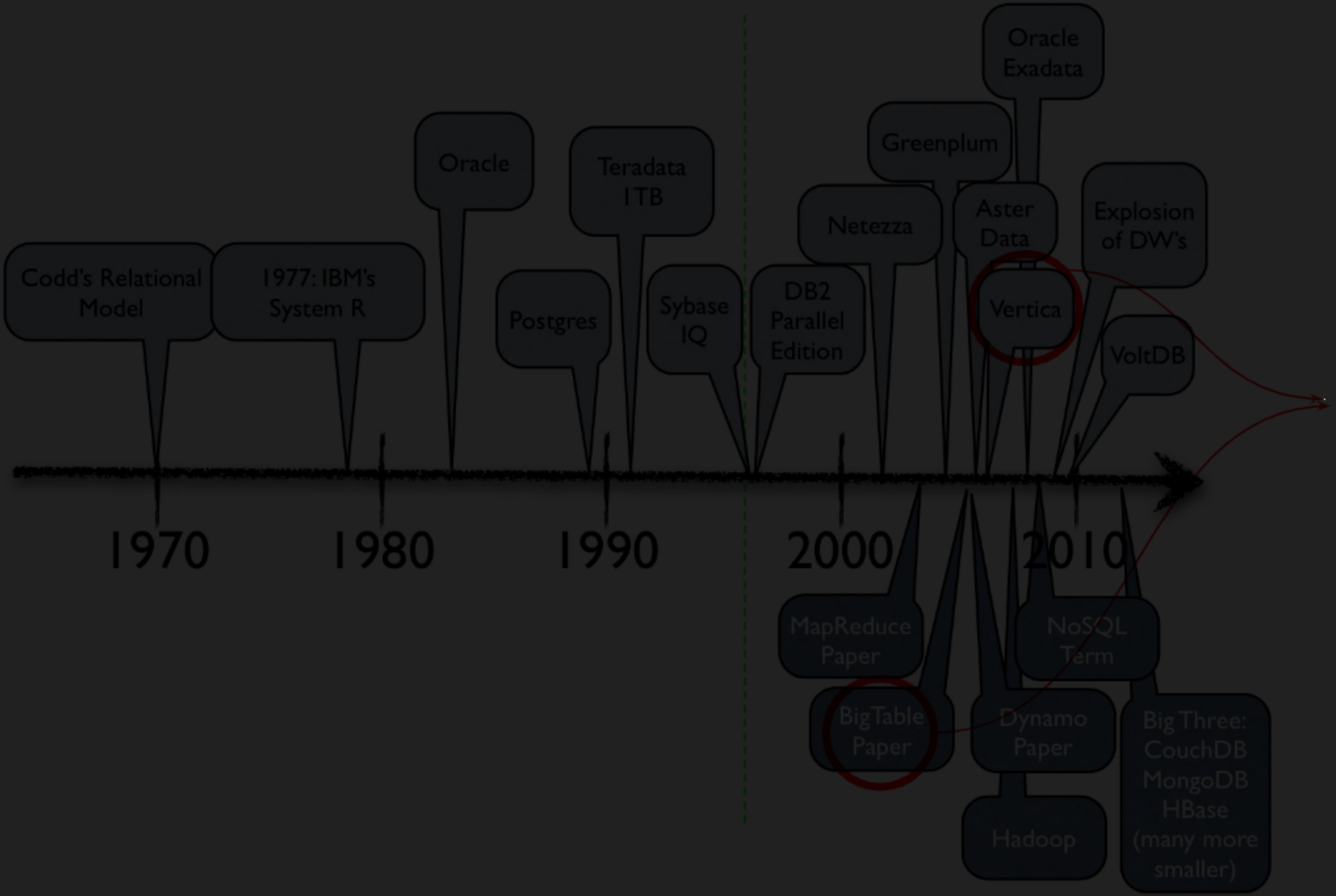
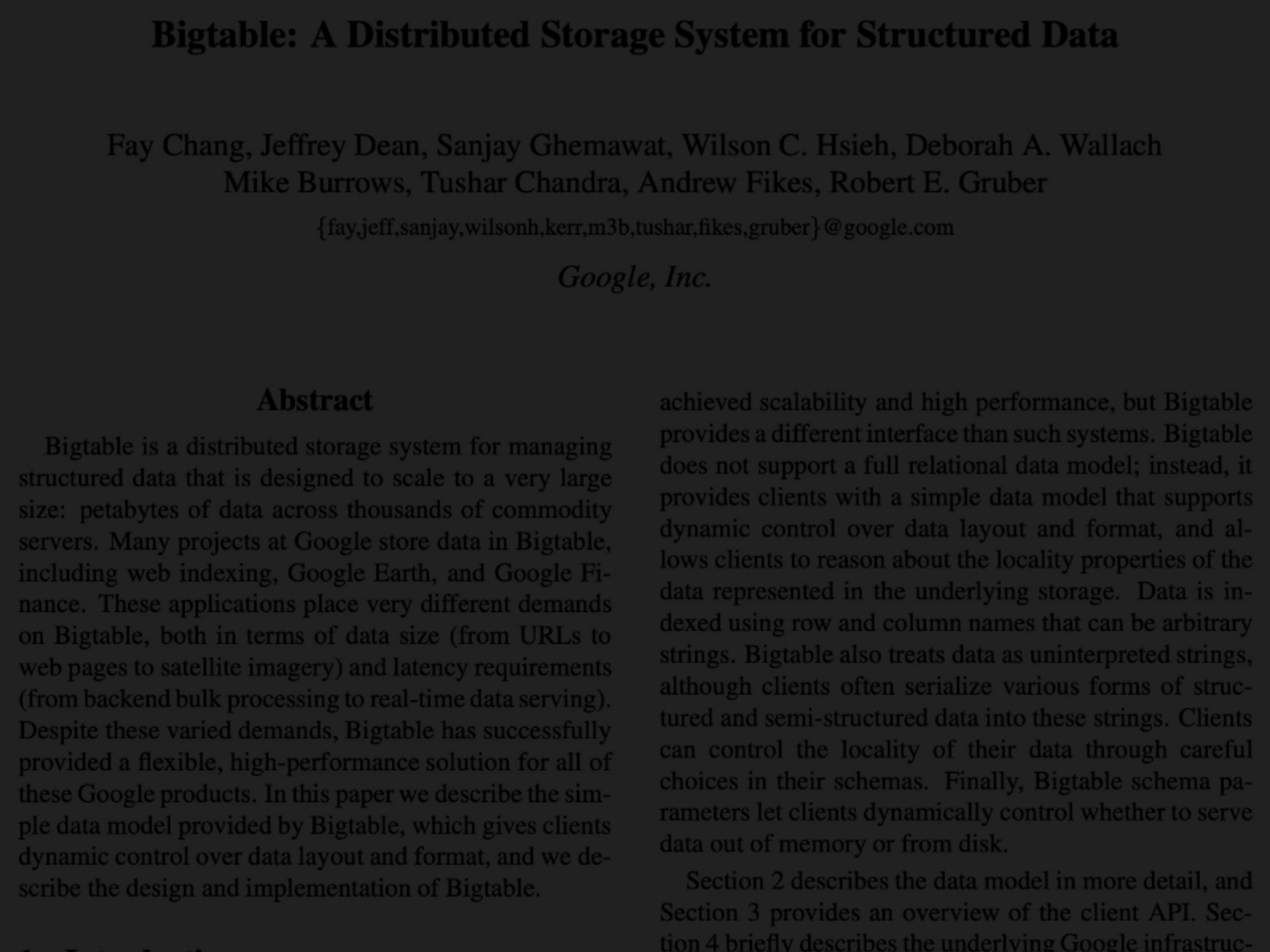
[고려대학교 특강] Edward J. Yoon

<edwardyoon@apache.org>


Codd's Relational Model
1977: IBM's System R
Oracle
Postgres
Teradata 1TB
Sybase IQ
DB2 Parallel Edition
Netezza
Greenplum
Oracle Exadata
Aster Data
Vertica
Explosion of DW's
VoltDB
1970
1980
1990
2000
2010
MapReduce Paper
BigTable Paper
Dynamo Paper
Hadoop
NoSQL Term
Big Three: CouchDB MongoDB HBase (many more smaller)



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

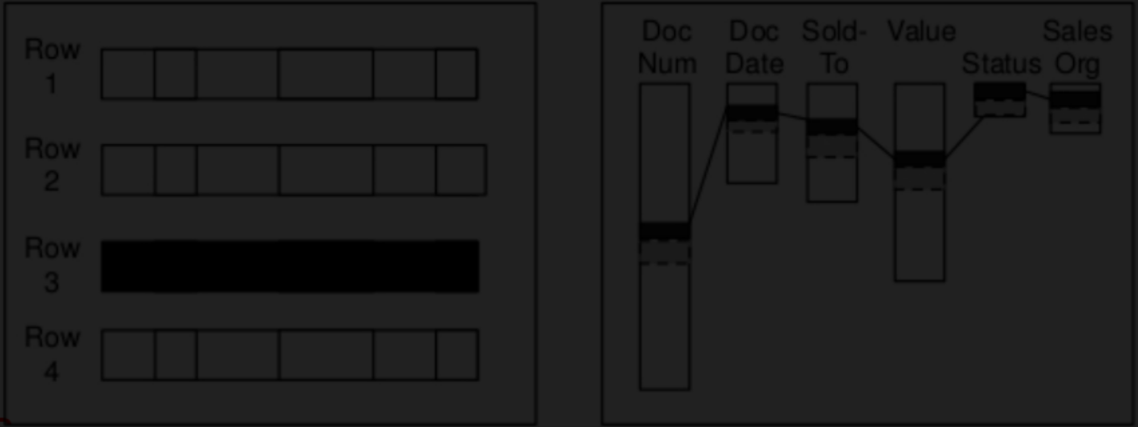
Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place 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.

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-

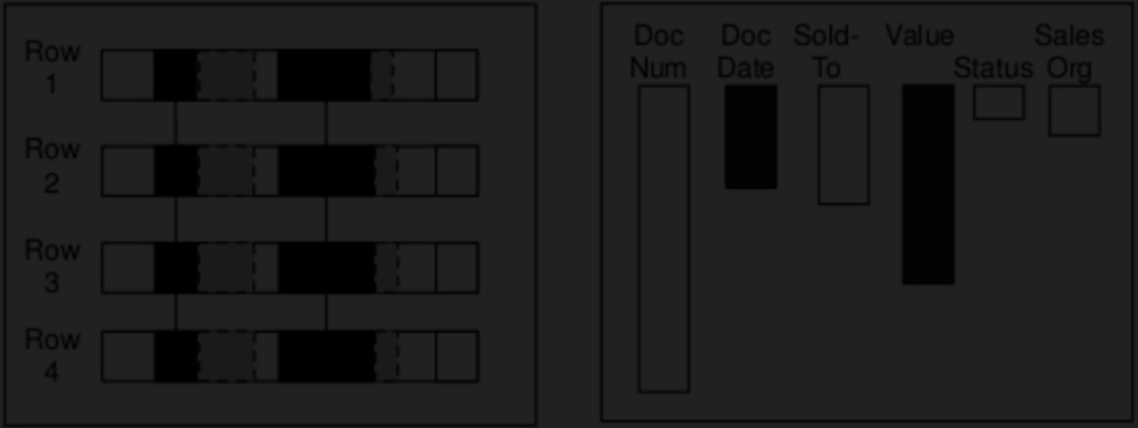
Optimal for row-wise access (e.g., SELECT *)

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

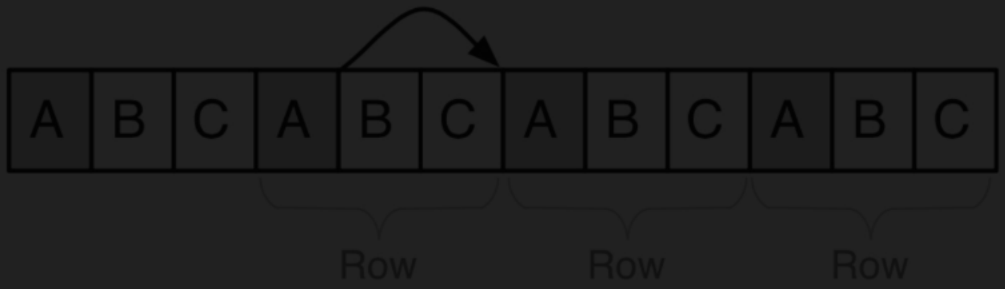


Optimal for attribute focused access (e.g., SUM, GROUP BY)

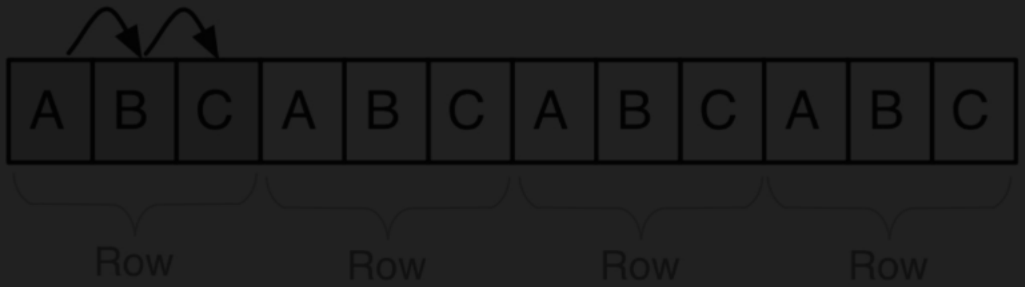
SELECT SUM(Value)
FROM Sales Orders
WHERE Document Date > 2011-08-28
(OLAP-style query)



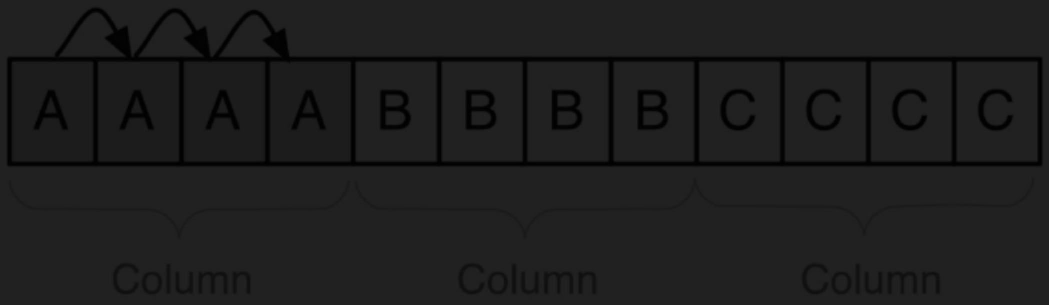
Column Operation



Row Operation



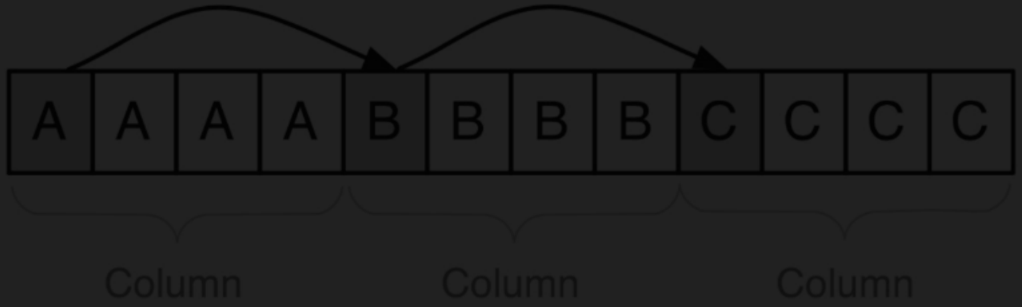
Column Operation



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

Stride access ~256 secs

Row Store – Layout

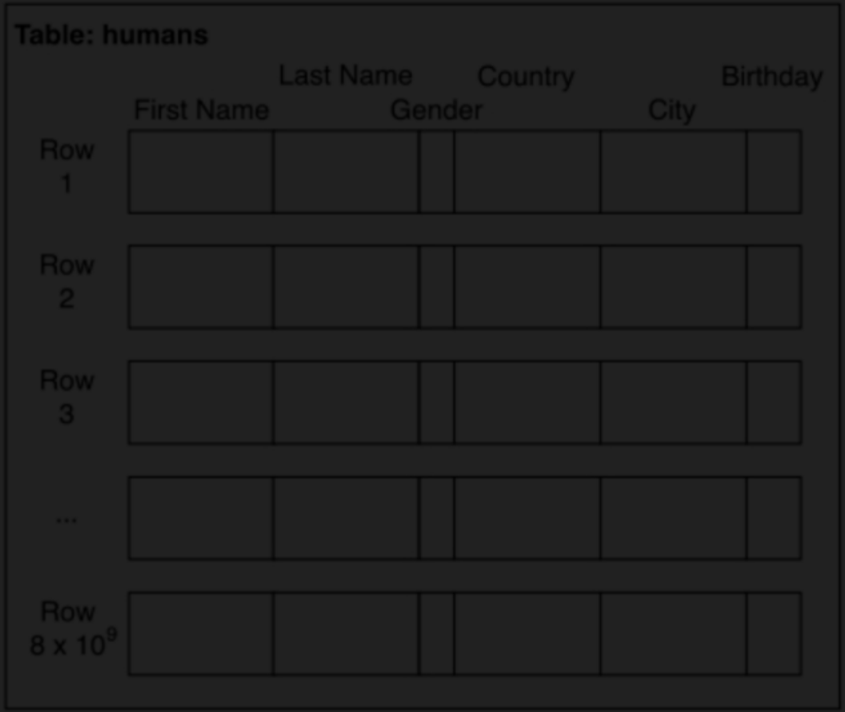


Table: humans

Last Name Country Birthday

First Name Gender City

Row 1

Row 2

Row 3

...

Row 8×10^9

Row Store – Full Table Scan

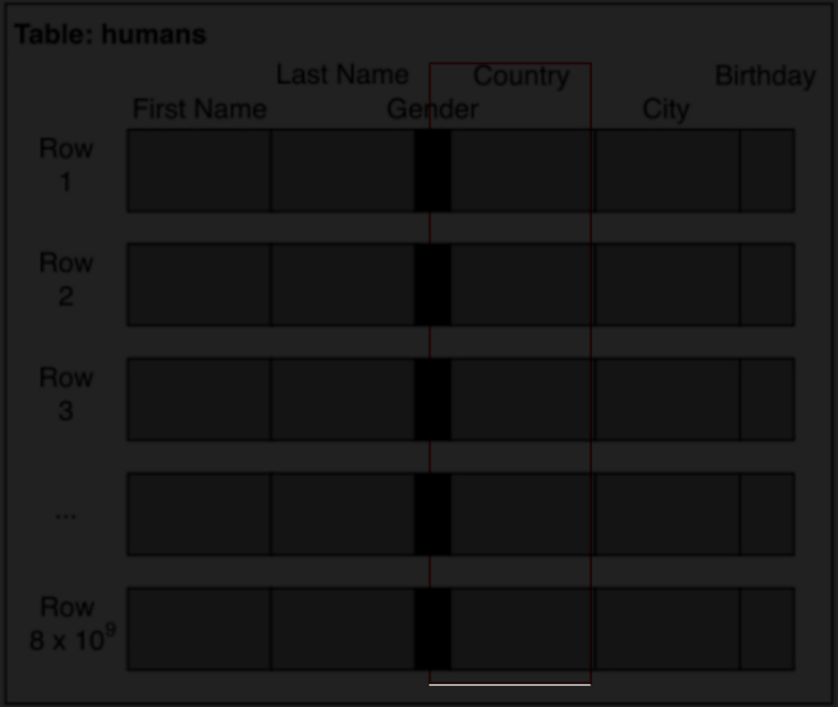


Table: humans

Last Name Country Birthday

First Name Gender City

Row 1

Row 2

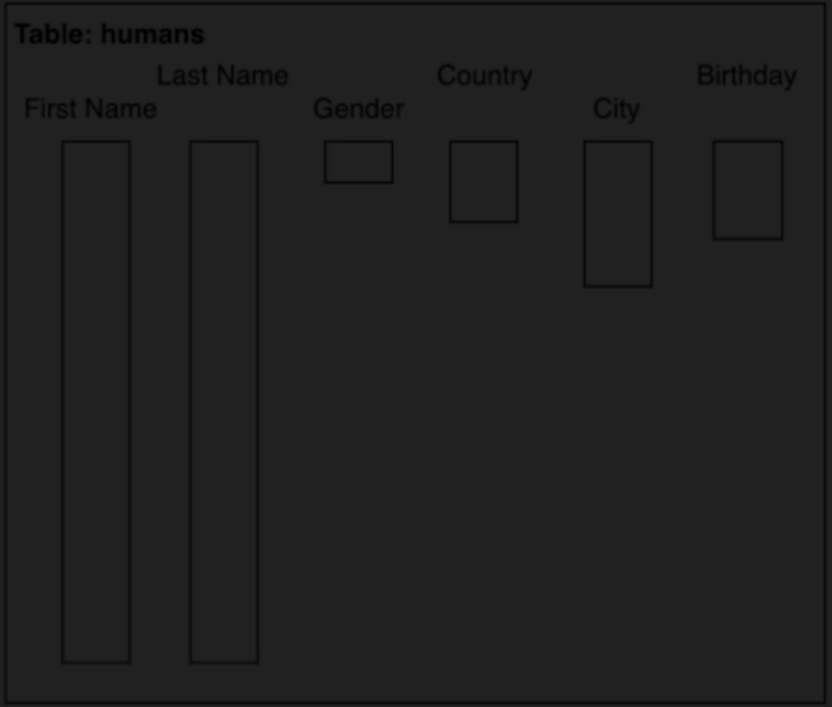
Row 3

...

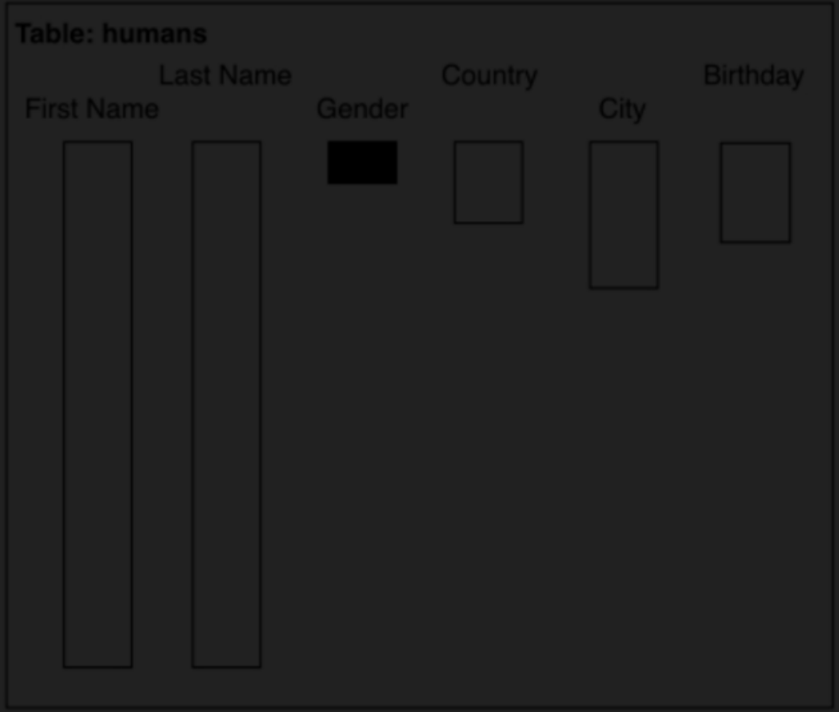
Row 8×10^9

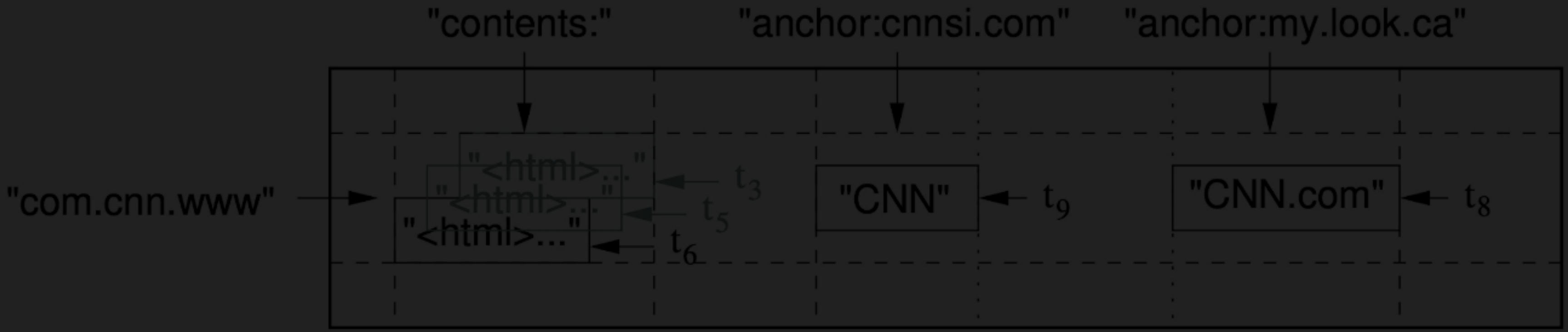
~0.5 secs

Column Store – Layout



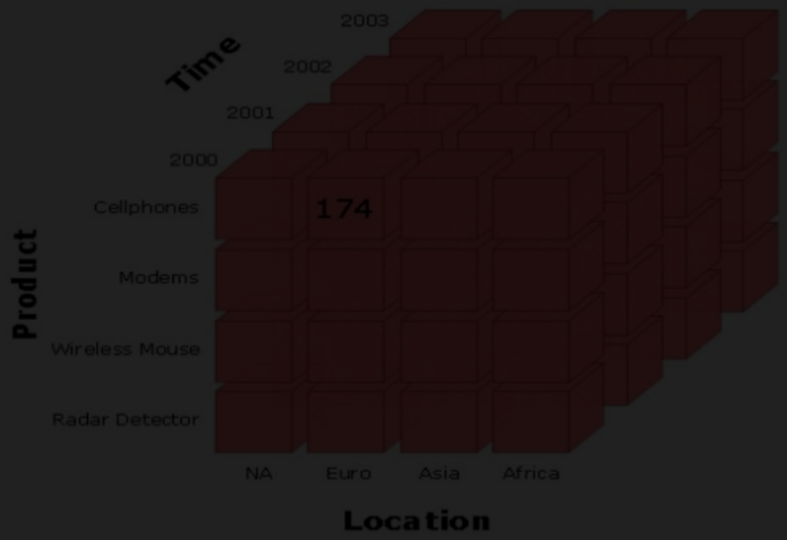
Column Store – Full Column



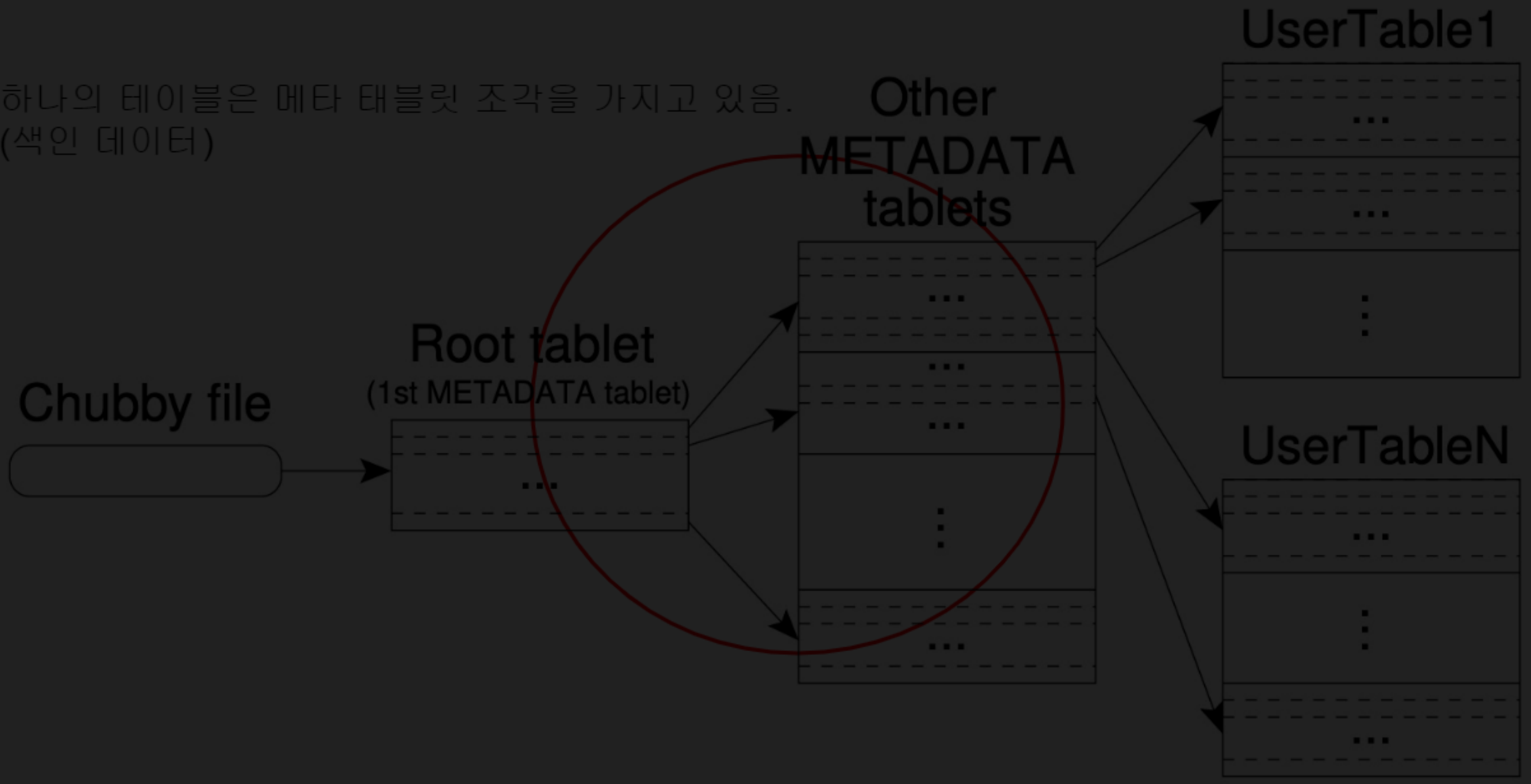

"contents:"
"anchor:cnnsi.com"
"anchor:my.look.ca"
"com.cnn.www"
"<html>..."
"<html>..."
"<html>..."
t3
t5
t6
"CNN"
t9
"CNN.com"
t8

예시)

Dimensions And Measures



웹 네트워크 구조의 진화 과정을



하나의 테이블은 메타 태블릿 조각을 가지고 있음.
(색인 데이터)

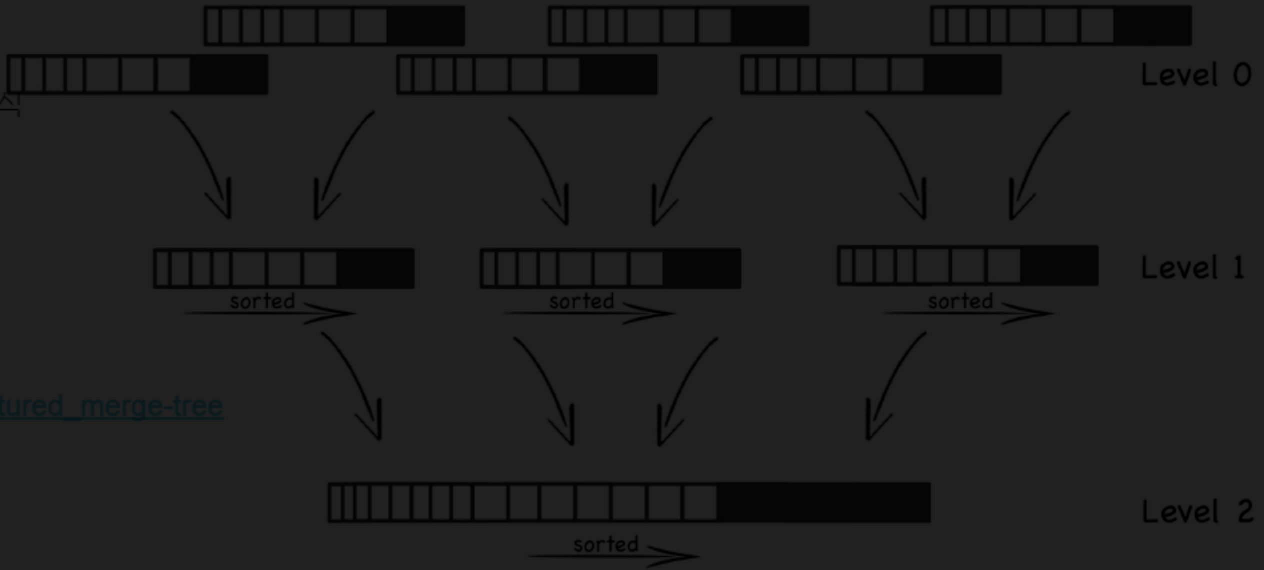
Figure 4: Tablet location hierarchy.

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



Compaction continues creating fewer, larger and larger files

DATABASE STORAGE ENGINES

B-TREE

ORACLE

Microsoft SQL Server

IBM DB2

PostgreSQL

MySQL

mongoDB

Couchbase

LSM TREE

cassandra

APACHE HBASE

Google Cloud Bigtable

elasticsearch

influxdb

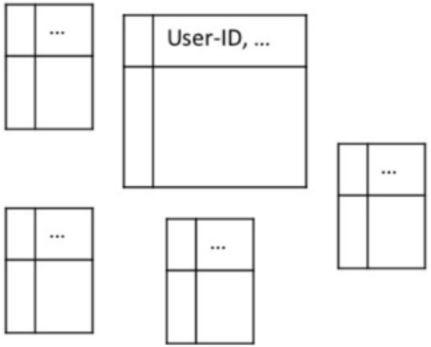
LEVELDB

RocksDB

YugaByte DB

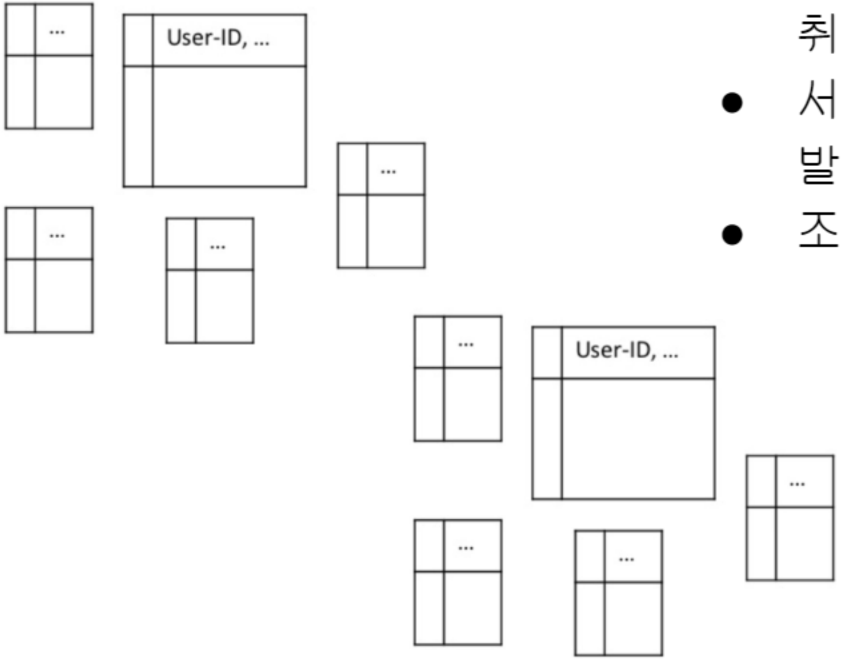
DB 학계에서는 옛날에 우리는 다 해봤던거라고 한 동안 무시
2010년 즈음 하여 VLDB 학회 등 70% 이상의 논문 타이틀에 MapReduce 출현.

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



- 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?

인류

문자

Web, full-text search

빅 데이터

Rise of AI (인공지능) 약 6년 전

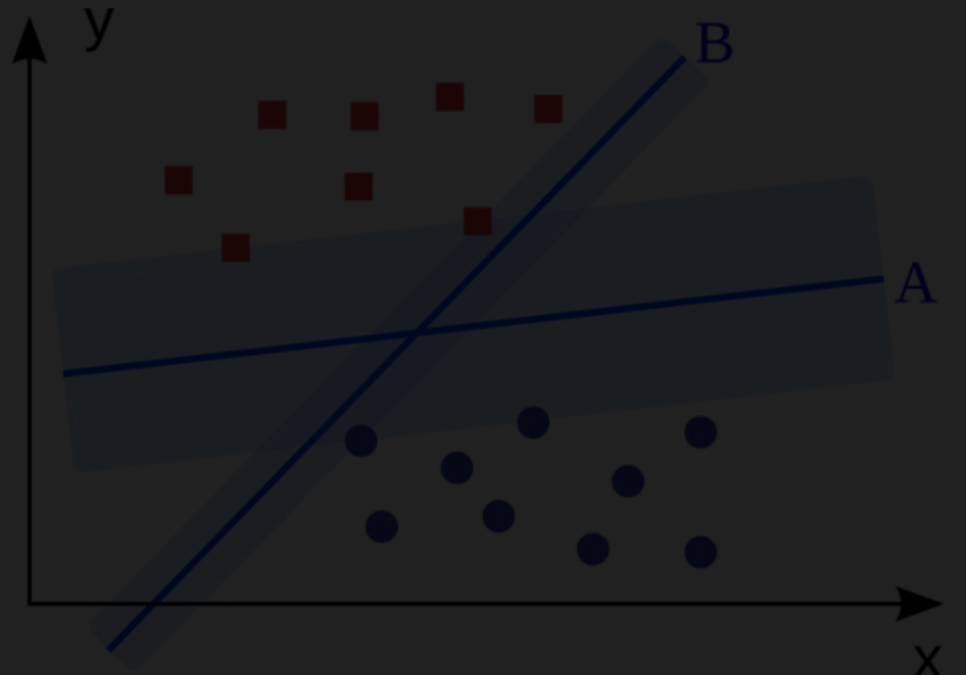
빅 데이터   컴퓨팅 파워, 알고리즘

Geoffrey Hinton

+ Jeff Dean

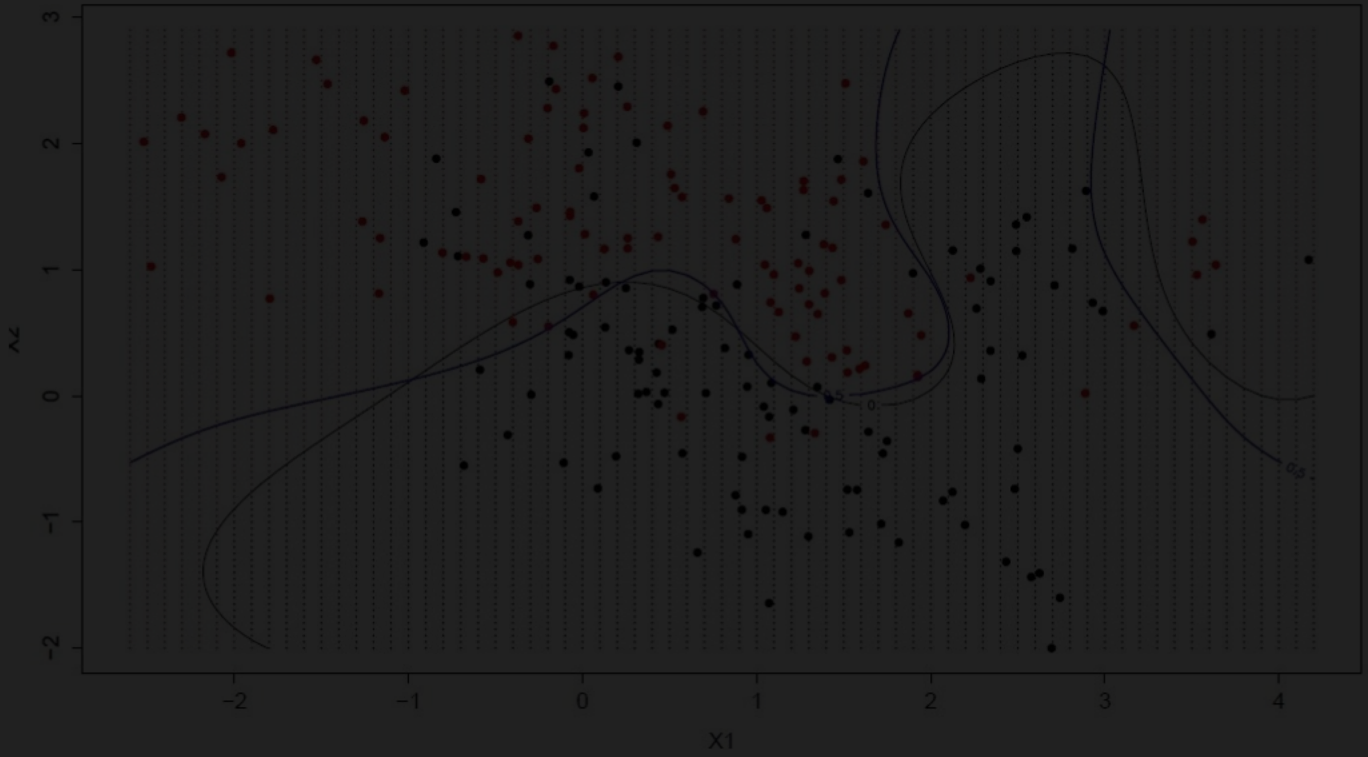
+ Google

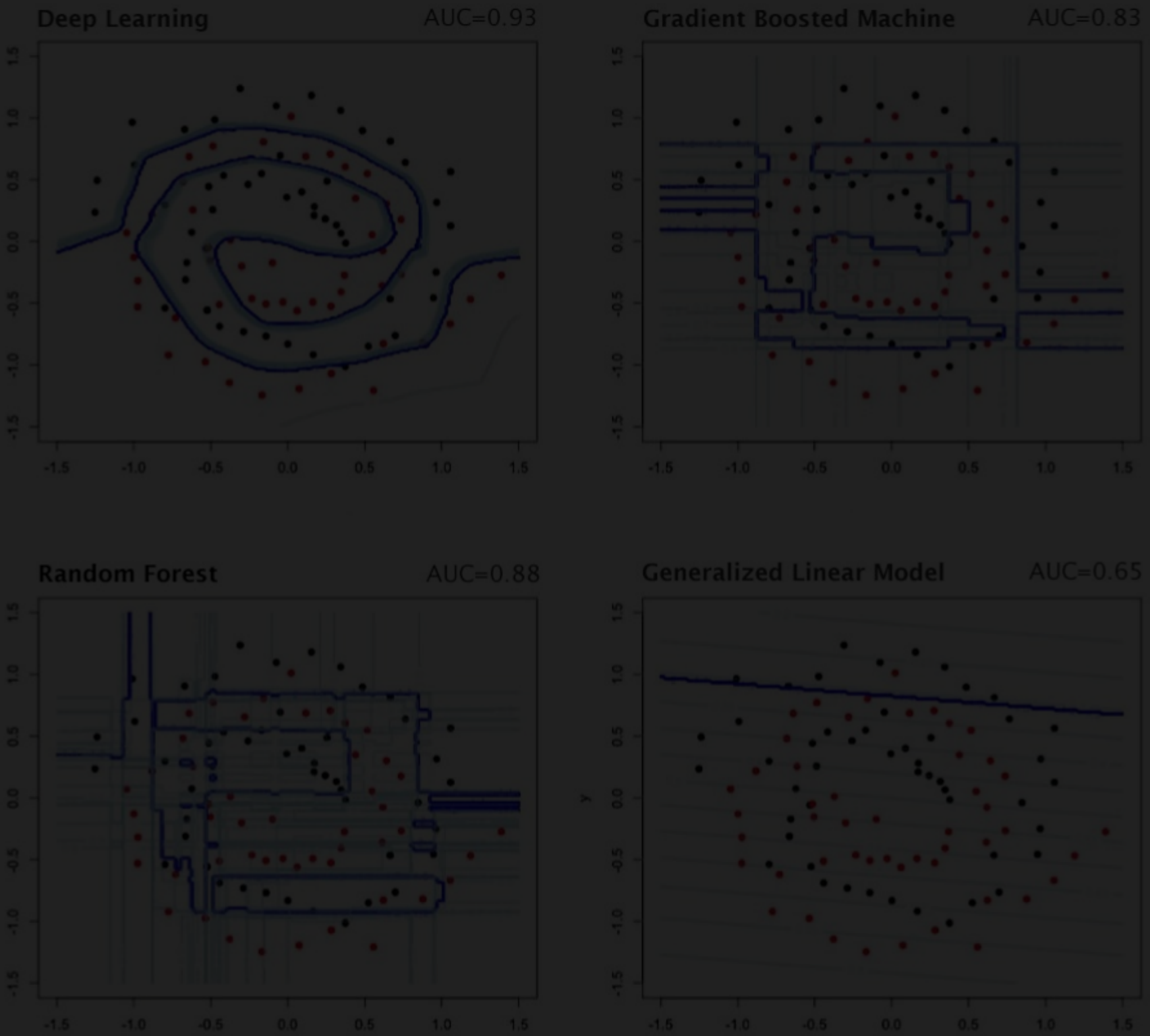
예) Support Vector Machine



락 장르

O/LOAD
12
OUT
6
IN
OUT
0dB
SCALE
30 Hz
LEVEL
A
O/LOAD
12
OUT
6
IN
OUT
0dB
SCALE
30 Hz
B


X1
X2
−2
−1
0
1
2
3
4
−2
−1
0
1
2
3
0.5
0.

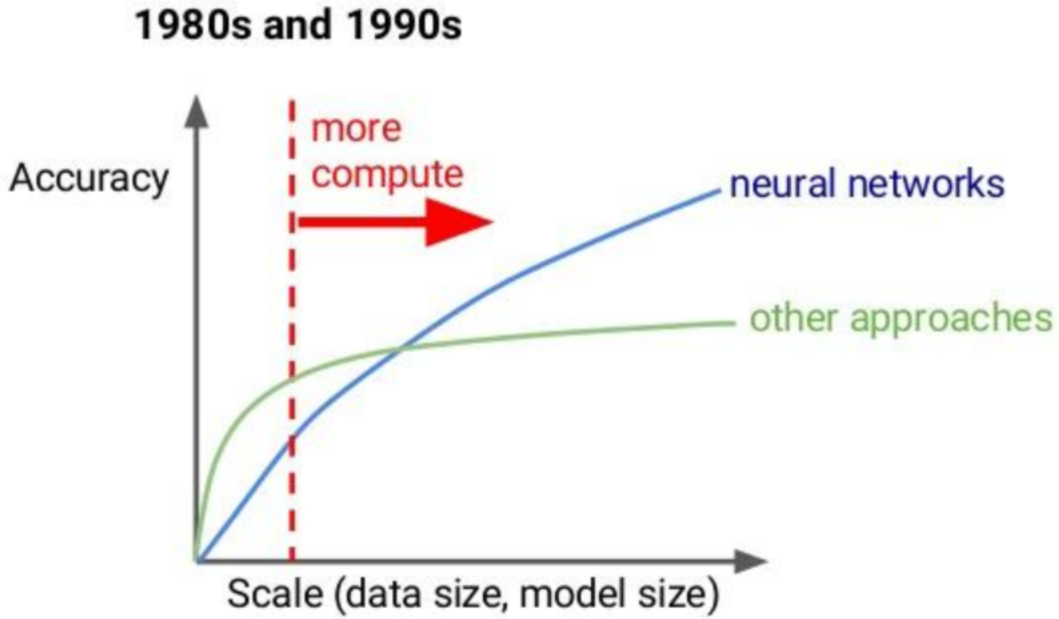


Deep Learning AUC=0.93

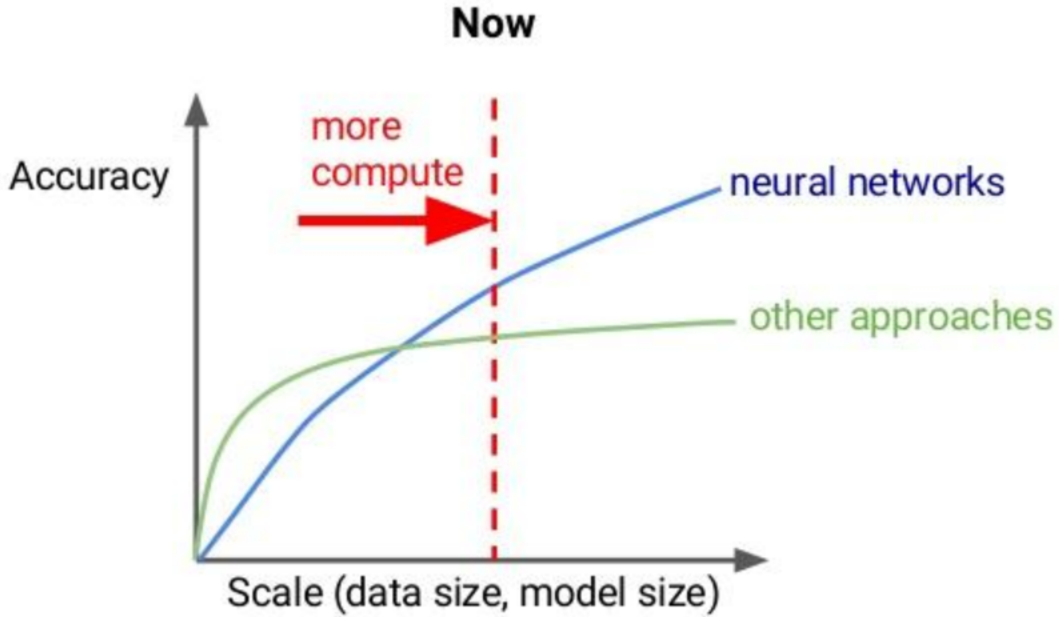
Gradient Boosted Machine AUC=0.83

Random Forest AUC=0.88

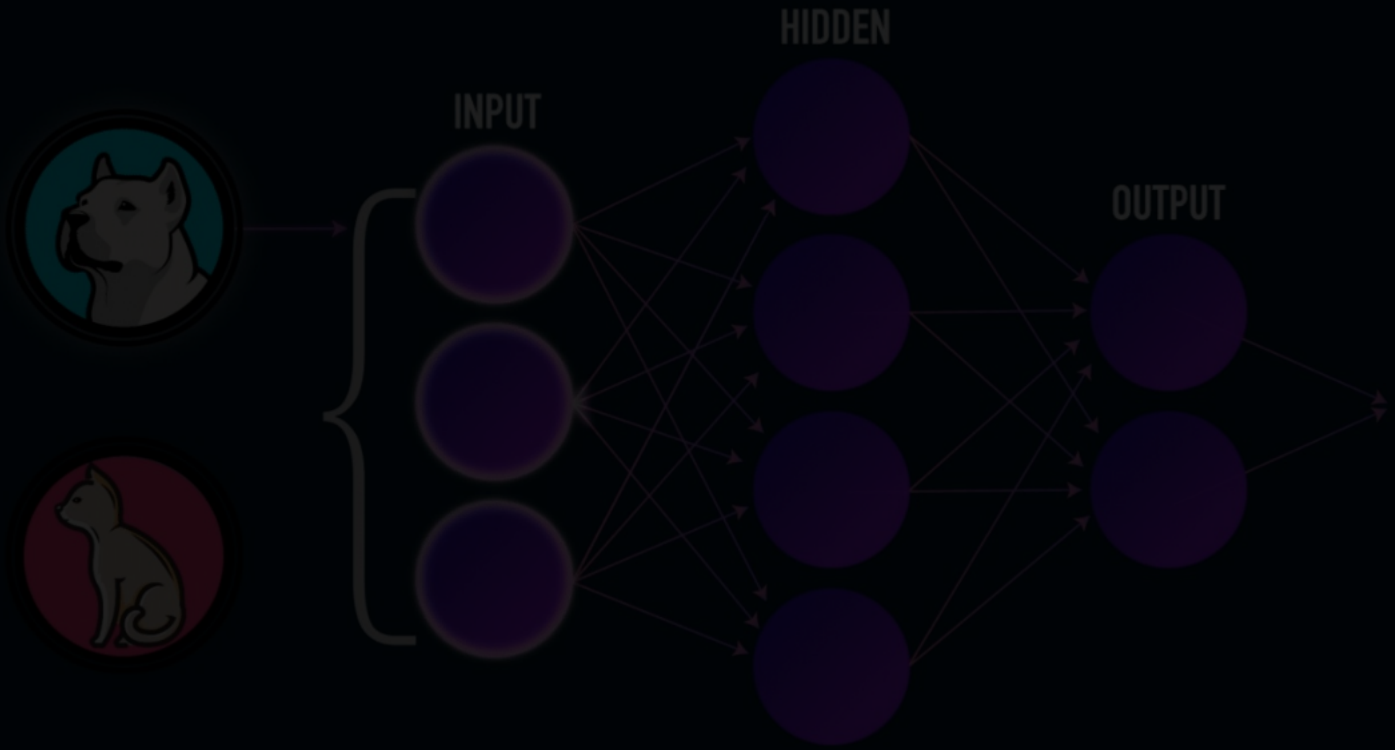
Generalized Linear Model AUC=0.65



과거 스케일에서는 다른 모델이 우수한 성능을 보여줬다면,



현재의 스케일에서는 딥 뉴럴 네트워크가 가장 우수하다.


HIDDEN
INPUT
OUTPUT