

L05 – Sisteme Distribuite

Balog David Alexandru

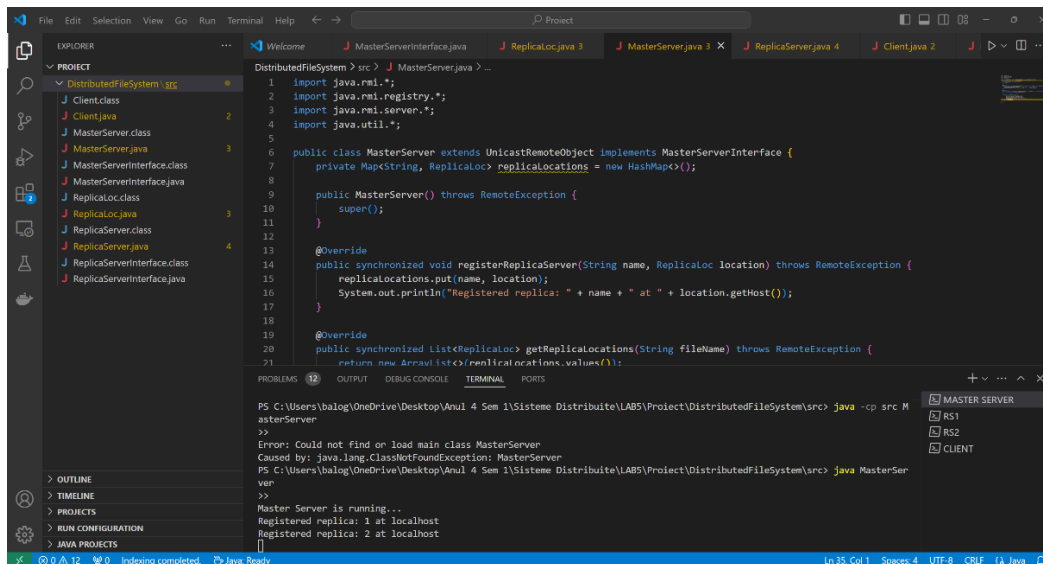
Link Github: <https://github.com/BalogDavid/L05-SISTEME-DISTRIBUITE>

Arhitectura

Sistemul de fișiere distribuit implementat utilizează **Remote Method Invocation (RMI)** pentru comunicarea între componente distribuite. Este compus din trei tipuri de noduri:

1. Master Server:

- Rol central care gestionează locațiile serverelor de replicare.
- Nu stochează date, doar coordonează replicarea și comunicarea între client și replici.



The screenshot shows an IDE with the following components:

- EXPLORER:** A project named 'DistributedFileSystem' with files including Client.class, Client.java, MasterServer.class, MasterServer.java, MasterServerInterface.class, MasterServerInterface.java, ReplicaLoc.class, ReplicaLoc.java, ReplicaServer.class, ReplicaServer.java, ReplicaServerInterface.class, and ReplicaServerInterface.java.
- EDITOR:** The file 'MasterServer.java' is open, showing the following code:

```
1 import java.rmi.*;
2 import java.rmi.registry.*;
3 import java.rmi.server.*;
4 import java.util.*;
5
6 public class MasterServer extends UnicastRemoteObject implements MasterServerInterface {
7     private Map<String, ReplicaLoc> replicaLocations = new HashMap<>();
8
9     public MasterServer() throws RemoteException {
10         super();
11     }
12
13     @Override
14     public synchronized void registerReplicaServer(String name, ReplicaLoc location) throws RemoteException {
15         replicaLocations.put(name, location);
16         System.out.println("Registered replica: " + name + " at " + location.getHost());
17     }
18
19     @Override
20     public synchronized List<ReplicaLoc> getReplicaLocations(String fileName) throws RemoteException {
21         return new ArrayList<>(replicaLocations.values());
22     }
23 }
```
- TERMINAL:** The output shows the command to compile and run the MasterServer, followed by an error message: 'Error: Could not find or load main class MasterServer'. The error is caused by 'java.lang.ClassNotFoundException: MasterServer'. The command to run the MasterServer is also shown: 'PS C:\Users\balog\OneDrive\Desktop\Anul 4 Sem 1\Sisteme Distribuite\LAB5\Project\DistributedFileSystem\src> java MasterServer'.

2. Replica Servers:

- Noduri responsabile pentru stocarea efectivă a fișierelor.
- Organizați într-un lanț utilizând mecanismul de **chained replication**:
 - Scriere: începe de la primul nod și trece prin fiecare nod din lanț.

- Citire: are loc întotdeauna de pe ultimul nod din lanț, pentru a garanta consistența datelor.

The screenshot shows an IDE with the following components:

- EXPLORER:** A project named 'DistributedFileSystem' with files: Client.class, Client.java, MasterServer.class, MasterServer.java, MasterServerInterface.class, MasterServerInterface.java, ReplicaLoc.class, ReplicaLoc.java, ReplicaServer.class, ReplicaServer.java, ReplicaServerInterface.class, and ReplicaServerInterface.java.
- EDITOR:** The file 'MasterServer.java' is open, showing the following code:


```

1 import java.rmi.*;
2 import java.rmi.registry.*;
3 import java.rmi.server.*;
4 import java.util.*;
5
6 public class MasterServer extends UnicastRemoteObject implements MasterServerInterface {
7     private Map<String, ReplicaLoc> replicaLocations = new HashMap<>();
8
9     public MasterServer() throws RemoteException {
10         super();
11     }
12
13     @Override
14     public synchronized void registerReplicaServer(String name, ReplicaLoc location) throws RemoteException {
15         replicaLocations.put(name, location);
16         System.out.println("Registered replica: " + name + " at " + location.getHost());
17     }
18
19     @Override
20     public synchronized List<ReplicaLoc> getReplicaLocations(String fileName) throws RemoteException {
21         return new ArrayList<>(replicaLocations.values());
22     }
23 }

```
- TERMINAL:** Shows the command 'PS C:\Users\balog\OneDrive\Desktop\Anul 4 Sem 1\Sisteme Distribuite\LAB5\Project\DistributedFileSystem\src> java -cp src R replicaServer 1' and the output:


```

Error: Could not find or load main class ReplicaServer
Caused by: java.lang.ClassNotFoundException: ReplicaServer
PS C:\Users\balog\OneDrive\Desktop\Anul 4 Sem 1\Sisteme Distribuite\LAB5\Project\DistributedFileSystem\src> java ReplicaSe
rver 1
Replica Server 1 is running...
[1] File written: testFile.txt

```
- PROBLEMS:** Shows the error message: 'Error: Could not find or load main class ReplicaServer'.
- OUTPUT:** Shows the output: 'Replica Server 1 is running...' and '[1] File written: testFile.txt'.
- PORTS:** Shows the ports: MASTER SERVER, RS1, RS2, and CLIENT.

The screenshot shows an IDE with the following components:

- EXPLORER:** A project named 'DistributedFileSystem' with files: Client.class, Client.java, MasterServer.class, MasterServer.java, MasterServerInterface.class, MasterServerInterface.java, ReplicaLoc.class, ReplicaLoc.java, ReplicaServer.class, ReplicaServer.java, ReplicaServerInterface.class, and ReplicaServerInterface.java.
- EDITOR:** The file 'MasterServer.java' is open, showing the following code:


```

1 import java.rmi.*;
2 import java.rmi.registry.*;
3 import java.rmi.server.*;
4 import java.util.*;
5
6 public class MasterServer extends UnicastRemoteObject implements MasterServerInterface {
7     private Map<String, ReplicaLoc> replicaLocations = new HashMap<>();
8
9     public MasterServer() throws RemoteException {
10         super();
11     }
12
13     @Override
14     public synchronized void registerReplicaServer(String name, ReplicaLoc location) throws RemoteException {
15         replicaLocations.put(name, location);
16         System.out.println("Registered replica: " + name + " at " + location.getHost());
17     }
18
19     @Override
20     public synchronized List<ReplicaLoc> getReplicaLocations(String fileName) throws RemoteException {
21         return new ArrayList<>(replicaLocations.values());
22     }
23 }

```
- TERMINAL:** Shows the command 'PS C:\Users\balog\OneDrive\Desktop\Anul 4 Sem 1\Sisteme Distribuite\LAB5\Project\DistributedFileSystem\src> java ReplicaSe rver 2' and the output:

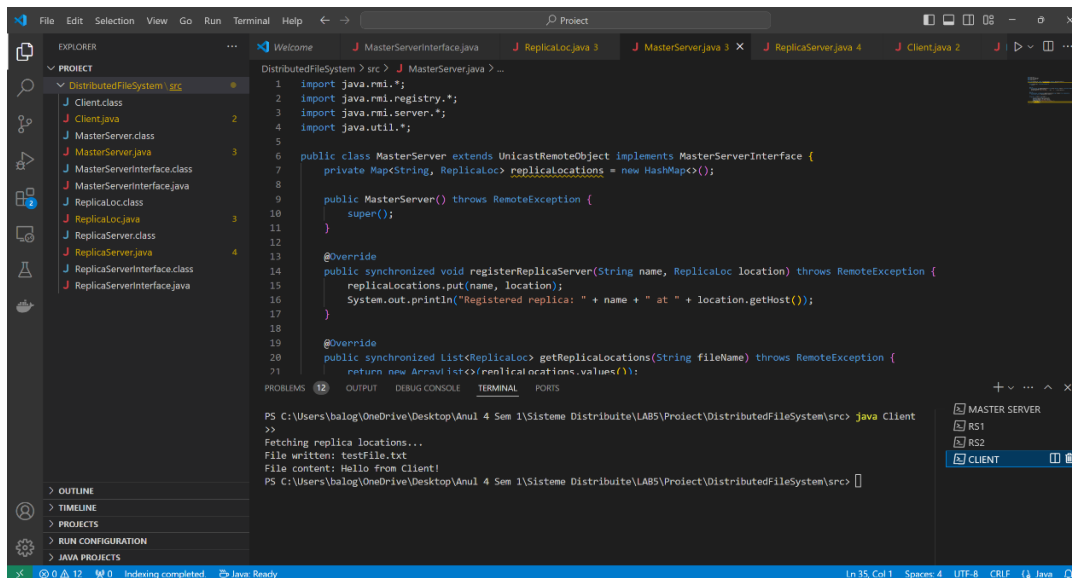

```

Replica Server 2 is running...

```
- PROBLEMS:** Shows the error message: 'Error: Could not find or load main class ReplicaServer'.
- OUTPUT:** Shows the output: 'Replica Server 2 is running...'.
- PORTS:** Shows the ports: MASTER SERVER, RS1, RS2, and CLIENT.

3. Client:

- Aplicație utilizator care trimite cereri de scriere sau citire către serverul master.
- Se conectează prin RMI la master pentru a primi locațiile replicilor și apoi interacționează direct cu acestea.



Funcționare

Scrierea fișierului

1. Clientul cere locația serverelor de replicare de la Master Server.
2. Clientul trimite fișierul către primul Replica Server din lanț.
3. Fiecare Replica Server scrie datele local și le transmite mai departe următorului server.
4. Ultimul server finalizează scrierea.

Citirea fișierului

1. Clientul cere locația serverelor de replicare de la Master Server.
2. Cererea de citire este redirecționată către ultimul Replica Server din lanț.
3. Replica Server returnează fișierul solicitat clientului.

Avantaje

1. **Consistența datelor:**

Citirea de pe ultimul nod din lanț garantează accesul la cele mai recente versiuni ale fișierelor.

2. **Toleranța la defectare:**

Dacă un Replica Server eșuează, restul lanțului poate continua să funcționeze, deși cu performanță redusă.

3. **Distribuirea încărcării:**

Scrierile sunt procesate de mai multe servere, ceea ce reduce riscul de suprasolicitare a unui singur nod.

4. **Extensibilitate:**

Noi servere de replicare pot fi adăugate relativ ușor în lanț.

Dezavantaje

1. **Timp de latență crescut:**

Scrierea fișierelor implică propagarea datelor prin toate nodurile din lanț, ceea ce poate deveni lent pentru sisteme mari.

2. **Dependență de noduri intermediare:**

Dacă un nod din mijlocul lanțului eșuează, trebuie repositionat lanțul pentru a asigura continuitatea.

3. **Single point of failure pentru Master Server:**

Dacă Master Server devine indisponibil, clienții nu mai pot accesa locațiile replicilor.

4. **Lipsa scalabilității pentru citire:**

Toate cererile de citire sunt redirecționate către ultimul nod din lanț, ceea ce poate deveni un punct de îngustare pentru sisteme cu trafic mare.