NoSQL Deep-Dive

[고려대학교 특강] Edward J. Yoon
<edwardyoon@apache.org>

연사 소개

Member, V.P., Committer at Apache Software Foundation

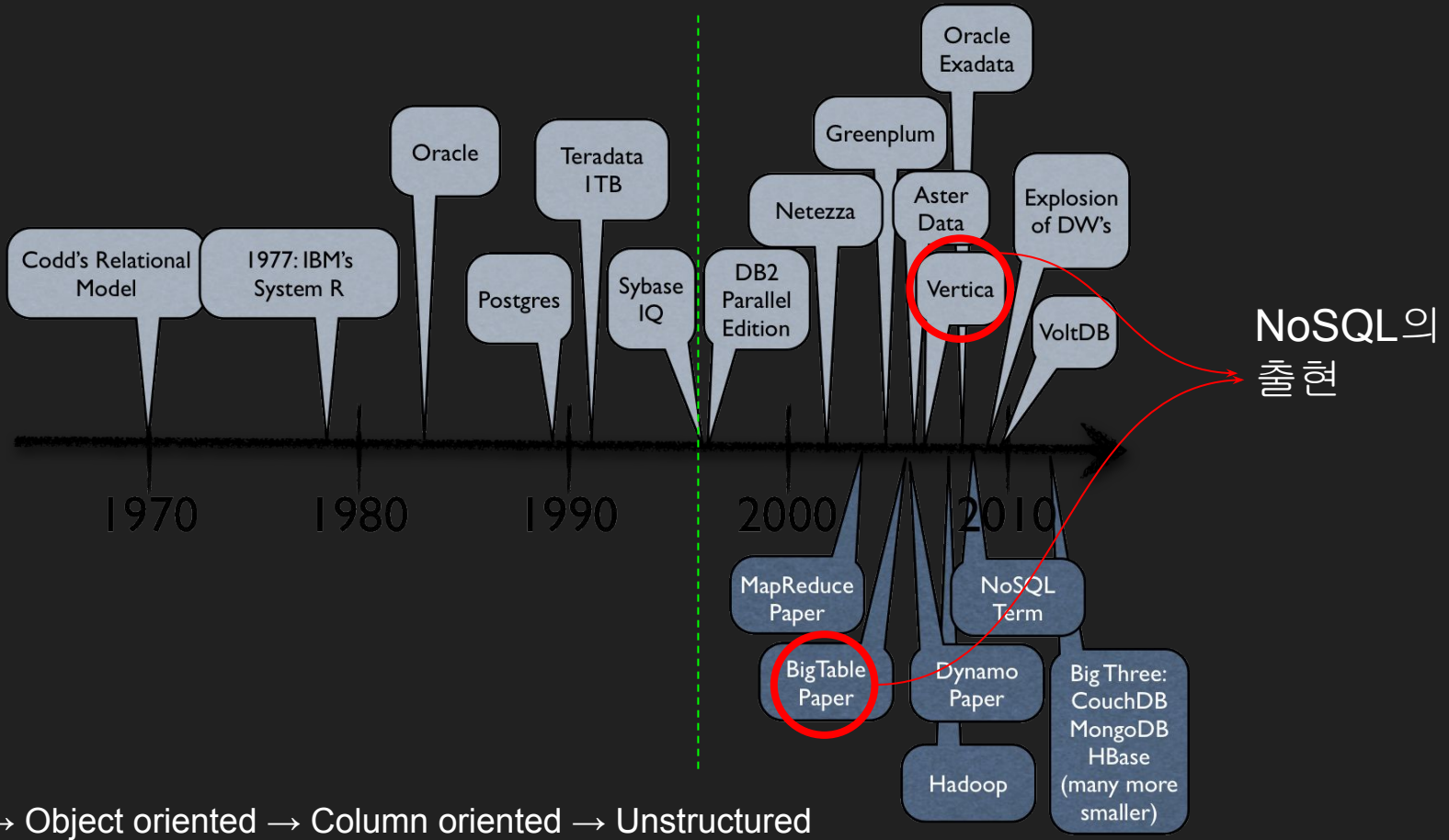
Apache Hama, OpenWhisk, BigTop, MRQL, …, etc.

- 2009년 ~ Stealth AI Startup 대표
- 2018년 ~ 2019년 스푼 라디오 R&D 총괄
- 2017년 ~ 2018년 위드이노베이션 CTO
- 2014년 ~ 2017년 삼성전자 AI팀 책임연구원
- 2012년 ~ 2013년 오라클 수석SW엔지니어
- 2011년 ~ 2012년 KT 유클라우드 추진본부 과장
- 2007년 ~ 2011년 네이버 소프트웨어 엔지니어

History of Database

- 1950년대 데이터베이스: 군사력 전체의 관리와 통제를 위한 데이터 기지 (base)
- 1970년대 - 에드거 프랭크 관계형 RDBMS 고안

“Data가 관계로 묶이면 Information이 된다”.
“구슬이 서 말이라도 꿰어야 보배”.



Structured → Object oriented → Column oriented → Unstructured

Large-scale NoSQL 출현 배경은?

- 기존 전통 DB는 단일 장비에 최적화된 구조로 진화
  - 인터넷 스케일의 데이터베이스가 필요해졌다.

- 즉, 월드 와이드 웹의 탄생과 무관하지 않음.
  - 기본적으로 WWW에는 국경이 없고 70억 인구를 대상으로 한다.
  - 사용자 데이터가 많다.
  - 이러한 데이터를 토대로 서비스를 고도화 할 수 있겠다.

예) 웹 메일, 웹 문서 기하급수적 증가, 인터넷 사용자 로그

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com
Google, Inc.

Abstract

Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to real-time data serving). Despite these varied demands, Bigtable has successfully provided a flexible, high-performance solution for all of these Google products. In this paper we describe the simple data model provided by Bigtable, which gives clients dynamic control over data layout and format, and we describe the design and implementation of Bigtable.

1 Introduction

achieved scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of the data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted strings, although clients often serialize various forms of structured and semi-structured data into these strings. Clients can control the locality of their data through careful choices in their schemas. Finally, Bigtable schema parameters let clients dynamically control whether to serve data out of memory or from disk.

Section 2 describes the data model in more detail, and Section 3 provides an overview of the client API. Section 4 briefly describes the underlying Google infrastruc-

WebTable

웹 문서 (Web Document) 저장 및 웹 페이지 랭킹 계산을 위해 고안

웹 문서 특징

- HTML attributes 가 너무 많고 계속 변화한다.
- 대부분 표준 가이드를 따르지 않기 때문에 웹 전체 데이터 구조가 굉장히 Sparse 하다.

요구사항

- 효율적인 저장 기법
- 특정 칼럼 광속 스캔, attribute-focused access 가 빨라야한다.

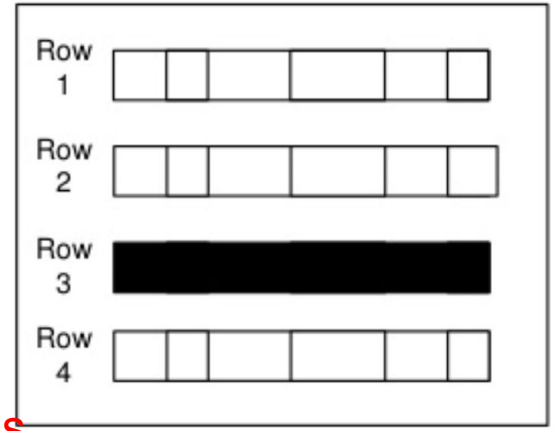
선택된 설계

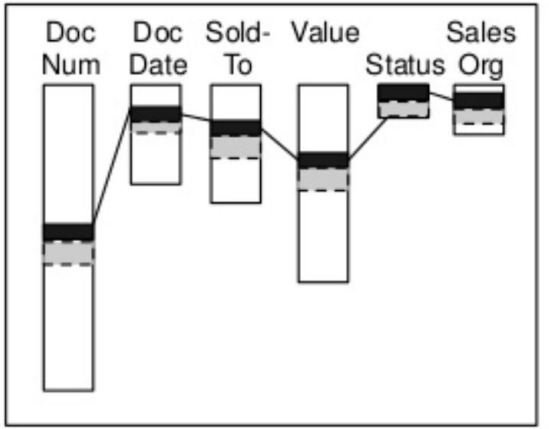
- Column-oriented → Sparse 테이블, Attribute-wise access
- Multi-dimensional → 지속변화하는 attributes, scheme-free
- Additional Time dimension → Time versions
- LSM tree + Distributed System → Fault-tolerant and large-scale

Row-wise vs Columnar

Optimal for row-wise access
(e.g., SELECT *)

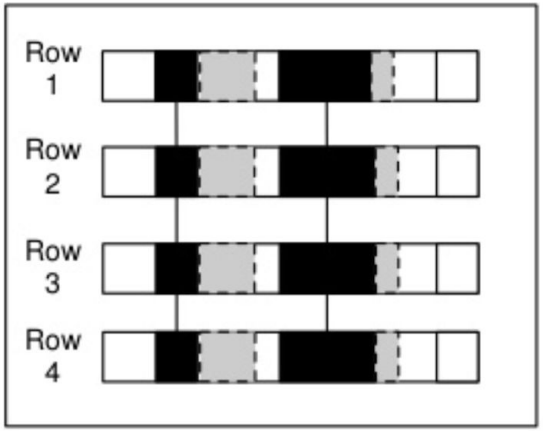
SELECT *
FROM Sales Orders
WHERE Document Number = '95779216'
(OLTP-style query)

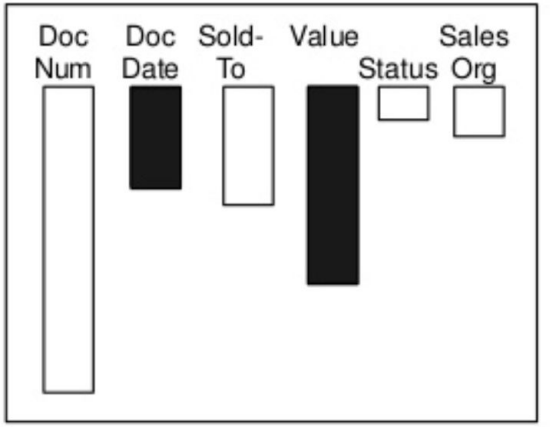




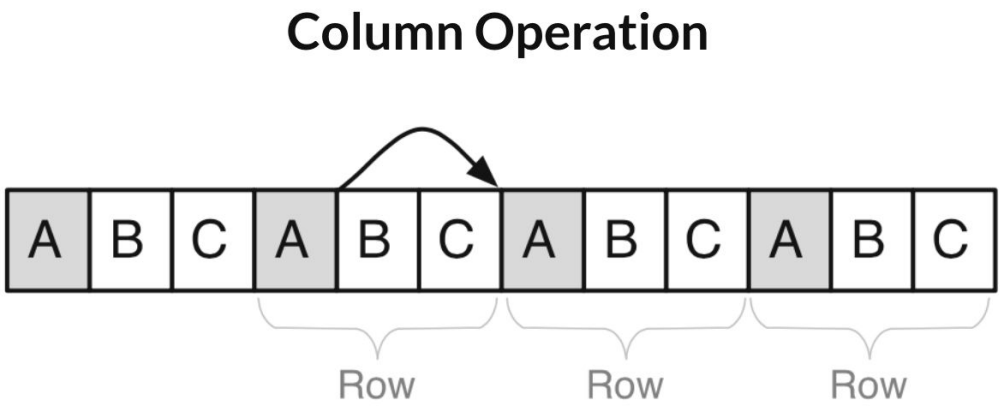
Optimal for attribute focused access
(e.g., SUM, GROUP BY)

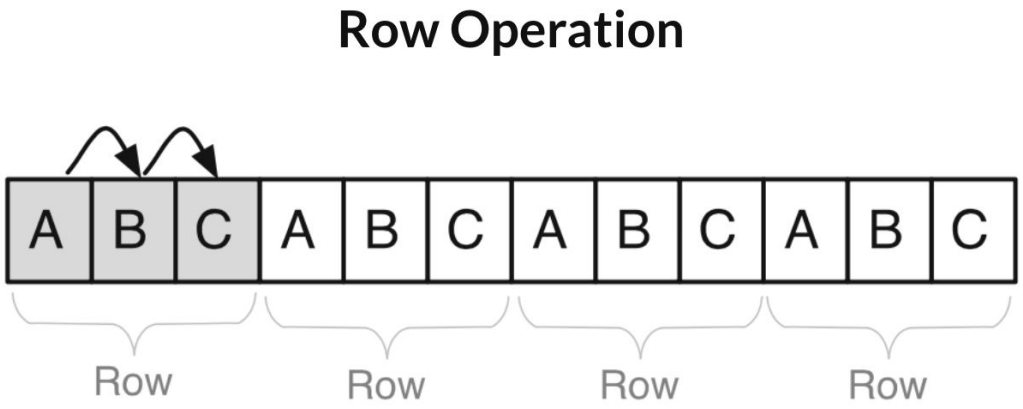
SELECT SUM(Value)
FROM Sales Orders
WHERE Document Date > 2011-08-28
(OLAP-style query)



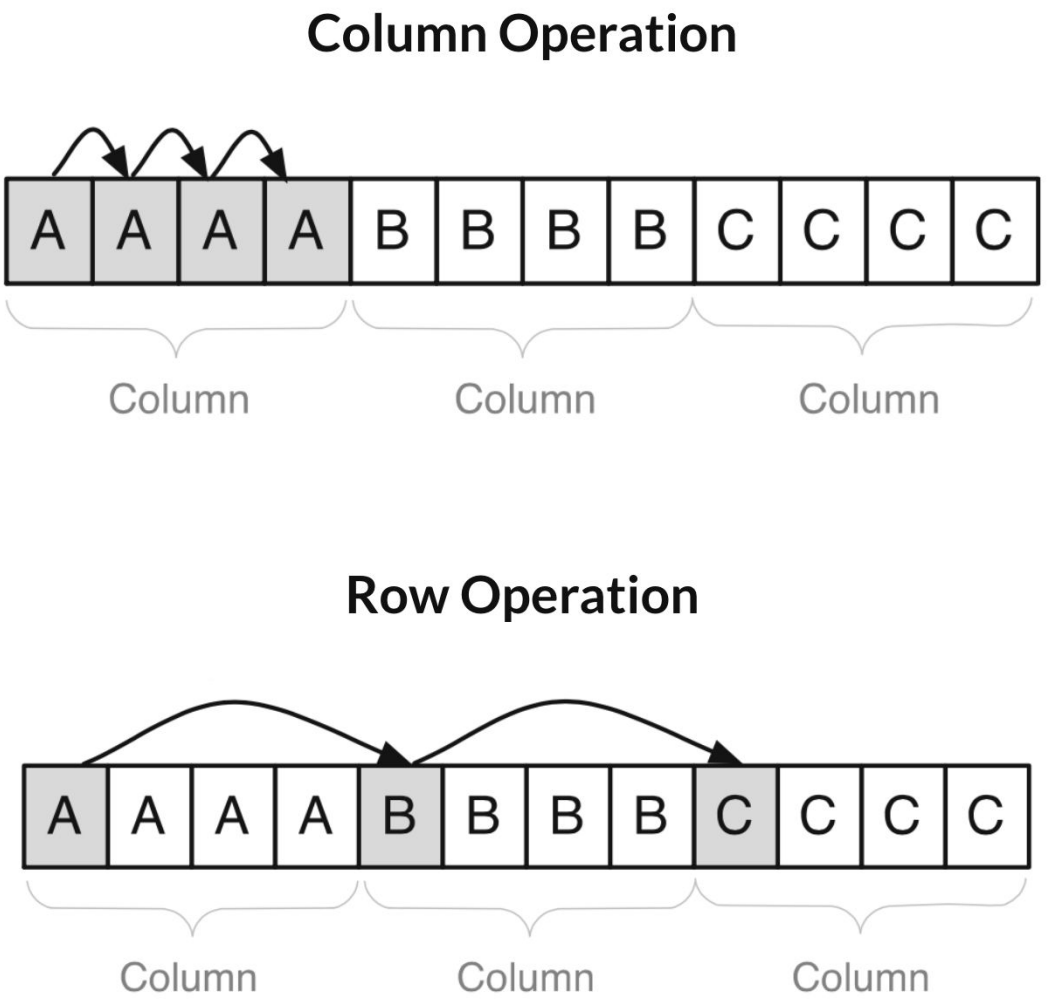


Row-wise

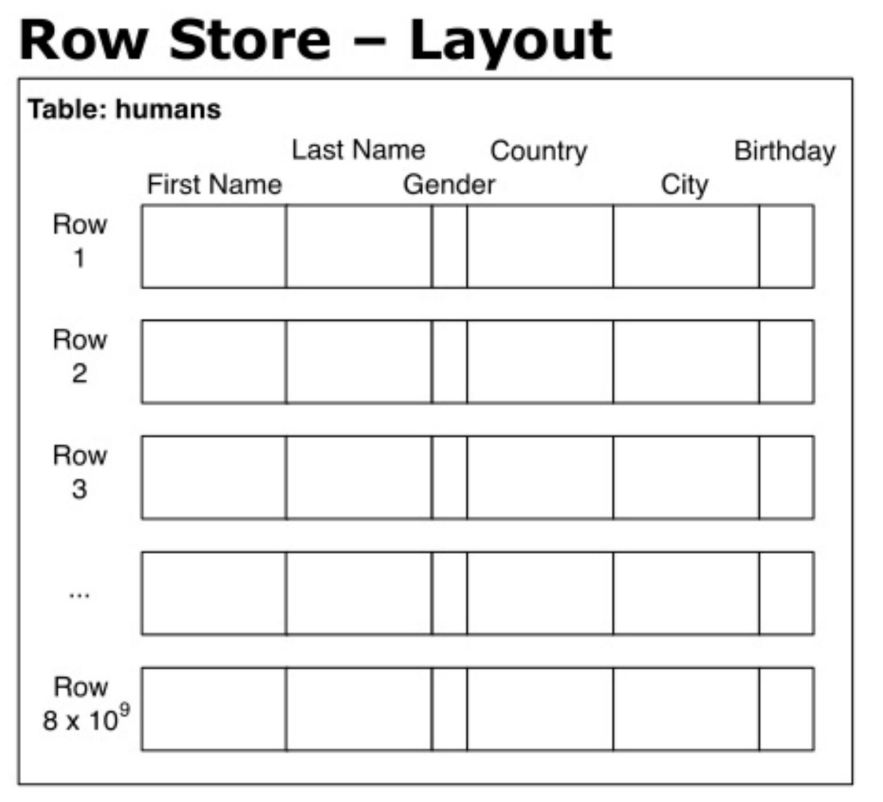

Column Operation
A B C A B C A B C A B C
Row
Row
Row

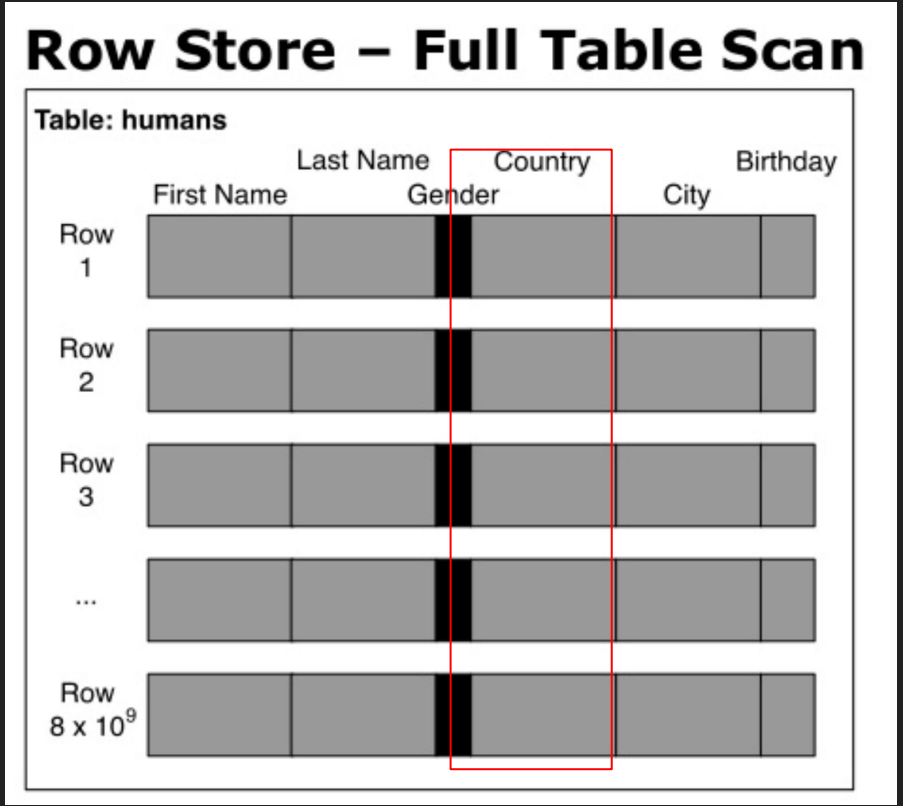

Row Operation
A B C A B C A B C A B C
Row
Row
Row
Row

Columnar

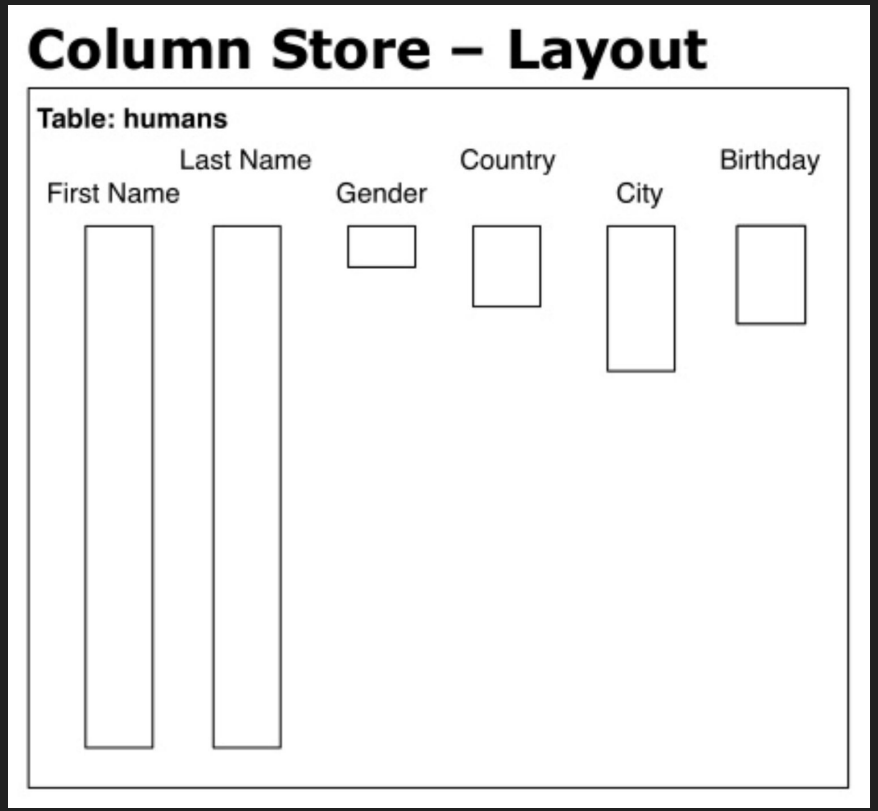

Column Operation
A A A A B B B B C C C C
Column
Column
Column
Row Operation
A A A A B B B B C C C C
Column
Column
Column

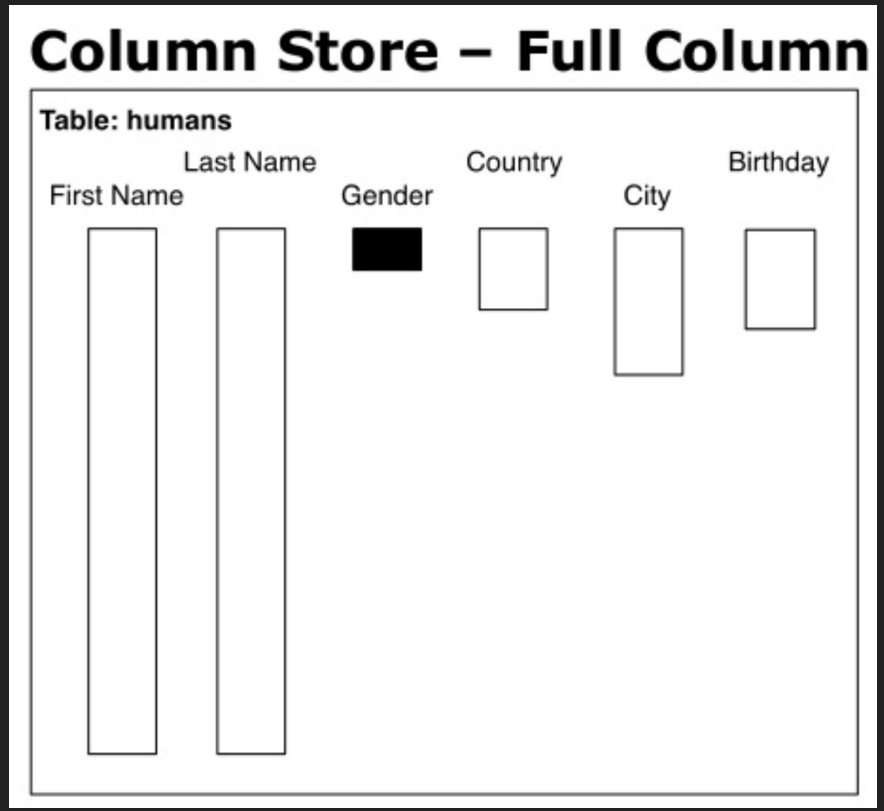
Scan Performance ~800 secs, Stride access ~256 secs

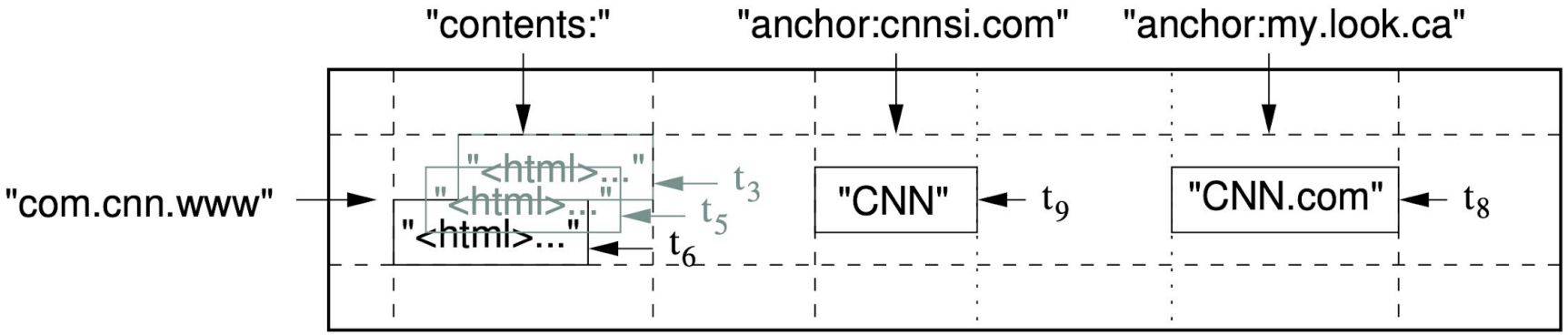




Scan Performance ~0.5 secs (scan through attribute vector)

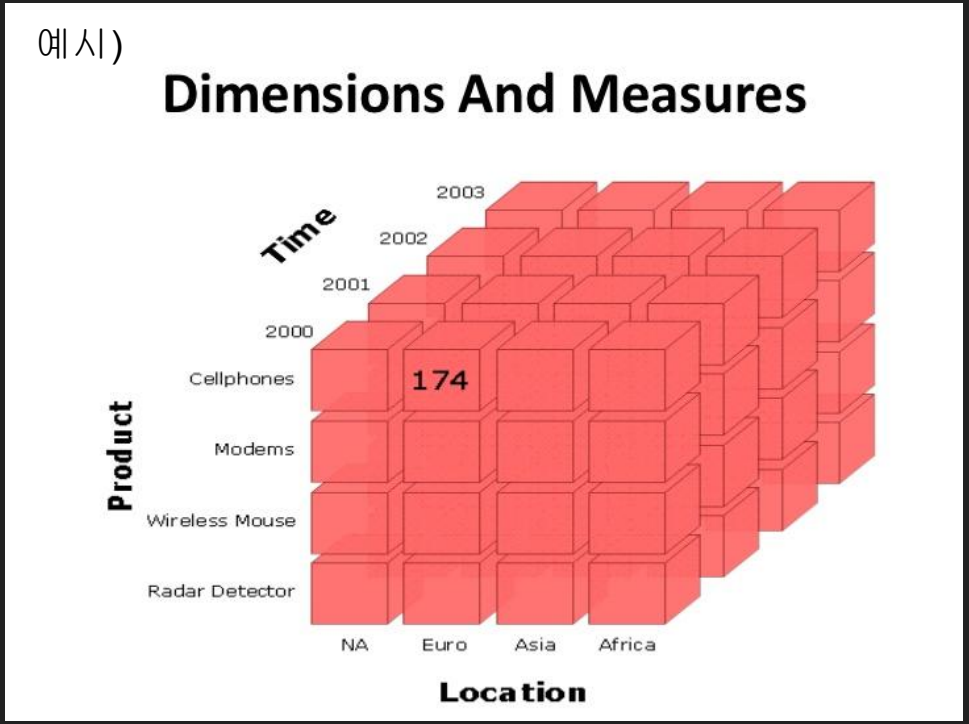





"contents:"
"anchor:cnnsi.com"
"anchor:my.look.ca"
"com.cnn.www"
"<html>..."
"<html>..."
"<html>..."
t_3
t_5
t_6
"CNN"
t_9
"CNN.com"
t_8

Multidimensional & Timestamp

예시)



WebTable
Row - URL, Columns - Attributes
수많은 3D table을 가지고 있음.

가령, 웹의 Row (in-link), Column (out-link) 와 Time dimension은

웹 네트워크 구조의 진화 과정을 모두 저장할 수 있으며, 광속 스캔하며 페이지랭크를 계산하는데 활용 가능함.

하나의 테이블은 메타 태블릿 조각을 가지고 있음.
(색인 데이터)

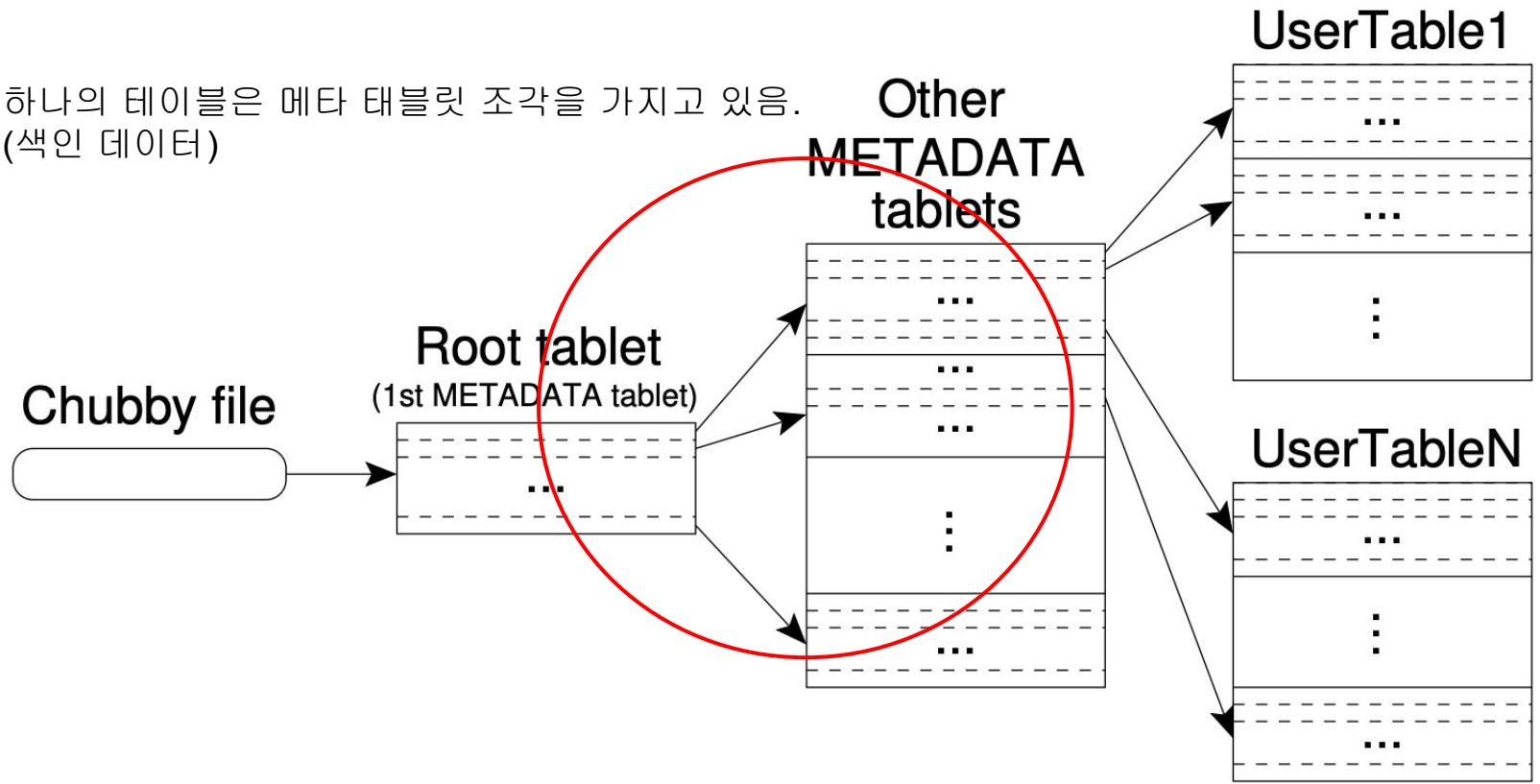


Figure 4: Tablet location hierarchy.

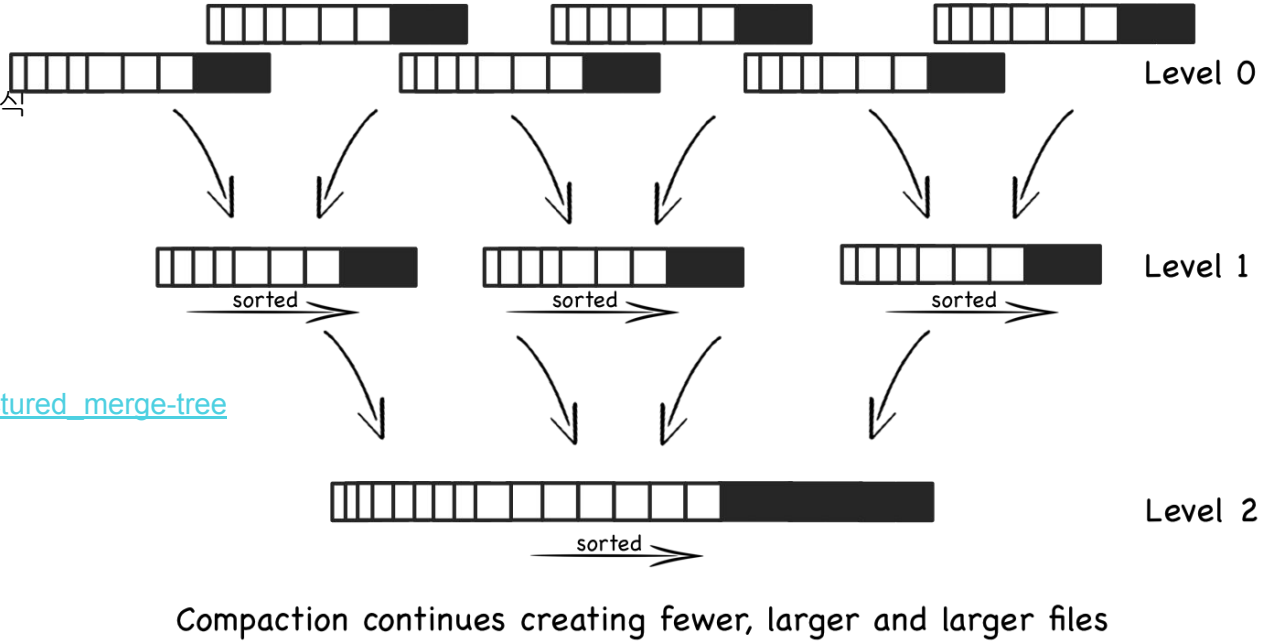
Log Structured Merge (LSM) Tree

로그 구조화 병합 트리

병합 정렬과 같은 방식으로
정렬된 파일 병합과 컴팩션하는 방식

대용량 쓰기 연산에 적합

https://en.wikipedia.org/wiki/Log-structured_merge-tree



DATABASE STORAGE ENGINES

B-TREE

LSM TREE

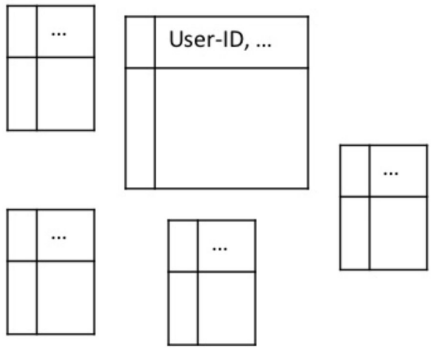
여담 (진지하게 받아들이지 말기)

- 사람들이 뭔지도 모르면서 여기저기 활용해보려고 시도.
- HBase Shell - (한국에 윤진석이) SQL 쿼리 문법 인터페이스 최초 구현
- 사람들, It's not a SQL이라 주장. 이 때문에 프로젝트 중단함
- 그때부터 NoSQL 이라는 용어가 탄생

- 구글이 Sawzall 이라는 쿼리 인터페이스 논문이 발행되자 그제서야
- 짭퉁들: Yahoo Pig, Facebook Hive 쿼리 기반 프로젝트 쏟아지기 시작

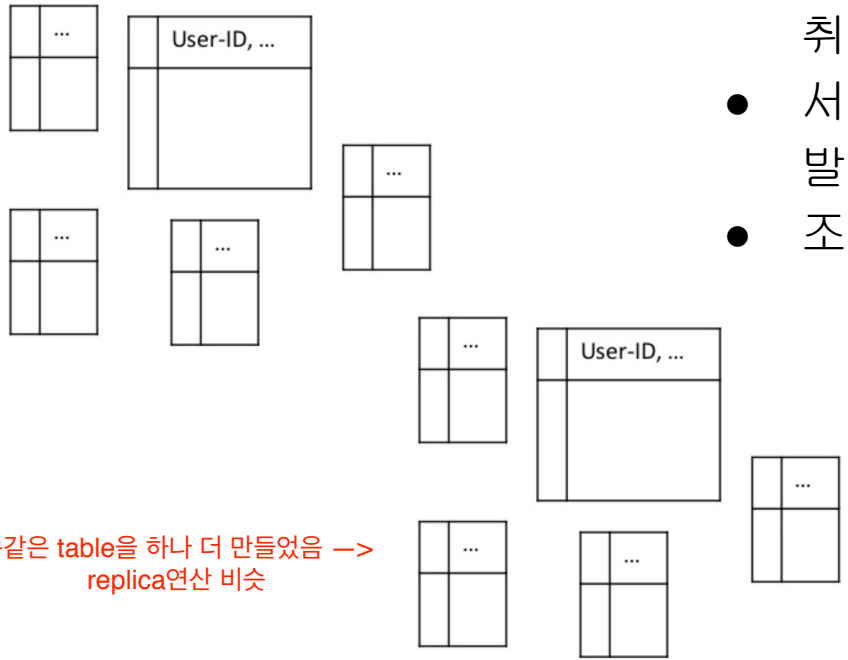
DB 학계에서는 옛날에 우리는 다 해봤던거라고 한 동안 무시
2010년 즈음 하여 VLDB 학회 등 70% 이상의 논문 타이틀에 MapReduce 출현.

관계형 데이터베이스의 문제점



- User profile을 저장하는 테이블이 너무 커지거나, 쓰기연산이 Heavy 한 경우 취할 수 있는 전략?
- 서비스 정책 변경에 따른 Scheme 변화가 발생할 때 해야하는 일들?
- 조인 쿼리가 너무 무겁다면?

관계형 데이터베이스의 문제점



똑같은 table을 하나 더 만들었음 —>
replica연산 비슷

- User profile을 저장하는 테이블이 너무 커지거나, 쓰기연산이 Heavy 한 경우 취할 수 있는 전략?
- 서비스 정책 변경에 따른 Scheme 변화가 발생할 때 해야하는 일들?
- 조인 쿼리가 너무 무겁다면?

일반적으로 수집 partitioning을 하는 이유는
gender table만 따로 빼놓으면 특정 column연산이 집중될 경우 대비
일반적으로 잘 하지 않음 - join 쿼리가 많아짐
새벽 4시부터 6시까지 임시점검중 - 보통 DB 작업 때문
어느 한 군데가 안 고쳐져 있으면 예상치 못한 영향
—> 코드를 변경, DB 수정작업
요즘은 안함

NoSQL

- Scheme-free
- Column-oriented
  - Sparse tables, attribute-focused
- LSTM tree + Distributed environments
- Fault-tolerant

100만, 200만까지는 RDBMS로 사용 가능 하지만 그 이상으로 넘어가면 너무 느려서 안 됨
schema를 정의하지 않고 사용
table을 처음 만들 때 column이 정해져 있지 않다
row도 무한히 증가할 수 있고 column도 무한히 증가할 수 있다.
구글 빅테이블 구조 안에서 나온 것
column-oriented —> sparse한 table을 저장하는 데에 유리
예를 들면 서비스를 하면서 거래내역에 대한 정보를 어딘가에 저장해야 되고 한다면
어떤 storage solution을 선택할 것인가에 대한 issue
NoSQL과 RDBMS간 장점이 다름 <— 스타일의 차이
이걸 명확히 알고 사용하는 것이 좋음

Coffee break ...

NoSQL open source projects

- Wide-column Store
  - HBase
    - Powerset 에서 개발 MS에 인수됨
  - Cassandra
    - Facebook 에서 개발되고, 장애로 팀 전체 해고
- Document Store
  - CouchDB
    - 오픈소스 개발자들이 개발, 웹 개발, Javascript와 연동 쉬움
  - MongdoDB
    - NoSQL 중에 인기가 좋음, 웹 개발, 특히 Javascript와 연동 쉬움

Golden age of Data Science

The Golden Age of Data Science

이미지 인식, 자연어 처리, 정보 검색의 비약적 발전

현생 인류의 출현 약 20만년 전

→ 문자의 출현 (정보 기록) 기원전 약 3,200년 경

→ Web, full-text search의 발전 (정보 검색) 약 30년 전

→ 빅 데이터 붐 약 10년 전

→ **Rise of AI (인공지능) 약 6년 전**

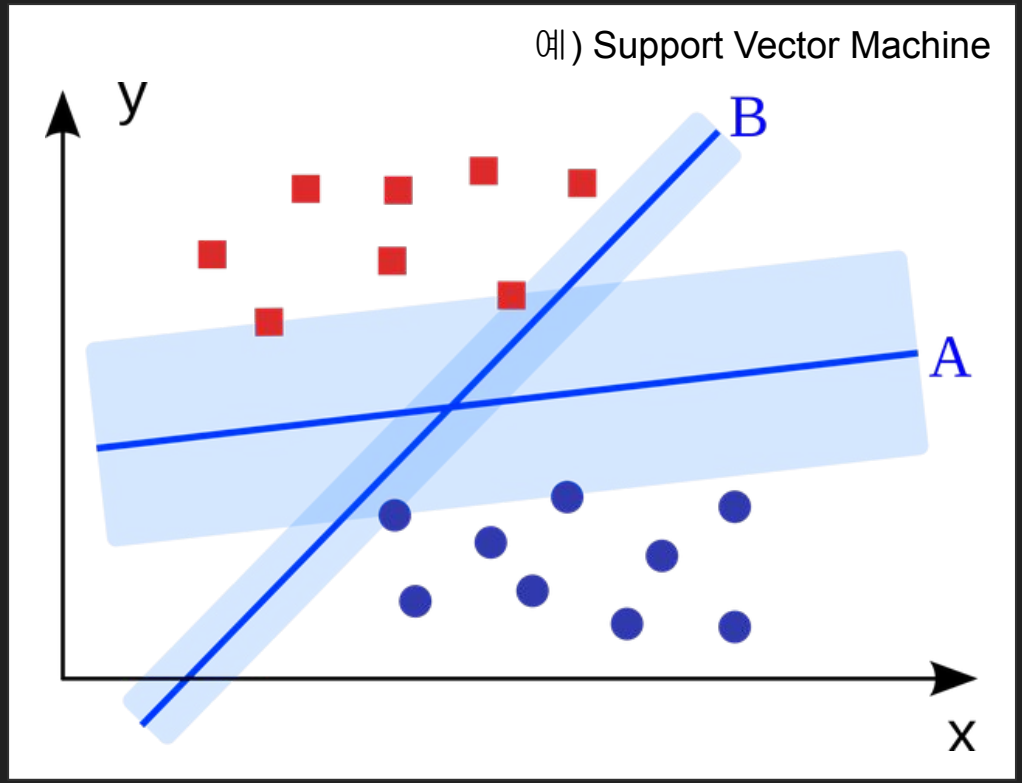
Rise of AI

빅 데이터와 컴퓨팅 파워, 알고리즘의 발전 배경에 기인

2012

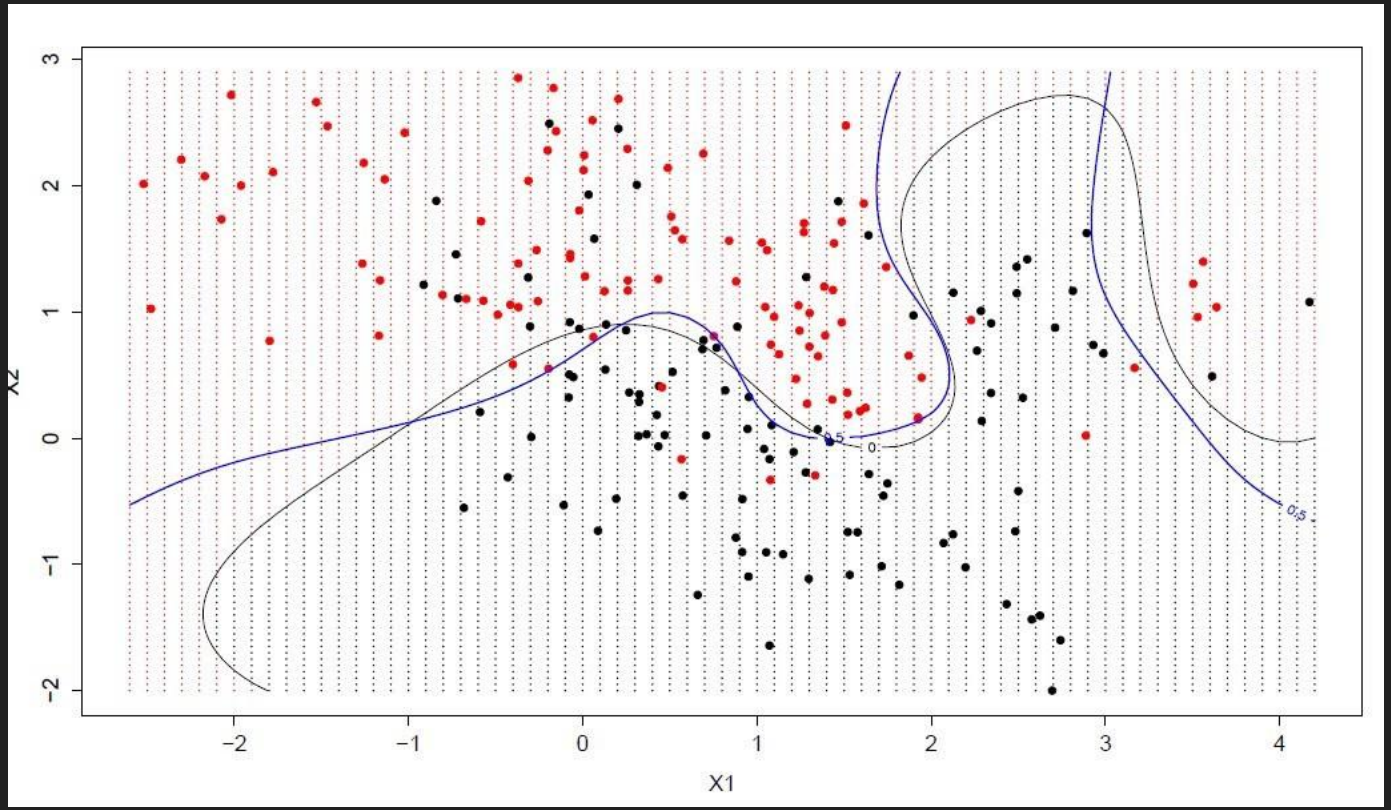
Youtube 동영상 내 고양이 출현 영상 분류 성공 = **Geoffrey Hinton** (뉴럴넷 대가) **+ Jeff Dean** (인프라 SW 대가) **+ Google** (보유 데이터)

기계학습 = Find the optimal hyperplane

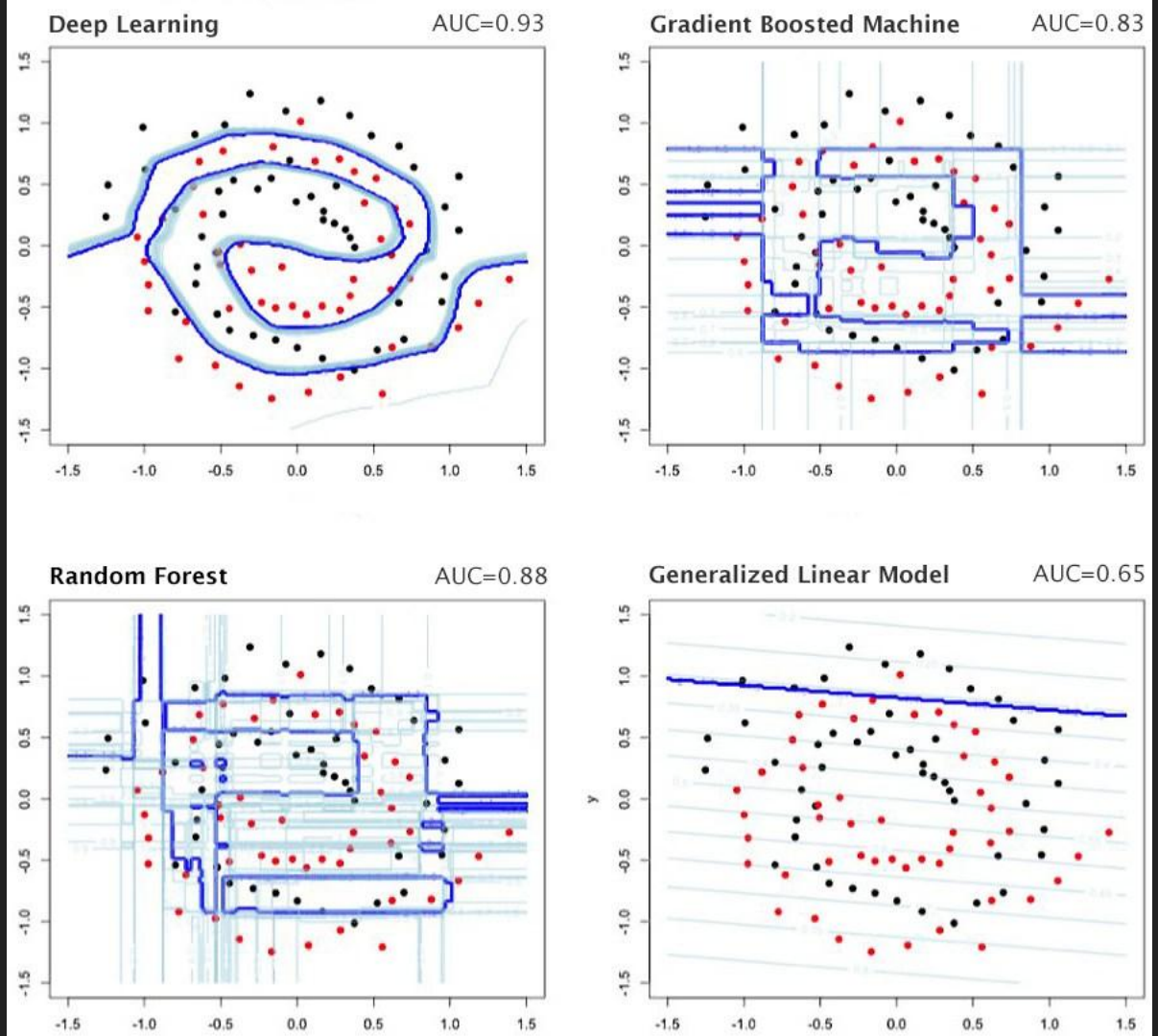


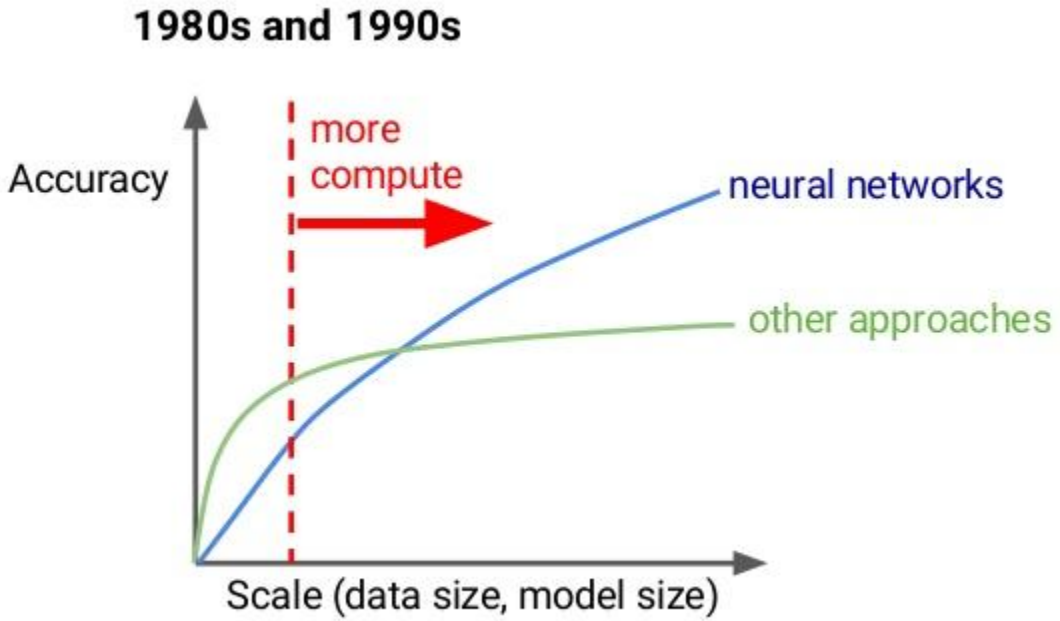
Machine learning algorithms are described as learning a target function (f) that best maps input variables (X) to an output variable (Y): Y = f(X)

락 장르 음악에 최적화된 세팅을 찾으시오

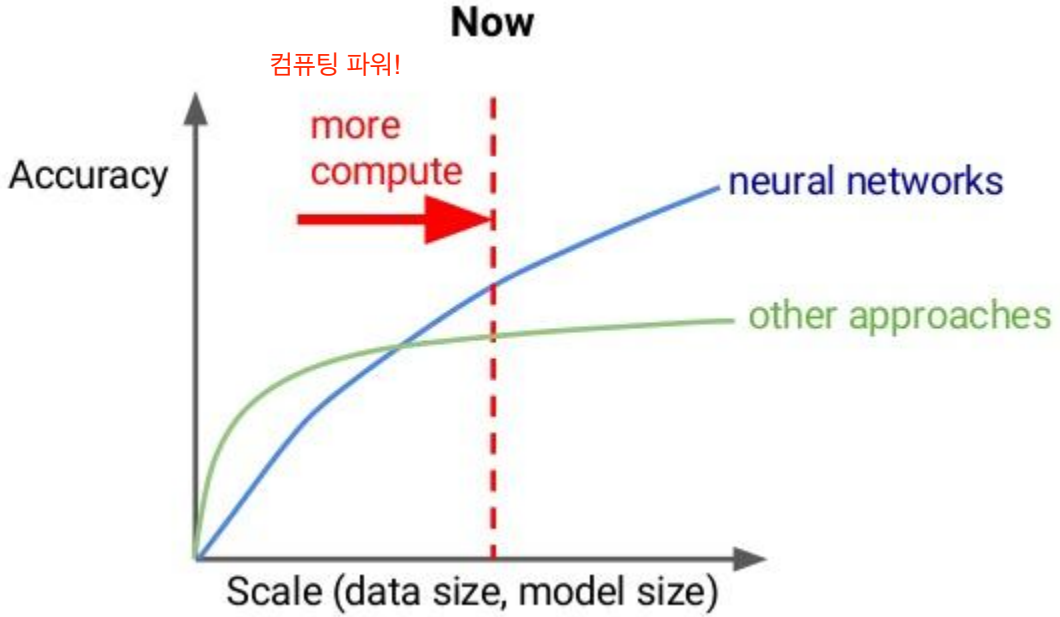


But, nearly all natural problems require nonlinearity

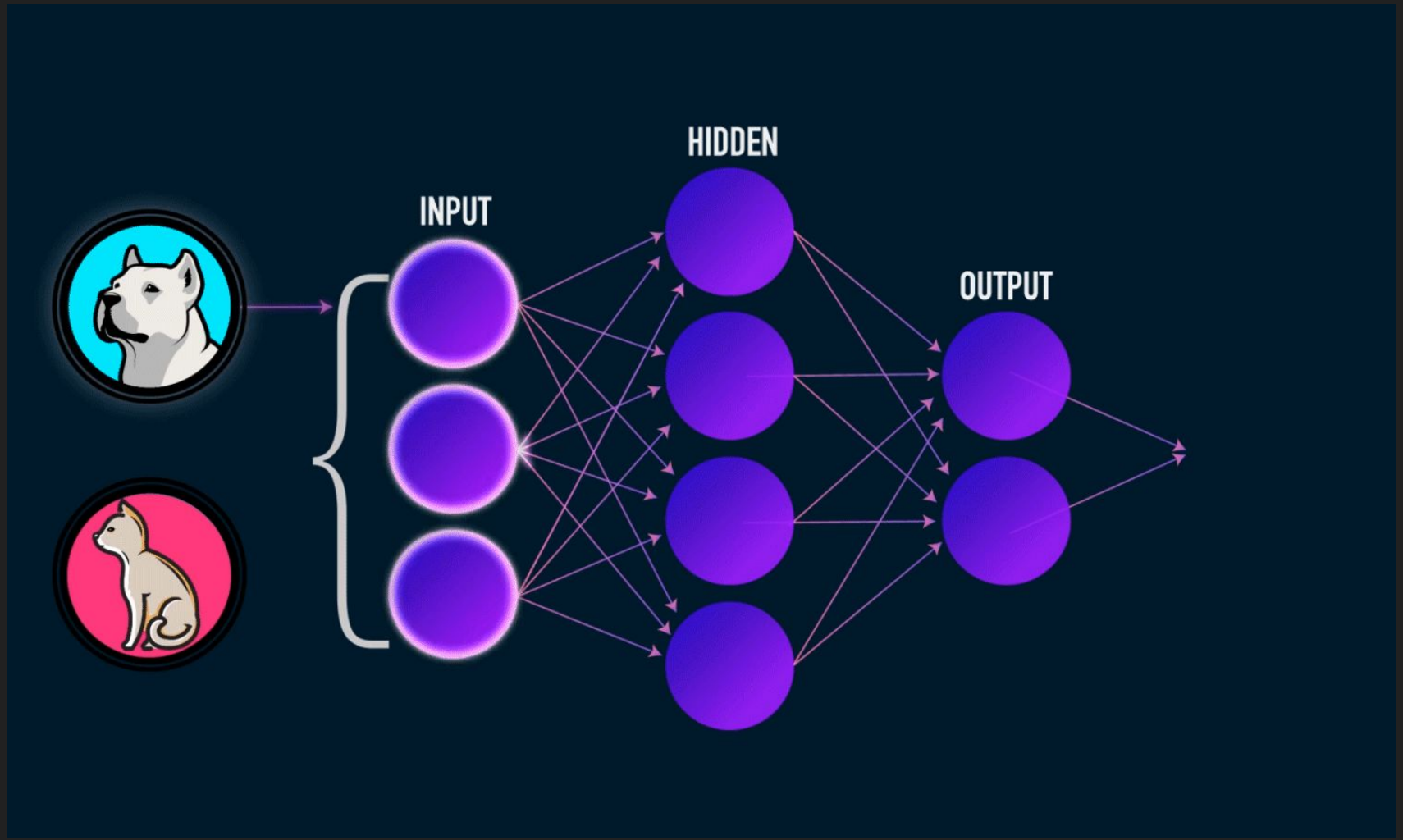

Deep Learning
AUC=0.93
Gradient Boosted Machine
AUC=0.83
Random Forest
AUC=0.88
Generalized Linear Model
AUC=0.65
y



과거 스케일에서는 다른 모델이 우수한 성능을 보여줬다면,



현재의 스케일에서는 딥 뉴럴 네트워크가 가장 우수하다.


HIDDEN
INPUT
OUTPUT

정보과학 기술 동향 - 웹의 진화

Web 전자문서 출현 → 개인 홈피
Web 2.0 개념 탄생 → 블로그 스피어, 집단지성, 소셜네트워크
Mobile 패러다임 → 1:1 또는 1:n 메신저와 SNS 폭발 성장
문서에서 Media로 → 유튜브, 아프리카TV, 스푼 라디오, 트위치, …

정보과학 기술 동향 - 기술 진화

Text → Multimedia

Text Mining
Full-text Search

Image Search
Collaborative Filtering

Vision, Voice recognition

Batch → Streaming

Batch computing

Mini-batch computing

DAG,
Computational graph

Sync → Async

Sequential code
compiler

Event-driven,
Asynchronous

NoSQL store는 text만 저장하나?

- Google의 BigTable은 Maps, Youtube, Gmail 에다 활용 중
- 즉, bytes array로 바이너리 파일도 저장한다는 뜻
- 기계학습 알고리즘에 NoSQL의 활용이 가져오는 이점은?
  - 중복적 pre-processing 을 최소화 할 수 있음
  - 이에 따라, intermediate temporary space 사용도 같이 줄일 수 있음