Guia da Cadeira de
COMPUTAÇÃO EM SISTEMAS DISTRIBUÍDOS

Mestrado em Engenharia Electrotécnica e de Computadores,
$2^{\underline{o}}$ Semestre

Prof. Luís E. T. Rodrigues

Secção de Sistemas Digitais e Computadores
Departamento de Engenharia Electrotécnica e de Computadores
Instituto Superior Técnico

Fevereiro de 1996

Nota prévia

O programa e organização da cadeira foi originalmente definido pelo Prof. Paulo Veríssimo. No fundamental, este guia segue as orientações definidas anteriormente.

1 Objectivos da Cadeira

Esta cadeira pretende instruir os alunos sobre os princípios da Computação Sistemas Distribuídos, com ênfase para modelos de sistemas distribuídos, confiaveis e tempo-real, para os mecanismos e paradigmas de interacção entre processos em computações multi-participante, algorítmica distribuída, transacções e gestão de dados replicados, memória partilhada distribuída.

Duas cadeiras no mestrado estão relacionadas em sequência lógica: Computação em Sistemas Distribuídos, e Comunicação em Sistemas Distribuídos. O programa está estruturado de modo a abranger as noções sistémicas da arquitectura de sistemas distribuídos e construção de aplicações sobre os mesmos. Nesse sentido, completa a formação em sistemas distribuídos dada no 4^o ano da LEIC (Sistemas Distribuídos, Sistemas Distribuídos em Ambientes Industriais), e em Sistemas Operativos Distribuídos no mestrado.

Distribuição, paralelismo e confiabilidade por tolerância a faltas tendem a ser indissociaveis, nos sistemas distribuídos modernos. O tempo-real, isto é, a capacidade de um sistema, de evoluir de acordo com especificações temporais (prazos para executar acções, instantes para ler entradas ou actuar sobre saídas), é um requisito cada vez mais frequente em aplicações distribuídas, despoletado em variados sectores como a automatização e gestão da produção, multimedia, trabalho cooperativo assistido por computador, etc.

Nesta cadeira pretende-se também perspectivar a tolerância a faltas, paralelismo e tempo-real sob a óptica dos sistemas distribuídos. Pretende-se assim expor os alunos ao contacto com paradigmas de programação não tradicionais, que aqueles não conheceram na licenciatura. Alguns deles, como programação orientada para grupos, baseiam-se em noções obtidas na cadeira de Comunicação em Sistemas Distribuídos.

2 Programa

Pretende-se estabelecer o seguinte fio condutor da matéria da cadeira:

- importância da distribuição; inevitabilidade da existência de atributos de tolerância a faltas e tempo-real, em maior ou menor grau, nos sistemas distribuídos evoluídos; o modelo e estruturação de sistemas distribuídos, com atributos de fiabilidade e pontualidade.

- exemplificar modelos de sistemas distribuídos, confiaveis e tempo-real;

- mostrar que, com o suporte de comunicação multi-participante (ou em grupo) adequado, se podem construir ambientes de desenvolvimento de aplicações que forneçam: algoritmos e protocolos para sistemas distribuídos que ultrapassem o mero contexto clássico de "comunicação" (como o envio de mensagens) ou de chamada a procedimento (RPC); o devido encapsulamento numa interface amigavel;

- como construir o suporte para a programação e execução daquelas aplicações é o objectivo principal da cadeira: RPC e grupos; transacções, dados replicados e memória distribuída; segurança.

1. A problemática dos sistemas distribuídos: revisão de conceitos (1 aula);

2. Noções e modelos de processamento distribuído aberto, tolerância a faltas em sistemas distribuídos (4 aulas):

   tolerância a faltas;

   processamento distribuído aberto;

modelo ANSA;

modelos ISIS e DELTA-4;

revisão de conceitos sobre comunicação multiparticipante;

filiação em grupos;

memória partilhada vs. passagem de mensagens;

3. Computação distribuída nos modelos cliente-servidor e orientado para grupos (2 aulas):

   gestão de actividades distribuídas; replicação; cooperação; competição;

   programação de actividades distribuídas utilizando o modelo cliente-servidor;

   a máquina de estado - vantagens/desvantagens; programação com máquina de estado: AtomicMaze, um jogo distribuído;

   breves exemplos de IPC: Unix BSd e V, SUNOS, ANSAware, Chorus, Mach, V-kernel/VMTP, x-Kernel, MARS, Autonet, LINDA/SNET;

   fiabilidade dos modelos de operações remotas e de difusão;

   programação com RPC - vantagens/desvantagens, RPC vs. grupos;

4. Métodos de construção de aplicações distribuídas (memória partilhada distribuída) (2 aulas):

   arquitecturas de memória partilhada distribuída (DSM): centralizada, migração, replicação em leitura, e em escr./leitura;

   protocolos de coerência de DSM: coerência estrita vs. frouxa; congruência na entrada vs. na saída; actualização vs. invalidação; actualização pronta vs. retardada;

5. Transacções e dados replicados) (5 aulas):

   transacções: as propriedades ACID; transacções sequenciais; arquitectura de uma transacção;

   transacções concorrentes: controlo da concorrência; trinco em duas fases; interbloqueio e aborto em cascata;

   transacções distribuídas: confirmação atómica (*atomic commitment*); janela de vulnerabilidade e 3P-commit;

   transacções replicadas: controlo da coerência; gestão de dados replicados; compromisso disponibilidade - congruência; serializabilidade 1-cópia; congruência forte - *read-1/write-all*, quora; congruência fraca e partições;

   Sistema de ficheiros;

6. Recuperação (1 aula):

   Salvaguardas coerentes. Salvaguardas sincronizadas e independentes.

7. Trabalho cooperativo distribuído (1 aula):

   Aplicações cooperativas;

   Impacto da distribuição; "awareness";

8. Sistemas distribuídos de tempo-real (2 aulas):

   Noções de tempo-real;

   Sistemas Mars de XPA;

9. Segurança em sistemas distribuídos (3 aulas):

   noções fundamentais;

   propriedades: confidencialidade, autenticidade, integridade, disponibilidade;

   técnicas de criptografia; canais seguros (revisão);

   controlo de acesso em sist. distribuídos;

   propriedades; tolerância a intrusões;

   metodologias de servidores de autenticação;

   aplicações de segurança;

10. Exemplos de distribuição usando vários sistemas e protocolos reais: ANSAware, ISIS, DELTA-4 pilots, x-Kernel, Highly Available Service da DEC-SRC, AAS da IBM, BANK'92, Munin, Arjuna (3 aulas);

3 Bibliografia

3.1 Referências Principais

ARCHITECTURE PROJECTS MANAGEMENT, Ltd, Cambridge, UK. *The ANSA Reference Manual*, release 1.1 edition, July 1989.

BIRMAN, K., & RENESSE, R., editors. *Reliable Distributed Computing With the ISIS Toolkit*. Number ISBN 0-8186-5342-6. IEEE CS Press, March 1994.

COMER, D.. *Internetworking With TCP/IP: Principles, Protocols, Architecture*. Prentice Hall, 1991.

HUTCHINSON, N. AND PETERSON, L.. Design of the x-Kernel. In *Proceedings of the SIGCOMM'88: Communications Architectures and Protocols*, Stanford, USA, August 1988. ACM.

MALLEY, S. AND LARRY PETERSON, L.. A new methodology for designing network software. Technical report, University of Arizona, Tucson, USA, 1990.

MULLENDER, S., editor. *Distributed Systems, 2nd Edition*. ACM-Press. Addison-Wesley, 1993.

PIMENTEL, J.. *Communication Networks for Manufacturing*. Prentice-Hall, 1990.

POWELL, D., editor. *Delta-4 - A Generic Architecture for Dependable Distributed Computing*. ESPRIT Research Reports. Springer Verlag, November 1991.

SANDERS, R.. The xpress transfer protocol (xtp)- a tutorial. Technical Report Computer Science Report TR-89-10, University of Virginia, University of Virginia, Charlottesville, Virginia 22903, January 1990.

STANKOVIC, J.. *Real-time Computing Systems: The Next Generation*. Technical Report TR-88-06, University of Massachussetts, Amherst, USA, January 1988.

VERÍSSIMO, P. & RODRIGUES, L.. Group Orientation: a Paradigm for Modern Distributed Systems. In *Proceedings of the 5th ACM SIGOPS European Workshop*, Mont Saint-Michel, France, September 1992. Extended and revised version as INESC RT/20-94.

4 Normas Gerais de funcionamento

Aulas

- Duas aulas teóricas semanais de 1.5 horas.

Destinam-se as aulas teóricas à exposição da matéria da cadeira. O material de apoio é constituído sempre que possível por livros de texto editados, e por alguns excertos de artigos, manuais e apontamentos.

Previsão do número de sessões:

- 24 aulas teóricas.

Regras de Funcionamento

- Parte Teórica

  É altamente desejavel a assistência às aulas teóricas, dado que nesta matéria inovadora a bibliografia é por vezes dispersa. Os alunos farão apresentações nas aulas teóricas que se pretende sejam de qualidade não só técnica mas também visual.

- Parte Laboratorial

  Existirão trabalhos para grupos de alunos. Os grupos de alunos (desejavelmente 2 alunos) servem tanto para a apresentação como para o trabalho. Os alunos manterão estes grupos até ao fim do semestre.

Avaliação

- Composição da nota:
  - Trabalho de laboratório - 50%;
  - Apresentações - 50%.
- Apresentações:

  Uma apresentação de 30 min. Apreciação e discussão durante a apresentação (pública, na aula).
- Trabalho:

  Visualização com discussão de 1/2 hora. Definição de notas individuais. Perguntas directamente relacionadas com o trabalho, e sobre a matéria referente ao trabalho em questão.

  Notas publicadas só ao fim de uma semana, por questões de harmonização de critérios.

  Chumbar no trabalho implica chumbar a cadeira.

5 Plano Pormenorizado e Sumários das Aulas (previsão)

Aula 1 A problemática dos sistemas distribuídos. *Caracterização, vantagens e desvantagens, complexidade, técnicas, requisitos do utilizador que procuram e justificam a distribuição.*

Aula 2 Tolerância a faltas. *Noções sobre tolerância a faltas.*

Aula 3 Processamento distribuído aberto, tolerância a faltas e tempo-real. *Noções sobre processamento distribuído aberto; modelo ANSA.*

Aula 4 Interligação. *Os problemas da interligação das máquinas.*

Aula 5 Suporte de comunicação. *Revisão de conceitos sobre comunicação em grupo.*

Aula 6 Replicação de actividades. *Replicação activa. Replicação passiva. Replicação semi-activa.*

Aula 7 Invocação remota. *Chamada a procedimentos remotos. Replicação.*

Aula 8 Memória partilhada distribuída. *Modelos de coêrencia.*

Aula 9 Memória partilhada distribuída. *Técnicas de concretização.*

Aula 10 Métodos de construção de aplicações distribuídas. Transacções: as propriedades ACID; transacções distribuídas; *atomic commit*.

Aula 11 Continuação da aula anterior.

Aula 12 Gestão de dados replicados. *Votação, primário-secundário, sistemas estáticos, sistemas dinâmicos.*

Aula 13 Continuação da aula anterior.

Aula 14 Gestão de dados replicados. *Sistemas de ficheiros.*

Aula 15 Recuperação em sistemas distribuídos. *Salvaguardas coerentes. Salvaguardas sincronizadas e independentes.*

Aula 16 Trabalho cooperativo em sistemas distribuídos. *Aplicações cooperativas; "awarness".*

Aula 17 Sistemas distribuídos de tempo-real. *Tempo-real; modelos XPA e MARS.*

Aula 18 Continuação da aula anterior.

Aula 19 Segurança em sistemas distribuídos. *Criptografia. Autenticação. Servidores.*

Aula 20 Continuação da aula anterior.

Aula 21 Continuação da aula anterior.

Aula 22 Exemplos de computação distribuída usando vários sistemas reais. *Apresentações por alunos durante 3 aulas, 1/2 hora cada.*

Aula 23 Continuação da aula anterior.

Aula 24 Continuação da aula anterior.

6 Referências

RICCIARDI, A.& BIRMAN, K.. Using process groups to implement failure detection in asynchronous environments. In *10th ACM Symposium on Principles of Distributed Computing*, Montreal - Canada, August 1991. ACM.

BIRMAN, K. & JOSEPH, T.. Reliable Communication in the Presence of Failures. *ACM, Transactions on Computer Systems*, 5(1), February 1987.

BIRMAN, K. SCHIPER, A. AND STEPHENSON, P.. Lightweight Causal and Atomic Group Multicast. *ACM Transactions on Computer Systems*, 9(3), August 1991.

BIRMAN, K. ET AL.. Maintaining consistency in distributed systems. Technical report, Cornell University, Ithaca, USA, 1991.

CARRIERO, N. & GELERTNER, D. The S/Net's Linda Kernel. *ACM Transactions on Computer Systems*, 4(2), May 1986.

F. CRISTIAN, AGHILI. H., R. STRONG, AND D. DOLEV. Atomic Broadcast: From simple message diffusion to Byzantine Agreement. In *Digest of Papers, The 15th International Symposium on Fault-Tolerant Computing*, IEEE, Ann Arbor-USA, June 1985.

CHANG, J. & MAXEMCHUCK, N.. Reliable broadcast protocols. *ACM, Transactions on Computer Systems*, 2(3), August 1984.

CHERITON, D.. VMTP: Versatile message transaction protocol. Technical Report Preliminary Version 0.7, Computer Science Department, Stanford University, February 1988.

CHERITON, D.. The V distributed system. Technical Report?3, Stanford University, March 1988.

COOPER, E.. Replicated procedure call. In *Proceedings of the 3rd ACM symposyum on Principles of Distributed Computing*, Berkeley, CA 94720, USA, August 1984. ACM.

COOPER, E. Replicated distributed programs. In *Proceedings of the 10th ACM Symposium on Operating Systems Principles*, Berkeley, California 94720, USA, November 1985. ACM.

COULOURIS, G. & DOLLIMORE, J.. *Distributed Systems, Concepts and Design*. Int'l Computer Science Series. Addison-Wesley, 1988.

CRISTIAN, F., AGHILI, H., STRONG, R. AND DOLEV, D.. Atomic Broadcast: From simple message diffusion to Byzantine Agreement. In *Digest of Papers, The 15th International Symposium on Fault-Tolerant Computing*, Ann Arbor-USA, June 1985. IEEE.

MACH NETWORKING GROUP. Network server design. Technical report, Carnegie-Mellon University, July 1988.

HUTCHINSON, N., MISHRA, S., PETERSON, L. AND THOMAS, V. Tools for implementing network protocols. Technical report, University of Arizona, Tucson, USA, June 1988.

HUTCHINSON, N. AND PETERSON, L.. The x-kernel: An architecture for implementing network protocols. Technical report, University of Arizona, Tucson, USA, 1990.

KANAKIA, H. AND CHERITON, D.. Universal network device interface protocol (UNDIP). In *Proceedings of the 13th Conference on Local Computer Networks*, Minneapolis, USA, October 1988.

KOPETZ, H. DAMM, A., KOZA, C., MULAZZANI, M., SCHWABL, W., SENFT, C. AND ZAINLINGER, R.. Distributed Fault-Tolerant Real-Time Systems: The Mars Approach. *IEEE Micro*, pages 25–41, February 1989.

LE LANN, G.. An analysis of different approaches to distributed computing. In *Proceedings of the 1st International Conference on Distributed Computing Systems*, Alabama-USA, 1979. IEEE.

LAMPORT, L.. Time, Clocks and the Ordering of Events in a Distributed System. *CACM*, 7(21), July 1978.

LAMPORT, L., SHOSTAK, R. AND PEASE, M. The Byzantine Generals Problem. *ACM Transactions on Prog. Lang. and Systems*, 4(3), July 1982.

LAMPORT, L.. Using Time Instead of Timeout for Fault-Tolerant Distributed Systems. *ACM Transactions on Prog. Lang. and Systems*, 6(2), April 1984.

MARQUES, J. & AND GUEDES, P.. *Fundamentos de Sistemas Operativos*. Editorial Presença, 1990.

MISHRA, S. & AND SCHLICHTING, R.. Abstractions for constructing dependable distributed systems. Technical Report TR 92-19, The University of Arizona, Departement of Computer Science, Tucson, Arizona, USA, 1992.

PETERSON, L., BUCHHOLDZ, N. AND SCHLICHTING, R.. Preserving and Using Context Information in Interprocess Communication. *ACM Transactions on Computer Systems*, 7(3), August 1989.

RODRIGUES, L. & VERÍSSIMO, P.. Reliable broadcast: a survey. *INESC Journal of R & D*, 1(1), June 1990. Also as INESC AR/39-90.

RODRIGUES, L. & VERÍSSIMO, P.. xAMp: a Multi-primitive Group Communications Service. In *Proceedings of the 11th Symposium on Reliable Distributed Systems*, Houston, Texas, October 1992. INESC AR/66-92.

ROSS, F.. An Overview of FDDI: The Fiber Distributed Data Interface. *IEEE J. on Selected Areas in Comm.*, 7(7), 1989.

ROSS, F., HAMSTRA, J. AND FINK, R. Fddi - a lan among mans. *ACM Sigcom 'CCR' (Computers Communications Review)*, 20(3), July 1990.

ROZIER, M. AND AL. Chorus distributed operating system. Technical Report CS/TR-88-7.6, Chorus Systemes, Paris, France, November 1988.

SALTZER, J. REED, D. AND CLARK, D.. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4), November 1984.

SCHNEIDER, F.. The state machine approach: a tutorial. In *Proceedings of the Workshop on Fault-tolerant Distributed Computing*, Lecture Notes in Computer Science. Springer-Verlag, 1988.

SCHNEIDER, F.. Understanding Protocols for Byzantine Clock Synchronization. Technical report, Cornell University, Ithaca, New York, August 1987.

SCHROEDER, M., BIRRELL, A., BURROWS, MURRAY, M., NEEDHAM, R., RODEHEFFER, SATTERTHWAITE, T., THACKER, C.. Autonet: a high-speed, self-configuring, local area network using point-to-point links. Technical Report?59, Digital,Systems Research Center, Palo Alto, California, April 1990.

SCHWARZ, R., & MATTERN, F.. Detecting causal relationships in distributed computations: In search of the holy grail. Technical report, Univ. Kaiserslautern, Kaiserslautern, Germany, February 1992.

STANKOVIC, J. & RAMAMRITHAM, K.. The Spring Kernel: A New Paradigm for Real-time Systems. *IEEE Software*, May 1991.

STEVENS, W.. *UNIX Network Programming*. Software Series. Prentice-Hall, 1990.

TANENBAUM, A. & MULLENDER, S. *AMOEBA, Collection of Papers*. Vrije University, 1991.

VERÍSSIMO, P. High-speed fibre-optic lan support for delta-4. Technical Report RT/108, Delta-4 Proj., INESC, Lisboa, Portugal, April 1988.

VERÍSSIMO, P. *Comunicação em Grupo Fiável, em Sistemas Distribuidos sobre Rede Local*. PhD thesis, Universidade Técnica - IST, Lisboa - Portugal, December 1989.

VERÍSSIMO, P., RUFINO, J., FONSECA, H., AND RODRIGUES, L. The performance of the xAMp protocol on token-bus and fddi nac's. Technical Report RT/109-91, INESC, Lisboa, Portugal, November 1991.

VERÍSSIMO, P. & RODRIGUES, L.. A posteriori Agreement for Fault-tolerant Clock Synchronization on Broadcast Networks. In *Digest of Papers, The 22th International Symposium on Fault-Tolerant Computing*, Boston - USA, July 1992. INESC AR/65-92.