

## Acceso a Recursos de Cómputo de Alto Rendimiento Mediante Correo Electrónico

---

Suilan Estévez Velarde  
[sestevez@matcom.uh.cu](mailto:sestevez@matcom.uh.cu)

*Universidad de La Habana*

Amanda Miotto

[miotto@griffith.edu.au](mailto:miotto@griffith.edu.au)

*Griffith University*

Andrew Lewis

[a.lewis@griffith.edu.au](mailto:a.lewis@griffith.edu.au)

*Griffith University*

Antonio Bolufé-Röhler

[bolufe@matcom.uh.cu](mailto:bolufe@matcom.uh.cu)

*Universidad de La Habana*

Stephen Chen

[syichen@yorku.ca](mailto:syichen@yorku.ca)

*York University*

### RESUMEN

El cómputo de alto rendimiento es una necesidad para el desarrollo de investigaciones con grandes volúmenes de datos. La creciente demanda de este tipo de resultados ha impulsado a varios centros de investigación a poner en funcionamiento recursos de cómputo de alto rendimiento. En Cuba no existe una solución definitiva que permita a todos los centros de investigación disponer de los recursos de cómputo necesarios para desarrollar sus proyectos. Este trabajo propone el empleo de un clúster de computadoras de la Universidad de Griffith a través de una interfaz basada en el correo electrónico. Esta solución permite disponer de recursos de cómputo de alto rendimiento sin necesidad de una alta conectividad. Como caso de estudio se analizan los resultados obtenidos en un proyecto de optimización global en grandes dimensiones desarrollado en la Universidad de La Habana. Para experimentos con un mes de duración (en una computadora estándar) los resultados muestran que al utilizar el recurso de alto rendimiento es posible alcanzar un incremento en el rendimiento relativo superior al 1300%.

**PALABRAS CLAVE:** Cómputo de Alto Rendimiento, Nimrod, Clúster de Computadoras, Programación Distribuida, Programación en Paralelo.

## INTRODUCCIÓN

En la actualidad se desarrollan numerosos proyectos e investigaciones que requieren un elevado poder de cómputo. Existen tres grandes áreas de investigación que tratan con problemas muy complejos: la minería de datos (Hand, 2007), la simulación (Chen et al., 2009) y la optimización en grandes dimensiones (Pardalos and Romeijn, 2002). Las aplicaciones de estas investigaciones tienen impacto directo en campos tan variados como la física cuántica (Joshi et al., 2009) y relativista (Shibata y Uryū, 2002), la computación financiera (Egloff, 2011), la ingeniería genética (Aluru et al., 2006) y el acoplamiento molecular (Kendall et al., 2000), por solo citar algunos. Si se usan solamente computadoras personales realizar un experimento en estos proyectos puede demorar meses. La programación en paralelo ha surgido precisamente para dar respuesta a estas limitaciones. En esta se distribuyen las tareas entre varios equipos, disminuyendo la carga asignada de forma individual y disminuyendo considerablemente el tiempo en que se obtienen los resultados. Los clúster de computadoras son actualmente muy utilizados como herramientas para la paralelización de tareas de alto costo computacional.

El incremento de la cantidad de información disponible en la actualidad ha provocado un gran interés en resolver problemas con grandes volúmenes de datos. En muchos problemas, como la minería de datos, los procesos de toma de decisiones, análisis y el diseño óptimo de sistemas complejos, se utiliza un enfoque de optimización. El propósito en este enfoque es encontrar la mejor solución posible entre todas las soluciones potenciales. En dependencia del tamaño de los datos (dimensiones del problema), el poder de cómputo requerido para dar solución al problema puede aumentar significativamente.

Por ello, al realizar optimización en grandes dimensiones se utiliza con frecuencia cómputo de alto rendimiento o cómputo voluntario. En el cómputo voluntario varias computadoras personales facilitan la utilización de sus recursos para contribuir al cómputo científico. El cómputo de alto rendimiento, también conocido como *High Performance Computing* (HPC), se basa en tener un conjunto de computadoras destinadas únicamente al cómputo científico. En la práctica el HPC devuelve resultados en mucho menos tiempo, es más eficiente y sus resultados son más confiables. Pero crear un clúster de computadoras para realizar cómputo de alto rendimiento es muy costoso.

Diversas instituciones brindan sus recursos de cómputo de alto rendimiento como servicio a terceros. La Universidad de Griffith en Australia dispone de un clúster de alto rendimiento que cuenta con una interfaz web. Los problemas de conectividad que existen actualmente en Cuba dificultan el uso de esta interfaz. La variante que se propone en nuestro trabajo utiliza el correo electrónico como interfaz para acceder a este servicio de cómputo de alto rendimiento.

## ESTADO DEL ARTE

Trabajar con grandes volúmenes de datos es prácticamente imposible con los recursos disponibles en las computadoras personales modernas. Para lidiar con este problema se divide la tarea en partes que trabajan de forma independiente. Esto es conocido como cómputo en paralelo (cómputo masivo) y puede realizarse sobre distintos núcleos en una misma computadora o sobre diferentes computadoras agrupadas en un clúster. Un clúster es un conjunto de computadoras que colaboran en la solución de una tarea.

Desde hace un tiempo se utiliza el cómputo de alto rendimiento para resolver problemas prácticos que requieren computar grandes volúmenes de cálculos, esto se evidencia en la variedad de publicaciones relacionadas con el tema como son (Jackson et al., 2010), (Rodríguez, 2012) y (Plaza et al., 2011). La utilización de estas soluciones permite crear recursos más poderosos que superan problemas que representaban un reto para las soluciones no paralelas.

Existen dos formas fundamentales de realizar cómputo en paralelo: el cómputo voluntario y el cómputo de alto rendimiento. El cómputo voluntario emplea el tiempo ocioso de las computadoras personales para realizar cómputo científico. Se considera tiempo ocioso cuando el usuario no utiliza los recursos de cómputo del ordenador. Algunos de los proyectos de cómputo voluntario más conocidos son SETI@home (Anderson et al., 2002) y Folding@home (Beberg et al., 2009).

El cómputo de alto rendimiento puede realizarse con supercomputadoras (Dowd et al., 1998), o en un concepto más amplio, con computadoras personales cuyo único objetivo es el cómputo científico. Las computadoras utilizadas para cómputo de alto rendimiento no son empleadas por usuarios para realizar tareas personales. Algunos de los clúster de alto rendimiento más conocidos son ASC (2014), Titan (2014), Blue Gene/Q (IBM Blue Gene/Q, 2014), Stampede (2014) y Piz Daint (Hpc-ch, 2014). Aunque más eficiente y confiable, el cómputo de alto rendimiento requiere inversiones iniciales mayores que el cómputo voluntario. Por ello resulta más difícil tener acceso a esta modalidad de cómputo paralelo.

En la actualidad una de las formas más usuales de acceder a un clúster es a través de la red, utilizando una comunicación directa con el mismo. En países con baja conectividad como Cuba esta comunicación con el clúster es limitada. Ante esta situación se requiere un cambio de enfoque para adaptarse a condiciones mucho más restrictivas.

En varios centros cubanos vinculados a la investigación se han realizado avances en el procesamiento de grandes volúmenes de datos (Barroso y Castillo, 2011) y (Gutiérrez et al., 2003). Sin embargo, aunque se han creado algunas soluciones internas, no existe una plataforma común para cómputo distribuido disponible a todos los centros de investigación del país. Por ello se requiere una plataforma basada en cómputo de alto rendimiento, que permita procesar grandes volúmenes de datos con alta disponibilidad y eficiencia.

La Universidad de La Habana desarrolla en estos momentos varios proyectos con grandes volúmenes de datos como (Bolufé-Röhler y Chen, 2014) que trabaja la optimización en grandes dimensiones. La mayoría forman parte de investigaciones de doctorados y maestrías. Sin embargo, en muchos casos no se cuenta con los recursos de cómputo masivos necesarios para la exitosa culminación de dichos proyectos. Bajo este contexto se propone la colaboración con otros centros que sí dispongan de los medios necesarios para realizar experimentos computacionalmente complejos.

Una de las principales dificultades son las condiciones de conectividad existentes. El uso de Internet es limitado por lo que no pueden intercambiarse grandes volúmenes de datos y su estabilidad varía en dependencia de los centros que la utilicen. En cambio, uno de los métodos con mayor disponibilidad, estabilidad y seguridad es el correo electrónico. El mismo no requiere de un ancho de banda significativo y funciona aunque los usuarios no se

mantengan conectados a la red. Por lo tanto es una solución más adecuada a las condiciones de conectividad actuales.

## **PROPÓSITO DEL TRABAJO**

La Universidad de Griffith en Australia dispone de un clúster de computadoras para cómputo de alto rendimiento, que brinda servicios a otras universidades. En este trabajo se propone la utilización de este clúster a través del correo electrónico en vez de utilizar la interfaz web.

A partir de lo anterior se propone extender este proyecto a diferentes universidades que se encuentren en países con baja conectividad a Internet. De esta forma se pueden incrementar los recursos de cómputo de las instituciones, sin la necesidad de hacer grandes inversiones. En estos momentos se desarrollan los primeros experimentos con la Universidad de La Habana. Se pretende extender el uso de la plataforma a otras instituciones, como la Universidad de Cape Coast en Ghana, África.

Este trabajo se propone proveer a la Universidad de La Habana de una plataforma para acceder a un recurso de cómputo de alto rendimiento. La plataforma está montada sobre un clúster de computadoras dedicado al cómputo científico donde se corren experimentos de distintas universidades del mundo. Como caso de estudio se analizarán los resultados obtenidos en un proyecto de optimización en grandes dimensiones realizado en la Universidad de La Habana. El propósito de este proyecto es el desarrollo de una nueva meta-heurística para optimizar funciones reales en grandes dimensiones. Debido a la complejidad computacional asociada a la evaluación de este tipo de funciones, se requiere de un elevado poder de cómputo. Hasta el momento no existían alternativas viables para realizar estas investigaciones en la Universidad de La Habana. La plataforma presentada en este trabajo abre las puertas a nuevas líneas de investigación en problemas computacionalmente complejos.

## **METODOLOGÍA**

El clúster HPC de la Universidad de Griffith está formado por 32 nodos de cómputo con el sistema operativo Fedora Core 3 y 13 nodos de cómputo con el sistema operativo Fedora Core 8, con un total de 90 unidades de procesamiento (Griffith University HPC Cluster - V20z, 2014). En este momento se brinda soporte para ejecutar aplicaciones implementadas en MATLAB. En el futuro se pretende brindar soporte a otros lenguajes utilizados por la comunidad científica.

La interfaz estándar para acceder al clúster es a través del protocolo SSH. Los usuarios autenticados en el clúster pueden ejecutar las tareas deseadas de forma directa empleando los recursos de cómputo disponibles en el clúster. Esta interfaz requiere una comunicación continua a través de Internet, por lo que resulta impracticable en entornos con escasa conectividad. Este proyecto presenta una interfaz alternativa utilizando el correo electrónico que no requiere que los usuarios dispongan de una conexión a Internet y realiza de forma transparente para el usuario el proceso de comunicación con el clúster. Para replicar la solución propuesta, la institución requiere de la creación de un usuario en el clúster HPC de la Universidad de Griffith y de un correo electrónico asociado. El correo debe cumplir las normas de seguridad para garantizar la autenticidad del autor.

Cada experimento que se desea ejecutar en el clúster se envía a través de un correo firmado digitalmente empleando el protocolo RSA. La llave pública debe ser enviada a los administradores del clúster antes de iniciar los experimentos. El cuerpo del correo consiste en un plan escrito en el lenguaje Nimrod. Este plan describe el experimento y es utilizado por el clúster para decidir cómo paralelizar las tareas. En archivos adjuntos se incluye el código fuente y los datos necesarios para realizar el experimento.

Un plan de Nimrod es un archivo con extensión .pnl que está conformado por 2 secciones, una descripción de parámetros y una descripción de tareas. Una combinación de los parámetros define una instancia del experimento que se ejecuta de manera secuencial en un nodo. El clúster determina de forma automática que parámetros se ejecutan en cada nodo y garantiza la ejecución de una instancia por cada combinación válida de los valores de los parámetros. La Figura 1 muestra un ejemplo de un plan de Nimrod.

Los experimentos deben definirse en dos niveles, el código del experimento en el lenguaje de programación correspondiente y el plan de Nimrod que define como va a paralelizarse el experimento.

### ***Parámetros:***

Los parámetros se definen como listas de valores, constantes, valores dinámicos, entre otros. La sintaxis para definir un parámetro en un plan de Nimrod es:

```
parameter <name> <type> [<domain>];
```

El nombre (name) debe ser único para el parámetro que se desee definir. La característica (type) determina el tipo de dato del parámetro. Existen 5 tipos de datos en Nimrod, que son float, integer, text, files y fromfile. Cada uno de estos tipos está representado en el plan de Nimrod utilizado como ejemplo, ver Figura 1. El dominio es un parámetro opcional que es usualmente necesario. Algunos dominios posibles son: single value, range, random, select anyof, select oneof y otros archivos de entrada. La forma de utilizarlos se muestra en el ejemplo. En caso de que ciertas combinaciones de los parámetros no sean válidas, Nimrod incluye un mecanismo para descartar combinaciones específicas de parámetros.

### ***Tareas:***

Esta parte del plan de Nimrod describe el proceso de ejecución de una instancia del experimento. Una tarea está compuesta por un conjunto de operaciones, comandos o directivas entre las que se encuentran copiar un archivo del nodo raíz hacia los nodos de cómputo (copy), ejecutar el código de un experimento (execute) o sustituir lugares vacíos en los archivos de entrada con valores reales. Las operaciones generalmente dependen de una ubicación, que puede ser node: o root:, y determinan si las operaciones se ejecutan en cada uno de los nodos donde se ejecuta cada instancia del experimento, o solamente en el nodo raíz antes de paralelizar, respectivamente.

*Figura 1: Fragmento de código de ejemplo de un plan de Nimrod*

```
parameter aircraft_model files select anyof "A3??.dat" "737-*.dat";
parameter AoA label "Angle of attack" float range from -45 to 45
step 2.5;
parameter winglets text select anyof "none" "fence" "blended"
"raked";
parameter airspeed integer range from 50 to 600 step 50;
parameter turbulence label "Normalized Reynolds" float random from
1 to 2;
task main
    copy ${aircraft_model} node:.
    copy wing_test.zip node:.
    node:execute unzip wing_test.zip
    node:execute ./run_wing_test.sh ${aircraft_model}
    ${winglets}\
        ${AoA} \
        ${airspeed} ${turbulence} >> output.${jobname}
    node:execute zip results.${jobname} *
    copy node:results.${jobname}.zip .
endtask
```

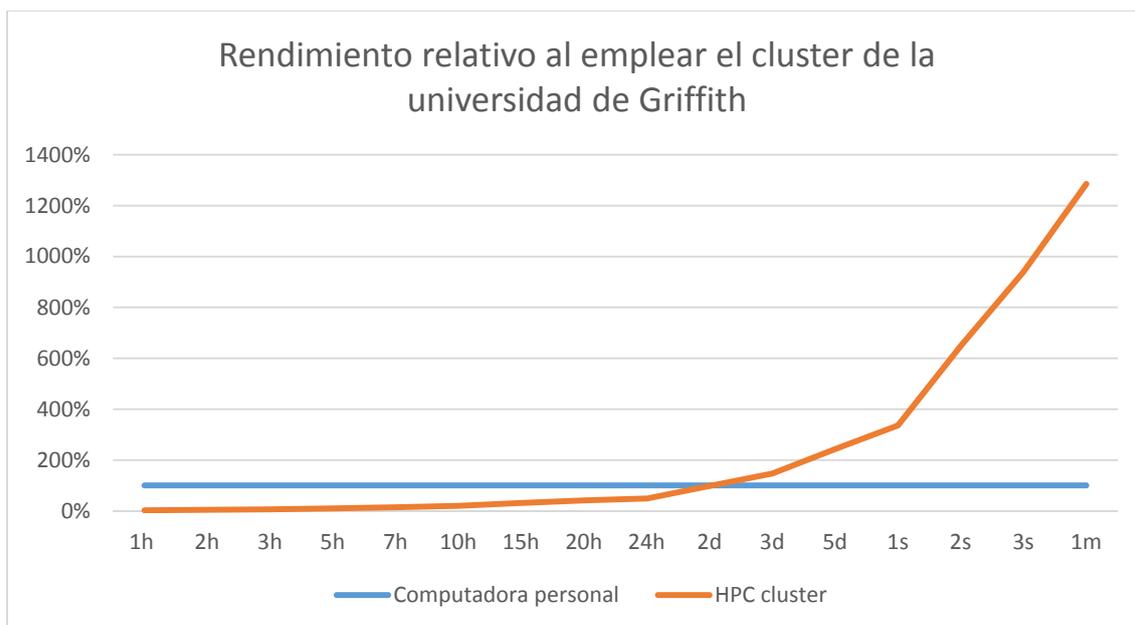
## RESULTADOS

El clúster fue puesto a prueba con diversos experimentos de un proyecto de optimización global con grandes volúmenes de datos. Dicho proyecto hace uso de las funciones de prueba del concurso de optimización en grandes dimensiones del Congreso de Computación Evolutiva de la IEEE (Li et al., 2012). Conforman el conjunto de prueba un total de 15 funciones de dimensión 1000. Según lo establecido en la competencia, para probar un algoritmo es necesario efectuar 25 corridas de cada una de las funciones. En cada corrida se debe realizar un total de 3 millones de evaluaciones de la función objetivo.

El tiempo de ejecución esperado de cada experimento supera las 200 horas en una computadora estándar (Core(TM)2 Duo CPU E8500 @3.16GHz). El diseño de nuevos algoritmos y el ajuste de sus parámetros requieren la realización de numerosos experimentos como parte del proceso de investigación. Por este motivo no es factible llevar a cabo este proyecto sin recursos de cómputo en paralelo. Para paralelizar un experimento se ejecutaron de manera independiente cada una de las corridas para cada una de las funciones, sumando un total de 375 procesos en paralelo. Los resultados obtenidos muestran una considerable disminución en el tiempo de ejecución de los experimentos.

Teóricamente el clúster debería reducir el tiempo de cómputo en un factor de 90, por lo que el tiempo de espera para una instancia del experimento sería de 5 horas. Se debe tener en cuenta que el clúster ejecuta tareas de diferentes usuarios y no todos los nodos están disponibles al mismo tiempo, por lo tanto se espera una demora adicional al tiempo de ejecución. En la práctica la demora asociada a ejecutar un experimento es de dos a tres días con la demanda actual del clúster. Incluso en este caso la reducción en el tiempo de ejecución es considerable.

**Gráfico 1: Comparación de rendimiento**



El rendimiento relativo entre la utilización del clúster y la ejecución de un experimento de forma secuencial se muestra en la Gráfica 1. Se considera un 100% de rendimiento en el tiempo de espera del algoritmo cuando este se ejecuta de forma secuencial. Como puede apreciarse después de los 2 días aparece una mejora significativa con la utilización del clúster.

## CONCLUSIONES

La utilización de recursos de cómputos masivos resulta fundamental en diversos proyectos de investigación, como por ejemplo la optimización en grandes dimensiones. Sin embargo, debido a su alto costo no siempre es posible contar con este tipo de recursos in situ. El acceso remoto constituye por tanto una solución efectiva y frecuentemente utilizada. En la mayoría de los casos la comunicación se realiza utilizando el protocolo SSH, pero esta alternativa no siempre es viable para naciones como Cuba donde el ancho de banda existente es muy bajo. Para dar solución a esta dificultad el presente trabajo presenta una alternativa basada en el correo electrónico.

Los requerimientos necesarios para este tipo de acceso son una cuenta de correo registrada en el clúster que permita firmar digitalmente los correos empleando el protocolo RSA. La comunicación se establece enviando en el cuerpo del correo código Nimrod y como adjunto el código de los experimentos a ser paralelizados. Actualmente los experimentos deben ser escritos en MATLAB, pero como parte de trabajos futuros se pretende incluir

otros lenguajes comúnmente utilizados por la comunidad científica como Python y C++. Los resultados obtenidos, al paralelizar experimentos de un alto costo computacional, muestran que posible incrementar el rendimiento relativo en aproximadamente un 1300%.

## REFERENCIAS

- Abramson, D., Foster, I., Giddy, J., Lewis, A., Sosič, R., Sutherst, R. and White, N. (1997) The Nimrod Computational Workbench: A Case Study in Desktop Metacomputing. *Proceedings of the 20th Australasian Computer Science Conference*.
- Aluru, S., Amato, N. Bader, D.A., Bhandarkar, S., Kale, L., Marinescu, D., and Samatova, N. (2006) *Parallel computational biology*. In Heroux M.A., Raghavan, P. and Simon, H.D. (ed), *Parallel Processing for Scientific Computing (Software, Environments and Tools)*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, p. 357-378.
- Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M. and Werthimer, D. (2002) SETI@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11): p. 56-61.
- ASC (2014) *Record Simulations Conducted on Lawrence Livermore Supercomputer*. <https://asc.llnl.gov/>
- Barroso, L.C. and E.S. Castillo (2011) Propuestas de Ingeniería de Software para la Computación de Alto Rendimiento. *Serie Científica*, Vol 4(3).
- Beberg, A.L., D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande (2009) Folding@home: Lessons from eight years of volunteer distributed computing. in *Parallel & Distributed Processing. IPDPS 2009. IEEE International Symposium*.
- Bolufé-Röhler A. and S. Chen (2014) Extending Minimum Population Search towards Large Scale Global Optimization. *IEEE Congress on Evolutionary Computation*. IEEE Press.
- Chen, F., Ge, W., Guo, L., He, X. Li, B., Li, J., Li, X., Wang, X. and Yuan, X. (2009) Multi-scale HPC system for multi-scale discrete simulation—development and application of a supercomputer with 1 Petaflops peak performance in single precision. *Particuology*, 7(4): 332-335.
- Dowd, K., Severance, C.R. and Loukides, M.K. (1998) *High performance computing*. vol. 2. O'Reilly.
- Egloff, D. (2011) *GPUs in financial computing part III: ADI solvers on GPUs with application to stochastic volatility*. *Wilmott mag.*, March, p. 51-53.
- Griffith University HPC Cluster - V20z (2014). Available from: <https://confluence.rcs.griffith.edu.au/display/v20zCluster/Griffith+University+v20z+Cluster+Home+Page>.
- Gutiérrez, D., González, A. and Febles J. P. (2003) El cluster BEOWULF del centro nacional de Bioinformática: diseño, montaje y evaluación preliminar. *Ingeniería Industrial*, vol. 24 (3):69-74.
- Hand, D.J. (2007) Principles of data mining. *Drug safety*, 30(7): p. 621-622.
- Hpc-ch (2014) Piz Daint, the first supercomputer with sustained petaflops-scale performance in Switzerland.
- IBM Blue Gene/Q (2014) <http://www-03.ibm.com/systems/technicalcomputing/solutions/bluegene/>
- Jackson, K.R., Ramakrishnan, L., Muriki, K., Canon S., Cholia S., Shalf, J., Wasserman H. J. and Wright N. J. (2010) *Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud*. 2nd IEEE. Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on

- Joshi, S., Pathak, R., Ahmed, S., Choudhary, K.K. and Mishra, D.K. (2009) *MPI and PVM based HPC Setup for Multi Scale Modeling*. IEEE International Advance Computing Conference (IACC'09).
- Kendall, R.A., Aprà, E., Bernholdt, D.E., Bylaska, E.J., Dupuis, M., Fann, G.I., Harrison, R.J., Ju, J., Nichols, J.A., Nieplocha, J., Straatsma, T.P., Windus, T.L., Wong, A.T. (2000) High performance computational chemistry: An overview of NWChem a distributed parallel application. *Computer Physics Communications*, 128(1): p. 260-283.
- Li, X., Tang, K., Omidvar, M.N., Yang, Z. and Qin, K. (2013) Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2012.
- Pardalos, P. and Romeijn, H.E. (2002) *Handbook of Global Optimization*, Vol. 2. Springer.
- Plaza, A., Du Q., Chang Y-L. and King R. L. (2011) Foreword to the Special Issue on High Performance Computing in Earth Observation and Remote Sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 4 (3): p. 503-507
- Rodríguez, D. (2012) *Despliegue de servicios HPC en entornos cloud multipropósito*. Informe técnico CESGA-2012-004, Fundación Centro de Supercomputación de Galicia (CESGA). Tesis de Máster, Máster en Computación de altas prestaciones.
- Shibata, M. and Uryū, K. (2002) Gravitational waves from the merger of binary neutron stars in a fully general relativistic simulation. *Progress of Theoretical Physics*, 107(2): 265-303.
- Stampede (2014) <https://www.tacc.utexas.edu/stampede/> Texas Advanced Computing Center.
- Titan (2014) Introducing Titan | The World's #1 Open Science Supercomputer. <https://www.olcf.ornl.gov/titan/>