Instituto Tecnológico de Costa Rica.

Sede de Cartago.

Escuela de Ingeniería en Computación.

Proyecto 2 - Cluster Kabré + Open MP.

Mary Paz Aguilar Herrera, 2017097669

Jossy Mejia Vargas 2017109693

I Semestre, 2018.

Fecha de Entrega: 11/06/2018.

Profesor: Esteban Arias.

### Abstract

In this document we will show the process to make the parallel computing with the Kabre cluster located in the Cenat, we will also describe the specs of our computers and show the screenshots about a program who show the floops of our computers.

En este trabajo se puso en práctica la programación paralela con Open MP y MPI donde se trabaja con algunos nodos del cluster Kabré con un código que a través de un for que empieza en 0 y termina en 9999999 realiza algunas operaciones de una ecuación para después imprimir el resultado, este código fue adaptado a las librerías Open MP y MPI para ejecutarlo en cluster de manera paralela.

Como documentación de apéndice del artículo en este documento se quiere probar la realización del trabajo por este motivo se demuestra con fotos los procesos realizados en el cluster para poder lograr las conclusiones del artículo científico.

#### Mini marco teórico

Cluster:El término clúster (del inglés cluster, que significa grupo o racimo) se aplica a 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.

Computación paralela: La computación paralela es una forma de cómputo en la que muchas instrucciones se ejecutan simultáneamente, operando sobre el principio de que problemas grandes, a menudo se pueden dividir en unos más pequeños, que luego son resueltos simultáneamente (en paralelo).

Cenat: El Centro Nacional de Alta Tecnología (CeNAT-CONARE) ha construido una plataforma sólida basada en laboratorios especializados en geomática, nanotecnología, computación avanzada y biotecnología que le han permitido impulsar diversos proyectos de investigación, beneficiando al sector empresarial, gubernamental y académico.

OpenMP: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.

Flops: operaciones de coma flotante por segundo son una medida del rendimiento de una computadora, especialmente en cálculos científicos que requieren un gran uso de operaciones de coma flotante.

Especificaciones técnicas de las computadoras usadas:

Computadora #1: Marca: ACER

Procesadores: Intel® Core™ I5-5200u cpu @2.20GHz(4 CPUs), ~2.20 GHz

Tipo: Notebook

RAM tipo: DDR3L SDRAM RAM cantidad: 6.00 GB RAM velocidad: 1600 MHz Tipo de disco duro: HDD Cantidad de disco duro: 1 TB Tlpos de memoria caché: 3 MB Flops de la computadora #1:

```
Input data or print help? Type [data]/help:
DATA
Number of equations to solve (problem size): 5000
Leading dimension of array: 5000
Number of trials to run: 200
Data alignment value (in Kbytes): 1000000
Warning: incorrect parameter Data alignment value (in Kbytes) (1000000),
must be not bigger than (64),
set to default value (64).
Current date/time: Mon Jun 11 19:23:58 2018

CPU frequency: 2.476 GHz
Number of CPUs: 1
Number of CPUs: 1
Number of tests: 2
Number of threads: 2

Parameters are set to:
Number of equations to solve (problem size): 5000
Leading dimension of array : 5000
Number of trials to run : 200
Data alignment value (in Kbytes) : 64

Maximum memory requested that can be used=200165536, at the size=5000
```

```
C:\Users\ACER\Desktop\w_mklb_p_2018.2.010\benchmarks_2018\windows\mkl\...
                                                                                                                                                            2.262585e-11
                           5000
5000
5000
5000
5000
5000
5000
 5000
5000
                                                                                                                                                                                                              3.154992e-
                                                                                1.759
1.934
1.839
1.923
2.208
1.911
                                                      64
64
64
                                                                                                                                                                                                                                                     -02
-02
-02
-02
-02
-02
-02
                                                                                                                                                                                                                                                                         pass
 5000
                                                                                                                                                                                                                                                                         pass
 5000
                                                                                                                                                                                                                                                                         pass
 5000
5000
                                                                                                                           43.3610
37.7682
                                                                                                                                                                                                                                                                         pass
                                                     64
64
64
64
5000
5000
                                                                                                                                                                                                                                                                         pass
                                                                                 1.869
                                                                                                                                                                                                                                                                         pass
                           5000
5000
5000
5000
5000
 5000
                                                                                                                                                                                                                                                                         pass
                                                                                                                                                                                                                                                    -02
-02
-02
-02
-02
-02
 5000
                                                                                                                                                                                                                                                                         pass
 5000
                                                                                                                                                                                                                                                                         pass
5000
5000
5000
5000
                           5000
5000
5000
                                                      64
64
64
64
                                                                                                                                                                                                                                                                         pass
                                                                                 1.890
                                                                                                                                                                                                                                                                         pass
                                                                                                                                                                                                                                                                         pass
                           5000
5000
5000
5000
5000
5000
5000
 5000
                                                                                                                                                                                                                                                                         pass
5000
5000
5000
5000
5000
                                                                                                                                                                                                                                                     -02
-02
-02
-02
-02
                                                                                                                                                                                                                                                                         pass
                                                                                                                                                                                                                                                                         pass
                                                                                                                           45.2522
43.0442
                                                                                 1.843
1.937
                                                      64
64
64
64
                                                                                                                                                                                                                                                                         pass
                                                                                 1.937
1.888
                                                                                                                                                                                                                                                                         pass
                                                                                                                            44.1583
 5000
                                                                                                                                                                                                                                                                         pass
                           5000
5000
                                                                                                                            44.1836
42.6695
                                                                                  1.887
                                                                                                                                                                                                                                                                         pass
```

Computadora #2:

Marca: HP

Procesadores: Intel® Core™ i5-6200U con gráficos Intel® HD 520 (2,3 GHz, hasta

2,8 GHz, 2 núcleos) Tipo: Notebook

RAM tipo: DDR3L SDRAM

RAM cantidad: 6 GB

RAM velocidad: 1600 MHz

Tipo de disco duro: Serial ATA-300

Cantidad de disco duro: 1TB Tlpos de memoria caché: 3MB Flops de la computadora #2:

Los programas usados para calcular los flops de cada computadora están en los siguientes links:

https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite https://www.dll-files.com/download/b942a89d465bba13d6a33c146d93b264/libiomp5 md.dll.html?c=RVIXcGJuQ0hGazJvTFB1amJTb0FOUT09

Se accesó al cluster como se muestra en la siguiente imagen:

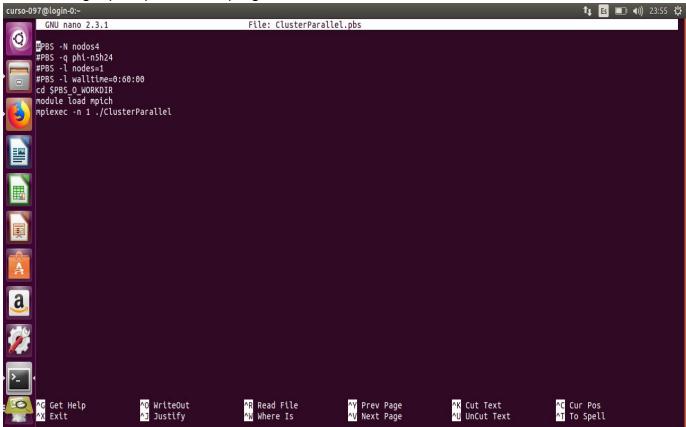
```
🚫 🖯 🗇 curso-097@login-2:~
Execution Time: 118.791
mary@mary-VirtualBox:~/Downloads$ ssh curso-097@cluster.cenat.ac.cr
curso-097@cluster.cenat.ac.cr's password:
Permission denied, please try again.
curso-097@cluster.cenat.ac.cr's password:
/opt/create_scratch.sh failed: exit code 1
Last failed login: Tue Jun 5 09:30:35 CST 2018 from meta.cnca on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Tue Jun 5 09:09:40 2018 from meta.cnca
./oyhhh-
/hhhhhhhhhh-
 /hhhhhhhhh-
   -hhhhhhhhh-
     yhhhhhhhh. '/yh-
      ohhhhhhhs/hhhhy
       /hhhho-
                  shhhhv
                    yhhhho
         :y/
       -os
                    yhhy-
     /yhhhy-
                    s/
                                                  .ys `/yhyhy+`
-Mm-NMo `mMN
                               -ohyhhs-.yys`
                              sMm. oMMyMMMd'
 .ohhhhhhhh:
                                                                     yMdMs
                                                              mMN'
                     - 5
                  `ohhy`
                                      ..:MN/Md` -MmdMm
                                                                   /My`NM:
`NN` +MN`
/hhhhhhhhhhh+
                             :MM:
                                        -MN /Md -MNMMh
 .shhhhhhhhhho/hhhhhy'
                            oMM-
    /hhhhhhhhs..yhhhhhs
                            /MM/
                                        -MN :Mm:MmdMm
                                                                   hMmssyMMh
```

Para trasladar archivos del ordenador al cluster se realizó de la siguiente forma:l

```
😰 🖨 🗊 mary@mary-VirtualBox: ~
Cluster.pbs: No such file or directory
mary@mary-VirtualBox:~$ scp ~/Downloads/ArchCluster/Cluster.pbs curso-097@cluste
r.cenat.ac.cr:~/mary/Downloads/ArchCluster/Cluster.pbs
curso-097@cluster.cenat.ac.cr's password:
scp: /home/curso-097/mary/Downloads/ArchCluster/Cluster.pbs: No such file or dir
ectory
mary@mary-VirtualBox:~$ scp ~/Downloads/ArchCluster/Cluster.pbs curso-097@cluste
r.cenat.ac.cr:~/mary/Downloads
curso-097@cluster.cenat.ac.cr's password:
scp: /home/curso-097/mary/Downloads: No such file or directory
mary@mary-VirtualBox:~$ scp ~/Downloads/ArchCluster/Cluster.pbs curso-097@cluste
r.cenat.ac.cr:
curso-097@cluster.cenat.ac.cr's password:
                                                100% 106
Cluster.pbs
                                                               0.1KB/s
                                                                         00:00
mary@mary-VirtualBox:~$ scp ~/Downloads/ArchCluster/Cluster.cpp curso-097@cluste
r.cenat.ac.cr:
curso-097@cluster.cenat.ac.cr's password:
Cluster.cpp
                                                100% 2165
                                                               2.1KB/s
                                                                         00:00
mary@mary-VirtualBox:~$ scp ~/Downloads/ArchCluster/Cluster curso-097@cluster.ce
nat.ac.cr:
curso-097@cluster.cenat.ac.cr's password:
                                                100% 8088
                                                               7.9KB/s
                                                                          00:00
Cluster
mary@mary-VirtualBox:~$ ssh curso-097@cluster.cenat.ac.cr
curso-097@cluster.cenat.ac.cr's password:
```

Se adjuntan diferentes archivos utilizados en el Cluster:

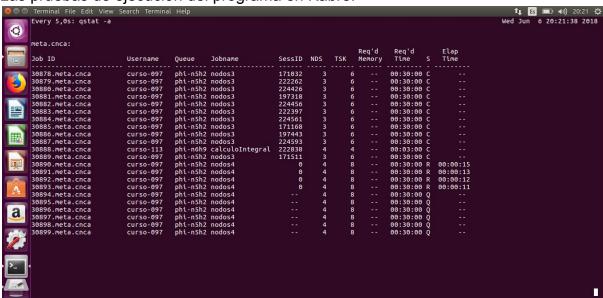
1. Código .pbs que corre el programa en el Cluster.

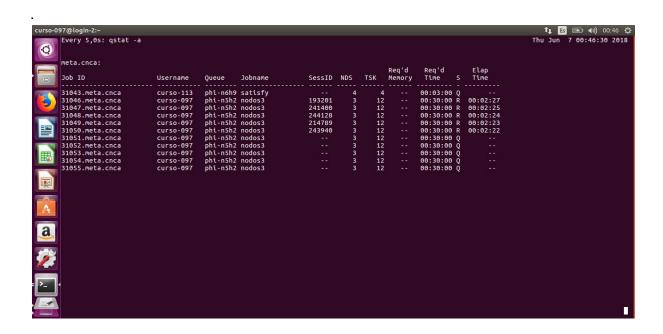


2. Un archivo .o generado por el Cluster.

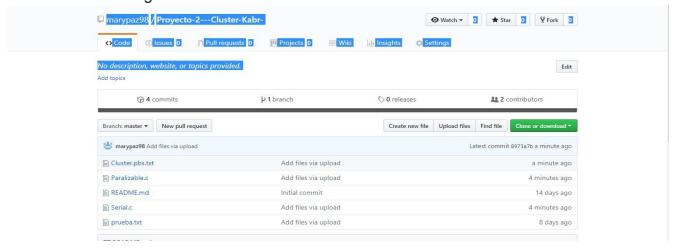


Las pruebas de ejecución del programa en Kabré:





# Pruebas de uso de github:



link de acceso a github: https://github.com/marypaz98/Proyecto-2---Cluster-Kabr-

#### **Apéndices**

# Apéndice A: Versión serial del código.

```
#include <stdio.h>
       #include <time.h>
       int main(int argc, char *argv[]){
             int i, op, upToVal;
                    upToVal = 999999999;
                    op = 0;
                    clock t start =clock();
                    for(i=0; i <= upToVal; i++) {</pre>
             op=pow(upToVal,9999)*pow(i,99999)*pow(op,8888)+pow(upToVal,9998)*pow(o
             p,99999)
                    double time = (((double) clock()-
             start) /CLOCKS_PER_SEC);
                    printf("\nOp is %d\n", op);
                         printf("Execution time: %f", time);
                    return 0;
Apéndice B: Versión paralela del código (Open MP y MPI).
#include <stdio.h>
```

```
#include "mpi.h"
#include <omp.h>
int main(int argc, char *argv[]){
int i, sum, sumTotal, upToVal;
int start, end, size, myRank;
upToVal = 999999999;
MPI_Init(&argc, &argv);
                         MPI_Comm size(MPI COMM WORLD, &size
MPI Comm rank (MPI COMM WORLD, &myRank);
start = myRank*(upToVal/size)+1;
if(myRank == (size-1)){
       end = upToVal;
}else{
      end = start + (upToVal/size) -1;}
op = 0;
opTotal = 0;
#pragma parallel for
for(i=start; i <= end; i++) {</pre>
op=pow(upToVal,9999)*pow(i,99999)*pow(op,8888)+pow(upToVal,9998)*pow(op,99999)
MPI Reduce(&op, &opTotal, 1, MPI INT, MPI SUM, 0, MPI COMM WORLD);
printf("\nRank: %d, local sum: %d, total sum: %d\n", myRank, op, opTotal);
MPI Finalize();
return 0;
```