

Actividad: Cálculo distribuido de PI

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=4,
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 int checksum(int[] a){
160     int c=0;
161     for(int i=0;i<a.length;i++){
162         c+=a[i];
163     }
164     return c;
165 }
166
167 public static void muestra_matriz(int[][] y){
168     for(int[] i:y){
169         for(int j:i){
170             System.out.print("\t"+j+"\t");
171         }
172         System.out.print("\n");
173     }
174 }
175 public static void main(String[] args) throws Exception{
176     int i;

```

```

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

```

```

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

```

```

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

```

2. Capturas y descripción del programa

```

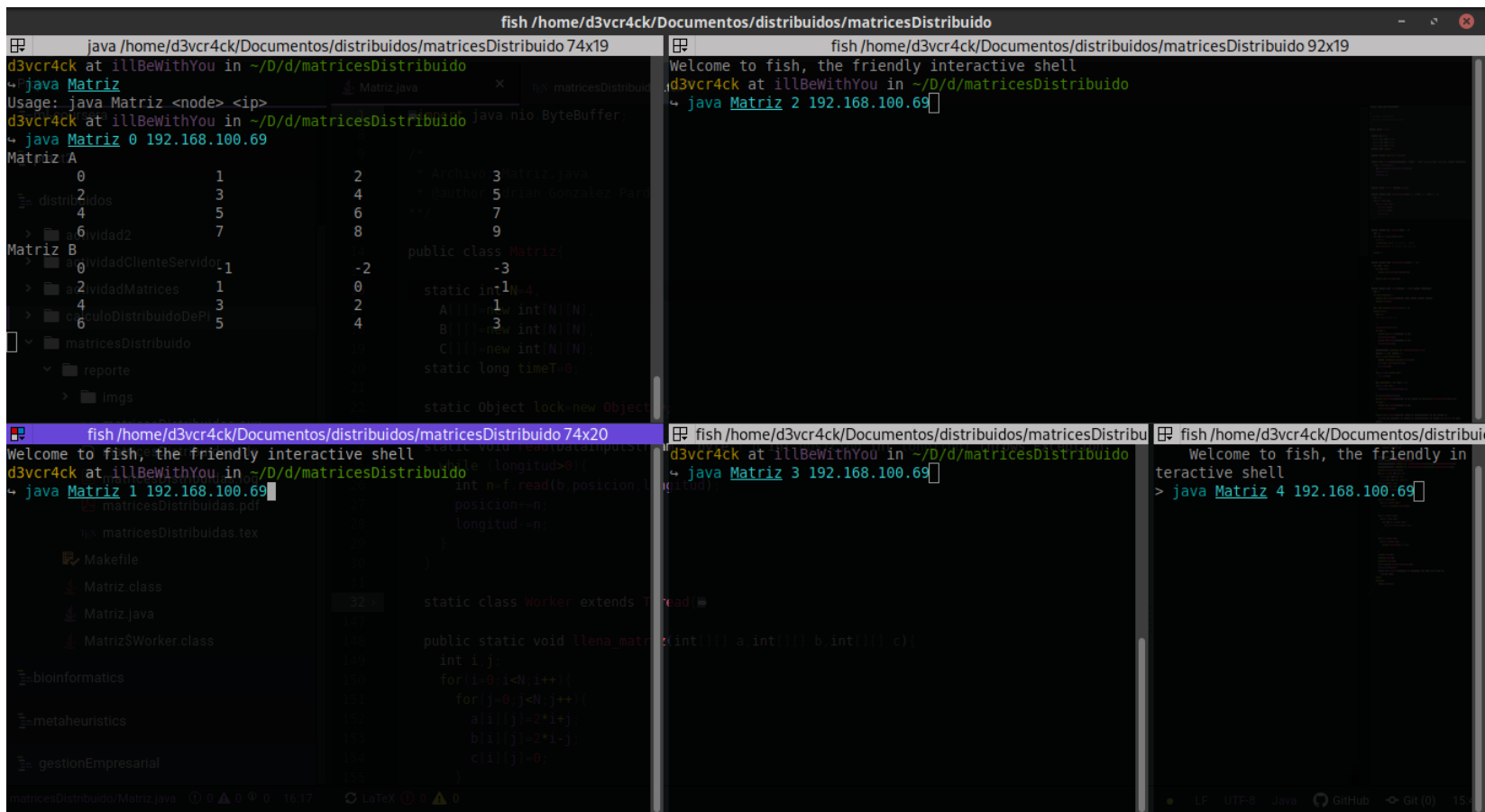
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido
ls
Makefile Matriz.java reporte/
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido lines=true,
ls
make
javac Matriz.java
d3vcr4ck at illBeWithYou in ~/D/d/matricesDistribuido
ls
Makefile 'Matriz$Worker.class' Matriz.class Matriz.java reporte/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

```

En esta captura podemos ver la compilación rápida del archivo `Matriz.java` gracias al archivo `Makefile` 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', displays 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 of 'Matriz.java' with line numbers 1 through 22. The code includes an import for 'ByteBuffer', a comment about the file and 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 window is partially visible, showing a list of numbers.

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.