Utilização de Redes Definidas por Software para Melhorar o Desempenho de Aplicações MapReduce

Marcelo Veiga Neves, Cesar A. F. De Rose

Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681, Prédio 32 – Porto Alegre, RS – Brasil

marcelo.neves@acad.pucrs.br, cesar.derose@pucrs.br

Resumo. *Este trabalho propõe a utilização de redes definidas por software (SDN) para o desenvolvimento de um sistema de controle de rede sensível à aplicação (isto é, que leva em consideração informações da camada de aplicação) para dinamicamente modificar o comportamento da rede de modo a acelerar a execução de aplicações MapReduce.*

1. Introdução

MapReduce (MR) é um modelo de programação para análise de dados de larga escala bastante utilizado atualmente. MR se tornou muito popular devido à sua simplicidade, eficiência e escalabilidade. Uma de suas principais características é sua habilidade de explorar a localidade dos dados e minimizar transferências via rede. No entanto, publicações recentes têm mostrado que comunicação de rede ainda é um dos principais gargalos de desempenho em aplicações MR [Al-Fares et al. 2010]. Por exemplo, foi reportado que transferências de rede podem ter um impacto de mais de 50% no tempo total de execução de um *job* MR [Chowdhury et al. 2011], bem como limitar a escalabilidade quando múltiplos *jobs* são executados concorrentemente. Existem diversos trabalhos propondo otimizações em *frameworks* MR para melhorar o desempenho de rede. No entanto, pouco tem sido desenvolvido no sentido de modificar dinamicamente o comportamento da rede para se adaptar às necessidades das aplicações MR.

2. Motivação e Proposta

MR normalmente executa em grandes *data centers* (DC) de *commodity hardware*. A rede de tais DCs normalmente utiliza topologias *multi-rooted* que oferecem múltiplos caminhos alternativos (*multipath*) entre cada par de *hosts*. Este tipo de topologia, combinado com a emergente tecnologia de redes definidas por software (SDN), possibilita a criação de protocolos inteligentes para distribuir o tráfego entre os diferentes caminhos disponíveis. Por exemplo, utilizando OpenFlow [1], um mecanismo para SDN já disponível em alguns *switches* comerciais, é possível executar algoritmos de forma logicamente centralizada (com uma visão global da rede) e manipular diretamente as tabelas de encaminhamento dos *switches*. Isto cria uma nova oportunidade para usar informações de nível de aplicação para modificar a operação da rede dinamicamente e melhorar o desempenho das aplicações.

Apesar destas possibilidades, os protocolos de encaminhamento atuais normalmente utilizam técnicas como ECMP (Equal Cost Multipath) [Hopps 2000] para distribuir os fluxos de dados entre os múltiplos caminhos disponíveis. ECMP mapeia estaticamente fluxos em caminhos através de um *hash* dos campos do cabeçalho do pacote relacionados ao fluxo. No entanto, esse mapeamento estático não leva em consideração

[1] http://www.openflow.org

a utilização atual da rede, características de fluxos individuais ou demandas futuras de tráfego, o que pode levar ao congestionamento de caminhos e à degradação no desempenho total da rede [Al-Fares et al. 2010]. Existem algumas iniciativas recentes para superar as limitações do ECMP e realizar balanceamento de carga entre os caminhos disponíveis na rede, tais como Hedera [Al-Fares et al. 2010] e MicroTE [Benson et al. 2011]. No entanto, como utilizam apenas estatísticas de nível de rede para tomar as decisões de escalonamento de fluxo, tais sistemas apresentam uma capacidade muito limitada de reagir às modificações no tráfego específicas de aplicação (por exemplo, aplicações com comportamento de rajada, tais como aplicações MR).

Neste contexto, este trabalho propõe a utilização de SDN para o desenvolvimento de um sistema de controle de rede sensível à aplicação (isto é, que leva em consideração informações da camada de aplicação). A abordagem de rede sensível à aplicação apresenta diversos benefícios. Por exemplo, permitindo que aplicações informem suas necessidades de rede, é possível prever com precisão as demandas de tráfego futuro e rapidamente reconfigurar a camada de encaminhamento da rede para se adaptar à aplicação. Além disso, muitos padrões de tráfego em DC dependem de informações internas à aplicação. Por exemplo, a quantidade de dados transmitidos por cada *host* na fase de *shuffle* de uma aplicação MR normalmente só é conhecida em tempo de execução e depende de diferentes fatores. O sistema proposto realizará a distribuição dos fluxos entre os caminhos disponíveis de forma a melhorar o desempenho de aplicações MR específicas e maximizar a utilização total da rede. Para isso, o sistema disponibilizará uma API para permitir que aplicações MR informem suas demandas de tráfego em tempo de execução e um escalonador de fluxos utilizará essas informações para a tomada de decisão.

3. Estado Atual

Este é um trabalho em andamento. A primeira parte consistiu em estudar os padrões de comunicação de aplicações MR e identificar as causas típicas de gargalos de desempenho [Neves et al. 2012]. Na sequência, elaborou-se uma arquitetura para o sistema de controle de rede baseado em SDN. O próximo passo é a construção de um protótipo utilizando um controlador OpenFlow (por exemplo, NOX/POX [2]). A avaliação será realizada tanto utilizando emulação de rede [3] quanto experimentos em ambiente real. A distribuição de fluxos em múltiplos caminhos é um problema clássico de otimização, conhecido como *multi-commodity flow* (MCF), que já possui diversas heurísticas publicadas, as quais poderão ser aproveitadas nesse trabalho.

Referências

Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., and Vahdat, A. (2010). Hedera: dynamic flow scheduling for data center networks. In *Proceedings of the USENIX NSDI 2010 conference*, pages 19–19, Berkeley, CA, USA. USENIX Association.

Benson, T., Anand, A., Akella, A., and Zhang, M. (2011). MicroTE: Fine Grained Traffic Engineering for Data Centers. In *Proceedings of the CONEXT 2011 Conference*, pages 1–12, New York, New York, USA. ACM Press.

Chowdhury, M., Zaharia, M., Ma, J., Jordan, M. I., and Stoica, I. (2011). Managing data transfers in computer clusters with orchestra. In *Proceedings of the ACM SIGCOMM 2011 conference*, pages 98–109, New York, NY, USA. ACM.

Hopps, C. (2000). Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, IETF.

Neves, M. V., Ferreto, T., and De Rose, C. (2012). Scheduling MapReduce Jobs in HPC Clusters. In *Proceedings of the Euro-Par 2012*, pages 179–190, Berlin, Heidelberg.

[2] http://www.noxrepo.org

[3] http://github.com/mininet