

Universidad Católica “Nuestra Señora de la Asunción”
Facultad de Ciencias y Tecnologías
Departamento de Electrónica e Informática
<http://www.uca.edu.py>
Computación Cooperativa

José A. Gómez G.
gomezgaona@gmail.com

13 de octubre de 2015

Resumen: La Computación Cooperativa consiste en un tipo de computación distribuida en el cual se reparten las tareas a conjunto de computadoras que participan en un proyecto específico, poniendo a disposición la capacidad de procesamiento para llevar a cabo una tarea específica, lográndose así un acceso transparente a una gran capacidad de cómputo.
KeyWords: Groupware, Sharing, Mainframes, Resource brokering, *Grid Computing*

1. Introducción

La Computación Cooperativa es una forma de computación distribuida, que sirve para resolver problemas computacionales complejos que involucran analizar grandes volúmenes de datos, por lo cual se ha convertido en una herramienta muy importante que permite ahorrar tiempo en la realización de proyectos que de otra forma llevarían mucho tiempo poder llevarlos a cabo. Internet es un sistema distribuido que se basa en diversos protocolos que permiten que se lo vea como una única red lógica por lo cual los proyectos colaborativos lo utilizan como plataforma de comunicación. Los proyectos de computo colaborativo están basados en la arquitectura *Grid Computing* lo cual representa una ventaja sobre los sistemas que utilizan supercomputadores.

En los últimos cinco años se ha alcanzado umbral tecnológico en cuanto a la velocidad de procesamiento, estableciendo un tope físico en la capacidad de los procesadores comerciales [1], por lo cual para mejorar el desempeño de las computadoras, la tendencia fue distribuir las tareas a varios procesadores logrando mejorar el desempeño y eficiencia energética sin tener que aumentar la velocidad del reloj. Esto indica que los procesadores que poseen nuestros dispositivos cuentan con una capacidad de cómputo que a veces no es aprovechado en un 100% [1], por lo cual el tiempo ocioso que poseen los procesadores actuales, puede ser aprovechado debido a que no siempre se encuentran realizando una tarea útil o no utilizan toda su capacidad. Con estos principios se lograron implementar proyectos científicos muy interesantes que consiguieron reducir considerablemente el tiempo de procesamiento el cual sería muy costoso implementarlos con supercomputadores.

No es reciente la iniciativa de compartir recursos que se encuentran distribuidos geográficamente. Muchos de los principios de diseño de esta nueva forma de computación se han ido desarrollando de distintas maneras a lo largo de la historia. Hoy en día, se pueden compartir recursos (*sharing* a diferencia de las décadas de los sesenta y setenta cuando el poder de procesamiento compartido era inviable por la poca capacidad de memoria, rapidez de procesamiento y que Internet no existía. En ese entonces, la computación se caracterizaba por estar dependiendo de grandes *mainframes* (Computadora Central) los cuales debían ser compartidos por las organizaciones por lo que era muy costoso que cada empresa o instituto tuviera uno.

En los comienzos del nuevo milenio los avances obtenidos en *Grid Computing* estuvieron encaminados en el desarrollo de protocolos, servicios y herramientas en la capa de aplicación, que sirvan de plataforma para la nueva infraestructura y tenga la posibilidad de ser escalable [2]. Estas tecnologías presentan soluciones confiables y pueden manejar de manera dinámica protocolos de administración de recursos y servicios que soportan el acceso remoto para procesar datos así como los recursos de datos, lo cual otorga las herramientas necesarias para poder realizar los proyectos de Computación Cooperativa.

2. Computación Grid

Para poder tener una mejor perspectiva es necesario entender la que es la Computación Grid, debido a que sobre este sistema de computación dis-

tribuida se basan los proyectos de Computación Cooperativa.

¿Por qué Grid? Existen varias maneras de solucionar los problemas que demanden de gran poder de cómputo, como pueden ser a través de supercomputadores que son caros y son especializadas para un tipo de trabajo, para los proyectos de Computación Cooperativa no se utiliza debido al costo y que el experimento en cuestión debe estar localizado. Asimismo resulta más complicado **alcanzar el poder de cómputo requerido de manera local** por lo que es más eficiente utilizar Grid Computing debido a que se dispone de mucho poder de cómputo remotamente. Además de tener un ahorro en costos y de tener el poder de un supercomputador, grid permite la Computación Distribuida sobre una red de recursos heterogéneos utilizando estándares abiertos [3]. Debido a la naturaleza de los proyectos de computación cooperativa Grid Computing provee una solución óptima debido a que se ajusta a las exigencias de cómputo requerido por dichos proyectos, si el proyecto en cuestión es sensible al desequilibrio en la rapidez de procesamiento de datos, Grid Computing no brinda la seguridad necesaria debido a que los recursos son heterogéneos y se deberá utilizar otro sistema más adecuado y totalmente diferente a lo que es Grid como la red de la NASA *Deep Space Network* que se encarga de recoger los datos de las diversas misiones espaciales actuales que están constantemente enviando datos a la tierra, por lo cual, la naturaleza del proyecto no permite usar Grid Computing como lo hace el proyecto SETI@home [4].

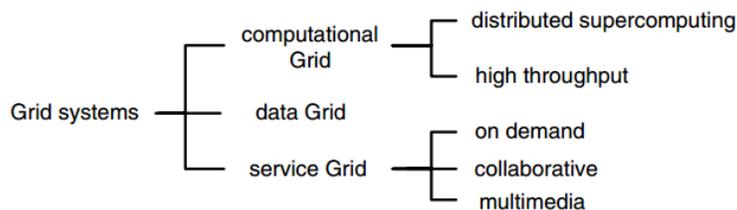


Figura 1: Grid Taxonomy [5]

2.1. Componentes de Grid [6]

Grid establece un nuevo paradigma que consiste en que se puedan compartir los recursos de manera coordinada y contribuir a una solución al problema en organizaciones virtuales multi-institucionales.

Sharing: El *Sharing* es la definición de como se intercambian los datos, así como también el acceso directo a procesadores, programas, datos y otros recursos como sean requeridos en el desarrollo de la solución y estrategias de *resourcebrokering* que surgen de la industria, la ciencia y la ingeniería. El sharing debe realizarse a través de controles a proveedores de recursos y consumidores definiendo de forma clara qué recursos se comparten, quién está habilitado a compartir y las condiciones bajo las cuales se efectuará..

Servicios: Primeramente definimos qué es una Organización Virtual (OV) la cual consiste en el conjunto de individuos o instituciones definidas por las reglas de sharing. Un servicio se entiende por el protocolo con el que habla y el comportamiento que implementa. La existencia de servicio estándar (de acceso a computación, datos, descubrimiento de recursos, replicación de datos y más) es muy importante porque permite mejorar los servicios ofrecidos a los colaboradores de las OV y también abstraer de detalles específicos de los recursos que de otra forma deberían ser posteriores al desarrollo de aplicaciones para OV. Relación con API's y los SDK's Los desarrolladores deben crear aplicaciones sofisticadas en complejidad y entornos de ejecución dinámica. La robustez de las aplicaciones, su integridad, costos de desarrollo y de mantenimiento son de gran importancia en este punto. Las abstracciones estándares, API's (*Application Programming Interface*), y los SDK's (*Software Development Kit*) pueden acelerar el desarrollo de código, permitiendo el sharing de código, y mejorar la portabilidad de aplicaciones [7].

2.2. Arquitectura Grid

La arquitectura Grid consiste en capas parecidas al modelo OSI, donde cada una proveerá funciones específicas. Las capas más altas de esta estructura se enfocan y acercan más a la relación directa con el usuario, mientras que las de nivel más bajo se aproximan al hardware, es decir a las computadoras y redes. [7]

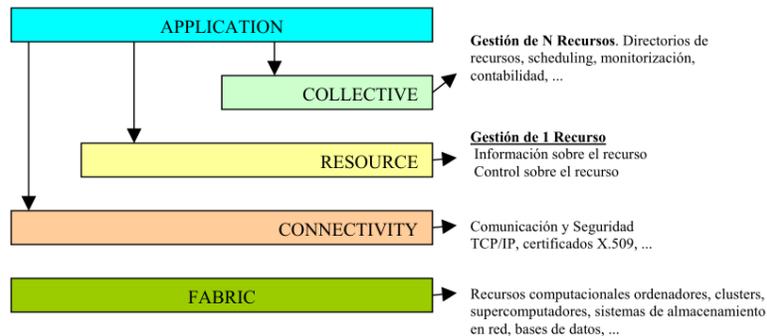


Figura 2: Arquitectura Grid [7]

Fabric: Provee el hardware necesario en donde el acceso compartido es administrado por protocolos. Un recurso puede ser una entidad lógica, un sistema de archivo distribuido, un cluster (Conglomerado de Computadoras) o un pool (Agrupamiento de Conexiones) de computadoras distribuidas, en esos casos la implementación del recurso puede involucrar protocolos internos. La funcionalidad de esta capa permite el sharing de operaciones sofisticadas.

Capa de Conectividad: La capa conectividad es el centro de las comunicaciones y protocolos de autenticación requeridos para las transacciones en la red Grid. Los Recursos y los protocolos de conectividad mantienen todas las transacciones específicas de Grid entre diferentes computadoras y otros recursos. La red empleada por GRID es Internet, la misma red usada por la web y muchos otros servicios como el correo electrónico. La diferencia es que las tecnologías de Internet direccionan el intercambio de comunicaciones e información entre computadoras pero no provee aproximaciones integradas para el uso coordinado de recursos de diversos sitios de computación. Los protocolos de comunicación proveen el intercambio de datos entre los recursos de la capa Fabric. Los protocolos de autenticación construyen la comunicación de servicios para proveer mecanismos seguros para la verificación e identificación de usuarios y recursos. Las comunicaciones requieren transporte, ruteo, y servicios existentes como TCP, DNS, IP y ICMP. Las soluciones de autenticación para algunos entornos pueden tener características como el tener que iniciar sesión de manera a proveer seguridad. Las soluciones de Grid de seguridad deben ser capaces de interoperar con varias soluciones locales. Además,

Usuarios basados en las relaciones confiables: en el cual el sistema de seguridad no debe requerir que cada proveedor de recursos coopere o interactúe con cada una de las configuraciones de los entornos de seguridad [7].

Capa de Recursos: La capa de Recursos se construye sobre la capa de Conectividad, se definen los protocolos de Comunicación y Autenticación para definir negociaciones seguras, iniciación, monitoreo, control, contabilidad y pagos de operaciones compartidas sobre recursos individuales. Las implementaciones de estos protocolos llaman a funciones de capa Fabric para acceder y controlar recursos locales. Los protocolos de esta capa se refieren a recursos individuales y por ende ignoran resultados de estado global y acciones atómicas a lo largo de colecciones distribuidas, éstos son de consideración para la capa colección. Existen dos tipos de protocolos en esta capa.

Capa Colectiva: Mientras la capa recurso se enfoca en interacciones de recursos simples, la próxima capa en la arquitectura contiene protocolos y servicios – API's Y SDK's, que no están asociadas con un recurso específico sino que es global y captura las interacciones a través de colecciones de recursos. Por esta razón, se la denomina capa colectiva. Dado que los componentes de capa colectiva construyen sobre capa Recursos y capa Conectividad pueden implementar una amplia variedad de comportamientos para el sharing sin la localización de nuevos requerimientos sobre los recursos almacenados. Algunas primitivas de servicio son las siguientes.

Servicio de Directorio: Permite a participantes de las OV descubrir la existencia y propiedades de los recursos de OV. Permite a sus usuarios hacer consultas para recursos por el nombre o por atributos, como ser tipo, disponibilidad o carga.

Co-Allocation Scheduling y Servicios Brokering: Dá la posibilidad a los recursos de los participantes de las OV averiguar sobre más recursos para propósitos específicos y para la programación de tareas sobre los recursos apropiados.

Monitoreo y Diagnóstico de Servicios: Soporta el monitoreo de los recursos de las OV para fallas, ataques de adversarios, detección de intrusos y sobrecargas.

Servicios de Replicación de Datos: Sostiene el manejo de almacenamiento de los recursos pertenecientes a las OV para maximizar el rendimiento en los accesos a datos con las respectivas métricas como el tiempo de respuesta o costo entre otros parámetros de medición.

Sistemas de Programación *Grid-Enabled*: Posibilita usar en los entornos GRID modelos de programación familiar usando varios servicios Grid para localizar recursos y brindar seguridad.

Sistemas de Manejo de Workload y Collaboration Frameworks: Se conocen como entornos de solución de problemas “PSE’s”) provee para la descripción, uso y manejo de multi steps, asíncronos, flujos de trabajo multi-componentes.

Sistemas de Ubicación de Software: Descubre y selecciona las mejores implementaciones de software y ejecución de plataformas basadas en parámetros del problema a resolver como por ejemplo son NetSolve y Ninf.

Servidor de Autorización de la Comunidad: Refuerza el gobierno de las políticas de acceso a recursos generando capacidades que los miembros de la comunidad pueden usar para el acceso a recursos comunitarios.

Servicios de Cuentas de la Comunidad y Pagos: Reune información acerca del uso de datos para el propósito de manejo de cuentas, pagos, y limitaciones en el uso de recursos a miembros de la comunidad.

Servicios de colaboración: Brinda soporte al intercambio coordinado de información dentro de comunidades de usuarios potencialmente grandes. Mientras los protocolos de capa Recursos deben ser generales en naturaleza y desarrollados ampliamente, los protocolos de capa Colectiva expanden su espectro desde propósitos generales a aplicaciones altas o dominios específicos sólo entre OV específicas. Las funciones de esta capa pueden implementarse como servicios persistentes con protocolos asociados o SDK’s designadas

para enlazarse con ciertas aplicaciones. En ambos casos sus implementaciones pueden construir en capa Recurso protocolos y API's. Los componentes de capa Collective pueden crearse para requerimientos de usuarios de comunidades específicas, OV, o dominios de aplicaciones como por ejemplo un SDK que implementa protocolos de coherencia de aplicaciones específicas, o un servicio co-reservado para un conjunto específico de recursos en la red. Otros componentes de esta capa pueden ser de propósitos más generales por ejemplo, servicios de replicación que manejen una colección internacional de sistemas de almacenamiento para múltiples comunidades o un servicio directorio designado a permitir el descubrimiento de OV's. Los servicios colectivos (collective services) se basan en protocolos: protocolos de información que obtienen datos sobre la estructura y estado de los recursos, y protocolos de manejo que negocian el acceso a recursos de una forma uniforme.

Capa de Aplicación: La capa final en la arquitectura Grid comprende aplicaciones de usuario que operan entre los entornos de las OV. Estas aplicaciones se construyen en términos de servicios definidos para cada capa. Cada una de estas capas tiene protocolos bien definidos que proveen acceso al uso de servicios: manejo de recursos, acceso a datos, y más. Si se considera que una aplicación de usuario necesita analizar datos contenidos en archivos independientes, tendrá que realizar entonces las siguientes tareas básicas:

- Obtener la credencial necesaria de autenticación para abrir los archivos (Recursos y Protocolos de Conectividad).
- Consultar el sistema de información y réplica de catálogos para determinar dónde pueden encontrarse las copias de los archivos en GRID así como también dónde se hallan los recursos más convenientes para hacer el análisis de datos (Collective Services).
- Pasar los pedidos a Fabric, la computadora apropiada, sistema de almacenamiento y redes, para extraer los datos, iniciar los procesos, y proveer los resultados (Recursos y Protocolos de Conectividad).
- Monitorear el progreso de varios procesos y transferencia de datos, notificando al usuario cuándo el análisis se completa y también detectando y respondiendo ante situaciones de fallas (Collective Services).

Hoy día hay diversos tipos de redes disponibles las cuales se caracterizan por su tamaño (Locales, Nacionales e Internacionales) y según su rendimiento en

términos de throughput que es la cantidad de datos transferidos desde un lugar a otro en un tiempo determinado. Típicamente el throughput se mide en Kbps, Mbps or Gbps. Grid está construida sobre redes de alto rendimiento tal es el caso de la red Intra-Europe GEANT o la red de UK Super Janet la cual exhibe 10 Gbps de rendimiento en el backbone de red (backbone: se utiliza para nombrar a aquellos enlaces de alta velocidad en las redes que unen grandes nodos). A medida que crece la velocidad de la red, el poder de GRID es determinado por el rendimiento de los recursos de computación disponibles en los nodos de la red. Los nodos mayores serán recursos de alto rendimiento como un cluster largo de computadoras o cualquier supercomputadora dedicada.

2.3. ¿Cómo Funcionan los Proyectos que Utilizan Grid Computing?

Un equipo central, denominado planificador(scheduler), distribuye un proceso entre pares dispersos geográficamente conectados por una red de datos que puede ser internet, donde tales pares pueden ser computadoras de escritorio o equipos de alto rendimiento (clusters, supercomputadoras, mainframes) especialmente dedicados al cómputo intensivo o no. El sistema toma las capacidades disponibles de todos los equipos vinculados a la infraestructura. Cuando no están siendo utilizados plenamente por el usuario reciben tareas del equipo planificador. Todos los recursos disponibles en la red son aprovechados, **independientemente de su ubicación geográfica, su arquitectura y sistema operativo**. Un punto fuerte de una red grid es su espina dorsal implementada como un canal alta velocidad y baja latencia. Cuando un usuario utiliza sólo una parte de la capacidad de su CPU, el resto se aprovecha. Por problemas derivados de fallas o de uso pleno de una computadora por parte de su usuario, la tarea se reasigna a otra máquina disponible. A la aplicación se la accede desde cualquier equipo. A través de una interface simple se pueden agregar más computadoras al sistema. La conexión directa entre las computadoras (P2P) evita la sobrecarga del equipo central [8].

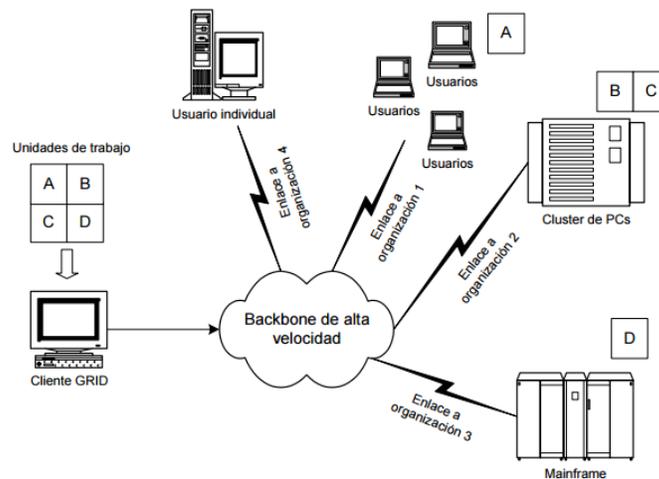


Figura 3: Arquitectura Grid [7]

2.4. Ventajas y Desventajas del Modelo Grid

Grid computing está creado con el fin de brindar una solución a determinadas cuestiones, como problemas que requieren de un gran número de ciclos de procesamiento o acceso a una gran cantidad de datos.

Ventajas

- Encontrar un hardware y un software que permitan brindar estas utilidades comúnmente proporciona inconvenientes de **costos**, **seguridad** por lo cual Grid presenta la ventaja de encontrar muchos recursos disponibles.
- Como se integran diferentes tipos de máquinas y de recursos compartidos por los usuarios, una red grid **nunca queda obsoleta** porque todos los recursos se van renovando.
- Brinda a las empresas el beneficio de la velocidad, lo que supone una ventaja competitiva, con lo cual se provee una mejora de los tiempos para la producción de nuevos productos y servicios.

- Facilita la posibilidad de compartir, acceder y gestionar información, mediante la colaboración y la flexibilidad operacional, aunando no sólo recursos tecnológicos dispares, sino también personas y aptitudes diversas.
- Con respecto a la seguridad en la grid, ésta está sustentada con las “intergrids”, donde esa seguridad es la misma que ofrece la red sobre la cual se utiliza tecnología grid.
- Si tenemos disponibilidad de un conjunto de máquinas heterogéneas de pequeño o mediano porte, cuya potencia computacional sumada sea considerable, eso permitiría generar sistemas distribuidos de muy bajo costo y gran potencia. computacional.

Desventajas

- El paralelismo puede estar visto como un problema, ya que el desarrollo de aplicaciones para computo paralelo presenta ciertas complicaciones.
- Como Grid Computing trabaja sobre la plataforma de Internet es susceptible a cualquier fallo que haya en la red global.
- La falta de estándares dificulta mucho el desarrollo del software necesario para su implementación.
- Grid Computing debe ser capaz de poder administrar cualquier tipo de recurso que maneje el sistema, si no resultará totalmente inútil, lo cual se complica al ser un sistema muy heterogéneo por lo cual la comunicación puede ser lenta y no uniforme.
- El descubrimiento, selección, reserva, asignación, gestión y monitorización de recursos son procesos que deben controlarse externamente y que influyen en el funcionamiento del grid.
- Se depende de la velocidad de conexión que dispone cada participante.

3. Proyectos de Computación Cooperativa

La plataforma más utilizada para los proyectos de Computación cooperativa es BONIC (*Berkeley Open Infrastructure for Network Computing*) aunque también existen otras que pertenecen a algunos proyectos y actualmente se están desarrollando Add-ons o Complementos para los navegadores. El cual es un software libre para la computación distribuida. Originalmente fue desarrollado para apoyar el proyecto SETI@home, pero ahora es utilizado como plataforma para otras aplicaciones distribuidas en áreas tan diversas como las matemáticas, la medicina, la biología molecular, la climatología y la astrofísica. El objetivo principal de este programa es hacer posible que los investigadores aprovechen la enorme potencia de procesamiento de las computadoras personales de todo el mundo. [9]. Actualmente dispone de unos 13.902.383 ordenadores activos en todo el mundo y con un rendimiento medio de 126.041.718 TeraFLOPS al 10 de octubre de 2015. [10]. Actualmente la plataforma está disponible para Windows, MAC OS X, Linux y Android. Algunos proyectos que utilizan BONIC son:

3.1. SETI@home

Es un proyecto científico cuya meta es detectar vida inteligente fuera de la Tierra. Un enfoque, conocido como radio SETI, usa radio telescopios para escuchar señales de radio de banda estrecha provenientes del espacio. No se tiene conocimiento de que estas señales ocurran de manera natural, por lo que su detección puede proporcionar evidencias de tecnología extraterrestre. [11]

Consiste en enviar señales que solamente puedan ser respondidas por una especie de vida que haya desarrollado los esquemas de modulación de señales parecidos al que el emitimos por radio en la tierra por lo cual el trabajo requiere de la filtración del ruido celestial producidos por los cuerpos que componen el universo, por lo cual el trabajo computacional que demanda es alto. [12]

Anteriores proyectos de radio SETI usaron ordenadores de propósito especial localizados en el mismo telescopio para realizar el grueso del análisis de datos. Hasta que el Congreso de los Estados Unidos le retiró el financiamiento [13]. En 1995, David Gedye se propuso convertir radio SETI un super

ordenador virtual compuesto de un gran número de ordenadores conectados a través de Internet y organizó el proyecto SETI@home para explorar esta idea. SETI@home fué lanzado originariamente en mayo de 1999 y actualmente funciona sobre la plataforma BONIC. [14]

3.2. GPUGRID.net

Es un proyecto de computación distribuida organizado por la Universidad Pompeu Fabra(España) y se desarrolla en BOINC. El proyecto consiste en realizar simulaciones de biología molecular a átomo-entero, y está diseñado para correr en GPUs Nvidia. El soporte para el procesador Cell del PlayStation 3 y el proyecto subsiguiente PS3GRID fue abandonado en 2009 debido a una actualización de firmware del PS3 que previene la instalación de programas terceros que son requeridos. Esto incluye distribuciones Linux que son requeridos para correr BOINC. El inmenso rendimiento de las GPUs Nvidia también ha echo al cliente PS3 prácticamente redundante. A partir de septiembre de 2009, una GPU Nvidia de rango medio corre las aplicaciones de GPUGRID aproximadamente cinco veces más rápido que el microprocesador Cell.

3.3. World Comunity Grid

Constituyen varios proyectos en el área de salud, que son impulsados por IBM, creando así la comunidad Grid mas grande del mundo. Fue lanzado en 2004 y se desarrolla sobre la plataforma BONIC, teniendo una capacidad de 448.063 Teraflops con unos 60.500 participantes activos a julio 2015. Algunos proyectos son:

FightAIDS@Home: *'Pelea Contra el SIDA en Casa'* consiste en el calculo de la evolución de la dinámica del VIH.El objetivo del proyecto es proteasa, que es la enzima que rompe los enlaces de proteínas y mediante el conocimiento de la maduracion puede ser bloqueada y así detener su crecimiento. Cada maquina simula una posible droga y verifica si es efectiva en detener el crecimiento del virus para así evitar su replicación. En Octubre de 2015 la fase 2 del proyecto fue presentado para evaluar a fondo los resultados obtenidos en la fase 1 del proyecto y establecer un ranking de las posibles curas.

The Clean Energy Project: El proyecto es apoyado por el Departamento de Química y Química Biológica de la Universidad de Harvard. El objetivo del proyecto es encontrar nuevos materiales para la siguiente generación de paneles solares y luego dispositivos de almacenamiento de energía. Los investigadores emplean el cálculo de estructuras de mecanismos moleculares y electrónicos para predecir las propiedades de óptica y transportes de moléculas que vendrán en la próxima generación de materiales para celdas solares. La fase 1 fue lanzada en 2008 y completada en 2009. Gracias al poder de cómputo se obtuvieron datos que los investigadores disponen para calcular las propiedades electrónicas de decenas de miles de materiales orgánicos, muchos más de los cuales pudieron haber testado en el laboratorio y gracias a eso determinaron cuáles son los candidatos más prometedores para el desarrollo de tecnología solar más favorable.

Mapping Cancer Markers: El objetivo del proyecto es identificar los marcadores asociados con varios tipos de cáncer, analizando millones de muestras de tejidos de pacientes con cáncer en todo el mundo, las muestras incluyen pacientes con cáncer en el pulmón, ovario, próstata, páncrea y de mama. Comparando estos datos, los investigadores quieren identificar patrones de marcas para diferentes tipos de cáncer y relacionarlos con distintos resultados, incluyendo la respuesta a varias opciones de tratamiento.

3.4. Folding@home:

Es un proyecto desarrollado por la Universidad de Stanford y consiste en realizar simulaciones de los plegamientos proteicos de enfermedades y otras dinámicas moleculares. Su objetivo es conocer cómo las proteínas llegan a su estructura final lo cual es de gran relevancia para el estudio de enfermedades y en consecuencia el diseño de nuevos medicamentos. Folding@home es el proyecto más grande de computación distribuida en el mundo reconocido por el *Guinness World Of Records* superando a SETI@home. Las simulaciones precisas de cómo se pliegan las proteínas permiten a la comunidad científica comprender mejor el desarrollo de muchas enfermedades, como el Alzheimer, la fibrosis quística, la enfermedad de las vacas locas o el cáncer. Hasta el momento, el proyecto folding@home ha tenido éxito simulando el plegamiento en un rango de 5-10 microsegundos, una escala de tiempo miles de veces más grande de lo que había sido posible anteriormente. [15]

3.5. Otros Proyectos

Proyecto	Categoría	Descripción
Einstein@Home	Astrofísica	Detecta fuentes de ondas gravitacionales.
DENIS@home	Medicina	Procesa la simulación de latidos electropsicológicos.
Enigma@home	Criptografía	Pretende decodificar tres mensajes que la máquina Enigma codificó en 1942.
Gerasim@home	Matemática	Investigación en matemática discreta y control lógico.
LHC@home	Física	Prueba el Gran Colisionador de Hadrones y sus sensores.
QMC@home	Química	Estudia la estructura y reacción de moléculas usando química cuántica y análisis Monte Carlo
Rosetta@home	Biología Molecular	Predice la estructura de proteínas para el estudio de enfermedades
AndrOINC	Criptografía	Rompe la clave de Motorola basada en RSA 1024-bits usado para firmar las particiones de inicio y recovery en los smartphones motorolas
AQUA@home	Computación Cuántica	Usa el análisis Monte Carlo para predecir el desempeño de computadoras cuánticas basadas en superconductores adiabáticos en diversos problemas.
DNA@home	Biología Molecular	Descubrió que regula los genes en el ADN usando algoritmos estáticos
SZTAKI Desktop Grid	Matemática	Busca plagios de ideas interidiomáticas en wikipedia

4. Razones para Participar de un Proyecto

- Aportar a un proyecto científico.
 - Aportar al avance en determinados campos científicos.
 - Los proyectos que luchan contra enfermedades pueden ser de gran ayuda para encontrar la cura.
- Poner a prueba la capacidad de la computadora.
 - Utiliza al máximo la CPU o GPU de tal manera que se puede comprobar su desempeño y con esta información los desarrolladores pueden mejorar el diseño de las mismas debido a que todo el tiempo esta siendo testeados.
- Reconocimiento y beneficios.
 - Algunos proyectos de astronomía tienen pensado dar la posibilidad a los usuarios nombrar los planetas que descubran.
 - Un proyecto puede ser propuesto por una persona o institución y de esa manera a tener la potencialidad de cómputo que brinda el modelo.

5. Conclusión

Esta forma de computación distribuida, establece una nuevo modelo de procesamiento de datos, altamente versátil, escalable y que permite combinar la potencia de muchos dispositivos para tener una capacidad de cómputo prácticamente ilimitada. Sus principales desventajas surgen de la dificultad para sincronizar los procesos de todos los equipos, la monitorización de recursos, asignación de cargas de trabajo y establecimiento de políticas de seguridad informática fiables.

Las plataformas se encuentran siempre en un proceso de desarrollo y renovación constante. Actualmente mucho proyectos se desarrollan gracias a que los equipos poseen la interfaz lo necesariamente avanzada para este tipo de computación y que con el tiempo va mejorando a la par que Internet ofrezca mejoras a la conectividad entre equipos, logrando así mayor robustez.

Los proyectos que se desarrollan de forma cooperativa establecen un nuevo paradigma de computación debido a que disponen de acceso transparente a gran potencia de procesamiento en comparación a Internet que posee acceso transparente a gran cantidad de información, por lo cual se ahorra en la solución de problemas computacionales en lo que relación al tiempo de procesamiento, debido a que se logra mejor rendimiento en ciertos proyectos haciéndolos de manera cooperativa que con un número limitado de supercomputadores, lo cual también tiene sus desventajas si es que los cálculos que se realizan en diversos equipos dependen entre sí, pero es una solución más económica, versátil y escalable que la implementación con supercomputadores.

Referencias

- [1] D. A. Patterson and J. L. Henessey, *Computer Organization and Design*. Elsevier, 2005.
- [2] “Screen Savers of the World Unite!” <http://cacs.usc.edu/education/cs653/Shirts-GridAD-Science00.pdf>. Accessed: 2015-09-26.
- [3] “Grid Computing .” <http://exa.unne.edu.ar/informatica/S0/Gridmonogcarol.pdf>. Accessed: 2015-09-27.
- [4] “Deep Space Network.” https://en.wikipedia.org/wiki/NASA_Deep_Space_Network. Accessed: 2015-10-13.
- [5] “A Taxonomy and Survey of Grid Resource Management Systems.” <http://www.cloudbus.org/papers/gridtaxonomy.pdf>. Accessed: 2015-09-27.
- [6] “REALIDAD DE LA COMPUTACION GRID EN AMERICA LATINA .” http://www.sociedadelainformacion.com/12/Paper_Grid.pdf. Accessed: 2015-09-23.
- [7] “GRID COMPUTING.” http://www.tyr.unlu.edu.ar/tyr/TYR-trab/2004/computacion_grid-luz-otros.pdf. Accessed: 2015-09-27.

- [8] “Diseño de una Plataforma de Computación Distribuida Cooperativa, Utilizando Servicios de una Red Compañero a Compañero.” http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes_de_Datos/Tesis/Bordignon.pdf. Accessed: 2015-09-04.
- [9] “BOINC o cómo donar recursos de tu computadora a proyectos de investigación.” <http://blog.desdelinux.net/boinc-computacion-distribuida>. Accessed: 2015-10-04.
- [10] “Bonic Stats.” <http://boincstats.com/es>. Accessed: 2015-10-04.
- [11] “Acerca de SETI@home.” http://setiathome.berkeley.edu/sah_about.php. Accessed: 2015-10-04.
- [12] “AstroSeti.” <http://astroseti.org/instituto-seti/historia-de-seti>.
- [13] “National Aeronautics and Space Administration NASA History Program Office.” <http://history.nasa.gov/seti.html>. Accessed: 2015-09-28.
- [14] “SETI@home project description.” <https://es.wikipedia.org/wiki/SETI@home>. Accessed: 2015-10-2.
- [15] “Folding@home.” <https://es.wikipedia.org/wiki/Folding@home>. Accessed: 2015-10-04.