SIMURED

-- Manual del usuario --

Fernando Pardo

02 de octubre de 2012

1 Introducción

SimuRed es un simulador de redes de multicomputadores. El programa simula el envío de paquetes a través de una red presentando estadísticas de tiempos consumidos, bloqueos producidos, etc. Permite una configuración casi completa de la red y del proceso de envío de los paquetes, permitiendo el uso de ficheros de trazas donde se especifican los paquetes a enviar.

El simulador es completamente visual y permite incluso ver la evolución de los paquetes a través de la red, por lo que es una buena herramienta para el aprendizaje del funcionamiento de las redes de multicomputadores.

Existe también una versión en línea de comandos cuyo funcionamiento interno es idéntico, pero que al no tener ningún tipo de capacidad visual, los resultados son puramente estadísticos. La configuración de los parámetros de la simulación se realiza redireccionando la entrada estándar desde un fichero donde se guardan los parámetros. Estos parámetros se incluyen en este manual entre paréntesis al lado de las explicaciones de las opciones de menú.

En este manual se explican los diferentes paneles de opciones que tiene el programa, de manera que los conceptos y el funcionamiento del simulador se explican a partir de estos paneles.

2 Configuración de la red

El primer panel sirve para configurar la red de interconexión. Una red tiene tres partes básicas:

- **Topología.** En este apartado introducimos los aspectos estructurales de la red, es decir, parámetros como el número de canales, el número de nodos, las dimensiones, etc.
- **Control de flujo.** Se refiere al mecanismo mediante el cual se transfieren los paquetes de un nodo a otro.
- **Encaminamiento.** Es el algoritmo utilizado para establecer un camino desde un nodo actual hasta un destino.

En este panel hay un tercer apartado dedicado a los retrasos de los elementos de la red. Esta parte se encuentra entre la topología y el control de flujo, por lo que se ha puesto de forma diferenciada.

2.1 Topología

En este subapartado se describen las características físicas de la red de interconexión de multicomputadores. Se toma como base siempre una red estrictamente ortogonal de tipo n-cubo k-aria (toro) que es la más general dentro de las estrictamente ortogonales. Aunque la red malla no es un caso particular de este tipo de redes, se pueden utilizar

también pues una red n-cubo k-aria es igual que una red malla siempre que el mecanismo de encaminamiento impida el paso en los "canales de vuelta".

Estos son los parámetros que se pueden modificar:

- **Dimensiones.** Son las dimensiones de la red. En principio no hay límite. Debe ser siempre mayor que cero. (DIMENSIONS).
- **Nodos por dimensión.** Son los nodos de cada dimensión. Por ejemplo, una red con 2 dimensiones y 4 nodos por dimensión (red bicubo cuaternaria) tiene 16 nodos en total. Debe introducirse un valor mayor que cero. (NODOSDIM).
- **Longitud del buffer.** Esta es la longitud de las colas FIFO de entrada y salida por cada canal. Tanto las colas de entrada como las de salida tienen la misma longitud, por lo que no pueden asignarse longitudes diferentes a unas y otras. El valor debe ser mayor que cero. (LONBUFFER).
- **Canales virtuales.** Cada canal puede subdividirse en varios canales. Por defecto estos canales serán virtuales, es decir, comparten el mismo cable físico pero cada uno tiene su propio buffer, por lo que funcionan como si fueran físicos pero con el aumento de la latencia que conlleva compartir el mismo cable. Si realmente se quiere que estos canales sean físicos, hay una opción para ello (ver un poco después en este mismo punto). (NUMVIRT).
- **Bidireccional.** Se puede especificar si los canales son bidireccionales o no. Hay que tener cuidado pues es posible que algunos algoritmos de encaminamiento no puedan encontrar un camino si el canal es unidireccional (el de orden de dimensión para mallas por ejemplo). (NUMDIR).
- **Adelanto en buffer.** En una cola fifo sencilla los flits van pasando de una posición a la siguiente, pero si hay posiciones vacías en la cola se puede adelantar el flit hasta justo antes de la siguiente posición ocupada, o directamente a la salida de la cola si ésta está vacía. Este adelanto es la opción por defecto, ya que es como funcionan normalmente las redes y se obtiene un mayor rendimiento. (FORWARDING).
- **Canales físicos.** Si se selecciona esta opción entonces los canales virtuales son realmente físicos, es decir, cada canal virtual tiene su propio cable y por lo tanto la transmisión se realiza en paralelo junto con el resto de canales virtuales. (PHYSICAL).

2.2 Retrasos

Se han implementado 4 retrasos en la red. Estos retrasos se dan siempre en ciclos de reloj.

- **Buffer.** Es el tiempo necesario para pasar de una posición de las colas fifo a la siguiente. Normalmente este tiempo es muy pequeño comparado con el resto, por lo que no se suele incluir en muchos simuladores; no porque sea despreciable, sino porque el tiempo de transmisión del paquete, al realizarse de forma segmentada, depende del máximo de este tiempo y del tiempo de atravesar el crossbar, y siempre suele ser el tiempo del crossbar mayor que este. (DELFIFO).
- **Crossbar.** Una vez establecido el camino interno en el encaminador, este tiempo es el que tarda un flit en atravesar el conmutador (crossbar). (DELCROSS).

- **Conmutación.** Es el tiempo que se tarda en establecer el camino dentro del encaminador, es decir, el tiempo que se tarda en decidir qué camino tomar y configurar el conmutador para ello. (DELSWITCH).
- **Canal.** Es el tiempo que necesita un flit para atravesar el canal físico. (DELCHANNEL).

2.3 Control de flujo (Conmutación)

En este caso sólo se ha implementado el control de flujo basado en el **agujero de lombriz (wormhole switching).** Algo parecido a conmutación de paquetes VCT se puede obtener haciendo las colas fifo lo suficientemente grandes como para alojar un paquete entero. La conmutación de circuitos se podría implementar pero dada la estructura interna del simulador, no sería una tarea simple. (SWITCH).

2.4 Encaminamiento

Realizar nuevos algoritmos de encaminamiento es muy sencillo a partir de las fuentes del simulador (red.cpp y red.h). Se han implementado 4 como ejemplo:

Orden de dimensión para mallas. Este es el famoso algoritmo determinista llamado XY para el caso de mallas bidimensionales. Está libre de bloqueos mortales (deadlocks). Funciona también con varios canales virtuales, en cuyo caso elige uno al azar siempre que esté libre. (ROUTING=0).

Orden de dimensión para toros. Igual que el anterior pero para redes n-cubo k-arias. Este algoritmo presenta bloqueos mortales, por lo que se ha incluido como ejemplo para ver cómo el simulador detecta estos bloqueos y los muestra. (ROUTING=1).

Duato para mallas. Se ha aplicado el protocolo de Duato utilizando un conjunto de canales vrtuales completamente adaptativo junto a otro conjunto libre de bloqueos mortales basado en XY. Se necesitan al menos dos canales virtuales (si sólo se pone un canal virtual entonces este algoritmo es exactamente igual que el primero). (ROUTING=2).

Completamente adaptativo para mallas. Este algoritmo elige cualquier canal libre que lleve al destino por un camino mínimo y sin preocuparse si se producirán bloqueos o no. Naturalmente puede presentar bloqueos mortales. (ROUTING=3).

3 Paquetes

En este apartado se muestran todas las opciones referidas a los paquetes y su generación automática, lectura de un fichero de trazas, etc. Se ha dividido en tres apartados, el de paquete, el de generación a partir de trazas y el de generación automática.

3.1 Paquete

- **Longitud del paquete.** Se refiere a la longitud de los datos del paquete. Se mide siempre en flits. (PACKLEN).
- **Cabecera.** Es la longitud de la cabecera. Por defecto está a cero. La longitud real del paquete será la longitud anterior más esta. Para lo único que se utiliza esta longitud es para las estadísticas, pues en ellas se distingue entre datos y cabeza. Por defecto es cero. (PACKHEADLEN).

3.2 Generación de paquetes (trazas)

En este apartado podemos elegir si queremos obtener los paquetes a partir de un fichero de trazas.

El **formato de un fichero de trazas** es el siguiente: para empezar se trata de un fichero texto normal y corriente. Cada línea especifica la generación de un paquete en la red o un comentario, si la línea empieza por el carácter almohadilla (#). Cada línea puede tener hasta cuatro campos numéricos, los tres primeros son obligatorios y el cuarto es opcional. El primer campo es el ciclo absoluto de reloj en el que se producirá el paquete, el segundo es el nodo origen, el tercero es el nodo destino y el cuarto, si está, indica la longitud del paquete. Los nodos están numerados de 0 a n*k-1 donde n es el número de dimensiones y k el número de nodos por dimensión.

Si se quiere utilizar un fichero de trazas, se debe especificar este fichero pulsando sobre el botón **"especificar fichero de trazas"** o eligiéndolo del menú desplegable. (TRACENAME, USETRACE).

En ocasiones es posible que queramos realizar una simulación sin tener en cuenta las longitudes del fichero de trazas, para ello no más que seleccionar la opción **"No leer las longitudes".** (NOREADLEN).

La simulación termina cuando se acaban los paquetes del fichero de trazas.

3.3 Generación automática

Si no tenemos un fichero de trazas se pueden generar los paquetes de forma automática. Esta generación consiste en mandar un paquete desde un origen a un destino, ambos aleatorios, según una productividad dada en flits/ciclo/nodo. La elección de los nodos origen y destino es plana, es decir, cualquier nodo tiene la misma probabilidad de salir. La simulación se prolonga hasta que se han enviado todos los paquetes especificados.

- **Paquetes o Flits por nodo.** Son las dos formas de especificar cuando debe terminar la simulación. Si se elige paquetes entonces se debe introducir el número de paquetes que se introducirán en la red. Si se elige Flits por nodo, entonces se enviarán tantos paquetes como sean necesarios para alcanzar esa cantidad de flits por nodo. (PACKNUM).
- **Productividad.** Es la tasa de emisión de paquetes y se especifica en flits/ciclo/nodo. Este valor se tiene en cuenta en una simulación simple o interactiva, pero si se utiliza una simulación múltiple, este valor no se tiene en cuenta pues viene especificado en la propia simulación interactiva. (PACKPROD).

4 Simulación

Este panel tiene numerosas opciones y posibilidades dadas las múltiples alternativas que existen para simular la red. Las opciones por defecto vienen preparadas para realizar lo que se llama una **simulación múltiple**, es decir, una simulación en la que se van cambiando diferentes parámetros para poder hacer gráficas comparativas. Existe otro tipo de simulación muy interesante, la **interactiva**, que nos permite ver el estado de la red y los paquetes en cada instante.

4.1 Simulación múltiple

Esta es la simulación establecida por defecto (JUSTONE=false). Durante su ejecución no se puede ver la evolución de la red. Sólo se muestra la estructura de la red al principio pero no la evolución de los paquetes. Si por alguna razón se interrumpe la ejecución o se produce un interbloqueo, entonces sí que se muestra el estado de la red en ese instante.

El objetivo de esta simulación en la de obtener estadísticas generales repitiendo los cálculos variando hasta dos parámetros a un tiempo (uno dentro de otro). Estas estadísticas se pueden guardar en un fichero fácilmente importable desde cualquier herramienta de hoja de cálculo. Este fichero también se puede leer desde el propio simulador para mostrar las gráficas correspondientes.

La simulación múltiple se realiza variando hasta dos parámetros. El parámetro del "bucle interior" es siempre la productividad expresada en flits/ciclo/nodo (SIMUVAR=0). El segundo parámetro puede ser ninguno (SIMUVAR2=0), las dimensiones (SIMUVAR2=1), los nodos por dimensión (SIMUVAR2=2), la longitud del buffer (SIMUVAR2=3) y los canales virtuales (SIMUVAR2=4). Si no se especifica ninguna variable secundaria entonces se genera una gráfica en función solamente de la productividad, si se especifica alguna de las posibles variables secundarias entonces se generan tantas gráficas como puntos se hayan especificado para esta variable.

Los valores de la variable del bucle interior (SIMUVAR) irán variando desde un valor inicial (LAVARINI) hasta un valor final (LAVARFIN) un número de veces dado (PUNTOS). La forma en que cambia la variable puede ser lineal (ESCALA=0) o logarítmica (ESCALA=1). Los mismos parámetros existen para la segunda variable (SIMUVAR2, LAVARINI2, LAVARFIN2, PUNTOS y ESCALA).

Pulsando el botón "Simular Múltiple" comienza la simulación. La simulación se puede parar pulsando sobre el botón "Interrumpit". Una vez detenida una simulación múltiple no se puede reanudar.

Los resultados se pueden guardar a un fichero en formato CSV (campos separados por comas). Para ello se debe especificar un nombre de fichero y activar la casilla correspondiente. Por defecto, el fichero de resultados se sobrescribe con los nuevos valores de la simulación. Si se quieren añadir nuevos datos a un fichero que ya tenía resultados de simulaciones anteriores, pulsaremos sobre la opción "Añade los resultados al fichero"; esto es interesante cuando se quieren mostrar gráficas conjuntas provenientes de distintas simulaciones.

4.2 Simulación Simple/Interactiva

En este mismo panel de simulación tenemos la opción "Habilitar simulación simple/interactiva" (JUSTONE=true). Esta es la simulación que permite ver la evolución del los paquetes a través de la red.

Al pulsar sobre esta opción aparece una ventana nueva donde se mostrará la red. En esta ventana estará activo el botón de "Simular". Por defecto no se muestra la evolución, por lo que si pulsamos sobre Simular se realizará una única simulación con los valores que se hubieran especificado.

Para ver los paquetes hay que pulsar sobre la opción "Mostrar evolución" entonces veremos que se activa el botón "Paso a paso" y las opciones de "Mostrar número de flit" y "Retraso entre ciclos".

Una vez en el modo mostrar evolución veremos que al simular aparecen los paquetes moviéndose por la red. Por defecto cada flit tiene un número según su posición y cada paquete se muestra en un color diferente. Se puede desactivar el que aparezca el número de flit clicando sobre la casilla “Mostrar número de Flit”.

Mientras se muestra la evolución de los paquetes, el intervalo de tiempo entre ciclo y ciclo se puede cambiar en la casilla “retraso entre ciclos”. Por defecto hay un intervalo de 100 milisegundos, pero se puede disminuir o aumentar para que vaya más deprisa o para que se vea mejor la evolución de los paquetes por la red.

La simulación mientras se muestran los paquetes se puede detener del todo pulsando el botón “Interrumpir”, pero se puede también detener un instante pulsando sobre “Pausa” o “Paso a Paso”. Si la simulación está detenida se puede continuar con el botón “Continuar” o con el de “Paso a Paso”, si reanudamos la simulación con este último botón, entonces sólo se ejecutará un ciclo y volverá a detenerse.

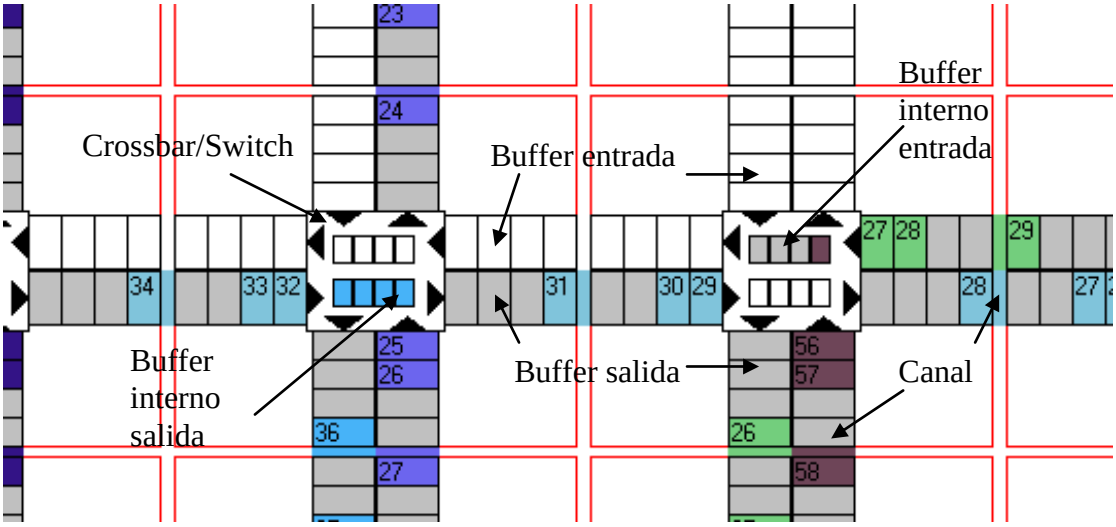

Fig. 1: Detalle de una porción de la red mostrando dos encaminadores y varios paquetes

En la simulación realizada desde la línea de comandos no se muestra nada, por lo que la única diferencia de esta simulación respecto de la múltiple es que sólo se realiza una vez tomando los valores originales de la simulación y no los de las variables de la simulación múltiple.

5 Gráfico

En la versión visual se tiene la posibilidad de leer el fichero de resultados CSV mostrando las gráficas en una ventana adicional. Una vez cargado el fichero de resultados se pueden elegir los parámetros tanto del eje X como del Y. Se han puesto unos valores por defecto para poder ver la latencia media en función de la productividad.

Cada gráfica se muestra con un color y en la parte inferior izquierda se muestra la variable que cambia de gráfica a gráfica con el color correspondiente.

6 Idioma

En este último panel se encuentra la opción para cambiar el idioma. También aparece la versión del programa, el autor y la web donde obtener las últimas actualizaciones, documentación, etc.

7 Línea de comandos

La versión en Java permite su ejecución como un comando. La versión en C++ tiene su propio ejecutable llamado simured_cmd. Los parámetros de la simulación se toman de la entrada estándar y existen valores por defecto para todos ellos. Lo normal es redireccionar la entrada estándar para que tome los valores de un fichero de configuración. Si no se redirecciona la entrada estándar habrá que introducir los valores a mano y pulsar fin de fichero al terminar (Ctrl-D).

Algunos valores estadísticos y los mensajes de simulación se muestran por la salida estándar de error. Los resultados siguiendo el formato CSV (campos separados por comas) se sacan por la salida estándar, por lo que resulta interesante redireccionar la salida estándar a algún fichero para luego importalo a cualquier herramienta gráfica. Un ejemplo de llamada al programa desde el intérprete de comandos sería:

```
simured_cmd < simured.conf > resultados.csv
java -jar jsimured -c < simured.conf > resultados.csv
```

7.1 Fuentes de la versión en línea de comandos

Las fuentes de esta versión están la página del simulador (http://simured.uv.es/). La versión en Java funciona en cualquier plataforma mientras que la versión en C++ debe ser compilada en cualquier plataforma que tenga gcc.

Los ficheros C++ que se tienen son los siguientes:

- **simured_cmd.cpp** Son las fuentes del front-end basado en línea de comandos para el simulador que se encuentra en red.cpp.
- **red.cpp** Es el código fuente del simulador. Aquí se definen todas las clases y métodos para el movimiento de los paquetes por la red. Tanto la versión visual como la de texto comparten estas mismas fuentes.
- **red.h** Es el fichero de cabecera para red.cpp.
- **simured_cmd** Es el ejecutable para Linux (seguramente necesita varias bibliotecas).
- **simured_cmd.exe** Es el ejecutable para DOS (seguramente no necesita ninguna biblioteca dinámica).
- **simured.conf** Es un fichero de configuración de ejemplo con todos los parámetros que necesita el simulador.
- **Makefile** Con este Makefile sólo se necesita hacer "make" y compila todo, tanto en Linux como en DOS (se debe editar previamente para ajustarlo a la plataforma).
- **README.txt** Un fichero de descripción que contiene un texto muy parecido a este.

La instalación consiste en copiar el ejecutable en cualquier directorio y nada más. Si se hacen cambios en las fuentes del programa basta con hacer "make" para generar el nuevo ejecutable. Si se usa cualquier otro compilador distingo al gcc basta con modificar el Makefile para apartarlo al nuevo.

8 Referencias

1. José Duato, Sudhakar Yalamanchili y Lionel Ni. "Interconnection Networks, an Engineering Approach". IEEE Computer Society Press. 1997.
2. Página web del simulador. http://simured.uv.es/