

# INTRODUÇÃO

MCTA025-13 - SISTEMAS DISTRIBUÍDOS

---

Emilio Francesquini  
e.francesquini@ufabc.edu.br

04 de junho de 2018

Centro de Matemática, Computação e Cognição  
Universidade Federal do ABC



- Prof. Dr. Emilio Francesquini
- `e.francesquini@ufabc.edu.br`
- `http://professor.ufabc.edu.br/~e.francesquini`
- Santo André, Bloco A, Sala 531-2

TODAS AS INFORMAÇÕES RELATIVAS À DISCIPLINA

TAIS COMO:

DATAS IMPORTANTES

CRITÉRIOS DE AVALIAÇÃO

BIBLIOGRAFIA

AVISOS

...

ESTARÃO DISPONÍVEIS EM:

<http://professor.ufabc.edu.br/~e.francesquini/sd/>

- Estes slides foram preparados para o curso de **Sistemas Distribuídos na UFABC**.
- Este material pode ser usado livremente desde que sejam mantidos, além deste aviso, os créditos aos autores e instituições.
- Estes slides foram adaptados daqueles originalmente preparados (e gentilmente cedidos) pelo professor **Daniel Cordeiro, da EACH-USP** que por sua vez foram baseados naqueles disponibilizados online pelos autores do livro “Distributed Systems”, 3ª Edição em:  
<https://www.distributed-systems.net>.

## Multitasking

*Attention, multitaskers (if you can pay attention, that is):  
Your brain may be in trouble.*

*People who are regularly bombarded with several streams of electronic information do not pay attention, control their memory or switch from one job to another as well as those who prefer to complete one task at a time, a group of Stanford researchers has found.*

*(...)*

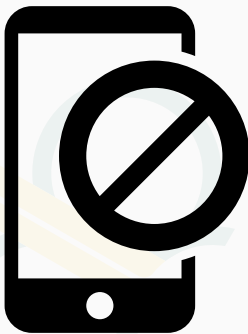
*So maybe it's time to stop e-mailing if you're following the game on TV, and rethink singing along with the radio if you're reading the latest news online. By doing less, you might accomplish more.*

<http://news.stanford.edu/2009/08/24/multitask-research-study-082409/>

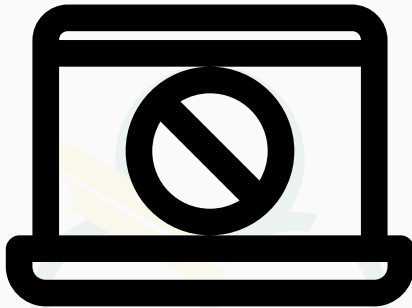
## O PERIGO DE FAZER VÁRIAS COISAS AO MESMO TEMPO

- Veja o vídeo de Clifford Nass (Stanford) em <https://youtu.be/PriSFBu5CLs>
- Se render às distrações do mundo digital (e-mail, mensagens instantâneas, Facebook, etc.) faz o cérebro lançar pequenas doses de dopamina
- Com o tempo, ficamos viciados nisso
- Resultado: *multitaskers* gastam muito mais poder de processamento cerebral do que *monotaskers* quando são distraídos
- Efeitos a longo prazo são difíceis de reverter

POR ISSO, NA SALA DE AULA:



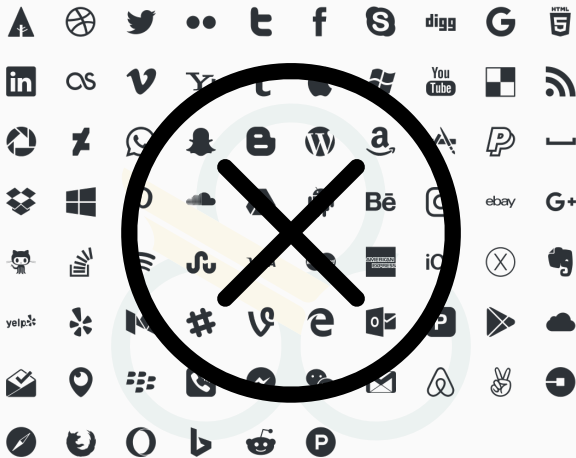
No Phone by Rflor from the Noun Project



blocked laptop by unlimicon from the Noun Project



# POR ISSO, NA SALA DE AULA:



Avise seu professor **o quanto antes** sobre a necessidade de cuidados extras para acessibilidade nos casos de deficiência:

- visual,
- física,
- auditiva,
- dislexia,
- etc.

<http://proap.ufabc.edu.br/>

- Professor: Emilio Francesquini —  
`e.francesquini@ufabc.edu.br`
- Aulas:
  - Segunda das 21:00 às 23:00, semanal, Sala A-107-0
  - Quarta das 19:00 às 21:00, quinzenal I, Sala A-108-0

Quartas, das 19:00 às 21:00, quinzenal II

- Turma NA1 — NA1MCTA025-13SA
  - Professor: Emilio Francesquini
  - Sala: 409-2
- Turma NA2 — NA2MCTA025-13SA
  - Professor: Fernando Teubl Ferreira
  - Sala: 407-2

- Teoria e Prática (Turma NA1) - Prof. Emilio Francesquini
  - Presencial
    - Na própria sala de aula, após as aulas.
    - Quarta-feira, das 17:00 às 19:00, sala 531-2.
    - Agendado por e-mail.
  - Online
    - Por e-mail.
    - Pelo fórum da disciplina (TIDIA – Procure por “SD - 2018.Q2 - Teoria - Turmas NA1 e NA2”)
- Prática (Turma NA2) – Prof. Fernando Teubl Ferreira
  - Quarta-feira, das 17:00 às 19:00, sala 525-2.



Qualquer tentativa de fraude nas provas, listas de exercícios ou projetos implicará:

- **Conceito final CF = F (reprovado)** para **TODOS** os envolvidos.
- **Denúncia apresentada à Comissão de Transgressões Disciplinares Discentes da Graduação**, a qual decidirá sobre a punição adequada à violação que pode resultar em **advertência, suspensão ou desligamento**, de acordo com os artigos 78-82 do Regimento Geral da UFABC.
- **Denúncia apresentada à Comissão de Ética da UFABC**, de acordo com o artigo 25 do Código de Ética da UFABC.

# CRITÉRIOS DE AVALIAÇÃO

A avaliação da disciplina será composta por duas notas, uma referente à teoria e outra à prática. Considere:

- $N_F$  é a nota final;
- $N_T$  é a nota de teoria;
- $N_P$  é a nota de prática.

A nota final  $N_F$  será determinada da seguinte maneira:

$$N_F = \begin{cases} \min\{N_T, N_P\}, & \text{se } N_T < 5 \text{ ou } N_P < 5 \\ 0.6 \cdot N_T + 0.4 \cdot N_P, & \text{caso contrário} \end{cases}$$

O conceito final  $C_F$  será obtido de acordo com a equação abaixo:

$$C_F = \begin{cases} \text{O,} & \text{se ausência total exceder 25\%} \\ \text{F,} & \text{se } N_F \in [0, 0; 5, 0) \\ \text{D,} & \text{se } N_F \in [5, 0; 6, 0) \\ \text{C,} & \text{se } N_F \in [6, 0; 7, 0) \\ \text{B,} & \text{se } N_F \in [7, 0; 8, 5) \\ \text{A,} & \text{se } N_F \in [8, 5; 10, 0] \end{cases}$$



A nota de teoria  $N_T$  será formada por duas provas  $P_1$  e  $P_2$ , de pesos iguais. Todas as provas serão efetuadas em sala de aula, sem auxílio de computador.

Haverá também uma prova substitutiva  $P_S$  que será aberta a todos os interessados, ainda que eles tenham feito tanto a  $P_1$  quanto a  $P_2$ .

**ATENÇÃO** — A nota da PS será utilizada obrigatoriamente em substituição à menor nota entre  $P_1$  e  $P_2$  ainda que isto diminua a nota final do aluno!

$$N_T = \begin{cases} \frac{\max\{P_1, P_2\} + P_S}{2}, & \text{caso tenha feito a } P_S \\ \frac{P_1 + P_2}{2}, & \text{caso contrário} \end{cases}$$

A avaliação da prática será feita através de dois projetos  $Pr_1$  e  $Pr_2$  de igual peso. Sua nota será, então, calculada pela seguinte equação:

$$N_P = \frac{Pr_1 + Pr_2}{2}$$

# RECUPERAÇÃO

A Resolução ConsEPE nº 182 garante a todos os alunos com  $C_F$  igual a **D** ou **F** o direito a fazer uso de mecanismos de recuperação.

A recuperação será feita através de uma prova  $P_R$ , sem consulta, e a sua nota será utilizada para compor a o conceito pós-recuperação  $C_R$  conforme as equações abaixo:

$$N_R = \frac{P_R + N_F}{2}$$

**Caso 1**  $C_F = D$ :

$$C_R = \begin{cases} C, & \text{se } N_R \geq 6,0 \\ D, & \text{caso contrário} \end{cases}$$

**Caso 2**  $C_F = F$ :

$$C_R = \begin{cases} D, & \text{se } N_R \geq 5,0 \\ F, & \text{caso contrário} \end{cases}$$

- Prova 1 — 16/07/2018
- Prova 2 — 27/08/2018
- Prova Substitutiva — 28/08/2018
- Prova de Recuperação — 17/09/2018 — Horário a definir
- Projeto 1 — A definir
- Projeto 2 — A definir

- [ST] — **Distributed Systems 3rd edition** (2017), por Maarten van Steen e Andrew S. Tanenbaum
- <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>
- Tópicos:
  - Introdução;
  - Arquiteturas de sistemas distribuídos;
  - Processos;
  - Comunicação;
  - Nomes;
  - MapReduce;
  - Coordenação;
  - Modelos de Consistência;

- **[CDKB]** Coulouris, J. Dollimore, T. Kindberg, and G. Blair. **Distributed Systems: Concepts and Design**. 5th Edition, Addison-Wesley, 2011.
- **[KS]** A.D. Kshemkalyani, M. Singhal, **Distributed Computing: Principles, Algorithms, and Systems**. Paperback edition, Cambridge University Press, 2011.
- **[NL]** Lynch, Nancy Ann. **Distributed algorithms**, Morgan Kaufmann Publishers, 1997.

Versões disponíveis na biblioteca tanto em português quanto em inglês. Contudo, (em particular as versões em português) podem ser de edições anteriores. Veja página da disciplina para mais detalhes.

O QUE É UM SISTEMA DISTRIBUÍDO?

# TODOS JÁ DEVEM TER OUVIDO ALGO SOBRE CLOUD COMPUTING

Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço



Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço

# TODOS JÁ DEVEM TER OUVIDO ALGO SOBRE CLOUD COMPUTING

Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço

Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço

# TODOS JÁ DEVEM TER OUVIDO ALGO SOBRE CLOUD COMPUTING

Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço

Ou ao menos algumas dessas ideias:

- “Computação em nuvem finalmente tornou realidade o sonho da computação utilitária”
- “Desenvolvedores não precisam mais se preocupar em conseguir grandes somas de dinheiro antes de colocar uma nova aplicação web no ar”
- “Adeus aos problemas de provisionamento de servidores (Elasticidade dos recursos)”
- Software como um serviço
- Plataforma como um serviço
- Infraestrutura como um serviço

# QUATRO PROBLEMAS QUE (AINDA) REQUEREM CONSTANTE INOVAÇÃO TECNOLÓGICA

- Problemas “em escala da web”
- **Grandes** *data centers*
- Computação paralela e distribuída
- Aplicações web interativas

- Características
  - Definitivamente *data-intensive*
  - Mas podem também ser *processing-intensive*
- Exemplos:
  - *Crawling*, indexação, busca, mineração de dados da web
  - Pesquisa em biologia computacional na era “pós-genômica”
  - Processamento de dados científicos (física, astronomia, etc.)
  - Redes de sensores
  - Aplicações Web 2.0
  - etc.

## DE QUAL VOLUME DE DADOS ESTAMOS FALANDO?

Problemas da ordem de **petabytes!**

$$\begin{aligned}1 \text{ PB} &= 1.000.000.000.000.000 \text{ B} \\ &= 1.000^5 \text{ B} \\ &= 10^{15} \text{ B} \\ &= 1 \text{ milhão de gigabytes} \\ &= 1 \text{ mil terabytes}\end{aligned}$$



### Muitos, mas muitos dados

- O Google processa cerca de **20 petabytes** de dados por dia (2008)
- O Wayback Machine tem cerca de 3 petabytes + 100 terabytes/dia (mar/2009)
- O Facebook tem cerca de 2,5 petabytes de dados de usuários + 15 terabytes/dia (abr/2009)
- O site eBay tem cerca de 6,5 petabytes de dados dos usuários + 50 terabytes/dia (mai/2009)
- O Grande Colisor de Hádrons do CERN irá gerar cerca de 15 petabytes/ano

### Muitos, mas muitos dados

- O Google processa cerca de **20 petabytes** de dados por dia (2008)
- O Wayback Machine tem cerca de **3 petabytes + 100 terabytes/dia** (mar/2009)
- O Facebook tem cerca de **2,5 petabytes** de dados de usuários + **15 terabytes/dia** (abr/2009)
- O site eBay tem cerca de **6,5 petabytes** de dados dos usuários + **50 terabytes/dia** (mai/2009)
- O Grande Colisor de Hádrons do CERN irá gerar cerca de **15 petabytes/ano**

### Muitos, mas muitos dados

- O Google processa cerca de **20 petabytes** de dados por dia (2008)
- O Wayback Machine tem cerca de **3 petabytes + 100 terabytes/dia** (mar/2009)
- O Facebook tem cerca de **2,5 petabytes de dados de usuários + 15 terabytes/dia** (abr/2009)
- O site eBay tem cerca de **6,5 petabytes de dados dos usuários + 50 terabytes/dia** (mai/2009)
- O Grande Colisor de Hádrons do CERN irá gerar cerca de **15 petabytes/ano**

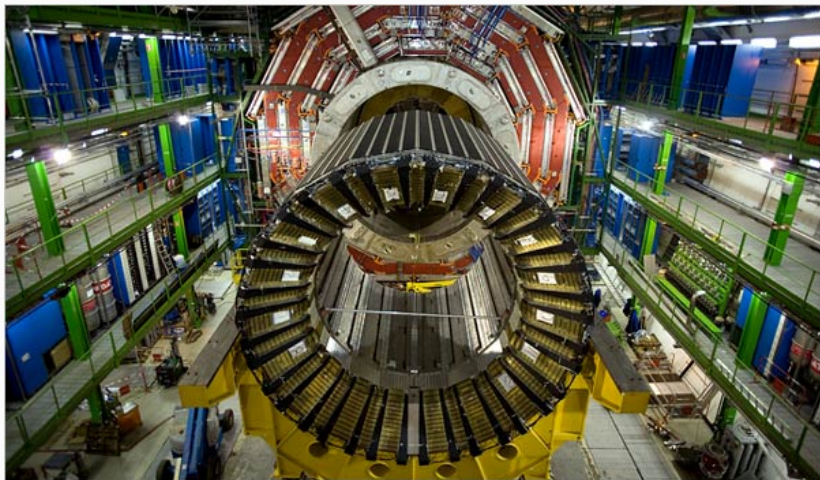
### Muitos, mas muitos dados

- O Google processa cerca de **20 petabytes** de dados por dia (2008)
- O Wayback Machine tem cerca de **3 petabytes + 100 terabytes/dia** (mar/2009)
- O Facebook tem cerca de **2,5 petabytes de dados de usuários + 15 terabytes/dia** (abr/2009)
- O site eBay tem cerca de **6,5 petabytes de dados dos usuários + 50 terabytes/dia** (mai/2009)
- O Grande Colisor de Hádrons do CERN irá gerar cerca de **15 petabytes/ano**

### Muitos, mas muitos dados

- O Google processa cerca de **20 petabytes** de dados por dia (2008)
- O Wayback Machine tem cerca de **3 petabytes + 100 terabytes/dia** (mar/2009)
- O Facebook tem cerca de **2,5 petabytes de dados de usuários + 15 terabytes/dia** (abr/2009)
- O site eBay tem cerca de **6,5 petabytes de dados dos usuários + 50 terabytes/dia** (mai/2009)
- O Grande Colisor de Hádrons do CERN irá gerar cerca de **15 petabytes/ano**

## DE QUAL VOLUME DE DADOS ESTAMOS FALANDO?



## DE QUAL VOLUME DE DADOS ESTAMOS FALANDO?

$$\begin{aligned}1 \text{ PB} &= 1.000.000.000.000.000 \text{ B} \\ &= 1.000^5 \text{ B} \\ &= 10^{15} \text{ B} \\ &= 1 \text{ milhão de gigabytes} \\ &= 1 \text{ mil terabytes}\end{aligned}$$

Ou seja, os **15 petabytes** que o CERN irá gerar por ano equivalem a **15 milhões de gigabytes**. Seriam necessários **1,7 milhão de DVDs *dual-layer*** para armazenar tanta informação!

# O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - no filme "O casamento do americano" (1971) o casamento do americano
    - ocorreu nos Estados Unidos em 1971
    - o filme "O casamento do americano" (1971) o casamento
    - ocorreu em 1971
    - o filme "O casamento do americano" (1971) o casamento
    - ocorreu em 1971
    - que por sua vez permitiu encontrar novos fatos



## O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - os fatos: “Nascimento-de(Mozart, 1756)” e “Nascimento-de(Einstein, 1879)”
    - levam aos dados: “Wolfgang Amadeus Mozart (1756–1791)” e “Einstein nasceu em 1879”
    - que levam a diferentes padrões: “PESSOA (DATA –)” e “PESSOA nasceu em DATA”
    - que, por sua vez, permitem encontrar novos fatos

# O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - os fatos: “Nascimento-de(Mozart, 1756)” e “Nascimento-de(Einstein, 1879)”
    - levam aos dados: “Wolfgang Amadeus Mozart (1756–1791)” e “Einstein nasceu em 1879”
    - que levam a diferentes padrões: “PESSOA (DATA –)” e “PESSOA nasceu em DATA”
    - que, por sua vez, permitem encontrar novos fatos

## O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - os fatos: “Nascimento-de(Mozart, 1756)” e “Nascimento-de(Einstein, 1879)”
    - levam aos dados: “Wolfgang Amadeus Mozart (1756–1791)” e “Einstein nasceu em 1879”
    - que levam a diferentes padrões: “PESSOA (DATA –)” e “PESSOA nasceu em DATA”
    - que, por sua vez, permitem encontrar novos fatos

## O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - os fatos: “Nascimento-de(Mozart, 1756)” e “Nascimento-de(Einstein, 1879)”
    - levam aos dados: “Wolfgang Amadeus Mozart (1756–1791)” e “Einstein nasceu em 1879”
    - que levam a diferentes padrões: “PESSOA (DATA –” e “PESSOA nasceu em DATA”
    - que, por sua vez, permitem encontrar novos fatos

## O QUE SE FAZ COM TANTOS DADOS?

- Encontram informações sobre novos fatos
  - Casamento de padrões com informações da web
  - ex: quem matou John Lennon?
- Procuram por novas relações entre os dados
  - Alguns padrões levam a novas relações:
    - os fatos: “Nascimento-de(Mozart, 1756)” e “Nascimento-de(Einstein, 1879)”
    - levam aos dados: “Wolfgang Amadeus Mozart (1756–1791)” e “Einstein nasceu em 1879”
    - que levam a diferentes padrões: “PESSOA (DATA –)” e “PESSOA nasceu em DATA”
    - que, por sua vez, permitem encontrar novos fatos

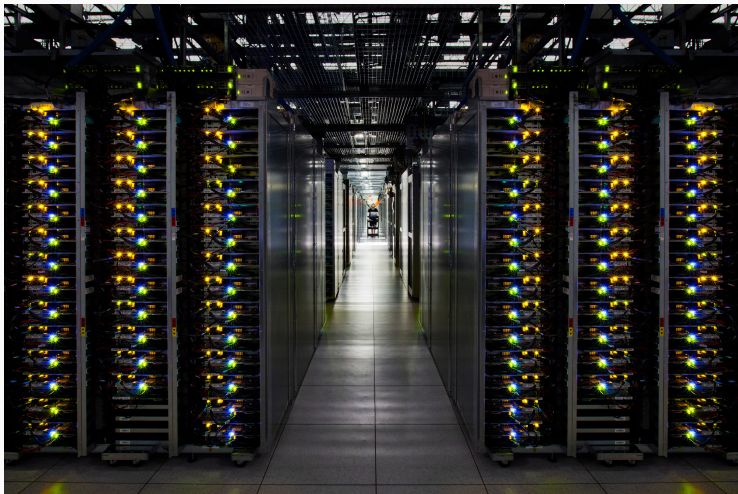
Estratégia simples (mas de difícil execução):

- Dividir para conquistar
- Usar mais recursos computacionais a medida que mais dados aparecerem

## Pergunta:

Quão grandes são os *data centers* que fazem sistemas que afetam a vida de quase todo mundo que se conecta a Internet (como os do Google, Facebook, etc.) funcionarem?

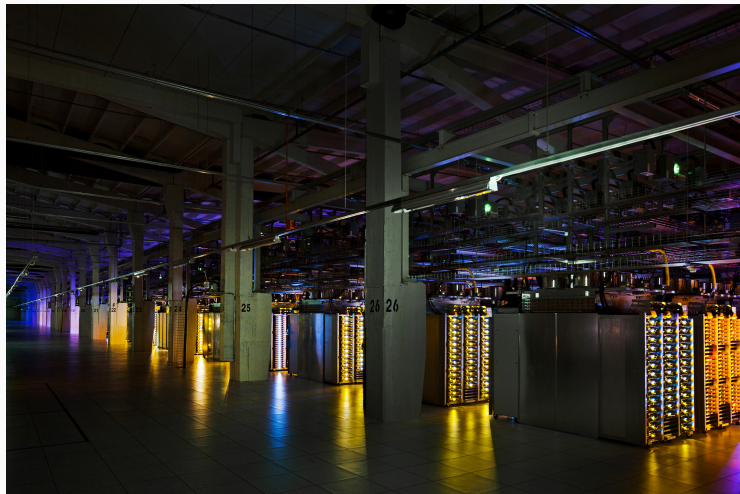
# GRANDES DATA CENTERS



Fonte: <http://www.google.com/intl/pt-BR/about/datacenters/>

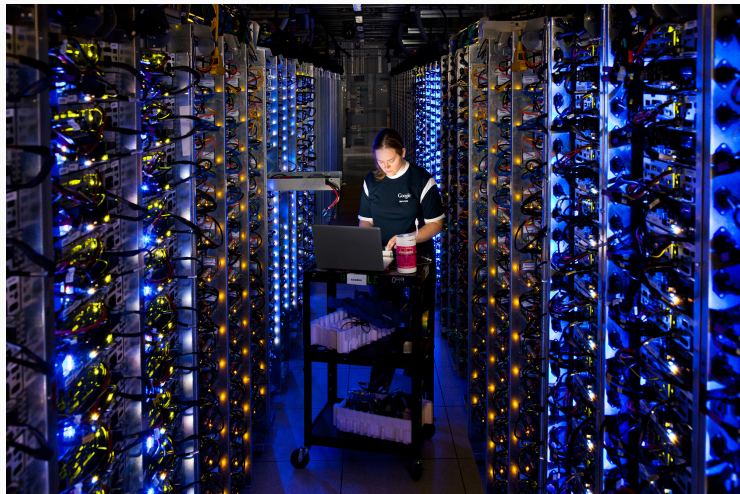


# GRANDES DATA CENTERS



Fonte: <http://www.google.com/intl/pt-BR/about/datacenters/>

# GRANDES DATA CENTERS



Fonte: <http://www.google.com/intl/pt-BR/about/datacenters/>

# GRANDES DATA CENTERS



Fonte: <http://www.google.com/intl/pt-BR/about/datacenters/>

Só o Google tem **treze** desses espalhados pelo mundo!

## Américas

- Berkeley County, Carolina do Sul
- Council Bluffs, Iowa
- Douglas County, Georgia
- Mayes County, Oklahoma
- Lenoir, Carolina do Norte
- The Dalles, Oregon
- Quilicura, Chile

Só o Google tem **treze** desses espalhados pelo mundo!

## Ásia

- Hong Kong
- Cingapura
- Taiwan



Só o Google tem **treze** desses espalhados pelo mundo!

## Europa

- Hamina, Finlândia
- St Ghislain, Bélgica
- Dublin, Irlanda



Como isso era feito até  
então?



- anos 50: computadores eram grandes calculadoras programadas com cartões perfurados; início da computação paralela
- final dos anos 60: ARPANET (computadores começavam a serem interconectados; noção de computação *como um serviço*)
- anos 70: surgem os primeiros microprocessadores
- anos 80: popularização dos computadores pessoais
- anos 90: popularização da Internet



# EVOLUÇÃO DA COMPUTAÇÃO

- anos 50: computadores eram grandes calculadoras programadas com cartões perfurados; início da computação paralela
- final dos anos 60: ARPANET (computadores começavam a serem interconectados; noção de computação *como um serviço*)
- anos 70: surgem os primeiros microprocessadores
- anos 80: popularização dos computadores pessoais
- anos 90: popularização da Internet

# EVOLUÇÃO DA COMPUTAÇÃO

- anos 50: computadores eram grandes calculadoras programadas com cartões perfurados; início da computação paralela
- final dos anos 60: ARPANET (computadores começavam a serem interconectados; noção de computação *como um serviço*)
- anos 70: surgem os primeiros microprocessadores
- anos 80: popularização dos computadores pessoais
- anos 90: popularização da Internet

# EVOLUÇÃO DA COMPUTAÇÃO

- anos 50: computadores eram grandes calculadoras programadas com cartões perfurados; início da computação paralela
- final dos anos 60: ARPANET (computadores começavam a serem interconectados; noção de computação *como um serviço*)
- anos 70: surgem os primeiros microprocessadores
- anos 80: popularização dos computadores pessoais
- anos 90: popularização da Internet

# EVOLUÇÃO DA COMPUTAÇÃO

- anos 50: computadores eram grandes calculadoras programadas com cartões perfurados; início da computação paralela
- final dos anos 60: ARPANET (computadores começavam a serem interconectados; noção de computação *como um serviço*)
- anos 70: surgem os primeiros microprocessadores
- anos 80: popularização dos computadores pessoais
- anos 90: popularização da Internet

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decodificação por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

# PARADIGMAS DE COMPUTAÇÃO

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decodificação por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decodificação por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decriptografia por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)



- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decifração por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decifração por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decifração por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

- Computadores Pessoais
- Computadores Paralelos
- Aglomerados de Computadores (*clusters*)
- Computação Voluntária (*Volunteer Computing*):
  - The Great Internet Mersenne Prime Search (1996): busca por primos de Mersenne (primos da forma  $2^n - 1, n \in \mathbb{N}$ )
  - distributed.net (1997): decifração por força-bruta
  - SETI@Home (1999): análise de sinais de rádio vindos do espaço em busca de evidência de vida extra-terrestre
- Computação em Grade (*Grid Computing*)

## PROBLEMAS ATUAIS

---



E como é feito agora?

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *foros* da *regueses*, a *Islândia*, o estado americano do *Oregon* e *minas* *desconodas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - *O que os fiordes noruegueses, a Islândia, o estado americano do Oregon e minas abandonadas tem em comum?*
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento



- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

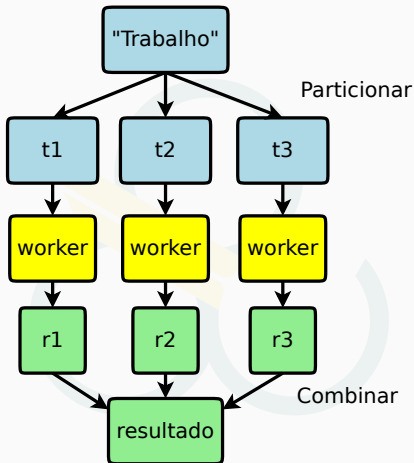
- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Seu problema aumenta na mesma escala da web?  
**Fácil:** basta adicionar mais máquinas
- Tendência: centralização dos recursos computacionais em grandes data centers
  - O que os *fiordes noruegueses*, a *Islândia*, o estado americano do *Oregon* e *minas abandonadas* tem em comum?
- Problemas a serem resolvidos:
  - Redundância
  - Eficiência
  - Utilização
  - Gerenciamento

- Escalabilidade horizontal, não vertical
  - Existem limites para máquinas SMP e arquiteturas de memória compartilhada
- Mova o processamento para perto dos dados
  - a banda de rede é limitada
- Processe os dados sequencialmente, evite padrões de acesso aleatórios
  - *seeks* são custosos, mas a vazão (*throughput*) do disco é razoável

# COMO PROGRAMAR APLICAÇÕES ESCALÁVEIS?

Divisão e conquista





- Como repartir as unidades de trabalho entre os *workers*?
- O que fazer quando temos mais trabalho do que *workers*?
- E se os *workers* precisarem compartilhar resultados intermediários entre si?
- Como agregar os resultados parciais?
- O que fazer se um *worker* parar de funcionar?
- Como saber se todos os *workers* terminaram seus trabalhos?

- Problemas de paralelização surgem por causa de:
  - comunicação entre os *workers*
  - acesso a recursos compartilhados (por exemplo, dados)
- Portanto, precisamos de algum mecanismo de sincronização

- É difícil, pois:
  - Não sabemos em que ordem cada *worker* será executado
  - Não sabemos quando um *worker* irá interromper outro *worker*
  - Não sabemos em qual ordem os *workers* irão acessar os dados compartilhados
- Por tanto, nós precisamos de:
  - Semáforos (**lock, unlock**)
  - Variáveis condicionais (**wait, notify, broadcast**)
  - Barreiras de sincronização
- Ainda assim, restam problemas como:
  - Deadlock, starvation, race conditions, ...

- Modelos de programação:
  - Memória compartilhada (pthreads)
  - Passagem de mensagens (MPI)
- Padrões arquiteturais:
  - Mestre–escravo
  - Produtor–consumidor
  - Filas de trabalho compartilhadas

- Tudo se resume ao nível mais adequado de abstração
- Esconda os detalhes do sistema dos desenvolvedores
  - Evita os problemas com race conditions, contenção em locks, etc.
- Separe o “quê” do “como”:
  - O desenvolvedor especifica apenas o **que** deve ser computado
  - O arcabouço deve se encarregar de **como** realizar a execução

O data center é o computador!

Quando a *Animoto*<sup>1</sup> tornou seu serviço disponível no Facebook, houve uma explosão na demanda que exigiu que o número de servidores fosse aumentado de 50 para 3.500 em **três** dias. Após esse pico de utilização, o tráfego caiu para um nível **muito** menor do que o pico.

- Se fosse uma companhia tradicional, o que teria acontecido?
- Com Computação em Nuvem: pague mais durante os picos, devolva os recursos desnecessários depois

---

<sup>1</sup>A Animoto é uma *startup* que oferece uma aplicação web que produz vídeos a partir de fotos, videoclipes e música.

# EVOLUÇÃO DO NÚMERO DE INSTÂNCIAS EC2 USADAS PELA ANIMOTO

