Architectures for Big Data Analytics

A database perspective

Fernando Velez

Director of Product Management

Enterprise Information Management, SAP

June 2013

Outline

Big Data Analytics Requirements

Spectrum of Analytic Computations

Solution Options

Summary

Big Data Analytics

The process of examining large amounts of data of a variety of types to uncover hidden patterns and unknown correlations to improve business performance or mitigate risk

Ebay Example: detect correlation between increased postal rates and decreased web traffic in time to launch promotion countermeasures

Requirements

- Scalability on terabyte range (1 – 1000) volumes
- “Deep” analytical computation
- Relational and non-relational data
- Response-time allowing interactivity
- Data management (querying, fault tolerance, backup, archiving, security, data quality…)

Analytic computations, from simple to deep

SQL (92)

– *For each employee, give the covariance of salary and bonus over 3 years*

SQL 99 Analytic extensions

– *For each employee, give the count of employees earning half less than them*

Packaged algorithms (clustering, linear regression, …)

– *Tell me which customers are likely to be early adopters of my new product*

– *Predict and fit to past data next month's sales by business unit and per geography*

Linear algebra packages (ScaLAPACK)

– *Discover communities among users of products sold to my enterprise customers*

Analytic programming language (R, MatLab, SAS…)

– Complete environment for data analysis including thousands of add-in packages

All require regularity in the data –at odds with *Big Variety*

Solution options

Analytic PL

Relational DBMS

Map-Reduce

Analytic PL + (Relational DBMS or Map-Reduce)

Redesign New DBMS

Analytic PL: R (and Matlab et. al.)

State of the art *deep analytics*

- Open source, high-level analytic & stats functions, matrix algebra, graphics

Standard file formats for data sharing across PL platforms

- HDF5

Limitations

Load data from file (or ODBC), operate entirely in main memory

Little or no parallelism

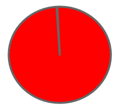
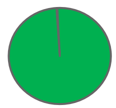
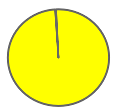
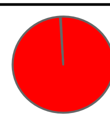
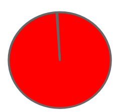
Little or no data management

R community working to address these needs

R community very active in parallelizing R across multiple CPUs or nodes

Revolution offers external memory implementation on proprietary file format

Evaluating against requirements

	Analytic PL
Scalability	
Deep analytics	
Relational & non-relational data	
Response time	
Data Management	

Relational DBMSs

The ultimate data management engines

Scalability and response time predicated mostly on

- The architecture for parallel processing
  - (incl. quality of data placement, quality of parallel algebra implementation)
- Row storage versus column storage

Analytics

Simple (SQL level) : good support

Deep analytics : much harder

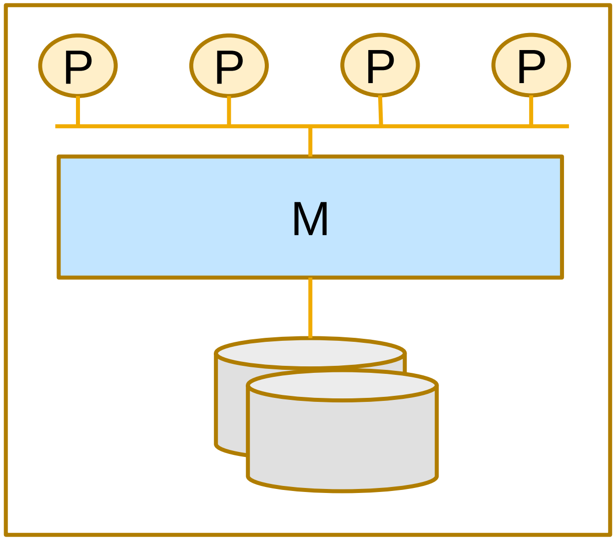
- Linear algebra on large matrices via SQL UDFs is hard to parallelize well
- But packaged algorithms becoming available as stored procedures

Big Variety is an issue

- Typical architecture is to “ETL data” from MR systems (e.g., Hadoop)

Architectures for parallel processing – 1

Shared Memory



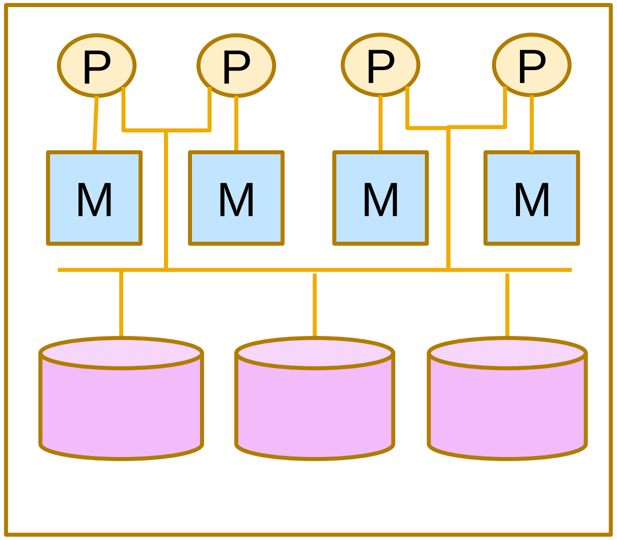
Shared Memory

The bus is the bottleneck beyond 32 processors – in practice used for 4-8 proc. machines

Architectures for parallel processing – 2

Shared Memory

Shared Disk



Shared Memory

The bus is the bottleneck beyond 32 processors – in practice used for 4-8 proc. machines

Shared Disk

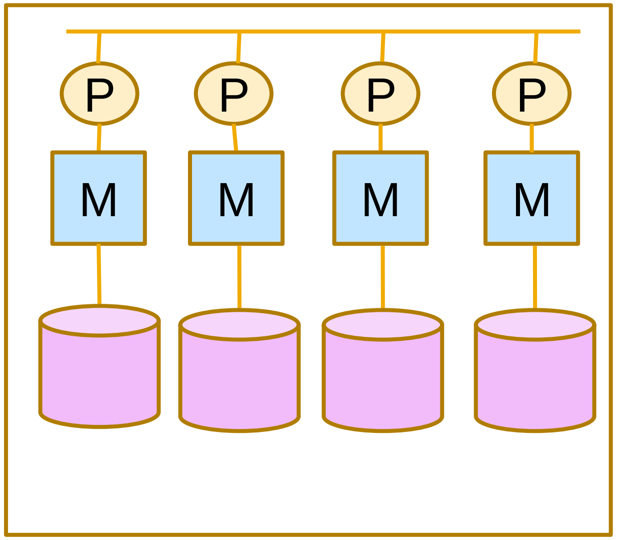
Slower inter-CPU communication, better fault-tolerance, bottleneck pushed to about 100 proc.

Architectures for parallel processing – 3

Shared Memory

Shared Disk

Shared Nothing



Shared Memory

The bus is the bottleneck beyond 32 processors – in practice used for 4-8 proc. machines

Shared Disk

Slower inter-CPU communication, better fault-tolerance, bottleneck pushed to about 100 proc.

Shared Nothing

Preferred parallel architecture for decision support, works on 1000's of commodity hardware processors. Costlier architecture for transactional workloads

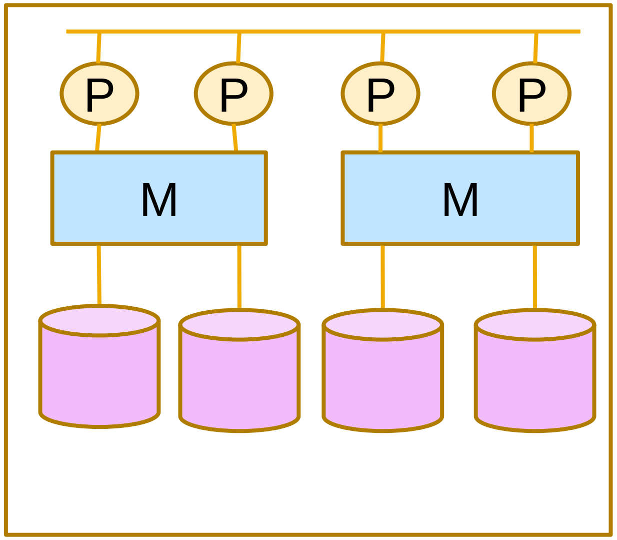
Architectures for parallel processing – 4

Shared Memory

Shared Disk

Shared Nothing

Hybrid (NUMA)



Shared Memory

The bus is the bottleneck beyond 32 processors – in practice used for 4-8 proc. machines

Shared Disk

Slower inter-CPU communication, better fault-tolerance, bottleneck pushed to about 100 proc.

Shared Nothing

Preferred parallel architecture for decision support, works on 1000’s of commodity hardware processors. Costlier architecture for transactional workloads

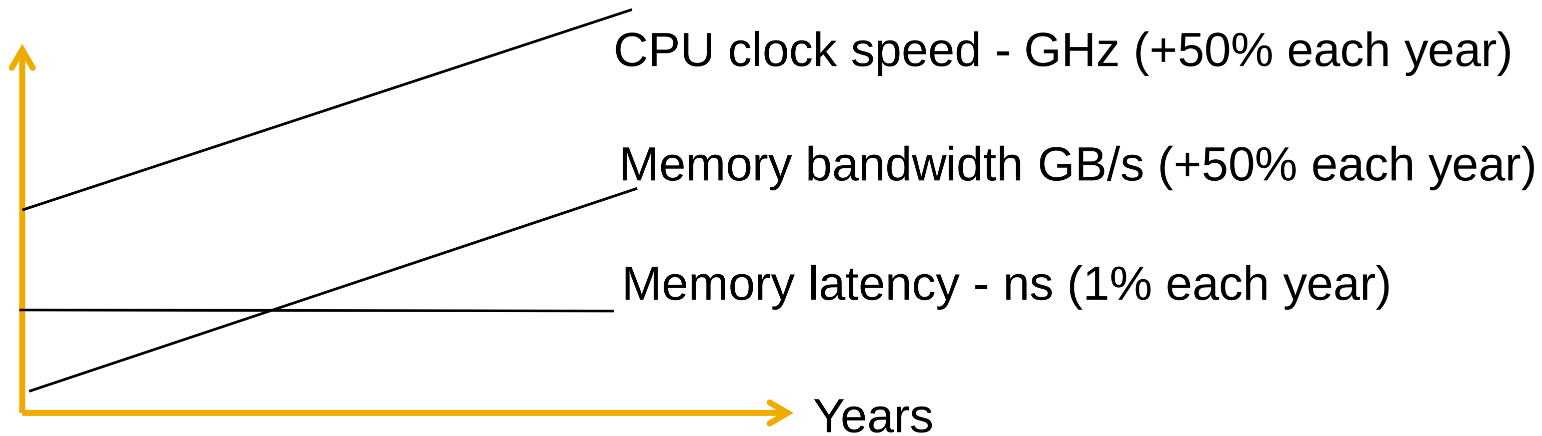
Row vs. column storage and future trends

Row Storage

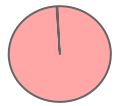
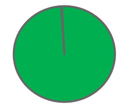
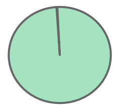
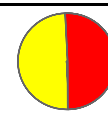
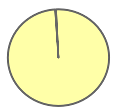
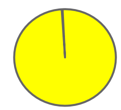
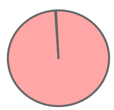
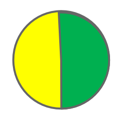
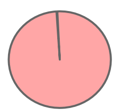
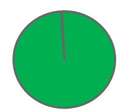
- Read / Write fields of a tuple
  - ALL fields is FAST
  - a SINGLE field is SLOW
- Limited compression
- Slower aggregations
- Fit for transactional workloads

Column Storage

- Read / Write fields of a tuple
  - ALL fields is SLOW
  - a SINGLE field is FAST
- High compression rates (5 to 10x)
- Fast aggregations
- Fit for analytic workloads



Evaluating against requirements

	Analytic PL	Relational DBMS
Scalability		
Deep analytics		
Relational & non-relational data		
Response time		
Data Management		

Map-Reduce

Shared nothing cluster computing

- Thousands of inter-connected commodity (low-cost) computer hardware

Distributed File System

- Architecture for storing redundantly chunks of very large files in the cluster

Map-Reduce

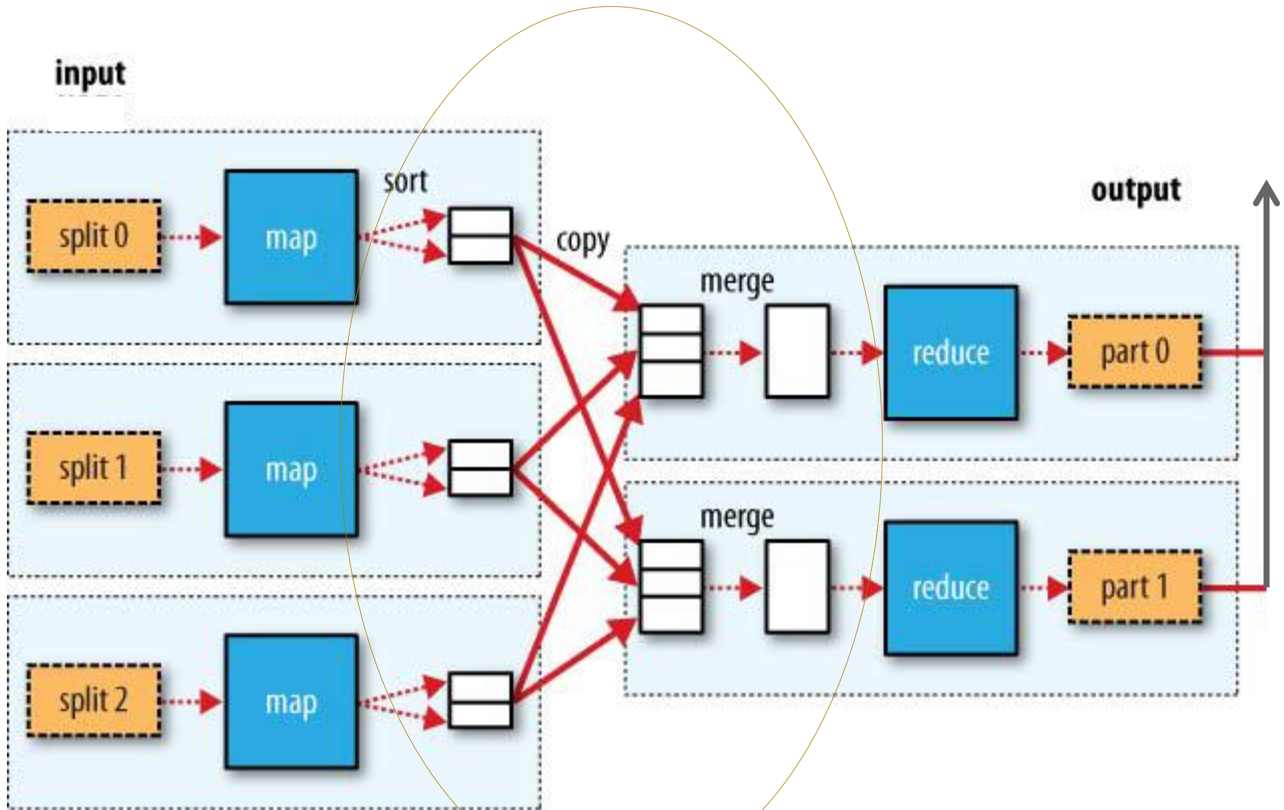
- A parallel computing model built on a DFS (Google, 2004)
- Programmer writes two functions: *map* and *reduce*
- Implementation coordinates parallel execution of map and reduce tasks

Higher level languages layered on top of M-R

- HIVE (a SQL variant), HBase (a No-SQL column oriented DBMS)
- Mahout (a machine learning package)

Hadoop is an open-source implementation of DFS and M-R

Map-Reduce example



Input | map | *Shuffle* | reduce > Output

0699**1943**05..
0699**1950**05..
0699**1050**05..

(0, 0699**1943**05...)
(106,0699**1950**05...)
(212,0699**1950**05...)

(**1950**, -11)
(**1950**, 22)
(**1949**, 111)

0699**1949**05.
.
0699194805

(0, 0699**1949**05...)
(106,0699**1948**05...)

(**1950**, 0)
(**1949**, 78)

(**1949**, [111, 78])
(**1950**, [0, 22 -11])

(**1949**, 111)
(**1950**, 22)

Map-Reduce for Big Data Analytics

No data model

- Typically solved by higher level layers (e.g., HIVE)

Hadoop MR slow on simple analytics

SQL-level analytics slower than parallel R-DBMSs [Pavlo 2009]

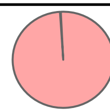
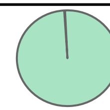
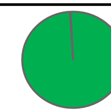
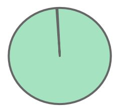
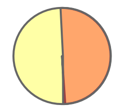
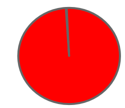
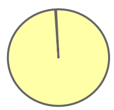
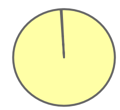
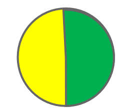
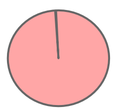
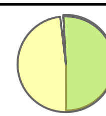
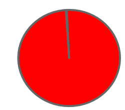
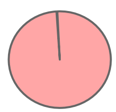
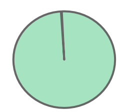
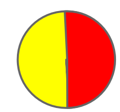
- MR has no indexing
- MR can only retain states between intermediate steps by writing to DFS
- MR can't bind specific processing to nodes in the cluster

Selection with full table scan	Same on large files, else MR is worse
Highly selective query (with index)	R-DBMS about 10 times faster
Simple aggregation	R-DBMS about 2 times faster
Selection, join and aggregation	R-DBMS about 50 times faster

Deep analytics experimentation on MR not yet conclusive, but

- Apache HAMA (matrix & graph computations) preferred the BSP model

Evaluating against requirements

	Analytic PL	Relational DBMS	Map-Reduce Hadoop
Scalability			
Deep analytics			
Relational & non-relational data			
Response time			
Data Management			

Analytic PL + (Relational DBMS or Map-Reduce)

1. Enhance a DBMS with deep analytics capabilities

MADLib: Open source library of SQL-based algorithms for statistics & machine learning, designed for shared-nothing parallel DBMSs

- Laudable effort, but portability is an issue, and performance today is not very good. Besides, will the analyst favor SQL over an Analytic PL?

2. Improve cooperation between DBMS and Analytic PL

Ricardo: Gets analysts to decompose which parts of their program remain in (e.g.,) R, and which by the DBMS (e.g., JAQL on M-R)

- Need sophisticated user to learn two systems and wanting to refactor code

3. Embedding Analytic PL scripts in stored procedures

SAP HANA: R scripts in stored procedure execution plan triggers processing in R, transfers intermediate tables

- As DBMS plans execute in parallel, multiple R runtimes run in parallel

Redesign New DBMSs

Example Is SciDB

New data model merging relations and arrays

New query language (Array SQL)

Shared nothing architecture

Scalable data management

Scalable deep analytics

No-overwrite, fault tolerance

Open source

SAP HANA – a contender in big data analytics

In-memory, parallel multi-core database

Column store

A predictive analytics library for statistics and data mining

- Server-side stored procedures

An integration with R

- Allowing to embed R script in stored procedure

ETL tool including Hadoop connector into HANA

- Parallel ETL through MR task generation

The most advanced BI toolset in the market

- Including BOBJ tools and new UX for predictive analysis

Summary

Lots of interest

- Deep analytics are becoming mainstream

Lots of players

- Industrial, research

Lots of challenges

- Common scalable execution model for linear algebra, statistic algorithms and query languages may prove elusive for a while

This area is ripe for disruption …!