

Avaliação de desempenho de *Heavy Hitters* utilizando P4 e XDP

Douglas L. Machado, Ariel G. de Castro, Francisco G. Vogt, Marcelo C. Luizelli

¹Universidade Federal do Pampa (UNIPAMPA)

Aplicações de monitoramento de infraestruturas de redes são cruciais para a correta operação e identificação de comportamentos e anomalias. Muitas aplicações de monitoramento se beneficiam da identificação dos fluxos de rede que mais consomem recursos (por exemplo, largura de banda) nas infraestruturas para identificar ataques, congestionamentos e para a engenharia de tráfego – apenas para mencionar alguns exemplos [Sivaraman et al. 2017]. Os fluxos de rede com o maior volume de dados são conhecidos como *Heavy Hitters*. A identificação dos *Heavy Hitters* consiste na contagem dos fluxos que mais contribuem para a utilização dos recursos nas infraestruturas [Ben Basat et al. 2017].

Com a consolidação das redes definidas por software (SDN – *Software-defined Networks*), a identificação de *Heavy Hitters* tem o potencial de ser feita de maneira *online* pelo próprio plano de dados [Sivaraman et al. 2017]. Ao se utilizar a identificação de *Heavy Hitters* no plano de dados, elimina-se a sobrecarga da transferência contínua de dados entre os planos de dados e de controle. Para além disso, a contagem dos fluxos tende a ser mais precisa que abordagens tradicionais de monitoramento baseadas em protocolos como, por exemplo, NetFlow ou SFlow. A programabilidade do plano de dados permite redefinir o processamento dos pacotes nos dispositivos, incluindo o suporte a novos protocolos e funcionalidades. A programabilidade do plano de dados emergiu recentemente com a consolidação da linguagem P4 [Bosshart et al. 2014] e de dispositivos de encaminhamento programáveis (por exemplo, Barefoot Tofino e SmartNICs). A linguagem P4 é independente de protocolo (isto é, os protocolos são definidos pelo programador) e independente de arquitetura (isto é, o compilador utilizado gera um código objeto de acordo com a arquitetura utilizada).

Outra abordagem recente para programabilidade do plano de dados é o XDP – *express Data Path* [Høiland-Jørgensen et al. 2018]. Ao contrário das aplicações escritas em P4 e compiladas para um hardware independente, XDP é um plano de dados baseado em eBPF (*Extended Berkeley Packet Filter*) [Xhonneux et al. 2018] executado diretamente no Kernel do Linux, disponível a partir da versão 4.8. A ideia do XDP consiste em adicionar um mecanismo de decisão baseado em eBPF logo após o pacote ser recebido pelo *driver* da interface de rede. Dessa forma, o pacote recebido pode ser tratado de maneira prioritária e personalizada pelo módulo de rede do sistema operacional, sem a necessidade de alocação prévia de memória para o pacote (o que é uma operação custosa), nem cópias desnecessárias entre o Kernel e o espaço de usuário.

Neste trabalho, objetiva-se conduzir uma avaliação de desempenho de algoritmos para identificar *Heavy Hitters* utilizando P4 e XDP e os algoritmos propostos por [Metwally et al. 2005, Ben Basat et al. 2017]. Pretende-se escrever uma aplicação de encaminhamento de pacotes que realize a contagem dos pacotes de acordo com os algoritmos de *Heavy Hitter*, utilizando como chave de identificação da contagem os endereços

IPs (origem/destino). Como a quantidade de pares de endereços IPs é exponencial, espera-se empregar estruturas de dados eficientes espacialmente como, por exemplo, *Counting Bloom Filter*. O programa escrito em P4 será compilado para ser executado em uma interface de rede programável (por exemplo, SmartNIC Netronome) e diretamente no Kernel do Linux a partir do compilador P4C-XDP. Com a avaliação, pretende-se estimar os custos em termos de pacotes processados por segundo, latência e vazão atingidas – as quais podem ser impactadas de acordo com a implementação do algoritmo de *Heavy Hitter* utilizado. Para além disso, pretende-se identificar os limites de utilização de memória em ambas as implementações e o consumo de ciclos de CPU.

Referências

- Ben Basat, R., Einziger, G., Friedman, R., Luizelli, M. C., and Waisbard, E. (2017). Constant time updates in hierarchical heavy hitters. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '17*, pages 127–140, New York, NY, USA. ACM.
- Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. (2014). P4: Programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95.
- Høiland-Jørgensen, T., Brouer, J. D., Borkmann, D., Fastabend, J., Herbert, T., Ahern, D., and Miller, D. (2018). The express data path: Fast programmable packet processing in the operating system kernel. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '18*, pages 54–66, New York, NY, USA. ACM.
- Metwally, A., Agrawal, D., and El Abbadi, A. (2005). Efficient computation of frequent and top-k elements in data streams. In *Proceedings of the 10th International Conference on Database Theory, ICDT'05*, pages 398–412, Berlin, Heidelberg. Springer-Verlag.
- Sivaraman, V., Narayana, S., Rottenstreich, O., Muthukrishnan, S., and Rexford, J. (2017). Heavy-hitter detection entirely in the data plane. In *Proceedings of the Symposium on SDN Research, SOSR '17*, pages 164–176, New York, NY, USA. ACM.
- Xhonneux, M., Duchene, F., and Bonaventure, O. (2018). Leveraging ebpf for programmable network functions with ipv6 segment routing. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '18*, pages 67–72, New York, NY, USA. ACM.