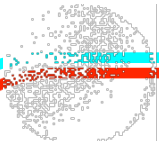


# Algoritmos Distribuídos

Lynch

Introdução



- **Autores**

- C. Geyer

- **Local**

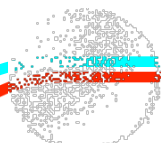
- Instituto de Informática

- UFRGS

- disciplinas:

- PDP - Programação Distribuída e Paralela

- versão V03 (2010)

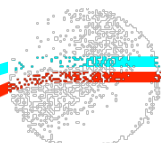


- **Essas transparências são baseadas (principalmente) em**

- Lynch, N. Distributed Algorithms. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996,

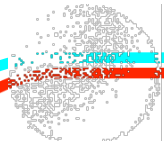
- **Bibliografia adicional**

- Barbosa, V. C. “An Introduction to Distributed Algorithms”. MIT Press, 1996.
- Raynal, M. “Distributed algorithms and protocols”. Wiley, Chichester, 1992.
- Raynal, M. and Helary, J-M. “Synchronization and Control of Distributed Systems and Programs”. Wiley, 1990.
- Tel, G. “Introduction to Distributed Algorithms”. Cambridge University Press, 2000 (second edition).



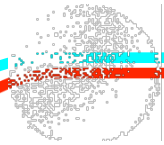
- **Bibliografia adicional:**

- R. Chow, T. Johnson. Distributed Operating Systems & Algorithms. 1997. Addison Wesley.
  - ❑ 2a parte somente sobre algoritmos de SOD
- Coulouris et al. Distributed Systems Concepts and Design. 4 edição.
  - ❑ Traduzida pela Bookman
  - ❑ preferido em muitos cursos de SD
  - ❑ algoritmos expressos em forma textual
- Andrews, G. Foundations of Multithreaded, Distributed and Parallel Programming. 2000.
- Tanenbaum, A. Distributed Systems.



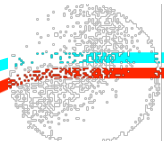
- **Súmula (desses slides)**

- conceitos de algoritmos distribuídos
- características de algoritmos distribuídos
- aspectos formais
- modelos formais
  - modelo síncrono
  - modelo assíncrono
  - modelo parcialmente síncrono
- estrutura do livro



- **Algoritmos Distribuídos (AD)**

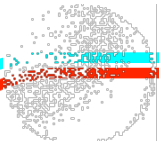
- uma grande variedade de algoritmos concorrentes
- originalmente
  - ❑ executados sobre muitos processadores distribuídos por larga área geográfica (redes largas)
- atualmente (também)
  - ❑ redes locais
  - ❑ multiprocessadores com memória compartilhada



- **Algoritmos Distribuídos (AD)**

- muitas aplicações

- ❑ Redes de computadores
- ❑ processamento distribuído de informações
- ❑ computação científica
- ❑ controle de processos em tempo-real
- ❑ tolerância a falhas
- ❑ sistemas operacionais distribuídos
- ❑ processamento P2P
- ❑ processamento em grade
- ❑ Suporte a jogos MMOG na Internet
- ❑ Suporte a programas paralelos (execução, ...)



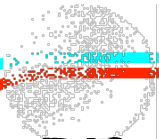
- **Atividades básicas**

- projeto ou concepção (design)
- implementação
- análise (complexidade, falhas, restrições, ...)

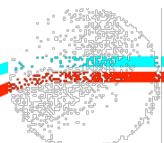


- **Diferenças entre AD**

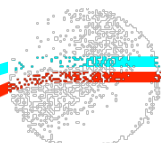
- método de IPC (InterProcess Communication)
  - ❑ memória compartilhada
  - ❑ troca de mensagens ponto-a-ponto
  - ❑ difusão de mensagens
  - ❑ chamada remota de procedures
- tempo
  - ❑ síncrono (totalmente)
  - ❑ assíncrono
  - ❑ parcialmente síncrono
    - ❑ limites de tempo
    - ❑ relógios aproximados
- modelo de falha



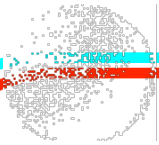
- **Diferenças entre AD (não Lynch)**
  - modelo de programa (?)
    - ❑ processos de código único (MPI)
    - ❑ OO distribuídos
    - ❑ módulos (Modula)
    - ❑ cliente/servidor (Java RMI, CORBA, ...)
    - ❑ ...
  - topologia da rede física
    - ❑ árvore
    - ❑ mesh (matriz, reticulado)
    - ❑ Hipercubo
    - ❑ Grafos randômicos (internet, ...)
  - Modelo de rede (comunicação)
    - ❑ Evolução afeta algoritmos ([Tel 2000])



- **Diferenças entre AD**
  - problema a resolver
    - ❑ alocação de recursos
    - ❑ comunicação
    - ❑ consenso
    - ❑ detecção de deadlock
    - ❑ outros



- **Diferenças entre AD (Lynch) e outros modelos como PRAM (= um tipo de Máquina Paralela Abstrata)**
  - número desconhecido de cpus
  - topologia desconhecida
  - entradas independentes em diferentes locais
  - vários programas concorrentes
    - ❑ iniciando em tempos diferentes
    - ❑ a velocidades diferentes
  - não determinismo do processador
  - tempos de entrega de mensagens variáveis
  - ordem de envio das mensagens desconhecida
  - falhas de comunicação e processadores



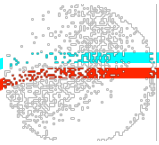
- **Resultados esperados pelo estudo**

- Lynch

- limites inferiores (piores) em complexidade
      - Por exemplo, quantidade de mensagens
    - resultados reais (implementações) serão sempre piores?
    - resultados impossíveis

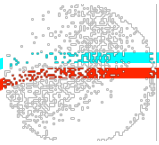
- Tel

- Caso médio; uso de distribuição de probabilidades

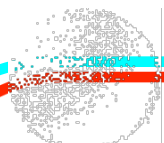


- **Framework**

- modelo baseado na teoria de autômatos
- não usa
  - ❑ linguagens de programação
  - ❑ lógica de prova formal



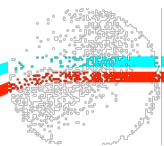
- **Rigorismo formal**
  - importante
  - evitar erros (enganos) fáceis
  - problema: usar rigorismo e ser didático



- **Modelo síncrono**

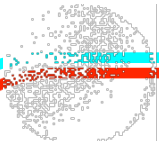
- + simples para descrever e usar
- componentes executam comandos em tempo uniforme
  - idem PRAM (algoritmos paralelos)
- resultados impossíveis do síncrono podem ser facilmente transportados ao modelo assíncrono
- freqüentemente
  - impossíveis ou muito ineficientes sobre sistemas reais



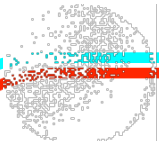


- **Modelo assíncrono**

- algoritmos genéricos, portáteis
- às vezes, não eficiente ou não resolve certo problema
- mais difícil de programar que o síncrono
  - devido não determinismo da ordem de eventos

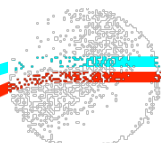


- **Modelo parcialmente síncrono**
  - restrições nos tempos dos eventos
  - tempo dos comandos não uniforme
  - mais reais
  - mais difíceis de programar
  - podem ser eficientes mas não confiáveis



- **Provas**

- asserções de invariantes
  - uso de indução no número de passos
- simulação
  - relação formal entre 1 par de sistemas
  - um: já provado; outro: a provar



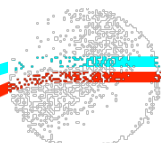
- **Estrutura modelo assíncrono**

- modelo redes

- ❑ sistemas send/receive (troca de mensagens ponto-a-ponto)
    - ❑ sistemas de difusão total (broadcast)
    - ❑ sistemas de difusão parcial (multicast)

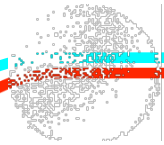
- algoritmos básicos

- ❑ eleição de líder em anel
    - ❑ eleição de líder em rede arbitrária
    - ❑ geração de “spanning tree”, difusão, coleta
    - ❑ pesquisa em largura, menor caminho
    - ❑ mínima “spanning tree”



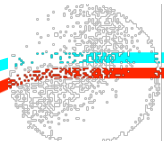
- **Estrutura modelo assíncrono**

- sincronizadores
  - técnicas para simplificar algoritmos em rede
- memória compartilhada X rede
  - técnica para simplificar (prova, desenvolvimento) algoritmos em rede
  - transformações nos dois sentidos
- tempo lógico
  - conceitos, implementação, aplicações
- fatos globais e propriedades estáveis
  - terminação
  - fatos globais consistentes



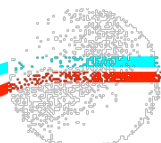
- **Estrutura modelo assíncrono**

- alocação de recursos em rede
  - exclusão mútua
  - alocação geral
- redes com falhas em processos
  - modelo, impossibilidade, detectores, consenso
- protocolos de comunicação de dados



- **Limitações do estudo de ADs**

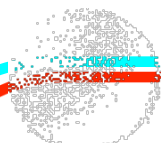
- não examina diversos aspectos importantes de um sistema distribuído (SD)
  - por exemplo: aspectos de hw, divisão em camadas e/ou módulos do sw, ...
- não examina diversas propriedades específicas de linguagens e bibliotecas de programação distribuída



- **Revisão**

- em que tipo de rede os ADs foram inicialmente usados?
- indique 3 pontos que distinguem os ADs dos algoritmos paralelos clássicos (por exemplo, para a PRAM)?
- porque é importante estudar os ADs?
- quais os 3 modelos de ADs da Lynch?





## ● Revisão

- qual a característica principal do modelo síncrono (MS)?
- quais as principais vantagens do MS?
- quais a principal desvantagem do MS?
- quais os 2 submodelos do modelo assíncrono?
- qual a principal vantagem do MA?
- qual a principal característica do modelo parcialmente síncrono?
- qual a principal vantagem do MPS?
- qual a principal desvantagem do MPS?