

# Actividad: Matrices Distribuidas

Instituto Politécnico Nacional  
Escuela Superior de Computo  
Desarrollo de Sistemas Distribuidos  
Adrian González Pardo  
4CV1  
21/01

23 de octubre de 2020

## 1. Código fuente:

```
1 import java.net.Socket;
2 import java.io.DataOutputStream;
3 import java.io.DataInputStream;
4 import java.net.Socket;
5 import java.net.ServerSocket;
6 import java.lang.Thread;
7 import java.nio.ByteBuffer;
8
9 /*
10  * Archivo: Matriz.java
11  * @author Adrian Gonzalez Pardo
12  */
13
14 public class Matriz{
15
16     static int N=1000,
17         A [][]=new int[N][N],
18         B [][]=new int[N][N],
19         C [][]=new int[N][N];
20     static long timeT=0;
21
22     static Object lock=new Object();
23
24     static void read(DataInputStream f,byte[] b,int posicion,int longitud) throws
        Exception{
25         while (longitud>0){
26             int n=f.read(b,posicion,longitud);
27             posicion+=n;
28             longitud-=n;
29         }
30     }
31
32     static class Worker extends Thread{
33         Socket conexion;
34         Worker(Socket conexion){
35             this.conexion=conexion;
36         }
37
38         public void run(){
39             try{
40                 long time0=0,timeI=0;
41                 timeI=System.currentTimeMillis();
42                 DataOutputStream salida=new DataOutputStream(conexion.getOutputStream())
43
44 ;
```

```

43 DataInputStream entrada=new DataInputStream(conexion.getInputStream());
44 int id=entrada.readInt(),i,j;
45 System.out.println("Nodo "+(id+1)+" comunicando con server");
46 switch(id){
47     case 0:{
48         for(i=0;i<N/2;i++){
49             for(j=0;j<N;j++){
50                 salida.writeInt(A[i][j]);
51             }
52         }
53         System.out.print("Se envio A1 al nodo "+(id+1)+"\n");
54         for(i=0;i<N;i++){
55             for(j=0;j<N/2;j++){
56                 salida.writeInt(B[i][j]);
57             }
58         }
59         System.out.print("Se envio B1 al nodo "+(id+1)+"\n");
60         for(i=0;i<N/2;i++){
61             for(j=0;j<N/2;j++){
62                 C[i][j]=entrada.readInt();
63             }
64         }
65         System.out.print("Se recibio C1\n");
66     }
67     break;
68     case 1:{
69         for(i=0;i<N/2;i++){
70             for(j=0;j<N;j++){
71                 salida.writeInt(A[i][j]);
72             }
73         }
74         System.out.print("Se envio A1 al nodo "+(id+1)+"\n");
75         for(i=0;i<N;i++){
76             for(j=N/2;j<N;j++){
77                 salida.writeInt(B[i][j]);
78             }
79         }
80         System.out.print("Se envio B2 al nodo "+(id+1)+"\n");
81         for(i=0;i<N/2;i++){
82             for(j=N/2;j<N;j++){
83                 C[i][j]=entrada.readInt();
84             }
85         }
86         System.out.print("Se recibio C2\n");
87     }
88     break;
89     case 2:{
90         for(i=N/2;i<N;i++){
91             for(j=0;j<N;j++){
92                 salida.writeInt(A[i][j]);
93             }
94         }
95         System.out.print("Se envio A2 al nodo "+(id+1)+"\n");
96         for(i=0;i<N;i++){
97             for(j=0;j<N/2;j++){
98                 salida.writeInt(B[i][j]);
99             }
100         }
101         System.out.print("Se envio B1 al nodo "+(id+1)+"\n");
102         for(i=N/2;i<N;i++){
103             for(j=0;j<N/2;j++){
104                 C[i][j]=entrada.readInt();
105             }
106         }
107         System.out.print("Se recibio C3\n");
108     }
109     break;

```

```

110         case 3:{
111             for(i=N/2;i<N;i++){
112                 for(j=0;j<N;j++){
113                     salida.writeInt(A[i][j]);
114                 }
115             }
116             System.out.print("Se envio A2 al nodo "+(id+1)+"\n");
117             for(i=0;i<N;i++){
118                 for(j=N/2;j<N;j++){
119                     salida.writeInt(B[i][j]);
120                 }
121             }
122             System.out.print("Se envio B2 al nodo "+(id+1)+"\n");
123             for(i=N/2;i<N;i++){
124                 for(j=N/2;j<N;j++){
125                     C[i][j]=entrada.readInt();
126                 }
127             }
128             System.out.print("Se recibio C4\n");
129         }
130         break;
131     }
132     salida.close();
133     entrada.close();
134     conexion.close();
135     time0=System.currentTimeMillis();
136     long time=time0-timeI;
137     System.out.print("Tiempo de procesamiento en el hilo del node "+
138         (id+1)+": "+time+" ms\n");
139     synchronized(lock){
140         timeT+=time;
141     }
142 }catch (Exception e) {
143     System.err.println(e.getMessage());
144 }
145 }
146 }
147
148 public static void llena_matriz(int[][] a,int[][] b,int[][] c){
149     int i,j;
150     for(i=0;i<N;i++){
151         for(j=0;j<N;j++){
152             a[i][j]=2*i+j;
153             b[i][j]=2*i-j;
154             c[i][j]=0;
155         }
156     }
157 }
158
159 public static long checksum(int[] a){
160     long c=0;
161     for(int i=0;i<a.length;i++){
162         c+=a[i];
163     }
164     return c;
165 }
166
167 public static long checksum(long[] a){
168     long c=0;
169     for(int i=0;i<a.length;i++){
170         c+=a[i];
171     }
172     return c;
173 }
174
175 public static void muestra_matriz(int[][] y){
176     for(int i:y){

```

```

177         for(int j:i){
178             System.out.print("\t"+j+"\t");
179         }
180         System.out.print("\n");
181     }
182 }
183 public static void main(String[] args) throws Exception{
184     int i;
185     if(args.length<2){
186         System.out.println("Usage: java Matriz <node> <ip>");
187         System.exit(1);
188     }
189     int node=Integer.valueOf(args[0]);
190     switch(node){
191         case 0:
192             /*Iniciar matriz A,B
193
194             */
195             llena_matriz(A,B,C);
196             if(N==4){
197                 System.out.print("Matriz A\n");
198                 muestra_matriz(A);
199                 System.out.print("Matriz B\n");
200                 muestra_matriz(B);
201             }
202             ServerSocket servidor=new ServerSocket(50000);
203             Worker w[]=new Worker[4];
204             for(i=0;i<w.length;i++){
205                 Socket conexion=servidor.accept();
206                 w[i]=new Worker(conexion);
207                 w[i].start();
208             }
209             for(i=0;i<w.length;i++){
210                 w[i].join();
211             }
212             long checksumA[]=new long[N],cT;
213             for(i=0;i<N;i++){
214                 checksumA[i]=checksum(C[i]);
215             }
216             cT=checksum(checksumA);
217             System.out.print("Checksum de la matriz C: 0x"+Long.toHexString(cT)+"\n");
218             if(N==4){
219                 System.out.print("Matriz C\n");
220                 muestra_matriz(C);
221             }
222             System.out.println("Tiempo total de procesamiento en los hilos: "+
223                 timeT+" ms\nPromedio de tiempo de procesamiento en hilos: "+(timeT/4)+"
224                 ms");
225             break;
226             case 1:case 2:case 3:case 4:
227                 Socket conexion=null;
228                 while(true){
229                     try{
230                         conexion=new Socket(args[1],50000);
231                         break;
232                     }catch (Exception e){
233                         Thread.sleep(100);
234                     }
235                 }
236                 long timeI=0,timeO=0;
237                 timeI=System.currentTimeMillis();
238                 DataOutputStream salida=new DataOutputStream(conexion.getOutputStream());
239
240                 ;
241                 DataInputStream entrada=new DataInputStream(conexion.getInputStream());
242                 salida.writeInt(node-1);
243                 int a[][]=new int[N/2][N],
244                     b[][]=new int[N/2][N],

```

```

242         c[][]=new int[N/2][N/2],j;
243     for(i=0;i<N/2;i++){
244         for(j=0;j<N;j++){
245             a[i][j]=entrada.readInt();
246         }
247     }
248     for(i=0;i<N;i++){
249         for(j=0;j<N/2;j++){
250             b[j][i]=entrada.readInt();
251         }
252     }
253     for(i=0;i<N/2;i++){
254         for(j=0;j<N;j++){
255             for(int k=0;k<N/2;k++){
256                 c[i][k]+=a[i][j]*b[k][j];
257             }
258         }
259     }
260     for(i=0;i<N/2;i++){
261         for(j=0;j<N/2;j++){
262             salida.writeInt(c[i][j]);
263         }
264     }
265     salida.close();
266     entrada.close();
267     conexion.close();
268     time0=System.currentTimeMillis();
269     timeT=time0-timeI;
270     System.out.println("Tiempo de ejecucion del nodo "+node+": "+
271         timeT+" ms");
272     break;
273     default:
274         System.exit(1);
275     }
276 }
277 }

```

```

1 # Archivo Makefile
2 # @author Adrian Gonzalez Pardo
3 JVC=javac
4 SRCC=$(wildcard *.java)
5 OBJS=$(SRCC:.java=.class)
6 all: ${OBJS}
7
8 %.class: %.java
9     ${JVC} $<
10
11 .PHONY: clean
12
13 clean:
14     rm *.class

```

## 2. Capturas y descripción del programa

```

d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido
$ make
javac Matriz.java
$ ls
Makefile  Matriz$Worker.class  Matriz.class  Matriz.java  reporte/

```

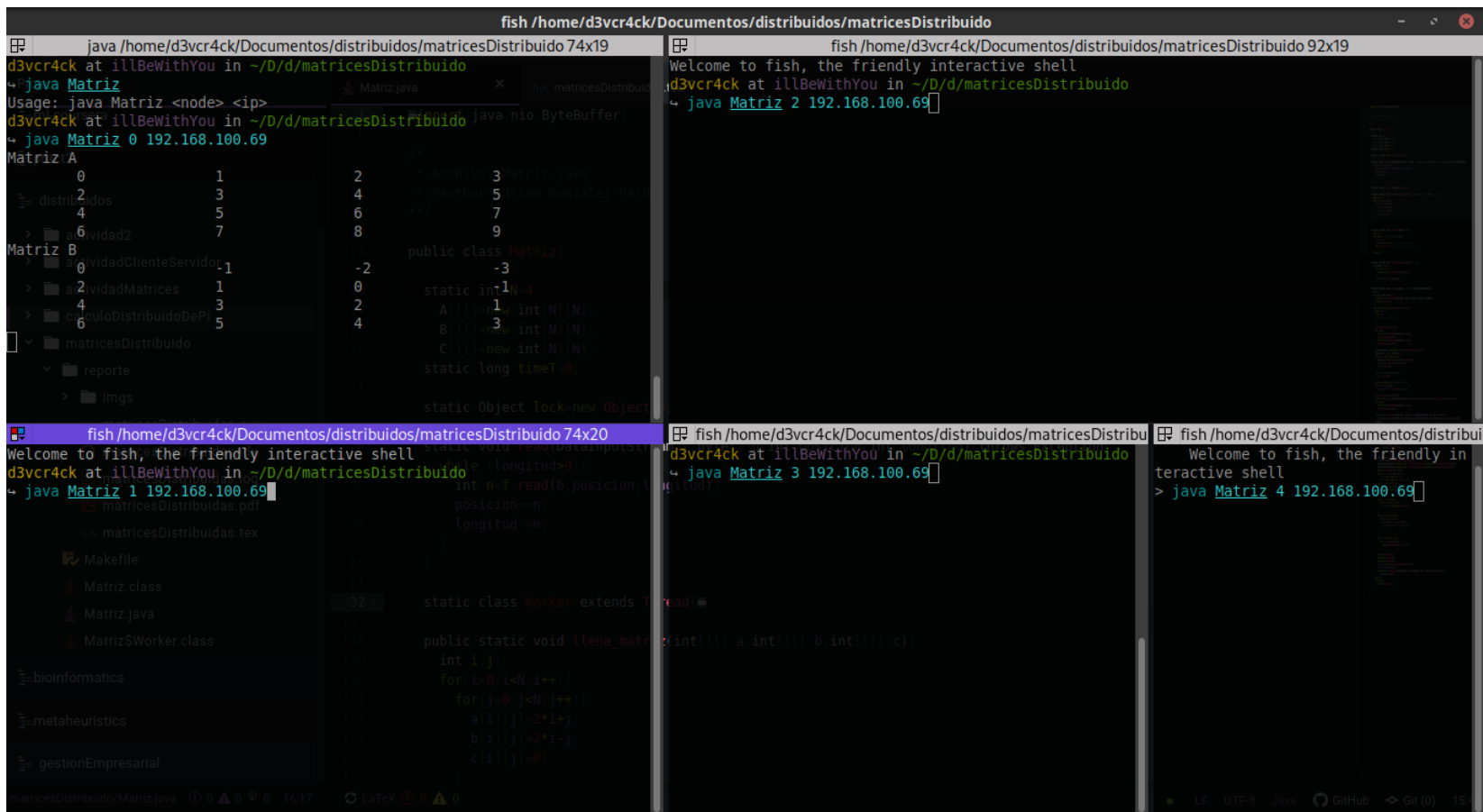
*En esta captura podemos ver la compilación rápida del archivo Matriz.java gracias al archivo y las tareas que realiza el archivo Makefile y la sencillez de simplemente escribir make en la terminal*

The image shows two side-by-side terminal windows. The left window, titled 'fish /home/d3vcr4ck/Documentos/distribuidos/matricesDistribuido 74x19', shows a file explorer on the left with a tree view containing folders like 'ml-coursera', 'pract2', 'distribuidos', 'actividad2', 'actividadClienteServidor', 'actividadMatrices', 'calculoDistribuidoDePi', 'matricesDistribuido', 'reporte', and 'imgs'. The main area shows the code for 'Matriz.java' with line numbers 1 to 22. The code includes imports for 'ByteBuffer' and 'FileChannel', a comment about the author, and the start of a 'public class Matriz' with static variables 'N=1000', 'A', 'B', 'C', 'timeT', and 'lock'. The right window, titled 'java /home/d3vcr4ck/Documentos/distribuidos/matricesDistribuido 92x19', shows the command 'java Matriz 2 192.168.100.69' being executed. The output on the right side of the right window is partially visible, showing a list of numbers.

```
fish /home/d3vcr4ck/Documentos/distribuidos/matricesDistribuido 74x19
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido
import java.nio.ByteBuffer;
/*
 * Archivo: Matriz.java
 * Author: Adrian Gonzalez Pardo
 */
public class Matriz{
    static int N=1000;
    A[][]=new int[N][N];
    B[][]=new int[N][N];
    C[][]=new int[N][N];
    static long timeT=0;
    static Object lock=new Object();
}
```

```
java /home/d3vcr4ck/Documentos/distribuidos/matricesDistribuido 92x19
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido
java Matriz 2 192.168.100.69
```

*En esta captura vemos que los nodos clientes se pueden ejecutar aun sin la existencia del servidor*



*En esta captura vemos que el server queda en espera de los clientes*



```

Nodo 1 comunicando con server
Se envio A1 al nodo 1
Se envio B1 al nodo 1
Se recibio C1
Tiempo de procesamiento en el hilo del node 1: 2 ms
Nodo 2 comunicando con server
Se envio A1 al nodo 2
Se envio B2 al nodo 2
Se recibio C2
Tiempo de procesamiento en el hilo del node 2: 3 ms
Nodo 3 comunicando con server
Se envio A2 al nodo 3
Se envio B1 al nodo 3
Se recibio C3
Tiempo de procesamiento en el hilo del node 3: 3 ms
Nodo 4 comunicando con server
Se envio A2 al nodo 4
Se envio B2 al nodo 4
Se recibio C4

```

En esta captura vemos la ejecución del servidor una vez que los clientes recibieron y enviaron sus datos con una matriz de 4x4.

```

Se envio A2 al nodo 4
Se envio B2 al nodo 4
Se recibio C4
Tiempo de procesamiento en el hilo del node 4: 2 ms
Nodo 3 comunicando con server
Se envio A2 al nodo 3
Se envio B1 al nodo 3
Se recibio C3
Tiempo de procesamiento en el hilo del node 3: 2 ms
Checksum de la matriz C: 0x250
Matriz C
28      22      16
52      38      24
76      54      32
100     70      40
Tiempo total de procesamiento en los hilos: 10 ms
Promedio de tiempo de procesamiento en hilos: 2 ms
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido

```

En esta captura vemos la continuación de la captura anterior.

```

Nodo 4 comunicando con server
Se envio B1 al nodo 1
Se envio B1 al nodo 3
Se envio A2 al nodo 4
Se recibio C1
Tiempo de procesamiento en el hilo del node 1: 20899 ms
Se recibio C3
Tiempo de procesamiento en el hilo del node 3: 18984 ms
Se envio B2 al nodo 2
Se envio B2 al nodo 4
Se recibio C2
Tiempo de procesamiento en el hilo del node 2: 24565 ms
Se recibio C4
Tiempo de procesamiento en el hilo del node 4: 17293 ms
Checksum de la matriz C: 0xfad8a880
Tiempo total de procesamiento en los hilos: 81741 ms
Promedio de tiempo de procesamiento en hilos: 20435 ms
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido

```

En esta captura vemos la ejecución del programa con una matriz de 1000x1000 datos.