Análise da Virtualização do TCP e da Marcação de Pacotes RED-ECN para Aplicações Hadoop MapReduce

Vilson Moro[1], **Maurício Pillon**[1], **Charles Miers**[1], **Guilherme Koslovski**[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPGCA) – LabP2D
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

Resumo. *Data centers com múltiplos inquilinos hospedam uma diversidade de algoritmos de controle de congestionamento do Transmission Control Protocol (TCP). A literatura especializada indicou que virtualizar o TCP e aplicar marcações de congestionamento nos comutadores pode contornar o problema causado pelas diferentes configurações. O presente trabalho analisa os efeitos e demonstra a ineficiência das técnicas para aplicações Hadoop MapReduce.*

1. Introdução

Os *Data Center*s (DCs) de nuvens computacionais se tornaram um ambiente no qual múltiplos inquilinos hospedam aplicações que divergem em necessidades tecnológicas e configurações. Neste cenário, algoritmos TCP, com diferentes formas de inferir a ocorrência de congestionamento passam a coexistir. Algoritmos recentes propõem a aplicação de *Explicit Congestion Notification* (ECN) e *Random Early Detection* (RED) marcando pacotes no núcleo da rede quando um determinado limite é atingido. Desta forma, o remetente reduz o envio de dados baseado na quantidade de pacotes com marcações ECN recebidos, sem que ocorra perdas de pacotes ou confirmações duplicadas, conforme ocorre com os algoritmos tradicionais.

Nesse contexto, a Virtualização do Controle de Congestionamento (VCC) foi proposta como uma alternativa para uniformizar os algoritmos de controle em execução nas máquinas virtuais dos inquilinos [Cronkite-Ratcliff et al. 2016, He et al. 2016]. Os provedores possuem acesso administrativo aos *switches* virtuais e hipervisores, responsáveis por encaminhar os pacotes. Assim, uma camada de virtualização pode interceptar o tráfego não padronizado (otimizado ou agressivo) e manipular, quando necessário, para atender aos requisitos dos protocolos do DC, inclusive interpretando as marcações ECN. Assim, nenhuma alteração é realizada no TCP da Máquina Virtual (MV) do inquilino.

As técnicas de virtualização propostas pela literatura foram estudadas com dados sintéticos [Cronkite-Ratcliff et al. 2016, He et al. 2016]. A proposta do presente trabalho é usar rastros de execuções de servidores de produção. Foi escolhida a aplicação *Hadoop MapReduce* (HMR) por sua popularidade na manipulação de dados. Para emular a aplicação foi utilizada a aplicação MRemu [Neves et al. 2015] que permite repetir a execução dos rastros de execuções, reproduzindo a comunicação em diferentes configurações. Os resultados mostram (através da análise das métricas tempo de execução e da perda de pacotes) o impacto que a aplicação HMR sofre quando submetida a diferentes cenários de configuração do TCP, mesmo com a virtualização do controle de congestionamento em execução.

2. Fundamentação e Motivação

Para fundamentação e motivação do trabalho, os conceitos de VCC e marcação de pacotes são discutidos. Ainda, os requisitos de comunicação do HMR são elencados.

2.1. Comunicação de Dados em HMR

É importante ressaltar que 33% do tempo de execução do HMR é atribuído à tarefas de comunicação [Neves et al. 2015]. Como DCs hospedam aplicações com padrões distintos de comunicação, àquelas baseadas em particionamento, processamento e agregação de informações trafegam dados de controle sensíveis à latência, bem como tráfego de fluxos para sincronização de massas de dados. Ou seja, os fluxos sensíveis à latência concorrem com o tráfego de *background*, constituído por pequenos e grandes fluxos. Analisando a ocupação dos *buffers* em *switches* e servidores, fluxos maiores que 1MB possuem baixa multiplexação e consomem grande parcela do espaço disponível [Alizadeh et al. 2010]. Consequentemente, induzem a formação de filas e o aumento da latência para os demais fluxos. Ainda, o tamanho dos pacotes de dados manipulados pelo HMR, estão entre 7MB e 9MB.

Estudos sobre o DC do *Facebook* avaliaram o comportamento de 3 padrões distintos de tráfego: *(i)* aplicação HMR; *(ii)* máquinas executando serviços de requisições web e *(iii) cache* de dados [Roy et al. 2015]. Os autores constataram que a escolha da topologia depende da demanda de dados dos serviços que oferecem. Ainda, identificaram uma tendência de baixa utilização da rede nas bordas, enquanto que nas camadas de agregação e núcleo ocorre uma grande demanda, criando pontos de congestionamento. Alguns padrões tem um tráfego mais intenso intra-rack enquanto que outros dependem de comunicação com outros *clusters* em diferentes *racks*. O tráfego de dados do HMR neste cenário, ocorre predominantemente em rajada curtas, apresentando uma variação de demanda através dos servidores, bem como em relação ao tempo.

2.2. Marcação de Pacotes e VCC

O ECN permite ao núcleo da rede sinalizar para as extremidades uma possível situação de congestionamento antes que ocorra perda de pacotes. Quando um limite de ocupação de fila é atingido, a ocorrência é marcada no cabeçalho do pacote e encaminhado para o destinatário, que acrescenta a informação no pacote de resposta enviado ao remetente. Então, o remetente reduz a janela de dados. Para implementação do ECN, o RED é usado para gerenciamento das filas, estabelecendo parâmetros, como o limite mínimo e máximo, além da probabilidade de marcação dos pacotes que excedem os limites.

A virtualização cria uma camada que permite ao DC a utilização de um algoritmo atualizado para controle de congestionamento, baseado em ECN, e com isso processar a comunicação dos diferentes algoritmos usado pelos clientes. Em resumo, o hipervisor monitora e participa do estabelecimento das conexões TCP. Quando um cliente com TCP legado solicita uma conexão, o hipervisor altera o cabeçalho do pacote alterando o bit *ECN Echo* (ECE). O destinatário confirma a solicitação para o hipervisor. Ao repassar a informação para o remente, as informações pertinentes ao ECN são removidas. Ao receber os pacotes de dados o hipervisor monitora a fila. Na ocorrência de congestionamento, o bit ECE é ativado. Quando o destinatário recebe pacotes com marcação ECN, ele ativa e empacota essa informação no pacote *Acknowledgement* (ACK). O hipervisor

do remetente, ao receber esse bit ativado, repassa a informação manipulando o *Receive Window* (RWND). Ou seja, para induzir a desaceleração de transmissão do hospedeiro sem aplicar uma técnica intrusiva, o controle de congestionamento é aplicado sobre a janela de recepção. Assim, as informações internamente aferidas pelo algoritmo legado sobre a janela *Congestion Window* (CWND) permanecem inalteradas. Nativamente, o TCP verifica $min(cwnd, rwnd)$ para identificar o volume de dados que podem ser trafegados em um determinado instante. Ou seja, manipulando as informações no pacote que será entregue a MV é possível reduzir o volume de dados para os próximos envios.

3. Análise Experimental

O objetivo desta análise é verificar o impacto que a marcação ECN/RED pode causar na aplicação HMR virtualizando o TCP aprofundando a discussão sobre a aplicabilidade de VCC em cenários compartilhados. As métricas utilizadas para a análise foram o tempo de execução da aplicação HMR e a perda de pacotes.

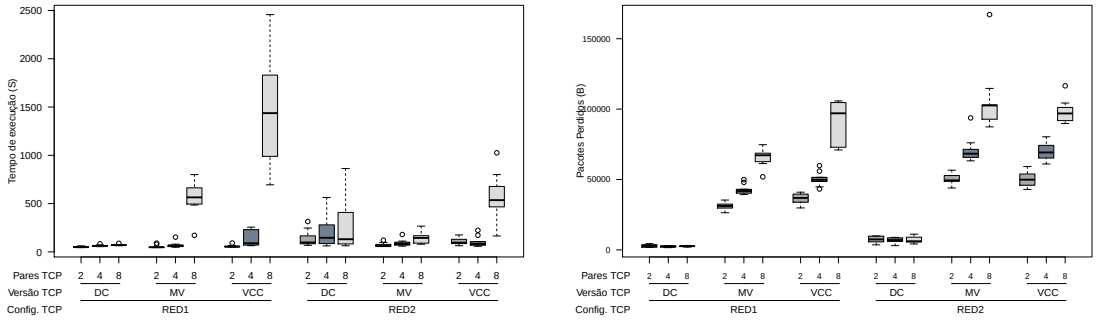
3.1. Ambiente Experimental

A topologia utilizada no experimento é formada por 2 *switches* interconectados por um enlace de 1 Gbps. Cada *switch* conecta 8 servidores (1 Gbps). A topologia simplifica a ocorrência de gargalo, representando um cenário comum em redes de DC. Foi utilizada a ferramenta MRemu [Neves et al. 2015] para emular a comunicação do HMR, enquanto a ferramenta *iperf* foi utilizada para injetar tráfego de *background* concorrendo pelos recursos. A aplicação HMR foi executada em três cenários de configuração do TCP: *(i)* atualizado (DC), *(ii)* legado (MV) e *(iii)* virtualizado (VCC). No entanto, o tráfego de *background* foi sempre executado com TCP atualizado. Para analisar o funcionamento do monitoramento de fila, duas configurações RED foram aplicadas: *(i)* RED1 com limite máximo e mínimo de ocupação da fila equivalente a 90000 bytes. *(ii)* RED2 com limite mínimo de 30000 e máximo de 90000 bytes, ou seja com um intervalo de marcação maior. Ao todo foram realizados 6 experimentos e cada rodada de testes foi executada 10 vezes. Foram introduzidos, inicialmente 2 pares de *background* e na sequência 4 e 8 pares, com configuração do TCP consciente de ECN.

3.2. Análise dos resultados

O gráfico de diagrama de caixas mostra a variação das execuções. Analisando inicialmente o tempo de execução, Figura 1(a), na configuração com cenário atualizado (DC) a configuração de gerenciamento de fila RED2 apresentou um tempo de execução superior ao RED1. No entanto, em um cenário legado (MV) a configuração RED1 apresentou um aumento significativo no tempo de execução quando executou com 8 pares, demonstrando uma predominância na ocupação da rede de aplicações com algoritmo atualizado afetando o tempo do HMR. Efeito semelhante pode ser observado no cenário virtualizado (VCC), no qual a execução com 8 pares mostrou um tempo de execução pior do que apresentado no cenário legado.

As perdas de pacotes são representadas na Figura 1(b). O cenário legado (MV) apresenta um aumento da perda conforme mais pares de execução são introduzidos. Sobretudo, com RED1 ocorre um aumento expressivo da perda com 8 pares, aumento esse que é ainda maior no cenário virtualizado. A configuração RED2 apresenta um maior



(a) Tempo de execução do HMR. (b) Pacotes perdidos.

Figura 1. Tempo de execução do HMR e perda de pacotes variando o número de pares cliente-servidor de tráfego de *background*.

número de perdas em relação a configuração RED1 em cenário legado. Por fim, o cenário virtualizado não conseguiu diminuir essa ocorrência.

Uma análise combinada das métricas aponta a ineficiência de VCC, RED e ECN no cenário estudado. É evidente que o desempenho é diretamente relacionado com as configurações RED e ECN, entretanto, a literatura não apresenta uma configuração canônica devido as particularidades de cada aplicação (*e.g.*, tamanho dos fluxos, tráfego em rajadas).

3.3. Conclusões e Trabalhos Futuros

A análise experimental indicou que o HMR quando configurado com variante do TCP que não reconhece o mecanismo de predição de congestionamento ECN é sensível ao número de fluxos de aplicações com configurações TCP atualizadas compartilhando a rede, sofrendo uma degradação do desempenho com o aumento do tempo de execução. Futuramente, será investigado a configuração dinâmica do RED.

Agradecimentos. Os autores agradecem ao LabP2D, UDESC e FAPESC.

Referências

Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. (2010). Datacenter TCP (DCTCP). *SIGCOMM Com. Rev.*, 41(4).

Cronkite-Ratcliff, B., Bergman, A., Vargaftik, S., Ravi, M., McKeown, N., Abraham, I., and Keslassy, I. (2016). Virtualized congestion control. In *Proc. of the SIGCOMM Conference*, pages 230–243, New York, NY, USA. ACM.

He, K., Rozner, E., Agarwal, K., Gu, Y. J., Felter, W., Carter, J., and Akella, A. (2016). Ac/dc tcp: Virtual congestion control enforcement for datacenter networks. In *SIGCOMM*, pages 244–257, New York, NY, USA. ACM.

Neves, M. V., Rose, C. A. F. D., and Katrinis, K. (2015). Mremu: An emulation-based framework for datacenter network experimentation using realistic mapreduce traffic. In *MASCOTS*, pages 174–177. IEEE.

Roy, A., Zeng, H., Bagga, J., Porter, G., and Snoeren, A. C. (2015). Inside the social network's (datacenter) network. *SIGCOMM Comput. Commun. Rev.*, 45(4):123–137.