IBM Data Retrieval Technologies: RDBMS, BLU, IBM Netezza, and Hadoop

Frank C. Fillmore, Jr.

The Fillmore Group, Inc.

Session Code: E13
Wed, May 06, 2015 (02:15 PM - 03:15 PM) | Platform: Cross-platform

Objectives

- Objective 1: Understand the technical differences between IBM data retrieval technologies.
- Objective 2: Be able to choose the most appropriate tool for a particular business use case.
- Objective 3: Articulate the value proposition and pluses/minuses of each solution.

Agenda

- The Fillmore Group
- Relational Database Management Systems (RDBMS)
- Netezza - Field-programmable Gate Array (FPGA)
- DB2 10.5 with BLU Acceleration - Columnar
- Hadoop - MapReduce
- Wrap-up, Q&A, Evaluations

"Rarely affirm, seldom deny, always distinguish."
St. Thomas Aquinas

THE FILLMORE GROUP
Relational Database Solutions

The Fillmore Group, Inc.

- Frank Fillmore, DB2 Gold Consultant, IBM Champion
- Kim May, IBM Champion
- Dozens of experienced, certified consultants
- Information Management Consulting Services
- IBM Authorized Training Partner
- IBM Information Management Software Reseller

IBM CHAMPION

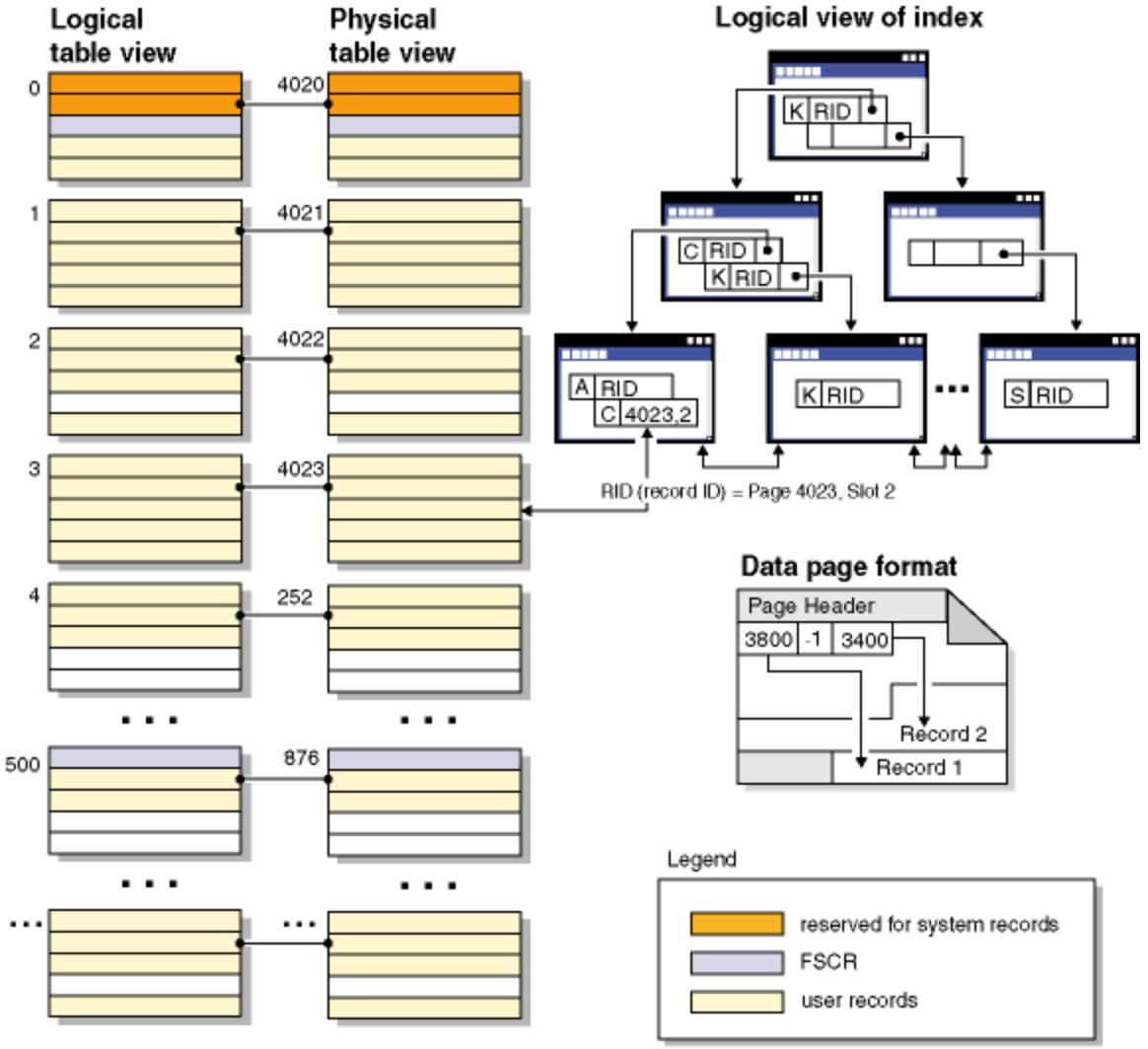
Relational Database Management System

Relational Database Management System

- The "foundation" characteristics
  - Data stored in ***tabular*** form (relations)
  - Schema stored in centralized repository
  - Strong datatyping
  - B-tree ***indexes*** used to create alternate access paths
  - Cost-based optimizer chooses access path
- Examples: DB2 for z/OS, LUW

Index review

- Root page
- Non-leaf page
- Leaf page
- Key
- Row ID (RID)
- Data page
- Slot



Relational Database Management System

- Typical use cases
  - Online transaction processing (OLTP)
  - E-Commerce
- Extended use cases
  - Data warehousing
  - Online analytical processing (OLAP)

Premier Business Partner IBM

Issues and limitations

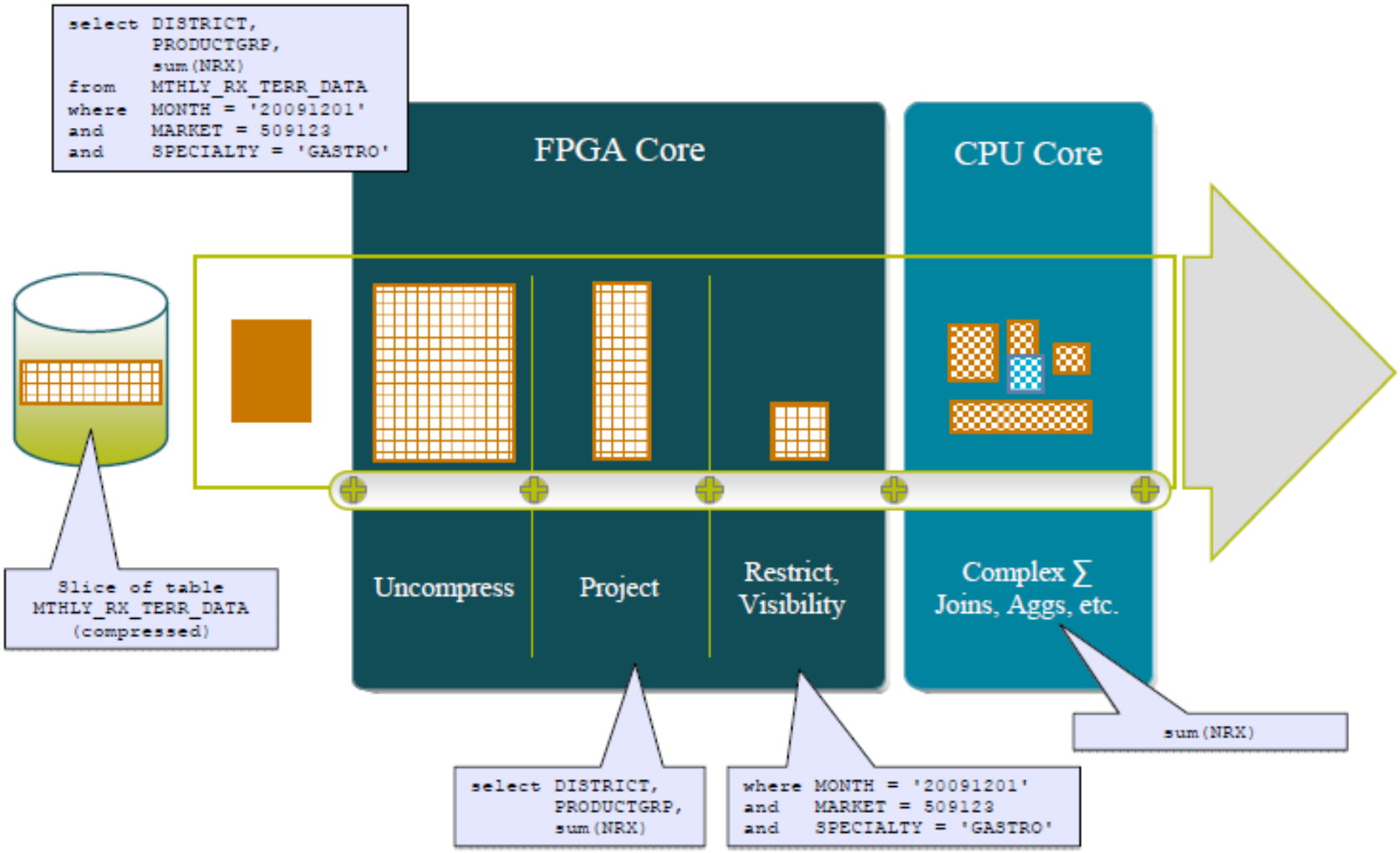
- Rigid schema and strong datatyping diminish flexibility and ***responsiveness*** to change
- Data is transferred in units of ***pages*** from disk to memory for processing
- At billions of rows in a table, B-tree index I/O becomes expensive, impacting ***response*** times

Netezza

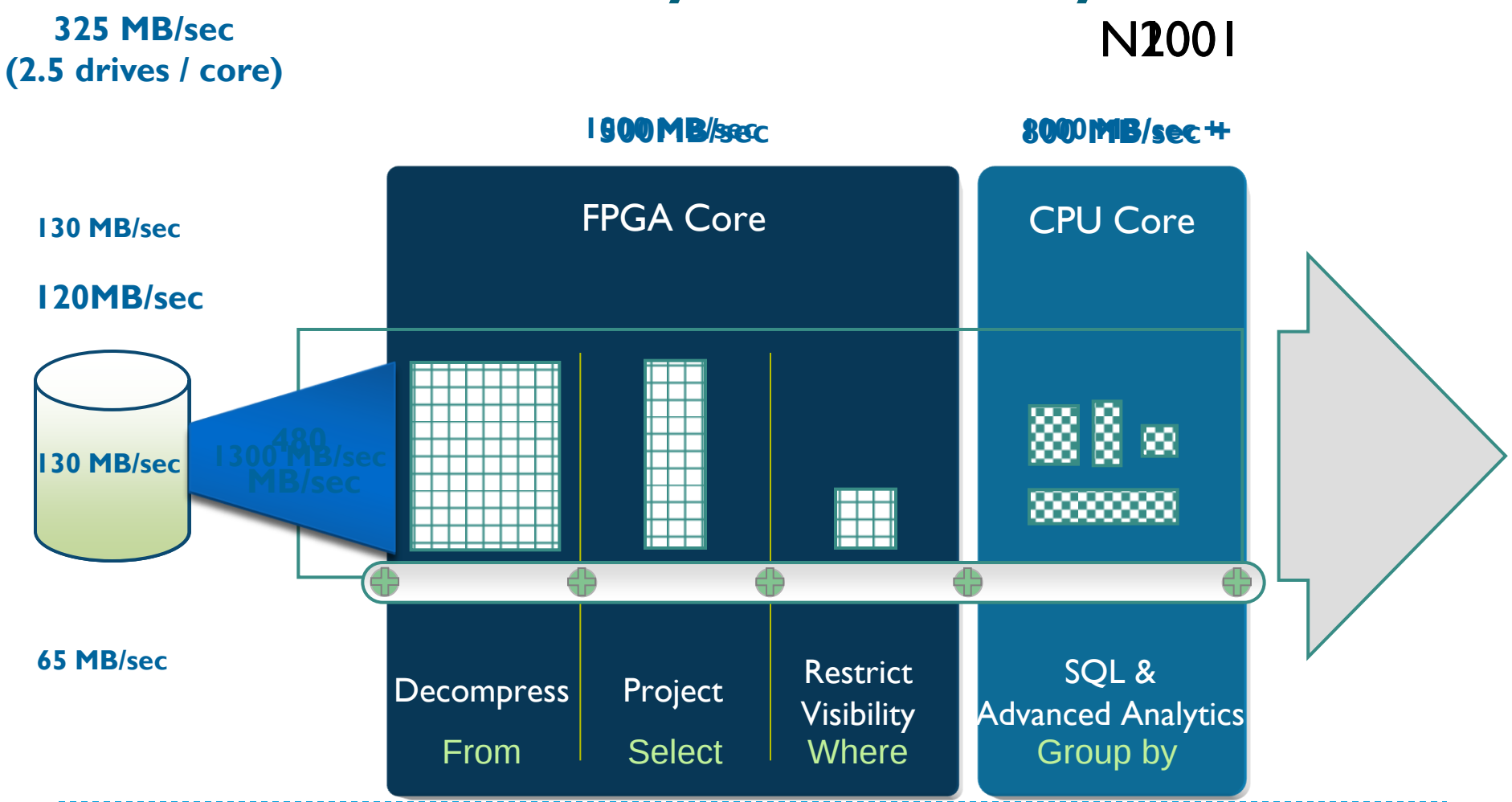
Netezza – aka PureData System for Analytics

- Appliance: integrated database, server, storage
- Key concepts
  - Field-programmable Gate Array (FPGA)
  - Data slice
  - Distribution Key
  - Zone Map

Premier Business Partner


select DISTRICT,
PRODUCTGRP,
sum(NRX)
from MTHLY_RX_TERR_DATA
where MONTH = '20091201'
and MARKET = 509123
and SPECIALTY = 'GASTRO'
FPGA Core
CPU Core
Slice of table
MTHLY_RX_TERR_DATA
(compressed)
Uncompress
Project
Restrict,
Visibility
Complex ∑
Joins, Aggs, etc.
sum(NRX)
select DISTRICT,
PRODUCTGRP,
sum(NRX)
where MONTH = '20091201'
and MARKET = 509123
and SPECIALTY = 'GASTRO'

PureData System for Analytics



Just say 'No'tezza

- No data modeling
- No ***indexes***/MQTs/MDCs
- No dbspace/tablespace sizing and configuration
- No page/block/extent sizing and configuration for tables
- No redo/physical/journaling/logical log sizing and configuration

- One partitioning strategy: ***hashing algorithm***

Data slices: how Netezza stores data on disk

- Data on the integrated drive is called a "data slice"
- Tables are split across data slices
- Netezza stores data on disk in groups according to row
- Data is compressed according to identical columnar values (***columnar compression***)
- Hashing algorithm distributes user data to primary partitions on each disk.
- Data is also mirrored

Distribution Keys

- Commonly joined tables (e.g. Dimension and Fact table of a star schema) use the same Distribution Key
- ***Collocated joins*** - rows from different tables with same key stored on the same data slice

```
CREATE TABLE customer
(    c_custkey        integer,
     c_name           varchar(25),
     c_address        VARCHAR(40),
     c_nationkey      integer,
     c_phone          character(15),
     c_acctbal        numeric(15,2),
     c_mktsegment     character(10),
     c_comment        VARCHAR(117)
) DISTRIBUTE ON ( c_custkey );
```

```
CREATE TABLE orders
(    o_orderkey             integer,
     o_custkey        integer,
     o_orderstatus    character(1),
     o_totalprice     numeric(15,2),
     o_orderdate      date,
     o_orderpriority  character(15),
     o_clerk          character(15),
     o_shippriority   integer,
     o_comment        VARCHAR(79)
) DISTRIBUTE ON ( o_custkey );
```

Zone Maps

- An extent is the smallest unit of Netezza disk allocation
- A Zone Map is an internal structure to show the range (***minimum*** and ***maximum***) of values within each extent
- During scans, zone maps are used to reduce I/O by ***skipping*** extents that don't qualify based on predicates
- Zone maps are internal to the system and require ***no administration***
- Especially useful in ever-ascending key data warehouses (e.g. date) when query predicates typically use that key

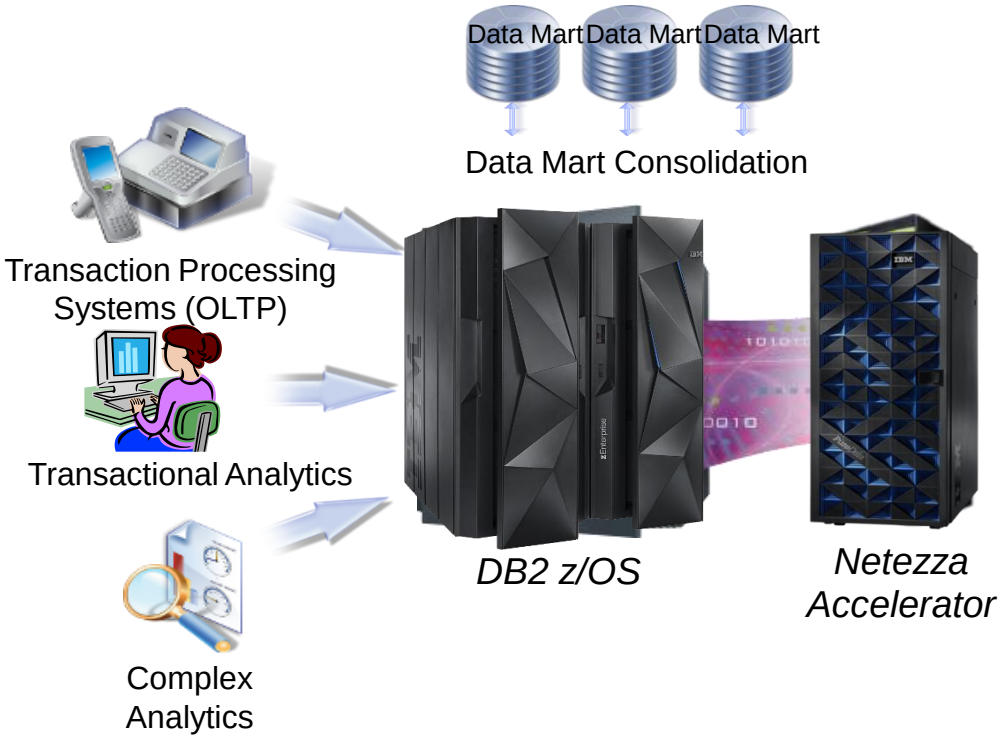
Zone Maps - example

Issues and limitations

- Rigid schema and strong datatyping diminish flexibility and ***responsiveness*** to change
- Netezza is not optimized to process SQL Data Manipulation Language (DML)
  - INSERTs
  - UPDATEs
  - DELETEs

  i.e. OLTP transactions

IBM DB2 Analytics Accelerator (IDAA): OLTP and Netezza *hybrid*



Decision to offload from DB2 to IDAA

- Simple, rule-based approach to figure out if a query is ***transactional*** (should be executed by DB2 directly) or ***analytical*** (should be offloaded to IDAA)
- Check is performed before the DB2 cost model is created
- Heuristics can be influenced by DB2 profile tables and they perform decisions based on DB2 statistics
- Keep the query on DB2 if
  - One Fetch Access
    - Single row to receive (Index access)
    - Aggregates like MIN MAX which can be answered by index
    - Equal unique access of joined tables
  - Query does not contain any grouping, aggregation and query does not contain predicate
  - Accessing small tables only

Checkpoint

- Q: Hashed Distribution Keys and collocation of data is similar to what DB2 for LUW technology?
- A: Data Partitioning Feature (DPF)
- Q: Zone Maps are *somewhat* similar to what DB2 index component?
- A: Leaf pages

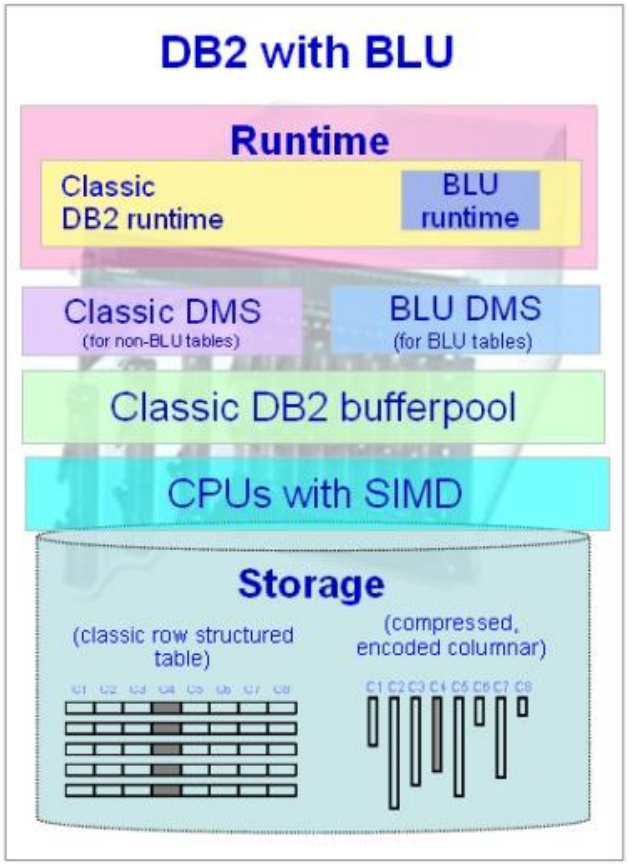
"There's no such thing as a free lunch."

Milton Friedman

DB2 10.5 BLU Acceleration

DB2 10.5 BLU Acceleration

- Memory optimized
  - In-memory ***columnar*** processing
  - Dynamic data movement from storage (no LRU)
- Actionable Compression
  - Patented ***compression*** technique that preserves order so that the data can be used without decompressing (column cardinality)
- Parallel Vector Processing
  - Multi-core and Single Instruction Multiple Data (SIMD) parallelism
- Data Skipping
  - Skips unnecessary processing of irrelevant data



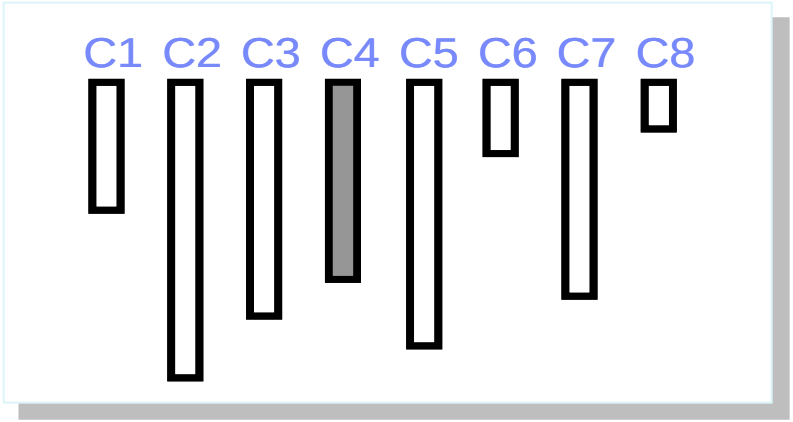
Premier Business Partner IBM

DB2 10.5 BLU Acceleration

- DB2 10.5 BLU Acceleration is a ***hybrid*** that supports mixed OLTP and ***analytic*** workloads
- Set DB2_WORKLOAD registry variable to ANALYTICS
- Column-organized tables will be the default table type
- Sets default page (32KB) and extent size (4) appropriate for analytics
- Data is always automatically compressed - no options
- For mixed table types can define tables as **`ORGANIZE BY COLUMN`** or **`ROW`**
- Utility to convert tables from row-organized to column-organized (**`db2convert`** utility)

Column-oriented storage

Columns stored separately and packed in different buffers in memory

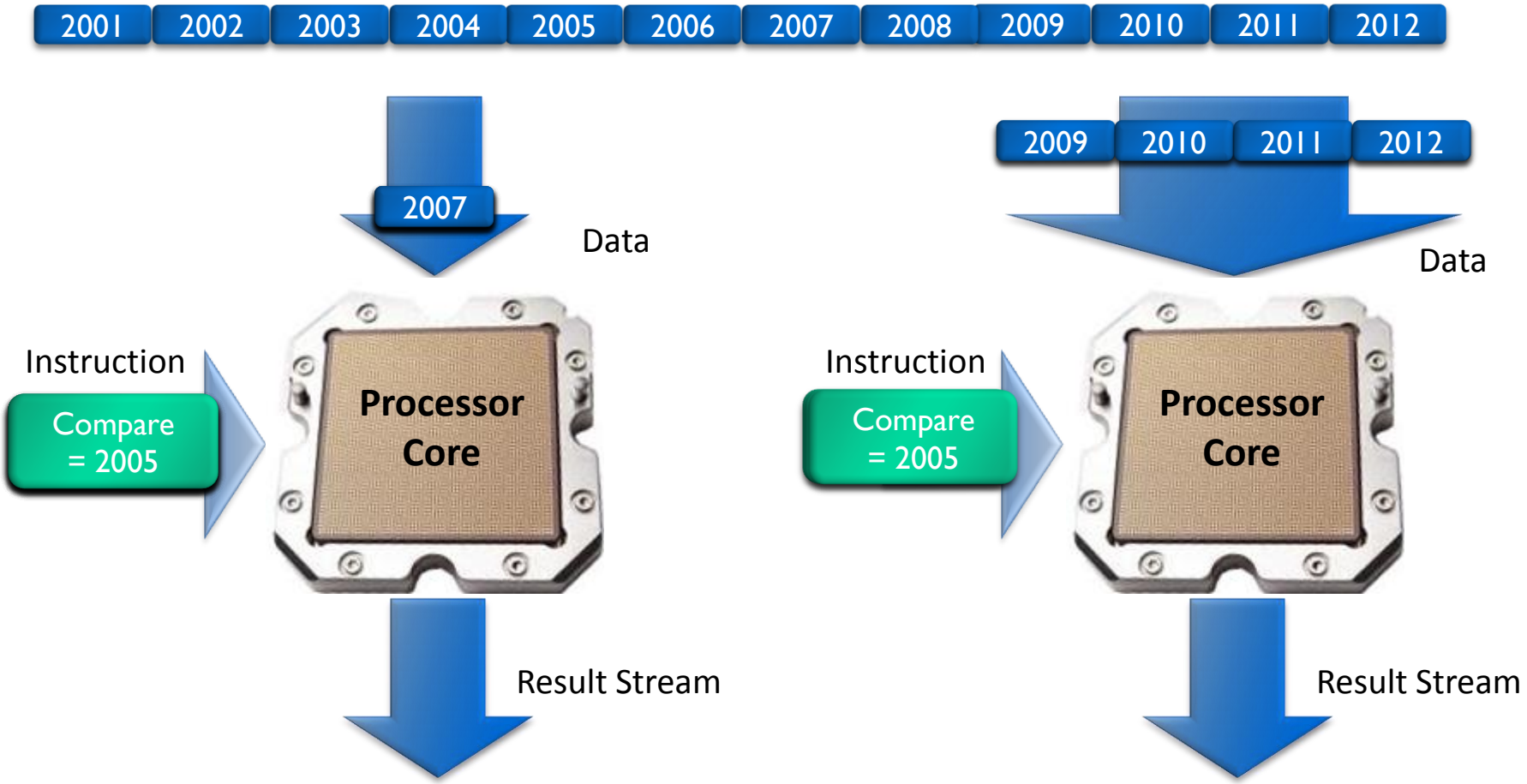


SELECT C4 ... WHERE C4=X

Consumes I/O bandwidth memory buffers and memory bandwidth only for C4

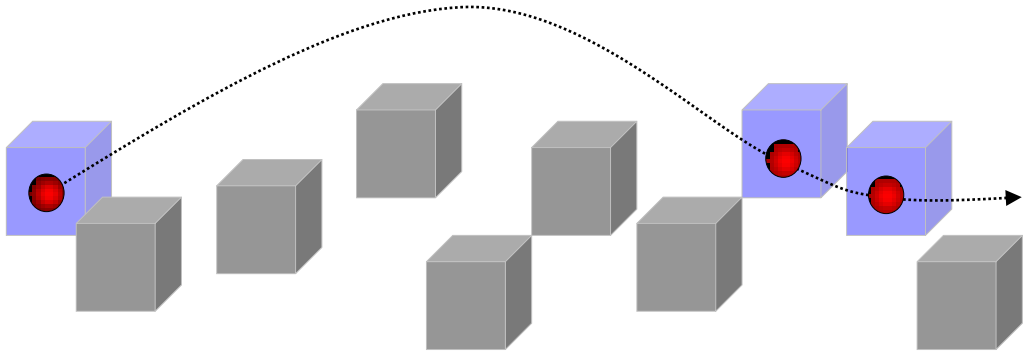
Premier Business Partner IBM

Single Instruction Multiple Data (SIMD)



Data skipping

- Retrieve only data that meets predicate criteria
- "Synopsis" automatically created and maintained as data is LOADed or INSERTed (***no administration***)
- Persistent storage of ***minimum*** and ***maximum*** values for sections of data values



Issues and limitations

- Rigid schema and strong datatyping diminish flexibility and ***responsiveness*** to change
- Columnar tables optimized for read-mostly

Checkpoint

- Q: Which two technologies surveyed so far focus on analytics?
- A: Netezza and DB2 10.5 BLU Acceleration
- Q: Why do the following matter?
  - number of columns in a row
  - ratio of columns retrieved to columns in the row
- A: If there are many columns in a row and most of them are retrieved most of the time, **`ORGANIZE BY ROW`** might be more efficient

Checkpoint

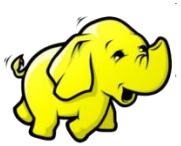
- DB2 10.5 is *similar* to IDAA as a ***hybrid*** platform
- Factors to consider when choosing between standalone Netezza and integrated DB2 10.5 BLU Acceleration
  - Business SLAs for ETL processing between OLTP and analytics databases
  - Existing skills
  - Appliance (Netezza) vs. hardware/software integration (DB2)
  - BLU is a single-node solution – 10Tb is a practical limit
  - Relative performance?

Checkpoint

- Q: Data skipping sounds like which Netezza feature?
- A: Zone Maps

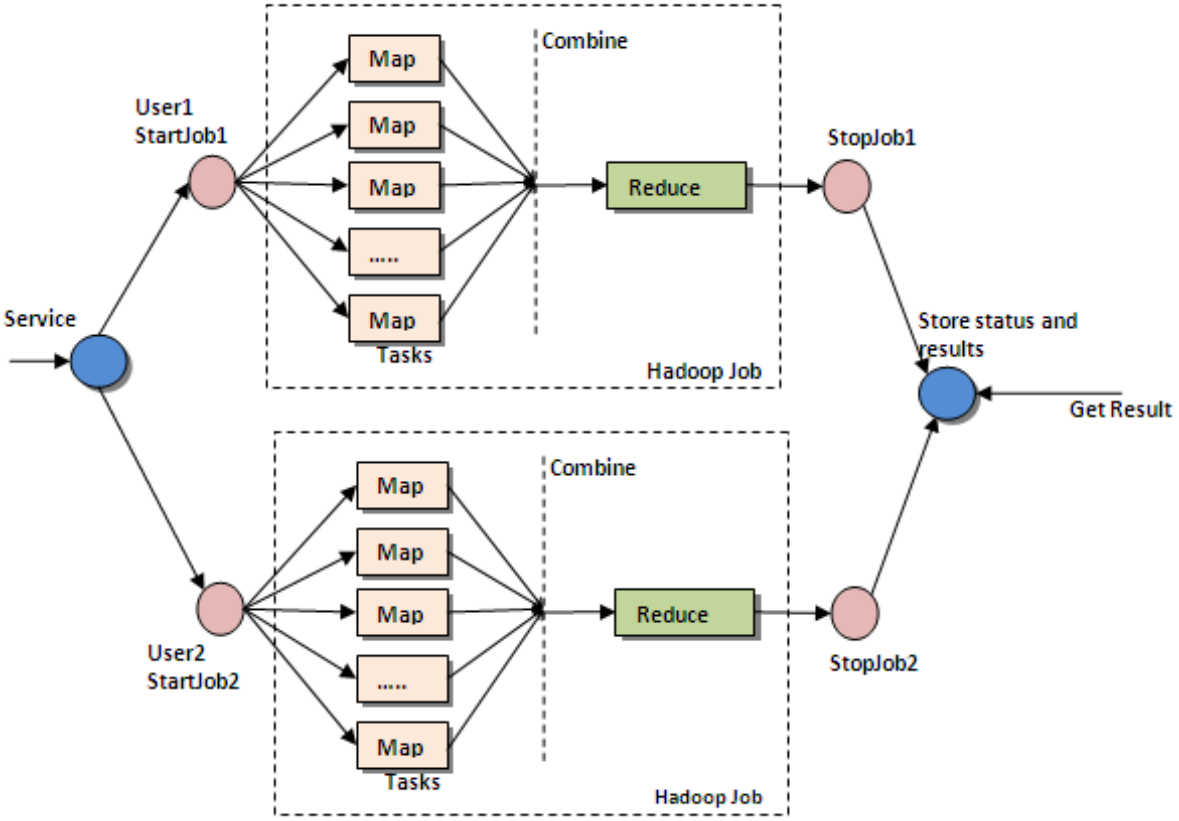
Hadoop

Hadoop



- IBM Hadoop distribution: **InfoSphere BigInsights**
- What is Hadoop?
  - Inspired by Google MapReduce
  - Batch-oriented, read intensive
  - Shared-nothing architecture
  - Exploits cheap commodity hardware
  - Hadoop Distributed File System (HDFS)
    - Provides high aggregate bandwidth across a cluster
    - Manages data across individual nodes in cluster

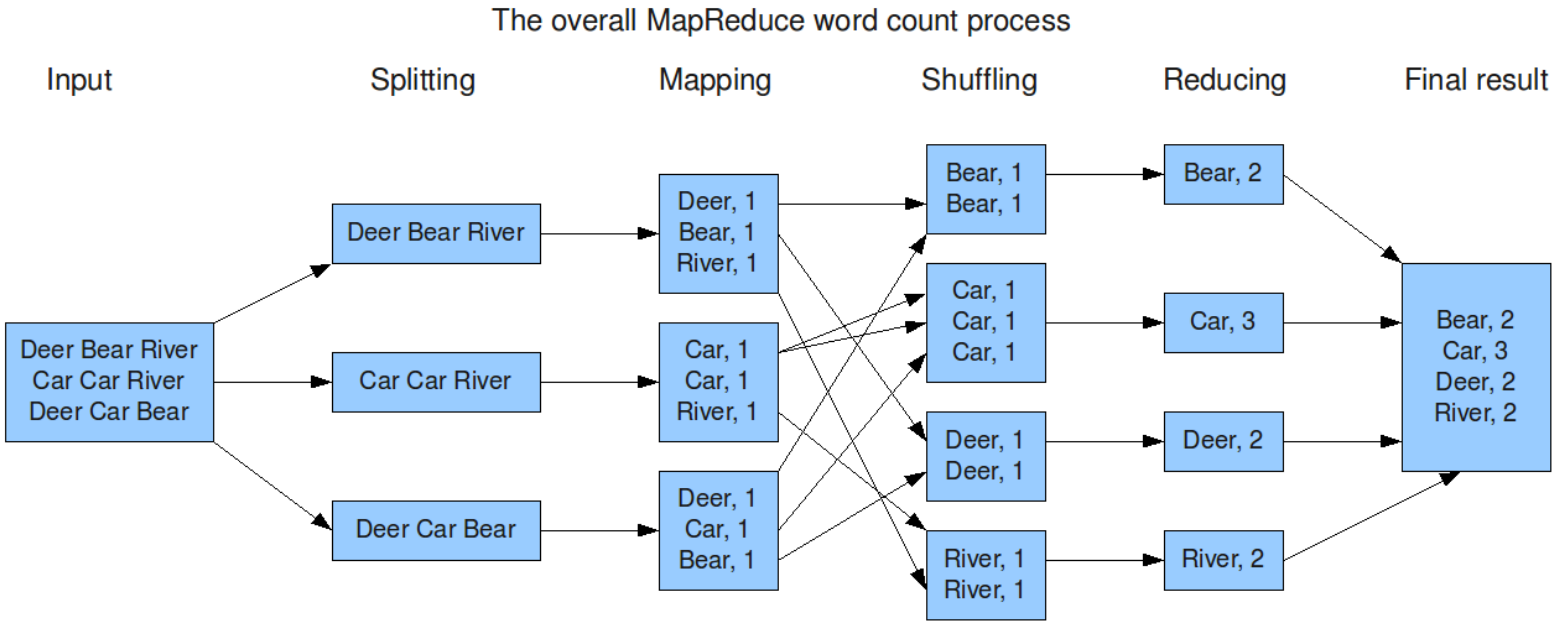
MapReduce process flow



Why Hadoop?

- Disk latency (speed of reads and writes) has not improved on an orders-of-magnitude scale as have CPU, RAM, disk capacity
- Volume, variety, and velocity of BigData
  - Solid state disk (SSD) is fast but *very* expensive
  - "Schema on read" flexibility for processing unstructured data
  - Store-and-access paradigm can't accommodate real-time processing
    - Google search advertisement placement
    - Stock trades
    - Sentiment analysis of Tweets

Hadoop conceptual example



Issues and limitations

- No update in place – insert and delete only
- Fault-tolerant, but potential for data loss
- Simple file compression
- Immature technology relative to others surveyed
- Not enough skills in the marketplace to meet demand
- Lots of customer integration of software stack ***or*** lose the commodity hardware cost benefit
- It takes 10 – 20 seconds to spawn a MapReduce job

Checkpoint

- Q: MapReduce distribution of portions of the data to different nodes (Map) and then reassembling the answer set (Reduce) is similar to what DB2 for LUW technology?
- A: Data Partitioning Feature (DPF)
  - DPF distributes data to nodes at load, then the data is queried repeatedly – favors response time over throughput
  - MapReduce distributes the data at query run time – favors throughput over response time

"When a farmers buys another 40 acres, he doesn't get a bigger ox, he gets a second ox." *Admiral Grace Hopper*

Audience Use Case Open Discussion

- What data management requirement do you need to support?
- Which technology of the four surveyed would you choose and why?

Contacts

- Kim May
  - kim.may@thefillmoregroup.com
  - twitter.com/KimMayTFG
  - www.linkedin.com/pub/kim-may/4/462/84
- Frank Fillmore
  - frank.fillmore@thefillmoregroup.com
  - twitter.com/ffillmorejr
  - www.linkedin.com/pub/frank-fillmore/6/597/9a6/
  - tinyurl.com/ChannelDB2
  - Flipboard for iPad, iPhone, Android: "BigData"

Resources

- IBM DB2 Express-C
- How to Handle the "V" for Volume in Big Data on DB2 for z/OS
- DB2 10.5 with BLU Acceleration Flashbook
- Optimizing analytic workloads using DB2 10.5 with BLU Acceleration
- BigData – Hadoop
- IBM InfoSphere BigInsights

Frank C. Fillmore, Jr.

The Fillmore Group, Inc.

frank.fillmore@thefillmoregroup.com

Session: E13

Title: IBM Data Retrieval Technologies: RDBMS, BLU, IBM Netezza, and Hadoop

Please fill out your session evaluation before leaving!

The March to Valley Forge. William B. T. Trego (1883)