

스루풋/응답시간 성능 개선을 위한 SDN 기반 스토리지 스케줄링 방안 제안

김기정, 허준식, 백현기, 이채은, 이상범
SK 브로드밴드

{gjkim7138, herjunsic, kgibsonb, chaeun, sb.lee}@sk.com

A Proposal of SDN based Storage Scheduling Scheme for Improvement of Throughput and Response Time

Gijeong Kim, Junsic Her, Hyunki Baik, Chae-eun Lee, Sangbeom Lee
SK Broadband

요 약

개인화 추천 데이터 생성/적재와 잦은 업데이트는 백엔드 스토리지의 응답시간 지연과 API 응답시간으로 이어져 고객 서비스 경험이 저하되는 현상이 발생하고 있다. 이를 해결하기 위해서 이중화 된 분산 데이터 스토리지를 구성하고 SDN(Software Defined Networking)을 활용하여 백엔드 데이터 스토리지를 스케줄링 하는 아키텍처 및 방안을 제안한다.

I. 서 론

과거에는 가구 단위로 단일 셋톱박스를 통해서 IPTV 서비스가 이용된 반면, 미디어를 소비하는 단말이 셋톱박스, 스마트폰, 태블릿 등으로 다양해지고, 고객의 선호 콘텐츠의 성향이 확고해지고 있으며, 콘텐츠의 다양성과 규모가 폭발적으로 증가하고 있다. 1) 이중의 미디어 소비 단말의 증가, 2) 콘텐츠 선호도 편향, 3) 콘텐츠 볼륨/복잡도 증가로 인해 고객의 스마트한 콘텐츠 소비 욕구가 커지고 있으며, 빅데이터와 AI 기반 개인화 추천 서비스를 통해서 최적화 된 미디어 서비스를 제공하고 있는 추세이다.

개인화 추천 서비스를 제공하는 과정에서 AI 알고리즘 기반 대용량 데이터 처리가 빈번하게 발생함에 따라 CPU 나 메모리 보다는 대체로 Hadoop, Elasticsearch 와 같은 대용량 분산 스토리지 입출력 과정에서 병목현상이 발생하고 있다. 이는 API 서버가 대용량 분산 스토리지 시스템에서 개인화 추천 결과를 조회하여 클라이언트에게 응답하는 구조에서 대규모 배치 또는 스트림 프로세싱이 발생하면 API 응답시간이 지연되거나 트래픽 최번시에 스토리지의 데이터 처리율이 낮아지면서 서비스 사용자 경험을 저하시킬 수 있다.

본 논문에서는 이중화 된 대용량 분산 스토리지를 사용하는 구조에서 SDN(Software Defined Networking)을 기반으로 스토리지 클러스터 간 스케줄링을 제공하여 처리율과 응답시간을 개선하는 방안을 제안한다.

II. 관련 연구

대용량 분산 스토리지는 Hadoop 이 가장 대표적이고, YARN, MESOS 등으로 Resource Management, Job Scheduling 을 수행하며, 멀티 노드로 구성된 대용량

분산 클러스터 내에서 처리율을 개선하기 위해서 작업 소요시간 예측 기법을 이용해 스케줄링을 하거나 I/O 성능을 극대화 하는 스케줄링 기법들이 연구되고 있다.[1][2][3][4][5] 이는 단일 클러스터 내에서 최적화 기법에 대한 연구로 제한되어 있으며, 실제 API 를 통한 서빙 품질을 보장하기 위해서 구축되는 멀티 클러스터 또는 이중 클러스터 환경에서의 스케줄링 기법에 대한 연구가 필요하다.

일반적으로 그림 1 과 같이 람다 아키텍처를 기반으로 하는 레퍼런스 모델 형태로 구현되는 경우가 많으며, 대용량 데이터 동기화나 I/O 입출력이 많은 경우, API 응답시간이 지연되는 현상이 발생한다. 대체로 Hadoop 을 많이 이용하며, Oracle 과 같은 RDMS 를 백엔드 스토리지를 이용할 경우, Master/Slave 구조로 구성하고 Shareplex 나 Golden Gateway 같은 CDC(Change Data Capture) 솔루션을 도입하여 실시간 동기화를 적용하기도 하지만 빅데이터 기반 개인화 추천 데이터는 부분적 변경 데이터가 발생하기 보다는 전체적으로 데이터가 변경되기 때문에 CDC 솔루션의 적용이 효율적인 동작하지 않을 수 있다.

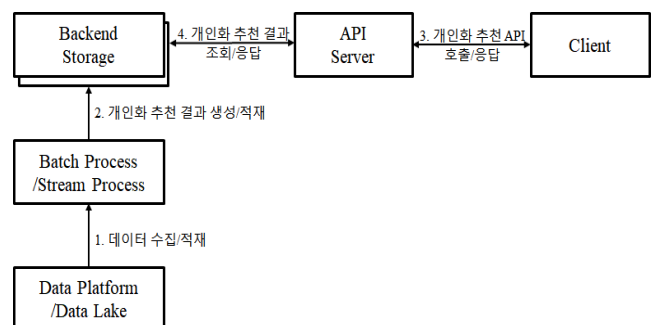


그림 1. 람다 아키텍처 기반 레퍼런스 모델 예시

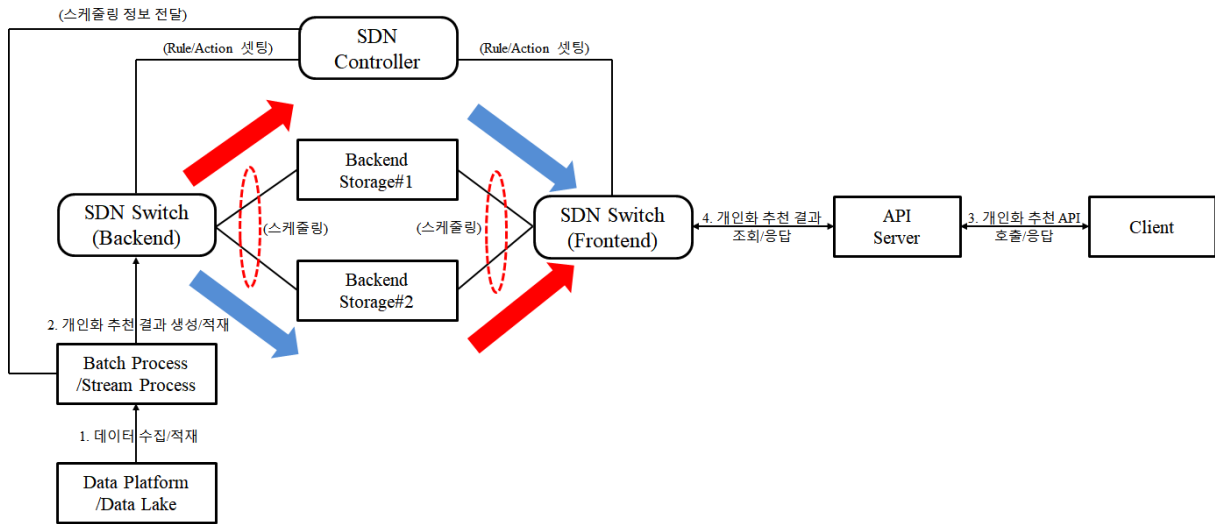


그림 2. SDN 기반 스토리지 스케줄링 아키텍처

III. 제안 방안

AI 알고리즘을 기반으로 대용량 데이터 처리 및 적재가 발생할 때 스토리지 처리율과 API 응답시간이 지연되는 문제를 해결하기 위해서, 대용량 분산 멀티스토리지 또는 이중화 된 대용량 분산 스토리지로 백엔드 스토리지를 구성하고 SDN 스위치와 컨트롤러를 이용해서 배치 프로세스와 API 서버가 백엔드 스토리지에 접근을 스케줄링 하는 방안을 제안한다. 일반적으로 백엔드 스토리지는 Hadoop, Elasticsearch, Cassandra 와 같은 분산 스토리지 클러스터로 구성된다.

최초에는 배치 프로세스가 이중화 된 백엔드 스토리지 2 곳 모두에 개인화 추천 데이터를 적재한다. 이후 배치 프로세스가 개인화 추천 데이터를 업데이트 하기 위해서 대용량 데이터 처리를 수행하고 개인화 추천 데이터를 생성하면 백엔드 스토리지에 적재하기 전에 SDN 컨트롤러에 백엔드 SDN 스위치에는 배치 프로세스의 트래픽은 백엔드 스토리지 #1 으로 트래픽이 포워딩 되도록 설정하고, 프론트엔드 SDN 스위치에는 API 서버의 트래픽을 백엔드 스토리지 #2 으로 포워딩 하도록 Rule/Action 명령어를 보낸다. 이후 배치 프로세스에서 개인화 추천 데이터에서 적재를 시도하면 백엔드 스토리지 #1 으로 데이터가 적재되고, API 서버는 백엔드 스토리지 #2 으로 개인화 추천 데이터를 조회하게 된다. 이 과정에서 대용량의 스토리지 I/O 가 발생하더라도 API 서버는 백엔드 스토리지로부터의 신속한 응답시간을 보장 받을 수 있는 환경을 제공 받게 된다. 마지막으로 개인화 추천 데이터 적재가 완료되면 백엔드 SDN 스위치에는 배치 프로세스의 트래픽이 백엔드 스토리지 #2 로 포워딩 되도록 설정하고, 프론트엔드 SDN 스위치에는 API 서버의 트래픽이 백엔드 스토리지 #1 로 포워딩 되도록 SDN 컨트롤러에 Rule/Action 명령어를 보낸다.

이와 같은 SDN 기반 스토리지 스케줄링 방식을 통해서 개인화 추천 데이터 업데이트를 위해서 배치 프로세스에 의해서 추가 대용량 데이터 처리가 발생하더라도 안정적이고 짧은 API 응답시간을 보장할 수 있고, 배치 프로세스에 의해서 업데이트 된 개인화 추천 데이터에 대한 최신성도 보장하게 된다.

IV. 결론 및 향후 연구 방향

개인화 서비스는 고객의 서비스 만족도를 높이고 기업의 매출을 극대화 하는데 핵심 기술로 자리 잡고 있다. 이와 같은 추세에서 대용량 데이터 처리와 개인화 추천 데이터의 잦은 변경으로 서비스 응답시간의 지연으로 고객 서비스 경험이 저하되는 케이스가 발생한다. 이를 해결하기 위해서 이중화 된 스토리지 구조 구성하고 SDN 을 활용하여 스토리지 I/O 병목 현상을 해소시키는 스토리지 스케줄링 아키텍처 및 방안을 제안하였고, SDN 기술이 네트워크의 유연성과 확장성을 개선하기 위한 기술로 제안 및 개발되었지만 어플리케이션이 네트워크를 제어함으로써 IP 주소의 변경 없이 컴퓨팅 및 스토리지 리소스를 효율적으로 사용하는 사례로 볼 수도 있을 것 같다. 향후에는 실제 구현을 통해서 기술 타당성 평가를 진행할 예정이다.

참 고 문 헌

- [1] Safaa Alkatheri et al. "A Comparative Study of Big Data Frameworks," International Journal of Computer Science and Information Security (IJCSIS), Vol. 17, No. 1, pp. 66-73, January 2019.
- [2] Ji Liu et al. "A Survey of Scheduling Frameworks in Big Data Systems," International Journal of Cloud Computing, Vol. 7, No. 2, pp. 1-27, January 2018.
- [3] AlbertReuther et al. "Scalable system scheduling for HPC and big data," Journal of Parallel and Distributed Computing, Vol. 111, pp. 76-92, January 2018.
- [4] Hatem Elshazly et al., "Towards enabling I/O awareness in task-based programming models," Future Generation Computer Systems, Vol. 121, pp. 74-89, August 2021.
- [5] N. Marz, J. Warren, Big Data: Principles and best practices of scalable realtime data systems, Manning Publications, 2013.