Confiabilidade: Segurança

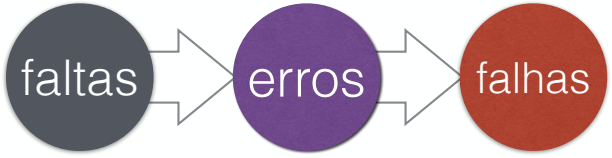
- **Autenticidade**
  - Os usuários comprovam suas **identidades**
  - senhas, chaves, etc.
- **Autorização**
  - estabelecimento de **controles de acesso** aos recursos
  - **listas** de controle de acesso
- **Privacidade**
  - As informações somente podem ser lidas por quem tiver direito.
  - mecanismos de **criptografia**
- **Integridade**
  - os dados não podem ser destruídos ou corrompidos por terceiros
- **Não-repudiação**
  - Todas as ações podem ser imputadas a seus **autores.**
  - mecanismos de **auditoria**

Confiabilidade: Tolerância a Faltas

- O que fazer em caso de **falha** de um servidor?
- Sistemas distribuídos podem ser projetados para **mascarar falhas**.
- Abordagens:
  - **replicação** de servidores
  - execução sem estado (***stateless*** *execution*)

Faltas, erros e falhas

- **Faltas**
  - **Situações incorretas** no estado interno de um sistema
  - Ex: um *bit* de memória inválido, um cabo de rede rompido
- **Erro**
  - **Decorrência da falta**
  - Estado interno incorreto do *software*
  - Ex: queda de uma conexão TCP, variável com valor errado
- **Falha**
  - **Decorrência do erro**
  - Serviço oferecido ao usuário não cumpre sua especificação
  - Ex: banco de dados fora do ar, aplicação mostrando dados incorretos
- Portanto:
  -



Desempenho

Aspectos de Projeto de Sistemas Distribuídos

Desempenho

- **Métricas** para medir desempenho:
  - **Número de mensagens** trocadas
  - **Tempo** de resposta
  - ***Throughput*** (número de tarefas executadas / tempo)
  - **Utilização** do sistema
- Em um sistema qualquer:
  - + processadores, + memória, + capacidade de armazenamento ⇒ **melhoria do desempenho**
- Ao se distribuir os processos entre os processadores na rede:
  - + **velocidade** final de computação?
  - + custo de **comunicação**!

Desempenho: Custos de Comunicação

- Componentes do **custo de comunicação**:
  - Tempo de **processamento** do protocolo/*middleware*
  - Tempo de **latência** do *hardware* e *software* de rede
  - Tempo de **transmissão** da mensagem
- Para obter um **bom desempenho**:
  - **Reduzir a comunicação** entre as entidades
  - Buscar manter um bom nível de **paralelismo**
  - Encontrar um ponto de **equilíbrio** entre ambos:
    - ***evitando: sobrecarga da máquina e comunicação excessiva***

Escalabilidade

Aspectos de Projeto de Sistemas Distribuídos

Escalabilidade

- Noção intuitiva:
  - Um sistema distribuído que opera bem com 10 máquinas também deve funcionar bem com 10.000 máquinas.
- O desempenho do sistema não deve ser **degradado de forma acelerada** à medida em que o número de processos distribuídos cresce.

Inimigos da Escalabilidade

- Componentes **centralizados**
  - por exemplo, um único servidor para todos os usuários
- Tabelas/BD **centralizadas**
  - por exemplo, um único arquivo com as informações acessadas pelo servidor
- Algoritmos **centralizados**
  - por exemplo, o roteamento de mensagens baseado em informações completas de caminho

Melhorando a Escalabilidade

- **Algoritmos descentralizados** com as seguintes características:
  - Nenhuma máquina possui **informações completas** sobre o estado do sistema.
  - Máquinas tomam decisões baseadas apenas nas informações disponíveis **localmente**.
  - Falha de **uma das máquinas** não impede o funcionamento do algoritmo.
  - Não existe um **relógio global** implícito.