

**Universidad Michoacana de San Nicolás de Hidalgo**



**Facultad de Ingeniería Eléctrica**



**“Administración de recursos cómputo utilizando  
Xgrid”**

**Tesis**

Que para obtener el Título de

**INGENIERO EN COMPUTACIÓN**

Presenta

**Luis Angel Feregrino Lemus**

Asesor

**M.C José Ortiz Bejar**

**Octubre del 2012**

# Agradecimientos

No creo poder expresar con palabras el agradecimiento que les tengo a todas las personas que han rodeado mi vida y de una u otra forma han hecho lo que soy. Y aunque me sienta incapaz de expresar lo que siento por cada persona, tratare de hacerlo.

A la primera persona que tengo agradecer, es la que me dio la vida, *mi mamá, Juana Lemus Villareal*, que siempre tuvo la paciencia, el amor y la dedicación para guiarme por el camino correcto. Gracias mamá por ese licuado en las mañanas para que no me fuera con el estómago vacío a la escuela, por todas esas noches que te desvelaste a mi lado cuando estaba enfermo, por esa clásica pregunta de: ¿Cómo te fue?, por la comida caliente cuando llegaba hambriento de la escuela, por todas las risas; gracias a tus ocurrencias, hasta por esos regaños para que no tomara el camino equivocado y sobre todo porque no importo cual fuese tu estado de ánimo o de salud, siempre has estado a mi lado. Por eso y muchas cosas más, gracias mamá.

A *mi papá, Luis Angel Feregrino Zamudio*, por su apoyo incondicional, por confiar en mí, por alentarme a seguir adelante, por darme los valores esenciales para ser un buen hombre, por ir todas esas mañanas a trabajar aunque estuvieras hartos; solo para darme lo mejor, por cumplir todos mis caprichos, por esas llamadas diarias del trabajo para solo saber cómo me encontraba, por mencionar algunas cosas de las cuales estoy agradecido. Gracias por todo papá.

A *mi hermana, Atziri Esmeralda Feregrino Lemus*, esa personita que ha sido mi compañera toda mi vida, con la que he pasado muy buenos momentos y cuando no son tan buenos me ha aguantado y apoyado, ella la que me ha sacado una sonrisa cuando más lo necesito, gracias hermana y **“Carnalitos forever, esa”**.

Agradezco a *mi asesor, M.C José Ortiz Bejar*, por la paciencia que me tuvo, por brindarme su tiempo, por haber estado detrás de este proyecto apoyándome a cumplir este gran sueño y sobre todo por haber confiado en mí, gracias profe.

A *mis sinodales*, que me brindaron un poco de su tiempo para revisar mi tesis y hacerme observaciones para mejorarla.

Tengo que agradecer a *mis profesores*, que me han ayudado con mi formación y me enseñaron que el trabajo duro siempre trae recompensas.

Agradezco a *todos mis amigos y sobre todo a los de la Facultad: Carreto (Carretiux), Manuel (Mane), Guillermo (Memo), Cesar (el patron), Josue (el panda)*, que nunca me dejaron rendirme fuese cual fuese la situación, los que me hicieron reír y olvidar los problemas, los que me explicaron algo cuando no lo entendía o simplemente estando a mi lado. Gracias amigos.

Y sobre todo, agradezco a *Dios* por haberme rodeado de tanta gente buena y dejar que cumpliera uno de mis grandes anhelos.

# Resumen

Actualmente los trabajos de investigación requieren realizar grandes cantidades de operaciones de cómputo en tiempos razonables debido a esto los Sistemas de administración de trabajo (JMS´s) se han convertido en un factor importante en el mundo de la investigación. Utilizando un JMS es posible llevar a cabo de forma eficiente tareas computacionales, procesos que van desde graficación en 3D hasta ejecutar procesos enfocados al desarrollo de medicamentos. En el presente trabajo se analizan distintos JMS´s con la finalidad de identificar uno que se adecúe a las necesidades del Posgrado de Ingeniería Eléctrica. Los tres JMS´s evaluados son Platform LSF, Condor y Xgrid. Después de una comparación de los tres JMS´s, se eligió Xgrid debido que se cuenta con la infraestructura para configurarlo fácilmente, adicionalmente no es necesario realizar el proceso de instalación, ya que, en MAC OS X el sistema Xgrid ya se encuentra instalado. Una vez configurado Xgrid, se instalaron diferentes aplicaciones para simplificar el envío de trabajos, así como su supervisión. Finalmente se realizaron distintas pruebas en diferentes lenguajes de programación para comprobar el funcionamiento y rendimiento del Grid.

# Índice general

<b>1. Introducción</b>	<b>10</b>
1.1. Antecedentes . . . . .	11
1.1.1. Computación en Grid . . . . .	11
1.2. Objetivo . . . . .	13
1.3. Justificación . . . . .	13
1.4. Configuración de la red . . . . .	14
1.5. Sistema de administración de trabajo . . . . .	14
1.6. Metodología . . . . .	15
<b>2. Fundamentos</b>	<b>16</b>
2.1. Arquitectura de un JMS . . . . .	16
2.2. Revisión y comparación de JMS's . . . . .	18
2.2.1. Condor . . . . .	18
2.2.2. Xgrid . . . . .	24
2.2.3. Platform LSF . . . . .	26
2.3. Criterio de evaluación . . . . .	28
2.3.1. Condor . . . . .	28
2.3.2. Xgrid . . . . .	29
2.3.3. Platform LSF . . . . .	29
<b>3. Configuración de Xgrid</b>	<b>32</b>
3.1. Configuración del agente Xgrid . . . . .	34
3.2. Configuración de controlador Xgrid . . . . .	36

<i>ÍNDICE GENERAL</i>	5
3.3. Xgrid Admin . . . . .	39
3.4. GridStuffer . . . . .	41
3.5. Herramienta de clasificación de Agentes . . . . .	43
3.5.1. Problemática con ART . . . . .	46
3.6. Comentarios Finales . . . . .	47
<b>4. Pruebas y Resultados</b>	<b>48</b>
4.1. Calendario . . . . .	49
4.2. División y cuadrado . . . . .	51
4.3. Método Monte-Carlo . . . . .	52
4.4. Comentarios Finales . . . . .	54
<b>5. Conclusiones y trabajo futuro</b>	<b>55</b>
5.1. Conclusiones . . . . .	55
5.2. Trabajo Futuro . . . . .	56
<b>A. Apéndice A</b>	<b>58</b>
A.1. Configuración del agente Xgrid . . . . .	58
A.2. Configuración de controlador Xgrid . . . . .	60
A.3. Características de las computadoras participantes en el Grid . . . . .	63
<b>B. Apéndice B</b>	<b>64</b>
B.1. Código del programa “División y cuadrado” . . . . .	64
B.2. Código del programa “Método Monte-Carlo” . . . . .	65
B.3. Script ART “imac3” . . . . .	65
B.4. Script ART “RAM” . . . . .	65

# Índice de figuras

1.1. Configuración de la red del Posgrado de Ingeniería Eléctrica . . . . .	13
2.1. Principales bloques de un sistema de administración de trabajo. . . . .	17
3.1. Roles e interacción de computadoras con Xgrid . . . . .	33
3.2. Configuración de agente . . . . .	34
3.3. Opciones de configuración . . . . .	35
3.4. Server Admin . . . . .	36
3.5. Configuración del controlador . . . . .	37
3.6. Configuración del controlador como un agente . . . . .	38
3.7. Información General de Xgrid . . . . .	39
3.8. Trabajos . . . . .	40
3.9. Agentes conectados a Xgrid . . . . .	41
3.10. Controladores existentes en Xgrid visto con XgridStuffer . . . . .	42
3.11. MetaJobs . . . . .	42
3.12. Nuevo trabajo . . . . .	43
3.13. Inspector de MetaJobs . . . . .	43
3.14. Proceso del controlador ante un ART . . . . .	44
4.1. Trabajos nuevos . . . . .	49
4.2. Monitoreo del estado de los agentes . . . . .	50
4.3. Tareas realizadas . . . . .	50
4.4. Salida de Calendario . . . . .	51

4.5. Trabajos del Programa operación aritmética . . . . .	52
4.6. Creación del trabajo de Monte-Carlo . . . . .	53
4.7. Envío del trabajo Monte-Carlo al controlador con GridStuffer . . . . .	53
A.1. Configuración de agente . . . . .	59
A.2. Opciones de configuración . . . . .	59
A.3. Server Admin . . . . .	60
A.4. Configuración de controlador . . . . .	61
A.5. Configuración del controlador como agente . . . . .	62
A.6. Últimos pasos para la configuración del controlador . . . . .	63



# Índice de Tablas

2.1. Plataformas soportadas por Condor 7.7.5 . . . . .	23
2.2. Comparación de características de los administradores de cargas de trabajo . . . . .	30
4.1. Aproximación a Pi . . . . .	53
A.1. Características de computadoras participantes . . . . .	63

# Índice de abreviaciones

JMS.- Job Management System

HTC.- High Throughput Computing

HPC.- High Performance Computing

NFS.- Network File System

IT.- Information Technology

SLA.- Service Level Agreement

ART.- Agent Ranking Tool

# Capítulo 1

## Introducción

El poder computacional ha crecido exponencialmente en estos últimos años, particularmente en el campo de microprocesadores, almacenamiento y redes. En el siglo 21 fue el auge para este crecimiento, gracias al aumento de computadoras en los hogares y el mundo de los negocios. Con una nueva sociedad, la exigencia de la velocidad de procesamiento se ha incrementando a medida que es posible realizar más trabajos con las computadoras actuales, de lo que hubiéramos hecho en las décadas de los 80's y 90's. En nuestra época las empresas utilizan las computadoras para realizar transacciones y procesar la información relacionada con sus procesos. Mientras los usuarios domésticos han comenzado a utilizar las computadoras para mucho más que escribir y jugar. Las Universidades también han aprovechado este crecimiento de potencia computacional y lo utilizan para resolver problemas que necesitan realizar grandes cantidades de cálculos en un tiempo corto, todo esto para beneficio de sus investigaciones.

La computación en Grid es una tecnología que permite usar de forma coordinada los recursos de las computadoras conectadas a una red, utilizando el poder conseguido para resolver problemas que necesiten hacer una gran cantidad de cálculos. Actualmente el cómputo en Grid es común en el ambiente empresarial y académico, y es necesario en las áreas donde se realicen grandes cantidades de procesamiento. Algunos de estas áreas son: investigación biomédica, matemática, estadística y análisis de imágenes.

Una de las principales tareas de la computación en Grid es la administración de trabajo, también conocida como administración de carga de trabajo, distribución de carga o manejo de trabajo. Los

sistemas de software capaces de realizar esta tarea se les denominan Sistemas de administración de trabajo (JMS). Dichos sistemas pretenden optimizar los recursos de cómputo de una red, mediante la asignación de trabajo a computadoras inactivas conectadas al Grid. Muchos proyectos de investigación necesitan una gran cantidad de ciclos de cómputo, y no es raro encontrar un problema que requiera semanas o hasta meses para resolverlo. Debido a lo anterior se necesita un ambiente de cómputo que proporcione gran poder de procesamiento por periodos de tiempo prolongados, a esto se le denomina “computación de alto rendimiento” (HTC por sus siglas en ingles). La clave de HTC es aprovechar todos los recursos disponibles eficazmente. Otro ambiente que se utiliza es la “computación de alto rendimiento” HPC (por sus siglas en ingles) que entrega una gran potencia computacional en un tiempo corto.

Algunas veces dichos ambientes utilizan la computación paralela para sacar un mayor provecho a los múltiples procesadores de cada una de las computadoras conectadas en el Grid.

## 1.1. Antecedentes

### 1.1.1. Computación en Grid

La computación en Grid comenzó en 1970 cuando el costo de procesamiento era alto. En aquellos días, las computadoras eran muy grandes y costaban cientos de miles de dólares. Como resultado, se crearon metodologías para maximizar el uso de ciclos y así nació el término Grid. Algunos de los primeros experimentos realizados fueron los llamados Creeper [Tho04] y Reaper [Wik12].

Estos programas se ejecutaron en el ARPA-net, que fue el precursor de la World Wide Web que utilizamos hoy. Estos proyectos fueron llamados “gusanos”, cuando creeper llegaba a alguna computadora, se auto ejecutaba y comenzaba a mostrar el siguiente mensaje: “I’m the creeper, catch if you can” que en español sería “Soy la enredadera, atrápame si puedes”. El virus no se replicaba en verdad, sino que iba de una computadora a otra, eliminándose de la computadora anterior. Reaper no era más que otro virus que actuaban como un antivirus de tal manera que iba extendiéndose por las computadoras de la red y cuando se encontraba con creeper, lo eliminaba.

En 1973, el centro de investigación de Xerox en Palo Alto hizo algunos avances importantes que influyeron en gran medida en el desarrollo de computación distribuida. Alguno de estos desarrollos incluyen el desarrollo de la estación de trabajo como una computadora personal, la primera insta-

lación en red que permitió a computadoras autónomas comunicarse entre sí y por último el PARC (Centro de investigación de Palo Alto) desarrollo el primer servidor de archivos distribuidos para que el usuario pudiera compartir archivos comunes. En el PARC otro “gusano” fue creado por los científicos Jonh F. Schoch y Jon A. Hupp. El propósito de este gusano era pasar de máquina en máquina usando recursos que no estaba utilizando para fines productivos.

Con la invención del Internet moderno en 1991, la computación distribuida creció más que la computación en redes de área local. Los Grids en la actualidad operan a través de múltiples plataformas y en zonas geográficamente dispersas. En un principio la participación en la computación distribuida era escasa, mientras ahora existen millones de computadoras que contribuyen en ella. El primer proyecto Grid fue llamado distributed.net en sus comienzos tenia miles de contribuyentes y se utilizaba para descifrar código encriptado. El segundo proyecto fue SETI@Home, un proyecto que buscaba inteligencia extraterrestre mediante el análisis de fluctuaciones de señales de radio.

El SETI@Home es uno de los proyectos de computación distribuida más exitosos, hasta la fecha cuenta con más de 3 millones de personas corriendo este software. Estas 3 millones de personas tienen un promedio de 500 teraflops (500 billones de operaciones de punto flotante por segundo) y han completado alrededor de 500,000 años de tiempo de procesamiento.

Las ventajas de esto son principalmente económicas, ya que el costo de una supercomputadora con la misma salida costaría millones de dólares. Desde la creación de este proyecto han surgido algunos mas, como es Find-A que se ha creado con la esperanza de proporcionar suficiente poder de procesamiento para ayudar a la investigación de nuevos fármacos que se puedan utilizar para curar el cáncer, VIH, y contrarrestar el bioterrorismo.

Existen proyectos de computación distribuida alrededor del mundo, los cuales permiten a los usuarios integrar sus equipos y recursos a un JMS, pero también existen proyectos menores que permiten a los individuos o pequeñas organizaciones crear con facilidad sus propios sistemas distribuidos, esto tiene grandes ventajas tales como: administrar su sistema distribuido y solo depender de las computadoras que están en su red.

## 1.2. Objetivo

Administrar los recursos de la red del Posgrado de la Facultad de Ingeniería Eléctrica, para optimizar el uso de la capacidad de cómputo de los equipos disponible, esto mediante un sistema de manejo de carga (Job Management System). Se pretende crear un medio ambiente de cómputo de alto rendimiento (HTC), para utilizar de manera efectiva los recursos disponibles en la red de trabajo. Adicionalmente se pretende instalar librerías de cómputo en paralelo, con la finalidad de proporcionar las herramientas necesarias para implementar programas que exploten los recursos disponibles en la red.

## 1.3. Justificación

Como se menciona anteriormente es común que los trabajos de investigación requieran una gran cantidad de recursos computacionales, para esto se necesita el uso intensivo del equipo donde se ejecutan los procesos relacionados, lo cual afecta la experiencia del usuario que ha iniciado sesión. Por este motivo es necesario crear un medio ambiente que aproveche al máximo el poder computacional disponible y que al mismo tiempo no afecte a los usuarios del equipo de cómputo. Se pretende crear un Grid utilizando una red que consta de: 6 equipos con el sistema operativo MAC OSX, 2 servidores MAC (*dep1* y *dep2*), el servicio de administración (JMS) estará hospedado en una MAC mini (*dep3*). La red está conectada como se muestra en la Figura 1.1.

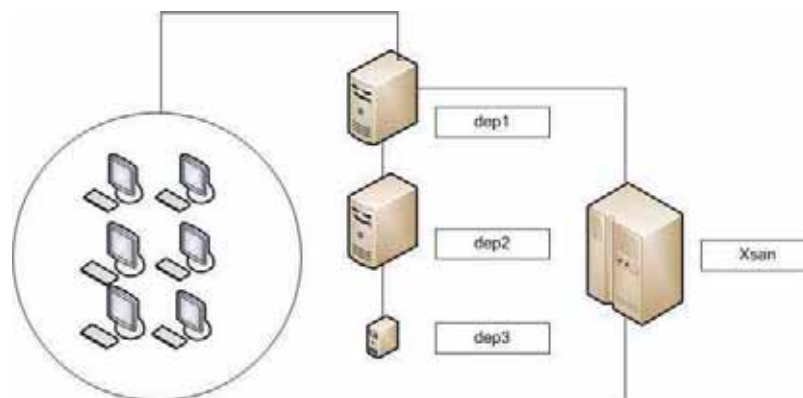


Figura 1.1: Configuración de la red del Posgrado de Ingeniería Eléctrica

## 1.4. Configuración de la red

En el servidor *dep1* se encuentra instalado el sistema archivos de red (NFS por sus siglas en inglés) y Opendirectory. NFS es un servidor para sistemas de archivos distribuidos en un entorno de red de computadoras de área local. Hace posible que distintos sistemas conectados en la red accedan a un sistema de archivos compartidos remotos como si se tratara de un sistema local [SEL]. Con NFS los datos se mantienen accesibles para todo tipo de usuarios en un nodo central. Así mismo los datos que consumen grandes cantidades de disco como los datos de administración se pueden mantener en un solo nodo. La velocidad de un acceso remoto a los archivos es tan eficiente como de forma local [JCL11].

Opendirectory implementa un servicio de directorio jerárquico y distribuido para acceder a la información referente a usuarios, contraseñas y otras entidades en un entorno de red, ofreciendo una amplia capacidad de filtrado sobre la información que está siendo solicitada. La implementación de Opendirectory tiene muchas ventajas como: ayuda a tener un sistema jerárquico de almacenamiento, permite replicar un servidor de forma muy sencilla, asegura que todas las entradas sean únicas gracias a su modelo de nombres globales, por mencionar algunas [Pub12]. La finalidad de tener instalado Opendirectory en el servidor *dep1* es para el manejo de cuentas de los usuarios de la red de Posgrado.

## 1.5. Sistema de administración de trabajo

Con la instalación de un sistema de administración de trabajo (JMS) en *dep3* se pretende crear un medio ambiente de alto rendimiento (HTC) para utilizar de manera efectiva el poder de cómputo de todas las computadoras conectadas a la red.

Se consideraron tres diferentes JMS, los cuales cumplen las necesidades de Posgrado ellos son: Condor, Xgrid y Platform LSF. Condor es un sistema de administración de trabajo especializado en trabajos de cómputo intensivo, proporciona una cola de trabajo, política de planificación, esquema de prioridades, seguimiento de recursos así como su administración. Condor es producto de Proyecto de Investigación de Condor (Condor Research Project) en la Universidad de Wisconsin-Madison y fue instalado por primera vez como un sistema de producción en el departamento de Ciencias de la Computación hace 15 años [UM12].

Xgrid es un programa para la computación distribuida desarrollado por la subdivisión de Cómputo Avanzado de Apple Inc. que permite a las computadoras conectadas en red contribuir a una sola tarea. Proporciona a los administradores de una red un medio para la creación de un cluster, lo que permite explotar el poder computacional.

Platform LSF es un software comercial vendido por Platform Computing para la administración de trabajo, el cual es el más poderoso por demanda, distribuido y de entornos de alto rendimiento de computación. Ofrece un conjunto completo de capacidades para la administración de carga de trabajo, todos diseñados para trabajar en conjunto para hacer frente a las necesidades de computación de alto rendimiento.

En el Capítulo 2 se describirán detalladamente, compararan y se elegirá alguno de los 3 JMS's mencionados anteriormente.

## 1.6. Metodología

Se consultarán varias bibliografías para obtener conocimiento básico sobre los conceptos de los entornos de MAC OSX [Appa], NFS [SEL] y Open Directory [Appab] debido a que estos servicios son los que se encuentran disponibles en la red de Posgrado. Se revisarán varios manejadores de carga para comparar sus características, requerimiento, beneficios y debilidades, gracias a esta comparación, se elegirá uno que se ajuste mejor a las necesidades particulares del Posgrado. Se realizará el proceso de instalación del manejador de carga elegido y así mismo su configuración en *dep3* (Mac mini) y por último se harán diferentes pruebas.

En el Capítulo de Introducción se dará una breve explicación a los JMS's. Así como antecedentes, objetivo, justificación y metodología empleada en esta tesis. En el siguiente Capítulo que corresponde a Fundamentos se revisarán distintos sistemas de administración de trabajo (JMS's) para la comparación de sus características, requerimientos, ventajas y desventajas. Y así elegir alguno que cumpla con las necesidades del Posgrado. A lo largo del Capítulo de Desarrollo se realiza la instalación del JMS elegido así como su configuración. Dentro del Capítulo de Pruebas y resultados se harán distintas pruebas en diferentes circunstancias para observar el comportamiento de nuestro JMS. Por último en el Capítulo de Conclusiones y trabajos futuros se hablará de las conclusiones a las cuales se llegó a través del uso de Sistemas de administración de trabajo, así como trabajos futuros que se pueden derivar del trabajo realizado.



## Capítulo 2

# Fundamentos

En el Capítulo anterior se mencionó que una de las mayores tareas de la computación en Grid es la administración de trabajo. Los sistemas que se encargan de realizar esta tarea se les denomina Sistema de administración de trabajo (JMS). Los JMS's pueden aprovechar los recursos de computadoras inactivas conectadas en una red, la mayoría de ellos puede trabajar en múltiples entornos, incluyendo cluster's heterogéneos de estaciones de trabajo, súper computadoras y sistemas paralelos.

### 2.1. Arquitectura de un JMS

El objetivo de un JMS es permitir a los usuarios ejecutar trabajos en un grupo de estaciones de trabajo no dedicadas con un impacto mínimo a los propietarios de dichas estaciones mediante el uso de recursos computacionales prestados por los propietarios.

El sistema debe ser capaz de realizar al menos las siguientes tareas:

1. Seguimiento de los recursos disponibles.
2. Aceptar trabajos presentados por los usuarios, junto a los recursos requeridos para cada uno.
3. Realizar una planificación centralizada de tareas donde coincidan todos los recursos disponibles con los trabajos presentados.
4. Asignar recursos y administrar la ejecución del trabajo.

5. Supervisar todos los trabajos y recolectar información.

Para realizar estas tareas básicas un JMS debe de incluir al menos las unidades de funcionamiento que se muestran en la Figura 2.1

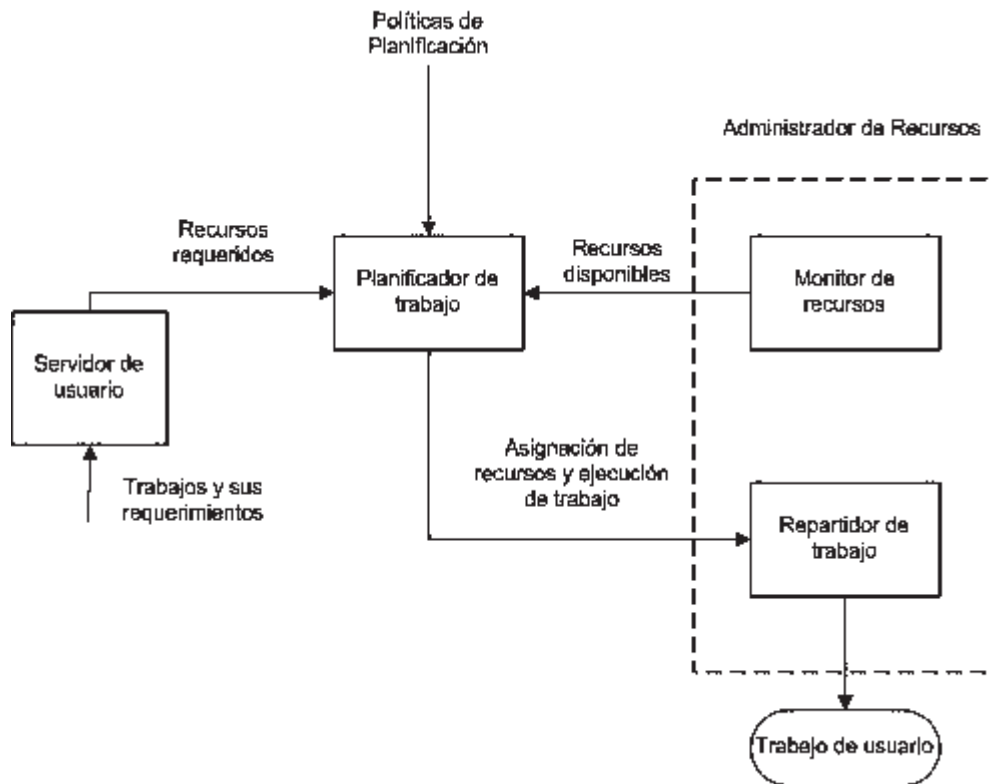


Figura 2.1: Principales bloques de un sistema de administración de trabajo.

1. **Servidor de usuario.-** Permite al usuario enviar trabajos y sus necesidades a un JMS, además permite obtener información acerca del estado del trabajo.
2. **Planificador de trabajo.-** Lleva a cabo la planificación de tareas, adicionalmente la asigna a una cola en base a las requerimientos de disponibilidad de recursos y políticas presentadas.
3. **Administrador de recursos.-** Se usa para controlar los recursos y el envío a los puestos de trabajo en un sistema de ejecución determinado.
  - a) **Monitor de recursos.-** Se encarga de supervisar los recursos existentes en el sistema.

- b) **Repartidor de trabajos.**- Distribuye los trabajos a las computadoras integrantes del JMS.

La elección de un JMS es difícil, debido a la existencia de un gran número disponible ya sean como software comercial o bien de código abierto.

## 2.2. Revisión y comparación de JMS's

Como ya se mencionó existen varios JMS's cada uno con sus propias características. En las siguientes secciones se realizará una comparación entre algunos de los más importantes encontrados en la literatura. Los JMS's analizados son Condor, Xgrid y LSF.

### 2.2.1. Condor

Basándose en los resultados del proyecto Remote-Unix (RU) que fue dirigido por los profesores D. Dewitt, R. Finkel y M. Solomon y como continuación de sus trabajos en el área de Administración de Recursos Distribuidos (DRM) por un grupo dirigido por el profesor M. Livni, el proyecto Condor comenzó en 1988. Se ha centrado en los clientes con grandes requerimientos de cómputo y entornos heterogéneos con recursos distribuidos. Con el esfuerzo de RU, el proyecto Condor ha heredado una gran colección de mecanismos y una base muy sólida.

La primera versión del sistema de administración de trabajo de Condor (inicialmente llamado el sistema RU) se implementó en 1986. El núcleo de políticas de la administración del proyecto Condor fue proporcionado por la asignación distribuida y las técnicas preventivas desarrolladas en la tesis del profesor M. Mutka del grupo de DRM [Mut88]. Aunque originalmente se centraba en el problema de balance de carga en un sistema distribuido, a mediados de la década de los 80's el grupo DRM desvió su atención a los ambientes de cómputo distribuido donde los dueños tenían control sobre los recursos que poseían [EGGA03].

Condor es producto de la investigación del proyecto Condor en la Universidad de Wisconsin-Madison y fue instalado por primera vez en el sistema de producción del departamento de ciencias computacionales de esa misma universidad. En la actualidad se ha convertido en un sistema de administración de trabajo especializado para puestos de trabajo de cálculo intensivo. Ofrece un mecanismo de colas de trabajo, política de planificación, esquemas de prioridades, seguimiento de

recursos y administración de recursos.

Los usuarios desde sus puestos de trabajo envían sus trabajos a Condor, ya sea en serie o paralelo, los coloca en una cola, elige cuando y donde ejecutar los trabajos en base a una política y cuidadosamente vigila su progreso e informa al usuario cuando terminan.

Aunque proporciona una funcionalidad similar a la de un sistema tradicional de colas, la novedosa arquitectura de Condor permite que tenga éxito en zonas donde los sistemas tradicionales fallan. Se puede utilizar para administrar un conjunto de nodos de computación dedicada. Además, los mecanismos únicos de Condor permiten aprovechar efectivamente el poder de CPU desperdiciado cuando las estaciones de trabajo están en modo de inactividad. Por ejemplo, Condor puede ser configurado para utilizar computadoras, donde el teclado y el ratón estén inactivos. En caso de detectar que una computadora ya no está disponible (por ejemplo, al pulsar una tecla), puede tener un punto de control del trabajo y migrarlo a una computadora que si lo este y empezar el trabajo en una nueva computadora precisamente donde se quedo dicho trabajo. En los casos donde Condor tenga un punto de control de un trabajo y lo migre, facilitara maximizar el número de máquinas que pueden ejecutar un trabajo.

Cuando se utiliza Condor no se requiere tener una cuenta en las máquinas donde se ejecutará el trabajo, esto debido a la tecnología de llamadas remotas al sistema, estas llamadas se transmiten a través de la red desde la máquina donde se envió el trabajo.

El uso de Condor puede ser un verdadero ahorro de tiempo cuando se requiere ejecutar un trabajo en diferentes momentos y con diferentes datos.

Cabe mencionar que Condor no requiere de un sistema de archivos compartidos entre las computadoras, en caso de no contar con un sistema de archivos compartidos, Condor puede transferir archivos de datos de trabajo en nombre del usuario, o ser capaz de redirigir de forma transparente las peticiones E/S de todos los puestos de trabajo. Como resultado, Condor puede ser usado para combinar todo el poder computacional de una organización en un solo recurso.

Existe un mecanismo en Condor llamado **ClassAd**, el cual ofrece un marco de trabajo muy flexible para hacer coincidir las solicitudes de recursos (trabajos) con ofertas de recursos (computadoras). Los trabajos pueden especificar tanto los requisitos y las preferencias. Del mismo modo, las máquinas pueden especificar los requisitos y preferencias sobre los puestos de trabajo que están dispuestos a ejecutar. Estos requisitos y preferencias se pueden describir en términos de expresiones de gran alcance, resultando en la adaptación de Condor a casi cualquier política deseada. **ClassAd**

funciona como la sección de clasificados de un periódico, ya que todas las máquinas en el grupo de Condor anuncia sus propiedades, tanto como estáticas como dinámicas, memoria RAM disponible, tipo de CPU, velocidad de CPU, tamaño de memoria virtual, ubicación física y promedio de carga actual. Un usuario especifica un anuncio de solicitud de recursos al enviar un trabajo. Dicha solicitud define un conjunto de propiedades de recursos para ejecutar el trabajo. Condor actúa como intermediario para hacer coincidir los anuncios de solicitud de recursos con los recursos ofertados, asegurándose que todos los requisitos en ambos anuncios se cumplen (match-making). Durante el proceso de match-making, Condor también considera los valores de prioridad: la prioridad del usuario asignado a la solicitud de anuncio de los recursos, la prioridad del usuario que envía el anuncio, y la disponibilidad de las máquinas del grupo a aceptar ciertos tipos de anuncios.

### Características

Algunas de las características que hacen ser un gran JMS a Condor son las siguientes:

#### ■ Punto de control y migración

Un punto de control es un conjunto completo de información que comprende el estado de un programa, teniendo en cuenta un punto de control, un programa puede utilizar dicho puesto de control para reanudar un programa que ha sido migrado. Cuando los programas se pueden vincular con las bibliotecas de Condor, los usuarios pueden asegurar que con el tiempo su trabajo se complete, gracias a que si el equipo que lo está corriendo deja de estar disponible, el trabajo guarda un punto de control. Cuando el trabajo se migra a otra computadora se empieza desde el punto de control guardado anteriormente. Con la finalidad de preservar el tiempo acumulado de un trabajo, ya sea por un fallo de sistema o alguna otra cosa, se implemento la característica de punto de control a Condor.

#### ■ Llamadas remotas al sistema

A pesar de los trabajos en ejecución en máquinas remotas, el modo estándar de ejecución universal conserva el ambiente de ejecución local a través de llamadas remotas al sistema. Los usuarios no tienen por qué preocuparse de hacer archivos de datos disponibles para las estaciones de trabajo remotas o incluso de obtener una cuenta de acceso a las estaciones de trabajo. El programa se comporta, como si se estuviera ejecutando en la misma computadora al cual fue enviado originalmente, no importa en qué máquina se termine de ejecutar.

- **No se necesita hacer cambios al código fuente**

No existe programación especial para utilizar Condor. Condor es capaz de ejecutar programas no interactivos. Los puntos de control y migración de los trabajos son transparentes y automáticos, como es el uso de llamadas remotas al sistema.

- **Flocking**

Es una característica de Condor que permite a los trabajos presentados dentro de un primer grupo de máquinas ejecutarse en un segundo. El mecanismo es flexible, siguiendo peticiones del envío de trabajos, mientras el segundo grupo establece las políticas sobre las condiciones en las que los trabajos se ejecutan.

- **Los trabajos pueden ser ordenados**

El orden de ejecución de los trabajos requeridos por las dependencias entre los puestos de trabajo en un conjunto de fácil manejo. El conjunto de trabajos se especifica usando un grafo dirigido acíclico, donde cada trabajo es un nodo en el grafo. Los trabajos enviados a Condor siguen las dependencias dadas por el grafo.

- **El usuario local tiene preferencias**

El propietario de una máquina tiene prioridad total sobre el uso de la máquina. El propietario no necesita hacer una acción especial para recuperar el control, ya que Condor se encarga de esto automáticamente.

- **ClassAds**

Es un mecanismo de Condor, el cual ofrece un marco de trabajo muy flexible para hacer coincidir las solicitudes de recursos (trabajos) con ofertas de recursos (computadoras).

## Limitaciones

- **Limitación en los trabajos que pueden tener punto de control**

1. Trabajos multi-procesos no están permitidos. Esto incluye las llamadas al sistema como `fork()`, `exec()` y `system()`.
2. La comunicación entre procesos no está permitido. Esto incluye pipes, semáforos y memoria compartida.

3. La comunicación en red debe ser breve. Un trabajo puede hacer que las conexiones de red que utilizan llamadas de sistema como `socket()`, pero una conexión de red que permanece abierta por un largo periodo puede retrasar los puntos de control y la migración.
4. Envío y recepción de las señales `SIGUSR2` o `SIGTSTP` (son señales enviadas a un proceso para indicar las condiciones definidas por el usuario) no están permitidas. Condor se reserva estas señales para su propio uso, pero enviar o recibir señales diferentes a las mencionadas anteriormente si lo está.
5. Las alarmas, temporizadores y `sleeping` no está permitido. Esto incluye `alarm()`, `gettimer()`, y `sleep()`.
6. Múltiples hilos de nivel de kernel no están permitidos. Sin embargo los múltiples hilos de nivel de usuario están permitidos
7. Los archivos de memoria asignada no están permitidos. Esto incluye las llamadas como `mmap()` y `munmap()`.
8. Los archivos bloqueados están permitidos, pero no se conservan entre puntos de control.
9. Todos los archivos deben ser abiertos en solo lectura o solo escritura. Un archivo abierto para lectura y escritura puede causar problemas si un trabajo debe ser revertido a un punto de control anterior. Por razones de compatibilidad, un archivo abierto para lectura y escritura tendrá como resultado una advertencia, pero no un error.
10. Una buena cantidad de espacio de disco debe estar disponible en la máquina para almacenar imágenes de punto de control. Una imagen de punto de control es aproximadamente igual a la memoria virtual consumida por un trabajo mientras se ejecuta. Si el espacio de disco es muy pequeño, un servidor especial de punto de control puede ser designado para almacenar todas las imágenes de punto de control para un grupo.
11. En Linux, el trabajo debe ser enlazado de forma estática.

#### ■ Implicaciones de seguridad

Condor hace una cantidad significativa de trabajo para prevenir los riesgos de seguridad, pero aun así las lagunas existen. Condor puede ser instruido para ejecutar programas de usuario solo como usuario `nobody` en Unix, un inicio de sesión de usuario que tradicionalmente tiene

un acceso muy restringido. Condor le ofrece al administrador un gran número de mecanismos de seguridad para aplicar las políticas deseadas.

- **Los trabajos necesitan ser re-vinculados para conseguir un punto de control y una llamada remota al sistema**

Aunque por lo general no es necesario los cambios de código fuente, Condor requiere que los trabajos se re-vinculen con las bibliotecas de Condor para tomar ventaja de los puntos de control y las llamadas remotas.

## Disponibilidad

Condor actualmente está disponible como descarga gratuita desde el Internet en el URL <http://www.cs.wisc.edu/condor/downloads-v2>. Las distribuciones de Condor están disponibles para las plataformas que se detallan en el Tabla 2.1.

Arquitectura	Sistema Operativo
Intel X86	Red Hat Enterprise Linux 3
	Red Hat Enterprise Linux 4
	Red Hat Enterprise Linux 5
	Fedora Core 1 – 11
	Debian Linux 5.0
	Windows 2000 Professional and Server
	Windows 2003 Server
	Windows 2008 Server
	Windows XP Professional
	Windows Vista
	Windows 7
	Macintosh OS X 10.4
	Opteron x86_64
Red Hat Enterprise Linux 5	
Red Hat Enterprise Linux 6	
Debian Linux 5.0	
Debian Linux 6.0	

Tabla 2.1: Plataformas soportadas por Condor 7.7.5



### 2.2.2. Xgrid

Xgrid es una aplicación de MAC OS X diseñado para llevar la facilidad de Apple al uso de forma paralela y distribuida de alto rendimiento proporcionando una manera más fácil de ejecutar aplicaciones de Cómputo intensivo. Xgrid permite formar un grupo de nodos (computadoras individuales) y crear un Grid. Con Xgrid se puede crear computación de largo funcionamiento conocidos como “trabajos” y presentar dichos trabajos a un controlador, el cual es un equipo dedicado en la red que se encarga de la administración de estos trabajos. Puede crear múltiples tareas de cada trabajo y distribuirlas entre los múltiples nodos.

Xgrid introduce el concepto conocido como “Recuperación de escritorio”<sup>1</sup> el cual se utiliza para descubrir la acción de acoplamiento de computadoras individuales en la red cuando las computadoras estarían normalmente inactivas. La recuperación de escritorio es una gran característica de Xgrid, ya que permite que los cálculos se ejecuten solo cuando las máquinas están en modo inactivo y como resultado no afecta a la persona que está trabajando en esa máquina. También permite que los sistemas funcionen como nodos dedicados, lo que significa dedicar el 100% de su potencia de procesamiento para los trabajos en el Grid. Esta combinación de recuperación de escritorio y máquinas dedicadas permiten al investigador maximizar el uso del poder de procesamiento.

En cuanto a las restricciones y limitaciones teóricamente Xgrid no tiene ninguna. En teoría, Xgrid debe ser capaz de permitir que cualquier número de sistemas puedan conectarse a la red, lo que resulta en N Gigahertz de poder efectivo. Sin embargo el número de máquinas que se pueden conectar no deben exceder de 300 GHz, aunque sin duda es una cifra es bastante baja en términos de poder computacional, para el laboratorio promedio, el poder Xgrid que se puede aprovechar es más que suficiente.

Xgrid es un JMS ampliamente utilizado por universidades en sus proyectos de investigación, algunos ejemplos son:

- **Universidad de Stanford.-** El Dr. Charles Parnot y Kobilka Brian utilizaron Xgrid para realizar grandes cantidades de poder computacional para modelar los cambios del receptor adrenérgico beta 2 y tener una mejor comprensión de su farmacología. Esto se hace a través del modelado de acoplación de las proteínas G. Esta compleja investigación podría proporcionar una mayor comprensión de como estos receptores podrían reaccionar a los medicamentos

---

<sup>1</sup>Desktop Recovery

utilizados para combatir varias enfermedades relacionadas con el corazón, el sistema coronario y el asma.

- **Universidad de Utah.-** James Reynolds ha creado una red con velocidades de hasta 500 GHz que se ha puesto a disposición de estudiantes de artes y de ciencias. Los estudiantes de artes han utilizado la red para la presentación de escenas en 3D que han creado con un programa de gráficos conocido como Maya. Los estudiantes de ciencias lo han utilizado para trabajar en proyectos específicos de investigación y han encontrado el poder de la red como una ayuda concreta para sus estudios.
- **Reed College.-** Bajo la supervisión del profesor Richard Crandalk, y teniendo en cuenta el financiamiento del gobierno de los Estados Unidos, los estudiantes de esta universidad han utilizado Xgrid en sus estudios de epidemiología.

### Características

Aunque Xgrid no proporciona todas las funcionalidades que ofrecen proyectos como Condor o Globus, tiene muchas que estos proyectos no hacen.

- **Instalación rápida y fácil utilización**

El proceso de instalaciones rápido y eficaz ya que requiere una configuración mínima para permitir a una máquina individual participar. El conocimiento de línea de comandos no es necesario, aunque puede ser utilizado por aquellos que se sienten más cómodos. Xgrid es capaz de ser completamente administrado a través de una terminal.

- **Tecnología exclusiva**

La utilización de Rendezvous de Apple [etu12], tecnología que permite a las computadoras individuales en la red conectarse a un controlador sin tener que saber la dirección IP. Con uso de cocoa [App12] es posible integrar Xgrid para ayudar a las aplicaciones con sus tareas de cálculo intensivo.

- **Ahorro de tiempo**

De la misma forma que Condor, Xgrid tiene la capacidad de crear sus propios plugins para ahorrar tiempo en el envío de trabajos. Esto ayuda a que el administrador envíe una tarea personalizada.

- **Simplicidad**

Ocultación de temas complejos como la administración de datos, la ejecución de trabajo y agregación de resultados por parte del usuario. GUI basado en las herramientas Grid para su monitoreo. La amplia gama de entrada y permitirá que cualquier operación se ejecute estilo UNIX.

- **Soporte**

Su objetivo es apoyar a todos los tipos de necesidad de computación que van desde la representación gráfica para el cálculo científico hasta la representación matemática.

### **Limitaciones**

- **Limitación de poder**

La suma del poder de cómputo de las computadoras conectadas al Grid no debe de exceder de los 300 GHz.

- **Falta de grupos heterogéneos**

Xgrid solo admite en el grupo de trabajo computadoras con el sistema operativo Mac OS X.

### **2.2.3. Platform LSF**

Platform LSF es un software comercial vendido por Platform Computing para la administración de trabajo, el cual es el mas poderoso por demanda, distribuido y de entornos de alto rendimiento de computación. Ofrece un conjunto completo de capacidades para la administración de carga de trabajo, todos diseñados para trabajar juntos y hacer frente a las necesidades de computación de alto rendimiento.

Platform LSF incluye un amplio conjunto de políticas inteligentes, basados en el manejo de la planificación, lo que permite aprovechar al máximo los recursos de la infraestructura de cómputo. Su arquitectura es altamente escalable, permite planificar complejas cargas de trabajo, y administrar hasta peta-flop de recursos. Con el mejor soporte en la industria de HPC, Platform LSF proporciona la mas completa solución de centro de datos HPC para la administración de carga de trabajo.

### Características

- **Alto rendimiento, arquitectura escalable.**

Gracias a su arquitectura escalable soporta miles de usuarios concurrentes y trabajos. Virtualiza los grupos con el fin de compartir los recursos IT asegurándose que se cumpla la demanda.

- **Conjunto amplio de políticas inteligentes de planificación.**

Los recursos apropiados son asignados de manera automática a los usuarios adecuados para maximizar la eficacia. Soporta múltiples centros de política con un control amplio y flexible. Las preferencias personalizadas y políticas fairshare garantizan que se cumplen los SLA. Soporta múltiples centros de política con un control amplio y flexible, la carga de trabajo se ejecuta de forma fiable a tiempo y la utilización de los recursos es óptima.

- **Funciones fáciles de usar.**

Se reduce el esfuerzo administrativo delegando controles administrativos a jefes de departamento. Mediante la administración de carga de trabajo a través de una interfaz web intuitiva se puede reducir errores de envío.

- **El soporte a plataformas es heterogéneo.**

Se utiliza la dinámica de arranque del sistema operativo para mejorar la utilización de recursos, así mismo mezcla y combina máquinas virtuales y física para uso óptimo de dichos recursos. Toma ventaja inmediata de los GPU's de alto rendimiento, aprovecha las inversiones existentes en los recursos de IT e integra cluster's con múltiples plataformas.

- **Auto-administración avanzada.**

Minimiza la intervención del administrador con trabajos automatizados y reintentos para garantizar cero tiempo de inactividad. Aplica cambios de configuración, sin la necesidad de re-configurar o reiniciar por lo tanto asegura la continuidad y una alta disponibilidad.

- **Soporte de aplicaciones integrada.**

Simplifica el envío de trabajos, reduce el tiempo de instalación y número de errores de operación, protege la inversión de software existente y se integra perfectamente con las aplicaciones de terceros.

- **Seguridad integral y extensible.**

La seguridad está basada en políticas de acceso a los recursos computacionales, es transparente a los archivos de datos locales y remotos, además utiliza aplicaciones basadas en Windows sin comprometer la seguridad.

### Limitaciones

- **Costoso**

Platform LSF es un software comercial, el cual se vende por licencias anuales de alto costo.

## 2.3. Criterio de evaluación

### 2.3.1. Condor

Aprovecha los recursos computacionales que de otra manera serían desperdiciados y los pone a buen uso. Además agiliza las tareas de los investigadores al permitir la presentación de muchos trabajos a la vez. De esta manera, enormes cantidades de cómputo se pueden hacer con muy poca intervención del usuario. Por otra parte, Condor permite a los usuarios tomar ventaja de las máquinas que no están siendo utilizadas.

Ofrece características importantes para los usuarios. Como es no tener que modificar el código fuente de ninguna manera para tomar ventaja de sus beneficios. El código que puede ser re-vinculado a las bibliotecas de Condor adquiere dos propiedades más: Los trabajos pueden producir puestos de control y pueden realizar llamadas remotas al sistema.

Para los cálculos de larga duración, la capacidad de producir y utilizar puntos de control guardados puede beneficiarnos a conservar tiempo de cálculo acumulado de días e incluso de semanas. Condor guarda los puntos de control periódicamente o cuando la máquina en la que está ejecutando el trabajo deja de estar disponible. De este modo, el trabajo puede continuar en otra máquina (migración).

### 2.3.2. Xgrid

Permite ejecutar programas usando varias computadoras y múltiples procesadores en una simple computadora para realizar cálculos en paralelo.

Es un sistema generalizado, capaz de reunir un cluster de procesadores en demanda, detección y corrección de fallas, y empaquetar las tareas como sea necesario para el cómputo en paralelo de propósito general en varios sistemas.

El controlador Xgrid es una característica de Mac OS X Server, pero cualquier computadora con Mac OS X (versiones 10.4 y posteriores) pueden enviar trabajos a Xgrid o actuar como agente para llevar a cabo cálculos.

Tiene una gran facilidad de configuración. Mas fácil de instalar que otros JMS's. Así mismo como de fácil uso, no requiere que los usuarios usen línea de comandos. Oculta la arquitectura compleja de distribución de software y datos, ejecución de trabajo y agregación de resultado. Utiliza Redezvous para el descubrimiento automático de los controladores disponibles Xgrid.

La seguridad de Xgrid no es un gran problema, ya que soporta el modelo de seguridad de UNIX.

Gracias a su propiedad de extensibilidad proporciona una arquitectura de plug-in que permite a los desarrolladores ampliar la funcionalidad de los cluster's Xgrid.

### 2.3.3. Platform LSF

Con Platform LSF se puede administrar y acelerar el proceso de la carga de trabajo para aplicaciones de cómputo intensivo o datos intensivos, resultando en una solución en un tiempo más corto. Permite planificar de manera inteligente y garantizar el cumplimiento de las cargas de trabajo a través de una red distribuida, gracias al IT virtualizado se puede aprovechar al máximo todos los recursos de la infraestructura IT independientemente del sistema operativo o la arquitectura.

Ayuda a asegurar que todos los recursos disponibles se utilicen plenamente. Mediante la entrega de la máxima potencia de cálculo, se acelera el desempeño de las aplicaciones de cómputo intensivo o de datos.

Proporciona una óptima administración de SLA, una gran flexibilidad, visibilidad y control de la planificación del trabajo. Esto permite a las IT mejorar el servicio que proporciona a los interesados. Al garantizar una óptima utilización de infraestructuras de IT existentes, mas trabajo se hace con menos recursos, reduciendo hardware adicional y costos de administración.

Los recursos computacionales de los grupos de trabajo y las aplicaciones de administración de carga de trabajo son unos entornos altamente distribuidos, que van desde un cluster local hasta uno global. Se le permiten distribuir carga de trabajo a cualquier combinación de sistemas de hardware incluyendo equipos de escritorio, servidores y súper computadoras.

Se dice que Platform LSF es la solución de la administración de carga de trabajo más completa para entornos complejos y distribuidos HPC. Se ofrece un conjunto completo de capacidades de administración de carga de trabajo, todos diseñados para trabajar juntos para hacer frente a sus necesidades de computación de alto rendimiento.

El Tabla 2.2 muestra una comparación de las características de los JMS's analizados anteriormente.

	Condor	Xgrid	LSF
<b>Distribución</b>	Código abierto	Comercial	Comercial
<b>Soporte de sistema operativo</b>			
Linux	Si	No	Si
Windows NT	Si	No	Si
Mac OS X	Si	Si	Si
<b>Tipos de trabajo</b>			
Trabajos interactivos	No	No	Si
Trabajos paralelos	Si	Si	Si
<b>Características</b>			
Etapa entrada-salida	Si	Si	Si
Proceso de migración	Si	Si	Si
Balanceo dinámico de carga	No	Si	Si
Puntos de control	Si	No	Si
Recuperación de Fallos de demonios	Solo hosts	Solo hosts	Maestro y host

Tabla 2.2: Comparación de características de los administradores de cargas de trabajo

Al analizar la tabla anterior, se puede observar que: el más completo es Platform LSF pero el gran inconveniente es que es comercial, por lo cual queda descartado. Condor es un buen JMS, pero existen inconvenientes, como la dificultad de instalación y configuración, debido que esto no es intuitivo y existe poca documentación que se refiera al proceso de instalación de Condor en Mac OS X. Aún cuando Xgrid es un software comercial, si se posee una computadora o servidor con el sistema operativo MAC OS X, este viene instalado por defecto. Como se mencionó la red de Posgrado de la Facultad de Ingeniería Eléctrica cuenta con dispositivos con este sistemas operativo, por lo cual se tiene la infraestructura para poder instalar de manera sencilla Xgrid. Sumado a lo

anterior, existen varias aplicaciones con interfaz gráfica, las cuales son de gran utilidad tanto para la administración de Xgrid hasta el envío de trabajos. Por lo tanto se elegirá Xgrid.



## Capítulo 3

# Configuración de Xgrid

Una vez seleccionado Xgrid como el JMS que satisface las necesidades de cómputo, se realizará la instalación y configuración en la red de Posgrado de la Facultad de Ingeniería Eléctrica.

El controlador de Xgrid ya se encuentra instalado por defecto en las computadoras que tienen como sistema operativo Mac OS X con lo cual ya no es necesario realizar el proceso de instalación.

Por lo que sólo es necesario configurar los componentes de nuestra red.

Al momento de incluir un dispositivo al Grid este puede ser configurado para realizar tres distintas funciones (como se muestra en la Figura 3.1). Las cuales se listan a continuación:

- Agentes
  - Los agentes son computadoras que realizan el trabajo de cómputo, cualquier equipo que ejecute Mac OS X puede funcionar como agente. Los agentes son responsables de recibir los trabajos enviados desde el controlador, el procesamiento de los trabajos y la devolución de los trabajos hacia el controlador. Los agentes pueden funcionar como agentes especializados, lo que significa que siempre aceptan trabajos, y dedican el poder máximo de procesamiento al trabajo aceptado. Los agentes también se pueden configurar para que acepten trabajos solo cuando el protector de pantallas se encuentre activado, lo que significa que sólo acepta trabajos cuando la computadora que va a actuar como agente esta inactiva.

- Clientes

- Cualquier dispositivo con sistema OS X puede ser cliente Xgrid, solo requiere tener la aplicación Xgrid instalada, y tener una conexión de red funcional. El cliente proporciona acceso al controlador, y permite a cualquier usuario local enviar trabajos al controlador utilizando la línea de comandos de Xgrid o una interfaz gráfica. Cuando un trabajo se ha completado el cliente es notificado y puede recuperar los resultados desde el controlador. Una de las mejores características del cliente, es la posibilidad de crear un trabajo personalizado llamado “plug-in” que se almacena en el controlador. Un plug-in puede ser utilizado más adelante para ejecutar instancias similares de la misma clase de trabajo.

- Controlador

- El controlador de Xgrid es el componente principal del Grid, ya que se encarga de administrar la comunicación y los recursos de todos los agentes disponibles en su grupo. El controlador recibe propuestas de trabajo del cliente, envía cada tarea a un agente y proporciona los resultados del trabajo al cliente. Se debe mencionar que solo puede haber un controlador por Grid. Cada controlador puede tener un número arbitrario de agentes conectados, sin embargo Apple aconseja sólo utilizar 128 agentes por controlador, debido al número sockets existentes en el sistema del controlador.

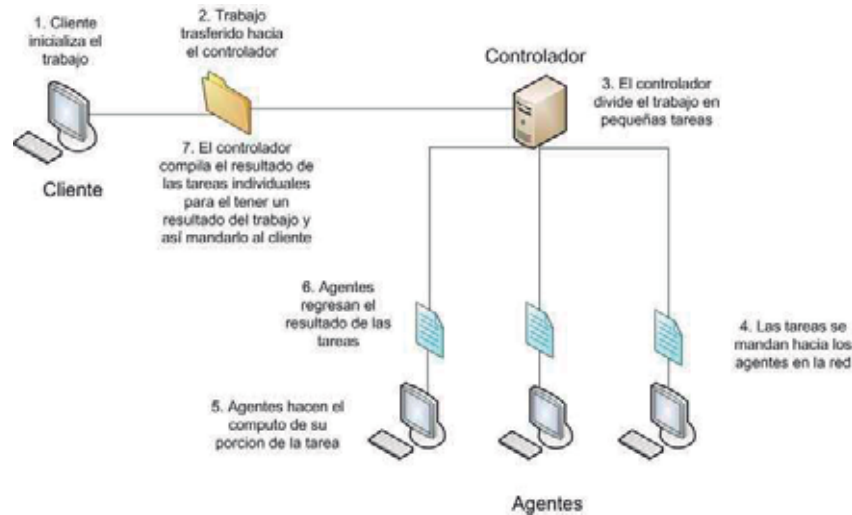


Figura 3.1: Roles e interacción de computadoras con Xgrid

### 3.1. Configuración del agente Xgrid

En primer lugar se configurará el agente Xgrid. Como ya se menciona un agente es una computadora que realiza los cálculos que contiene un trabajo.

Para comenzar a configurar el agente, se debe habilitar a cada componente como se muestra en la Figura 3.2.



Figura 3.2: Configuración de agente

A cada miembro del Grid se le debe configurar los siguientes parámetros:

- Controlador.- Se tienen dos opciones para elegir el controlador con el que nos vamos a conectar:
  - Usar el **primer controlador disponible**.- Esta opción permitirá que el sistema Xgrid busque un controlador en la red y se conecte al primero que este disponible.
  - Usar un **controlador específico**.- Permite ingresar la dirección IP, FQDN (Fully Qualified Domain Name) o elegir un controlador detectado vía Bonjour como el controlador a utilizar. En nuestro caso se utiliza esta última opción, donde se elige *dep3* o se ingresa la dirección IP 148.216.38.92.
- Agente acepta tareas.- En ésta opción se especifica la política para la aceptación de tareas en

el agente, puede ser solo cuando el agente este sin utilizar o siempre.

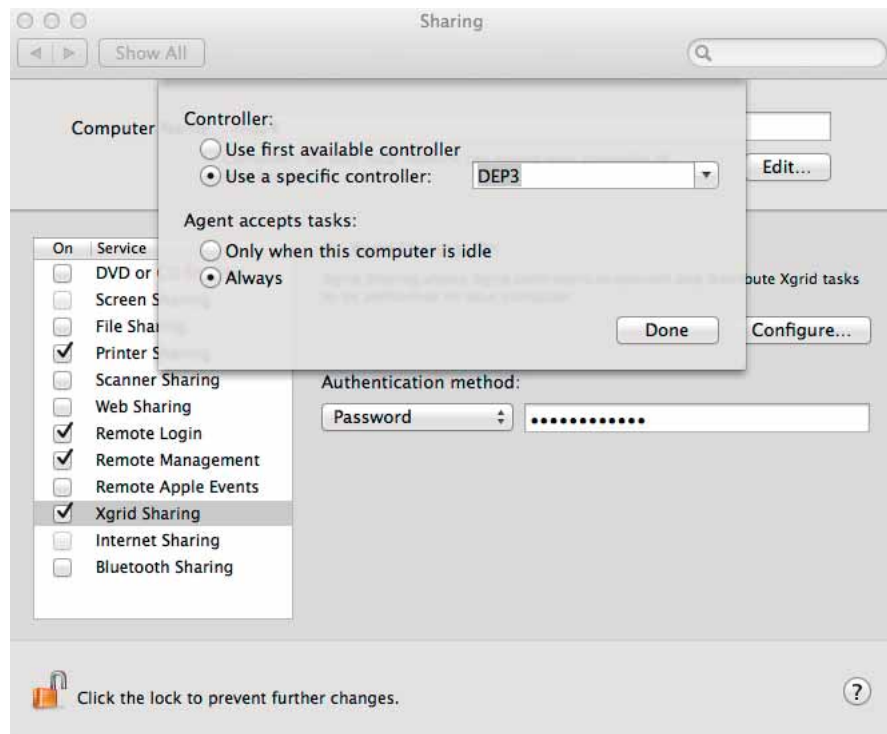


Figura 3.3: Opciones de configuración

Especificación del método de autenticación, existen 3 maneras de hacerlo:

- Ninguno.- No se necesita realizar ninguna autenticación para recibir parte del trabajo enviado por el controlador.
- Single sign on Kerberos<sup>1</sup>.- Es mas segura que la opción de contraseña, esta forma de autenticación necesita una base de datos de claves para cada uno de los agentes.
- Contraseña.- Se utiliza una cadena encriptada para autenticarse con el controlador y así recibir parte del trabajo. En éste caso se debe proporcionar una clave.

<sup>1</sup>Sistema por el cual las computadoras pueden delegar la autenticación de usuarios autorizados a un servidor de confianza, por lo general esto forma parte de una configuración de LDAP o un DNS local.

## 3.2. Configuración de controlador Xgrid

El controlador realiza la función mas importante, por que es quien recibe el trabajo y es el encargado de dividirlo en tareas mas pequeñas para que puedan ser procesadas por los agentes. Cabe mencionar que para facilitar la configuración del contralor es necesario tener instalado Mac OS X Server, de lo contrario se necesitaría instalar la aplicación XgridLite.

Para configurar el controlador se debe ingresar a “Server Admin” [App] como se muestra en la Figura 3.4.

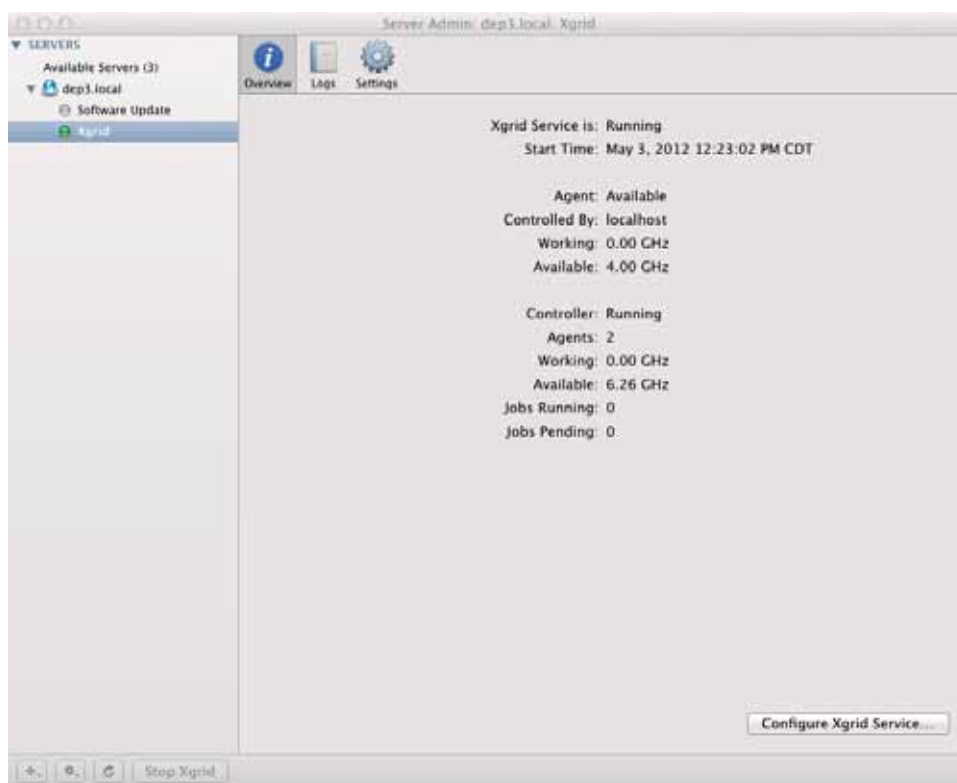


Figura 3.4: Server Admin

En la pestaña de configuración, se tienen 2 opciones:

- Controlador: En esta opción se puede activar el servicio de controlador, la ubicación donde se guarda los datos importantes del controlador, por ejemplo una base de datos que contiene el nombre de los agentes que están conectados. Así como la manera en que se va a autenticar el

cliente con el controlador y el controlador con el agente.



Figura 3.5: Configuración del controlador

- Agente: Esta opción permite configurar el controlador como agente, habilitando el servicio de agente, y se configurará de la misma forma que un agente (ver apéndice A). En la sección de “Agente acepta trabajos” se recomienda utilizar la opción “Siempre” si nuestro servidor es una máquina multi-núcleo y no se utiliza para otro servicio, en cambio si a menudo esta ocupada prestando servicios se debe elegir la opción “Solo cuando el equipo este inactivo”.

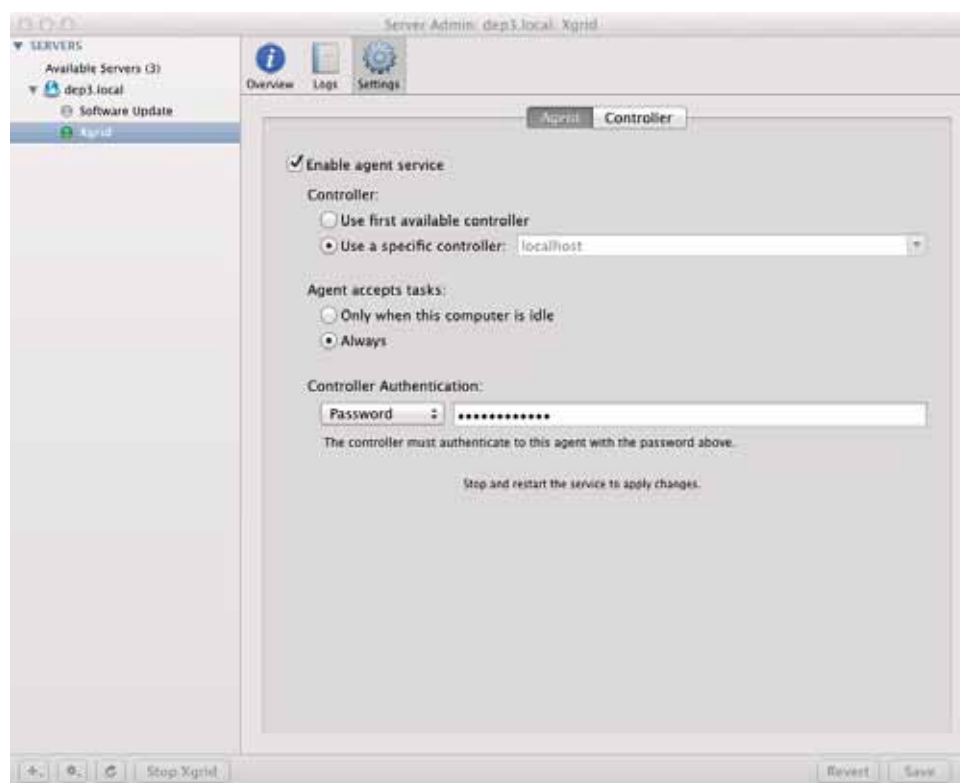


Figura 3.6: Configuración del controlador como un agente

Para terminar de configurar el controlador, se inicia el servicio presionando el botón “Start Xgrid”. Para ver la configuración detallada ver el Apéndice A.

### 3.3. Xgrid Admin

Xgrid Admin es una aplicación gráfica que permite la administración tanto de los trabajos, como de los agentes que participan en un Grid (ver Figura 3.7 ).



Figura 3.7: Información General de Xgrid

Existen tres pestañas diferentes en Xgrid Admin las cuales son:

- Información General.- Muestra cuantos agentes están configurados en la red, CPU disponible utilizado y los trabajos en ejecución.
- Agentes.- Contiene las características de los agentes conectados a la red; datos tales como: nombre, ip, estado, poder de CPU y poder activo.
- Trabajos.- Se puede observar los trabajos que se están ejecutando así como los que ya han terminado. Se muestran propiedades tales como la hora de envío, estado, progreso y el porcentaje de CPU que esta siendo utilizando (ver Figura 3.8).



Name	Status	Date Submitted	Progress	CPU Power	Tasks
/bin/hostname	Finished	12:00:48 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:04:04 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:04:14 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:11:26 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:15:29 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:16:29 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:20:19 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:20:47 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:23:08 03 May 12	100%	0.00 GHz	1
/bin/hostname	Finished	12:50:28 03 May 12	100%	0.00 GHz	1

Name: /bin/hostname  
 Identifier: 0  
 Date Started: Thursday, May 3, 2012 12:06:08 PM Central Daylight Time  
 Date Stopped: Thursday, May 3, 2012 12:06:08 PM Central Daylight Time

Total: 10 jobs / 10 finished

Figura 3.8: Trabajos

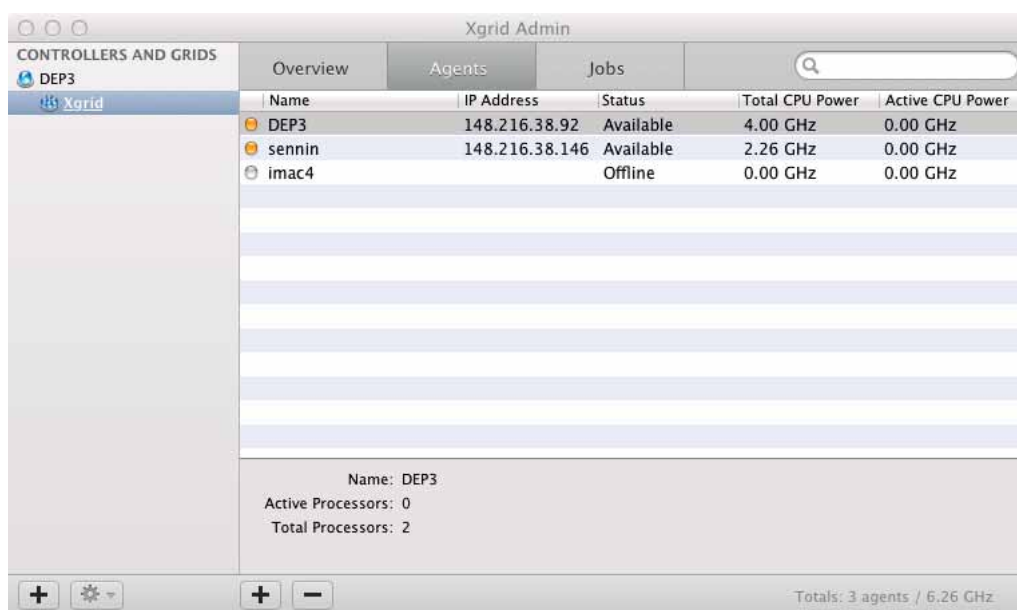
Con Xgrid Admin podemos:

- Comprobar el estado de la red y su actividad, incluyendo el número de agentes que trabajan y están disponibles, la potencia de procesamiento en el uso y disposición, el número de trabajos en ejecución y pendientes.
- Agregar o quitar controladores y redes administradas.
- Ver una lista de agentes en la red, así como la potencia de CPU disponible.
- Agregar o eliminar los agentes a la red.
- Ver la lista de trabajos en la red, la fecha y hora que se presentó cada trabajo y su progreso, y la potencia de CPU utilizado para cada uno.
- Eliminar trabajos de la red.
- Parar un trabajo en proceso.
- Empezar un trabajo que había sido detenido o completado.

Xgrid admin dispone de controles en su interfaz gráfica y comandos de menú para todas sus opciones.

La interfaz gráfica de Xgrid Admin indica mediante esferas de colores el estado de controladores, agentes y trabajos. Dependiendo del color se pueden leer como sigue:

- Transparente: controlador o agente esta desconectado.
- Gris: trabajo enviado.
- Verde: Controlador conectado; agente trabajando ; trabajo corriendo.
- Amarillo: agente disponible pero no corriendo.
- Rojo: agente no disponible; trabajo fallido o cancelado.
- Azul: trabajo completado.



The screenshot shows the Xgrid Admin window with the 'Agents' tab selected. The table below lists the connected agents:

Name	IP Address	Status	Total CPU Power	Active CPU Power
DEP3	148.216.38.92	Available	4.00 GHz	0.00 GHz
sennin	148.216.38.146	Available	2.26 GHz	0.00 GHz
imac4		Offline	0.00 GHz	0.00 GHz

Below the table, the details for the selected controller (DEP3) are shown:

```

Name: DEP3
Active Processors: 0
Total Processors: 2

```

At the bottom of the window, a status bar indicates: Totals: 3 agents / 6.26 GHz.

Figura 3.9: Agentes conectados a Xgrid

### 3.4. GridStuffer

GridStuffer es una aplicación Cocoa para enviar trabajos multi-tareas usando un formato muy simple que se asemeja mucho al formato de línea de comando de Xgrid. La idea es emular el formato batch plist sin complejidad y proporcionar una interfaz gráfica para la recuperación de resultados. La aplicación utiliza datos básicos para almacenar información sobre el progreso del trabajo.

GridStuffer permite la conexión simultánea a varios controladores al mismo tiempo. Los controladores encontrados en la red local (vía Bonjour) se en listarán automáticamente. Si deseamos añadir algún otro controlador solo basta con presionar el botón de “+” y teclear la dirección IP.



Figura 3.10: Controladores existentes en Xgrid visto con XgridStuffer

El concepto de MetaJob es definido por GridStuffer como una lista de comandos de shell escritos en un archivo de entrada. En la ventana de MetaJobs se puede agregar, correr y parar lo trabajos que se envían al controlador, además se puede observar el nombre, el número de tareas y su progreso.



Figura 3.11: MetaJobs

Con GridStuffer se pueden crear trabajos de una manera simple (ver Figura 3.12), con solo ingresar el nombre del nuevo trabajo, en archivo de entrada y el directorio de salida. Donde el archivo de entrada es un script, el cual contiene una secuencia de instrucciones cíclicas y el directorio de salida es la ubicación donde se van a almacenar los resultados.

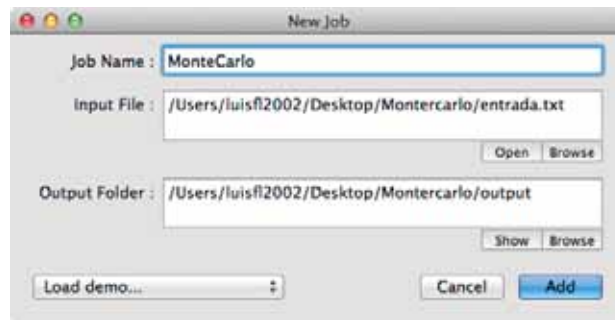


Figura 3.12: Nuevo trabajo

Con GridStuffer se puede realizar el seguimiento de cada uno de los comandos en MetaJob es posible observar el estado de los comandos en la ventana MetaJob Inspector.



Figura 3.13: Inspector de MetaJobs

### 3.5. Herramienta de clasificación de Agentes

La herramineta ART funciona en base a un archivo (script) que se puede ejecutar en los agentes, este devuelve una puntuación, la cual es un valor numérico en la salida estándar (STDOUT). El controlador mantiene un registro de todas las puntuaciones, una por cada agente, y las escribe en un marcador virtual. Los agentes deben pasar un valor de umbral para ser candidatos a ejecutar la

tarea.

Para elegir los agentes que capaces de realizar el trabajo, el controlador debe realizar las siguientes acciones (ver Figura 3.14):

- Asignar puntuaciones: El controlador envía el ART a todos los agentes disponibles, y recibe de vuelta las puntuaciones para cada agente.
- Clasificar: Los agentes con una puntuación de 0 no serán candidatos. Los agentes restantes se clasifican en base a su puntuación.
- Asignar los trabajos: Las tareas del trabajo se envían a los agentes seleccionados. Los agentes con las puntuaciones más altas tiene prioridad en ser utilizados.
- Obtener los Resultados: el cliente sólo recibe los resultados de vuelta, y no tiene que preocuparse por el marcador.

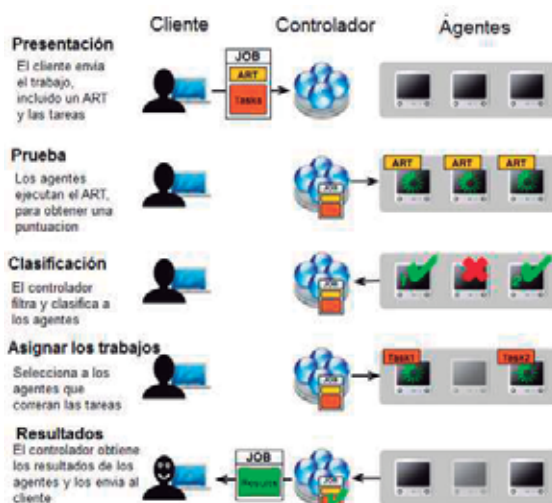


Figura 3.14: Proceso del controlador ante un ART

Para determinar las especificaciones de un agente, existen cuatro comandos de gran utilidad para ser utilizados un script ART, los cuales son: `/usr/sbin/sysctl`, `/bin/hostname`<sup>2</sup>, `/usr/bin/sw_versions` y `/usr/sbin/system_profiler`. El comando `sysctl` muestra datos como el tamaño de memoria, la velocidad y tipo de CPU. El comando `hostname` regresa el nombre del agente. El comando `sw_versions`

<sup>2</sup>Muchas veces el nombre que tiene la maquina internamente no es el mismo presentado externamente

indica la versión del SO directamente sin tener que descifrar el número de construcción del kernel que devuelve sysctl. Por último debemos tomar en cuenta que el comando `system_profiler` es lento y difícil de usar, ya que puede regresar un formato que varía muy poco computadora a computadora.

Cabe mencionar que existen 5 comandos para utilizar ART, los cuales son los siguientes:

- **-art [ruta-del-art]**: Se utiliza para clasificar agentes.
- **-artid [identificador-art]**: Identifica de forma única la herramienta de clasificación de los agentes.
- **-artequal [valor-del-art]**: Restringe las tareas para ejecutarse únicamente en agentes los cuales la herramienta de clasificación devuelva un valor especificado por el art.
- **-artmax [valor-del-art]**: Restringe las tareas para ejecutarse únicamente en agentes los cuales la herramienta de clasificación devuelva un valor que no supere al especificado por el art.
- **-artmin [valor-del-art]**: Restringe las tareas para ejecutarse únicamente en agentes los cuales la herramienta de clasificación devuelva un valor que supere al especificado por el art.

Para ejemplificar lo anterior tenemos el script llamado “imac3” (ver Apéndice B), el cual fuerza a un trabajo funcionar en un agente específico llamado “imac3”.

Para asegurarnos que el script es ejecutable utilizamos la siguiente línea:

```
chmod a+x imac3.pl
```

Para ejecutar el script debemos de introducir la siguiente línea de comando:

```
xgrid -job submit -art imac3.pl -artid foo -artequal 1 el_comando_de_envio
```

Otro ejemplo de un script ART es “RAM”, el cual clasifica los agentes en función de la cantidad de RAM (ver Apéndice B).

Desde MAC OS X 10.7, los scripts ART son una herramienta de clasificación, también se puede utilizar esta técnica para simplemente alterar puntaje para preferir un agente, y no tanto para excluirlo.

### 3.5.1. Problemática con ART

Algunas causas por las que trabajos que utilizan ART pudieran no ser ejecutados son:

- Los scripts ART deben tener permiso de ejecución en el host local.
- La ruta o el nombre del archivo ART mal escrito.
- El script ART tiene un error o no clasifica ningún agente.
- Un script ART se envía a un agente que luego fue desconectado o por otra causa no puede responder.

En estos casos, el trabajo quedará pendiente con ningún mensaje de error. Estas parecen ser las fallas de diseño, ya que se trata no sólo de requisitos innecesarios, los dos primeros son detectables al momento del envío. El cliente xgrid debería al menos dar un mensaje de error y rechazar el envío o el controlador debe marcar el trabajo como fallido. En el caso de los dos primeros, el registro no mostrará el envío del script ART. En el tercero se enviará el script a los agentes disponibles. En el caso tres, los resultados nunca son reportados en el registro, y los trabajos permanecen permanentemente en espera.

Otras veces el controlador hace caso omiso al script ART en la selección del agente, existen distintas causas por lo que puede ocurrir esto, las cuales son:

- Desde línea de comando:
  - Cuando se utiliza `-art`, muchas veces se requiere `-artid` y `-artequal`.
  - Al script ART se le debe de dar la ruta relativa en el equipo local (no hay rutas absolutas o implícitas en CLI).
  - Los envíos por grupos ignoran la mayoría de los argumentos en la línea de comando.

En esos casos, el script ART se ejecuta, pero el trabajo no se ejecuta en el primer agente disponible. Se podrá ver que el script falta en el registro, pero no existen mensajes de error ni advertencias.

- En general:
  - El valor que regresa el script debe ser un número en ASCII.
  - Si devuelve una cadena que no se evalúa como un número, este se evalúa como un cero.

- El comando `-artequal` debe ser un número, los intentos de hacer comparaciones utilizando un argumento cadena fallará.

En estos tres últimos casos, el script ART se ejecuta y los resultados aparecerán en el registro, pero el valor no es lo que esperaba, por lo que los trabajos pueden ejecutarse en agentes inesperados. No hay mensajes de error ni advertencias. Con lo anterior se puede concluir que el uso de ART es responsabilidad del cliente.

### 3.6. Comentarios Finales

En este Capítulo se pudo observar que al momento de incluir un dispositivo a Xgrid este puede ser configurado para realizar tres distintas funciones, las cuales son controlador, agente y cliente. Cada uno, tiene una función específica en el Grid y un proceso distinto de configuración. Se ha dado proporcionado una breve explicación de como se configuran cada uno de los componentes que participan en Xgrid. Adicionalmente se introdujeron algunas aplicaciones existentes para interactuar con Xgrid, las cuales facilitan el uso del Grid. Por último se dio una breve explicación del uso de los scripts ART, los cuales nos ayudan a seleccionar los agentes donde se quiere ejecutar un trabajo, cabe mencionar que esta tarea es responsabilidad del cliente.



## Capítulo 4

# Pruebas y Resultados

Para comprobar la correcta configuración de Xgrid, se realizaron algunas pruebas utilizando GridStuffer.

Las pruebas tienen por objeto comprobar la correcta configuración de Xgrid en la red. La primera ejecuta un comando Unix que obtiene el calendario del sistema desde línea de comandos; mientras la segunda es un programa en perl, el cual recibe como parámetros dos números los divide y el resultado lo eleva al cuadrado; la tercera es un poco más compleja, debido que se trata de la implementación del método de Monte-Carlo.

Para enviar un trabajo en GridStuffer es necesario proporcionar el nombre para nuevo trabajo, el archivo de entrada y el directorio de salida, tal como se muestra en la Figura 4.1. El archivo de entrada es un script, el cual contiene una secuencia de instrucciones cíclicas. GridStuffer permite mediante el archivo de entrada especificar las instrucciones que se desean ejecutar en el Grid. El uso del archivo de entrada ayuda a prevenir errores y facilita su corrección. Si no se cuenta con GridStuffer el formato que se tiene que seguir para mandar un trabajo es el siguiente:

**xgrid -job run (ruta del compilador del lenguaje utilizado) programa.**

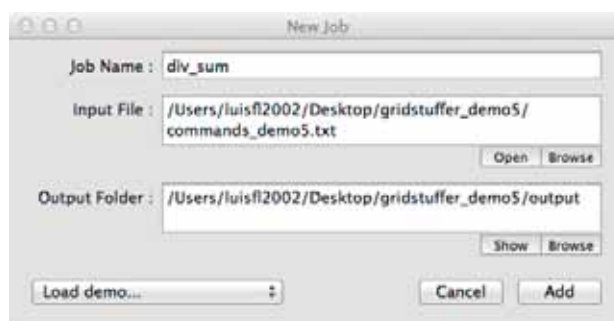


Figura 4.1: Trabajos nuevos

**Nota:** Los programas enviados con Grid Stuffer deben tener en su primer línea `#!` con la ruta donde se encuentra el el interprete del lenguaje utilizado para escribir el script del trabajo, por ejemplo si el script estuviera escrito en Python se agregaría lo siguiente: `#!/usr/bin/python`.

## 4.1. Calendario

Para esta prueba se utiliza el comando `cal`, el cual despliega el calendario del sistema en pantalla. El archivo de entrada tiene el nombre de “Calendario”, contiene las siguientes líneas de comando:

```
/usr/bin/cal 1 2000
/usr/bin/cal 2 2000
/usr/bin/cal 3 2000
/usr/bin/cal 7 2000
/usr/bin/cal 8 2000
.
.
.
/usr/bin/cal 12 2000
```

El script anterior pide el calendario de un mes del año 2000, el mes está determinado por el primer parámetro (1 para enero, 2 para febrero y así sucesivamente). Con el fin de probar el funcionamiento del Grid, se repiten las instrucciones para cada mes 600 veces.

Para observar si en verdad el trabajo fue enviado a Xgrid, se ingresa a la aplicación de Xgrid Admin para monitoriar el avance del trabajo.

Como se puede observar en la Figura 4.2 todos los agentes están participando en la resolución del trabajo.

Name	IP Address	Status	Total CPU Power	Active CPU Power
DEP3	127.0.0.1	Working	4.00 GHz	4.00 GHz
sennin		Offline	0.00 GHz	0.00 GHz
imac	148.216.38.162	Working	4.32 GHz	4.32 GHz
imac4	148.216.38.187	Working	2.66 GHz	2.66 GHz
imac3	148.216.38.195	Working	2.66 GHz	2.66 GHz

Name	Active Processors	Total Processors
imac3	1	1

Totals: 5 agents / 13.64 GHz

Figura 4.2: Monitoreo del estado de los agentes

Con Xgrid admin podemos ir monitoreando las tareas del trabajo de “Calendario” como se muestra en la figura 4.3.

Name	Status	Date Submitted	Progress	CPU Power	Tasks
Calendario [586]	Finished	11:33:10 21 Jun 12	100%	0.00 GHz	1
Calendario [587]	Finished	11:33:10 21 Jun 12	100%	0.00 GHz	1
Calendario [588]	Running	11:33:11 21 Jun 12	0%	0.00 GHz	1
Calendario [590]	Pending	11:33:11 21 Jun 12	0%	0.00 GHz	0
Calendario [591]	Running	11:33:11 21 Jun 12	0%	0.00 GHz	1
Calendario [592]	Finished	11:33:12 21 Jun 12	100%	0.00 GHz	1
Calendario [593]	Pending	11:33:12 21 Jun 12	0%	0.00 GHz	0
Calendario [594]	Pending	11:33:12 21 Jun 12	0%	0.00 GHz	0
Calendario [595]	Starting	11:33:12 21 Jun 12	0%	0.00 GHz	1
Unknown	Unknown		0%	0.00 GHz	0

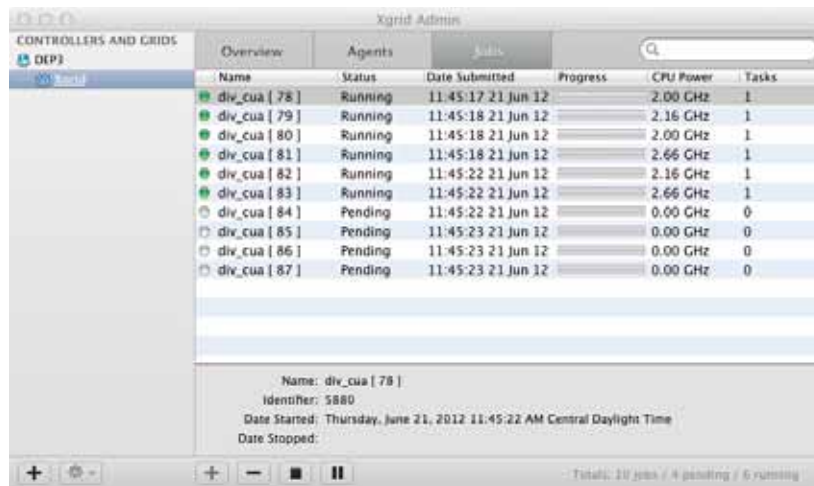
  

Name	Identifier	Date Started	Date Stopped
Calendario [586]	5106	Thursday, June 21, 2012 11:33:11 AM Central Daylight Time	Thursday, June 21, 2012 11:33:12 AM Central Daylight Time

Totals: 10 jobs / 3 pending / 3 starting / 2 running / 3 finished

Figura 4.3: Tareas realizadas





The screenshot shows the Xgrid Admin interface with a table of jobs. The table has columns for Name, Status, Date Submitted, Progress, CPU Power, and Tasks. Below the table, there is a detailed view for a specific job.

Name	Status	Date Submitted	Progress	CPU Power	Tasks
div_cua [ 78 ]	Running	11:45:17 21 Jun 12		2.00 GHz	1
div_cua [ 79 ]	Running	11:45:18 21 Jun 12		2.16 GHz	1
div_cua [ 80 ]	Running	11:45:18 21 Jun 12		2.00 GHz	1
div_cua [ 81 ]	Running	11:45:18 21 Jun 12		2.66 GHz	1
div_cua [ 82 ]	Running	11:45:22 21 Jun 12		2.16 GHz	1
div_cua [ 83 ]	Running	11:45:22 21 Jun 12		2.66 GHz	1
div_cua [ 84 ]	Pending	11:45:22 21 Jun 12		0.00 GHz	0
div_cua [ 85 ]	Pending	11:45:23 21 Jun 12		0.00 GHz	0
div_cua [ 86 ]	Pending	11:45:23 21 Jun 12		0.00 GHz	0
div_cua [ 87 ]	Pending	11:45:23 21 Jun 12		0.00 GHz	0

Job Details:

- Name: div\_cua [ 78 ]
- Identifier: 5880
- Date Started: Thursday, June 21, 2012 11:45:22 AM Central Daylight Time
- Date Stopped:

Total: 10 jobs / 4 pending / 6 running

Figura 4.5: Trabajos del Programa operación aritmética

Los resultados se almacenan en el directorio salida el cual se específico al momento de crear el trabajo.

### 4.3. Método Monte-Carlo

El siguiente ejemplo es un problema mas complejo, es la implementación del método de Monte-Carlo, el cual es un método numérico que permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias. En nuestro caso se utiliza un script escrito en Python, que implementa el método de Monte-Carlo para obtener una aproximación de  $\pi$ .

En la Figura 4.6 se puede observar los parámetros proporcionados a GridStuffer para la creación del trabajo Monte-Carlo, donde el archivo de entrada contiene varias líneas como la siguiente:

**montecarlo.py N**

Donde montecarlo.py es el programa (ver código en el Apéndice B) y N es el número de muestras, para nuestro caso N va desde 10 hasta 100000000.

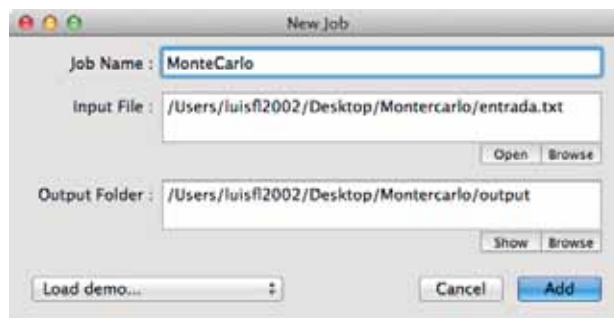


Figura 4.6: Creación del trabajo de Monte-Carlo

Una vez creado el trabajo de Monte-Carlo se envía al controlador para que reparta las tareas, esto se puede observar en la Figura 4.7.

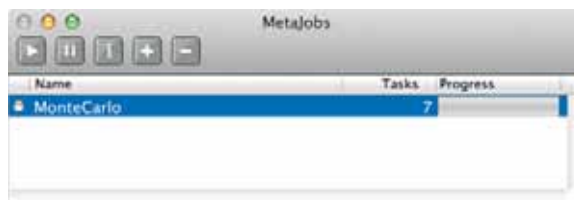


Figura 4.7: Envío del trabajo Monte-Carlo al controlador con GridStuffer

Ya finalizado el trabajo, se pueden consultar los resultados en el directorio de salida. El Tabla 4.1 muestra el resultado de las aproximaciones.

número de muestras	Aproximación a Pi
100	2.8
1000	3.192
10000	3.1572
100000	3.14112
1000000	3.141108
10000000	3.1413304

Tabla 4.1: Aproximación a Pi

En la tabla 4.1 se puede observar que conforme se aumenta el número de iteraciones la aproximación de  $\pi$  es mejor.

#### 4.4. Comentarios Finales

En el Capítulo se hicieron varias pruebas para verificar el funcionamiento de Xgrid con ayuda de GridStuffer. Las pruebas realizadas son sencillas debido a que se pretende verificar el correcto funcionamiento y el rendimiento del Grid, y no el desarrollo de un proyecto.

## Capítulo 5

# Conclusiones y trabajo futuro

### 5.1. Conclusiones

En la actualidad los JMS's se han convertido en un factor importante en el mundo de la investigación, esto debido a que muchos proyectos requieren hacer un gran cantidad de operaciones en un corto tiempo. Los proyectos que se han llevado acabo haciendo uso de un JMS van desde la graficación en 3D hasta problemas tales, como encontrar la cura de algunas enfermedades. Contar con un JMS es de gran importancia ya que su uso facilita el desarrollo de los proyectos en un centro de investigación, tal como es el caso del Posgrado de Ingeniería Eléctrica.

En el presente trabajo se analizaron tres JMS, lo anterior con el objetivo de seleccionar uno para su instalación y configuración en el Departamento de Estudios de Posgrado de la FIE.

Cada uno de los JMS's comparados en el documento cuenta con características particulares, Platform LSF no fue seleccionado debido a que es un software comercial en este momento no es justificable su adquisición. En el caso de Condor aún cuando cuenta con muchas características deseables, se descarto debido a su mayor complejidad en el proceso de instalación y configuración. Pero Xgrid es amigable en su configuración, cuenta con una interfaz gráfica y su utilización es fácil.

Por tanto se eligió Xgrid, debido que se cuenta con la infraestructura para configurarlo, adicionalmente no es necesario realizar el proceso de instalación, ya que, en MAC OS X el sistema Xgrid ya se encuentra instalado. Una vez configurado Xgrid, se instalaron herramientas para facilitar el proceso de envío de trabajos y su supervisión. Para la supervisión de trabajos se utiliza la apli-



cación Xgrid Admin, en la cual es posible observar información tal como que máquinas participan en el Grid, el estado de los trabajos, etc. Para facilitar el envío de trabajos se utiliza una aplicación desarrollada por la Universidad de Stanford llamada Grid Stuffer, la cual permita ingresar un script de shell el cual facilita la descripción del trabajo. Escribir las instrucciones en un archivo facilita el proceso de corrección de errores, además de ayudar a prevenirlos. Un claro caso de la utilidad de esta forma de enviar tareas, puede observarse en los ejemplos presentados en este trabajo en los que se muestra como probar un mismo programa con diferentes parámetros. Adicionalmente no es necesario escribir los comandos de xgrid, ya que estos son manejados como opciones de Grid Stuffer.

Este gran beneficio de Grid Stuffer se vio reflejado cuando se hicieron las pruebas debido que se requerían hacer pruebas sencillas pero que explotarían los recursos del sistema por lo que se requería de enviar un gran número de trabajos al sistema, con un solo script se mandaron centenares de tareas para que el sistema las ejecutará, que sin ayuda de Grid Stuffer esto se vería un poco tedioso y tardado.

Como se puede observar en las pruebas se utilizaron diferentes lenguajes de programación para ser ejecutados en Xgrid, se deduce que si se tiene la infraestructura de software adecuada para la ejecución del programa que es enviado en el controlador de Xgrid, este es capaz de ejecutar y concluir la tarea exitosamente. Por lo tanto no hay restricciones acerca de esto y los investigadores como los alumnos pueden enviar cualquier tipo de trabajo a Xgrid.

## 5.2. Trabajo Futuro

Como trabajo futuro se propone la instalación de otro JMS de los mencionados en la parte de desarrollo para poder comparar el rendimiento de los varios JMS en la misma infraestructura. Además de realizar pruebas con programas reales, para explotar los beneficios del Grid en los proyectos del Posgrado.

Un proyecto casi necesario sería la instalación de un JMS en los laboratorios de cómputo de la Facultad de Ingeniería Eléctrica que se encuentran en el edificio B de Ciudad Universitaria, los laboratorios cuentan con 120 computadoras, las cuales están encendidas durante toda la jornada laboral, la mayoría del tiempo se encuentran inactivas. Por lo que la Facultad contaría con un Grid muy poderoso, y no solo podría usarlo investigadores o alumnos de la Facultad, sino también cualquier otro miembro de la Universidad que requiera gran capacidad de cómputo. Adicionalmente

se puede utilizar para realizar prácticas como parte de los cursos curriculares. A continuación se listan algunas de las aplicaciones posibles en que podrían ser utilizados los recursos por otras Facultades y Escuelas de la Universidad:

- **Facultad de Arquitectura:** Para acortar el tiempo del proceso de renderización<sup>1</sup> de imágenes.
- **Facultad de Biología:** Resolver algoritmos y técnicas estadísticas de información biológica que típicamente consisten en grandes secuencias de ADN, ARN, o proteínas
- **Facultad de Química:** Dichos sistemas pueden ser usados para calcular las estructuras y propiedades de moléculas o de cuerpos solidos, o simplemente ver la reacción química de lo que sucede si se mezclan ciertos componentes químicos con una sustancia. En la Industria el cómputo en Grid es ampliamente utilizado para diseñar nuevos medicamentos y materiales.
- **Escuela Popular de Bellas Artes:** Los alumnos podrían acelerar la edición y procesamiento de videos o películas.
- **Facultad de Ingeniería Mecánica:** Para la simulación de flujo de líquidos, de túneles de viento y transferencias de calor, por poner algunos ejemplos.

Como se puede observar en muchas áreas se necesita hacer una gran cantidad de cálculos por lo tanto es necesario contar con un JMS, con el fin de optimizar el uso de recursos de cómputo, así como de acelerar los tiempos de procesamiento..

---

<sup>1</sup>La renderización es un proceso de cálculo complejo desarrollado por un ordenador destinado a generar una imagen 2D a partir de una escena 3D

# Apéndice A

# Apéndice A

## A.1. Configuración del agente Xgrid

Para comenzar a configurar el agente, se debe ingresar al menú de preferencias, una vez dentro, seleccionamos la opción “Compartir Xgrid”, en esta misma ventana podremos seleccionar el método de autenticación como se muestra en la Figura A.1, existen 3 maneras de hacerlo las cuales son:

- Ninguno.- No se necesita ninguna autenticación para recibir alguna parte del trabajo enviado por el controlador.
- Single sign on Kerberos<sup>1</sup>.- Es mas segura que la opción de contraseña, esta forma de autenticación necesita una base de datos de claves secretas para cada uno de los agentes.
- Contraseña.- Se utiliza una cadena encriptada para autenticarse con el controlador y así recibir parte del trabajo.

---

<sup>1</sup>Sistema por el cual las computadoras pueden delegar la autenticación de usuarios autorizados a un servidor de confianza, por lo general esto forma parte de una configuración de LDAP o un DNS local.

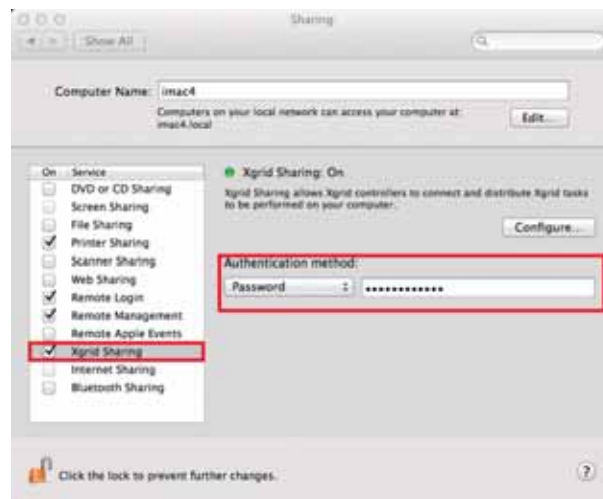


Figura A.1: Configuración de agente

A continuación se hace clic en el botón de configurar en donde se desplegará una ventana como la siguiente:



Figura A.2: Opciones de configuración

En esta ventana podremos seleccionar que controlador utilizar; ya sea el primer controlador disponible en nuestra red o un controlador específico, en la última opción debemos elegir ya sea el nombre del controlador desplegado en el menú o poner la dirección IP que tiene asignada nuestro controlador.

En esta misma ventana se podrá elegir en que circunstancia el agente puede aceptar tareas, existen dos opciones, la primera es que solo cuando el agente este inactivo, a que se refiere esto, solo aceptará tareas, si no existe ningún usuario tecleando o usando el mouse. La segunda opción es que el agente siempre aceptará tareas, no importa lo que este realizando el usuario.

## A.2. Configuración de controlador Xgrid

Para facilitar la configuración del controlador es necesario tener instalado Mac OS X Server, de lo contrario se necesitaría instalar un aplicación llamada XgridLite.

Para configurar el controlador se debe ingresar a “Server Admin” y seleccionamos la opción Xgrid del servidor como se muestra en la imagen.

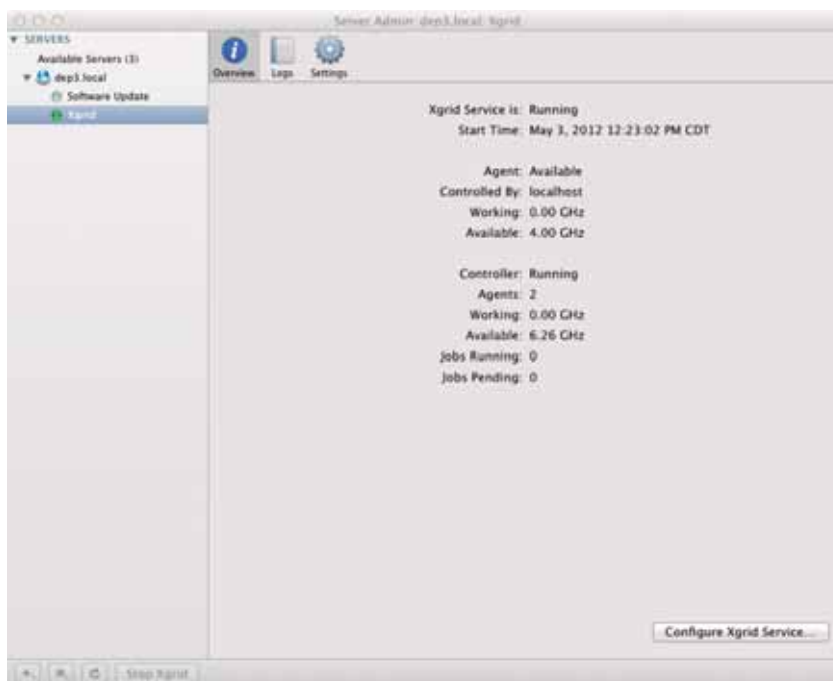


Figura A.3: Server Admin

A continuación hacemos clic en la pestaña de “Configuración” y después en la opción controlador como se observa en la Figura A.3.

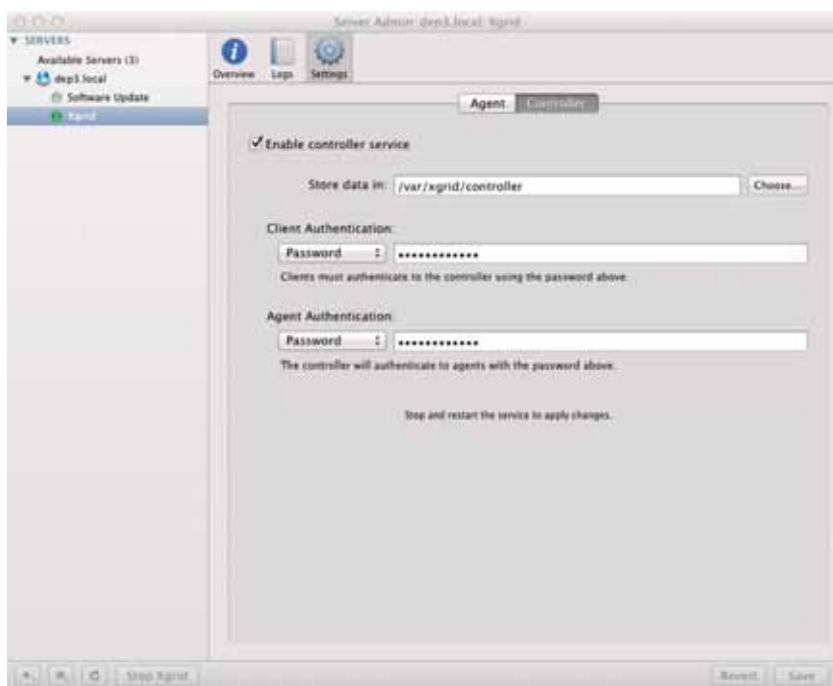


Figura A.4: Configuración de controlador

Seleccionamos la opción “Habilitar servicio de controlador”, ajustamos “autenticación del cliente” y “autenticación del agente” como contraseña. Después de llenar los campos de contraseña de autenticación se guardara al dar clic al botón de “Guardar”.

De manera siguiente hacemos clic sobre la pestaña de “Agente”. Dentro de esta pestaña seleccionamos la opción “Habilitar el servicio de agente”, y posteriormente “Uso de un controlador específico”, escribimos la dirección IP o el nombre del controlador que vamos a utilizar.

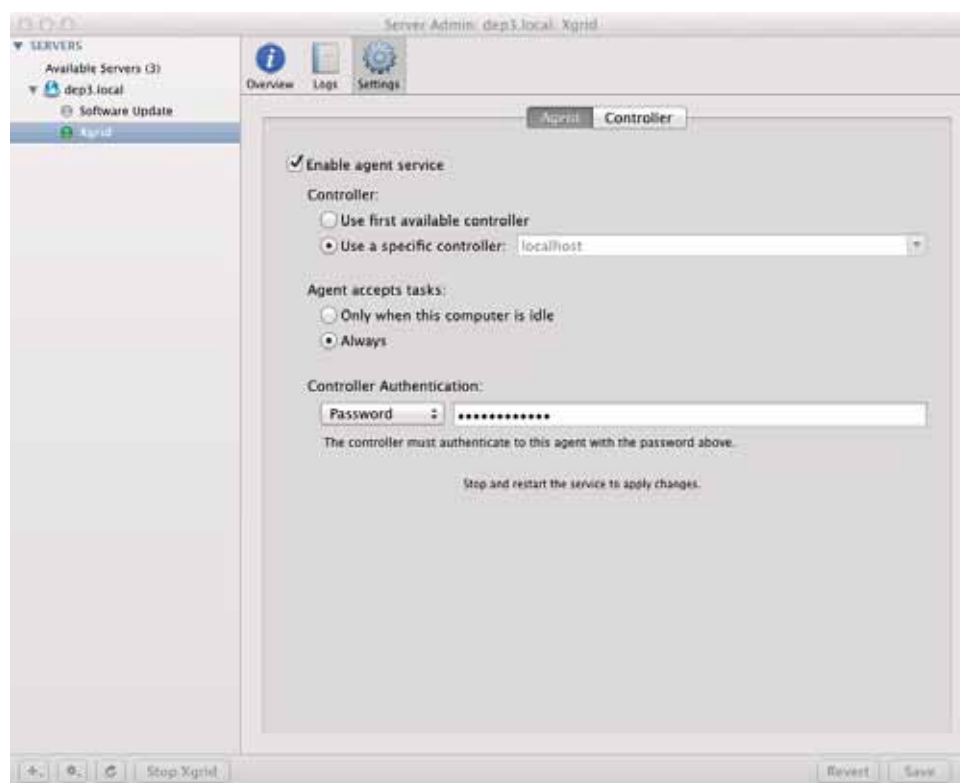


Figura A.5: Configuración del controlador como agente

Seleccionamos alguna de las opciones dadas por la sección de “Agente acepta trabajos”, Si nuestro servidor es una máquina multi-núcleo y no se utiliza para otro servicio podremos elegir la opción de “Siempre”, en su defecto, si la máquina esta a menudo ocupada prestando sus servicios elegiremos la opción “Solo cuando el equipo este inactivo”. Por ultimo establecemos una de las opciones para “la autenticación del controlador”, la opción elegida por nosotros sera con contraseña, por lo tanto introducimos la contraseña correspondiente. Hacemos clic a la opción de “Guardar” . Para terminar de configurar nuestro controlador, hacemos clic en el botón “Start Xgrid” para iniciar el servicio de Xgrid.

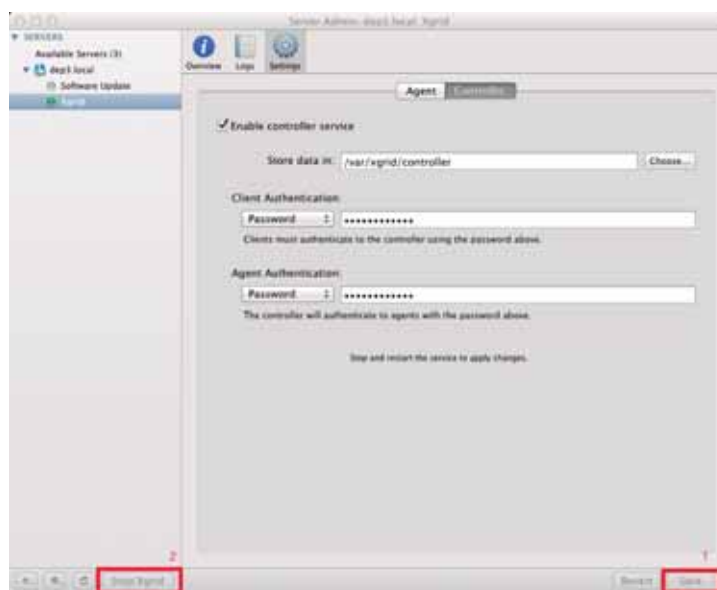


Figura A.6: Últimos pasos para la configuración del controlador

### A.3. Características de las computadoras participantes en el Grid

Nombre	Procesador	Velocidad	HDD	SO	RAM
imac	Intel Pentium 4	4 GHz	320 GB	MAC OS X 10.7.3	2 GB
imac3	Intel Pentium Core Duo	2.66 GHz	320 GB	MAC OS X 10.7.3	2 GB
imac4	Intel Pentium Core Duo	2.66 GHz	320 GB	MAC OS X 10.7.3	2 GB
dep3	Intel Pentium Core i7	2.00 GHz	500 GB	MAC OS X SERVER 10.7.3	4 GB

Tabla A.1: Características de computadoras participantes



## Apéndice B

# Apéndice B

### B.1. Código del programa “División y cuadrado”

```
#!/usr/bin/perl -w
my $sleep_duration = 5;
warn "starting...\n";
warn "sleeping $sleep_duration seconds...\n";
sleep $sleep_duration;
my $a = $ARGV[0];
my $b = $ARGV[1];
warn "calculating $a/$b...\n";
my $c=$a/$b;
my $d=$c*$c;
print "$c\n";
print "$d\n";
warn "done!\n";
exit 0;
```

## B.2. Código del programa “Método Monte-Carlo”

```
#!/usr/bin/python
import random import math import sys
N = 1000000 if len( sys.argv ) > 1: N = int( sys.argv[1] )
count_inside = 0 for count in range( N ):
d = math.hypot(random.random(), random.random())
if d < 1: count_inside += 1 count += 1
print 30 * '-' print N, ":", 4.0 * count_inside / count
```

## B.3. Script ART “imac3”

```
#!/usr/bin/perl
$h = '/bin/hostname' || "Failed";
chomp($h);
if ($h eq "imac3") { print 1; }
else { print 0;};
```

## B.4. Script ART “RAM”

```
#!/usr/bin/perl
# profiler values are in the form: ' __key__: _____value_____' (one per line)
my $memory = '/usr/sbin/system_profiler system_profiler SPHardwareDataType SPSoftware-
DataType grep Memory';
my $regex = '^\\s*Memory: (\\d*) (.*?)$';
my ( $value, $unit ) = ( $memory =~ /$regex/m );
# print RAM value in megabytes if ( $unit eq 'MB' ) { print $value }
if ( $unit eq 'GB' ) { print $value * 1024 }
```

# Bibliografía

- [Appa] Apple.Inc. *Mac OS X Server Advanced Server Administration*. 2nd edition.
- [Appb] Apple.Inc. *Mac OS X Server Open Directory Administration*. 2nd edition.
- [App12] Apple.Inc. Cocoa, 2012. Available from: <https://developer.apple.com/technologies/mac/cocoa.html>.
- [BD00] Chansup Byun and Christopher Duncan. A comparison of job management systems in supporting hpc clustertools. Indiana University, 2000.
- [Bow05] Stuart Bowness. *Xgrid and cross platform grid computing*. University College of the Fraser Valley, 2005.
- [Col10] Smith College. Xgrid tutorial part 1: Monte carlo, 2010. Available from: [http://cs.smith.edu/dftwiki/index.php/XGrid\\_Tutorial\\_Part\\_1:\\_Monte\\_Carlo](http://cs.smith.edu/dftwiki/index.php/XGrid_Tutorial_Part_1:_Monte_Carlo).
- [EGGA03] Tarek El-Ghazawi, Kris Gaj, and Nikitas Alexandridis. A performance study of job management systems. George Mason University, 2003.
- [etu12] etutorials.org. What rendezvous?, 2012. Available from: <http://etutorials.org/Mac+OS/mac+os+x+unwired/Chapter+8.+Rendezvous/8.1+What+Is+Rendezvous/>.
- [JCL11] Grupo JCL. Ventajas del servidor nfs, 2011. Available from: <http://servidornfs.blogspot.mx/2011/07/ventajas-del-servidor-nfs.html>.
- [JD11] David Johnson Davies. An xgrid primer, 2011. Available from: <http://xgrid.plasticki.com/show?7KE>.

- [Mut88] Matt Walter Mutka. *Sharing In A Privately Owned Workstation Environment*. PhD thesis, University of Wisconsin-Madison, 1988.
- [Par07] Charles Parnot. The xgrid tutorials (part i): Xgrid basics, 2007. Available from: [http://www.macresearch.org/the\\_xgrid\\_tutorials\\_part\\_i\\_xgrid\\_basics](http://www.macresearch.org/the_xgrid_tutorials_part_i_xgrid_basics).
- [Pub12] Public. Que es ldap, 2012. Available from: [http://www.linuxorientado.edu.sv/descargas/2563b6\\_OpenLDAP.pdf](http://www.linuxorientado.edu.sv/descargas/2563b6_OpenLDAP.pdf).
- [SEL] Hal Steren, Mike Eisler, and Ricardo Labiaga. *Managing NFS and NIS*. 2nd edition.
- [Tho04] Chen Thomas. *The Evolution of Viruses and Worms*. 2004.
- [UM12] UW-Madison. Condor high throughput computing, 2012. Available from: <http://research.cs.wisc.edu/condor/>.
- [Wik12] Wikipedia. Reaper (antivirus), 2012. Available from: [http://es.wikipedia.org/wiki/Reaper\\_28antivirus29](http://es.wikipedia.org/wiki/Reaper_28antivirus29).