

Construção de Ferramentas de Comunicação de Grupo em Sistemas Tempo Real

Rafael Campello
campello@inf.ufrgs.br

Taisy Weber
taisy@inf.ufrgs.br

João Netto
netto@inf.ufrgs.br

Curso de Pós-Graduação em Ciência da Computação
Instituto de Informática
Universidade Federal do Rio Grande do Sul

Resumo

A comunicação de grupo tem sido um dos principais paradigmas de comunicação em sistemas distribuídos e muitas ferramentas têm sido construídas. Este trabalho apresenta as questões de projeto que devem ser levadas em consideração na construção de tais ferramentas para aplicações tempo real. A análise é baseada na experiência com um serviço de comunicação de grupo, chamado *RealGroup*, recentemente desenvolvido.

Abstract

The group communication has been one of the main communication paradigms in distributed systems and many tools have been built. This work presents the design issues that must be addressed in the building of those tools for real time applications. The analysis is based on the experience with a group communication service, called *RealGroup*, recently developed

1. INTRODUÇÃO

No desenvolvimento de sistemas distribuídos, um dos pontos mais positivos é o emprego de técnicas de tolerância a falhas. Com custos geralmente altos, devido às exigências de redundância, essas técnicas visam manter um fornecimento contínuo de serviços, mesmo na presença de falhas. Em ambientes distribuídos, entretanto, esse custo adicional é reduzido pela existência de elementos naturalmente redundantes e independentes. Por outro lado, o fraco acoplamento existente nesses sistemas, onde a única forma de interação é a troca de mensagens, gera uma maior dificuldade de desenvolvimento, agravada pelo paralelismo real na execução das diferentes partes do sistema.

A despeito de estilos de computação distribuída bem conhecidos como RPC e cliente/servidor, a comunicação de grupo vem ganhando grande aceitação. Mesmo úteis e maduros, esses estilos conhecidos mostram que sua natureza ponto-a-ponto e requisição-resposta não oferecem uma solução universal ao aumento da demanda de desenvolvimento de aplicações distribuídas. Nessas aplicações, interações mais complexas e altamente concorrentes entre os vários participantes são necessárias, reforçando as vantagens de um paradigma como a comunicação de grupo.

Além dos obstáculos naturais na implementação de uma ferramenta de comunicação de grupo, novas variantes surgem quando essa comunicação visa ser efetuada em sistemas distribuídos tempo

real. Nesses ambientes, fortes restrições temporais são impostas à comunicação, exigindo um projeto diferenciado e condizente com os novos cenários de falha possíveis.

Outra grande característica de sistemas computacionais aplicados à ambientes tempo real é a freqüente utilização de soluções proprietárias. Tanto em *software* como em *hardware* são usados elementos especificamente criados para cada aplicação, dada a necessidade de características não encontradas em componentes padronizados ou devido ao caráter demasiadamente crítico de tais aplicações. Isso acaba agregando, a determinados projetos, custos muito elevados.

O Grupo de Tolerância a Falhas da UFRGS está desenvolvendo uma ferramenta para comunicação de grupo em sistemas distribuídos, voltada para o trabalho em ambientes tempo real, chamada *RealGroup*. Essa ferramenta usa *hardware* padrão, com computadores pessoais convencionais interligados por uma rede privada (ninguém mais tem acesso ao canal de comunicação) padrão Ethernet e não duplicada, além de um sistema operacional comercial, chamado QNX [QNX93].

Serão apresentados, na segunda seção deste artigo, detalhes sobre comunicação de grupo tempo real. Na seção 3 serão mostradas algumas diretrizes de projeto para ferramentas de comunicação de grupo em sistemas tempo real. Finalmente, serão apresentadas as conclusões e perspectivas futuras.

2. COMUNICAÇÃO DE GRUPO TEMPO REAL

A comunicação entre grupos de processos em ambientes distribuídos possui vários pontos importantes a serem analisados, incluindo o envio confiável de mensagens, a manutenção da consistência nas visões de grupo e a detecção e tratamento de falhas de processo e de comunicação [ABD96]. Em aplicações tempo-real distribuídas, que operam sob restrições de tempo e de confiabilidade, o problema torna-se mais complexo, já que atividades de difusão confiável e gerência de grupos devem ser realizadas em limites de tempo restritos, mesmo na presença de falhas. Nesse caso, cada processo membro precisa validar mensagens de difusão recebidas, dados seus prazos temporais, assim como a gerência de grupos deve garantir limites máximos de tempo no caso de falha ou recuperação de membros, ou em falhas na comunicação.

Outro aspecto importante em sistemas tempo-real é o seu sincronismo. Os sistemas distribuídos podem ser separados em duas correntes principais: síncronos e assíncronos. Uma definição comumente aceita de sincronismo é descrita pelas seguintes propriedades: velocidade de processamento limitada e conhecida; atrasos limitados e conhecidos na entrega de mensagens; taxa de desvio do relógio local limitada e conhecida. Com relação a protocolos de difusão para uso em sistemas tempo-real, uma discussão sobre a validade ou não de protocolos baseados em relógios lógicos (*timer-driven* ou *clock-less*) em aplicações tempo-real é apresentada por Veríssimo [VER90]. No entanto, para garantir um alto nível de sincronismo no sistema, serviços de sincronização de relógios são utilizados na maioria dos protocolos de difusão confiável tempo-real.

Vários protocolos tolerantes a falhas para difusão atômica e ordenada e para gerência de grupos foram propostos, tanto para sistemas assíncronos [CHA84] [MOS96] [REN96] [KAA92] como para sistemas síncronos [ABD96] [CRI85] [VER89] [KOP94].

O principal problema encontrado na implementação de protocolos de difusão confiável, gerência de grupos e sincronização de relógios, em sistemas tempo-real distribuídos, é a obtenção das propriedades de sincronismo descritas acima, principalmente valendo-se de componentes padronizados. Embora o determinismo de processamento seja, de certa forma, garantido pelas características de um sistema operacional tempo real, o determinismo em relação à comunicação é

um ponto muito delicado. Como toda a característica síncrona do sistema e dos protocolos está baseada nesse determinismo, é importantíssima a observação dessas propriedades. As características fundamentais para uma comunicação tempo-real, segundo Veríssimo [VER93], são: atrasos conhecidos e limitados na entrega de mensagens; comportamento determinístico na presença de distúrbios (sobrecargas, falhas); diferentes classes de latência, isto é, possível estabelecimento de prioridades à nível de mensagens; e finalmente conectividade.

Alguns protocolos padronizados de acesso ao meio tratam distúrbios, como omissões e sobrecargas, assim como implementam mecanismos de prioridades e tratam problemas de conectividade. São redes que oferecem serviços isócronos e de alta disponibilidade, valendo-se, muitas vezes, de algum tipo de redundância espacial. Um bom exemplo disso é a FDDI (*Fiber Distributed Data Interface*) [TAN96] [CHE97]. Com protocolos de acesso probabilístico ao meio, como é o caso das distintas variedades de CSMA (*Carrier Sense Multiple Access*), surgem problemas com o indeterminismo no atraso máximo de entrega das mensagens e na presença de distúrbios, a falta de classes de latência e a conectividade. Em qualquer ferramenta baseada nesse tipo de rede, é fundamental criar algum mecanismo de controle de acesso ao meio, eliminando as colisões e obtendo atrasos limitados. Outras características como conectividade e comportamento determinístico na presença de falhas só poderão ser alcançadas com o auxílio de algum tipo de redundância física do meio, aumentando os custos e prendendo a ferramenta a uma plataforma padronizada.

3. QUESTÕES DE PROJETO EM FERRAMENTAS DE COMUNICAÇÃO DE GRUPO TEMPO REAL

Como observado no desenvolvimento do *RealGroup*, a determinação dos critérios de projeto de uma ferramenta de comunicação de grupo passa pela idealização de um serviço voltado ao tipo de aplicação escolhida, pela determinação dos recursos disponíveis e, então, baseado em necessidades concretas, acaba na escolha de determinado protocolo de difusão e na adequação desses critérios às características desse protocolo.

Tendo em vista atender a demanda de sistemas com restrições temporais, é notória a importância em garantir todas as características de uma comunicação tempo real, apresentando um comportamento confiável e temporal. Características como atrasos limitados na entrega das mensagens, comportamento determinístico na presença de distúrbios e previsão de classes de prioridades são de extrema importância em face ao tipo de aplicação esperada. A rigidez com que essas características serão atendidas depende, basicamente, do tipo de aplicação esperada. Além disso, a possibilidade do envio tanto de mensagens esporádicas como de mensagens periódicas torna-se de grande valia para esse tipo de aplicação.

Outro critério importante a ser determinado é o tipo de plataforma a ser empregada. Como já mencionado, é comum o uso de plataformas proprietárias, com elementos de *hardware* duplicados, facilitando e garantindo a manutenção das características temporais exigidas. Com vantagens em relação a custos e portabilidade, entretanto, a utilização de uma plataforma padronizada, com elementos de grande aceitação no mercado, é outra alternativa possível. Essa escolha deve ser feita com base nas aplicações alvo dessa ferramenta. Baseada em componentes padronizados, por exemplo, não serão esperados sistemas de caráter extremamente crítico sobre tal serviço, requisitos normalmente satisfeitos por soluções proprietárias.

Por fim, detalhes sobre o protocolo de comunicação de grupo a ser usado surgem. Mesmo aparentando pouca relação com o tipo de aplicação prevista para o serviço, certas decisões não

podem deixar de lado o caráter temporal de tais aplicações. Decisões como semânticas de envio e de resposta, por exemplo, precisam levar em consideração a existência de limites temporais a serem cumpridos, inviabilizando qualquer comportamento não determinístico. As seguintes diretrizes precisam ser observadas:

- **Endereçamento:** mesmo sendo esse um critério independente das características tempo real a serem atendidas, é importante dar ao usuário de nível superior facilidades quanto à transparência no endereçamento de membros. Tendo isso em vista, o tipo de endereçamento ideal seria o via *kernel*. Assim, toda vez que um membro do grupo solicita o envio de uma mensagem para os outros, ele passa a mensagem para a camada de nível inferior (serviço de difusão), e este, de posse de uma lista de membros, difunde à todos. A maneira pela qual o processo gerenciador dissemina a mensagem está diretamente ligada à interface de rede existente, sendo tratada como detalhe de implementação. Com esse tipo de endereçamento, um membro pode disseminar uma mensagem para o resto do grupo sem saber quem são ou onde estão os membros destino.
- **Confiabilidade:** diretamente ligada ao tipo de aplicação alvo, a confiabilidade é um fator primordial na maioria dos sistemas tempo real. Assim, ferramentas que visem esses sistemas devem comportar-se de maneira confiável, mesmo em eventuais distúrbios como perdas ou duplicações de mensagens.
- **Semântica de falhas:** o tipo de falha a ser tolerado é outro critério importante. Diretamente ligada ao número de mensagens trocadas pelo protocolo, a semântica de falhas a ser usada geralmente tende a falhas por omissão. Essa escolha por suportar somente falhas por omissão, e não semânticas mais robustas, é justificada pelo alto custo das outras semânticas, associado à baixa ocorrência de outros tipos de falhas. Por outro lado, o suporte a essas outras semânticas pode ser importante, dependendo do tipo de aplicação alvo.
- **Ordenação:** Embora a ordenação total seja uma propriedade que, normalmente, eleva a quantidade de mensagens trocadas e a necessidade de processamento, é de relativa importância para a maior parte das aplicações distribuídas a observação dessa característica. Assim, é de grande interesse a garantia de uma ordenação total das mensagens enviadas pelo grupo, para que todos os membros do grupo recebam as mensagens sempre na mesma ordem.
- **Semântica de envio:** assim como o critério anterior, a semântica de envio a ser utilizada está fortemente relacionada com as necessidades da maioria das aplicações distribuídas. Para tanto, a garantia de uma entrega atômica das mensagens torna-se fundamental em muitos algoritmos distribuídos, principalmente em tarefas que envolvem a obtenção de concordância, sendo, também, ponto fundamental na maioria das ferramentas.
- **Semântica de resposta:** esse critério é limitado pelo comportamento temporal esperado. Semânticas que envolvem espera por respostas podem tornar indeterminado os atrasos possíveis, ou, quando *timeouts* são usados, agravar os cenários de pior caso. Por outro lado, a inexistência de respostas pode acarretar perda de confiabilidade pelo serviço. Assim, esse é um critério que deverá ser analisado com muito cuidado, devendo ser pesado bem na escolha do protocolo de difusão a ser utilizado.
- **Estrutura de grupos:** tratando de estrutura de grupos, quanto mais flexíveis e dinâmicas forem suas características, mais versátil às aplicações de nível superior o serviço será. Para tanto, grupos abertos, não-hierárquicos, com gerenciamento distribuído e com possibilidade de

sobreposição são as escolhas mais sensatas. Esses critérios estão, também, diretamente relacionados com o protocolo de difusão a ser utilizado.

Sendo esses os detalhes a serem observados, resta definir, baseado na realidade de recursos disponíveis ao serviço, quais seriam as maiores dificuldades na efetivação da ferramenta, adequando a mesma à essa realidade. No caso da ferramenta *RealGroup*, todo o sistema está colocado sobre uma plataforma de *hardware* padronizada, com interfaces, topologias e protocolos de rede existentes e comuns no mercado atual. Isso dificultou a efetivação do serviço, gerando a necessidade de adaptações para que o projeto inicial fosse viável, principalmente pela necessidade de um serviço síncrono, com atrasos determinísticos tanto quanto à entrega de mensagens como quanto ao tratamento de distúrbios.

4. CONCLUSÃO

Com base na peculiar forma de interação com o ambiente, apresentada pelos sistemas tempo real, pode-se afirmar que qualquer algoritmo ou mecanismo a ser utilizado nesse tipo de sistema precisa estar adequado às restrições temporais impostas. Teoricamente, como mencionado, esses limites temporais são razoavelmente claros, passando pela utilização de um sistema de comunicação tempo real, com características síncronas, e estendendo-se até o uso de protocolos tempo real, com tempos de latência determinísticos.

No nível de comunicação, técnicas que visam uma troca de mensagens confiável, com mecanismos que tratam a ocorrência de distúrbios, não fogem a essa regra, exigindo canais de comunicação com comportamento determinístico e protocolos que tenham tempos máximos de execução limitados. Usar recursos computacionais que atendam esses limites, como um sistema de comunicação com componentes replicados, por exemplo, bem como *softwares* básicos projetados especificamente para determinada aplicação, como um sistema operacional proprietário, são soluções eficientes e muito utilizadas em aplicações altamente críticas. Por outro lado, valendo-se de recursos computacionais padronizados, de grande aceitação e utilização no mercado atual, assim como de um sistema operacional não proprietário, buscar uma solução alternativa que obedeça os mesmos limites teóricos pode representar ganhos importantes no custo, principalmente para aplicações de caráter menos crítico.

5. REFERÊNCIAS

- [ABD96] ABDELZAHER, Tarek et al. **RTCAST**. lightweight multicast for real-time process groups. [S.l.:s.n.], 1996. Disponível por FTP anônimo em ftp.eecb.umich.edu.
- [CHA84] CHANG, J.; MAXEMCHUK, N. F. Reliable broadcast protocols. **ACM Transactions on Computer Systems**, New York, v.2, n.3, p.251-273, Aug. 1984
- [CHE97] CHEN, Biao; KAMAT, Sanjay; ZHAO, Wei. Fault-tolerant, real-time communication in FDDI-based networks. **Computer**, Los Alamitos, v.30, n.4, p.83-90, Apr. 1997.
- [CRI85] CRISTIAN, F. et al. Atomic broadcast: from simple message diffusion to byzantine agreement. In: 15th International Symposium on Fault Tolerant Computing Systems, 1985, Ann Arbor, USA. **Proceedings**. [S.l.:s.n.], 1985.
- [KAA 92] KAASHOEK, Frans; TANENBAUM, Andrew. **Efficient reliable group communication for distributed systems**. Amsterdam: Department of Mathematics and Computer Science of Vrije Universiteit, 1992.

- [KOP 94] KOPETZ, Hermann; GRÜNSTEIDL, Günter. TTP-a protocol for fault-tolerant real-time systems. **Computer**, Los Alamitos, v.27, n.1, p.14-23, Jan. 1994.
- [MOS 96] MOSER, L. et al. Totem: a fault-tolerant multicast group communication system. **Communications of the ACM**, New York, v.39, n.4, p.54-63, Apr. 1996.
- [QNX 93] QNX SOFTWARE SYSTEMS. **QNX: system architecture**. Ontario: QNX Software Systems, 1993.
- [REN 96] RENESSE, Robert van; BIRMAN, Kenneth P.; MAFFEIS, Silvano. Horus: a flexible group communication system. **Communications of the ACM**, New York, v.39, n.4, p.76-83, Apr. 1996.
- [TAN 96] TANENBAUM, Andrew. **Computer networks**. 3. ed. Upper Saddle River: Prentice-Hall, 1996.
- [VER 89] VERÍSSIMO, P.; RODRIGUES, L.; BAPTISTA, M. AMp: a highly parallel atomic multicast protocol. In: SIGCOMM, 1989, Arlington, Texas. **Proceedings...** New York:ACM, 1989. p.83-93.
- [VER90] VERÍSSIMO, P. Real-Time Data Management with Clock-Less Reliable Broadcast Protocols. In: Workshop on the Management of Replicated Data, 1990, Houston. **Proceedings**. New York: IEEE, 1990.
- [VER 93] VERÍSSIMO, Paulo. Real-time communication. In: MULLENDER, S.J. (Ed.). **Distributed systems**. 2. ed. New York: ACM-Press, 1993. p.447-490.