

# SISTEMAS DISTRIBUIDOS

KÉLVYN COSTA<sup>1</sup>, MICHEL ALMORIN MARTINS<sup>2</sup>

Universidade Vale do Rio do Sinos (UNISINOS)  
CEP – 93.022-000 – São Leopoldo– RS – Brasil

[kelvyn.costa@gmail.com](mailto:kelvyn.costa@gmail.com), [michelalmorinmartins@hotmail.com](mailto:michelalmorinmartins@hotmail.com)

**Abstract.** *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

**Resumo.** *Este artigo visa descrever o que são os sistemas distribuídos, bem como suas funcionalidades. Como são feitas as distribuições de processos para que sejam executados da maneira mais compartilhada possível para que se tenha o desempenho desejado, através do compartilhamento de hardware, software e dados. Será também relatado um experimento feito para demonstração desses sistemas.*

## 1. Introdução

Segundo Tanenbaum, sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas por trocas de mensagens. Com essa simples definição em mente, é possível abranger toda a parte de sistemas em que computadores conectados em rede possam ser distribuídos de uma maneira útil.

Não é necessário que os computadores estejam localizados em um mesmo ambiente físico, distância não é um delimitador para esse tipo de sistema, podendo até mesmo estar em continentes separados, como também podem estar no mesmo prédio ou na mesma sala. Com esse conceito definido, a definição de sistemas distribuídos tem como consequência:

- Concorrência: programas executando concorrentemente é uma norma dentro de uma rede de computadores. Um trabalho pode estar sendo feito em um computador, ao mesmo tempo em que outro é feito em uma máquina diferente, compartilhando recursos como arquivos ou até mesmo uma página web quando necessário. A capacidade que o sistema possui de manipular recursos que estão em compartilhamento são facilmente ampliados com a adição de mais recursos (computadores, por exemplo) na rede.
- Inexistência de relógio global: quando é necessário a cooperação entre programas, suas ações são coordenadas através de trocas de mensagens. É necessário haver uma noção compartilhada do tempo em que as ações dos programas acontecem. É possível verificar que limites para a precisão com o qual computadores podem sincronizar seus relógios em uma rede existem, não existindo assim uma noção global única do tempo preciso.

- Falhas independentes: Qualquer sistema de computador é suscetível a falhas e cabe aos projetistas dos sistemas ponderar as consequências dessas possíveis falhas. No caso dos sistemas distribuídos, as falhas tem consequências diferentes. Falhas na rede irão resultar no completo isolamento dos computadores que a ela estão conectados, mas não quer dizer que os mesmos irão parar de funcionar. É provável que os programas que neles existem, não consigam distinguir se a rede se tornou extremamente lenta ou se a mesma simplesmente falhou. Cada componente de um sistema pode falhar por si só, o que não impede que os outros componentes continuem em funcionamento.

O estímulo de se construir um sistema distribuído é devido a vontade de compartilhar um mesmo recurso. Apesar de ser um termo bem abstrato, “recurso” define bem o conjunto de coisas que poderão ser compartilhados de uma maneira vantajosa em um sistema interligado através de uma rede. Este termo pode abranger desde componentes de hardware, como discos de armazenamento, até arquivos de quaisquer tipos.

## 2. Desafios na implementação de sistemas distribuídos

A implementação de um sistema distribuído deve atender a uma série de requisitos que garanta seu funcionamento pleno a maior parte do seu tempo, sendo eles:

- Heterogeneidade: Um sistema distribuído pode possuir diferentes tipos de rede, diferentes tipos de hardware (diferentes tipos para representação de dados, código de máquina diferentes), sistemas operativos diferentes e linguagens de programação diferentes. Para resolver esse problema, é necessário definir uma camada de software intermediária nomeada *middleware*, essa camada faz a mediação entre o software e as demais aplicações, é utilizada para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação.
- Escalabilidade: Um sistema distribuído deve ser capaz de se manter funcional de forma correta independentemente do número de usuários (outros sistemas) e sem que haja impactos na performance quando necessário adicionar novos. É necessário desenvolver o software de forma que o aumento de usuários não impacte em grandes alterações no sistema.
- Sistema Aberto: Um sistema distribuído deve ser capaz de ser extensível, tanto em hardware quanto em software, quando necessário adicionar novos componentes ao sistema, deve ser feito sem que comprometa o funcionamento dos já existentes e que haja comunicação com os antigos.
- Tratamento de falhas: Na ocorrência de falhas, o sistema deve ser capaz de contê-las sem que comprometa o seu funcionamento. Se não houve resposta a um pedido, deve-se ser capaz de distinguir se isso ocorreu devido a uma falha na rede ou no nó do destino. Certas falhas podem ser ocultadas se houver redundância para isso. Se uma mensagem não é entregue, ela pode ser retransmitida, um arquivo pode ser escrito em duplicidade (em discos diferentes, por exemplo), a cada dois roteadores, é necessário existir dois percursos diferentes, bancos de dados podem ser replicados em vários servidores.
- Segurança: Garantir a segurança de recursos computacionais significa manter o nível de confidencialidade exigido pelos usuários (proteger contra acessos sem

autorização), manter a integridade dos dados (proteger contra a alteração ou corrupção de informações) e manter a disponibilidade do sistema, ou seja, garantir que não haja interferências aos meios de acesso aos recursos.

- **Transparência:** Um sistema distribuído deve ser visto como um todo e não como um conjunto de componentes distribuídos. Dentro do desafio da transparência, pode-se separá-la em:
  - **Acesso:** recursos locais e recursos remotos podem ser acessados utilizando as mesmas operações.
  - **Localização:** recursos podem ser acessados sem o conhecimento da sua localização.
  - **Concorrência:** clientes de um componente não precisam saber o estado de concorrência do mesmo.
  - **Transparência de replicação:** clientes de um componente não precisam saber se existe uma replicação e se estão utilizando uma réplica ou o original.
  - **Falhas:** o sistema deve funcionar na presença de falhas de hardware e seus usuários e não necessitam ter conhecimento de como a falha foi ou será resolvida.
  - **Migração:** recursos podem mudar de localização sem que isso afete a sua utilização.
  - **Desempenho:** o sistema pode ser reconfigurado para melhorar seu desempenho sem a percepção de seus usuários.
  - **Escalabilidade:** o sistema pode ser expandido sem que seus usuários tenham conhecimento disso.

### **3. Exemplos**

Alguns exemplos de sistemas distribuídos, que mostram de uma forma clara o que é essa forma de compartilhamento de processamento.

#### **3.1 Pesquisa na Web**

As pesquisas na World Wide Web são um excelente exemplo de sistemas distribuídos, devido ao fato de que todo o conteúdo pesquisado, passa por diversos locais e é processado por diversos servidores trabalhando em conjunto para proporcionar os conteúdos.

O Google é a prova disso, sendo o líder de pesquisas na web, fez um trabalho muito forte com sistemas distribuídos, diversas estações espalhadas pelo mundo contendo vários computadores trabalhando em rede para dar acesso rápido e seguro à arquivos de diversos tamanhos. Sistema de arquivos distribuídos para dar suporte aos seus aplicativos como Google Earth e Drive. Um sistema de gerenciamento de cálculos paralelos e distribuídos em suas estações, para acelerar o processamento.

#### **3.2 Programa SETI - Search for Extraterrestrial Intelligence**

Esse programa tem como objetivo encontrar vida extraterrestre no espaço. Um dos projetos paralelos baseados no SETI, é o SETI@HOME, que como o próprio nome indica, é um sistema que utiliza os dados coletados pelo SETI, dividindo em partes que possam ser analisadas por computadores comuns. Ao se conectar na internet, o usuário cadastrado carrega em seu computador dados fornecidos por radiotelescópios, e durante o tempo livre do processador, são analisados para posteriormente serem enviados para o controle do projeto.

### **3.3 Plataforma BOINC - Berkeley Open Infrastructure For Network Computing**

É uma plataforma em framework, que tem como objetivo, facilitar a computação voluntária através de sistemas distribuídos. Foi desenvolvido originalmente fora do projeto SETI@HOME, mas que tem seu objetivo voltado para projetos do mesmo segmento do SETI.

Ele serve basicamente para tornar mais rápido e possível, processamentos que levariam muito tempo para serem executados, como por exemplo, na astrofísica ou Climatologia. Com a possibilidade de usar certa parte do poder de processamento dos computadores pessoais dos usuários cadastrados, de uma forma que não deixe o computador lento.

## **5. Conclusão**

Conclui-se através desse artigo que os Sistemas distribuídos tem papel fundamental no processamento compartilhado. A possibilidade de compartilhamento via rede de processamentos que demorariam muito tempo em apenas uma CPU, tornam sistemas como o SO Amoeba, possíveis para facilitar o uso de softwares mais robustos.

## **6. Referências**

- Sistemas Distribuídos – 5ed: Conceitos e Projeto – George Coulouris, Jean Dollimore, Tim Kindberg e Gordon Blair;
- Sistemas Distribuídos – Wikipédia;
- Distributed operating systems - Andrew S. Tanenbaum, Robbert Van Renesse;