



Trabalho Prático (Exame Especial)

Realizado por: João Leitão

Professor: Luís Assunção

Relatório realizado no âmbito de Sistemas Distribuídos,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2017/2018

Julho de 2018

Índice

1. Introdução.....	3
2. Requisitos	4
3. Solução	5
3.1 Arquitetura	5
3.2 Implementação	6
3.3 Storage.....	6
3.4 Broker.....	6
3.5 Comunicação.....	6
3.6 Tratamento de erros.....	6
4. Conclusão	7

1. Introdução

Este trabalho teve como objetivo o desenvolvimento de um sistema distribuído usando objetos distribuídos (.NET Remoting) e Serviços WCF na plataforma .NET.

Este sistema distribuído constata o armazenamento de dados por utilizadores. De modo que este sistema se encontra distribuído em três partições: um serviço .NET Remoting (Storage), um serviço WCF (Broker) e uma aplicação cliente que expõe as funcionalidades do sistema.

A correta implementação e interligação dos componentes deste sistema referidos anteriormente, faz com que este sistema proponha as boas maneiras da programação distribuída, promovendo a tolerância a falhas, bem como o balanceamento de carga.

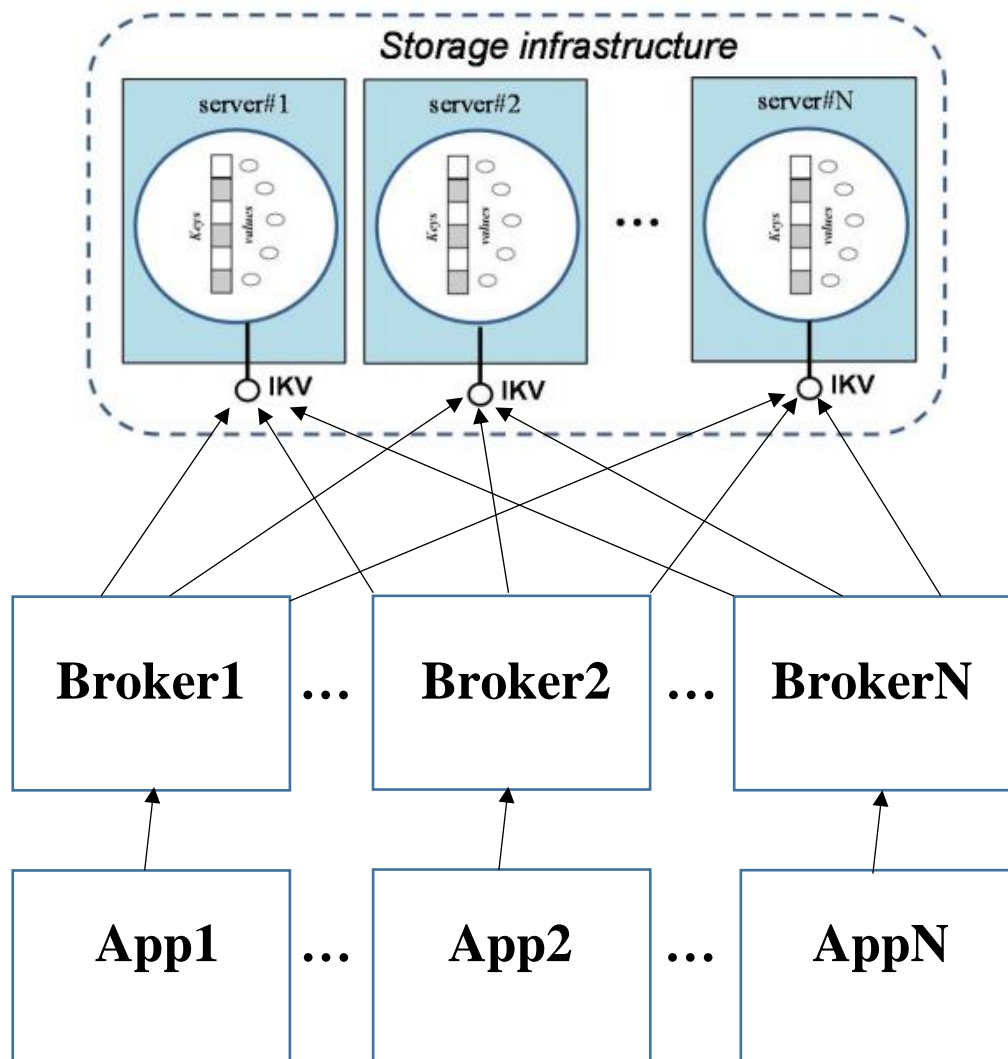
2. Requisitos

- Armazenamento de dados em memória usando o conceito de par chave-valor;
- Recolha de dados através da chave fornecida;
- Remoção de dados através da chave fornecida;
- Não são permitidas chaves repetidas num mesmo servidor;
- Os servidores executam-se em máquinas diferentes acedidas por TCP no porto 7000 ou na mesma máquina nos portos 7000, 7001, ...;
- Configuração do servidor através de ficheiros de configuração, para permissão do lançamento de múltiplas instâncias em máquinas diferentes ou na mesma máquina.
- Balanceamento de carga através da gestão do armazenamento de dados em servidores menos ocupados.

3. Solução

3.1 Arquitetura

A seguir é apresentada a arquitetura do sistema que ilustra as interligações com os componentes utilizados.



A arquitetura apresentada acima revela a possibilidade de existência de diversas instâncias dos diversos componentes. O componente Storage (servidor .NET Remoting) é o responsável pelo armazenamento dos dados neste sistema. Por forma a promover o balanceamento de carga é necessário que existam múltiplas instâncias de Storage para que não haja excesso de carregamento nos servidores. O componente Broker tem a função de criar transparência à existência de múltiplos servidores de Storage, para isso comunica diretamente com os servidores Storage de modo a executar as operações a eles referentes. O componente App é responsável por aceder ao serviço do componente Broker e ilustrar as funcionalidades do sistema descrito.

3.2 Implementação

Nesta secção são apresentados todos os aspetos inerentes à implementação dos componentes do sistema distribuído.

3.3 Storage

O objeto distribuído .NET Remoting Storage implementa uma interface que disponibiliza as funcionalidades de armazenamento, leitura e remoção de dados, que são guardados em sequências de bytes. Através do ficheiro de configuração deste objeto é possível verificar que o modo de ativação é Singleton, dado que se pretende que tenha estado entre chamadas aos métodos, também é permitido concorrência visto que as funcionalidades são thread-safe. Ao disponibilizar este objeto num canal TCP, é necessário colocar o atributo `rejectRemoteRequests` a falso, pois assim permite que haja conexão com máquinas remotas.

3.4 Broker

O componente Broker é um serviço WCF que neste caso, é cliente do componente Storage, isto é, o Broker é um proxy (intermediário) entre o Storage e a App. O Broker tem a função de disponibilizar as operações do Storage, através da implementação de um contrato a ele inerente com as mesmas funcionalidades do componente Storage. O Broker tem ainda como função, fazer o balanceamento da carga, verificando qual dos servidores Storage se encontra menos ocupado para realizar um armazenamento. No ficheiro de configuração é definido o *binding* a ser utilizado, neste caso `netTcpBinding`, devido ao protocolo de transporte ser TCP, é definido também os *endpoints* dos clientes do serviço, neste caso os *endpoints* dos diversos servidores Storage.

3.5 Comunicação

A comunicação entre o Storage e o Broker é feita através da instanciação de objetos Storage, pelos *endpoints* dos servidores Storage no ficheiro de configuração do serviço WCF Broker. A comunicação entre a aplicação App e o serviço WCF Broker é feita através da instanciação de objetos `BrokerClient` que é um *proxy* que contém os metadados referentes ao serviço WCF Broker gerado através da adição de uma referência para o serviço WCF Broker na aplicação App.

3.6 Tratamento de erros

Para existir tolerância a falhas num sistema distribuído é necessário que haja uma boa gestão dos erros para ainda assim promover uma boa utilização aos clientes.

Para tal, no serviço WCF Broker, foi implementada um contrato de falha, `BrokerExceptionFaultContract` que possibilita a propagação de erros de forma simples e com informação controlada para os clientes. Este contrato de falha é composto por um construtor que recebe uma mensagem do erro por parâmetro e um atributo `Message` com propriedades `get` e `set`.

4. Conclusão

Após a realização deste trabalho foi possível reter aspetos importantes sobre sistemas distribuídos, a sua programação e as diferenças entre as tecnologias usadas para os desenvolver, relativamente a sistemas já ultrapassados (.NET Remoting) com sistemas mais recentes (WCF).

O problema do trabalho baseava-se na utilização e integração de serviços .NET Remoting e serviços WCF, o que inicialmente se revelou um desafio, a utilização das duas tecnologias no mesmo sistema distribuído, levou a um esforço maior no estudo das duas tecnologias por forma a tirar partido das boas maneiras da programação distribuída.

É possível afirmar que a forma como os serviços WCF possibilitam uma transparência de serviços externos aos clientes e também como fazem a gestão de erros, de modo a permitir tolerância a falhas, revelou-se algo superior à tecnologia .NET Remoting.

Por fim é possível concluir que estes aspetos anteriormente referidos promovem um novo conhecimento que com certeza fará diferença em projetos semelhantes de sistemas distribuídos.