Informe técnico

Resumen: Este informe recoge los principales resultados extraídos del estudio del rendimiento de diferentes herramientas de paralelización en el superordenador Finis Terrae II instalado en el CESGA. Se incluyen resultados tanto de los rendimientos de comunicaciones punto a punto como de comunicaciones colectivas, en las que se analiza como afecta el aumento del número de procesos por nodo a las eficiencias de las diferentes herramientas MPI estudiadas.

Id. Documento:	**CESGA-2016-002**
Fecha :	**27/12/2016**
Responsables:	**Ignacio Santos Díaz** **Sandra González Rodríguez** **Jose Carlos Mouriño Gallego**
Estado:	**FINAL**

Estudio del rendimiento de diferentes herramientas MPI en el sistema FinisTerrae II

Centro de Supercomputación de Galicia

Autores

Ignacio Santos Díaz

Sandra González Rodríguez

Jose Carlos Mouriño Gallego

Indice

1 Introducción.....7

2 Infraestructura.....7

3 Resultados.....8

4 Discusión de resultados.....21

4.1 Comunicación punto a punto.....21

4.2 Comunicación colectiva.....22

5 Conclusiones.....23

Figuras

Fig.1 Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de gran tamaño en un mismo nodo.....8

Fig.2: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de tamaño intermedio en un mismo nodo.....9

Fig.3: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de pequeño tamaño en un mismo nodo.....9

Fig.4: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de gran tamaño en dos nodos distintos.....10

Fig.5: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de tamaño intermedio en dos nodos distintos.....10

Fig.6: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de pequeño tamaño en dos nodos distintos.....11

Fig.7: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 e intel MPI 5.1 intranodal e internodal.....11

Fig.8: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 e openMPI 1.10.2 intranodal e internodal.....12

Fig.9: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 e openMPI 2.0.0 intranodal e internodal.....12

Fig.10: Comparativa de comunicación punto a punto de dos procesos con intel 2016 e openMPI 2.0.1 intranodal e internodal.....13

Fig.11: Comparativa de comunicación punto a punto de dos procesos con Intel 2016 e intel MPI 5.1 intranodal e internodal.....13

Fig.12: Comparativa de comunicación punto a punto de dos procesos con Intel 2016 y openMPI 1.10.2 intranodal e internodal.....14

Fig.13: Comparativa de comunicación punto a punto de dos procesos con Intel 2016 y openMPI 1.10.2 con HPCX intranodal e internodal.....14

Fig.14: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de gran tamaño.....15

Fig.15: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de tamaño intermedio.....16

Fig.16: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de pequeño tamaño.....16

Fig.17: Comparativa de comunicación colectiva para las herramientas MPI para dos nodos, con un proceso por nodo.....17

Fig.18: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo.....17

Fig.19: Comparativa de comunicación colectiva para las herramientas MPI para ocho nodos, con un proceso por nodo.....18

Fig.20: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo.....18

Fig.21: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con doce procesos por nodo.....19

Fig.22: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con veinticuatro procesos por nodo.....19

Fig.23: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo, para mensajes pequeños.....20

Fig.24: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con doce procesos por nodo, para mensajes pequeños.....20

Fig.25: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con veinticuatro procesos por nodo, para mensajes pequeños.....21

Fig.26: Comparativa de comunicación colectiva de las diferentes herramientas 22

Tablas

Tabla 1: Herramientas MPI estudiadas vs. compiladores.....7

1. Introducción

En el siguiente estudio se representará de manera breve el rendimiento de diferentes herramientas de paralelización en el superordenador FinisTerrae II, instalado en el CESGA.
Los objetivos que se abordarán se resumen en los siguientes puntos:

a) Análisis de las diferentes herramientas sobre dos procesos: en un caso dentro de un mismo nodo, y, por otro lado, en dos nodos diferentes. De esta manera se podrá realizar una evaluación del rendimiento de las comunicaciones punto a punto.
b) Análisis de las diferentes herramientas en procesos repartidos entre dos, cuatro y ocho nodos. Se estudiará, por su parte, utilizando tanto uno, como doce y como veinticuatro procesos por cada nodo. Así se analizará la eficiencia de las operaciones colectivas.

2. Infraestructura

Para el pertinente análisis se utilizarán una serie de ocho nodos procedentes del superordenador FinisTerrae II, que presentan las siguientes características técnicas:

- 2 procesadores Haswell 2680v3 con 24 cores
- 128 GB de memoria
- 1 disco de 1 TB
- 2 conexiones 1 GbE
- 1 conexión Infiniband FDR@56Gbps

Por su parte, las herramientas MPI a comparar, son las siguientes:

- Intel MPI versión 5.1.3.210 instalado utilizando los compiladores de Intel 2016.
- OpenMPI versión 1.10.2 instalado con los compiladores de Intel 2016.
- OpenMPI versión 1.10.2 instalado utilizando los compiladores de Intel 2016 y con las herramientas de Mellanox HPC-X.
- OpenMPI versión 2.0.1 instalado utilizando los compiladores de Intel 2016.
- Intel MPI versión 5.1 instalado utilizando los compiladores de gcc 6.1.0.
- OpenMPI versión 1.10.2 instalado utilizando los compiladores de gcc 6.1.0.
- OpenMPI versión 2.0.0 instalado utilizando los compiladores de gcc 6.1.0.

En la Tabla 1, se muestran las combinaciones de compilador y librería de MPI anteriormente mencionadas.

		COMPILADOR	
		Intel 2016	gcc 6.1.0
HERRAMIENTA MPI	OpenMPI 1.10.2	X	X
	OpenMPI 1.10.2 HPCX	X	
	OpenMPI 2.0.0		X
	OpenMPI 2.0.1		X
	Intel MPI 5.1	X	X

Tabla 1: Herramientas MPI estudiadas vs. compiladores.

Los nodos empleados han sido los c6603 y 6604 para los estudios de comunicación de punto a punto y los nodos c7225 y 7228-7234 para los estudios de comunicación colectiva de las tres primeras herramientas MPI mencionadas; para el resto de herramientas MPI se han empleados los nodos c6641-6642 para los estudios de comunicación punto a punto y los c7301-7308 para los estudios de comunicación colectiva.

El benchmark que se ejecutará para el estudio es el facilitado por el paquete de Intel, versión 4.1.1, que ha sido compilado para cada una de las herramientas antes mencionadas.
Para la ejecución de las pruebas se utilizará el sistema de colas presente en la infraestructura FinisTerrae II: Slurm 14.11.10-Bull 1.0. El método de ejecución se ejemplifica en el bloque de Resultados.

3. Resultados

La primera parte del estudio se basa en las comunicaciones punto a punto. Para ello se han lanzado dos procesos en un mismo nodo, y, posteriormente, en dos nodos diferentes. El modo de ejecución ha sido el siguiente:

```
[irsantos@fs6801 src]$ srun -t 00:40:00 -p thinnodes -n2 -N1 --tasks-per-node=2 -w c7225 ./IMB-MPI1
```

```
[irsantos@fs6801 src]$ srun -t 00:40:00 -p thinnodes -n2 -N2 --tasks-per-node=1 -w c[7225,7228] ./IMB-MPI1
```

Para la interpretación de los resultados se utilizará el test Ping-Pong, usado para medir la puesta en marcha y el rendimiento de un mensaje enviado entre dos procesos. Los datos se representarán en dos gráficas para realizar la comparativa: por un lado se compararán las diferentes herramientas propuestas, y por otro lado se analizarán las diferencias que aparecen entre los procesos internodales y los intranodales.

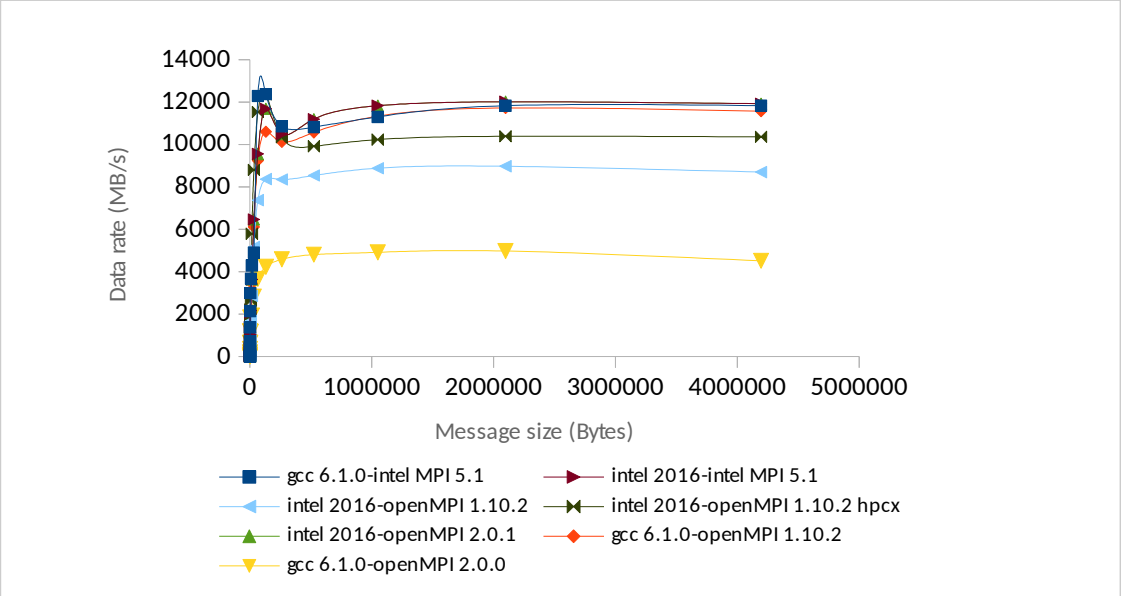

Figura 1: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de gran tamaño en un mismo nodo.

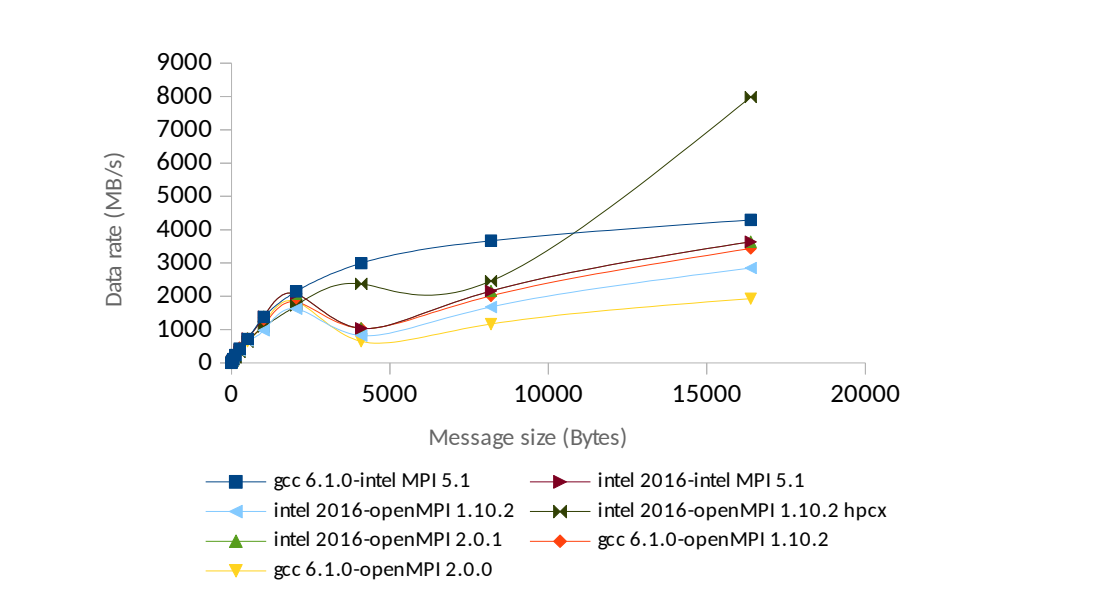

Figura 2: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de tamaño intermedio en un mismo nodo.

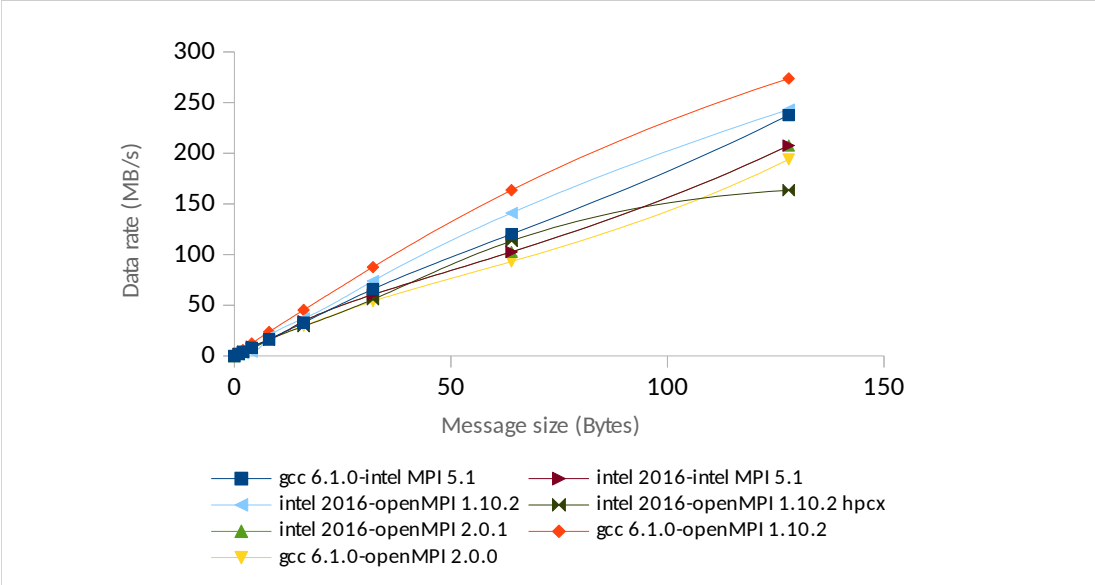

Figura 3: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de pequeño tamaño en un mismo nodo.

A continuación, se realiza exactamente la misma representación de resultados para el par de procesos en dos nodos diferentes.

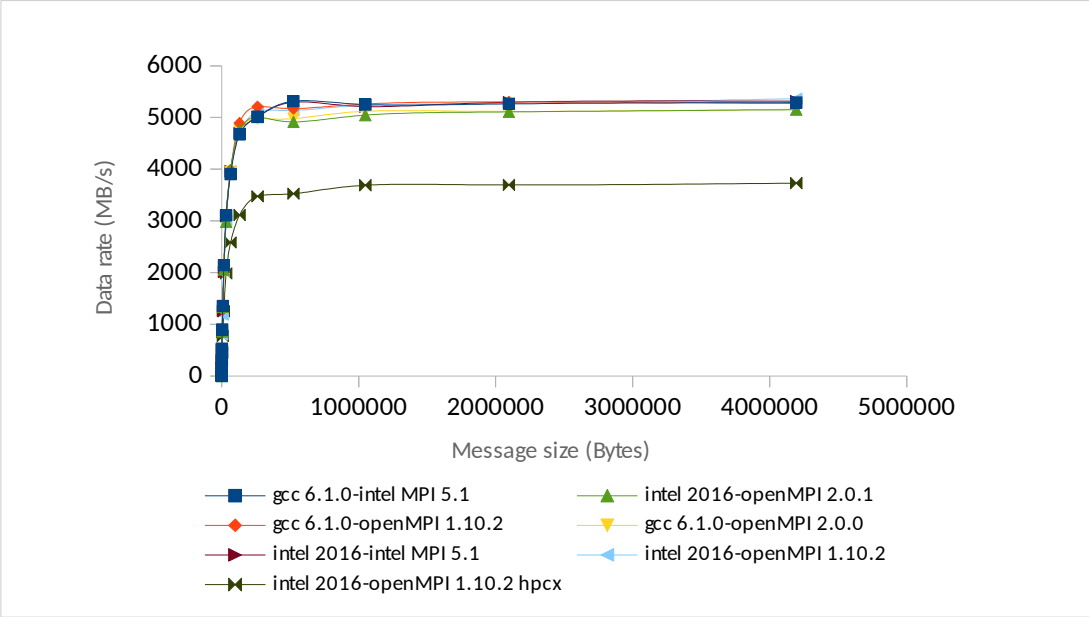

Figura 4: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de gran tamaño en dos nodos distintos.

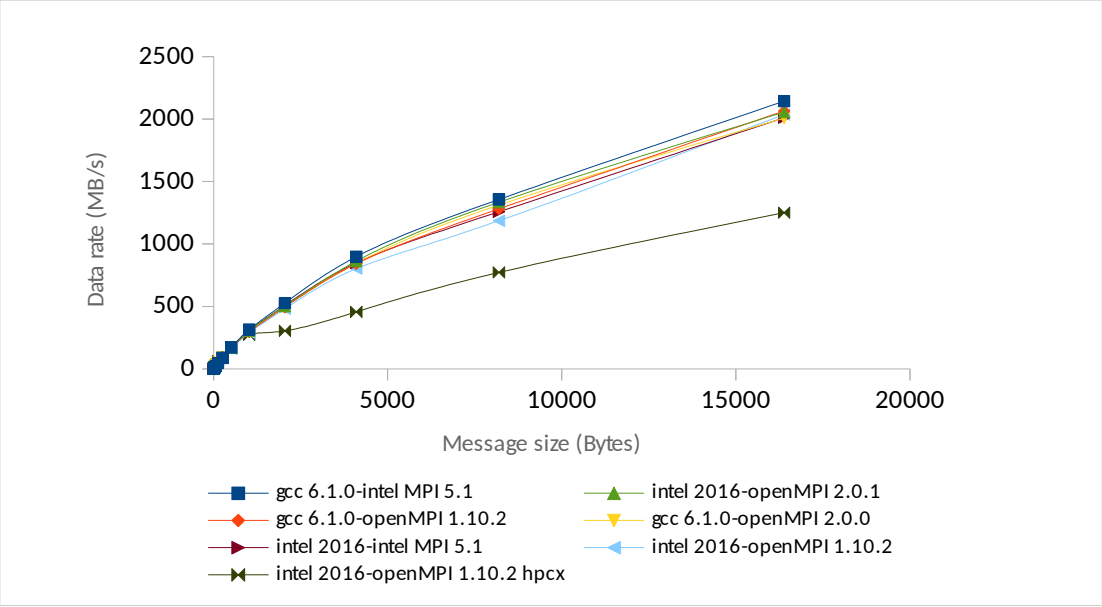

Figura 5: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de tamaño intermedio en dos nodos distintos.

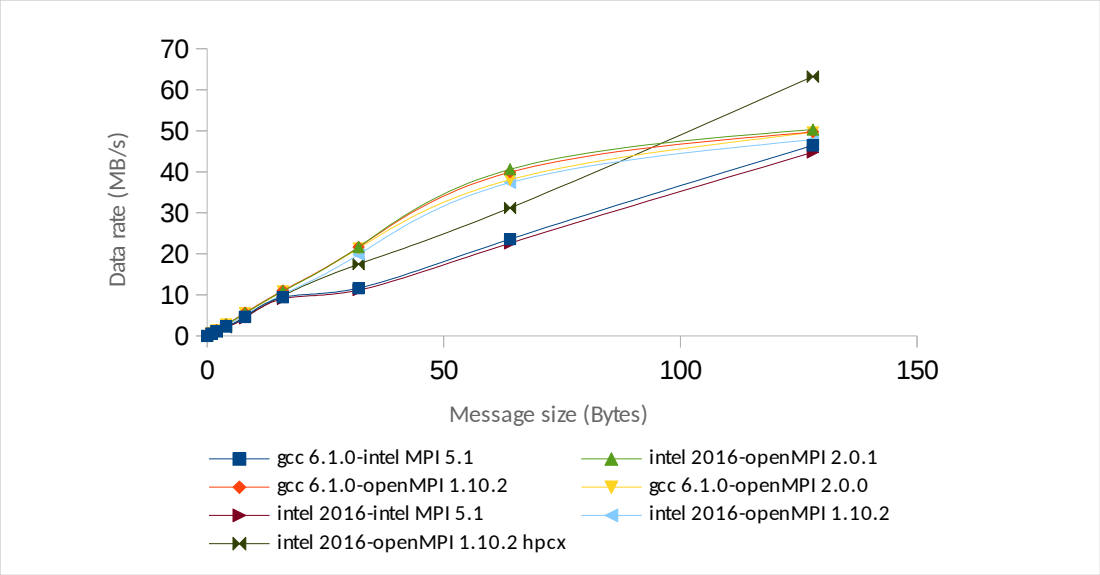

Figura 6: Comparativa de comunicación punto a punto de las diferentes herramientas para mensajes de pequeño tamaño en dos nodos distintos.

Posteriormente, se representan los resultados en una comparativa entre los procesos internodal e intranodal por cada herramienta MPI.

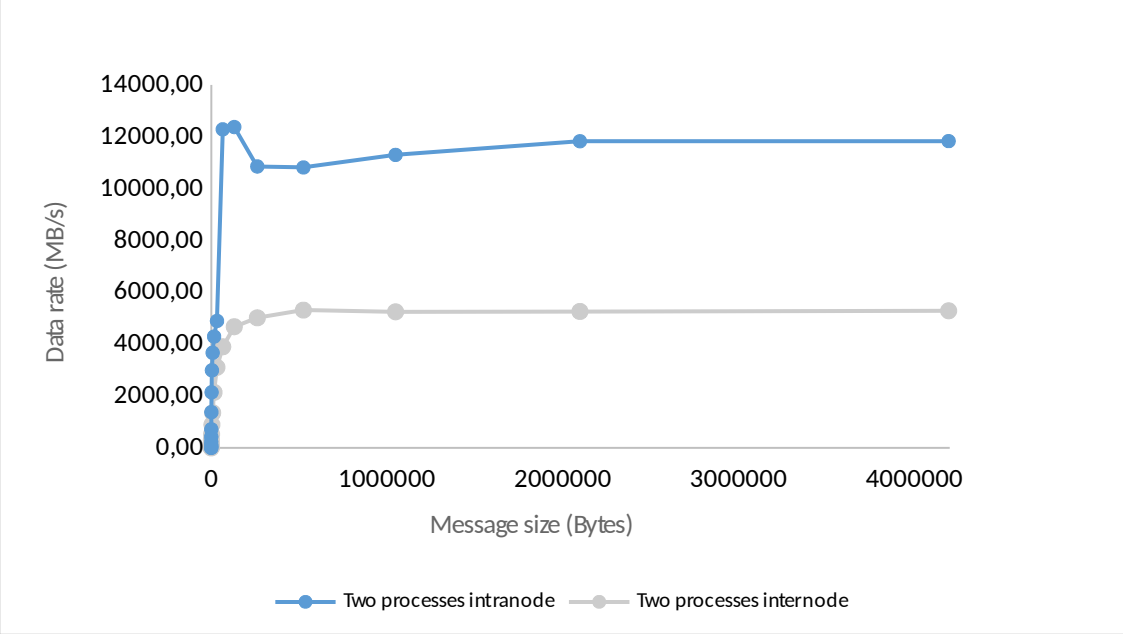

Figura 7: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 e intel MPI 5.1 intranodal e internodal.

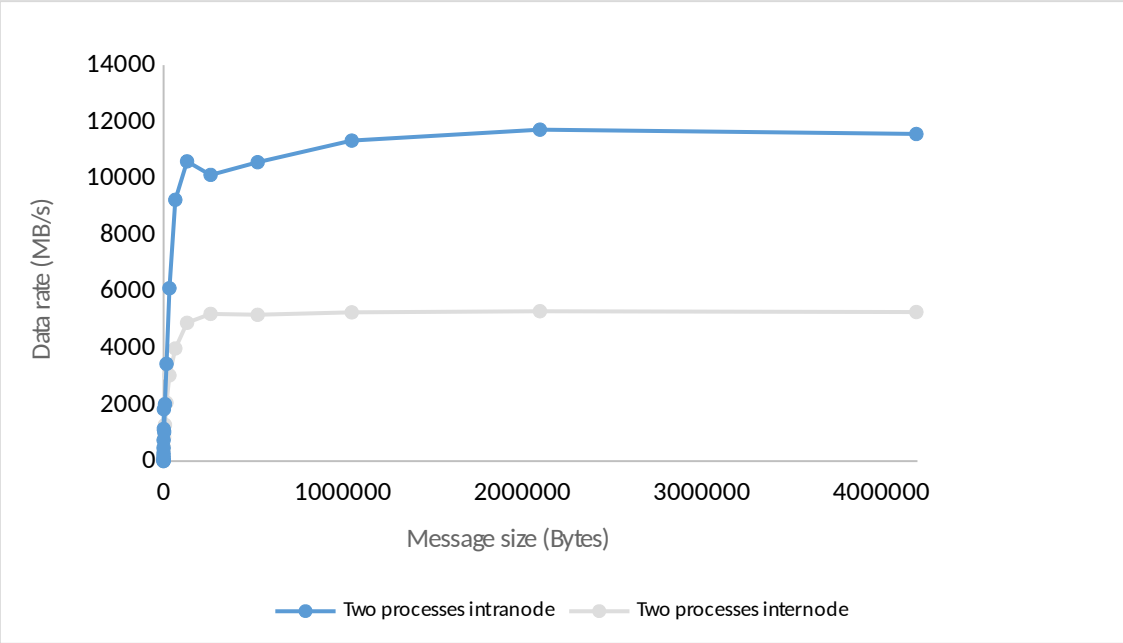

Figura 8: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 y openMPI 1.10.2 intranodal e internodal.

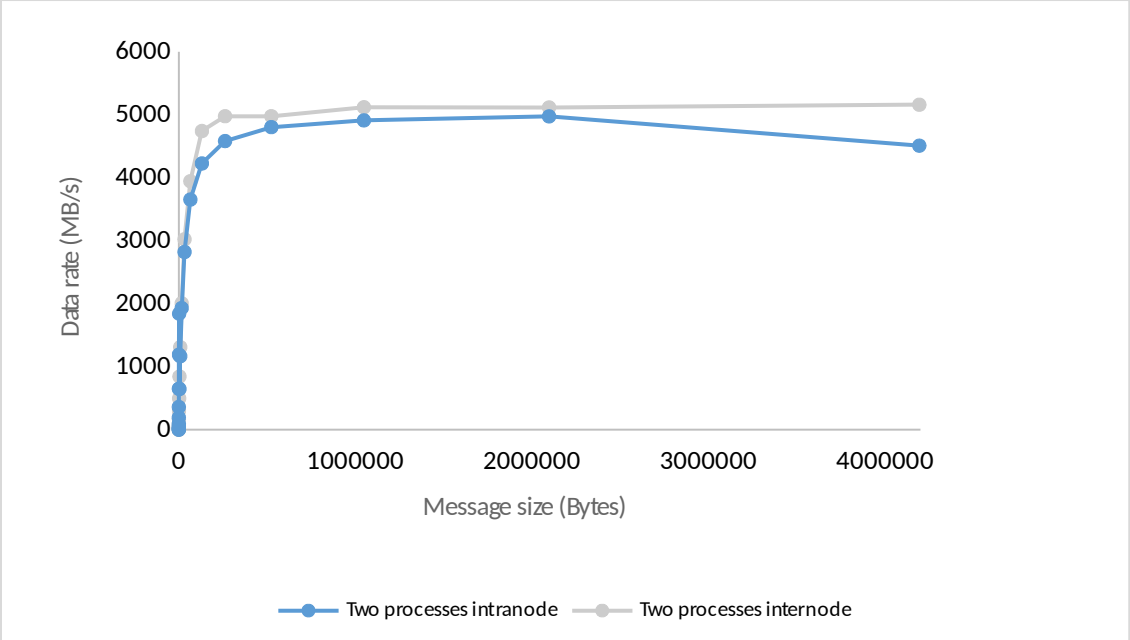

Figura 9: Comparativa de comunicación punto a punto de dos procesos con gcc 6.1.0 y openMPI 2.0.0 intranodal e internodal.

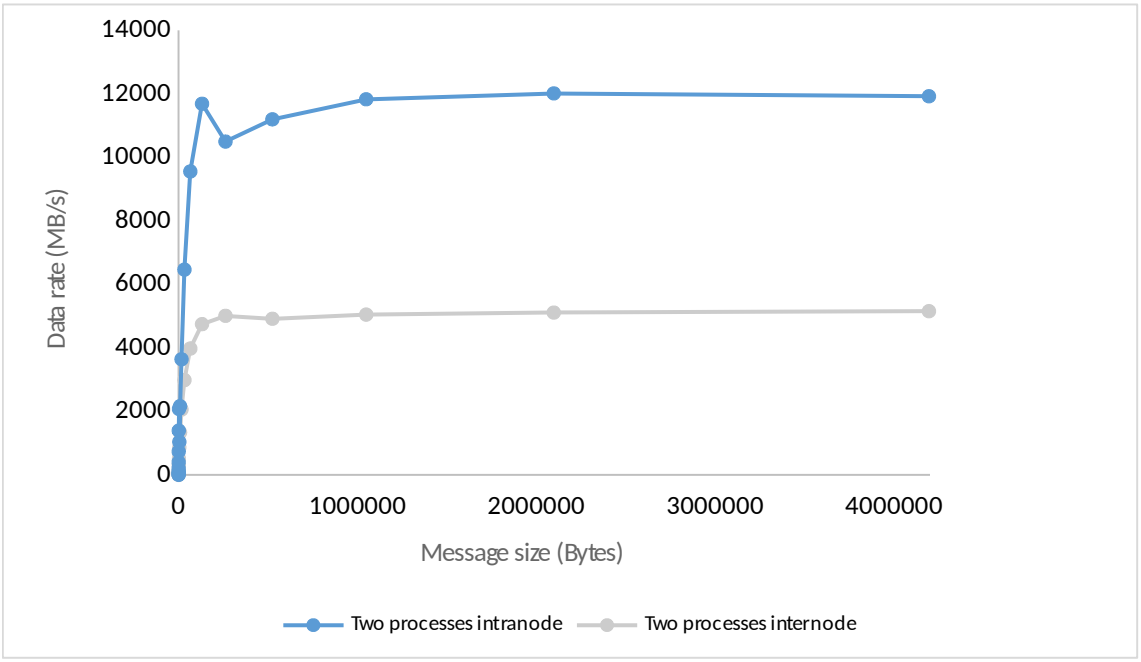

Figura 10: Comparativa de comunicación punto a punto de dos procesos con intel 2016 y openMPI 2.0.1 intrano-dal e internodal.

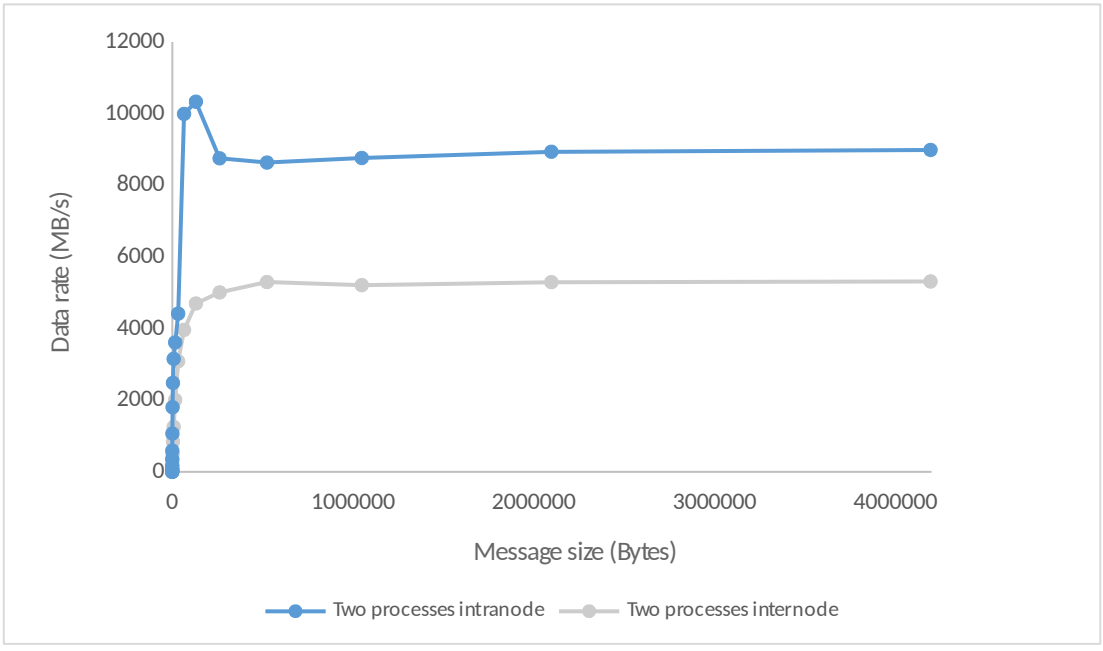

Figura 11: Comparativa de comunicación punto a punto de dos procesos con Intel 2016 e intel MPI 5.1 intranodal e internodal.

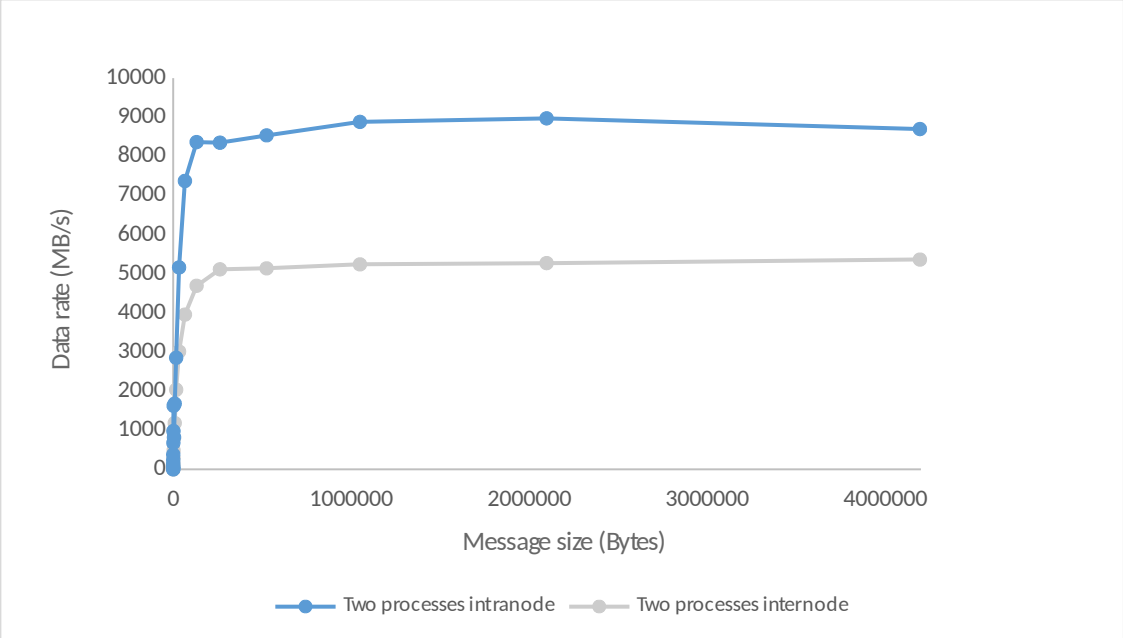

Figura 12*: Comparativa de comunicación punto a* punto de dos procesos con Intel 2016 y open*MPI 1.10.2 intranodal e internodal.*

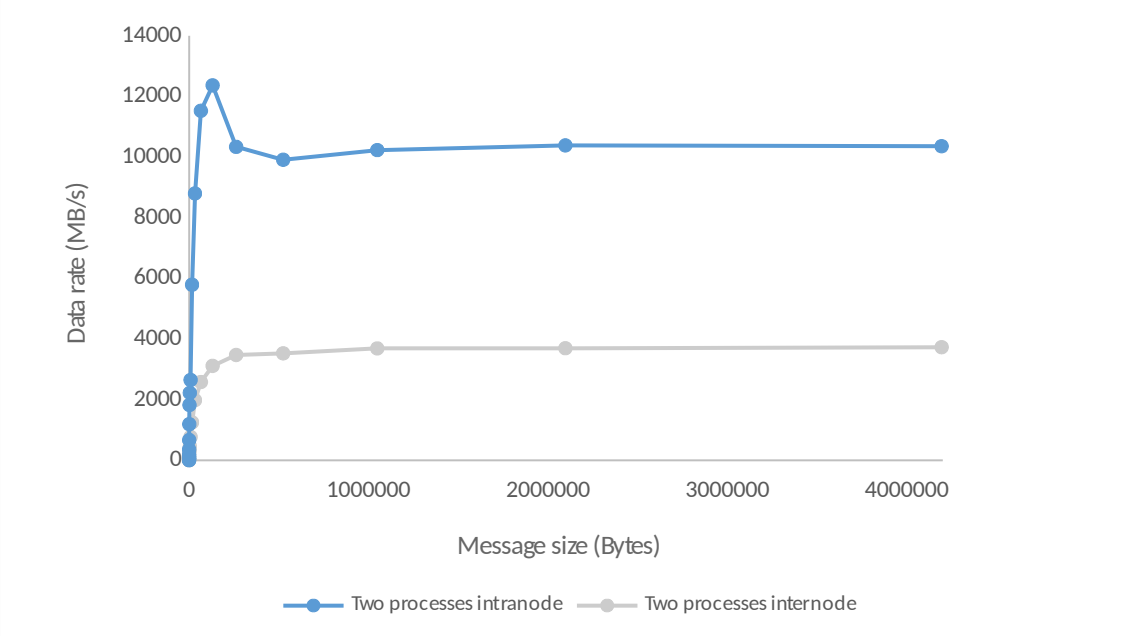

Figura 13: Comparativa de comunicación punto a punto de dos procesos con Intel 2016 y openMPI 1.10.2 con HPCX intranodal e internodal.

Finalmente, se procede al estudio de los procesos colectivos. Para ello se lanzará el conjunto de benchmarks para uno, doce y veinticuatro procesos por nodo; para dos, cuatro y ocho nodos, respectivamente. El modo de lanzamiento es el siguiente (se muestran los ejemplos solo para el lanzamiento de ocho nodos, el resto de casos es extrapolable):

```
[irsantos@fs6801 src]$ srun -t 00:40:00 -p thinnodes -n8 -N8 --tasks-per-node=1 -w c[7225,7228-7234] ./IMB-MPI1
```

```
[irsantos@fs6801 src]$ srun -t 00:40:00 -p thinnodes -n96 -N8 --tasks-per-node=12 -w c[7225,7228-7234] ./IMB-MPI1
```

```
[irsantos@fs6801 src]$ srun -t 00:40:00 -p thinnodes -n192 -N8 --tasks-per-node=24 -w c[7225,7228-7234] ./IMB-MPI1
```

La prueba escogida para este estudio será la de AllgatherV, en el que cada proceso envía X bytes de mensaje y recibe de cada uno del resto de procesos el mismo número de bytes que el propio envía. Por lo tanto, si “np” es el número de procesos, cada proceso recibirá np x X bytes del resto. En primer lugar se analizará la manera que afecta la comunicación colectiva según el tipo de herramienta MPI utilizada. Para ello se utilizarán los datos obtenidos usando cuatro nodos, con un proceso por cada nodo.

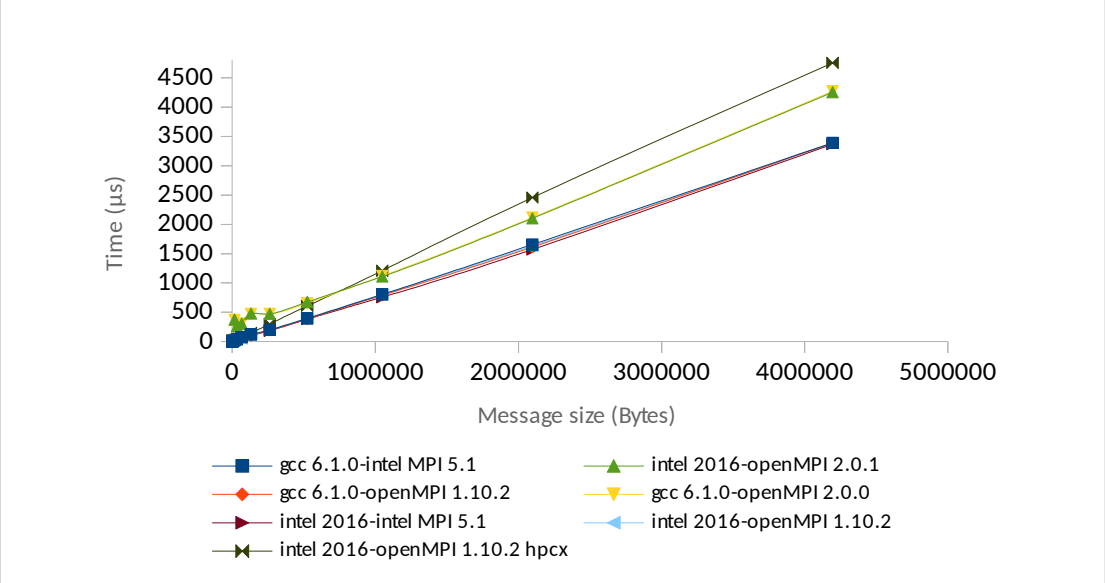

Figura 14: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de gran tamaño.

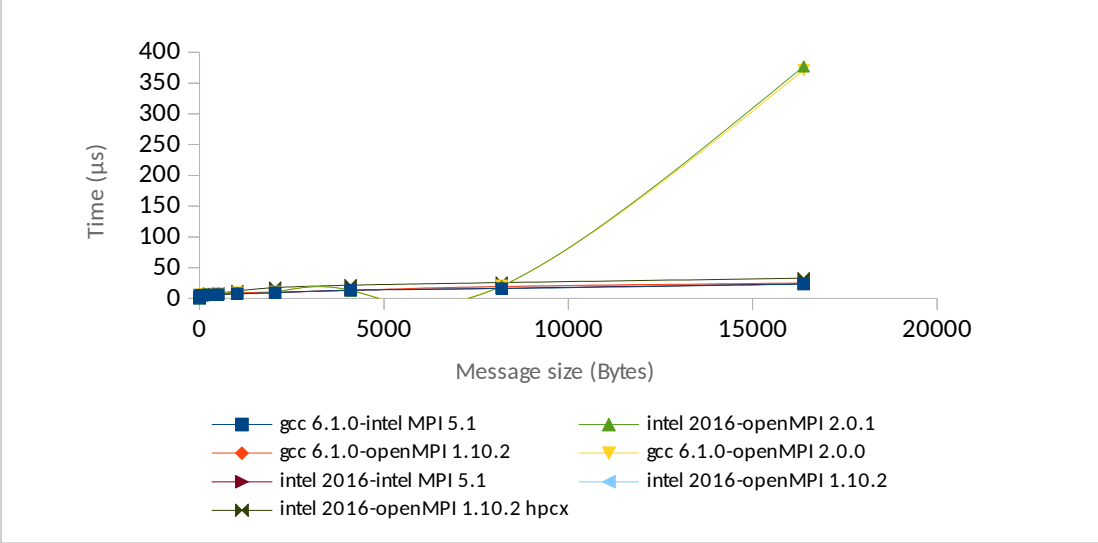

Figura 15: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de tamaño intermedio.

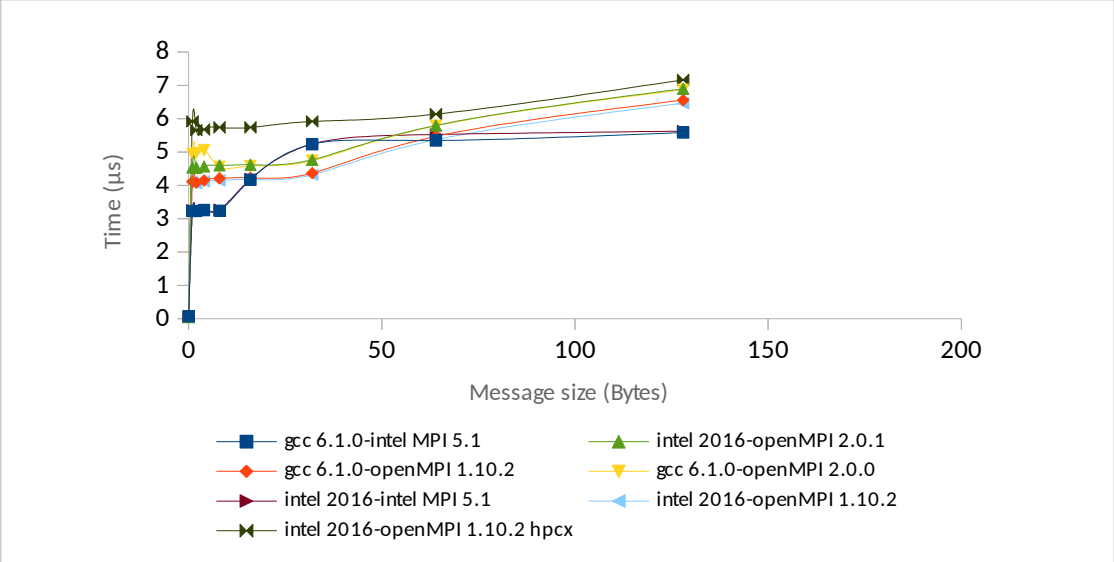

Figura 16: Comparativa de comunicación colectiva de las diferentes herramientas para mensajes de pequeño tamaño.

Posteriormente, se procede a estudiar gráficamente como afecta a cada uno de los MPI a analizar, el aumento del número de nodos, para un mismo número de procesos por nodo. En este caso se trabajará con un proceso por cada nodo.

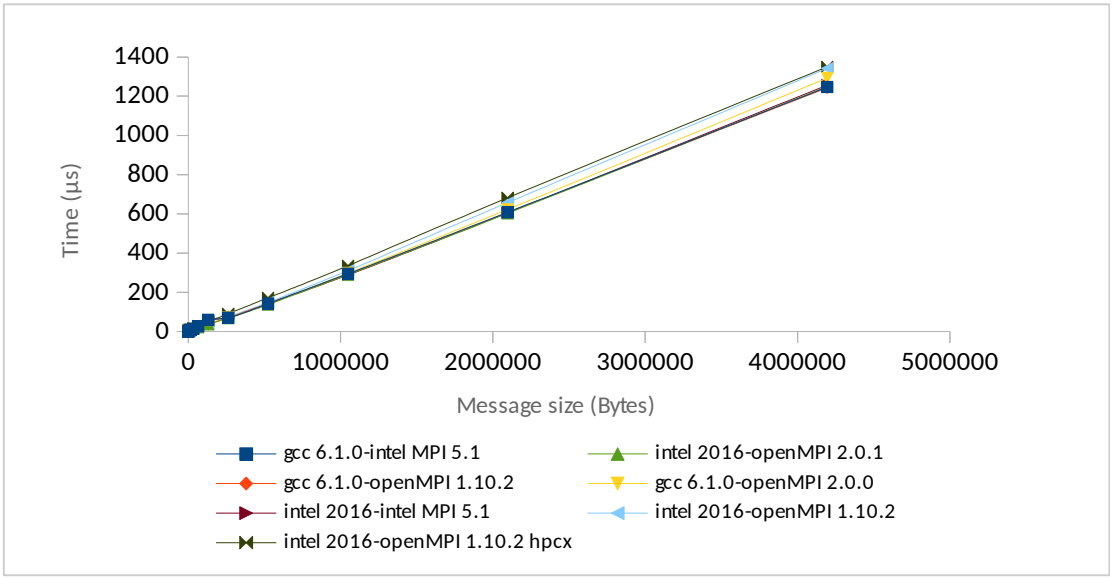

Figura 17: Comparativa de comunicación colectiva para las herramientas MPI para dos nodos, con un proceso por nodo.

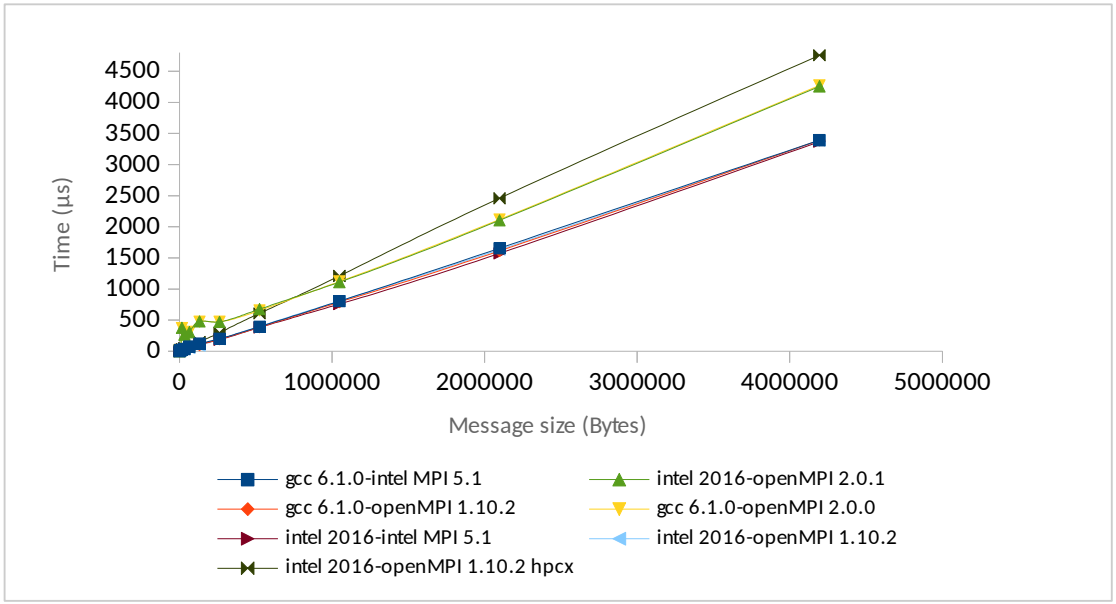

Figura 18: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo.

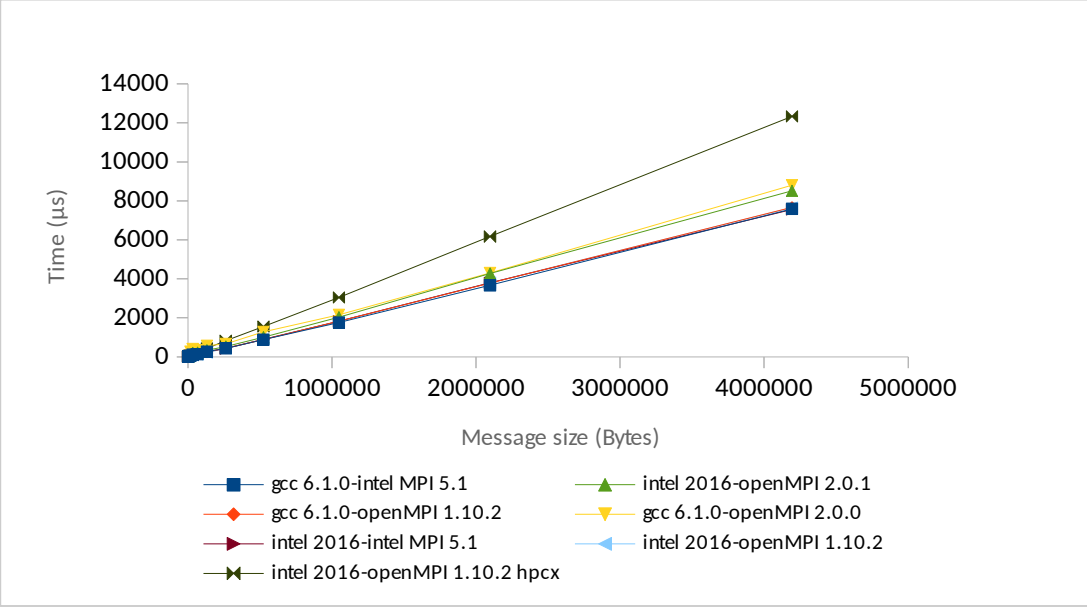

Figura 19: Comparativa de comunicación colectiva para las herramientas MPI para ocho nodos, con un proceso por nodo.

Finalmente, se realizará un análisis del efecto del aumento del número de procesos en el rendimiento de las herramientas MPI. Para ello se considera un número concreto de nodos, en este ejemplo se trabajará con cuatro nodos, pasando de uno, a doce y a veinticuatro procesos por cada nodo.

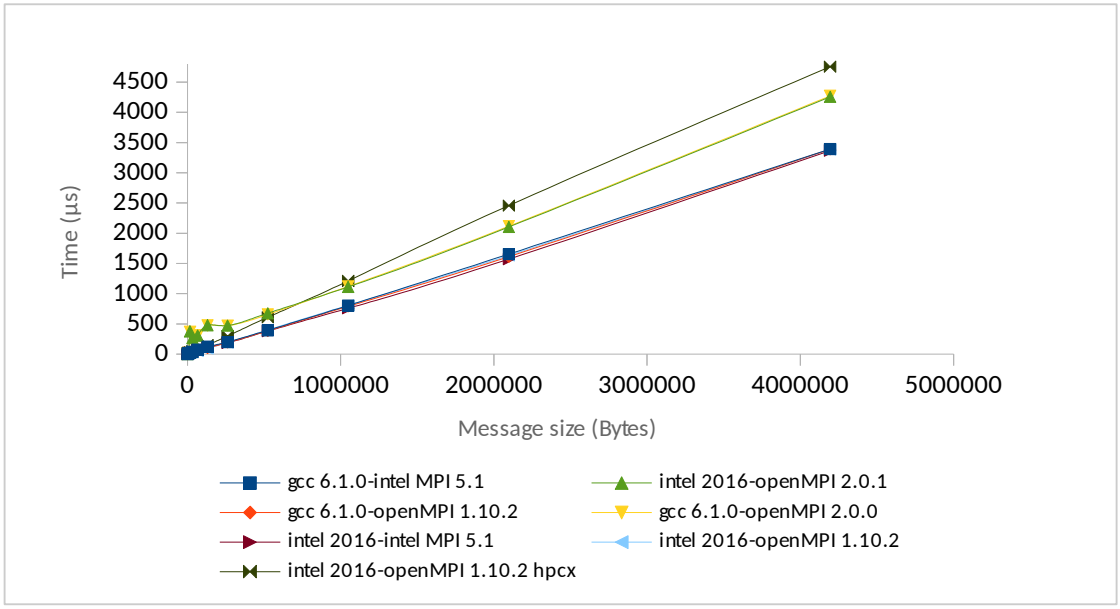

Figura 20: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo.

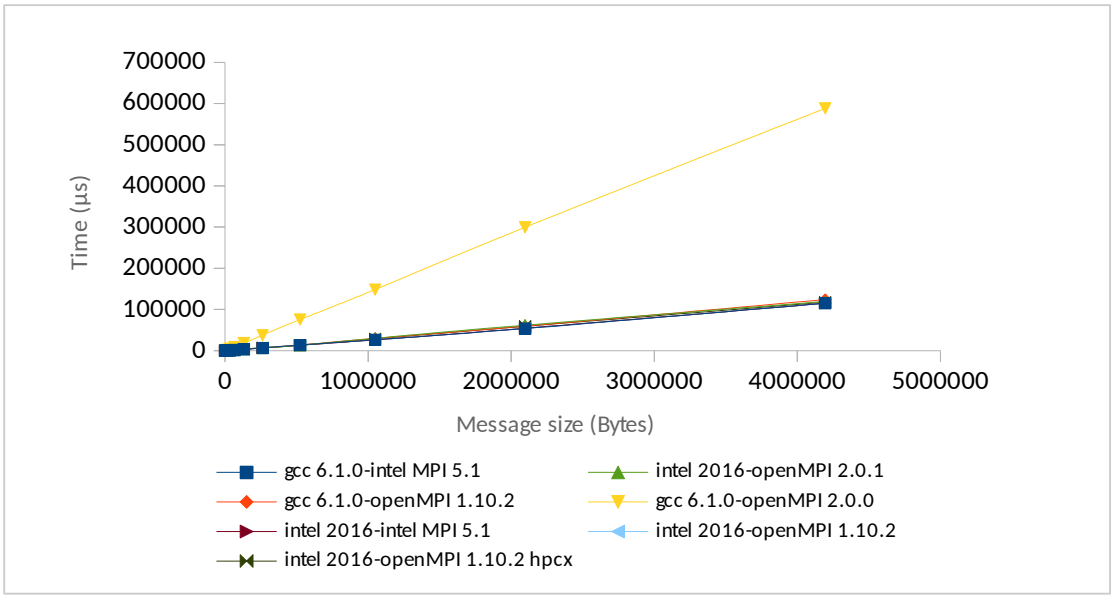

Figura 21: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con doce procesos por nodo.

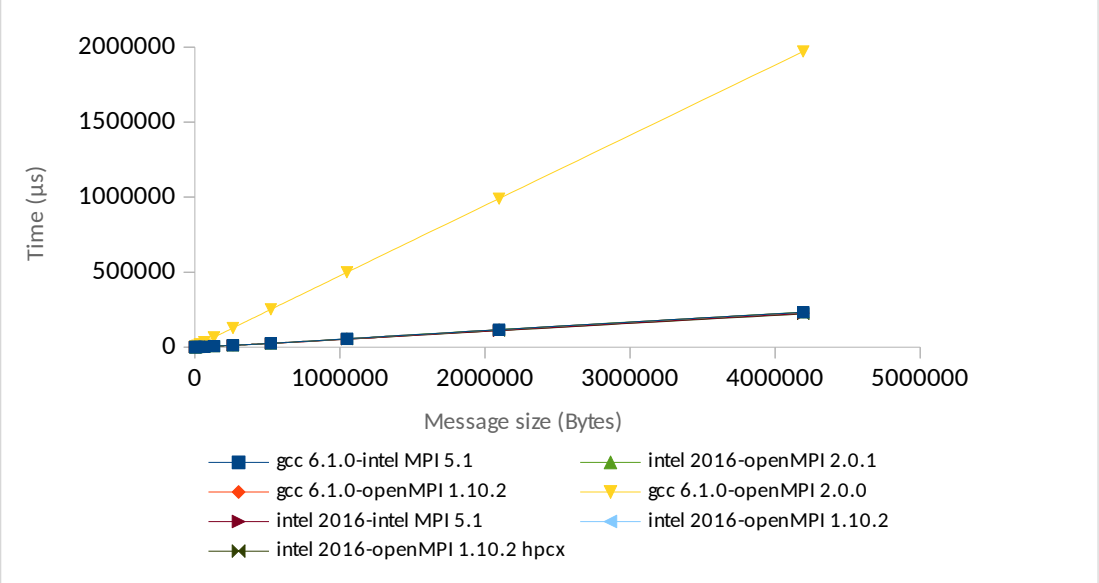

Figura 22: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con veinticuatro procesos por nodo.

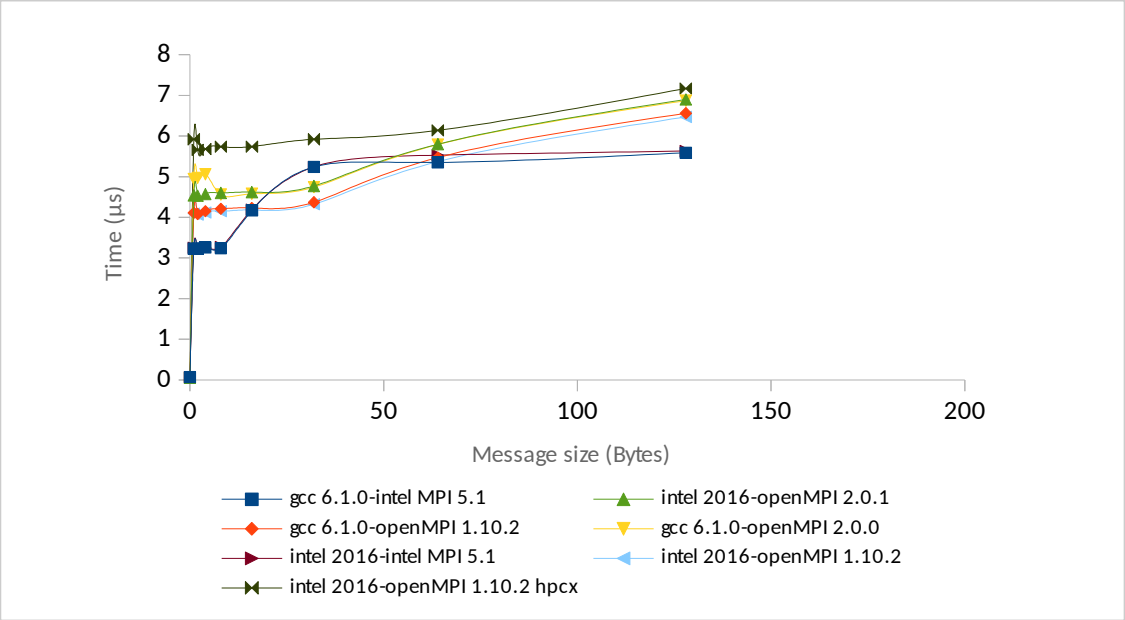

Figura 23: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con un proceso por nodo, para mensajes pequeños.

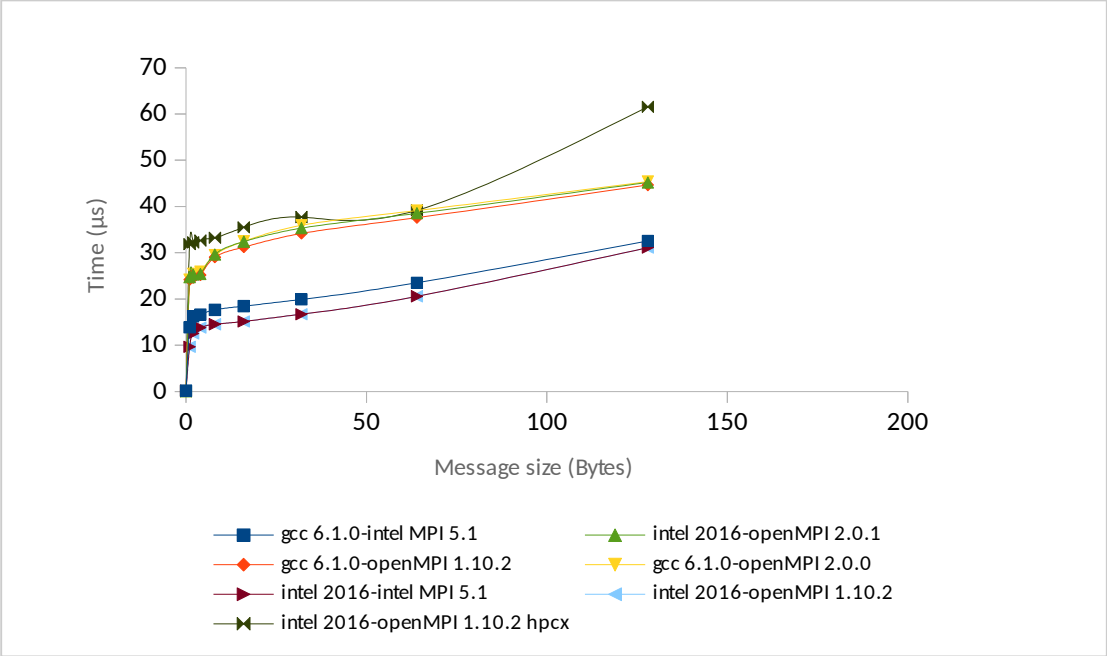

Figura 24: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con doce procesos por nodo, para mensajes pequeños.

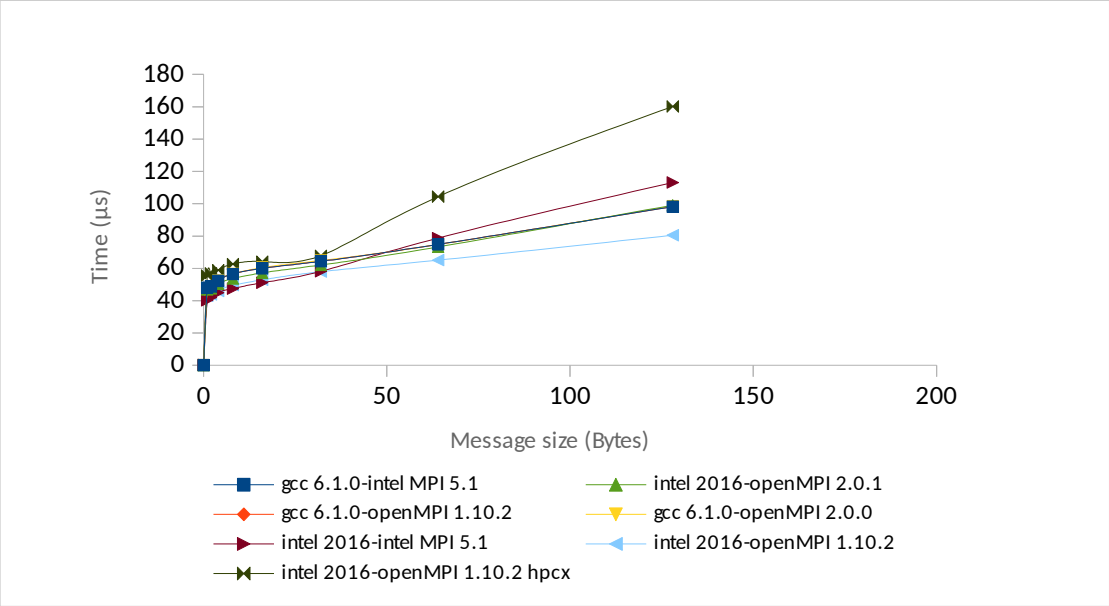

Figura 25: Comparativa de comunicación colectiva para las herramientas MPI para cuatro nodos, con veinticuatro procesos por nodo, para mensajes pequeños.

4. Discusión de resultados

4.1. Comunicación punto a punto

Se comienza analizando las diferencias que puedan presentarse en la comunicación punto a punto dentro de un mismo nodo. Se aprecian una serie de discrepancias dependiendo del tamaño del mensaje a enviar. En los tamaños grandes de mensaje (a partir de 1GB) se obtienen rendimientos similares al comparar las diferentes herramientas MPI, a excepción del openMPI 1.10.2 instalado con intel 2016 y openMPI 2.0.0 instalado con gcc 6.1.0 para los cuales se obtienen rendimientos menores, aproximadamente un 30 y 60% menos de rendimiento que para el resto de herramientas, respectivamente. Al disminuir el tamaño del mensaje (entre 5000 y 10000 bytes) el OpenMPI con HPCX presenta una disminución de su velocidad de envío. Finalmente, para tamaños pequeños de mensaje, el comportamiento es similar para todas las combinaciones de herramientas estudiadas, sin existir grandes diferencias de rendimiento (ver Figura 3).
Por su parte, al interpretar los datos de la comunicación internodal, se observa que el rendimiento del openMPI con HPCX instalado con intel 2016 es inferior al del resto de herramientas MPI a medida que aumenta el tamaño de mensaje, llegando a ser aproximadamente un 40% inferior para mensajes de gran tamaño (Figura 5).
Para finalizar con el análisis de este punto, se procede a realizar una comparativa de cada herramienta para la comunicación punto a punto. Como es obvio, para la mayoría de herramientas MPI se observa una disminución del rendimiento al pasar de la comunicación de dos procesos en un nodo a la comunicación en dos nodos, exceptuando el openMPI 2.0.0 compilado con gcc 6.1.0, que presenta ligeramente un mejor rendimiento en el proseso internodal. En el caso del openMPI 1.10.2 HPCX compilado con intel 2016, se aprecia una gran mejora en el rendimiento al pasar de la comunicación internodal a la intranodal, siendo la transferencia de datos aproximadamente tres veces mayor en el proceso intranodal que en el internodal. Estudiando cada una del resto de herramientas en particular (a excepción de los ya citados openmpi 2.0.0 compilado con gcc 6.1.0 y penMPI 1.10.2 HPCX compilado con intel 2016), se observa que la disminución de rendimiento entre los procesos internodal e intranodal se sitúa en valores de entre el 43 y 59%.

4.2. Comunicación colectiva

Antes de profundizar con los resultados obtenidos de la prueba AllGatherV seleccionada para el estudio, cabe destacar que de toda la serie de pruebas presentes en el paquete de Benchmarking de Intel, la prueba Gather no se ha podido completar para su ejecución con OpenMPI 1.10.2 compilado con Intel 2016 ni con OpenMPI 1.10.2 compilado con gcc 6.1.0 a partir de los 2048 bytes de mensaje en 32 procesos.
En cuanto a los resultados obtenidos en la prueba de AllGatherV, se empieza observando la tendencia para cada herramienta al utilizar cuatro nodos y un proceso por nodo. Al igual que se ha realizado en la comunicación punto a punto, se analizará el comportamiento según el tamaño de mensaje. En cuanto a los mensajes de gran tamaño (ver Figura 14) se aprecia que el OpenMPI 1.10.2 con HPCX presenta un aumento del tiempo de latencia, y por tanto una bajada de rendimiento de, aproximadamente, un 29%, con respecto a las otras dos herramientas compiladas con Intel 2016 (OpenMPI 1.10.2 e Intel MPI 5.1). Esta bajada de la eficiencia también se ve reflejada al disminuir el tamaño del mensaje, aunque cada vez se va haciendo menos pronunciada. Por otra parte, se vislumbra un comportamiento similar para mensajes grandes en el caso de OpenMPI 2.0.1 compilado con Intel 2016 y OpenMPI 2.0.0 compilado con gcc 6.1.0, los cuales presentan rendimientos mejores que el OpenMPI 1.10.2 con HPCX compilado con Intel 2016, pero peores que el resto de herramientas. En el caso de los mensajes de tamaño intermedio de 9000 a 15000 bytes el tiempo de latencia de estas dos herramientas aumenta con respecto a las otras, reduciendo notablemente su eficacia. A pesar de esto, entre 20 y 50 bytes, los Intel MPI presentan una reducción de su eficacia respecto a los OpenMPI sin HPCX, como se ve en el gráfico siguiente.

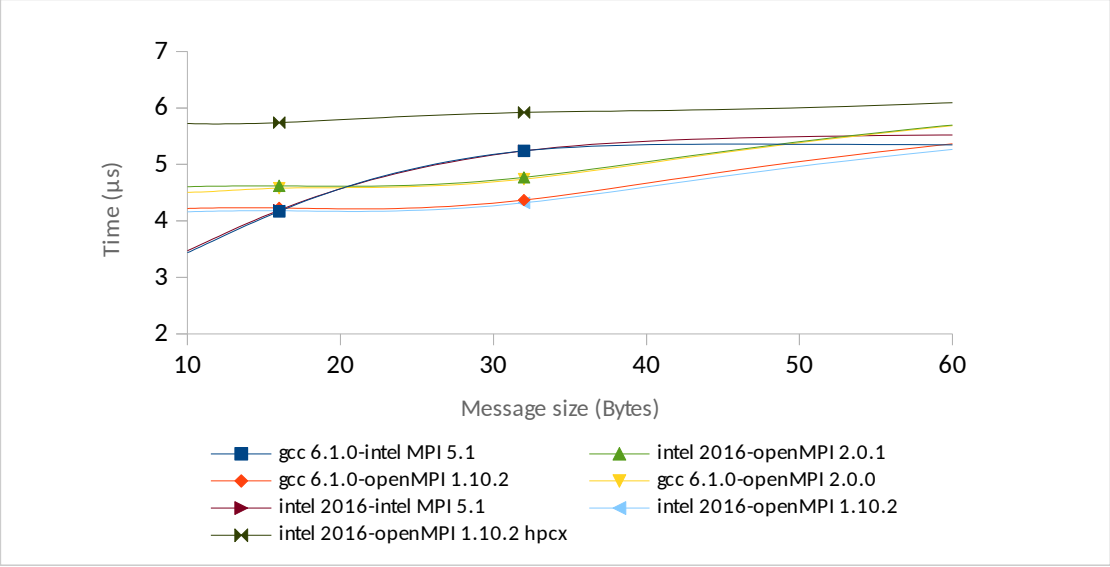

Figura 26: Comparativa de comunicación colectiva de las diferentes herramientas.

A continuación se ha profundizado más con el análisis estudiando la manera en la que afecta al rendimiento de cada herramienta el aumento del número de nodos en la comunicación. Como se puede observar, para dos nodos (Figura 17) los rendimientos son similares para todas las herramientas, sin embargo, al aumentar el número de nodos a 4 (Figura 18), se observa una menor eficacia del openMPI 1.10.2 con HPCX compilado con Intel 2016, y, también, aunque menos notable, una reducción del rendimiento del openMPI 1.10.2 compilado con intel 2016 y el openMPI 2.0.0 compilado con gcc 6.1.0. Esto se acentúa más al pasar a ocho nodos (Figura 19), donde el aumento del tiempo del openMPI 1.10.2 compilado con intel 2016 con la optimización HPCX llega a ser del 63% respecto a las herramientas que presentan mejores rendimientos.
Finalmente se hará una observación muy similar a lo anterior al estudiar el rendimiento al ir aumentando el número de procesos con la misma cantidad de nodos. Por lo visto en los resultados de los estudios de comunicación punto a punto, cabe esperar que al ir aumentando el número de procesos, se observe una mejoría del openMPI con HPCX respecto al resto, ya que su comunicación intranodal es destacada. Efectivamente, ocurre lo esperado, y, al trabajar con un proceso por nodo, se

ve un tiempo de latencia un 41% mayor para el openMPI optimizado respecto a las otras herramientas MPI compiladas con intel 2016 (Figura 20); el cual se reduce a un 2% con doce procesos por nodo (Figura 21); y que incluso llega a superar a los otros dos un 1% para los nodos completos con veinticuatro procesos (Figura 22). Por el contrario, para le openMPI 2.0.0 compilado con gcc 6.1.0 se observa el comportamiento opuesto, ya que al aumentar el número de procesos por nodo aumenta el tiempo de latencia respecto al resto de herramientas MPI estudiadas.

5. Conclusiones

En resumen, salvo para las herramientas openMPI compiladas con gcc 6.1.0 y para el openMPI 1.10.2 con HPCX compilado con intel 2016, en la mayoría de resultados obtenidos se observa una misma tendencia en los rendimientos de las herramientas MPI estudiadas . Unido a esto cabe destacar que en la comunicación colectiva, tanto el OpenMPI 1.10.2 compilado con Intel 2016 como el compilado con gcc 6.1.0 no ha podido desarrollar el benchmark Gather a partir de 2048 bytes de tamaño de mensaje para 32 procesos. Este hecho debería ser estudiado y conocer las consecuencias que pudiese provocar. Por otro lado se encuentra el OpenMPI 1.10.2 con la optimización Mellanox compilado con Intel 2016, cuyo comportamiento permite sacar más conclusiones. Se ha comprobado que, efectivamente, su uso en varios procesos dentro de un mismo nodo otorga una serie de mejoras en el rendimiento frente a las otras dos herramientas; pero cuando se trabaja con una comunicación entre nodos, este rendimiento se ve bastamente reducido, llegando a ser muy inferior a los demás. Se debería buscar las causas de esta bajada de eficiencia internodal e intentar conseguir una actualización que minimice estos conflictos.
En el sistema FinisTerrae II sería recomendable la utilización del intel MPI, ya que prácticamente obtiene mejores rendimientos para todos los casos. Además de ya encontrarse configurado y de su facilidad de uso. Se podría valorar la utilización del OpenMPI para el envío de mensajes MPI de tamaño pequeño, entre 16 y 64 bytes. Una alternativa viable sería la optimización avanzada del OpenMPI en presencia de Mellanox para escapar de las bajadas de rendimiento presentes y conseguir unos resultados más satisfactorios. Por último, está pendiente de profundizar en llos motivos del bajo rendimiento del OpenMPI 2.0.0 compilado con gcc en las comunicaciones dentro de nodo.