

Instituto Tecnológico de Costa Rica  
Sede Central Cartago

Escuela de Computación  
Arquitectura de Computadoras

Cluster Kabré + OpenMP

Alonso Jiménez Loaiza

2017142321

Fabrizio Ceciliano Navarro

2017111236

Profesor: Esteban Arias Méndez

Reporte proyecto 2

I Semestre

21 de mayo del 2018

Abstract: In this Project, we search for a program we can parallelize using the OpenMP API and execute at the Cluster Kabré. In the actual report, we will just show the access to the Cluster and the executions, and the article in IEEE format is attached in the same carpet.

## Pruebas de acceso al Cluster

Para el proyecto actual, utilizamos solamente la computadora de Fabricio para hacer las pruebas, pues la computadora de Alonso se utilizó para recolectar las pruebas de ejecución y buscar información.

En esta imagen se muestra cuando el programa enviado se está ejecutando en la queue del cluster:

```
hal
curso-095@login-3:~
Every 5.0s: qstat -a                                     Sat Jun  9 11:18:46 2018

meta.cnca:

Job ID      Username  Queue  Jobname      SessID  NDS   TSK   Req'd  Req'd  Elap
-----  -
31215.meta.cnca  curso-111  cpu-n5h2  satisfy      --      3     3     --    00:03:00  Q    --
31408.meta.cnca  curso-094  cpu-n5h2  pi-3n        --      3     3     --    00:05:00  Q    --
31545.meta.cnca  dag       phi-debu  STDIN        124516  --    --    --     --      R    00:00:00
31590.meta.cnca  curso-087  phi-n2h7  MontecarloPiPara 230424  1     1     --    00:45:00  C    --
31591.meta.cnca  curso-094  phi-n2h7  proyecto2     225618  1     1     --    00:15:00  C    --
31593.meta.cnca  curso-096  phi-n2h7  proyecto3     260891  1     1     --    00:15:00  R    00:09:10
31595.meta.cnca  curso-095  phi-n5h2  paralelo     233710  1     1     --    00:20:00  C    --
31596.meta.cnca  curso-095  phi-n5h2  paralelo     232037  1     1     --    00:20:00  R    00:01:46
```

En la siguiente imagen, se muestra el resultado que se obtuvo al terminar el programa de ejecutarse, el cual es el tiempo que tardó:

```
hal
curso-095@login-3:~
GNU nano 2.3.1                                     Fichero: paralelo.e31596

real    2m44.929s
user    2m44.932s
sys     0m0.006s

[ 4 líneas leídas ]
^G Ver ayuda      ^O Guardar      ^R Leer Fich    ^Y Pág Ant      ^K CortarTxt    ^C Pos actual
^X Salir          ^J Justificar   ^W Buscar       ^V Pág Sig      ^U PegarTxt     ^T Ortografia
```

En la siguiente imagen se muestran los archivos que se encontraban en el usuario de Fabricio:

```

curso-095@login-3:~
cluster      integrales      integrales.pbs  mpd.hosts      paralelo.c      paralelo.o31595  scratch
cluster.pub  integrales.c      intel          paralelo        paralelo.e31595  paralelo.pbs      torque-scripts
[curso-095@login-3 ~]$ watch -n 5 qstat -a
[curso-095@login-3 ~]$ nano paralelo.e31596
[curso-095@login-3 ~]$ qsub paralelo.pbs
31597.meta.cnca
[curso-095@login-3 ~]$ watch -n 5 qstat -a
[curso-095@login-3 ~]$ ls
cluster      integrales.c      mpd.hosts      paralelo.e31595  paralelo.o31595  paralelo.pbs
cluster.pub  integrales.pbs    paralelo        paralelo.e31596  paralelo.o31596  scratch
integrales  intel            paralelo.c      paralelo.e31597  paralelo.o31597  torque-scripts
[curso-095@login-3 ~]$ nano paralelo.e31597
[curso-095@login-3 ~]$ qsub paralelo.pbs
31600.meta.cnca
[curso-095@login-3 ~]$ qsub paralelo.pbs
31601.meta.cnca
[curso-095@login-3 ~]$ watch -n 5 qstat -a
[curso-095@login-3 ~]$ ls
cluster      integrales.pbs    paralelo.c      paralelo.e31600  paralelo.o31597  scratch
cluster.pub  intel            paralelo.e31595  paralelo.e31601  paralelo.o31600  torque-scripts
integrales  mpd.hosts        paralelo.e31596  paralelo.o31595  paralelo.o31601  paralelo.pbs
integrales.c paralelo          paralelo.e31597  paralelo.o31596  paralelo.pbs
[curso-095@login-3 ~]$ nano paralelo.e31600
[curso-095@login-3 ~]$

```

En la siguiente imagen se muestran varios procesos que se mandaron a ejecutar simultáneamente al cluster:

```

curso-095@login-3:~
Every 5.0s: qstat -a                                     Sat Jun  9 11:46:40 2018

meta.cnca:
Job ID      Username      Queue      Jobname      SessID  NDS  TSK  Req'd  Req'd  S  Elap
Memory      Time                                     Time
-----
31215.meta.cnca  curso-111    cpu-n5h2    satisfy      --      3    3    --     00:03:00  Q  --
31408.meta.cnca  curso-094    cpu-n5h2    pi-3n        --      3    3    --     00:05:00  Q  --
31545.meta.cnca  dag         phi-debu    STDIN        124516  --   --    --     --      R  00:00:00
31605.meta.cnca  curso-094    phi-n2h7    proyecto3    1628    1    1    --     00:15:00  R  00:14:12
31608.meta.cnca  curso-094    phi-n2h7    proyecto2    234232  1    1    --     00:20:00  R  00:04:06
31609.meta.cnca  curso-095    phi-n5h2    paralelo     229033  2    2    --     00:20:00  R  00:00:45
31610.meta.cnca  curso-095    phi-n5h2    paralelo     236137  2    2    --     00:20:00  R  00:00:41
31611.meta.cnca  curso-095    phi-n5h2    paralelo     233244  2    2    --     00:20:00  R  00:00:39
31612.meta.cnca  curso-095    phi-n5h2    paralelo     232650  2    2    --     00:20:00  R  00:00:36

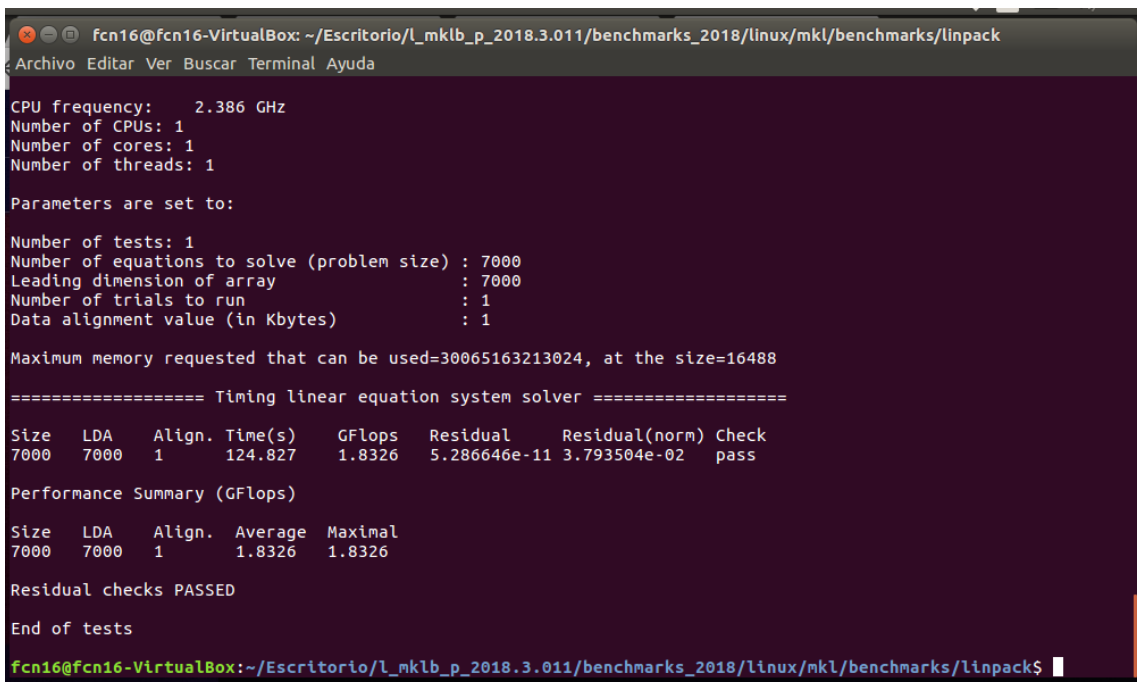
```

En esta imagen se muestra lo que resultaba al correr el mismo programa enviado al clúster, en la computadora de Fabricio, donde se puede observar que duraba mucho más que en el clúster, lo que confirma todo lo mencionado en el trabajo escrito.



```
fcn16@fcn16-VirtualBox: ~/Escritorio
fcn16@fcn16-VirtualBox:~/Escritorio$ gcc paralelo.c -o paralelo -fopenmp
fcn16@fcn16-VirtualBox:~/Escritorio$ ./paralelo
3.141593e+00
Time spent: 211.738708
fcn16@fcn16-VirtualBox:~/Escritorio$ ./paralelo
3.141593e+00
Time spent: 211.659077
fcn16@fcn16-VirtualBox:~/Escritorio$
```

En la siguiente imagen se muestra el programa que se utilizó para verificar la cantidad de GFlops que corría la computadora de Fabricio.



```
fcn16@fcn16-VirtualBox: ~/Escritorio/L_mklb_p_2018.3.011/benchmarks_2018/linux/mkl/benchmarks/linpack
Archivo Editar Ver Buscar Terminal Ayuda

CPU frequency: 2.386 GHz
Number of CPUs: 1
Number of cores: 1
Number of threads: 1

Parameters are set to:

Number of tests: 1
Number of equations to solve (problem size) : 7000
Leading dimension of array : 7000
Number of trials to run : 1
Data alignment value (in Kbytes) : 1

Maximum memory requested that can be used=30065163213024, at the size=16488

===== Timing linear equation system solver =====

Size  LDA  Align. Time(s)  GFlops  Residual  Residual(norm) Check
7000  7000   1    124.827    1.8326  5.286646e-11  3.793504e-02  pass

Performance Summary (GFlops)

Size  LDA  Align. Average  Maximal
7000  7000   1     1.8326    1.8326

Residual checks PASSED

End of tests

fcn16@fcn16-VirtualBox:~/Escritorio/L_mklb_p_2018.3.011/benchmarks_2018/linux/mkl/benchmarks/linpack$
```

En la siguiente imagen se muestra el programa que utilizó para verificar la cantidad de GFlops que corre la computadora de Alonso:

```
@ubuntu-VirtualBox: ~/Escritorio/l_mklb_p_2018.3.011/benchmarks_2018/linux/mkl/benchmarks/linpack
Number of equations to solve (problem size): 7000
Leading dimension of array: 7000
Number of trials to run: 1
Data alignment value (in Kbytes): 1
Current date/time: Mon Jun 11 22:02:22 2018

CPU frequency: 1.785 GHz
Number of CPUs: 1
Number of cores: 1
Number of threads: 1

Parameters are set to:

Number of tests: 1
Number of equations to solve (problem size) : 7000
Leading dimension of array : 7000
Number of trials to run : 1
Data alignment value (in Kbytes) : 1

Maximum memory requested that can be used=30065163213024, at the size=3086319510

===== Timing linear equation system solver =====

Size  LDA  Align. Time(s)  GFlops  Residual  Residual(norm) Check
7000  7000  1      21.662    10.5605  5.286646e-11  3.793504e-02 pass

Performance Summary (GFlops)

Size  LDA  Align. Average Maximal
7000  7000  1      10.5605  10.5605

Residual checks PASSED

End of tests

ubuntu@ubuntu-VirtualBox:~/Escritorio/l_mklb_p_2018.3.011/benchmarks_2018/linux/mkl/benchmarks/linpack$
```

## Mini marco teórico

Seguidamente se explicará un poco las herramientas utilizadas en el desarrollo del proyecto.

**OpenMP:** es una interfaz de programación de aplicaciones (API) para la programación multiproceso de memoria compartida en múltiples plataformas. Permite añadir concurrencia a los programas escritos en C, C++ y Fortran sobre la base del modelo de ejecución fork-join. Está disponible en muchas arquitecturas, incluidas las plataformas de Unix y de Microsoft Windows. Se compone de un conjunto de directivas de compilador, rutinas de biblioteca, y variables de entorno que influyen el comportamiento en tiempo de ejecución [1].

**Clúster:** se define como los conjuntos o conglomerados de ordenadores unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora [2].

**Algoritmo paralelo:** es un algoritmo que puede ser ejecutado por partes en el mismo instante de tiempo por varias unidades de procesamiento, para finalmente unir todas las partes y obtener el resultado correcto.

### **Programa ejecutado**

A continuación se adjunta el programa utilizado para la ejecución y paralelización en el clúster.

```
#include <omp.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

static long num_steps = 1000000000000;
double step;

void main()
{
    clock_t begin, end;
    double result;
    begin = clock();
    int i;
    double x, pi, sum = 0.0;
    step = 1.0/(double) num_steps;

    // hilos
    omp_set_num_threads(1);
    //declarar una region completa

    #pragma omp parallel
    {
        #pragma omp for private(x,i) reduction(+:sum)
```

```

        for(i=0; i<num_steps; i++)
        {
            #pragma omp critical
            x = (i+0.5)*step;
            sum = sum + 4.0/(1.0+x*x);
        }
    }

    pi = step*sum;
    printf("%e \n",pi);

    end = clock();
    result = (double) (end-begin)/CLOCKS_PER_SEC;
    printf("Time spent: %f \n",result);
}

```

### **Link repositorio Git**

<https://github.com/fabriciocn16/Proyecto2>

### **Referencias bibliográficas**

- [1] Wikipedia. (2017). OpenMP. Recuperado de:  
<https://es.wikipedia.org/wiki/OpenMP>
- [2] Wikipedia (2018). Clúster(Informática). Recuperado de:  
[https://es.wikipedia.org/wiki/Cl%C3%BAster\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cl%C3%BAster_(inform%C3%A1tica))
- [3] Wikipedia. (2018). Algoritmo paralelo. Recuperado de:  
[https://es.wikipedia.org/wiki/Algoritmo\\_paralelo](https://es.wikipedia.org/wiki/Algoritmo_paralelo)