

Testando uma Infra-estrutura FT-CORBA: O Caso GROUPPAC *

Alexandre Schulter¹, Alysson Neves Bessani¹, Eduardo A. P. Alchieri¹,
Joni da Silva Fraga¹, Lau Cheuk Lung²,

{schulter, neves, alchieri, fraga}@das.ufsc.br, lau@ppgia.pucpr.br

¹DAS - Departamento de Automação e Sistemas
UFSC - Universidade Federal de Santa Catarina
Campus Universitário, Caixa Postal 476 - CEP 88040-900 - Trindade - Florianópolis - SC

²PPGIA - Programa de Pós-Graduação em Informática Aplicada
PUC-PR - Pontifícia Universidade Católica do Paraná
R. Imaculada Conceição, 1155 - Prado velho - CEP 80215-901 - Curitiba -PR

Abstract. *This paper presents a case-study about the application of unit tests in the GROUPPAC fault tolerance infrastructure. Our approach is pragmatic: given the system's dimensions and the current state of his development we opt for apply techniques and tools widely used, testing only the services' interfaces that compose the infrastructure. This approach allowed us to improve the overall quality of the system.*

Resumo. *Este artigo apresenta um estudo de caso sobre a aplicação de testes de unidade na infra-estrutura para tolerância a faltas GROUPPAC. Nossa abordagem é pragmática: dadas as dimensões do sistema e o estágio atual de seu desenvolvimento optamos por aplicar técnicas e ferramentas largamente utilizadas testando apenas as interfaces dos serviços que compõem a infra-estrutura. Esta abordagem nos permitiu melhorar a qualidade do sistema em geral.*

1. Introdução

A tolerância a faltas (TF) é um requisito de grande importância em aplicações críticas distribuídas. Em geral, esta qualidade de serviço é implementada através da utilização de algum suporte que implemente os protocolos e mecanismos necessários para que uma aplicação mantenha-se ativa mesmo com a falha de alguns de seus componentes. Estes suportes são comumente implementados em nível de *middleware*, fornecendo serviços básicos de TF à aplicação. Um requisito fundamental de qualquer suporte de TF é a confiabilidade e correção: se a aplicação deve tolerar faltas a infra-estrutura não pode conter erros que levem a falhas.

Dentre os padrões de *middleware* atualmente em uso, o CORBA (*Common Object Request Broker Architecture*) se destaca pela sua maturidade, sendo aceito nos mais diversos segmentos. Além dos mecanismos básicos de comunicação, interoperabilidade e serviços, este padrão contempla uma arquitetura para objetos distribuídos tolerantes a faltas (de parada), chamado FT-CORBA (*Fault-Tolerant CORBA*) [Object Management Group, 2002]. Este suporte é definido a partir de um conjunto de objetos de serviço usados no gerenciamento de objetos de aplicação que são replicados usando abstrações de grupos.

O GROUPPAC é um suporte para tolerância a faltas que implementa o padrão FT-CORBA [Lung et al., 2001]. Este suporte vem sendo desenvolvido no LCMI/DAS/UFSC desde 1999 e atualmente se encontra em sua terceira versão [Bessani et al., 2004].

Como em qualquer outro software concebido para ser utilizado em projetos reais, espera-se que o mesmo funcione corretamente. Este requisito se acentua se levamos em consideração o

*Realizado com recursos do CNPq através do edital Software Livre, projeto GroupPac/MJaco (número 401802/2003-5).

ambiente para o qual o GROUPPAC foi desenvolvido: *software* crítico. Desta forma, uma melhora da qualidade do sistema pode ser obtida através da aplicação de testes de software.

O processo de teste é uma atividade básica do ciclo de desenvolvimento de software. Durante este processo um programa é executado (exercitado) com intenção de encontrar erros [Fairley, 1985]. O processo de testes não demonstra que um programa está correto, mas mostra seu comportamento considerando um conjunto de entradas pré-definidas (casos de teste). Se os casos de testes considerados forem de boa qualidade é possível ter uma boa estimativa da qualidade de um sistema e encontrar uma série de possíveis erros em seu código.

Este artigo apresenta uma breve descrição do processo de testes empregado para validar a implementação da infra-estrutura GROUPPAC. Este processo baseou-se na utilização de técnicas simples e largamente aceitas como testes de unidade [Beck and Gamma, 1998] apoiado por ferramentas. As próximas seções do artigo apresentam uma breve descrição da arquitetura do GROUPPAC (seção 2), a abordagem utilizada para testes e seus resultados (seção 3) e algumas considerações finais sobre o trabalho (seção 4).

2. GROUPPAC

O GROUPPAC [Lung et al., 2001] é uma implementação completa do padrão FT-CORBA desenvolvida inicialmente visando dar suporte a sistemas de larga escala. O objetivo básico desta infra-estrutura de TF é fornecer um conjunto de serviços úteis na construção de aplicações tolerantes a falhas. Sua arquitetura é composta por uma série de serviços, dentre os quais destacam-se:

- **Serviço de Gerenciamento de Replicação - SGR** Este serviço é responsável pelo ciclo de vida dos grupos de réplicas. Duas funcionalidades básicas são oferecidas pelo mesmo: **Gerenciamento de Propriedades**, onde as propriedades da replicação (tipo de replicação usada, número mínimo de réplicas, etc.) são definidas e alteradas; e o **Gerenciamento de Grupos**, que oferece mecanismos para o controle da criação e destruição de grupos, bem como de seus membros (réplicas). Sendo assim, responsável pela manutenção da filiação (*membership*) dos grupos gerenciados.
- **Serviço de Gerenciamento de Falhas - SGF** Envolve um conjunto de objetos que cooperam na detecção, notificação, análise e diagnóstico de falhas. Quando uma falha é detectada esta é repassada ao SGR para que este execute as ações necessárias e mantenha uma lista de membros sempre atualizado dos grupos.
- **Serviço de Gerenciamento de Recuperação e Logging - SRL:** Este serviço define mecanismos para a recuperação de réplicas e do próprio grupo, bem como da configuração de estados em membros que entram no grupo após o serviço replicado já estar em funcionamento. Entre estes mecanismos está um suporte para construção de *logs* usados no armazenamento de requisições enviadas ao grupo e um mecanismo para atualização de membros através de *checkpoints*.

Além destes serviços básicos, outros serviços como o de transferência de estados e o de comunicação de grupo, entre outros, também fazem parte da arquitetura do sistema. Entretanto, conforme será visto a seguir, os testes realizados se basearam diretamente no SGR, SGF e SRL.

3. Testes

Existe uma infinidade de abordagens e ferramentas desenvolvidas visando melhorar o processo de teste de software, grande parte destas são de cunho acadêmico e/ou experimental. Desta forma, nesse trabalho, optou-se por uma abordagem pragmática: dadas as dimensões do sistema e o estágio avançado de seu desenvolvimento optamos por testes que exercitem apenas as interfaces dos serviços que compõem a infra-estrutura verificando se os resultados obtidos estão de acordo com o esperado segundo as especificação do FT-CORBA.

Dentro desta abordagem essencialmente prática, optou-se pelo uso da ferramenta JUNIT [JUnit staff, 2005]: um *framework* para criação de testes de unidade [Beck and Gamma, 1998]. No entanto, os testes escritos caracterizam-se como testes de sistema, pois cada teste inicializa

a maioria dos serviços do GROUPPAC de forma integrada e atua como um usuário da infraestrutura executando diferentes casos de uso (chamando os métodos das interfaces) com entradas (parâmetros) corretas ou não. Estes testes estão divididos em quatro grupos, apresentados a seguir.

3.1. Fábrica Genérica

Uma fábrica genérica é um objeto utilizado pelo SGR para criação e remoção de réplicas (objetos) localmente nos *hosts* do sistema. Estes testes utilizam uma fábrica diretamente, sem interferência do Gerenciador de Replicação e verificam a criação de objetos CORBA passando parâmetros corretos e incorretos. No caso de parâmetros corretos, é verificado se o objeto foi realmente criado e o identificador retornado é correto. No caso de parâmetros incorretos, os parâmetros são passados incorretamente de diversas formas e o comportamento esperado é o lançamento da exceção apropriada. No teste de remoção é verificado se o objeto é realmente removido.

3.2. Serviço de Gerenciamento de Replicação - SGR

A suíte de testes do Serviço de Gerenciamento de Replicação consiste dos testes do Gerenciamento de Propriedades e do Gerenciamento de Grupos.

A cada grupo de objetos é associado um conjunto de propriedades e o SGR possibilita que estas sejam definidas por padrão ou pelo tipo do objeto, no momento da criação do grupo ou dinamicamente (com restrições). Os testes do Gerenciamento de Propriedades verificam o funcionamento dos métodos para manipulação dos conjuntos de propriedades, determinando se propriedades definidas por padrão ou por tipo são realmente atribuídas e se as devidas exceções são lançadas caso propriedades não suportadas ou com valores incorretos sejam passadas ao gerenciador. Certas propriedades como o estilo de replicação e o número inicial de réplicas não podem ser definidas dinamicamente.

O controle de filiação dos grupos de objetos pode ser feito pela infra-estrutura ou pela aplicação. No primeiro caso, o gerenciador de replicação invoca fábricas, nas localizações apropriadas, para criar os membros do grupo tanto inicialmente, para satisfazer a propriedade “número inicial de réplicas”, quanto depois da perda de um membro devido a uma falha detectada, visando satisfazer a propriedade “número mínimo de réplicas”. No segundo caso, a interface de gerenciamento de grupos do SGR permite que a aplicação crie um membro de um grupo, adicione um membro já existente ou remova, citando a localização do membro criado, adicionado ou removido. Desta forma, foram criados dois testes distintos que juntos verificam exaustivamente o funcionamento correto dos métodos da interface de Gerenciamento de Grupos.

3.3. Serviço de Gerenciamento de Falhas - SGF

A especificação FT-CORBA define interfaces para a comunicação dos detectores com os objetos monitorados e uma interface para a comunicação com o notificador. A especificação, no entanto, não define a quantidade e o arranjo dos detectores, os filtros do notificador, nem o funcionamento dos analisadores. Os testes do SGF não criam detectores específicos, e sim testam o funcionamento correto dos detectores implementados no GROUPPAC, que são detectores *pull* em 2 níveis (*host* e objeto) e um notificador com filtro simples que elimina relatórios duplicados [Lung et al., 2001]. Cada *host* possui um detector em nível de objeto, detectando falhas individuais de objetos. Todos os membros do grupo de detectores em nível de *host* detectam falhas de cada um dos detectores em nível de objeto, além de detectarem falhas de membros do seu próprio grupo.

Sendo assim, os testes do SGF registram-se no notificador e injetam falhas de parada em *hosts* e objetos para verificar se os detectores enviam, após determinado tempo, os relatórios no formato e com as informações corretas, como o tipo de falha (*host* ou objeto) e sua origem. Quando uma falha de *host* é detectada, também é verificado se o detector em nível de *host* é retirado de seu grupo, o que é responsabilidade do SGR. Os testes também consideram o fato de que não devem ser enviados relatórios de falhas de objetos de um grupo que tem a propriedade de estilo de monitoramento “não monitorado”.

3.4. Serviço de Gerenciamento de Recuperação e Logging - SRL

O SRL não tem interfaces definidas pois a aplicação nunca interage diretamente com ele. As suas responsabilidades de recuperação de réplicas faltosas a partir de estados armazenados e requisições interceptadas (*logging*) dependem do estilo de replicação usado.

No estilo de replicação passiva, o SRL deve substituir um membro primário falho por um membro reserva correto. Para verificar se isto ocorre corretamente, os testes injetam falhas nas réplicas primárias e verificam se os estados das réplicas reservas (substitutas) são consistentes com os estados das réplicas primárias faltosas no momento em que as reservas assumem o lugar das faltosas. Outra funcionalidade testada é envio de *checkpoints* periódicos as réplicas.

Em um grupo com replicação ativa, somente é necessário testar a atualização feita pelo SRL do estado de uma réplica que acaba que se juntar ao grupo, uma vez que as características deste tipo de replicação asseguram a consistência do estado das réplicas.

3.5. Considerações sobre os Testes

O GROUPPAC é um sistema relativamente grande, com aproximadamente **39000** linhas de código (Java e IDL) distribuídas por mais de **120** classes (sem considerar os arquivos gerados pela tradução de interfaces em IDL e os testes). Levando este fato em consideração, os casos de testes utilizados se baseiam em grande parte nos comportamentos especificados para os serviços definidos no documento do FT-CORBA (cap. 23 de [Object Management Group, 2002]).

O processo todo foi automatizado através da ferramenta ANT [Apache Software Foundation, 2005] utilizando as extensões para a ferramenta de testes JUNIT [JUnit staff, 2005] juntamente com a ferramenta de avaliação de cobertura de testes COBERTURA [Doliner and Thomerson, 2005].

Ao todo são **19** testes que alcançaram uma taxa de cobertura de código de **78%**. É um bom resultado, haja vista a abordagem simplificada utilizada e a existência de trechos de código descontinuados (e não mais utilizados) do GROUPPAC (como as extensões de larga escala).

De fato, a aplicação dos testes mostrou-se efetiva no sentido que diversos erros foram detectados no código, e posteriormente corrigidos. Uma grande vantagem da abordagem usada é a capacidade de executar os testes sempre que alterações forem feitas no sistema, verificando desta forma sua correção em relação aos casos de uso definidos. E já que os testes atuam como um usuário da infra-estrutura, acessando apenas as interfaces do FT-CORBA, podem ser aplicados em qualquer outra infra-estrutura que siga a mesma especificação.

4. Considerações Finais

A aplicação de testes em *middleware* TF é uma atividade essencial para garantir a confiabilidade deste importante componente das aplicações críticas distribuídas. Este artigo relata a estratégia de testes utilizada no desenvolvimento do sistema GROUPPAC. A estratégia implementada é pragmática e se baseia na automação dos testes através de ferramentas largamente utilizadas pela comunidade Java. Esta estratégia está longe da perfeição, porém foi eficiente no sentido de que permitiu uma boa cobertura do código do sistema e identificou vários erros antes não percebidos.

Referências

- Apache Software Foundation (2005). Ant homepage. <http://ant.apache.org>.
- Beck, K. and Gamma, E. (1998). Test-infected: Programmers love writing tests. *Java Report*, 3(7).
- Bessani, A. N., Alchieri, E. A. P., Lung, L. C., and da Silva Fraga, J. (2004). GroupPac 3: Estendendo o FT-CORBA para Gerenciamento e Replicação Ativa. In *Anais do WTF 2004*, Gramado, RS.
- Doliner, M. and Thomerson, J. (2005). Cobertura homepage. <http://cobertura.sourceforge.net>.
- Fairley, R. (1985). *Software engineering concepts*. McGraw-Hill, New York, NY.
- JUnit staff (2005). JUnit homepage. <http://www.junit.org>.
- Lung, L. C., da Silva Fraga, J., Padilha, R., and Souza, L. (2001). Adaptando as especificações FT-CORBA para redes de larga escala. In *Anais do SBRC 2001*, Florianópolis, SC.
- Object Management Group (2002). The Common Object Request Broker Architecture: Core Specification v3.0. OMG Standart formal/02-12-06.