

5

230927 5. 안정 해시 설계

5. 안정 해시 설계

▼ 해시 키 재배포치(refresh) 문제

서버들에 부하를 균등하게 나누는 보편적 방법

서버 풀의 크기가 고정되어 있고 데이터 분포가 균등할 때는 잘 작동함

서버가 추가되거나 기존 서버가 삭제되면 문제가 생김

```
-- N개의 캐시 서버  
serverIndex = hash(key) % N
```

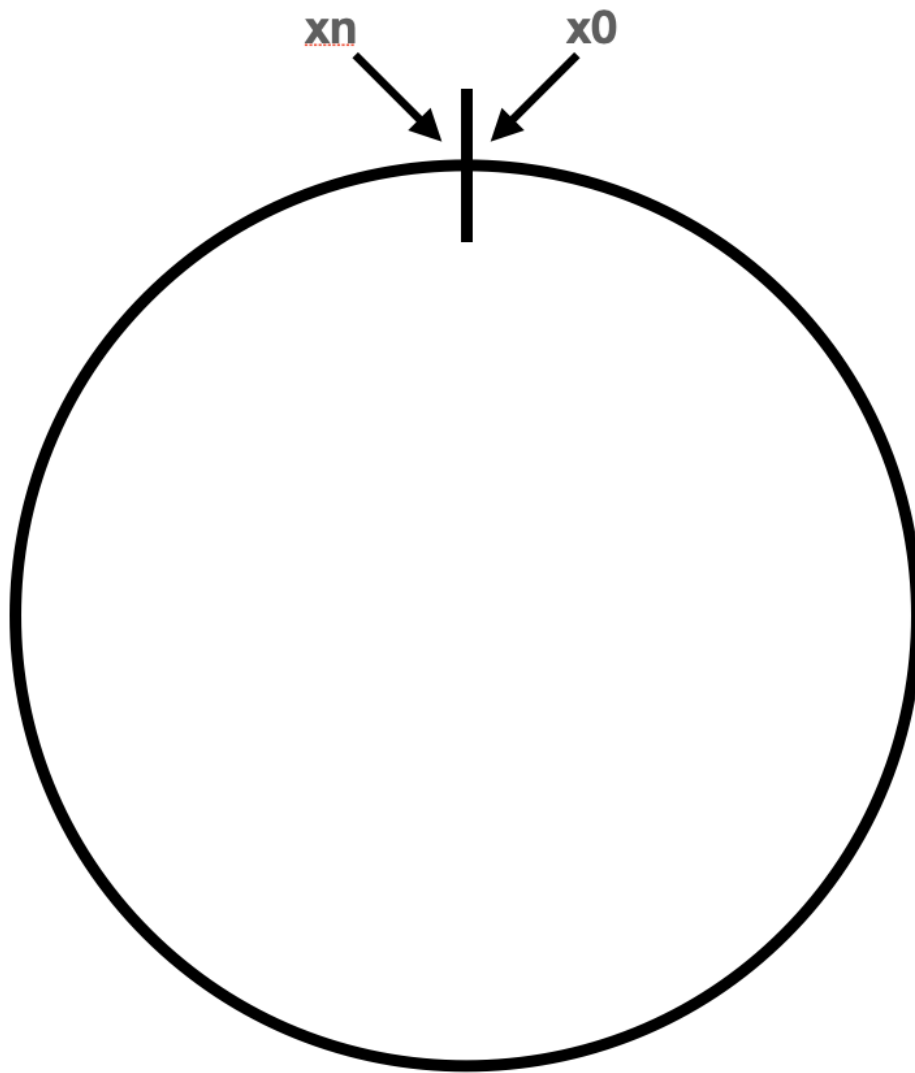
▼ 안정 해시

해시 테이블 크기가 조정될 때 평균적으로 오직 k/n 개의 키만 재배포하는 해시 기술

▼ 해시 공간과 해시 링

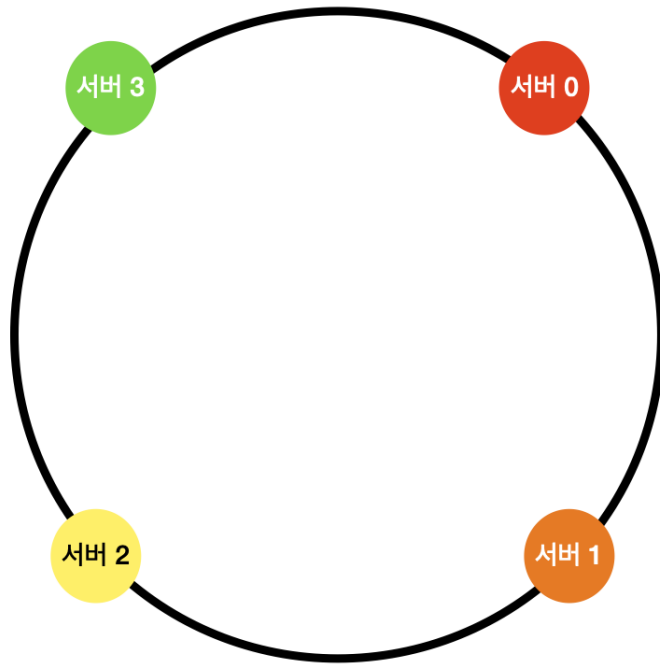
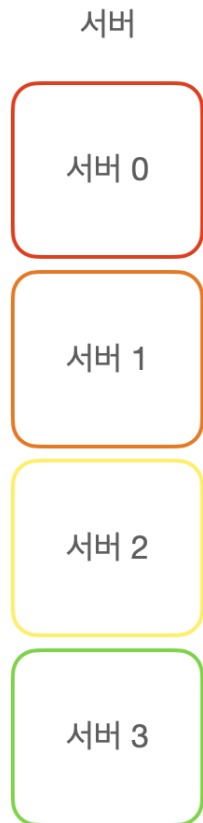
해시 공간의 양쪽을 구부려 접으면 해시 링





▼ 해시 서버

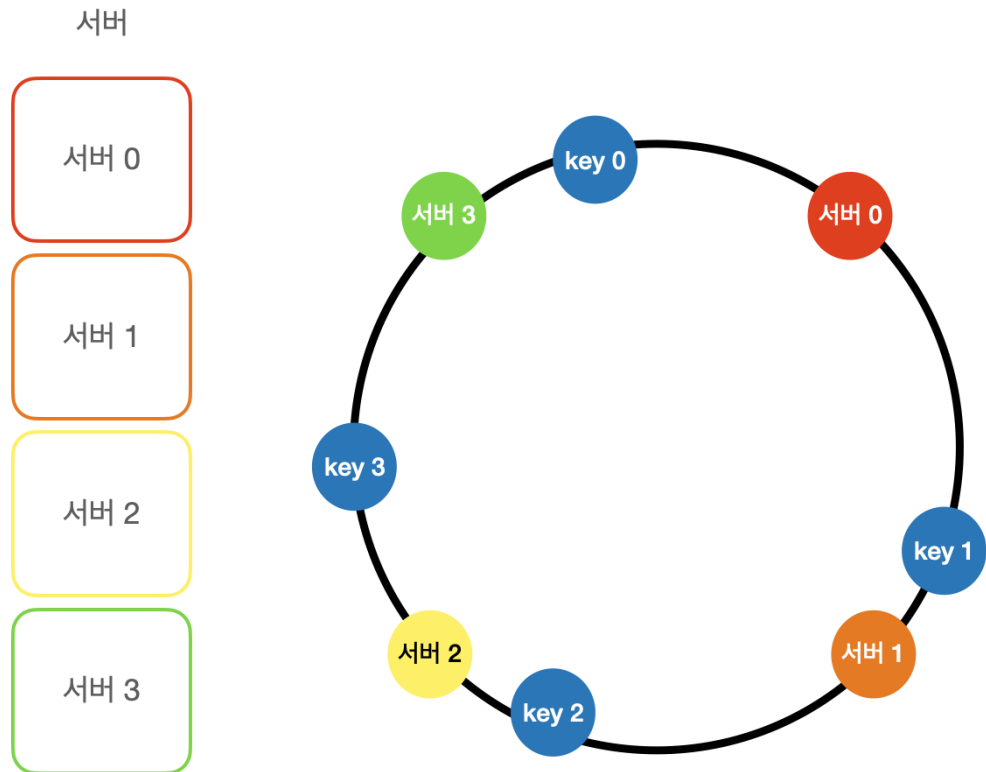
해시 함수 f 를 사용하면 서버 IP나 이름을 이 링 위의 어떤 위치에 대응시킬 수 있음



▼ 해시 키

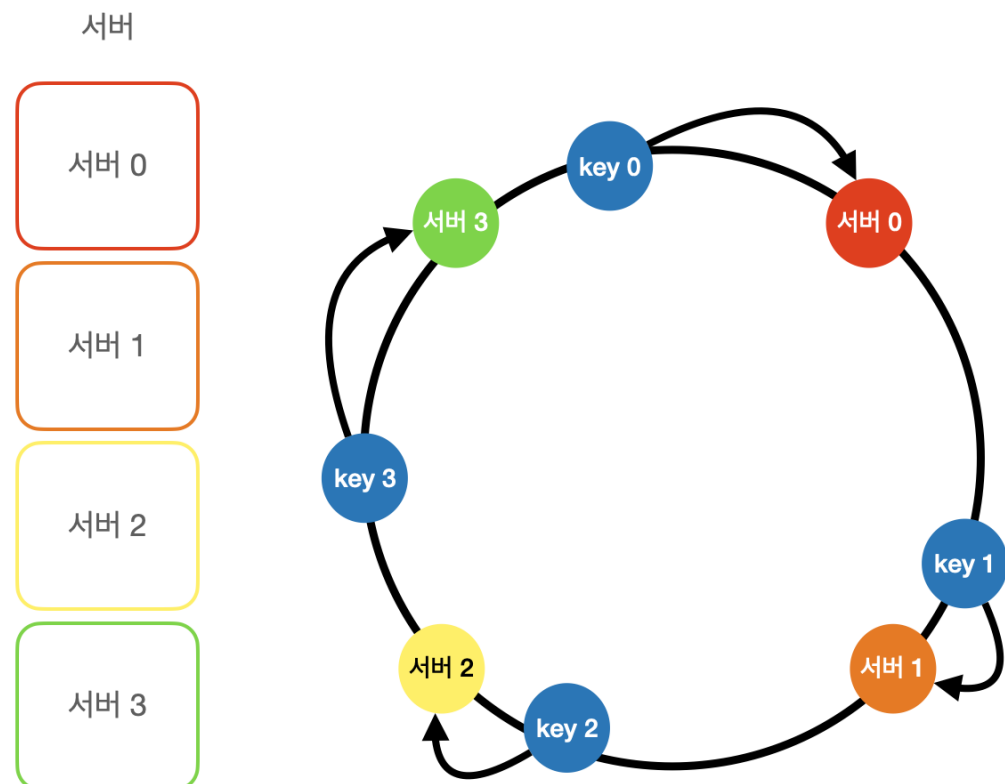
캐시할 키를 해시 링 위의 어느 지점에 배치

“해시 키 재배치 문제”에 언급된 함수와 다르며, 나머지 연산을 사용하고 있지 않음



▼ 서버 조회

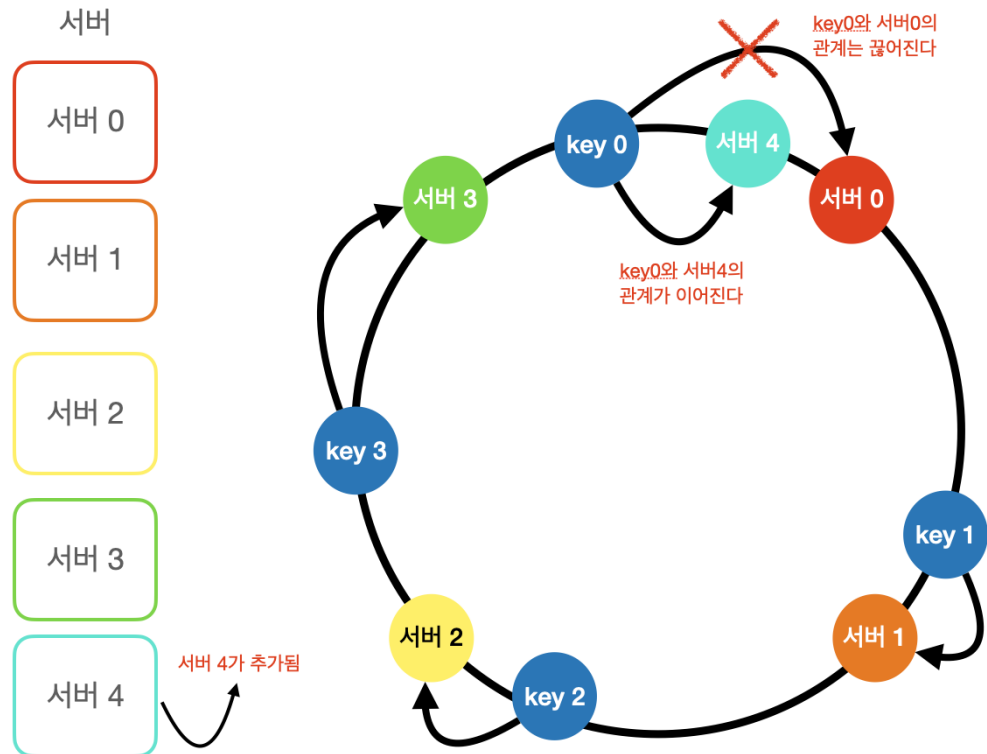
어떤 키가 저장되는 서버는 해당 키의 위치로부터 시계 방향으로 링을 탐색해 나가면서 만나는 첫 번째 서버



▼ 서버 추가

키 가운데 일부만 재배포하면 됨

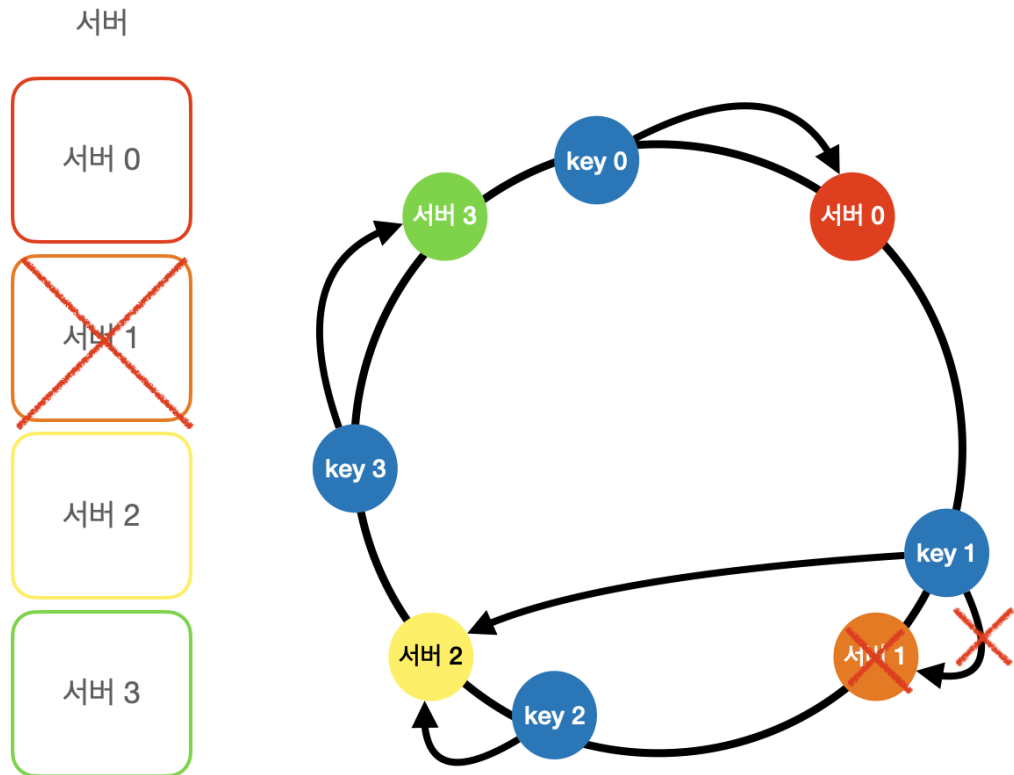
새로 추가된 서버를 기준으로 시계 반대 방향으로 키를 탐색해 다른 서버가 나오기 전 까지 만난 키 재배포



▼ 서버 제거

서버 추가와 비슷하게 키 가운데 일부만 재배포하면 됨

삭제된 서버에 배치되었던 키 재배포



▼ 기본 구현법의 두 가지 문제

안정 해시의 기본 절차

1. 서버와 키를 균등 분포 해시 함수를 사용해 해시 링에 배치
2. 키의 위치에서 링을 시계 방향으로 탐색하다 만나는 최초의 서버가 키가 저장될 서버

기본 절차의 두 가지 문제

- 서버가 추가되거나 삭제되는 상황을 감안하면 파티션의 크기를 균등하게 유지하는 게 불가능
 - 파티션은 인접한 서버 사이의 해시 공간
 - 서버마다 할당받는 공간의 크기가 다를 수 있음
- 키의 균등 분포를 달성하기 어려움

⇒ 가상 노드/복제 기법을 통해 해결 가능

▼ 가상 노드

실제 노드 또는 서버를 가르키는 노드

하나의 서버는 링 위에 여러 개의 가상 노드를 가질 수 있음

가상 노드의 개수를 늘리면 표준 편차가 작아져서 데이터가 고르게 분포되기 때문에 키의 분포는 점점 더 균등해짐

가상 노드의 수를 더 늘리면 표준 편차의 값은 더 떨어지지만 가상 노드 데이터를 저장할 공간은 더 많이 필요하게 됨

⇒ 시스템 요구사항에 맞도록 가상 노드 개수를 적절히 조정해야 함

안정 해시의 이점

- 서버가 추가되거나 삭제될 때 재배포되는 키의 수 최소화
- 데이터가 보다 균등하게 분포하게 되므로 수평적 규모 확장성을 달성하기 쉬움
- 핫스팟(hotspot)키 문제를 줄임
 - 특정한 샤드에 대한 접근이 지나치게 빈번하면 서버 과부하 문제가 생길 수 있는데 안정 해시는 데이터를 좀 더 균등하게 분배하므로 이런 문제가 생길 가능성을 줄임

안정 해시의 유명한 예시

- 아마존 다이نامो 데이터베이스의 파티셔닝 컴포넌트
- 아파치 카산드라 클러스터에서의 데이터 파티셔닝
- 디스코드 채팅 어플리케이션
- 아카마이 CDN
- 매그레프 네트워크 부하 분산기

▼ 토론

가상 노드 개수는 시스템의 요구 사항에 맞게 적절하게 조절하라고 했는데 어떤 경우에 비교적 적은 수를, 어떤 경우에 비교적 많은 수의 노드를 필요로 할까?

1. 적은 수의 가상 노드:

- 작은 규모의 네트워크: 작은 규모의 애플리케이션이나 웹사이트의 경우에는 비교적 적은 수의 가상 노드로 충분할 수 있습니다.
- 낮은 트래픽 부하: 트래픽이 많이 발생하지 않는 경우에는 가상 노드의 수를 적게 유지하여 시스템의 자원을 효율적으로 사용할 수 있습니다.

- 예산 제약: 가상 노드의 추가 비용을 고려해야 할 때, 적은 수의 가상 노드를 사용하는 것이 경제적일 수 있습니다.

2. 많은 수의 가상 노드:

- 대규모 시스템: 대규모 애플리케이션이나 분산 시스템의 경우, 많은 수의 가상 노드가 필요할 수 있습니다. 이는 시스템의 성능과 확장성을 향상시키는 데 도움이 됩니다.
- 높은 트래픽 부하: 많은 수의 사용자 또는 트래픽 부하가 예상되는 경우, 많은 수의 가상 노드를 사용하여 부하를 분산시킬 수 있습니다.
- 고가용성 및 내결함성: 많은 수의 가상 노드를 사용하면 시스템의 고가용성 및 내결함성을 향상시킬 수 있습니다.

요구 사항 및 시스템의 특성에 따라 가상 노드의 개수를 조절하여 최적화하는 것이 중요합니다.