

Legión - Sistema de Computación en Grid

Legion - Grid Computing System

Genghis Ríos Kruger

*Pontificia Universidad Católica del Perú, Dirección de Informática Académica
grios@pucp.edu.pe*

Martín Iberico Hidalgo

*Pontificia Universidad Católica del Perú, Dirección de Informática Académica
martin.iberico@pucp.edu.pe*

Oscar Díaz Barriga

*Pontificia Universidad Católica del Perú, Dirección de Informática Académica
diaz.oa@pucp.edu.pe*

Abstract

It is not often possible to acquire a large computing infrastructure dedicated to the execution of computationally intensive applications within university environments in developing countries due to their high cost. Drawing on the resources available in the computer labs of universities themselves, the problem can be solved by implementing a Desktop Grid Computing.

This document shows the Legion System, which allows you to manage multiple projects using computer-intensive Desktop Grid Computing implementation raised by the BOINC infrastructure. With the solution implemented in Legion, researches in different areas will have access to broad existent computational resources inside university campuses, through a web user interface, hiding the complexity of the grid computing system to the end user.

1. Introducción

La computación en grid tiene sus orígenes a principios de los años 80's, donde la investigación intensiva en el campo de la algoritmia, la programación y arquitecturas que soportan paralelismo, provocaron que los primeros éxitos en este campo ayuden a otros procesos de investigación en diversas áreas. A esto último se agrega el exitoso uso de la computación grid en la ingeniería y en el campo empresarial [1], permitiendo la ejecución de procesos en múltiples equipos, pudiendo tener cada uno de ellos características heterogéneas, sin la necesidad de estar ubicados en ambientes dedicados.

Durante los últimos años, se han enfocado los esfuerzos en la investigación y desarrollo de protocolos, servicios y herramientas que facilitan la implementación de sistemas en grid escalables [9]. Entre las principales

implementaciones tenemos a Condor y BOINC, como sistemas de gestión de trabajos, y a Globus Toolkit, como un conjunto de módulos independientes [10] elaborados para el desarrollo de servicios orientados a las aplicaciones de computación distribuida [11].

La infraestructura BOINC [2], Berkeley Open Infrastructure for Network Computing por sus siglas en inglés, surge con la finalidad de aprovechar la donación voluntaria de recursos de computadores de escritorio distribuidos alrededor del mundo, donde los usuarios brindan un porcentaje de dichos recursos a investigaciones científicas de todo tipo. Además, BOINC plantea su uso como una Desktop Grid, donde se realiza el aprovechamiento de recursos computacionales localizados en ambientes supervisados o controlados, dentro de una organización en particular.

Conforme pasa el tiempo, se tiene una mayor cantidad de aplicaciones computacionalmente intensivas, por lo que en muchos casos se hace necesario de una infraestructura de hardware muy costosa. Una posible solución se encuentra en el uso de Desktop Grid Computing, que permite el aprovechamiento de recursos disponibles en ambientes controlados locales o cercanos, siendo para nuestro estudio, computadores ubicados en los laboratorios dentro de un campus universitario.

El objetivo principal del trabajo realizado es canalizar esta capacidad de procesamiento hacia proyectos de investigación computacionalmente intensivos, realizando un manejo eficiente de la infraestructura existente dentro de laboratorios informáticos, y motivando a investigadores a realizar estudios más ambiciosos.

2. La infraestructura BOINC

BOINC es una plataforma de código abierto que permite a diversos proyectos hacer uso de capacidad computacional ociosa la cual proveniente de computadores ubicados alrededor del mundo de manera voluntaria. Una computadora ejecutando el cliente BOINC puede realizar peticiones al servidor BOINC en busca de tareas disponibles y, si las encuentra, empieza a trabajar para el proyecto al cual se ha unido procesando los datos asociados.

El cliente BOINC puede ser limitado en la cantidad de recursos computacionales que puede utilizar, priorizando las necesidades del usuario que utiliza el computador. En el presente caso los usuarios de los laboratorios no deben percibir cambios en el rendimiento del equipo. La flexibilidad y control que permite el cliente BOINC sobre los recursos donados hacen que la infraestructura BOINC sea adecuada para implementar la solución planteada.

Con BOINC el proceso general se divide en tareas pequeñas llamada workunits. Tales tareas deben ser independientes unas de las otras en cuanto a secuencia de procesamiento,

lo que se conoce como “paralelismo perfecto” [3].

Entre las características más importantes de BOINC, relacionadas con la implementación de la solución en este proyecto, tenemos:

Autonomía de Proyectos: Varios proyectos independientes pueden definirse en el servidor BOINC.

Administración de Recursos: Cada uno de los clientes BOINC puede participar en múltiples proyectos, permitiendo configurar la cantidad de recursos asignados a cada uno de éstos: cpu, ram, espacio en disco, velocidad de red.

Disponibilidad de Recursos: El cliente BOINC se pueden configurar para disponer permanentemente de la capacidad de procesamiento de la computadora anfitriona, esté o no siendo utilizada por sus usuarios habituales.

Multiplataforma: Disponibilidad del cliente BOINC para múltiples plataformas: Windows, Linux y Mac OS X.

3. La Arquitectura Legión

A diferencia de la implementación tradicional de BOINC, utilizada como plataforma de encolamiento y distribución de workunits, el Sistema Legión es un desarrollo propio destinado a la administración a alto nivel de usuarios y proyectos, automatizando los procesos de generación y recopilación de tareas, así como el control de computadores cliente. La arquitectura está formada por:

- La plataforma BOINC, como aplicación base.
- Base de datos en MySQL para la administración de usuarios, computadoras, proyectos y tareas.
- Servidor de archivos web en Apache para almacenamiento de los resultados de cada tarea.
- Aplicaciones en PHP para la interfaz web de usuario.

- Aplicaciones en C++ para la generación de workunits y resumen de los resultados.
- Herramientas en java para el mantenimiento y administración de los computadores.

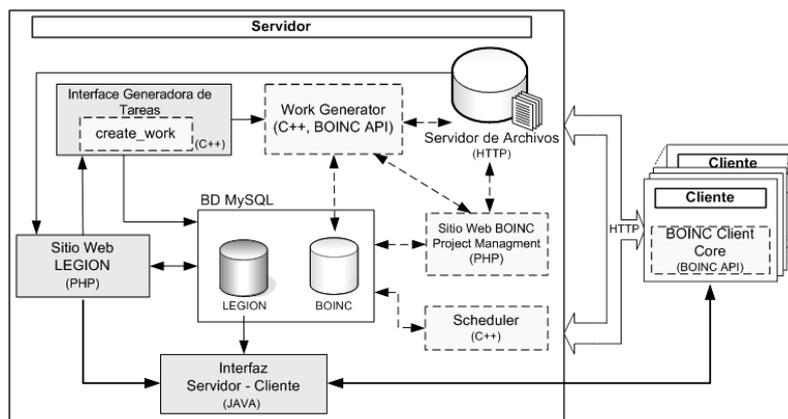


Figura 1. La Arquitectura Legion

Una característica particular de la computación en grid es que los equipos que la conforman no son necesariamente computadores dedicados. Esta propiedad, denominada pertenencia dinámica, implica que cualquier equipo puede adherirse o abandonar una determinada grid en tiempo de ejecución. Además un equipo que forme parte de una determinada grid no implica que deje de ser operativo para cualquier otro tipo de tarea, pudiendo seguir siendo utilizado por sus usuarios habituales [6].

3.1. Interfaz Web

Es la interfaz que permite el acceso de múltiples usuarios a múltiples proyectos, que forman parte del Sistema Legión, vía web. Esta interfaz fue desarrollada utilizando el lenguaje de programación PHP, requiriendo en algunos casos, el uso de técnicas de programación AJAX vía el framework Prototype [4], para indicar el progreso o avance de ejecución de una tarea.

La interfaz permite a los usuarios la generación y observación automática de diversas tareas que les son permitidas, además de la finalización de éstas para la posterior descarga del resultado respectivo.

En la mayoría de los casos la interfaz es la única para cada proyecto, dado que muchos de los parámetros de creación de las tareas son específicos al proyecto correspondiente.

3.2. Interfaz Generadora de Tareas

Es la aplicación en C++ encargada de la generación automatizada de workunits. Utiliza la función `create_work`, propia del API de BOINC, para invocar al demonio Work Generator propio de BOINC (ver Figura 1).

3.3. Interfaz Servidor - Cliente

Es un conjunto de aplicaciones realizadas en Java, que permiten comunicar al servidor Legión con los computadores clientes a través de BOINC RPC. De este modo los computadores pueden realizar peticiones forzadas de tareas hacia el servidor, minimizando el tiempo muerto de espera de las peticiones de tareas provenientes desde los clientes BOINC.

3.4. Administración de computadores cliente

El uso de una Desktop Grid implica la necesidad de administrar, de forma automática

y remota, los múltiples computadores que forman o formarán parte del grid. Debido a esto, se han desarrollado una serie de herramientas que permiten interactuar con los computadores a través de BOINC RPC. Dichas

herramientas permiten realizar acciones remotas sobre los computadores cliente, como unirse a un proyecto, suspenderse, no solicitar más tareas, separarse de un proyecto, etc.

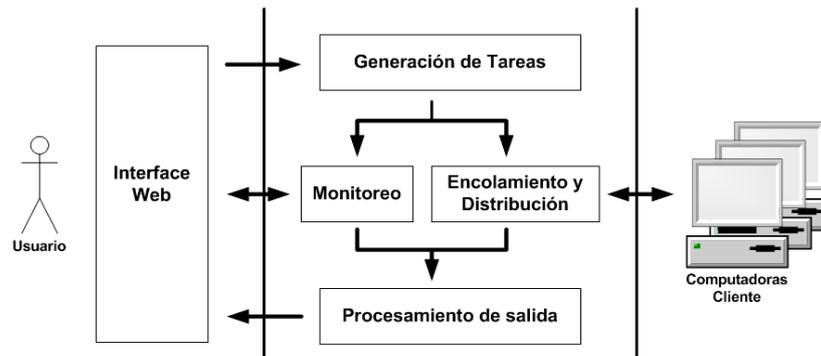


Figura 2. Flujo de trabajo de Legión

4. Funcionamiento de Legión

La interfaz web permite al usuario, entre otras cosas, la generación de tareas correspondientes a un proyecto específico. Estas tareas, internamente generan workunits mediante la Interfaz Generadora de Tareas, las cuales son enviadas a los computadores cliente vía el sistema de encolamiento y distribución, para su posterior ejecución dentro del grid. Un sistema de monitoreo supervisa los cambios de estado de todos los workunits. Una vez que todos los workunits pertenecientes a una tarea específica finalizan, se realiza un procesamiento de salida que resume los resultados obtenidos. Finalmente, se genera un reporte que se envía al usuario vía correo electrónico. La interfaz web ofrece además la posibilidad de seleccionar y descargar los resultados obtenidos.

4.1. Generación de Tareas

Las tareas creadas por los usuarios generan una serie de workunits, mediante una aplicación en C++ y una serie de librerías que provee el API de BOINC. Una vez creadas, las tareas son registradas en la base de datos MySQL. Cabe señalar, que para la mejor administración de las tareas y las workunits que la conforman, además de los usuarios que

las generan, se tuvo que realizar una personalización de la base de datos original de BOINC.

4.2. Encolamiento y Distribución

Una vez se generan los workunits, éstos son encolados por BOINC, para luego ser distribuidos y procesados en computadores cliente con recursos disponibles. Los computadores cliente seleccionados retornan sus resultados al servidor, donde son almacenados dentro de un servidor de archivos propio de BOINC. Los resultados esperan allí por el procesamiento de salida (punto 4.4).

4.3. Monitoreo

El sistema de monitoreo permite tener un control sobre los estados actuales de cada uno de los workunits que conforman una tarea. Este monitoreo se realiza vía disparadores configurados en la base de datos MySQL, permitiendo conocer el progreso de la ejecución de cada una de las tareas. La ejecución de la última tarea invoca el procesamiento de salida.

4.4. Procesamiento de salida

El procesamiento de salida es el último paso en el flujo de ejecución de una tarea, permitiendo realizar un procesamiento sobre

cada uno de los archivos de resultados parciales. Este procesamiento es particular para cada proyecto según sus requerimientos.

5. La Infraestructura utilizada

El servidor con el que se cuenta para albergar al Sistema Legión es un IBM eServer xSeries 336 8837 – Intel® Xeon® 3.2 GHz, con un total de memoria RAM de 4Gb y HDD de 73.4GB 10 000 RPM, corriendo un sistema operativo CentOS 5.3.

El Sistema Legión cuenta, actualmente, con 467 computadores que son utilizados como computadores cliente, encargados del procesamiento de todos los workunits generados. Los laboratorios informáticos de la universidad suman un total de 13, y se encuentran distribuidos según los distintos tipos de computador con los que se cuenta, donde cada uno de ellos alberga aproximadamente 40 equipos. La distribución de los mismos es como sigue:

- GenuineIntel Intel Core 2 CPU 6300 1.8 GHz – 120 computadores distribuidos en un total de 3 laboratorios.
- GenuineIntel Intel Core 2 CPU 6400 2.13 Ghz – 212 computadores distribuidos en un total de 6 laboratorios.
- GenuineIntel Intel Core 2 Duo CPU E8200 2.66 Ghz – 135 computadores distribuidos en un total de 4 laboratorios.

Sumando la capacidad de todos los computadores medida por cada cliente BOINC en FLOPS, se ha llegado a una capacidad máxima estimada de procesamiento de 1 TeraFLOPS.

6. Caso de Aplicación

El caso de aplicación esta enmarcado dentro de la Física de Altas Energías, donde el Dr. Alberto Gago, y su equipo de trabajo, han formado el Grupo de Altas Energías, en la Sección Física de la Universidad Católica del Perú.

Su trabajo consiste en la comparación de dos modelos de producción de neutrinos de alta energía en Núcleos de Galaxias Activas (NGA), a través de la variación de los valores de los parámetros involucrados en la definición de cada modelo. El programa calcula el número de neutrinos producidos por cada modelo y el número de éstos que serán detectados en el telescopio de neutrinos IceCube[5], ubicado en el Polo Sur. Estos datos simulados serán utilizados posteriormente para cuantificar, primero, la posibilidad de identificar los neutrinos provenientes de NGA, de acuerdo a cada modelo, y, segundo, la capacidad que tendría IceCube para saber cuál de los dos modelos es el que explica la producción de neutrinos.

El equipo de Legión recibió por parte del Grupo de Altas Energías el ejecutable de la aplicación antes descrita. Luego del análisis correspondiente, se llegó a la conclusión de que la aplicación podía ser “boincificada”. La solución en estos casos es utilizar el wrapper que brinda el API de BOINC para la ejecución en los computadores cliente de una aplicación previamente desarrollada.

El primer paso de la “boincificación”, es realizar la creación del proyecto BOINC, para luego realizar la personalización de la base de datos que permita llevar el control adecuado en la creación de las tareas. A medida que se van realizando pruebas controladas de la aplicación a “boincificar”, la Interfaz Generadora de Tareas es diseñada y desarrollada, para luego proceder con las primeras pruebas de generación de workunits. Luego de validar ejecuciones con pocos workunits, el monitoreo de los mismos y el Procesamiento de Salida son diseñados. Este último requiere un trabajo muy cercano con el equipo que propone el proyecto, tomando los requerimientos de sumarización propios de su estudio. Luego de ello se empiezan a realizar pruebas con el flujo completo desarrollado, desde la generación, hasta la obtención de los resultados finales.

Dado que el objetivo es que los usuarios no tengan la necesidad de conocer la arquitectura que soporta el grid, ésta se encuentra oculta a través de un módulo web desarrollado

específicamente para el proyecto, donde cada una de las partes del desarrollo mostradas pueda ser ejecutada vía la interfaz Legión. Cada usuario posee a un ambiente propio donde puede acceder a los proyectos a los que pertenece, administrar sus tareas, y acceder a los resultados de las mismas. El acceso a estos ambientes personales es validado vía web mediante un login con usuario y contraseña.

En la Figura 3 se muestra la interfaz web del Sistema Legión, específicamente la pantalla de generación de una nueva tarea. Como se muestra, el usuario debe ingresar todos los parámetros necesarios para que la aplicación se ejecute en los computadores cliente. La imagen muestra algunos de los parámetros necesarios para esta generación en particular.

Figura 3. Interfaz Legión [7]

En la Figura 4 se muestra a una tarea en ejecución, indicando además el progreso de la misma y la cantidad de computadores que se encuentran ejecutando en ese instante la tarea.

finalice, se mostrará la cantidad de Horas CPU que la tarea ha tomado, pudiendo compararse con la cantidad de horas que la tarea hubiera tomado si se ejecutase en una sola computadora.

Luego de que el Procesamiento de Salida

Fecha	Nombre de la Tarea	Progreso	PC's	Horas/CPU		
16:32:16 19-05-09	BB loopclass=1, Nalpa=Ninner=400, Emin=1.D5	62.42%	411	-	↓	□
14:59:11 19-05-09	KT con evolucion Nalpa = 400, Emin = 1.D5	100%	-	38.26	↓	□
14:43:52 19-05-09	KT con evolucion Nalpa = 400, Emin = 1.D5	100%	-	38.26	↓	□
14:26:04 19-05-09	KT sin evolucion Nalpa = 400, Emin = 1.D5	100%	-	38.25	↓	□
13:26:18 19-05-09	KT sin evolucion Nalpa = 400, Emin = 1.D5	100%	-	39.03	↓	□

Figura 4. Interfaz Legión [7]

La Figura 5 muestra a la tarea antes mencionada luego de finalizar su ejecución, tomando un tiempo total de 15 236 Horas/CPU, obteniendo un tiempo real de procesamiento dentro del Sistema Legión de

aproximadamente 60 horas.

Finalmente el link de descarga de resultados se activaría, y este estaría disponible para la descarga por parte del usuario que ha generado la tarea.

The screenshot shows the Legión web interface. At the top, there is a navigation bar with the 'Legión' logo and the Pontificia Universidad Católica del Perú logo. Below the navigation bar, the page title is 'Inicio > Proyecto Física GAE1'. The main content area is titled 'Física GAE1' and contains a table of tasks. The table has columns for 'Fecha', 'Nombre de la Tarea', 'Progreso', 'PC's', and 'Horas/CPU'. The tasks listed are:

Fecha	Nombre de la Tarea	Progreso	PC's	Horas/CPU
06:32:56 22-05-09	BB loopclass=1, Nalpha=Ninner=400, Emin=1.D6	19.81%	412	-
16:32:16 19-05-09	BB loopclass=1, Nalpha=Ninner=400, Emin=1.D5	100%	-	15236.83
14:59:11 19-05-09	KT sin evolucion Nalpha = 400, Emin = 1.D6	100%	-	38.26
14:43:52 19-05-09	KT sin evolucion Nalpha = 400, Emin = 1.D5	100%	-	38.26
14:26:04 19-05-09	KT sin evolucion Nalpha = 400, Emin = 1.D6	100%	-	38.25
13:26:18 19-05-09	KT sin evolucion Nalpha = 400, Emin = 1.D5	100%	-	39.03

Figura 5. Interfaz Legión [7]

7. Conclusiones y Trabajos Futuros

Diversos campos de investigación no pueden ser abordados de manera satisfactoria sin la ayuda de una capacidad de cómputo realmente importante. Tradicionalmente esto se resuelve con un cluster de computadoras dedicado, el cual puede representar un costo muy elevado, en términos de adquisición, instalación y mantenimiento; considérese además que, por razones de espacio, el cluster puede limitarse a un reducido número de computadores.

Por otro lado, notamos que muchas universidades en Latinoamérica cuentan con laboratorios de cómputo equipados con computadores de última generación, cuya potencia de procesamiento realmente excede las necesidades del estudiante promedio, generándose una gran capacidad de cómputo ocioso que puede ser aprovechado en proyectos de investigación computacionalmente intensivos.

La tecnología de computación en grid brindada por BOINC, permite aprovechar la infraestructura de aulas informáticas, tomando de forma controlada los recursos computacionales ociosos, de tal forma que no

interrumpa el trabajo de los usuarios, con lo que se logra obtener grandes capacidades de cómputo.

La infraestructura BOINC puede ser adaptada a diversos proyectos gracias a que posee su propia API de desarrollo, la cual puede ser incluida en aplicaciones ya implementadas, con ciertos cambios de lógica, pero no muy significativos para un desarrollador experimentado permitiendo incorporar funcionalidades de checkpoint para ejecuciones de workunits muy largas. Los proyectos también pueden ser boincificados y ejecutados en paralelo sin necesidad de utilizar el API de manera directa, como es el caso de la aplicación desarrollada en el presente proyecto.

El Sistema Legión trabaja como un excelente complemento del sistema BOINC y ofrece a los investigadores una interfaz amigable para que puedan interactuar con la grid, ocultándoles la complejidad de la misma y permitiéndoles mayor fluidez en los trabajos que deseen procesar. Además, se alivia el trabajo del administrador del sistema BOINC, ya que Legión puede configurar los nodos

automáticamente mediante las herramientas en java antes descritas.

El proyecto desarrollado optimiza el aprovechamiento de la infraestructura con que la universidad cuenta, ya que se puede obtener todo el potencial que tienen los computadores que se poseen.

Con el presente estudio se sientan las bases para que otras universidades latinoamericanas puedan replicar el estudio desarrollado, adaptándolo a sus realidades, y así fomentar el ingreso, por parte de éstas, a la eCiencia.

Nuestro trabajo futuro estará enfocado en la implementación de una interfaz web que permita la administración de los computadores, a fin de realizar en forma eficiente los procesos de mantenimiento requeridos.

El uso de un servidor de datos externo es otro de los objetivos futuros de este proyecto. Esto debido a que algunos proyectos podrían generar archivos resultado bastante extensos, por lo que un mismo servidor sería ineficiente.

El desarrollo del Sistema Legión tiene como objetivo captar a la mayor cantidad de investigadores para que hagan uso de éste, y puedan terminar sus proyectos de investigación de forma satisfactoria. Dado que los computadores de los laboratorios utilizan el sistema operativo Microsoft® Windows® XP, se esta trabajando en lograr que éstos puedan a la vez soportar un sistema operativo GNU/Linux, trabajándolos como máquinas virtuales corriendo sobre Microsoft® Windows®. Con esto logramos tener el soporte para aplicaciones nativas tanto en Microsoft® Windows® como en GNU/Linux.

8. Referencias

- [1] R. Smith, "Grid Computing: A Brief Technology Analysis", CTO Network Library, http://www.ctonet.org/documents/GridComputing_analysis.pdf, 2005
- [2] BOINC <http://boinc.berkeley.edu>
- [3] A. Abbas, "Grid Computing: A Practical Guide to Technology and Applications", Charles River Media, Massachusetts, 2004
- [4] Prototype Javascript framework <http://www.prototypejs.org>
- [5] IceCube Neutrino Observatory
- [6] P. Morillo Tena, "Grid Computing: Compartición de recursos y optimización del

hardware", Mundo Electrónico N° 346, Cetisa, Madrid, Oct. 2003, pp. 50-55.

- [7] Sistema Legión <http://legion.pucp.edu.pe>
- [8] A. Grama, A. Gupta, G. Karypis and V. Kumar, "Introduction to parallel computing", Pearson, Harlow, 2003.
- [9] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the Grid : Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, Vol. 15, N°3, 2001.
- [10] P. Plaszczak Jr., R. Wellner, "Grid Computing : The Savvy Manager's Guide", Morgan Kaufmann, San Francisco, 2005.
- [11] I. Foster, "Globus Toolkit Version 4 : Software for Service-Oriented Systems", Globus Alliance, <http://www.globus.org/alliance/publications/papers/EFIP-2006.pdf>, 2006.