

# On Existing Mobile Grids for Android Devices

Roberto García, Leonardo Flórez-Valencia, Mariela Curiel  
Pontificia Universidad Javeriana  
Bogotá, Colombia  
roberto.garcia@javeriana.edu.co,  
florez-l@javeriana.edu.co, mcuriel@javeriana.edu.co

## ABSTRACT

The rise of wireless technology and mobile devices has increased the number and types of resources to be integrated to the grid. As a result, the concept of grid has been enriched and new categories have emerged. One of these new categories is the mobile grid, which include mobile devices as resource providers. Many articles have been written in the past decade about mobile grids. These include surveys, implementations and challenges of this young technology. We want to explore the use of mobile grids for medical image processing and analysis. This impose certain constraints such as the use of C++ and MPI library. We start by evaluating existing implementations of mobile grids under these restrictions. One of our main finding is that the number of available mobile grids is really scarce. BOINC was the open source system more suitable for our research project.

## CCS Concepts

- Software and its engineering → Grid computing;
- Hardware → Wireless devices;

## Keywords

Mobile Grids, Mobile Computing, BOINC, Smartphones, Android

## 1. INTRODUCCIÓN

El uso de los dispositivos móviles se ha incrementado de forma importante alrededor del mundo en los últimos años, tanto es así que en el 2014 el acceso a la Internet fue superior desde dispositivos móviles, comparado con el acceso desde computadores personales o de escritorio [44]. Algunos autores consideran que los *smartphones* son una de las mejores invenciones disponibles para esta generación y su acelerada adopción hace pensar en que la tecnología aún seguirá desarrollándose a medida que los usuarios se vayan dando cuenta de todo lo que puede estar al alcance de algunos clicks [36].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

2016 8th Euro American Conference on Telematics and Information System (EATIS), Cartagena de Indias – Colombia. April 27 – 29

© 2016 ACM. ISBN 978-1-5090-2435-3/16/\$31.00 © 2016 IEEE.

DOI: 10.1145/1235

Por otro lado, las capacidades actuales de estos dispositivos han aumentado considerablemente; hoy en día existen *smartphones* cuyas características se comparan a las que tenían algunas estaciones de trabajo de gama alta, como la SGI Indigo-2 unos años atrás [14]. Es por esta razón que ya comienzan a considerarse como plataformas de cómputo para diversos tipos de aplicaciones, especialmente si están ociosos. Algunos autores han explorado el uso de los *smartphones* para la computación científica ([49, 21]) o como una infraestructura de cómputo alterna en una empresa que provee de *smartphones* a sus empleados [6].

En el mismo orden de ideas, desde la década anterior se han desarrollado muchos trabajos de investigación que plantean la problemática de incorporar dispositivos móviles a la grid. Una Grid es una arquitectura de hardware y software que provee acceso de manera confiable a recursos computacionales y de almacenamiento, geográficamente dispersos, pertenecientes a distintas organizaciones [24]. La extensión del modelo de grid tradicional para usar dispositivos móviles llega naturalmente debido al gran auge de la tecnología móvil en los últimos años. En particular, se denomina **Grid Móvil** a la grid donde pueden conectarse los usuarios a través de sus dispositivos móviles (*smartphones*, *Tablets*, etc) con dos propósitos: 1) obtener acceso a todos los recursos que provee el grid y/o 2) colocar sus dispositivos móviles a disposición de los usuarios de la grid tradicional. Este trabajo se enfoca en el uso de los dispositivos móviles como proveedores de recursos de cómputo [2, 41, 48, 50, 25].

El objetivo de nuestro trabajo es utilizar dispositivos móviles para la ejecución de tareas (secuenciales y paralelas) escritas en C++ y, en el caso de tareas paralelas, comunicándose mediante MPI. Se decidió utilizar una grid, en lugar de hacer una implementación propia, como en los casos de [19],[6] y [14] porque esta tecnología ofrece valores agregados, principalmente en lo que corresponde a descubrimiento de recursos, planificación y calidad de servicio. Una vez que se decide utilizar una grid móvil, el siguiente paso es evaluar el estado actual de la tecnología existente. Si bien se han escrito muchos artículos sobre middleware y herramientas ([3, 4, 5, 58, 16, 17, 37, 47, 46, 20]), no todos estos sistemas están disponibles para su uso, poseen buena documentación u ofrecen el soporte adecuado, debido a que son proyectos que no han seguido actualizándose. Uno de los usos planeados para la grid móvil a mediano plazo, en nuestro proyecto de investigación, es el procesamiento de imágenes médicas. Esto impone ciertas restricciones a la grid, como la ejecución de algoritmos en C++ y el uso de la librería de comunicaciones MPI; ambos son estándares actuales para el procesamiento

de imágenes.

En este artículo se proveerá información sobre las herramientas disponibles y se seleccionará la más idónea de acuerdo con las restricciones del proyecto de investigación. Una vez seleccionada, se explicarán los pasos necesarios para el despliegue. El documento está estructurado de la siguiente forma: en la Sección 2 se describen los principales trabajos relacionados a la investigación. En la Sección 3 se exponen los principales sistemas evaluados, sus ventajas y desventajas, así como los criterios de evaluación. Finalmente, se selecciona BOINC [4], la cual se describe en detalle en la Sección 4. El despliegue se presenta en la Sección 5. El trabajo concluye en la Sección 6.

## 2. TRABAJOS RELACIONADOS

Kurdi en [35] utiliza el término **Grid Accesible** para referirse a una grid que se centra en el acceso a los recursos. Las grid móviles son un tipo de grid accesible donde se usan los dispositivos móviles como recursos para ejecutar tareas.

El uso de los dispositivos móviles como plataformas de cómputo implica una cantidad de desafíos que provienen de las características de estos dispositivos: heterogeneidad, capacidad de CPU, tamaño de la pantalla, vida corta de la batería, movilidad, desconexiones intermitentes, entre otros. Adicionalmente, las redes inalámbricas (*wireless*) se caracterizan por limitados ancho de banda, baja confiabilidad y altas latencias. Por otro lado, la ejecución de tareas en grid móviles presenta otros retos como el tipo de aplicaciones que sería más adecuado, la seguridad y la disposición que pueden tener los usuarios a prestar sus recursos. Varios artículos se dedican a las definiciones básicas y destacan los principales desafíos de las grids móviles ([2, 26, 41, 48]). Otro grupo de trabajos expone problemas particulares tales como: arquitecturas middleware ([16, 47, 42, 29, 43]), manejo de recursos ([7, 8, 53, 31, 9, 11, 63, 22, 23, 39, 18]), planificación ([40, 56, 27, 10, 34, 1, 38]), y seguridad ([15, 51, 52, 57, 62, 60, 13]).

Entre los sistemas grid implementados se encuentran: IBIS [47], MoGrid [20], Akogrimo [32] MORE [59], MiPeG [16], Mobile OGS.NET [37] y BOINC [4].

El uso de dispositivos móviles para la ejecución de trabajos no está necesariamente ligado al concepto de grids. Existen otros grupos de investigadores que desarrollan sus propias implementaciones para la ejecución de tareas ([19], [61], [6])

## 3. EVALUACIÓN DE LAS HERRAMIENTAS DISPONIBLES

Inicialmente se realizó una revisión bibliográfica para identificar todas las posibles implementaciones de grids accesibles, especialmente móviles. Muchas se descartaron básicamente porque la herramienta no se encontraba disponible para el público. Tal es el caso de Akogrimo [32], MiPEG [16], MORE [59], Mobile OGS.NET [37] y MADAM [3].

Se seleccionaron las herramientas cuyo código fuente estaba disponible: BOINC [4], Ibis[47], MoGrid[20] y OurGrid [5]. Para evaluar estas herramientas se utilizó la metodología DAR [30]. En la primera parte de la metodología se escogen los atributos más importantes para el proyecto de investigación. Los atributos se dividieron en dos grupos de acuerdo con su importancia: atributos de prioridad alta y prioridad baja. Es obligatorio que la herramienta seleccionada soporte los atributos de alta prioridad. Los atributos de prioridad

menor pueden ser prescindibles por las siguientes razones: a) Son deseables para algunos usos futuros del sistema pero no imprescindibles actualmente. b) Puede que sin la existencia de estos atributos se compliquen algunas tareas (modificación de código, por ejemplo), lo cual no quiere decir que sea imposible su realización. De los ocho atributos que se mencionan a continuación sólo los tres primeros se consideraron de alta prioridad.

**Ejecución en dispositivos móviles:** Este atributo es de alta prioridad, ya que uno de los principales objetivos de la investigación es poder ejecutar tareas en una grid móvil.

**Lenguaje de las tareas que se pueden ejecutar:** Este atributo es de alta prioridad porque se requiere que la grid móvil soporte la ejecución de tareas en C++, ya que en este lenguaje están escritas muchas aplicaciones científicas, además de la mayoría de las librerías y *frameworks* para el procesamiento y análisis de imágenes médicas. Mientras mayor cantidad de lenguajes soporte, mejor será la evaluación de la herramienta.

**Mantenimiento del proyecto:** Es importante saber en qué estado está el proyecto, es decir, si sigue en desarrollo y si ha logrado una versión de distribución al público, por lo que existe una comunidad que ha puesto a prueba el software. Estas características garantizan un software más completo, libre de errores y la existencia de una comunidad que puede dar soporte en el caso de problemas. Por otro lado, un proyecto que ha sido abandonado no presenta estas ventajas, ni estará incorporando las innovaciones tecnológicas que surgen continuamente.

**Tipo de grid accesible:** Kurdi define las grids accesibles como grid móvil, ad-hoc o inalámbrica. Una meta a largo plazo del proyecto de investigación es desarrollar una grid ad-hoc que pueda conformarse con dispositivos móviles cercanos, en caso de emergencias o de desastres. Sin embargo, aunque este es el caso ideal, se trata de una meta a largo o mediano plazo, y al no dominar aún la tecnología de la grid móvil, se puede comenzar desplegando una grid móvil formada por dispositivos móviles y fijos.

**Uso de MPI:** el soporte de MPI es un atributo deseable para el procesamiento y análisis de datos (en particular imágenes) médicos, ya que la versión paralela de la mayoría de algoritmos usa MPI. No obstante, la mayoría de estos algoritmos utilizan un esquema Maestro-Esclavo bajo el esquema SIMD (*Single Instruction Multiple Data*) y la librería de comunicaciones se utiliza casi exclusivamente para repartir los datos entre los nodos. En una grid, el uso de MPI con este propósito no es fundamental ya que la transmisión de los datos es parte de las responsabilidades de la grid. Como trabajo futuro se tendrá que estudiar la modificación de los algoritmos existentes, que usan MPI, para que se ejecuten en la grid.

**Calidad de la documentación:** Este atributo facilita el despliegue de la grid y futuras modificaciones.

**Facilidad para ejecución de tareas:** Mientras más fácil sea el uso de la grid, existirán más usuarios interesados en esta tecnología.

**Tabla 1: Herramientas y Atributos**

Atributos	BOINC	Ibis	MoGrid	OurGrid
Ejecución en móviles	4	3	2	1
Lenguajes de las tareas a ejecutar	4	3	3	2
Mantenimiento del proyecto	4	2	1	3
Tipo de grid	3	2	4	1
Uso de MPI	3	4	2	2
Calidad de la documentación	4	1	2	3
Facilidad de ejecución de las aplicaciones	4	2	1	3
Facilidad de modificar el código	1	4	2	3
<b>Total</b>	<b>90</b>	<b>64</b>	<b>54</b>	<b>54</b>

**Facilidad de modificar el código:** Este atributo está relacionado con el diseño del sistema, lo cual influye en qué tan fácil es entenderlo y modificarlo.

Después de seleccionar los atributos, se le dio un coeficiente de 5 a los atributos de primera prioridad, y un coeficiente de 2 a los prioridad menor. Luego se organizaron los cuatro sistemas de acuerdo a la valoración de los atributos. Al atributo más alto se le da el indicador 4, y al peor 1. Por último, se calcula el puntaje total para cada proyecto o herramienta sumando la multiplicación de coeficiente e indicador. Este proceso es el definido por la metodología DAR.

### 3.1 Evaluación

La tabla 1 muestra la puntuación de las herramientas después de aplicar la metodología DAR. Con respecto a la ejecución en dispositivos móviles, BOINC toma el primer puesto, ya que incluye una implementación de computación voluntaria que está actualmente en funcionamiento con decenas de miles de instalaciones en Android [33, 54]. Ibis toma el segundo lugar, ya que por estar todos sus módulos implementados en Java, se ha explorado portar las partes principales del sistema al sistema operativo Android. Aunque los resultados reportados fueron positivos [47], en el portal de la herramienta no se han agregado las implementaciones para Android. MoGrid no cuenta con clientes móviles disponibles para uso público y en el caso de OurGrid no se reporta su uso en dispositivos móviles.

Con referencia a los lenguajes soportados para las tareas que se ejecutarán en el grid, BOINC toma de nuevo el primer lugar ya que soporta varios lenguajes utilizando *wrappers* o envoltorios. Los *wrappers* ejecutan las aplicaciones como subprocesos y manejan toda la comunicación con el cliente BOINC. Actualmente existen *wrappers* para C/C++, Java y Python. Para Fortran existe un método de convertir el código a C y luego tratarlo como tal o utilizar los *wrappers* de C. MoGrid e IBIS sólo soportan programas escritos en Java.

El tercer atributo de alta prioridad se refiere a las actua-

lizaciones del proyecto. BOINC es un proyecto muy activo, cuyo último hito fue la conferencia BOINC:FAST 2015 que se llevó a cabo en septiembre. Se ha formado una comunidad alrededor de BOINC que se ha involucrado y ha enriquecido el proyecto. OurGrid toma el segundo lugar dado que su última versión es de finales de 2013. El proyecto, al unir los recursos computacionales de múltiples laboratorios pequeños y medianos, tiene una comunidad participando y mejorando el producto. Ibis toma el tercer lugar, debido a que la última actualización reportada fue en 2012, y el último hito ocurrió en 2014 que fue la entrega del premio *Euro-Par Achievement Award* 2014 al profesor Henri Bal, miembro del proyecto. MoGrid no tiene su código a disposición del público en la actualidad.

Con respecto a los atributos de menor prioridad se puede comentar que el único de los tres sistemas que soporta la formación de grids ad-hoc es MoGrid [28]. BOINC necesita de un gestor central para su despliegue.

El soporte de la librería de comunicación MPI o similares se observa mejor en Ibis. Este sistema ofrece MPJ que es una implementación en Java de la librería MPI. Ofrece alto rendimiento y se puede desplegar en varias plataformas, desde clusters basados en Myrinet hasta grids, aunque en la documentación no hay referencia a su uso en dispositivos móviles. En BOINC es posible ejecutar aplicaciones que usen MPI, pero los procesos que se comunican deben estar ejecutándose en una sola máquina host [45]. No se observa en la documentación de la herramienta el uso de MPI sobre dispositivos Android. El resto de los sistemas no soporta esta librería de comunicaciones.

La documentación extensa de BOINC facilita la ejecución de aplicaciones. BOINC fue diseñado para ser usado por científicos, no por programadores o profesionales de la TIC y por esa razón las herramientas son simples y están bien documentadas. No obstante, la estructura modular de Ibis, hace inferir que sería el sistema más adecuado para la introducción de nuevas funcionalidades.

Finalmente se elige BOINC, dado que es el sistema con mayor puntuación. En la próxima sección se describe este sistema de forma más detallada.

## 4. BOINC

### 4.1 Descripción General

BOINC es una infraestructura para la computación voluntaria distribuida, desarrollada originalmente en la Universidad de Berkley para el proyecto SETI@home. Actualmente hay otros proyectos usando BOINC en diversos campos como la física, la medicina nuclear, climatológica, etc. [12]. BOINC es un medio de aprovechar una capacidad de cómputo enorme, utilizando inicialmente computadores personales alrededor del mundo y, posteriormente, también el poder de procesamiento de dispositivos móviles [54]. BOINC está diseñado para ser una plataforma libre para cualquier persona o institución que quiera crear un proyecto de computación distribuida.

### 4.2 Componentes

Los principales conceptos usados en la Arquitectura de BOINC se describen a continuación:

Un **proyecto** (*project*) se asocia con una institución o grupo de investigación que tiene un proyecto de computación voluntaria o distribuida. Se identifica por una URL maestra

(*master URL*), que es la página de un sitio WEB y también sirve como un directorio de servidores que actúan como planificadores. Los participantes o voluntarios se registran con un determinado proyecto. Desde el proyecto, o gestor, se administran las aplicaciones y sus versiones, las tareas pendientes, las completadas y canceladas, los usuarios, la asignación de recursos, etc. Una o más aplicaciones pueden estar asociadas a un proyecto.

Dentro de BOINC una **aplicación** es un conjunto de programas (para diferentes sistemas operativos y tipos de procesadores) que realizan el mismo cálculo. Estos programas individuales se denominan versiones (*app versions*). Por ejemplo, para que una aplicación utilice un recurso Android debería existir la versión de la aplicación específica para este sistema operativo.

Para poder ejecutar tareas o trabajos, deben asociarse a las aplicaciones, y BOINC decide cuál *app version* usar cuando envía una tarea a un determinado recurso.

Otro de los componentes principales es el **servidor**. Las funciones de un servidor son ejecutadas por un conjunto de servicios web o procesos demonios: Los **servidores de planificación** manejan los RPCs que realizan los clientes y los reportes de trabajos culminados. Los **servidores de datos** manejan las cargas de los usuarios o *file uploads* usando un mecanismo basado en certificados para asegurar que los usuarios sólo carguen archivos legítimos, con tamaños específicos.

El **cliente** BOINC es ejecutado en la máquina o dispositivo voluntario. El cliente se encarga de ejecutar las tareas que le son asignadas por el gestor. Dependiendo de las propiedades de la máquina se tiene un cliente específico, por ejemplo, se tienen clientes para Windows, para Android y para Linux, entre otros. El **BOINC Manager** es una interfaz gráfica que permite la administración y configuración del cliente BOINC.

BOINC provee un **API** definido en lenguaje C/C++. El API contiene los métodos que deben ser ejecutados por una tarea para iniciar y finalizar su ejecución, comunicar avances, acceder a archivos de entrada y salida y reportar errores, entre otros. Para que una tarea pueda ser ejecutada en la plataforma BOINC debe utilizar el API o los *wrappers*. Si los *wrappers* están disponibles para una determinada plataforma, no es necesario reescribir las aplicaciones para insertar llamadas a BOINC. Aún no hay *wrappers* para ejecutar tareas en Android.

Para participar como voluntario se debe descargar un cliente BOINC. Nuestro objetivo, una vez realizada la evaluación de herramientas, es instalar BOINC en nuestra institución como una plataforma de cómputo distribuido con recursos móviles y fijos. En la próxima Sección se explican los pasos necesarios para desplegar BOINC y ejecutar un programa simple escrito en C++.

## 5. DESPLIEGUE Y PRUEBAS

Para realizar el despliegue se decide utilizar el sistema operativo Linux Debian, debido a que gran parte de la documentación de BOINC está escrita específicamente para este sistema operativo. Para el despliegue y pruebas de utilizaron los siguientes equipos:

- Teléfonos celulares Samsung S4, Arquitectura ARM, Android Versión 4.2.
- BOINC Versión 7.4.4.1.

- Máquina host con Windows 8.1, 6GB de RAM, intel i7, 2360QM.
- Servidor DNS MaraDNS.
- Gestor de máquinas virtuales Oracle VM VirtualBox versión 4.3.20.
- Máquina virtual Linux Debian con servidor Web Apache y MySQL.

El gestor de BOINC se instaló en la máquina virtual Debian. Para crear un proyecto se debe ejecutar un script cuyo resultado es una aplicación alojada en el servidor web de la máquina virtual. Esta aplicación web es el gestor que está asociado a un proyecto.

Una vez instalado el gestor se procede a instalar el cliente en los dispositivos Android. El cliente BOINC para dispositivos móviles Android se encuentra en la Play Store de Google. Se debe tener en cuenta que este cliente requiere de una versión de Android 4.1 o superior. De igual forma sólo se soportan dispositivos de arquitectura ARM, MIPS y x86.

### 5.1 Proceso de Despliegue

Para el despliegue se utilizó la metodología PPDIOO[55], cuyos pasos son: Preparar, Planear, Diseñar, Implementar, Operar y Optimizar de manera iterativa al interior de una organización. Aunque el método fue creado para el diseño y gestión de redes de comunicación, se utilizó en este caso debido a que se preveía que los problemas con el despliegue, uso y gestión del sistema BOINC estarían relacionados con la red. Para el trabajo descrito en este artículo sólo se aplicaron las tres primeras fases de la metodología. Todas las fases se describen a continuación:

**Preparar** : En esta fase se definen los requerimientos del despliegue para el proyecto de investigación, a saber: facilidad de replicar, portabilidad y conectividad con dispositivos móviles.

**Planear** : Durante la planeación se determinan los recursos que serán necesarios en el diseño e implementación: a) tecnología de máquinas virtuales, ya que la generación de una máquina virtual pre-configurada facilita la replicación de la grid en otras plataformas del proyecto; b) un switch inalámbrico, para conectar los componentes en una red local.

**Diseñar** : En esta fase se define el diseño detallado del despliegue a realizar.

**Implementar** : Esta fase se realizó en tres etapas. Primero, se despliega el grid a nivel local, en una máquina virtual; luego, se agregan al sistema los recursos de una máquina host; por último, se forma la grid agregando dispositivos móviles. En cada una de las etapas se resolvieron problemas particulares, por ejemplo, cuando se incluye la máquina host, fue necesario asegurar que todas las comunicaciones pertinentes estuvieran llegando al gestor BOINC desde una máquina real. La incorporación de dispositivos móviles requirió de la modificación y compilación cruzada del código a ejecutar en la grid.

**Operar** : En esta fase, como su nombre lo indica, se opera el sistema desplegado en un contexto de día a día.

**Optimizar** : En esta fase, a partir de lo observado en su operación, se realizan pequeñas optimizaciones sobre la grid. También se realizan optimizaciones a partir de la evaluación de los requerimientos de negocio originales y se asegura que éstos no hayan cambiado; si es el caso se inicia el proceso de nuevo.

## 5.2 Ejecución de Programas en C++

Cada vez que se crea un proyecto en BOINC es necesario desplegar una aplicación, y dentro de ella configurar diferentes versiones para varios tipos de clientes, por ejemplo Windows x86, Linux x86, y Android ARM.

Infortunadamente para recursos Android no se tiene un *wrapper* que permita ejecutar de forma automática programas en C++ en la grid móvil. Por lo tanto, se debe modificar la aplicación, utilizando el API de BOINC para controlar el inicio y final de la tarea, al igual que las operaciones de entrada/salida. Una vez modificada la nueva aplicación ésta se debe compilar usando la técnica de compilación cruzada para la arquitectura Android deseada, en este caso ARM. Para la compilación cruzada se debe realizar un nuevo procedimiento que consiste en los siguientes pasos:

1. Se descarga el *Android Native Development Kit*. Este Kit de desarrollo contiene un conjunto de herramientas para generar un *Toolchain*.
2. Una vez que se tiene el *Toolchain* se hace primero la compilación cruzada del código fuente de BOINC, incluyendo las librerías.
3. Por último, se realiza la compilación cruzada de la aplicación, lo cual produce una versión para Android ARM dentro de la carpeta del proyecto.

Para ejecutar la aplicación ya compilada en la grid hay que realizar varios pasos: 1) agregar la aplicación a la base de datos, para que esté visible al *scheduler*; 2) crear templates para los archivos de entrada y salida; 3) preparar el archivo template para la descarga del ejecutable en los dispositivos, este template define la estructura que deben cumplir los archivos de entrada y salida; 4) finalmente, se deben crear las tareas con los comandos de BOINC apropiados. El cliente BOINC, ejecutándose en los dispositivos móviles, periódicamente pregunta al gestor por tareas y cuando las encuentra las ejecuta enviando los resultados de nuevo al gestor. Una estrategia usada por el gestor de BOINC para tolerancia a fallos es replicar la misma tarea y comparar posteriormente los diferentes resultados para asegurar que sean iguales.

El usuario voluntario puede observar cómo está siendo utilizado su recurso, así como controlar la utilización de su dispositivo móvil por parte de la grid.

## 6. CONCLUSIONES

Durante el desarrollo de este trabajo se evaluaron varias herramientas para el despliegue de una grid móvil donde pudieran ejecutarse programas escritos en C++. Después de evaluar 4 sistemas, se escogió el proyecto BOINC, que es una infraestructura para la computación voluntaria distribuida, desarrollada en la Universidad de Berkley, pero que también está diseñada para su uso por cualquier institución que quiera crear un proyecto de computación distribuida. Entre las principales ventajas de BOINC para los propósitos del proyecto de investigación se encontraron:

- Permite la inclusión de dispositivos móviles entre sus recursos. No obstante, dado que necesita un gestor central, sería una **grid híbrida** con recursos móviles y fijos.
- Puede ejecutar tareas en varios lenguajes de programación. No obstante, la ejecución de código en C++ en dispositivos móviles requiere de la modificación del programa para introducir ciertas operaciones del API relacionadas con el inicio y la terminación de las tareas, y con operaciones de E/S.
- Los programas en C++ tienen que ser previamente compilados usando compilación cruzada. BOINC ofrece ayudas para la compilación cruzada y para la colocación de tareas en los dispositivos; ambas tareas son complejas si se tienen que realizar manualmente.
- Posee buena documentación.
- Es un proyecto activo con muchos recursos que permiten resolver posibles dificultades en su despliegue y uso.

Entre las desventajas de la grid escogida se encuentran: a) con BOINC no se puede formar una grid ad-hoc ya que necesita de un gestor central, b) el uso de MPI está limitado a un nodo, c) es necesario modificar el código fuente de los programas. Como trabajo futuro se diseñará un *wrapper* para ejecutar aplicaciones en dispositivos móviles sin necesidad de modificarlas.

Finalmente, después de una búsqueda bibliográfica extensa y de evaluar varias herramientas, encontramos que muy pocas se adaptaban totalmente a los requerimientos del proyecto de investigación. La existencia de pocas plataformas da cuenta de una tecnología aún inmadura y de un campo muy fértil de investigación.

## 7. REFERENCES

- [1] A. Adeyelu, E. Olajubu, A. Aderounmu, and T. Ge. Article: A model for coordinating jobs on mobile wireless computational grids. *International Journal of Computer Applications*, 84(13):17–24, December 2013. Published by Foundation of Computer Science, New York, USA.
- [2] S. P. Ahuja and J. R. Myers. A survey on wireless grid computing. *The Journal of Supercomputing*, 37(1):3–21, 2006.
- [3] M. Alia, F. Eliassen, S. Hallsteinsen, and E. Stav. Madam: towards a flexible planning-based middleware. In *Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems*, pages 96–96. ACM, 2006.
- [4] D. P. Anderson. Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 4–10. IEEE, 2004.
- [5] N. Andrade, W. Cirne, F. Brasileiro, and P. Roisenberg. Ourgrid: An approach to easily assemble grids with equitable resource sharing. In *Job scheduling strategies for parallel processing*, pages 61–86. Springer, 2003.
- [6] M. Y. Arslan, I. Singh, S. Singh, H. V. Madhyastha, K. Sundaresan, and S. V. Krishnamurthy. Computing

- while charging: building a distributed computing infrastructure using smartphones. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 193–204. ACM, 2012.
- [7] M. Birje and S. Manvi. Multiagent model for device state control in the wireless grid. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 3, pages 456–460. IEEE, 2011.
- [8] M. Birje and S. Manvi. Wigramma: A wireless grid monitoring model using agents. *Journal of Grid Computing*, 9(4):549–572, 2011.
- [9] M. N. Birje and S. S. Manvi. Multiagent model for device state control in the wireless grid. In *Proc. of the 3rd International Conference on Electronics Computer Technology (ICECT)*, 2011.
- [10] M. N. Birje, S. S. Manvi, and C. Bulla. Economical job scheduling in wireless grid. In *Proc. of the 3rd International Conference on Electronics Computer Technology (ICECT)*, 2011.
- [11] M. N. Birje, S. S. Manvi, and S. K. Das. Reliable resources brokering scheme in wireless grids based on non-cooperative bargaining game. *Journal of Network and Computer Applications*, 2013.
- [12] D. P. A. BOINC. A system for public-resource computing and storage, 5th iee. In *ACM International Workshop on Grid Computing*, 2004.
- [13] T. Brooks and L. McKnight. Securing wireless grids: Architecture designs for secure wicket-to-wicket interfaces. *International Journal of Information and Network Security (IJINS)*, 2(1):1–16, 2012.
- [14] F. Büsching, S. Schildt, and L. Wolf. Droidcluster: Towards smartphone cluster computing—the streets are paved with potential computer clusters. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 114–117. IEEE, 2012.
- [15] E. Cody, R. Sharman, R. H. Rao, and S. Upadhyaya. Security in grid computing: A review and synthesis. *Decision Support Systems*, 44(4):749–764, 2008.
- [16] A. Coronato and G. D. Pietro. Mipeg: A middleware infrastructure for pervasive grids. *Future Generation of Computer Systems*, 24(3-21):17–29, 2008.
- [17] G. Coulson, P. Grace, G. S. Blair, D. Duce, C. Cooper, and M. Sagar. A middleware approach for pervasive grid environments. In *Workshop on Ubiquitous Computing and e-Research National eScience Centre*, 2005.
- [18] M. Curiel. Wireless grids: Recent advances in resource and job management. In *Handbook of Research on Next Generation Mobile Communication Systems*, pages 293–320. IGI Global, 2016.
- [19] P. Datta, S. Dey, H. S. Paul, and A. Mukherjee. Angels: A framework for mobile grids. In *Applications and Innovations in Mobile Computing (AIMoC), 2014*, pages 15–20. IEEE, 2014.
- [20] L. dos S Lima, A. T. Gomes, A. Ziviani, M. Endler, L. F. Soares, and B. Schulze. Peer-to-peer resource discovery in mobile grids. In *Proceedings of the 3rd international workshop on Middleware for grid computing*, pages 1–6. ACM, 2005.
- [21] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand. Motivating smartphone collaboration in data acquisition and distributed computing. *Mobile Computing, IEEE Transactions on*, 13(10):2320–2333, 2014.
- [22] U. Farooq and W. Khalil. A generic mobility model for resource prediction in mobile grids. In *Collaborative Technologies and Systems, 2006. CTS 2006. International Symposium on*, pages 189–193. IEEE, 2006.
- [23] U. Farooq, S. Mahfooz, and W. Khalil. An efficient resource prediction model for mobile grid environments. *Punjab University College of Information Technology Lahore, Pakistan*, 2006.
- [24] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid, an open grid services architecture for distributed systems integration. Technical report, Global Grid Forum, 2002.
- [25] J. Furthmüller and O. P. Waldhorst. A survey on grid computing on mobile consumer devices. *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing*, pages 313–363, 2010.
- [26] L. Gang, S. Hongmei, H. Gao, H. Yu, and Y. Cai. A survey on wireless grids and clouds. In *Eighth International Conference on Grid and Cooperative Computing*, 2009.
- [27] P. Ghosh and S. K. Das. Mobility-aware cost-efficient job scheduling for single-class grid jobs in a generic mobile grid architecture. *Future Generation Computer Systems*, 26(8):1356–1367, 2010.
- [28] A. T. A. Gomes, A. Ziviani, L. d. S. Lima, and M. Endler. Dichotomy: A resource discovery and scheduling protocol for multihop ad hoc mobile grids. In *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*, pages 719–724. IEEE, 2007.
- [29] P. Grace, D. Hughes, G. Coulson, G. S. Blair, B. Porter, and F. Taiani. Overlay-based middleware for the pervasive grid. *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing, IGI-Global*, pages 981–1002, 2010.
- [30] T. P. Group. Decision analysis and resolution process. *newsletter*, 2005.
- [31] M. Hijab and D. Avula. Resource discovery in wireless, mobile and ad hoc grids: Issues and challenges. In *Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT)*, 2011.
- [32] J. Jähnert, S. Wesner, and V. Villagrà. The akogrmo mobile grid reference architecture-overview, whitepaper, 2008.
- [33] F. Joachim. Boinc/android: Status, september 2015.
- [34] K. Katsaros and G. C. Polyzos. Evaluation of scheduling policies in a mobile grid architecture. In *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, pages 390–397. IEEE, 2008.
- [35] H. Kurdi, M. Li, and H. Al-Raweshidy. A classification of emerging and traditional grid systems. *Distributed Systems Online, IEEE*, 9(3):1–1, 2008.
- [36] D. K. O. Leigh A. Mutchler, Jung P. Shim. Exploratory study on users’ behavior: Smartphone

- usage. In *AMCIS 2011 Proceedings*, 2011.
- [37] A. Litke, D. Halkos, K. Tserpes, D. Kyriazis, and T. Varvarigou. Fault tolerant and prioritized scheduling in ogsa-based mobile grids. *Concurrency and Computation: Practice and Experience*, 21(4):533–556, 2009.
- [38] A. Litke, D. Skoutas, K. Tserpes, and T. Varvarigou. Efficient task replication and management for adaptive fault tolerance in mobile grid environments. *Future Generation Computer Systems*, 23(2):163–178, 2007.
- [39] A. Litke, D. Skoutas, and T. Varvarigou. Mobile grid computing: Changes and challenges of resource management in a mobile grid environment. In *5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004)*, 2004.
- [40] A. Mahmood and E. S. Elmallah. Incremental routing and scheduling in wireless grids. In *Global Telecommunications Conference, GLOBECOM 2009*, 2009.
- [41] S. Manvi. A review of wireless grid computing. *International Journal of Computer and Electrical Engineering*, 2(3):1793–8163, 2010.
- [42] M. Masinde, A. Bagula, and V. Ndegwa. Mobigrid: a middleware for integrating mobile phone and grid computing. In *Network and Service Management (CNSM), 2010 International Conference on*, pages 523–526. IEEE, 2010.
- [43] L. W. McKnight, T. Bose, J. Marsden, E. Nanno, J. Treglia, H. Volos, X. Chen, P. Sharma, and X. Ma. Open specifications for wireless grids technical requirements. *Version 0.1 approved: WiGiT Group. Syracuse, NY: School of Information Studies, Syracuse University*, 2012.
- [44] R. Murtagh. Mobile now exceeds pc: The biggest shift since the internet began, Julio 2014.
- [45] U. of California. Boinc wiki: Mpiapps.
- [46] K. Ohta, T. Yoshikawa, T. Nakagaw, and H. Inamura. Design and implementation of mobile grid middleware for handsets. In *Proceedings. 11th International Conference on Parallel and Distributed Systems*, 2005.
- [47] N. Palmer, R. Kemp, T. Kielmann, and H. Bal. Ibis for mobility: solving challenges of mobile computing using grid techniques. In *Proc. of the 10th workshop on Mobile Computing Systems and Applications*, 2009.
- [48] K. Parmar, N. Jani, P. Shrivastav, and M. Patel. Mobile grid computing: Facts or fantasy? *International Journal of Multidisciplinary Sciences and Engineering*, 4(1), 2013.
- [49] J. M. Rodríguez, C. Mateos, and A. Zunino. Are smartphones really useful for scientific computing? In *Advances in New Technologies, Interactive Interfaces and Communicability*, pages 38–47. Springer, 2012.
- [50] J. M. Rodriguez, A. Zunino, and M. Campo. Introducing mobile devices into grid systems: a survey. *International Journal of Web and Grid Services*, 7(1):1–40, 2011.
- [51] D. G. Rosado, E. Fernández-Medina, J. López, and M. Piattini. Obtaining security requirements for a mobile grid system. *International Journal of Grid and High Performance Computing (IJGHPC)*, 1(3):1–17, 2009.
- [52] D. G. Rosado, E. Fernández-Medina, J. López, and M. Piattini. Analysis of secure mobile grid systems: A systematic approach. *Information & Software Technology - INFISOFT*, 52:517–536, 2010.
- [53] M. Sedaghat and M. Othman. Agent based resource brokering and allocation in wireless grid: Revisited. *Journal of Theoretical and Applied Information Technology*, 2009.
- [54] M. Szpakowski. Native boinc, 2012.
- [55] J. Tiso and D. Teare. Chapter 1: The cisco enterprise architecture. In *Designing Cisco Network Service Architectures (ARCH): Foundation Learning Guide*, pages 1–21. 2011.
- [56] S. Vaithiya and S. Bhanu. Scheduling tasks in mobile grid environment using mobility based resource prediction. In *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on*, pages 89–94. IEEE, 2010.
- [57] P. Varalakshmi, S. T. Selvi, R. G. Kumar, and S. S. Raman. Security in mobile grid services. In *First International Conference on Advanced Computing, 2009. ICAC 2009.*, 2009.
- [58] S. Wesner, T. Dimitrakos, and K. Jeffrey. Akogrimo—the grid goes mobile. *ERCIM News*, (59), pages 32–33, 2004.
- [59] A. Wolff, S. Michaelis, J. Schmutzler, and C. Wietfeld. Network-centric middleware for service oriented architectures across heterogeneous embedded systems. In *EDOC Conference Workshop, 2007. EDOC'07. Eleventh International IEEE*, pages 105–108. IEEE, 2007.
- [60] Y. Xiao. *Security in distributed, grid, mobile, and pervasive computing*. CRC Press, 2007.
- [61] Z. Yannes. Portable mpich2 clusters with android devices. 2015.
- [62] Z. S. Y. Yue Hong Zhang, Li Hao. Security scheme in wireless grid. *Advanced Research on Computer Education, Simulation and Modeling Communications in Computer and Information Science*, 175:453–457, 2011.
- [63] Y. Zhang and Y. Pei. A resource discovery algorithm in mobile grid computing based on ip-paging scheme. In C. Ding, Z. Shao, and R. Zheng, editors, *Network and Parallel Computing*, volume 6289 of *Lecture Notes in Computer Science*, pages 402–411. Springer Berlin Heidelberg, 2010.