

대규모 브로드캐스팅의 처리

하나의 예로 3만 연결을 갖는 채팅 서버에서 전체 사용자에게 메시지를 전송한다. 전송의 지연 시간과 개별 전송의 부하가 영향을 미친다. 개별 전송의 부하가 크다면 전송 지연이 커질 수 있다. 한번에 전송한다면 급격하게 부하가 올라가 부하 자체가 다시 지연을 증가시키는 악순환이 발생할 수 있다.

선택한 전송 방식을 S_1, S_1 하의 전송 지연을 d_1 이라고 한다. S_k 하의 전송 지연은 d_k 가 된다. 메시지의 전송 요청은 큐에 넣는다고 가정한다. 그러면 메시지 수신 자체의 부하는 크지 않다. 단지, 큐에 넣는 것으로 완료된다.

전송 지연 d_k 가 동시에 전송 되는 개수에 따라 증가할 가능성이 있고 실제 그럴 것으로 보인다. 따라서 d_k 는 동시 전송의 개수의 함수가 되므로 $d_k(n) : M \rightarrow R$ 인 함수이다. M 은 메시지, R 은 실수이다.

방향

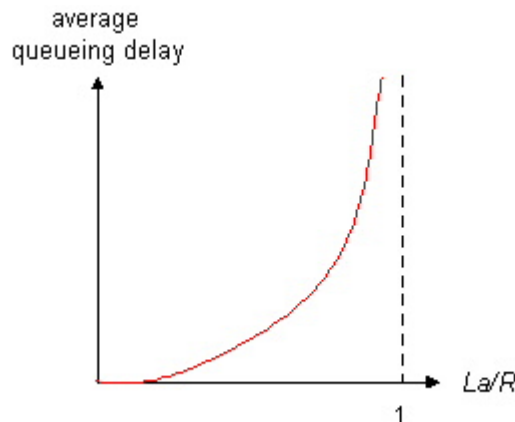
- 동시 전송 개수 n 을 조절하여 지연을 조절한다.
- 수신 개수를 조절하여 큐 길이가 무한정 늘어나지 않도록 한다.
 - 일정 개수 이상의 수신은 드랍시킨다.

알고리즘

상태를 갖고 있어야 할 대상은 세션이다. 각 브로드캐스팅 메시지는 고유한 순서를 갖는다. 브로드캐스팅 메시지를 $m_{seq} \in M$ 으로 표시한다. 실제로는 $m_1, m_2, \dots, m_i, \dots$ 이 된다. i 번째 메시지를 전송할 때 대상이 되는 세션을 조절하는 것이 핵심이 된다.

큐에서 peek만 하고 꺼내지 않는 기능이 필요하다. 한번에 모든 연결에 보내는 것이 아니라 나눠서 보내기 때문이다. 전송한 세션 그룹을 메시지에 기억해 두어야 한다.

대상이 되는 세션들을 나누어야 한다. 미리 연결을 맺을 때 적절한 개수의 그룹으로 분할 하는 것이 필요하다. 수 천 개의 연결에 대한 브로드캐스팅은 대부분의 경우 문제가 되지 않는다. 수 만개의 경우 PC 급 서버에서 급격한 부하 증가를 불러올 수 있다. 병목이 발생하면 지연은 기하 급수적으로 올라간다.



적절한 부하 증가 지점을 찾아서 그룹만 분할하고 입력 개수를 제한하면 오히려 더 많은 것을 처리할 수 있다. 왜냐하면 지연을 증가시키지 않는 부하로 나뉘었을 때 초당 처리 개수가 올라갈 가능성이 매우 높다.

파라미터 튜닝

- 세션 그룹의 크기
- 입력 개수의 제한

두 가지를 변경하면서 테스트를 진행해 보고 대상 시스템 장비와 환경에서 적절한 구성을 찾는다.