

# GCC129 - Sistemas Distribuídos

## Relatório Técnico 1

### **Desafios na Implementação de Aplicações Distribuídas**

Nome: Mateus Carvalho Gonçalves

Turma: 10A

#### 1. Introdução

Tanenbaum define um sistema distribuído como “um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”. Essa definição engloba importantes aspectos, porém uma pequena parte merece destaque: com o desenvolvimento de tecnologias de rede, IA e IoT, esses computadores independentes seriam melhor ilustrados por “componentes independentes”, que também podem ser sensores, câmeras, entre outros, que trocam mensagens entre si para o funcionamento do sistema.

Uma rede social qualquer, como o Facebook, é um sistema distribuído que troca mensagens entre milhões de pessoas no mundo. Forma uma grande rede de troca de mensagens entre os aplicativos/páginas web e os servidores, entre outros, para funcionar.

Para suportar computadores e redes heterogêneas utiliza-se uma camada de software, o middleware, que conecta os dispositivos em um sistema distribuído. Ele é necessário para fazer a interface entre as várias máquinas e aplicações desse sistema e mantê-lo consistente, oferecendo assim uma visão de sistema único.

Busca-se cumprir algumas metas ou requisitos para garantir a viabilidade e qualidade de um sistema distribuído, são esses: oferecer fácil acesso a seus recursos; ocultar razoavelmente bem o fato de que os recursos são distribuídos por uma rede (transparência); deve ser aberto (uso de padrões) e deve poder ser expandido facilmente (escalabilidade).

A principal meta de um sistema distribuído é facilitar aos usuários e às aplicações o acesso a recursos remotos. Os avanços de conectividade e recursos têm proporcionado que pessoas muito distantes geograficamente possam trabalhar por meio de groupware - software para edição colaborativa, teleconferência, etc. - como as ferramentas Google, Figma e Meet, por exemplo.

A transparência visa melhorar a experiência do usuário ao passo que eles não precisam se preocupar com informações não relevantes à eles e garantir que o

sistema seja visto como único e coerente. Esse requisito é avaliado em sete formas: acesso, localização, migração, relocação, replicação, concorrência e falha.

A abertura de um sistema é medida pelo uso de padrões computacionais estudados e definidos por entidades internacionais, como os padrões web por exemplo. Isso garante que o sistema possa se comunicar facilmente com outros sempre que necessário (interoperabilidade) e também a portabilidade de aplicações.

Um sistema escalável é aquele que está preparado para lidar facilmente com mudanças, sejam expansões ou retrações. Essas mudanças podem estar ligadas à demanda de acesso, distância física e ao número de domínios administrativos.

Voltando ao exemplo do Facebook, são fatores que mostram que esse sistema cumpre (pelo menos em alguns fatores) os requisitos citados acima: possui facilidade de acesso e de uso; é transparente, uma vez que os usuários não precisam se preocupar com questões sobre onde, geograficamente, estão os dados que estão consumindo; usa padrões abertos para se conectar a todas as plataformas (smartphones e computadores, navegadores web ou aplicativos, iOS, Android, Windows, Linux, etc); e é escalável ao passo que está preparado para o aumento de usuários.

Por fim, por serem sistemas complexos e descentralizados, existem vários desafios a serem considerados no desenvolvimento, alguns deles serão discutidos na seção 2.

## 2. Principais Desafios

### 2.1 Concorrência e consistência

Servidores e bancos de dados desses sistemas devem estar preparados para atender um grande número de requisições. Geralmente, um sistema grande, que responde várias requisições tem múltiplos servidores e bancos de dados replicados, dessa forma são necessários algoritmos para controlar o fluxo de requisições e também para garantir a consistência dos dados em todas as réplicas de um banco.

Um exemplo é o uso colaborativo de um arquivo no Google Documentos. Todos os colaboradores, em qualquer lugar do mundo que estejam, devem conseguir acompanhar a edição em tempo real. Também, se um servidor de uma rede social está sobrecarregado ou cai, deve-se direcionar as novas requisições para outro servidor ativo.

### 2.1 Desempenho, confiabilidade e disponibilidade

Um aspecto importante ao projetar sistemas distribuídos é a relação entre desempenho, confiabilidade e disponibilidade. Deve-se avaliar qual modelo é mais adequado para a finalidade do componente ou do próprio sistema como um todo.

Em geral, um sistema deve oferecer um desempenho razoável, ter baixa tolerância a falhas ou ter uma boa recuperação, e uma boa disponibilidade. Entretanto, alguns sistemas de pesquisa dependem de alto desempenho, já sistemas de e-commerce devem ser extremamente intolerantes à indisponibilidade, mantendo funcionamento quase 100% do tempo.

## 2.1 Heterogeneidade de componentes

Atualmente, vivenciamos uma variedade imensa de plataformas e hardware na computação, logo, percebe-se a necessidade de uma boa portabilidade. O uso de padrões assiste para que a maior parte das aplicações funcione na maior diversidade de hardware e sistemas operacionais.

Além disso, quando trata-se de clusters, por exemplo, é imprescindível que o sistema esteja preparado no caso de haver máquinas com especificações e sistemas operacionais diferentes. Por exemplo: sistemas operacionais podem ter comandos e padrões de dados diferentes; também, se um servidor possui duas máquinas, uma possui hardware suficiente para atender a 10 requisições em um período de tempo e a outra 50 requisições, é inviável que o sistema opere dividindo igualmente a carga de trabalho entre as duas máquinas.