23 Novembre
2021

Esercitazione 6

Introduzione alle JAVA RMI

Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Obiettivi esercitazione

Hard skills

1. Invocazione di procedure remote mediante JAVA RMI
2. Manipolazione di file di testo remoti

Soft skills

1. Collaborazione per raggiungere un fine comune
2. Organizzazione del proprio lavoro in team

conta_righe

Cliente

1. Recupero riferimento oggetto remoto da RMI Registry

```
try {
    registryHost = args[0];
    serviceName = args[1];
    ...
    String completeName = "//" + registryHost + ":" + REGISTRYPORT + "/" + serviceName;
    IServerImpl serverRMI = (IServerImpl) Naming.lookup(completeName);
    System.out.println("ClientRMI: Servizio \"" + serviceName + "\" connesso");
    ...
} catch (Exception e) { ... }
```

2. Lettura dell'input dall'utente

3. Invocazione metodo remoto **conta_righe** e presentazione del risultato

Servitore

1. Registrare il riferimento remoto sul registry alla locazione corretta

```
public static void main(String[] args) throws RemoteException {
    final int REGISTRYPORT = 1099;
    String registryHost = "localhost";
    String serviceName = "ServerImpl";

    // Registrazione del servizio RMI
    String completeName = "//" + registryHost + ":" + REGISTRYPORT + "/" + serviceName;
    try {
        ServerImpl serverRMI = new ServerImpl();
        Naming.rebind(completeName, serverRMI);
        System.out.println("Server RMI: Servizio \"" + serviceName + "\" registrato");
    } catch (Exception e) { ... }
}
```

2. Attesa di invocazioni da parte del cliente

3. Elaborazione richiesta e restituzione del risultato

elimina_righe

Cliente

1. Recupero riferimento oggetto remoto da RMI Registry

```
try {
    registryHost = args[0];
    serviceName = args[1];
    ...
    String completeName = "//" + registryHost + ":" + REGISTRYPORT + "/" + serviceName;
    IServerImpl serverRMI = (IServerImpl) Naming.lookup(completeName);
    System.out.println("ClientRMI: Servizio \"" + serviceName + "\" connesso");
    ...
} catch (Exception e) { ... }
```

2. Lettura dell'input dall'utente

3. Invocazione metodo remoto **elimina_righe** e presentazione del risultato

Servitore

1. Registrare il riferimento remoto sul registry alla locazione corretta

```
public static void main(String[] args) throws RemoteException {
    final int REGISTRYPORT = 1099;
    String registryHost = "localhost";
    String serviceName = "ServerImpl";

    // Registrazione del servizio RMI
    String completeName = "//" + registryHost + ":" + REGISTRYPORT + "/" + serviceName;
    try {
        ServerImpl serverRMI = new ServerImpl();
        Naming.rebind(completeName, serverRMI);
        System.out.println("Server RMI: Servizio \"" + serviceName + "\" registrato");
    } catch (Exception e) { ... }
}
```

2. Attesa di invocazioni da parte del cliente

3. Elaborazione richiesta e restituzione del risultato

4

Implementazione conta_righe

```java
public interface IServerImpl extends Remote {

    int conta_righe(String filePath, int wordsInLineCount) throws RemoteException;

    ...
}
```

Cliente

```java
while ((service = stdIn.readLine()) != null) {
    // C = CONTA RIGHE
    if (service.equals("C")) {
        System.out.print("Path del file? ");
        fileName = stdIn.readLine();
        int wordsInLineCount = -1;
        System.out.print("Numero di parole nella riga?");

        while (wordsInLineCount <= -1) {
            wordsInLineCount = Integer.parseInt(stdIn.readLine());
            if (wordsInLineCount < 0) {
                System.out.println("Inserisci un intero positivo!");
                System.out.print("Numero di parole nella riga?");
            }
        }

        try {
            System.out.println("ClientRMI: righe che contengono un numero di parole maggiore"
                    + "dell'intero inviato: "
                    + serverRMI.conta_righe(fileName, wordsInLineCount));
        } catch (RemoteException e) {
            System.out.println(e.getMessage());
        }
    }
    ...
}
```

Servitore

```java
@Override
public int conta_righe(String filePath, int wordsInLineCount) throws RemoteException {
    File file = new File(filePath);

    try {
        BufferedReader reader = new BufferedReader(new FileReader(file));
        int c;
        int rowWords = 0;
        int result = 0;

        while ((c = reader.read()) != -1) {
            if (c == 13) {
                rowWords++;

                if (rowWords > wordsInLineCount) {
                    result++;
                }

                rowWords = 0;
            } else {
                if (c == ' ') {
                    rowWords++;
                }
            }
        }
        reader.close();

        return result;
    } catch (FileNotFoundException e) {
        throw new RemoteException("Server RMI: Il file " + filePath + " non esiste!");
    } catch (IOException e) {
        throw new RemoteException("Server RMI: Errore nella lettura del file!");
    }
}
```

Implementazione elimina_righe

```java
public interface IServerImpl extends Remote {

    ...

    String[] elimina_riga(String filePath, int rowToDelete) throws RemoteException;
}
```

Cliente

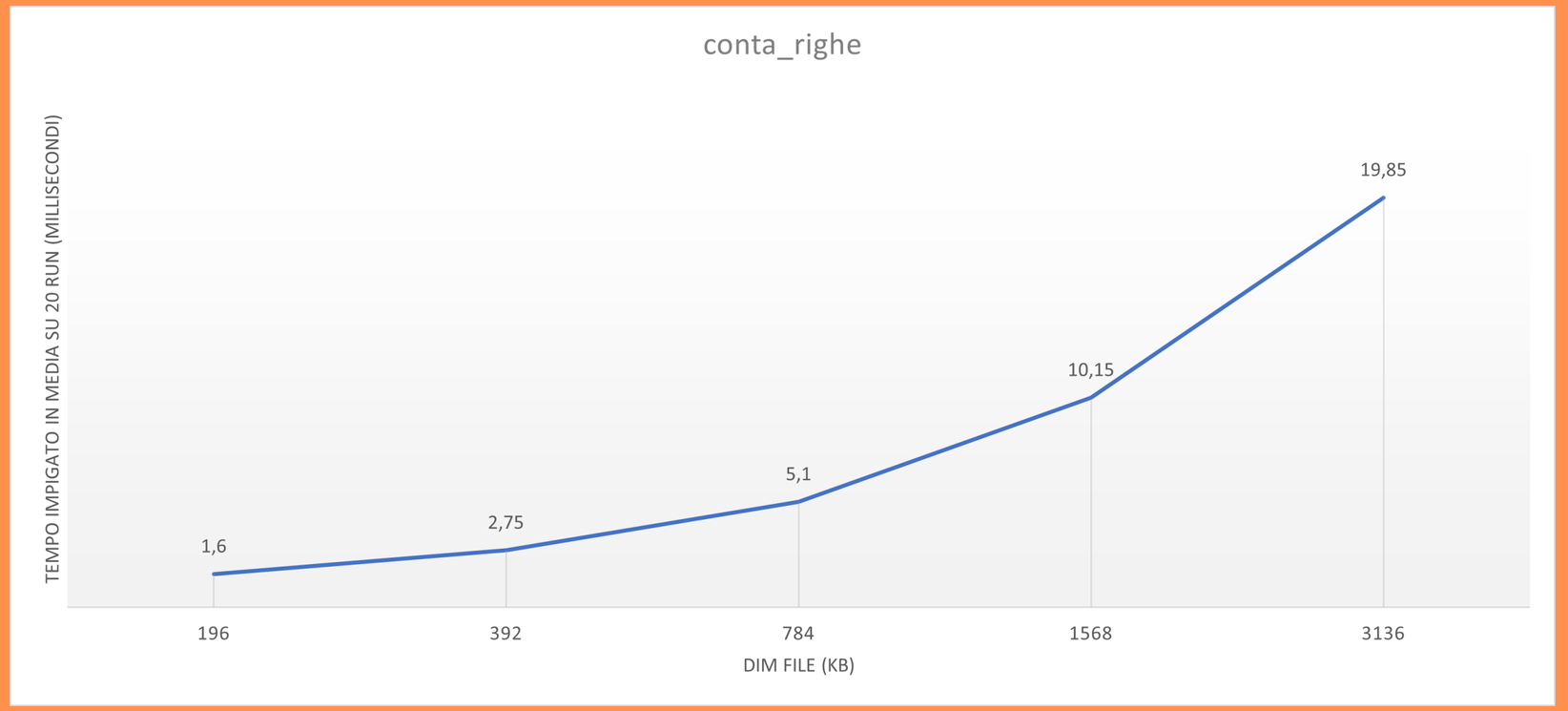
```java
while ((service = stdIn.readLine()) != null) {
    ...

    // E = ELIMINA RIGA
    else if (service.equals("E")) {
        System.out.print("Path del file? ");
        fileName = stdIn.readLine();
        int rowToDelete = -1;
        System.out.print("Indice riga da eliminare");

        while (rowToDelete <= -1) {
            rowToDelete = Integer.parseInt(stdIn.readLine());
            if (rowToDelete < 0) {
                System.out.println("Inserisci un numero di parole positive da contare!");
            }
        }

        try {
            String[] res = serverRMI.elimina_riga(fileName, rowToDelete);
            System.out.println("Nome del file modificato: " + res[0]);
            System.out.println("Nuove totale righe nel file: " + res[1]);
        } catch (RemoteException e) {
            System.out.println(e.getMessage());
        }
    } else {
        System.out.println("Servizio non disponibile");
    }
}
```

Servitore

```java
@Override
    public synchronized String[] elimina_riga(String filePath, int rowToDelete) throws
RemoteException {
        String[] result = new String[2];
        File file = new File(filePath);
        File tempOut = new File(filePath + "_temp");

        try {
            BufferedReader reader = new BufferedReader(new FileReader(file));
            BufferedWriter writer = new BufferedWriter(new FileWriter(tempOut));
            String line;
            int rowIndex = 1;

            while ((line = reader.readLine()) != null) {
                if (rowIndex != rowToDelete) {
                    writer.write(line + System.lineSeparator());
                }
                rowIndex++;
            }
            reader.close();
            writer.close();

            if (rowIndex < rowToDelete) {
                throw new RemoteException("Server RMI: il file contiene solo "
                 + rowIndex
                 + "righe, riga: " + rowToDelete + " inesistente!");
            }
            if(!file.delete()) { ... }
            if(!tempOut.renameTo(file)) { ... }

            result[0] = filePath;
            result[1] = String.valueOf(rowIndex - 1);
            return result;
        } catch (FileNotFoundException e) {
            throw new RemoteException("Server RMI: Il file " + filePath + " non esiste!");
        } catch (IOException e) {
            throw new RemoteException("Server RMI: Errore nella lettura del file!");
        }
    }
```

Benchmark – conta_righe

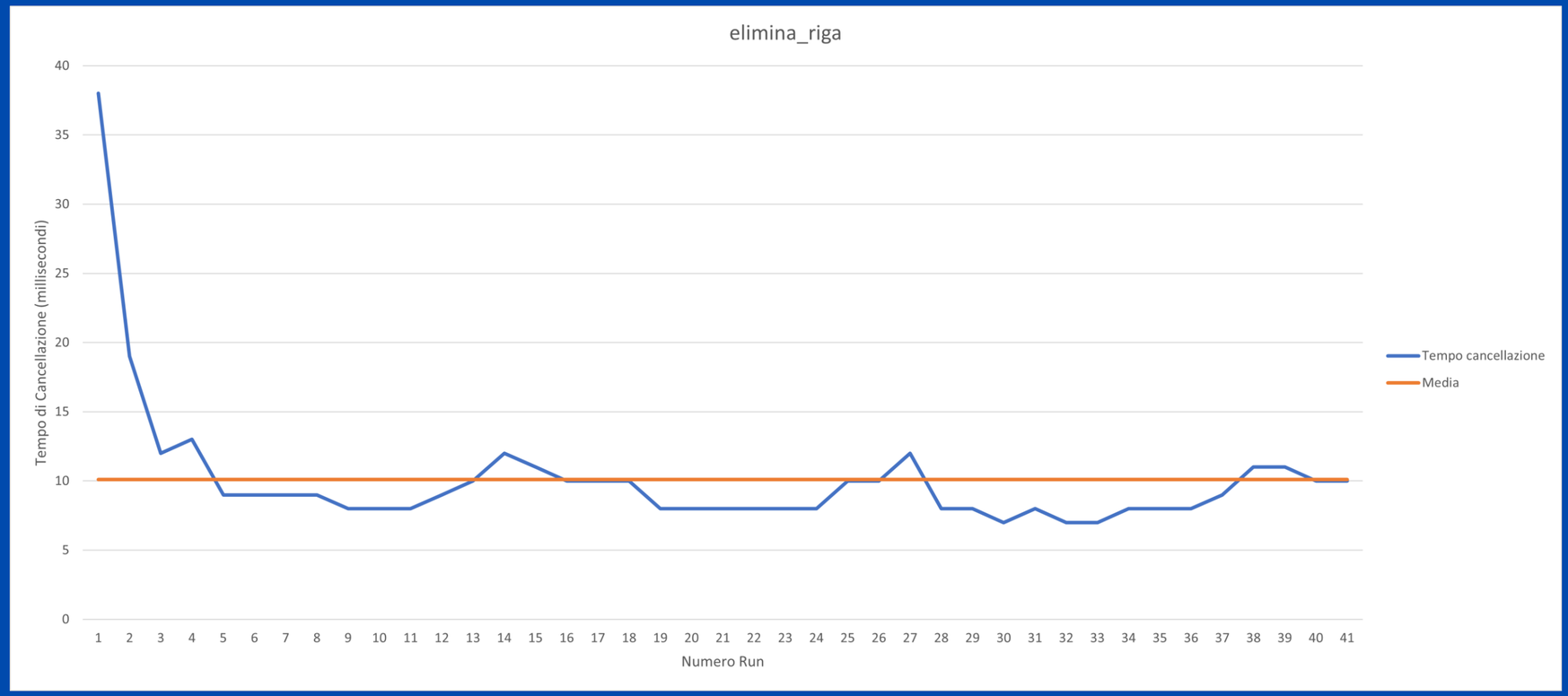
I valori riportati nel grafico sono il risultato della media dei valori ottenuti durante 20 esecuzioni.



Caratteristiche elaboratore: CPU Ryzen 7 5800X | RAM 16gb ddr4 3600mhz

Benchmark – elimina_riga

La linea arancione rappresenta la media dei valori ottenuti durante 40 esecuzioni, effettuate su lo stesso file di dimensione fissa 3136 KB.



Caratteristiche elaboratore: CPU Ryzen 7 5800X | RAM 16gb ddr4 3600mhz

Gestione progetto

Coding

Team

23 Novembre 2021

Grazie !

Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo