

PROCESSOS

DCE540 - Computação Paralela e Distribuída

Atualizado em: 26 de julho de 2021

Iago Carvalho

Departamento de Ciência da Computação



A definição de **processo** vem da área de sistemas operacionais

- Um programa em execução
- Gerenciar processos é uma das tarefas mais importantes de um sistema operacional

A definição de processo no conceito de computação paralela e distribuída é um pouco diferente

- Threads
- Multi-core
- Multi-processadores
- Virtualização

Em um sistema distribuído, um processo não pertence exclusivamente a um sistema operacional

- Um processo pode ser movido de um nó para outro
 - Escalabilidade
 - Tolerância a falhas
 - Balanceamento de carga

Para falarmos de processos, primeiramente devemos apresentar o conceito de **threads**

- Um processo em um sistema distribuído pode ser dividido em múltiplas threads

Threads também são chamadas de *processos leves*

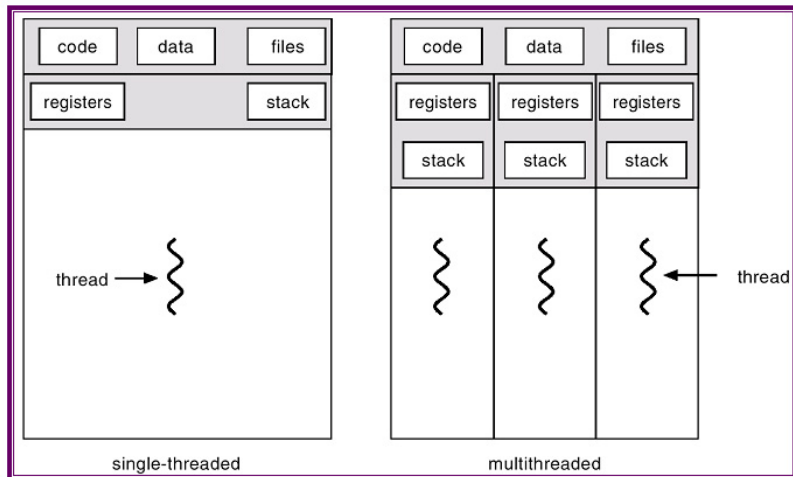
- Sistema operacional faz o gerenciamento de threads de forma mais simples e rápida
- Seu contexto é muito menor do que o de um processo
- Menor segurança
 - Segurança é de responsabilidade do desenvolvedor/programador

Existe um grande número de outras diferenças entre processos e threads, como a seguir

PROCESSO VS THREAD

Processo	Thread
Processo pesado	Processo leve
Depende do sistema operacional para realizar troca entre processos	Independente do sistema operacional para realizar troca entre threads
Processos são independentes	Uma thread pode alterar outras
Possui memória e arquivos próprios	Compartilham memória e arquivos
Pesado para ser criado	Leve para ser criado
Criação depende de chamada de sistema	Criação independente de chamada de sistema
Utiliza grande quantidade de recursos	Utiliza uma menor quantidade de recursos
Se um processo é bloqueado, os processos relacionados também são	Se uma thread é bloqueada, outras threads relacionadas ainda estão livres

PROCESSO MONO VS MULTI-THREAD



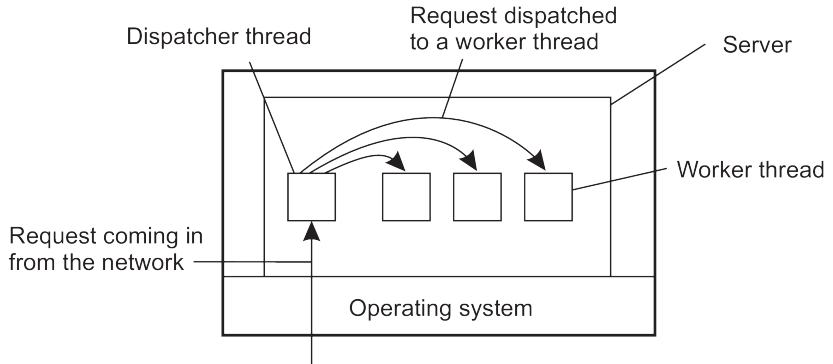
Uma das propriedades mais importantes de threads em sistemas distribuídos é sua capacidade de ser bloqueada sem que o processo inteiro seja também bloqueado

- Maneira simples de implementar comunicação assíncrona

Outra aplicação de threads é no desenvolvimento de servidores

- Servidores podem receber um grande número de requisições
- Cada requisição cria uma thread
 - Requisições são tratadas de forma paralela

SERVIDOR MULTI-THREAD

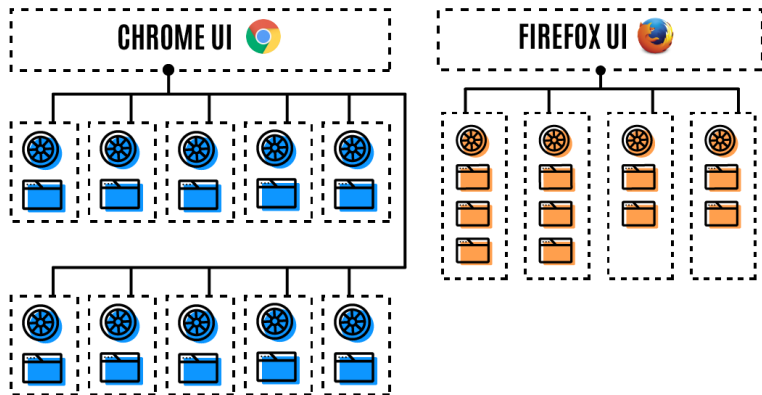


Além de servidores, também é interessante que clientes sejam implementados utilizando o conceito de multi-thread

- Realizar comunicação assíncrona
- Utiliza a natureza não-blocante das threads

Exemplo de aplicação: navegador *web*

NAVEGADORES *WEB*: GERENCIAMENTO DE JANELAS



NAVEGADORES WEB: REQUISIÇÕES ASSÍNCRONAS

```
<html>
  <head>...</head>
  <body background="http://164.125.34.198:5110/static/background.png?
etag=LlFYDas_">
    <p>There are no articles in the blog</p>
    <form method="post" enctype="application/x-www-form-urlencoded">
      <input type="hidden" name="_token" value="oK332zTXsw">
      <div class="required">
        <label for="hident2">Article Title</label>
        <input id="hident2" name="f1" type="text" required value>
      </div>
      <div class="required">
        <label for="hident3">Article Content</label>
        <div unselectable="on" style="width: 600px;">...</div>
        <div style="width: 600px; border-width: 0px 1px 1px; border-top-
style: initial; border-right-style: solid; border-bottom-style:
solid; border-left-style: solid; border-top-color: initial; border-
right-color: rgb(204, 204, 204); border-bottom-color: rgb(204, 204,
204); border-left-color: rgb(204, 204, 204); border-image: initial;
overflow-y: auto; overflow-x: hidden;">...</div>
        <textarea class="html" id="hident3" name="f2" style="display:
none;"></textarea>
      </div>
      <div>
        <input type="submit" value="Publish to blog">
      </div>
    </form>
    <script src="http://js.nicedit.com/nicEdit-latest.js"></script>
    <script>
      (function(){new nicEditor({fullPanel:true}).panelInstance("hident3")})
      ();
    </script>
  </body>
</html>
```

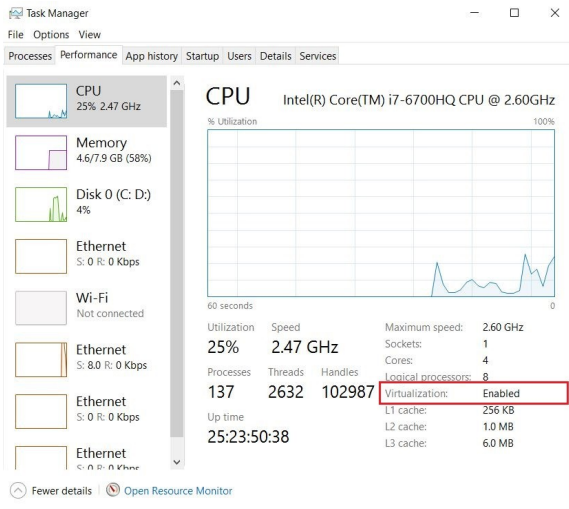
A ideia de utilizar diversas threads é executar múltiplas tarefas em paralelo.

Entretanto, em um único processador, somente uma única instrução (ou algumas, em casos de pipeline) pode ser realizada por clock

- É criada a ilusão de paralelismo através da troca extremamente rápida entre threads (ou processos)

A essa ilusão de processamento paralelo, nós damos o nome de *virtualização* de recursos

VIRTUALIZAÇÃO EM PROCESSADORES



Além disso, existem três usos adicionais de virtualização em sistemas distribuídos

Infrastructure-as-a-Service (IaaS)

- Serviços de nuvem, como processamento e armazenamento de dados

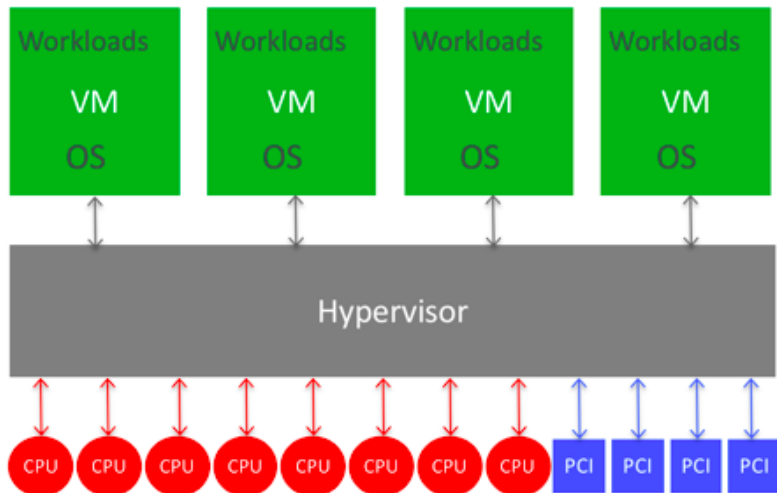
Platform-as-a-Service (PaaS)

- IaaS + plataformas de software

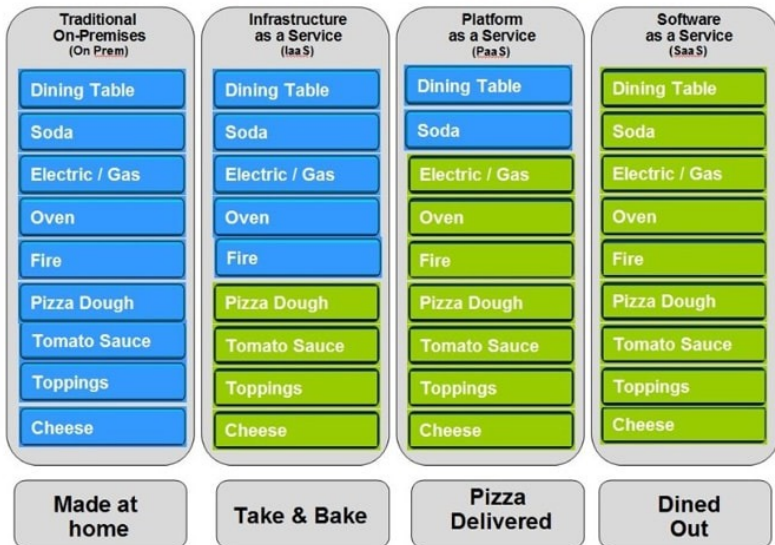
Software-as-a-Service (SaaS)

- PaaS + Aplicações

VIRTUALIZAÇÃO EM SISTEMAS DISTRIBUÍDOS



Pizza as a Service



COMPUTADORES!

