

시스템 규모 확장

시스템 사용자 수에 따라 설계해야 하는 규모가 달라진다.
수백만의 이용자가 존재하는 시스템을 개발해야 한다면, 어떤 것들을 고려해야 할 지 알아보자

1. 무상태(stateless) 웹 계층

수평적으로 확장하기 위해 필요하다. 즉, 사용자 세션 정보와 같은 상태 정보를 데이터베이스와 같은 지속 가능한 저장소에 맡기고, 웹 계층에서는 필요할 때 가져다 사용하는 방식으로 만든다.

웹 계층에서는 무상태를 유지하면서, 어떤 사용자가 http 요청을 하더라도 따로 분리한 공유 저장소에서 해당 데이터를 불러올 수 있도록 구성한다.

수평적 확장은 여러 서버를 추가하여 Scale out하는 방식으로, 이처럼 웹 계층에서 상태를 지니고 있지 않으면, 트래픽이 늘어날 때 원활하게 서버를 추가할 수 있게 된다.

2. 모든 계층 다중화 도입

데이터베이스를 주-부로 나누어 운영하는 방식을 다중화라고 말한다. 다중화에 대한 장점은 아래와 같다.

- 더 나은 성능 지원 : 모든 데이터 변경에 대한 연산은 주 데이터베이스 서버로 전달되는 반면, 읽기 연산은 부 데이터베이스 서버들로 분산된다. 병렬로 처리되는 쿼리 수가 늘어나 성능이 좋아지게 된다.
- 안정성 : 데이터베이스 서버 가운데 일부분이 손상되더라도, 데이터를 보존할 수 있다.
- 가용성 : 데이터를 여러 지역에 복제하여, 하나의 데이터베이스 서버에 장애가 발생해도 다른 서버에 있는 데이터를 가져와서 서비스를 유지시킬 수 있다.

3. 가능한 많은 데이터 캐시

캐시는 데이터베이스 호출을 최소화하고, 자주 참조되는 데이터를 메모리 안에 두면서 빠르게 요청을 처리할 수 있도록 지원해준다. 따라서 데이터 캐시를 활용하면, 시스템 성능이 개선되며 데이터베이스의 부하 또한 줄일 수 있다. 캐시 메모리가 너무 작으면, 액세스 패턴에 따라 데이터가 너무 자주 캐시에서 밀려나 성능이 떨어질 수 있다. 따라서 캐시 메모리를 과할당하여 캐시에 보관될 데이터가 갑자기 늘어났을 때 생길 문제를 방지할 수 있는 솔루션도 존재한다.

4. 여러 데이터 센터를 지원

데이터 센터에 장애가 나는 상황을 대비하기 위함이다. 실제 AWS를 이용할 때를 보더라도, 지역별로 다양하게 데이터 센터가 구축되어 있는 모습을 확인할 수 있다. 장애가 없는 상황에서 가장 가까운 데이터 센터로 사용자를 안내하는 절차를 보통 '지리적 라우팅'이라고 부른다. 만약 해당 데이터 센터에서 심각한 장애가 발생한다면, 모든 트래픽을 장애가 발생하지 않은 다른 데이터 센터로 전송하여 시스템이 다운되지 않도록 지원한다.

5. 정적 콘텐츠는 CDN을 통해 서비스

CDN은 정적 콘텐츠를 전송할 때 사용하는 지리적으로 분산된 서버의 네트워크다. 주로 시스템 내에서 변동성이 없는 이미지, 비디오, CSS, Javascript 파일 등을 캐시한다.

시스템에 접속한 사용자의 가장 가까운 CDN 서버에서 정적 콘텐츠를 전달해주므로써 로딩 시간을 감소시켜준다. 즉, CDN 서버에서 사용자에게 필요한 데이터를 캐시처럼 먼저 찾고, 없으면 그때 서버에서 가져다가 전달하는 방식으로 좀 더 사이트 로딩 시간을 줄이고, 데이터베이스의 부하를 줄일 수 있는 장점이 있다.

6. 데이터 계층은 샤딩을 통해 규모를 확장

데이터베이스의 수평적 확장을 말한다. 샤딩은 대규모 데이터베이스를 shard라고 부르는 작은 단위로 분할하는 기술을 말한다. 모든 shard는 같은 스키마를 사용하지만, 보관하는 데이터 사이에 중복은 존재하지 않는다. 샤딩 키(파티션 키라고도 부름)를 적절히 정해서 데이터가 잘 분산될 수 있도록 전략을 짜는 것이 중요하다. 즉, 한 shard에 데이터가 몰려서 과부하가 걸리지 않도록 하는 것이 핵심이다.

- 데이터의 재 샤딩 : 데이터가 너무 많아져서 일정 shard로 더이상 감당이 어려울 때 혹은 shard 간 데이터 분포가 균등하지 못하여 어떤 shard에 할당된 공간 소모가 다른 shard에 비해 빨리 진행될 때 시행해야 하는 것
- 유명인사 문제 : 핫스팟 키라고도 부름. 특정 shard에 질의가 집중되어 과부하 되는 문제를 말한다.
- 조인과 비 정규화 : 여러 shard로 쪼개고 나면, 조인하기 힘들어지는 문제가 있다. 이를 해결하기 위한 방법은 데이터베이스를 비정규화하여 하나의 테이블에서 질의가 수행가능하도록 한다.

7. 각 계층은 독립적 서비스로 분할

마이크로 서비스라고 많이 부른다. 서비스 별로 독립적인 체계를 구축하면, 하나의 서비스가 다운이 되더라도 최대한 다른 서비스들에 영향을 가지 않도록 할 수 있다. 따라서 시스템 규모가 커질수록 계층마다 독립된 서비스로 구축하는 것이 필요해질 수 있다.

8. 시스템에 대한 모니터링 및 자동화 도구 활용

- 로그 : 에러 로그 모니터링. 시스템의 오류와 문제를 쉽게 찾아낼 수 있다.
- 메트릭 : 사업 현황, 시스템 현재 상태 등에 대한 정보들을 수집할 수 있다.
- 자동화 : CI/CD를 통해 빌드, 테스트, 배포 등의 검증 절차를 자동화하면 개발 생산성을 크게 향상시킨다.

[참고 자료]

- 가상 면접 사례로 배우는 대규모 시스템 설계 기초