

Sistemas Distribuídos 2017/2018 — Meta 1
25 de outubro de 2017



iVotas: Voto Eletrónico na UC

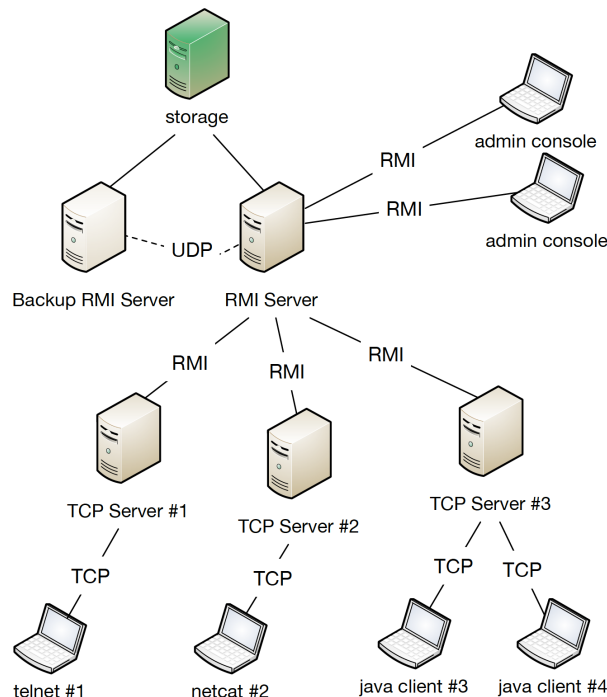
Guilherme Cardoso Gomes da Silva - 2014226354

João Pedro Costa Ferreira - 2014197760

Contents

1	Arquitetura do Software	2
2	Funcionamento TCP	3
3	Funcionamento RMI	3
4	Distribuição de Tarefas	3
5	Testes de Software	4

1 Arquitetura do Software



1- Arquitetura do Projeto

O software tem a estrutura recomendada no enunciado. Existem dois servidores RMI com código idêntico, em que um assume o papel de servidor principal e outro o papel de RMI de Backup. Os servidores RMI são responsáveis por gerir toda a informação, através de ficheiros de objectos. Os servidores comunicam entre si através de uma conexão UDP na qual o servidor backup recebe periodicamente pings do primário. Quando deixa de receber pings durante uma certa quantidade de tempo, assume que o primário foi abaixo e tenta substituir este, ligando-se à porta de registry a qual o primeiro servidor estava ligado.

As admin consoles e os servidores TCP simplesmente ligam-se ao RMI na porta fornecida. Para fins de notificações a tempo real, as consolas de administração têm também 2 threads a correr em paralelo à principal: uma para verificar quando uma eleição fecha, outra para controlar quando são associadas/removidas mesas de voto. Os clientes ligam-se aos servidores TCP através de uma ligação UDP.

O nosso projeto ainda não implementa base de dados. Os dados são armazenados em java objects e posteriormente guardados em ficheiros de objetos.

2 Funcionamento TCP

As máquinas TCP servem como ponto de ligação entre o cliente e o RMI, garantindo que este não tem acesso a informação que não deve. O servidor TCP conecta-se ao servidor RMI através do registo criado por este e ao cliente através de um socket TCP.

Aquando de uma ligação de um novo cliente, este identifica-se através do seu ID. Se o ID existir é desbloqueado a consola de terminal de voto, onde o utilizador terá de fazer login através de user e pass, escolher uma das eleições dipsoníveis e a lista na qual quer votar (Todas as eleições têm um campo "0. " que serve como voto em branco e qualquer escolha de lista de candidatos que não consta na lista apresentada é contabilizado como voto nulo).

Não foi implementado protocolo TCP, ou seja, a comunicação entre o servidor e cliente é feita da seguinte forma: O servidor envia uma string que descreve o que estará à espera de receber e o cliente envia a resposta também na forma de string.

3 Funcionamento RMI

O servidor RMI é o servidor central (replicado) que armazena todos os dados da aplicação, no nosso caso através de ficheiros de objetos, suportando todas as operações necessárias para a aplicação.

O servidor RMI que se liga primeiro é tratado como o servidor principal e enquanto está ativo manda pings por UDP ao servidor backup, que está receptivo a estes. Se o servidor principal é fechado/desliga-se, o servidor secundário espera por 5 heartbeats. Quando não os recebe, tenta então ligar-se à porta da registry onde o primário estava ligado. Caso seja bem sucedido, toma o papel de servidor primário e começa a mandar pings UDP. O servidor que foi abaixo, quando se liga novamente, torna-se então o novo servidor secundário.

Todas as funções que estão a ser usadas pelo servidor RMI são descritas em detalhe no javadoc.

4 Distribuição de Tarefas

Guilherme Silva – Consola Admin, Interface Admin, funções RMI relativas a esta e relatório (Arquitetura geral, servidor RMI, testes de software)

João Ferreira – TCP Server, Interface Server, funções RMI relativas a este, relatório (Servidor TCP, testes de software), javadoc, readme.txt

5 Testes de Software

Teste	Descrição	Resultado Esperado	Resultado Obtido
Registar user	Adicionar user aos registos	User acrescentado ao ficheiro de objetos	Pass
Registar departamento	Registar novo departamento no sistema	Departamento acrescentado com sucesso	Pass
Criar eleição	Começar nova eleição (e acrescentar esta ao sistema)	Eleição acrescentada à base de dados com sucesso	Pass
Adiciona mesas de voto	Permite que departamentos sejam utilizados para votar em eleições	Permitir o cliente apenas votar em departamentos com mesas	Pass – As mesas de voto são criadas; Fail – O cliente consegue ainda votar em departamentos sem mesa associada
Remover mesas de voto	Remove uma mesa de voto associada a um departamento	Deixar de permitir um cliente votar num certo departamento	Pass – As mesas são removidas; Fail – Como os clientes votam independentemente se a mesa existe ou não, é redundante
Criar listas de candidatos	Associar users registados a uma lista	Criação e registo de uma nova lista candidata	Pass
Editar listas de candidatos	Editar dados de uma lista especificada	Mudar os dados de uma lista de candidatos já registada	Pass
Remover lista de candidatos	Remover uma lista de candidatos da base de dados	Remover a lista especificada dos ficheiros de objetos	Pass
Alterar propriedades de uma eleição	Editar a eleição especificada se ainda não tiver começado	Edição editada	Pass
Saber em que local votou cada eleitor	Saber em que mesas cada eleitor votou	Lista de mesas onde o eleitor votou	Fail
Histórico de votos de um eleitor	Saber em que listas cada eleitor votou por eleição	Listar cada eleição onde o eleitor votou e em que lista votou	Pass
Consola de admin actualizada a tempo real	Receber notificações relativamente aos estados das mesas de votos/eleições expiradas/votos	Notificações em Real Time de eventos que se passam enquanto a consola está ligada	Pass – Eleições expiradas e mesas criadas/removidas mostradas em real time; Fail – Não mostra quando um user vota
Consola mostra mesas on/off	Verificar estado das mesas de voto	Verificar mesas que estão online no momento da invocação	Pass – Apesar de os users não verificarem se o departamento tem mesa ou não, é possível saber quais estão ligadas
Eleição termina correctamente na data indicada	Deixa de ser possível votar nesta eleição quando a data de fim desta passa	Fechar a eleição e anunciar o vencedor	Pass – A eleição termina na altura indicada; Fail – Não demonstra o vencedor desta"

Consultar dados das eleições passadas	Consultar vencedores/votos das eleições passadas	Estatísticas relativamente à eleição passada (vencedor/votos brancos/nulos,...)	Fail
Identificar eleitor na mesa de voto	Pedir ao eleitor para se autenticar na mesa de voto (por ID)	Verificar se o eleitor está nos registos	Pass
Autenticar eleitor na terminal	Pedir a password e nome do eleitor	Autenticar o eleitor	Pass
Votar no terminal de voto	Escolher lista em qual votar	Registar voto nos ficheiros de objetos	Pass
Avaria de um servidor RMI não tem qualquer efeito nos clientes	-	Não se perde informação quando o RMI falha	Pass
Avárias temporárias (<30s) dos 2 RMIs são invisíveis para clientes	-	Clientes não se apercebem de falha no RMI	Fail
Terminal de voto bloqueado automaticamente após 120s sem uso	-	Clientes não conseguem efetuar mais operações após 120 segundos sem usar terminal	Pass
Heartbeats via UDP entre o servidor primário e o secundário	-	RMI principal e secundário trocam informação	Pass
Em caso de avaria longa os servidores TCP ligam ao secundário	-	-	Pass
Servidor RMI secundário substitui o primário em caso de avaria longa	-	Após falhar uma quantidade definida de heartbeats o RMI secundário torna-se primário	Pass
Os dados são os mesmos em ambos os servidores RMI	-	O código de ambos os RMI's é idêntico	Pass
O failover é invisível para utilizadores	-	Utilizadores não perdem a sessão quando há uma falha do RMI	Pass
O servidor original, quando recupera, torna-se secundário	-	-	Pass