

ucDrive: Repositório de ficheiros na UC

Sistemas Distribuídos 2021/22 — Meta 1 — 26 de março de 2022 (23:59)

Resumo

O ucDrive é uma plataforma de armazenamento de ficheiros partilhados para alunos e professores da Universidade de Coimbra. Este projeto consiste em criar esta plataforma para que professores e alunos possam armazenar ficheiros num repositório remoto.

1 Objetivos do projecto

No final do projeto cada estudante deverá ter:

- programado um sistema de armazenamento de dados com arquitetura cliente-servidor;
- seguido um modelo multithreaded para desenhar os servidores;
- aplicado sockets para comunicação entre clientes e servidores;
- criado uma consola de administração usando Java RMI;
- garantido a disponibilidade da aplicação através de redundância com mecanismos de failover.

2 Visão geral

O armazenamento de dados em servidores é uma necessidade entre professores e alunos da Universidade de Coimbra para poderem armazenar ficheiros resultantes das diversas actividades académicas. Com esta necessidade em mente, o corpo docente da disciplina decidiu conceptualizar o ucDrive. Para isto, fizeram o levantamento de um conjunto de funcionalidades a desenvolver, atributos de qualidade a serem respeitados pela solução e respectiva arquitetura.

Tendo isto em conta, os alunos de Sistemas Distribuídos do DEI@UC, dotados de capacidades para levar a cargo esta tarefa, foram incumbidos de idealizar, conceptualizar e implementar a plataforma ucDrive. Para tal, estes alunos devem organizar-se em grupos de 2 ou 3 alunos (o número de alunos terá impacto nas funcionalidades a implementar), e cada grupo deve apresentar uma solução até ao dia 26 de março de 2022 (23:59).

A plataforma ucDrive deve permitir realizar todas as operações indispensáveis para que os utilizadores possam armazenar de maneira simples os seus ficheiros. Com este objectivo em mente, a plataforma deve permitir operações como: carregar um ficheiro, descarregar um ficheiro e listar ficheiros.

Como todo e qualquer software está sujeito a falhas, a plataforma ucDrive deve considerar a possibilidade de ocorrência de falhas e lidar com estas de forma adequada através de mecanismos de *failover*.

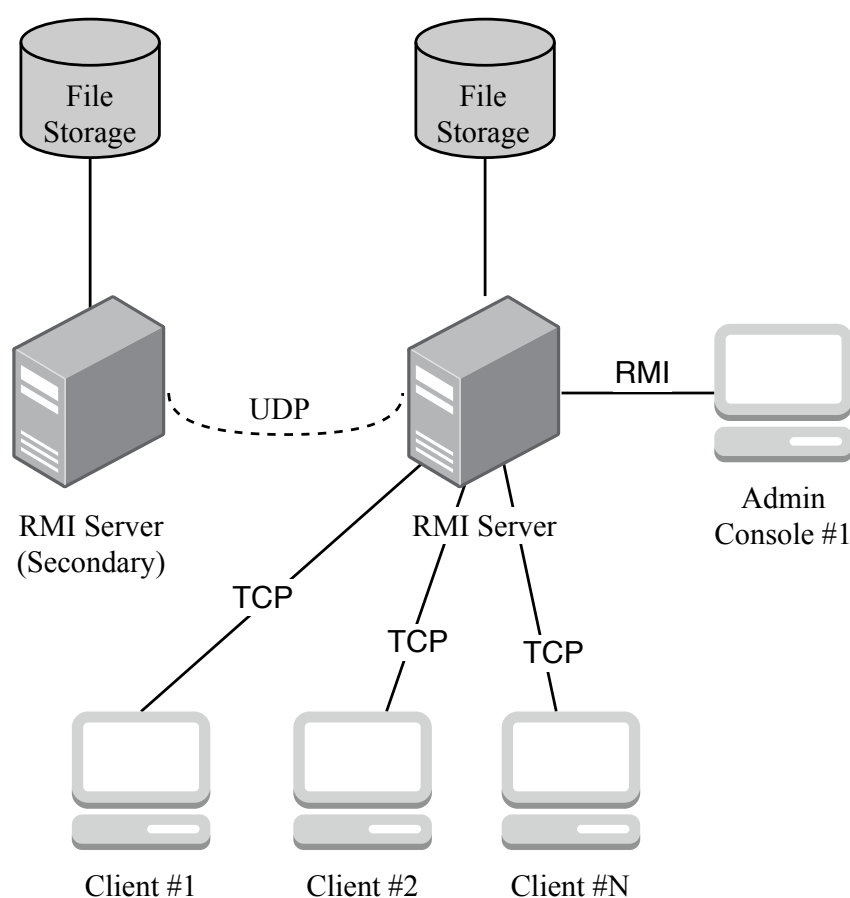


Fig. 1: Arquitetura da plataforma ucDrive.

A Figura 1 mostra a arquitetura global da plataforma ucDrive. Cada grupo deverá programar os *servidores de armazenamento* (*ucDrive Server – Primary Server e Secondary Server*), que são idênticos embora um seja inicialmente primário e o outro secundário. Cada grupo deverá também programar a *consola de administração* (*Admin Console*), à parte, que se liga ao ucDrive Server por RMI e permite ao administrador da plataforma gerir, organizar e realizar modificações ao funcionamento interno da plataforma, e a *consola de cliente* (*Client*), que se liga ao ucDrive Server por TCP e permite aos clientes da plataforma carregar, descarregar e listar ficheiros.

3 Funcionalidades a desenvolver

Os utilizadores da aplicação interagem apenas com os terminais de cliente, e os *administradores* (gestores da plataforma) utilizam a consola de administração. Deverá ser possível realizar as seguintes operações:

3.1 Consola de Administração

A consola de administração deve ser implementada apenas pelos grupos de 3 alunos. No caso de grupos de 2 alunos, as informações que são alvo destas funcionalidades devem ser geridas de forma estática com ficheiros de configuração localizados diretamente no servidor.

1. **Registar utilizador.** Deve ser possível registar utilizadores. Deverá guardar toda a informação pessoal que considere necessária, bem como uma password (código de acesso) e o departamento/faculdade ao qual o utilizador pertence. A informação pessoal deverá incluir também dados de contacto telefónico, morada, número e validade do cartão de cidadão. Ao ser registado o utilizador, deve ser configurado de imediato a directoria de base (home) para esse mesmo utilizador no servidor.
2. **Listar as directorias/ficheiros por utilizador.** Para efeitos de gestão a plataforma deve permitir listar as directorias e ficheiros por utilizador. Esta funcionalidade deve ter em conta os privilégios do utilizador que a executar. Como administrador deve ter visão total sobre o sistema de ficheiros e, para tal, deve ser-lhe permitido listar as directorias e ficheiros de todos os utilizadores.
3. **Configurar o mecanismo de failover.** Para aumentar ou reduzir a capacidade de detectar avarias entre servidores, a plataforma deve permitir aos administradores alterar os parâmetros do mecanismo de heartbeat, nomeadamente, número de *pings* perdido até se considerar o serviço indisponível e o tempo entre cada *ping* em milissegundos.
4. **Listar detalhes sobre o armazenamento.** Para evitar o consumo de recursos de armazenamento descontrolado, os administradores devem poder listar detalhes do espaço ocupado por utilizador e no total.
5. **Validar a replicação dos dados entre os vários servidores.** Para garantir a replicação dos dados, os servidores comunicam entre si de modo a trocarem informações para realizarem a replicação e consenso de toda a informação guardada. Para efeitos de manutenção e verificação, deve ser permitido aos administradores verificarem o estado de replicação dos dados entre os vários servidores. Para isto, devem poder obter um sumário dos dados que estão devidamente replicados e do estado destes, por exemplo, deve ser visível em que servidores se encontram replicados cada directoria e respectivos ficheiros.

3.2 Aplicação cliente

1. **Autenticação do utilizador no terminal de cliente.** Os utilizadores devem poder autenticar-se num terminal de cliente através de nome de utilizador e respectiva password. Após a autenticação o servidor deve colocar o utilizador na directoria Home, ou na directoria da última sessão caso este já se tenha autenticado no sistema no passado.
2. **Alterar a password do utilizador.** O utilizador deve poder alterar a sua password. Para que isto seja possível, estes utilizadores devem estar autenticados através do terminal de cliente. Após alteração, o terminal deve desconectar o cliente e pedir uma nova autenticação de modo a manter a segurança na alteração da password.
3. **Configurar endereços e portos de servidores primário e secundário** Na aplicação cliente deve ser possível configurar os endereços *IP* e *portos* dos dois ucDrive Servers, o primário e o secundário.
4. **Listar os ficheiros que existem na directoria atual do servidor.** Os utilizadores devem poder listar ficheiros existentes numa directoria do servidor. Esta funcionalidade tem como objectivo poder listar os ficheiros de uma directoria remota.
5. **Mudar a directoria atual do servidor.** Os utilizadores devem poder mudar a directoria corrente no servidor para poderem navegar entre as várias directorias a que têm acesso, estando sempre limitados às directorias dentro da sua *home*.
6. **Listar os ficheiros que existem na directoria atual do cliente.** Os utilizadores que estejam a aceder ao sistema através de um terminal de cliente devem poder listar os ficheiros existentes na directoria corrente do cliente. Esta funcionalidade tem como objectivo poder listar os ficheiros de uma directoria local.
7. **Mudar a directoria atual do cliente.** Os utilizadores devem poder mudar a directoria corrente no seu sistema de ficheiros local para poderem navegar entre as várias directorias existentes no sistema local.
8. **Descarregar um ficheiro do servidor.** Os utilizadores devem poder descarregar um ficheiro do servidor. Para que o possam fazer, devem estar autenticados como tal e devem ter permissões de leitura no ficheiro que pretendem descarregar. Ao descarregar um ficheiro do servidor, o cliente deve ficar com um cópia do ficheiro armazenado localmente. A descarga do ficheiro deve ser efetuada através de um socket independente conectado com o servidor atual mas num porto diferente.
9. **Carregar um ficheiro para o servidor.** Os utilizadores devem poder carregar um ficheiro para o servidor. Para que o possam fazer, devem estar autenticados como tal e devem ter permissões de escrita na directoria para onde pretendem carregar o ficheiro. Ao carregar um ficheiro para o servidor, o servidor deve ficar com um cópia do ficheiro armazenado na directoria. O carregamento do ficheiro deve ser efetuado através de um socket independente conectado com o servidor atual mas num porto diferente.

4 Arquitetura

A arquitetura apresentada na Figura 1 mostra que a plataforma ucDrive é composta por três aplicações, o ucDrive Server, Client e Admin Console. O sistema deverá aceitar qualquer número de clientes, que devem conter todas as funcionalidades acima identificadas. Assim, deverão ser criadas três aplicações:

- ucDrive Server – É o servidor central (replicado) da plataforma que armazena, obtém e distribui toda a informação. Este servidor comunica com: 1) Admin Console(s) através do protocolo RMI, para efeitos de administração e configuração da plataforma, 2) Client(s) via TCP, para interações dos utilizadores com a plataforma e 3) Outras réplicas de ucDrive Server através do protocolo UDP, para questões de replicação de dados e mecanismos de failover.
- Client – São os clientes que estão disponíveis para os utilizadores. Podem existir um ou mais clientes, que lhes permitem realizar as funcionalidades acima definidas. Estas consolas comunicam com o ucDrive Server apenas através do protocolo TCP.

- [REDACTED]

O servidor deve implementar dois canais de comunicação com cada cliente, um para troca de comandos e um outro para troca de ficheiros. Sempre que for necessário descarregar ou carregar um ficheiro, deve ser aberto um outro canal num *porto* diferente para tratar destas operações. Como podem suceder falhas de rede ou de hardware no ucDrive Server, deve existir um ucDrive Server de backup, pronto a substituir o outro em caso de falha. Estes servidores devem ter os dados persistidos de forma a que nunca se percam, mesmo que os processos sejam terminados. Para tal, poderá ser usado o sistema de ficheiros ou uma base de dados.

5 Atributos de qualidade

A aplicação deverá lidar corretamente com quaisquer exceções que estejam previstas. Por forma a garantir que o repositório de ficheiros está sempre disponível, deverá usar uma solução de failover para garantir que a aplicação continua a funcionar mesmo que um servidor avarie.

5.1 Tratamento de exceções

Como o hardware pode falhar, é necessário que os utilizadores não notem nenhuma falha de serviço. Como tal, no caso do servidor ucDrive falhar, é preciso garantir que os ficheiros não se percam, nem apareçam duplicados.

Também do lado dos clientes é possível que a ligação se perca a meio. É necessário garantir que nenhuma falha do lado do cliente deixe nenhuma operação a meio,

por exemplo, um download ou upload de um ficheiro, devendo ser implementado um mecanismo de *retry* quando o download não é devidamente terminado.

Os terminais de cliente podem *crashar* e deve bastar reiniciá-los para que se possa retomar as operações no mesmo estado em que se encontrava antes deste evento.

5.2 Failover

De modo a que quando o servidor ucDrive falhar, o servidor secundário o substitua, é necessário ter alguns cuidados. Em primeiro lugar o servidor secundário deve trocar periodicamente, via UDP, *pings* ou *heartbeats* com o servidor primário. Se algumas destas mensagens se perderem (por exemplo cinco pings seguidos - de acordo com a configuração definida para o servidor) o secundário assume que o primário avariou e assume essa função. Em segundo lugar, para a implementação deste mecanismo, o servidor secundário, enquanto o primário está ativo não aceita ligações dos clientes, passando a aceitar apenas quando detecta que o primário tem um problema e assume a sua função. A troca de servidor conectado é da responsabilidade do cliente, que deve passar a ligação do primário para o secundário no caso de falha. Outros mecanismos equivalentes serão aceites, devendo ser devidamente explicitados.

Finalmente, se ou quando o servidor ucDrive original recuperar, deverá tomar o papel de secundário e não de primário.

6 Relatório

Devem reservar tempo para a escrita do relatório no final do projeto. Devem escrever o relatório de modo a que um novo colega que se junte ao grupo possa perceber a solução criada, as decisões técnicas efetuadas e possa adicionar novos componentes ou modificar os que já existem. O relatório deve incluir:

- Arquitetura de software detalhadamente descrita. Deverá ser focada a estrutura de threads e sockets usadas, bem como a organização do código.
- Detalhes sobre o funcionamento do servidor ucDrive. Deverá explicar detalhadamente o funcionamento dos métodos remotos disponibilizados e eventuais callbacks usados.
- Detalhes do mecanismo de *failover*.
- Distribuição de tarefas pelos elementos do grupo.
- Descrição dos testes feitos à plataforma (tabela com descrição e pass/fail de cada teste).

7 Distribuição de tarefas

A divisão do trabalho é da responsabilidade do grupo, no entanto ressalva-se que todos os elementos do grupo devem conhecer de forma global a totalidade do trabalho e entender e discutir as decisões tomadas.

8 Entrega do projeto

O projeto deverá ser implementado em Java e entregue num arquivo ZIP. Para além dos ficheiros .java, esse arquivo deverá conter um ficheiro README.txt com toda a informação necessária para instalar e executar o projeto sem a presença dos alunos. Projetos sem este ficheiro, sem informações suficientes, que não compilem ou não executem corretamente **não serão avaliados**.

Dentro do ficheiro ZIP deverá também estar um PDF com o relatório, que deve seguir a estrutura fornecida, dado que a avaliação irá incidir sobre cada um dos pontos.

Também no ficheiro ZIP deverão existir 3 ficheiros JAR: terminal.jar (terminal cliente), ucDrive.jar (servidor) e, para o caso de grupos de 3 alunos, a console.jar (consola de administração).

Finalmente, o ficheiro ZIP deverá ter também uma pasta com o código fonte completo do projeto. A ausência deste elemento levará à **não avaliação do projeto**.

O ficheiro ZIP deverá ser entregue na plataforma inforestudante até ao dia 26 de março de 2022 (23:59), via <https://inforestudante.uc.pt>.