



UNIVERSIDAD NACIONAL DE SAN AGUSTIN

CIENCIA DE LA COMPUTACIÓN

Informe de los ejercicios en el super computador de la UNSA

Alumnas :

Judith Escalante Calcina

Profesor:

Mg. Alvaro Henry Mamani

Aliaga

Índice

1. Script que se utilizará para compilar en el super computador	2
2. Ejercicios en OpenMP	2
2.1. Data Histogram	2
2.2. Monte Carlo	4
2.3. Count Sort	5
2.4. Odd Sort	6
2.5. Cola en el supercomputador	7
3. Ejercicios en Pthread	7
3.1. Mutex	7
3.2. Multiplicación de una matrix y un vector	8
3.3. Tokenize	9
3.4. Cola en el supercomputador	11
4. Ejercicios en MPI	11
4.1. Multiplicación de una matrix y un vector	11
5. Conclusión	12

1. Script que se utilizará para compilar en el super computador

```
#!/bin/bash
#PBS -N data
### Output files
#PBS -e data.err
#PBS -o data.log
### Mail to user
#PBS -M jec3430@gmail.com
### Queue name (small, medium, long, verylong) batch is default queue
#PBS -q batch
#PBS -l nodes=3:ppn=16
./data
```

- Donde solo le pasaremos el ejecutable de nuestro programa e este caso el ejecutable se llama data.
- El data.err nos mostrará si tenemos algún error.
- El data.log nos mostrará los mensajes que imprimimos en nuestro código original.

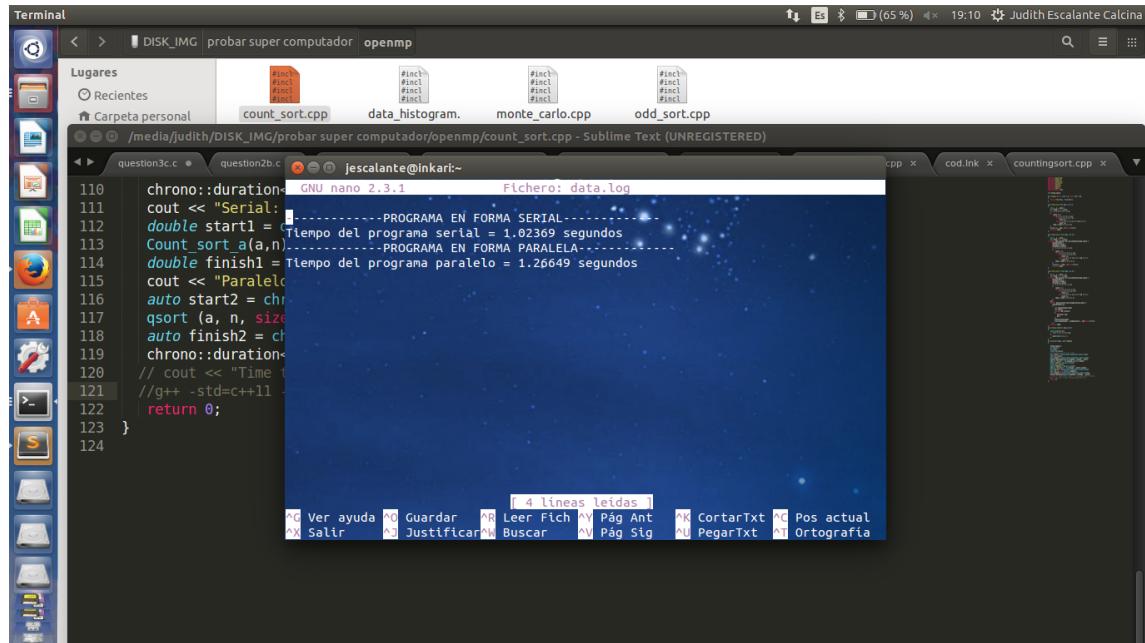
2. Ejercicios en OpenMP

A continuación mostraré la comparación de tiempos entre la compilación del CPU y el super computador ; no en todos los casos el super computador lo hace más rápido.

2.1. Data Histogram

- Super computador

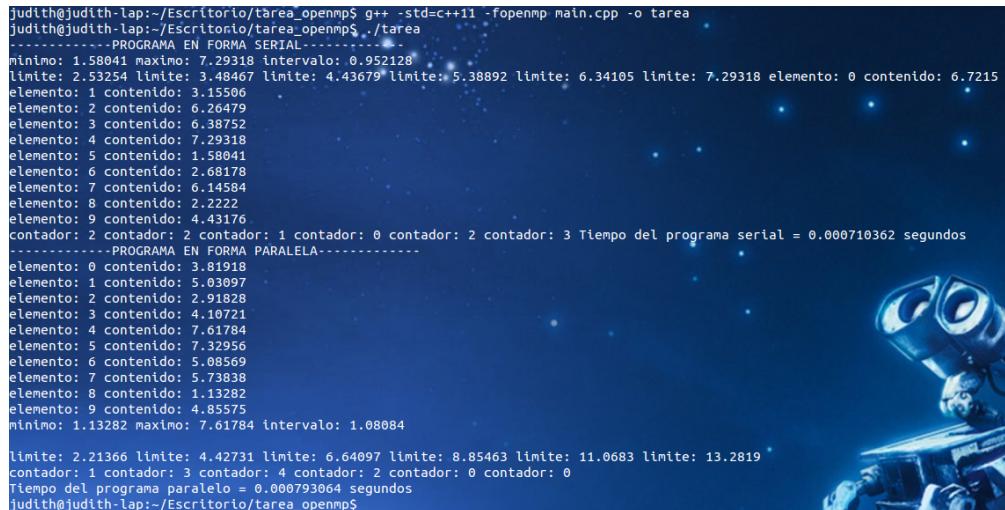
Informe de los ejercicios en el super computador de la UNSA



```
110 chrono::duration<double> start1 = chrono::duration<double>::zero(); Fichero: data.log
111 cout << "Serial: " << endl;
112 double start1 = chrono::duration<double>::zero();
113 Count_sort(a, n);
114 double finish1 = chrono::duration<double>::zero();
115 cout << "Paralelo: " << endl;
116 auto start2 = chrono::duration<double>::zero();
117 qsort(a, n, size);
118 auto finish2 = chrono::duration<double>::zero();
119 chrono::duration<double>::duration();
120 // cout << "Time " << finish1 - start1 << endl;
121 // g++ -std=c++11 -fopenmp -o main main.cpp
122 // return 0;
123 }
124
```

Figura 1: data.log

■ CPU

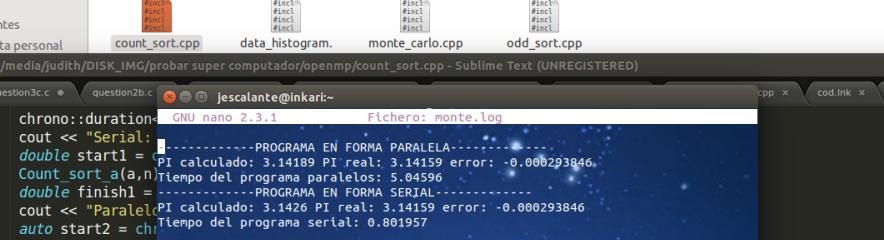


```
judith@judith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp main.cpp -o tarea
judith@judith-lap:~/Escritorio/tarea_openmp$ ./tarea
-----PROGRAMA EN FORMA SERIAL-----
minimo: 1.58041 maximo: 7.29318 intervalo: 0.952128
limite: 2.53254 limite: 3.48467 limite: 4.43679 limite: 5.38892 limite: 6.34105 limite: 7.29318 elemento: 0 contenido: 6.7215
elemento: 1 contenido: 3.15506
elemento: 2 contenido: 6.26479
elemento: 3 contenido: 6.38752
elemento: 4 contenido: 7.29318
elemento: 5 contenido: 1.58041
elemento: 6 contenido: 2.68178
elemento: 7 contenido: 6.14584
elemento: 8 contenido: 2.2222
elemento: 9 contenido: 4.43176
contador: 2 contador: 2 contador: 1 contador: 0 contador: 2 contador: 3 Tiempo del programa serial = 0.000710362 segundos
-----PROGRAMA EN FORMA PARALELA-----
elemento: 0 contenido: 3.81918
elemento: 1 contenido: 5.03097
elemento: 2 contenido: 2.91828
elemento: 3 contenido: 4.10721
elemento: 4 contenido: 7.61784
elemento: 5 contenido: 7.32956
elemento: 6 contenido: 5.08569
elemento: 7 contenido: 5.73838
elemento: 8 contenido: 1.13282
elemento: 9 contenido: 4.85575
minimo: 1.13282 maximo: 7.61784 intervalo: 1.08084
limite: 2.21366 limite: 4.42731 limite: 6.64097 limite: 8.85463 limite: 11.0683 limite: 13.2819
contador: 1 contador: 3 contador: 4 contador: 2 contador: 0 contador: 0
Tiempo del programa paralelo = 0.000793064 segundos
judith@judith-lap:~/Escritorio/tarea_openmp$
```

Figura 2: CPU

2.2. Monte Carlo

- Super computador



The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window, titled 'Terminal', displays the command 'ls' and its output, showing files like 'count_sort.cpp', 'data_histogram.cpp', 'monte_carlo.cpp', and 'odd_sort.cpp'. The code editor, titled 'jescalante@inkar:-', shows a C++ program named 'question3c.c'. The program includes code for calculating PI using parallel and serial methods, and it uses the `chrono::duration` class to measure execution time. The code editor interface includes tabs for 'question3c.c' and 'question2b.c', and a sidebar with file navigation icons. The desktop background is a dark blue space-themed image.

```
ls
count_sort.cpp  data_histogram.cpp  monte_carlo.cpp  odd_sort.cpp
question3c.c  question2b.c

jescalante@inkar:~$ g++ question3c.c -o question3c
jescalante@inkar:~$ ./question3c
----- PROGRAMA EN FORMA PARALELA -----
PI calculado: 3.14189 PI real: 3.14159 error: -0.000293846
Tiempo del programa paralelos: 5.04596

----- PROGRAMA EN FORMA SERIAL -----
PI calculado: 3.1426 PI real: 3.14159 error: -0.0010600000000000000
Tiempo del programa serial: 0.801957
```

Figura 3: monte.log

- CPU

```
judith@judith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp monte_carlo.cpp -o tarea2
judith@judith-lap:~/Escritorio/tarea_openmp$ ./tarea2
----- PROGRAMA EN FORMA PARALELA -----
PI calculado: 3.14159 PI real: 3.14159 error: -2.34641e-06
Tiempo del programa paralelos: 3.68269
----- PROGRAMA EN FORMA SERIAL -----
PI calculado: 3.1426 PI real: 3.14159 error: -2.34641e-06
Tiempo del programa serial: 0.355452
judith@judith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp monte_carlo.cpp -o tarea2
judith@judith-lap:~/Escritorio/tarea_openmp$ ./tarea2
----- PROGRAMA EN FORMA PARALELA -----
PI calculado: 3.14128 PI real: 3.14159 error: 0.000313654
Tiempo del programa paralelos: 2.39077
----- PROGRAMA EN FORMA SERIAL -----
PI calculado: 3.1426 PI real: 3.14159 error: 0.000313654
Tiempo del programa serial: 0.658256
judith@judith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp monte_carlo.cpp -o tarea2
judith@judith-lap:~/Escritorio/tarea_openmp$ ./tarea2
----- PROGRAMA EN FORMA PARALELA -----
PI calculado: 3.14195 PI real: 3.14159 error: -0.000353346
Tiempo del programa paralelos: 0.178256
----- PROGRAMA EN FORMA SERIAL -----
PI calculado: 3.1426 PI real: 3.14159 error: -0.000353346
Tiempo del programa serial: 0.367439
```

Figura 4: CPU

2.3. Count Sort

- Super computador

```

Fichero: count.log
110 chrono::duration<chrono::nanoseconds> start1 = chrono::duration<chrono::nanoseconds>::zero();
111 cout << "Serial: ";
112 double start1 = chrono::high_resolution_clock::now().time_since_epoch().count();
113 Count_sort_a(a,n);
114 double finish1 =
115 cout << "Paralelo: ";
116 auto start2 = chrono::high_resolution_clock::now();
117 qsort(a, n, size);
118 auto finish2 = chrono::high_resolution_clock::now();
119 chrono::duration<chrono::nanoseconds> start3 = chrono::duration<chrono::nanoseconds>::zero();
120 // cout << "Time taken: " << (finish1 - start1) / 1000000000.0 << endl;
121 //g++ -std=c++11 -fopenmp -o tarea3 question3c.c question2b.c
122 //return 0;
123 }
124

```

Figura 5: count.log

- CPU

```

jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp sorting_algorithm.cpp -o tarea3
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ ./tarea3
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 1.42037 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.469134 segundos
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp sorting_algorithm.cpp -o tarea3
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ ./tarea3
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 1.42718 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.469134 segundos
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp sorting_algorithm.cpp -o tarea3
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ ./tarea3
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 0.019655 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.666789 segundos
jjudith@jjudith-lap:~/Escritorio/tarea_openmp$ 

```

Figura 6: CPU

2.4. Odd Sort

- Super computador

```

#include <iostream>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <chrono>
using namespace std;
void first_odd_even()
{
    int phase,i,tmp;
    for (phase = 0; p
    {
        if (phase % 2
# pragma omp parallel for num_threads(n_th) \
            default(shared) private(i, tmp)
        for (i =
        {
            if (e
            {
                t
                e
            }
        }
    }
    # pragma omp parallel for num_threads(n_th) \
        default(shared) private(i, tmp)
    for (i = 1; i < n - 1; i += 2)
    {
        if (a[i] > a[i+1])
    }
}

```

Figura 7: Odd.log

- CPU

```

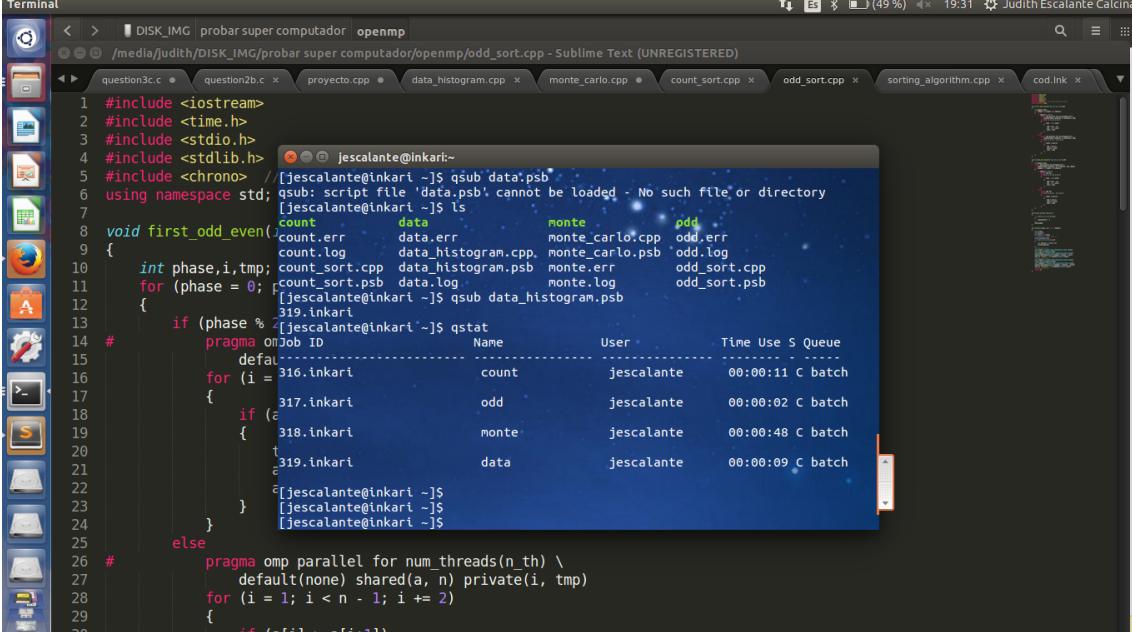
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp main.cpp -o tarea
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ ./tarea
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 0.281101 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.646444 segundos
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp main.cpp -o tarea
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ ./tarea
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 0.284854 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.622777 segundos
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp main.cpp -o tarea
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ ./tarea
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 0.842255 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 1.82956 segundos
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ g++ -std=c++11 -fopenmp main.cpp -o tarea
judgeth@judgeth-lap:~/Escritorio/tarea_openmp$ ./tarea
-----PROGRAMA EN FORMA SERIAL-----
Tiempo del programa serial = 0.829268 segundos
-----PROGRAMA EN FORMA PARALELA-----
Tiempo del programa paralelo = 0.129868 segundos

```



Figura 8: CPU

2.5. Cola en el supercomputador



```
Terminal
Disk_IMG | probar super computador | openmp
/media/Judith/DISK_IMG/probar super computador/openmp/odd_sort.cpp - Sublime Text (UNREGISTERED)

question3.c  question2b.c  proyecto.cpp  data_histogram.cpp  monte_carlo.cpp  count_sort.cpp  odd_sort.cpp  sorting_algorithm.cpp  cod.ink

1 #include <iostream>
2 #include <time.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <chrono> / [jescalante@inkari ~]$ qsub data.psb
6 using namespace std; qsub: script file 'data.psb' cannot be loaded - No such file or directory
7 [jescalante@inkari ~]$ ls
8 void first_odd_even( count  data  monte  odd
9 {  count_err  data_err  monte_carlo.cpp  odd_err
10   int phase,i,tmp;  count.log  data_histogram.cpp  monte_carlo.psb  odd.log
11   for (phase = 0;  count_sort.cpp  data_histogram.psb  monte_err  odd_sort.cpp
12   {  count_sort.psb  data.log  monte.log  odd_sort.psb
13     if (phase % 319.inkari
14   #   pragma onJob ID  Name  User  Time Use S Queue
15   default----- 316.inkari  count  jescalante  00:00:11 C batch
16   for (i = 317.inkari
17   {  317.inkari  odd  jescalante  00:00:02 C batch
18     if (c  318.inkari  monte  jescalante  00:00:48 C batch
19     {  t  319.inkari  data  jescalante  00:00:09 C batch
20     }
21   }
22   }
23   }
24   }
25   else
26   #   pragma omp parallel for num_threads(n_th) \
27   default(None) shared(a, n) private(i, tmp)
28   for (i = 1; i < n - 1; i += 2)
29   {
30     if (a[i] > a[i+1])
```

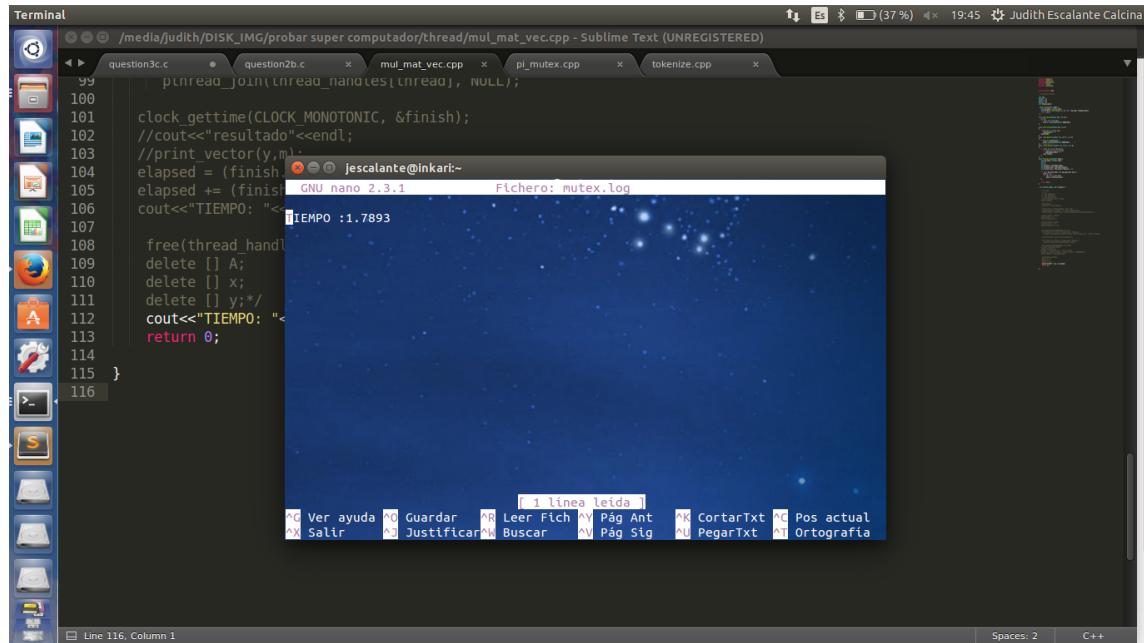
Figura 9: qstat del super computador

3. Ejercicios en Pthread

3.1. Mutex

- Super computador

Informe de los ejercicios en el super computador de la UNSA



```
Terminal /media/Judith/IMG/probar super computador/thread/mul_mat_vec.cpp - Sublime Text (UNREGISTERED)
question3c.c question2b.c mul_mat_vec.cpp pi_mutex.cpp tokenize.cpp
99     pjoin(pthread_join, &thread_handles[&thread], NULL);
100
101    clock_gettime(CLOCK_MONOTONIC, &finish);
102    //cout<<"resultado"<<endl;
103    //print vector(y,m);
104    elapsed = (finish - start);
105    elapsed += (finish - start);
106    cout<<"TIEMPO: "<< elapsed;
107
108    free(thread_handles);
109    delete [] A;
110    delete [] x;
111    delete [] y;/*
112    cout<<"TIEMPO: "<< elapsed;
113    return 0;
114
115 }
116

GNU-nano 2.3.1  Fichero: mutex.log
[1 linea leida]
^A Ver ayuda ^D Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^H Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

Line 116, Column 1  Spaces: 2  C++
```

Figura 10: mutex.log

- CPU

```
judith@judith-lap:~/Escritorio$ g++ pi_mutex.cpp -o mutex -lpthread
judith@judith-lap:~/Escritorio$ ./mutex
TIEMPO :3.85443
judith@judith-lap:~/Escritorio$
```

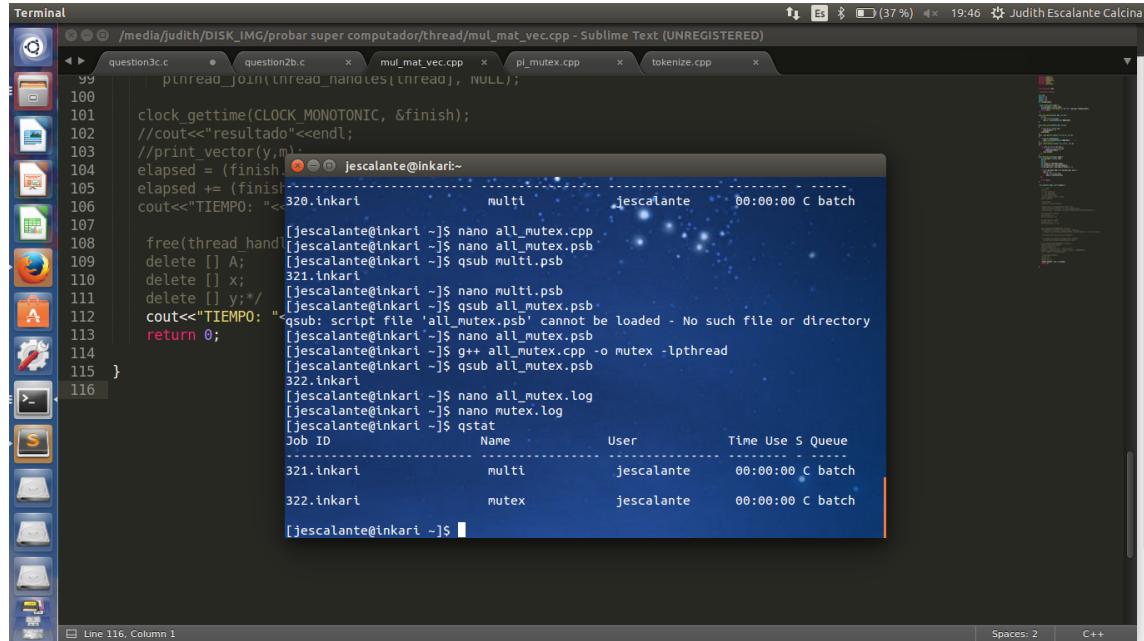


Figura 11: CPU

3.2. Multiplicación de una matrix y un vector

- Super computador

Informe de los ejercicios en el super computador de la UNSA



```
Terminal /media/Judith/DISK_IMG/probar super computador/thread/mul_mat_vec.cpp - Sublime Text (UNREGISTERED)
question3c.c question2b.c mul_mat_vec.cpp pl_mutex.cpp tokenize.cpp
99     pInread_join(pthread_handles[lnread], NULL);
100
101    clock_gettime(CLOCK_MONOTONIC, &finish);
102    //cout<<"resultado"endl;
103    //print vector(y,m);
104    elapsed = (finish - start);
105    elapsed += (finish - start);
106    cout<<"TIEMPO: "<< elapsed;
107
108    free(thread_handles);
109    delete [] A;
110    delete [] x;
111    delete [] y;
112    cout<<"TIEMPO: "<< elapsed;
113    return 0;
114
115 }
```

```
[jescalante@inkari ~]$ nano all_mutex.cpp
[jescalante@inkari ~]$ nano all_mutex.psb
[jescalante@inkari ~]$ qsub multi.psb
321.inkari
[jescalante@inkari ~]$ nano multi.psb
[jescalante@inkari ~]$ qsub all_mutex.psb
322.inkari
[jescalante@inkari ~]$ qsub all_mutex.psb
321.inkari
[jescalante@inkari ~]$ nano all_mutex.log
[jescalante@inkari ~]$ nano mutex.log
[jescalante@inkari ~]$ qstat
-----+-----+-----+-----+-----+-----+
Job ID          Name          User          Time Use S Queue
-----+-----+-----+-----+-----+-----+
321.inkari      multi        jescalante  00:00:00 C batch
322.inkari      mutex        jescalante  00:00:00 C batch
-----+-----+-----+-----+-----+-----+
[jescalante@inkari ~]$
```

Figura 12: mult.log

- CPU

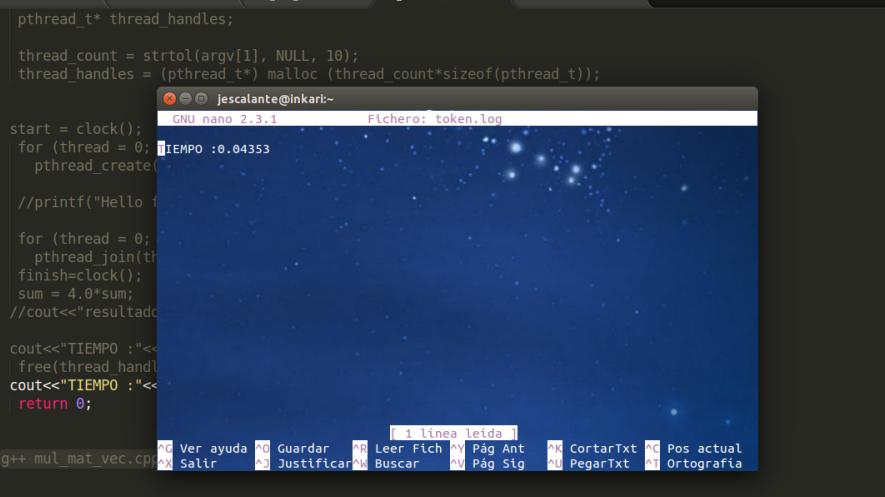
```
judith@judith-lap:~/Escritorio$ g++ mul_mat_vec.cpp -o mult -lpthread
judith@judith-lap:~/Escritorio$ ./mult
TIEMPO: 1.6573
judith@judith-lap:~/Escritorio$
```

Figura 13: CPU

3.3. Tokenize

- Super computador

Informe de los ejercicios en el super computador de la UNSA



```
Terminal /media/judith/DISK_IMG/probar super computador/thread/pi_mutex.cpp - Sublime Text (UNREGISTERED)
question3c.c question2b.c mul_mat_vec.cpp pi_mutex.cpp tokenize.cpp

97 pthread_t* thread_handles;
98
99 thread_count = strtol(argv[1], NULL, 10);
100 thread_handles = (pthread_t*) malloc (thread_count*sizeof(pthread_t));
101
102 start = clock();
103 for (thread = 0; thread < thread_count; thread++)
104     pthread_create(&thread_handles[thread], NULL, thread_routine, (void*) thread);
105
106 //printf("Hello from thread %d\n", thread);
107
108 for (thread = 0; thread < thread_count; thread++)
109     pthread_join(thread_handles[thread], NULL);
110 finish=clock();
111 sum = 4.0*sum;
112 //cout<<"resultado: "<<sum<<endl;
113
114 cout<<"TIEMPO :"<<double(clock() - start)/CLOCKS_PER_SEC<<endl;
115 free(thread_handles);
116 cout<<"TIEMPO :"<<double(clock() - finish)/CLOCKS_PER_SEC<<endl;
117
118 return 0;
119
120 }
121 //g++ mul_mat_vec.cpp -lpthread
122
```

GNU nano 2.3.1 Fichero: token.log

TIEMPO :0.04353

Ver ayuda Guardar Leer Fich Pág Ant CortarTxt Pos actual
Salir Justificar Buscar Pág Sig PegarTxt Ortografía

1 linea leida

37 characters selected Spaces: 3 C++

Figura 14: mult.log

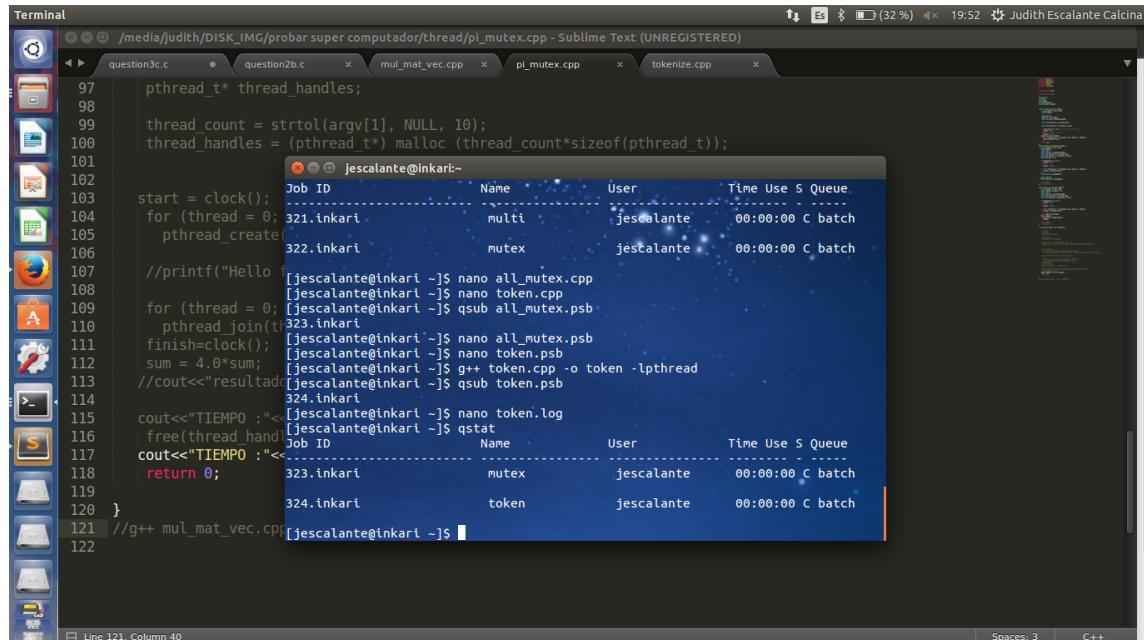
■ CPU

```
judith@judith-lap:~/Escritorio$ g++ tokenize.cpp -o token -lpthread
judith@judith-lap:~/Escritorio$ ./token
TIEMPO :1.7898
judith@judith-lap:~/Escritorio$
```



Figura 15: CPU

3.4. Cola en el supercomputador



```
Terminal /media/judith/DISK_IMG/probar super computador/thread/pi_mutex.cpp - Sublime Text (UNREGISTERED)
question3c.c question2b.c mut_mat_vec.cpp pi_mutex.cpp tokenize.cpp Judith Escalante Calcina
97     pthread_t* thread_handles;
98
99     thread_count = strtol(argv[1], NULL, 10);
100    thread_handles = (pthread_t*) malloc (thread_count*sizeof(pthread_t));
101
102    start = clock();
103    for (thread = 0; thread < thread_count; thread++)
104        pthread_create(&thread_handles[thread], NULL, jescalante@inkari-
105                      321.inkari           multi      jescalante  00:00:00 C batch
106                      322.inkari           mutex      jescalante  00:00:00 C batch
107 //printf("Hello f
108
109 for (thread = 0; thread < thread_count; thread++)
110     pthread_join(&thread_handles[thread], NULL);
111
112     finish=clock();
113     sum = 4.0*sum;
114 //cout<<"resultado
115 cout<<"TIEMPO : "<<[jescalante@inkari ~]$ nano token.log
116 free(thread_handles);
117 cout<<"TIEMPO :"<<[jescalante@inkari ~]$ qstat
118
119 return 0;
120
121 //g++ mul_mat_vec.cpp[jescalante@inkari ~]$
```

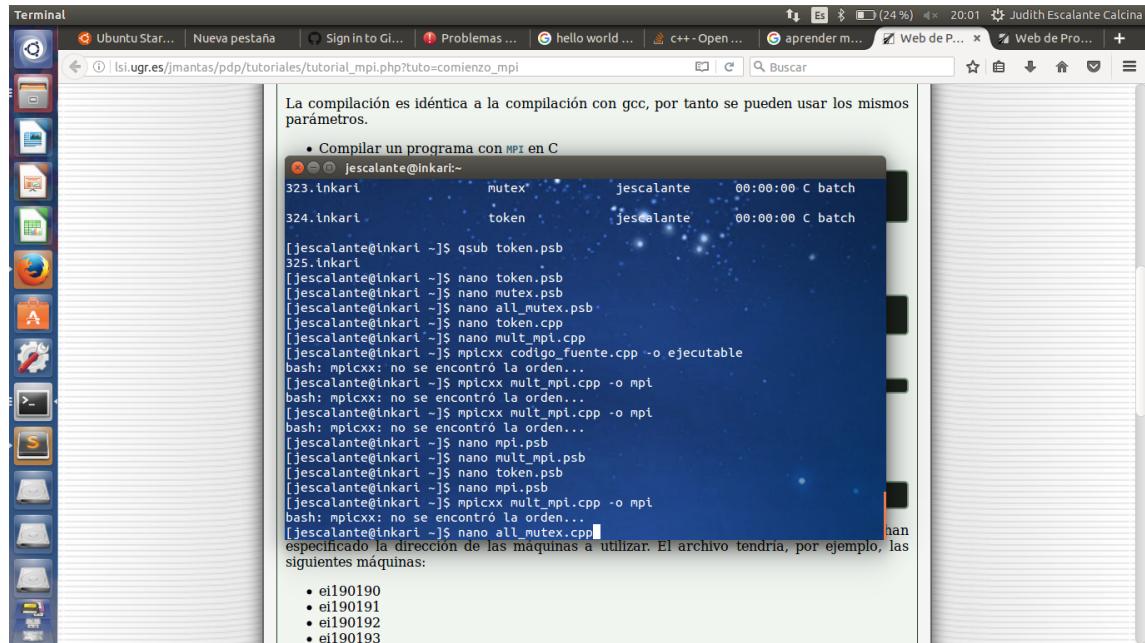
Figura 16: qstat en el super computador

4. Ejercicios en MPI

Con los programas parallelizados con MPI existe un error el cual podemos ver en la siguiente imagen , en este nos dice que no existe la librería instalada en el mismo super computador por ende no puede compilar ni generar .log que es el archivo que necesitamos.

4.1. Multiplicación de una matrix y un vector

- Super computador



The screenshot shows a terminal window on an Ubuntu desktop environment. The terminal is running a script to compile MPI programs. The log output is as follows:

```
La compilación es idéntica a la compilación con gcc, por tanto se pueden usar los mismos parámetros.

• Compilar un programa con MPI en C
[jescalante@inkari ~]$ gcc -fopenmp -o mutex mutex.c
[jescalante@inkari ~]$ gcc -fopenmp -o token token.c
[jescalante@inkari ~]$ qsub token.pbs
323.inkari          mutex          jescalante 00:00:00 C batch
324.inkari          token          jescalante 00:00:00 C batch

[jescalante@inkari ~]$ qsub token.pbs
325.inkari
[jescalante@inkari ~]$ nano token.pbs
[jescalante@inkari ~]$ nano mutex.pbs
[jescalante@inkari ~]$ nano all_mutex.pbs
[jescalante@inkari ~]$ nano token.cpp
[jescalante@inkari ~]$ nano mult_mpi.cpp
[jescalante@inkari ~]$ mpicxx codigo_fuente.cpp -o ejecutable
bash: mpicxx: no se encontró la orden...
[jescalante@inkari ~]$ mpicxx mult_mpi.cpp -o mpi
bash: mpicxx: no se encontró la orden...
[jescalante@inkari ~]$ mpicxx mult_mpi.cpp -o mpi
bash: mpicxx: no se encontró la orden...
[jescalante@inkari ~]$ nano mpi.pbs
[jescalante@inkari ~]$ nano token.pbs
[jescalante@inkari ~]$ nano mpi.pbs
[jescalante@inkari ~]$ mpicxx mult_mpi.cpp -o mpi
bash: mpicxx: no se encontró la orden...
[jescalante@inkari ~]$ nano all_mutex.pbs
ha especificado la dirección de las máquinas a utilizar. El archivo tendría, por ejemplo, las siguientes máquinas:
• e1190190
• e1190191
• e1190192
• e1190193
```

Figura 17: mult.log

5. Conclusión

Solo en algunos ejercicios de openmp el super computador no supera en tiempo de ejecución a la CPU normal , pero en general al tener más nodos en los que compilar , el super computador es muy útil en la compilación de estructuras de gran tamaño como matrices o vectores; esto se ve claramente en la multiplicación de una matrix y un vector que se realiza en los ejercicios de *Pthread*.