Combinando Elasticidade Reativa e Proativa para Aumentar o Desempenho de Aplicações HPC

Vinicius F. Rodrigues[1], **Rodrigo da R. Righi**[1]

[1]Programa de Pós-Graduação em Computação Aplicada – UNISINOS

{vfrodrigues,rrrighi}@unisinos.br

Resumo. *A elasticidade de recursos em ambientes de Computação em Nuvem é uma característica explorada por aplicações que demandam alto desempenho e gerenciamento de recursos sob demanda. Porém, a definição dos parâmetros de configuração dessa funcionalidade dependem diretamente do conhecimento e experiência do usuário. Neste contexto, este artigo busca apresentar uma análise detalhada de desempenho da técnica de elasticidade híbrida Live Thresholding em que não há necessidade de configuração de parâmetros.*

1. Introdução

A elasticidade de recursos em ambientes de Computação em Nuvem oferece inúmeros benefícios para aplicações que demandam variável poder computacional ao longo de sua execução [Al-Dhuraibi et al. 2017]. Porém, estratégias de elasticidade geralmente necessitam a correta configuração de parâmetros, o que não é uma tarefa trivial. A maioria das soluções emprega o controle de elasticidade de forma reativa ou proativa/preditiva [Farokhi et al. 2015, Nikravesh et al. 2015]. No presente trabalho, é proposta a técnica de elasticidade Live Thresholding (LT), a qual combina os dois métodos. LT calcula automaticamente os limites inferior (T_i) e superior (T_s) inicializando seus valores em 0% e 100% respectivamente. A carga de processamento (CPS) do ambiente, a qual é calculada periodicamente, baseia-se no consumo de CPU dos recursos e é utilizada para definição dos limites e disparo de ações de elasticidade. Em cada período de monitoramento, é calculada a variação da carga atual ($CPS(o)$) para carga do período anterior ($CPS(o-1)$), atribuindo esse valor para ΔCPS ($\Delta CPS = CPS(o) - CPS(o-1)$). Os limites são reajustados utilizando esse valor de variação em que, se negativo, seu módulo é adicionado ao limite inferior. Caso contrário o módulo da variação é subtraído do limite superior. Em cada período, após a reconfiguração dos limites, ações de elasticidade são disparadas se CPS violar um dos limites. Para realização de experimentos, foram investigadas 6 abordagens diferentes A_z ($A_z|z \in \{a, b, c, d, e, f\}$) para tratar a adaptatividade dos limites após uma ação de elasticidade em um ambiente de Nuvem privada. Quando o limite inferior é violado o valor desse limite pode ser reconfigurado usando uma das estratégias apresentadas na Equação 1. Para a reconfiguração do limite superior, as estratégias investigadas são apresentadas pela Equação 2.

$$\mathrm{T}_i = \begin{cases} 0 & para\ A_a \\ \frac{CPS(o)}{2} & para\ A_b \\ CPS(o-1) - \left|\frac{CPS(o-1)-CPS(o)}{2}\right| & para\ A_c \end{cases} \quad (1)$$

$$\mathrm{T}_s = \begin{cases} 1 & para\ A_d \\ CPS(o) + \frac{1-CPS(o)}{2} & para\ A_e \\ CPS(o-1) + \left|\frac{CPS(o-1)-CPS(o)}{2}\right| & para\ A_f \end{cases} \quad (2)$$

2. Resultados Preliminares

A Figura 1 apresenta a execução de uma aplicação paralela com todas as combinações de LT. As figuras (a), (b) e (c) apresentam comportamento similar e o uso da estratégia A_a.

A variação da estratégia de reconfiguração do limite superior causou variações somente em momentos em que recursos foram adicionados. Nos cenários das figuras (d), (e) e (f) o número de recursos disponíveis foi diferente para cada um. A maior diferença ocorreu no cenário (e) em que a quantidade disponível de recursos chegou a 8 VMs quando a carga já estava diminuindo próximo aos 1000 segundos. Isso ocorreu pois uma nova ação de elasticidade já havia sido iniciada e os recursos ficaram disponíveis somente após esse ponto. Da mesma forma, os cenários (g), (h) e (i) empregam A_c com diferentes estratégias para calcular o limite superior. Enquanto em (g) e (i) duas ações de elasticidade removeram recursos no primeiro pico de queda da carga, em (h) nenhum recurso foi removido nesse pico. Isso ocorreu pois quando a carga começou a diminuir uma ação de elasticidade já estava sendo realizada. Como duas ações de elasticidade não são disparadas concorretemente, nenhuma operação para remoção de recursos foi permitida naquele intervalo.

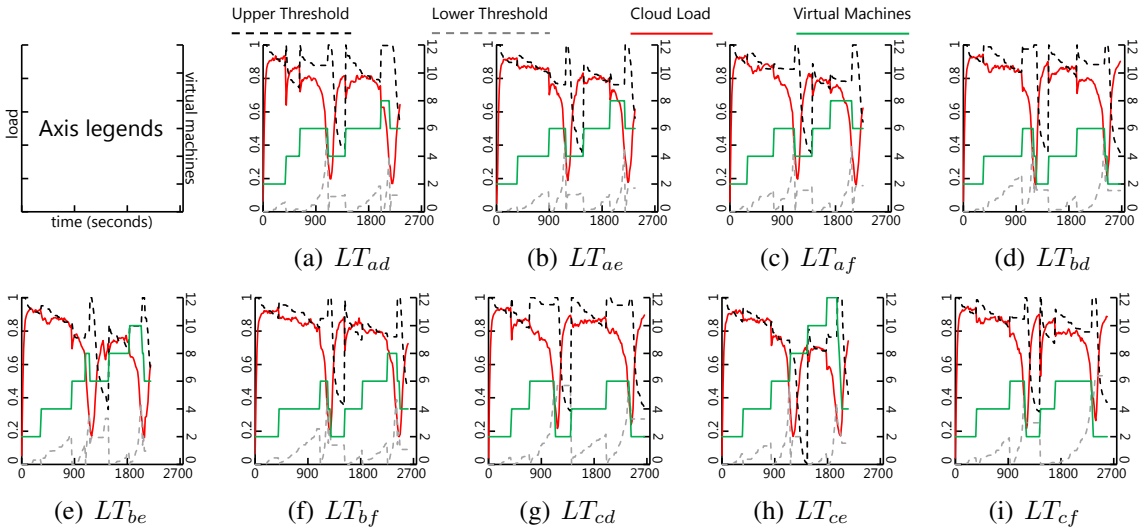

Figura 1. Comportamento da elasticidade para uma aplicação de carga variável.

3. Conclusão

Este artigo apresentou diferentes modelos de configuração de limites de elasticidade explorando a combinação de técnicas reativas e proativas. Os limites são adaptados automaticamente a cada atividade periódica de monitoramento, bem como quando uma ação de elasticidade ocorre. Resultados com uma aplicação que apresenta uma carga variável demonstraram que os melhores resultados foram obtidos pela estratégia LT_{ce}.

Referências

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., and Merle, P. (2017). Elasticity in cloud computing: State of the art and research challenges. *IEEE Transactions on Services Computing*, PP(99):1–1.

Farokhi, S., Jamshidi, P., Brandic, I., and Elmroth, E. (2015). Self-adaptation challenges for cloud-based applications : A control theoretic perspective. In *10th International Workshop on Feedback Computing (Feedback Computing 2015)*. ACM.

Nikravesh, A. Y., Ajila, S. A., and Lung, C.-H. (2015). Towards an autonomic auto-scaling prediction system for cloud resource provisioning. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '15, pages 35–45, Piscataway, NJ, USA. IEEE Press.