**Relatório sobre o Projeto de Sistemas Distribuídos em Java**

Fabrício Silva & Mauro Aldo

O projeto em questão trata-se de um sistema distribuído em Java, envolvendo servidores de aplicação, balanceadores de carga e clientes. O sistema é concebido para realizar operações de leitura e escrita em arquivos, além de calcular o Máximo Divisor Comum (MDC) de dois números.

A estrutura do projeto é organizada em quatro principais pacotes: `main`, `clientes`, `balanceadores`, e `Servidores`. Cada um desses pacotes desempenha um papel específico no funcionamento do sistema.

**Funcionalidades Principais:**

**1. Servidores de Aplicação:**

- Os servidores de aplicação (`ServidorDeAplicacao`) são criados com base em uma porta específica e um nome de arquivo associado.

- Cada servidor de aplicação espera por conexões de clientes e, ao receber uma conexão, cria uma nova thread (`ClienteHandler`) para lidar com a requisição do cliente.

**2. Clientes:**

- Os clientes (`Cliente`) são threads que enviam requisições aleatórias para o sistema, simulando operações de leitura e escrita.

- As requisições dos clientes são encaminhadas para um balanceador de carga, que decide qual servidor de aplicação será responsável por atendê-las.

**3. Balanceadores de Carga:**

- Os balanceadores de carga (`BalanceadorDeCarga`) são responsáveis por distribuir as requisições dos clientes entre os servidores disponíveis.

- Os balanceadores mantêm uma lista de servidores e utilizam um algoritmo simples de round-robin para encaminhar as requisições para os servidores.

**Fluxo de Execução:**

1. Os servidores de aplicação aguardam conexões na porta especificada.

2. Os balanceadores de carga também aguardam conexões em suas respectivas portas.

3. Os clientes enviam requisições aleatórias para os balanceadores de carga.

4. Os balanceadores encaminham as requisições para os servidores de aplicação.

5. Os servidores de aplicação processam as requisições e respondem aos clientes.

**Observações Adicionais:**

- O código utiliza threads para permitir a execução concorrente de múltiplos servidores de aplicação, balanceadores de carga e clientes.

- A comunicação entre clientes e balanceadores, bem como entre balanceadores e servidores de aplicação, é realizada via sockets.

- As operações de leitura e escrita são simuladas nos servidores de aplicação, com resultados exibidos no console.

- As exceções são tratadas principalmente com mensagens de log, sem ações específicas para lidar com essas situações.

**Potenciais Melhorias e Considerações:**

- Adição de tratamento mais robusto para exceções, como a implementação de estratégias de retry ou fallback.

- Implementação de mecanismos de segurança, como autenticação e autorização.

- Consideração de técnicas de balanceamento de carga mais avançadas.

- Exploração de opções de persistência mais eficientes para os dados do sistema.

Em resumo, o projeto demonstra uma arquitetura básica de sistemas distribuídos em Java, proporcionando uma visão geral de como os servidores de aplicação, balanceadores de carga e clientes podem interagir em um ambiente distribuído.