

Uma Implementação do Detector de Falhas do FT-CORBA

Luelson Marlos Nunes
Elias Procópio Duarte Jr.

Universidade Federal do Paraná
Departamento De Informática
Caixa Postal 19081 Curitiba 81531-990 PR Brasil
E-Mail: {lnunes,elias}@inf.ufpr.br

Resumo

Neste trabalho apresentamos uma abordagem de implementação do detector de falhas do FT-CORBA (*Fault Tolerant-Common Object Request Broker Architecture*), utilizando a linguagem de programação Java integrada e o serviço COSNaming de uma implementação aberta do padrão CORBA. A abordagem utilizada neste trabalho para a implementação do detector de falhas propõe a monitoração periódica exclusivamente da réplica primária dos servidores. A monitoração das demais réplicas é executada apenas em caso de falha da réplica primária. Neste caso um novo servidor primário é eleito entre as réplicas sem falhas. Também é apresentado um exemplo de construção de uma aplicação tolerante a falhas utilizando o detector implementado.

Palavras-Chave: Sistemas Distribuídos, Tolerância a Falhas, Replicação, CORBA, Detecção de Falhas.

1 O Detector de Falhas do FT-CORBA

O padrão CORBA define um dos modelos de computação para objetos distribuídos mais amplamente utilizados [1]. O CORBA é a especificação de uma arquitetura e interface que permite que aplicações efetuem requisições a objetos de forma transparente e independente da linguagem de programação, sistema operacional ou mesmo da localização dos objetos[4].

O padrão para tolerância a falhas FT-CORBA [2] tem por objetivo fornecer um suporte para aplicações que necessitam de garantias de disponibilidade. O foco deste trabalho é a implementação de um dos componentes deste padrão: o Detector de Falhas. Também foi implementada uma versão simplificada do Gerenciador de Replicação.

O Detector de Falhas é responsável pela percepção da presença de falhas no sistema e geração de um relatório de falhas para o Notificador de Falhas. O Notificador de Falhas efetua uma seleção sobre este relatório visando eliminar informações desnecessárias ou duplicadas enviando então um evento de notificação para os consumidores deste serviço, como o próprio Gerenciador de Replicação [2].

Um exemplo de interação cliente/servidor tolerante a falhas baseada no FT-CORBA é ilustrada na figura 1. Um servidor de hora é utilizado no exemplo. O componente *Client* representa um cliente que deseja utilizar o servidor. O componente *COS-Naming* representa o serviço de nomes do CORBA [3] onde os grupos e os objetos replicados são registrados. O componente *ReplicationManager*, neste caso, representa o objeto responsável por iniciar o processo de recuperação da referência da réplica do servidor primário do grupo de objetos solicitado pelo cliente. O componente *PrimaryTimeServer* representa o servidor primário do grupo de objetos. O componente *FaultDetectorImpl* representa a classe responsável pela monitoração dos servidores primários dos grupos de objetos. O componente *Monitor* representa a *thread* iniciada para monitoração de cada servidor primário dos grupos de objetos existentes.

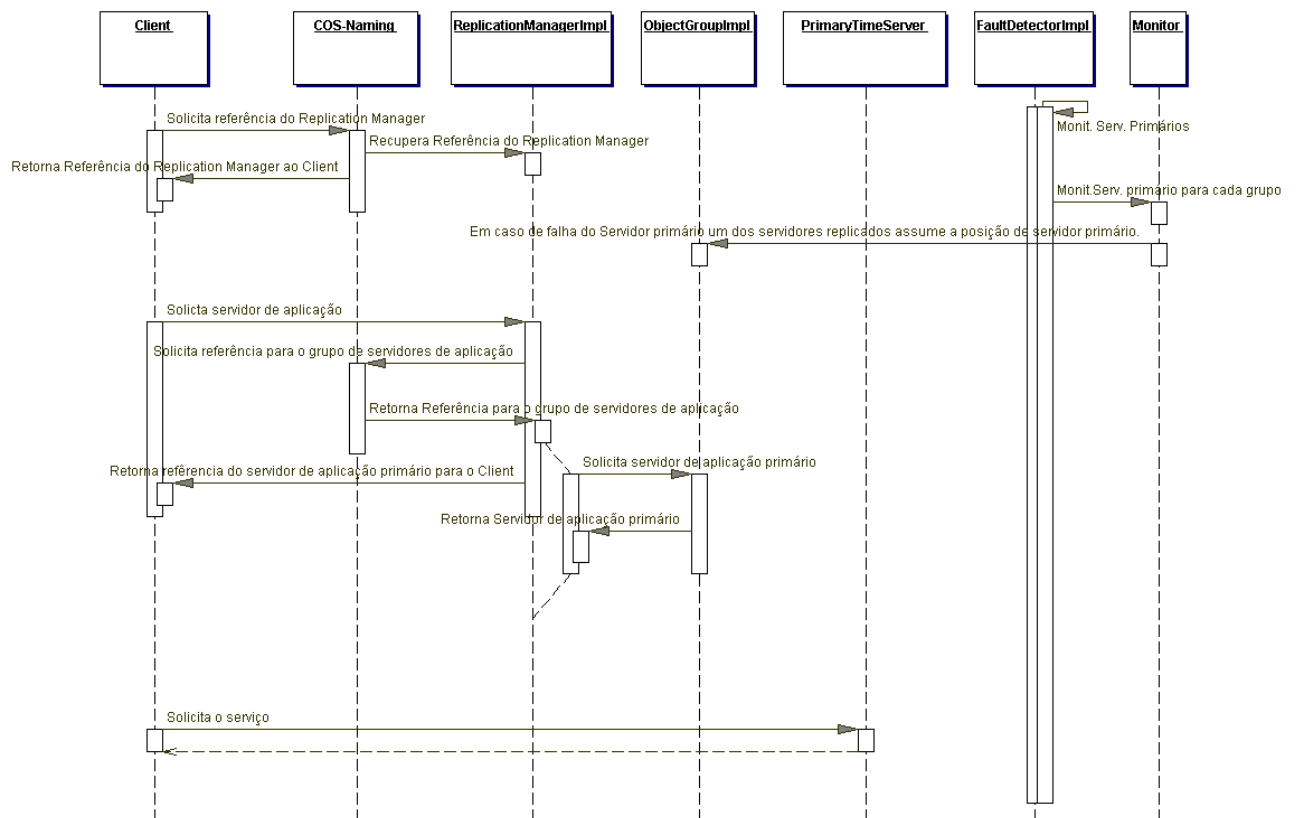


Figura 1: Interação cliente/servidor tolerante a falhas.

O processo apresentado na figura 1 ilustra a requisição do *Client* ao *COS-Naming* para obtenção da referência ao *ReplicationManager*. A partir desta recuperação o cliente solicita ao *ReplicationManager* a referência do servidor de hora pelo nome do grupo de servidores. O *ReplicationManager* por sua vez efetua uma consulta ao *COS-Naming* buscando a referência para o grupo de objetos desejado, que é representado na figura 1 pelo componente *ObjectGroupImpl*. Ao obter a referência para o grupo de objetos, este retorna a referência para o servidor primário do grupo desejado que é então retornado para o *ReplicationManager* e, por sua vez, retornado para o *Client* que pode então efetuar a solicitação desejada.

O servidor primário de cada grupo é periodicamente monitorado através do componente representado na figura 1 por *FaultDetectorImpl*. O intervalo de monitoração é configurável. O componente *FaultDetectorImpl* cria uma *thread* para cada grupo de objetos, estas *threads* são representadas na figura 1 pelo componente *Monitor*.

O processo de monitoração consiste na obtenção da referência apenas do servidor primário de cada grupo, checando se o mesmo está funcional. A monitoração não é feita para todas as réplicas, o que oferece um ganho potencial de performance se considerarmos um grande número de réplicas. Por outro lado, caso múltiplas réplicas falhem, o intervalo de tempo até que um novo servidor seja eleito pode ser consideravelmente elevado. No processo de monitoração implementado, somente quando o servidor primário não está ativo, a monitoração das réplicas é efetuada elegendo-se então um novo servidor primário para o grupo de objetos em questão e configurando esta nova réplica como servidor primário deste grupo conforme representado na figura 1 pela interação entre o componente *Monitor* e o componente *ObjectGroupImpl*.

2 Integração de uma Aplicação com o Detector de Falhas

A integração ao Detector de Falhas de uma aplicação exemplo que consiste em um sistema cliente-servidor de hora é descrita a seguir. Para que o servidor de hora *TimeServerImpl*, representado na figura 2 no canto superior direito, possa ser monitorado é necessário que ele implemente duas interfaces. A primeira interface é chamada de *OrbInterface* e define o método *initOrb*. A partir desta implementação é possível disponibilizar o servidor via ORB. A segunda interface é chamada de *PullMonitorableOperations* e define a implementação do método *isAlive()*. Este método é utilizado teste do objeto replicado e é invocado periodicamente pelo monitor do detector de falhas para monitorar se o servidor primário do grupo de objetos está falho ou sem-falhas.

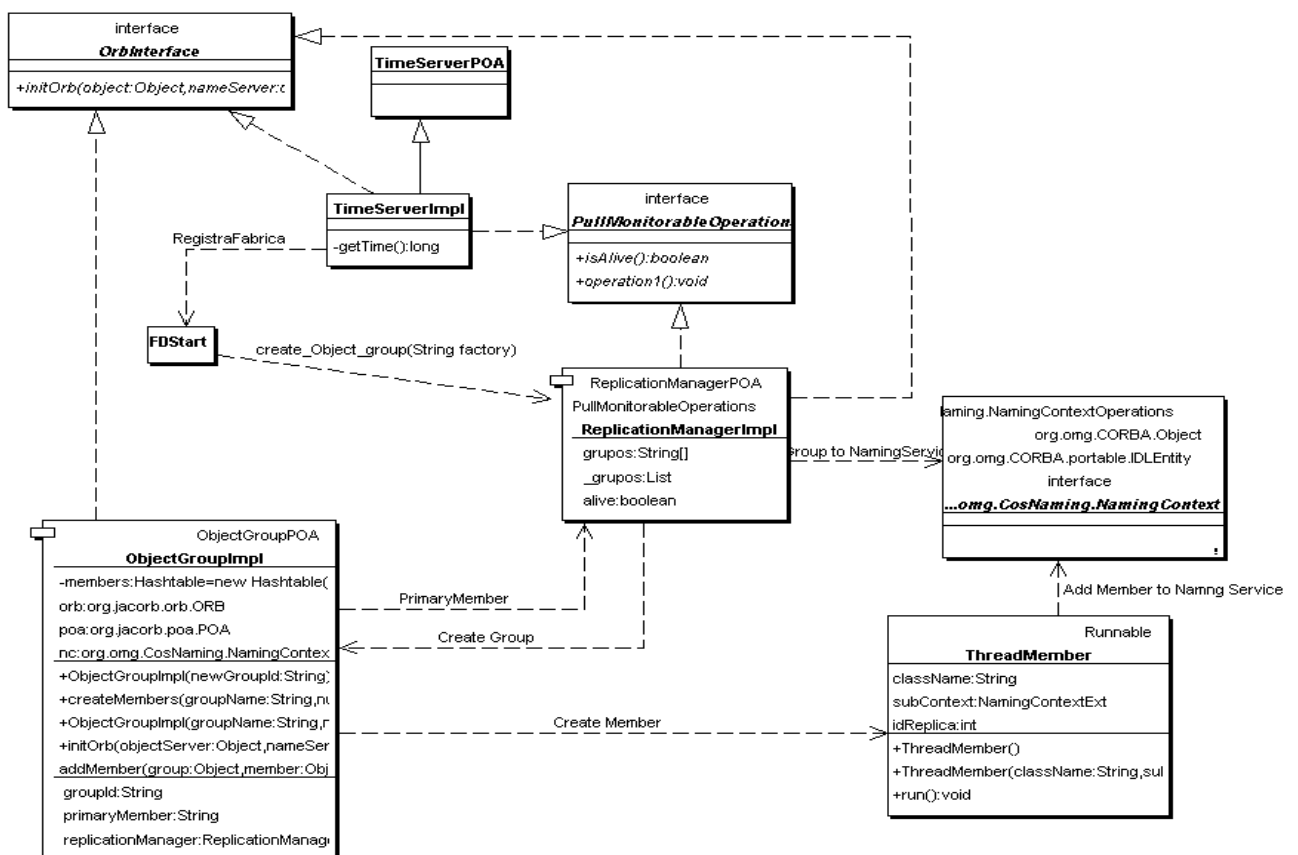


Figura 2: Integração de uma aplicação ao Detector de Falhas.

Qualquer aplicação servidora monitorada pelo detector de falhas irá utilizar os componentes *FDStart* e *ObjectGroupImpl*. A partir do componente *FDStart* a aplicação servidora faz o registro do seu nome. A partir deste nome é possível que seja efetuada a instanciação dinâmica das réplicas do servidor pelo componente *ObjectGroupImpl* que é o responsável pela criação e disponibilização das réplicas dos grupos de objetos a serem monitorados.

A figura 2 apresenta a integração de uma aplicação com a arquitetura de detecção de falhas. O início do processo de criação de grupos de objetos replicados é feito pelo *FDStart* que aciona o *ReplicationManagerImpl*, o qual registra o novo grupo no serviço de nomes CORBA e invoca o *ObjectGroupImpl* que por sua vez inicia o processo de disponibilização dos servidores replicados.

O componente responsável pela criação das réplicas dos grupos a serem monitorados é representado na figura 2 pelo componente *ObjectGroupImpl*. Este componente realiza a instanciação dinâmica das réplicas do servidor e disponibiliza os servidores em diferentes *threads* representadas na figura 2 pelo componente *ThreadMember*. O número de réplicas é configurável por host e a existência de réplicas em diferentes hosts é possível.

3 Considerações Finais

A abordagem utilizada neste trabalho para a implementação do Detector de Falhas propõe a monitoração periódica de atividade apenas do servidor primário, iniciando o processo de monitoração para as demais réplicas somente em caso de falha deste servidor. Neste caso o servidor primário é substituído por um novo servidor replicado ativo, isto aumenta o desempenho do sistema pois elimina parte do *overhead* de monitoração de múltiplas réplicas. Por outro lado, caso múltiplas réplicas falhem, o intervalo de tempo até que um novo servidor seja eleito pode ser consideravelmente elevado.

Uma avaliação quantitativa da utilização da estratégia proposta está prevista como trabalho futuro. Trabalhos futuros incluem também, além da implementação do Notificador de Falhas, a implementação da Fábrica Genérica de objetos replicados.

Referências Bibliográficas

- [1] A. Pope, *The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture*, Addison-Wesley, 1997.
- [2] Object Management Group, "Fault Tolerant CORBA Specification V1.0," *OMG Document ptc/2000-04-04 ed.*, 2000. Disponível em URL: <ftp://ftp.omg.org/pub/docs/ptc/00-04-04.pdf>
- [3] Object Management Group - OMG Home. Disponível em URL: <http://www.omg.org/>
- [4] Object Management Group, "A discussion of the Object Management Architecture," *OMG Document 00-06-41 ed.*, 1997. Disponível em URL: <ftp://ftp.omg.org/pub/docs/formal/00-06-41.pdf>