

안녕하세요. 네트워크 트래픽 로드 밸런싱 시스템 설계를 주제로 발표할 최시몬 정찬우 조의 000입니다. 발표시작하겠습니다.

p1 목차

목차는 선정 동기, 알고리즘, 성능비교기준, 결론 및 향후 계획입니다.

p2. 선정동기

이미지에서 보시다시피 한국의 망사용료는 타국가들에 비해 상당히 높은 편입니다. 특히 넷플릭스, 유튜브 등 대규모 트래픽을 발생시키는 글로벌 기업들이 망 사용료 문제와 관련해 국내 통신사들과 충돌하는 상황에서, 이러한 트래픽을 어떻게 효과적으로 처리하고 관리할 것인가가 주요 이슈로 부각되고 있습니다.

따라서 대규모 데이터를 처리하는 네트워크 환경에서 성능 저하나 과부하를 막기 위한 로드 밸런싱 기술이 필수적인 상황입니다.

이에 대규모 트래픽 관리에 대한 관심이 생겨 이 주제를 선정하게 되었습니다.

p3. 데이터 (트래픽 처리 기준 선정)

사용할 데이터셋은 캐글에서 받아왔습니다. 이 데이터는 6일 동안 특정 시간대에 수집된 트래픽 데이터입니다. 데이터가 매우 방대하여 약 100만 개의 트래픽 데이터로 축소하여 분석을 진행할 예정입니다. 이를 통해 네트워크의 다양한 흐름을 효율적으로 처리할 방법을 찾기 위해, 지속시간, 전체 송수신 패킷 수, 패킷 간 평균 간격, 초당 전송되는 바이트 수, 총 네 가지 기준을 고려해 우선 처리 기준을 설정했습니다.

Flow Duration (지속 시간)

지속 시간이 짧은 트래픽을 우선 처리하면, 네트워크에서 대기 시간을 줄일 수 있습니다. 또한, 장시간 네트워크에 쌓이는 트래픽을 방지하여 전반적인 네트워크 흐름을 원활하게 유지하는 데 도움이 됩니다.

Total Packets (전체 송수신 패킷 수)

송수신된 패킷 수를 기준으로 정렬하면, 많은 패킷을 처리해야 하는 대규모 트래픽 흐름을 우선 처리할 수 있습니다. 이를 통해 많은 트래픽을 생성하는 흐름이 원활하게 처리될 수 있습니다.

Flow IAT Mean (패킷 간 평균 간격)

패킷 간 평균 간격이 짧은 트래픽은 더 높은 우선순위를 부여할 수 있습니다. 이 기준은 트래픽 밀도가 높은 경우 빠르게 처리할 수 있는 장점이 있습니다.

Flow Bytes per Second (초당 전송되는 바이트 수)

Flow Bytes per Second는 초당 전송되는 바이트 수를 기준으로 합니다. 이 기준은 비디오 스트리밍, 대용량 파일 전송, 실시간 데이터 처리 등 네트워크 자원을 많이 차지하는 트래픽 흐름을 효율적으로 관리할 수 있습니다. 네트워크 자원이 집중되는 흐름을 우선 처리함으로써 대역폭을 많이 차지하는 트래픽을 효율적으로 관리하고 네트워크 병목 현상을 방지할 수 있습니다.

이러한 이유로 **Flow Bytes per Second(초당 전송되는 바이트 수)**가 네트워크 트래픽 관리에서 가장 적합한 기준이라고 판단하여 최종적으로 선택했습니다.

p4. 알고리즘 설명

퀵 정렬

정렬은 피벗을 선택하여 데이터를 두 개의 부분으로 나누고, 각 부분을 재귀적으로 정렬하는 방식입니다. 평균적인 성능은 매우 우수하지만, 피벗 선택에 따라 성능이 달라질 수 있습니다.

시간복잡도는 최선의 경우 $O(n\log(n))$ 이고 최악의 경우 $O(n^2)$ 입니다.

장점으로는 $O(n\log(n))$ 인 다른 정렬 알고리즘과 비교해도 속도가 빠르다는 점이 있고 단점으로는 정렬된 데이터들에 대해서 최악의 시간 복잡도가 나오게 된다는 점입니다.

트래픽 분석에서 중간 크기의 데이터셋을 빠르게 정렬하는 데 적합하고, 일반적으로 트래픽 데이터를 빠르게 정렬할 수 있어, 트래픽 처리량을 기준으로 데이터 흐름을 정렬할 때 적합하기 때문에 사용할 계획입니다.

병합정렬

병합 정렬은 리스트를 반으로 나누어 각각을 정렬한 후, 병합하는 방식입니다. 안정적인 정렬 알고리즘으로 항상 일정한 시간 복잡도를 보장하며, 대규모 데이터에서도 일관된 성능을 제공합니다.

시간복잡도는 최선의 경우와 최악의 경우 모두 $O(n\log(n))$ 입니다.

병합 정렬은 메모리 사용량은 다소 높지만, 100만 개 이상의 데이터에서도 안정적인 성능을 보여주며, 트래픽 처리 성능 비교에 적합합니다.

힙 정렬

힙 정렬은 비교 기반 정렬 알고리즘입니다.

선택 정렬 알고리즘과 비슷하게 정렬된 영역과 아닌 영역을 나누어 가장 큰 요소를 찾아 정렬된 영역으로 배치시킵니다.

선택 정렬보다 개선된 점은 선형 시간이 소요되는 선택 정렬의 탐색과 달리 힙 자료구조를 사

용하여 최댓값을 탐색합니다.

시간복잡도는 최선의 경우와 최악의 경우 모두 $O(n\log(n))$ 으로 같습니다.

장점으로는 최악의 경우에도 $O(n\log(n))$ 의 빠른 속도가 보장되고 추가적인 메모리를 필요로 하지 않습니다.

단점으로는 정렬간의 데이터 안정성을 보장받지 못한다. 즉, 중복의 데이터의 경우 순서가 바뀌는 불안정한 알고리즘이며 데이터의 상태에 따라 다른 정렬법보다 느릴 수 있습니다.

힙 정렬은 메모리 사용량을 고려하고, 대규모 데이터를 정렬하기 위한 시간 복잡도까지 고려했을 때 가장 적합한 정렬 알고리즘이라 생각되어서 사용할 계획입니다.

그리디 알고리즘

트래픽 흐름에서 특정한 특성을 기준으로 정렬하기 때문에 어떻게 나올지 궁금해서 넣어봤습니다.

p5. 시간 / 공간 복잡도

p6.

알고리즘 성능 비교 기준은

응답 시간

- 각 정렬 알고리즘이 데이터를 정렬할 때 걸리는 시간을 측정합니다

메모리 사용량

- 각 알고리즘이 정렬 작업을 처리하는 동안 사용한 메모리 양을 기록합니다.

특히 힙 정렬과 합병 정렬의 메모리 사용량을 중점적으로 관찰할 예정입니다.

힙 정렬은 추가 메모리가 거의 필요 없으나, 합병 정렬은 $O(n)$ 만큼의 추가 메모리 공간이 필요하기 때문입니다.

- 최적화 가능 여부

퀵 정렬은 피벗의 선택에 따른 성능차가 큼으로 최적화 가능한 부분을 관측할 수 있을 것입니다.

- 비교 및 시각화

가능시 그래프 등으로 시각화하여 명확하게 볼 수 있도록 할 예정입니다.

p7. 조언

1. 추가적인 성능 지표 : 응답시간, 메모리 사용량에 초점을 두고 성능 지표를 도입했지만 cpu 활용률, 대기 시간, 네트워크 지연 시간 등 다른 중요한 성능 지표들도 추가할 수 있겠다.

2. 다른 유형의 트래픽 분석 : 알고리즘이 다른 유형의 트래픽에 어떻게 반응하는지 비교 분석을 통해 다양한 환경에서 성능을 테스트

3. 다양한 환경에서 테스트 : 2와 마찬가지로 다양한 실제 세계 네트워크 환경에서의 최적화 방안 모색

4. 보안 : 트래픽 로드 밸런싱 시스템에서 가장 중요한 것은 보안이라고 생각한다.

특히 클라우드 서비스가 점점 더 큰 서버 시장 파이를 차지하고 있는데 이런 멀티 클라우드와 같은 환경에서의 악성 트래픽 탐지 등의 솔루션 적용 방안 모색

p8. 결론

초당 전송되는 바이트를 기준으로 대규모 트래픽 흐름을 우선 처리함으로써 네트워크 성능을 최적화하는 것입니다.

-시간복잡도, 공간복잡도, 규칙성등을 통한 성능 비교합니다

p9. 일정

11월 22일까지 알고리즘 코드작성 및 결론도출,

11월 29일까지 프로젝트 보고서 작성 예정입니다.

p10. 끝

오늘 발표를 통해 네트워크 트래픽 로드 밸런싱과 다양한 알고리즘 성능들을 비교하여 어떤 방식으로 최적화 해보고 적용할지 살펴보았습니다.

이상으로 발표를 마치겠습니다. 감사합니다.