Universidad Zaragoza
1542

ACTAS DE LAS JORNADAS SARTECO

12-14 SEPT

AVANCES EN ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

Teruel

Editado por:

Francisco J. Martínez

Julio A. Sangüesa

Piedad Garrido

Arturo González-Escribano

Diego R. Llanos

Sergio Cuenca Asensi

Jesús González Peñalver

sarteco

iNiT

Avances en arquitectura y tecnología de computadores

Actas de las Jornadas SARTECO 2018

Teruel, 12 a 14 de Septiembre de 2018

Avances en arquitectura y tecnología de computadores
Actas de las Jornadas SARTECO 2018

Editores: Francisco J. Martínez, Julio A. Sangüesa, Piedad Garrido, Arturo Gonzalez-Escribano, Diego R. Llanos, Sergio Cuenca Asensi, Jesús González Peñalver

ISBN-13: 978-84-09-04334-7

Teruel, 2018

Prólogo

La Sociedad de Arquitectura y Tecnología de Computadores (SARTECO) presenta una vez más las Jornadas SARTECO, las cuales integran las XXIX Jornadas de Paralelismo (JP2018) y las III Jornadas de Computación Empotrada y Reconfigurable (JCER2018). Además, en el contexto de las jornadas SARTECO se celebra también el IV Concurso "Tu tesis en 3 minutos" (T3M 2018) que pretende premiar las mejores tesis en el área.

Este año las Jornadas se celebran en Teruel, donde hemos aprovechado para incorporar algunas actividades novedosas al programa científico:

- Un curso titulado "Programación y Arquitectura de Sistemas Heterogéneos", impartido por Darío Suárez, Víctor Viñals, Ruben Gran, Alejandro Valero, Xavier Martorell, Rafael Asenjo y María Ángeles González.
- SARTECO-Pro, una sesión plenaria enfocada en dar orientación profesional, este año a cargo de la empresa HP.
- II Encuentro Women in SARTECO: Retos de la carrera profesional, que este año contará con la presencia de Elisa Martín Garijo, Chief Technology Officer IBM Spain , Inmaculada García, Catedrática de ATC de la Universidad de Málaga, Mª Ángeles González, Profesora Titular de ATC en la Universidad de Málaga, y María Villarroya, Profesora contratada doctora de ATC en la Universidad de Zaragoza, presidenta de AMIT-Aragón.

En esta edición de las Jornadas contamos con un excelente programa de sesiones técnicas con un total de 56 artículos que se presentarán en las JP y otros 18 artículos en las JCER. A fecha de edición del presente libro de actas, el número de asistentes (incluyendo keynotes y voluntarios) es de 147, de los cuales 21 son mujeres. Un gran porcentaje de los asistentes son profesores, pero también asistirán un 29 % de estudiantes de doctorado, un 2 % de estudiantes de máster, y un 7 % de estudiantes de grado. Los asistentes provienen de 23 Universidades españolas, aunque también contamos con 1 asistente de Portugal y 1 de México.

La celebración conjunta de estas actividades y eventos de carácter científico-técnico constituye un referente nacional imprescindible para la comunidad científica agrupada en SARTECO. Estas jornadas reúnen a un nutrido grupo de investigadores, procedentes de diferentes universidades y centros de investigación, con el objeto de intercambiar experiencias, presentar y debatir resultados de investigación, facilitar colaboraciones y sinergias entre grupos, y potenciar nuestras oportunidades de transferencia tecnológica a la industria.

Desde la organización de esta nueva edición de las Jornadas SARTECO, os deseamos a todos una placentera y productiva estancia en Teruel.

Comités de coordinación

Comité de Dirección de las Jornadas SARTECO

Inmaculada García Fernández (UMA) (Presidenta)
Víctor Viñals Yufera (UNIZAR) (Vicepresidente)
Katzalin Olcoz Herrero (UCM) (Secretaria)
Francisco Tirado Fernández (UCM) (Presidente de Honor)

Comité de Organización

Francisco José Martínez Domínguez (UNIZAR)
Piedad Garrido Picazo (UNIZAR)
Julio Alberto Sangüesa Escorihuela (CUD)
Fernando Naranjo Palomino (UNIZAR)
Vicente Torres Sanz (UNIZAR)
Luis Carlos Aparicio Cardiel (UNIZAR)
Mirialys Machín Navas (UNIZAR)
Rafael Asenjo Plaza (UMA)
Jesús González Peñalver (UGR)
Sergio Cuenca Asensi (UA)
Diego R. Llanos Ferraris (UVa)
Arturo González Escribano (UVa)

Comité de Coordinación JP 2018

Ramón Beivide Palacios (UC)
Jesús Carretero Pérez (UCIIIM)
José Duato Marín (UPV)
Inmaculada García Fernández (UMA)
Antonio Garrido Del Solo (UCLM)
Emilio López Zapata (UMA)
Emilio Luque Fadón (UAB)
Alberto Prieto Espinosa (UGR)
Francisco José Quiles Flor (UCLM)
Ana Ripoll Aracil (UAB)
Francisco Tirado Fernández (UCM)
Mateo Valero Cortés (UPC)
Victor Viñals Yúfera (UNIZAR)

Comité de Coordinación JCER 2018

Jesús González Peñalver (UGR)
Sergio Cuenca Asensi (UA)
Miguel A. Vega Rodríguez (UNEX)
Miguel Damas Hermoso (UGR)
Antonio Martínez Alvarez (UA)
Gustavo Sutter (UAM)
Ignacio Bravo (UAH)
José Torres (UV)
Jordi Carrabina (UAB)
Juan Suardíaz (UPCT)
Jesús Barba Romero (UCLM)
Goiuria Sagardui Mendieta (UMONDRAGON)
Jorge Portilla Berrueco (UPM)

Comité T3M 2018

Inmaculada García Fernández (UMA) (Presidenta)
Katzalin Olcoz Herrero (UCM) (Secretaria)
Francisco Tirado Fernández (Presidente de Honor)
Enrique S. Quintana Ortí (UJI) (Vocal, Junta directiva SARTECO)
Miquel Moretó Planas (UPC) (Vocal, Junta directiva SARTECO)

Caracterizadión de Cargas para Redes de Computación Exascale

José Duro, Salvador Petit, Julio Sahuquillo y María E. Gómez [1]

Resumen— **La computación exascale es el siguiente paso en la computación de alto rendimiento proporcionada por sistemas compuestos por millones de núcleos de procesamiento interconectados. Para guiar el diseño e implementación de dichos sistemas, se requieren múltiples estudios de caracterización de carga de trabajo y evaluaciones del sistema.**

Este documento proporciona un estudio de caracterización de la carga de trabajo en el contexto del proyecto europeo ExaNeSt, que se centra, entre otros, en el desarrollo de la tecnología de red necesaria para implementar futuros sistemas de exascale. En este trabajo, caracterizamos diferentes aplicaciones ExaNeSt desde la perspectiva de la red informática mediante el análisis de la distribución de mensajes por tamaño, el consumo de ancho de banda a lo largo de la ejecución y los patrones espaciales de comunicación entre los núcleos.

El análisis destaca tres observaciones principales; i) los tamaños de los mensajes son, en general, inferiores a 50 kB; ii) los patrones de comunicación suelen ser de ráfaga; y iii) la comunicación espacial entre los núcleos se concentra en los puntos críticos para la mayoría de las aplicaciones. Tomando en cuenta estas observaciones, uno puede concluir que para desbloquear enlaces de red congestionados, una red exascale debe diseñarse para soportar brevemente anchos de banda superiores a la media en las proximidaes de los núcleos claves de la red.

Palabras clave— ***Caracterización de Carga*; *Computación Exascale*; *MPI*.**

I. Introducción

La computación *Exascale*, pretende alcanzar el exaflop (10^{18} operaciones de coma flotante por segundo), que es el próximo desafío para la comunidad de supercomputación. De acuerdo con la creciente tendencia computacional actual, se espera que esta meta se alcanceel año 2020. Para lograr una capacidad informática tan grande, los sistemas requerirán millones de elementos de cómputo interconectados que ejecuten aplicaciones paralelas masivas.

Las redes Exascale estarán compuestas por miles de núcleos informáticos, por lo que la transmisión de datos entre ellos se convierte en una gran preocupación de diseño. En este contexto, surgen nuevos requisitos, no solo en términos de rendimiento, sino también en relación con las demandas de energía. En dichos sistemas, la tecnología de red subyacente [1], [2] y la topología son opciones de diseño fundamentales. El enfoque del proyecto europeo de interconexión y almacenamiento del sistema Exascale (European Exascale System Interconnect and Storage project - ExaNeSt) [3], [4], que se está desarrollando actualmente, es proporcionar una implementación factible que cumpla con los requisitos mencionados.

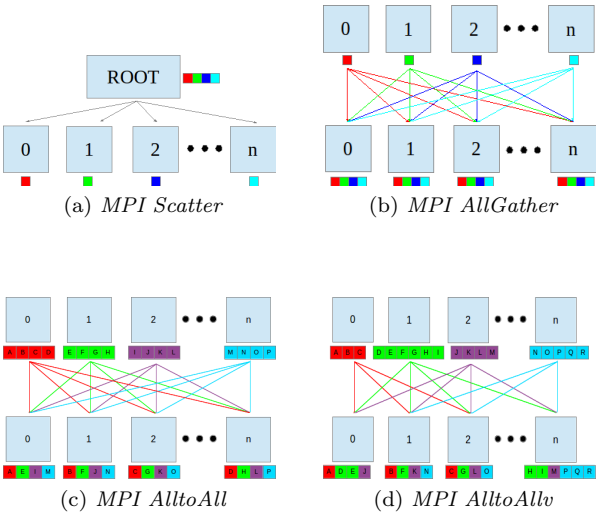
[1]Departamento de Informaática de Sistemas y Computadores Universitat Politècnica de València, email: jodugo1@gap.upv.es

Los entornos de simulación de ExaNeSt modelan las redes de interconexión eléctricas [5], [6] y óptica [7]. Para realizar un análisis realista de todo el comportamiento del sistema, las plataformas de simulación deben alimentarse con aplicaciones reales o trazas que modelen el comportamiento de dichas aplicaciones. El objetivo principal de este documento es presentar una caracterización detallada de las cargas de trabajo de ExaNeSt para proporcionar un conocimiento profundo de los requisitos de la red y para guiar a los diseñadores de red en la toma de decisiones.

Para caracterizar los requisitos de red de las cargas de trabajo ExaNeSt, se perfilaron para recopilar su tiempo de ejecución y los mensajes, tando los punto a punto como las colectivas MPI (interfaz de paso de mensajes) [8]. La información se utiliza para generar trazas que se analizan para obtener información sobre la distribución del tamaño de los mensajes, la evolución temporal de la utilización del ancho de banda y los patrones de comunicación espacial. Los datos de caracterización serán útiles para seleccionar y/o desarrollar topologías y tecnologías de red adecuadas para una implementación de red exascale factible y eficiente.

El estudio de caracterización destaca tres observaciones principales; i) la comunicación entre núcleos, en general, se realiza principalmente con mensajes cuyo tamaño es inferior a 50 KB; ii) aunque las aplicaciones presentan una amplia gama de consumos de ancho de banda, los patrones de comunicación suelen ser de ráfagas, y iii) nuestro análisis muestra que la comunicación espacial entre los núcleos puede concentrarse en los puntos críticos. Como resultado, podemos concluir que para evitar pérdidas de rendimiento debido a la red informática, una red exascale debe diseñarse para admitir anchos de banda superiores a la media en núcleos clave de red en puntos específicos de tiempo.

El resto de este documento está organizado de la siguiente manera. La sección II motiva este trabajo en el contexto del proyecto ExaNeSt. La sección III proporciona información general sobre las colectivas MPI. La sección IV presenta una descripción general de las cargas de trabajo estudiadas. Las secciones V, VI y VII analizan, respectivamente, la distribución del tamaño de los mensajes, la evolución temporal del consumo de ancho de banda y los patrones de comunicación espacial de las cargas de trabajo estudiadas. Finalmente, la sección VIII presenta algunas observaciones finales.



(a) *MPI Scatter* (b) *MPI AllGather*

(c) *MPI AlltoAll* (d) *MPI AlltoAllv*

Fig. 1: Colectivas MPI.

II. Motivación

Se espera que los requisitos para la computación Exascale en la década actual aumenten el rendimiento de la red. El proyecto ExaNeSt actualmente está diseñando y construyendo un prototipo de arquitectura de red capaz de alcanzar la computación Exascale.

El objetivo de ExaNeSt es desarrollar un sistema que se pueda ampliar hasta las decenas de millones de núcleos de bajo consumo de energía interconectados para resolver problemas científicos y de big data a una gran escala. Para soportar un sistema de este tamaño, ExaNeSt se enfrenta al enorme desafío de diseñar una interconexión capaz de cumplir con un desempeño muy estricto, la capacidad de recuperación y las limitaciones en terminos de costes para la serie de desafíos computacionales.

La red ExaNeSt es una red de interconexión de múltiples niveles que se puede dividir en dos partes distintas. Los niveles inferiores, que se fijan físicamente mediante paneles y planos posteriores en los armarios, y los niveles superiores que son completamente reconfigurables utilizando enrutadores personalizados basados en FPGA [9]. Esta flexibilidad permite construir cualquier topología de red, es decir, directa, indirecta o híbrida, o incluso el uso de conmutadores estándar de productos básicos.

Para cumplir con los requisitos de una interconexión tan exigente, en este trabajo analizamos las aplicaciones reales que se utilizan para probar todo el sistema (núcleos de cómputo, red de interconexión y almacenamiento).

III. Background de las Colectivas MPI

Las aplicaciones ExaNeSt consisten en miles de hilos que se han codificado con colectivas de la interfaz de paso de mensajes (MPI). Para ayudar a comprender el estudio de caracterización esta sección identifica las colectivas MPI utilizadas por las cargas de trabajo estudiadas y describe cómo funcionan.

Al estudiar las cargas de trabajo ExaNeSt, encontramos el siguiente conjunto de primitivas: *MPI_Bcast*, *MPI_Scatter*, *MPI_Scatterv*, *MPI_Gather*, *MPI_Allgather*, *MPI_Reduce*, *MPI_Allreduce*, *MPI_Alltoall*, *MPI_Alltoallv*, y *MPI_Scan*.

- *MPI_Bcast*: esta primitiva permite que un proceso (es decir, la raíz) envíe un fragmento de la matriz a todos los procesos en un comunicador (es decir, el conjunto de núcleos involucrados en la colectiva).
- *MPI_Scatter*: esta colectiva es similar a *MPI_Bcast*. La principal diferencia es que *MPI_Scatter* envía diferentes fragmentos de igual tamaño de una matriz a diferentes procesos (ver Figura 1a).
- *MPI_Scatterv*: una variación de la colectiva *MPI_Scatter* donde los fragmentos pueden ser de diferentes tamaños.
- *MPI_Gather*: implementa el comportamiento opuesto de *MPI_Scatter*. En lugar de separar elementos de un proceso (raíz) a muchos procesos, *MPI_Gather* toma elementos de muchos procesos y los envía al mismo proceso (raíz).
- *MPI_Gatherv*: como *MPI_Gather*, pero con fragmentos de diferentes tamaños.
- *MPI_Allgather*: dado un conjunto de elementos de datos distribuidos en todos los procesos, esta colectiva envía todos los elementos a todos los procesos. En la primera etapa, *MPI_Allgather* reenvía los elementos al proceso raíz, y luego, en una segunda etapa, este proceso envía los datos recopilados a todos los procesos (ver Figura 1b).
- *MPI_Reduce*: esta primitiva es similar a *MPI_Gather*. *MPI_Reduce* toma una matriz de elementos en cada proceso y devuelve una matriz procesada de elementos al proceso raíz. Por lo tanto, esta primitiva implica un cálculo con los datos de cada elemento.
- *MPI_Allreduce*: como *MPI_Reduce* pero el resultado del cálculo se distribuye entre todos los procesos.
- *MPI_Barrier*: esta primitiva implementa una barrera. Por lo tanto, el proceso que lo llama se detiene hasta que todos los procesos en el comunicador también lo hayan llamado. Se implementa como *MPI_Bcast* pero con mensajes muy cortos (es decir, tokens).
- *MPI_Alltoall*: es similar a *MPI_Scatter* pero en este caso, todos los procesos dividen las matrices de entrada con el mismo tamaño en partes iguales y envían cada fragmento a todos los procesos en el comunicador (ver Figura 1c). Con esta primitiva, cada proceso envía y recibe la misma cantidad de datos.
- *MPI_Alltoallv*: esta primitiva presenta dos diferencias principales con respecto a *MPI_Alltoall*. Por un lado, las matrices de entrada pueden tener diferentes tamaños y, por otro lado, un proceso puede recibir trozos de diferentes tamaños de cada emisor (o no recibir nada de un núcleo

Application		Tiempo de Ejecución (ciclos)	MB Transf (Total)	MB/s (media)
Lammps	24 Núcleos	43,754,790,287	24,644	1,126
	48 Núcleos	21,713,810,259	31,141	2,868
	96 Núcleos	10,887,071,229	40,934	7,520
	192 Núcleos	5,983,794,523	55,338	18,496
	384 Núcleos	152,820,600,368	71,924	941
	768 Núcleos	322,882,228,358	97,993	607
RegCM	24 Núcleos	139,543.000,917	22,976	329
	48 Núcleos	80,112,643,804	33,157	828
	96 Núcleos	49,580,028,588	47,213	1,905
	144 Núcleos	51,640,032,689	58,138	2,252
	192 Núcleos	40,411,947,679	69,131	3,421
Gadget	24 Núcleos	316,651,890,866	152,567	964
	48 Núcleos	257,497,854,652	267,104	2,075
	72 Núcleos	207,637,515,308	415,640	4,004
DPSNN 32x32	32 Núcleos	8,795,221,526,254	34,533	8
	64 Núcleos	4,568,949,694,490	45,690	20
	128 Núcleos	2,744,786,342,258	65,125	47
DPSNN 64x64	64 Núcleos	23,829,773,201,115	170,572	14
	128 Núcleos	12,287,105,198,156	199,951	33

TABLA I: Ancho de banda medio requerido para cada traza.

particular) (ver Figura 1d).

- *MPI_Scan*: calcula una reducción parcial incremental entre los procesos de los participantes. Esto significa que cada proceso i calcula la reducción del proceso 0 a sí mismo. Es decir, el último proceso n obtendrá la reducción total entre todos los procesos.

IV. Cargas ExaNeSt y Metodología de Análisis de Trazas

Las cargas de trabajo de ExaNeSt consisten en cuatro aplicaciones principales existentes o desarrolladas por los socios de ExaNeSt, concretamente *Lammps* [10], *RegCM*[11], *DPSNN*[12] y *Gadget*[13]. Los detalles sobre cómo funciona cada aplicación se pueden encontrar en cada referencia particular.

Para proporcionar información sobre la relación entre los requisitos de ancho de banda y el recuento de núcleos de la red, se han considerado diferentes tamaños de carga de trabajo. Más exactamente, los socios de ExaNeSt proporcionaron 19 trazas generadas con hardware real utilizando diferentes herramientas de recopilación de estadísticas de red, tales como *scalasca* [14], que proporcionan información de las primitivas MPI además de los tiempos de computación y las marcas de tiempo de los mensaje.

Estas trazas corresponden a 6 ejecuciones diferentes (variando el recuento de núcleos de 24 a 768) de Lammps, 5 de RegCM (de 24 a 192 núcleos), 5 de DSPNN (de 32 a 128 núcleos en la configuración de columnas neuronales de 32x32 y de 64 a 128 núcleos en la configuración de columnas neuronales 64x64) y 3 de Gadget (de 24 a 72 núcleos). En este trabajo presentamos el análisis de un subconjunto representativo de las trazas proporcionadas.

El primer paso en el estudio de caracterización es proporcionar una visión global de las trazas desde la perspectiva del ancho de banda de la red. Para este propósito, reunimos dos parámetros clave: el tiempo de ejecución y los datos generales transferidos. De ellos, calculamos el ancho de banda promedio consumido por cada aplicación.

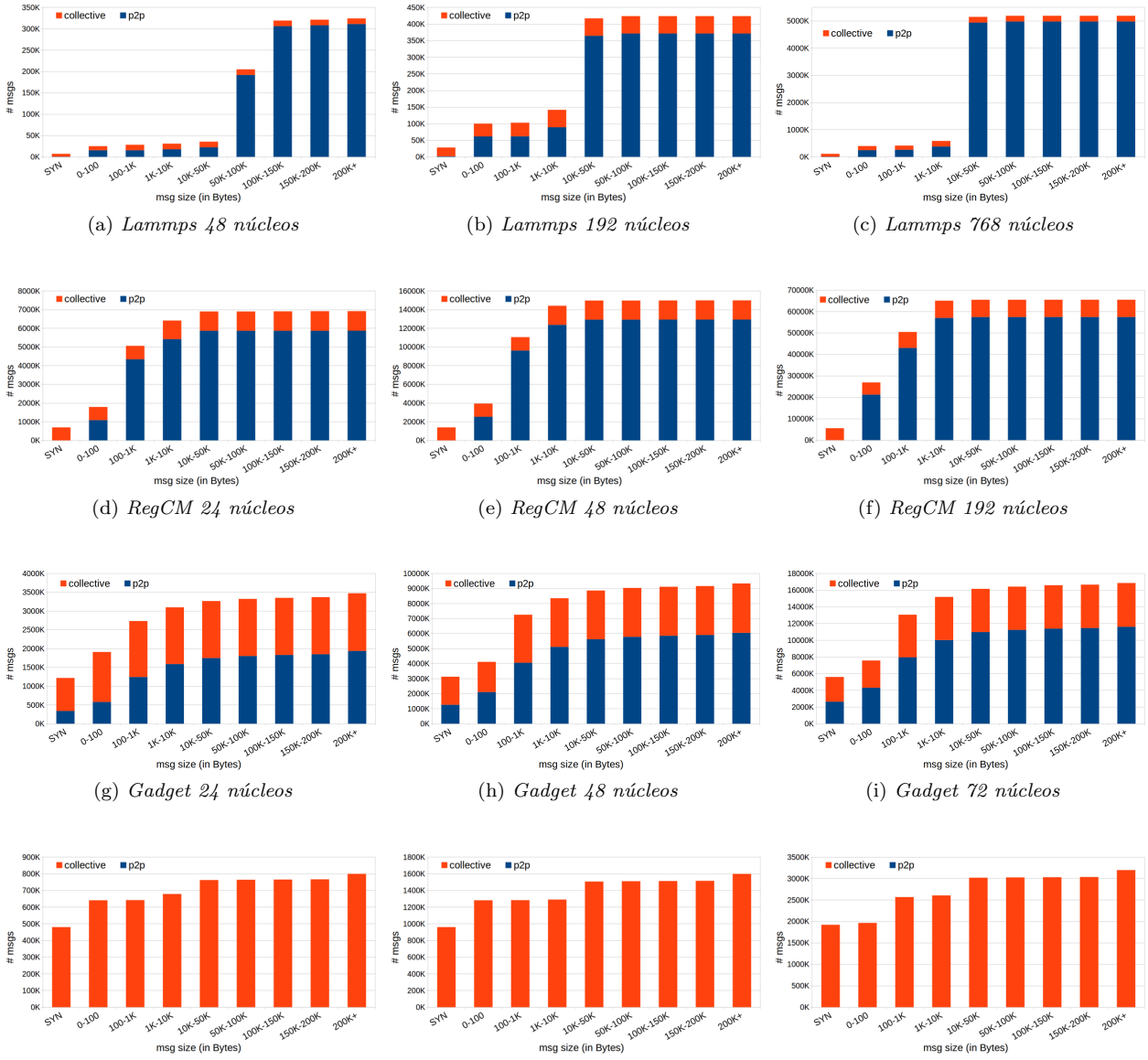
Consideramos todos los datos transferidos, tanto la cabecera como la carga. Los mensajes se dividen en paquetes de 72 bytes para ser inyectados en la red, donde 64 bytes corresponden a la carga útil y los restantes 8 bytes restantes corresponden a la cabecera. Más concretamente, un mensaje de 1KB se divide en 16 paquetes como se hace en algunas máquinas modernas [15], incurriendo en una sobrecarga de 128B para las cabeceras.

La tabla I muestra los resultados. Como se observa, los valores obtenidos varían ampliamente entre las variables estudiadas. El tiempo de ejecución presenta diferencias que van desde alrededor de 6 hasta 23 billones de ciclos. Los requisitos de ancho de banda presentan grandes variaciones independientemente del tiempo de ejecución. Dado que las trazas no proporcionan la frecuencia del procesador, hemos asumido un reloj con procesador de 2 GHz para calcular el ancho de banda medio en segundos. Los resultados muestran que hay aplicaciones con relativamente pocos requisitos de ancho de banda (aproximadamente 8 MBps) y aplicaciones con grandes requisitos de ancho de banda (alrededor de 18,5GBps). Tenga en cuenta que cuando el número de núcleos excede 192 en Lammps, el tiempo de ejecución aumenta. La razón es que el sistema original utilizado para generar estas trazas tiene menos núcleos que los hilos paralelos que se generan en estas configuraciones, lo que afecta el tiempo de ejecución.

V. Análisis del Tamaño de Mensajes

El análisis del tamaño del mensaje se realizó para discernir si la entrega del mensaje podría mejorarse al priorizar la latencia o el ancho de banda. Por ejemplo, si hay una gran cantidad de mensajes cortos podríamos optar por priorizar la latencia sobre el ancho de banda; sin embargo, en los tamaños de mensajes grandes, debemos priorizar el ancho de banda para mejorar la red y, por lo tanto, el rendimiento de la aplicación. Aunque se han analizado 19 trazas, solo los patrones representativos identificados se presentan y discuten con fines ilustrativos.

La figura 2 muestra la distribución del tamaño de mensaje acumulativo que varía el recuento de núcleos a través de las trazas estudiadas. El eje Y indica la cantidad de mensajes (tanto debidos a primitivas MPI colectivas como punto a punto) transferidos y el eje X el tamaño del mensaje distribuido en rangos. La primera columna (etiquetada como SYN) se refiere al número de mensajes de sincronización, cuya característica principal es que no incluyen la carga



(a) *Lammps 48 núcleos* (b) *Lammps 192 núcleos* (c) *Lammps 768 núcleos*

(d) *RegCM 24 núcleos* (e) *RegCM 48 núcleos* (f) *RegCM 192 núcleos*

(g) *Gadget 24 núcleos* (h) *Gadget 48 núcleos* (i) *Gadget 72 núcleos*

(j) *DPSNN 32x32 32 núcleos* (k) *DPSNN 64x64 64 núcleos* (l) *DPSNN 64x64 128 núcleos*

Fig. 2: Distribución basada en el Tamaño del Mensaje.

útil; sin embargo, también se analizan porque, como se verá en los resultados, pueden generar una cantidad considerable de tráfico en algunas aplicaciones.

Con respecto a la distribución del tamaño del mensaje en Lammps (ver figuras 2a - 2c), se puede apreciar que la mayoría de los tamaños de mensaje están ubicados entre 10kB y 50kB para un número de núcleos mayor o igual de 192. Por el contrario, para 48 núcleos, el tamaño del mensaje dominante varía de 50 kB a 100 kB. En resumen, en promedio, cuanto menor es el recuento de núcleos, se utilizan mensajes de tamaños más grandes.

Las figuras 2d - 2f muestran la distribución del tamaño del mensaje en las trazas de la aplicación RegCM. El tamaño del mensaje en esta aplicación es bastante homogéneo al variar el recuento de núcleos. El tamaño dominante oscila entre 100B y 1kB en todas las trazas, aunque la traza de 192 núcleos también tiene una cantidad significativa de mensajes más pequeños (de 0 a 10 kB).

En gadget (figuras 2g - 2i) muestra un alto porcentaje de mensajes de sincronización, independientemente de la cantidad de núcleos. Como se observa, una cantidad significativa de mensajes (que van del 30 % al 50 %) presenta un tamaño inferior a 1 kB, aunque alrededor del 20 % de los mensajes son más grandes.

Finalmente, las Figuras 2j - 2l representan la distribución del tamaño del mensaje en las trazas desde DPSNN. El primer gráfico (Figura 2j) corresponde a la aplicación que trabaja con 32x32 columnas neuronales (aproximadamente 1.2M neuronas y 2.6G sinapsis), mientras que las dos trazas restantes se refieren a la aplicación con 64x64 columnas neuronales (aproximadamente 5M neuronas y sinapsis 10G). En promedio, el tráfico generado por los mensajes de sincronización representa hasta un 60 % de la cantidad total. A diferencia de las trazas analizadas previa-

(a) *Lammps 192 núcleos*

(b) *Lammps 768 núcleos*

(c) *RegCM 24 núcleos*

(d) *RegCM 192 núcleos*

(e) *Gadget 24 núcleos*

(f) *Gadget 72 núcleos*

(g) *DPSNN 64x64 128 núcleos*

(h) *Zoomed Y axis of DPSNN 64x64 128 núcleos*

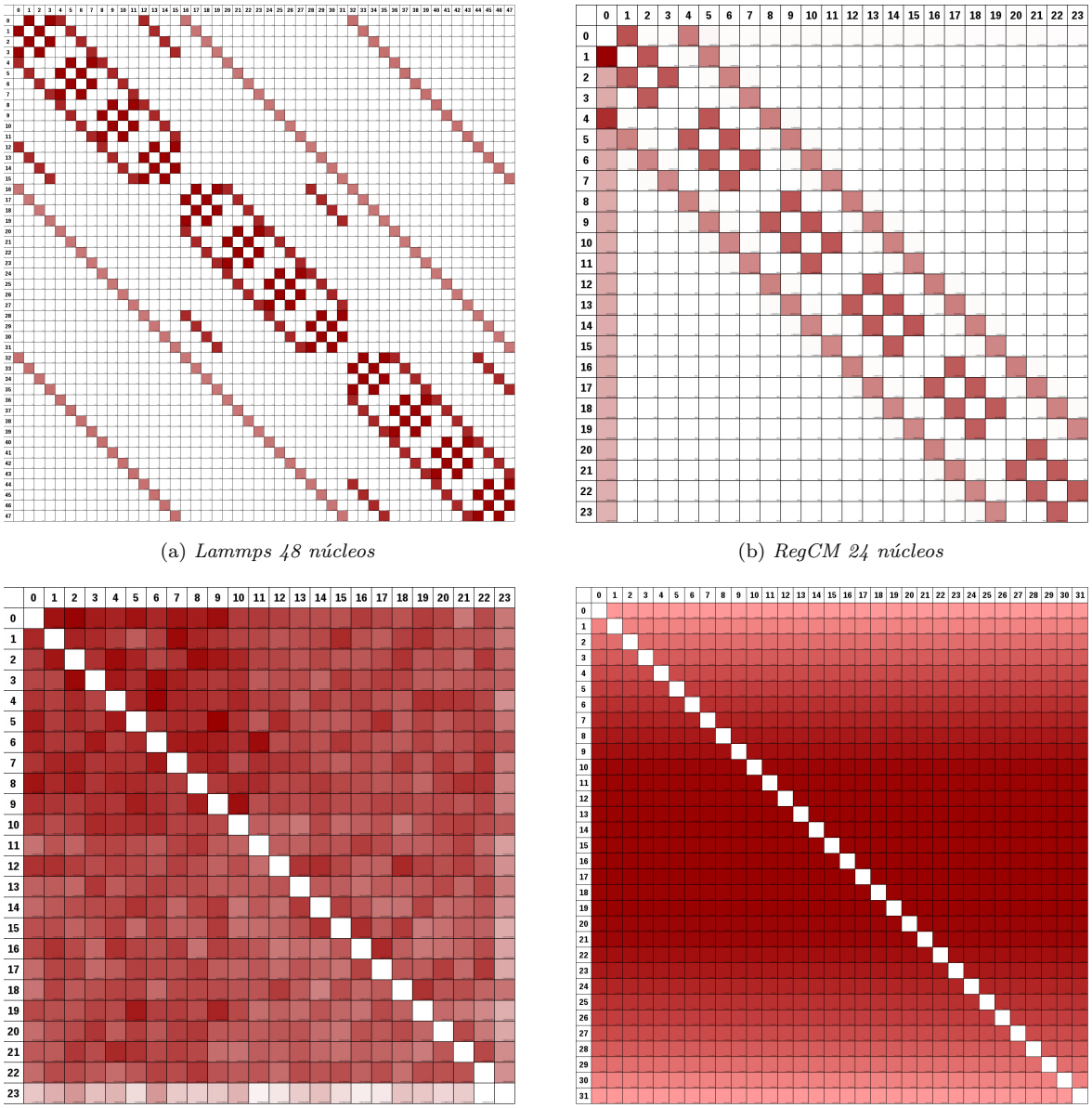
Fig. 3: Evolución temporal del mínimo, media y máximo ancho de banda.

mente, el tráfico en esta aplicación se debe a colectivas MPI en lugar de mensajes punto a punto.

VI. Análisis de la Evolución Temporal

En la sección IV (Tabla I) mostramos el consumo de ancho de banda para cada traza estudiada en promedio a lo largo del tiempo de ejecución de la aplicación. El objetivo de esta sección es analizar los requisitos de ancho de banda dinámico para explorar cómo evolucionan los requisitos con el tiempo.

Analizamos todas las trazas de cada aplicación y encontramos que el comportamiento de algunas aplicaciones cambia al variar la cantidad de núcleos. Con fines ilustrativos, mostramos un ejemplo de cada uno de los múltiples patrones exhibidos por las aplicaciones estudiadas al variar el recuento de núcleos. Para facilitar el análisis visual y homogeneizar la representación de las tramas, dividimos el tiempo de ejecución de cada aplicación en 200 intervalos de la misma longitud. Luego, dividimos cada intervalo en subintervalos de 10M-ciclos y calculamos el consumo de ancho de banda de cada subintervalo. Estos valores se promedian para obtener el consumo de ancho de banda de cada intervalo, que se etiqueta como Mean en las figuras. Además, los consumos de ancho de banda máximo (Max) y mínimo (Min) entre los



(a) *Lammps 48 núcleos*

(b) *RegCM 24 núcleos*

(c) *Gadget 24 núcleos*

(d) *DPSNN 32x32 32 núcleos*

Fig. 4: Matriz de comunicación entre los núcleos.

subintervalos también se trazan para discernir fácilmente patrones de comunicación de ráfagas.

La figura 3a y la figura 3b muestran los patrones de ancho de banda de Lammps para 192 y 768 núcleos, respectivamente; que son representativos de todos los patrones de esta aplicación. La figura 3a muestra el comportamiento exhibido para un recuento de núcleos inferior o igual a 192, mientras que la otra figura muestra el comportamiento cuando el número de núcleos supera los 192. Se puede apreciar que en el primer caso, las variables estudiadas muchas veces se superponen entre sí. Sin embargo, cuando la cantidad de núcleos aumenta por encima de 192, hay una clara diferenciación entre las tres variables representadas. Como se esperaba (ver Tabla I), el tráfico de red es mucho menos importante en 768 núcleos que en 192.

La aplicación RegCM muestra patrones de ancho de banda similares independientemente del recuento de núcleos, con la única excepción de la traza de 24 núcleos. La figura 3c y la figura 3d muestran ambos patrones. Elegimos la traza de 192 núcleos, porque es la que presenta el mayor ancho de banda promedio. Una observación interesante es que el ancho de banda experimenta un fuerte aumento al final de la ejecución en ambos comportamientos exhibidos. En cuanto a la figura 3d, RegCM muestra en promedio menos requerimientos de ancho de banda que Lammps (en 40 %) y requisitos de ancho de banda máximo similares.

La figura 3e y la figura 3f muestran la evolución temporal de Gadget. Esta aplicación presenta una gran diferencia entre el promedio y el máximo, lo que implica un patrón de comunicación a ráfagas; es decir, hay subintervalos con altos requisitos de comunicación y otros con muy poco tráfico.

DPSNN exhibe un comportamiento homogéneo en todas las trazas estudiadas. Por esta razón, elegimos

la traza que presenta los requisitos de ancho de banda más altos, es decir, una matriz neuronal de 64x64 con 128 núcleos. La figura 3g muestra los resultados. En comparación con las aplicaciones anteriores, se puede observar que DPSNN tiene requisitos de ancho de banda bajos durante casi toda la ejecución. Sin embargo, en los primeros intervalos, DPSNN presenta grandes consumos de ancho de banda. Debido a este hecho, tuvimos que reducir el eje Y en la figura 3g para apreciar el promedio como se muestra en la figura 3h.

VII. Análisis de la Comunicación Espacial

Una vez que se han analizado las aplicaciones en tiempo de ejecución, esta sección estudia la cantidad de tráfico que cada núcleo envía/recibe entre sí. En otras palabras, la distribución espacial de la comunicación entre los núcleos (fuente-destino de la matriz).

La figura 4 muestra la matriz de comunicaciones resultante para las cuatro aplicaciones estudiadas. Cuanto más oscuro es el color, mayor es la cantidad de bytes transferidos. Se puede ver que el tráfico se concentra en un pequeño porcentaje de núcleos en Lammps y RegCM, mientras que se extiende entre todos los núcleos en DSPNN y Gadget. En DSPNN, el tráfico sigue un patrón regular (más oscuro en los núcleos centrales y más claro en los núcleos superiores e inferiores), mientras que en los núcleos de gadgets se comunican entre todos de forma aleatoria.

VIII. Conclusiones

Las caracterizaciones de carga de trabajo son necesarias para guiar a los investigadores en el diseño de nuevos sistemas. En este documento, hemos analizado las trazas obtenidas a partir de aplicaciones reales utilizadas en el proyecto europeo ExaNeSt, que se utilizan para diseñar e implementar la red de interconexión para un sistema de exascale.

El análisis se ha realizado teniendo en cuenta tres características principales: la distribución de tipos y tamaños de mensajes, el ancho de banda consumido durante el tiempo de ejecución y la comunicación espacial entre núcleos.

Con respecto al análisis de distribución de tamaños de mensajes, la mayoría de las aplicaciones (tres de las cuatro estudiadas) presentan una mayor cantidad de mensajes punto a punto, aunque una de las aplicaciones (DPSNN) está completamente dominada por colectivos MPI. En general, la mayoría de los mensajes están por debajo de 50 kB independientemente del tamaño de la carga de trabajo.

El análisis del ancho de banda consumido durante el tiempo de ejecución indica que las aplicaciones presentan una amplia gama de requisitos de ancho de banda en promedio; sin embargo, la mayoría de las aplicaciones presentan patrones de comunicación por ráfagas que pueden estresar la red de interconexión en determinados momentos.

Finalmente, el análisis de la matriz de comunicaciones espaciales para las diferentes aplicaciones muestra patrones de comunicación espacial muy diferentes entre aplicaciones. Por ejemplo, en algunas aplicaciones, el tráfico se distribuye entre todos los núcleos, mientras que en otros el consumo de ancho de banda se concentra en los puntos críticos. Esto significa que, para soportar ráfagas de comunicación y desbloquear enlaces de red congestionados, una red de exascale adecuada debe proporcionar un ancho de banda superior al promedio en el entorno de núcleos clave en puntos específicos de tiempo.

Agradecimientos

Este trabajo fue apoyado por el proyecto ExaNest, financiado por el programa de investigación e innovación Horizon 2020 de la Unión Europea bajo el acuerdo de subvención No. 671553, y por el Ministerio de Economía y Competitividad (MINECO) y fondos del Plan E bajo Grant TIN2015-66972-C5-1-R.

Referencias

[1] Avinash Karanth Kodi, Brian Neel, and William C Brantley, "Photonic interconnects for exascale and datacenter architectures," *IEEE Micro*, vol. 34, no. 5, pp. 18–30, 2014.

[2] Sébastien Rumley, Dessislava Nikolova, Robert Hendry, Qi Li, David Calhoun, and Keren Bergman, "Silicon photonics for exascale systems," *Journal of Lightwave Technology*, vol. 33, no. 3, pp. 547–562, 2015.

[3] M Katevenis, N Chrysos, M Marazakis, I Mavroidis, F Chaix, N Kallimanis, J Navaridas, J Goodacre, P Vicini, A Biagioni, et al., "The exanest project: Interconnects, storage, and packaging for exascale systems," in *Digital System Design (DSD), 2016 Euromicro Conference on*. IEEE, 2016, pp. 60–67.

[4] "ExaNeSt website," 2018, May.

[5] Fco. Javier Ridruejo Perez and José Miguel-Alonso, "Insee: An interconnection network simulation and evaluation environment," in *Proceedings of the 11th International Euro-Par Conference on Parallel Processing*, Berlin, Heidelberg, 2005, Euro-Par'05, pp. 1014–1023, Springer-Verlag.

[6] Javier Navaridas, Jose Miguel-Alonso, Jose A. Pascual, and Francisco J. Ridruejo, "Simulating and evaluating interconnection networks with {INSEE}," *Simulation Modelling Practice and Theory*, vol. 19, no. 1, pp. 494 – 515, 2011, Modeling and Performance Analysis of Networking and Collaborative Systems.

[7] José Duro, Salvador Petit, Julio Sahuquillo, and María E Gómez, "Modeling a photonic network for exascale computing," in *High Performance Computing & Simulation (HPCS), 2017 International Conference on*. IEEE, 2017, pp. 511–518.

[8] William Gropp, Ewing Lusk, and Anthony Skjellum, *Using MPI: portable parallel programming with the message-passing interface*, vol. 1, MIT press, 1999.

[9] Caroline Concatto, Jose A. Pascual, Javier Navaridas, Joshua Lant, Andrew Attwood, Mikel Lujan, and John Goodacre, "A cam-free exascalable hpc router," 2018, Architecture of Computing Systems (ARCS), p. to appear.

[10] Steve Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of computational physics*, vol. 117, no. 1, pp. 1–19, 1995.

[11] Filippo Giorgi, E Coppola, F Solmon, L Mariotti, MB Sylla, X Bi, N Elguindi, GT Diro, V Nair, G Giuliani, et al., "Regcm4: model description and preliminary tests over multiple cordex domains," *Climate Research*, vol. 52, pp. 7–29, 2012.

[12] Pier Stanislao Paolucci, Roberto Ammendola, Andrea Biagioni, Ottorino Frezza, Francesca Lo Cicero, Alessandro Lonardo, Elena Pastorelli, Francesco Simula, Laura Tosoratto, and Piero Vicini, "Distributed simulation of polychronous and plastic spiking neural networks: strong and weak scaling of a representative mini-application

benchmark executed on a small-scale commodity cluster," *arXiv preprint arXiv:1310.8478*, 2013.
[13] Volker Springel, "The cosmological simulation code gadget-2," *Monthly notices of the royal astronomical society*, vol. 364, no. 4, pp. 1105–1134, 2005.
[14] "scalasca website," 2018, May.
[15] Saïd Derradji, Thibaut Palfer-Sollier, Jean-Pierre Panziera, Axel Poudes, and François Wellenreiter Atos, "The bxi interconnect architecture," in *High-Performance Interconnects (HOTI), 2015 IEEE 23rd Annual Symposium on*. IEEE, 2015, pp. 18–25.