数据库系统概论新技术篇

数据仓库与联机分析处理技术(5)

陈红

中国人民大学信息学院

新的研究方向

- 传统问题
  - 实体化视图的增量维护
  - 数据集成
  - ……
- 新的多维数据分析方法
  - SKYLINE
  - TOP-K
  - KNN
- 新的硬件环境
  - 内存OLAP
  - 多核OLAP
  - 基于协处理器的OLAP
  - 实时数据仓库
- 新的应用场景
  - 大数据OLAP
  - 流数据的联机分析
  - 物联网中的联机分析

内存OLAP和多核OLAP的困境

- OLAP不仅是数据密集型应用，更是计算密集型应用，即使数据全部放在内存中，数据的计算量也不是短时间内完成的；
- 多核OLAP虽然在一定程度上缓解了计算瓶颈，但由于CPU缓存大小有限以及CPU缓存与内存之间的带宽较低，随着内存容量的增大，会进一步增加缓存缺页的次数，从而无法保证好的效率。

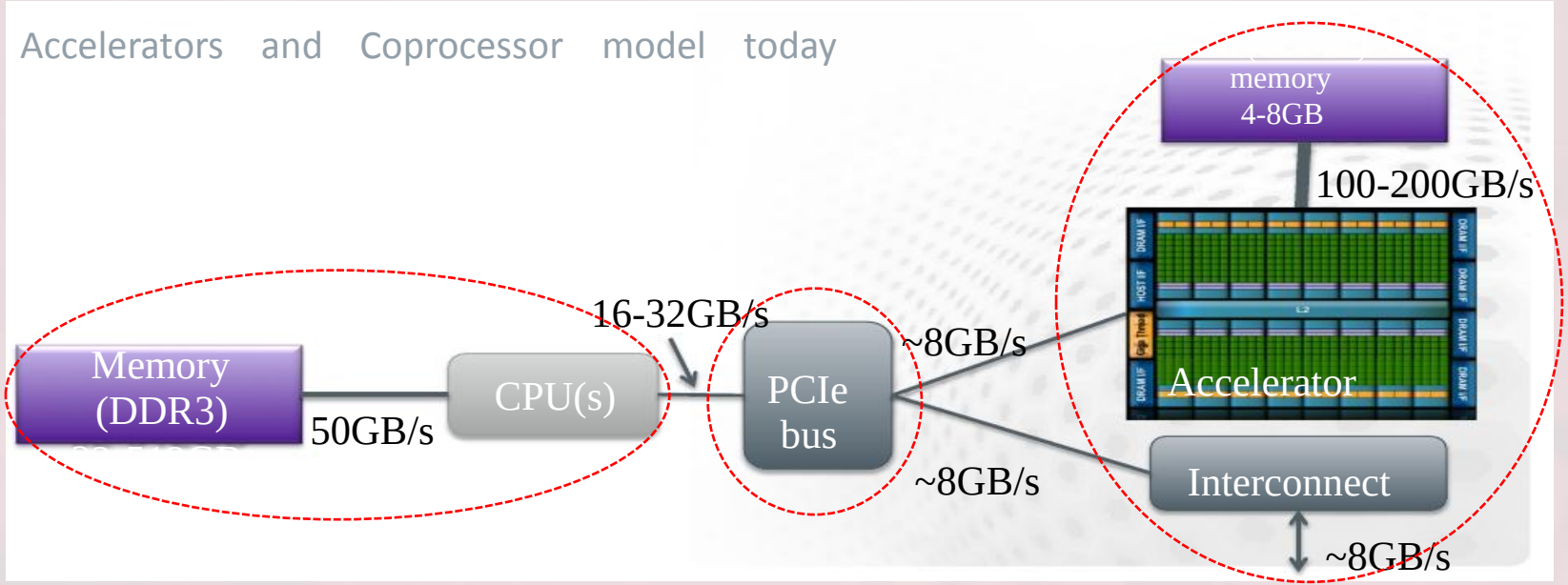
基于协处理器的OLAP

- 内存OLAP的性能瓶颈本质上由两个原因造成：第一，CPU计算速度慢；第二，存储器与处理器之间带宽低。
- 协处理器的高带宽及其先天具备的大数据量并行计算能力，为解决内存OLAP的性能问题提供了良好的硬件基础

协处理器产品体系

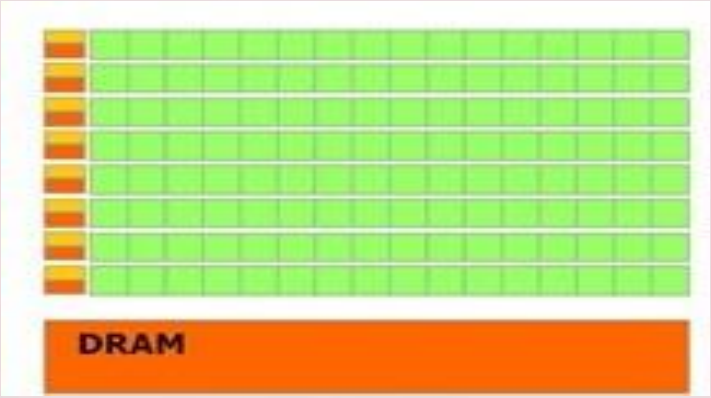
- Different families of technologies
  - GPGPU (Nvidia Tesla, AMD)
  - Manycores (Intel MIC, Adapteva)
  - FPGA (Convey etc.)

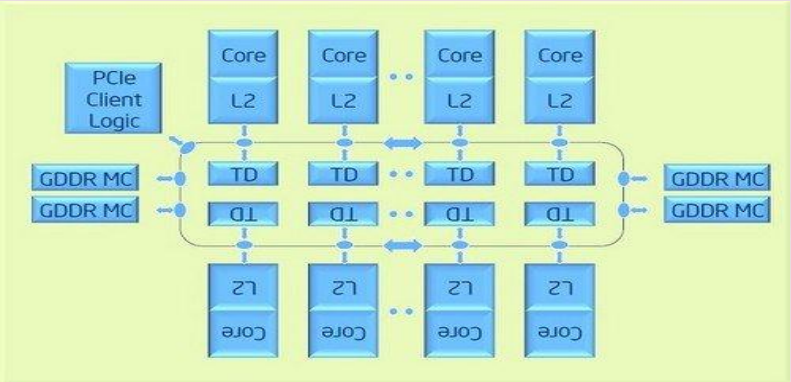
协处理器一般架构



- 与较大的CPU内存(512G)相比,协处理设备缓存较小(4~8G); 与CPU内存访问带宽(50GB/s)相比,协处理设备内存访问带宽更佳(100~200GB/s);
- 加速设备通过PCIe通道与CPU相连.
- 多个协处理设备之间的通信速度较慢(8GB/s).
- 整体架构中,PCIe数据传输(16~32GB/s)通常被认为是性能制约最重要因素.

协处理器特性





类型	Xeon E7-4890 v2	Xeon Phi 7120X	NVIDIA Tesla K40
核心数量/线程数量	15 /30	61/244	2880 CUDA cores
主频	2.80 GHz	1.24 GHz	732MHz
内存容量	1536 GB	16GB	12GB
缓存容量	37.5MB	30.5MB	1.5MB
内存类型	DDR-3	GDDR5 ECC	GDDR5
内存带宽	85GB/s	352 GB/s	288 GB/s
价格	$6619.00	$4129.00	$5500.00

GPU OLAP的研究

- **the University of North Carolina**
  - 研究了包括比较运算,布尔运算以及范围查询等基本操作符在**GPGPU**上实现
    - **Fast Computation of Database Operations using Graphics Processors. SIGMOD, 2004:215--226.**
    - **Fast and Approximate Stream Mining of Quantiles and Frequencies Using Graphics Processors. SIGMOD, 2005:611--622.**
    - **GPUTeraSort: High Performance Graphics Co-processor Sorting for Large Database Management. SIGMOD 2006: 325—336**

GPU OLAP的研究

- **University of Virginia**
  - 基于CUDA的数据库操作
    - **Accelerating SQL database operations on a GPU with CUDA. GPGPU 2010: 94-103**
    - **A Performance Study for Iterative Stencil Loops on GPUs with Ghost Zone Optimizations. International Journal of Parallel Programming 39(1): 115-142 (2011)**
    - **Pannotia: Understanding irregular GPGPU graph applications. IISWC 2013: 185-195**
    - **BenchFriend: Correlating the performance of GPU benchmarks. IJHPCA 28(2): 238-250 (2014)**

GPU OLAP的研究

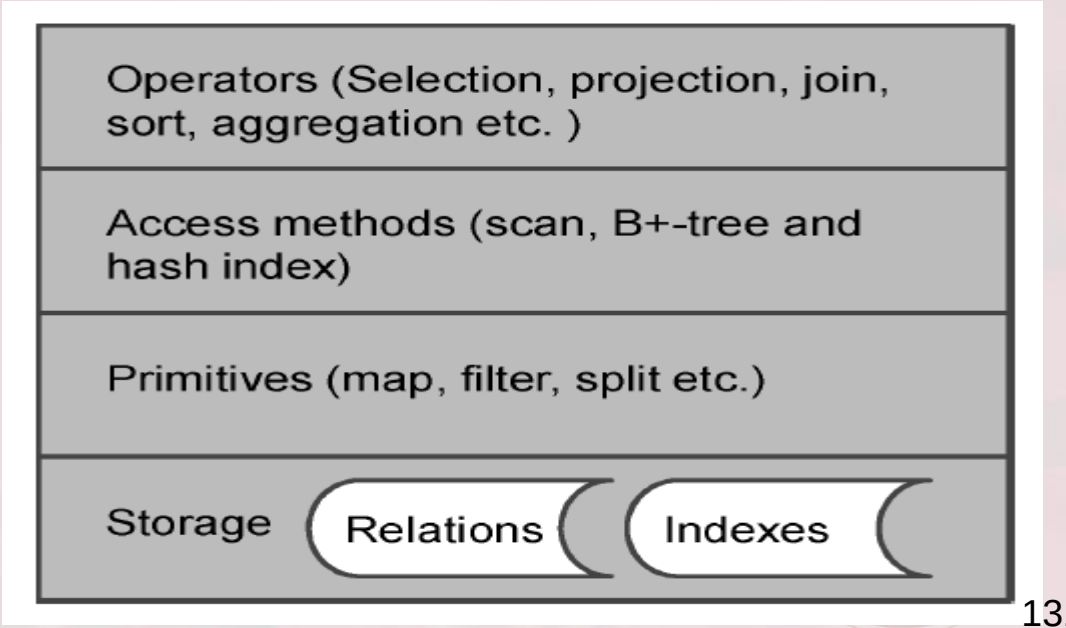
- **University of Magdeburg**
  - 研究了混和平台上的查询计划自调优模型，及相应原型系统**HyPE**
    - **Efficient co-processor utilization in database query processing. Inf. Syst. 38(8): 1084-1096 (2013)**
    - **Why it is time for a HyPE: A Hybrid Query Processing Engine for Efficient GPU Coprocessing in DBMS. PVLDB 6(12): 1398-1403 (2013)**
    - **Towards Optimization of Hybrid CPU/GPU Query Plans in Database Systems. ADBIS Workshops 2012: 27-35**
    - **Self-Tuning Distribution of DB-Operations on Hybrid CPU/GPU Platforms. Grundlagen von Datenbanken 2012: 89-94**

GPU OLAP的研究

- Hong Kong University of Science and Technology
  - gather，scatter，join等在GPU上的实现
  - 原型系统研发(GDB系统)
    - Efficient gather and scatter operations on graphics processors. Nov. 2007 Proceedings of the 2007 ACM/IEEE conference on Supercomputing.
    - Bingsheng He, Wenbin Fang, Qiong Luo, Naga K. Govindaraju, Tuyong Wang: Mars: a MapReduce framework on graphics processors. PACT 2008: 260-269
    - Relational query coprocessing on graphics processors. ACM Trans. Database Syst. 34(4) (2009) ---GDB

基于GPU的分析型数据库系统GDB

- GDB系统中，将数据库操作细化并定义为一系列的原语，原语是原子级操作，通过原语构成关系操作符。
  - Map
  - Scatter and gather
  - Prefix Scan
  - Split
  - Sort



基于GPU的分析型数据库系统GDB

- 在GPU上实现了一系列操作符
  - 连接：
    - Non-indexed NLJs (NINLJ)
    - Indexed NLJs (INLJ)
    - Sort-Merge Joins (SMJ)
    - Hash joins(HJ)
  - 排序
    - Bitonic Sort
    - GPU TeraSort
    - Quick Sort

Joins	CPU (sec)	GPU (sec)	Speedup
NINLJ	528.0	75.0	7.0
INLJ	4.2	0.7	6.1
SMJ	5.0	2.0	2.4
HJ	2.5	1.3	1.9

CPU与GPU的协同处理 -1

Hong Kong University of Science and Technology

❖协同处理基本代价

$$T_{overall} = T_{mm\_dm}(I) + T_{GPU} + T_{dm\_mm}(O).$$

- 三种处理方式
  - **on the CPU only (EXEC CPU)**
  - **on the GPU only (EXEC GPU)**
  - **using both processors (EXEC COP)**
- 协同处理
  - 针对每一种操作建立具体的代价计算公式,在查询处理过程中对于每个操作任务计算其代价,再通过预先制定的规则分派到**CPU**或**GPU**执行.**GPU**执行的操作需要数据由**host**端向**device**端的传输

CPU与GPU的协同处理 -2

Technical University of Lisbon

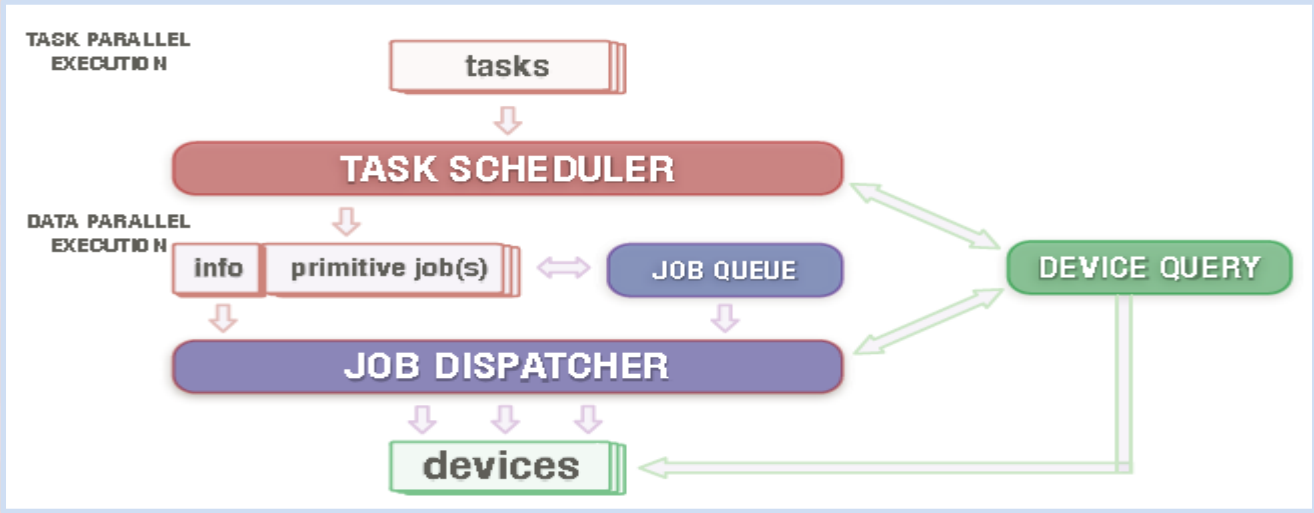
- **The Task Scheduler**
  - 粗粒度任务的入口,对应任务级并行
- **Primitive Jobs**
  - 原子级操作,可按任意顺序执行

Job Queue

– 将 Primitive Jobs组织为多维数据结构

Job Dispatcher

– 将Job Queue中的操作分派到Device执行



CPU与GPU的协同处理 -3

University of Magdeburg

❖调度优化

self-tuning execution time estimator

algorithm selector

hybrid query optimizer

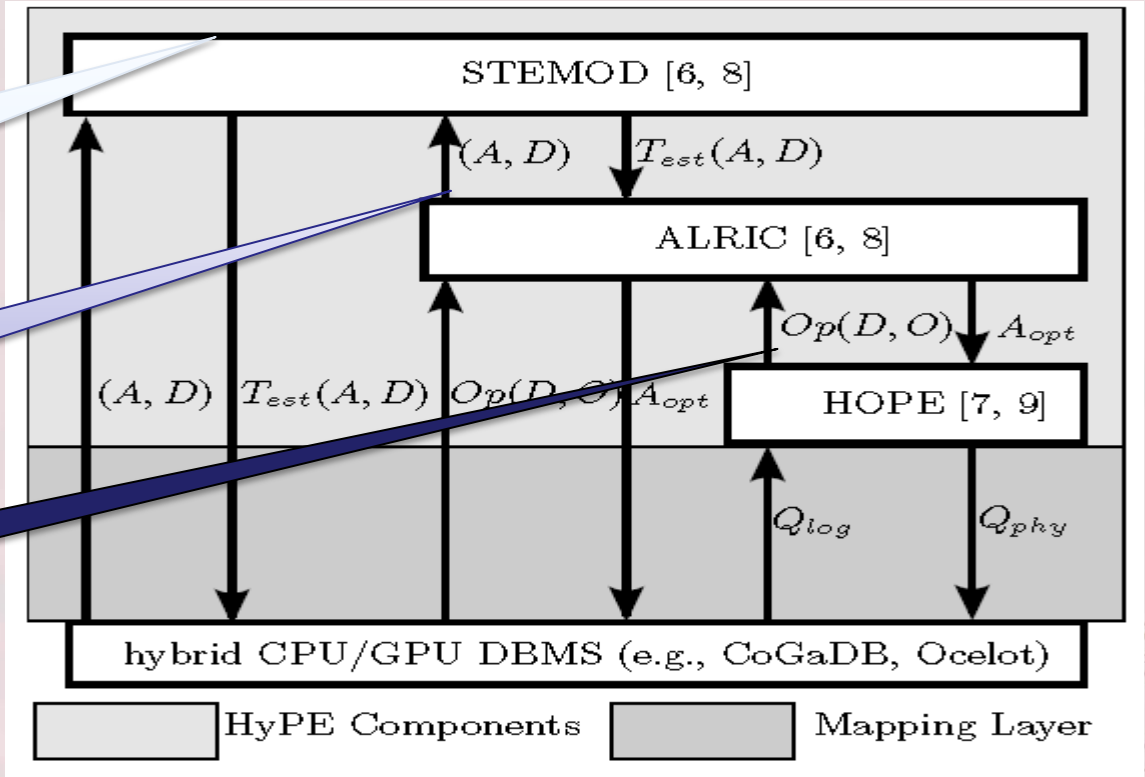
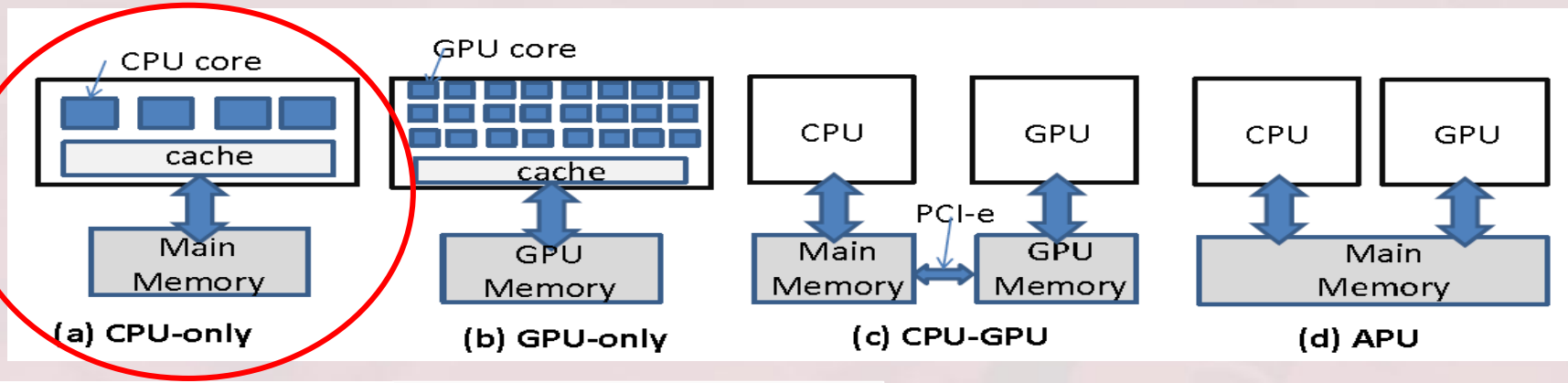


Fig. Architecture of HyPE

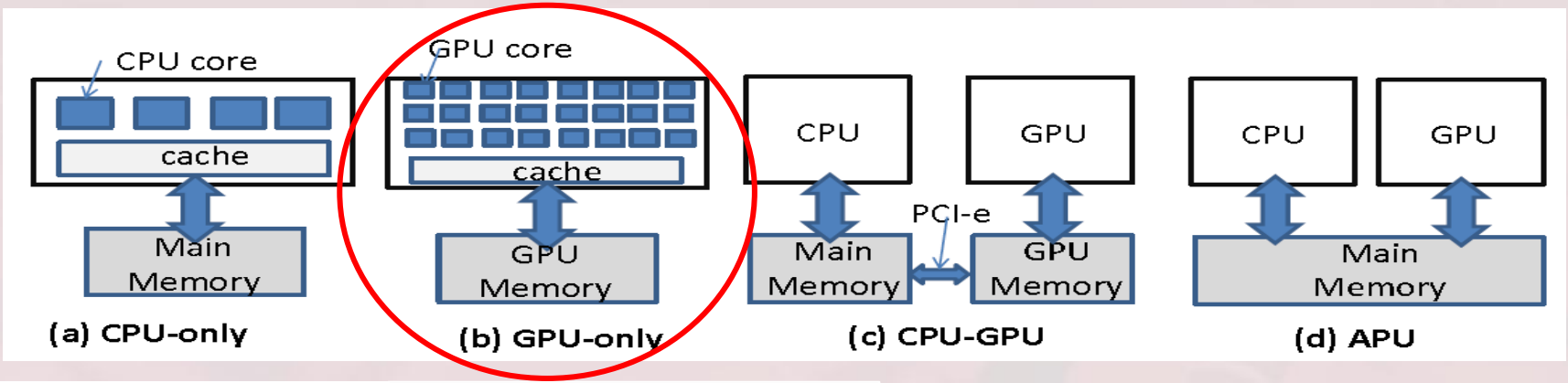
CPU,GPU与APU

❖从离散到融合



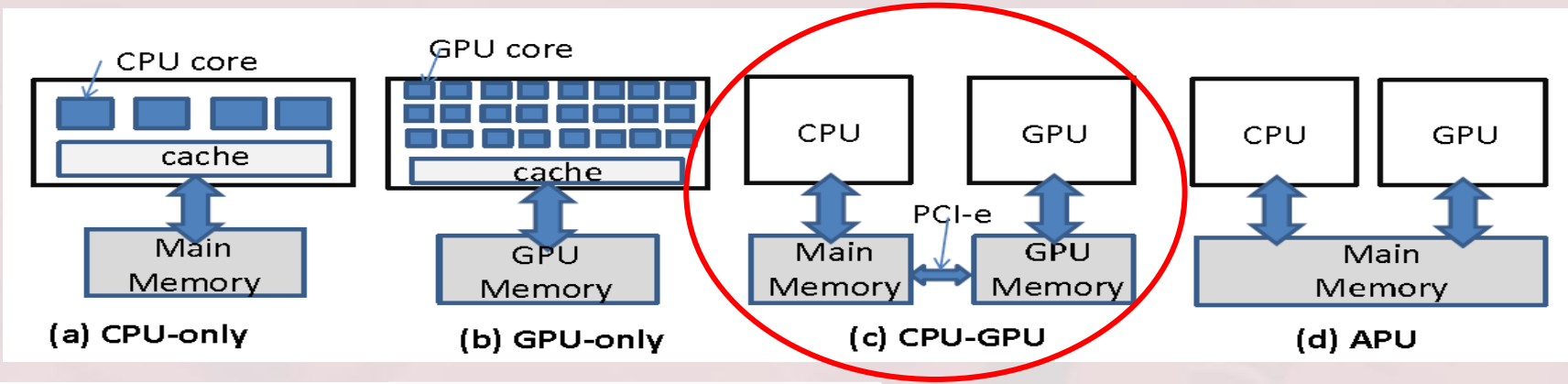
CPU,GPU与APU

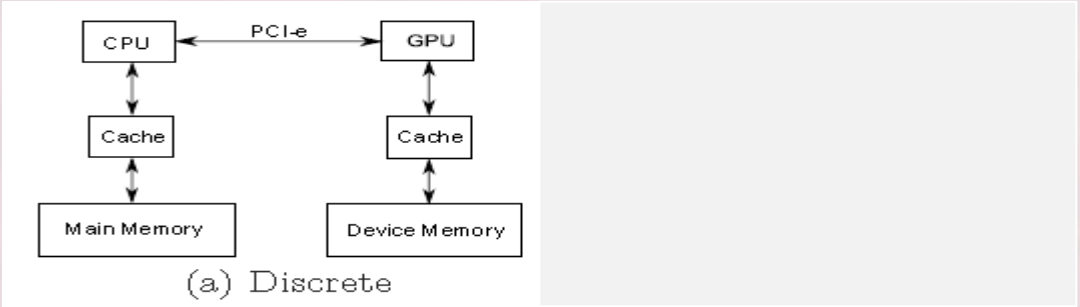
❖从离散到融合



CPU,GPU与APU

❖从离散到融合

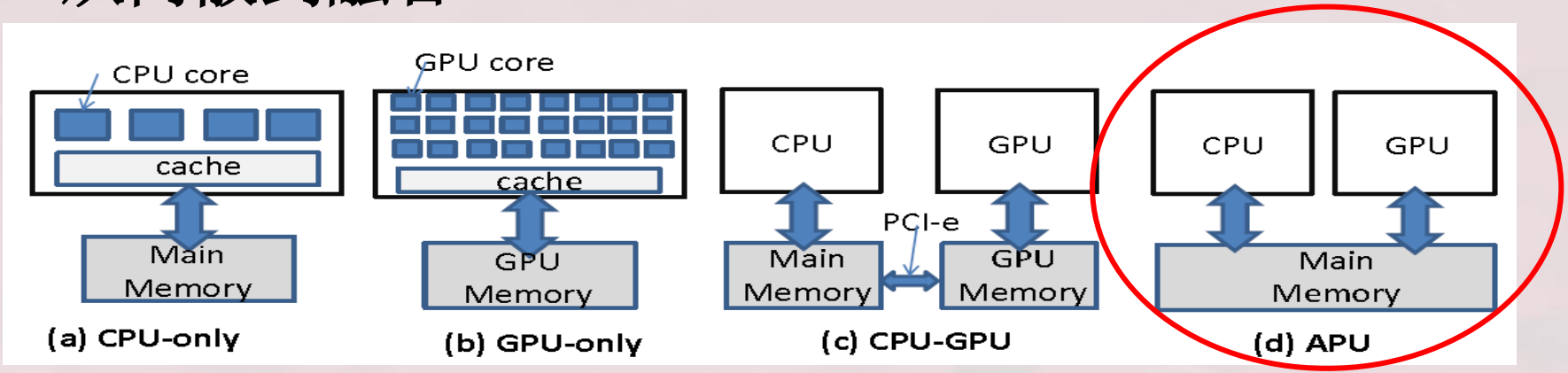


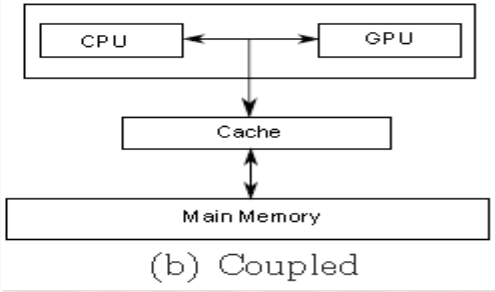


- 在CPU与GPU混合架构中,二者各自使用自己的Cache进行数据访问,数据可重用性差.并且导致数据必须经由相对低速的PCIe通道进行往复传输,在很大程度上削减了GPU并行计算所带来的收益.

CPU,GPU与APU

❖从离散到融合





- 在CPU与GPU混合架构中,二者各自使用自己的Cache进行数据访问,数据可重用性差.并且导致数据必须经由相对低速的PCIe通道进行往复传输,在很大程度上削减了GPU并行计算所带来的收益.
- AMD公司的产品思路是将CPU与GPU放在同一个芯片Chip上,并且使二者可以共同访问同一块存储区域,有统一的内存控制,从而做到数据的共享

APU的研究

❖ APU的数据库应用

■ Hash Join （ VLDB 2013）

- He.等研究了no-partition Join算法和Partitioned Hash Join在APU上的实现;
- the coupled architecture enables fine-grained co-processing and cache reuses;
- the cost model can automatically guide the design and tuning knobs in the design space;
- fine-grained co-processing achieves up to 53%, 35% and 28% performance improvement over CPU-only, GPU-only and conventional CPU-GPU co-processing, respectively.

Phi协处理器

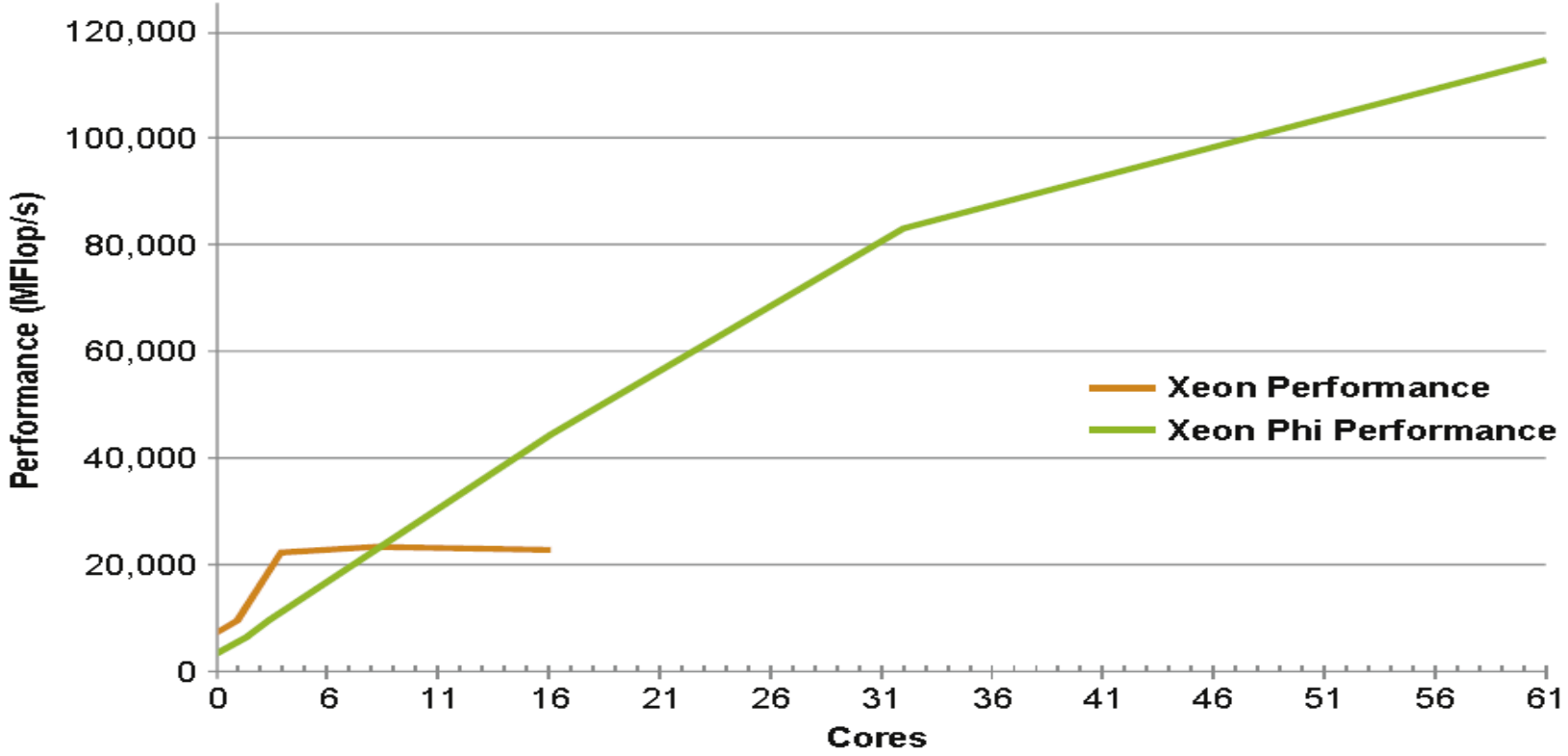
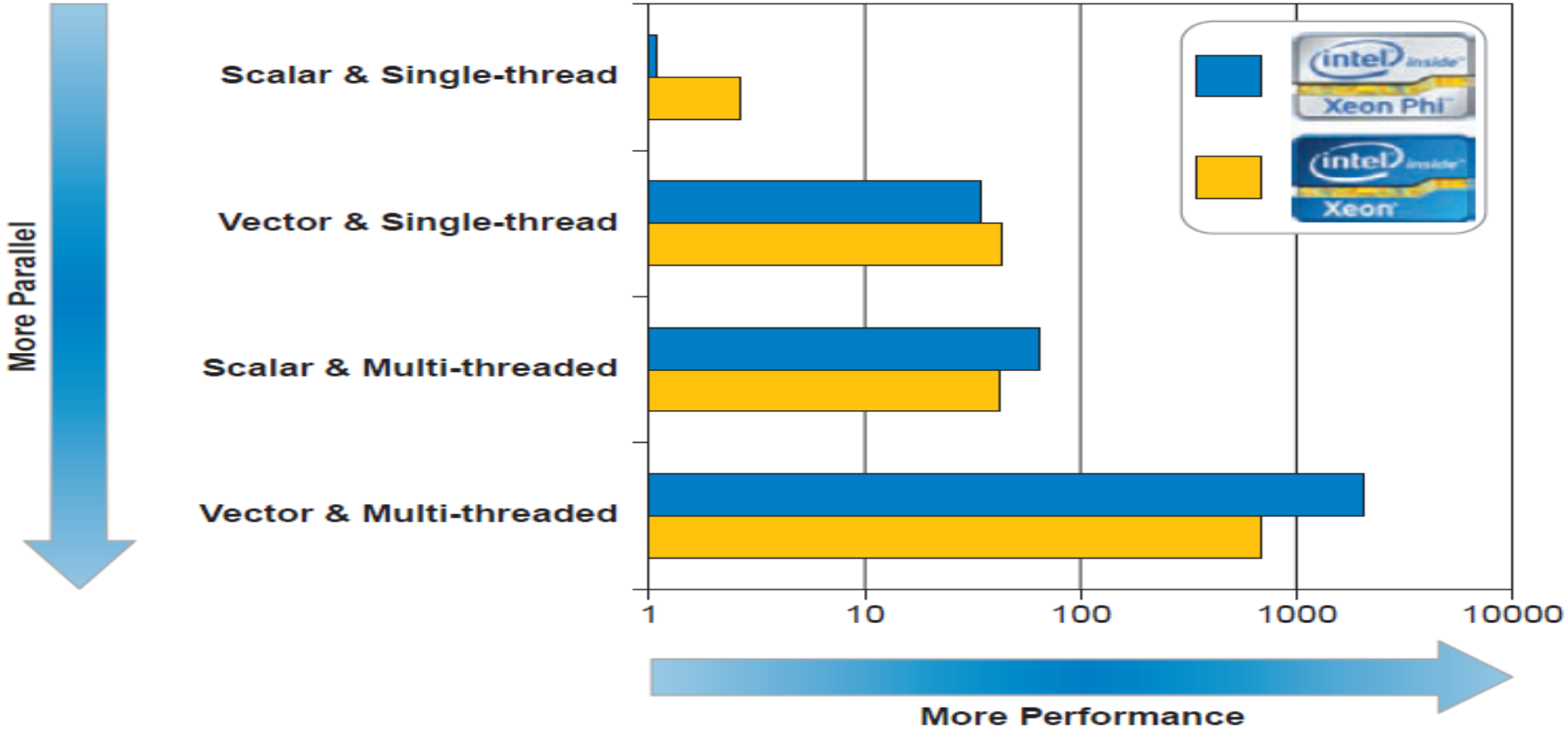


FIGURE 3.7

Processor and Coprocessor Performance on the Highly Parallel 9-Point 2D Stencil Code.

Phi协处理器



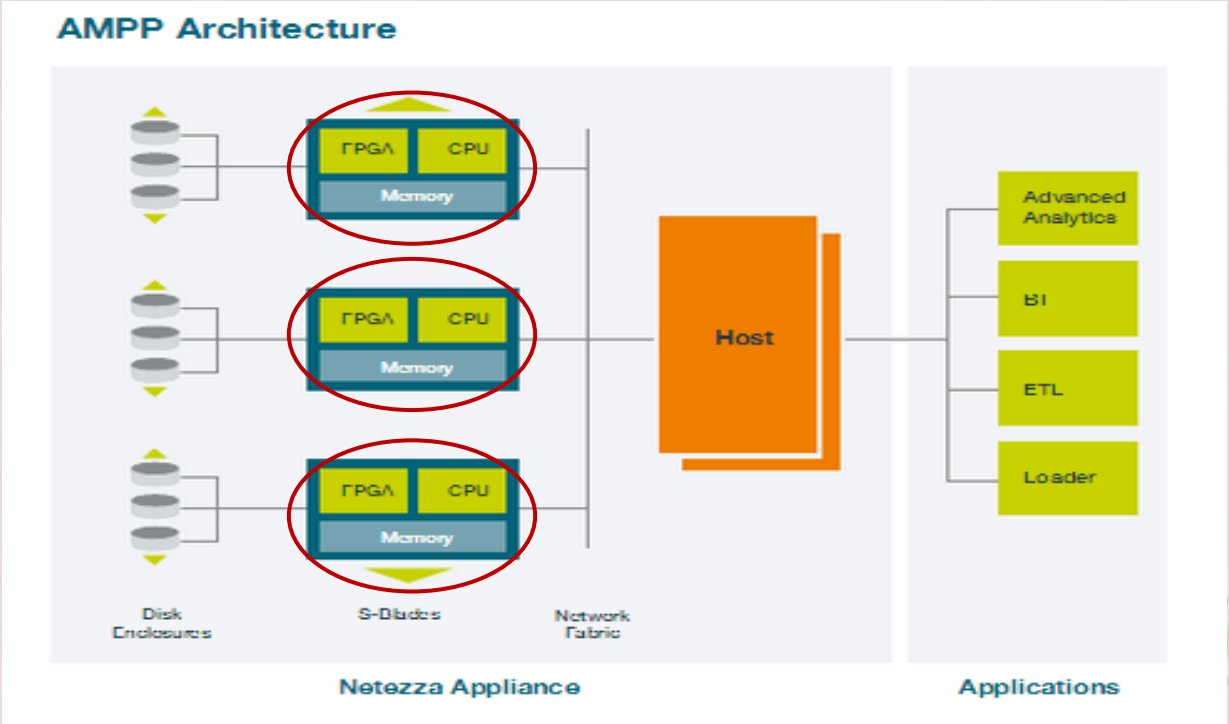
Processor and Coprocessor Performance on the Highly Parallel 9-Point 2D Stencil Code.

Phi协处理器

- 算法调优
  - **Vectorizing**
    - **refers to using the fundamental data parallel engines that support single instruction, multiple data (SIMD) usage available in Intel architecture processors with the SSE and AVX instruction extensions and the Intel Xeon Phi instruction set architecture.**
  - **Scaling**
    - **refers to enabling the code to run across the many cores and hardware threads as independent parallel tasks**

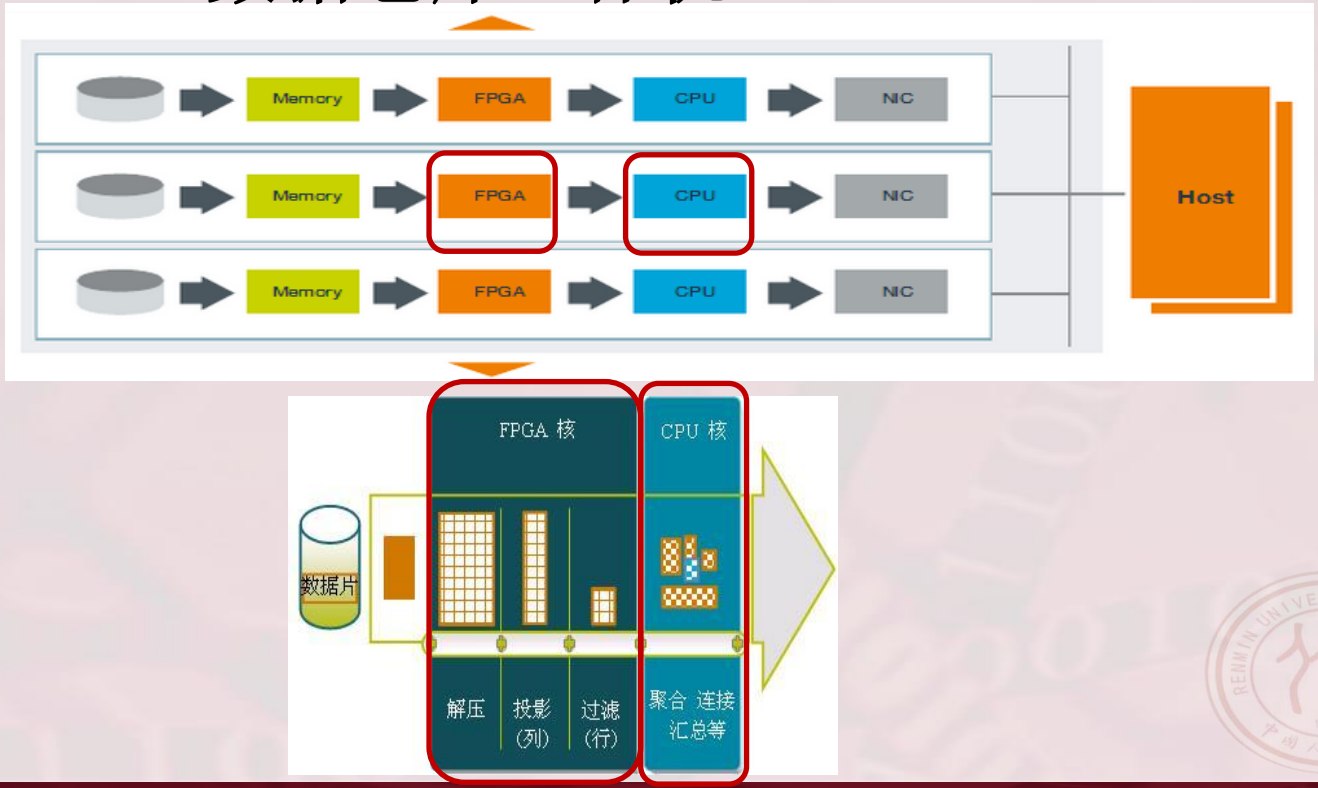
FPGA的研究

- **IBM Netezza 数据仓库一体机**



FPGA的研究

- **IBM Netezza 数据仓库一体机**



典型系统

❖基于GPU的OLAP系统

产品名称	研究机构	类型
MapD	**Massachusetts Institute of Technology**	商业系统
GPUdb	**Ohio State University**	准商业系统
Ocelot	柏林工业大学	开源
CoGaDB	马格德堡大学	开源
GPUQP	香港科技大学	原型系统
OmniDB	南洋理工大学	原型系统
MultiQx-GPU	**Ohio State University**	原型系统
virginian	**NEC Laboratories America**	原型系统

❖基于PHI的OLAP原型系统

产品名称	研究机构	类型
PhiDB	南洋理工大学	原型系统

新的研究方向

- 传统问题
  - 实体化视图的增量维护
  - 数据集成
  - ……
- 新的多维数据分析方法
  - SKYLINE
  - TOP-K
  - KNN
- 新的硬件环境
  - 内存OLAP
  - 多核OLAP
  - 基于协处理器的OLAP
  - 实时数据仓库
- 新的应用场景
  - 大数据OLAP
  - 流数据的联机分析
  - 物联网中的联机分析

什么是实时数据仓库

- 传统数据仓库
  - OLTP和OLAP分离
  - 通过ETL工具将事务型 数据加载到数据仓库中
  - 分析数据滞后

- 实时数据仓库(Real-Time Data Warehouse, RTDW)
  - 基于实时数据，进行实时数据分析
  - 新硬件下的快速OLAP技术为RTDW提供了可行性

实时数据仓库技术

- 实时数据加载
  - 持续地将变化的数据从OLTP系统中读取到暂存表（Staging Tables），暂存表拥有和事实表相同的结构，同时存储着当前时段的数据拷贝。数据将按需要周期性地从暂存表拷贝到事实表，允许短暂的OLAP服务器暂停
- 外部实时数据缓存
  - 对于进行实时ETL的数据库服务器，将数据暂存在一个外部实时数据缓存中

实时数据仓库技术

- 准时制信息归并JIM
  - 通过对实时数据和与其有关联的历史数据进行归并分析，并结合事务日志提供的日志信息，确定当前所需要的实时数据。

- 反向准时制信息归并(Reverse JIM)
  - RJIM会将所需要的历史数据加载到存储实时数据的缓存中进行分析合并，对应的查询也将在数据缓存中执行。

典型系统

- SAP HANA
  - 采用完全内存存储，行列混合存储的内存计算技术，实现事务处理（OLTP）和分析处理（OLAP）的集成，将事务型数据库和分析型数据库合二为一。
- HyPer（原型系统，慕尼黑技术大学）
  - OLTP&OLAP混合系统
  - 高性能内存数据管理系统，采用快照模式，避免并发控制开销

小结

- 基于协处理器的OLAP技术
  - 基于GPU的OLAP技术
  - 基于PHI的OLAP技术
  - 基于FPGA的OLAP技术
- 实时数据仓库
  - 实时数据加载
  - 外部实时数据缓存
  - 准时制信息归并
  - 反向准时制信息归并