Avaliação de Desempenho de Sistemas

Professor: Alexandre Magno de Souza

Aluno: Ícaro Bicalho Quintão, EC - 14.1.8083

Pensamento metódico sobre performance

No artigo “Thinking Methodically about Performance”, o autor Brendan Gregg aborda métricas para problemas de desempenho, como o método USE e os anti-métodos “blame-someone-else” e “streetlight”. Pode-se ter pouca ou nenhuma pista da origem dos problemas de desempenho, sendo assim, sem um ponto de partida, o caminho o problema é analisado aleatoriamente, ou seja, tentar adivinhar onde o problema está para depois tratá-lo. Embora isso possa fornecer resultados, caso adivinhado corretamente, também pode consumir tempo, causar interrupções e ignorar outros problemas. Além de descrever os problemas de desempenho de sistemas e algumas metodologias, é proposto uma nova metodologia para abordar e resolver uma classe de problemas.

Algumas das metodologias abordadas no artigo foram:

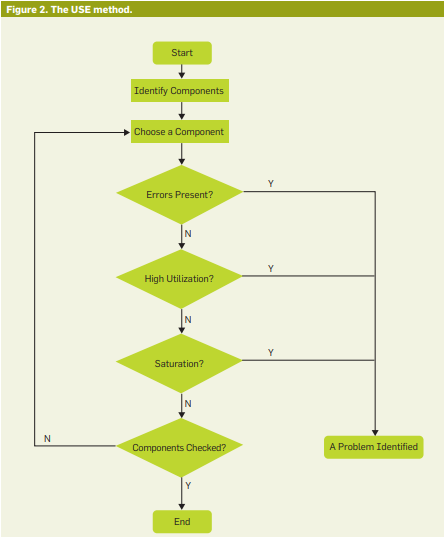
“The Blame-Someone-Else”: Ao invés de investigar problemas de desempenho, essa metodologia joga o problema para outra pessoa, ou seja, a culpa é de outra equipe de trabalho. A falta de análise de dados, ou mesmo dados para começar, pode levar à esse método.

“The Streetlight Anti-Method”: Este anti-método nada mais é do que a ausência de qualquer metodologia, ou seja é feita a análise de desempenho selecionando as ferramentas de observação que são familiares, encontradas na Internet ou encontradas aleatoriamente. As ferramentas mais familiares são executadas primeiro, mesmo que não façam mais sentido.

Aprender mais ferramentas ajuda quando se usa esse método, mas ainda é uma abordagem limitada. Certos componentes ou recursos do sistema podem ser negligenciados devido à falta de ferramentas ou métricas. Além disso, o usuário, sem saber que a exibição está incompleta, não tem como identificar "incógnitas desconhecidas".

“Problem Statement Method”: É comumente usado pela equipe de suporte para coletar informações sobre um problema, e foi adaptada para análise de desempenho. Normalmente é a primeira metodologia utilizada para tentar resolver o problema, ela consiste nada mais nada menos do que coletar uma descrição detalhada do problema que direciona uma análise mais profunda. A descrição por si só pode resolver o problema.

“USE method”: É o principal método trabalhado no artigo e tem seu foco para ser usado no início de uma investigação de problemas de desempenho, após a declaração do problema, para identificar gargalos sistêmicos rapidamente. Pode ser resumido como: para cada recurso, verifique a utilização, saturação e erros. Recurso, neste caso, significa todos os componentes funcionais do servidor físico (CPUs, discos, barramentos etc.) examinados individualmente. Alguns recursos de software podem ser examinado usando a mesma metodologia, desde que as métricas façam sentido. Utilização é a porcentagem de tempo que o recurso está ocupado atendendo trabalhar durante um intervalo de tempo específico. Enquanto ocupado, o recurso ainda pode estar capaz de aceitar mais trabalho; o grau para o qual não pode fazê-lo é identificado por saturação. Esse trabalho extra geralmente é esperando em uma fila. O esquema contido na Figura 2 retirada do artigo mostra claramente o fluxograma de funcionamento desse método.



O método USE foi desenvolvido para abordar deficiências em outras metodologias comumente usadas e é uma estratégia simples para executar uma verificação completa da integridade do sistema. Ele considera todos os recursos para evitar negligenciar os problemas e usa métricas limitadas para que possa ser seguido rapidamente. Incluindo computação em nuvem, onde muitos sistemas estão envolvidos, e precisam ser verificados. Ela é ideal para gargalos e erros, mas é uma das diversas ferramentas que podem ser utilizadas como recursos para que os problemas possam ser tratados.

O desempenho bom ou ruim é subjetivo, o que pode ser uma latência inaceitável para um usuário pode ser aceitável para outro. Depende do ponto de vista. Sem um meio de identificar claramente os problemas, pode ser difícil saber não apenas se um problema está presente, mas também quando ele é corrigido. Os problemas de desempenho também podem surgir de interações complexas entre componentes que funcionam bem isoladamente. A solução desse tipo de problema pode exigir a conexão de diversos setores dentro de um mesmo projeto.