

# Implementação de Replicação Ativa no CORBA a Partir da Integração das Especificações FT-CORBA e UMIOP

Eduardo Adilio Pelinson Alchieri\*,  
Alysson Neves Bessani<sup>†</sup>, Joni da Silva Fraga<sup>‡</sup>

<sup>1</sup>DAS - Departamento de Automação e Sistemas  
UFSC - Universidade Federal de Santa Catarina  
Campus Universitário, C.P. 476 - CEP 88040-900 - Trindade - Florianópolis - SC

{alchieri, neves, fraga}@das.ufsc.br

**Abstract.** *This paper presents an integration model for the FT-CORBA and UMIOP specifications in a single middleware platform. This model defines a large spectrum of middleware support for distributed objects group communication that is used to develop an active replication mechanism for FT-CORBA. The way to integrate group communication tools in the GROUPPAC infrastructure (our FT-CORBA implementation) is discussed and some performance results are presented together with related experiences described in the literature.*

**Resumo.** *Este artigo apresenta um modelo de integração das especificações FT-CORBA e UMIOP em uma plataforma única de middleware, resultando em um amplo espectro de primitivas de comunicação que sustentam a replicação ativa do padrão FT-CORBA. Também é discutida a forma de integração de ferramentas de comunicação de grupo na infraestrutura do GROUPPAC (nossa implementação do FT-CORBA) e alguns resultados de desempenho são apresentados juntamente com experiências similares da literatura.*

## 1. Introdução

O padrão FT-CORBA (*Fault-Tolerant CORBA*) surgiu de um esforço da OMG (*Object Management Group*) no sentido de especificar uma arquitetura para objetos distribuídos tolerantes a falhas [Object Management Group, 2002]. Este padrão introduz suporte à tolerância a falhas na arquitetura CORBA (*Common Object Request Broker Architecture*) fazendo uso de técnicas de replicação de objetos. Este suporte é construído a partir de um conjunto de objetos de serviço usados no gerenciamento de objetos de aplicação replicados. A tolerância a falhas fornecida usando os serviços definidos pelo padrão está fundamentada em premissas de falhas de parada (*crash*).

A especificação FT-CORBA define quatro tipos de replicação para os grupos de objetos: **Sem Estado** (estilo de replicação em que o estado dos objetos não é alterado pelas requisições); **Passiva Fria** (replicação com apenas um objeto ou réplica primária que grava

---

\*Bolsista PIBIC/CNPq.

<sup>†</sup>Co-orientador - Bolsista PGI/CNPq.

<sup>‡</sup>Orientador - Bolsista de Produtividade em Pesquisa do CNPq - Nível 2.

periodicamente informações de estado em meio persistente); **Passiva Morna** (funciona de maneira semelhante ao anterior com a diferença de que os *checkpoints* da réplica primária são difundidos entre as réplicas secundárias); e, por fim, a **Replicação Ativa** (onde todas as réplicas executam as requisições do cliente).

O único tipo de replicação que é previsto e não tem seus mecanismos necessários especificados é a replicação ativa. A especificação atual do FT-CORBA permite que as abstrações definidas para a gestão de replicações possam ser também usadas com réplicas ativas, desde que ferramentas de comunicação de grupo proprietárias (que não são baseadas em padrões abertos) forneçam os protocolos de comunicação necessários. Assim, as requisições de método são difundidas, através de difusão atômica [Hadzilacos and Toueg, 1994, Defago et al., 2000] a todos os objetos membros do grupo (réplicas ativas) usando meios externos ao ORB. Neste caso, se considerarmos as necessidades de suporte de uma replicação ativa, do ponto de vista conceitual, diríamos que o FT-CORBA fornece os mecanismos para controle de *membership* do grupo formado pelas réplicas ativas e a ferramenta proprietária fornece o suporte para comunicação de grupo.

A arquitetura CORBA tem no ORB (*Object Request Broker*), seu principal componente, a implementação das semânticas de comunicação. As mensagens geradas nas comunicações seguem uma sintaxe de transferência própria: o GIOP (*General Inter ORB Protocol*). Esta sintaxe torna as mensagens envolvidas nas comunicações independentes das implementações de ORBs e das conseqüências de um ambiente heterogêneo. O mapeamento do GIOP sobre o TCP/IP é concretizado através do IIOP (*Internet Inter ORB Protocol*). O IIOP, seguindo as características do TCP, foi definido para comunicações ponto a ponto confiável.

Muitas aplicações distribuídas dependem de abstrações de grupos de objetos ou da necessidade da disseminação de dados entre vários *hosts* da rede. Estas aplicações demandam um melhor aproveitamento dos serviços da rede. Assim, em 1999 a OMG publicou um RFP (*Request For Proposal*) onde definia uma série de requisitos para um serviço de difusão multiponto não confiável que deveria ser suportado pelo *multicast* IP, visto que este compreende um conjunto de extensões ao protocolo IP que o habilita na concretização deste tipo de comunicação [Deering, 1986]. Em 2002, como conseqüência de esforços de padronização, surgiram as especificações UMIOP (*Unreliable Multicast Inter-ORB Protocol*) [Object Management Group, 2001] que deram origem ao protocolo MIOP (*Multicast Inter-ORB Protocol*), responsável pelo mapeamento do GIOP sobre a pilha UDP/*multicast* IP. O MIOP é o elemento chave para tornar disponível um serviço de *multicast* não confiável em suportes de objetos distribuídos CORBA.

Prover suportes de comunicação de grupo que oferecem garantias de entrega e de ordem de mensagens para aplicações distribuídas, envolve uma combinação de protocolos que lidam tanto com gerenciamento de grupo (*membership*, detecção de faltas, transferência de estados, etc.) como com a própria comunicação de grupo. Estas abstrações estão sendo padronizadas separadamente pela OMG: o gerenciamento de grupo é tratado nas especificações FT-CORBA e a comunicação de grupo através de protocolos como o MIOP, definido nas especificações UMIOP.

O GROUPPAC [Lung et al., 2001] é uma implementação completa do padrão FT-CORBA, já as especificações UMIOP foram implementadas no ORB MJACO

[Bessani et al., 2002], ambos desenvolvidos pelo grupo de sistemas distribuídos do LCMI/DAS/UFSC e disponibilizados livremente em (<http://grouppac.sf.net>).

Este trabalho apresenta os esforços realizados, no contexto de um projeto de iniciação científica, para a concepção de uma solução completa para replicação ativa no FT-CORBA a partir das primitivas de comunicação definidas no padrão UMIOP. Esta solução representa a integração destas especificações complementadas com um protocolo de difusão atômica e uma arquitetura de integração do mesmo (através de objetos de serviço) no modelo do FT-CORBA. A implementação dos mecanismos e protocolos especificados se deu através da continuidade do projeto GROUPPAC usando os serviços implementados no ORB MJACO.

O presente artigo está organizado da seguinte forma: na seção 2 temos um breve resumo sobre os grupos nas especificações da OMG. A seção 3 apresenta o protocolo desenvolvido para dar suporte a replicação ativa no FT-CORBA usando o MIOP. A arquitetura do SCG (Suporte de Comunicação de Grupo), que permite a inclusão de novas ferramentas de comunicação de grupo na forma de *plugins*, é discutida na seção 4. Ainda nesta seção, temos a descrição de como o *plugin* para o nosso protocolo foi desenvolvido. A seção 5 apresenta medidas de desempenho, comparando-o com outros *plugins* para ferramentas proprietárias. Finalmente, a seção 6 relata alguns experimentos similares da literatura e na seção 7 temos as conclusões do trabalho.

## 2. As especificações FT-CORBA e UMIOP

A arquitetura **FT-CORBA** [Object Management Group, 2002] define os serviços, em nível de *middleware*, responsáveis por prover as funcionalidades básicas para aplicações tolerantes a falhas através da replicação de objetos CORBA. Esses serviços estão divididos em três módulos básicos:

- **Gerenciamento de Replicação (SGR):** Este é o serviço responsável pelo ciclo de vida dos grupos. Duas funcionalidades são oferecidas por ele: gerenciamento de propriedades, onde as características funcionais da replicação (tipo de replicação usada, número mínimo de réplicas, etc.) são definidas; e o gerenciamento de grupos, que oferece mecanismos para a criação de membros (através de fábricas de objetos) e para o controle de *membership* dos grupos gerenciados;
- **Gerenciamento de Falhas (SGF):** É o serviço responsável pela detecção, notificação, análise e diagnóstico de falhas. Este serviço trabalha em conjunto com o SGR para que este último mantenha um *membership* sempre atualizado dos grupos;
- **Gerenciamento de Recuperação e Logging (SRL):** O SRL é responsável pela manutenção da consistência de estados das réplicas. Este serviço define mecanismos para a recuperação de réplicas e do próprio grupo. Entre estes mecanismos está um suporte para construção de *logs* usados no armazenamento de requisições enviadas ao grupo. O SRL também fornece mecanismos para atualização de membros através de *checkpoints* além de atuar na recuperação de réplicas faltosas através da atualização de seus estados.

Note que entre estes módulos nada é dito sobre o suporte de comunicação de grupo, indispensável principalmente na replicação ativa: não existem interfaces e nem tampouco um protocolo padronizado para tal. As especificações apenas sugerem que se utilize qualquer suporte proprietário para este fim.

Em relação a interoperabilidade com grupos onde objetos são mantidos por diferentes ORBs que implementam o FT-CORBA, a OMG criou um formato padronizado para a referência de grupo de objetos: IOGR (*Interoperable Object Group Reference*). A IOGR consiste basicamente em um conjunto de endereços (perfis IIOP) que identificam cada membro do grupo ou um *proxy* para acesso a estes.

A especificação **UMIOP** [Object Management Group, 2001] define um protocolo baseado em *multicast* IP e um modelo de objetos que permite a difusão de mensagens sem garantia de confiabilidade em grupos de objetos de aplicação. O modelo de objeto definido permite que vários objetos CORBA possam ser invocados simultaneamente através de uma única referência, contrastando com o modelo convencional do CORBA, onde uma referência de objeto corresponde a uma única implementação do mesmo.

Em termos de interoperabilidade, o padrão UMIOP também define uma referência de grupo que permite endereçar zero ou mais objetos. Esta IOR (*Interoperable Object Reference*) de grupo utiliza um tipo de perfil diferente do perfil IIOP para difundir mensagens via UDP/*multicast* IP. Este perfil contém as informações necessárias para o acesso ao grupo em nível de transporte (endereço IP classe D e porta), além da identificação lógica do grupo, utilizada para acessar os objetos membros em nível de ORB.

A grande diferença entre a IOGR do FT-CORBA e a IOR de grupo do UMIOP são as informações contidas nas mesmas: enquanto a IOGR fornece a lista de membros do grupo (*membership*), a IOR de grupo contém apenas uma chave de grupo e um endereço *multicast* IP usados para se chegar aos membros do grupo.

### 3. Protocolo para Replicação Ativa a Partir das Especificações UMIOP

A implementação de replicação ativa (também chamada replicação por máquina de estados) exige o chamado determinismo de réplicas [Schneider, 1990]. Este determinismo pode ser atendido fazendo com que os estados dos objetos replicados sejam determinados apenas pelo conjunto de requisições recebidas por estes, e, difundindo as requisições com garantias de confiabilidade e ordem total (todos recebem na mesma ordem). O recebimento de mensagens em uma mesma ordem é implementado a partir da difusão atômica, uma primitiva de comunicação que satisfaz as propriedades de Validade, Acordo, Integridade e Ordenação Total Local [Hadzilacos and Toueg, 1994].

Partindo de um ORB como o MJACO, que fornece primitivas de comunicação ponto a ponto confiável (via IIOP) e difusão não confiável (via MIOP), como ferramenta de comunicação de grupo base para replicação ativa, nos deparamos com alguns problemas:

- O MIOP, protocolo definido nas especificações UMIOP, é não confiável e portanto o seu uso puro e simples é inadequado;
- O modelo de objetos do UMIOP define que o protocolo MIOP só pode ser utilizado para difusão de requisições sem resposta (*oneway*). As requisições que necessitam de resposta devem ser enviadas para um representante de grupo, cujo endereço está definido na referência UMIOP, via IIOP.

O primeiro empecilho é superado através da inclusão de funcionalidades que tornem os serviços fornecidos mais elaborados. A segunda limitação é vencida fazendo com que as

mensagens difundidas no grupo não demandem resposta, implementando um mecanismo de *callback*. Estas funcionalidades e mecanismos são descritos a seguir.

### 3.1. Protocolo de Difusão Atômica

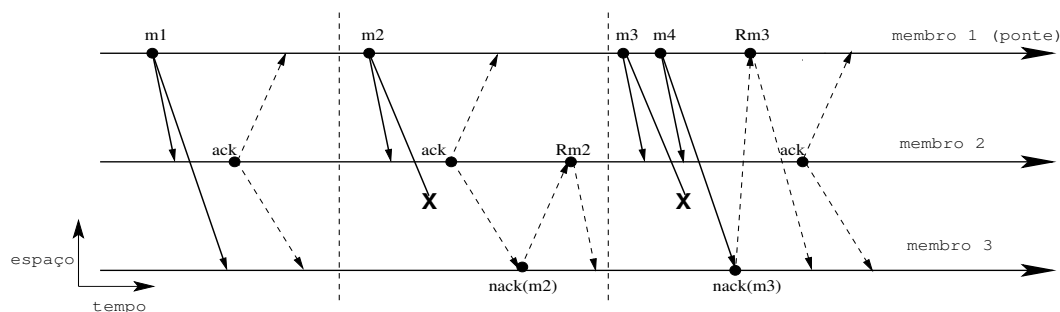
A implementação do protocolo de difusão atômica foi feita em camadas: primeiro definiu-se um serviço de difusão confiável a partir dos mecanismos de comunicação disponíveis no MJACO, depois, o protocolo de difusão atômica que fornece ordem total foi implementado sobre esse serviço. Desta forma temos o FT-CORBA usando um protocolo de difusão atômica que usa um mecanismo de difusão confiável que se baseia somente em serviços de comunicação do ORB (IIOP e MIOP).

A difusão confiável implementada a partir de difusão não confiável (MIOP) se dá através da troca de mensagens de controle entre os membros do grupo. Existem dois tipos de mensagens de controle: ACKs (confirmação de recebimento) e NACKs (recuperação de mensagens perdidas). Os parâmetros dos ACKs são a identificação do emissor e a identificação da última requisição entregue (executada) por este membro. Já nos NACKs, os parâmetros enviados são a identificação das requisições que devem ser reenviadas.

Quanto ao mecanismo de NACKs, as requisições perdidas podem ser recuperadas de duas formas:

1. Quando do recebimento de um ACK e detecção da perda (através do identificador da última mensagem entregue), uma solicitação de reenvio da(s) requisição(s) perdidas (NACK) é feita pelo receptor. Esta solicitação é enviada ao emissor do ACK, uma vez que se ele está confirmando o recebimento de uma mensagem, presume-se que ele a tenha recebido. Se a mensagem não for reposta no grupo, outro processo (membro do grupo) é selecionado e o NACK é enviado a este;
2. Quando do recebimento de uma requisição e detecção da perda (através da identificação de uma falha na sequência de requisições recebidas), o receptor envia o NACK diretamente ao processo que difundiu esta requisição. Caso este processo falhe, outro membro do grupo é escolhido, assim como no item anterior.

Esta troca de mensagens pode ser visualizada na figura 1, onde são considerados apenas os ACKs do membro 2 para facilitar a visualização.



**Figura 1: Recuperação de mensagens perdidas.**

No protocolo, os ACKs são enviados via MIOP (difusão não confiável), isto significa que a perda de alguns deles não trará problemas ao membro que os perdeu, o único prejuízo

é o aumento do tamanho do *buffer* de requisições não estáveis<sup>1</sup>. Quando o próximo ACK for recebido esta situação tende a se normalizar uma vez que os ACKs trazem o identificador da última (maior) mensagem entregue pelo membro que o emitiu.

Um fator importante para a máxima confiabilidade deste protocolo é o fato das mensagens de recuperação (NACKs) serem enviadas via IIOP, o que garante que elas serão recebidas e respondidas se não houver falhas nos processos participantes da recuperação.

Protocolos que se baseiam em NACKs são chamados “controlados pelo receptor” [Levine and Garcia-Luna-Aceves, 1998], pois em geral é este o responsável por identificar perdas e pedir retransmissões. Apesar de escalável e eficiente em ambientes onde a taxa de perda de mensagens é baixa, esta classe de protocolo sofre do problema do *buffer* infinito: o acordo só pode ser provado se considerarmos que os processos armazenam todas as mensagens recebidas para poderem responder a eventuais NACKs. O uso de ACKs como confirmações de recebimento de mensagens, apesar de limitar a escalabilidade do protocolo (devido a necessidade de um serviço de *membership*), evita o problema do *buffer* infinito, já que os processos podem saber quais mensagens são estáveis, tornando possível a retirada das mesmas de seus *buffers*. A necessidade do serviço de *membership* não é um problema quando consideramos uma implementação do padrão FT-CORBA, já que a lista de membros esta na própria referência de grupo (IOGR). Outro problema decorrente do uso de ACKs no protocolo é a elevada quantidade de confirmações que será enviada:  $\#G$  ACKs para cada mensagem. Este problema é minimizado no protocolo proposto fazendo com que os processos enviem ACKs a cada  $N$  mensagens recebidas ou a cada  $T$  unidades de tempo.  $N$  e  $T$  são parâmetros do protocolo ajustados de acordo com o ambiente de execução.

Partindo de uma primitiva de difusão confiável, a implementação de difusão atômica requer apenas a implementação da propriedade de ordem total. Assim, basta a definição de um mecanismo que garanta que todos os processos corretos membros do grupo entreguem a mesma sequência de mensagens difundidas neste. O mecanismo usado em nossa implementação é baseado no paradigma de ordenador fixo [Defago et al., 2000], onde os processos recebem as requisições difundidas no grupo através de um serviço de difusão confiável e cabe a um deles, o ordenador, impor a ordem de entrega destas requisições. As mensagens contendo as ordens das requisições são difundidas no grupo de forma semelhante às requisições, ou seja, via *multicast* confiável e quando o processo que esta difundindo as requisições também é o ordenador as mesmas são enviadas já com sua ordem definida, aumentando assim o desempenho do protocolo. Em caso de falha do ordenador o notificador de falhas avisa a todos os membros do grupo que definem, de maneira determinística, um novo ordenador.

Uma apresentação formal dos protocolos descritos nesta seção, juntamente com as provas de corretude dos mesmos, pode ser encontrada em [Bessani et al., 2004a].

#### 4. Integração de Ferramentas de Comunicação de Grupo ao FT-CORBA

O modelo de replicação ativa, proposto para o FT-CORBA, baseia-se na idéia de um grupo fechado<sup>2</sup> de réplicas acessado por clientes leves através do mecanismo de comunicação usual

---

<sup>1</sup>Uma requisição é considerada estável quando ela já tiver sido entregue por todos os membros do grupo, isto é descoberto através do envio de acks.

<sup>2</sup>Grupos fechados são aqueles em que somente membros do grupo podem difundir mensagens neste.

do CORBA (chamada de métodos remotos via IIOP). A especificação FT-CORBA também prevê o uso de ferramentas de comunicação de grupo proprietárias para difusão atômica [Object Management Group, 2002]. Entretanto, estas mesmas especificações não definem como estas ferramentas de comunicação de grupo devem ser integradas na arquitetura FT-CORBA.

Diante disto foi criado o *framework* SCG (Suporte de Comunicação de Grupo). O SCG é parte integrante do GROUPPAC e define um mecanismo padrão através do qual a infraestrutura FT-CORBA interage com diferentes ferramentas de comunicação de grupo visando prover propriedades de comunicação necessárias a replicação ativa. Através deste mecanismo é possível a integração de qualquer suporte de comunicação de grupo ao FT-CORBA por meio da construção de adaptadores que se integram como *plugins* no GROUPPAC.

O SCG foi concebido através da instanciação de vários padrões de projeto utilizados em três componentes:

- **Proxy de Requisições:** Este componente recebe todas as requisições enviadas a réplica e as repassa ao *plugin* de comunicação de grupo. Sua implementação é baseada no conceito de servidores genéricos.
- **Invocador de Requisições:** É o responsável por passar as requisições recebidas do *plugin* de comunicação de grupo para a réplica do grupo invocado.
- **Suporte ao Plugin de Comunicação de Grupo:** Compreende as classes e interfaces de suporte a *plugins* de ferramentas de comunicação de grupo.

A partir destes três mecanismos é possível implementar adaptadores que integram o GROUPPAC a qualquer sistema de comunicação de grupo. Em [Bessani et al., 2004b] temos uma explicação detalhada sobre a construção e o funcionamento do SCG.

#### 4.1. Integração das especificações FT-CORBA e UMIOP: O *plugin* MJACO

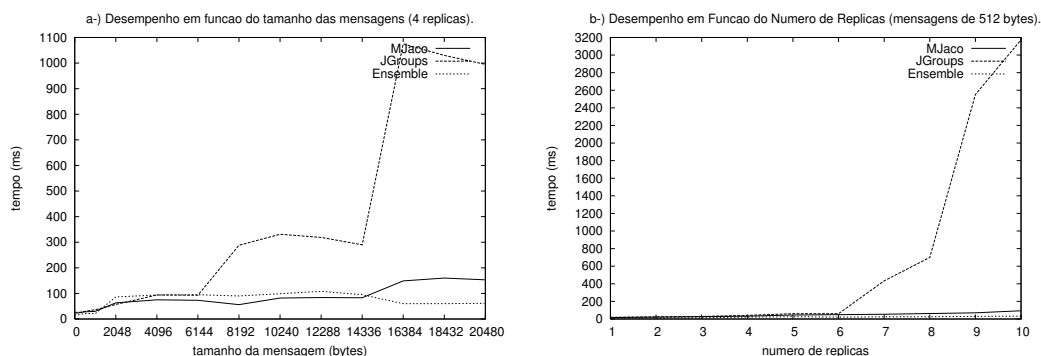
O *plugin* MJACO explora o modelo de objetos definidos no padrão UMIOP, que permite o acesso a vários objetos a partir de uma única referência de grupo, e os serviços de detecção de falhas do FT-CORBA, obtendo assim, um serviço de replicação ativa inteiramente baseado em padrões da OMG. A idéia básica para a construção deste *plugin* é refletir o grupo FT-CORBA, representado pela IOGR, em um grupo UMIOP, usado no contexto do SCG, para então fazer uso dos serviços de comunicação do MJACO.

A implementação do *plugin* foi feita a partir de duas interfaces definidas em IDL: *UMIOPAdaptorReceiver* e *ReceiverResponses*. A primeira é implementada por todos os adaptadores receptores, que são registrados como membros do grupo UMIOP que reflete o grupo FT-CORBA acessado. Esta interface fornece métodos de recebimento de mensagens como *receiveRequest()*, de confiabilidade (*ack()* e *nack()*) e de ordenação (*setGlobalOrder()*). A interface *ReceiverResponses* tem sua implementação ativada pelo adaptador emissor para vencer a limitação de requisições sem resposta imposta pelo padrão UMIOP. Desta forma, apesar das requisições serem difundidas no grupo através do método *receiveRequest()* da interface *UMIOPAdaptorReceiver*, as respostas são enviadas via IIOP através da operação *receiveResponse()* da interface *ReceiverResponses*. A referência para o objeto receptor (que implementa esta interface) é passada como um parâmetro em *receiveRequest()*, implementando desta forma o mecanismo de *callback*.

## 5. Desempenho e Escalabilidade da Replicação Ativa

A fim de validar a implementação do protocolo de difusão atômica proposto foram realizados alguns experimentos com o *plugin* SCG implementado comparando-o com outros dois *plugins* (implementados anteriormente) para ferramentas proprietárias: ENSEMBLE [van Renesse et al., 1998] e JGROUPS [JGroups, 2004]. A aplicação exemplo utilizada foi um serviço de armazenamento remoto simples que provê uma operação para o armazenamento de um bloco de dados retornando um resumo deste (*hash*). O estilo de replicação empregado nos testes foi a ativa com votação, onde a resposta retornada ao cliente é o valor majoritário escolhido a partir do conjunto de respostas retornadas por todas as réplicas. As ferramentas de comunicação de grupo foram configuradas de maneira equivalente para prover um serviço de difusão atômica para o GROUPPAC. O cenário dos testes foi uma rede local onde não ocorreram falhas nos objetos.

A partir do cenário definido, dois tipos de experimentos foram realizados medindo o tempo entre o envio da requisição ao serviço replicado e a recepção da resposta pelo cliente. A figura 2 apresenta o resultado destes experimentos.



**Figura 2: Desempenho do mecanismo de replicação ativa.**

Na figura 2-a temos o tempo de resposta do serviço replicado considerando diferentes tamanhos de mensagens. Já na figura 2-b temos este tempo considerando um número variável de réplicas. Em ambos os casos temos resultados equivalentes em média entre a solução que implementa nossos protocolos e a baseada na ferramenta ENSEMBLE. A ferramenta de comunicação JGROUPS obteve resultados bastante inferiores, principalmente em termos de escalabilidade.

Os resultados apresentados na figura 2 são bastante positivos, principalmente se levarmos em consideração que nossa implementação é um primeiro protótipo enquanto o ENSEMBLE é uma ferramenta consagrada e bastante estável. Outro ponto a ser ressaltado é que o *plugin* implementado utiliza apenas mecanismos de comunicação do ORB, que apesar de garantir portabilidade, impõem um *overhead* natural do *middleware*.

## 6. Trabalhos Relacionados

Existem vários trabalhos que apresentam soluções para replicação ativa no CORBA, entretanto apenas dois destes sistemas seguem as especificações FT-CORBA. O sistema Eternal



[Moser et al., 1999] utiliza interceptação em nível de sistema operacional para integrar a ferramenta de comunicação de grupo Totem [Moser et al., 1996]. Esta abordagem, além de não ser interoperável prende a infraestrutura a apenas uma ferramenta de comunicação de grupo.

O projeto IRL (*Interoperable Replication Logic*) [Baldoni et al., 2002] tem por finalidade implementar uma infraestrutura FT-CORBA completamente portátil que possa ser integrada a ORBs já existentes. A abordagem do IRL para replicação ativa consiste em dividir o sistema em três camadas instaladas em ambientes diferenciados em termos de sincronismo: cliente (sistema assíncrono), camada intermediária (parcialmente síncrono) e réplicas (assíncrono). Neste contexto, o IRL implementa *gateways* replicados na camada intermediária, onde os protocolos de acordo podem ser implementados, pois não estão sujeitos a impossibilidade FLP [Fischer et al., 1985]. Estes *gateways* recebem chamadas dos clientes via IIOP e as reenviam (também via IIOP) as réplicas do objeto invocado. Além desta abordagem ter um custo maior em termos de desempenho, pois existe um passo a mais de comunicação (duas chamadas IIOP e uma execução do protocolo de acordo para a atualização das réplicas do *gateways* na camada intermediária), ela sofre do problema de concentração dos *gateways* em redes especiais: devido as propriedades de sincronismo requeridas, este mecanismo só pode ser implementado em redes locais ou controladas, que não podem estar espalhadas geograficamente.

## 7. Conclusão

Este trabalho apresenta nossos esforços na construção de um suporte para replicação ativa no modelo de objetos CORBA. O objetivo traçado, neste caso, era a integração das especificações FT-CORBA e UMIOP em uma plataforma única de *middleware*. Para suportar a replicação ativa foi definido um protocolo que faz uso das pilhas MIOP/UDP/*multicast* IP e IIOP/TCP/IP definidas para ORBs CORBA. Este protocolo foi implementado na forma de um *plugin* para o *framework* SCG. Experimentos realizados demonstraram um bom desempenho e escalabilidade deste *plugin*.

O resultado prático desta integração é um suporte de comunicação de grupo com primitivas básicas de difusão seletiva (ainda sem garantia e sem ordenação), que é estendido para prover qualidades de serviço mais elaboradas em termos de garantia e ordem. Neste caso, a replicação ativa passa também a ser suportada usando somente padrões definidos pela OMG, dentro do próprio *middleware*.

## Referências

- Baldoni, R., Marchetti, C., and Termini, A. (2002). Active software replication through a three-tier approach. In *Proceedings of the 21st Symposium on Reliable Distributed Systems (SRDS'02)*, pages 109–118, Osaka, Japão. IEEE.
- Bessani, A. N., da Silva Fraga, J., and Lung, L. C. (2002). MJaco: Integração do multicast IP na arquitetura CORBA. In *Anais do XX Simpósio Brasileiro de Redes de Computadores - SBRC'2002*, Buzios, RJ.
- Bessani, A. N., da Silva Fraga, J., Lung, L. C., and Alchieri, E. A. (2004a). Replicação Ativa no CORBA: Padrões, Protocolos e Framework de Implementação. In *Anais do XXII Simpósio Brasileiro de Redes de Computadores - SBRC'2004*, Gramado, RS.

- Bessani, A. N., Lung, L. C., Alchieri, E. A., and da Silva Fraga, J. (2004b). GroupPac 3: Extendendo o FT-CORBA para gerenciamento e replicação ativa. In *Anais do V Workshop de Testes e Tolerância a Falhas - WTF 2004*, Gramado, RS.
- Deering, S. E. (1986). Host Extensions for IP Multicasting (RFC 988). IETF RFC.
- Defago, X., Schiper, A., and Urban, P. (2000). Totally ordered broadcast and multicast algorithms: a comprehensive survey. Technical Report TR DSC/2000/036, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.
- Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374–382.
- Hadzilacos, V. and Toueg, S. (1994). A modular approach to the specification and implementation of fault-tolerant broadcasts. Technical report, Department of Computer Science, Cornell University, New York - USA.
- JGroups (2004). JGroups: A toolkit for reliable multicast communication. Acessível em <http://www.jgroups.org>.
- Levine, B. N. and Garcia-Luna-Aceves, J. J. (1998). A comparison of reliable multicast protocols. *Multimedia Systems*, 6(5):334–348.
- 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 XIX Simpósio Brasileiro de Redes de Computadores - SBRC'2001*, Florianópolis, SC.
- Moser, L. E., Melliar-Smith, P. M., Agarwal, D. A., Budhia, R. K., and Lingley-Papadopoulos, C. A. (1996). Totem: A fault-tolerant multicast group communication system. *Communications of the ACM*, 39(4):54–63.
- Moser, L. E., Melliar-Smith, P. M., Narasimhan, P., Tewksbury, L. A., and Kalogeraki, V. (1999). The eternal system: an architecture for enterprise applications. In *Proceedings of the 3rd International Enterprise Distributed Object Computing Conference (EDOC'99)*, Mannheim - Alemanha.
- Object Management Group (2001). Unreliable multicast inter-orb protocol specification v1.0. OMG Standart ptc/03-01-11.
- Object Management Group (2002). The Common Object Request Broker Architecture: Core specification v3.0. OMG Standart formal/02-12-06.
- Schneider, F. B. (1990). Implementing fault-tolerant service using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4):299–319.
- van Renesse, R., Birman, K. P., Mark Hayden, A. V., and Karr, D. (1998). Building adaptive systems using ensemble. *Software - Pra. and Exp.*, 28(9):963–979.