

# 数据库系统原理与设计

## (第2版)

### 志存高远      脚踏实地

- 树立一个长期（十年，二十年）奋斗目标
- 制定短期的符合SMART的学习、工作计划
  - 必须是具体的 (Specific)
  - 可度量的 (Measurable)
  - 可以行动的 (Actionable)
  - 合乎情理可以到达的 (Realistic)
  - 要能在一定时间内完成的 (Time-Bound)
- 经常回顾进度，并进行适当修正

# 数据库系统原理与设计

## (第2版)

### 第12章 数据管理技术前沿

# 本章内容

- 12.1 大数据的兴起
  - 大数据的概念、从数据库到大数据、大数据处理模式
- 12.2 大数据处理平台和框架
  - Apache Hadoop、Spark、Storm
- 12.3 数据库面临的挑战
  - 数据库架构变化、数据库可扩展性问题的解决方法、数据库的发展
- 12.4 NoSQL数据库
  - 键值数据库Redis、文档数据库MongoDB、列数据库HBase、图数据库Neo4j
- 12.5 NewSQL数据库
  - VoltDB、NuoDB

# 12.1 大数据的兴起

- 大数据的概念
  - 3V定义，即认为大数据需满足3个特点
    - **Volume(容量大)**:
      - 容量大，达到PB级别
    - **Variety(多样性)**:
      - 数据类型繁多，半结构化、非结构化数据在飞速增长
    - **Velocity(速度)**:
      - 数据产生和更新的频率高，
      - 数据(特别是流数据)处理的速度要求高
    - 其他的V:
      - **Veracity(真实性)**: 数据的真实性和质量才是获得真知和思路最重要的因素
      - **Value(价值)**: 指大数据中蕴含着巨大的价值

- 大数据的来源：

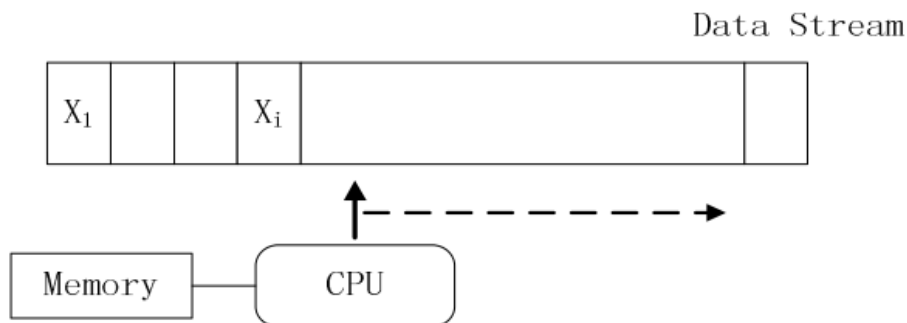
- 互联网大数据尤其社交媒体是近年来大数据的主要来源之一，大数据技术主要源于快速发展的国际互联网企业
- 生物工程、遥感测绘、物理等领域的实验数据
- 越来越多的机器配备了连续测量和报告运行情况的装置，这些机器传感数据也属于大数据的范围
- 企业中的生产、销售数据等，以及非结构化的数据，如电子邮件、文本、音视频、图片，以及传感器数据等
- 随着**5G**的部署

# 从数据库到大数据

- 从数据库到大数据：“池塘捕鱼” → “大海捕鱼”
  - 数据规模：池塘——海洋
  - 数据类型：池塘的鱼——海洋中的鱼
  - 模式(schema)和数据的关系：
    - 先有池塘，后有水
    - 海洋是自然形成的
  - 处理对象：
    - 池塘中的鱼仅仅是处理对象
    - 海洋中的鱼不仅是处理对象，也是资源
  - 处理工具：
    - “池塘捕鱼”：One Size Fits All
    - “大海捞鱼”：No Size Fits All

# 大数据处理模式

- **流处理**：流处理的处理模式将数据视为流，源源不断的数据组成了数据流。当新的数据到来时就立刻处理并返回所需的结果



- 由于响应时间的要求，流处理的过程基本在内存中完成，其处理方式更多地依赖于在内存中设计巧妙的概要数据结构(synopsis data structure)，内存容量是限制流处理模型的一个主要瓶颈。

## • 批处理：

- 批处理主要是对大容量静态数据集进行处理，并在计算过程完成后返回结果。
- 批处理模式中使用的数据集通常符合下列特征：
  - 有界：批处理数据集代表数据的有限集合
  - 持久：数据通常始终存储在某种类型的持久存储设备中
  - 大量：批处理操作通常是处理极为海量数据集的唯一方法
- 大量数据的处理需要付出大量时间，因此批处理不适合对处理时间要求较高的场合。



# 批处理的代表——MapReduce

- MapReduce模型的基本处理过程包括：

**MapReduce**擅长处理大数据，它为什么具有这种能力呢？这可由**MapReduce**的设计思想发觉。**MapReduce**的思想就是“分而治之”。

（1）**Mapper**负责“分”，即把复杂的任务分解为若干个“简单的任务”来处理。“简单的任务”包含三层含义：一是数据或计算的规模相对原任务要大大缩小；二是就近计算原则，即任务会分配到存放着所需数据的节点上进行计算；三是这些小任务可以并行计算，彼此间几乎没有依赖关系。

（2）**Reducer**负责对**map**阶段的结果进行汇总。至于需要多少个**Reducer**，用户可以根据具体问题，通过在**mapred-site.xml**配置文件里设置参数**mapred.reduce.tasks**的值，缺省值为1。

# 批处理的代表——MapReduce

- MapReduce模型的基本处理过程包括：
  - (1) 从Hadoop分布式文件系统(Hadoop distributed file system, HDFS)读取数据集；
  - (2) 将数据集拆分成小块并分配给所有可用结点；
  - (3) 针对每个结点上的数据子集进行计算(这一过程成为Map，计算的中间态结果会重新写入HDFS)；
  - (4) 重新分配中间态结果并按照键进行分组；
  - (5) 通过对每个结点计算的结果进行汇总和组合，实现对每个键的值进行“Reducing”；
  - (6) 将计算的最终结果重新写入HDFS。

## 12.2 大数据处理平台和框架

- Apache Hadoop: 批处理
- Spark: 批处理、流处理
- Storm: 流处理

# 数据库面临的挑战

- 可扩展性：
  - 随着数据量、访问量和业务量的增加，需要扩展数据库以满足需求
  - 如果扩展：
    - 纵向扩展(scale up, 也译为向上扩展或者垂直扩展)
      - 纵向扩展是指购买更好的机器，配置更多的磁盘、更多的内存等等。
    - 横向扩展(scale out, 也译为向外扩展或者水平扩展)。
      - 横向扩展即使用很多廉价的机器组成集群，这样就可以以较低的成本扩展到较大的规模。
    - 对于可扩展性问题，企业往往首先尝试用纵向扩展方案来解决，但是大型企业发现，纵向扩展不可持续。于是很多大型互联网公司最后都转向横向扩展。

# 数据库的发展

- 关系数据库在大数据时代已经力不从心了：
  - 关系数据库体系结构设计落后于大数据时代
    - 云计算、集群、新型硬件
  - 关系数据库的限制过于严格
    - 模式要求、ACID
- 对策：
  - 摒弃关系数据库的体系结构和关系数据库理论的限制，完全重新设计，设计出了**NoSQL数据库**；
  - 着眼于云计算、集群计算等新的计算环境，对关系数据库的体系结构进行大幅改造，但仍然保留关系数据库的关键特性：模式、SQL语言和ACID特性，产生了**NewSQL数据库**。

## 12.4 NoSQL数据库

- NoSQL: No SQL / Not Only SQL
- 特性:
  - 无模式，或者模式不重要；
  - 不保证强一致性；
  - 不采用SQL作为查询语言；
  - 可扩展性好，并发读写性能高。

# NoSQL分类

因此Redis适合的场景主要局限在较小数据量的高性能操作和运算上。目前使用Redis的网站有 github, Engine Yard。

- 面向高性能并发读写的**Key-Value数据库**:
  - 主要特点是具有极高的并发读写性能。典型代表为Redis和Tokyo Cabinet。
- 面向海量数据访问的**文档数据库**:
  - 主要特点是可以在海量的数据中快速地查询数据。典型代表有MongoDB和CouchDB。
- 面向可扩展性的**分布式数据库**:
  - 目标是解决传统数据库在可扩展性上的缺陷，主要特点是可以较好地适应数据量的增加以及数据结构的变化。典型代表是Google的Big Table。
- 面向图形关系存储的**图数据库**:
  - 目标是解决图形关系的存储问题，主要特点是可以以原生的格式存储图形数据，并支持对图的灵活遍历和查询。典型代表是Neo4j和FlockDB。

# 文档数据库——MongoDB

- MongoDB将数据存储为一个一个文档，即一个类似于JSON格式的对象

```
{  
  name: "sue"  
  age: 26  
  status: "A"  
  groups: ["news", "sports"]  
}
```



# MongoDB和关系数据库的对比

关系数据库概念	MongoDB概念	说明
数据库	数据库(database)	一个数据库可以包含多个集合
表	集合(collection)	一个集合中可以包含多个文档
行	文档(document)	一个文档描述一个对象
列	字段(field)	
索引	索引(index)	MongoDB也支持索引
连接		MongoDB不支持连接
主键	主键(primary key)	MongoDB自动将_id字段设置为主键

关系数据库概念	MongoDB概念	说明
数据库	数据库(database)	一个数据库可以包含多个集合
表	集合(collection)	一个集合中可以包含多个文档
行	文档(document)	一个文档描述一个对象
列	字段(field)	
索引	索引(index)	MongoDB也支持索引
连接		MongoDB不支持连接
主键	主键(primary key)	MongoDB自动将_id字段设置为主键

# MongoDB和关系数据库的区别

- 一个集合没有固定的结构，但通常情况下插入集合的数据都会有一定的关联性。
- 一个集合的文档不需要设置相同的字段，并且相同的字段不需要相同的数据类型
- 文档中的键/值对是有序的
- 文档中的值不仅可以是双引号中的字符串，还可以是其他几种数据类型(甚至可以是整个嵌入的文档)
- 每个文档有一个\_id字段作为它的主键，这个字段的值必须是唯一的

```
{
  ... name: { first: "Alan", last: "Turing" },
  contact: { phone: { type: "cell", number: "111-222-3333" } }, ...
}
```

# 列数据库——HBase

- 列数据库基于宽列存储模型（简称列式存储）
  - 起源于Google的BigTable
  - 传统的关系数据库是行式存储，即以行为单位存储

表 12-8 person 表结构

<u>id</u>	lastname	firstname	username	pwd	timestamp
1	张	三	zhangsan	111	20160719
2	李	四	lisi	222	20160720

行式存储

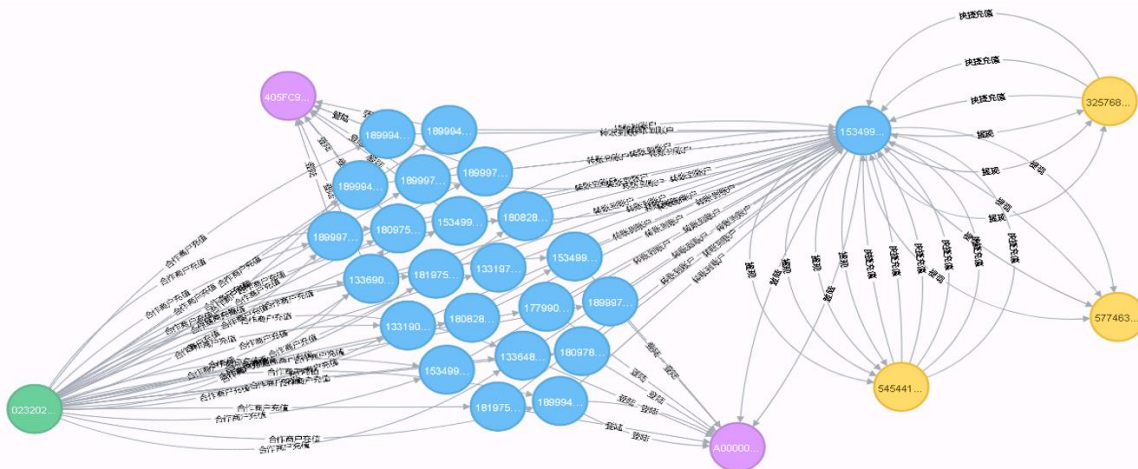
表 12-9 personColumn 表结构

<u>Row-Key</u>	info 列族	login 列族
1	info{'lastname': '张', 'firstname': '三'}	login{'username': 'zhangsan', 'pwd': '111'}
2	Info{'lastname': '李', 'firstname': '四'}	login{'username': 'lisi', 'pwd': '222'}

列式存储

# 图数据库Neo4j

- 图数据库使用图(graph)相关的概念来描述数据模型，把数据保存为图中的结点以及结点之间的关系
  - 非常适合描述如社交网络等数据
- Neo4j
  - 最基本的概念是结点和边。结点表示实体，边则表示实体之间的关系。结点和边都可以有自己的属性。不同实体通过各种不同的关系关联起来，形成复杂的对象图
  - 提供了在对象图上进行查找和遍历的功能



# 反洗钱检测

## 主要功能

异常交易  
结构发现

社区洗钱  
风险评级

## 核心技术

连通子图算法

Tarjan算法

GCN

Louvain算法

反洗钱规则

充值

转账

消费

提现

Neo4j



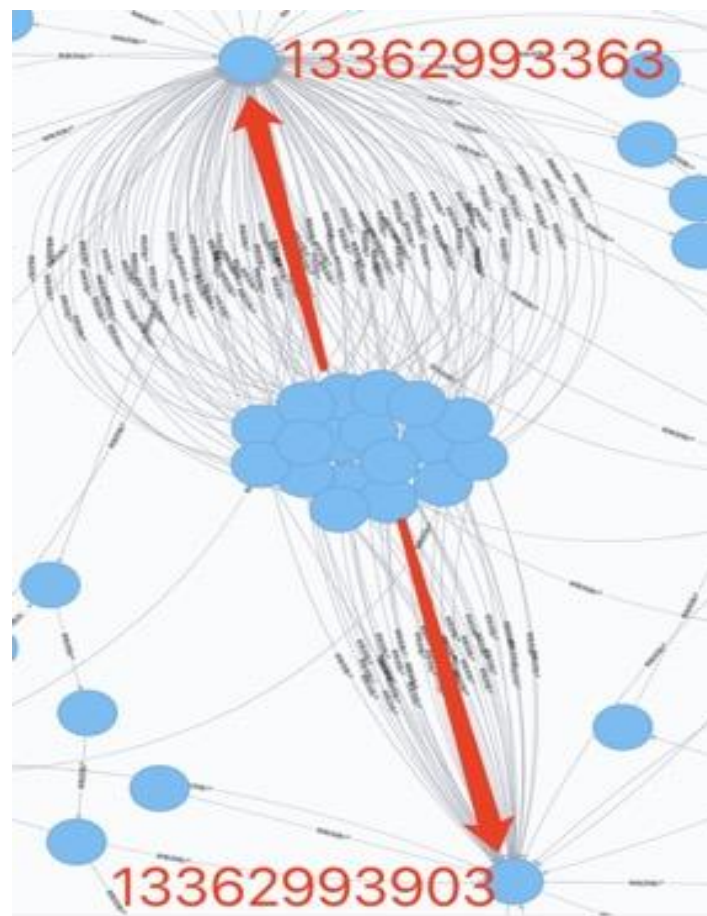
实体表

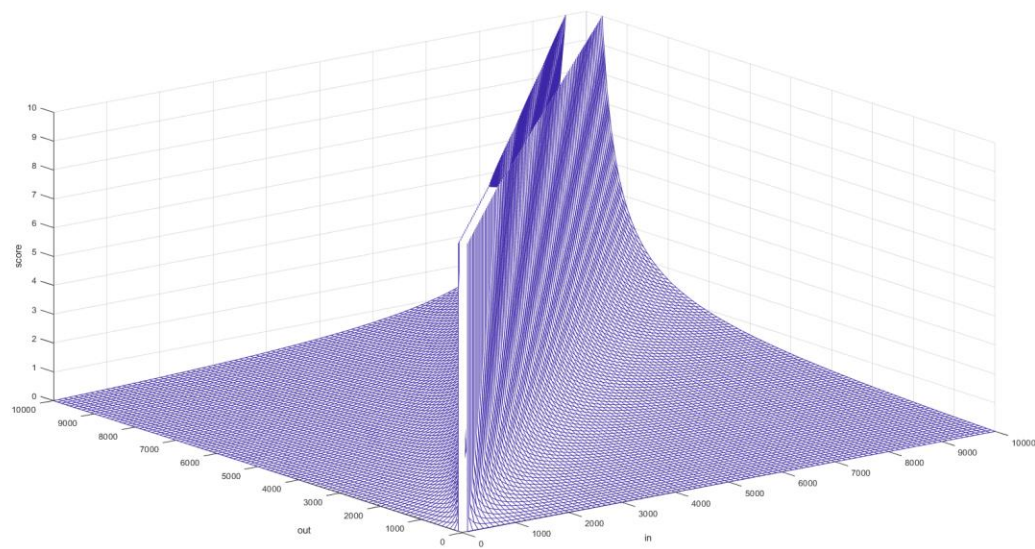
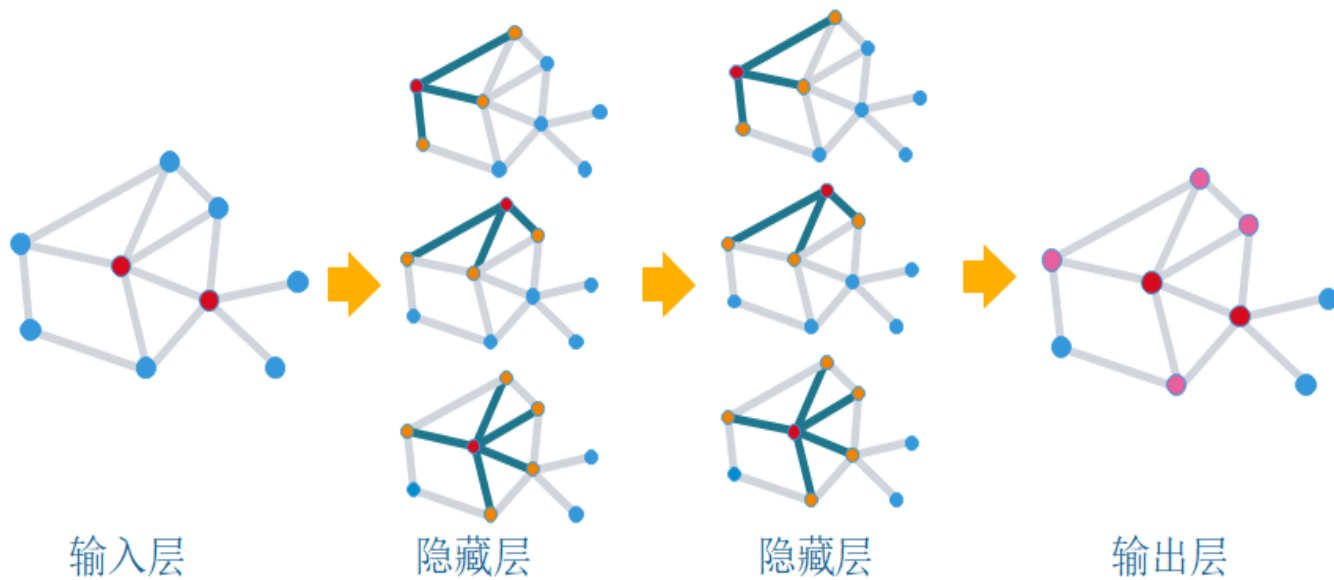


操作表



交易表





# NewSQL数据库

- NoSQL的主要不足：
  - 不支持SQL
  - 不支持ACID
- NewSQL:
  - NewSQL数据库不仅具有NoSQL对海量数据的存储管理能力，还保持了传统数据库支持ACID和SQL等特性
  - 但在体系结构上与传统的RDBMS已经有很大的区别
  - 目前还在探索阶段，没有统一的架构



# NewSQL分类

- 新的体系结构
    - 完全基于新的体系结构，为在分布式集群中运行而设计
    - 集群中结点采用**share-nothing**结构，每个结点都有部分数据
    - 包含了分布式并发控制、流量控制和分布式查询处理等
    - 典型的系统包括：Google Spanner，NuoDB等。
  - SQL引擎
    - 为SQL高度优化的存储引擎，仍然使用SQL接口，但是比内置存储引擎具有更好的可伸缩性
    - 这种类型的产品包括：MySQL Cluster，InfiniDB等
  - 透明分片
    - 提供专门分片的中间件，节点之上。ScaleBase就是
- Share-Nothing**架构又分为两种，首先是分布式架构。将数据库中的数据按照某一标准分布到多台机器中，查询或插入时按照条件查询或插入对应的分区。另一种是每一个节点完全独立，节点之间通过网络连接，通常是通过光钎等专用网络



图表1：数据库的演进



资料来源：华为官网，华泰证券研究所

头条 @未来智库

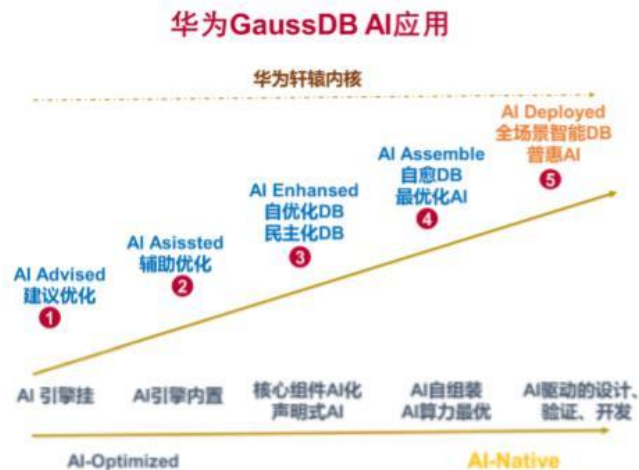
图表3：华为 GaussDB 特点



资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表2：华为 GaussDB 对 AI 技术的应用



资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表4：华为 GaussDB 产品线



资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表6： 华为 GaussDB OLTP&OLAP 特点

华为GaussDB特性



资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表7： 华为 GaussDB HTAP 特点

华为GaussDB HTAP特性



资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表8： 数据库：行式存储

数据库：行式存储

传统行式数据库（关系型数据库）

	1	2	3	4	5	6	7	8	9	...
R1	...	...	...	...	...	...	...	...	...	...
R2	...	...	...	...	...	...	...	...	...	...
R3	...	...	...	...	...	...	...	...	...	...
R4	...	...	...	...	...	...	...	...	...	...
R5	...	...	...	...	...	...	...	...	...	...

- 1、数据是按行存储的
- 2、没有索引的查询使用大量I/O
- 3、建立索引和物化视图需要花费大量时间和资源
- 4、面对查询的需求，数据库必须被大量膨胀才能满足性能需求

资料来源：王顶男，华泰证券研究所

头条 @未来智库

图表9： 数据库：列式存储

数据库：列式存储

列式数据库

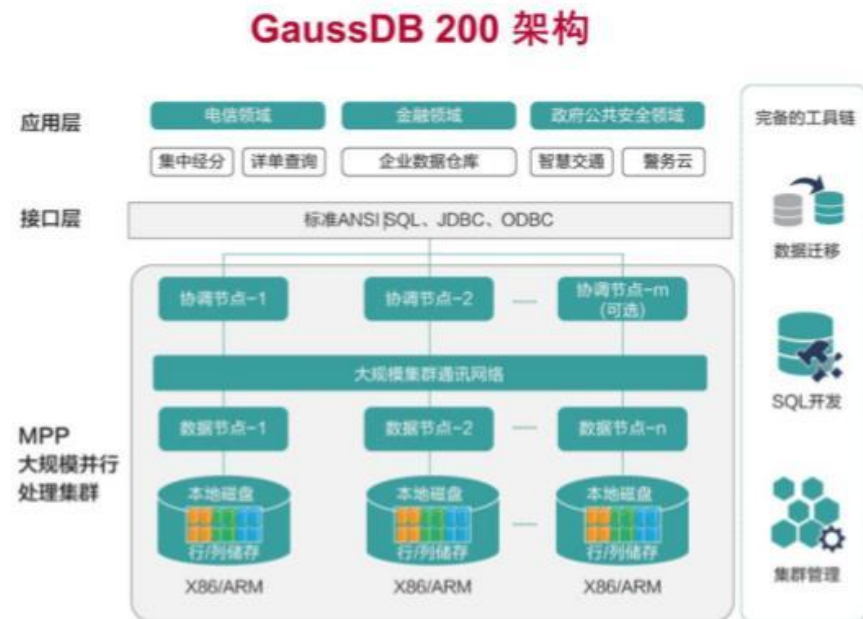
	1	2	3	4	5	6	7	8	9	...
R1	1	1	1	1	1	1	1	1	1	1
R2	1	1	1	1	1	1	1	1	1	1
R3	1	1	1	1	1	1	1	1	1	1
R4	1	1	1	1	1	1	1	1	1	1
R5	1	1	1	1	1	1	1	1	1	1

- 1、数据按列存储-每一列单独存放
- 2、数据即是索引
- 3、只访问查询涉及的列-大量降低系统IO
- 4、每一列由一个线程来处理-查询的并发处理
- 5、数据类型一致，数据特征相似-高效压缩

资料来源：王顶男，华泰证券研究所

头条 @未来智库

图表11： 华为 GaussDB 200 架构

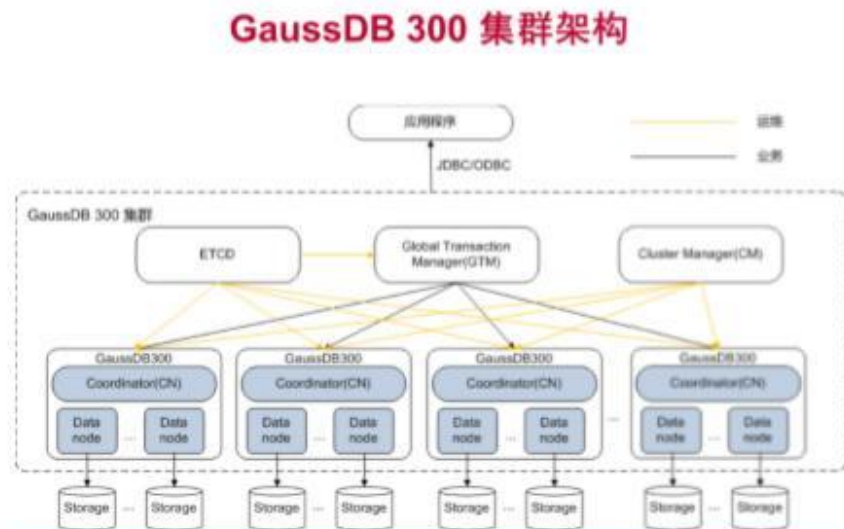


资料来源：华为官网，华泰证券研究所

头条 @未来智库

摘自：  
<https://www.vzkoo.com/news/1476.html>

图表16： 华为 GaussDB 300 集群架构

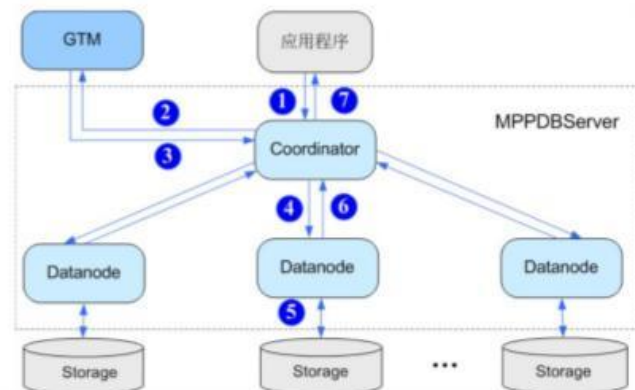


资料来源：华为官网，华泰证券研究所

头条 @未来智库

图表15： 华为 GaussDB200 数据查询过程

### GaussDB 200 数据查询流程



资料来源：华为官网，华泰证券研究所

头条 @未来智库

数据库技术正在进化当中，希望各位同学为数据库的发展添砖加瓦！