

# 数据库系统概论新技术篇

## 内存数据库

张延松

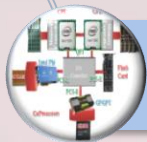
中国人民大学信息学院

2017年4月

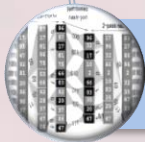
# 目录



什么是内存数据库



新硬件技术推动内存数据库发展



内存数据库技术示例



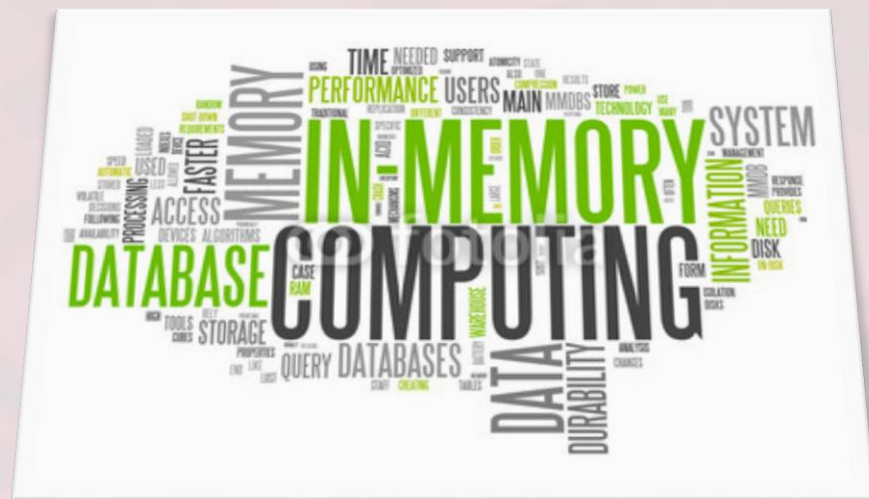
内存数据库发展历程



内存数据库发展趋势



# 什么是内存数据库



# 什么是内存数据库

## 理解内存数据库概念

- ▼ 内存数据库与磁盘数据库的区别
- ▼ 内存数据库的特点
  - ◆ 存储结构
  - ◆ 索引结构
  - ◆ 查询处理

## 内存数据库两大应用领域

- ▼ In-Memory Database Management Systems
- ▼ In-Memory Analytics

## 内存数据库应用架构

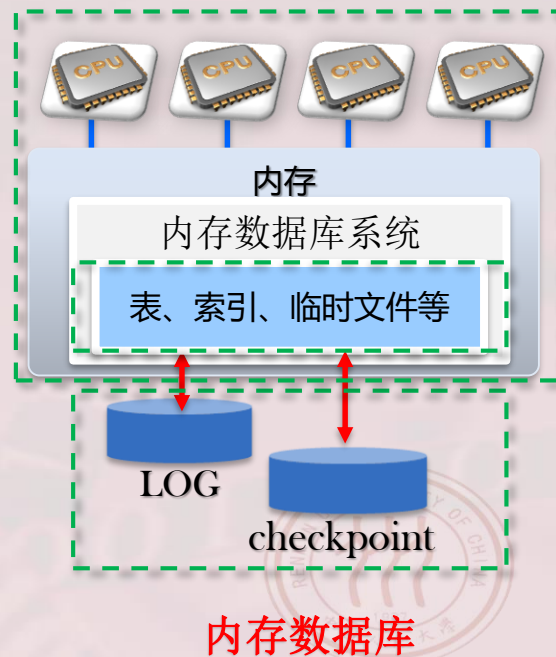
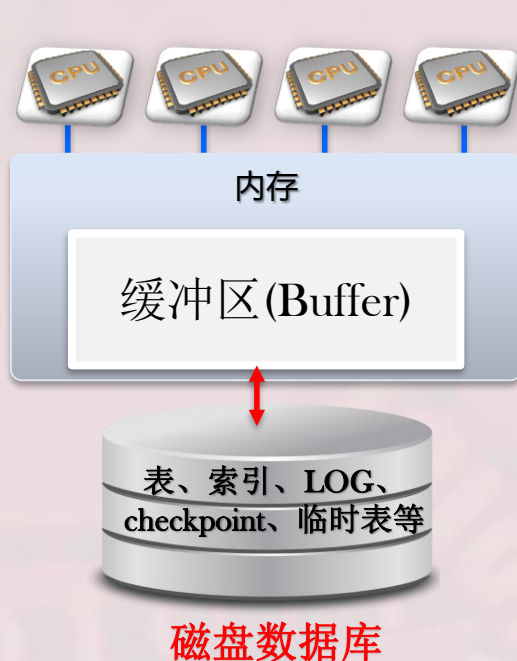
- ▼ 独立型内存数据库
- ▼ 协同型内存数据库
- ▼ 混合型内存数据库

## 内存数据库代表性产品

- ▼ 内存OLTP数据库系统
- ▼ 内存OLAP数据库系统

# 磁盘数据库与内存数据库

- ❖ 定义：使用**内存**作为常规数据存储设备的数据库系统，简称为IMDB(In-Memory Database)或MMDB(Main-Memory Database)
- ❖ 特点：
  - 内存为主要存储设备
  - 磁盘为后备存储和持久化设备
  - 面向内存访问而非I/O的优化



# 内存数据库≠cached磁盘数据库

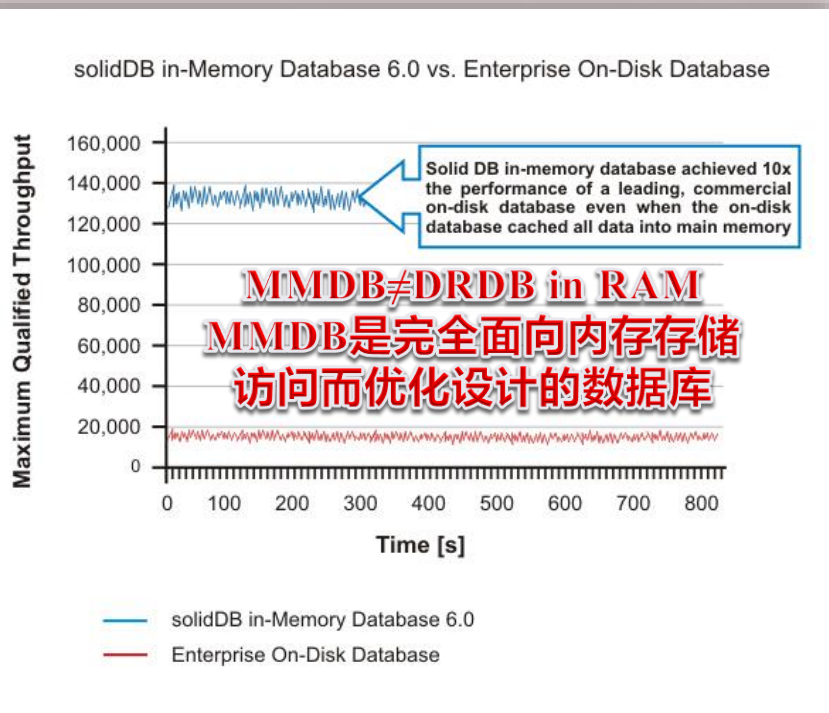
- ❖ 磁盘数据库的缓冲区是磁盘数据在内存的副本，采用与磁盘存储一致的基于page-slot的数据结构，当内存足够将全部磁盘数据缓存在缓冲区时，数据访问只是相当于在内存磁盘（RAM Disk）上的访问。
- ❖ 内存数据库的存储和访问算法以内存访问特性为基础，实现处理器对数据的直接访问，在算法和代码效率上高于以磁盘I/O为基础的磁盘数据库。
- ❖ 在内存数据库中，使用针对内存访问特性进行优化的T树索引和hash索引、面向cache优化的索引算法和多种面向连接操作的优化技术，进一步优化了内存数据库的性能。
- ❖ 与数据全部缓存到内存的磁盘数据库相比，内存数据库的性能仍然超出数倍。





# 内存数据库≠cached磁盘数据库

- ❖ 磁盘数据库的存储一致的基于数据缓存在缓存(Disk)上的访问
- ❖ 内存数据库的处理器对数据的基础的磁盘数据库
- ❖ 在内存数据库和hash索引、优化技术，
- ❖ 与数据全部缓存仍然超出数倍。



采用与磁盘存储  
够将全部磁盘  
内存磁盘 (RAM

为基础，实现处  
高于以磁盘I/O为

优化的T树索引  
种面向连接操作

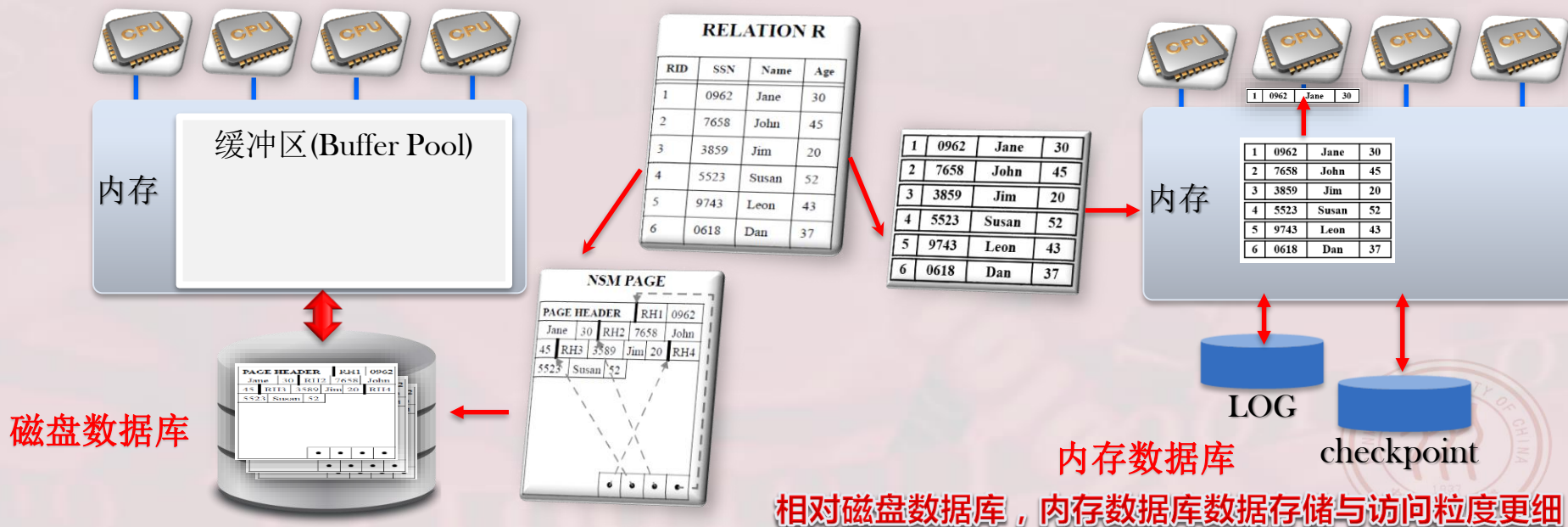
存数据库的性能



# 内存数据库特点

## ❖ 存储结构:

- 磁盘数据库在磁盘为page为单位存储数据，一次I/O访问一个数据page(4KB或8KB)
- 内存数据库在内存中按行或列存储，数据访问的单位为cache line(64字节)

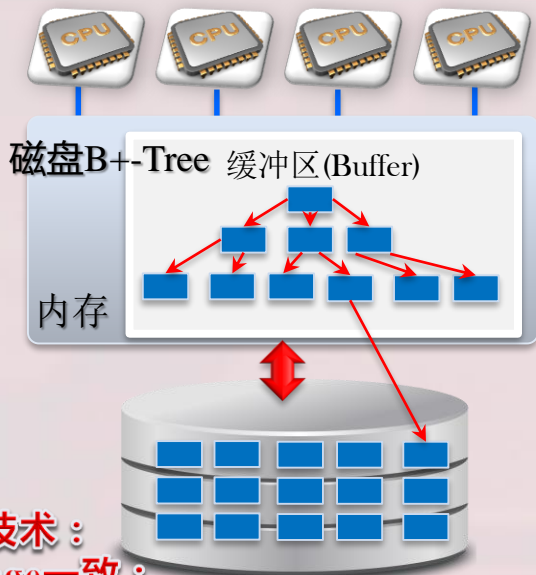




# 内存数据库特点

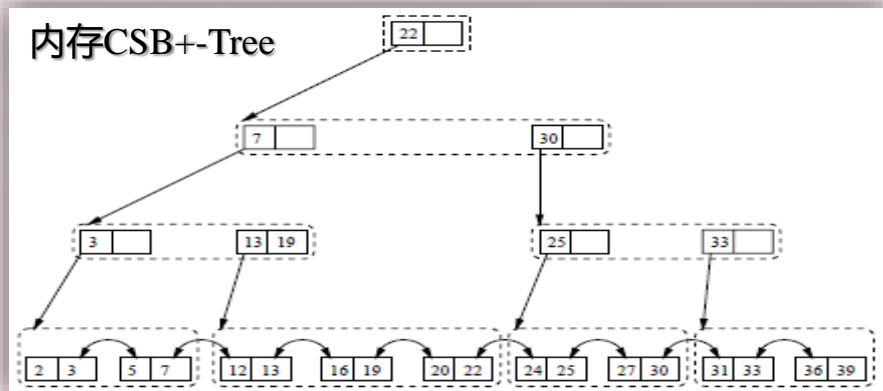
## ❖ 索引结构:

- 磁盘B+-Tree索引: 以 **page** 为单位存储索引节点, 索引访问涉及多个I/O操作
- 内存CSB+-Tree索引: 以 **cache line** 为存储和访问单位, 优化指针存储



磁盘B+树优化技术:

1. 节点大小与page一致;
2. 内部节点多指针减少I/O访问代价



CSB+树的cache优化技术:

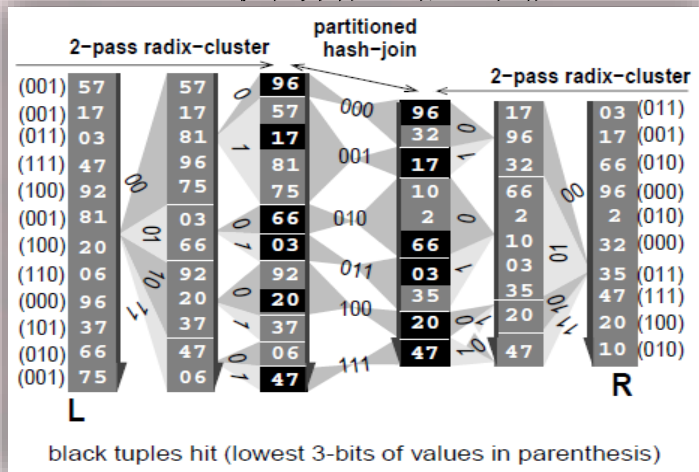
1. 节点大小与cache line一致;
2. 内部节点对应的子节点存储在连续区域



# 内存数据库特点

## ❖ 查询处理:

- 磁盘数据库查询优化以I/O估算为基础, 目标是**提高缓冲区利用率, 减少I/O代价**
- 内存数据库查询优化以**cache line miss**为基础, 目标是**提高cache利用率, 降低内存访问延迟, 提高处理器的并行处理效率**
  - 适合**cache line**大小的数据存储和数据访问结构
  - 将大表划分为适应**cache**大小的分区
  - 优化数据访问, 提高**cache**利用率
  - 优化并行查询处理性能



多趟基数划分

hash连接  
或嵌套连接

```
partitioned-hashjoin(L, R, H):  
  radix-cluster(L,H)  
  radix-cluster(R,H)  
  FOREACH cluster IN [1 ... H]  
    hash-join(L[c], R[c])
```

```
radix-join(L, R, H):  
  radix-cluster(L,H)  
  radix-cluster(R,H)  
  FOREACH cluster IN [1 ... H]  
    nested-loop(L[c], R[c])
```

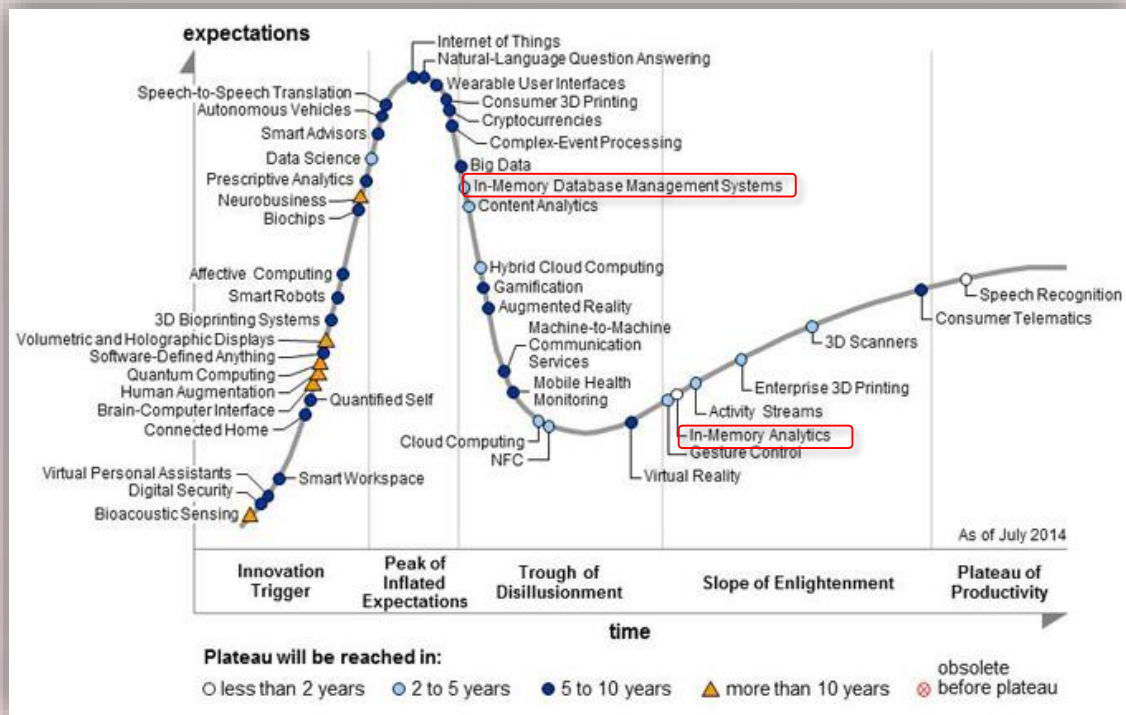
# 内存数据库的应用领域

## ❖ In-Memory Database Management Systems:

- 应用于内存事务处理
- 面向高吞吐、低延迟、高响应事务处理应用
- 可以作为独立的内存数据库事务处理引擎，也可以作为磁盘数据库引擎的高速cache

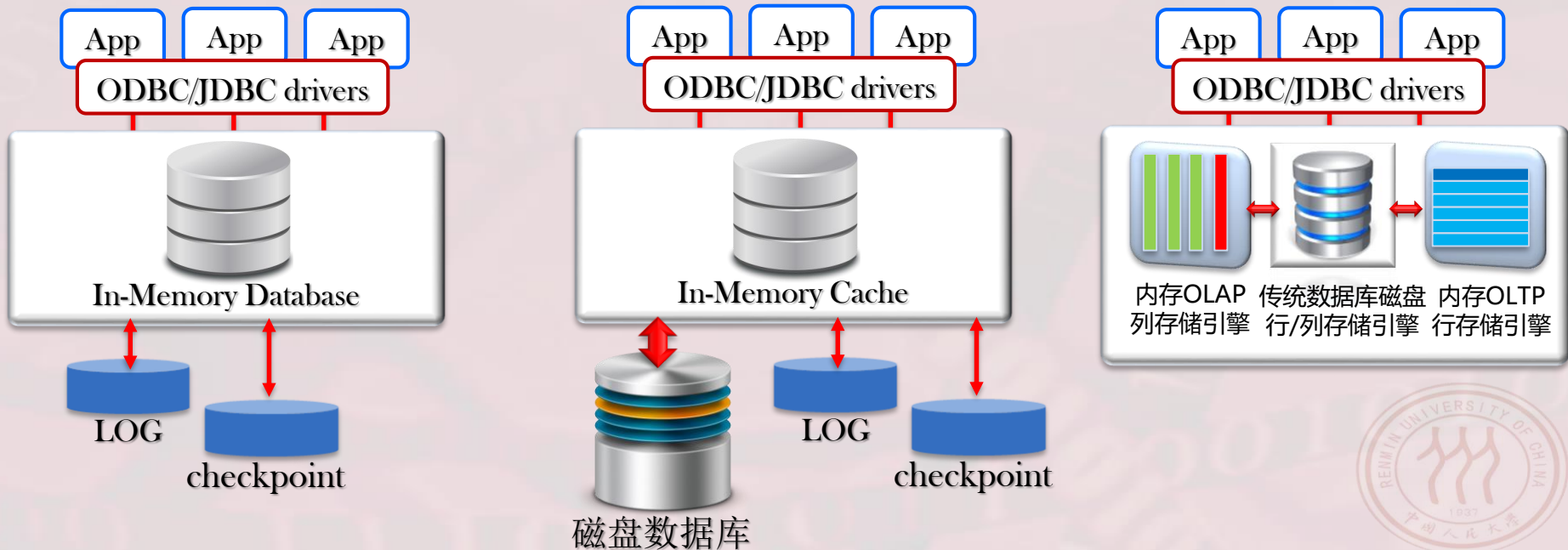
## ❖ In-Memory Analytics:

- 应用于内存分析处理
- 面向大数据、内存实时分析处理应用
- 主要面向数据仓库和联机分析处理应用



# 内存数据库应用架构

- ❖ 独立的内存数据库
- ❖ 作为传统磁盘数据库的前端**Cache**数据库
- ❖ 集成的内存数据库引擎



# 内存数据库代表性产品

## ❖ 代表性内存OLTP数据库

- 独立及高速缓存内存数据库: **Oracle TimesTen, IBM SolidDB**
- 内存-磁盘混合架构内存数据库: **Altibase**
- 实时及嵌入式内存数据库: **McObject eXtremeDB**
- 融合内存数据库: **Microsoft SQL Server Hekaton**
- 分布式NewSQL内存数据库: **VoltDB**

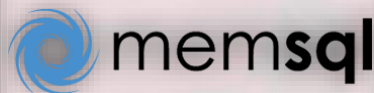




# 内存数据库代表性产品

## ❖ 代表性内存OLAP数据库

- 学术系统: **MonetDB, Hyper**
- 向量处理技术内存数据库: **Vectorwise**
- 内存分析加速技术: **IBM BLU Acceleration**
- 内存分析引擎: **Oracle Database In-Memory, SQL Server, MEMSQL**
- OLTP&OLAP内存数据库: **SAP HANA**
- MPP内存数据库: **EXASOL**



# 小结

- ❖ 内存数据库完成了传统磁盘数据库的技术**升级**
  - 物理存储引擎从磁盘升级为内存：系统架构、性能、数据访问粒度
- ❖ 内存数据库**释放**了现代多核处理器的计算潜能
  - 高带宽、并行访问的内存充分利用多核处理器的并行处理能力
  - 多级缓存优化技术进一步提高多核处理器的数据访问性能
- ❖ 内存数据库使**实时数据分析处理**成为可能
  - 内存OLTP和OLAP支持高性能事务及分析处理
  - 内存OLTP和OLAP融合为统一的数据库引擎成为新趋势

