Institut de Recherche en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Pyramid

Data Engineering

Evolution of Data Management Systems:

Fundamental Concepts, Methods and Applications

Emerit. Prof. Abdelkader Hameurlain

hameurlain@irit.fr

Informatics Research Institute of Toulouse IRIT

Pyramid Team*

Paul Sabatier University PSU
Toulouse , France

*** Query Processing & Optimization in Parallel & Large-scale Distributed Environments**

0. Introduction (1/2): Main Problems of Data Management

[Sto 98, Ozs 16, …]

"Data needs to be: <Captured, Cleaned, Stored, Queried, Processed and Turned in Knowledge>"

- **Data Modelling & Semantic**
- **Query Processing & Optimization** (OLAP Online Analytical Processing)
- **Concurrency Control/Transactions** (OLTP Online Transactional Processing)
- **Replication & Caching**
- **Cost Models**
- **Security & Privacy**
- **Monitoring Services**
- **Resource Discovery**
- **Autonomic Data Management (self-tuning, self-repairing, …), …**
- …

➡ **Data Management Systems DMS**

0. Introduction (2/2) : Evolution of Data Management Systems [Gra 96]

➡ *"The present without past has not future"* **Fernand Braudel**

▶ <Concept ➔ Systems: *Objective*>

- ....
- File Management Systems FMS: *Storage Device Independence*
- Uni-processor DB Systems DBMS [Codd 70]: *Prog-Data Independence*
- Parallel DBMS [Dew 92, Val 93]: *High Perf., Scalable & Data Availability*
- Distributed DBMS [Ozs 16]: *Location/Frag./Replication Transparency*
- Data Integration Systems [Wie 92]: *Uniform Access to Data Sources*

  Characteristics =<Distribution, *Heterogeneity, Autonomy*>

- Data Grid Systems [Fos 04]: *Sharing of Available Resources*
- Mobile Database Systems [Oza 08, Mor 11]: *Decentralized Control*
- Cloud Data Mana. Systems [Aba 09, Sto 10]: *Pay-Per-Use* ➡ *Economic Models*

  Characteristics *=<Elasticity, Fault-Tolerant >*

Evolution of Data Management Systems

I. From File Mana. Systems FMS to Database MS DBMS

- Motivations, Objectives, Files Organizations & Drawbacks
- Databases & Rel. DBMS: Motivations & Objectives

II. Parallel Relational DBMS

- Motivations Objectives, Characteristics and Challenges
- Parallel Query Processing
- Optimization of Data Communications: Plague of Parallelism

III. From Distributed DBMS to Data Integration Systems DIS

- Motivations , Objectives & Designing of Distributed DB
- Distributed Query Processing & Soft. Architecture
- Mediator-Wrappers Architecture & Query Processing Methodologies

IV. Cloud Data Management Systems CDMS

- Motivations, Objectives & Main Characteristics of CDMS
- Classification of CDMSs : 3 Generations (G1, G2 & G3)
- Advantages & Weakness of MR Systems & Parallel DBMSs
- Comparison between Parallel DBMSs & MR Systems

V. Conclusion & References

Evolution of Data Management Systems: Cloud Data Management Systems CDMS

Outline

I. Background & Fundamentals: [Codd 70, Sel 76, Dew 92, Val 93, ...]

- From FMS to DBMS: Objectives & Limitations
- Parallel Rel. DBMS: Motivations, Characteristics & Challenges
- From Distributed DBMS to Data Integration Systems DIS

II. Cloud Data Management Systems CDMS [Aba 09, Sto 10, ...]

- Motivations & Main Characteristics of CDMS
- Classification of CDMS : 3 Generations (G1, G2 & G3)
- Applications: Petasky – Mastodons Project [Mas 16]
- Advantages & Weakness of MR Systems & Parallel DBMSs
- Evolution of DML for CDMS (G1)
- Comparison between Parallel Rel. DBMSs & MR Systems (G1)

III. Future Research Directions [Abadi 22, The Seattle Report on DB Research]

IV. Conclusion

I. Background & Fundamentals B & F
1. File Management Systems (1/2)

- **File Concept**

  ➡ ***Program and Storage Device <u>Independence</u>***

  [Storage] <File> [Program/Application]

  ▶ **Software Eng. Requirements**

- **File Organizations: 4 types**
  - **< Sequential /Indexed > Organizations**
  - **< Hashing/Relative> Organizations**

I. Background & Fundamentals B & F
2. File Management Systems (2/2)

- **Access Methods AM**
  - Sequential AM
  - Key AM :=<Indexed/Hashing> AM
- **Drawbacks of FMS**
  - Data description must be done in each program
  - Relationships/Links between files are materialized (➔ New files)

➔ **Database Concept**

I. Background & Fundamentals B & F

3. Databases DB and Relational DBMS [Codd 70]

- **DB Objectives:**
  - Separation between Data Structures (DB Schema) & Program.
  - Prog-Data Independence = <Physical & Logical> Independence
- **DB Models: <Hierarchical, Network, Relational & Object>**
- **Main Characteristics (Rel. DB)**
  - Structured Data: Relation Concept to describe <Entities & Links>
  - Relational Algebra: Commutative, Internal Law
  - Rel. Languages: From Procedural ➔Declarative Languages: SQL [Cham 76], QUEL [Sto 76], QBE [Zlo 77], ….
    - The System will find the (near) Optimal Access Path
      - ➡ **Optimizer** [Sel 79, Wong 76, Gan 92, …]

I. Background & Fundamentals B & F

4. Uni-proc. Rel. DBMS: Query Optimization [Sel 79]

- **Problem Position [Gan 92]:**

  q $\in$ Query , p $\in$ {Execution Plans}, $Cost_p$ (q):

  - **Find p calculating q such as $Cost_p$ (q) is minimum**
  - **Objective : Find the best trade-off between**

    Min (Response Time) & Min (Optimization Cost)

- **Optimizer Structure= < St, Sp, C> [Gan 92]**

  – **St: Search Strategies (➔ Intelligence)**
  - **<Physical Optim., Parallelization, Resource Allocation, …>**

  – **Sp: Search Space (➔ Control)**
  - **Data Structures/Queries: Linear Spaces, Bushy Space**
  - **Type/Nature of Queries**

  – **C: Cost Models (➔ Knowledge)**
  - **<Metrics, System Environment Description>**

I. Background & Fundamentals B & F

5. Limitations of Uni-proc. Query Optimization Methods wrt Decision Support Systems /OLAP

- **Complex Queries***: Number of Joins >6*
- **Size of Research Space [Tan 91]:** *Very Large (e.g. 2^{N-1})*
- **Optimization Cost [Lan 91]:** *can be very expansive (e.g. Deterministic Strategies wrt Random Strategies)*
- **Optimal Execution Plan:** *not guaranteed (e.g. Random Strategies)*

➡ *Requirements in:* **High Performance HP & Resource Availability**

➡ **Introducing a New Dimension:** ***Parallelism***

▶ **Parallel Relational Database Systems [Dew 92, Val 93, ...]**

I. B & F: 6. Parallel Relational DBMS (1/2) [Dew 92, Val 93, Lu 94,.. .]

- **Motivations: Declarative Relational Languages (e.g. SQL)**
  - Automatic Parallelization of <Intra-operation & Inter-operation>
  - Parallelism Forms: <Partitioned & Independent, Pipelined> //
  - Regular Data Structures : ➔ *Static Annotations*
  - Decision Support Queries: Complex Queries, Huge DB (TB, PB, …)

- **Objectives [Dew 92]:**
  - Best Trade-off between Cost/Performance wrt Mainframe
  - High Performance HP
    - Minimizing the Response Time
    - Maximizing the Parallel System Throughput
  - **Scalability (≠ Elasticity)**
    - Adding New resources (CPU, Memory, Disk)
    - Adding New Users (Applications)

      ➡ Holding the Same Performance
  - Resource Availability: Complex Queries, Fault-Tolerant

I. B & F: 6. Parallel Rel. DBMS (2/2) [Dew 92, Val 93, Lu 94,.. .]

Main Characteristics

- Parallel Architect. Models: SM, SD, DM= Shared-Nothing Architecture
- Parallelism Forms: <Partitioned, Independent, Pipelined>
- Data Partitioning:
  - Approaches: <Full Declustering, Partial Declustering>
  - Methods: <Round Robin, Range Partitioning, Hashing>

Main Challenges

- Parallelism Degree of each Relation/Operator (e.g. Join)?
- Parallelization Strategies: <One-Phase, 2-Phases> Approaches>
- Resource Allocation: Data & Tasks Placement/Scheduling
- Optimization of Data Communications: Plague of Parallelism!

Weakness of Parallel Rel. DBMS

- Run only on Expensive servers
- Web Data Sets are not structured (Relational Schemas)
- Weak Fault - Tolerance
- Communication Costs: Data Redistribution (=Reshuffling in MR)

➔.... Towards Cloud Data Management Why ?

II. Towards Cloud Data Management Systems CDMS

[Aba 09, Sto 10/13, Agr 10-12, Chaud 12, Zhou 12, Kald 12, Gra 13, LI 14, Unt 14, Norvag 14, Akba 15, Bon 15, Aba 16 ...]

Outline

- **Big Data, Cloud Computing & MapReduce MR: Motivations?**
- **Main Characteristics of Cloud Systems [D. Agrawal 2011]**
  - **"Hot Debate" on: MapReduce Versus Parallel DBMS: friends or foes?**
    [M. Stonebraker et al., 2010], [D. Agrawal et al. 2010, S. Chaudhauri 2012]
  - **" Reconciling Debate" [Zhou 2012, Kaldewey 2011]**
    "SCOPE : Parallel Databases Meet MapReduce" [Zhou 2012]
- **Classification of Cloud Data Management Systems CDMS**
- **Advantages & Weakness of Parallel RDBMS & MR Systems**
- **Applications: Petasky – Mastodons Project [Mas 16]**
- **Evolution of DML & Compar. between Par. RDBMS & MR Systems**

II.1 Motivations (1/2): Big Data & Cloud Computing

- **Big Data? : Generated from specific requirements of Web Appli.**
  + Tradit. Appli. : C. Sim, Sat. , Astronomy, Live Sc, Buisness, ….

Remark: 50th Intl. Conf. on Very Large DB; 49th Intl . Conf. On Manag. of Data

➡ **Big Data → "Moving Target" [Val 16]**

- **Big Data Characteristics: the 3 V's (Volume, Velocity, Variety)**
  ➡ **What are the Solution for "the 3 V's" [Val 14] ?**
  - **Volume: Refers to very large amounts of Data**
    ➡ **Parallel Database Systems [Dew 92]**
  - **Velocity: Streaming Data**
    ➡ **Data Stream Management Systems [Ozu 16]**
  - **Variety: Heterogeneity of Data Formats, Semantics & Resources**
    ➡ **Data Integration Systems [Wied 92]**

However, why these systems are not naturally used?

II.1 Motivations (2/2): Towards Cloud Computing & MapReduce

➡ **Observation (Buisness Idea!): "One size does not fit all" [Sto 2010]**

- **Current Solutions (Infrastructures & Software/RDBMS) are:**
  Proprietary & Expensive

  ➡ **Open Source Alternatives, Simple Programming Model ! (e.g. MapReduce), Low Costs LC (Commodity Hardware CH)**

- **Ability to scale resources out up and down dynamically on- demand: ➡ Elasticity (➔ Pay-Per-Use PPU)**

- **How the systems should react "strongly" to Failures?**
  <Commodity Hard./LC, Data Replication, HDFS> ➡ Fault-Tolerance

- **Cloud Environments do not to be Owned nor Managed by a Customer (PPU Approach): Users ➔ Multi-tenant**
  ➡ <Tenant, Provider> through SLA (Service Level Agreement)

II.2 Main Characteristics of Cloud Systems [Agra. et al. 2011]

- **Scalability (Infrastructure: Shared-nothing Architecture)**
- **Elasticity [Ozu 16]:**

  «The ability to scale resources out up and down dynamically to accommodate changing conditions»

- **Strong Fault-Tolerance:**
  - **Ability to run on Commodity Hardware CH (Low Cost!)**
  - **Data Replication (e.g. HDFS)**

- **Users ➔ Multi-tenant [Nara 13]: <Tenant, Provider> trough SLA (Service Level Agreement) Meeting**

➡ **New Context = <Service on-demand, Multi-tenant, Commodity Hardware >**

➡ **Introduction of Economic Models in the Resource Managemen**

II.3 Classification of Cloud Data Manag. Systems CDMS (1/3)

- **1st Generation G1: From MapReduce MP ➔ SQL- Like**
  - **MP Systems ➔SQL on-Hadoop Systems based on Type of Data Store:**
    <Key-value Store, Document Store, Column –Family, Graph DB >
    - **Simple Queries= Selection Queries**
    - **Bigtable, Hive, MongoDB, Cassandra, Neo4j, Riak, Spark, ...**

- **2nd Generation G2: From Parallel RDBMS ➔ Multi-tenant Par. RDBMS**
  - **Extension of Parallel Rel. RDBMSs with the "Cloud Concept"**
    ➔ <High Performance & Elasticity> [Won15, Yin 18, ...]
    - **Complex Queries= Join Queries**
    - **Amazon Redshift, Azure SQL DW, Google BigQuery, Snowflake DW, ...**

- **3rd Generation G3: =<Distribution, *Heterogeneity, Autonomy*>**

based on the concepts: <Multibase/Federated DB & Data Integration>

- **Multistore/Polystores Systems: Polybase [Dew 13], SCOPE [Zho 12], CoherentPaas Proj. [Bon 15], BigDAWG [Sto/Dug 15], [Sol 20], [Lec 18], ...**

II.4 1^{st} Gener. G1 : From MR ➔ SQL Like on-Hadoop Systems

- **Classification of NoSQL Systems : Type of Data Store**
  - ➡ **Observation (Buisness Idea!): "One size does not fit all" [Sto 2010]**

- **Key-value Store:** <Azure Table Storage, DynamoDB, Redis, Riak, Voldemort, ...>
- **Document Store (XML, JSON):** <MongoDB, CouchDB, RavenDB>
- **Column-family (Rel. DB, Data is stored in column):** <Hbase, Cassandra, Hypertable>
- **Graph Databases (Social Networks):** <Neo4j, Infinity Graph, InfoGrid, ...>

II.5 1st Gener. G1 : From MR ➔ SQL - Like on-Hadoop Systems (1/2)

➡ **Advantages and Weakness of MR**

Advantages of MapReduce MR

- Scaling very well (to manage massive data sets)
- Strong Fault -Tolerance (Data Replication, HDFS)
- Mechanism to achieve Load-Balancing
- Support **only** the Intra-Oper. & Independent Parallelisms (**Pipeline Par.?**)

Weakness of MR: Side Applications

Developers:

- Are forced to translate their business logic to MR model
- Have to provide implementation for the M & R functions
- Have to give the best scheduling of M & R operations

➡ **More Hot Problems wrt Data Management!**

- Prog-Data Structure Independence **is lost** (DB Objective !)
- Extensive Materialization **(I/O)** (the Pipeline // is not implemented)
- Data Reshuffling (Redistribution) between M & R ➡ **Plague of Parallelism**

II.5 1st Gener. G1 : From MR ➔ SQL Like on-Hadoop Systems (2/2)

➡ Advantages and Weakness of Par. RDBMS

- **Advantages of Par. RDBMS [Dew 92]**
  - Relational Schemas (➔ Easy Annotations/Metadata)
  - Declarative Query Languages (➔ Automatic Optimization Process)
  - Sophisticated Query Optimizers-Parallelizers : {Partitioned, Indep., Pipelined //}
  - +/- Comm. Costs : Avoid the Data Redistribution (+/-: in some cases)

- **Weakness of Par. RDBMS**
  - Run only on expensive servers
  - Weak Fault - Tolerance
  - Web Data Sets are not structured (Relational Schemas)
  - Communication Costs: Data Redistribution (=Reshuffling in MR)

II.6 Comparison between Par. RDBMS & MR Systems 1st Gener. (G1)

Systems / Parameters	Par. RDBMS	MapReduce Systems (Hadoop Env.)/1st Generation
Type of Applications	OLAP & OLTP (ACID)	OLAP: Yes; **OLTP: Not suitable (Initially!)** ➔ NewSQL (HTAP)
Data Models	Structured Data (Relational Schema)	Unstructured or semi-Structured , ...(more Flexible!)
Data – Prog Independ.	Yes	No (Initially)
Query Languages	Declarative	Procedurals (initially)
Optimization & Parallelization	Automatic Optim. & // Annotations: Easy	Explicit Optim. (initially) Annotations: Very difficult
Scalability & Elasticity	Scalable & Dynamic	Scalable & Elastic
Fault-Tolerance	Weak	Strong
Location --- Maturity	Known in advance --- Strong	SLA Negotiation --- Weak (at this moment!)

II.7 Evolution of Data Manip. Languages for CDMS/1st Generation

Charact. ➔ Nature of Languages	Functions (Power)	Advantages	Drawbacks
L1: Proc./Func. Languages (e.g. MapReduce) [Bigtable, PNUTS]	Filter & Project Google, Yahoo!	– Simplicity of Programming Model!	– Complexity to read and optimize prog. – Data Str. Dependency
L2: P/FL with Relational Operators [PIG Latin, Jaql]	Rel. Operators Towards SQL func Yahoo!, IBM	– Prog. are more readable – Automatic Logical Optim.	Developers provide Scheduling of Rel. Op ➔ No Physical Optimization
L3: Declarative Languages [HiveQL, SQL/ SPARK, TEZ, ...]	Close to SQL + Specific Operators MS, FB, IBM & Google	Automatic : – Optimization – Parallelization (➔ Avoid Data Reshuffling)	"Lack of statistics stored in The catalog" ➔ "Blinds the optimization Process"

II.8 Petasky – Mastodons Project (CNRS, LIMOS/LIRIS)

"Benchmarking SQL on MapReduce systems using large astronomy databases"; A. Mesmoudi et al.; In: Intl journal PDBD, 34(3), 2016

- **Objectives:** **"They report on the capability of 2 MR systems (Hive and HadoopDB) to accommodate LSST* data management requirements" in terms of loading & execution times : < Data Loading & Indexing and Queries (Selection, Group By, Join) >**
- **Conclusions [Mes 2016] :**
  - ➡ **"We believe that the model is efficient for queries that need one pass on the data (e.g. Selection and Group By)"**
  - ➡ **" We believe that MR model is not suitable for handling Join queries "**

*** LSST : Large Synoptic Survey Telescope**

II.9 Classification of Cloud Data Manag. Systems CDMS (2/3)

- **1st Generation G1: From MapReduce MP → SQL- Like**
  - **MP Systems →SQL on-Hadoop Systems based on Type of Data Store:**
    <Key-value Store, Document Store, Column –Family, Graph DB >
    - **Simple Queries= Selection Queries**
    - **Bigtable, Hive, MongoDB, Cassandra, Neo4j, Riak, Spark, ...**

- **2nd Generation G2: From Parallel RDBMS → Multi-tenant Par. RDBMS**
  - **Extension of Parallel Rel. RDBMSs with the "Cloud Concept"**
    → <High Performance & Elasticity> [Won15, Yin 18, ...]
    - **Complex Queries= Join Queries**
    - **Amazon Redshift, Azure SQL DW, Google BigQuery, Snowflake DW, ...**

- **3rd Generation G3: =<Distribution, *Heterogeneity, Autonomy*>**
  based on the concepts: <Multibase/Federated DB & Data Integration>
  - **Multistore/Polystores Systems: Polybase [Dew 13], SCOPE [Zho 12] , CoherentPaas Proj. [Bon 15], BigDAWG [Sto/Dug 15], [Sol 20], [Lec 18], ...**

II.9.1 2nd Gen.: Multi-tenant Par. RDBMS (Par DB-as-a-Service PDBaaS)

2 Approaches to provide PDBaaS: A1 & A2

<Elasticity, High Performance, Cost-effectiveness>

- **A1: Exclusive Resource Approach**
  - Hard Isolation between Tenants
    - ➡ Meeting tenant SLAs
  - Poor Resource Utilization (Low Cost – Effectiveness)
- **A2: Shared Resource Approach**
  - Soft Isolation between tenants
    - ➡ SLAs may not be guaranteed (➔ Penalty, thresh-hold)
  - Better Resource Utilization: Avoid Resource Contention
    - ➡ Elastic Resource Allocation Models [Kan 19, Won 15,. ..]

II.10 Classification of Cloud Data Manag. Systems CDMS (3/3)

- **1st Generation G1: From MapReduce MP → SQL- Like**
  - **MP Systems →SQL on-Hadoop Systems based on Type of Data Store:**
    <Key-value Store, Document Store, Column –Family, Graph DB >
    - **Simple Queries= Selection Queries**
    - **Bigtable, Hive, MongoDB, Cassandra, Neo4j, Riak, Spark, ...**

- **2nd Generation G2: From Parallel RDBMS → Multi-tenant Par. RDBMS**
  - **Extension of Parallel Rel. RDBMSs with the "Cloud Concept"**
    → <High Performance & Elasticity> [Won15, Yin 18, ...]
    - **Complex Queries= Join Queries**
    - **Amazon Redshift, Azure SQL DW, Google BigQuery, Snowflake DW, ...**

- **3rd Generation G3: =<Distribution, *Heterogeneity, Autonomy*>**

based on the concepts: <Multibase/Federated DB & Data Integration>

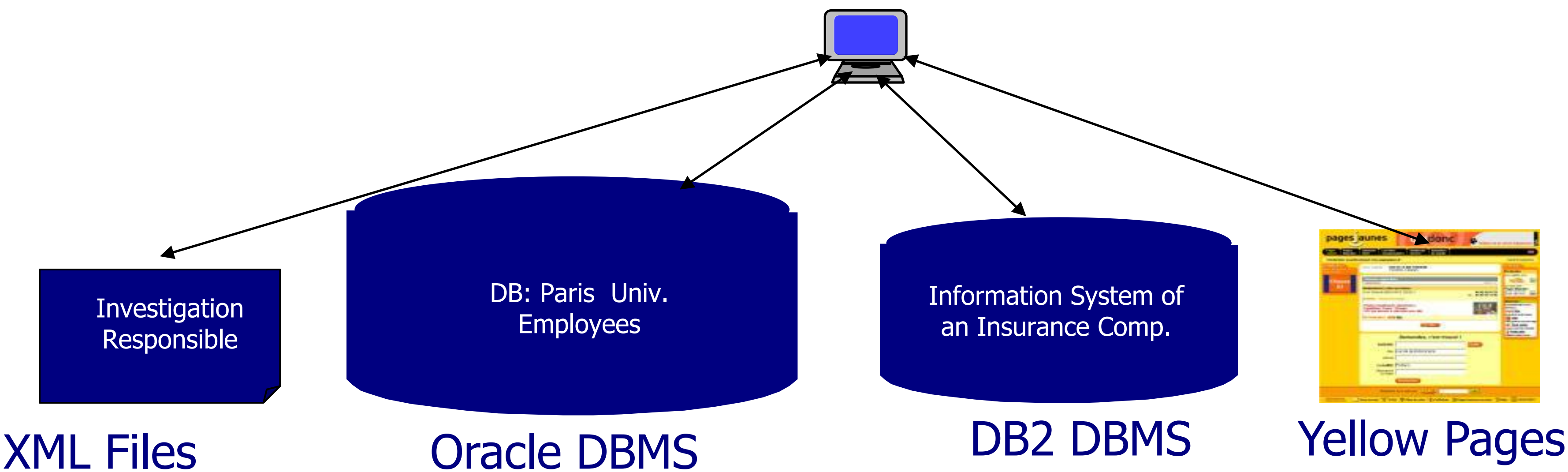
- **Multistore/Polystores Systems: Polybase [Dew 13], SCOPE [Zho 12], CoherentPaas Proj. [Bon 15], BigDAWG [Sto/Dug 15], [Sol 20], [Lec 18], ...**

II.10.1 3rd Generation: Multistore/Polystore Systems (1/2)

Strongly Inspired: Data Integration Systems [Wied 92, Gol 00, Yer 99, ...]

Question : Quels sont les numéros de téléphones des Medecins traitant des employés de la section Informatique dont Durand est le responsable ?

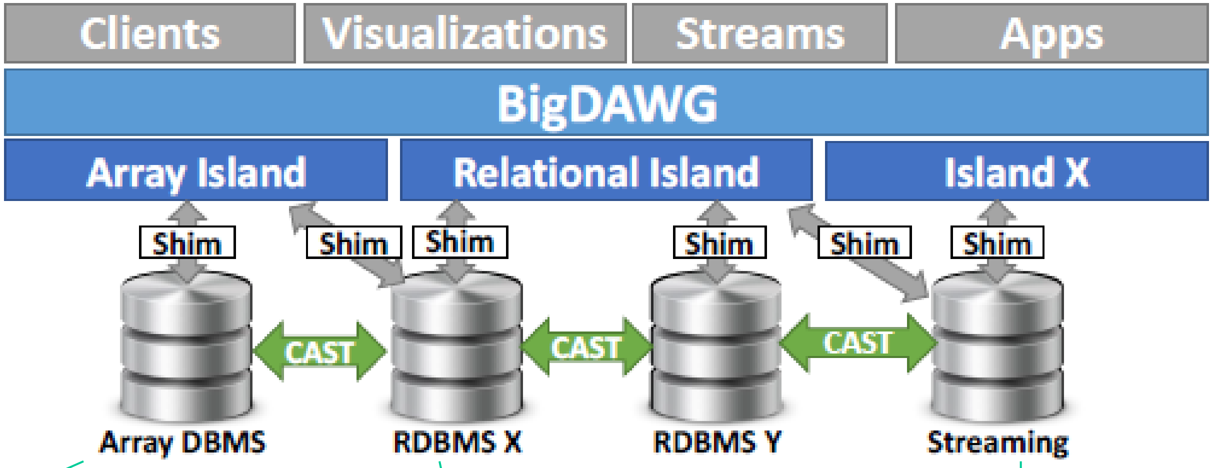
- **Requirement: Data Integration arising from Several Sources**
- **Characteristics: <Distribution, Heterogeneity, Autonomy>**
- **Objective: Uniform Access to Data Sources**



➡ **Mediator-Wrapper Architecture [WIE 92]**

3rd Generation: A Case Study

II.10.2: Polystore System with BigDAWG [Dug 15]



e.g., **SciDB [Bro10] : Archived Time Series Data**

e.g., **Postgres : Patient Metadata**

e.g., **S-Store [Cet14] : Real-time Waveform Data**

III.1 Future Research Directions (1/5)

- **New Context in CC= <Service on-demand, Multi-tenant, Commodity Hardware>**

  ➡ **Introduction of Economic Models in the Resource Management**

- **Research Challenges [Abadi et al. 2022; "The Seattle Report on DB Research"]**

  RC1: "Data Science"

  < Data-to-Knowledge Pipeline, Data Context & Provenance, Data Manag. in support of Machine Learning, …>

  RC2: "Data Governance"

  <Data Use Policy & Data Sharing, Data Privacy , Ethical Data Science, …>

  RC3: "Scalable Big/Fast Data Infrastructures"

  <New Hardware (CPU/GPU), Parallel & Distributed Processing, *Query Proc. & Optimization,* Cost-efficient Storage, NewSQL, HTAP (Hybrid Transaction Analytical Processing), Metrics & Benchmarks, …>

  RC4: "Cloud Services"

  <*Elasticity, Multi-tenancy*, Performance Isolation, Multistore/Polystores Systems, Leveraging Machine Learning, Auto-Tuning, ….>

III.2. HTAP Hybrid Transaction Analytical Processing (2/5) [Val 22]

- **Context:**
  - **OLAP (Online Analytical Processing): Querying and analyzing data for decision-making and strategic purposes**
  - **OLTP (Online Transactional Processing): Updating a consistent DB**

- **Main Characteristics MC:**
  - **OLAP: Complex Queries, High Performance & Availability**
  - **OLTP: Simple Update Queries (Insert, Delete, Modify), Consistency (Coherence) of DB, ACID Properties.**

  → **"OLTP helps run a business while OLAP helps to understand it"**

- **Limitations: OLAP & OLTP are separately Managed (See their MC)**

- **Objective: is to unify the 2 systems in a single system, making it possible to simultaneous perform complex queries and updating requests in real-time.**

III.3 Future Research Directions (3/5)

➡ **Contribution of Machine Learning for Query Optimization**

- **Optimizer Structure= < St, Sp, C> [Gan 92]**
  - **St: Search Strategies** (➔ Intelligence)
    - <Physical Optim., Parallelization, Resource Allocation, …>
  - **Sp: Search Space** (➔ Control)
    - Data Structures/Queries: Linear Spaces, Bushy Space
    - Type/Nature of Queries
  - **C: Cost Models** (➔ Knowledge)
    - <Metrics, System Environment Description>

➡ **Estimation errors in metric values**

Could Machine Learning ML effectively improve estimation errors?

III.3 Future Research Directions (4/5)

- **Open Issues wrt *Query Processing and Optimization***

  P1: Elastic Resource Allocation & Dynamic Data Replication

  [Kouri 13, Gra 13, Unter 14, Wong 15, Tan 16, Yin 18, Mok 20, …]

  P2: Data Skew & Load Balancing

  [Ram 12, Guf 12, Kwon 12/13, Elm 14, Akba 15, ….]

  P3: Data Partitioning & Redistribution (Reshuffling Issue in MR)

  (Optimization of Data Comm. in // DB Systems) [Chu 15, Lir 13, Sakr 12, …]

  P4: Big Data Indexing [Val 14, …., Knuth 73]

  ➔ [Val 14] "Indexing and Processing Big Data"

  In: Mastodons Indexing Scientific Big Data, Paris, January 2014.

III.4 Future Research Directions (5/5)

- **P1: Elastic Query Optimization** [...,Yin 18, Mok 20, ...]
  - Resource Allocation: Scheduling/Placement of Data/Tasks
  - Dynamic Data Replication
  - Cost Models :=<High Performance, Cost-effectiveness>

➡ **Designing of Dynamic Execution Models:**

Efficient (Tenant) & Cost-effective (Provider)

➡ **Objective Function:** Find the best trade-off between

- Multi-tenant Satisfaction (QoS (e.g. Response Time)) &
- Cost-effectiveness of Provider Services <IaaS, PaaS, DBaaS/ SaaS>

IV. Summary & Conclusion : Evolution of Data MS:
<Concept ➔ Systems: *Objective*>

- **File Management Systems:** *Storage Device Independence*
- **Uni-processor Rel. DB Systems DBMS [Codd 70]:** *Data –Prog. Indepen*
- **Parallel DBMS [Dew 92, Val 93]:** *High Perfor., Scalable & Data Availability*
- **Distributed DBMS [Ozs 16]:** *Location/Frag./Replication Transparency*
- **Data Integration Systems [Wie 92]:** *Uniform Access to Het. Data Sources*

  Characteristics =<Distribution, Heterogeneity, Autonomy>

- **Data Grid Systems [Fos 04, Pac 07]:** *Sharing of Available Resources*
- **Mobile Database Systems [Oza 08, Mor 11]:** *Decentralized Control*
- **Cloud Data Manag. Systems:** *<Pay-Per-Use>* ➔ **Economic *Models***

  1st Gen. : SQL-on-Hadoop Systems; 2nd Gen.: Extension of Par. RDBMS with "Cloud Concept"; 3rd Gen.: Multistore/Polystores Systems

  Characteristics =<Elasticity, High Performance , Fault-Tolerance>

IV. Summary: Main Characteristics of Cloud DMS: G1, G2 & G3

Main Characteristics of 1st G1: From MapReduce → SQL Like

- "One size does not fit all" : Systems are based on Type of Data Store
- Low Performance : <Selection Queries=one pass>
- Extensive Materialization I/O: initially, the Pipeline has not been implemented!
- Loss of Data Structure – Prog. Independence (Initially!)

  ➡ Weak Fault-Tolerance (Pipeline Parallelism)

Ind. Prod. : Bigtable, Hive, MongoDB, Cassandra, Neo4j, Riak, Spark, ...

Main Characteristics of 2nd G2: Multi-tenant Parallel RDBMS

- + High Performance (Partitioned, Indep., Pipelined //) : → Complex Queries
- + Decla. Query Languages & Optimizer – Parallelizer & Minimization of Comm. Costs
- - Poor Semantic (Relational Model, "One size does not fit all"!)

  ➡ <High Performance & Elasticity> ... Weak Fault-Tolerance ?

Ind. Prod. : Amazon Redshift, Azure SQL DW, Google BigQuery, Snowflake DW, ...

Main Characteristics of 3rd Generation G3: Multistore/Polystores Systems

- <Distribution, *Heterogeneity, Autonomy*>
- "Provide integrated access to different data stores (e.g. HDFS, SQL, NoSQL) through one or more query languages"

IV. Conclusion (1/4): Maturity of Big Data Manag. Systems/Cloud

- **Query Languages**
  - **Declarative Languages**
  - **Standardization**
- **More Experimentation & Benchmarking**
  - **TPC – H & TPC - DS**
- **Administration & Tuning/Supervision Tools**
- **Let time do its work!**

IV. Conclusion (2/4): Criteria for Choosing a Data Mana. System?

- **C1: Price ➡ Investment VS Pay-Per-Use (Cloud Computing Platform)**

- **C2: Characteristics of Applications (Objectives & Evolution)**
  - **Nature of Applications: OLAP, OLTP, Hybrid (HTAP)**
  - **Data Models/Structures: File, DB, XML, ...**
  - **Degree of Schema (Sem) Evolution (Data – Prog. Independence)**
  - **Template Queries: Type & Nature of Queries and Indexing**

- **C3: Characteristics of DM Systems (System Infrastructures)**
  - **Environment: Uni-proc., Parallel, Distributed**
  - **Fundamental Functionalities: DDL, DML, Programming Languages (Java/C + SQL), Consistency Constraints, ...**
  - **DMS Administration & Tuning**

IV. Conclusion (3/4): Impacts of CDMS on Scientific & Social Aspects

1. Scientific Aspects (1/2) :

"The Beckman Report on Database Research" [Abadi et al. 2016]

- "Many early Big Data Mana. Systems BDMS Abandoned of DBMS Principles (e.g. Declarative Programming and Transactional Data Consistency) in favour of Scalability, Elasticity & Fault-Tolerance on Commodity Hardware" .

- "The latest generation of DBMS is rediscovering the value of these principles and is adopting concepts and methods…"that have been mastered by the DB Community DBC.
  - ➡ "Building these systems on these principles, the DBC is well positioned to drive improvements ….."

IV. Conclusion (4/4): Impacts of CDMS on Scientific & Social Aspects

1. Scientific Aspects (2/2)

<Concepts, Approaches, Methods, Tech/Tools> & <Applications>

- **New "Concept" introduced by the Cloud Computing CC**
  In terms of: <Data Models, DM Languages> ?

  ➡ **Economic Models in the Resource Management (Elasticity)**
  Rationalization & Cost-effectiveness!

- **Risk of a Gradual Shift of Fundamental Research Activities towards only Engineering Activities**

  ➡ **Best trade-off between: <Fund. Research & R&D>**

2. Social Aspects: Feedback from Industry and Institutions

➡ **Evaluation of benefits and social impacts of CDMS?**

Thank you for your attention

===================

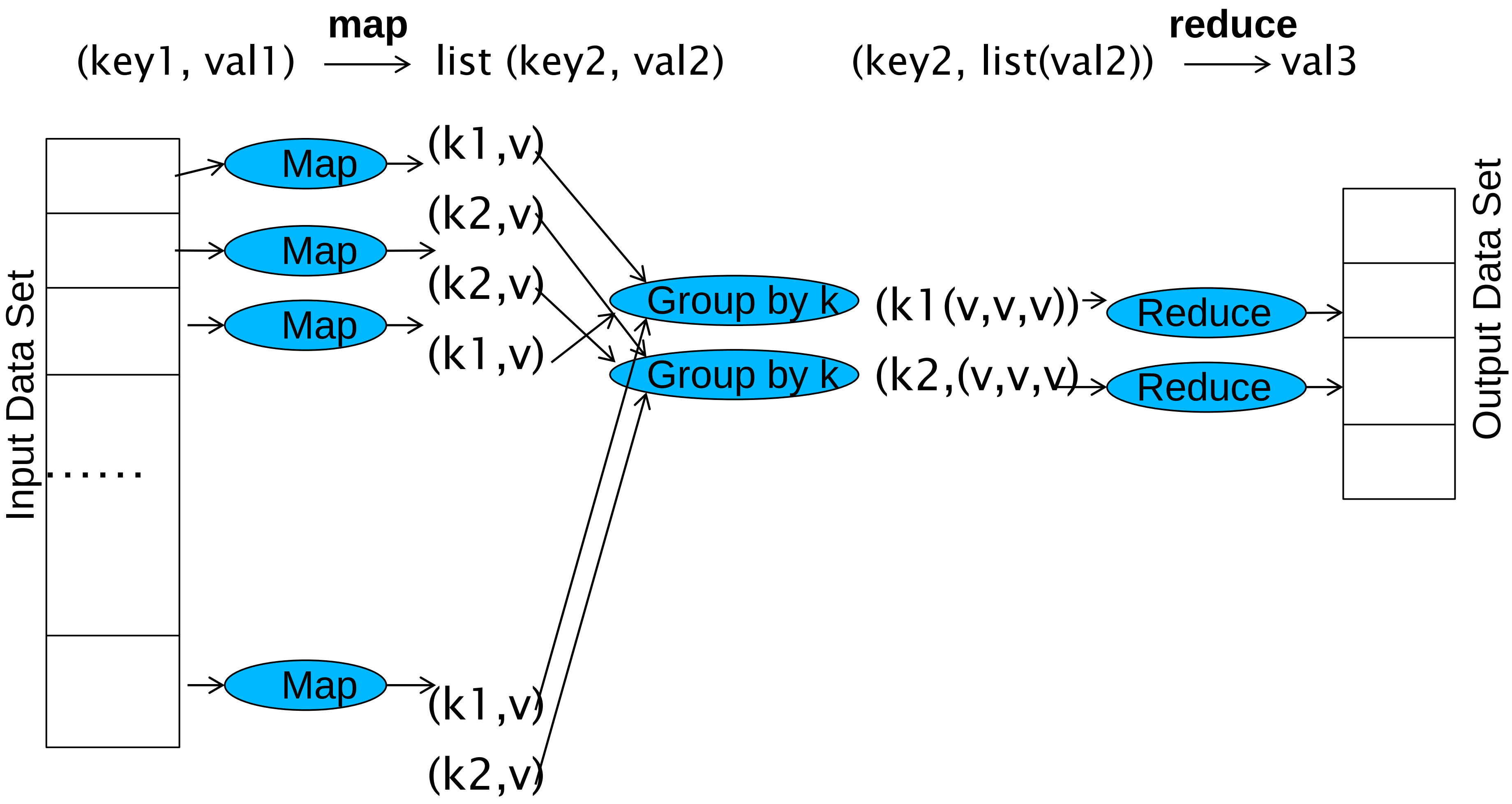
Contact: Hameurlain@irit.fr

Informatics Research Institute of Toulouse IRIT

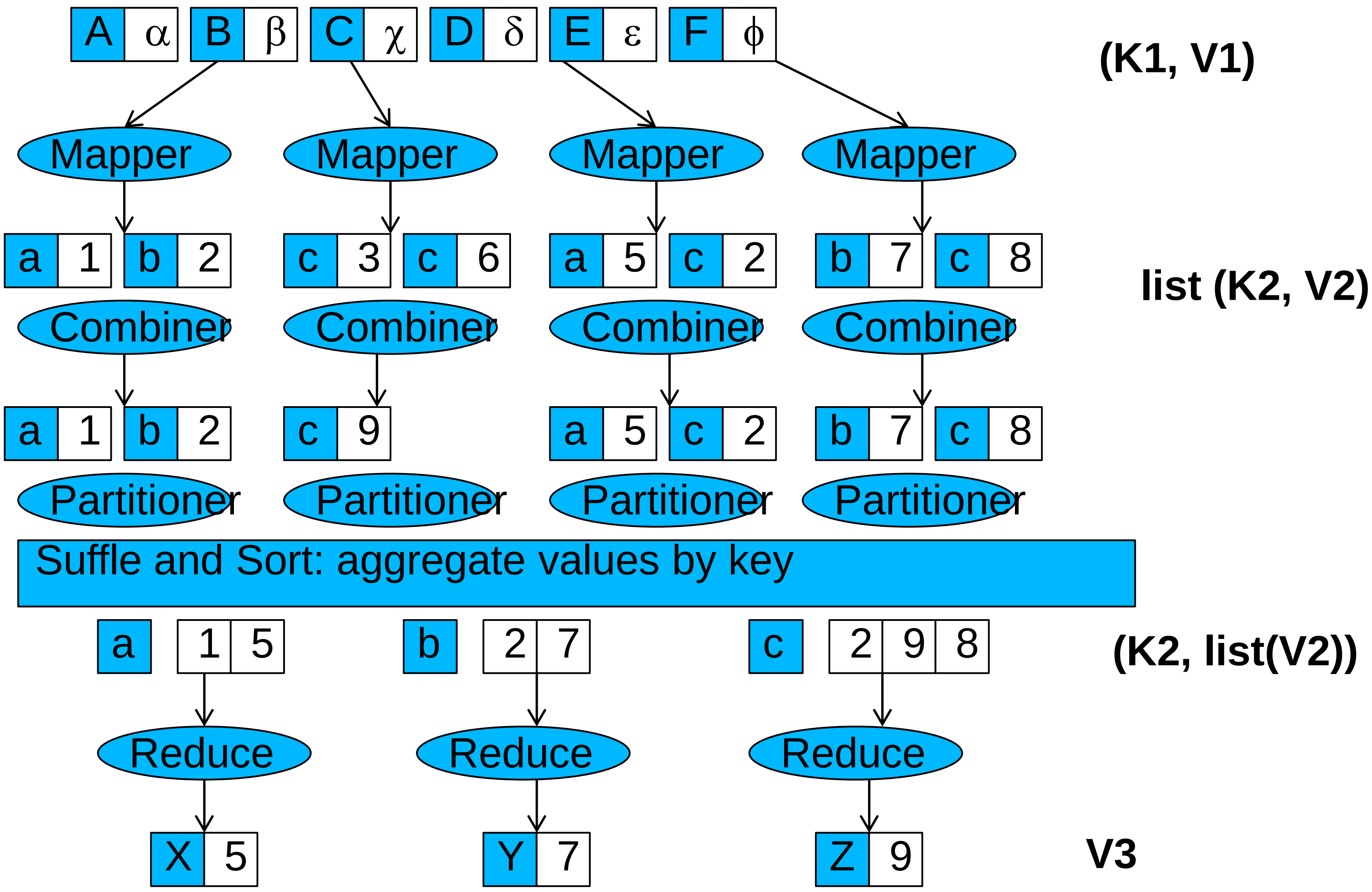
Pyramid Team

Paul Sabatier University
Toulouse , France

MapReduce Processing [Val 2010]



Combiner & Partitionner [Val 2010]



V.1 References: // DB Systems

- **D.J. DeWitt, J. Gray, "*Parallel Database Systems: The Future of High Performance DB Systems*" , in: *Comm. of the ACM*, Vol. 35, 1992, pp. 85-98.**
- **P. Valduriez, : "*Parallel Database Systems: Open Problems and News Issues*", in: Distributed and Parallel DB, Vol. 1, pp. 137--165, Kluwer Academic, (1993)**
- **H. Lu et al., "*Query Processing in Parallel Relational Database Systems*", IEEE CS Press, 1994**
- **D. Taniar et al., "*High Performance Parallel DB Processing and Grid Databases*", Ed. Wiley, 2008**
- **A. Gounaris et al. ; " *Adaptive Query Processing: A Survey* ", Proc. of the 19th British National Conf. on DB, Sheffield, UK, July 2002, pp. 11-25**
- **A. Hameurlain, F. Morvan ; " *Parallel query optimization methods and approaches: a survey* ", Intl. Journal of Computers Systems Science & Engineering, CRL Publishing, Vol. 19, No.5, Sept. 2004, pp. 95-114**

V.2 References: Distributed DB Systems

- M.T. Özsu, P. Valduriez, Principles of Distributed Database Systems , 3rd Edition, April 2011, Ed. Springer Verlag
- D. Kossman , The State of the Art in Distributed Query Processing, ACM Computing Surveys, Vol. 32, No. 4; 2002
- M. Stonebraker , Hellerstein J.M. : Reading in DB Systems, M. Kaufmann Publisher, 3rd Ed., 1998
- M. Stonebraker, et al..: Mariposa: A Wide-Area Distributed Database System. In:VLDB Jour., 5(1), pp. 48--63, Springer, (1996)
- P. Valduriez, Principles of Distributed Data Management in 2020? Invited Talk, in:Dexa 2011, Toulouse/France), LNCS 6860, pp. 1-11.
- …

V.3 References: Cloud Computing & Data Management (1/3)

- F. Afrati & Ullman; Optimizing Joins in a MR Environment; EDBT'2010
- F. Afrati & Ullman; Optimizing Multiway Joins in a MR Environment; IEEE TKDE 23(9), 2011, pp; 1282 – 1298.
- S. Agarwal, et al., « Re-optimizing data-parallel computing », In Proc. of USENIX NSDI Conf., 2012.
- D. Agrawal et al., "Big Data and Cloud Computing: New Wine or Just New Bottles?", In: VLDB'2010 Tutorial, PVLDB, Vol. 3, No. 2, pp. 1647-1648.
- D. Agrawal et al., "Big Data and Cloud Computing: Current State and Future Opportunities", In: EDBT 2011, Tutorial, March, Uppsala, Sweden.
- D. Agrawal, et al., « The evolving landscape of data management in the cloud », Int. J. Computational Science and Engineering 7(1), 2012.
- Blanas et al. ; A Comparison of Join Alg. for Log Processing in MR; SIGMOD'2010.

V.3 References: Cloud Computing & Data Management (2/3)

- K.S. Beyer et al., « Jaql: a scipt language for large scale semi-structured data analysis », Proc. of VLDB Conf., 2011.
- Campbell et al.; Cloudy Skies for Data Management, ICDE'201
- R. Chaiken et al., « SCOPE: easy and efficient parallel processing of massive data sets », Proc. of VLDB Conf., 2008.
- S. Chaudhuri, « What next?: a half-dozen data management research goals for big data and the cloud », Proc. of PODS 2012.
- F. Chang et al., « Bigtable: A Distributed Storage System for Structured Data », ACM Trans. Comput. Syst. 26(2), 2008.
- B. F. Cooper et al., « PNUTS: Yahoo!'s hosted data serving platform », Proc. of VLDB, 2008.

V.3 References: Cloud Computing & Data Management (3/6)

- J. Dean, G. Ghemawat, « MapReduce: simplified data processing on large clusters », Proc. of OSDI Conf., 2004.
- G. De Candia, et al., « Dynamo: amazon's highly available key-value store », Proc. of the 21st ACM Symp. on Operating Systems Principles, 2007.
- A. Floratou, et al., « Can the Elephants Handle the NoSQL Onslaught? », Proc. of the VLDB Endowment, 2012.
- A.F. Gates, et al., « Building a High-level Dataflow system on top of Map-Reduce: The Pig Experience », Proc.of VLDB Conf., 2009.
- S. Ghemawat, et al., « The Google File System », Proc. of the 19th ACM symposium on Operating Systems Principles, 2003.
- Hadoop. http://hadoop.apache.org
- F. Deprez et al., «Special Theme : Cloud Computing, Platforms, Software and Applications », in ERCIM News, Number 83, Oct. 2010, pp. 12 – 51.

V.3 References: Cloud Computing & Data Management (4/6)

- T. Kaldewey, et al., « Clydesdale: structured data processing on MapReduce», Proc. of EDBT Conf., 2012.
- A. Lakshman, P. Malik, « Cassandra: a decentralized structured storage system », Operating Systems Review, 44(2), 2010.
- R. S. G. Lanzelotte, P. Valduriez, « Extending the Search Strategy in a Query Optimizer », Proc. of VLDB Conf., 1991.
- V. Narasayya, et al., « SQLVM: Performance Isolation in Mutli-tenant Relational Database-as_a_Service », Proc of CIDR'13, January 2013, Asilomar, CA, USA
- C. Olston, et al., « Pig Latin: a not-so-foreign language for data processing », Proc. of Sigmod Conf., 2008.
- C. Collet et al.; « De la gestion des bases de données à la gestion de grands espaces de données», Comité Bases de Données Avancées; July 2012.
- Maria Indrawan-Santiago, « Database Research: Are We At A Crossroad », Proc. of NBIS 2012, Melbourne, Australia, Sept. 26-28; pp. 45-51.

V.3 References: Cloud Computing & Data Management (5/6)

- A. Paramswaran, "An interview with S. Chaudhuri" , In: XRD Vol. 19, No. 1, Sept. 2012
- M. Stonebraker, et al., « MapReduce and Parallel DBMSs: friends or foes? », Commun. ACM 53(1), 2010.
- Thakar & Szalay; Migration a large Science DB to the Cloud, HPDC'2011
- A. Thusoo, et al., « Hive- a warehousing solution over a MapReduce framework », Proc. of VLDB Conf., 2009.
- A. Thusoo, et al., « Hive- a petabyte scale data warehouse using Hadoop », Proc. of ICDE Conf., 2010.
- Y. Yu et al., « DryadLINQ: a system for general purpose distributed data-parallel computing using a high level language », Proc. of OSDI Conf., 2008.
- J. Zhou, et al., « SCOPE : Parallel databases meet MapReduce », VLDB Jounal, 2012.
- M.F. Sakr et al.; "Center of Gravity Reduce Task Scheduling to Lower MapReduce Network Trafic"; IEEE Cloud Conf. , 2012, pp. 49-58.
- S. Ibrahim, et al.; " LEE: Locality/fairness-aware key partitioning for MapReduce in the Cloud"; Conf. on Cloud Computing Technology & Science; pp. 17 – 24.

V.3 References: Cloud Computing & Data Management (6/6)

- F. Li et al., "Distributed Data Management Using MapReduce"; ACM CS, Vol. 46. No. 3, January 2014.
- G. Graefe et al. "Elasticity in Cloud Databases and Their Query Processing"; Intl Journal of Data Warehousing and Mining, Vol. 9, No. 2 April-June 2013
- P. Unterbrunner et al.; "High availability, elasticity, and strong consistency for massively parallel scans over relational data"; in VLDB Jo, Vol. 23, pp. 627-652, 2014.
- P. Valduriez, « Indexing and Processing Big Data"; Seminar: Mastodons Indexing Scientific Big Data, Paris, January 2014.
- C. Doulkeridis, K. Norvag, "A Survey of Large-scale Analytical Query Processing in MapReduce"; VLDB Journal, 23(3), 2014
- Liroz-Gistau et al. " Data Partitioning for Minimizing Transferred Data in MapReduce" in: Globe Conf. , 2013, p. 1 – 12; Also, in: PhD Thesis, Dec. 2013
- A. Hameurlain, , «Large-scale Data Management Approaches: Evolution and Challenges ». In: ACOMP 2013 (Invited Talk), Ho Chi Minh City, Vietnam, 23-25 Oct. 2013.

V.4 References (1/2): Query Optimization; Multi_Objective, SLA/SLO

➔ **Multi-Objective Query Optimization**

- **Trummer, I., and Koch, C. Approximation Schemes for Many-Objective Query Optimization. In Proceedings of the ACM SIGMOD international conference (SIGMOD '14) (Snowbird, UT, USA, June 22-27, 2014). ACM Press, New York, NY, 2014, 1299-1310.**
- **Trummer, I., and Koch, C. A Fast Randomized Algorithm for Multi-Objective Query Optimization. In Proceedings of the ACM SIGMOD international conference (SIGMOD '16) (San Francisco, USA, June 26th - July 1st, 2016). ACM Press, New York, NY, 2016.**
- **Kllapi, H., Sitaridi, E., Tsangaris, M. M., and Ioannidis, Y. Schedule optimization for data processing flows on the cloud. In Proceedings of the ACM SIGMOD international conference (SIGMOD '11) (Athens, Greece, June 12-16, 2011). ACM Press, New York, NY, 2011, 289-300.**

➔ **SLA/SLO Papers**

- **Ortiz, J., de Almeida, V. T., and Balazinska, M. Changing the Face of Database Cloud Services with Personalized Service Level Agreements. In Proceedings of the Seventh Biennial Conference on Innovative Data Systems Research (CIDR ’15) (Asilomar, CA, USA, January 4-7, 2015). Online Proceedings, www.cidrdb.org, 2015**
- **Lang, W., Shankar, S., Patel, J. M., and Kalhan, A. Towards Multi-tenant Performance SLOs. In Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE ’12) (Washington, DC, USA, 1-5 April, 2012). IEEE Computer Society, 2012, 702-713**

V.4 References (2/2): Multi-tenant DBMS

- F. Chong, G. Carraro, and R. Wolter. Multi-Tenant Data Architecture. Microsoft Corporation. June 2006.
- P.Wong, Z. He, and Eric Lo. Parallel Analytics as a Service. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD'13), 25-36.
- O. Schiller, B. Schiller A. Brodt, and B. Mitschang. Native Support of Multi-tenancy in RDBMS for Software as a Service. EDBT 2011, March 22–24, 2011, Uppsala, Sweden.
- Z. Tan, and S. Babu. Tempo: Robust and Self Tuning Resource Management in Multitenant Parallel Databases. Proc. of the VLDB Endowment, Vol. 9, No. 10, 2016, pp. 720-731.
- Petrie Wongy, Zhian He, Ziqiang Feng, Wenjian Xu, and Eric Lo. Thrifty: Offering Parallel Database as a Service using the Shared-Process Approach. In Proceedings of the ACM SIGMOD intl. conf. (SIGMOD'15), May 31–June 4, 2015, Melbourne, Victoria, Australia.

V.4 References: Data Replication in Cloud Env.(1/2)

- B.A. Milani, N.J. Navimipour. A comprehensive review of the data replication techniques in the cloud environments: major trends and future directions. Journal of Network and Computer Applications, 64 , pp. 229–238, (2016)
- Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng. CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster. Proc. of the IEEE Int. Conf. on Cluster Computing (CLUSTER), pp. 188-196, (2010).
- N. Bonvin, T. G. Papaioannou, K. Aberer. Autonomic SLA-driven Provisioning for Cloud Applications. Proc. of Int. Symp. on Cluster, Cloud and Grid Computing, pp. 434- 443, (2011).
- Z. Cheng, et al. ERMS: An Elastic Replication Management System for HDFS. Proc. of the IEEE Int. Conf. on Cluster Computing Workshops , pp. 32-40, (2012)
- W. Lang, S. Shankar, J. Patel, A. Kalhan. Towards Multi-Tenant Performance SLOs. IEEE Trans. On Knowledge and Data Engineering, V. 26, No. 6, pp. 702–713, (2014).
- J.-W. Lin, C.-H. Chen, and J.M. Chang, “QoS-Aware Data Replication for Data Intensive Applications in Cloud Computing Systems,” IEEE Trans. Cloud Computing, vol. 1, no. 1, pp. 101-115, June 2013

V.4 References: Data Replication in Cloud Env. (2/2)

- F. R. C. Sousa, J.C. Machado. Towards Elastic Multi-Tenant Database Replication with Quality of Service. In Proc. of Int. Conf on Utility and Cloud Computing, UCC '12, pp. 168-175. IEEE Computer Society, Washington, DC, USA, (2012)
- G. Silvestre, S. Monnet, R. Krishnaswamy & P. Sens. AREN: A Popularity Aware Replication Scheme for Cloud Storage. Int. Conf. on Parallel and Distributed Systems, pp. 189–196, (2012).
- K. A. Kumar et al.. SWORD: Workload-Aware Data Placement and Replica Selection for Cloud Data Management Systems. The VLDB Journal, Special Issue, Vol. 23, N. 6, pp. 845-870, (2014)
- Y. Mansouri, A.N. Toosi, R. Buyya. Cost optimization for dynamic replication and migration of data in cloud data centers. IEEE Transactions on Cloud Computing (2017).
- C.L. P. Chen and C- Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. Information Sciences, 275: pp. 314–347, (2014).
- P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu, and H. Hacigumus. Intelligent Management of Virtualized Resources for Database Systems in Cloud Environment. Proc. of Int. Conf. of Data Engineering (ICDE), pp. 87–98. (2011).