

# Un modelo genérico para CluSim: modelado de aplicaciones pipeline y divide/conquer

Abigaíl R. N. Verazay, C. Marcelo Pérez Ibarra, Nilda M. Pérez Otero  
*Universidad Nacional de Jujuy, Facultad de Ingeniería*

## Abstract

*La demanda de poder computacional generada por las aplicaciones de alto rendimiento promovió el desarrollo de centros de cómputos de altas prestaciones. En el diseño y desarrollo de estos sistemas los simuladores resultan de gran utilidad para el desarrollo de técnicas y mecanismos de optimización de rendimiento. En este marco, el GIS desarrolló un simulador de cluster basado en OMNeT++ que permite evaluar y predecir el impacto en el rendimiento del sistema considerando distintas configuraciones de cluster. La primera versión de CluSim, sólo permitía simular la ejecución de aplicaciones paralelas bajo el paradigma master/worker. A fin de extender el modelo de simulación de CluSim para abarcar todos los paradigmas de programación paralela el GIS definió un proceso de modelado que se utilizó para modelar aplicaciones SPMD. En este artículo, se presenta el trabajo realizado para completar el modelo de CluSim con los paradigmas pipeline y divide/conquer. La aplicación del proceso de modelado a un conjunto de aplicaciones representativas de ambos paradigmas determinó que los parámetros que caracterizan el comportamiento de las aplicaciones pipeline se corresponden uno a uno con los parámetros definidos en el modelo previo. Sin embargo, la ausencia de patrones de comportamiento similares en las trazas de las aplicaciones divide/conquer seleccionadas, no permitió identificar parámetros comunes entre ellas. Como trabajos futuros, se pretende determinar las distribuciones de probabilidad asociadas a los parámetros identificados en este trabajo así como su implementación en el modelo en CluSim y su correspondiente validación.*

## Palabras Clave

Cómputo de altas prestaciones, HPC, cluster, simulación, modelado de aplicaciones, CluSim.

## Introducción

En las últimas décadas las aplicaciones de cómputo de altas prestaciones experimentaron un increíble crecimiento en varios campos tales como medicina, genética, física, procesamiento de imágenes, computación, entre otros. La demanda de poder computacional generada

por estas aplicaciones estuvo ligada estrechamente al desarrollo de centros de cómputos de altas prestaciones. A junio de 2013, según el ranking de top500.org, los clusters representan cerca del 83% de los centros de supercomputación más importantes, donde varios de ellos cuentan con cientos de miles de cores e incluso algunos que superan el millón.

En el diseño y desarrollo de estos sistemas es indispensable utilizar herramientas de simulación como soporte del modelado de rendimiento a fin de evaluar las opciones de diseño y ayudar a optimizar el rendimiento de procesadores, redes de interconexión y eventualmente el sistema completo, incluyendo el software y las aplicaciones High Performance Computing (HPC) [1].

Cada vez es más frecuente el uso de modelos de simulación computacional en HPC, ya sea como ayuda al modelado de prestaciones [1], como para explorar arquitecturas o aplicaciones [2] [3] o como una herramienta de predicción de tráfico [4].

En la literatura se encuentran muchos trabajos centrados en simular grandes redes y aplicaciones de HPC. La mayoría de los simuladores de redes están centrados en arquitecturas específicas. En [5] se presenta un simulador de SANs (Redes de Área de Almacenamiento) que permite trabajar tanto con trazas de tráfico reales como sintéticas, y simula fallos en enlaces y switches, canales virtuales, diferentes algoritmos de ruteo, etc. PARSEC [6] es un entorno de simulación de eventos discretos, que, mediante un compilador mejorado de C++ permite simular entidades y constructores de mensajes de comunicación entre entidades. SIMCAN [7] es un entorno de simulación para grandes redes complejas de almacenamiento que permite simular estas

redes y sus subsistemas subyacentes correspondientes (I/O, networking, etc.). También es posible encontrar entornos de simulación de redes de propósito general que permiten crear diferentes configuraciones de redes, con diferentes tipos de nodos, switches, topologías, protocolos, etc. Ejemplos de éstos son OPNET Modeler<sup>1</sup> y OMNeT++<sup>2</sup>.

Considerando estos antecedentes, desde el año 2010, el Grupo de Ingeniería de Software (GIS) desarrolló CluSim, un simulador de clusters basado en OMNeT++ que permite parametrizar la configuración de recursos de un cluster, los patrones de cómputo y comunicación de la aplicación paralela y el tamaño del problema, de modo que sea posible evaluar y predecir el impacto en el rendimiento del sistema considerando diferentes configuraciones (número de nodos del cluster). Este simulador, presentado en los trabajos [8] [9] [10] [11], pretende servir como herramienta de soporte a la toma de decisiones para la selección de las configuraciones más adecuadas para un cluster que ejecuta un determinado tipo de aplicación paralela.

A fin de extender la funcionalidad de CluSim, el GIS definió las etapas de un proceso de modelado que permite la formulación de un modelo genérico de aplicaciones paralelas bajo distintos paradigmas de programación, proceso que ya fue utilizado para caracterizar aplicaciones master/worker y SPMD (Single Program Multiple Data) [12] [13]. En este trabajo se presentan los resultados de aplicar dicho proceso al modelado de aplicaciones pipeline y divide/conquer.

El resto del trabajo se estructura como sigue: el proceso de modelado y las aplicaciones objetivo se describen en el apartado siguiente, a continuación se muestran los resultados obtenidos y luego se discuten. Finalmente se presentan las conclusiones y trabajos futuros.

## Elementos del Trabajo y metodología

<sup>1</sup> www.opnet.com

<sup>2</sup> www.omnetpp.org

En este apartado se describen brevemente la metodología de trabajo seguida, las aplicaciones objetivo y el entorno de experimentación.

### Proceso de Modelado

Para llevar a cabo el presente trabajo se aplicó el proceso de modelado (Figura 1) presentado en [13]. Este proceso permite generar un modelo que caracteriza una aplicación paralela identificando los parámetros que definen su comportamiento. Esto puede realizarse aplicando las técnicas de análisis de rendimiento que se muestran en la Figura 2 [14].

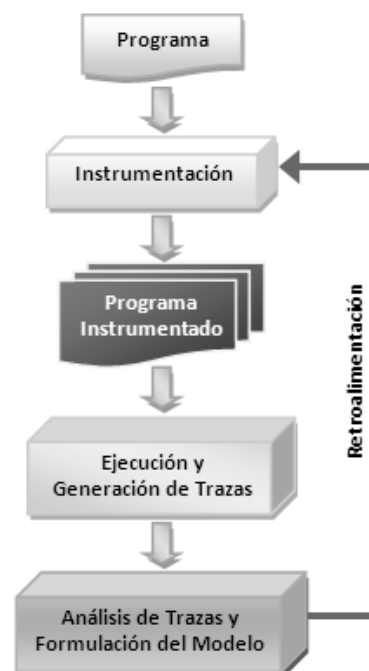


Figura 1. Proceso de modelado de aplicaciones.

En este trabajo el proceso de modelado se realiza aplicando el método accionado por eventos (las medidas se realizan cuando se activan determinados eventos) para registrar los sucesos del programa a través de la instrumentación de código implementada con la interposición de las librerías MPE de MPICH. El registro del comportamiento dinámico de la aplicación se lleva a cabo mediante trazas de ejecución (secuencia temporal de los valores de rendimiento en cada instante) para determinar las propiedades algorítmicas subyacentes a la sucesión de eventos [15] [16].

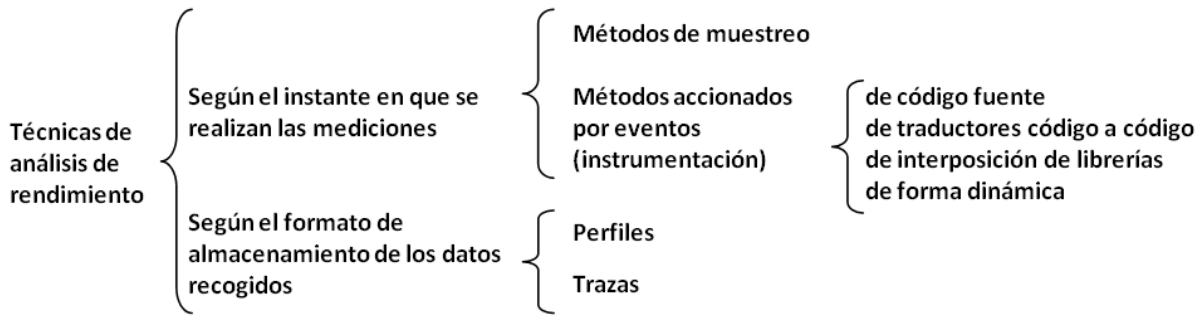


Figura 2. Clasificación de técnicas análisis de rendimiento.

### Entorno de Experimentación

El entorno de experimentación utilizado en este trabajo es el cluster del Laboratorio de Altas Prestaciones de la Facultad de Ingeniería de la Universidad Nacional de Jujuy. Este cluster está compuesto de 14 nodos con las siguientes características (software y hardware):

- procesador core I3
- 4 GB de RAM
- 500 GB de disco duro
- Placa Ethernet 10/100M
- Sistema operativo: Linux Ubuntu 12.04
- Librería de paso de mensajes: MPICH2
- Librerías adicionales: librería MPE y visualizador Jumpshot

### Aplicaciones objetivo

Para llevar a cabo el estudio abordado en este trabajo se seleccionaron dos conjuntos de aplicaciones MPI codificadas en C. El primero utiliza el paradigma pipeline para resolver la eliminación de Gauss-Jordan, la factorización de Cholesky y el problema de los n cuerpos. El segundo conjunto, correspondiente a aplicaciones divide/conquer, incluye algoritmos de cálculo de números primos, cálculo de Pi, merge-sort, alineamiento múltiple de secuencias y el fractal de Mandelbrot.

Cada una de las aplicaciones se ejecutó en 8 nodos del cluster con los parámetros que se muestran en la Tabla 1.

En el apartado siguiente se presentan los resultados obtenidos de la ejecución de las aplicaciones detalladas anteriormente.

Tabla 1. Aplicaciones objetivo y sus parámetros

	Aplicación	Parámetros Requeridos	Aplicación en C
Pipeline	Eliminación Gauss-Jordan		ge_pipe.c
	Factorización de Cholesky	Tamaño del vector: 1000 Tamaño del bloque: 100	cholesky.c
	Problema de n cuerpos	n: 1000	n-body.c
Divide/Conquer	N° Primos		primes.c
	Merge-Sort	1024 datos	mergesort.c
	Alineamiento Múltiple de Secuencias		clustalw.c
	Fractal de Mandelbrot	x real: (-2, 2) x imag.: (-2, 2)	fractal.c
	Cálculo de Pi por el método de la integral	32 intervalos	pi_calc.c

### Resultados

Como resultado de la experimentación se obtuvieron trazas de ejecución para cada una de las aplicaciones objetivo. Los archivos de traza fueron procesados utilizando Jumpshot, el visor de trazas incluido en la librería MPE.

Las figuras 3, 4 y 5 presentan segmentos correspondientes a trazas de las aplicaciones de tipo pipeline, mientras que en las figuras 6 a 10 se muestran segmentos de trazas para las aplicaciones divide/conquer.

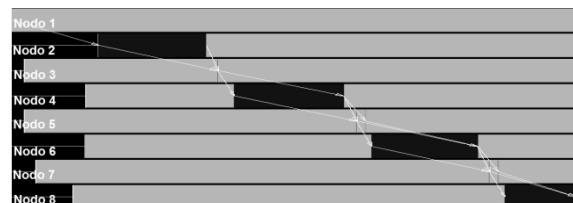


Figura 3. Segmento de una traza de ge\_pipe.c.

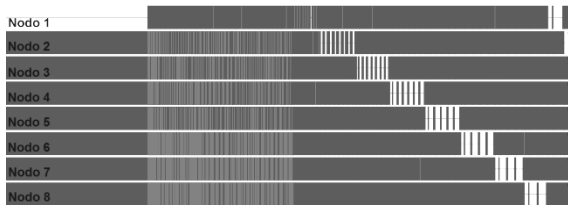


Figura 4. Segmento de una traza de *cholesky.c*.

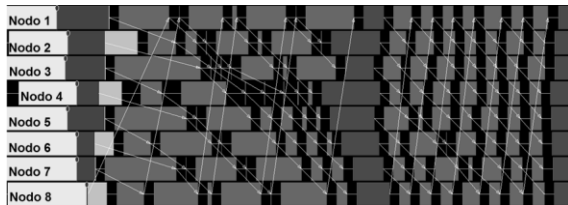


Figura 5. Segmento de una traza de *n-body.c*.

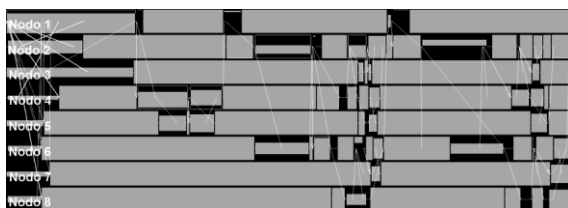


Figura 6. Segmento de una traza de *clustalw.c*.

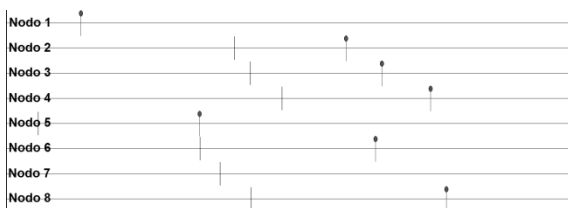


Figura 7. Segmento de una traza de *primes.c*.

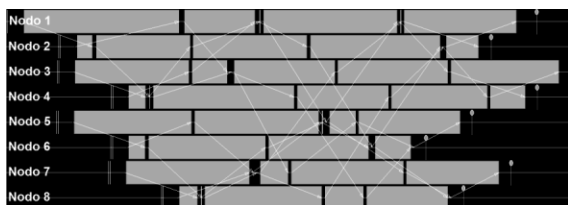


Figura 8. Segmento de una traza de *mergesort.c*.

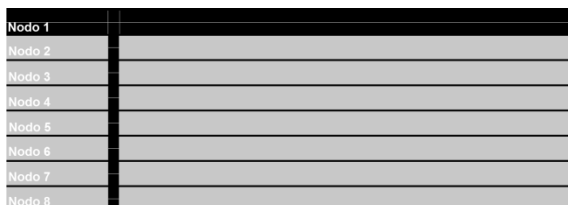


Figura 9. Segmento de una traza de *pi\_calc.c*.

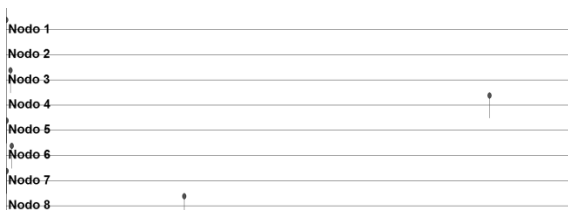


Figura 10. Segmento de una traza de *fractal.c*.

## Discusión

La formulación del modelo requiere:

1. la identificación de los parámetros asociados a los patrones de comportamiento de las aplicaciones y
2. la determinación de las distribuciones de probabilidad que siguen dichos parámetros.

En este apartado se discuten por separado los resultados correspondientes a cada uno de los paradigmas analizados

### Aplicaciones pipeline

El análisis de la representación gráfica de las trazas permitió identificar seis parámetros. Estos parámetros se corresponden con aquellos definidos para las aplicaciones master/worker y SPMD [13]. Esta correspondencia se muestra en la tabla 2.

Con respecto a las distribuciones de probabilidad que siguen los parámetros para cada tipo de aplicación, actualmente se trabaja en el procesamiento de los datos numéricos correspondientes a las trazas de ejecución generadas.

### Aplicaciones divide/conquer

Como puede observarse en las figuras 6 a 10, las aplicaciones que siguen este paradigma no presentan patrones similares de comportamiento. Por lo tanto, siguiendo el modelo planteado anteriormente, resulta difícil la identificación de parámetros que caractericen a las aplicaciones que siguen este paradigma.

## Conclusiones y trabajos futuros

En este artículo se presentó el modelado de aplicaciones pipeline y divide/conquer. Para ello se seleccionaron aplicaciones representativas de estos paradigmas que se instrumentaron utilizando la librería MPE de MPICH. Luego, en los escenarios de experimentación establecidos, se ejecutaron las distintas aplicaciones para generar las trazas de ejecución correspondientes. Las trazas así obtenidas se analizaron para identificar patrones de comportamiento y

Tabla 2. Correspondencia entre parámetros y tiempos de aplicación master/worker, SPMD y Pipeline.

Parámetro	Tiempos master/worker	Tiempos SPMD	Tiempos Pipeline
t_cómputo_inicial	Preparación para enviar trabajo a los workers	Preparación para difusión inicial	Preparación para distribución inicial
t_computo_final	Compilación de resultados	Reducción final	Reducción de resultados
t_nodos	tiempo entre el envío de tareas a los workers	Igual a 0	Igual a 0
t_computo_nodo	Cómputo del worker	Cómputo del nodo	Cómputo del nodo
t_envío_trabajo	Envío de trabajo al worker	Difusión	Envío de trabajo / difusión
t_envío_resultado	Devolución de resultados al máster	Reducción / Difusión de resultados parciales	Envío de resultados / difusión

consecuentemente determinar los parámetros asociados a ellos. En el caso de las aplicaciones pipeline fue posible identificar un comportamiento común al conjunto estudiado y extender el modelo de aplicación existente para este tipo de aplicación. Sin embargo, en el caso de las aplicaciones divide/conquer los resultados obtenidos no presentan un patrón común de comportamiento que permita asimilar este tipo de aplicación a los parámetros definidos para los otros paradigmas.

A partir de estos resultados se prevé que los trabajos futuros incluirán:

- la determinación de las distribuciones de probabilidad de los parámetros del paradigma pipeline,
- la implementación del modelo extendido en CluSim y
- la validación del simulador.

## Referencias

1. Denzel, W.E.; Li, J.; Walker, P. and Y. Jin. A framework for end-to-end simulation of high-performance computing systems. In Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, pp. 1--10, ICST, Brussels, Belgium, Belgium. ICST. (2008)
2. Hammond, S.D.; Mudalige, G.R.; Smith, J.A.; Jarvis, S.A.; Herdman, J.A. and A. Vadgama. Warpp: a toolkit for simulating highperformance parallel scientific codes. In Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, pp. 1--10, ICST, Brussels, Belgium, Belgium. ICST. (2009)
3. Minkenber, C. and Rodriguez, G. Tracedriven co-simulation of high-performance computing systems using OMNeT++. In Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, pp. 1--8, ICST, Brussels, Belgium, Belgium. ICST. (2009)
4. Tikir, M.M.; Laurenzano, M.A.; Carrington, L. and A. Snavely. PSINS: An open source event tracer and execution simulator for MPI applications. In Euro-Par '09: Proceedings of the 15th International Euro-Par Conference on Parallel Processing, pp. 135--148, Berlin, Heidelberg. Springer-Verlag. (2009)
5. Molero, X.; Silla, F.; Santonja, V. and J. Duato. Modeling and simulation of storage area networks. Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proceedings. 8th International Symposium on , vol., no., pp.307,314. (2000)
6. Bagrodia, R.; Meyer, R.; Takai, M., Chen, Y., Zeng, X.; Martin, J.; Park, B. and H. Song. Parsec: A Parallel Simulation Environment for Complex Systems. Computer, Vol. 31(10), October 1998, pp. 77-85. (1998)
7. Núñez, A.; Fernández, J.; García, J.D.; García, F. and J. Carretero. New techniques for simulating high performance MPI applications on large storage networks. *J. Supercomput.* 51, 1 (January 2010), 40-57. (2010)
8. Pérez Ibarra, C.M., Valdiviezo, L.M., Pérez Otero, N.M., Liberatori, H.P., Rexachs, D., Luque, E. y C.M. Lasserre. CLUSIM: Simulador de Clusters para Aplicaciones de Cómputo de Altas prestaciones basado en OMNeT++. XVI Congreso Argentino de Ciencias de la Computación., Buenos Aires (2010).
9. Valdiviezo, L.M.; Pérez Otero, N.M.; Pérez Ibarra, C.M. y C.M. Lasserre. Caracterización de una aplicación paralela con distintas configuraciones en CluSim. Investigaciones en Facultades de Ingeniería del NOA – 2010. ISSN 3367-5072. Ed. EdiUNJu. S. S. de Jujuy. pp. 499-504. (2010)
10. Lasserre, C.M.; Pérez Ibarra, C.M.; Valdiviezo, L.M.; Verazay, A.R.N.; Quispe, G.L.; Nolasco, S.A.; Chosco, V.H.; y N.M. Pérez Otero. Adaptación de CluSim a cluster heterogéneos. ISSN 1853-7871. Editorial Científica Universitaria. Catamarca, Argentina. (2011)
11. García, A.; Pérez Otero, N.M.; Pérez Ibarra, C.M. y C.M. Lasserre. Simulación de Clusters: Integración de INET a CluSim. XVII Congreso

- Argentino de Ciencias de la Computación. La Plata, Buenos Aires. (2011)
12. Lasserre, C.M.; García, A.; Pérez Otero, N.M.; Verazay, A.R.N.; Pérez Ibarra, M.; Nolasco S.A. y J.G. Martínez. Hacia un Modelo Genérico de Aplicaciones Paralelas. XVIII Congreso Argentino de Ciencias de la Computación (CACIC 2012). pp. 296-305. ISBN 978-987-1648-34-4. Universidad Nacional del Sur. Bahía Blanca. (2012)
  13. García, A.; Lasserre, C.M.; Verazay, A.R.N.; Pérez Otero, N.M.; Pérez Ibarra, M.; Martínez, J.G. y S.A. Nolasco. Modelado de Aplicaciones SPMD. VIII Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA. Ed. Grupo Loza Impresiones S.R.L. ISSN 1853-7871 San Miguel de Tucumán. (2012)
  14. Rodríguez Martínez, D.: Modelado Analítico del Rendimiento de Aplicaciones en Sistemas Paralelos. Tesis Doctoral. Departamento de Electrónica e Computación. Universidad de Santiago de Compostela. (2011)
  15. Blanco, V. Análisis, predicción y visualización del rendimiento de métodos iterativos en HPF y MPI. Tesis Doctoral. Departamento de Electrónica e Computación. Universidad de Santiago de Compostela. (2002).
  16. Grosclaude, E. Caracterización de Aplicaciones de Paso de Mensajes con Técnicas de Caja Negra. Trabajo Final de Especialidad de Cómputo de Altas Prestaciones. Universidad Nacional de La Plata. Buenos Aires. (2012)

**Datos de Contacto:**

*Abigail R. N. Verazay. Facultad de Ingeniería. Universidad Nacional de Jujuy. Ítalo Palanca 10. [abigailrn@gmail.com](mailto:abigailrn@gmail.com)*