

# Engenharia de Sistemas de Computação

## Otimização do Desempenho

### Aula 1 – Medição de Desempenho

Vítor Oliveira ([vitor.s.p.oliveira@gmail.com](mailto:vitor.s.p.oliveira@gmail.com))

6 Abril 2021

1

# Quem sou eu

Vítor Oliveira

- Performance Architect/Engineer (Oracle de 2014 até 2021, Huawei Stockholm Research Institute desde 2021);
- Focado em otimização de desempenho e profiling de bases de dados, em particular MySQL, em ambientes empresais e infraestruturas Cloud;
- <https://www.linkedin.com/in/vitor-oliveira-8b49134/>

## Outras experiências

- Consultor para o desenho de sistemas de computação avançada, incluindo os clusters HPC Search (UM) e Darwin (INL), infraestruturas de suporte a correio eletrónico para milhares de utilizadores, sites de operadores de telecomunicações, infraestruturas hospitalares, etc., de 2000 até 2013.
- Docente convidado na Universidade do Minho e no Instituto Politécnico de Bragança, na área da arquitetura de computadores e dos sistemas de computação avançada, intermitentemente desde 1997.
- Investigador do Laboratório de Instrumentação e Física Experimental de Partículas desde 2012, incluindo a organização local da escola de verão do CERN CSC'2014.
- Diretor de Sistemas de Informação do Teatro Nacional S. João entre 2007 e 2014, assessor da direção do teatro entre 2002 e 2006 e responsável pela informática de 1997 a 2001;

# Áreas de actividade

## **Análise e optimização do desempenho de sistemas e aplicações**

- As aplicações são, na maior parte das vezes, complexas e a utilização que fazem dos recursos computacionais subjacentes pode não ser a melhor.
- Para optimizar o seu desempenho é preciso estudar o comportamento das aplicações sobre hardware específicos, perceber o que pode ser melhorado e implementar essas medidas.

## **Desenho e dimensionamento de infraestruturas de computação**

- Partindo de aplicações pré-definidas, para disponibilizar um qualquer serviço a partir delas torna-se necessário avaliar quais os recursos necessários aos mais variados níveis, incluindo o tipo e número de servidores, os sistemas de armazenamento de dados, as redes de comunicação, etc.

## **Sistemas de computação de alto-desempenho (HPC)**

- Os sistemas HPC são necessários para as utilizadores com os requisitos mais elevados, em que um pequeno conjunto de máquinas é insuficiente para lidar com um problema em particular.
- Estes sistemas tipicamente possuem servidores, redes e sistemas de armazenamento optimizados para a computação paralela.

## 2 *Performance Engineering*

- Especialidade que avalia o desempenho de aplicações e sistemas de computação e que propõe medidas para melhorar esse desempenho segundo critérios específicos de cada caso, nomeadamente em termos de tempo de resposta, débito, custo, eficiência e escalabilidade.

# Objectivos

## Melhorar o experiência do utilizador

- O tempo de resposta pode ser determinante para o sucesso de algumas empresas. Os tempos de resposta elevados podem não só frustrar utilizadores como diminuir o seu número
- Mas também pode custar muito dinheiro, como no caso do high-speed trading e processamento de block-chain, em que há competição entre sistemas concorrentes por um mesmo serviço.

## Reduzir o consumo de recursos

- Os servidores são muito dispendiosos em termos de aquisição inicial, energia eléctrica, arrefecimento e recursos humanos.
- Os sites à escala da internet têm por vezes centenas de milhares de servidores por data-center, sendo um dos maiores consumidores de energia do planeta. A esta escala mesmo pequenos aumentos de desempenho podem significar reduções de custo muito significativas.

# Objectivos

## Escalabilidade/adaptabilidade

- Os sistemas têm de ser desenhados para atender não só as cargas actuais, mas também os picos de utilização e permitir a escalabilidade futura.
- Tal implica identificar cedo qual a resposta do sistema ao aumento dos requisitos e adaptar-se com antecedência.
- Há empresas que estão em permanência em modo de aquisição de novos recursos para juntar aos seus sistemas, pelo que a escalabilidade é fundamental.

## Vantagem competitiva

- O desempenho de um sistema é frequentemente utilizado para mostrar uma capacidade técnica superior em relação aos concorrentes (o, às vezes, chamado *benchmarking*).
- Esta vantagem pode levar empresas a prosperar em ambientes de competição agressiva e a falta dela pode levar empresas à falência.

# Áreas de intervenção

## Otimização de desempenho do software

- Melhorar o desempenho de uma aplicação específica nos vários domínios/sistemas em que será utilizada.
- Introduzir uma alteração no código que pode trazer regressões no tempo de execução.

## Otimização de desempenho dos sistemas

- Melhorar o desempenho das infraestruturas de computação tendo em conta as aplicações que acolherão.

## Site-reliability/Qualidade de serviço

- Monitorizar os tempos de resposta dos vários sistemas de forma a que seja cumprida uma determinada qualidade de serviço (QoS, contratada ou esperada).
- Adaptar as aplicações e sistemas para que os sistemas cumpram eficientemente essa QoS.



---

## 3 Desafios na avaliação de desempenho

# Complexidade do hardware

## A complexidade dos sistemas subjacentes

- Os servidores modernos têm múltiplos processadores, cada um com cada vez mais núcleos, múltiplos níveis de memória cache e tempos de acesso às memórias não-uniformes, o que obriga a recorrer ao paralelismo para conseguir explorar os recursos disponíveis.
- Os processadores modernos ajustam agressivamente a frequências às condições de execução, pelo que uma optimização pode não produzir o efeito esperado por causa das implicações em termos energéticos no processador.
- Os sistemas distribuídos passaram a ser a norma, obrigando a considerar as implicações da rede de interligação dos servidores e da comunicações com o mundo exterior.
- O armazenamento de dados passou dos discos móveis para memórias flash e semelhantes, com níveis de desempenho mais elevados mas também exigindo optimização do código que lhes faz acesso, deixou de ser possível esconder código ineficiente atrás da latência do disco.

# Complexidade do software

## A dimensão do código

- A maior parte das aplicações “interessantes” atingem dimensões em linhas de código que são difíceis de conhecer e a optimização do software requer a escolha dos pontos mais eficazes de ataque;

## A interdependência de componentes

- A remoção de um ponto de contenção num ponto particular do código pode não melhorar o desempenho, acabando apenas por mover esse ponto de contenção para o gargalo seguinte.
- Aliás, chega a acontecer que o desempenho baixa quando esse gargalo é removido ao agravar outros pontos de contenção mais graves.

## A necessidade de adaptação

- A optimização do desempenho pode ser demorada e implicar esforços de múltiplos anos, o que nem sempre é compatível com a introdução da inovação no código que torna a optimização do código um “alvo móvel”.

# Complexidade da interacção do software com o hardware

## O paralelismo

- O problema maior que se tem de lidar com a optimização está na escalabilidade, na capacidade de aproveitar o paralelismo e evitar os pontos de contenção quando se tem muitas tarefas a executar sobre o mesmo conjunto de recursos.

## Os sistemas não são todos iguais

- Algumas optimizações de desempenho podem funcionar eficazmente num processador específico, ou família de processadores, mas ter impact negativo noutros processadores, pelo que terá de se avaliar o seu efeito numa gama alargada de sistemas.

## Os sistemas como um todo

- A optimização de uma parte do código poderá não ter o impacto desejado, mas pode verificar-se que são essenciais para o bom desempenho se outras dependencias forem cumpridas.

## 4 Medição de desempenho (*benchmarking*)

Para a engenharia do desempenho é essencial medir corretamente os vários elementos que se pretende otimizar.

# Medição de desempenho

## O que é o desempenho

- O que se entende por desempenho?
  - Muitas coisas, depende de cada contexto!
  - Em bases de dados é vulgar utilizar métricas como tempo de resposta, débito de pedidos atendidos ao longo do tempo, tempo de propagação dos dados a toda a infraestrutura, eficiência de utilização do CPI, memória, energia.
  - Diferentes sistemas podem ter diferentes requisitos.

# Medição de desempenho

## Medir o desempenho do sistema

- Utilizar/desenvolver testes de desempenho para simular as cargas que se pretendem otimizar
- Aplicar carga real no sistema em ambiente de testes
- Verificar as métricas mais relevantes em cada contexto, como o débito, o tempo de resposta, o tempo de execução e o consumo de recursos como o tempo de CPU, a memória utilizada e a largura de banda da rede.

# Medição de desempenho

## Métricas

- Débito máximo de unidades de trabalho por medida de tempo
  - Quantos pedidos http forem atendidos por segundo
  - Quantas transacções forem executadas
- Métrica importante para avaliar a utilização máxima do sistema e contenção no acesso a recursos quando o sistema é colocado em carga máxima



# Medição de desempenho

## Métricas

- Latência
  - Quanto tempo demorou a executar uma pesquisa?
  - Quando tempo demorou a actualizar um registo?
- Métrica importante em processos que dependem de um tempo de resposta máximo, como jogos interactivos, high-speed trading, etc.
- A latência tem um reflexo no débito se o número de processos concorrentes for baixo, uma vez que o próximo pedido se vai iniciar apenas depois do anterior terminar.
- (NET-CALC)

# Medição de desempenho

## Métricas

- Tempo total de execução
  - Quando tempo demorou a gerar uma simulação?
- Para processo mais longos, o tempo total de execução é uma métrica complementar à latência, mais robusta até por considerar todas as operações necessárias à aplicação.
- Ter em atenção que tempo de CPU é muito diferente do tempo de relógio, e na maior parte dos casos é este último que se pretende.

# Medição de desempenho

## Métricas

- Consumo de recursos
  - CPU
  - Memória
  - Armazenamento externo
  - Largura de banda de rede
  - Interrupções
- Dependendo das circunstâncias, é importante conhecer o consumo dos diversos recursos do sistema ao longo do tempo para perceber de que forma o sistema está no limite ou pode escalar para cargas de trabalho acrescidas.

# Medição de desempenho

## Metodologia

- Testar todas as combinações de parâmetros relevantes para o problema em causa
- Repetir observações de cada combinação até que a variância nos resultados seja aceitável;
- Usar estatísticas robustas
  - A média das medições é muito sensível a outliers (se entre quatro pessoas tiverem sido comidos dois frangos, em média estariam bem alimentados, não fosse uma das pessoas ter comido os dois frangos)
  - A utilização dos valores mínimos é muito sensível ao “alinhamento dos astros”, pode haver uma circunstancia específica em que é atingido esse valor mínimo, mas na prática ser improvável e comparar diferentes execuções em que esse valor mínima não aparece pode invalidar qualquer raciocínio sobre os valores.
  - As medianas, MAD e os percentis são mais adequados às medições de desempenho, porque são medidas de tendência e de dispersão robustas.

# Medição de desempenho

## Metodologia

- Não medir as repetições da mesma combinação seguidamente, mas intercaladas com as repetições das outras combinações;
- Mudar uma única variável de cada vez;  
  
Se se mudarem múltiplas variáveis ao mesmo tempo não se consegue identificar a responsabilidade de cada uma no comportamento final;
- Evitar medir configurações diferentes em circunstâncias diferentes  
  
Como medir uma combinação com a base de dados fria e testar a combinação seguinte com a base de dados quente
- Verificar se os scripts usados nos testes não beneficiam indevidamente alguma combinação de testes

# Medição de desempenho

## Metodologia

- Não confiar nos números até prova em contrário;
  - avaliar os números de diferentes ângulos e não precipitar a apresentação de resultados.
  - verificar os padrões e procurar resultados estranhos mais difíceis de explicar;
  - confirmar que os parâmetros escolhidos são de facto os aplicados;
  - repetir os testes sempre que as configurações subjacentes mudem e desconsiderar os resultados anteriores (mesmo que isso seja mais doloroso que uma zaragatoa).
  - remover execuções erráticas para não poluir os dados com erros de medição.
- Os gráficos são ótimos para para mostrar os resultados aos outros, mas as tabelas dinâmicas são uma forma muito eficaz de validar os resultados dos testes.

# Medição de desempenho

## Lembrem-se que

- O sistema operativo não adivinha:
  - As diferentes heurísticas do sistema operativo tentam otimizar o desempenho do sistema com um conjunto de assunções genéricas sobre as aplicações;
  - Se as heurísticas não forem adequadas ao comportamento da aplicação em estudo, o desempenho pode ser muito inferior ao que teria se o comportamento fosse adaptado a cada aplicação em particular.
  - Por vezes é possível que a aplicação informe o sistema operativo que espera ter de forma a que ajustar as heurísticas do sistema operativo (fadvise/madvise).
  - Noutras situações, as aplicações podem ser desenhadas de forma a não deixar que o sistema operativo use as heurísticas menos interessantes contornando algumas funcionalidades (thread/cpu binding, busy waiting, interrupt control, etc).

# Medição de desempenho

## Lembrem-se que

- “não são os únicos”:
  - Outros utilizadores e tarefas agendadas podem executar aplicações enquanto o benchmark está a correr e afetar os resultados na máquinas;
  - Infraestruturas como a rede e os sistemas de armazenamento podem responder de forma diferentes ao longo do tempo por causa dessas mesmas atividades de fundo, por vezes dramaticamente;
- A honestidade compensa (mesmo que a longo prazo)
  - Apresentem as combinações relevantes e utilizem todos os dados corretos que forem observados, não escolham apenas alguns dos números mais favoráveis, há fenómenos importantes que podem ser perdidos por falta dessa informação;
  - Apresentem os dados ao longo de um tempo suficiente, não apenas durante um período em que beneficia de algum benefício que um utilizador normal não beneficiará)



# Medição de desempenho

## Exemplos

- <https://mysqlhighavailability.com/author/vitor/>
- <https://mysqlhighavailability.com/performance-evaluation-mysql-5-7-group-replication/>
- <http://mysqlhighavailability.com/mysqlha/gr/files/perf/gr-ga-part1.pdf>
- <http://mysqlhighavailability.com/mysqlha/gr/files/perf/gr-ga-part2.pdf>