



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

anlisis de recursos mpi en biomed

Mihaita Alexandru Lupoiu

8 Abril 2016

Modelos de Programacin en Grid (Mpg)

trabajo propuesto

1. Introducin
2. Ejemplo LDL'
3. Configuracin
4. Resultados Ping Pong
5. Resultados LDL'
6. Conclusion y Trabajos Futuros
7. Cdigo

introducín

El objetivo de este proyecto es analizar los recursos MPI de Biomed. Para ellos se van a analizar los recursos mediante:

- El anlisis del retardo que existe a la hora de enviar un mensaje mediante el programa Ping Pong.
- El anlisis del rendimiento de una aplicacin que tiene un coste computacional importante y adems que tiene dependencia de datos.

La aplicacin que tiene un coste computacional importante es la la factorizacin LDL', que es una forma de factorizacin de una matriz A como el producto de una matriz triangular inferior L por una diagonal D y por una matriz inferior traspuesta L'.

$$A = L * D * L^T$$

Para evitar complicaciones las pruebas se harn solo para matrices simtricas, eso significa que $A = A'$.

ejemplo ldl'

ejemplo ldl'

Ejemplo sin sobre-escritura:

$$A = \begin{bmatrix} 4 & 3 & 1 & 1 \\ 3 & 8 & 1 & 2 \\ 1 & 1 & 16 & 1 \\ 1 & 2 & 1 & 10 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 \\ 0.25 & 0.0435 & 1 & 0 \\ 0.25 & 0.2174 & 0.0442 & 1 \end{bmatrix} \times \begin{bmatrix} 4 \\ 5.75 \\ 15.7391 \\ 9.4475 \end{bmatrix} \times \begin{bmatrix} 1 & 0.75 & 0.25 & 0.25 \\ 0 & 1 & 0.0435 & 0.2174 \\ 0 & 0 & 1 & 0.0442 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$L \qquad D \qquad L'$

Ejemplo con sobre-escritura:

$$A = \begin{bmatrix} 4 & 3 & 1 & 1 \\ 3 & 8 & 1 & 2 \\ 1 & 1 & 16 & 1 \\ 1 & 2 & 1 & 10 \end{bmatrix}$$

$$LDL' = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0.75 & 5.75 & 0 & 0 \\ 0.25 & 0.0435 & 15.7391 & 0 \\ 0.25 & 0.2174 & 0.0442 & 9.4475 \end{bmatrix}$$

configuracin

Para el lanzamiento de la aplicacin en BIOMED se tienen que crear varios ficheros:

1. El fichero de pre configuracin "pre-hook.sh" que se encargar de compilar el programa en la mquina remota, copiar ficheros necesarios, etc.
2. El fichero de "start.sh", para que se puedan utilizar distintas versiones de MPI. En este caso se utiliza el mismo que se utiliz en las practicas.
3. El fichero "program.jdl" que se encarga de especificar los requisitos de la aplicacin para que se pueda ejecutar con exito.
4. El fichero de post configuracin "post-hook.sh" que se encargar de realizar el procesamiento de los resultados de la aplicacin en caso de que sea necesario. En este caso no lo es.

El fichero de configuracin en este caso es el siguiente:

```
1  #!/bin/sh
2  pre_run_hook () {
3      tar xzvf file.tar.gz
4      make
5
6      return 0
7  }
```

Dentro de "file.tar.gz" est todo el cdigo de la aplicacin que se quiere lanzar y el makefile.

El fichero de los requisitos del sistema es el siguiente:

```
1  JobType           = "Normal";
2  nodeNumber        = 4;
3  Executable         = "starter.sh";
4  Arguments          = "mpi.out OPENMPI";
5  InputSandbox       = {"starter.sh", "ldl.tar.gz", "pre-hook.sh"};
6  StdOutput          = "std.out";
7  StdError           = "std.err";
8  OutputSandbox      = {"std.out", "std.err"};
9  Requirements       =
10 Member("MPI-START", other.GlueHostApplicationSoftwareRunTimeEnvironment)
11 && Member("OPENMPI", other.GlueHostApplicationSoftwareRunTimeEnvironment)
12 && Member("MPI-ETHERNET", other.GlueHostApplicationSoftwareRunTimeEnvironment);
13 Environment        = {"I2G_MPI_PRE_RUN_HOOK=pre-hook.sh"};
```

Los comandos utilizados fueron:

- Para listar los recursos donde se puede ejecutar la aplicacin:

```
glite-wms-job-list-match -a program.jdl
```

- Para el envo del y arranque de la aplicacin:

```
glite-wms-job-submit -a program.jdl
```

- Para cancelar una aplicacin:

```
glite-wms-job-cancel https://URL_submit
```

- Para la observar el estado de la aplicacin:

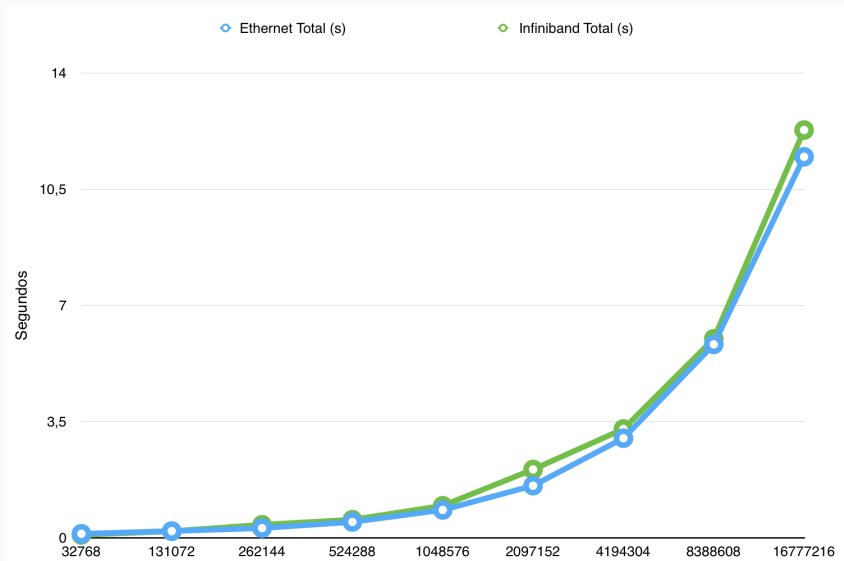
```
glite-wms-job-status https://URL_submit
```

- Para recuperar la salida estndar de la aplicacin:

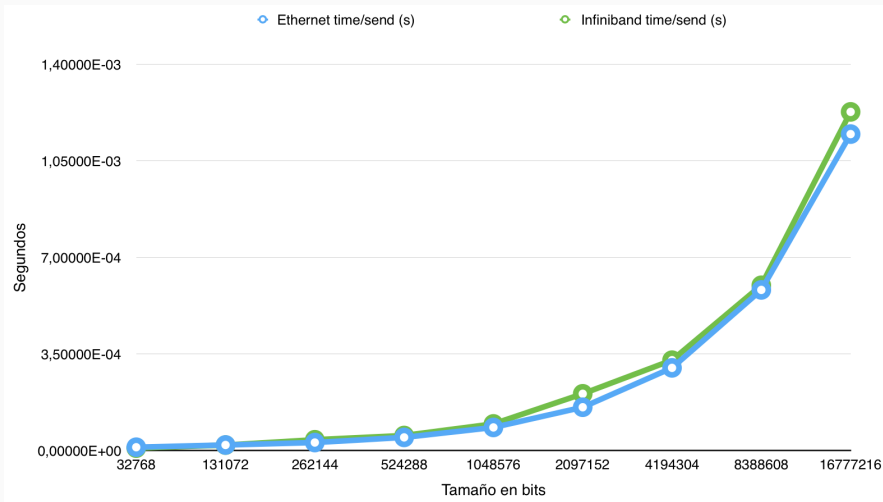
```
glite-wms-job-output --dir . https://URL_submit
```

resultados ping pong

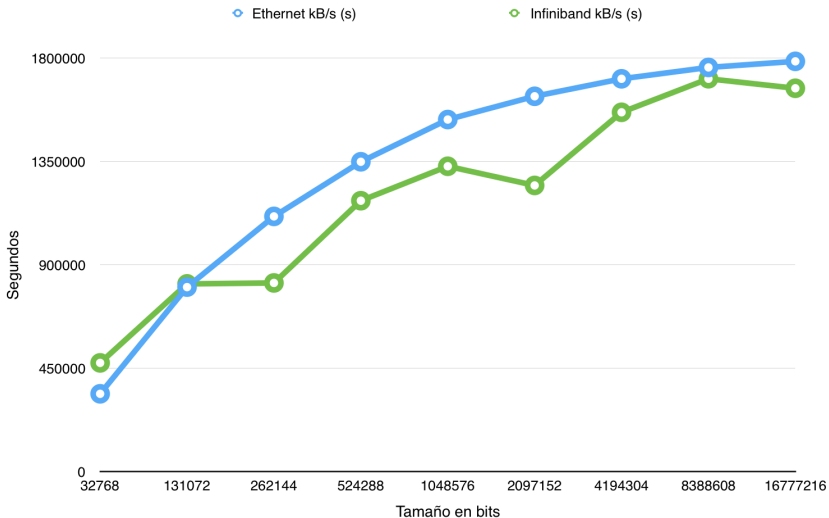
resultados: tiempo empleado



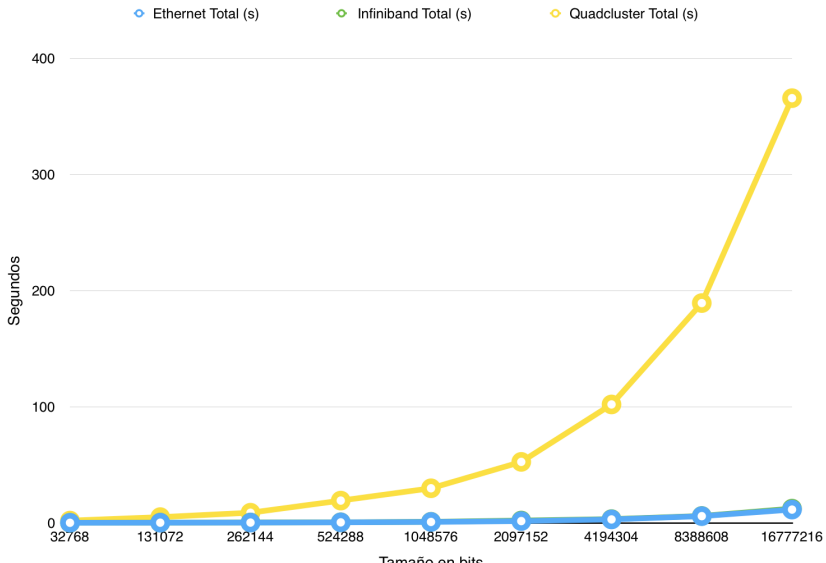
resultados: tiempo/envío (s) 4 kb



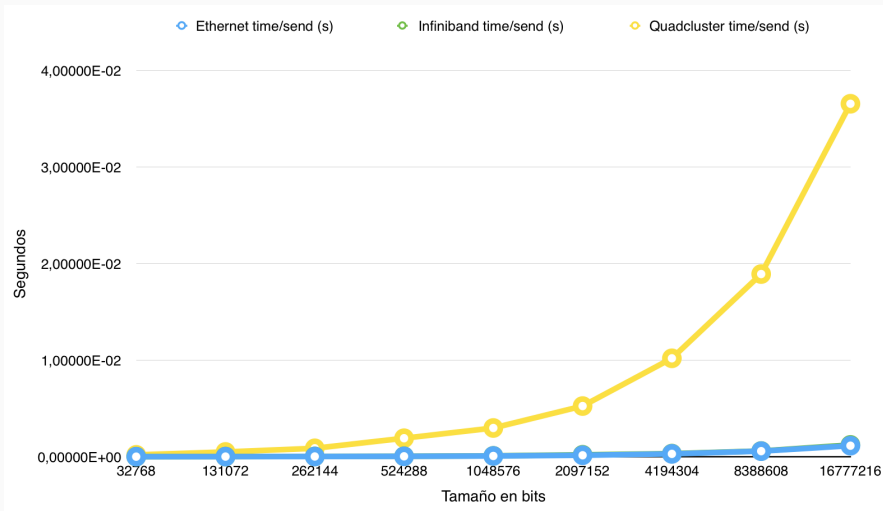
resultados: kb/s 4 kb



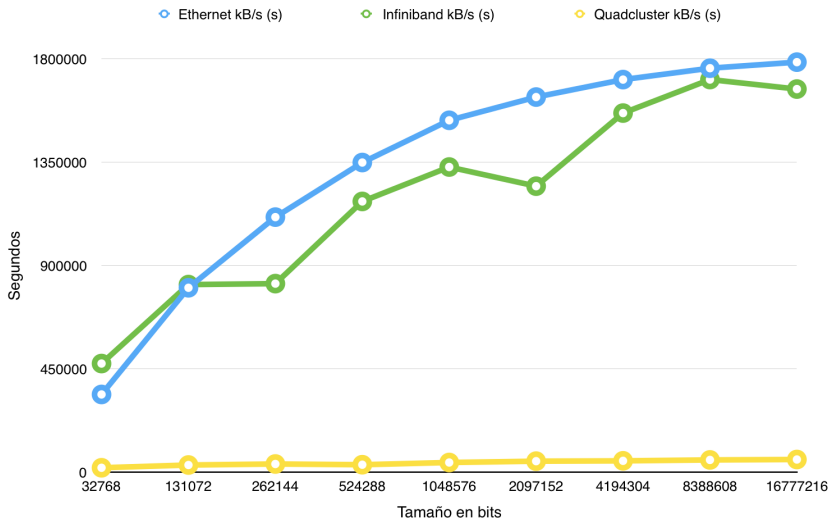
resultados: tiempo empleado



resultados: tiempo/envío (s) 128kb

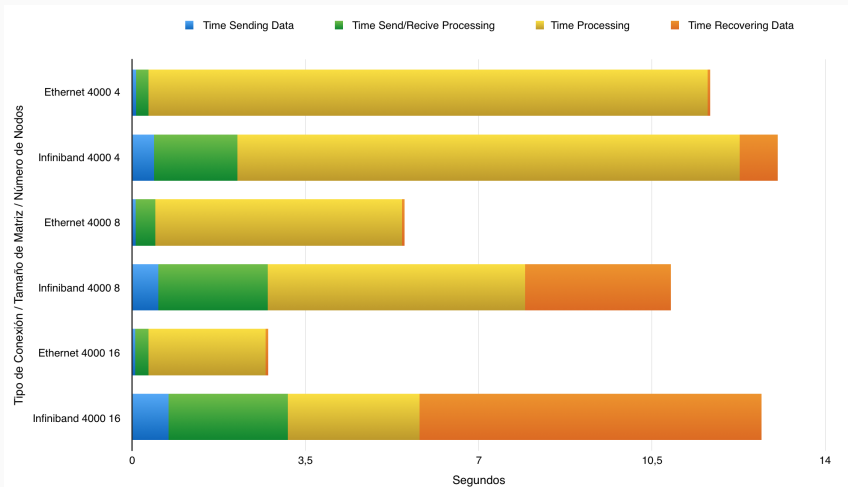


resultados: kb/s 128kb

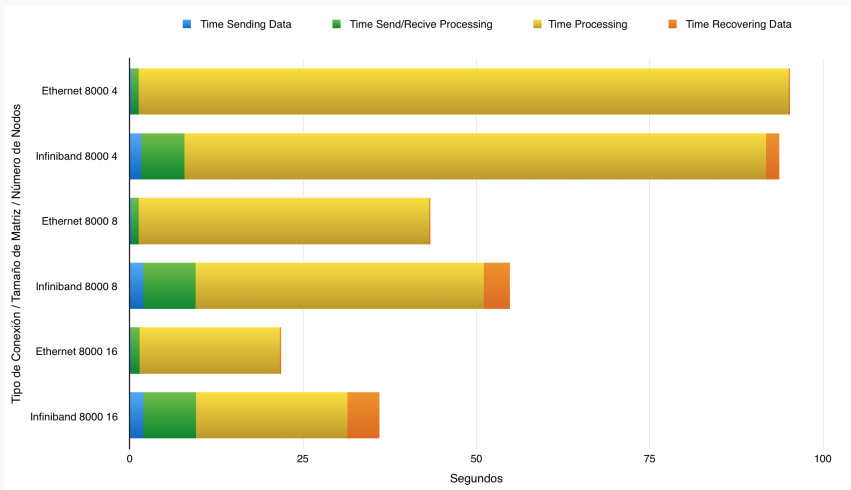


resultados IdI'

resultados: matriz 4000x4000



resultados: matriz 8000x8000



conclusion y trabajos futuros

conclusion y trabajos futuros

El algoritmo de la factorización LDL^T implementar de manera concurrente y con unas mejoras considerables.

Se ha podido observar que el cloud computing de Azure puede ser una buena alternativa y mucho ms barata para clculos pesados si no se dispone de un equipo adecuado.

Para futuras mejoras se debera de utilizar BLASH para la actualizacin de la matriz en MPI y utilizar un algoritmo por bloques para reducir las comunicaciones.

Se debera de intentar realizar pruebas con ms de 4 nodos de tipo D1 para ver cuando se puede conseguir de mejor, adems se deberan de probar otros proveedores de cloud computing como Amazon para comparar sus servicios.

cdigo

Pueden conseguir todo el cdigo fuente en

`github.com/MihaiLupoiu/MPG`

Todo bajo la licencia Creative Commons Attribution-ShareAlike 4.0 International License.



Gracias!

Alguna pregunta?