Uma investigação sobre a eficiência energética em servidores Web virtualizados utilizando o HAProxy

Viernanryck Luciano[1], **Carlos Oliveira**[1]

[1]Instituto Federal do Rio de Janeiro (IFRJ)

1. Introdução

No cenário atual da tecnologia da informação, a eficiência e o desempenho dos servidores web desempenham um papel fundamental na entrega eficaz de conteúdo online. O balanceamento de carga emerge como fator crítico para garantir a disponibilidade contínua de serviços web e aprimorar a experiência do usuário. Este artigo tem como objetivo investigar o desempenho do balanceamento de carga em um ambiente virtualizado de servidores web usando o **HAProxy**[Edition] como balanceador. Nosso estudo visa analisar o impacto de diferentes algoritmos de balanceamento de carga, como Round Robin e Leastconn, nas métricas de desempenho do sistema. Além disso, pretendemos avaliar a eficácia desses algoritmos em lidar com cargas de trabalho variáveis e identificar as melhores práticas para otimizar o balanceamento de carga em ambientes similares.

2. Descrição do cluster virtualizado

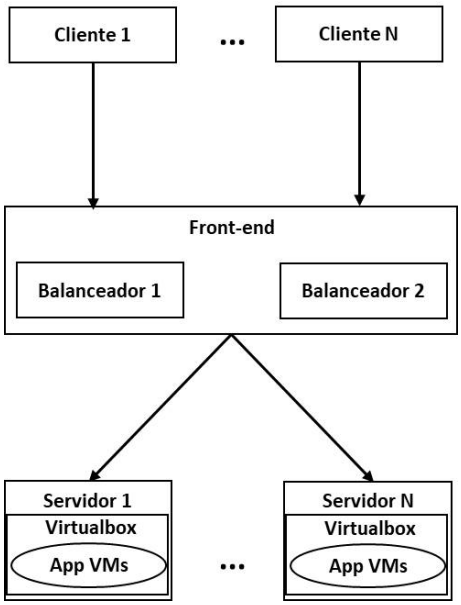

Figura 1. Arquitetura do cluster de servidores

Nossa arquitetura (apresentada na Figura 1) consiste em um cluster de servidores web replicados. O cluster apresenta uma visão única aos clientes por meio de duas máquinas front-end, que distribuem as solicitações recebidas entre os servidores que processam as solicitações. Esses servidores executam a distribuição Linux **Debian 11** x64 sem interface gráfica. Para realizar o balanceamento de carga utilizamos o HAProxy, utilizando 2 diferentes algoritmos de balanceamento de carga: (1) RoundRobin, que distribui as solicitações de forma igualitária, independentemente da carga de trabalho de

cada servidor, enquanto o (2) Leastconn prioriza servidores com menos conexões ativas no momento. Portanto, o Leastconn seria o mais adequado quando a carga de trabalho não é uniforme, e alguns servidores podem estar mais sobrecarregados do que outros. Isso ajuda a evitar a sobrecarga de servidores individuais, assegurando uma distribuição mais equilibrada da carga. O testbed utilizado para implementar a arquitetura proposta é apresentado na Figura 2. As solicitações web dos clientes são redirecionadas para as máquinas virtuais (VMs) correspondentes que executam o servidores web. Cada VM possui uma cópia de um script PHP simples vinculado que faz uso da CPU para caracterizar uma aplicação web. Para gerar carga para a aplicação web nós utilizamos uma máquina com jMeter [Foundation]. Todas as máquinas do cluster possuem a mesma configuração (Intel Pentium Gold 5400 (Dual core 3,7Ghz. Quad core núcleos lógicos), Memória DDR3 8GB, Disco rígido 500GB, Rede em 1Gbs full duplex)). As máquinas que rodam os balanceadores e os servidores executam a distribuição Linux Debian 11 x64 sem interface gráfica, enquanto o gerador de carga executa o Windows 10.

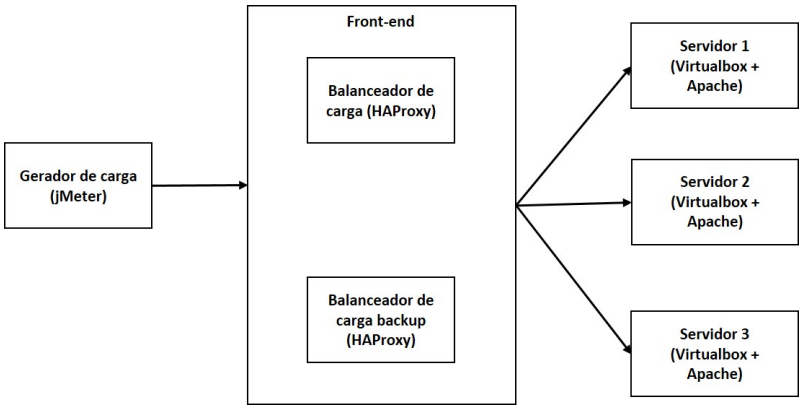

Figura 2. Configuração do cluster

3. Experimentos

Para analisar o desempenho de nossa infraestrutura de servidores, realizamos uma série de experimentos em cenários variados, alternando entre diferentes estratégias de balanceamento de carga e modos de teste. Por uma questão de falta de páginas não podemos apresentar os experimentos em detalhes, mas eles visaram oferecer uma visão abrangente de como nosso sistema opera em diferentes condições, destacando sua capacidade de manter um serviço confiável e de alta disponibilidade. Em nossos experimentos, empregamos o JMeter [Foundation] para simular a interação de diversos usuários com nossos servidores, resultando em uma significativa carga de processamento para nossa infraestrutura. Através dessas simulações, buscamos avaliar o desempenho e a capacidade de resposta de nossos sistemas sob condições de alto tráfego, replicando um cenário realista de utilização.

Nos testes realizados, simulamos a atividade de um grupo composto por 1000 usuários que acessam uma página web em PHP. Essa página é projetada para gerar carga de CPU por meio de um simples loop. Cada usuário faz 10 acessos, resultando em um total de 10.000 acessos. Configuramos o Jmeter para distribuir esses acessos ao longo de 60 segundos. No contexto desses testes, consideramos que ocorre um erro quando: (1) o tempo de conexão com o servidor excede 21 segundos (21000ms); ou (2) o servidor

demorar mais de 50 segundos (50000ms) para criar a página solicitada. Para calcular o quanto foi gasto com energia elétrica levamos em consideração a tarifa residencial normal (B1) vigentes para o Rio de Janeiro pela concessionária Enel que no momento do teste era de R$ 0,88834 kWh na bandeira verde (os valores de impostos não foram levados em consideração para os cálculos). O monitoramento do consumo de energia das CPUs em cada servidor é realizado em intervalos de um segundo, e os valores são registrados regularmente.

Os experimentos foram realizados em 3 diferentes cenários: (1) servidor único: Neste primeiro teste, não houve uso de balanceador; em vez disso, um único servidor recebeu toda a carga. O teste teve uma duração de 307 segundos, o equivalente a cerca de 5 minutos, e obteve uma taxa de sucesso de 72% nas respostas, com uma média de 12 segundos (12544 ms) para cada resposta. Durante esse período, A CPU atingiu uma temperatura máxima de 43°C, trabalhando em uma potência de média de 16.17 Watts, o que se traduz em um custo aproximado de R$ 0.001226 centavos durante esse teste; (2) Round-Robin: Nesta série de testes, nosso balanceador de carga entrou em operação, distribuindo o trabalho entre nossos três servidores por meio do algoritmo Round-Robin. A duração total do teste foi de 126 segundos, com um índice de sucesso de 95% ou seja 23% a mais que o primeiro teste, obtivemos apenas 494 erros contra 2755 no teste anterior. A média do tempo de resposta foi de aproximadamente 6 segundos (5931 ms), metade do tempo anterior. Durante esse período, o consumo médio de energia pelos processadores foi de 16 watts por segundo, operando em carga máxima, e a temperatura máxima atingida foi de 44°C no Servidor 3. A potência somada de todos os CPUs foi de 48.70 Watts, Esse consumo se traduz em um custo aproximado de R$ 0.001515 centavos durante o teste; (3) Failover: Neste teste, aplicamos os mesmos parâmetros utilizados no teste anterior, mas desta vez, deliberadamente desativamos nosso balanceador de carga principal (10.0.0.1) para simular uma interrupção. Como resultado, o nosso servidor de backup (10.0.0.2) entrou em ação, ativando o failover. Assim, ele assumiu o endereço IP 10.0.0.1 e começou a responder às requisições, distribuindo a carga. Nesse cenário, nosso teste teve uma duração de 166 segundos, um pouco mais de três minutos, apenas 40 segundos a mais em relação ao teste anterior. Alcançamos uma taxa de sucesso de 87%, com uma média de tempo de resposta de aproximadamente 6 segundos (5950 ms). Entre os 1278 erros registrados, 923 deles foram identificados como **'Non HTTP response code: java.net.SocketTimeoutException/Non HTTP response message: Read timed out'**, correspondendo às requisições perdidas devido ao desligamento do balanceador principal. Os outros 295 erros indicaram que as requisições foram atendidas, mas com um tempo de resposta significativamente alto. A temperatura máxima registrada nos CPUs atingiu 44°C, enquanto a potência média de consumo dos CPUs foi de 11 watts, com uma potência pico de 18 watts no servidor 3. A potência somada de todos os servidores foi de 33.01 Watts, resultando em um custo de energia de aproximadamente R$ 0.001352 centavos durante o período de execução do teste; (4) Leastconn: A configuração do HAProxy foi modificada para utilizar o algoritmo Leastconn. Em seguida, conduzimos uma bateria de testes sob essa nova configuração. Este teste teve uma duração de 130 segundos, o que equivale a pouco mais de dois minutos, e apresentou uma taxa de sucesso de 94,5%, registrando somente 544 erros ao longo desse período. Além disso, a média do tempo de resposta foi de aproximadamente cinco segundos (5458ms). Durante essa avaliação, o Servidor 3 alcançou a temperatura mais alta, atingindo 45°C, enquanto

Cenário	Tempo	Sucesso	Erros	Média Latência	Potência CPUs	Gasto
1	307s	72,45%	2755	12544ms	16.17 W	R$ 0.001226
2	126s	95,06%	494	5931ms	48.70 W	R$0.001515
3	166s	87,22%	1278	5950ms	33.01 W	R$ 0.001352
4	130s	94,56%	544	5458ms	49.41 W	R$ 0.001593

Tabela 1. Comparativo entre os testes

os processadores mantiveram uma potência média de 16 watts por segundo. A potência somada foi de 49.41 Watts, Como resultado, o consumo de energia foi calculado em R$ 0.001593 centavos.

4. Conclusão

O foco deste trabalho está no consumo energético dos servidores web. Como era de se esperar, no cenário 1, em que menos máquinas estão ligadas, o consumo de energia foi menor. Contudo, a experiência do usuários foi muito pior (como pode ser observado na tabela 1). Além do consumo de energia, a empresa que tem um serviço Web precisa cuidar da experiência do usuário ou o mesmo não retornará ao site e o serviço perderá seu propósito [Guerra] . Os usuários tiveram uma experiência significativamente melhor nos cenários em que as requisições foram atendidas por mais de um servidor. Isso pode ser observado nas colunas sucesso, erros e média latência na tabela 1. Como era de se esperar, houve um impacto na experiência do usuário quando o balanceador de backup precisou ser acionado. Contudo, a variação foi pequena (com uma taxa de sucesso apenas 8% menor que o cenário 2 e tempo de resposta igual ao cenário 2). No entanto, quando se compara o cenário em que o balanceador caiu (cenário 3) com o cenário 1 (em que havia um único servidor), as métricas para avaliar a experiência do usuário também são melhores no cenário 3.

Ao comparar os dois cenários em que a única diferença nos experimentos foi o algoritmo utilizado para o balanceamento de carga (cenários 2 e 4), observamos que a diferença foi mínima. No entanto, notou-se um desempenho ligeiramente inferior com o algoritmo Leastconn. Contudo, é preciso observar que, neste trabalho, estamos comparando experimentos realizados em um intervalo de tempo muito curto. Em um cenário real em que o serviço Web é prestado 24 horas por dia, a diferença pode ser relevante no consumo de energia elétrica. Foi observado que à medida que a carga de trabalho aumenta nos servidores, a temperatura da CPU também aumenta, o que resulta em um maior consumo de energia. Nos experimentos pudemos observar que o aumento do consumo de energia segue o aumento da temperatura. Além disso, quando a CPU começa a gerar mais calor, o sistema automaticamente aumenta a rotação das ventoinhas para resfriá-la, o que, por sua vez, gera ainda mais consumo de energia.

Referências

Edition, H. C. `https://www.haproxy.org/`. Acessado em 22/04/2023.

Foundation, A. S. `https://jmeter.apache.org/`. Acessado em 22/03/2023.

Guerra, B. `https://www.mazag.com.br/marketing-digital/a-importancia-da-experiencia-do-usuario-para-sua-empresa/`. Acessado em 19/10/2023.