



**Escuela de Ingeniería en computación**

**Sede central**

**Arquitectura de computadores  
IC-3101**

**Esteban Arias Méndez  
I semestre 2018**

**Fecha de entrega: Sábado 11 de Junio**

**Reporte Proyecto 2**

**Tecnologías de procesadores  
Wilson Lopez Rubi  
Oscar vega Morua**

**Abstract**

**The main objective of this project is to run a parallel program in a cluster Kabre with different configurations of nodes, to know the structure of a super computer through a simple example programmed with openmpi.**

**Introducción:**

El presente reporte es un apéndice donde se adjuntan algunas pruebas del uso del cluster kabre del CeNat durante la realización del segundo proyecto del curso de arquitectura de computadoras , importado en el instituto tecnológico de Costa Rica , durante el primer semestre del 2018

## Código del programa:

```
#include<omp.h>
#include<iostream>
#include<sys/time.h>
#include<stdio.h>

using namespace std;
void suma(){//empesamos con el metodo suma
int m = 500;//dimension de las matrices
struct timeval start, end;//declaramos esta variable para controlar el
tiempo de ejecucion
long mtime, seconds, usecons;//variable de long para definir rangos de
segundos y mili segundos
gettimeofday(&start,NULL);

//omp_set_num_threads(32);//colocamos este comando para el numero de
hilos

int A[m][m],B[m][m],C[m][m];//declaramos las matrices con una
dimencion de m
//en los sigues lineas de codigo llenamos las matriz A con la suma de
j+i
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        A[i][j]=j+i;
    }
}
//en las siguientes lineas de codigo llenamos las matriz B con la
multiplicacion j*i
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        B[i][j]=j*i;
    }
}
#pragma omp parallel for //este comando se ejecuta la programacion
paralela en el proceso de la suma de la matriz A +B y almacenamos en C
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        C[i][j]=A[i][j]+B[i][j];
    }
}
```

```

    }
}

cout<<endl;
cout<<"-----"<<endl;
cout<<"suma de matrices "<<endl;
cout<<"-----"<<endl;
//en las siguientes lineas de codigo estamos imprimiendo la matriz C
//Se limita a imprimir solamente 10 elementos ya que 200 son muchos
for( int i=0;i<10;i++){
    cout<<"      "<<endl;
    for(int j=0;j<10;j++){

        cout<<"      "<<C[i][j];

    }
}
cout<<endl;
gettimeofday(&end, NULL);
seconds=end.tv_sec -start.tv_sec;//almacenamos el tiempo de ejecucion
usecons=end.tv_usec - start.tv_usec;//almacenamos el tiempo de
ejecucion
mtime=((seconds)*2000+usecons/2000.0)+0.5;//hacemos el calcula del
tiempo de ejecucion para que nos imprima en milisegundos
cout<<"el tiempo duro "<<mtime<<" mili segundo de la suma de
matrices";//imprimimos el tiempo de ejecucion

}

void resta(){
int m = 500;//dimension de las matrices
struct timeval start, end;//declaramos esta variable para controlar el
tiempo de ejecucion
long mtime, seconds, usecons;//variable de long para definir rangos de
segundos y mili segundos
gettimeofday(&start,NULL);

//omp_set_num_threads(32);//colocamos este comando para el numero de
hilos para la programacion paralela

```

```

int  A[m][m],B[m][m],C[m][m];//declaramos las matrices con una
dimension de m

//en los sigues lineas de codigo llenamos las matriz A con la suma de
j+i
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        A[i][j]=j+i;
    }
}

//en las siguientes lineas de codigo llenamos las matriz B con la
multiplicacion j*i
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        B[i][j]=j*i;
    }
}

#pragma omp parallel for //este comando se ejecuta la programacion
paralela en el proceso de la suma de la matriz A +B y almacenamos en C
for( int i=0;i<m;i++){
    for(int j=0;j<m;j++){
        C[i][j]=A[i][j]+B[i][j];
    }
}

cout<<endl;
cout<<"-----"<<endl;
cout<<"resta de matrices "<<endl;
cout<<"-----"<<endl;

//en las siguientes lineas de codigo estamos imprimiendo la matriz C
//Se limita a imprimir solamente 10 elementos ya que 200 son muchos
for( int i=0;i<10;i++){
    cout<<"      "<<endl;
    for(int j=0;j<10;j++){

        cout<<"      "<<C[i][j];

    }
}

cout<<endl;
gettimeofday(&end, NULL);
seconds=end.tv_sec -start.tv_sec;//almacenamos el tiempo de ejecucion

```

```

usecons=end.tv_usec - start.tv_usec;//almacenamos el tiempo de
ejecucion
mtime=((seconds)*2000+usecons/2000.0)+0.5;//hacemos el calcula del
tiempo de ejecucion para que nos imprima en milisegundos
cout<<"el tiempo duro "<<mtime<<" mili segundo de la resta de
matrices";//imprimimos el tiempo de ejecucion
cout<<"\n";

}

void multi(){
int m = 500,i,j,k;//declaramos variables
struct timeval start, end;//declaramos esta variable para controlar el
tiempo de ejecucion
long mtime, seconds, usecons;//variable de long para definir rangos de
segundos y mili segundos
gettimeofday(&start,NULL);

omp_set_num_threads(1);//colocamos este comando para el numero de hilos
para la programacion paralela

int A[m][m],B[m][m],C[m][m];//declaracion de las matrices
//con los siguientes codigos empesamos a llenar las matriz A con la i+j
for (i=0;i<m;i++)
{
    for (j=0;j<m;j++)
    {
        A[i][j]=i+j;
    }
}
//con los siguientes codigos empesamos a llenar las matriz B con la i*j
for (i=0;i<m;i++)
{
    for (j=0;j<m;j++)
    {
        B[i][j]=i*j;
    }
}

```

```
    }  
}
```

//este comando se ejecuta la programacion paralela en el proceso de la multiplicacion de la matriz A\*B y almacenamos en C

```
#pragma omp parallel for  
for (i=0;i<m;i++)  
{  
    for (j=0;j<m;j++)  
    {  
        C[i][j]=0;  
        for (k=0;k<m;k++)  
        {  
            C[i][j]=C[i][j]+A[i][k]*B[k][j];  
        }  
    }  
}
```

```
cout<<"-----"<<endl;  
cout<<"multiplicaion de matrices"<<endl;  
cout<<"-----"<<endl;  
//en las siguientes lineas de codigo estamos imprimiendo la matriz C  
//Se limita a imprimir solamente 10 elementos ya que 200 son muchos  
for (i=0;i<10;i++)  
{  
    cout<<"      "<<endl;  
    for (j=0;j<10;j++)  
    {  
        cout<<"      "<<C[i][j];  
    }  
}  
cout<<"  "<<endl;  
gettimeofday(&end, NULL);  
seconds=end.tv_sec -start.tv_sec;//almacenamos el tiempo de ejecucion  
usecons=end.tv_usec - start.tv_usec;//almacenamos el tiempo de  
ejecucion  
mtime=((seconds)*2000+usecons/2000.0)+0.5;//hacemos el calcula del  
tiempo de ejecucion para que nos imprima en milisegundos
```

```

cout<<"el tiempo duro "<<mtime<<" mili segundo de la multiplicación de
matrices";//imprimimos el tiempo de ejecución
cout<<"\n";
}

```

```

int main(){
    cout<<"-----"<<endl;
    cout<<"Operaciones matriciales"<<endl;
    cout<<"-----"<<endl;
    suma();//en la primera opcion tenemos a la suma y por tanto
instanciamos al metodo suma
    resta();//en la segunda opcion tenemos a la resta y por tanto
instanciamos al metodo resta
    multi();

return 0;
}

```

## Código del Makefile

```

default: all

all:
    g++ -Wall -o Run_file -fopenmp proyectoArqui.cpp

clean:
    rm Run_file

```



# Muestras de ejecución

- Conexión:

```
Actividades Terminal lun 08/24 curso-106@login-2~
Wilson@wilson-Inspiron-5559:~$ ssh curso-106@cluster.cenat.ac.cr
No se ha encontrado la orden «ssh», quizás quiso decir:
La orden «ssh» del paquete «dash» (main)
La orden «sch» del paquete «scheme2» (universe)
La orden «sic» del paquete «sic» (universe)
La orden «ssh» del paquete «openssh-client» (main)
ssh: no se encontró la orden
Wilson@wilson-Inspiron-5559:~$ ssh curso-106@cluster.cenat.ac.cr
/opt/create_scratch.sh failed: exit code 1
Last login: Fri Jun 8 22:53:29 2018 from meta.cncs

./oyhhh-
/hhhhhhhh-
-hhhhhhhhh-
'yh-hhhhhhh- /yh-
-ohhhhhhhhh/ hhhhy
/ hhhho- shhhhy
'yy/ 'yh-hho
'yyh-
'yyhhy- 's/ -ohyhhs..yys' .ys 'yhyhy' 'yyy'
ohhhhhhh- sm- ohyoyoyd- -m- mmo 'mm' ymms
/ hhhhhhhhh- 'ohhy- hds- ..mmd/ -mmdm- ... /my m-
-shhhhhhhhh/ hhhhy- om- -m- /m/ -mmdm- 'm' +m-
/ hhhhhhhhh- 'yh-hhs- /m/ -m- mmdm- mmsm-
'ohhy- ohhs- hds- -m- mmdm- +m- -m/
-yh- +ho +mmdm- m- -m- .ymhdym- sm-

Cluster CNCs

Para verificar el estado de los nodos con el comando "check_nodes.py"
Cualquier consulta dirigirla a cnc@cenat.ac.cr

Bienvenido
[curso-106@login-2 ~]$ ls
lsort          Proyecto      result.o31619  torque-scripts
intel          mpd.hosts    proyectoArqui.cpp  Run_file
Makefile       Progra2.pbs  result.e31619  scratch
[curso-106@login-2 ~]$
```

- 2 Nodos:

```
Actividades Visual Studio Code lun 20/20 curso-106@login-2~
[curso-106@login-2 ~]$ qsub Progra2.pbs
32976.meta.cncs
[curso-106@login-2 ~]$ cat result.o3297
result.o32970 result.o32971 result.o32973 result.o32974 result.o32975 result.o32976
[curso-106@login-2 ~]$ cat result.o32976

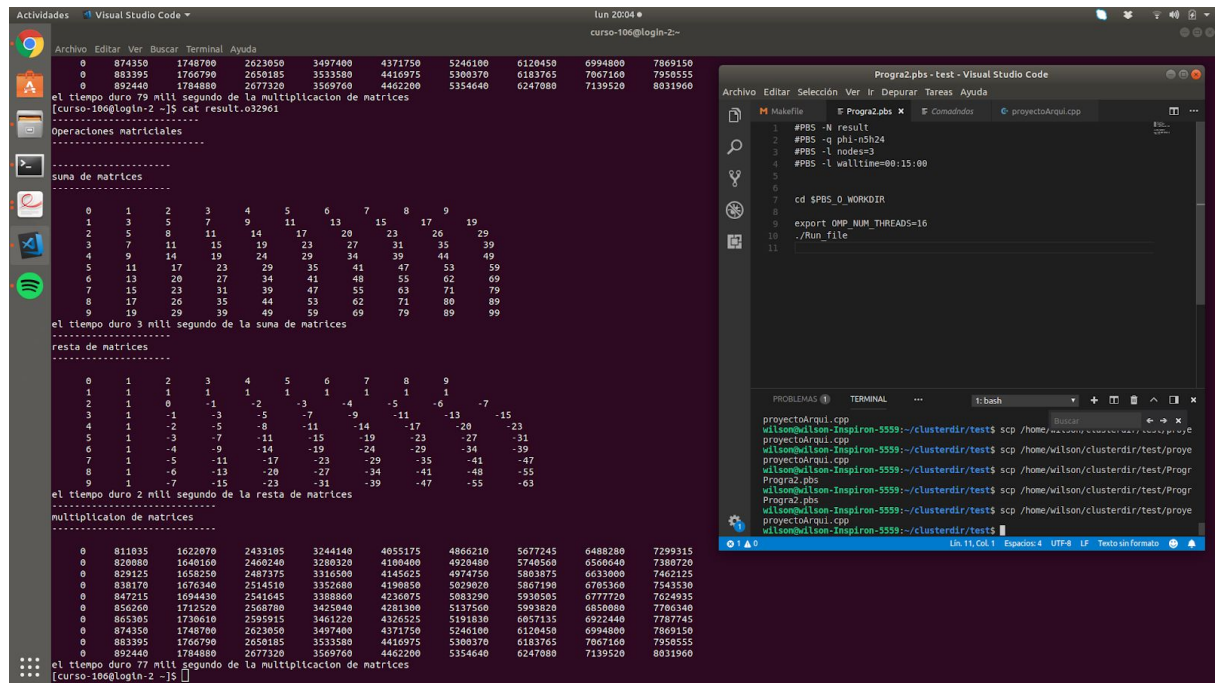
Operaciones matriciales
-----
suma de matrices
-----
 0  1  2  3  4  5  6  7  8  9
 1  3  5  7  9 11 13 15 17 19
 2  5  8 11 14 17 20 23 26 29
 3  7 11 15 19 23 27 31 35 39
 4  9 14 19 24 29 34 39 44 49
 5 11 17 23 29 35 41 47 53 59
 6 13 20 27 34 41 48 55 62 69
 7 15 23 31 39 47 55 63 71 79
 8 17 26 35 44 53 62 71 80 89
 9 19 29 39 49 59 69 79 89 99

el tiempo duro 4 mil segundo de la suma de matrices
-----
resta de matrices
-----
 0  1  2  3  4  5  6  7  8  9
 1  1  1  1  1  1  1  1  1  1
 2  1  0 -1 -2 -3 -4 -5 -6 -7
 3  1 -1 -3 -5 -7 -9 -11 -13 -15
 4  1 -2 -5 -8 -11 -14 -17 -20 -23
 5  1 -3 -7 -11 -15 -19 -23 -27 -31
 6  1 -4 -9 -14 -19 -24 -29 -34 -39
 7  1 -5 -11 -17 -23 -29 -35 -41 -47
 8  1 -6 -13 -20 -27 -34 -41 -48 -55
 9  1 -7 -15 -23 -31 -39 -47 -55 -63

el tiempo duro 3 mil segundo de la resta de matrices
-----
multiplicacion de matrices
-----
 0  904890 1809780 2714670 3619560 4524450 5429340 6334230 7239120 8144010
 0  914620 1829240 2743860 3658480 4573100 5487720 6402340 7316960 8231580
 0  924350 1848700 2773850 3697400 4621750 5546100 6470450 7394800 8319150
 0  934080 1868160 2802240 3736320 4670400 5604480 6538560 7472640 8406720
 0  943810 1887620 2831430 3775240 4719050 5662860 6606670 7550480 8494290
 0  953540 1907080 2860620 3814160 4767700 5721240 6674780 7628320 8581860
 0  963270 1926540 2889810 3853080 4816350 5779620 6742890 7706160 8669430
 0  973000 1946000 2919000 3892000 4865000 5838000 6811000 7784000 8757000
 0  982730 1965460 2948100 3930920 4913650 5896380 6879110 7861840 8844570
 0  992460 1984920 2977380 3969840 4962300 5954700 6947220 7939680 8932140

el tiempo duro 86 mil segundo de la multiplicacion de matrices
[curso-106@login-2 ~]$
```

### ● 3 Nodos :



```

Actividades Visual Studio Code lun 20:04
curso-106@login-2-~

0 874350 1746700 2623050 3497400 4371750 5246100 6120450 6994800 7869150
0 881395 1766790 2650185 3533580 4416975 5300370 6183765 7067160 7950555
0 892440 1786880 2677320 3569760 4462200 5354640 6247080 7139520 8031960

el tiempo duro 79 mil segundo de la multiplicacion de matrices
[curso-106@login-2 ~]$ cat result.o32931

-----
Operaciones matriciales
-----

suma de matrices

0 1 2 3 4 5 6 7 8 9
1 3 5 7 9 11 13 15 17 19
2 5 8 11 14 17 20 23 26 29
3 7 11 15 19 23 27 31 35 39
4 9 14 19 24 29 34 39 44 49
5 11 17 23 29 35 41 47 53 59
6 13 20 27 34 41 48 55 62 69
7 15 23 31 39 47 55 63 71 79
8 17 26 35 44 53 62 71 80 89
9 19 29 39 49 59 69 79 89 99

el tiempo duro 3 mil segundo de la suma de matrices

-----
resta de matrices
-----

0 1 2 3 4 5 6 7 8 9
1 1 1 1 1 1 1 1 1 1
2 1 0 -1 -2 -3 -4 -5 -6 -7
3 1 -1 -3 -5 -7 -9 -11 -13 -15
4 1 -2 -5 -8 -11 -14 -17 -20 -23
5 1 -3 -7 -11 -15 -19 -23 -27 -31
6 1 -4 -9 -14 -19 -24 -29 -34 -39
7 1 -5 -11 -17 -23 -29 -35 -41 -47
8 1 -6 -13 -20 -27 -34 -41 -48 -55
9 1 -7 -15 -23 -31 -39 -47 -55 -63

el tiempo duro 2 mil segundo de la resta de matrices

-----
multiplicacion de matrices
-----

0 811835 1622070 2433105 3244140 4055175 4866210 5677245 6488280 7299315
0 828880 1646100 2460240 3280320 4090400 4900480 5710560 6520640 7330720
0 828125 1649250 2463750 3316500 4145625 4974750 5803875 6633000 7462125
0 838170 1676340 2514510 3352680 4190850 5029020 5867190 6705360 7543530
0 847215 1694430 2541645 3388860 4236075 5073240 5910405 6747570 7584735
0 856260 1712520 2568780 3425940 4283100 5120265 5957430 6794595 7631760
0 865305 1730610 2595915 3461720 4325525 5162690 5999855 6837020 7674185
0 874350 1748700 2623050 3497400 4371750 5246100 6120450 6994800 7869150
0 883395 1766790 2650185 3533580 4416975 5300370 6183765 7067160 7950555
0 892440 1786880 2677320 3569760 4462200 5354640 6247080 7139520 8031960

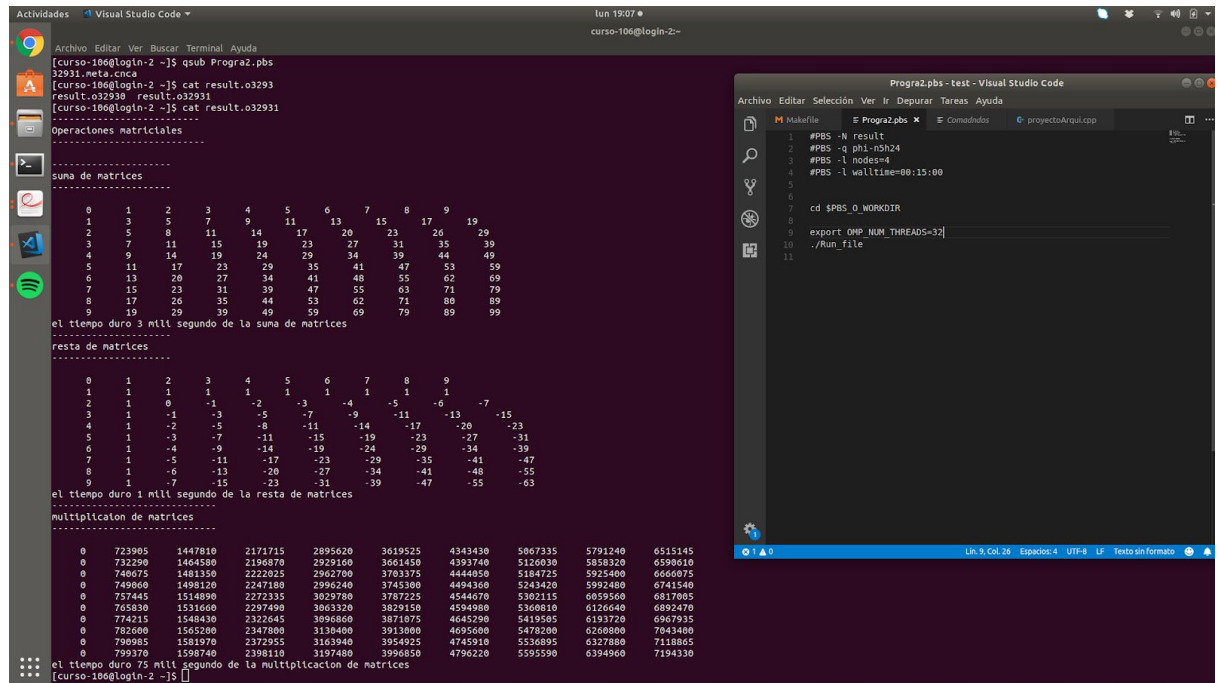
el tiempo duro 77 mil segundo de la multiplicacion de matrices
[curso-106@login-2 ~]$

Progra2.pbs - test - Visual Studio Code
Archivo Editar Selección Ver Ir Depurar Tareas Ayuda

1 #PBS -N result
2 #PBS -q phi-nsh24
3 #PBS -l nodes=3
4 #PBS -l walltime=00:15:00
5
6
7 cd $PBS_0_WORKDIR
8
9 export OMP_NUM_THREADS=16
10 ./Run_file
11

```

### ● 4 Nodos:



```

Actividades Visual Studio Code lun 19:07
curso-106@login-2-~

[curso-106@login-2 ~]$ qsub Progra2.pbs
32931.meta.cnca
[curso-106@login-2 ~]$ cat result.o32931
result.o32931 result.o32931
[curso-106@login-2 ~]$ cat result.o32931

-----
Operaciones matriciales
-----

suma de matrices

0 1 2 3 4 5 6 7 8 9
1 3 5 7 9 11 13 15 17 19
2 5 8 11 14 17 20 23 26 29
3 7 11 15 19 23 27 31 35 39
4 9 14 19 24 29 34 39 44 49
5 11 17 23 29 35 41 47 53 59
6 13 20 27 34 41 48 55 62 69
7 15 23 31 39 47 55 63 71 79
8 17 26 35 44 53 62 71 80 89
9 19 29 39 49 59 69 79 89 99

el tiempo duro 3 mil segundo de la suma de matrices

-----
resta de matrices
-----

0 1 2 3 4 5 6 7 8 9
1 1 1 1 1 1 1 1 1 1
2 1 0 -1 -2 -3 -4 -5 -6 -7
3 1 -1 -3 -5 -7 -9 -11 -13 -15
4 1 -2 -5 -8 -11 -14 -17 -20 -23
5 1 -3 -7 -11 -15 -19 -23 -27 -31
6 1 -4 -9 -14 -19 -24 -29 -34 -39
7 1 -5 -11 -17 -23 -29 -35 -41 -47
8 1 -6 -13 -20 -27 -34 -41 -48 -55
9 1 -7 -15 -23 -31 -39 -47 -55 -63

el tiempo duro 1 mil segundo de la resta de matrices

-----
multiplicacion de matrices
-----

0 723905 1447810 2171715 2895620 3619525 4343430 5067335 5791240 6515145
0 732290 1464580 2196870 2925160 3649450 4373740 5100030 5826320 6550610
0 740675 1481350 2222025 2962700 3703375 4444050 5184725 5925400 6666075
0 749060 1498120 2247180 2996240 3745380 4496050 5243420 5992480 6741540
0 757445 1514890 2272335 3029780 3779225 4544070 5302115 6059560 6817085
0 765830 1531660 2297490 3063320 3829150 4594980 5360810 6126640 6892470
0 774215 1548430 2322645 3096860 3871075 4645290 5419505 6193720 6967935
0 782600 1565200 2347800 3130400 3913080 4695600 5478200 6268880 7043480
0 790985 1581970 2372955 3163940 3954925 4745910 5536995 6327080 7118865
0 799370 1598740 2398110 3197480 3996850 4796220 5595590 6394960 7194330

el tiempo duro 75 mil segundo de la multiplicacion de matrices
[curso-106@login-2 ~]$

Progra2.pbs - test - Visual Studio Code
Archivo Editar Selección Ver Ir Depurar Tareas Ayuda

1 #PBS -N result
2 #PBS -q phi-nsh24
3 #PBS -l nodes=4
4 #PBS -l walltime=00:15:00
5
6
7 cd $PBS_0_WORKDIR
8
9 export OMP_NUM_THREADS=32
10 ./Run_file
11

```

- 5 Nodos:

```

Actividades Visual Studio Code lun 18:37
curso-106@login-2~

32891:meta:cnos
[curso-106@login-2 ~]$ ls
a.out Makefile mpd.hosts Proyecto result.e32884 result.e32887 result.e32891 result.e32886 result.o32890 Run_file torque:scripts
[curso-106@login-2 ~]$ cat result.o32891
-----
Operaciones matriciales
-----
suma de matrices
-----
  0  1  2  3  4  5  6  7  8  9
  1  3  5  7  9 11 13 15 17 19
  2  5  8 11 14 17 20 23 26 29
  3  7 11 15 19 23 27 31 35 39
  4  9 14 19 24 29 34 39 44 49
  5 11 17 23 29 35 41 47 53 59
  6 13 20 27 34 41 48 55 62 69
  7 15 23 31 39 47 55 63 71 79
  8 17 26 35 44 53 62 71 80 89
  9 19 29 39 49 59 69 79 89 99

el tiempo duro 7 mil segundo de la suma de matrices
-----
resta de matrices
-----
  0  1  2  3  4  5  6  7  8  9
  1  1  1  1  1  1  1  1  1  1
  2  1  0 -1 -2 -3 -4 -5 -6 -7
  3  1 -1 -3 -5 -7 -9 -11 -13 -15
  4  1 -2 -5 -8 -11 -14 -17 -20 -23
  5  1 -3 -7 -11 -15 -19 -23 -27 -31
  6  1 -4 -9 -14 -19 -24 -29 -34 -39
  7  1 -5 -11 -17 -23 -29 -35 -41 -47
  8  1 -6 -13 -20 -27 -34 -41 -48 -55
  9  1 -7 -15 -23 -31 -39 -47 -55 -63

el tiempo duro 1 mil segundo de la resta de matrices
-----
multiplicacion de matrices
-----
  0 328350 656700 985050 1313400 1641750 1970100 2298450 2626800 2955150
  1 333300 666600 999900 1333200 1666500 1999800 2333100 2666400 2999700
  2 338250 676500 1014750 1333800 1691250 2024500 2367750 2700900 3044050
  3 343200 686400 1029600 1372800 1716000 2059200 2402400 2745600 3088800
  4 348150 696300 1044450 1392600 1740750 2088900 2437050 2785200 3133350
  5 353100 706200 1059300 1412400 1765500 2118600 2471700 2824800 3177900
  6 358050 716100 1074150 1432200 1790250 2148300 2506350 2864400 3222450
  7 363000 726000 1089000 1452000 1815000 2178000 2541000 2904000 3267000
  8 367950 735900 1103850 1471800 1839750 2207700 2575650 2943600 3311550
  9 372900 745800 1118700 1491600 1864500 2237400 2610300 2983200 3356100

el tiempo duro 67 mil segundo de la multiplicacion de matrices
[curso-106@login-2 ~]$
  
```