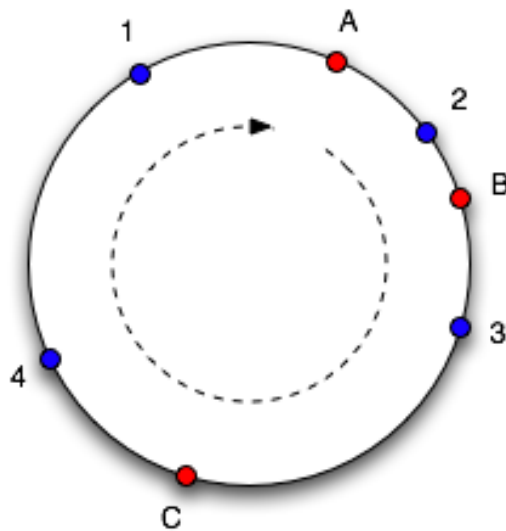


5장 안정 해시 설계

해시 키 재배치 (rehash) 문제

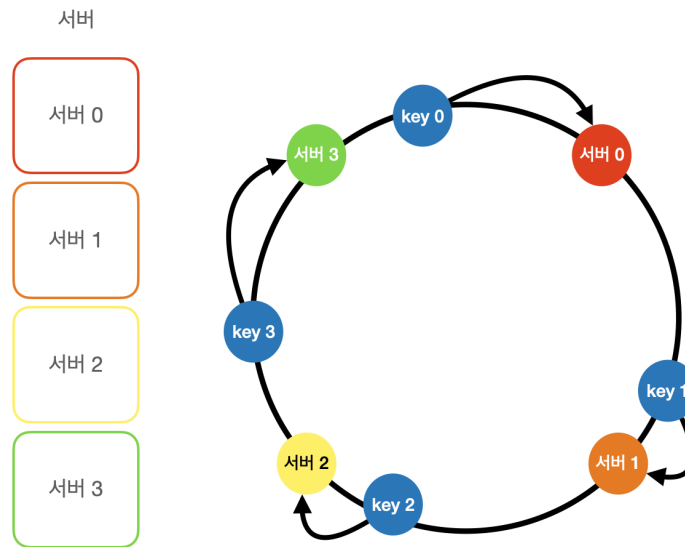
- 보편적으로 쓰이는 해시 함수
 - $\text{serverIndex} = \text{hash}(\text{key}) \% N(\text{서버의 개수})$
 - 서버 풀의 크기가 고정되어 있을 때, 데이터 분포가 균등할 때 잘 작동.
 - 서버 추가 및 삭제시 대규모 캐시 미스 (cache miss) 가 발생.

안정 해시

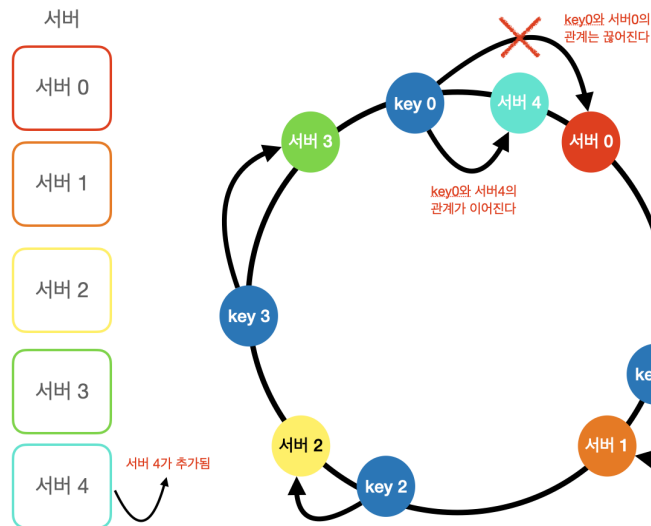


- 해시 테이블의 크기가 조정될 때 평균적으로 k/n (k : 키 개수, n : 슬롯 개수) 개의 키만 재배치하는 해시 기술.
- 함수 SHA-1의 해시 공간 범위는 0부터 $2^{160}-1$ 까지라고 알려져 있으면 출력값 범위인 $x_0 - x_n$ 은 0 부터 $2^{160} - 1$ 이다.
- 해시 공간을 구부리면 해시 링 (hash ring) 이 만들어진다
- **해시 서버** : 서버 IP나 이름을 링 위의 어떤 위치에 대응시킬 수 있다.
- **해시 키** : 해시 링 어떤 위치에 대응시킬 수 있다.

- **서버 조회** : 어떤 키가 저장되면 서버는 해당 키의 위치로부터 시계 방향으로 링을 탐색하면서 만나는 첫번째 서버에 키를 저장한다.



- **서버 추가** : 키는 시계 방향으로 순회하면서 가장 먼저 만나는 서버에 저장되므로 서버가 추가되어도 그 직전의 키만 재배포되고 다른 키들은 그렇지 않다.



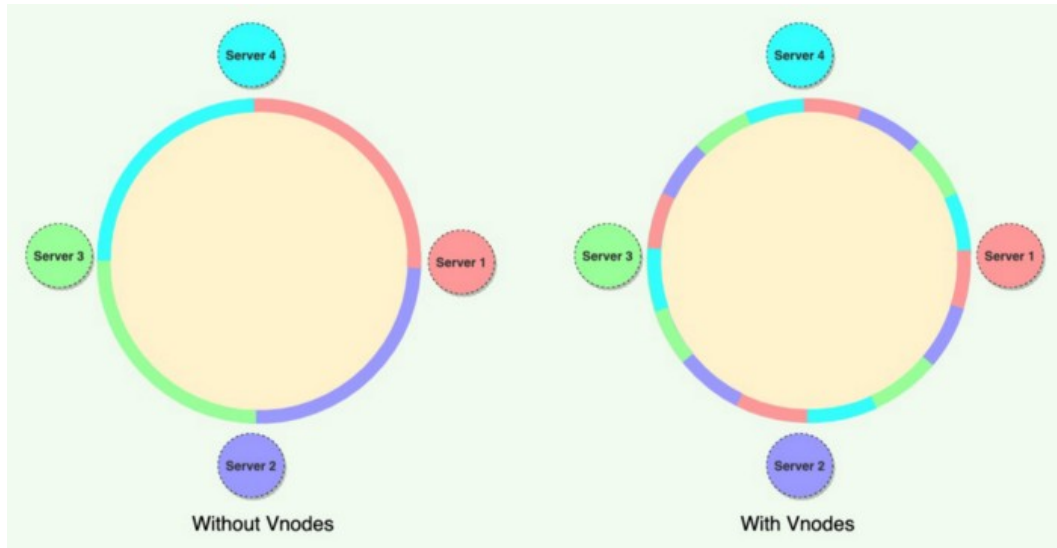
- **서버 제거** : 하나의 서버가 제거되어도 키 가운데 일부만 재배포된다.

기본 구현법의 두 가지 문제

- 서버가 추가, 삭제 상황 생각하면 파티션 (partition) 의 크기를 균등하게 유지 불가하다.
 - 서버 하나가 삭제되면 그 다음으로 인접한 서버의 파티션이 매우 커짐.

- 키의 균등 분포 (uniform distribution) 을 달성하기 어렵다는 것.
 - 서버의 배치에 따라 아무 데이터를 가지지 않을 수도 있음.

가상 노드



- 가상 노드 (virtual node) 는 실제 노드 또는 서비스를 가리키는 노드이다.
- 하나의 서버는 링 위에서 여러 개의 가상 노드를 가질 수 있음.
- 시계 방향으로 순회하면서 만나는 최초의 가상 노드가 나타내는 서버에 저장됨.
- 가상 노드 개수 늘리면 키의 분포가 점점 균등해짐. 표준편차가 작아지므로.
 - 100 ~ 200 개의 가상 노드 사용할 경우 평균 표준 편차 값은 5% (200개인 경우) 에서 10% (100 개인 경우) 가 된다.
- 가상 노드 데이터를 저장할 공간은 더 많이 필요할 것이므로 타협적 결정이 필요하다.
- 키 재배치
 - 서버가 추가된 경우 반시계 방향의 직전 서버와 추가된 서버 사이의 키들만 재배치한다.
 - 서버 s0가 삭제된 경우 그 반시계 방향에 있던 s1과 시계 방향 다음에 있던 서버 s2 사이의 키들이 s2로 재배치된다.

안정 해시의 이점

- 서버 추가 혹은 삭제시 재배치되는 키 개수 최소화.

- 데이터가 보다 균등하게 분포되므로 수평적 규모 확장성 달성 가능.
- 핫 스팟 (hotspot) 키 문제 줄임.