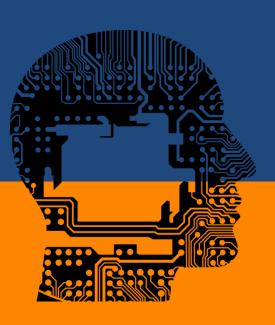


3강 - 데이터베이스 아키텍처

By SoonGu Hong





1. 아키텍처란?

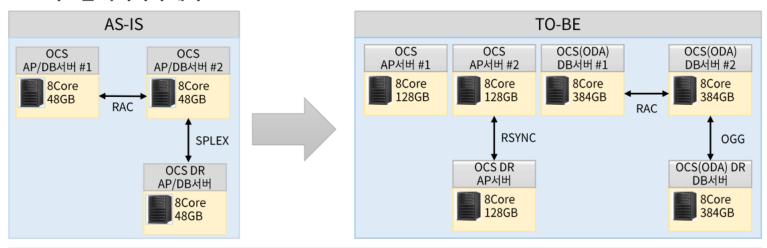
아키텍처 (Architecture)

- ① 시스템을 만들기 위한 물리 레벨의 조합(서버의 기능, 저장소와 네트워크 기기의 조합 등)
- ② 데이터베이스 설계에서 시스템의 구성
- ③ 아키텍처 통해 시스템의 용도와 목적 추측 가능

➤ IT 아키텍처 (Architecture)

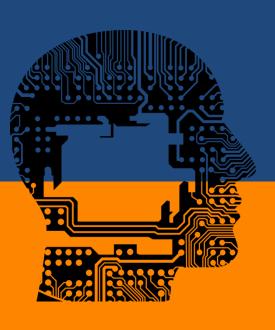
① 일정 기준과 절차에 따라 조직 전체의 정보화 구성요소를 통합분석 후 그 관계를 구조적으로 정리한 체제 이를 바탕으로 정보시스템을 효율적으로 구성하기 위한 방법

❖ IT 시스템 아키텍처 예시



- 아키텍처 구성을 시스템의 목적에 맞게 결정하는 과정, 서버, OS, 미들웨어, 저장소 등 폭넓은 지식 필요
- 적정 비용으로 필요 시스템 구축하기 위해서도 매우 중요





2. 아키텍처의 역사와 개요

▶ IT 아키텍처의 역사

단계	특징
Stand-alone	- 1980년대까지 널리 이용 - 데이터베이스만으로 시스템 운용
클라이언트/서버	- 1990년대~2000년 - 클라이언트와 서버로 계층 분리 - 상호 네트워크 접속
WEB 3계층	- 2000년~현재 - WEB, WAS, DBMS로 구분 - 클라이언트/서버 단계를 발전시킨 현재의 주류 모델

- 최근에는 클라우드 시대를 맞이하여 기존의 틀을 뒤엎은 다양한 IT 아키텍처가 등장함
- 서버가 마치 존재하지 않는 것처럼 운영되는 서버리스 개념도 존재함
- 다양한 클라우드 서비스 업체가 경쟁 중
- 클라우드를 이해하기 위해서도 기존의 전통적인 방식의 IT 아키텍처의 지식은 필수임









➤ Stand-alone 단계(DBMS 관점)

- ① DBMS 서버가 네트워크 접속 없이 독립적으로 작동
- ② 20세기 현대적 컴퓨터의 최초 등장 시기

장점	단점
 구축 과정이 간단함(소규모 작업 및 테스트를 빨리 처리할 수 있음) 높은 보안(네트워크 연결 불가) 	 물리적으로 떨어진 장소에서 접근 불가 복수 사용자가 동시에 작업 불가(1명만 이용가능) 낮은 가용성(Availability) - 서버가 단 1대임 확장성 부족(성능개선 여지가 부족함)

❖ Stand-alone

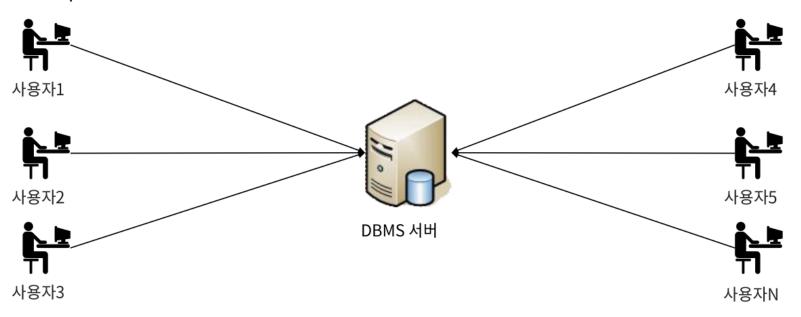


➤ 클라이언트/서버 단계(DBMS 관점)

- ① 네트워크 연결을 통한 복수 사용자가 동시에 사용가능
- ② DB 서버 한대에 복수 사용자가 접속하는 구성이 주를 이름

장점	단점
- 원격지에서도 사용 가능	인터넷 환경에서 접속 시 보안이 위험각각의 사용자 PC에서 애플리케이션 설치
- 복수의 사용자가 <mark>동시에 사용가능</mark>	(수정 및 배포의 어려움 존재)

Client/Server

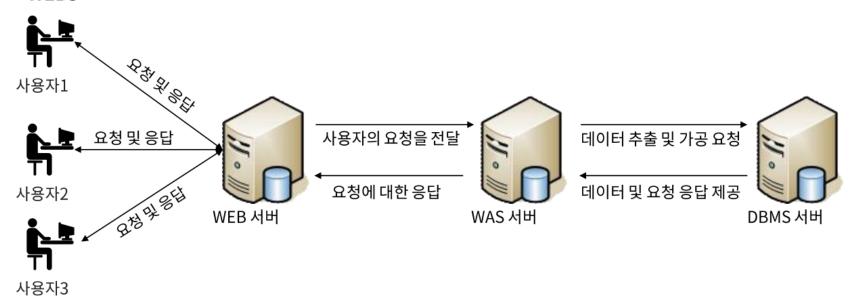


➤ WEB3 계층(DBMS 관점)

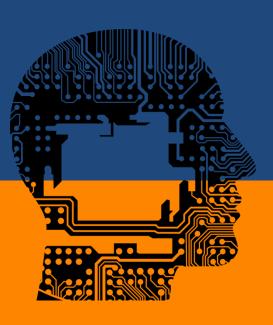
- ① 애플리케이션을 WAS 서버에서 관리
- ② 시스템을 3가지 계층의 조합으로 인식하기 시작함(WEB, WAS, DBMS)

장점	단점
 직접적인 접속 요청을 웹서버 계층에 한정하여 보안이 향상됨 애플리케이션 계층에 비즈니스 로직이 집중됨(관리의 용이성 및 비용 절감) 	- Stand Alone 및 클라이언트/서버 방식에 비해 크게 단점 이 존재하지 않음

❖ WEB3







3. 가용성과 확장성 확보

▶ 가용성이란?

- ① 가용성(Availability)이란 서버와 네트워크, 프로그램 등의 정보 시스템이 <mark>정상적으로 사용 가능한 정도</mark>를 말한다. 가동률과 비슷한 의미이다.
- ② 가용성을 수식으로 표현할 경우, 가용성(Availability)이란 정상적인 사용 시간(Uptime)을 전체 사용 시간 (Uptime+Downtime)으로 나눈 값을 말한다. 이 값이 높을수록 "가용성이 높다"고 표현한다.
- ③ 가용성이 높은 것을 고가용성(HA, High Availability)이라고 한다.

Availability(%) =
$$\left(\frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}}\right) * 100$$
 - 가용성이 1 을 뜻한다.

가용성이 100이라 하면 단 한번도 장애없이 동작한 시스템을 뜻한다.

▶ 확장성이란?

- ① 확장성(Scalability)는 IT 시스템에서 대규모적인 재설계 및 재설치가 필요없이 <mark>확장이 얼마나 쉽고 가능한지에</mark> 대한 용이성을 뜻한다.
- ② DBMS 설계자는 DBMS의 확산이나 거대한 성장을 도모해야 함
- ③ 절대적인 사용자 수가 증가하더라도 이를 수용할 수 있도록 확장성 있게 설계해야 함을 의미함
- ④ 최근의 클라우드 시스템이 각광을 받는 이유 중 하나가 바로 탁월한 확장성에 있음

▶ 가용성을 높이는 전략

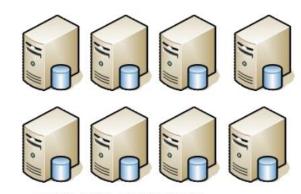
고품질-소수	저품질-다수
 DBMS 서버의 고품질을 추구함 소수의 DBMS 서버를 이용함 소수의 DBMS 서버를 높은 견고함과 신뢰성으로 무장 시킴 	- DBMS 서버가 품질이 떨어지더라도 <mark>다수의 DBMS를 사용</mark> 하는 전략 - 클러스터링 전략(<mark>동일 기능의 DBMS 서버를 다수로 구축</mark>)

❖ 고품질-소수



- DBMS 서버의 신뢰성을 높여간다.

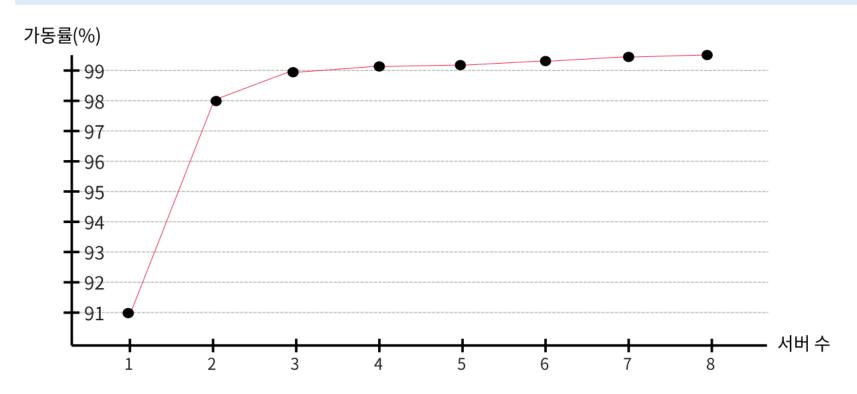
❖ 저품질-다수



- DBMS 서버를 병렬화 한다.

➤ 저품질-다수 전략의 용이성

- ① 동일 기능을 하는 DBMS 서버를 여러 대 설치 및 운영하여 병렬화 시킴
- ② 여러 대의 DBMS 서버가 한 개의 시스템을 위해서 존재함
- ③ 다중화/여유도 확보
- ④ 서버를 늘릴수록 장애 발생률은 확률적으로 자연 감소함



➤ 단일 장애 점(SPOF, Single Point Of Failure)

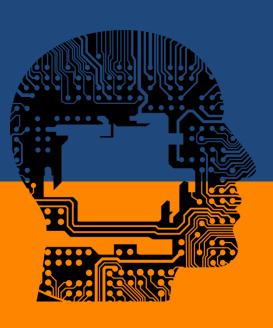
- ① 단일 장애 점(single point of failure, SPOF)은 시스템 구성 요소 중에서, <mark>동작하지 않으면 전체 시스템이 중단되는 요소</mark>를 말한다. 높은 가용성을 추구하는 네트워크, 소프트웨어 애플리케이션, 상용 시스템에 단일 장애점이 있는 것은 바람직하지 않다.
- ② 높은 신뢰성을 요구하는 시스템은 단일 컴포넌트에 의존하지 않는 것이 좋다. (ex. 금융권 시스템)

▶ 신뢰성 VS 가용성

신뢰성(Reliability)	가용성 (Availability)
하드웨어나 소프트웨어가 고장 나는 빈도 및 고장 기간컴포넌트 자체의 문제	사용자 입장에서 시스템을 어느 정도 사용할 수 있는가시스템 전체 수준의 문제

- 신뢰성이 낮아도 여러 대의 서버를 구축하는 클러스터링 기법(저품질-다수)을 통해 가용성 확보가 가능





4. DB 서버의 다중화

▶ DB 서버의 다중화

- ① 다른 컴포넌트에 비해 다중화 어려움
- ② 영속(Persistence) 계층의 특성
 - 데이터 장기간 보존 필요
 - 일시적 처리만 담당하는 애플리케이션 서버 등과 차이
 - 데이터 다중화 시 갱신 통한 정합성 중요

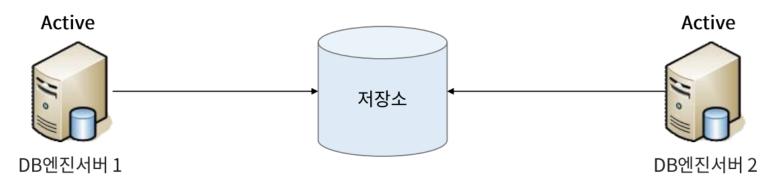
▶ DB서버 다중화 유형

유형	설명
Active-Active	- 두개의 DBMS 엔진 서버가 동시에 가동됨 - 저장소는 한 곳을 바라봄(저장소는 1개)
Active-Standby	- 평소에는 Active만 운영하고 나머지 서버는 Standby 상태 - 저장소는 한 곳을 바라봄
리플리케이션	 DB 서버와 저장소를 하나 세트로 하여 미리 준비 데이터 동기화가 중요함(운영->DR) 리플리케이션은 데이터베이스 서버와 저장소가 동시에 사용 불능일 때 서비스를 계속할 수 있도록 해주는 매우 가용성이 높은 아키텍처(주로 DR 시스템에 주로 사용됨)

- DB 서버는 업무적 측면, 기술적 측면, 비용적 측면, 조직 운영 등을 고려하여 결정해야함

➤ Active-Active 구성 다중화

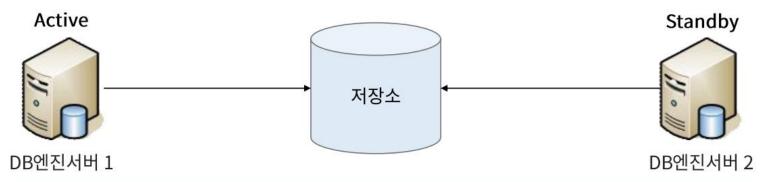
- ① 장애 발생 시 Downtime이 거의 없다고 해도 무방함
- ② 하나의 서버가 다운되어도 나머지 하나가 계속 처리함
- ③ 2대의 서버가 운영되므로 성능상 유리(저장소 병목이 없을 경우에)



- DB엔진서버 1번에 장애가 발생 시 DB 엔진서버 2가 Active 상태로 있으므로 업무 처리가 가능함
- 평소에(장애상황이 아닌 경우)도 2대의 서버가 운영되므로 성능상 유리함
- 저장소의 병목 현상이 있을 경우 순간적으로 성능에 문제가 발생하는 경우가 있음
- Oracle DBMS의 RAC가 대표적인 Active-Active 다중화 제품임
- 국산 DBMS인 티베로도 TAC라는 Active-Active 다중화 기술을 지원함

Active-Standby 구성 다중화

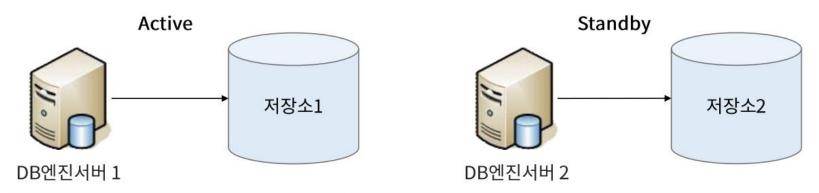
- ① 평소에는 Active 서버로만 업무를 처리함
- ② Active 서버가 장애 상황이 된 경우 Standby 서버가 업무를 처리함
- ③ Standby 서버로 전환 시까지 Downtime이 존재함



- Cold-Standby는 평소에는 Standby 서버는 작동을 안함 Active 장애시에만 작동
- Hot-Standby는 평소에도 Standby 서버가 작동함 -> Downtime 시간이 감소됨
- 저장소 병목으로 인한 성능 상 이슈는 없음
- 무조건 1대의 서버로만 운영되므로 그로인한 성능 상 불리한 점이 있음
- Active-Active 구성에 비해서 비용상 유리하고 관리가 용이한 장점이 있음
- 대부분의 DBMS가 채택하는 방식임

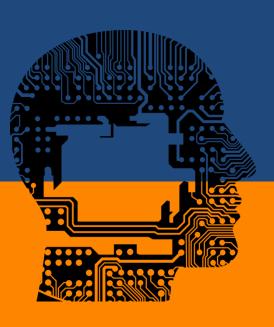
▶ 리플리케이션 다중화

- ① Active-Active, Active-Standby는 저장소는 다중화 되지 않음
- ② 하나의 저장소에 문제가 생기면 바로 장애 상황이 발생
- ③ 하나의 저장소에만 데이터가 저장되므로 데이터 소실 가능성이 있음
- ④ 리플리케이션 방식은 DB서버와 저장소를 복수의 세트로 준비함



- DB엔진서버1번과 DB엔진서버2번은 거의 실시간으로 데이터가 동기화됨
- 특수한 재해상황(전쟁, 정전 등)에 대한 대책으로 이용함
- Active 세트(시스템)과 Standby 세트(시스템)은 서로 다른 지역에 서버를 설치함
- 금융기관, 공공기관 등에서 시스템 위험 분산의 대책으로 이용됨
- 거의 실시간 데이터 동기화로 인한 성능 이슈가 발생할 수 있음
- 리플리케이션 다중화는 비용이 매우 많이 들어감





5. 성능 추구를 위한 다중화

> Shared Disk와 Shared Nothing

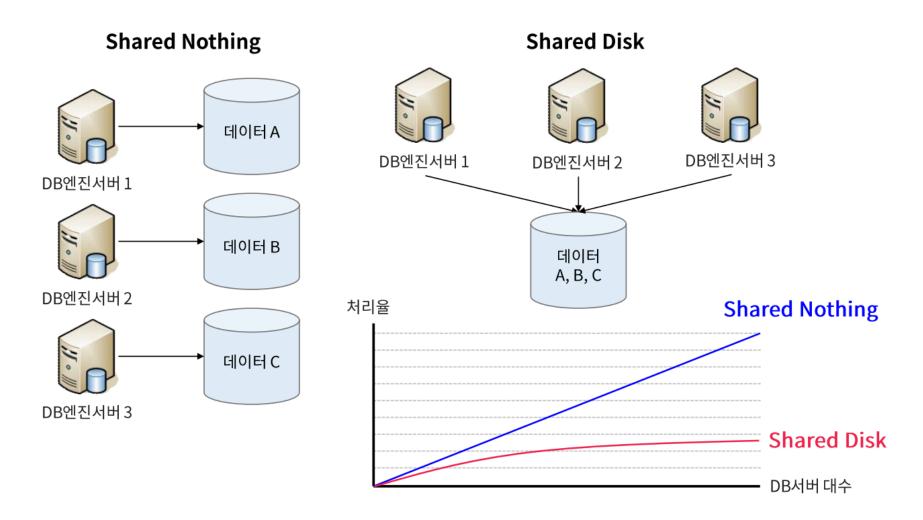
Shared Disk	Shared Nothing
- Active-Active 구성 DB - 저장소 공유로 인한 병목 현상 발생 - DB서버 늘려도 한계점에 도달함	 서버와 저장소 세트를 늘려 병렬처리 세트를 늘린 것만큼 비례하여 처리율 증가함 샤딩 기술이 대표적임 DB서버 다운 시 다른 서버가 이어받아 처리할 수 있는 커버링 전략이 필요

> Shard

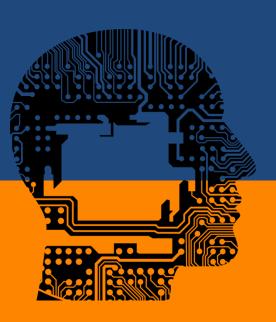
- 데이터베이스 샤드(database shard)는 데이터베이스나 웹 검색 엔진의 데이터의 수평 분할이다. 개개의 파티션은 샤드(shard) 또는 데이터베이스 샤드(database shard)로 부른다. 각 샤드는 개개의 데이터베이스 서버 인스턴스에서 부하 분산을 위해 보유하고 있다.
- 데이터베이스 내의 일부 데이터는 모든 샤드에 존재하지만 일부는 하나의 샤드에만 존재한다. 각 샤드(또는 서버)는 이 데이터 부분을 위해 "하나"의 소스로서 동작한다.

<u>샤드 (데이터베이스 아키텍처) - 위키백과, 우리 모두의 백과사전 (wikipedia.org)</u>

➤ Shared Disk와 Shared Nothing의 성능 비교



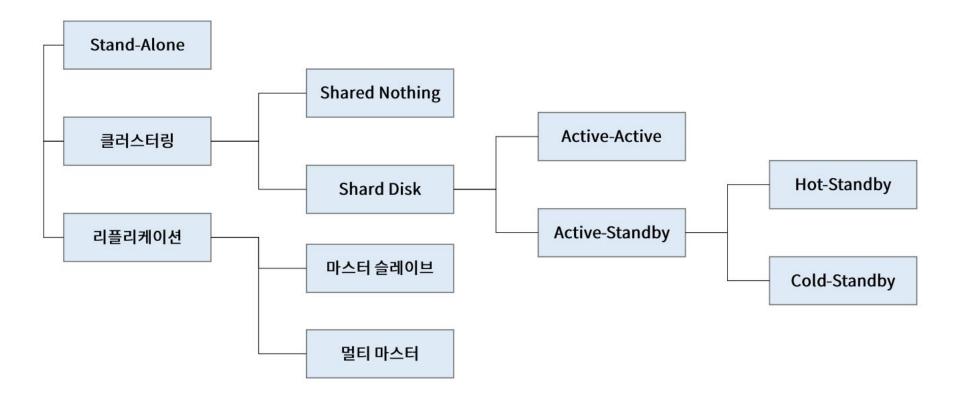




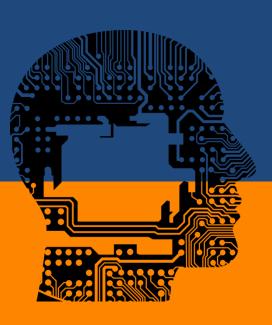
6. 적합한 아키텍처를 설계하기 위해

▶ 최적의 아키텍처 설계 전략

- ① 가용성, 신뢰성, 재해대책, 성능, 보안, 조직, 비용 등 다양한 조건을 고려해야함
- ② 해당 비즈니스의 성장성이나 사용자의 유입 예측도 감안해야 함
- ③ 한번 구축하면 변경 시 비용과 시간이 소요됨







7. 연습 문제

문제 1. Web 3계층 모델 구성하는 컴포넌트가 아닌 것은?

- ① DB 서버
- ② 파일서버
- ③ 애플리케이션 서버
- 4 웹 서버

문제 2. DB 서버를 Active-Active 클러스터 구성으로 구축한 경우의 장점이 아닌 것은?

- ① 시스템의 가용성이 높아진다.
- ② 데이터베이스의 성능이 향상된다.
- ③ DB 서버 장애 시 다운 시간이 거의 걸리지 않는다.
- ④ Active-Standby와 비교해서 비용이 적게 든다.

문제 3. 특정 시스템의 Uptime은 1000분이었고 Downtime은 10분이었다 가용성(%)은 얼마인가?

- 1) 99
- 2 100
- 3 90
- (4) 98

감사합니다 THANK YOU