

Simulación de Clusters: Integración de INET a CluSim

Adelina García¹, Nilda M. Pérez Otero¹, C. Marcelo Pérez Ibarra¹, Cecilia M. Lasserre¹

¹ Grupo de Ingeniería de Software, Facultad de Ingeniería,
Universidad Nacional de Jujuy, Argentina
{agarcia, nilperez, cmperezi, classerre}@fi.unju.edu.ar

Abstract. El rendimiento de un sistema paralelo depende tanto de la configuración de recursos de la computadora paralela, como del tipo de aplicación a ejecutar. 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. 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 a fin de analizar el comportamiento del sistema, 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 este trabajo se presentan los resultados de incorporar el framework INET a CluSim.

Keywords: cluster, simulación, OMNeT++, INET, CluSim.

1 Introducción

El estudio y resolución de problemas cuyos modelos son muy complejos o manipulan enormes volúmenes de datos demandan un creciente poder de cómputo. El procesamiento paralelo o computación paralela de alto rendimiento proporciona el entorno tecnológico apropiado que permite dar solución a estos problemas.

Los clusters de computadoras implementan el cómputo masivo a través de una colección de equipos independientes que interconectados trabajan juntos como un único recurso de computación [1]. El avance de diferentes tecnologías (microprocesadores multinúcleo, redes de alta velocidad, bibliotecas e interfaces para programación de procesos, etc.) hicieron posible el surgimiento de estos sistemas.

La política de reparto y descentralización de recursos de los clusters proporciona una mayor capacidad de proceso, menores tiempos de procesamiento y aprovechamiento de la escalabilidad potencial de los recursos.

Considerando esto, 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.

Consecuentemente, el rendimiento de un sistema paralelo depende tanto de la configuración de recursos de la computadora, como del tipo de aplicación a ejecutar. El estudio del comportamiento de aplicaciones paralelas en clusters, considerando diferentes configuraciones, puede permitir establecer qué configuración resulta más adecuada de acuerdo al tipo de aplicación.

El estudio de rendimiento de estos sistemas no es fácil de realizar ya que implica detener un cluster en funcionamiento y reconfigurar sus recursos para llevar a cabo la evaluación de rendimiento y analizar el comportamiento del sistema, con el consecuente impacto en tiempo y costo.

En virtud de lo expuesto, el Grupo de Ingeniería de Software de la Facultad de Ingeniería de la Universidad Nacional de Jujuy desarrolló CluSim, un simulador basado en OMNeT++, que permite parametrizar la configuración de un cluster y los patrones de cómputo y comunicación de una aplicación paralela [2]. A través de este simulador se pretende evaluar y predecir el rendimiento del sistema paralelo considerando diferentes configuraciones para distintos tipos de aplicaciones paralelas. Las versiones iniciales de CluSim utilizan un componente ad hoc (simplificado) para emular el comportamiento de la red de comunicaciones sobre la que se modela el cluster [3]. En este trabajo se presenta la incorporación del framework INET al modelo de simulación de CluSim, y los primeros resultados experimentales de esta nueva versión de CluSim.

2 OMNeT++

OMNeT++ es una herramienta pública de modelado y simulación, basada en componentes modulares y con un entorno de simulación de arquitectura abierta y con un fuerte soporte de GUI, creado por Andrés Varga en el año 2001 en la Universidad Técnica de Budapest. Es un entorno de simulación discreto [4].

Su principal área de aplicación es la simulación de redes de comunicaciones, y debido a que su arquitectura es genérica y flexible, se utilizó exitosamente en el modelado de redes cableadas e inalámbricas, modelado de protocolos, evaluación de aspectos rendimiento en sistemas software complejos, validación de arquitecturas de hardware, etc.

OMNeT++ provee una arquitectura modular basada en componentes (módulos) programados en C++, que se pueden ensamblar para obtener componentes y modelos más complejos utilizando un lenguaje de alto nivel (*NED*). Aunque OMNeT++ no es un simulador en sí, actualmente está ganando popularidad como una plataforma de simulación de redes tanto en la comunidad científica como en la industria.

OMNeT++ proporciona las herramientas básicas para realizar simulaciones, pero por sí mismo no brinda ningún componente específico para la simulación de redes de computadoras, simulaciones de colas, simulaciones de arquitectura de sistemas o cualquier otra área. Lo que OMNeT++ proporciona es una librería de clases C++ que permite la creación de componentes de simulación como módulos simples y canales; también, proporciona la infraestructura para reunir las simulaciones de estos componentes y configurarlos en el lenguaje *NED* o en archivos tipo *ini*. Además proporciona interfaces en las que se puede observar y manipular el tiempo de

ejecución o entornos para la simulación como *Tkenv*, *Cmdenv*; herramientas para facilitar la creación de simulaciones y evaluación de resultados.

3 INET

INET es un framework de simulación de redes para el entorno OMNeT++. Este framework permite simular redes (802.11, Ethernet, TCP, IP, IPv6, OSPF, MPLS, RSVP, y otros protocolos) ad-hoc, cableadas e inalámbricas [6]. INET proporciona:

- Modelos bien documentados y claramente programados que permiten experimentar con protocolos y varios tipos de arquitecturas
- Modelos que implementan el software de redes del mundo real, tales como el demonio de enrutamiento Quagga o la implementación de AODV-UU, lo que garantiza la precisión de la simulación.
- Modelos de protocolos existentes que pueden combinarse libremente para formar servidores y dispositivos de red.

Al estar basado en OMNeT++, INET utiliza el mismo concepto: módulos que se comunican por paso de mensajes. Los servidores, routers, switches y otros dispositivos de red están representados en OMNeT++ por módulos compuestos. Estos módulos están ensamblados con módulos simples que representan protocolos, aplicaciones y otras unidades funcionales. En OMNeT++ una red es un módulo compuesto que contiene servidores, routers, y otros módulos. Las interfaces externas de los módulos están descritas en archivos NED. Estos archivos describen los parámetros y compuertas de módulos, submódulos y conexiones de módulos compuestos.

Los módulos de INET se organizan en paquetes jerárquicos (en un árbol de carpetas como los paquetes de Java), de acuerdo a las capas del modelo OSI. En el nivel más alto de la jerarquía se incluyen los paquetes: *inet.applications*, *inet.transport*, *inet.networklayer* e *inet.linklayer*. Otros paquetes incluidos son: *inet.base*, *inet.util*, *inet.world*, *inet.mobility* e *inet.nodes*. Estos paquetes corresponden a las capas de aplicación y transporte modeladas en INET. El paquete *inet.nodes* contiene varios servidores preensamblados, routers, switch, puntos de acceso y otros módulos, por ejemplo, *StandardHost*, *Router* y *EtherSwitch*, y *WirelessAP*. En la jerarquía los niveles más bajos, en general, corresponden a protocolos específicos o familias de protocolos.

Los módulos de INET contienen opciones de personalización lo que permite generar modelos que se ajusten a cada situación. Además INET brinda al desarrollador la posibilidad de crear sus propios componentes para escenarios de simulación particulares.

4 Experimentación y Resultados

Partiendo del supuesto que la integración de INET a CluSim permitirá emular con mayor precisión las características de un cluster real se ejecutaron una serie de experimentos sobre los siguientes escenarios (Tabla 1):

Tabla 1. Escenarios de experimentación.

| Cluster | Aplicación Paralela | Sistema Operativo y Librerías MPI |
|--|--|-----------------------------------|
| Heterogéneo Red cableada 2 Nodos | Producto de Matrices con Asignación Dinámica (100x100 hasta 2400x2400) | Ubuntu 10.04 MPICH2 y MPE |
| Heterogéneo Red cableada 4 Nodos | Producto de Matrices con Asignación Dinámica (100x100 hasta 2400x2400) | Ubuntu 10.04 MPICH2 y MPE |
| Heterogéneo Red cableada 8 Nodos | Producto de Matrices con Asignación Dinámica (100x100 hasta 2400x2400) | Ubuntu 10.04 MPICH2 y MPE |

Como puede observarse en la tabla precedente, la experimentación se realizó sobre un cluster heterogéneo (considerando 2, 4 y 8 nodos) ejecutando aplicaciones paralelas tipo Master/Worker (producto de matrices con asignación dinámica) y utilizando las librerías MPICH2 y MPE en equipos con sistema operativo Ubuntu 10.04. Por cada experimento generaron las trazas de ejecución (usando la librería MPE) y se registraron tiempos de cómputos, tiempos de comunicación y tiempo de ejecución. El análisis de las trazas de ejecución (patrones de cómputo y comunicación de la aplicación) determinó que los tiempos de cómputo registrados siguen una distribución Weibull con parámetros forma igual a 0,470358 y escala igual a 930,568 (figura 1); en tanto que los tiempos de comunicación siguen una distribución Weibull con parámetros forma igual a 0,743210 y escala igual a 240,433 (figura 2).

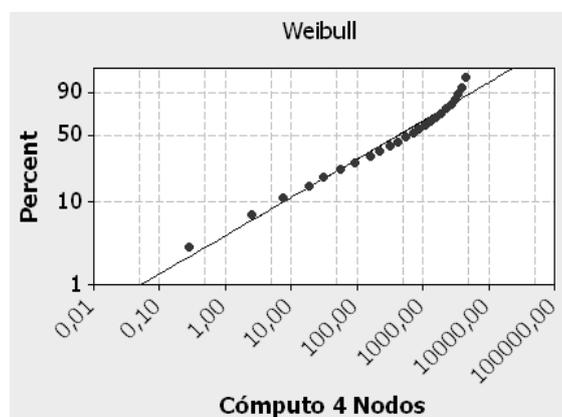


Figura 1. Análisis de Distribución de los tiempos de cómputo del cluster con 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,988.

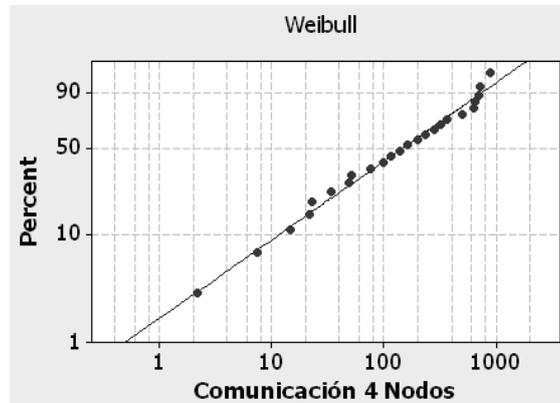


Figura 2. Análisis de Distribución de los tiempos de comunicación del cluster con 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,995.

Respecto a los tiempos de ejecución (figura 3) se estableció que éstos corresponden a una distribución Weibull con parámetro forma igual a 0,567307 y escala igual a 311,422.

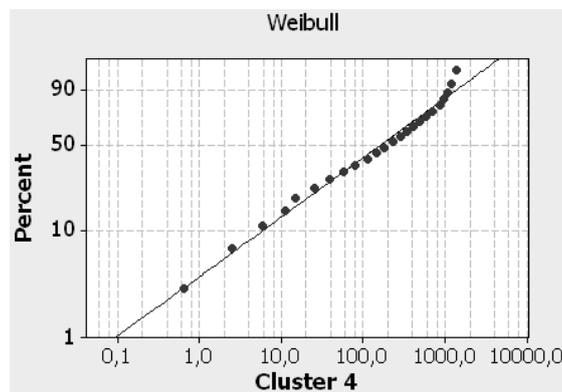


Figura 3. Análisis de Distribución de los tiempos de ejecución del cluster con 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,993.

Con estos datos se ajustaron los parámetros de CluSim y las opciones personalizables de los componentes INET integrados al simulador. Se ejecutaron las simulaciones correspondientes a cada escenario de experimentación y se obtuvieron siguientes resultados:

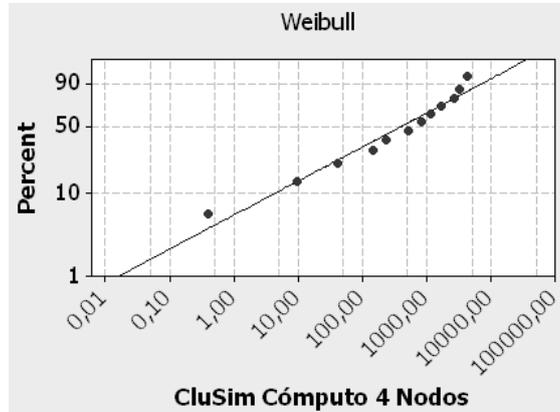


Figura 4. Análisis de Distribución de los tiempos de cómputo generados por CluSim configurado para 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,980.

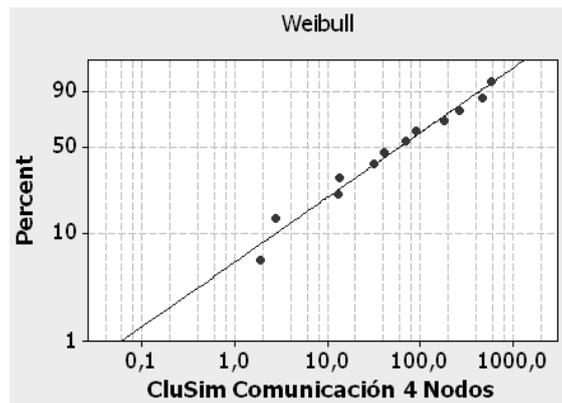


Figura 5. Análisis de Distribución de los tiempos de comunicación generados por CluSim configurado para 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,986.

Al analizar los valores generados por el simulador se observa que los tiempos de cómputo (figura 4), comunicación (figura 5) y ejecución (figura 6) responden a las mismas distribuciones estadísticas que el cluster real.

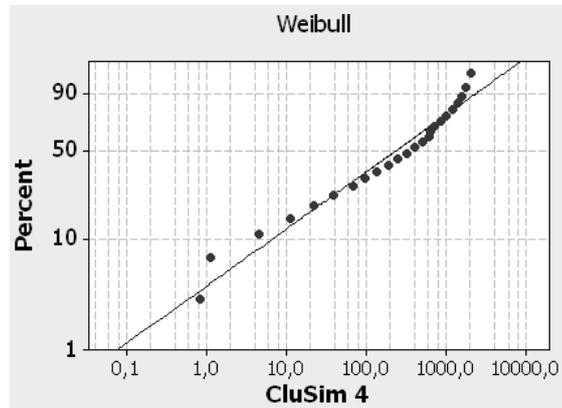


Figura 6. Análisis de Distribución de los tiempos de ejecución de CluSim configurado con 4 nodos. Los valores se ajustan a la distribución Weibull con un coeficiente de correlación de 0,982.

4 Conclusiones y Trabajos Futuros

La contrastación de los resultados obtenidos en la experimentación y en la simulación permiten indicar que tanto los aspectos prestacionales (tiempos de ejecución) como los patrones de cómputo y comunicación de las aplicaciones paralelas consideradas pueden emularse en CluSim con buena precisión, proporcionando nociones acerca de su escalabilidad. Considerando esto, se prevé que CluSim incorpore la simulación de otros tipos de aplicaciones paralelas (SPMD, Divide/Conquer, Pipeline) y que además permita incluir tolerancia a fallos.

5 Referencias

1. Buyya, Rajkumar (editor). High Performance Cluster Computing: Architectures and Systems, Vol. 1. ISBN 0-13-013784-7, Prentice Hall PTR, NJ, USA, 1999.
2. 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.
3. 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.
4. Varga, Andrés. "The OMNeT++ Discrete Event Simulation System". Proceedings of the European Simulation Multiconference (ESM'2001). 2001.
6. INET, <http://www.omnест.com/inet-framework.php>.