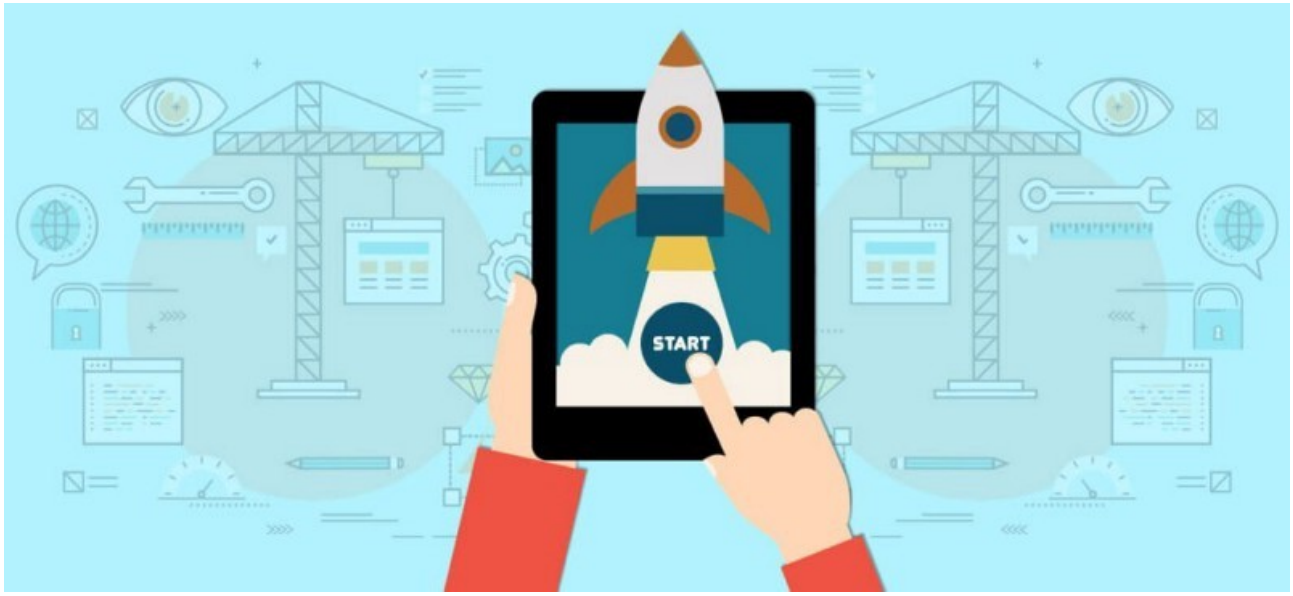
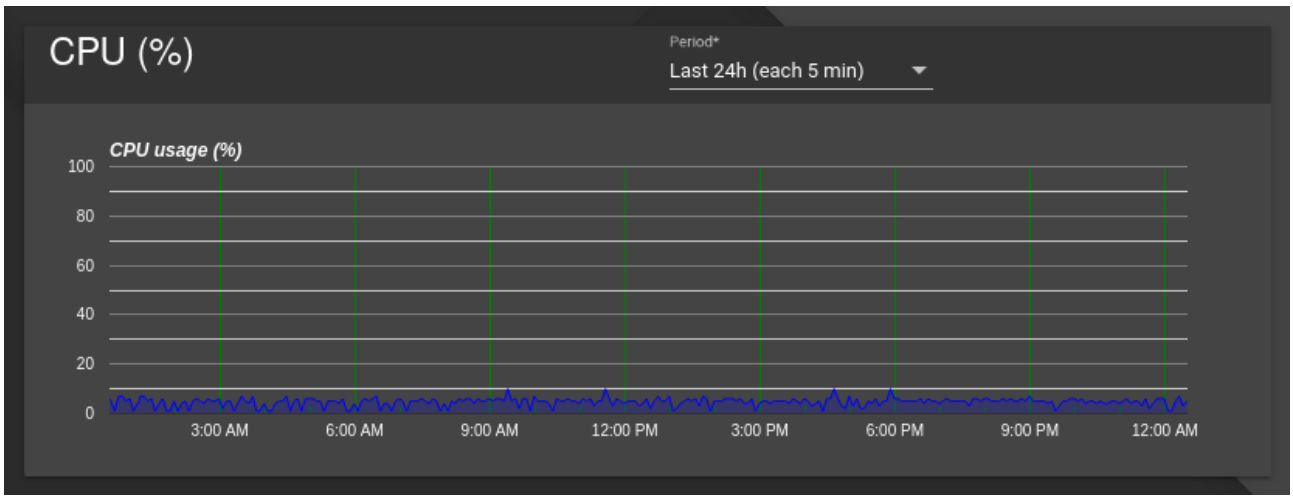


Cuidados e armadilhas na análise de performance

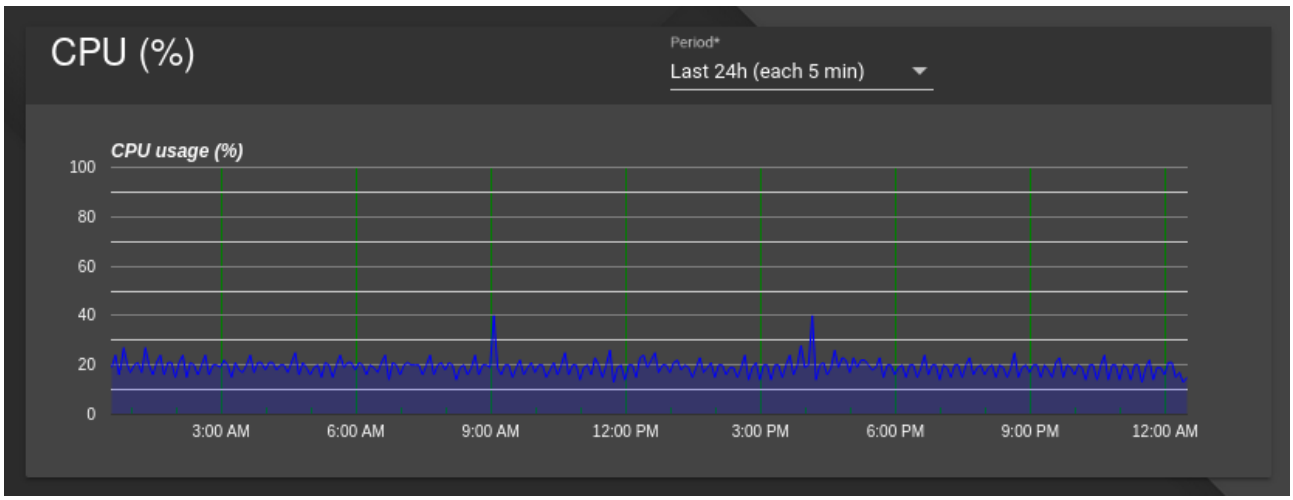
Primeiramente quero colocar que “análise de performance” é um termo que considero muito abrangente e aberto a diferentes interpretações, pois afinal, o que é performance?
Na ótica de quem está analisando a performance de uma aplicação, por exemplo, performance pode ser o tempo decorrido desde a solicitação de um usuário até o recebimento da informação.
Já em relação à análise de um servidor, performance está mais relacionado ao uso de seus recursos de hardware, como percentual de uso da CPU, memória, espaço em disco e uso das interfaces de rede.
Claro que a abordagem de monitoramento de um servidor pode abranger outros pontos, como tempo de resposta das operações de I/O, latência em comunicação de rede e outros, mas percebo que a análise dessas variáveis muitas vezes acaba sendo subestimada quando os indicadores de CPU, memória e rede não apresentam um problema aparente.

Para esse artigo vou abortar o termo “performance” no sentido de uso de recursos de um servidor.
Existem diversos monitores que mostram o uso dos recursos de um servidor e nos permitem identificar possíveis lentidões geradas pela falta de recursos, no entanto, a análise dessas informações de forma isolada, sem considerar outras variáveis, pode esconder algumas armadilhas, as quais deviríamos analisar com um certo cuidado.

Vou dar um exemplo hipotético, onde queremos migrar um servidor de uma estrutura on-premise para uma estrutura em Cloud.
Para tanto, se analisa o gráfico de uso de CPU de um servidor com 4 vCPUs e se constata uma média de utilização de CPU de 5% e pico máximo de 10%.
Em uma análise isolada dessas informações concluímos que, para atender as demandas atuais, podemos dimensionar um novo servidor na nuvem usando apenas 1 vCPU, desde que ela seja equivalente as características das vCPUs on-premise, sem com isso comprometer a performance desse servidor e nem das aplicações que nele estejam rodando.



Então criamos o novo servidor e logo no primeiro dia em produção nos é informado que usuários estão reportando que o tempo de resposta da aplicação está muito alto e que não estão conseguindo trabalhar.
Então abrimos o mesmo software de análise de uso dos recursos de hardware e identificamos que a média de uso de CPU subiu de 5% no ambiente on-premise para 20% no ambiente em Cloud e o pico máximo de uso de CPU subiu de 10% para 40%, exatamente como havia sido estimado matematicamente durante nossa análise de dimensionamento.



Pois bem, se temos uma média de uso de CPU de apenas 20%, como o tempo de resposta de nossa aplicação pode ter mudado tanto para o usuário final? Pois bem, aí está uma das armadilha que citei no título deste artigo.

Vamos compreender um pouco melhor alguns detalhes:

Primeiramente deixe-me dizer que as informações que estão plotadas no gráfico de uso de determinado recurso de hardware se tratam de médias, ou seja, são médias de uso em um determinado intervalo de tempo. Quanto maior o período das informações em um gráfico, maior deve ser o intervalo ao qual a média corresponde.
Por exemplo, em um gráfico que mostre o uso de um determinado recurso nas últimas 24h, talvez cada ponto plotado no gráfico seja a média de uso de 1 minuto. Se o gráfico de uso é do período dos últimos 7 dias, talvez cada ponto plotado no gráfico seja a média de uso de 5 minutos.

Nesse ponto cabe se fazer uma análise de como é a natureza de utilização de determinados recursos, que pode diferenciar muito de um tipo de recurso para outro.
Por exemplo, o uso de memória tende a ser mais linear, embora hajam variações quando novos elementos são alocados e removidos da memória, ainda assim, em um gráfico com o percentual do montante total de uso, a oscilação não é muito grande.
No entanto, existem outros recursos onde o uso funciona na forma de rajadas, dando picos de 0% a 100% e, posteriormente, retornando a 0% novamente.
Vou citar aqui dois desses recursos, muito presentes no dia a dia de profissionais de tecnologia e que trabalham nesse modelo de rajadas, são eles CPU e rede, dois elementos fundamentais no desempenho de praticamente qualquer sistema.

Lembro que a quase duas década atrás, quando trabalhava no desenvolvimento de sistema operacional para roteadores e, quando estava prestando consultoria ou treinamento para outras pessoas da área de redes e de telecomunicações, com uma certa frequência vinha justamente esse ponto.
Um dos questionamentos que frequentemente eram feitos era algo do tipo: *"Fizemos uma análise no uso do link de conexão com a Internet, onde temos uma média de uso em horário de produção de 40% mas, ainda assim, percebemos uma grande lentidão nos acessos se comparado a outros horários fora de produção, onde a média baixa para 10% de uso"*.
Essa lógica de raciocínio, em geral, leva ao seguinte questionamento: *"Se estou usando 40% do meu link, ou seja, ainda me resta 60% livre, como posso ter lentidão? Ou ainda, como poderia ter alguma melhora de performance se fizer um aumento de um recurso que já esta subutilizado?"*

Se nesse momento você acaba de se fazer a mesma pergunta, não se preocupe, essa é dúvida é muito comum e agora vou explicar seus detalhes e apresentar uma outra maneira de interpretar um gráfico de uso de certos recursos.

Neste ponto precisamos aprofundar um pouco mais os detalhes técnicos de como funciona fisicamente a comunicação de rede. Mas fique tranquilo, não vamos entrar em detalhes de mais baixo nível, como uso de buffers e flags de sinalização, vou citar detalhes que acredito serem de muito mais fácil compreensão.

Imagine que você precisou transmitir um arquivo de 1GB e no período daquela transferência, que durou 2 minutos, você identificou que a média de uso no seu link de 100Mbps, que é feita no intervalo de 5 minutos, foi de 40%. Mas o que esse gráfico nos diz? Você acredita que realmente estava utilizando apenas 40% do seu link e deixando 60% livre?
Não, o que o gráfico está lhe dizendo é que, durante aquele período de 5 minutos o seu link foi utilizado a 100% da capacidade durante 40% do tempo. Sim, esse tipo de recurso não trabalha a 10, 20, 30 ou 40%, ele trabalha sempre a 100%, ou seja, se você precisa transmitir 100MB, o seu dispositivo de rede vai tentar transmiti-lo o mais rápido possível, utilizando 100% dos recursos disponíveis.
Ao meu ver, essa é a chave para mudar a visão a respeito da interpretação de um gráfico.

Então, no caso hipotético citado, os dados foram transmitidos na velocidade de 100 Mbps e, tão logo a transferência foi finalizada, ele caiu para 0.
Sendo assim, podemos pensar que se precisamos transmitir dados que demoram 30 segundos em um link de 100Mbps e isso gera um uso de 10% no meu no meu gráfico, se aumentarmos o link para 200Mbps, o tempo de transmissão desta mesma quantidade de dados deveria cair para algo em torno de 15 segundos e a média de uso para 5%.
Esse é um exemplo onde, mesmo analisando um gráfico que mostra uma média de uso de apenas 10%, dobrar a capacidade do recurso reduz pela metade o tempo necessário para efetuar a mesma tarefa.

Esse mesmo pensamento pode ser levado para a análise de uso de CPU, visto que este recurso também trabalha por rajadas, ou seja, se você tem um núcleo de processamento com capacidade de 2GHz e seu gráfico mostra uso de 20%, isto não quer dizer que a sua CPU estava trabalhando a 400MHz. Podemos pensar que ela estava operando a 2GHz durante 20% do tempo.
Dessa forma, se rodarmos uma mesma tarefa em uma CPU de apenas 1GHz, ainda que o gráfico mostre que estejamos utilizando cerca de apenas 40% dessa CPU, o tempo para conclusão desse processamento tende a ser aproximadamente o dobro.

Isso explica o porquê, mesmo com um gráfico que nos mostra que o uso de CPU passou de 5% no ambiente on-premise para 20% no ambiente em Cloud, ou seja, que teoricamente ainda teríamos 80% de CPU livre, ainda assim o tempo de resposta para o nosso usuário final ficou muito pior do que era no servidor anterior.

Outro ponto a considerarmos é a frequência e intervalos no quais nossa aplicação é chamada. Quando temos uma aplicação com uso mais constante de processamento, a análise de um gráfico de uso de recurso pode refletir um pouco melhor o que acontece com o consumo de determinado recurso, mas quando ela é solicitada de forma mais esporádica, com intervalos maiores de ociosidade, a análise isolada de um gráfico de utilização do recurso pode esconder as armadilhas mencionadas.
Por exemplo, se a aplicação recebe uma solicitação que é processada durante 30 segundos e somente após cinco minutos recebe outra requisição que, novamente, é processada durante 30 segundos, se formos analisar o gráfico de uso deste recurso, com média de 5 minutos, veremos um uso de cerca de 10% e, se essa aplicação fizer uso de multiprocessamento, poderíamos dobrar o número de núcleos e reduzir esse tempo de execução para algo em torno de 15 segundos.
Então, em um exemplo simplista, este seria um caso onde, mesmo com um gráfico de uso de CPU nos mostrando uso de apenas 10%, temos o tempo de resposta da aplicação sendo limitado pela capacidade de processamento, ainda que o nosso gráfico mostre que, hipoteticamente, teríamos 90% de CPU Livre.

Apresentei aqui um exemplo simples, sem entrar em detalhes de multiprocessamento da aplicação, concorrência de conexões, tempo aguardando por I/O, intervalos ociosos e alguns outros detalhes que são relevantes, mas o objetivos era justamente esse, explicar da forma mais simples possível alguns detalhes que considero relativamente complexos em uma análise e dimensionamento de uso de recursos. Mostrar que cada recurso trabalha de uma maneira diferente e a análise de uso de alguns deles precisa considerar outras variáveis.

É importante fazer uma análise mais abrangente do uso de certos recursos, ter uma boa visão ao interpretar os dados em um gráfico de monitoramento, ter em mente que cada ponto nele se trata de uma média, considerar outros fatores relevantes, como a natureza das aplicações e o comportamento de quem as consome, sejam eles pessoas ou sistemas.
Isso nos ajuda não só a fazermos um dimensionamento mais assertivo dos recursos necessários para um bom desempenho de nossas aplicações, mas também nos ajuda a identificar e otimizar recursos que, em uma primeira análise simplista, poderiam não parecer um gargalo.

Espero que essas informações sejam úteis e que, de alguma forma, consigam contribuir para uma melhor análise de performance e provisionamento de recursos.
Fiquem à vontade para comentar, contribuir com outros pontos e fazer observações. Toda a informação construtiva é bem-vinda e só agrega.