

FT-OSGi: Extensões à Plataforma OSGi para Tolerância a Faltas*

(Resumo Estendido)

Carlos Torrao, Nuno Carvalho, and Luís Rodrigues

INESC-ID/IST

`carlos.torrao@ist.utl.pt, nonius@gsd.inesc-id.pt, ler@ist.utl.pt`

1 Introdução

A plataforma OSGi [1] foi criada pela organização *OSGi Alliance* e concretiza uma especificação para a composição de serviços que possam ser instalados dinamicamente e que se invocam mutuamente. Esta plataforma foi concretizada para a linguagem Java e fornece primitivas que permitem a construção de aplicações a partir de pequenos módulos reutilizáveis, que colaboram entre si. Muitas áreas aplicacionais que usam a plataforma OSGi têm requisitos de disponibilidade e tolerância a faltas. Na área da domótica foram reportados vários problemas de fiabilidade, que se reflectem na (falta de) satisfação do utilizador final [2]. Assim sendo, é da maior importância desenhar e implementar ferramentas que forneçam tolerância a faltas neste tipo de aplicações, sem, no entanto, perder a compatibilidade entre os vários módulos.

2 Trabalho Relacionado

O núcleo da plataforma OSGi suporta a instalação e remoção, em tempo de execução, de aplicações extensíveis, a que chamamos *módulos*. Uma das principais vantagens da plataforma OSGi é a sua capacidade de instalação em tempo de execução, sendo possível instalar, actualizar, remover, iniciar e parar módulos numa instância OSGi em execução.

Do nosso conhecimento até à data, não existe nenhum trabalho que forneça suporte de tolerância a faltas para a arquitectura OSGi de uma forma genérica e completa, apesar de existirem os seguintes trabalhos nessa direcção: OSGi-based Gateway Replication [3] e Towards Reliable OSGi Framework [4].

3 FT-OSGi

Esta secção apresenta o FT-OSGi, que consiste num conjunto de extensões à plataforma de serviços OSGi. Estas extensões visam melhorar a disponibilidade e fiabilidade de aplicações OSGi, de uma forma transparente e simples.

* Este trabalho foi parcialmente suportado pela FCT através do projecto Pastramy (PTDC/EIA/72405/2006).

Os serviços são replicados em vários nós de um sistema distribuído. O cliente, quando deseja aceder a um serviço, comunica com o grupo de réplicas desse serviço e não com uma única instância. Os serviços podem ter dois modos de funcionamento relativamente ao seu estado interno: com estado ou sem estado. A manutenção de estado replicado requer suporte explícito por parte do programador do serviço. São suportados dois tipos de replicação: replicação activa e replicação passiva.

É também possível definir como são seleccionadas as respostas retornadas ao cliente do serviço replicado. No caso de replicação activa, um pedido é executado em todas as réplicas e o cliente obtém um número de respostas equivalente ao número de réplicas existentes. O intermediário do lado do cliente tem a responsabilidade de filtrar os pedidos de forma a tornar a replicação transparente. O FT-OSGi suporta três modos de selecção de respostas: **primeira-resposta**, **todas-as-respostas**, e **maioria-das-resposta**. No modo **primeira-resposta**, a primeira resposta a ser recebida é logo processada para o cliente, e todas as respostas que lhe seguem são descartadas. No modo **todas-as-respostas**, só é passada uma resposta para o cliente quando todas as respostas foram recebidas e verificadas como sendo iguais. Caso uma ou mais respostas sejam diferentes, é lançada uma excepção com essa informação. Finalmente, o modo **maioria-das-respostas** retorna a resposta maioritária, mesmo que existam respostas divergentes.

Finalmente, visto que os clientes têm uma instância OSGi local, a semântica dos eventos OSGi é mantida, fazendo chegar a todos os clientes notificações dos eventos gerados pelos serviços replicados, assim como eventos acerca do ciclo de vida destes serviços. Para mascarar a replicação, um intermediário do lado do cliente é responsável por filtrar as cópias do mesmo evento recebidas pelo cliente (usando algoritmos semelhantes aos usados para processar respostas aos pedidos).

Sendo o FT-OSGi uma extensão para a plataforma OSGi, existe uma instância OSGi em execução em todos os nós do sistema. Para além disso, o FT-OSGi combina componentes desenvolvidos por terceiros com componentes desenvolvidos pelos autores para suportar as funcionalidades acima descritas. Os componentes desenvolvidos por terceiros são o R-OSGi e o serviço de comunicação em grupo, que se descrevem de seguida.

O R-OSGi [5] consiste numa plataforma capaz de distribuir uma aplicação OSGi por vários nós de uma rede. A camada R-OSGi assenta sobre a arquitectura OSGi original de uma forma transparente para as aplicações, sendo possível colocar qualquer aplicação que assente sobre OSGi a funcionar sobre o R-OSGi. O R-OSGi baseia-se no uso de intermediários (que representam um serviço numa máquina remota) e no uso do protocolo *Service Location Protocol* (SLP) [6]. Para cada serviço publicado por um módulo OSGi num nó específico, quando é necessário num outro nó, é criado um intermediário que representa o serviço remotamente. Este intermediário simula localmente o serviço que está no nó remoto, efectuando invocações remotas através da rede. O SLP é usado para permitir a descoberta dos serviços na rede.

O FT-OSGi usa um serviço genérico de comunicação em grupo (jGCS) [7] que suporta o uso de várias concretizações. Neste caso foi concretizado usando o Appia [8]. O Appia permite a comunicação entre de elementos fora de um grupo com elementos do grupo.

Sobre os serviços acima descritos, foram desenvolvidos os seguintes componentes: *FT Service Handler* e *FT-Core* que, no seu conjunto, permitem fornecer tolerância a faltas a serviços OSGi. O *FT Service Handler* é responsável por obter e disponibilizar a informação relativa aos serviços remotos que podem ser acedidos através da instância local de OSGi. Em particular, as propriedades dos serviços registam as opções de configuração do FT-OSGi (tal como o tipo de replicação utilizada, a forma como se processam as respostas, etc). O *FT-Core* é responsável por manter a coerência das réplicas que se executam em diferentes instâncias de OSGi e por esconder dos clientes a complexidade associada a esta tarefa. O componente *FT-Core* encontra-se dividido em sub-componentes, que serão descritos de seguida. O *Appia Channel Factory* é o componente responsável pela definição do serviço de replicação. O grupo de comunicação suporta dois fluxos: a comunicação entre réplicas e a comunicação entre o cliente e o grupo de réplicas. Para cada um destes fluxos de comunicação, são criados canais Appia. O canal de comunicação entre réplicas é criado quando é registado um serviço com propriedades de tolerância a faltas numa instância de OSGi (num servidor), sendo concretizado pelo componente *Appia Channel (Server)*. O canal de comunicação do cliente para o grupo é criado quando esse cliente deseja invocar o serviço replicado e é concretizado pelo componente *Appia Channel (Client)*. O componente *Replication Mechanisms* é responsável pela gestão da replicação usada nos serviços. Este componente implementa as duas técnicas de replicação suportadas (activa e passiva), tanto para serviços com estado como para serviços sem estado. Sendo também responsável pela recuperação de faltas e pela transferência de estado a novos elementos que são adicionados ao grupo.

4 Avaliação

Para efectuar a avaliação do FT-OSGi foram utilizados testes que executam várias invocações de um método vazio, sem processamento, que recebe 4 KBytes de dados como argumento. Para estes testes, medimos os tempos de resposta, para a versão centralizada usando R-OSGi e para a versão replicada com o FT-OSGi.

Através dos resultados obtidos (ver Figura 1) foi possível extrair as seguintes conclusões. Enquanto a execução remota de um serviço não replicado só exige dois passos de comunicação (pedido e resposta), a execução de um serviço replicado exige pelo menos quatro passos de comunicação (uma vez que inclui o custo adicional da coordenação entre as réplicas) e, no caso da replicação passiva, a transferência de estado entre a réplica primária e as réplicas secundárias, o que justifica as diferenças de desempenho observadas. Pode também observar-se que quanto maior for o número de réplicas, maior é o custo no desempenho.

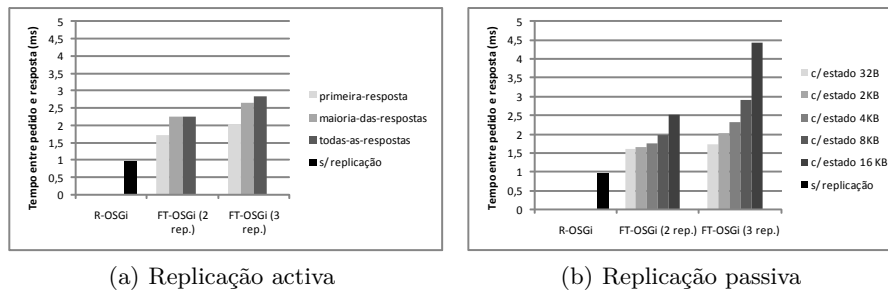


Figura 1. Custo adicional da replicação.

5 Conclusões

Este artigo descreve e avalia o FT-OSGi, um conjunto de extensões à plataforma OSGi orientada a fornecer tolerância a faltas. No FT-OSGi, cada serviço pode ser configurado para usar replicação passiva e activa, assim como propriedades de QoS. Apesar da replicação induzir um custo adicional na invocação dos serviços, para números reduzidos de réplicas os valores são aceitáveis para muitas aplicações.

Agradecimentos: Os autores agradecem ao João Leitão pelos comentários feitos a versões preliminares deste artigo.

Referências

- OSGi Alliance: Osgi service platform core specification (April 2007) <http://www.osgi.org/Download/Release4V41>.
- Kaila, L., Mikkonen, J., Vainio, A.M., Vanhala, J.: The ehome - a practical smart home implementation. In: Proceedings of the workshop Pervasive Computing @ Home, Sydney, Australia (May 2008)
- Thomsen, J.: Osgi-based gateway replication. Proceedings of the IADIS Applied Computing Conference 2006 (2006) 123–129
- Ahn, H., Oh, H., Sung, C.: Towards reliable osgi framework and applications. Proceedings of the 2006 ACM symposium on Applied computing (2006) 1456–1461
- Rellermeyer, J., Alonso, G., Roscoe, T.: R-osgi: Distributed applications through software modularization. Middleware (2007) 1–20
- Guttman, E.: Service location protocol: automatic discovery of ip network services. Internet Computing, IEEE **3**(4) (Jul/Aug 1999) 71–80
- Carvalho, N., Pereira, J., Rodrigues, L.: Towards a generic group communication service. In: Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA), Montpellier, France (Oct 2006)
- Miranda, H., Pinto, A., Rodrigues, L.: Appia, a flexible protocol kernel supporting multiple coordinated channels. In: Proceedings of the 21st International Conference on Distributed Computing Systems, Phoenix, Arizona, IEEE (Apr 2001) 707–710