

# Extensión Funcional de CluSim para Tolerancia a Fallos

Nilda M. Pérez Otero<sup>1</sup>, Luis M. Valdiviezo<sup>1</sup>, C. Marcelo Pérez Ibarra<sup>1</sup>, Cecilia M. Lasserre<sup>1</sup>, Germán Montejano<sup>2</sup>

<sup>1</sup>Grupo de Ingeniería de Software/Facultad de Ingeniería/Universidad Nacional de Jujuy  
Avda. Italia esq. Avda Martiarena  
+54-388-4221591

<sup>2</sup>Departamento de Informática/Universidad Nacional de San Luis  
{nilperez, cmperezi, classerre, hliberatori, mlvaldiviezo}@fi.unju.edu.ar, gmonte@unsl.edu.ar

## RESUMEN

El rendimiento de un sistema paralelo depende tanto de la configuración de recursos de la computadora, como del tipo de aplicación a ejecutar. Por ello, el estudio del comportamiento de aplicaciones paralelas en clusters, considerando diferentes configuraciones, permitiría establecer qué configuración resulta más adecuada de acuerdo al tipo de aplicación. Además, otro factor a considerar es la ocurrencia de fallos en el sistema paralelo y la presencia de un sistema robusto de tolerancia a fallos que mantenga disponible la aplicación. El estudio de rendimiento de estos sistemas no es fácil, ya que implica detener el cluster en funcionamiento y reconfigurar sus recursos para llevar a cabo la evaluación de rendimiento o inyectar fallos en el cluster a fin de analizar el comportamiento del sistema tolerante a fallos; todo esto con el consecuente impacto en tiempo y costo. CluSim es un simulador de cluster para aplicaciones paralelas, basado en OMNeT++, que permite parametrizar la configuración de un cluster de modo que sea posible evaluar y predecir el impacto en el rendimiento de diferentes configuraciones para distintos tipos de aplicaciones. En la actual línea de investigación se pretende extender la funcionalidad del simulador CluSim para incluir aspectos de tolerancia a fallos.

**Palabras clave:** Simulador, OMNeT++, Clusters, Cómputo de Altas Prestaciones.

## CONTEXTO

La actual línea de investigación está inserta en el proyecto *Sistemas de Cómputo de*

*Altas Prestaciones y Alta Disponibilidad: Evaluación de la Performance en Diferentes Configuraciones.*

El equipo de trabajo del proyecto está conformado por:

- Directora: MSc. Cecilia Lasserre
- Co Directora: Ing. Nilda Pérez Otero
- Master Sandra Mendez
- Ing. Marcelo Pérez Ibarra
- Ing. Héctor Liberatori
- Luis Valdiviezo

El proyecto está acreditado y financiado por la Secretaria de Ciencia y Técnica y Estudios Regionales de la Universidad Nacional de Jujuy (SECTER-UNJu) y se inserta en el programa de incentivos al docente-investigador bajo el código 08-D0093. El proyecto está asesorado por los doctores Emilio Luque Fadón y Dolores Isabel Rexachs del Rosario, pertenecientes a la Escuela Técnica Superior de Ingeniería, Dpto. Arquitectura de Computadoras y Sistemas Operativos de la Universidad Autónoma de Barcelona, quienes además dirigen la tesis doctoral de la Master Sandra Méndez. También se cuenta con la colaboración del Mg. Germán Montejano (Universidad Nacional de San Luis) y la Dra. Jussara de Marques Almeida (Universidad Federal de Minas Gerais) como co-directores de la tesis de maestría de la Ing. Nilda Pérez Otero.

## 1. INTRODUCCIÓN

Los sistemas de Cómputo de Altas Prestaciones (*High Performance Computing, HPC*) se aplican en multitud de campos, incluyendo física nuclear, simulación de accidentes, procesamiento

de datos de satélites, dinámica de ruidos, modelado del clima, bioinformática y modelado financiero. La gran variedad de organizaciones científicas, comerciales y gubernamentales que hacen uso de estos sistemas ilustra el creciente predominio e impacto de las aplicaciones de HPC en la sociedad moderna [Carver, 2007].

Actualmente el uso del cómputo masivo se está introduciendo en aplicaciones comerciales como el correo electrónico, servidores web, bases de datos, etc.; lo que supone el arranque definitivo de los entornos distribuidos de cómputo en la resolución de problemas de diversa índole. Esto se ha conseguido gracias a la utilización de entornos cluster, multi-cluster y grid como paradigma de resolución de problemas en cómputo de altas prestaciones.

Un cluster es un sistema de procesamiento paralelo compuesto por un conjunto de computadoras que, interconectadas por red y software especial, cooperan para llevar a cabo el procesamiento masivo [De Giusti *et al.*, 2007]. La configuración de un cluster requiere un cuidadoso diseño de su arquitectura y funcionalidad, de modo que se garantice la eficiente ejecución de una aplicación paralela, en un entorno con un gran número de nodos y por un largo período de tiempo. Además es necesario considerar que la probabilidad de ocurrencia de un fallo crece al aumentar los recursos y/o el tiempo de procesamiento; y tal ocurrencia puede derivar en la pérdida total del trabajo realizado [Pradhan, 1996] [Weissman, 1999]. Por ello, un cluster debe mantener en todo momento la prestación de servicios encubriendo los fallos que puedan producirse. Esto hace necesario utilizar técnicas de tolerancia a fallos [Avizienis, 1997] que permitan garantizar (confiabilidad) Fiabilidad (*Reliability*), Disponibilidad (*Availability*) y Facilidad de mantenimiento (*Serviceability*) del sistema.

Evidentemente es necesario utilizar alguna técnica o mecanismo que proteja al sistema

mediante la prevención (generar redundancia), detección (monitorización), diagnóstico (latencia del error), recuperación y reconfiguración del sistema (retornar el sistema a condiciones operativas razonables después un fallo) como un todo [Avizienis *et al.*, 2004]. Estas tareas se implementan a través de un sistema de tolerancia a fallos.

RADIC es una arquitectura tolerante a fallos para sistemas de paso de mensajes que provee alta disponibilidad. Esta arquitectura es escalable, flexible, transparente y descentralizada, y tiene por objetivo asegurar que una aplicación paralela-distribuida finalice correctamente aún si ocurren fallos en algunos nodos de la computadora paralela [Duarte *et al.*, 2006]. RADIC establece un modelo que define la interacción de la arquitectura tolerante a fallos y la estructura de la computadora paralela, implementando una capa entre el nivel de paso de mensajes y la estructura de la computadora. La operación de RADIC se basa en el mecanismo de *rollback-recovery* con protocolo log pesimista. En este protocolo, se realizan *checkpoints* regularmente y todos los mensajes recibidos por cada proceso de la aplicación deben ser registrados por el receptor.

Como sistema transparente de tolerancia a fallos, RADIC realiza automáticamente una serie de actividades requeridas por el protocolo de *rollback-recovery*. Es por ello que cada actividad de RADIC se encuentra enmarcada en uno de los cuatro procedimientos generales (fases funcionales) de un mecanismo transparente de tolerancia a fallos:

- Protección: salvar el estado
- Detección de Fallos
- Recuperación
- Reconfiguración y Enmascaramiento de Fallos

Las actividades de RADIC se suceden concurrentemente con la ejecución de la aplicación paralela y no interfieren en sus resultados.

La operación de RADIC tiene dos entidades que trabajan juntas: observadores y protectores. Cada nodo del computador paralelo tiene un protector dedicado y hay un observador dedicado por cada proceso de la aplicación paralela. Estas entidades realizan el log pesimista de mensajes basado en receptor y el *checkpointing* no coordinado de manera de recuperar la aplicación paralela de un fallo, y un mecanismo de *heartbeat/watchdog* para detectar fallos.

RADIC presenta dos niveles de protección. El nivel básico provee sus funcionalidades sin la necesidad de recursos pasivos usando algunos de los nodos activos de la configuración actual para recuperar cualquier proceso fallido que pudiera degradar el rendimiento. Esto resulta adecuado para aplicaciones de corto tiempo de ejecución, o bien, aplicaciones que puedan tolerar pérdida de recursos, como aquellas con balanceo de carga dinámico. El segundo nivel de protección está destinado a aplicaciones que demandan un comportamiento non-stop. En este nivel, RADIC proporciona una redundancia dinámica flexible a través del manejo transparente de nodos *spare* [Duarte, 2007].

## 2. LINEAS DE INVESTIGACION y DESARROLLO

Las líneas de investigación del proyecto original son:

- Definir la configuración adecuada de tolerancia a fallos para diferentes tipos de aplicaciones, teniendo en cuenta los requerimientos de rendimiento y prestaciones del usuario.
- Definir un modelo genérico de aplicación-prestación-tolerancia a fallos.
- Validar experimentalmente el modelo.

La actual línea del proyecto está dirigida a:

- Simulación de RADIC usando el entorno de simulación OMNeT++

## 3. RESULTADOS OBTENIDOS/ ESPERADOS

El proyecto, previsto para el período 2009-2011, obtuvo los siguientes resultados en sus 2 primeros años:

- Instalación y configuración de un cluster homogéneo.
- Obtención de trazas de aplicaciones paralelas tipo Master/Worker ejecutadas sobre el cluster real.
- Desarrollo de un simulador de clusters (CluSim) para aplicaciones HPC basado en OMNeT++ [Pérez Ibarra *et al.*, 2010].
- Simulación de una aplicación Master/Worker considerando diferentes configuraciones de cluster [Valdiviezo *et al.*, 2010].
- Configuración de la arquitectura tolerante a fallos RADIC sobre el cluster.

En este último año se prevé obtener los siguientes resultados:

- Obtención de trazas de aplicaciones paralelas ejecutadas sobre el cluster real con RADIC y sin fallos.
- Obtención de trazas de aplicaciones paralelas ejecutadas sobre el cluster real con RADIC y con fallos.
- Extensión funcional de CluSim para incluir la arquitectura tolerante a fallos RADIC.

La versión actual de CluSim permite parametrizar la configuración de un cluster y los patrones de cómputo y comunicación de una aplicación paralela para evaluar y predecir el rendimiento del sistema paralelo. El objetivo de este año es ampliar la funcionalidad de CluSim de modo que sea posible:

- crear y probar nuevas políticas en sistemas de altas prestaciones que no están físicamente disponibles;
- predecir la escalabilidad de aplicaciones y clusters.
- analizar el comportamiento del sistema (desbalanceo de carga, cuellos de botellas causados por fallos) mediante la inyección de diferentes

patrones de fallos; y dar soporte al proceso de toma de decisiones respecto a la inclusión y configuración de sistemas tolerantes a fallos.

#### 4. FORMACION DE RECURSOS HUMANOS

El equipo de trabajo dedicado a la actual línea de investigación está integrado por:

- Ing. Nilda Pérez Otero (Directora)
- MSc. Cecilia Lasserre
- Ing. Marcelo Pérez Ibarra
- Luis Valdiviezo

Tesis de Posgrado y Tesinas de Grado:

- Doctorado en curso: 1
- Tesis de Maestría aprobadas: 1
- Tesis de Maestría en curso: 1
- Trabajo final de Especialización en curso: 2
- Tesinas en curso: 1

#### 5. REFERENCIAS

[Avizienis *et al.*, 2004] Avizienis A., Laprie J. and Randel B. & Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing [Journal] // *IEEE Transactions on Dependable and Secure Computing* v 1, n.1. - 2004. - pp. 11-33.

[Avizienis, 1997] Avizienis A. Toward systematic design of fault-tolerant systems [Journal] / *IEEE, Computer* v.30, Iss.4. 1997. pp. 51-58.

[Carver, 2007] Carver, Jeffrey C. Third international workshop on software engineering for high performance computing (hpc) applications. In *ICSE COMPANION '07: Companion to the proceedings of the 29th International Conference on Software Engineering*, page 147, Washington, DC, USA, 2007. IEEE Computer Society.

[De Giusti *et al.*, 2007] De Giusti, A. E., R. M. Naiouf, L. C. De Giusti, F. Chichizola, M. Denham, I. Rodriguez, A. Pousa, J. E.

Pettoruti, D. Montezanti, D. Encinas, L. Iglesias, H. Villagarcía Wanza, M. Iglesias. *Algoritmos Paralelos sobre Arquitecturas Multicluster y GRID*. IX Workshop de Investigadores en Ciencias de la Computación (WICC). Chubut. Mayo 2007.

[Duarte *et al.*, 2006] Duarte, A., D. Rexachs y E. Luque. Increasing the cluster availability using RADIC. In *CLUSTER*. 2006.

[Duarte, 2007] Duarte, A. *RADIC: A powerful fault tolerance architecture*. PhD thesis, Department d'Arquitectura de Computadors i Sistemes Operatius. Universitat Autònoma de Barcelona, <http://www.tdx.cat/TDX-1126107-101-303>, Mayo 2007.

[Pérez Ibarra *et al.*, 2010] Pérez Ibarra, C. M., L. M. Valdiviezo, N. M. Pérez Otero, H. P. Liberatori, D. Rexachs, E. Luque 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 (Morón). Octubre 2010.

[Pradhan, 1996] Pradhan, D. K. *Fault tolerance System Design*. New Jersey. Prentice Hall. 1996.

[Valdiviezo *et al.*, 2010] Valdiviezo, L. M., N. M. Pérez Otero, C. M. Pérez Ibarra y C. M. Lasserre. *Caracterización de Aplicaciones Paralelas para CluSim*. VI Jornadas de Ciencia y Tecnología de las Facultades de Ingeniería del NOA. Jujuy. Noviembre 2010.

[Weissman, 1999] Weissman J.B. Fault Tolerant Computing on the Grid: What are My Options [Conference] // *High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on*. 1999.