

► Arquitetura P2P

- Componentes reparados por hosts com papéis semelhantes.
- Demanda de um nó pode ser porida por outro nó.
- Intermitente \Rightarrow Nós entram e saem da rede, porém ela se mantém operacional.

o Tempo estimado para distribuir um arquivo de X Bytes para N hosts.

- Transmissão do Servidor

↳ Pelo menos 1 cópia do upload

$$\text{Tempo de envio 1 cópia} \Rightarrow T_0 = \frac{X}{\text{Taxa de upload}}$$

$$\text{Tempo para enviar } N \text{ cópias para cada cliente} \Rightarrow T_1 = N \cdot T_0$$

- Download do Cliente

$$\text{Tempo min download} = \frac{X}{\text{Taxa download}}$$

Tempo para distribuir arquivo de X bytes para N hosts.

$$T_3 \geq \max \left\{ \frac{T_1, T_2, N \cdot X}{\left(T_{\text{upload}} + \sum_{\text{cliente}} T_{\text{upload}} \right)} \right\} \Rightarrow T_3 \text{ não é linear}$$

Tempo de transmissão Cliente/Servidor

↳ Deve enviar N cópias sequencialmente do arquivo X

- Transmissão do Servidor

↳ Pelo menos 1 cópia do upload

$$\text{Tempo de envio 1 cópia} \Rightarrow T_0 = \frac{X}{\text{Taxa de upload}}$$

$$\text{Tempo para enviar } N \text{ cópias para cada cliente} \Rightarrow T_1 = N \cdot T_0$$

$$\text{Tempo para distribuir } T_3 \geq \max \{ T_1, T_2 \}$$

$$T_3 \text{ é linear}$$

- Download do Cliente

$$\text{Tempo min download} = \frac{X}{\text{Taxa download}}$$

➡ Distributed hash table (DHT)

- forma mais conveniente de armazenar e buscar uma representação numérica da chave. Onde a ideia é distribuir pares (key, value) para o conjunto de pros.

* Salt

↳ Valor aleatório adicionado a uma senha antes de aplicar o hash

- Garante que mesmo senhas idênticas resultem em hashes diferentes.
- Proteção contra ataques de "força bruta" e Rainbow-table

⇒ Medição de Desempenho

$$\text{Desempenho} = \frac{1}{\text{Tempo de execução}}$$

$$\text{Ganho de desempenho (speedup)} = \frac{A}{B}$$

$$\text{Speedup} = \frac{T(1)}{T(N)} \rightarrow \begin{array}{l} \text{Tempo de execução em 1 processador} \\ \text{Tempo de execução em N ps} \end{array}$$

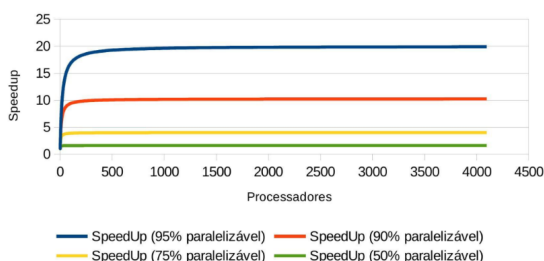
⇒ Lei de Amdahl

$$S = \frac{1}{\left((1-p) + \frac{p}{N} \right)}$$

Lei de Amdahl

* Speedup máximo é limitado pela parte do programa que não pode ser paralelizada

* À medida que mais processadores são adicionados, o ganho de desempenho adicional diminui especialmente se grande parte do programa for sequencial.



⇒ Performance de Pico

$$P = N.C.F.R$$

