

**Aluno: André Luiz N. Carneiro De Castro 92854**

## **Sistemas Computacionais Distribuídos**



Segunda Avaliação  
Prof. Klausner

---

FUNDAÇÃO HERMÍNIO OMETTO

**ARARAS/SP  
06/2021**

**Q1)** *Explique quais soluções de Nomeação Simples podem ser utilizadas Sistemas Distribuídos, citando um exemplo prático de cada.*

Para nomeação simples, temos uma certa limitação dada que as técnicas que podem ser utilizadas para resolução de problemas só funcionam bem em redes locais, porém acabam tendo problemas de escalabilidade em broadcasting, multicasting e ponteiros repassadores.

**Broadcasting** contém uma mensagem que possui um identificador e é enviada para cada computador da rede, do qual, depois dela ter sido espalhada para todas as entidades, cada entidade deve verificar se existe tal informação para assim ser enviado o recurso. Broadcasting tem problema quando a banda cresce (aumento da probabilidade de colisões de mensagem) do qual gera desperdício. Também distribui a mensagem para todas as entidades, de tal forma que se identificar, avisa tal identificação para que seja enviado o endereço para ter acessibilidade. Assim sendo, ela é uma técnica muito deficiente, pois só funciona bem em uma rede local, consome muito da rede gerando desperdício e tem problema de escalabilidade.

Já no **multicasting**, diferente do broadcasting que envia a mensagem para todos na rede local, ele cria um grupo restrito de host's que direciona para pessoas/recursos específicos, tendo um uso menor de rede e diminuindo o tráfego. Ele também é utilizado para associar réplicas e localizar a réplica mais próxima.

Temos os **ponteiros repassadores**, do qual como o próprio nome fala, temos uma entidade que foi posicionada em um determinado local e ela se move para outro local (tendo uma movimentação da entidade), nessa técnica de ponteiro deixamos uma marca na identificação que seria o ponto A ("de partida") apontando para a nova localização no ponto B ("ponto de chegada"), ou seja, o recurso se move de A para B, guardando o apontamento/endereçamento do qual onde ele foi após a mudança, podendo acontecer várias vezes o mesmo procedimento.

Além dos ponteiros repassadores, temos a técnica de **localização nativa** que permite o acerto da localização da referência que foi perdida. A **tabela de hash (dht)** toda vez que temos uma mudança, vamos sempre guardar o antecessor e o sucessor, fazendo as combinações de hash para obter a localização do objeto dentro da estrutura do sistema distribuído, o sistema chord é um exemplo de hash. E por ultimo temos as **implementações hierárquicas** que consiste na combinação de nomeações estruturadas e nomeações baseadas em atributos. A organização dela é dada de tal forma por registros ou entradas de diretório e composto por um conjunto de pares.

**Q2)** *Pensando em Sincronização em Sistemas Distribuídos, explique como funcionam os relógios físicos e lógicos.*

**Relógios lógicos** referem-se à implementação de um protocolo em todas as máquinas em seu sistema distribuído, de modo que as máquinas sejam capazes de manter a ordenação consistente de eventos em algum intervalo de tempo virtual. Um relógio lógico é um mecanismo para capturar relações cronológicas e causais em um sistema distribuído. Os sistemas distribuídos podem não ter relógio global fisicamente síncrono, portanto, um relógio lógico permite a ordenação global de eventos de diferentes processos em tais sistemas.

Exemplo: Se resolvermos viajar, fazemos um plano completo para qual lugar devemos ir primeiro. Sempre mantemos o procedimento ou uma organização planejada anteriormente. De maneira semelhante, devemos fazer as operações em nossos PCs, uma a uma, de maneira organizada.

Ou seja, supondo que temos mais de 10 pc's em um sistema distribuído, para realizar o procedimento de todos funcionarem juntos e em sincronia, precisamos de um relógio lógico.

Agora o **relógio físico** para muitos sistemas é muito desejável, sendo utilizado para ordenação temporal de eventos produzidos por processos concorrentes, sejam eles: Sincronização entre remetentes e receptores de mensagens, coordenação de atividade conjunta, serialização de acesso simultâneo para compartilhamento de objetos.

O controle de um relógio físico é feito através de um cristal de quartzo, que temos um contador e um registrador de retenção. Cada volta do relógio temos uma interrupção e a cada ciclo do relógio é somada a hora armazenada, mantendo-o sempre atualizado.

Tendo a estrutura de Berkeley sendo a mais conhecida, o servidor consulta as máquinas de tempos em tempos, ou seja, cada máquina ou cada nó da rede em questão é consultado e é extraído o tempo de cada máquina. Então o algoritmo gera uma média de todas as horas geradas e informa todos os computadores o deslocamento de tempo a ser feito.

Ele possui alguns problemas, um deles é que o tempo do relógio não pode retroagir, assim diminuimos a frequência do clock. Também existem os delays da rede, do qual dificulta 100% a sincronização envolvendo os relógios físicos e não realizam a parcialidade de tempo com exatidão no horário de outras máquinas, entretanto, são eficientes em sistemas de tempo real.

**Q3)** *Explique detalhadamente as estratégias de Tolerância a Falhas, citando um exemplo prático de cada.*

Temos a **resiliência de processos** onde temos adaptação, flexibilização das mudanças decorrentes, da replicação dos processos em grupos e grupos simples ou hierárquicos, preocupando-se em criar mecanismos para auxiliar processos, tentando fazer com que os mesmos sobrevivam a eventuais falhas. Uma forma de tornar um processo resiliente é a utilização de processos redundantes (redundância física). Além de protocolos serem utilizados, técnicas também são utilizadas, sejam elas tanto no hardware quanto no software, para evitar falhas no sistema.

Na **comunicação confiável cliente-servidor**, falamos de uma comunicação que deve ocorrer de forma eficiente, evitando falhas de comunicação. Elas são abstrações de comunicação que permitem que os vários processos de um programa distribuído, geralmente modelados como clientes e servidores, se comuniquem entre si através dos limites da máquina. Um requisito fundamental das abstrações é mascarar as falhas de máquina e comunicação que podem ocorrer durante as comunicações cliente-servidor.

Quando um cliente de banco acessa serviços de banco on-line com um navegador da web (o cliente), o cliente inicia uma solicitação ao servidor da web do banco. As credenciais de login do cliente podem ser armazenadas em um banco de dados e o servidor da web acessa o servidor de banco de dados como um cliente, com isso deve-se atentar na confiabilidade de conectividade.

Na **Comunicação confiável de grupo** acontece quando um processo de origem tenta se comunicar com vários processos ao mesmo tempo. Um grupo é uma coleção de processos interconectados com abstração. Essa abstração é para ocultar a passagem da mensagem para que a comunicação pareça uma chamada de procedimento normal. A comunicação em grupo também ajuda os processos de diferentes hosts a trabalharem juntos e realizarem operações de maneira sincronizada, aumentando, portanto, o desempenho geral do sistema. Podemos ter: *comunicação de difusão, comunicação multicast e comunicação unicast.*

**Comprometimento distribuído** isso acontece em relação a entrega de mensagens, ou acontece com cada membro do grupo ou não acontece. Não adianta falar de um envio de uma mensagem se não tiver o recebimento dele então para que isso aconteça, devemos enviar a mensagem e ela deve ser recebida pelo receptor. Existem os comprometimentos de uma fase, duas fases e três fases (do qual é mais difícil lidar com a falha sistêmica).