강화학습 기반 링크가중치 조정 로드밸런싱 알고리즘 연구

임지윤°, 남석현 유재형, 홍원기
포항공과대학교 컴퓨터공학과

{limjiyoon, obiwan96, jhyoo78, jwkhong}@postech.ac.kr

A Study on the reinforcement learning based link weight adjustment load balancing algorithm

Jiyoon Lim°, Sukhyun Nam, Jae-Hyoung Yoo, James Won-Ki Hong
Department of Computer Science and Engineering, POSTECH

요 약

데이터센터 네트워크의 토폴로지와 트래픽 패턴은 기존의 인터넷과 다른 양상을 보인다. 특히 트래픽 패턴은 짧은 플로우 처리 완료 시간(flow completion time)과 높은 처리량(throughput)을 요구한다. 기존에 제안된 데이터센터 로드밸런싱 알고리즘의 경우 짧은 플로우에 대해 처리 능력이 부족하거나, 확장성에 한계가 있다. 본 연구는 이러한 기존 알고리즘의 문제점을 개선하기 위해 강화학습 기반 링크가중치 조정 알고리즘을 제안한다. 제안하는 알고리즘은 프로그래머블 스위치와 In-band 네트워크 텔레메트리 기법을 사용하여 높은 네트워크 가시성을 확보하고, 강화학습을 통해 모든 링크의 가중치를 조정 후 각 링크의 가중치에 따라 패킷을 전송함으로써 기존 로드밸런싱 알고리즘보다 빠른 플로우 처리 완료 시간을 갖을 것으로 기대한다.

Ⅰ. 서 론

데이터센터 내에서 동작하는 어플리케이션들은 네트워크 트래픽을 효과적으로 사용하기 위해 높은 양방향 대역폭 (bisection bandwidth)을 요구한다. 데이터센터 네트워크는 이러한 요구에 따라 기존의 인터넷과 다르게 각 호스트 간에 다중 경로를 제공하는 토폴로지를 사용하며[1], 트래픽은 기존의 인터넷과 다른 패턴을 보인다[2][3]. 즉, 10KB 이하의 짧은 플로우가 전체 플로우의 약 80%를 차지하며, 트래픽 양의 90%는 100MB 이상의 elephant 플로우에 의해 발생하고 있다. 이러한 추세에 따라 데이터센터 네트워크는 점점 더 빠른 플로우 처리 시간(flow completion time)과 높은 처리량(throughput)을 요구하고 있다. 이러한 요구사항을 만족하기 위해서는 데이터센터 네트워크의 특성을 반영하는 새로운 로드밸런싱 알고리즘이 필요하다. 기존에 주로 사용되어 온 로드밸런싱 알고리즘으로 ECMP (Equal-Cost Multipath)[3]가 있다. ECMP 는 패킷 헤더에 있는 플로우 관련 필드를 해싱하고 이를 바탕으로 트래픽을 각 최단 경로에 균등하게 분배하는 방식이다. ECMP 는 데이터센터 네트워크의 다중 경로를 활용하지만, 둘 이상의 elephant 플로우를 같은 경로로 보낼 경우 네트워크 혼잡이 발생하여 처리량이 매우 떨어진다는 한계점이 있다.

ECMP 의 한계점을 해결하기 위해 네트워크의 정보를 로드밸런싱에 활용하는 방법들이 제안되고 있다. 중앙 집중화된 컨트롤러에서 네트워크 전체의 정보를 활용하여 라우팅하는 방식[4], 각 스위치에서 네트워크 일부분의 정보만을 사용하여 라우팅하는 방식[5], 머신 러닝 기법을 사용하여 네트워크 상태를 학습하고 이를 사용하여 라우팅하는 방식[6][7]등의 연구가 진행되고 있다.

본 연구는 강화학습 기반 링크가중치 조정 로드밸런싱 알고리즘을 제안한다. 제안하는 알고리즘은 강화학습을 통해 네트워크 상태에 따라 모든 링크의 가중치를 학습한다. 또한 INT (In-band Network Telemetry)[8] 기법을 활용하여 높은 네트워크 가시성을 확보하고 이를 강화학습의 입력 값으로 활용한다. 제안하는 알고리즘은 모든 크기의 플로우에 대해 효과적으로 라우팅하여 빠른 플로우 처리 시간을 제공하고자 한다.

Ⅱ. 관련 연구

1. INT (In-band Network Telemetry)

정밀한 네트워크 정보를 모니터링 하기 위해 프로그래머블 스위치와 INT 기법을 활용하는 연구가 많이 진행되고 있다[8]. 프로그래머블 스위치는 P4[9] 라는 DSL (Domain-specific Language)로 헤더 포맷, 헤더 파싱 방법, match-action 테이블 구조 및 컨트롤 플로우를 프로그래밍 할 수 있다.

INT 는 프로그래머블 스위치를 활용하여 제어 평면의 개입없이 패킷 단위의 네트워크 정보를 모니터링하기 위해 설계된 프레임워크이다. INT 의 아키텍처 모델은 Source 스위치, Transit 스위치, Sink 스위치로 구성된다. Source 스위치는 패킷에 “telemetry instruction”이라고 불리는 헤더 필드를 삽입한다. Telemetry instruction 은 수집할 네트워크 정보를 정의하고, 수집된 정보를 어떻게 처리할지 결정한다. Transit 스위치에서는 telemetry instruction 에 의해 네트워크 디바이스에서 네트워크 정보를 추출하여 패킷에 삽입시키고, 패킷을 전송한다. Sink 스위치는 telemetry instruction 에 의해 수집된

네트워크 정보를 데이터베이스에 전송하고, 패킷에 삽입된 telemetry instruction 과 네트워크 정보를 제거한다. INT 로 수집할 수 있는 정보는 스위치 ID, 입출력 포트 번호, 패킷의 입출 시간, 홉 지연시간, 출력 포트에 대한 링크 이용률, 큐 점유율, 버퍼 점유율 등이 있다.

2. 강화학습

강화학습은 환경과 상호작용하여 피드백을 통해 학습하는 머신러닝 기법 중 하나이다. 그림 1 은 강화학습의 각 요소를 나타낸 것이다. 에이전트(Agent)는 환경을 관측하여 현재 상태(State)를 유추한다. 그리고 에이전트는 현재 상태에 대해서 자신만의 정책(Policy)을 통해 행동(Action)을 결정한다. 환경(Environment)은 에이전트의 행동에 영향을 받아 변화하고, 에이전트는 환경으로부터 행동에 따른 보상(Reward)을 얻게 된다. 강화학습의 목적은 최종적으로 얻는 보상의 총합을 최대화하는 것이다. 지도학습 (Supervised learning)과 비교하여 강화학습은 알파고와 같이 연관성이 있는 연속된 데이터에 대한 학습이나 레이블을 붙이기 어려운 데이터를 사용하는 경우에 대해 상대적 강점을 갖는다.

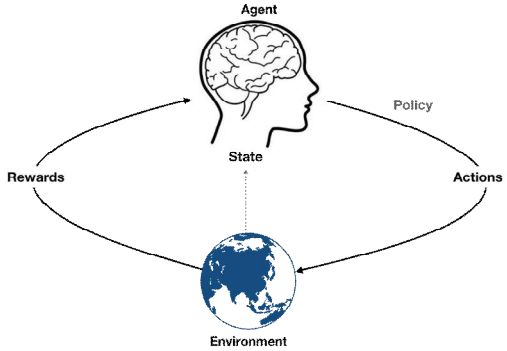

그림 1 강화학습의 요소

3. 로드밸런싱 알고리즘

중앙 집중화된 컨트롤러에서 모든 정보를 처리하는 로드밸런싱 알고리즘은 짧은 플로우에 대한 처리능력에 한계가 있다. 또한 네트워크의 일부분의 정보를 각 스위치에서 저장하고 스위치에서 라우팅하는 방식의 로드밸런싱 알고리즘은 네트워크 정보를 오버헤드가 크다는 문제가 있다. 다음은 이러한 문제를 해결하기 위해 제시된 머신러닝 기반 로드밸런싱 알고리즘들이다.

RILNET[7]은 로드밸런싱 알고리즘에 강화학습을 적용한 기법이다. RILNET 은 모든 edge 스위치 페어에 대해 플로우의 대역폭을 관측한다. 관측값은 합성곱 신경망(Convolution Neural Network)으로 상태를 축약한다. RILNET 의 action 은 모든 edge 스위치 페어에 대한 경로 가중치이다. 학습 알고리즘으로는 DDPG[10]를 사용하여 모든 edge 스위치 페어에 대해서 병목에 해당하는 링크 이용률의 평균값을 최소화하도록 학습시킨다. RILNET 은 강화학습에 대한 오버헤드가 크기 때문에 짧은 플로우에 대한 결정이 제대로 이루어지지 않는다는 한계가 있다.

HMMLB[8]는 은닉 마르코프 모형(Hidden Markov Model)을 로드밸런싱에 활용한다. 은닉 마르코프 모형을 통해 네트워크의 일부 스위치에서 추출한 링크 대역폭 데이터를 네트워크 전체 대역폭으로 추론한다. 그리고 추론한 네트워크 전체 대역폭을 기반으로 패킷 경로를 결정한다. HMMLB 는 Hedera[4] 대비 비슷한 성능을 유지하면서 시간 오버헤드를 유의미하게 감소시켰다.

본 연구는 INT 와 강화학습을 활용하는 로드밸런싱 알고리즘을 제안한다. 제안하는 알고리즘은 INT 를 사용하여 플로우에 대한 네트워크 정보를 수집하고, 순환 신경망[15]을 사용하여 연속적인 플로우 데이터로부터 네트워크 상태로 추론한다. 또한 추론한 네트워크 상태로부터 모든 링크에 대한 가중치를 출력한다. 제안하는 알고리즘은 학습에 사용하는 뉴럴 네트워크의 크기를 감소시킴으로써 더 빠른 라우팅이 가능하고, 짧은 플로우에 대해서도 경로를 결정할 수 있게 한다.

Ⅲ. 강화학습기반 알고리즘 제안

본 논문에서는 강화학습을 통해 각 스위치 별 출력 포트에 대한 가중치를 결정하는 로드밸런싱 알고리즘을 제안한다. 그림 2 는 제안하는 알고리즘 모델을 나타낸다. 알고리즘은 1) 네트워크 정보 수집 단계 2) 강화학습 단계로 이루어져 있다.

네트워크 수집 단계에서는 프로그래머블 스위치와 INT 기법을 활용하여 네트워크 정보를 수집하는 단계이다. 데이터평면에서는 프로그래머블 스위치에 P4 프로그램이 배포되어 다음과 같이 동작한다. 가장 먼저 각 스위치에 Source 스위치, Transit 스위치, Sink 스위치의 역할을 부여한다. Edge 스위치는 Source 스위치와 Sink 스위치의 역할을 한다. 패킷이 edge 스위치를 지날 경우 만약 해당 패킷에 telemetry instruction 이 삽입되어 있지 않다면 telemetry instruction 이 헤더 필드에 삽입된다. 만약 패킷에 telemetry instruction 이 삽입되어 있다면 삽입된 telemetry instruction 과 수집한 네트워크 정보를 추출하여 데이터베이스로 전송한다. aggregation 스위치와 core 스위치는 Transit 스위치가 되어 패킷이 해당 스위치를 지날 때마다 네트워크 디바이스로부터 네트워크 정보를 수집하여 패킷에 삽입한다. INT 기법으로 수집하는 네트워크 정보로는 각 스위치의 홉 지연시간, 버퍼 점유율, 링크 별 큐 점유율, 링크 이용률, 링크 대역폭 등이 있다.

강화학습 단계에서는 수집된 네트워크 정보를 바탕으로 모든 링크에 대한 가중치를 계산하고 이를 제어평면에 전달한다. 강화학습의 요소는 다음과 같이 정의한다. 에이전트는 데이터평면으로부터 수집한 INT 데이터를 관측한다. INT 데이터는 T 시간 동안 수집한 과거 K 개의 플로우 데이터를 말한다. 플로우 데이터는 각 링크의 사용량, 각 링크의 대역폭, 각 스위치의 버퍼 사용량, 각 스위치의 홉 지연시간을 사용한다. 네트워크 상태는 네트워크 전체의 링크 이용률, 각 링크의 대역폭, 각 스위치의 버퍼 사용량, 각 스위치의 홉 지연시간으로 정의한다. 제안하는 알고리즘은 연속된 데이터를 처리하는데 강점을 갖고 있는 순환 신경망을 사용하여 INT 데이터로부터 네트워크 상태를 추론한다. 강화학습의 행동은 네트워크 상태로부터 모든 링크에 대한 가중치를 결정하는 것으로 이는 순환신경망의 출력값에 해당한다. 각 링크의 가중치는 0 과 1 사이의 값이다. 강화학습의 보상으로는 평균 플로우 처리시간에 -1 을 곱한 값을 사용한다. 학습 알고리즘으로 DDPG 를 사용하여 보상을 최대화 하도록 학습시킨다.

DDPG 알고리즘은 Actor 네트워크, Target Actor 네트워크 Critic 네트워크, Target critic 네트워크, 총 네 개의 순환 신경망이 학습에 사용된다. Actor 네트워크는 행동을 결정하는 정책을 계산하는 네트워크이고 Critic 네트워크는 행동을 평가하는 Q-value 를 계산하는 네트워크이다. 나머지 두 Target 네트워크는 각각 Actor 네트워크와 Critic 네트워크를 복사한 네트워크로 학습

결과를 천천히 반영하여 학습에 안전성을 높이기 위해 사용된다. Actor 네트워크는 현재 상태와 Critic 네트워크에서 계산한 Q-value 를 사용하여 정책을 경사하강법(Gradient Descent)으로 학습한다. Critic 네트워크는 (현재 상태, 현재 행동, 보상, 다음 상태)로 이루어진 배열로 정의되는 transition 을 사용하여 손실함수(Loss function)을 최소화하도록 학습한다. 손실함수는 Actor 네트워크와 Critic 네트워크로 계산한 Q-value 와 Target Actor 네트워크와 Target Critic 네트워크로 계산한 Q-value 의 차이로 정의한다. 이후 각 Target 네트워크들의 값은 현재 Target 네트워크의 값과 해당하는 네트워크의 값을 일정 비율로 합친 값으로 업데이트한다.

강화학습 모듈은 INT 데이터가 들어오면 Actor 네트워크가 결정한 현재의 정책에 따라 행동을 결정하고 transition 을 버퍼에 저장한다. 이후 버퍼로부터 N 개의 transition 을 랜덤하게 선택하고 transition 으로부터 Critic 네트워크를 학습한다. 그리고 학습된 Critic 네트워크를 바탕으로 Actor 네트워크를 학습한다.

강화학습으로 결정된 모든 링크에 대한 가중치는 컨트롤러에 전달된다. 이후 플로우가 흐를 경우 각 스위치에서는 패킷 재배열 문제를 해결하기 위해 flowlet[11] 단위로 출력포트를 선택한다. 각 출력포트는 해당 스위치의 모든 출력포트 의 링크 가중치의 합에 대한 현재 출력포트의 링크 가중치의 비율로 선택된다. 이는 모든 경로로 패킷을 보내어 네트워크 전체의 정보가 전달될 수 있도록 하는 역할을 한다.

제안하는 알고리즘은 네트워크 일부분의 정보만을 사용하고 전체 링크의 가중치를 선택한다. 기존 강화학습기반 로드밸런싱 알고리즘에 비해 입력값과 출력값의 차원이 모두 작아 뉴럴 네트워크 크기 또한 작아지게 된다. 이는 강화학습에 의한 시간 오버헤드를 줄여 짧은 플로우에도 대응할 수 있도록 한다. 네트워크 전체 데이터를 사용하지 않고 일부만을 사용함에 따라 예상되는 성능이 저하되는 문제는 연속된 데이터에 맞는 순환 신경망을 사용하여 보완할 수 있다. 또한 주어진 네트워크 환경에서 데이터수집과 학습을 끊임없이 반복한다. 따라서 네트워크 환경이 바뀌어도 지속적인 학습을 통해 새로운 네트워크에 적응하여 기존 로드밸런싱 알고리즘과 비슷한 성능을 보여줄 수 있다.

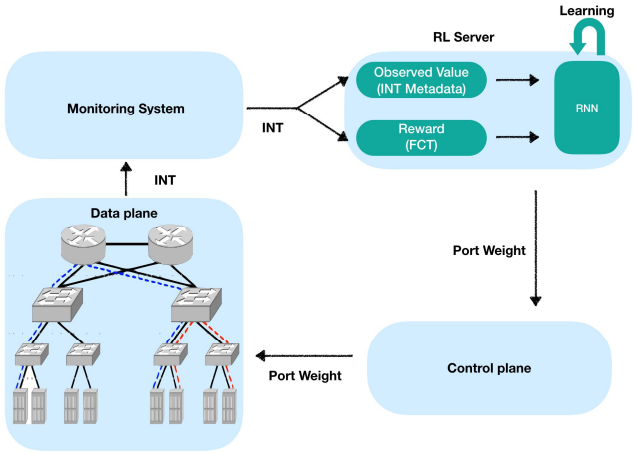

그림 2 강화학습기반 링크가중치 조정 알고리즘

IV. 결론

본 논문에서는 프로그래머블 스위치와 INT 기법을 사용해 네트워크 정보를 수집하고, 이를 활용한 강화학습 기법 알고리즘으로 링크가중치를 결정하는 로드밸런싱 알고리즘을 제안하였다. 제안하는 알고리즘은 네트워크 전체의 데이터가 아닌 연속된 플로우 데이터만을 사용하여 전체 네트워크 상태를 추측하고, action 의 수를 감소시켜 강화학습에 의해 발생하는 오버헤드를 최소화한다. 향후 연구로는 프로그래머블 스위치를 통해 실제 시스템을 구축하여, 데이터센터 네트워크 트래픽을 생성하여 기존 다른 알고리즘과 성능을 비교•분석할 예정이다. 마지막으로, 네트워크 크기에 따른 강화학습 알고리즘의 학습 속도와 패킷 처리시간을 고려하는 로드밸런싱 알고리즘을 개발한다.

ACKNOWLEDGMENT

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발)

참 고 문 헌

[1] Al-Fares et al: A scalable, commodity data center network architecture. In: ACM SIGCOMM Computer Communication Review, vol. 38, pp. 63- 74. ACM, 2008

[2] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, “Vl2: a scalable and flexible data center network,” in ACM SIGCOMM computer communication review, vol. 39, pp. 51- 62, ACM, 2009

[3] Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, pp. 267- 280. ACM, 2010

[3] M. Chiesa, G. Kindler, and M. Schapira, “Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective,”IEEE Conference on Computer Communications, 2014

[4] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic Flow Scheduling for Data Center Networks,” NSDI, 2010

[5] M. Alizadeh et al., “CONGA: distributed congestion-aware load balancing for datacenters,”Proceedings of the 2014 ACM conference on SIGCOMM, 2014.

[6] Lin Q., Gong Z., Wang Q., Li J. “RILNET: A Reinforcement Learning Based Load Balancing Approach for Datacenter Networks,” In: Renault É., Mühlethaler P., Boumerdassi S. (eds) Machine Learning for Networking. MLN 2018. Lecture Notes in Computer Science, vol 11407. Springer, Cham, 2019

[7] Binjie He, Dong Zhang, Chang Zhao, Hidden Markov Model-based Load Balancing in Data Center Networks, The Computer Journal, , bxz142, https://doi.org/10.1093/comjnl/bxz142, 2019

[8] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L. J. Wobker, “In-band Network Telemetry via Programmable Dataplanes,” In ACM SIGCOMM, 2015.

[9] P. Bosshart et al., “P4: programming protocol-independent packet processors,” ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, pp. 87-95, 2014.

[10] Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015

[11] E. Vanini, R. Pan, M. Alizadeh, P. Taheri, and T. Edsall, “Let it flow: Resilient asymmetric load balancing with flowlet switching,” in Proc. USENIX NSDI,Boston,MA, USA, 2017, pp.407-420