

# Data-Centric Application Design

<https://product.kyobobook.co.kr/detail/S000001766328>

인터넷은 잘 동작하고 있어서 대부분의 사람들은 인공의 무언가라기보다 태평양같은 천연자원으로 생각한다. 이런 규모의 기술이 이토록 오류가 없었던 적은 언제가 마침가일까?

요즘 애플리케이션 = 데이터 중심 > (이전) 계산 중심

애플리케이션 표준 구성 요소

- : 데이터베이스, 캐시, 검색 색인, 스트림 처리, 일괄 처리 등
- : 기능적 요구 사항 & 비기능적 요구사항 (보안, 신뢰성, 법규 준수, 호환성, 유지보수성 등)

데이터 시스템의 원칙과 실용성

복합 데이터 시스템

신뢰성 : 결함성, 탄력성 : 넷플릭스의 카오스 몽키

- 소프트웨어 오류, 인적 오류

확장성 : 부하 매개변수 : 처리량, 응답 시간, [꼬리]지연 시간

- 서비스 수준 목표, 선두 차단
- 용량 확장, 규모 확장

유지보수성 : 운용성, 단순성=복잡도 관리/추상화, 발전성, 유연성, 수정 가능성, 적응성

내 언어의 한계는 내 세계의 한계를 의미한다.

네트워크 모델=코다실 모델 / 게층 모델 > 관계형 모델 / 문서 모델=문서 참조

NoSQ

다중 저장소 지속성

임피던스 불일치

지역성 : 저장소 지역성

다대일과 다대다 관계

IMS (정보 관리 시스템)

스키마 우연성 - 스키마리스

マイグ레이션

맵리듀스

몽고 DB - 집계 파이프라인

속성 그래프 모델 / 트리플 저장소 모델

사이퍼 질의 언어

재귀 공통 테이블식

시맨틱 웹

자원 기술 프레임워크

스파클(SPARQL)

데이터로그

게놈 - 염기 서열 유사도 검색

전문 검색

항상 주변을 단정히 정돈하는 사람은 단지 찾기를 너무 귀찮아하는 사람이다.

로그 구조 계열 저장소 엔진 / 페이지 지향 계열 저장소 엔진

해시 색인

정렬된 문자열 테이블 (SS테이블) / 멤테이블

볼륨 필터

크기 계층

레벨 컴팩션

B 트리 - 분기 계수

LSM 트리 - 쓰기 증폭

기본키 색인, 보조 색인, 클러스터드 색인, 커버링 색인, 포괄열이 있는 색인

다중 칼럼 색인 - 결합 색인

레벤슈타인 오토마トン

트랜잭션 일반적인 특징 : ACID - 원자성, 일관성, 격리성, 지속성

온라인 트랜잭션 처리 (OLTP)

온라인 분석 처리 (OLAP)

비즈니스 인텔리전스

데이터 웨어하우징 < ETL 과정

드릴 다운, 슬라이싱, 다이싱

별 모양 스키마 ( 차원 모델링

눈꽃송이 모양 스키마

칼럼지향 저장소

비트맵 부호화

백터화 처리

구체화 집계 > 구체화 뷰

데이터 큐브 (OLAP 큐브

만물은 변한다. 그대로 있는 것은 아무것도 없다.

순회식 업그레이드 (단계적 롤아웃

부호화(직렬화/마샬링) / 복호화(피싱/역직렬화/언마샬링)

스리프트 : 바이너리프로토콜, 컴팩트프로토콜

스키마 발전

데이터가 코드보다 더 오래 산다. (Data outlives code.)

서비스 지향 설계 (SOA) > 마이크로서비스 설계

원격 프로시저 호출

위치 투명성

비동기 메세지 전달 시스템

메시지 브로커/메시지큐/메시지 지향 미들웨이

분산 액터 프레임워크

기술이 성공하기 위해서는 홍보보다 현실이 우선돼야 한다. 자연을 속일 수는 없기 때문이다.

잘못될 수 있는 것과 잘못될 수 없는 것 사이의 주된 차이점은 잘못될 수 없는 것이 잘못됐을 때는 잘못을 파악하거나 고치는 것은 거의 불가능하다는 점이다.

복제 - 단일 리더, 다중 리더, 리더 없는

리더 기반 복제(능동/수동, 마스터 슬레이브 복제)

- 리더(마스터, 프라이머리) + 팔로워(읽기 복제 서버, 슬레이브, 2차 핫 대기)

동기식 / 비동기식 팔로워

팔로워 장애, 리더 장애

구문 기반 복제, 쓰기 전 로그 배송, 논리적(로우 기반) 로그 복제 = 변경 데이터 캡쳐, 트리거 기반 복제

쓰기 후 일관성, 단조 읽기, 일관된 순서로 읽기

다중 데이터 센터 운영 - 성능 > 데이터센터 중단 내성 > 네트워크 문제 내성

오프라인 작업하는 클라이언트

협업 편집

쓰기 충돌 다루기

동기 대 비동기 충돌 감지

충돌 회피

일관된 상태 수렴

사용자 정의 충돌 해소 로직

전체 연결 / 원형 토플리지

정족수

암시된 핸드오프

최종 쓰기 승리 (LWW)

동시성, 시각 상대성

형제 값

[도티드] 버전 백터

과성 컨텍스트

순차성에서 벗어나야 하고 컴퓨터에 제한을 가하지 않아야 한다는 것은 분명하다. 데이터 정의를 명시하고 데이터의 우선순위와 데이터에 관한 설명을 제공해야 한다.

파티셔닝(샤딩) - 동적, 노드 비례

전역 색인 / 지역 색인

## 재균형화

### 스캐더 / 개더

### 대규모 병렬 처리

어떤 저자들은 2단계 커밋에서 유발되는 성능이나 가용성 문제 때문에 생기는 비용이 너무 커서 이를 지원할 수 없다고 주장했다. 우리는 항상 트랜잭션 없이 코딩하는 것보다 트랜잭션을 과용해서 병모깃점이 생기는 성능 문제를 애플리케이션 프로그래머가 처리하게 하는 게 낫다고 생각한다.

ACID / BASE(기본적으로 가용성을 제공하고, 유연한 상태를 가지며, 최종적 일관성을 지닌다)

### 어보트 능력 (abortability)

### 커밋 후 읽기

### 더티 읽기/쓰기 방지

### 커밋 후 읽기 구현

### 스냅숏 격리와 반복 읽기

### 스냅숏 격리 구현 (-다중 버전 동시성 제어(MVCC))

### 직렬성 - 직렬성 스냅숏 격리(SSI)

2단계 잠금 (2PL) - 공유 모드, 독점 모드 - 서술 잠금, 색인 범위 잠금

- 비관성 동시성 제어 매커니즘 ( $\neq$  낙관적 동시성 제어 매커니즘)

이봐 방금 널 만났어. 네트워크가 느려터졌군. 하지만 여기 내 데이터가 있으니. 아마도 저장 해주길.

네트워크 분단

타임아웃 / 기약 없는 지연

네트워크 혼잡

흐름제어 - 혼잡 회피, 배압

T

CP, UDP

변동성 (지퍼, jitter)

제한있는 지연

순간적으로 몰리는 트래픽

서비스 품질, 진입 제어

지원 시간, 자원 사용률

신뢰성 없는 시계

- 일 기준 시계(벽시계 시간), 단조 시계

논리적 시계 / 물리적 시계

## 엄격한 실시간 시스템

### 실시간 운영체제

비잔틴 결함 = 비잔틴 장군 문제

- 비잔틴 내결함성을 지닌다

### 시스템 모델

- 동기식 모델, 부분 동기식 모델, 비동기식 모델
- 죽으면 중단하는 결함, 죽으면 복구하는 결함, 비잔틴 결함

### 안전성, 활동성

### 부분실패

살아 있지만 틀린 게 나은가, 올바르지만 죽은 게 나은가?

### 스플릿 브레인

### 최종적 일관성

선형성 = 최신성 보장

- 원자적 일관성, 강한 일관성, 즉각 일관성, 외부 일관성

### 직렬성

CAP (일관성, 가용성, 분단 내성)

## 순서화

- 인과적으로 일관적
- 수학적 집합은 부분적으로 순서 결정
- 일렬번호(타임랩스) 순서화

## 램포트 타입스탬프

### 전체 순서 브로드캐스트, 원자적 브로드 캐스트

- 신뢰성 있는 전달과 전체 순서가 정해진 전달의 속성 항상 만족해야함

## 상태 기계 복제

## 보상 트랜잭션

### 데이터베이스 내부 분산 트랜잭션

### 이종 분산 트랜잭션

## XA 트랜잭션

### 경험적 결정 - 원자성을 깨 가능성 있음

### 에포크 번호, 투표 번호, 뷰 번호, 템 번호

한 사람의 영향도가 너무 큰 시스템은 성공하기 어렵다. 초기 설계가 완료되고 상당히 견고해지면 여러 사람이 다양한 관점을 가지고 각각 실험을 진행하면서 실제 테스트가 진행된다.

### 요청 > 응답 > 질의 > 결과

서비스(온라인 시스템), 일괄 처리 시스템(오프라인 시스템), 스트림 처리 시스템(준실시간 시스템)

느슨한 결합 (지연 바인딩, 제어 반전)

NAS, SAN

네임 노드

삭제 코딩 방식

맵리듀스

워크플로

이벤트 로그 (활동 이벤트, 클릭 스트림 데이터)

세션화

린치핀 객체 (핫 키)

쓸린 조인

맵사이드 조인

브로드캐스트 해시 조인

버킷 맵 조인

중간 상태

이행적 패쇄 알고리즘

벌크 동기식 병렬 연산 모델 (BSP)

K 최근접 이웃 알고리즘

복잡하지만 잘 작동하는 시스템은 예외 없이 간단하지만 잘 작동하는 시스템으로부터 발전 한다. 이 문제는 역도 참이다. 처음부터 복잡하게 설계된 시스템은 절대 작동할 리도 없고 작동하게 만들지도 못한다.

파생 데이터

스트림 처리

생산자 (발송자, 발행자) - 소비자 (구독자, 수신자)

메시징 시스템 (발행/구독 모델)

복수 소비자 - 로드 밸런싱, 팬 아웃

로그 기반 메시지 브로커

메시지 오프셋 ~ 로그 순차 번호

원형 버퍼 (링 버퍼)

변경 데이터 캡처 (CDC)

로그 컴팩션

명령과 질의 책임 분리 (CQRS)

복잡한 이벤트 처리 (CEP)

분석

구체화 뷰

분산 RPC

윈도우 - 텀블링 윈도우, 흡핑 윈도우, 슬라이딩 윈도우, 세션 윈도우

스트림 스트림 조인, 스트림 테이블 조인, 테이블 테이블 조인

천천히 변하는 차원 (SCD)

정확히 한 번 시맨틱 == 결과적으로 한번

마이크로 일괄 처리

멱등성

어떤 것이 결국 다른 것이 될 운명이라면 그 끝은 존재의 보존일 수 없다. 그러므로 선장은 그가 맡은 배를 보존하려 하지 않는다. 배는 결국 항해를 할 운명이기 때문이다.

철도 스키마 이전

람다 아키텍처

언벌들링

느슨한 결합

미분 데이터플로

함수형 반응형 프로그래밍 ~ 논리적 프로그래밍

관찰자 패턴

쓰기 경로 / 읽기 경로

오프라인 우선

종단 간 논증

적시성 / 무결성

코디네이션 회피

로해머 기법

자가 검증, 자가 감사

작업 증명

머클트리

인증서 투명성

예측 분선, 편견과 차별, 책임과 의무, 피드백 루프, 사생활과 추적, 감시, 동의와 선택의 자유, 사생활과 데이터 사용, 산업 혁명의 기억, 자산과 권력으로서의 데이터, 법률과 자기 규제