

P. KEGLEVICH DE BUZIN *

SUMARIO

Este trabalho aborda o problema de concepção de sistemas distribuídos. São estabelecidos alguns conceitos fundamentais, e é proposto um modelo para descrever um ambiente computacional distribuído. A partir deste modelo é constatado que a descrição de sistemas distribuídos introduz um novo estilo de programação.

ABSTRACT

This paper deals with the problem of distributed systems conception. It establish some fundamental concepts and proposes a model to describe an distributed computational environment. From this model it was found out that the description of distributed systems introduces a new stile of programation.

* MsC. Ciência da Computação (UFRGS, 1984); Sistemas Distribuídos, controle de processos e sistemas em tempo-real, engenharia de software, projeto de linguagens de programação. UFRGS-PGCC, Av. Osvaldo Aranha, 99 - Porto Alegre, RS - CEP 90000.

1. INTRODUÇÃO

Avanços tecnológicos tem tornado viável construir sistemas complexos sobre uma rede de computadores. Para assistir tais sistemas existe uma crescente necessidade por maneiras efetivas de organizar e manter sistemas distribuídos. Assim os custos do equipamento reduzem-se assombrosamente, porém surge um impasse, que um número ainda insuficiente de pesquisadores está tentando resolver: a complexidade e o custo do desenvolvimento do software [MCQ 80].

Neste trabalho procura-se obter uma abordagem adequada para a concepção de sistemas distribuídos, para tanto é necessário primeiro um entendimento apropriado de tais sistemas, e a seguir é feita uma proposta de modelo para ambientes computacionais distribuídos, que irá fornecer uma abordagem para um projeto de sistemas distribuídos.

2. O PROBLEMA DA CONCEITUAÇÃO

Para o desenvolvimento de sistemas distribuídos é fundamental um respaldo de um bom entendimento de conceitos consistentes sobre o tipo de sistema a ser desenvolvido. No caso de sistema distribuídos, este problema é fortemente sentido. A carência de um entendimento mais acurado nesta área torna a tarefa de projeto de uma aplicação distribuída tremendamente complexa e onerosa. Como exemplo, cabe observar que a própria definição do que seja um sistema distribuído é obscura, pois tenta-se defini-lo em função da aplicação, o que torna difícil a obtenção dos conceitos fundamentais para estruturar um sistema de computação distribuída.

3. CARACTERIZAÇÃO DOS SISTEMAS DISTRIBUÍDOS

De uma maneira informal, um sistema distribuído pode ser caracterizado como sendo um conjunto de elementos processadores autônomos que se comunicam para realizar um sistema coerente de computação. Cada elemento processador possui pelo menos um processador operante, além de memória local. A operação de cada elemento é assíncrono em relação aos outros. A comunicação é efetuada através de uma rede de interconexão, cuja topologia pode ser qualquer, embora usualmente seja em barra ou em anel [FRA 81], como no caso de redes locais de computadores.

3.1 POTENCIALIDADES

O desenvolvimento dos sistema distribuídos é motivado por vários benefícios antevistos, além de outros ainda a serem constatados. A importância de tais benefícios, e a habilidade para materializá-los depende do ambiente da aplicação. Alguns destes benefícios são citados a seguir:

- a) melhor confiabilidade e recuperabilidade em relação aos sistemas centralizados,
- b) Extensibilidade em tempo de carga ou execução,

- c) melhores condições de aumentar o desempenho que os sistemas centralizados,
- d) a melhora de desempenho tem o potencial de poder ser feita a um custo menor que em sistemas centralizados.
- e) permite uma distribuição da capacidade de processamento, reduzindo com isto o custo de transmissão de dados,
- f) finalmente, possui o potencial de ser mais fácil de programar que sistemas centralizados.

O potencial de ser mais fácil de programar ainda não foi totalmente constatado na prática, em parte devido ao rústico entendimento atual das relações entre os requisitos do processamento de aplicações e os necessários mecanismos de comunicação entre processos [FRA 81].

3.2 SISTEMAS ABERTOS X SISTEMAS FECHADOS

Basicamente os sistemas distribuídos podem ser classificados em duas grandes classes: os sistemas abertos (OSI) e os sistemas fechados. Para os sistemas abertos já existe uma recomendação largamente aceita para a estruturação dos níveis de protocolo, que é a recomendação ISO. Porém, embora esta recomendação se adapte bem para grandes redes de teleprocessamento, a mesma já apresenta problemas para ser usada em redes locais. As redes locais não precisam de uma estrutura de protocolo tão complexa quanto a dos sistemas de transmissão de dados a longa distância, onde é comum um grande número de nodos com sistemas heterogêneos interconectando-se e desconectando-se dinamicamente.

Para os sistemas fechados não existe uma recomendação consagrada como o caso da ISO [DAT 81], mas existe certo consenso quanto ao tipo de arquitetura [ADV 80]. A abordagem conceitual neste caso é totalmente diferente do caso dos sistemas abertos. No caso dos sistemas fechados toda a rede é considerada como a arquitetura de uma máquina, ou seja, a máquina alvo é a rede. No caso dos sistemas abertos, o nodo é a máquina alvo, e a rede é encarada como um recurso disponível a esta máquina. Todo o padrão ISO baseia-se neste conceito.

Se for comparada a estrutura de níveis do modelo de sistema fechado com a estrutura do modelo de referência da ISO, será observado que o modelo da ISO enquadraria os níveis de aplicação, sistema operacional e kernel distribuído do sistema fechado como somente o nível de protocolo de aplicação. Por outro lado, devido a sua característica, o modelo do sistema fechado não se preocupa com os níveis de apresentação e sessão que existem no modelo de referência da ISO. Portanto o modelo de sistemas abertos são totalmente diferentes do modelo dos sistemas fechados, e não podem ser confundidos. Assim para o caso de redes locais, onde se necessita de uma estrutura de protocolos mais simples, o modelo de sistemas fechados se presta muito mais do que o modelo OSI.

4. A OBTENÇÃO DE UM MODELO PARA SISTEMAS DISTRIBUIDOS

Para a obtenção do modelo deste ambiente computacional, é necessário fazer uma consideração mais apurada no que concerne a interação entre processos, a identificação de elementos dentro do ambiente, o tipo de controle, a proteção de recursos e a sincronização de processos. Tais aspectos são considerados na abordagem do problema dos sistemas distribuídos, e a seguir é apresentado um modelo que descreva o ambiente computacional distribuído. Maiores detalhes sobre estas considerações podem ser encontrados em [KEG 84a].

4.1 O PROBLEMA DOS SISTEMAS DISTRIBUÍDOS

A peculiaridade dos sistemas distribuídos é que os mesmos são sistemas fracamente conceitados, devido a arquitetura do hardware, para os quais o estado global não é conhecido. Esta característica causa o surgimento de vários problemas para o projeto do software. A solução destes problemas leva a um entendimento mais acurado do que seja um ambiente de programação distribuído e consequentemente permite definir com mais clareza os problemas das aplicações distribuídas, assim como os problemas das aplicações em qualquer ambiente computacional.

Os principais problemas que surgem, devido as características dos sistemas distribuídos, são a maior dificuldade de controlar a ocorrência de eventos (sincronização), que se reflete no problema da manutenção de consistência do sistema, e a diversidade de aplicações e operações que podem ser feitas ao mesmo tempo sobre tal sistema. A diversidade de aplicações ao mesmo tempo irá se refletir na abordagem do modelo de controle de execução do sistema distribuído, e tornará o problema de proteção mais complexo que em sistemas operacionais.

É importante observar que é o modelo de controle de execução que vai dar a característica de sistema distribuído a um determinado ambiente computacional. Assim, por exemplo, um sistema com vários nodos, mas com um nodo mestre que controla todos os outros, não necessariamente pode ser caracterizado como sistema distribuído, embora tenha nodos distribuídos geograficamente. Tal sistema deve antes ser caracterizado como algo similar ao convencional sistema de teleprocessamento [ADV 80]. A maior complexidade dos sistemas distribuídos surge exatamente do fato de se constituir em vários processadores autônomos que devem cooperar juntos para produzir uma computação coerente. Portanto um sistema distribuído é principalmente caracterizado por um modelo de controle de execução distribuído, ao passo que um sistema centralizado é caracterizado por um modelo de controle de execução centralizado.

A abordagem do problema da sincronização implica em uma análise mais acurada do controle da ocorrência de eventos em sistemas distribuídos, examinando a nominação e endereçamento em sistemas computacionais, os mecanismos de sincronização e os tipos de interação entre processos. É interessante notar que existem basicamente três modelos de interação entre processos [AND 81]: monitor (ou procedure), passagem de mensagem e operações remotas. Tais modelos permitem analisar as classes de mecanismos de sincronização mais adequados para cada tipo de ambiente computacional e para cada tipo de aplicação [AND 83].

4.2 ENTENDENDO UM AMBIENTE COMPUTACIONAL DISTRIBUÍDO

Um ambiente computacional pode ser visto como um conjunto de recursos disponíveis dentro de um mesmo sistema e acessíveis de uma maneira uniforme, de acordo com as regras do ambiente.

4.2.1 TIPOS DE AMBIENTES

Foram considerados quatro tipos diferentes de ambientes computacionais: distribuído, paralelo, concorrente e sequencial. Devido às suas características, estes ambientes relacionam-se fortemente, de forma que um ambiente distribuído engloba um ambiente paralelo, que por sua vez engloba um ambiente concorrente, que por sua vez engloba um ambiente sequencial.

Assim, até ao nível de ambiente concorrente, inclusive, não existe preocupação com a localização de unidades de programa. Com o processamento paralelo já existe a preocupação com a localização das unidades de programa ou com a associação algoritmo-processador para a computação de um programa, sem no entanto preocupar-se com a característica do meio de comunicação. Já em um ambiente distribuído existe a preocupação com o meio de comunicação, pois é preciso controlar as situações de erro e de retardo de mensagens.

Quanto ao tipo de recursos que oferecem, os ambientes computacionais podem ser subdivididos em ambientes físicos (hardware) e ambientes lógicos (software ou parte algorítmica). Portanto, para formar um ambiente computacional é necessário localizar sobre o ambiente físico o ambiente lógico.

4.2.2 AMBIENTES FÍSICOS

O ambiente físico pode ser somente uma máquina hospedeira, para o caso de ambiente computacional concorrente, ou uma arquitetura com vários processadores, para o caso de um ambiente paralelo, ou uma rede, para o caso distribuído. Assim, os elementos de um ambiente físico podem ser processadores, memórias compartilhadas ou nodos de uma rede.

No caso de ambientes físicos de sistemas distribuídos, deve ser considerado um aspecto importante que é a característica do meio de comunicação. De um modo geral, um sistema físico pode ser considerado como fracamente conectado quando possui retardo, de modo que a produção de um evento e a sua materialização é distinguível. Isto sem dúvida torna bastante complexo o problema da sincronização e controle de eventos. Por outro lado, o sistema pode ser assumido como fortemente conectado quando não há retardo na comunicação, ou seja, a produção de um evento e a sua materialização são indistinguíveis. Como exemplo do último caso tem-se uma arquitetura de multiprocessadores que compartilham uma memória.

4.2.3 AMBIENTES LÓGICOS

A estrutura da parte lógica de um ambiente computacional deve ser totalmente modular para poder comportar aplicações complexas. Assim o problema da manutenção de consistência de um sistema, que reside no controle da ocorrência dos eventos, fica subdividido em problemas menos complexos. Dentro do ambiente, tais módulos só se comunicam através de mensagens.

Estas mensagens dentro do ambiente lógico podem ser de tipos diferentes conforme a semântica associada no envio e recebimento das mesmas. Assim estas mensagens podem ser operações, exceções em execução, ou interrupções. As operações podem implementar desde o sistema de mensagens comum, até as chamadas remotas de procedure [AND 81]. Desta forma, dentro deste modelo é feito um tratamento homogêneo tanto para exceções como para a sincronização de processos.

4.2.4 O INTERRELACIONAMENTO ENTRE AMBIENTES

No caso do interrelacionamento entre dois ambientes computacionais é usado o conceito de interface computacional. Com isto o modelo permite a convivência de vários ambientes computacionais sobre o mesmo sistema, facilitando a implementação de um sistema distribuído com vários aplicativos diferentes. Por outro lado, no caso de relacionamento entre ambiente lógico e componente de ambiente físico, é usado o conceito de interface hardware. O relacionamento global entre todo o ambiente físico e todo o ambiente lógico é estabelecido na construção do ambiente computacional.

5. CONCLUSÕES

O presente trabalho constitui em um estudo preliminar que precedeu o projeto de uma ferramenta de suporte para sistemas distribuídos [KEG 84a, KEG 84b]. A abordagem do projeto é fornecer uma ferramenta básica onde os problemas básicos inerentes a sistemas distribuídos

já estejam resolvidos, de modo que a preocupação com sincronização e roteamento de mensagens assim como com o fornecimento de outros recursos, fica aliviada da implementação de uma aplicação em um sistema distribuído.

Uma das principais conseqüências desta modelagem preliminar é a introdução de recursos, na nova linguagem projetada, de descrever o meio físico (ou arquitetura de hardware) e a localização de módulos sobre esta arquitetura. A maioria das linguagens para sistemas distribuídos desenvolvidas atualmente não se preocupam com este detalhe. Isto é feito externamente, o que torna o sistema ainda mais complexo, pois não há um tratamento uniforme neste caso. Assim, não é uma tarefa trivial mudar uma aplicação de uma rede para outra, por exemplo.

O modelo de ambiente computacional fornece ao programador de sistemas dois pontos de vista do ambiente de programação: o ponto de vista físico e o ponto de vista lógico. Portanto o programador de um sistema distribuído deve ter condições de:

- a) Especificar a distribuição das unidades do programa.
- b) Estabelecer processos, programando as partes que podem ser executadas em paralelo.
- c) Desenvolver os mecanismos de comunicação entre os componentes do programa, projetando a política de sincronização entre os processos.
- d) Providenciar estratégias para o manuseio de exceções.

Estas características introduzem um novo estilo de programação, que é diferente dos já existentes, correspondentes a programação seqüencial e concorrente [TOB 82].

BIBLIOGRAFIA

- [ADV 80] ADVANCED COURSE ON DISTRIBUTED SYSTEMS ARCHITECTURE AND IMPLEMENTATIONS, München Mar. 4-13, 1980. Distributed systems architecture and implementation. Berlin, Springer-Verlag, 1981. (Lecture Notes in Computer Science, 105)
- [AND 81] ANDREWS, G.R. Synchronizing resources. ACM Trans. on Programming Languages and systems, 3(4):405-30, Oct. 1981.
- [AND 83] _____. & SCHNEIDER, F.B. Concepts and notations for concurrent programming. Computing Surveys, 15(1):3-43, Mar.1983.
- [DAT 81] DATA processing - Open systems interconnection - Basic reference model. Computer Networks, 5(2):81-118, Apr.1981.
- [FRA 81] FRANTA, W.R. et alii. Real time distributed computer systems. In: ADVANCES in Computer. New York, Academic Press, 1981. v. 20, p.40-82.

- |KEG 84a| KEGLEVICH DE BUZIN, P.F.W. Uma abordagem para a concepção e implementação de sistemas distribuídos. Porto Alegre, PGCC da UFRGS, 1984. (Dissertação de mestrado).
- |KEG 84b| _____. Constructor: projeto de uma ferramenta para descrição de computadores. Prêmio SERPRO de monografia âmbito nacional, primeiro lugar. Nov. 1984.
- |MCQ 80| McQUILLAN, J.M. Local network technology and lessons of history. Computer Networks, 4(5):235-8, Oct. 1980.
- |TOB 82| TOBIASCH, R. & RAFFLER. H. Configuratig software - a method for bridging over the gap between concurrent processing and distributing processing. In: INFOCOM 82, Nevada, Mar/Apr. 1982. Proceedings. s.n.t. p. 44.8