

Esercitazione 7

Belano Andrea, Ceccolini Gabriele, Loddo Filippo, Merenda Simone.
Gruppo 12

Specifiche del Progetto

Lo scopo del progetto è quello di modificare un progetto già esistente implementando metodi per la ricerca tramite tag e l'associazione di questi ai servizi. È inoltre necessario modificare il Cliente e Servitore del progetto per implementare i nuovi metodi.

RegistryRemoteTagImpl

La richiesta era di aggiungere i metodi `cercaTag()` e `associaTag()`. Questi fanno uso di una matrice in cui in ogni colonna ci sono i nomi logici dei servizi che hanno associato l'ennesimo tag. Per agevolare l'accesso a questa struttura dati, usiamo un enum per i tag.

```
String[][] tagTable = new String[tagNum][tableSize];

public String[] cercaTag(String tag) throws RemoteException {
    try {
        lock.readLock().lock();
        int tagEq = Tag.valueOf(tag).getValue();
        String result[] = new String[indexTag[tagEq]];
        for (int i = 0; i < indexTag[tagEq]; i++) {
            result[i] = tagTable[tagEq][i];
        }
        return result;
    } finally {
        lock.readLock().unlock();
    }
}

public boolean associaTag(String nomeLogicoServer, String tag) throws RemoteException {
    if (nomeLogicoServer == null) {
        return false;
    }
    try {
        lock.writeLock().lock();
        int tagEq = Tag.valueOf(tag).getValue();
        tagTable[tagEq][indexTag[tagEq]] = nomeLogicoServer;
        indexTag[tagEq] = indexTag[tagEq] + 1;
    } finally {
        lock.writeLock().unlock();
    }
    return true;
}
```


ServerCongressoImpl

In questa classe, per far uso del metodo associaTag(), all'avvio del servizio l'utente può inserire come parametri i tag a cui vuole che il servizio sia associato.

Registriamo il servizio nel registry e successivamente gli associamo i tag presi come argomento

```
if (args.length == 0) {
    System.out.println("Inserire il nome dell'host remoto e, opzionalmente, la porta e i tag");
    System.exit(1);
}
String registryRemotoHost = args[0];
if (args.length >= 2) {
    try {
        registryRemotoPort = Integer.parseInt(args[1]);
    } catch (Exception e) {
        System.out.println("Inserire il nome dell'host remoto e, opzionalmente, la porta e i tag");
        System.exit(2);
    }
    if (args.length >= 3) {
        tags = new String[args.length - 2];
        for (int i = 2; i < args.length; i++) {
            tags[i - 2] = args[i];
        }
    }
}
```

```
String completeRemoteRegistryName = "/" + registryRemotoHost + ":" + registryRemotoPort + "/" + registryRemotoName;

try {
    RegistryRemotoTagServer registryRemoto = (RegistryRemotoTagServer) Naming.lookup(completeRemoteRegistryName);
    ServerCongressoImpl serverRMI = new ServerCongressoImpl();
    registryRemoto.aggiungi(serviceName, serverRMI);
    if (args.length >= 3) {
        for (int i = 0; i < tags.length; i++) {
            registryRemoto.associaTag(serviceName, tags[i]);
        }
    }
    System.out.println("Server RMI: Servizio \"" + serviceName + "\" registrato");
} catch (Exception e) {
    System.err.println("Server RMI \"" + serviceName + "\": " + e.getMessage());
    e.printStackTrace();
    System.exit(1);
}
```

ClientCongresso

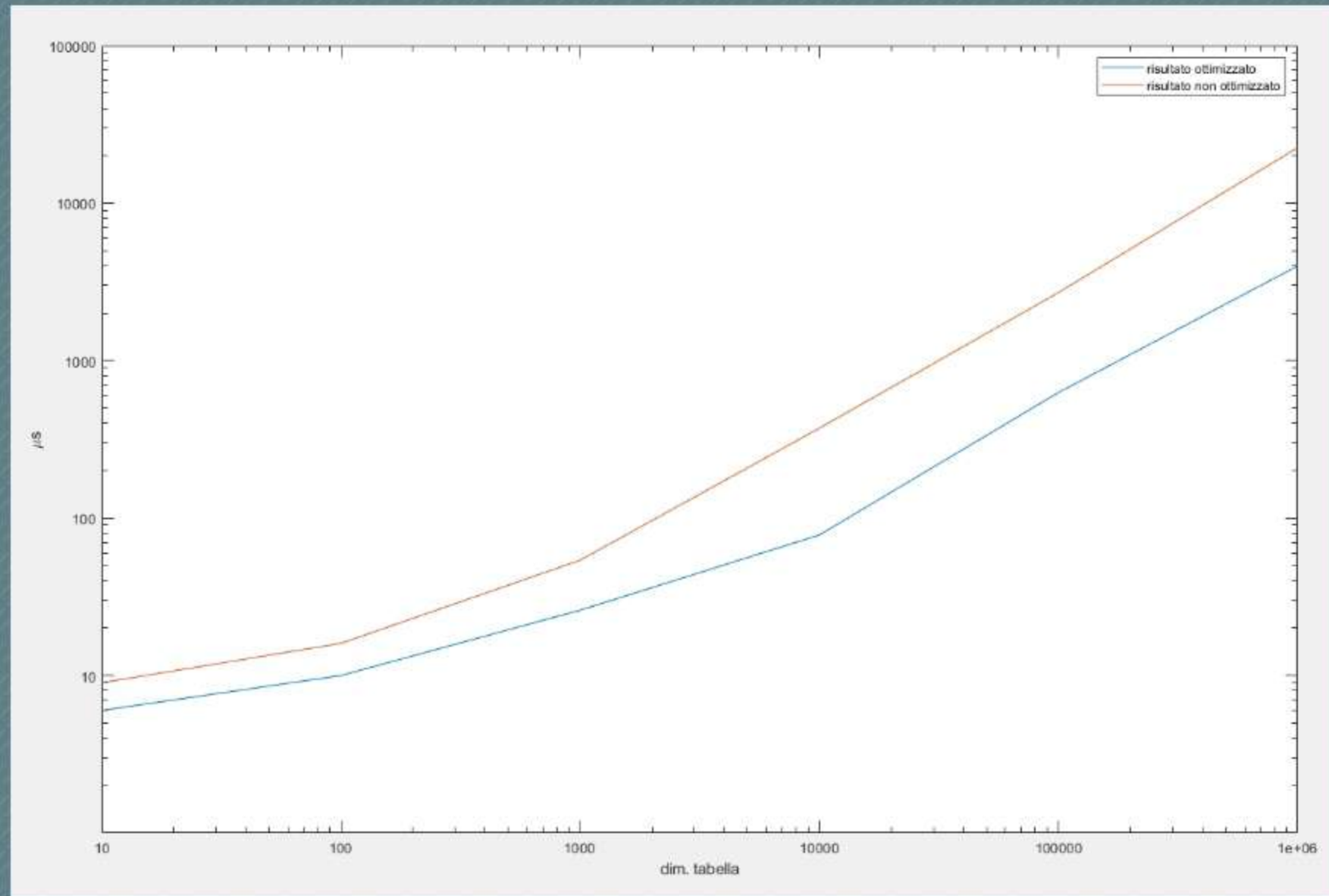
Alla classe è stata aggiunta una parte iniziale in cui l'utente può scegliere a che servizio collegarsi. Questi infatti può recuperare:

- il riferimento remoto inserendo il nome logico del servizio;
- tutti i riferimenti remoti inserendo un nome logico e scegliere quale utilizzare;
- tutti i nomi logici dei servizi a cui è associato un certo tag inserito dall'utente

```
while (serverRMI == null) {
    req = stdin.readLine();
    if (req.equals("C")) {
        System.out.println("Inserisci il nome del servizio");
        nome = stdin.readLine();
        server = registryRemoto.cerca(nome);
        if (server == null) {
            System.out.println("Nessun servizio disponibile con quel nome");
        } else if (!(server instanceof ServerCongresso)) {
            System.out.println("Servizio non supportato");
        } else {
            serverRMI = (ServerCongresso) server;
        }
    } else if (req.equals("A")) {
        System.out.println("Inserisci il nome del servizio");
        nome = stdin.readLine();
        servers = registryRemoto.cercaTutti(nome);
        if (servers.length == 0) {
            System.out.println("Nessun servizio disponibile con quel nome");
        } else {
            index = 0;
            for (int i = 0; i < servers.length; i++) {
                if (servers[i] instanceof ServerCongresso) {
                    possibile[index] = (ServerCongresso) servers[i];
                    index++;
                }
            }
            if (index == 0) {
                System.out.println("Non ci sono servizi supportati");
            } else {
                System.out.println("Trovati " + index + " servizi corrispondenti a quel nome");
                System.out.println("Quale vuoi? (1 - " + index + ")");
                num = Integer.parseInt(stdin.readLine());
                if (num > index || num < 1) {
                    System.out.println("Sbagliato");
                } else {
                    serverRMI = (ServerCongresso) possibile[num - 1];
                }
            }
        }
    } else if (req.equals("T")) {
        System.out.println("Inserisci il tag da cercare");
        tag = stdin.readLine();
        nomi = registryRemoto.cercaTag(tag);
        if (nomi.length == 0) {
            System.out.println("Non ci sono servizi con quel tag");
        } else {
            System.out.println("Trovati i seguenti servizi:");
            for (int i = 0; i < nomi.length; i++) {
                System.out.println(nomi[i]);
            }
        }
    } else {
        System.out.println("Riprova");
    }
}
```


Test

I test sono stati eseguiti sulla ricerca per Tag. I tempi comprendono anche il tempo di elaborazione della risposta. Al nostro algoritmo è stato affiancato uno che, al posto di utilizzare una struttura dati a parte, utilizza la tabella dei riferimenti aggiungendo dei campi dove sono inseriti gli eventuali tag.



Grazie per l'ascolto

Belano Andrea, Ceccolini Gabriele, Loddo Filippo, Merenda Simone.
Gruppo 12