가상 면접 사례로 배우는

대규모 시스템 설계 기초

이정민

5. 안정 해시 설계

수평적 규모 확장에서 요청/데이터를 서버에 균등하게 나누는 기법 설계

해시 키 재배치(refresh) 문제

■ 해시 함수

- **서버들에 부하를 균등하게 나누는 보편적 방법**
  - serverIndex = hash(key) % N
- **장점**
  - 서버 풀 크기가 고정일 때, 데이터 분포가 균등할 때
- **단점**
  - 서버가 추가 되거나 기존 서버 삭제 시 문제 발생
  - 서버 인덱스 값 변겨으로 대부분의 키가 재분배됨
  - 대규모 캐시 미스 발생

안정 해시

- 안정 해시 (consistent hash)
  - **서버 추가 제거 시 최소한의 재분배 발생 알고리즘**
    - 평균적으로 k/n 개의 키만 재배치 (k: 키의 수, n: 슬롯 수)
  - **해시 공간**
    - 가능한 모든 해시값의 집합을 의미
    - SHA-1 해시값의 크기는 160비트 (범위: 0~2의 160승 -1)
  - **해시 링**
    - 해시 공간을 원형으로 구성 (최소값과 최대값이 연결)

- 동작 원리
  - **해시 서버/키**
    - 해시링에 서버와 키를 배치
  - **서버 조회**
    - 해당 키 해시값에서 시계 방향으로 첫번째 탐색되는 서버
  - **서버 추가**
    - 새로 추가 된 서버에서 시계 방향으로 첫번째 탐색되는 서버 구간에 속하는 키가 추가된 서버로 재배치
  - **서버 제거**
    - 해당 노드가 담당하는 키는 다음 시계 방향의 서버로 재배치
  - **문제점**
    - 파티션(서버 사이의 공간) 크기 균등 유지 불가능
    - 키의 균등 분포를 달성하기 힘듬

안정 해시

■ 가상 노드 기법

- **하나의 서버는 여러개의 가상노드를 가질 수 있음**
  - 각 서버는 여러개의 파티션을 관리
  - 가상 노드에 할당된 데이터는 실제 서버에 저장
- **장점**
  - 가상 노드 증가 시 키의 분포는 점점 더 균등해짐

마치며

■ 안정 해시의 이점

- 서버가 추가되거나 삭제 될 때 재배치되는 키의 수가 최소화됨
- 데이터가 보다 균등하게 분포하게 되므로 수평적 규모 확장성을 달성하기 쉬움
- 핫스팟(hotspot) 문제를 줄임 (특정 샤드 접근이 지나치게 빈번하면 서버 과부화 문제 발생)

■ 안정 해시의 실제 사례

- 아마존 다이나모 DB 파티셔닝 관련 컴포넌트, 아파치 카산드라 클러스터 데이터 파티셔닝
- 디스코드 채팅 어플리케이션, 아카미아 CDN, 매그레프 네트워크 부하 분산기