

MI - Concorrência e Conectividade

Acertando os Ponteiros

Vinicius Pereira Santana¹

¹Graduando em Engenharia de Computação.

Universidade Estadual de Feira de Santana (UEFS), Feira de Santana, Brasil.

vpsantana@ecomp.uefs.br

Abstract.

Resumo.

1. Introdução

Na modernidade, tempo tornou-se um elemento precioso. Não obstante, sua noção é bastante difusa e, por diversas vezes, é tema de discussões filosóficas e alvo de pesquisas na Física. O que se sabe é que ele norteia a vida das sociedades e constitui-se como moeda de troca nas relações de trabalho.

Nos sistemas computacionais não é diferente. Ter uma referência para a execução das tarefas é importante. Essas tarefas podem ser das mais variadas: desde a criação de um arquivo até a execução de um *back-up* agendado, por exemplo. Entretanto, por um aspecto construtivo, dispositivos diferentes (ou até mesmo iguais) podem mensurar os instantes de maneira diferente e, com o passar do tempo, naturalmente atrasar ou adiantar. Assim, é preciso buscar maneiras confiáveis de manter a sincronização dos relógios.

2. Desenvolvimento

2.1. Sistemas Distribuídos

A utilização dos computadores é vital para o funcionamento e realização de diversas atividades. Deste modo, diversas arquiteturas

Coulouris define um sistema distribuído cujos componentes localizam-se em uma rede de computadores e coordenam suas ações por meio da troca de mensagens. Por exemplo, inclui-se os jogos *multiplayer* online e os sistemas de buscas web. Esta definição consegue caracterizar bem a maioria dos sistemas existentes e faz menção às características de concorrência no uso dos componentes, tolerância às falhas e ausência de um relógio comum.

A concorrência no uso dos componentes é uma realidade quase que inevitável em uma rede de computadores. Se um grupo de estudantes, nas suas residências, compartilham arquivos e escrevem o trabalho simultaneamente, é importante que o sistema tenha a capacidade de coordenar as ações da melhor forma.

Falhas podem acontecer a qualquer instante e é importante que seja dimensionado três itens: identificação das falhas, ações de emergência e os impactos. Assim, por exemplo, mensagens podem não ser recebidas ou atrasarem e os processos podem finalizar de maneira inesperada (*crash*). Em qualquer situação, o sistema deve continuar funcionando corretamente.

2.2. Sincronização em Sistemas Distribuídos

2.3.

3. Resultados

4. Conclusões

Referências

Caelum (2017). Java e Orientação a Objetos - Curso FJ - 11.
www.caelum.com.br/apostilas.