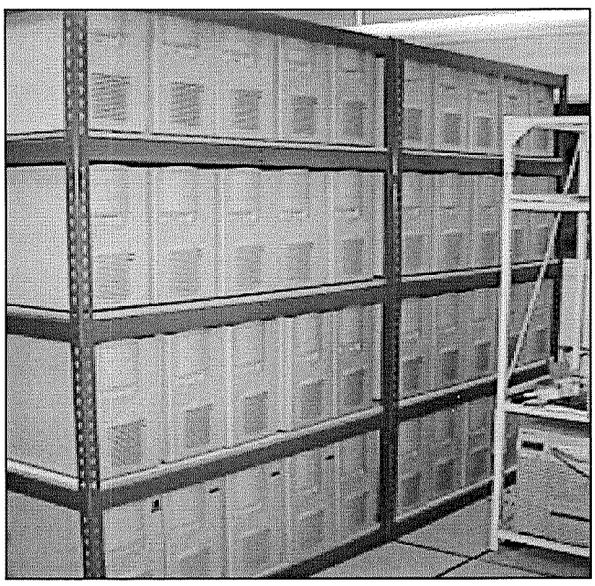


Figura 5. Cluster tipo Beowulf llamado "Avalon".

Los Beowulf se distinguen de otros sistemas de clusters, en que no imponen una arquitectura del sistema. Los componentes primarios del sistema que manejan la arquitectura se pueden descomponer en el procesador, memoria, red de trabajo y sistema de almacenamiento secundario. Pero el procesador y la red de trabajo han tenido el impacto más visible en los cambios de esta arquitectura. [Beowulf 03]

En estos días, los clusters Beowulf son usados para resolver problemas que requieren de gran capacidad de cómputo. Tales problemas son el mapeo del genoma humano, predicción del clima, simulación de explosiones nucleares, búsquedas de curas para el SIDA, entre otros. La disponibilidad de microprocesadores poderosos, del avance en las redes de alta velocidad y de software para dar soporte a aplicaciones de alto desempeño, ha permitido una revolución de los Beowulf. [Aspen 03]

Más allá del programador paralelo, el cluster Beowulf ha sido construido y usado por programadores con pequeña o nula experiencia en programación paralela. De hecho, el cluster Beowulf proporciona, con recursos limitados, una excelente plataforma para enseñar cursos de programación paralela y provee un excelente costo efectivo de cómputo a sus científicos computacionales también. En la taxonomía de las computadoras paralelas, el cluster Beowulf cae en algún lugar entre MPP (Procesadores Masivamente Paralelos, como nCube, CM5, Convex SPP, Cray T3D, Cray T3E, etc.) y NOWs (red de estaciones de trabajo) [Clumex 03]. El proyecto Beowulf se beneficia del desarrollo de ambas clases de arquitectura.

Los clusters son considerados las supercomputadoras modernas ya que combinan el rendimiento de las computadoras vectoriales, la escalabilidad de las máquinas MPPs, el precio atractivo de las PCs y la estabilidad del sistema operativo Linux. El cluster proporciona buen precio/rendimiento y la habilidad de escalar a decenas de Teraflops [Earl 03].

Analizando el TOP 500, que es un proyecto iniciado en 1993 para establecer cuales son las 500 supercomputadoras más rápidas y más poderosas del mundo, se puede observar que ha habido un incremento en el uso de los clusters y las constelaciones (cluster de nodos SMP) (ver figura 6). Esto debido principalmente al costo y la mejora en las tecnologías de redes [Top 03].

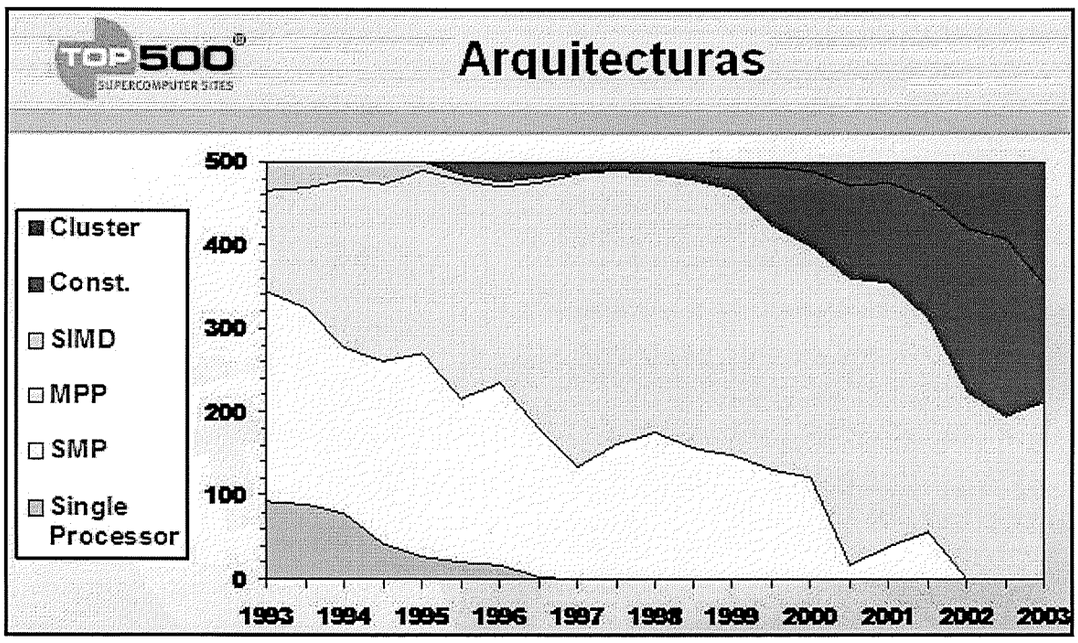


Figura 6. Evolución de las supercomputadoras. De grandes Mainframes a clusters de estaciones de trabajo (NOW).

La figura 6, fue tomada del top500 y muestra como algunas de las supercomputadoras más rápidas de todo el mundo están conformadas por clusters de estaciones de trabajo conocidos como NOW (Network of Workstations) [Top 03]. Las estaciones de trabajo son en su mayoría computadoras personales pero con un superpoder de cómputo, podríamos decir en este momento que son aquellas que tienen procesadores de 64 bits, como son Itanium, Opteron, Sparc, entre otras.

II.3 Tendencias del cómputo de alto rendimiento.

La adquisición de equipo de cómputo de alto rendimiento, especialmente supercomputadoras, presenta una serie de problemas: difícil y costoso mantenimiento, contratos ventajosos por parte de los proveedores, necesidad de contratación de personal especializado, poca o nula tolerancia a fallos, etc.

Pero de entre todas estas desventajas, la falta de escalabilidad es la que mayores problemas ha ocasionado a las entidades, que eventualmente, al llegar a un punto de bloqueo tecnológico, se ven irremediablemente obligadas a un cambio total en la infraestructura informática de la empresa [Inner 02]. Esto provoca que los altos costos de migración de plataformas se vean incrementados sustancialmente y deban incluirse dentro de los costos de escalamiento y de portabilidad.

La aparición, a finales de la década de los '90, de la tecnología Grid está cambiando radicalmente esta situación.

La posibilidad de unir todos los recursos informáticos de la red corporativa de un centro de investigación o de una empresa para emplearlos como una supercomputadora virtual, está permitiendo suplir las deficiencias de las supercomputadoras tradicionales.

La figura 7, muestra un estudio realizado por la empresa Gridsystems, el cual contiene una comparación entre las tres tecnologías de cómputo de alto rendimiento existentes (Supercomputadoras, clusters de estaciones de trabajo con 2 o mas procesadores y clusters de computadoras personales) y la nueva tecnología que esta siendo adoptada por diversas instituciones en todo el mundo [Inner 02].

En la figura 7, a mayor número de estrellas significa que tiene más ventajas.

Características: (A mayor número de estrellas, más ventajas)	Super Computador	Cluster de Estaciones multiprocesador	Cluster de PCs	GRID
Capacidad Crecimiento	*	**	***	****
Escalab. del rendimiento	****	***	***	***
Potencia de Cálculo	***	***	**	****
Precio	*	**	***	****
Tolerancia a Fallos	*	*	*	****
Rapidez Desarrollo	*	*	*	***
Portabilidad	*	*	**	***
Fácil Integración	*	*	*	***
Fácil administración	**	*	*	****
Fácil mantenimiento	*	**	***	****
No Rápida Obsolescencia	*	*	**	****

Figura 7. Comparación entre las diversas tecnologías de cómputo de alto rendimiento.

Como se puede observar en la figura, la tecnología grid muestra mucho más ventajas que las otras tecnologías de cómputo de alto rendimiento. Esta es la tendencia a nivel mundial.

En la República Mexicana, algunos Centros de Investigación como CICESE y Universidades como la UABC, la UNAM y la U. de G. ya empiezan a realizar estudios sobre la tecnología grid, y cómo ésta puede ayudar a los proyectos de investigadores que necesiten de un mayor poder de cómputo o de grandes montos de datos. Además, en el pasado mes de Abril del 2003 en la reunión de primavera CUDI 2003 [Cudi 03], CICESE, por conducto de Salvador Castañeda, lanzó una convocatoria a las demás entidades sobre la creación de un Grid Nacional para la unificación de recursos computacionales entre las diferentes entidades que estén interesadas en el tema y que cuenten con Internet 2.

III. TECNOLOGÍA GRID

III.1 Fundamentos.

Un Grid computacional es un tipo de sistema paralelo y distribuido que permite el compartir, seleccionar y agregar recursos distribuidos geográficamente en forma dinámica dependiendo de su disponibilidad, capacidad, rendimiento, costo y requerimientos en "calidad del servicio" del usuario [Grcomp 03].

Este sistema de redes permite la venta y distribución de tecnologías de la información (TI) como un servicio a través de Internet con un sistema de pago por uso (similar a como se distribuye la electricidad o el agua), eliminando la necesidad por parte de los usuarios de desarrollar y actualizar permanentemente costosas infraestructuras informáticas propias [IBM 02].

Los Grids son un nuevo tipo de redes que permiten a diversas organizaciones dispersas geográficamente compartir aplicaciones, datos y recursos computacionales de manera tal que sea transparente para el usuario.

El término "Grid" (malla por su significado en español) proviene del grid que suministra energía eléctrica, el cual proporciona grandes cantidades de electricidad a los usuarios en forma tal, que sea transparente para ellos. The economist publicó la siguiente analogía para dar una idea más clara de lo que es un grid computacional a sus lectores:

Imagine que cada vez que conecte un tostador, usted tenga que elegir la estación de energía la cual proporcione la electricidad necesaria para encender el tostador. Peor aun, que usted solo pueda seleccionar las estaciones de energía que la compañía que fabricó la tostadora decida. Que tal si en ese lapso de tiempo, la estación de energía que eligió esta

trabajando a su máxima capacidad, la tostadora no podrá funcionar.

Como sabemos, cada vez que necesitemos conectar algún dispositivo eléctrico, no es necesario elegir que estación de energía nos proveerá la electricidad, en cierta forma es transparente para nosotros [Econ 03].

Una de las metas mas importante de los grids computacionales es lograr la transparencia que los grids de energía eléctrica tienen, es decir, que el usuario utilice todos los recursos computacionales que se encuentren distribuidos geográficamente sin la necesidad de tener conocimiento de ello. Otra meta de la tecnología Grid es que cada nodo en un grid computacional, además de utilizar recursos de otros nodos, también aporte sus propios recursos computacionales para que sean utilizados por otros usuarios del grid.

Surgimiento de la tecnología Grid.

La tecnología Grid surgió, a finales de la década de los 90', como una solución a una serie de problemas que se presentan con la adquisición de supercomputadoras, tales como:

- Una costosa inversión inicial del equipo.
- Difícil y costoso mantenimiento.
- Contratos abusivos por parte de los proveedores.
- Necesidad de contratación de personal especializado.
- Poca o nula tolerancia a fallos.
- Falta de escalabilidad.
- Falta de disponibilidad.

Para muchos científicos, la calidad en sus investigaciones depende de su capacidad de cómputo. Hoy en día, es común encontrar problemas que requieran de semanas, meses y algunas veces años de cómputo para poder resolverse.

Científicos involucrados en este tipo de investigaciones necesitan un entorno computacional que realice grandes cantidades de cómputo sobre un largo periodo de tiempo. Este entorno computacional es conocido como High-Throughput Computing (HTC). En contraste, el entorno de High-Performance Computing (HPC) entrega grandes cantidades de cómputo en un corto periodo de tiempo.

El entorno de HPC es usualmente medido en operaciones de punto flotante por segundo. Muchos científicos de hoy les importan poco los flops; sus problemas radican en una escala mucho mayor. Ellos están enfocados en operaciones de punto flotante por mes o por año y su interés principal es la cantidad de trabajos que pueden completar en un largo periodo de tiempo.

La clave al HPC es el eficiente uso de los recursos disponibles. Hace años, la comunidad científica tenía que utilizar grandes computadoras llamadas Mainframe, para realizar trabajos que requerían de mucha capacidad de cómputo. Un gran número de individuos y grupos de trabajo tenían que juntar sus recursos financieros para poder adquirir una de estas supercomputadoras ya que eran demasiado costosas.

Después de la adquisición, venía el problema de compartir el tiempo de uso del equipo entre todos los que financiaron su compra. Esto era, y en algunas instituciones todavía lo es, un verdadero problema.

Con el paso del tiempo y con el avance de la tecnología, las computadoras se hicieron mas pequeñas, rápidas y menos costosas, esto dió como resultado que los científicos se

olvidaran un poco de los Mainframes y empezaran a comprar computadoras personales y estaciones de trabajo.

La capacidad de cómputo que las computadoras personales y estaciones de trabajo ofrecían, era menor que la de un Mainframe, pero proporcionaba acceso exclusivo al dueño del equipo.

Una solución que vino a dar gran capacidad de cómputo a un costo mucho menor que el de una supercomputadora, fue la creación de un cluster de computadoras personales a mediados del 1994, tal proyecto fue llamado Beowulf.

Hoy en día, las instituciones que se dedican a la investigación y necesitan de una gran capacidad de cómputo, cuentan con uno o varios clusters de computadoras, alguna supercomputadora (en caso de contar con suficientes recursos financieros), computadoras personales y estaciones de trabajo para los usuarios.

La mayoría de los usuarios de los institutos no utilizan su computadora todo el tiempo dando a lugar que estas máquinas pasen a un estado de inactividad conocido como 'idle', por lo tanto se está desperdiciando gran capacidad de recursos de cómputo que en lugar de estar ociosos o inactivos, podrían estar ejecutando trabajos que requieran de un cierto periodo de cómputo.

Con lo mencionado anteriormente, los científicos empezaron a formar grupos de trabajo para crear software que permitiera la utilización de recursos de cómputo de los equipos que no se estuvieran utilizando en ese momento.

Uno de los proyectos que mayor auge ha tenido es el Condor Project de la Universidad de Wisconsin, Madison, EUA [Condor 03].

Este proyecto toma ventaja de recursos de cómputo que de otra forma estarían desperdiciados y los utiliza para darles buen uso. Condor permite a los científicos enviar muchas tareas (jobs) al mismo tiempo. De esta forma, tremendas cantidades de cómputo pueden ser utilizadas con muy poca intervención del usuario. Mas a fondo, Condor permite a los usuarios tomar ventaja de las máquinas que se encuentren en estado de ociosas (idle) que, de otra forma, no podrían tener acceso. En un principio, el proyecto CONDOR estaba enfocado para uso departamental, o más bien, para uso local.

Debido al éxito obtenido con el proyecto Condor, otras universidades y empresas están desarrollando software (middleware) que, además de poder contar con los recursos que se encuentren localmente, tome ventaja de recursos que estén disponibles en otras partes del mundo de tal manera que sea transparente para el usuario. Es aquí en donde radica el concepto de Grid (malla computacional). Ver figura 8.

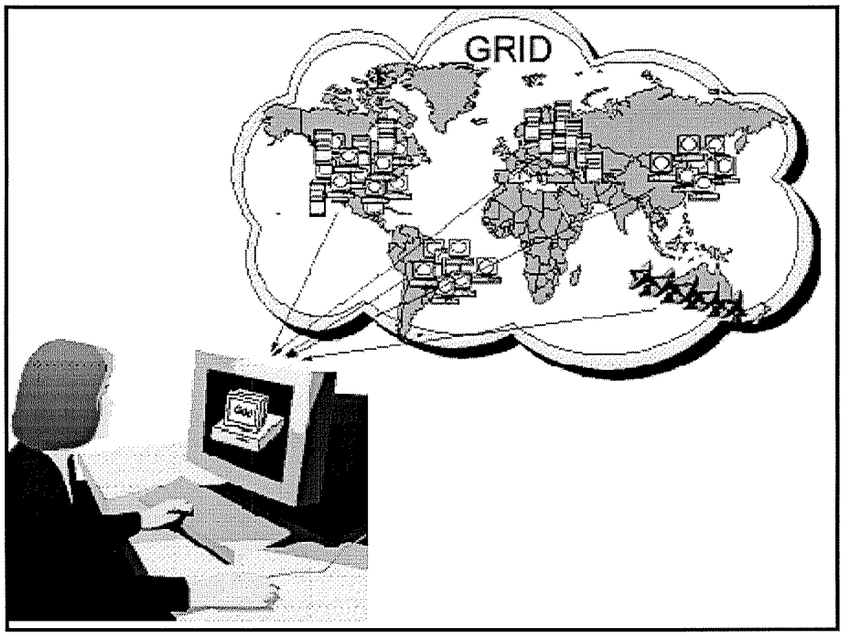


Figura 8. El usuario tiene acceso a recursos de cómputo distribuidos geográficamente en forma transparente para él.

A continuación se describen algunos de los beneficios que los grids computacionales ofrecen tanto a los usuarios finales como a las empresas:

Entre los beneficios del grid computacional en el marco de las tecnologías de la información podemos enumerar:

1. Integrar sistemas y dispositivos heterogéneos. El grid computacional proporciona un conjunto de capacidades de integración horizontal que dirige de forma efectiva los recursos de toda la empresa, e incluso extienden la solución entre múltiples organizaciones. Por ejemplo, un científico que participe en una investigación grid, podría obtener el acceso a una única supercomputadora conectada a un laboratorio.

2. Mejora del costo efectivo de los entornos operativos. A través de la visualización de la consolidación, reservas, compartición y gestión de recursos de las funciones heterogéneas de tecnologías de la información, el grid computacional ayuda a simplificar los entornos operativos y su gestión reduciendo la administración de su supervisión. Además, como consecuencia de fomentar la utilización eficiente de los recursos, el grid computacional ayuda a las empresas a construir una infraestructura de tecnologías de la información de costos efectivos que asegure la completa utilización de las inversiones en tecnología existente.

3. Creación de competencias virtuales seguras y flexibles. Las tecnologías grid adoptan las nociones de flexibilidad, libertad de elección de estándares abiertos. Los grids son capaces de descubrir dinámicamente los recursos disponibles y ajustarse a los entornos cambiantes y fluctuantes de las tecnologías de la información. Por esta razón, la tecnología grid facilita el establecimiento, reestablecimiento y cambios de los parámetros que requieren los negocios respecto a la seguridad y compartición de recursos.

4. Incrementa la capacidad de recursos para responder a las fluctuaciones de la demanda. Permitiendo a las organizaciones de tecnologías de la información agregar recursos distribuidos y explotar una capacidad no utilizada, los grids incrementan de forma importante la cantidad de recursos computacionales y de datos disponibles. El grid computacional ayuda a crear infraestructuras de tecnologías de la información que pueden responder rápidamente a oleadas inesperadas en el tráfico y uso de los recursos.

5. Aumenta la fiabilidad de la infraestructura. Sacando ventaja de los recursos del grid como una alternativa ante la recuperación de los desastres tradicionales, los departamentos de tecnologías de la información pueden mejorar la fiabilidad y disponibilidad de sus infraestructuras tecnológicas para aumentar la resistencia a una fracción del costo de los sistemas duplicados.

Entre los beneficios del grid computacional en el marco de las empresas podemos enumerar:

1. Mejorar el tiempo de los resultados para nuevos productos y servicios. Aumentando la productividad y la colaboración, las organizaciones mejoran el tiempo en la obtención de resultados. Tanto si estos resultados incluyen llevar un nuevo producto al mercado más rápidamente, resolver un complejo problema de negocio más pronto, realizar un análisis de datos en profundidad para lanzar un nuevo servicio, el grid ofrece a las compañías acelerar el tiempo y tomar la delantera al mercado.

2. Facilitar la colaboración y promover la flexibilidad de las operaciones. Las ofertas de grid pueden agrupar no sólo recursos tecnológicos distintos, sino también a la gente. Facilitando que el personal pueda compartir, acceder y gestionar la información, la tecnología grid puede hacer que las organizaciones sean capaces de mejorar la colaboración

en todas las unidades de negocio y geográficas para dar soporte a estrategias de globalización.

3. Efectivamente escalable para favorecer distintos tipos de demandas. Con el grid los negocios pueden crear infraestructuras flexibles y elásticas que gestionen rápidamente las fluctuaciones de las demandas de los clientes permitiendo el acceso instantáneo a los recursos de computación y datos que respondan a las necesidades del negocio. La capacidad de resolver problemas de negocio complejos más rápido significa que las organizaciones pueden moverse más rápidamente y ganar ventajas competitivas en el mercado.

4. Incrementar la productividad. Proporcionar a los usuarios finales un acceso sin restricciones a los recursos informáticos, de datos y de almacenamiento que necesitan, la tecnología grid puede ayudar a las compañías a mejorar la gestión de recursos humanos.

5. Mantener las inversiones de capital. Maximizar la productividad mediante la utilización eficaz de los recursos existentes es una de las claves para minimizar los costos. Asegurando la utilización óptima de las capacidades informáticas, la tecnología grid puede ayudar a las empresas a evitar las dificultades comunes de sobreprovisionamiento o incurrir en el exceso de costos para infraestructura. Debido a que el grid computacional se basa en estándares abiertos para crear una infraestructura única y unificada, la tecnología libera a las organizaciones de tecnologías de la información del peso de administrar sistemas no integrados, reduciendo de esta forma la supervisión. [Gcomp 03]

III.2 Arquitectura del Grid.

Los grids computacionales son creados para servir a diferentes comunidades con una gran variedad de características y requerimientos. Por lo tanto, probablemente no se tenga una sola arquitectura del grid. Sin embargo, se pueden identificar algunos servicios básicos que la mayoría de los grids proporcionan.

Varias personalidades involucradas en el estudio de la tecnología grid, en su definición de grid, concuerdan en que las organizaciones que quieran formar un entorno de este tipo deben establecer relaciones que puedan compartir entre ellas. A esto se le conoce como interoperabilidad.

En un entorno de red, interoperabilidad significa tener protocolos comunes. Por lo tanto, la arquitectura del grid es principalmente una arquitectura de protocolos que a su vez definen un mecanismo básico por el cual los usuarios y los recursos negocian, establecen, administran y explotan las relaciones que comparten.

Esta descripción de la arquitectura del grid identifica requerimientos para clases generales de componentes. El resultado es una gran estructura abierta de la arquitectura, dentro de la cual, puede contener soluciones para requerimientos clave del usuario.

La arquitectura es organizada por capas como se muestra en la figura 9. Los componentes dentro de cada capa comparten características comunes.

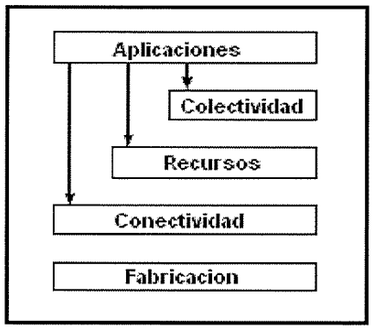


Figura 9. Las capas de la arquitectura del grid.

Capa de fabricación: Interfaces para un control local.

La capa de fabricación del grid contiene los recursos que van a ser compartidos. Esto puede incluir poder de cómputo, almacenamiento de información, sensores de carga, etc. La habilidad de compartir los recursos está bajo control de los protocolos del grid. En caso de que el recurso incluya a redes locales, los protocolos locales son los que se encargan en este punto. El sistema grid solo se enfoca con el acceso por encima de este punto.

En el nivel de fabricación se pueden implementar sistemas que den soporte a la calendarización de recursos y otras operaciones de alto nivel. Esto incrementa la complejidad y por lo tanto, aumenta el costo para agregar más recursos al grid.

Los recursos deben de contar con mecanismos de consulta que permitan a niveles más altos asegurar la estructura, estado y capacidades de un recurso junto con un mecanismo de administración de recursos para asegurar la calidad en el servicio.

Capa de conectividad: Comunicación fácil y segura.

La capa de conectividad contiene los protocolos de comunicación y autenticación requeridos para transacciones específicas del grid sobre la red. Los protocolos de comunicación permiten el intercambio de datos entre diferentes recursos de la capa de fabricación. Los protocolos de autenticación proveen servicios de comunicación que proporcionan mecanismos seguros para verificar la identidad de los usuarios y recursos. Los requerimientos en comunicación típicamente incluyen tareas de transporte y ruteo. Para la mayoría de los casos el protocolo TCP/IP contiene las herramientas pertinentes. Los sistemas grid en un futuro, probablemente requieran extensiones a este protocolo.

Los sistemas de seguridad pueden ser basados en estándares existentes para reducir la complejidad y así el grid puede utilizar las herramientas disponibles para Internet en donde sean aplicables.

Capa de recursos: Compartiendo un solo recurso.

La capa de recursos utiliza los protocolos de comunicación y seguridad de la capa de conectividad para controlar la negociación, inicialización, monitoreo, conteo y la contabilidad de operaciones compartidas sobre recursos individuales. Los protocolos de la capa de recursos están enfocados principalmente con recursos individuales.

Hay dos clases de protocolos en la capa de recursos:

- *Protocolos de información*: Son utilizados para obtener información sobre la estructura y estado del recurso, por ejemplo, su configuración, la carga actual, políticas de uso, etc.

- *Protocolos de administración*: Son utilizados para negociar el acceso a los recursos compartidos, especificando por ejemplo, requerimientos en recursos y las operaciones a llevar a cabo.

Capa de colectividad: Coordinando múltiples recursos.

Mientras la capa de recursos está enfocada en interacciones con un solo recurso, la capa de colectividad contiene protocolos y servicios que son globales en naturaleza y capturan interacciones sobre una colección de recursos.

- Los componentes de colectividad permiten implementar una gran variedad de métodos para compartir los recursos sin tener que realizar nuevos requerimientos sobre los recursos de la capa de fabricación a compartir. Este componente puede ser: Un servicio de directorio que permita a los usuarios consultar los recursos por nombre o por atributos como tipo, disponibilidad, carga de trabajo, etc.
- Servicios de monitoreo y diagnóstico que den soporte a los recursos en caso de que se presente alguna falla.
- Servicios de replicación de datos, servidores comunitarios de autorización.
- Sistemas de programación para el grid los cuales estén basados en modelos familiares de programación para poder desarrollar sistemas en un entorno grid.
- Sistemas de administración de carga de trabajo y estructuras de colaboración, mejor conocidas como entorno de solución de problemas.

Capa de aplicaciones.

La capa final de la arquitectura del grid comprende las aplicaciones del usuario. Las aplicaciones son construidas haciendo llamadas a servicios definidos en cada capa de la estructura del grid.

En cada capa, protocolos bien definidos proporcionan acceso a algunos servicios útiles como: administración de recursos, acceso a datos, descubrimiento de recursos y muchos otros más. Los protocolos y servicios de cada capa son utilizados para desempeñar las acciones que se necesiten.

Se pueden realizar interfaces para las aplicaciones (API) que se deseen ejecutar en un ambiente grid, utilizando el conjunto de herramientas (kit) de desarrolladores de programas (SDK) que permita interactuar con los protocolos del grid y con los servicios de la red [Foster 99].

III.2.1 Arquitectura de Servicios Abiertos para el Grid (OGSA).

La Arquitectura de Servicios Abiertos Grid (Open Grid Services Architecture) es un conjunto de especificaciones y estándares que combina los beneficios de la tecnología Grid y los servicios Web, actualmente, está siendo desarrollada por Globus e IBM [Ogsa 03].

De esta forma, los usuarios pueden, por primera vez, compartir y acceder a los recursos computacionales que necesitan en Internet, contando con el soporte de una infraestructura muy robusta, con capacidad de autogestión y siempre disponible. Así, los usuarios pueden integrar aplicaciones, compartir datos y poder de procesado (cpu), consiguiendo niveles de eficiencia muy altos, además de un ahorro considerable en costos.

El nuevo conjunto de especificaciones OGSA completa los estándares XML, WDSL y SOAP (todos ellos importantes para los servicios Web), con los estándares desarrollados por Globus para tecnologías de redes Grid, utilizados para localizar, planificar y asegurar recursos informáticos.

OGSA cuenta ya con el apoyo de empresas de diferentes industrias, incluyendo Avaki, proveedor de soluciones comerciales de software Grid; Entropía, proveedor de cómputo de redes Grid distribuida basada en PC; Microsoft; y Platform Computing, proveedor de software de cómputo distribuido.

IBM tiene como objetivo la implementación de OGSA como punto clave en su "Proyecto eLiza". El proyecto eLiza es la iniciativa de cómputo autonómico de IBM para construir un servidor de infraestructura autogestionable, abierto y heterogéneo para el comercio electrónico y la puesta en práctica de Grids comerciales [Ibgl 03]. Esto forma parte de la estrategia a largo plazo de IBM para tener mayor presencia en el mercado.

II.3 Clases de grid y su topología.

Hay varias clases o tipos de grid dependiendo del uso al que se va a dedicar. Por ejemplo, si una empresa necesita potencia de cálculo (cpu), el diseño de su grid es muy diferente a que si la empresa necesita compartir datos. Los tipos de grid mas utilizados son los siguientes:

- Grid de cómputo.
- Grid de datos.

Grid de cómputo.

Un grid de cómputo agrega el poder de procesamiento de una colección de sistemas distribuidos en áreas dispersas pero generalmente con una buena conectividad, digamos por medio de Internet 2, aunque no necesariamente. Un muy conocido ejemplo de un grid de cómputo es el SETI@HOME grid. Este tipo de grid utiliza el poder de procesamiento de computadoras personales que no estén siendo utilizadas por el usuario. Los ciclos en estado de ocioso de las computadoras personales en el grid SETI@HOME son combinados para crear un grid de cómputo utilizado para analizar transmisiones de radio recibidas del espacio exterior en el proyecto denominado "Búsqueda de inteligencia extra terrestre" [Seti 03]. El éxito de este proyecto es muy sobresaliente, de hecho, fue el punto de partida de varios proyectos los cuales se enfocan en la búsqueda de curas para enfermedades como el SIDA.

La mayoría de los usuarios del grid de cómputo lo utilizan para resolver problemas donde se demanda una gran cantidad de poder de cómputo y generalmente están relacionados con matemáticas, simulación, paralelización, fármacos, medicina, economía o cualquier aplicación que necesite de gran capacidad de cómputo.

Los grids de cómputo cuentan con las siguientes características:

- Creación de clusters de clusters.
- Permite la búsqueda de CPUs en estado de ocioso para mejorar la utilización de los recursos.
- Proporciona el poder de cómputo necesario para procesar tareas de gran escala.

Grid de datos.

Mientras que los grids de cómputo son utilizados para un buen aprovechamiento de los recursos disponibles, el enfoque de los grids de datos se basa en proporcionar un acceso seguro a minas de datos distribuidos en forma heterogénea.

Por ejemplo, en la creación de bases de datos, un grid de datos hace que un grupo de bases de datos disponibles funcionen como una sola base de datos virtual. Estos son muy usados por ejemplo en meteorología para ver el comportamiento global del clima.

Un grid de datos también administra el almacenamiento de los datos según las políticas locales y globales que gobiernan el uso de los datos, además de proporcionar un acceso más rápido, integro y seguro a dicha información. Algunas compañías como Oracle ya ofrecen soluciones para ser incorporadas a los grids computacionales [Oracle 03].

Topología del grid.

El grid está dividido en tres topologías o niveles lógicos:

- Grid departamental o intragrid.
  - Organizaciones simples.
  - Incapaz de integrarse con otras organizaciones (no sociedades).
  - Un solo cluster o racimo (son las granjas típicas).

- Grid empresarial o extragrid.
  - Organizaciones múltiples.
  - Integración con otras organizaciones (sociedades).
  - Múltiples clusters.

- Grid global o intergrid.
  - Muchas organizaciones.
  - Múltiples sociedades.

La figura 10 representa lo citado anteriormente. Como se puede observar, un Intergrid abarca extragrids y este a su vez abarca intragrids.

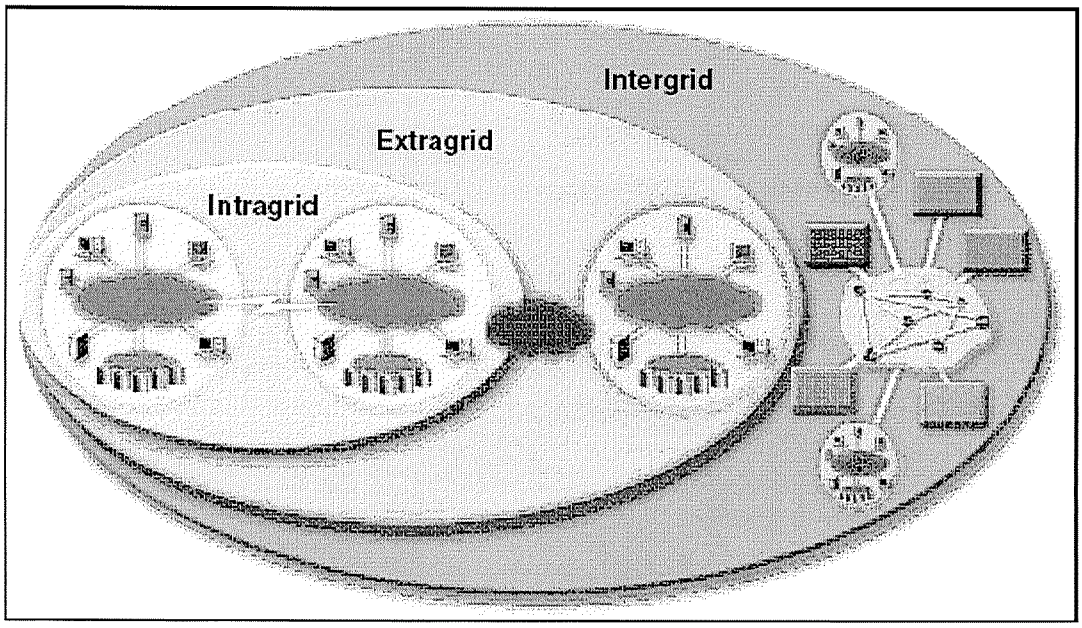


Figura 10. Topologías del Grid.

Un *grid departamental o intragrid* es la forma más sencilla de un grid y proporciona servicios de cómputo a nivel de grupo o departamento, incluso de institución. El tipo de software que permite este servicio puede ser un sistema de administración de recursos distribuidos (DRM), un sistema de administración de trabajos (JMS) o un sistema de calendarización de trabajos.

Esta topología también es conocida como *cluster grid.* Ver figura 11.

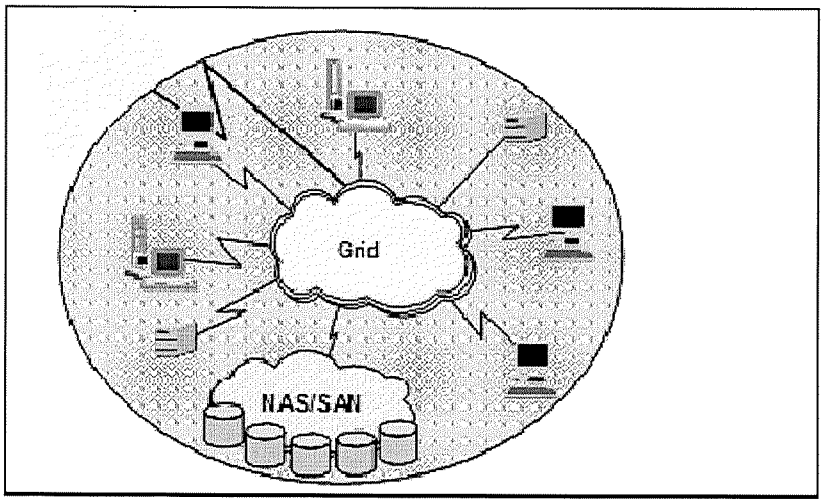


Figura 11. Intragrid o grid departamental.

Los *grids empresariales o extragrids* permiten a múltiples proyectos, departamentos de una empresa o campus compartir los recursos entre ellos, y no necesariamente tienen que introducir los temas de seguridad y manejo de políticas que están asociados con los grids globales. Ver figura 12.

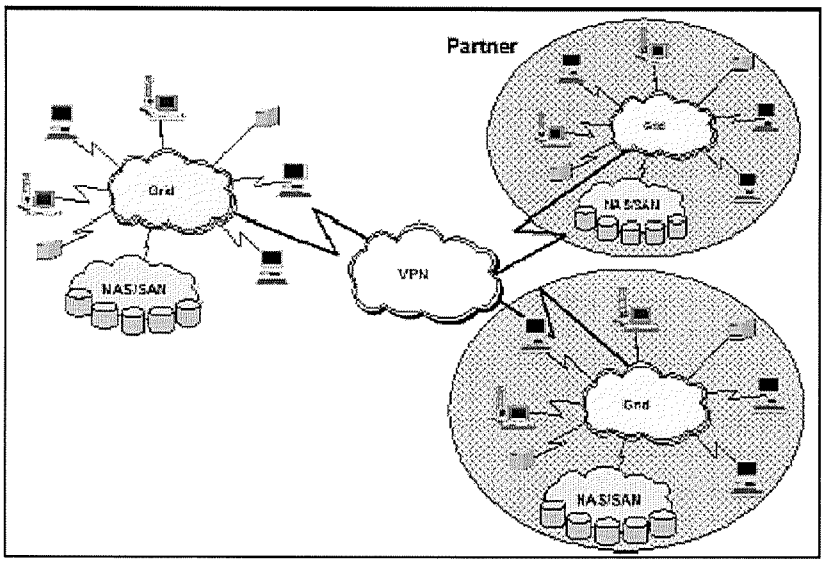


Figura 12. Extragrid o grid empresarial.

Los *grids globales o intergrids* son colecciones de grids empresariales y de grids departamentales además de otros recursos distribuidos geográficamente, los cuales han acordado su uso global e implementado protocolos y políticas para compartir los recursos.

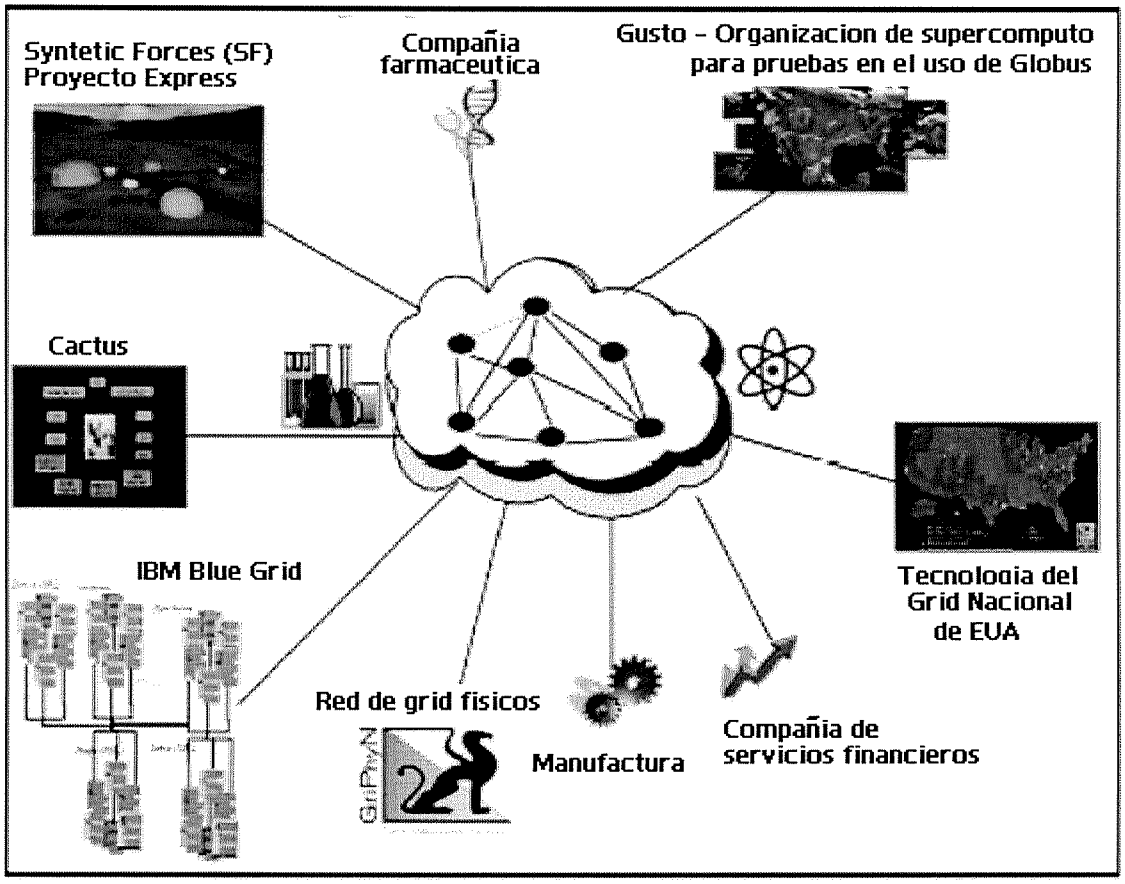


Figura 13. Ejemplo de un Intergrid o grid global.

III.4 Software.

Perspectiva general sobre los productos para el grid.

Hoy en día, alrededor del mundo se encuentran muchas empresas, universidades y centros de investigación que están desarrollando software para el grid.

El objetivo principal de los productos que proporcionan una plataforma de software para el grid, es que el producto proporcione las bases principales que se necesitan para la construcción de un grid.

A continuación se describen aquellos productos que son considerados los más importantes en la arena del grid computacional.

El proyecto *Globus.*

El Globus Toolkit es un software de código abierto que esta siendo desarrollado principalmente por las siguientes entidades: El laboratorio Nacional de Argonne, la Universidad del Sur de California y la Universidad de Chicago.

Hay tres componentes principales en el Globus Toolkit:

- Administración de recursos.
- Servicios de información.
- Administración de datos.

Dentro de las tecnologías usadas para realizar estos tres componentes se incluyen Administración de asignación de recursos del Grid (GRAM), Servicio de monitoreo y descubrimiento (MDS), y el protocolo de transferencia de archivos para el grid (GridFTP).

Todos estos componentes utilizan la Infraestructura de Seguridad del Grid (GSI), protocolo de seguridad en la capa de conexión [Globus 03].

Avaki.

Avaki es un vendedor de software localizado en Cambridge, Massachussets. Avaki ofrece una plataforma de Grid, la cual proporciona ambas clasificaciones de grid: grid de cómputo y grid de datos.

Avaki está diseñado especialmente para recursos dedicados y servidores existentes, no para desktops o computadoras personales. En la plataforma Avaki hay tres tipos de máquinas: bootstrap o arranque, de servicio y clientes de comandos.

Avaki 2.5 es soportado por varios sistemas operativos como: Windows NT/2000, Linux, Tru64, AIX, IRIX y Solaris.

La máquina de arranque es instalada en primer lugar para configurar e inicializar el grid, después se instalan los equipos de servicio. En caso de que el usuario desee tener acceso al Avaki Grid sin la necesidad de enrolar su máquina, esta puede ser configurada como máquina cliente.

Además de proporcionar poder de cómputo, también proporciona un ambiente grid para los datos. Los usuarios del Avaki Grid pueden copiar archivos al grid o compartir archivos locales en el grid. Cuando estos archivos son copiados al grid, pueden ser almacenados en cualquier recurso del grid, y el usuario no le debe de importar en que máquina está almacenado el archivo, solo se debe de tener en cuenta la ruta del archivo en el directorio global [Avaki 03].

DataSynapse.

DataSynapse es un vendedor de software localizado en Nueva York, E.U.A. Ellos proporcionan una plataforma de búsqueda de CPU conocida como LiveCluster.

La plataforma LiveCluster consiste de tres componentes principales: Un servidor, motores de ejecución y un controlador. LiveCluster es soportado sobre Windows, Linux y Solaris.

El servidor es responsable de la administración del sistema y la calendarización. Se tiene una interfaz Web al servidor que permite realizar la administración del sistema en forma remota. Los motores de ejecución pueden ser instalados sobre servidores dedicados o sobre computadoras personales. Cuando un motor de ejecución determina que el sistema se encuentra en estado ocioso, este le hará una petición al servidor indicando que esta listo para ejecutar tareas. Si el usuario interrumpe al motor de ejecución cuando esta procesando una tarea, la tarea se suspenderá y el servidor la va a reiniciar en otro motor de ejecución.

El controlador es responsable para el ingreso de tareas al servidor, así como la recepción de la salida de las tareas. Los controladores también pueden enviar comandos administrativos para obtener el estado de algún recurso o tarea [Data 03].

Entropía.

Entropía es un vendedor de software para el grid localizado en San Diego, California, E.U.A. El producto de Entropía, DCGrid, proporciona los medios para encontrar CPUs en estado de ocioso de máquinas Windows en una empresa.

DCGrid tiene tres componentes principales:

- Clientes DCGrid.
- Administrador DCGrid.
- Calendarizador DCGrid.

El cliente DCGrid es instalado en todas las máquinas de la empresa que quieran ser parte del Grid. Este cliente corre las tareas sin obstruir el trabajo normal del usuario en el equipo. Si una máquina no puede dedicar suficientes recursos para ejecutar una tarea, esta es reasignada a otra. El cliente contiene una "caja de arena" (sandbox) para que las tareas se ejecuten ahí. Esta es un área aislada para la aplicación a ejecutar, la aplicación no puede acceder a ningún archivo fuera de esta "caja de arena". Además, todas las aplicaciones y los archivos almacenados en una máquina cliente son cifrados con Triple-DES encryption. Estos dos hechos no solo protegen a la aplicación del usuario, sino que también protegen al usuario de la aplicación.

Los componentes del servidor del DCGrid están conformados por el calendarizador DCGrid y el administrador DCGrid. El calendarizador DCGrid es responsable de despachar las tareas a las máquinas clientes. El administrador DCGrid ofrece una interfaz Web para la administración del grid empresarial. Esta interfaz Web permite al administrador monitorear el estado de las tareas y los clientes, además de administrar el control de acceso de los usuarios.

DCGrid soporta cualquier aplicación Win32 y no requiere alteración alguna al código fuente. Sin embargo, no todo tipo de aplicaciones pueden funcionar de manera correcta en la plataforma DCGrid. Las aplicaciones deben contar con cierto tipo de características para poder ejecutarse a través del DCGrid. Las aplicaciones que necesitan de gran capacidad de cómputo y las que son altamente paralelas son buenas candidatas para ejecutarse en la plataforma [Entro 03].

Platform Computing.

Platform es un vendedor de software localizado en Toronto, Canadá. Platform proporciona muchos productos de cómputo distribuido que están relacionados con el cómputo de grids. Algunos de sus productos son LSF, ActiveCluster y Multicluster.

LSF es un producto que ofrece administración de carga de trabajo. LSF es soportado por AIX, IRIX, Tru64, HP-UX, Solaris, Linux y Windows. LSF puede ser instalado en un grupo heterogéneo de servidores y balancea la carga de trabajo a través de estos servidores.

LSF consiste de tres componentes principales:

- Maestro.
- Servidor.
- Cliente.

Se debe de tener una máquina maestra por cada cluster, y ésta es responsable de calendarizar todas las tareas en el cluster. El servidor es cualquier máquina que está corriendo los demonios de LSF y puede ejecutar tareas. El cliente es una máquina que no está corriendo ningún demonio de LSF y solo puede enviar tareas y ejecutar comandos de LSF. LSF proporciona varios tipos de calendarizadores y también permite desarrollar calendarizadores basados en políticas de la organización. LSF soporta aplicaciones existentes y no requiere cambio algunos en el código. LSF también proporciona un buen conjunto de API's para el desarrollo de aplicaciones.

ActiveCluster es una extensión de LSF que puede ser usado en las computadoras Windows. ActiveCluster requiere LSF, porque necesita su calendarizador. Un servidor dentro de un cluster LSF puede ser convertido a un ActiveCluster y es responsable de enviar tareas a las

máquinas ActiveCluster. ActiveCluster proporciona un camino para utilizar los equipos que se encuentren en estado ocioso dentro de una empresa.

MultiCluster es otra extensión de LSF que es usada para juntar multiples LSF clusters. Una empresa grande puede contar con varios LSF clusters los cuales pueden pertenecer a cada departamento de la empresa. MultiCluster permite a una empresa no solo balancear la carga de trabajo de un cluster departamental, sino que puede balancear la carga de trabajo a través de todos los cluster departamentales. Por ejemplo, si el cluster dentro de un departamento necesita recursos adicionales, MultiCluster puede enviar tareas a otro departamento cuyo cluster tenga recursos en estado ocioso [Platf 03].

United Devices.

United Devices (UD) es un vendedor de software localizado en Austin, Texas. Ellos proporcionan una plataforma de búsqueda de CPU llamada Plataforma MetaProcessor. Esta plataforma consiste de dos partes principales, un servidor conocido como Servidor MP, y una parte cliente, conocido como Agente UD. Además del Servidor MP, también se tiene un servidor de base de datos y una consola Web para administrar la plataforma MetaProcessor.

El servidor MP es soportado solamente en Red Hat Linux, mientras que el servidor de base de datos puede correr sobre cualquier sistema operativo que soporte IBM DB2. Los agentes UD son soportados por Microsoft Windows 98, ME, NT, 2000, y XP, así como Linux. El servidor MP es responsable de calendarizar tareas, coleccionar datos y administrar la plataforma. La consola MP proporciona una interfaz Web para el servidor, la cual permite administración remota. Las tareas pueden ser ejecutadas a través de esta interfaz o por medio de la línea de comandos. El servidor de base de datos IBM DB2, es usado para

almacenar todas las aplicaciones, la estadística de las tareas, los usuarios, los agentes, etc.

Los agentes UD son instalados en las máquinas deseadas y son responsables de ejecutar las tareas que se les envíe. El agente solo ejecuta una tarea en un tiempo determinado y con baja prioridad, de esta forma las tareas del usuario del equipo no son afectadas. United Devices ha proporcionado gran escalabilidad con su plataforma MetaProcessor.

Los usuarios pueden bajar e instalar el agente UD y de esta forma donar sus ciclos de CPU para la comunidad de United Devices a través de su sitio Web. Esta plataforma ofrece su poder de procesamiento para investigaciones relacionadas en la lucha contra el cáncer [Udev 03].

GridSystems.

Esta empresa está situada en Mallorca, España. Innergrid es su producto multiplataforma para la aplicación de la tecnología Grid en los actuales entornos empresariales.

El sistema InnerGrid está diseñado para posibilitar la ejecución de programas de cálculo intensivo sobre un conjunto heterogéneo de máquinas, aprovechando al 100% todos los CPUs disponibles.

La operación general del sistema consiste en dividir una tarea en micro tareas, más pequeñas y de menor duración, que pueden ser resueltas por una máquina de potencia media en un tiempo razonable.

El sistema, a través de una interfaz sencilla y amigable, controla las máquinas que tiene disponibles y les encarga la ejecución de micro tareas, realizando un seguimiento de las mismas [Inner 02].

Sun One Grid Engine.

Este software esta siendo implementado por Sun Microsystems, cuya oficina corporativa se encuentra en Santa Clara, CA, EUA. El Sun One Grid Engine es un software de código abierto que proporciona todas las funciones de un administrador de recursos distribuidos (DRM), tales como: Encolamiento de tareas, balanceo de carga de trabajo, estadísticas de las tareas, suspensión y reinicialización de tareas, acceso a recursos específicos, entre otras funciones. Plataforma en las que trabaja: Solaris, Linux e Irix, entre otras. Actualmente existen dos variantes de este software:

Sun One Grid Engine Enterprise Edition (SGEEE): Resuelve las necesidades de un Campus grid o grid empresarial, por ejemplo, muchos usuarios, equipos y departamentos compartiendo recursos comunes pero trabajando en diferentes proyectos con diferentes metas y calendarios.

Sun One Grid Engine (SGE): Resuelve las necesidades de un cluster grid o grid departamental, por ejemplo, un usuario, equipo o departamento trabajando en conjunto sobre un proyecto. Este software es de código abierto [Sun 03].

En el capítulo IV se da una descripción más amplia de este software.

III.5 Aplicaciones para el grid.

Las aplicaciones más utilizadas en los sistemas grid son aquellas que involucran una gran cantidad de poder de cómputo en un largo periodo de tiempo.

Los primeros ejemplos de Grid Globales surgieron en entornos científicos, ya que a menudo los grandes retos científicos conllevan también grandes retos tecnológicos.

Tal vez los ejemplos más conocidos son el procesamiento y almacenamiento de datos de los grandes aceleradores de partículas o de los grandes telescopios que desde el espacio siguen de cerca todos los cambios de la Tierra, así como también las investigaciones del proyecto Genoma Humano. Sin embargo, los posibles campos de aplicación son muchos y muy variados:

- Estudios médicos para la prevención, el diagnóstico y tratamiento de enfermedades como el cáncer o el SIDA. El incremento en el número de casos de estudio al compartir datos entre diferentes hospitales puede sacar a la luz correlaciones que permitan entender el proceso de nacimiento, contagio y/o evolución de una enfermedad, o de virus.

- Estudios metereológicos para la mejora de las predicciones de los cambios climáticos a corto y largo plazo. No hay duda que en una disciplina de la que se ha dicho que el aleteo de una mariposa en la China puede provocar un huracán en el otro extremo de planeta, requiere de una computación extrema y en un entorno colaborativo (en este caso todo el planeta) para conseguir resultados fiables.

- Estudios de fluctuaciones de mercado en un entorno de economía global. La evolución de la economía internacional no permite hacer previsiones económicas

con la información de una región del planeta. Lo que ha sucedido y lo que sucede en cada momento en los diferentes mercados de los cinco continentes puede tener repercusiones inmediatas o a largo plazo, que hay que considerar para optimizar las inversiones de los grandes bancos así como los productos dirigidos a sus clientes [Appg 03].

En la industria privada, muchas empresas han recurrido a la aplicación de la tecnología grid en sus departamentos para poder lograr un mejor desempeño y obtener un mayor beneficio de sus recursos computacionales. Algunas empresas son:

Ford Motor Company: Se construyó un grid empresarial para el área de Ingeniería Colaborativa – Cómputo de Alto Rendimiento para el grupo de Motores y Transmisiones. El software utilizado fue el Sun Grid Engine de Sun Microsystems, junto con 500 estaciones de trabajo Sun Blade 1000 y un servidor Sun enterprise 3000. La aplicación principal del grid es en el área de simulación y análisis de elementos. Con esta implementación, los ingenieros del grupo de Motores y Transmisiones pueden dividir las tareas y obtener resultados que antes tardaban días, en solo 15 minutos. Debido al éxito obtenido, la compañía Ford tiene pensado implementar la tecnología grid en otras áreas de la empresa [Sun 03].

Motorola: Se implementó un grid computacional para el área de diseño de integrados. Esta área ya contaba con una infraestructura de supercómputo pero se necesitaba aun más. Al realizar algunos estudios se dieron cuenta que en varios departamentos que se contaba con infraestructura de supercómputo, no se estaba utilizando al 100%. Así que en vez de adquirir más equipo, descargaron el software de Sun (Sun One Grid Engine) y lo implementaron en sus diferentes departamentos. Con esto incrementaron su capacidad de cómputo sin necesidad de invertir en más equipo [Sun 03].

IV. DISEÑO DE UN MODELO GRID PARA LA RED-CICESE

IV.1 Supercómputo en CICESE.

A finales del año de 1996, el CICESE, a través de la Subdirección de Cómputo y Redes (hoy Dirección de Telemática) adquirió un servidor tipo gabinete Origin 2000 de la compañía Silicon Graphics, mismo que entró en operación en abril del siguiente año, con el firme propósito de brindar y reforzar diversas actividades científicas y académicas sustentadas en los diferentes proyectos de investigación, y de desarrollo tecnológico que día a día requieren de una infraestructura tecnológica de punta en el campo del cómputo de alto rendimiento. Debido a los grandes recursos con que contaba la supercomputadora: 10 procesadores R10000 de 195 MHZ cada uno con 4 MB de memoria caché, con un total de memoria RAM de 1 GB, discos SCSI con un total de 40.9 GB, etc., su uso se enfocó al desarrollo de aplicaciones e investigaciones de procesamiento en paralelo, de compilación y ejecución de programas que requerían de gran velocidad en cómputo, gran cantidad de memoria y de una entrega inmediata de resultados.

Sin duda la incorporación de esta supercomputadora SGI, vino a darle una salida a los proyectos que demandan de una gran cantidad de cómputo de alto rendimiento, sin embargo después de 5 años de la aparición de esta tecnología, y del aumento considerable en complejidad de las actuales aplicaciones científicas, fue imprescindible disponer de caminos alternos para la ejecución de dichas aplicaciones.

Fue así, que a partir de 1999 se iniciaron los estudios encaminados a la justificación de la restauración de la capacidad de cómputo del CICESE. Después de un sin número de presentaciones, estudios y replanteamientos, se logró el presupuesto para la compra de equipo. Por tal motivo, a principios del año 2002, se decidió la adquisición de otra supercomputadora, una Sun Fire 4800 llamada "calafia" y 5 estaciones de trabajo Sun

Blade 1000 formando éstas un cluster, todo este equipo de la compañía Sun Microsystems, con ello el centro ha logrado restituir su capacidad de cómputo [Cicese 03].

IV.1.1 Infraestructura.

El equipo de supercómputo que cuenta el CICESE para la creación del grid es el siguiente:

1 Supercomputadora SGI Origin 2000 (cicese2000).
1 Supercomputadora Sun Fire 4800 (calafia).
1 Cluster de 5 estaciones de trabajo Sun Blade 1000 (tribus).

Supercomputadora SGI Origin 2000 "cicese2000".

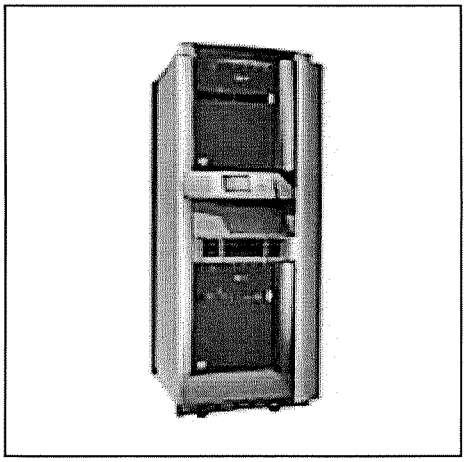


Figura 14. Supercomputadora Origin 2000 llamada "cicese2000".

Características generales.

10 procesadores superescalares de 195 Mhz, con topología de hipercubo, con un rendimiento de 3.9 Gflops. Tiene en total 1 GB de memoria; la memoria es físicamente distribuida y lógicamente compartida.

El sistema es modular para poder incrementar su tamaño; el sistema puede crecer hasta 128 procesadores. La arquitectura de SGI es de memoria lógicamente compartida entre procesadores y físicamente distribuida entre nodos. La interconexión entre nodos es de baja latencia.

Procesadores.

Cada módulo esta comprendido por 4 nodos y cada nodo posee 2 procesadores. Cicese2000 cuenta con 10 procesadores R10000 de 195 Mhz.

El módulo 1 posee 1 nodo con 2 procesadores y el módulo 2 tiene 4 nodos con 8 procesadores totalizando 10 procesadores en los 5 nodos. El gabinete tiene capacidad de hasta 16 procesadores y su máximo crecimiento es de 8 gabinetes totalizando 128 procesadores.

Cada procesador tiene la capacidad máxima de efectuar 390 millones de operaciones de punto flotante por segundo, totalizando 3.90 GFlops en la supercomputadora cicese2000.

Memoria.

La memoria principal de cicese2000 es de 1280 MB (1.25 GB) distribuida entre todos los 5 nodos (cada nodo con 256 MB) y se comparte lógicamente. Se conoce como Memoria

Compartida - Distribuida (DSM), para un procesador la memoria aparece como un sencillo espacio direccionable de 1280 MB.

Cada procesador tiene 64 KB de caché primario (32 KB de instrucciones y 32 KB de datos) y 4 MB de caché secundario.

Un solo nodo puede tener de 64 MB hasta 4 GB de memoria principal, lo que significa que un gabinete con sus 8 nodos puede alcanzar 32 GB de memoria principal, y si se juntan 8 gabinetes se tendrán 256 GB

Dispositivos de almacenamiento.

Son 5 discos SCSI cuya capacidad de almacenamiento suman 40.9 GB, distribuidos en los 2 módulos. El disco principal del sistema de 4.5 GB y los otros 4 de 9.1 GB. Se cuenta con una Unidad de Cinta (DAT) para cintas de 4 mm. y con una Unidad de CDROM. Cada módulo posee 5 ranuras para discos SCSI.

Dispositivos de entrada y salida.

La tarjeta base de E/S se localiza en la primera ranura de E/S en la parte posterior del módulo. La tarjeta base de E/S que viene con cada módulo contiene lo siguiente:

1 Conector Fast Ethernet 100 Base-T.
2 Puertos serial de 9 pins.
1 Puerto externo UltraSCSI (68 pins).
2 Interruptores de tiempo real.

Se tienen disponible 11 ranuras de E/S en cada módulo para agregar tarjetas de red, UltraSCSI, Canal de Fibra, ATM, HIPPI-Serial [Casdel 99].

Supercomputadora Sun Fire 4800 "calafia".

Características generales.

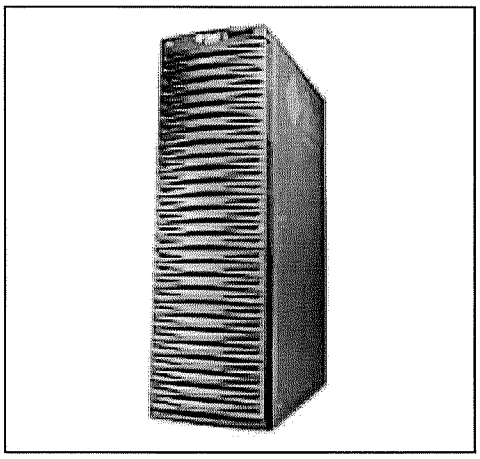


Figura 15. Supercomputadora Sun FIRE 4800 llamada "calafia".

El Sun Fire 4800 cuenta con 8 procesadores UltraSPARC III 900 Mhz con 8 MB de cache (**Rendimiento total 14.4 Gflops**), 8 GB de memoria RAM, un arreglo de discos con un total de 324 GB para almacenamiento.

Procesadores.

Los procesadores UltraSPARC III de calafia son procesadores superescalares montados en superficie altamente conductora de cobre para un disipado de calor eficiente, son de 64 bits en una arquitectura de Mutiprocesamiento Simétrico (SMP), con dos niveles de memoria cache cada uno:

- Nivel 1 (en el chip) de 64 KB de datos y 32 KB de instrucciones.
- Nivel 2 (fuera del chip) de 8MB de datos e instrucciones.

Cada procesador provee un rendimiento de 1.8 Gflops. Calafia puede crecer hasta 12 procesadores distribuidos en 3 tarjetas CPU/Memoria (4 por tarjeta).

Memoria.

La memoria es de 8 GB distribuida físicamente en 2 tarjetas CPU/Memoria, y puede crecer hasta 96 GB. Para el programador y aplicaciones, calafia presenta una memoria plana, global, compartida, con una ancho de banda de 9.6 GB/s (76.8 Gbits).

Dispositivos de almacenamiento.

Se cuenta con un medio de almacenamiento de alta disponibilidad y rendimiento como el arreglo de discos "Sun StorEdge T3" el cual provee una capacidad de almacenamiento total de 324 GB (consta de 9 discos de 36 GB y con una velocidad de rotación de 10K RPM) y cuyo canal e interconexión a la Sun Fire es por fibra óptica.

Otro medio de almacenamiento con el que se cuenta es el "Sun StorEdge D240" que contiene 2 discos de 36 GB (uno para el sistema operativo y otro de espejo), una unidad de CDROM/DVD y una unidad de Cinta DAT. Y su interconexión es SCSI.

Interconexión del sistema.

Este provee la comunicación entre las tarjetas de CPU/Memoria y los dispositivos de E/S, es un conmutador (switch) de interconexión punto a punto algo mas que un simple ducto común y cuyo ancho de banda es de 9.6 GB/s o 76.8 Gbits. Es conocido además como backplane. El "Fireplane" es capaz de dividir el sistema dentro de particiones completamente aisladas y de dividir la partición dentro de dominios lógicamente aislados (esta presentación de generar dominios es como dividir la supercomputadora en máquinas independientes con sus propios recursos y su propio sistema operativo). Calafia está configurada con un solo dominio.

Dispositivos de entrada y salida.

Calafia presenta en su parte trasera 2 "módulos" de E/S con un total de 16 ranuras PCI, en donde actualmente se encuentra conectado en una de sus ranuras el "Sun StorEdge T3", en otra el "Sun StorEdge D240" y en otra la tarjeta de red 10/100. [Casdel 02]

Cluster "tribus".

El cluster tribus está formado por 5 estaciones de trabajo Sun Blade 1000, y proporciona un rendimiento total de 9 Gflops.

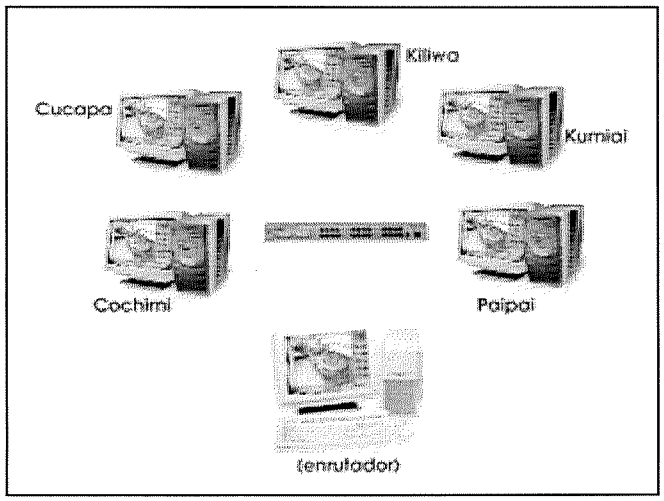


Figura 16. Cluster de 5 estaciones de trabajo Sun Blade 1000 llamado "tribus".

Cada estación de trabajo del cluster cuenta con:

- Un procesador UltraSparc III de 900 Mhz 64 bits, similar a los de la supercomputadora calafia, además es expandible a dos procesadores.
- 1 GB de memoria RAM expandible a 4 GB.
- 36 GB en disco duro Ultra SCSI.
- Tarjeta de red 10/100.

IV.1.2 Situación actual.

Actualmente, la infraestructura de supercómputo que se tiene en el CICESE, se encuentra de manera independiente. Se cuenta con Cicese2000 con 8 procesadores (2 procesadores se encuentran fuera de servicio debido a una falla en una de las fuentes de voltaje del equipo), Calafia de 8 procesadores y un cluster de 5 estaciones de trabajo Sun Blade 1000.

Con la adquisición de Calafia y el cluster vino a mejorar el servicio de supercómputo que con Cicese2000 no era suficiente, debido a las características "obsoletas" de este equipo, poco a poco se han ido migrando usuarios y aplicaciones de Cicese2000 a Calafia, la cual ha llegado a un punto en que se tiene bastante carga de trabajo, como consecuencia, Cicese2000 se ha quedado rezagada con poca actividad y el cluster aun no cuenta con el espacio y lugar adecuado donde se pueda obtener un mejor beneficio.

Se puede decir que existe un inapropiado balance en la carga de trabajo que a la postre deja a Calafia funcionando a su máxima capacidad y esto conlleva a que varios usuarios no puedan ejecutar sus tareas en ese momento y tengan que esperar a que se libere un poco los recursos de Calafia, generando algunas veces cierto malestar entre los mismos usuarios.

Con la adquisición del cluster tribus, cuyo principal uso es la visualización científica y cómputo paralelo, se ha logrado tener la operabilidad suficiente e independiente de Calafia para usuarios que lo soliciten, pero aun así, no se ha logrado disminuir su carga de trabajo debido a la falta de un buen esquema de distribución de cargas entre los 3 equipos.

Los inconvenientes que se han presentado con la supercomputadora Calafia se pueden resumir en lo siguiente:

1) Demasiada carga de trabajo con tareas secuenciales (en serie) y paralelas.

2) No existe un control en la asignación de los recursos.

Se pueden listar los siguientes inconvenientes que se han presentado con el cluster:

1) No todas las máquinas del cluster estaban siendo utilizadas por igual.
2) La administración se torno un poco más difícil ya que el verificar que máquina estaba siendo utilizada y que máquina no, se tenía que hacer en forma manual por el Administrador, y éste a su vez, informar a los usuarios que algunas máquinas ya estaban saturadas y que utilizaran otras.

Ahora, no basta con distribuir la carga de trabajo entre Calafia y el cluster, sino que es necesario llevar una mejor administración y control de las tareas y aplicaciones y de esta manera, brindar un mejor servicio a los usuarios de supercómputo del CICESE.

IV.1.3 Propuesta para la implementación de la tecnología grid.

Debido a la situación actual que presenta el equipo de supercómputo de CICESE, y la necesidad de llevar una mejor administración de los recursos computacionales y dada la investigación realizada sobre la tecnología grid, se propone realizar una implementación de la tecnología grid en un ambiente departamental, y de esta forma unificar los recursos computacionales que ofrecen las supercomputadoras Calafia, Cicese2000 y el cluster tribus.

Con la implementación de un grid departamental, se mejorará la administración de los recursos computacionales.

También permitirá la creación de "políticas para el uso de los recursos del Grid" y de esta forma lograr una mejor asignación de recursos a los usuarios.

Además, la implementación de esta tecnología permitirá, en un futuro, la creación de nuevos proyectos de investigación en colaboración con otras entidades que estén relacionadas con el tema tomando como medio de comunicación el canal de Internet 2, y de esta forma, se puede lograr constituir un grid a mayor escala, como lo es un grid global.

IV.2 Modelo ideal del GRID-CICESE.

Para el CICESE, lo ideal es conjuntar todo el poder de cómputo que ofrece el equipo que se tiene en sus diferentes departamentos por medio de los sistemas grid, y ponerlos a disposición para sus usuarios internos y externos, así como asociarse con otras entidades que se encuentren envueltas en la implementación de la tecnología Grid utilizando el canal de Internet 2 y formar organizaciones virtuales que compartan sus recursos computacionales. La figura 17 muestra el esquema ideal de Modelo Grid en la Red-CICESE.

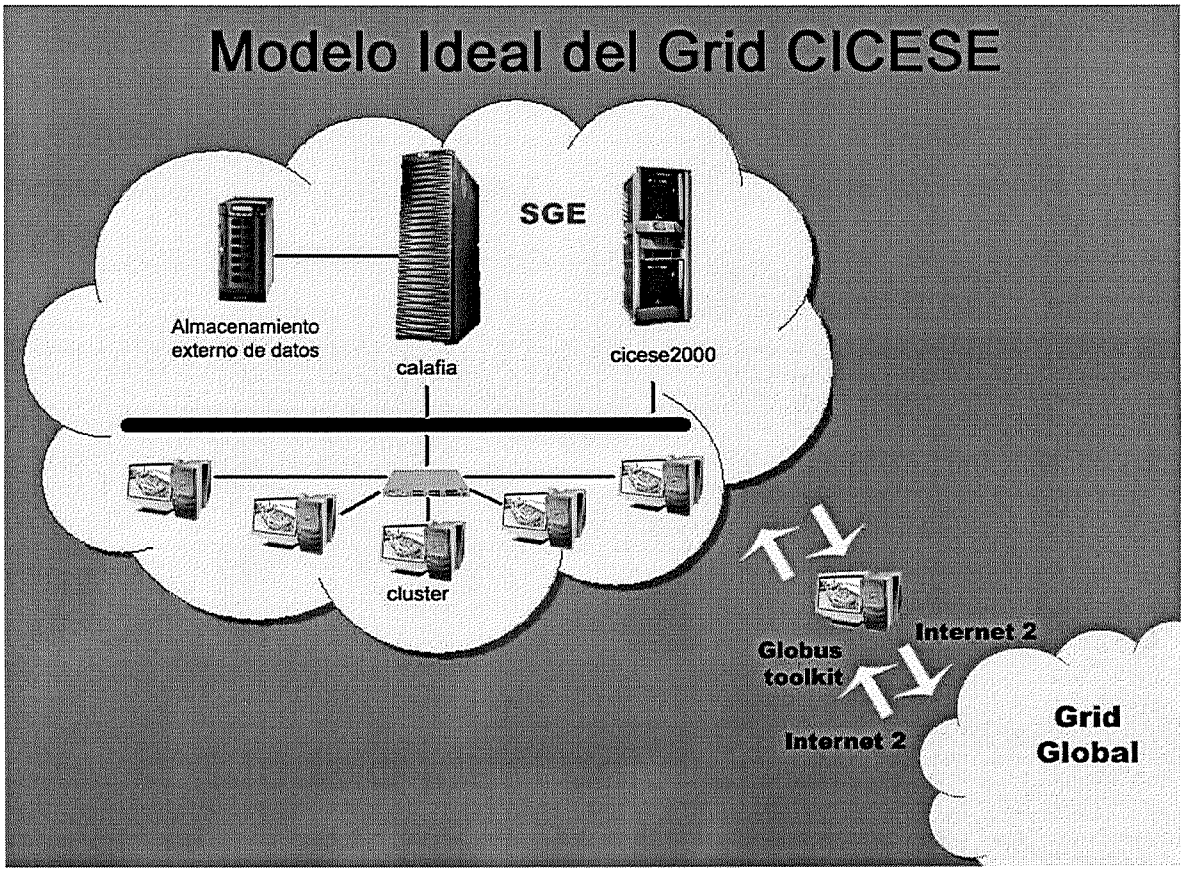


Figura 17. Modelo ideal del Grid-CICESE.

Como se observa en la figura 17, el grid departamental va a estar conformado por las supercomputadoras Calafia y Cicese2000, el cluster Tribus y un dispositivo de almacenamiento externo de datos el cual tiene la capacidad de 3 TB ya que los discos de alta velocidad de la supercomputadora Calafia que se utilizan para los temporales, se encuentran funcionando al 90% de su capacidad.

El software a utilizar para la integración de los recursos del Departamento de Cómputo del CICESE es el Sun Grid Engine Software, ya que el equipo de supercómputo de CICESE (Calafia y el cluster Tribus) son propietarios de Sun Microsystems y existe un soporte por parte de la empresa.

Para enlazarse con otras entidades, se utilizará una máquina la cual tenga instalado el software de Globus Toolkit, ya que este software cuenta con una muy buena infraestructura de seguridad que maneja certificados para usuarios y también para máquinas. Además, el Globus Toolkit puede utilizar al Sun Grid Engine como su calendarizador de tareas.

El medio de comunicación con otras entidades va a ser por medio del canal de Internet 2.

IV.3 Modelo experimental del GRID-CICESE.

El propósito principal de este grid departamental es proporcionar mayor poder de cómputo a los usuarios de supercómputo de CICESE.

Actualmente el grid se encuentra en fase de experimentación. La integración de los recursos es la siguiente:

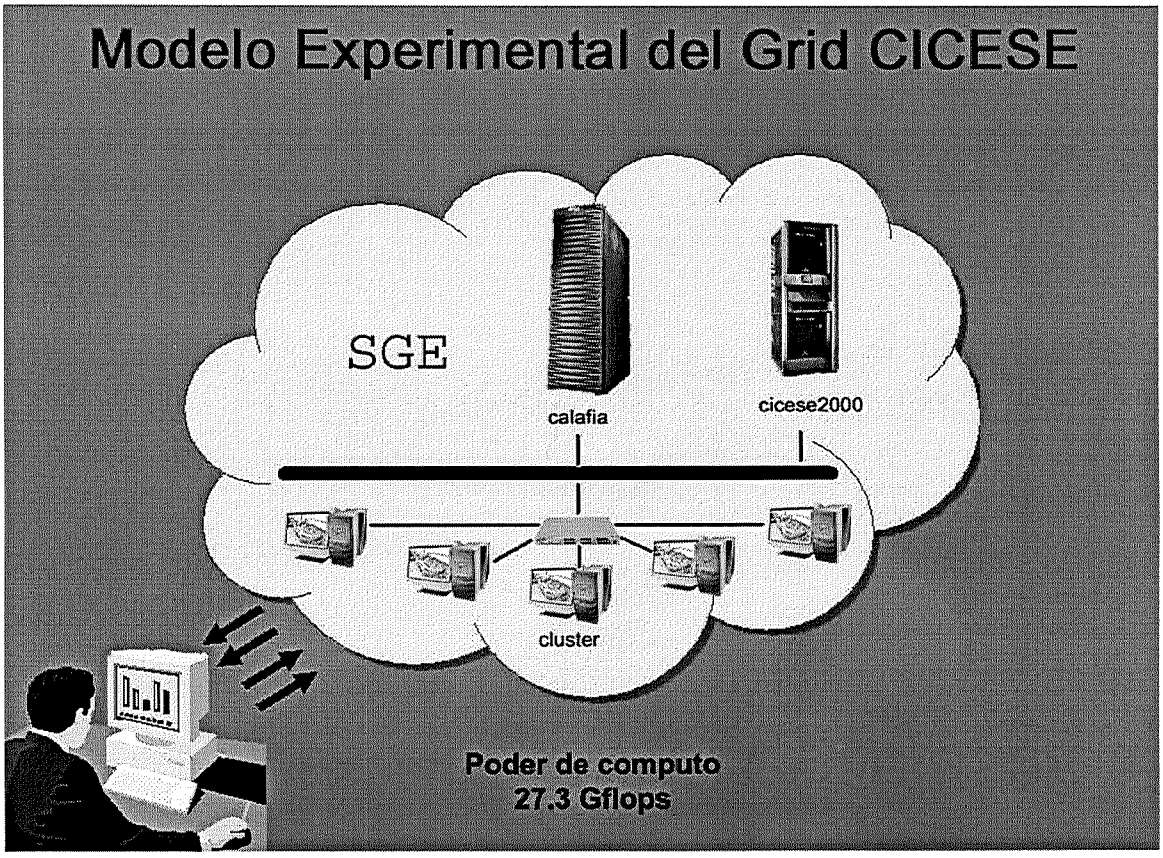


Figura 18. Modelo experimental del GRID-CICESE.

El software que se está utilizando para realizar la integración de los diferentes equipos de supercómputo es el Sun Grid Engine Software versión 5.3p3, el cual se describirá a continuación.

IV.3.1 Sun Grid Engine Software.

El software de Sun Grid Engine (SGE) proporciona todas las funciones tradicionales que un Administrador de Recursos Distribuidos (DRM) contiene. Tales funciones son:

- Encolamiento de tareas.
- Balanceo de carga de trabajo.
- Estadística de tareas.
- Petición de recursos específicos por el usuario.
- Petición de recursos a nivel cluster.
- Migración de tareas.

Además, el software de SGE también incluye mejoras como el shell qtcsh el cual permite utilizar aplicaciones interactivas dentro del SGE.

El software de Sun Grid Engine, Enterprise Edition (SGEEE) proporciona capacidades adicionales en el módulo de políticas. Así que, si quitamos este módulo, el software SGE y el SGEEE son idénticos. El módulo de políticas permite a los administradores realizar una compartición de recursos para diferentes departamentos y diferentes proyectos, por lo tanto, el SGEEE es el software considerado para la construcción de grids empresariales y el SGE es el software utilizado para la construcción de grids departamentales.

Hay cuatro tipos de máquinas lógicas en un entorno SGE, como es mostrado en la figura 19 y descrito en la tabla I. Dependiendo del tamaño, complejidad y deseo de escalabilidad del cluster grid, un grid puede ser configurado con uno o dos sistemas con múltiples roles lógicos.

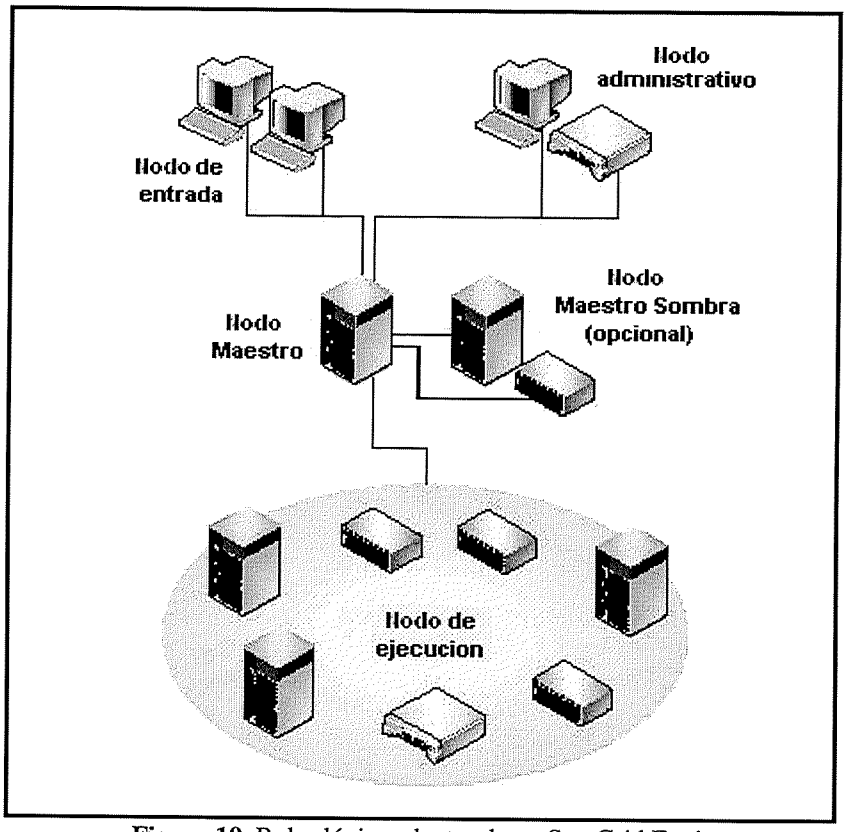


Figura 19. Roles lógicos dentro de un Sun Grid Engine.

Rol	Descripción
Maestro	Administra todas las peticiones de los usuarios, realiza la calendarización de tareas, y despacha las tareas a los equipos de ejecución. Siempre debe de existir un equipo maestro en cada implementación del software de SGE. Los demonios que se ejecutan en este equipo son los siguientes: • Schedd.- Es el demonio calendarizador que toma las tareas del área temporal y las manda a los equipos de ejecución, dependiendo de la prioridad de la tarea, requerimientos de la tarea, etc. • Qmaster.- Acepta las peticiones de las tareas y las pasa al demonio calendarizador, e implementa las decisiones de calendarizado realizadas por el demonio calendarizador.

Maestro Sombra	Mientras que existe un equipo Maestro, otras maquinas en el cluster pueden ser designadas como Maestro Sombra para proporcionar mayor seguridad en el grid. El Maestro sombra monitorea de manera continua al Maestro, y automáticamente y en forma transparente, asume el control en los eventos que fallan en el Maestro. Demonio: • Shadowd.- Monitorea la existencia de los demonios en el Maestro, y toma control de las funciones que fallan en ese equipo.
Ejecución	Las máquinas en el cluster que están disponibles para ejecutar tareas son llamadas *máquinas de ejecución.* Demonio: • Execd.- Acepta tareas del demonio qmaster y permite que se ejecute la tarea en ese equipo. Reporta la carga de trabajo al demonio maestro.
Entrada	Son máquinas configuradas para enviar, monitorear y administrar tareas. No se necesitan demonios en los equipos de entrada.
Administración	Son máquinas utilizadas para realizar cambios en la configuración del cluster, así como modificar algunos parámetros en el Administrador de Recursos Distribuidos (DRM), agregar nuevos nodos, agregar o cambiar usuarios. No se necesitan demonios.

Tabla I. Descripción de los roles lógicos dentro de un entorno SGE.

Flujo de tareas en el SGE.

Todas las tareas que son ingresadas al SGE, primeramente son tomadas por el maestro y puestas en un área temporal hasta que el calendarizador determine que la tarea esta lista para ejecutarse. El software de SGE verifica que existan los recursos suficientes para que se puedan ejecutar las tareas, tal como: suficiente memoria, tiempo de CPU, licencias de

software, etc. Tan pronto como los recursos apropiados estén disponibles para la ejecución de una nueva tarea, el software de SGE despacha la nueva tarea con la más alta prioridad.

El calendarizador de SGE toma en cuenta el orden en el que la tarea fue ingresada al grid, que máquinas están disponibles, y la prioridad del trabajo.

La siguiente descripción y la figura 20 ilustran el típico flujo de las tareas a través del SGE.

1.- Ingreso de tareas. Cuando un usuario ingresa una tarea desde una máquina de entrada, una petición de entrada es enviada al maestro.

2.- Calendarización de tareas. El maestro determina el equipo al cual le va a ser asignada la tarea. Verifica los recursos que necesite la tarea.

3.- Ejecución de tareas. Después de obtener la información del calendarizador, el maestro envía la tarea al los equipos de ejecución, éste la guarda en la base de datos de información de tareas, y comienza un proceso llamado "shepherd", el cual comienza la tarea y espera a que se complete.

4.- Información y estadística. Cuando la tarea se termina de ejecutar, el proceso "shepherd" regresa la información de la tarea y el equipo de ejecución reporta al maestro que ésta se ha terminado de ejecutar y la remueve de la base de datos. El maestro actualiza la base de datos de estadística de tareas para indicar que se ha completado el trabajo.

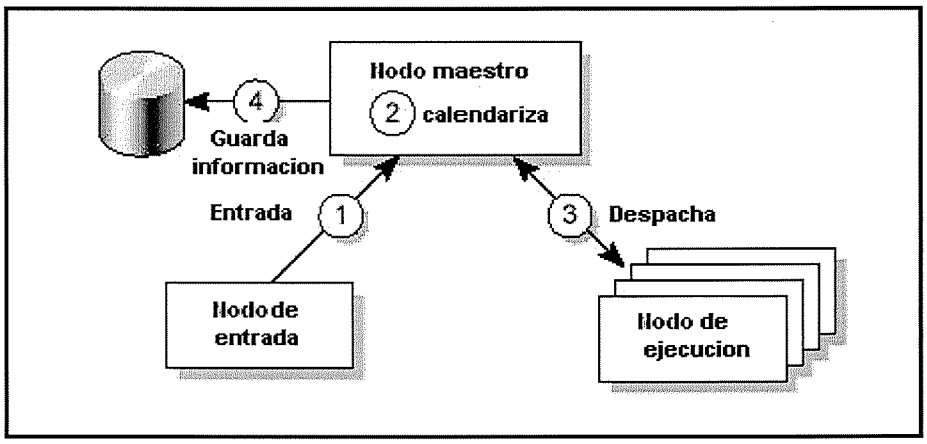


Figura 20. Flujo de tareas en el SGE.

Migración de tareas.

El SGE permite migrar las tareas de un equipo a otro en caso de que se suspenda la cola de trabajo o se suspenda la tarea misma. También permite utilizar el checkpointing con las aplicaciones que cuenten con esta habilidad.

¿Que es el checkpointing?

El checkpointing es un proceso que guarda, en una imagen en el disco, el estado en que se encuentra corriendo en ese momento una aplicación y, cuando se necesite volver a ejecutar dicha aplicación, ésta inicia desde donde se quedó. La imagen guardada se conoce como checkpoint. Es muy útil para cuando se desee migrar o abortar una aplicación sin la necesidad de volver a iniciar desde el principio, sino que se ejecute en donde se quedó. Se utiliza generalmente en aplicaciones que tardan horas o días en ejecutarse.

El SGE permite la creación de métodos para enviar señales a las tareas que se encuentren en los equipos de ejecución.

Paralelización.

El SGE permite ejecutar aplicaciones que requieran de un entorno paralelo. El SGE permite el uso de aplicaciones en PVM (Parallel Virtual Machine, Máquina Virtual Paralela) o MPI (Message Passing Inteface, Interfaz para el Envío de Mensajes).

Si los equipos de ejecución cuentan con algún entorno paralelo instalado, este puede ser integrado con el SGE para que funcione por medio del grid. Por ejemplo, si en los equipos de ejecución se tiene instalado el paquete de Sun HPC Cluster Tools, este puede ser configurado para que funcione mediante el grid. De la misma manera para MPICH o PVM.

Sensores de inactividad.

El SGE permite el uso de sensores de inactividad. Este sensor es muy útil para el caso de que se necesiten ingresar al grid, máquinas de usuarios las cuales pasen su mayor tiempo en estado ocioso. El sensor funciona de la siguiente manera:

La cola se habilita para ejecutar tareas cuando pasan x minutos que el usuario ha dejado de utilizar el teclado o de mover el ratón, en cuanto el usuario utiliza el teclado o el mouse, la cola se suspende y las tareas son migradas a otro equipo de ejecución.

Calendarios para las colas de trabajo.

El SGE permite la implementación de calendarios los cuales son útiles para manejar las políticas de uso de las colas de trabajo en el grid.

Por ejemplo, se pueden crear calendarios para que las colas de trabajo ejecuten tareas en paralelo todos los días por las noches o que se suspendan las colas por ser día festivo, etc.

Petición de recursos.

El SGE permite la petición de recursos tales como, tiempo de CPU, memoria RAM, espacio en disco, licencias de software, sistema operativo o arquitectura, etc.

La petición puede ser a nivel máquina o a nivel cluster.

En caso de contar con los recursos, el SGE ejecuta la aplicación en los equipos los cuales cuenten con dichos recursos.

Manejo y administración de usuarios.

En el SGE hay dos tipos de usuarios:

Administradores. Son encargados de administrar el SGE en todos sus módulos, como configuración del entorno, creación de usuarios, etc, no tienen ninguna restricción administrativa.

Operadores. Son encargados de administrar el SGE pero con algunas restricciones administrativas las cuales son declaradas por los Administradores.

También se pueden crear listas de usuarios los cuales tengan ciertas restricciones, como por ejemplo: Que dicha lista de usuarios solo puedan ejecutar tareas en ciertas colas de trabajo, o que solo tengan cierto tiempo de uso de procesador.

IV.3.2 Especificaciones técnicas del GRID-CICESE.

A las máquinas que conforman el GRID-CICESE se les instaló el Sun Grid Engine quedando configurado como lo muestra la Tabla II.

	Maestro	Ejecución	Entrada	Administrativo	Plataforma
Cluster tribus					
Kumiai	X	X	X	X	Solaris 8
Kiliwa		X	X	X	Solaris 8
Cucapa		X	X	X	Solaris 8
Paipai		X	X	X	Solaris 8
Cochimi		X	X	X	Solaris 8
Supercomputadora					
Calafia		X	X	X	Solaris 8
cicese2000		X	X	X	Irix 6.4

Tabla II. Funciones de cada nodo dentro del grid.

Todos los nodos que conforman el GRID-CICESE pueden enviar tareas (entrada), ejecutar tareas (ejecución), realizar operaciones administrativas y el nodo *kumiai* se configuró como nodo maestro del grid.

Una vez instalado el software se definieron las colas de trabajo necesarias para que el usuario pueda enviar sus tareas.

Se establecieron dos entornos para el envío de tareas. La Tabla III describe las colas de trabajo y el tipo de tarea que se ejecuta en cada equipo del GRID-CICESE.

Colas de trabajo en serie: Son utilizadas para ejecutar tareas secuenciales e interactivas en el grid, esto significa, que las tareas que se ingresen en esta cola no podrán ser de tipo paralelas (que utilicen MPI o MPICH).

Colas de trabajo en paralelo: Son utilizadas para ejecutar tareas que necesiten de un entorno paralelo (MPI o MPICH). El cluster Tribus y Calafia utilizan el SUN HPC Cluster Tools para ejecutar trabajos en paralelo, en Cicese2000 se tiene instalado MPICH.

Host	En serie	En paralelo	Procesadores
Cluster tribus			
kumiai	kumiai.q	kumiai.cre	1
kiliwa	kiliwa.q	Kiliwa.cre	1
cucapa	cucapa.q	cucapa.cre	1
paipai	paipai.q	paipai.cre	1
cochimi	cochimi.q	cochimi.cre	1
Supercomputadora			
calafia	calafia.q	calafia.cre	8
cicese2000	cicese2000.q	cicese2000.mpi	8

Tabla III. Colas de trabajo en el grid.

Las colas de trabajo en serie y las colas de trabajo en paralelo se encuentran sincronizadas para evitar la sobrecarga de los equipos de ejecución. La figura 21 muestra los equipos de ejecución dentro del grid utilizando la interfaz gráfica propia del SGE "qmon".

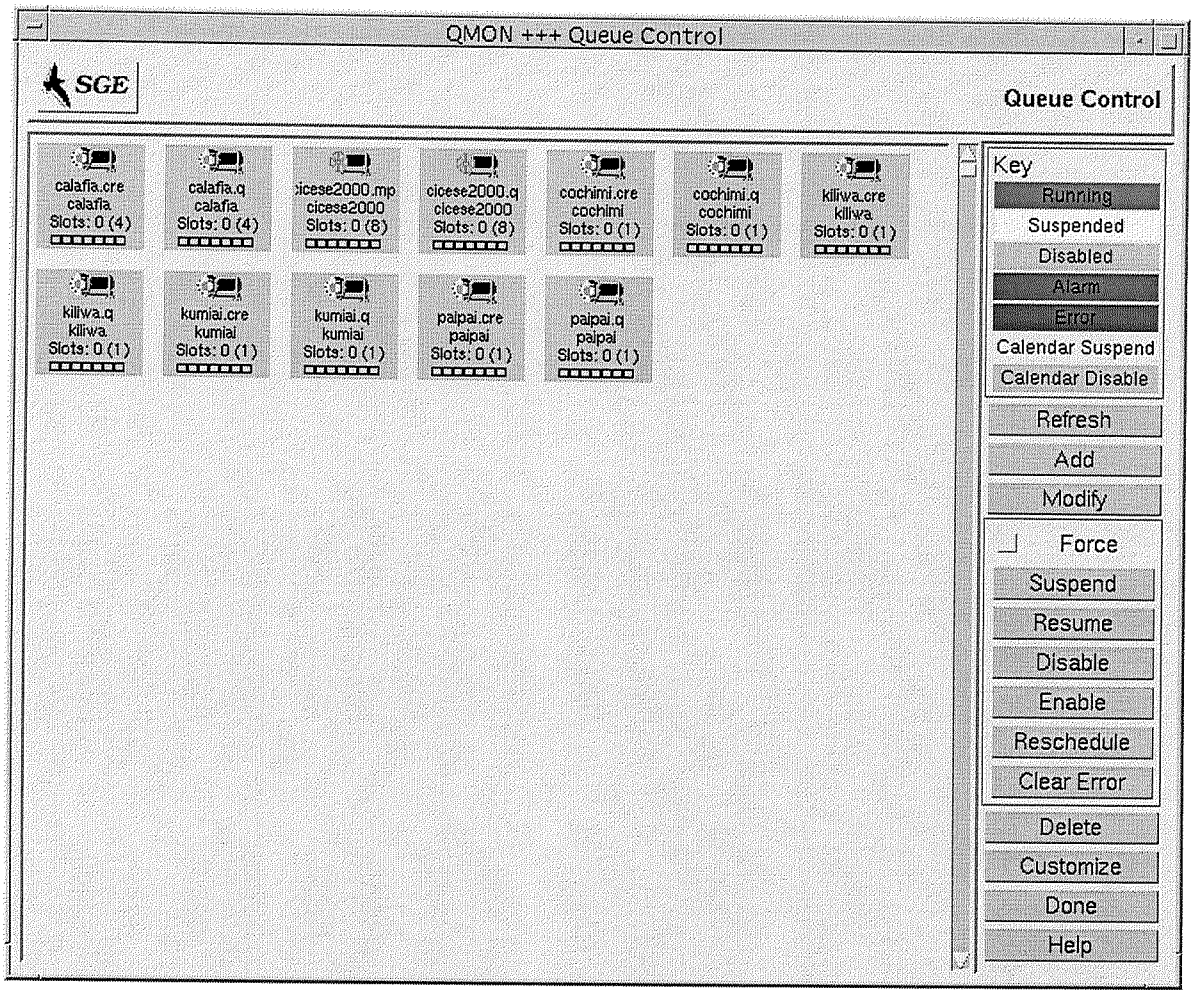


Figura 21. Nodos de ejecución dentro del GRID-CICESE.

Entorno paralelo en el GRID-CICESE.

Para ejecutar tareas en paralelo en el GRID-CICESE, actualmente se está manejando paralelización en el cluster tribus y en la supercomputadora Calafia, en Cicese2000 se está manejando paralelización con el paquete MPICH en forma independiente debido a la arquitectura de esta máquina.

En Calafia, para ejecutar tareas en paralelo se tiene instalado el paquete Sun HPC Cluster Tools 5. Para hacer la integración de este paquete con el SGE, se siguieron los pasos para

realizar una "integración ligera" entre el SGE y el Sun HPC Cluster Tools 5 que vienen en la documentación.

La cola para ejecutar tareas en paralelo en Calafia se llama "calafia.cre", cuenta con 4 procesadores. En el SGE, para identificar a esta integración, se creó un entorno paralelo llamado "hpc-calafia".

En el cluster tribus para ejecutar tareas en paralelo, se tiene instalado el paquete Sun HPC Cluster Tools 5. Para hacer la integración de este paquete con el SGE, se siguieron los pasos para realizar una "integración ligera" entre el SGE y el Sun HPC Cluster Tools 5 que vienen en la documentación.

Las colas para ejecutar tareas en paralelo en cada máquina del cluster se llaman "kumiai.cre, kiliwa.cre, paipai.cre, cochimi.cre y cucapa.cre", cuentan con 5 procesadores en total. En el SGE, para identificar a esta integración, se creó un entorno paralelo llamado "hpc-cluster".

En Cicese2000 la cola para ejecutar tareas en paralelo se llama "cicese2000.mpi" y cuenta con 8 procesadores. Se realizó una integración entre MPICH y SGE. En el SGE, para identificar a esta integración, se creó un entorno paralelo llamado "hpc-cicese2000".

Migración de tareas en el GRID-CICESE.

Se configuraron algunas aplicaciones para que éstas, en caso de suspenderse la cola de ejecución en la que se encuentren corriendo, migren a otro equipo de ejecución y vuelvan a ejecutarse desde el principio.

Si se desea que estas aplicaciones no vuelvan a ejecutarse desde el principio sino desde

donde se quedaron corriendo la última vez, se tiene que implementar el checkpointing, ya sea a un nivel de la aplicación misma, por definición del usuario o por el kernel del sistema operativo. Por ejemplo, si el usuario envía al grid una tarea implementada con checkpointing y el nodo asignado se suspende por sobrecarga, esta tarea migrará a otro nodo y continuará con su ejecución desde donde se quedó.

En Calafia y el cluster Tribus, debido a que el sistema operativo Solaris no permite el checkpoint a nivel kernel, se tienen que utilizar bibliotecas independientes para poder lograr el checkpointing de tareas. El SGE permite utilizar las bibliotecas del Condor Project.

Estas bibliotecas solo permiten el checkpointing con tareas secuenciales (seriales), no para tareas en paralelo.

Para realizar el checkpointing, primeramente se tiene que enlazar el programa o tarea que se quiere ejecutar, con las bibliotecas de Condor. Se tiene que compilar el archivo fuente o el archivo objeto, utilizando algunas herramientas de Condor para ligarlo con las bibliotecas y así darle la habilidad de checkpointing.

Actualmente, ya se han ligado varios programas y, manualmente, pueden crear una imagen de su estado y volver a ejecutarse en donde se quedaron.

Para realizar el checkpointing de tareas sin intervención del usuario, se puede configurar el SGE para que realice el checkpoint de manera automática, ya sea en algún intervalo de tiempo, por la suspensión de la cola o la tarea, y en caso de ser necesario, migre la tarea a otro nodo para que continúe con la ejecución de la tarea desde donde se quedó, dependiendo de la ultima imagen que se haya creado.

El SGE permite la creación de métodos para enviar señales a las tareas que se encuentren ejecutándose dentro del grid. En caso de aplicaciones ligadas a las bibliotecas de Condor, se envían las señales SIGUSR2 y SIGTSTP. El método se llama *condor_ckpt*.

Con SIGUSR2, la aplicación crea una imagen en disco de su estado actual y continua con su ejecución. Con SIGTSTP la aplicación crea una imagen en disco de su estado actual y aborta su ejecución.

En Cicese2000, no se necesita utilizar alguna biblioteca independiente, ya que el sistema operativo Irix 6.4 soporta el checkpointing a nivel kernel (por medio del sistema operativo) utilizando el comando *cpr*. Se realizó la implementación de este comando con el SGE para que se realice el checkpointing de aplicaciones que así lo requieran. El método se llama *cicese2000_ckpt*.

Sensores de inactividad en el GRID-CICESE.

Actualmente, se tiene implementado un sensor de inactividad en una máquina del cluster (kiliwa). El sensor de inactividad que se está utilizando es uno que está incluido en el paquete de SGE.

Este sensor es muy útil para el caso de que se necesiten ingresar al grid maquinas de usuarios las cuales pasen su mayor tiempo en estado de ocioso (idle). El sensor funciona de la siguiente manera:

La cola se habilita para ejecutar trabajos cuando pasan 3 minutos de que el usuario ha dejado de utilizar el teclado o de mover el ratón, en cuanto el usuario utiliza el teclado o el ratón, la cola se suspende.

Tareas interactivas en el GRID-CICESE.

Se pueden ejecutar tareas interactivas y gráficas en el SGE utilizando algunos comandos como: qsh y qrsh.

Por medio de estos comandos se ha podido ejecutar Matlab, OpenDX y Opnet en los nodos de ejecución del SGE que se encuentren mas desocupados o con menos carga de trabajo.

Compartición de datos en el GRID-CICESE

Sun Grid Engine no permite unificar recursos de almacenamiento por lo que la compartición de datos se hizo mediante las herramientas propias del sistema operativo tal como lo es NFS (Network File System).

Los datos que se están compartiendo entre Calafia y el cluster Tribus son los directorios de los usuarios y los discos de alta velocidad (T3 Sun StoreEdge) que almacenan información temporal.

En Cicese2000 aún está pendiente la compartición de la información entre esta, el cluster y Calafia.

V. DISEÑO Y EJECUCION DE PRUEBAS

Para verificar el funcionamiento del GRID-CICESE, se realizaron las siguientes pruebas:

V.1) Distribución de tareas secuenciales en el grid.

V.2) Palalelización con MPI en el grid.

V.3) Tolerancia a fallas (Migración de tareas).

Las pruebas fueron documentadas bajo el siguiente esquema:

1) Descripción de la prueba a realizar.

2) Pasos para realizar la prueba.

3) Desarrollo de la prueba.

4) Conclusiones.

V.1 Distribución de tareas secuenciales en el grid.

1) Descripción breve de la prueba a realizar:

En esta prueba, se enviaran al grid varias tareas en forma simultánea.

Se estará monitoreando el comportamiento de las colas de ejecución en el grid y de esta forma verificar el balanceo de carga de trabajo que presenta.

2) Pasos para realizar la prueba.

- **a)** Se verificará el estado actual de la carga de los nodos de ejecución.
- **b)** Se enviarán las tareas al grid.
- **c)** Se dará seguimiento a la ejecución de las tareas mediante la herramienta *qmon* para verificar la distribución de tareas secuenciales entre Calafia y el cluster.

3) Desarrollo de la prueba.

a) Para verificar la carga de los nodos de ejecución se utiliza el comando *qstat*. El campo *load_avg* indica la carga actual de cada nodo dentro del grid, como lo muestra la figura 22.

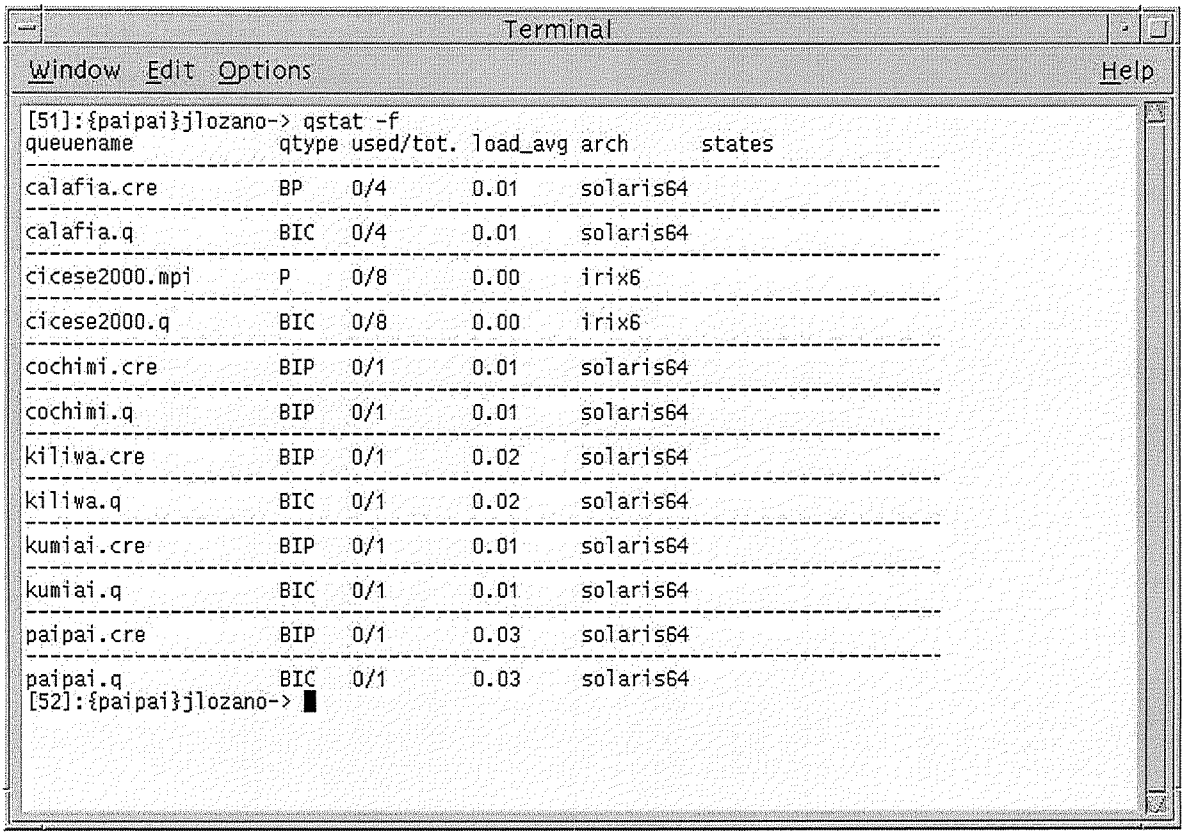


Figura 22. Verificación de la carga de trabajo en los nodos del Grid.

En la supercomputadora Calafia, el promedio de carga máximo es de 12.50, si se llegara a dar ese caso, Calafia se encontraría trabajando casi al 100% de su capacidad. Lo mismo sucedería para Cicese2000. Para las máquinas del cluster, el promedio de carga máximo es de 1.00.

En este caso, todos los nodos de ejecución se encuentran disponibles.

b) Para enviar las tareas al grid, se desarrolló el script de entrada *prueba_serial.csh*:

```
#!/bin/csh
#  Script para probar la distribución de tareas dentro del grid-cicese
qsub -cwd -l a=solaris64 btA.csh
qsub -cwd -l a=solaris64 cgA.csh
qsub -cwd -l a=solaris64 epA.csh
qsub -cwd -l a=solaris64 ftA.csh
qsub -cwd -l a=solaris64 isA.csh
exit
#fin del archivo prueba_serial.csh
```

Donde:

qsub -> Es el comando que se utiliza para enviar tareas al grid.
-cwd -> Indica que la tarea a enviar se encuentra en el directorio actual.
-l -> Indica que se va a realizar la petición de algún recurso para condicionar la ejecución de la tarea.
a= solaris64 -> Abreviatura de arch (arquitectura). Se le está solicitando al grid que el host que ejecute dicha tarea sea de arquitectura solaris 64 bits.
btA.csh .. isA.csh -> Son los scripts que contienen los binarios a ejecutar.

En la ejecución de las pruebas los binarios fueron tomados de los Benchmarks de NAS. Los Benchmarks de NAS son un conjunto de programas diseñados para evaluar el rendimiento de supercomputadoras, clusters, estaciones de trabajo, etc [Nasft 03]. Hay varias versiones de estos programas: en paralelo y en serie. La versión serial es la que se utilizó para verificar la distribución de tareas en el grid.

Al ejecutar el script de entrada se obtuvo la siguiente salida:

```
[73]:{paipai}jlozano-> prueba_serial.csh
your job 674 ("btA.csh") has been submitted
your job 675 ("cgA.csh") has been submitted
your job 676 ("epA.csh") has been submitted
your job 677 ("ftA.csh") has been submitted
your job 678 ("isA.csh") has been submitted
```

c) Las tareas fueron enviadas al grid. El nodo maestro del grid verifica que nodo de ejecución es el que tiene la menor carga de trabajo y además, que cumpla con los requisitos para que ejecute la tarea. La figura 23 muestra como se distribuyeron las tareas en el grid.

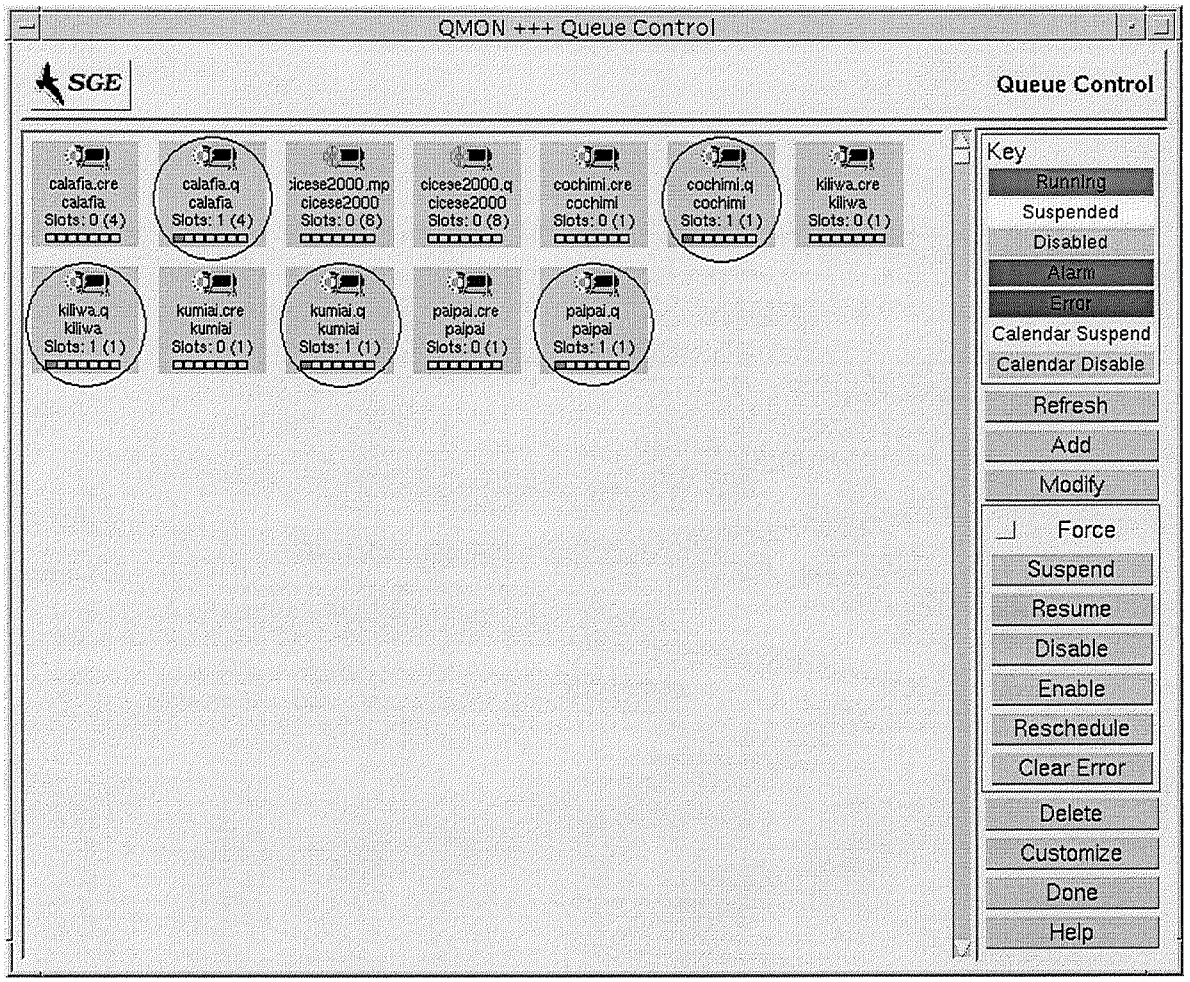


Figura 23. Distribución de tareas secuenciales en el grid.

Las colas en las que se ejecutaron las tareas son las que terminan con ".q" ya que estas están configuradas para que solo ejecuten tareas seriales o interactivas, no para tareas que necesiten de un entorno paralelo.

Como se observa en la Figura 23, las tareas fueron ejecutadas solo en los equipos que cuentan con la arquitectura Solaris 64 bits, tal es el caso de la supercomputadora Calafia y el cluster Tribus.

4) Conclusiones.

Al término de esta prueba, se verificó la apropiada distribución de tareas que proporciona el grid, además de que el usuario puede requerir de ciertos recursos como arquitectura, tiempo de cpu, memoria RAM disponible, almacenamiento en disco, entre otros y estar seguro de que su tarea solo se va a ejecutar si los nodos de ejecución cumplen con esos requisitos.

Como se pudo observar en el desarrollo de la prueba, otro de los aspectos en los que se mejoró con la implementación de este grid, es que el usuario no tiene que especificar que nodo va a ejecutar su tarea, solo se enfoca en enviar la tarea al grid y este se encarga de distribuirlo a los demás nodos, a esto se le conoce como *transparencia,* que es una de las características principales del GRID.

V.2 Paralelización con MPI en el GRID.

1) Descripción de la prueba a realizar.

Como se describió en el capítulo anterior, se pueden enviar tareas al Grid-CICESE que necesiten de un entorno paralelo utilizando la integración entre el Sun HPC Cluster Tools 5 (para Calafia y el cluster Tribus), MPICH (para Cicese2000) y el Sun Grid Engine.

Para verificar el funcionamiento del entorno paralelo configurado en el Grid, se utilizarán aplicaciones basadas en MPI, se desarrollarán los scripts de entrada para dichas tareas, se estará monitoreando la ejecución de éstas tareas y, una vez finalizada su ejecución, se comparará su desempeño. También se describirá el funcionamiento de una aplicación real dentro del grid.

2) Pasos para realizar la prueba.

a) Se enviará una tarea en paralelo por medio del grid a la supercomputadora Calafia (*hpc-calafia*).

b) Se enviará una tarea en paralelo por medio del grid al cluster tribus (*hpc-cluster*).

c) Se comparará el tiempo de ejecución de algunas aplicaciones para verificar la integración entre en SGE y el HPC Cluster Tools.

d) Se dará a conocer el funcionamiento de una aplicación real tanto en Calafia como en el cluster.

e) Se enviará una tarea en paralelo por medio del grid a la supercomputadora Cicese2000 (*hpc-cicese2000*).

3) Desarrollo de la prueba.

a) Primero se va a enviar una tarea en paralelo usando el Grid. Para Calafia y el cluster Tribus, el envío de tareas en paralelo se hace mediante un script, por ejemplo, el script parluA4.csh contiene lo siguiente:

```
#!/bin/csh
# Script de entrada entorno paralelo cluster - calafia
$SGE_ROOT/mpi/sunhpc/loose-integration/MPRUN -np $NSLOTS lu.A.4
#fin del archivo parluA4.csh
```

Donde:
lu.A.4 -> es el binario a ejecutar.

Posteriormente se realizo el script *prueba_par.csh*, el cual contiene las instrucciones para enviar la tarea al grid.

```
#!/bin/csh
#  Script prueba_par.csh
#  Script para probar el entorno paralelo dentro del grid-cicese.  Supercomputadora calafia
qsub -cwd -l cre -pe hpc-calafia 4 parluA4.csh
exit
```

Al ejecutar el script de entrada se obtuvo la siguiente salida:

```
[74]:{paipai}jlozano-> prueba_par.csh
your job 679 ("parluA4.csh") has been submitted
```

La figura 24 muestra que la tarea se está ejecutando solo en Calafia con 4 procesadores.

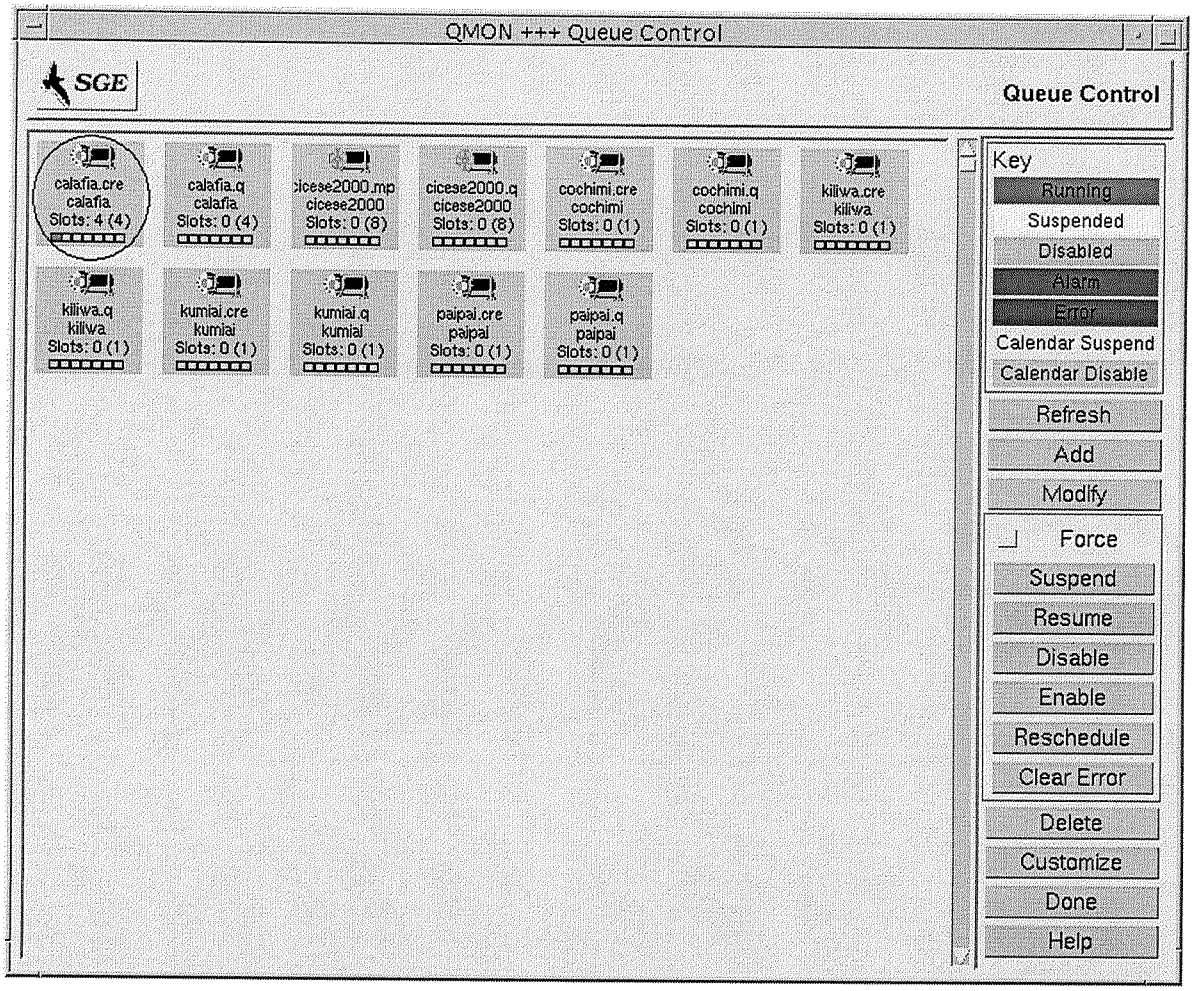


Figura 24. Calafia ejecutando aplicaciones en paralelo dentro del grid.

La cola de ejecución para tareas en paralelo de la supercomputadora calafia es "calafia.cre" que solo cuenta con 4 procesadores para este tipo de tareas, esto debido a las políticas de uso del equipo [Casdel 02].

b) La siguiente prueba será enviar otra tarea en paralelo al cluster por medio del Grid. Se modificó, en el script de entrada utilizado en Calafia, la línea que indica el entorno paralelo

a utilizar, cambiando *hpc-calafia* por *hpc-cluster* como se muestra en el script prueba_par2.csh.

```
#!/bin/csh
# Script para probar el entorno paralelo dentro del grid-cicese
# Cluster tribus
qsub -cwd -l cre -pe hpc-cluster 4 parluA4.csh
exit
#Fin del archivo prueba_par2.csh
```

Al ejecutar el script de entrada se obtuvo la siguiente salida:

```
[75]:{paipai}jlozano-> prueba_par2.csh
your job 680 ("parluA4.csh") has been submitted
```

La figura 25 muestra las tareas que se están ejecutando en el Grid. Tanto calafia como el cluster se encuentran ejecutando tareas en paralelo.

c) Para verificar que el ejecutar tareas en paralelo en el grid no afecte al rendimiento de la aplicación, se compararon las corridas de los mismos programas dentro del grid (integración) y de manera independiente. Esta prueba se realizó en el cluster y los resultados se muestran en la Tabla IV.

Programa	Num. Procesadores	Independiente (seg)	Integracion (seg)
1.- Lu.A	2	399.23	403.62
2.- Cg.W	2	3.58	3.62
3.- Ep.A	2	66.21	68.03
4.- Lu.A	4	200.28	202.06
5.- Cg.W	4	3.79	4.31
6.- Ep.A	4	36.35	38.04

Tabla IV. Resultados de rendimiento utilizando los benchmarks de NAS en el cluster Tribus.

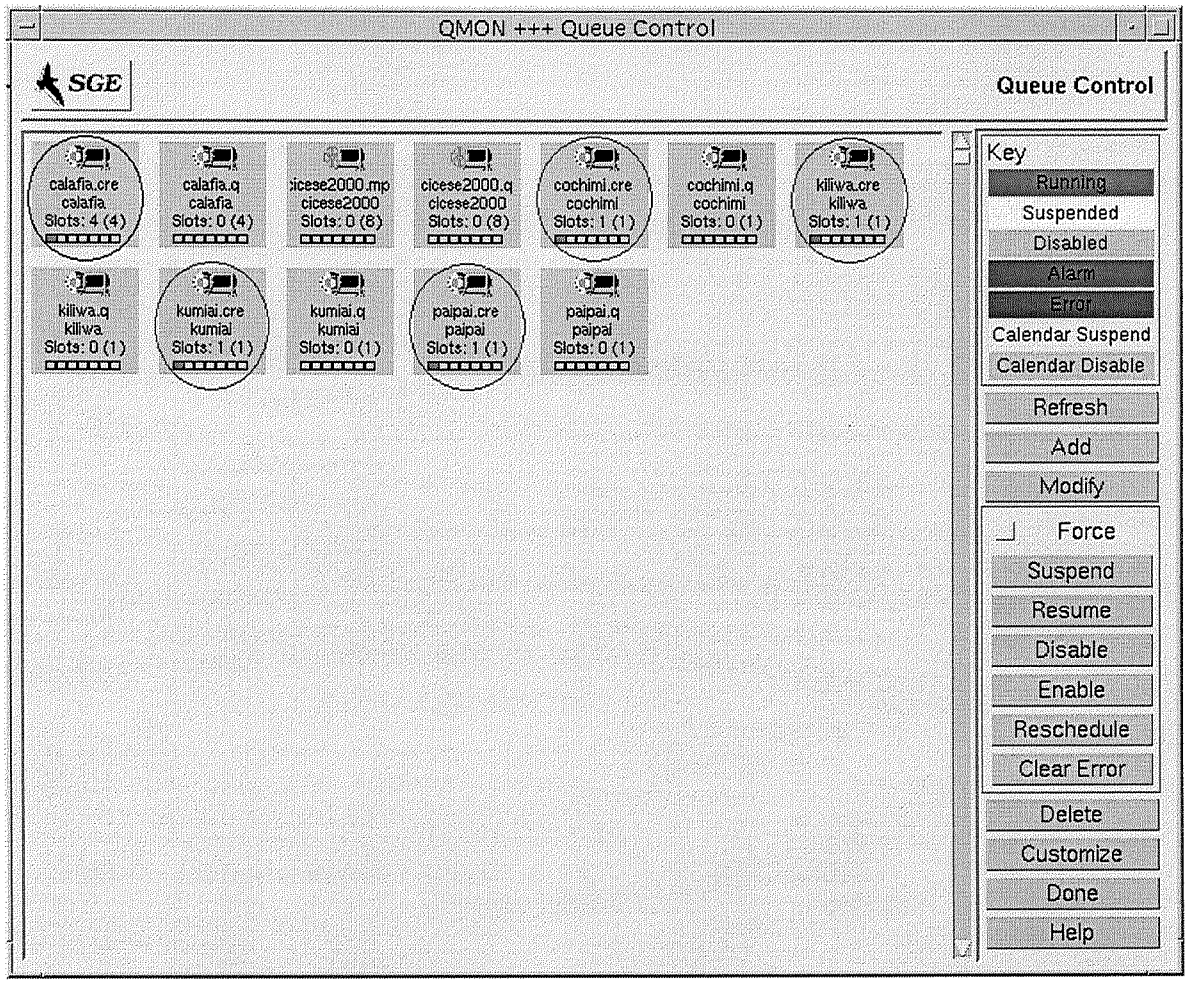


Figura 25. Calafia y el cluster Tribus ejecutando aplicaciones en paralelo.

En la Tabla IV, el campo *independiente* muestra el tiempo de duración del programa en segundos utilizando el paquete de Sun HPC Cluster Tools 5 sin el SGE, y el campo *integración* muestra el tiempo de duración del programa en segundos utilizando la integración entre el paquete de Sun HPC Cluster Tools y el Sun Grid Engine.

La figura 26 muestra una comparativa de los resultados obtenidos anteriormente. Como se puede observar, el retardo es mínimo y es debido a la recopilación de resultados por parte del calendarizador del Grid y también por el hecho de verificar la disponibilidad y la carga de los nodos antes de asignar la tarea.

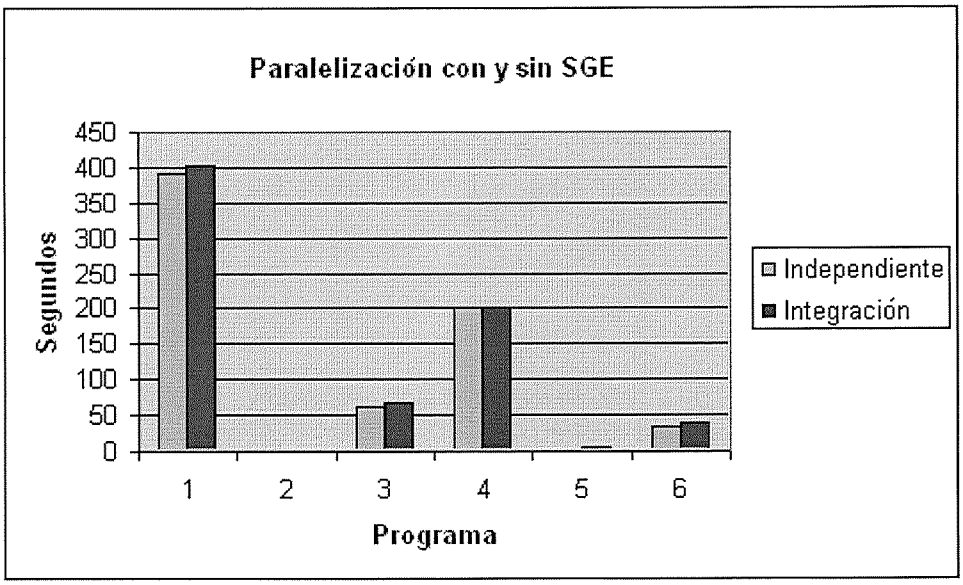


Figura 26. Comparativa de MPI con SGE y sin SGE.

d) Una de las aplicaciones reales que se ejecutan en la supercomputadora Calafia es el modelo MM5 (Mesoscale Model Fifth-Generation) [Mmc 03], que es utilizado por el departamento de Oceanología de CICESE para investigación atmosférica (pronóstico metereológico). Este modelo se ejecuta en su versión secuencial y en su versión paralela, su gran demanda de CPU y memoria han llegado a afectar el desempeño de la supercomputadora a un grado de ocasionar incomodidad entre los usuarios debido a la lentitud de la máquina.

Esta aplicación tarda aproximadamente 8 horas con 35 minutos en ejecutarse utilizando 6 procesadores de la supercomputadora calafia. Al probarse esta aplicación dentro del grid, se utilizaron 4 procesadores del cluster tribus y esta aplicación duró aproximadamente 8 horas con 55 minutos.

El cluster ejecutó el modelo MM5 con 4 procesadores casi con el mismo tiempo de ejecución que Calafia con 6, esto debido a que la aplicación consume, además de tiempo de procesador, bastante memoria RAM (600MB en cada hijo).

Con la distribución de tareas a través del Grid, se permite balancear la carga de trabajo disminuyendo la posibilidad de saturación de los recursos de los equipos más demandados.

f) En Cicese2000, para ejecutar tareas en paralelo a través del grid, se realiza mediante un script como el que a continuación se muestra:

```
#!/bin/csh
# Script de entrada entorno paralelo cicese2000
mpirun –np $NSLOTS cg.W.4

#fin del archivo parcgW4.csh
```

Donde:
cg.W.4 es el binario a ejecutar.

Para enviar la tarea al grid, se ejecutó el siguiente script:

```
#!/bin/csh
#  Script para probar el entorno paralelo dentro del grid-cicese. Super cicese2000
qsub -pe hpc-cicese2000  4 parcgW4.csh
exit
#Fin del archivo prueba_par3.csh
```

Se obtuvo la siguiente salida:

```
[76]:{paipai}jlozano-> prueba_par3.csh
your job 681 ("parcgW4.csh") has been submitted
```

La figura 27 muestra que la tarea se está ejecutando en Cicese2000 en 4 procesadores de 8 disponibles.

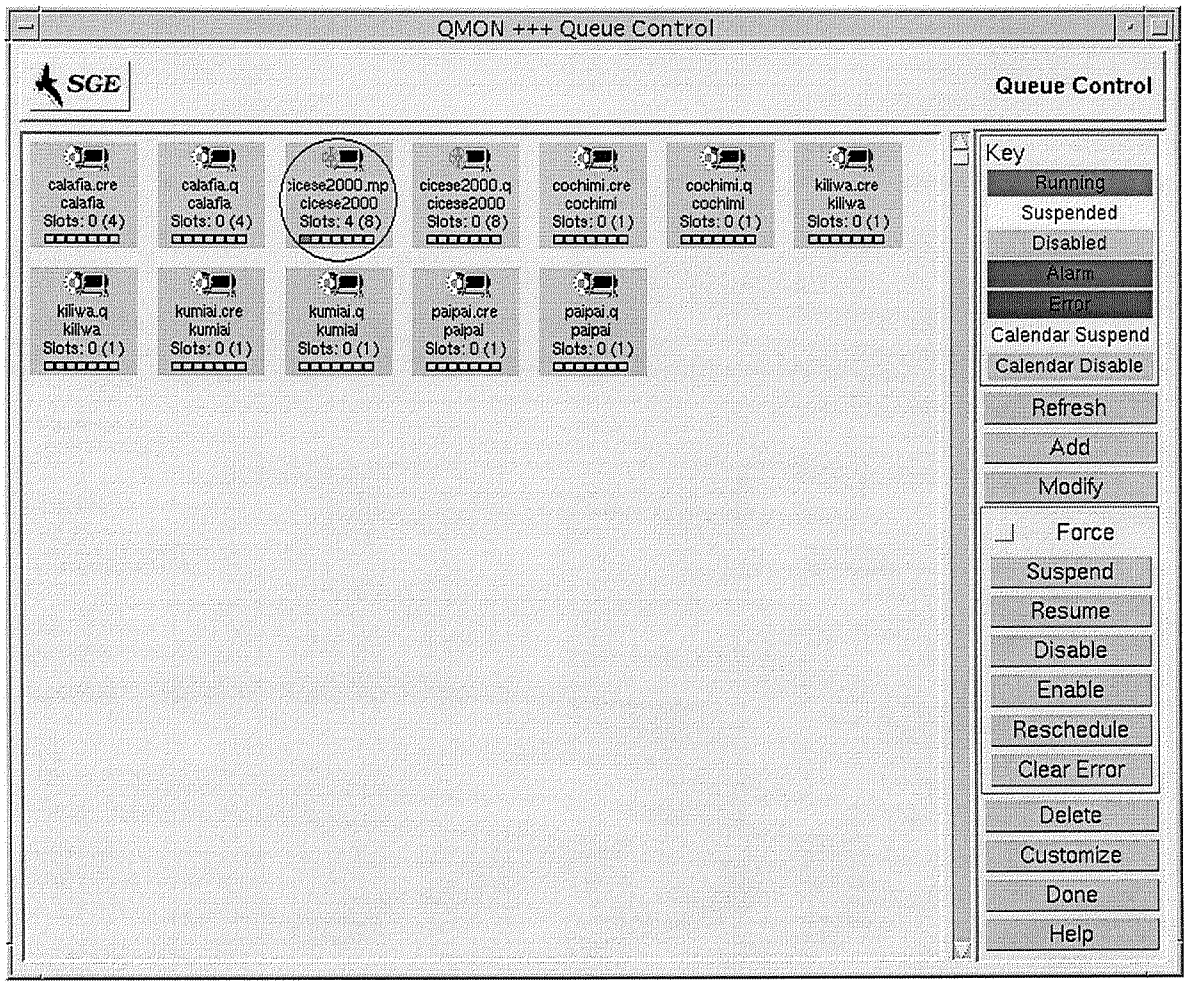


Figura 27. Supercomputadora cicese2000 ejecutando tareas en paralelo por medio del grid.

4) Conclusiones.

Como se observó en el desarrollo de esta prueba, el tiempo de retardo entre la integración de SGE con el HPC Cluster Tools es mínimo. Con la aplicación real se obtuvo un buen rendimiento y la aceptación por parte del usuario, así que ya se contempla la posibilidad de configurar otro tipo de aplicaciones para que estas puedan ser ejecutadas por medio del grid.

Con esta integración, se puede resolver la problemática de asignación de recursos (tiempo de procesador) para aplicaciones en paralelo ya que se pueden implementar calendarios que gobiernen a las colas de ejecución para que solo se ejecuten este tipo de tareas en ciertas horas del día, y de esta manera, evitar la sobrecarga de los equipos.

V.3 Tolerancia a fallas (Migración de tareas).

1) Descripción de la prueba a realizar.

En esta prueba, se enviará al grid una aplicación de los benchmarks de NAS en su versión serial. Al suspenderse la cola de ejecución del nodo en el cual se encuentre corriendo, la aplicación deberá migrar a otro nodo y comenzar a ejecutarse de nuevo. También se describirá el funcionamiento del grid con aplicaciones que puedan implementar el checkpointing.

2) Pasos para realizar la prueba.

a) Se enviará al grid el programa secuencial a ejecutar.

b) Al estar ejecutando el grid la tarea, se suspenderá la cola de ejecución del nodo el cual se encuentre corriendo el programa.

c) Se verificará que el programa migre a otro nodo que este más desocupado y comience a ejecutarse de nuevo.

d) Se describirá el funcionamiento de aplicaciones con la habilidad de checkpointing.

3) Desarrollo de la prueba.

a) La aplicación a ejecutar se encuentra en el script *btA.csh*:

```
#!/bin/csh
./bt.A
#fin del archivo btA.csh
```

Se envió la aplicación mediante el siguiente script:

```
#!/bin/csh
#
# Script para probar la migración de tareas en el grid
qsub -cwd -l a=solaris64 –ckpt reubicar btA.csh
#fin del archivo prueba_migracion.csh
```

donde:

-ckpt -> Indica al SGE que se utilizará un método para migración de tareas.
reubicar -> Método que contiene la información para que el SGE recalendarice la tarea y la reinicie en caso de suspensión de la cola de ejecución.

Se obtuvo la siguiente salida de confirmación de envió:

```
[77]:{paipai}jlozano-> prueba_migracion.csh
your job 682 ("btA.csh") has been submitted
```

La figura 28 muestra que la tarea se está ejecutando en el nodo kumiai, en la cola de ejecución llamada "*kumiai.q*".

b) Los casos en los que se puede suspender una cola de trabajo pueden ser: en forma manual, por sobrecarga del nodo, por algún calendario, por algún sensor de inactividad o por la caída del demonio de ejecución.

En este caso, se procedió a realizar la suspensión de la cola de ejecución en forma manual a través de la interfaz gráfica (qmon).

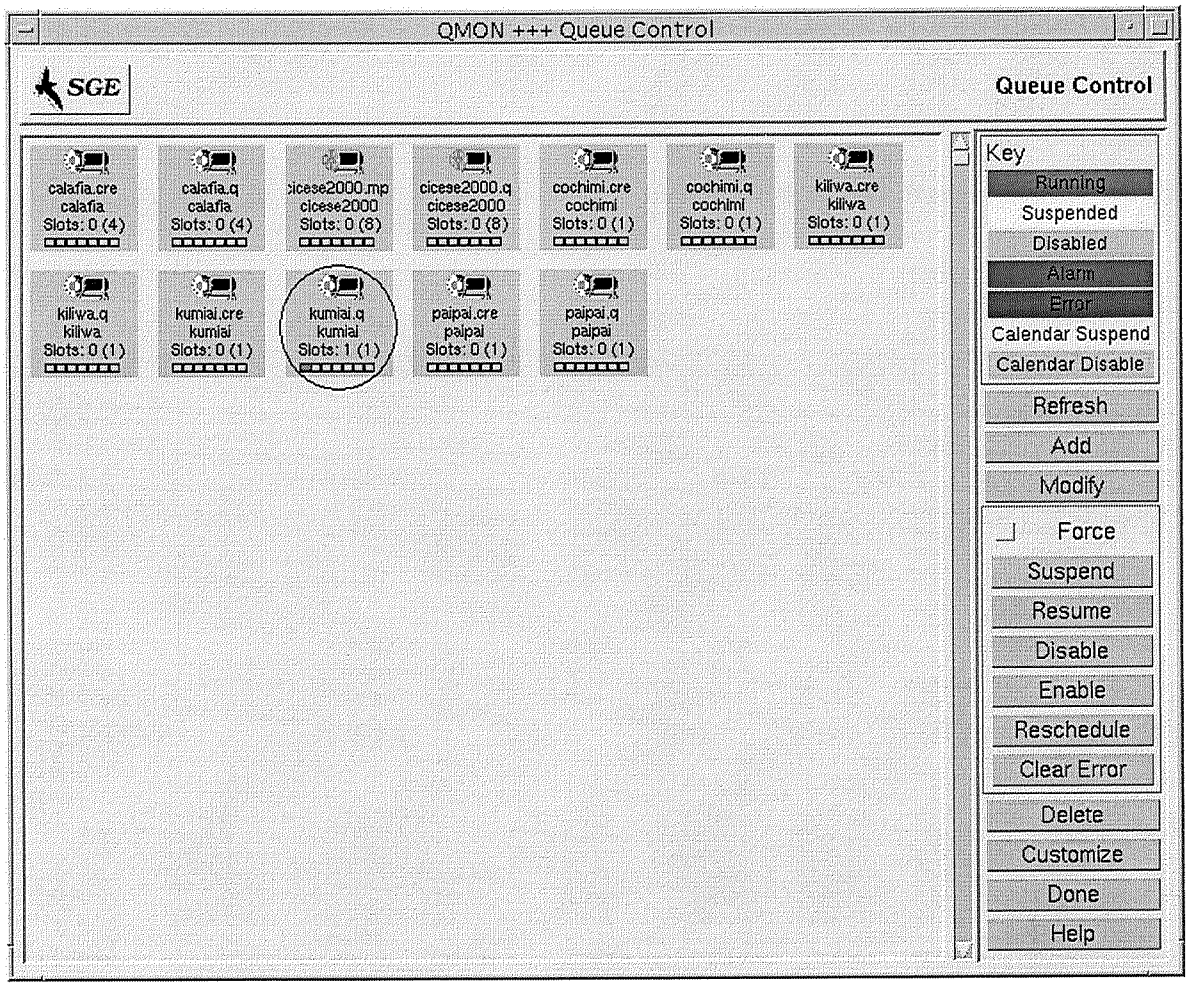


Figura 28. Nodo kumiai ejecutando la tarea antes de la suspensión.

c) Al suspenderse la cola de ejecución "*kumiai.q*", la tarea se recalendarizó y migró a otro nodo que contaba con menos carga, tal como se muestra en la figura 29.

En la figura 29, la cola "kumiai.q" se encuentra suspendida mientras que "kiliwa.q" comienza a ejecutar de nuevo la tarea.

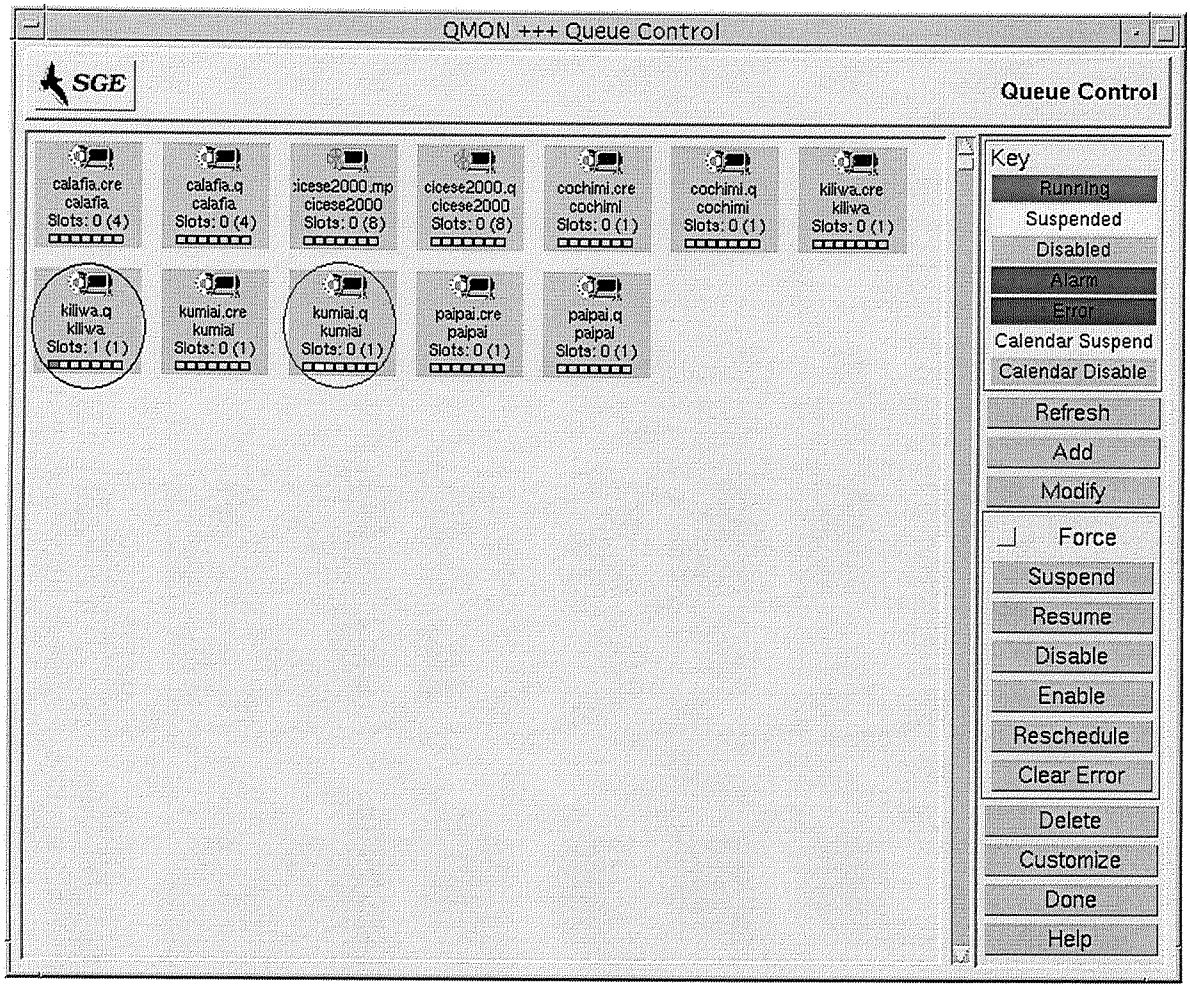


Figura 29. Migración de tareas. De kumiai a kiliwa después de la suspensión.

d) A continuación se describirá el proceso para ejecutar aplicaciones que cuenten con la habilidad de checkpointing.

La aplicación a utilizar es un programa desarrollado en lenguaje c. El nombre del programa es *prueba.c*.

La aplicación dura aproximadamente 6 minutos en ejecutarse tanto en Calafia como en el cluster Tribus. Para el caso de Calafia y el cluster Tribus, se utilizó el método *condor_ckpt*,

para lograr el checkpointing de la aplicación y la forma de enlazar la aplicación con las bibliotecas de Condor es la siguiente:

Primeramente se necesita extraer del paquete de Condor, el directorio *lib* y el script *condor_compile* que se encuentra en el directorio *bin.*

En el script *condor_compile* se modifica la siguiente línea:

CONDOR_LIB=
por
CONDOR_LIB= *Ruta en donde se encuentra el directorio lib de Condor.*

Después, para ligar la aplicación se realiza lo siguiente:

```
condor_compile -condor_standalone cc -o prueba_ckpt prueba.c
```

donde *prueba_ckpt* es la aplicación con el checkpointing implementado.

Una vez que se envió la aplicación al grid, ésta se ejecutó en kumiai, en la cola llamada "*kumiai.q*", al suspenderse la cola de ejecución del nodo kumiai debido a que se sobrepaso la tolerancia de carga definida para este nodo, la aplicación creó una imagen de su estado (checkpoint image) y migró a otro nodo que se encontraba con menos carga de trabajo, la aplicación se ejecutó desde la imagen creada y de esta forma su tiempo de duración fue el de 6 minutos aproximadamente, que es el tiempo que la aplicación dura en ejecutarse sin suspensión alguna.

Para Cicese2000, no es necesario ligar la aplicación con las bibliotecas de Condor, ya que el sistema operativo Irix de SGI permite el checkpointint a nivel kernel utilizando el comando "cpr" propio de Irix.

Una vez configurado lo necesario para implementar el checkpointing a una tarea secuencial, ésta se envía a la cola de ejecución "cicese2000.q". Al suspenderse ésta cola, la tarea no se migró a otro nodo ya que la aplicación necesita una arquitectura Irix y solo cicese2000 cuenta con esta arquitectura, sino que creó una imagen de su estado en ese momento y, al habilitarse de nuevo la cola de cicese2000, la aplicación empezó a ejecutarse desde la imagen creada. En caso de que se contara en el Grid con otro nodo de ejecución con arquitectura Irix, la tarea hubiera migrado a ese nodo para continuar con su ejecución.

El método para ligar una aplicación con checkpointing en cicese2000 se llama *cicese2000_ckpt*.

4) Conclusiones.

Con ésta prueba se comprobó que se pueden configurar algunas aplicaciones para que estas se vuelvan a ejecutar de manera automática en caso de suspenderse la cola de ejecución en donde se encuentren corriendo. También, en caso de ser necesario, se pueden configurar algunas aplicaciones (con ciertas restricciones) para que cuenten con la habilidad de checkpointing y hagan uso de ésta dentro del grid.

Una de las desventajas del checkpointing es que no todos los sistemas operativos soportan esta habilidad a nivel kernel teniendo que ligar aplicaciones con bibliotecas independientes. Además, con el uso de éstas bibliotecas, no todas las aplicaciones pueden ser ligadas, por tal motivo, resulta muy poco atractivo para los usuarios.

VI. CONCLUSIONES

Hoy en día vemos como va evolucionando el interés por los grids computacionales en todo el mundo, tanto así, que en la página de Internet *www.gridpartners.com*, crearon un índice llamado "El espejo del grid" el cual mide el grid computacional según las consultas realizadas en el buscador de páginas Google. Lo que descubrieron fue que el número de peticiones se ha más que triplicado desde Junio del 2002, y que la mayoría de ellas vienen de los E.U.A. y del Reino Unido.

Pero nuestro país no se esta quedando atrás, aunque si un poco mas lento, la tecnología grid ya esta siendo implementada por diversas instituciones a nivel nacional, tal es el caso de CICESE, la UNAM, la UABC, la U. de. G. entre otras. De hecho, el pasado 23 de septiembre del 2003, se celebró un día virtual CUDI de grids computacionales en la que se trataron temas muy importantes, de los cuales podemos destacar la necesidad de crear un comité técnico para el desarrollo de los grids computacionales en México y la necesidad de formar recursos humanos relacionados con el tema.

Con la presente investigación, se dió a conocer una de las tecnologías que están revolucionando al mundo entero, tanto ha sido la popularidad de esta tecnología que algunos directivos de empresas importantes como IBM, afirman que la tecnología grid va a lograr tener tanto auge como lo hizo Internet en su tiempo.

También en esta investigación, se dió a conocer la implementación de un grid departamental en el CICESE, y como ésta vino a solucionar algunas problemáticas que se tenían antes con el equipo de supercómputo, como lo era la inapropiada distribución de carga entre los equipos. Con este grid, se ofrece un mejor servicio a los usuarios de supercómputo de CICESE.

Pero lo más importante, es que la presente investigación abrió el camino para futuros proyectos relacionados con el tema.

Uno de los proyectos, es la implementación de grids entre instituciones que estén interesadas con el tema y que cuenten con el canal de Internet 2. Un ejemplo de esto, podría ser la creación de un grid interinstitucional entre el CICESE y la UABC.

GLOSARIO

Benchmarks.
Son aplicaciones o programas los cuales sirven para obtener el rendimiento de los equipos (velocidad de procesamiento).

Calendarizacion.
Referente a la tecnología Grid, es un proceso el cual, elige que host es el más adecuado para ejecutar alguna tarea dependiendo de los requerimientos de la tarea.

Constelación.
Se les llama constelaciones a un conjunto de nodos los cuales cuentan con muchos procesadores por nodo, más de 16 procesadores por nodo. Son maquinas SMP.

CPU.
Abreviatura de Central Processing Unit. Unidad central de procesamiento. Es el cerebro de la computadora.

Ducto.
Un ducto o bus en ingles, es un conjunto de cables por lo cuales pasan datos de un lugar de la computadora a otro.

Gigaflop.
Un billón de operaciones de punto faltante por segundo.

Grid.
Malla por su significado en español

Hardware.
Es todo aquello que podemos tocar, el monitor, el teclado, la computadora en si, (lo que alberga las tarjetas, el disco duro, la unidad de disquete, el procesador, memoria, etc.), la impresora, el ratón (mouse), los cables, conexiones, etc.

Idle.

Estado en el cual el procesador se encuentra activo, pero sin ejecutar alguna operación, por lo tanto se dice que el procesador esta siendo desperdiciado o que esta inactivo.

Memoria caché.
Es un tipo de memoria de alta velocidad que es usada para almacenar datos que se utilizan con más frecuencia por algunos programas y de esta forma ganan más velocidad de procesamiento.

Middleware.

Software que conecta dos o más aplicaciones independientes.

NFS.

Abreviatura de Network File System. Sistema de archivos de red.

Pipelining.

Consiste en ejecutar al mismo tiempo en diversas etapas las instrucciones del programa; mientras en una etapa se hace la ejecución de una instrucción, simultáneamente en otra etapa se está realizando una lectura de la siguiente instrucción, es un esquema muy similar al de una línea de ensamble de autos.

Procesador vectorial
Es un procesador capaz de ejecutar instrucciones en el cual los operandos pueden ser arreglos más bien que escalares. [Telecom Glosary 2000 \
www.its.bldrdoc.gov/projects/devglossary/]

Racimos.
Se refiere a clusters.

Software.
Todo el hardware que hay no puede funcionar si no hay un programa o programas que hacen que funcione de manera adecuada. Estos programas hacen que los usuarios puedan interactuar con la computadora.

Supercomputadora vectorial
Contiene un conjunto de unidades funcionales, utilizadas para procesar elementos de arreglos o vectores eficientemente.

Teraflop.
Un trillón de operaciones de punto flotante por segundo.

REFERENCIAS

[Appg 03]
Articulo. "Aplicaciones del grid". http://www.ifae.es/pic/grid_intro_cas.htm

[Arqui 03]
UNAM. Departamento de supercómputo. "Arquitectura de supercomputadoras". http://www.labvis.unam.mx/new/docencia/planbec/cursos/arquitecturas/

[Aspen 03]
Aspen Systems, Inc."The Era of Supercomputing"
http://www.aspsys.com/clusters/beowulf/history/. 2003

[Ava 03]
Avalon. "The Avalon Cluster". http://cnls.lanl.gov/avalon/.

[Avaki 03]
Avaki. www.avaki.com

[Beowulf 03]
Proyecto Beowulf. Articulo."Historia del proyecto Beowulf". www.beowulf.org.

[Casdel 99]
Reporte técnico. "Supercomputadora cicese2000". Salvador Castañeda, Julian Delgado. 1999. Clave: CTCOT9901. 1999. CICESE.

[Casdel 02]
Reporte técnico. "Guía de usuario de la supercomputadora calafia". Salvador Castañeda, Julian Delgado. 2002. Clave: 10396. CICESE.

[Cicese 03]
CICESE. "Web CICESE". www.cicese.mx.

[Clumex 03]
UNAM. Departamento de supercomputo. "Historia de los clusters". http://clusters.unam.mx/Historia/index.php

[Condor 03]
University of Winsconsin. "Condor project". http://www.cs.wisc.edu/condor/

[Cray 03]
Cray Inc. Articulo "Cray Inc. History". www.cray.com/company/history.html. 2003.

[Cudi 03]
Cudi. Reunión de primavera Cudi 2003 en Ensenada, B.C.. http://cudi2003.cicese.mx/.

[Data 03]
DataSynapse. www.datasynapse.com

[Earl 03]
Earl Joseph, Ph.D. "The Evolution of the Supercomputer". Linux Network. http://www.lnxi.com/news/history/index.php. 2003.

[Econ 03]
The economist. Articulo. www.economist.com

[Entro 03]
Entropia. www.entropia.com

[Foster 99]
Libro. "The Grid : blueprint for a new computing infrastructure". Ian Foster, Carl Kesselman. San Francisco : Morgan Kaufmann, 1999

[Gcomp 03]
Articulo. "El grid computacional". http://www.vnunet.es

[Globus 03]
Globus. The Glubus Project. www.globus.org.

[Grcomp 03]
Articulo. ¿Qué es el grid?. www.gridcomputing.com.

[Ibgl 03]
IBM. Articulo "IBM y Globus anuncian servicios Grid abiertos para informática de usuario" http://www-5.ibm.com/es/press/notas/2002/febrero/globus.html.

[IBM 02]
Articulo. "IBM inaugura el primer centro de innovación de tecnología grid para clientes" http://www-5.ibm.com/es/press/notas/2002/abril/grid.html

[Inner 02]
GridSytems. Documento "Innergrid". www.gridsystems.com.

[Irbk 02]
IBM RedBook. "Introduction to grid computing with Globus" www.redbooks.ibm.com

[Leru 93]
Lerman, G., Rudolph, L., "Parallel Evolution of Parallel Processors", Plenum Press, New York (1993).

[Mmc 03]
MM5. MM5 Community Model. http://www.mmm.ucar.edu/mm5/mm5-home.html
[Morr 92]
Morris, Ch., "Academic Press Dictionary of Science and Technology", Academic Press (August 1992).

[Nasft 03]
NASA. Software de la NASA. http://www.nas.nasa.gov/Software/NPB/

[Ogsa 03]
El proyecto Globus. Articulo. "Arquitectura de servicios abiertos para el grid" www.globus.org/ogsa

[Oracle 03]
Oracle white papers. "Oracle grid computing technologies". http://otn.oracle.com/products/oracle9i/grid_computing/index.html

[Origin 97]
Origin Servers Technical Report. Silicon Graphics. Abril 1997.

[Platf 03]
Platform Computing. www.platform.com

[Seti 03]
SETI@HOME. El proyecto SETI. http://setiathome.ssl.berkeley.edu/.

[Steen 02]
Aad J. van der Steen & Jack J. Dongarra , "TOP 500 Overview of Recent Supercomputers"

[Sun 03]
Sun Microsystems. www.sun.com/software/grid

[Tecnova 99]
Tecnova. "Revista electrónica". http://www.cea.es/tecnova/default.asp.

[Top 03]
Top 500. Las 500 supercomputadoras más rápidas del mundo. www.top500.org.

[Udev 03]
United Devices. www.ud.com