Enterprise Applications – OLTP and OLAP – Share One Database Architecture

Prof. Dr. Hasso Plattner

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

History of OLTP and OLAP

Motivation

- Today's data management systems are separated into **transactional and analytical** systems storing their data along rows or columns.
- Modern ERP systems are challenged by a **mixed workload** including OLAP-style queries, e.g.,
  - Dunning-run,
  - Available-to-promise, and
  - Real-time operational reporting

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

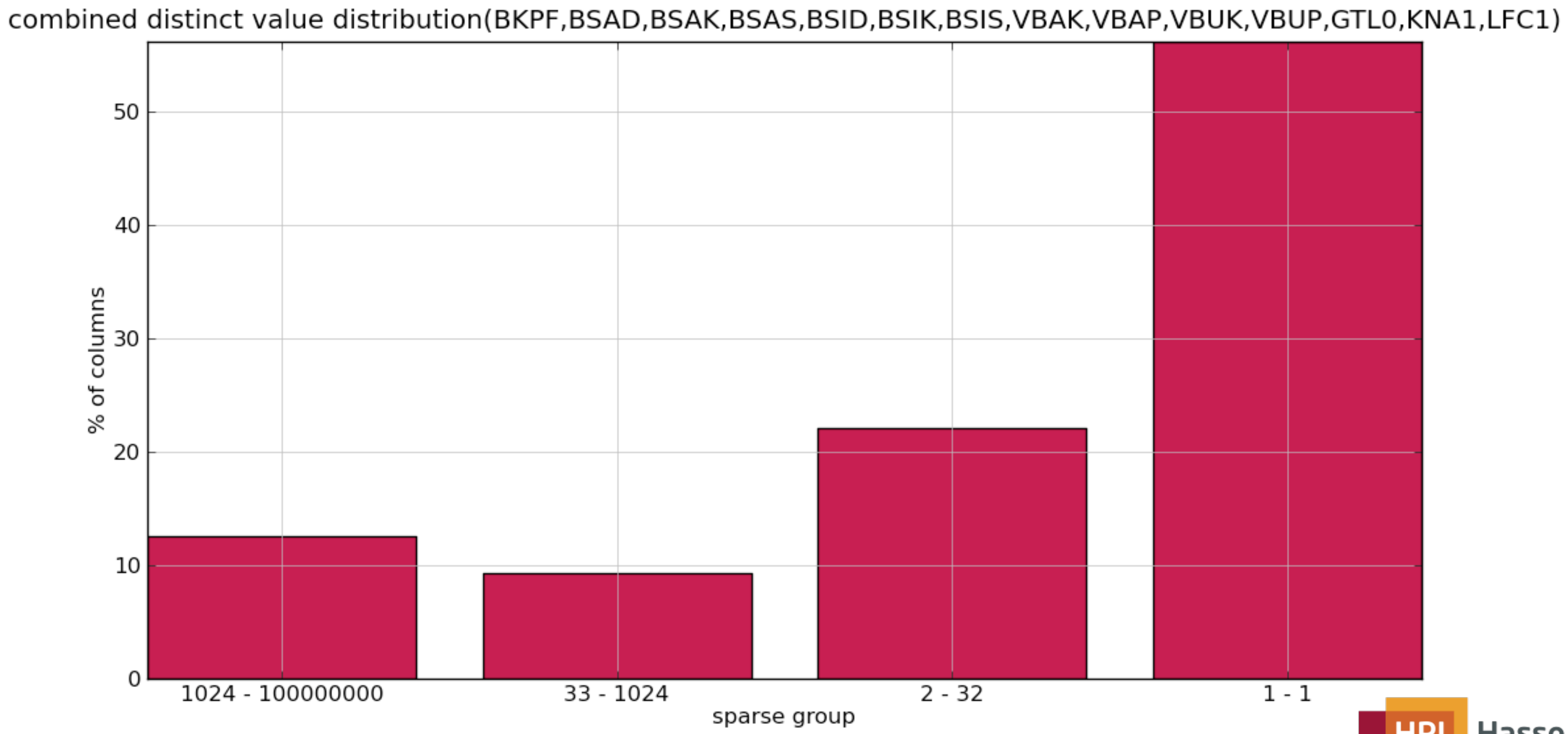
Enterprise Data is Sparse Data

- Many columns are not used even once
- Many columns have a low cardinality of values
- NULL values/default values are dominant
- Sparse distribution facilitates high compression

Sparse Data

55% unused columns per company in average

40% unused columns across all companies

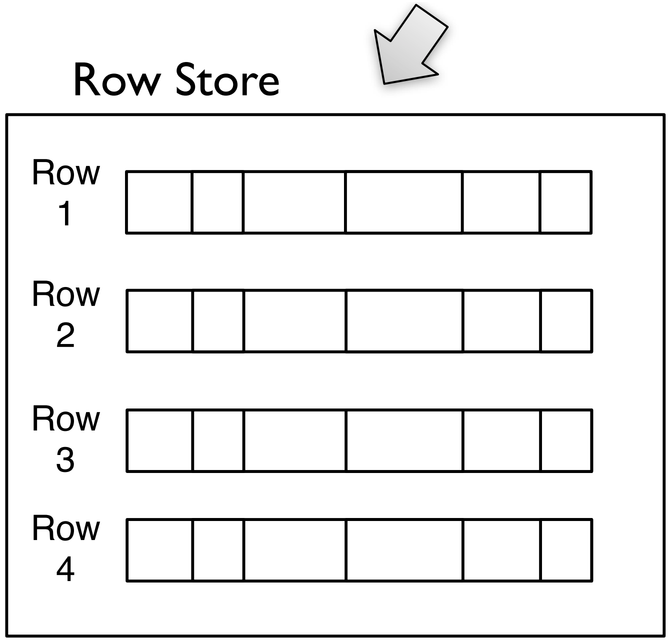


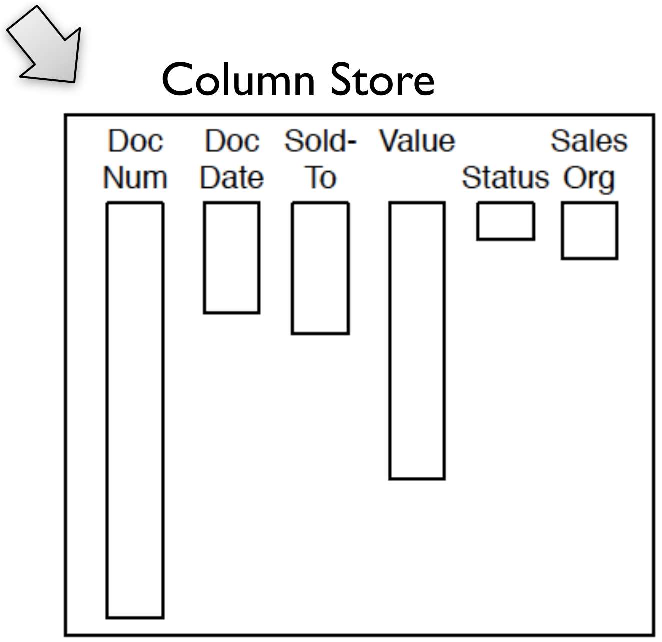
Column Store is Best Suited for Modern CPUs

Row vs. Column Store

(Compressed)

Document Number	Document Date	Sold-To Party	Order Value	Status	Sales Organization	...
95769214	2009-10-01	584	10.24	CLOSED	Germany Frankfurt	...
95769215	2009-10-01	1215	124.35	CLOSED	Germany Berlin	...
95779216	2009-10-21	584	47.11	OPEN	Germany Berlin	...
95779217	2009-10-21	454	21.20	OPEN	Germany Frankfurt	...



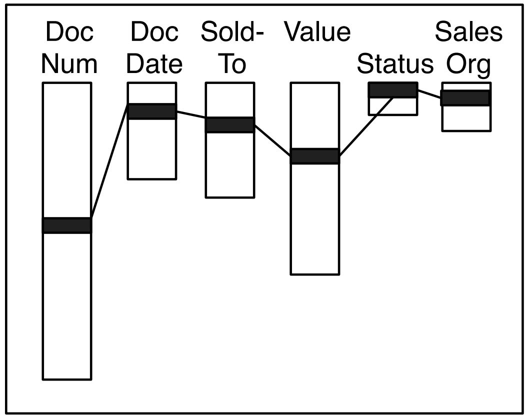


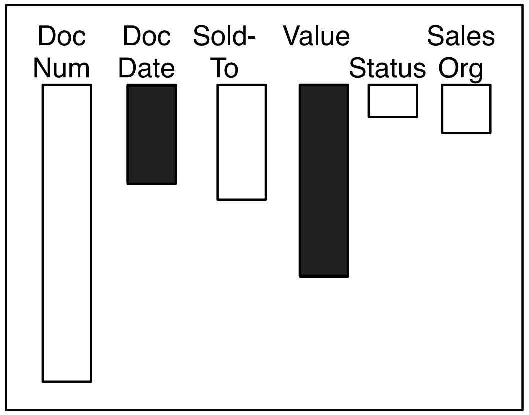
OLTP vs. OLAP Queries

SELECT *
FROM Sales Orders
WHERE Document Number = '95779216'

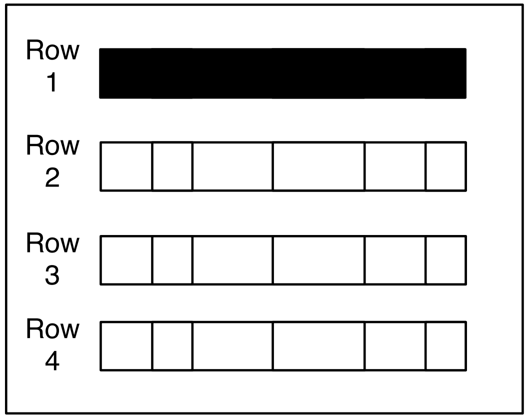
SELECT SUM(Order Value)
FROM Sales Orders
WHERE Document Date > 2009-01-20

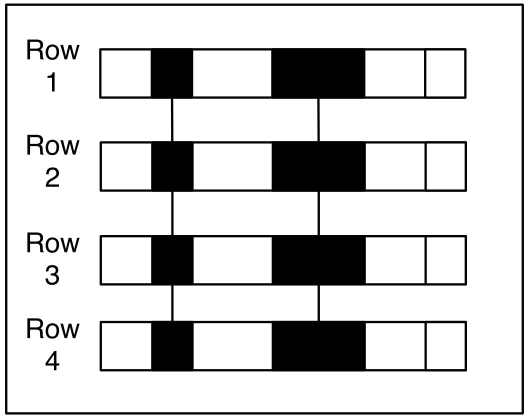
Column Store





Row Store





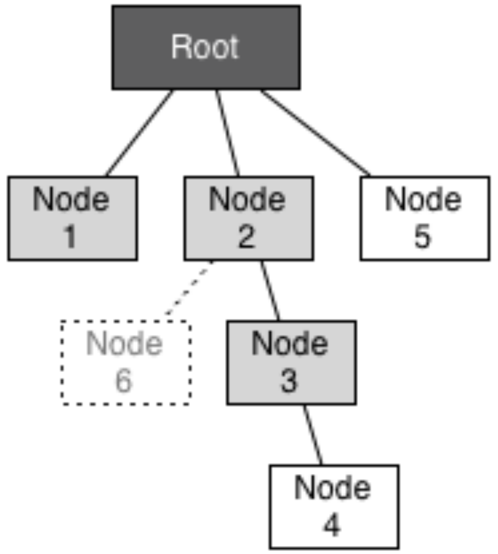
HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Column Stores for Modern Enterprise Applications

- Single object instance vs. set processing on attributes of nodes of objects
- Enterprise applications perform **set processing** (items for an order, orders for a customer)
- Bring application logic closer to the storage layer using stored procedures

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Object Data Guides



ID	Type	...	...	ODG
1	Order	...	...	(1,1,1,0,0)

1 = table is relevant

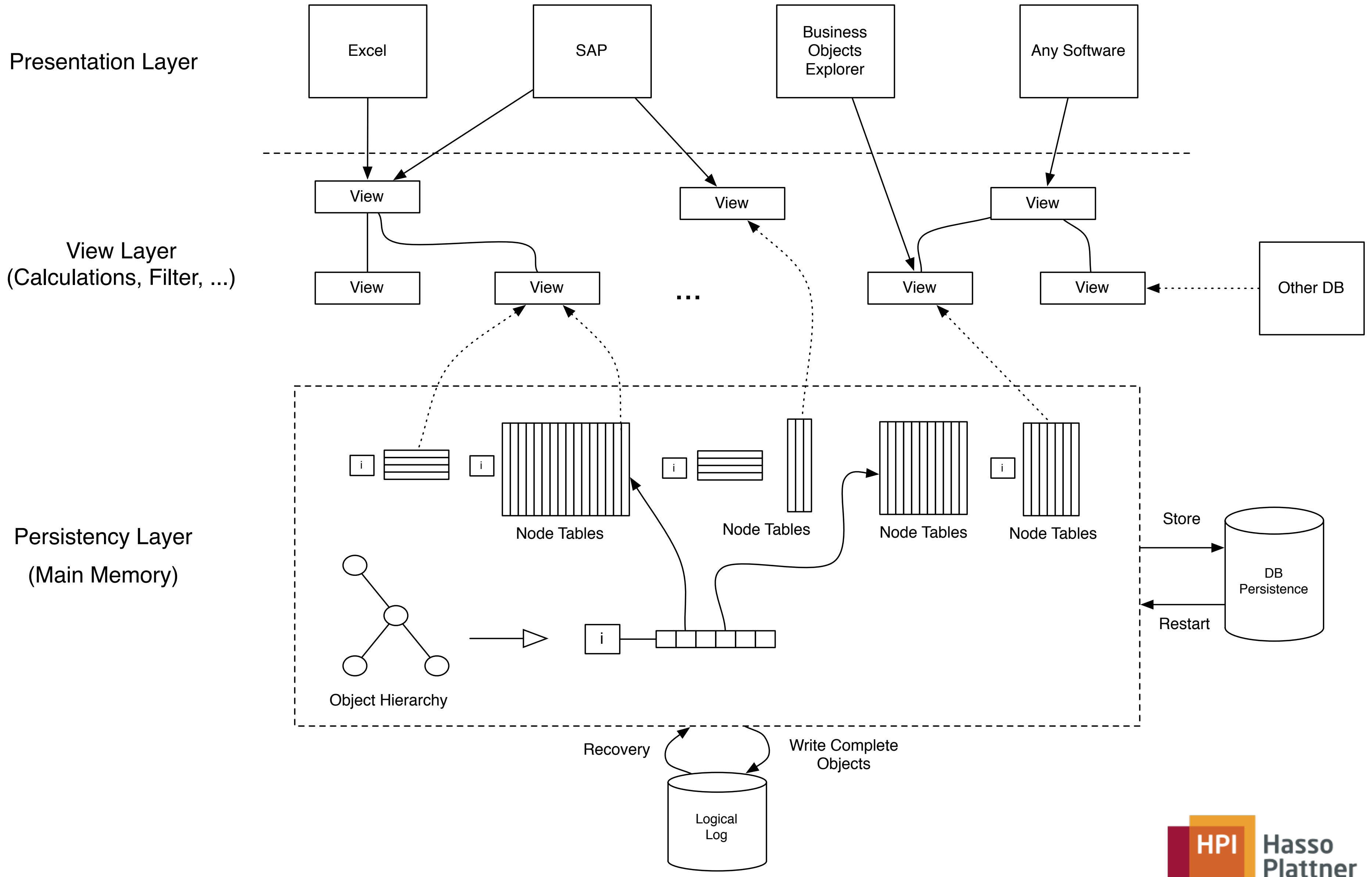
0 = table not relevant

- Enterprise systems make heavy use of objects - objects must be mapped to relations
- Often, objects are distributed sparsely over all tables representing nodes
- Relevant tables can now be queried in parallel
- When adding new tables, only add another bit

■	Root Table
▨	Used Table
□	Unused Table
⬚	New Table

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

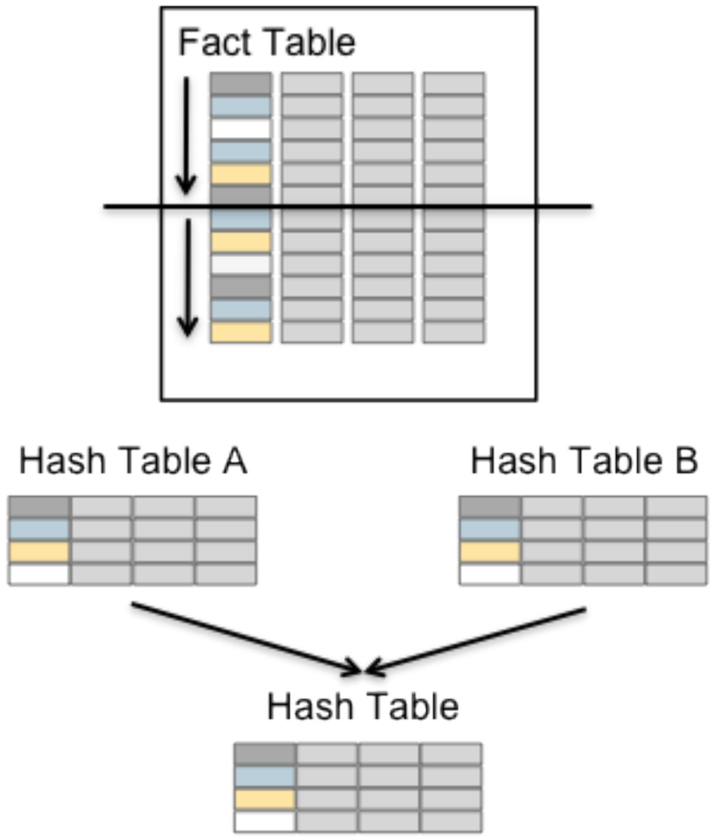
Dynamic Views



Multi-Core Usage

Parallelization in Column Stores

IntraOperator Parallelism



- Columns are optimal for dynamic range partitioning
- One sequential block can be easily split into many (as number of cores) blocks

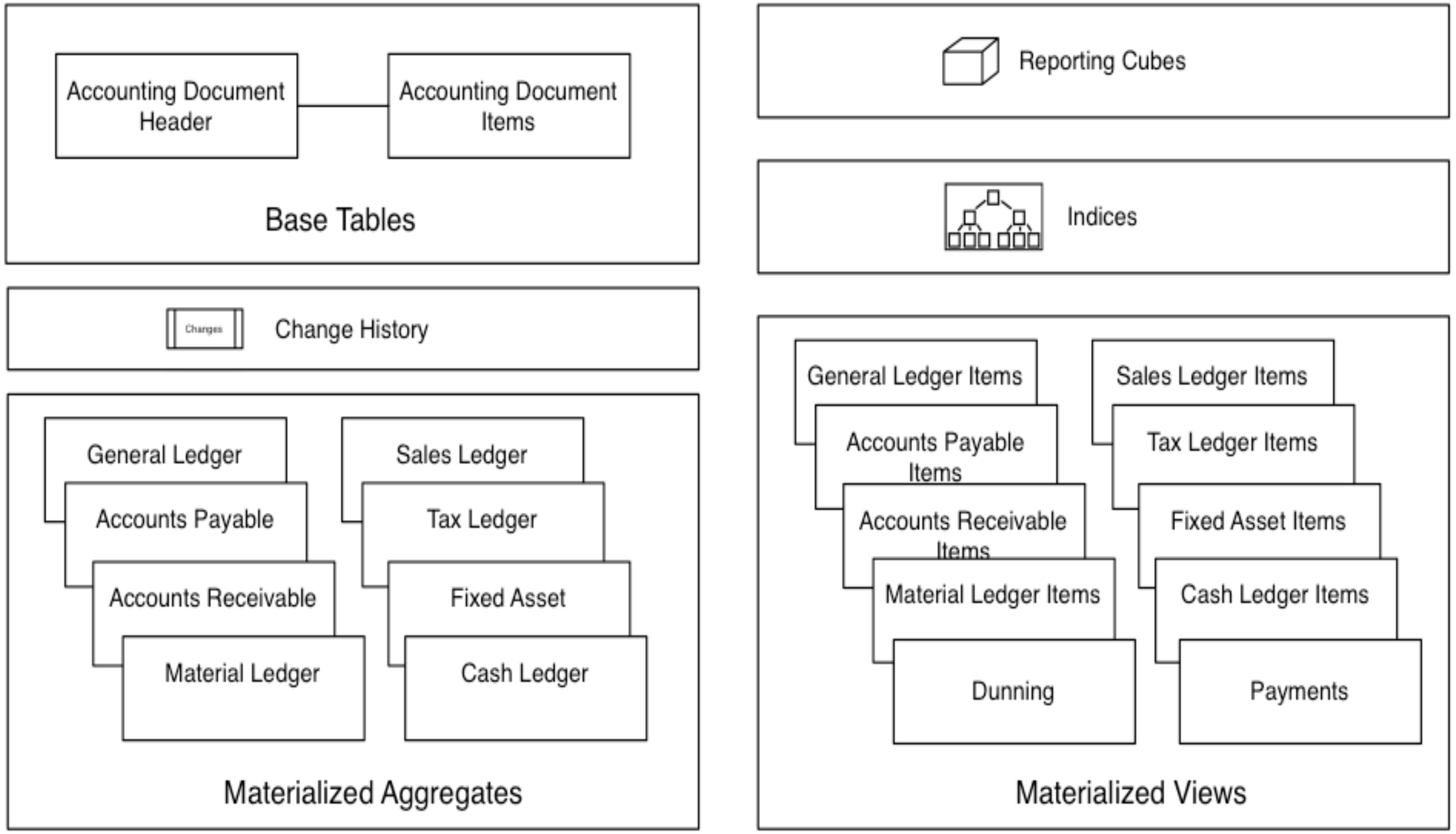
Stored Procedures

- New enterprise data management requires rethinking of how application logic is written
- Identify common application logic
- Rethink how applications are developed

Claim:

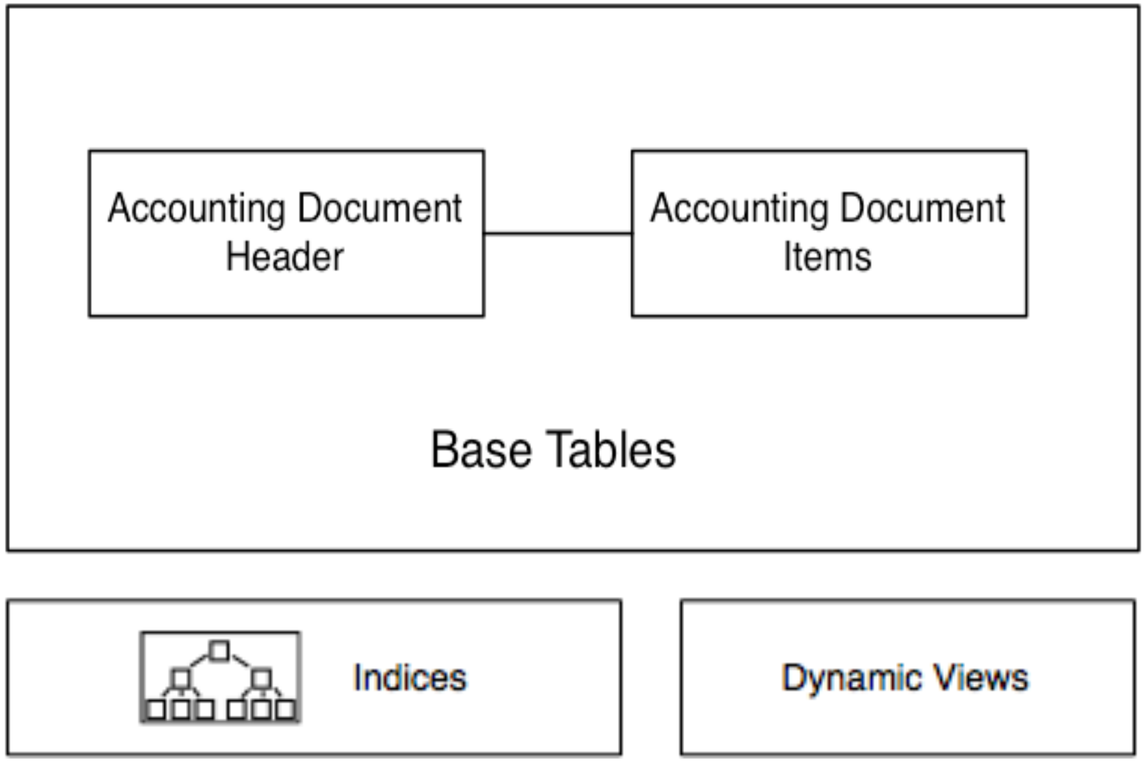
Columnar storage is suited for update-intensive applications

Nowadays Financials



Simplified Financials System (Target)

Only base tables, algorithms, and some indices



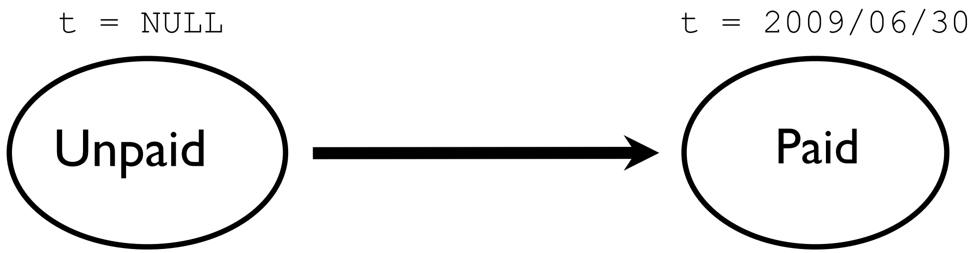
Insert Only

- Tuple visibility indicated by timestamps (*POSTGRES*-style time-travel*)
- Additional storage requirements can be neglected due to low update frequency (5 – 15%)
- Timestamp columns are not compressed to avoid additional merge costs
- Snapshot isolation
- Application-level locks

* Michael Stonebraker:
The Design Of The Postgres Storage System (1987)

Status Updates

- When updates of status fields are changed by replacement, do we need to insert a new version of the tuple?
- Most status fields are binary
- Idea: uncompressed in-place updates with row timestamp



Optimizing Write

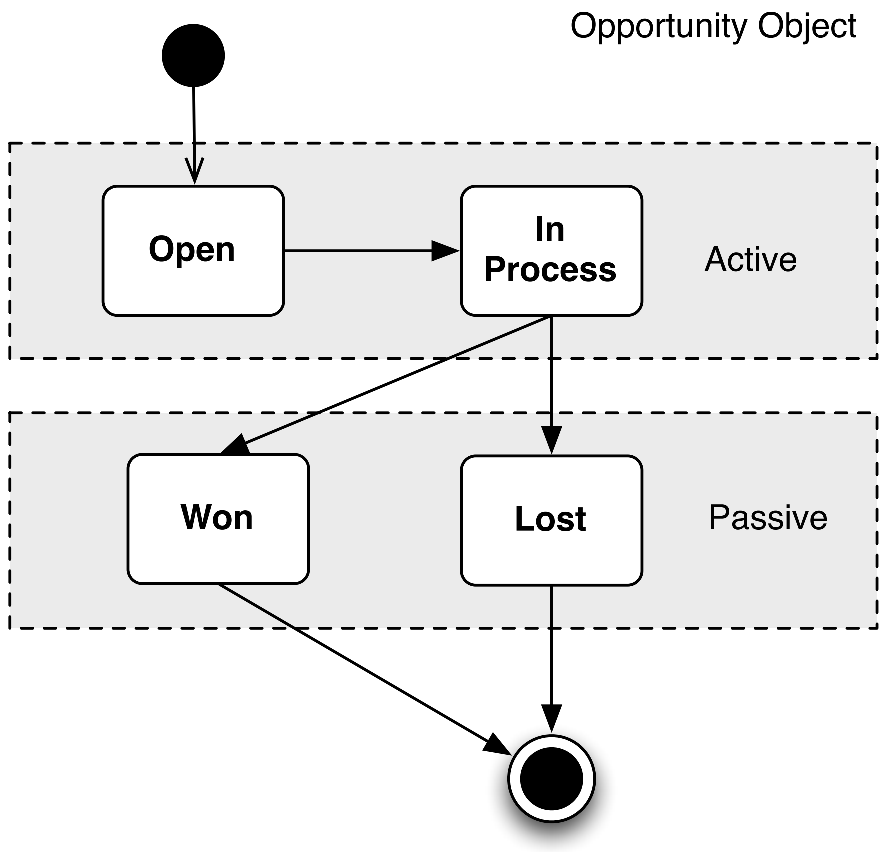
- OLTP workload requires many appends
- Instantly applying compression has a severe impact on the performance
- New values are written transactionally safe to a special write optimized storage
- Asynchronous re-compression of all values
- Current binary representation is stored on secondary storage (Flash) for faster recovery

Memory Consumption

- Experiments show a general factor 10 in compression (using dictionary compression and bit-vector encoding)
- Additional storage savings by removing materialized aggregates, save ~2x
- Keep only the active partition of the data in memory (based on fiscal year), save ~5x
- In total 100x is possible

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Aging = Partitioning



- Each enterprise object has a dedicated lifecycle - modeled using a state-transition diagram
- Events determine the status of an object
- Map states to partitions
- Multiple partitions = parallel queries

Memory Consumption (contd.)

- Arrays of 100 blades already available
- Next generation of rack servers will allow up to 2TB RAM
- 50 TB main memory will easily allow to cover the majority of SAP Business Suite customers

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Customer Study: Dunning Run in < 1s?

- Dunning run determines all open and due invoices
- Customer defined queries on 250M records
- Current system: 20 min
- New logic: **3 sec**
    - In-memory column store
    - Parallelized stored procedures
    - Simplified Financials

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Why?

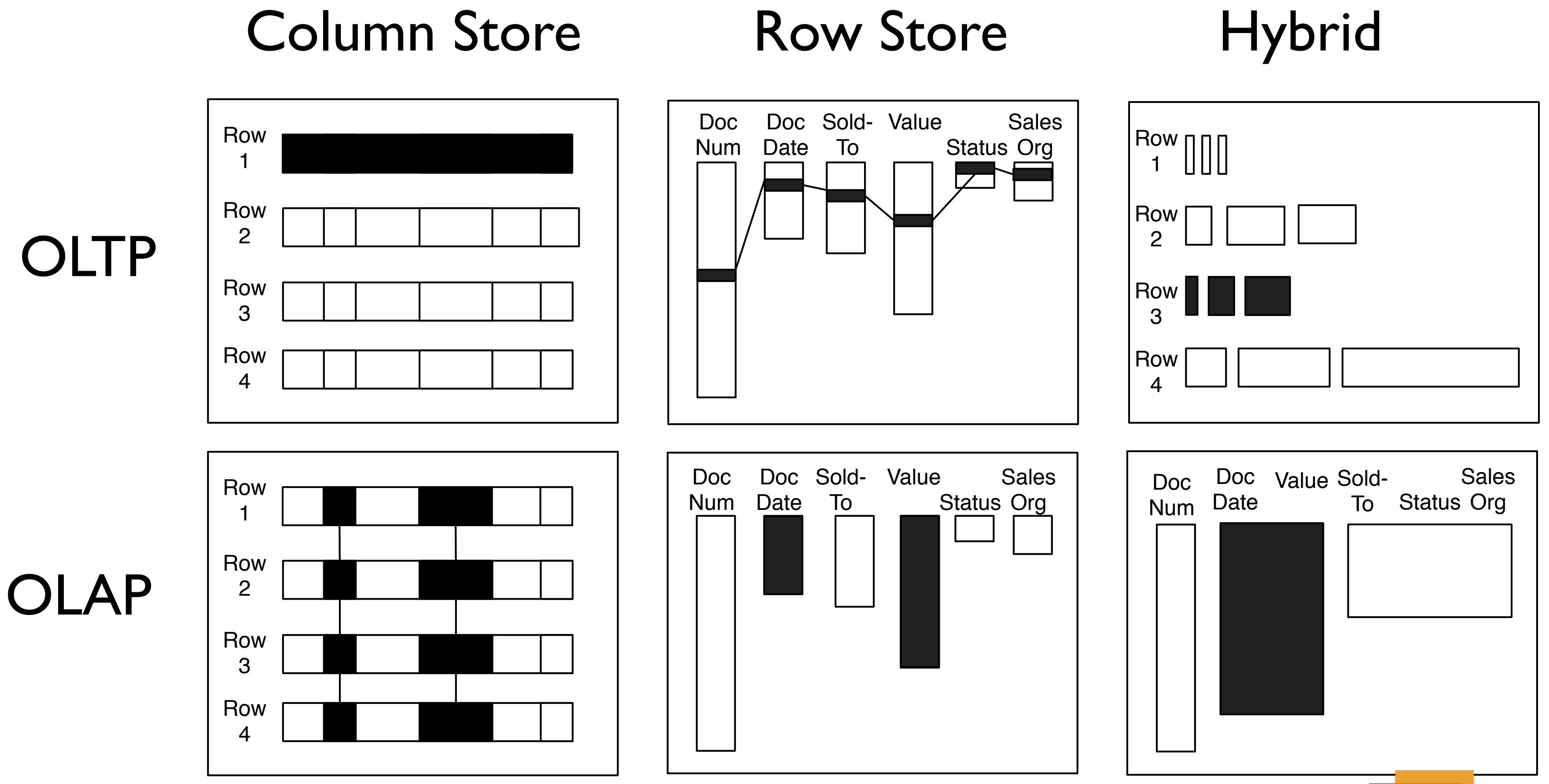
- Being able to perform the dunning run in such a short time **lowers TCO**
- Add more functionality!
- Run other jobs in the meantime! - in a multi-tenancy cloud setup hardware must be used wisely

Next: Hybrid Storage

- **Coarse**-grained hybrid - a single table can be either stored all rows or all columns
- **Fine**-grained hybrid - a single table will be vertically partitioned into groups of columns which are stored independently
- Enterprise workload is mixed workload and the hybrid provides best performance

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Hybrid Storage



Recovery in On-Demand Systems

- Recovery must be handled differently in on-demand scenarios
- Multiple tenants per system
  - Should all tenants be reloaded at the same time?
  - Prioritization inside a single tenant?
  - Use parallelization

Transition

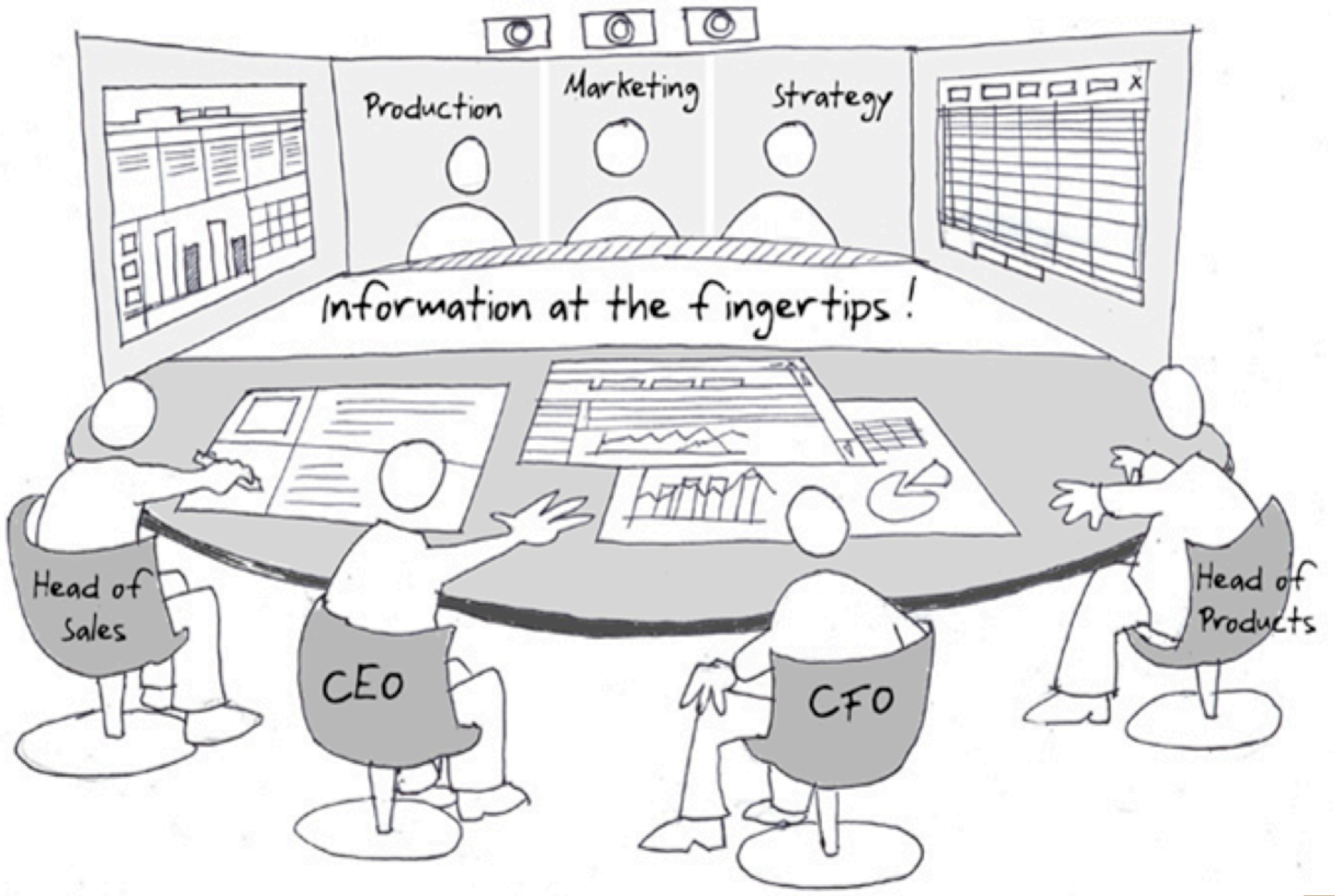
- Millions of "old" unoptimized lines of code at the customers' site
- Transition required
    - Row-store replacement
    - Part-for-part replacement with bypass
    - Transform row-store to column-store on the fly
    - Change of application code

Conclusion

- Technology improvements allow re-thinking of how we build enterprise apps:
  - A combined OLTP and OLAP system can share the same in-memory column store data base
  - Our experiments with real applications and data prove it
- Open research challenges:
  Disaster recovery, extension for unstructured data, life cycle based data management

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Outlook



Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam