HaQoop: scientific workflows over BigData

3rd HOSCAR Meeting

Bordeaux, Sept. 2013

Fabio Porto, Douglas Ericson Oliveira
Matheus Bandini, Henrique Kloh
Reza Akbarinia, Patrick Valduriez

Outline

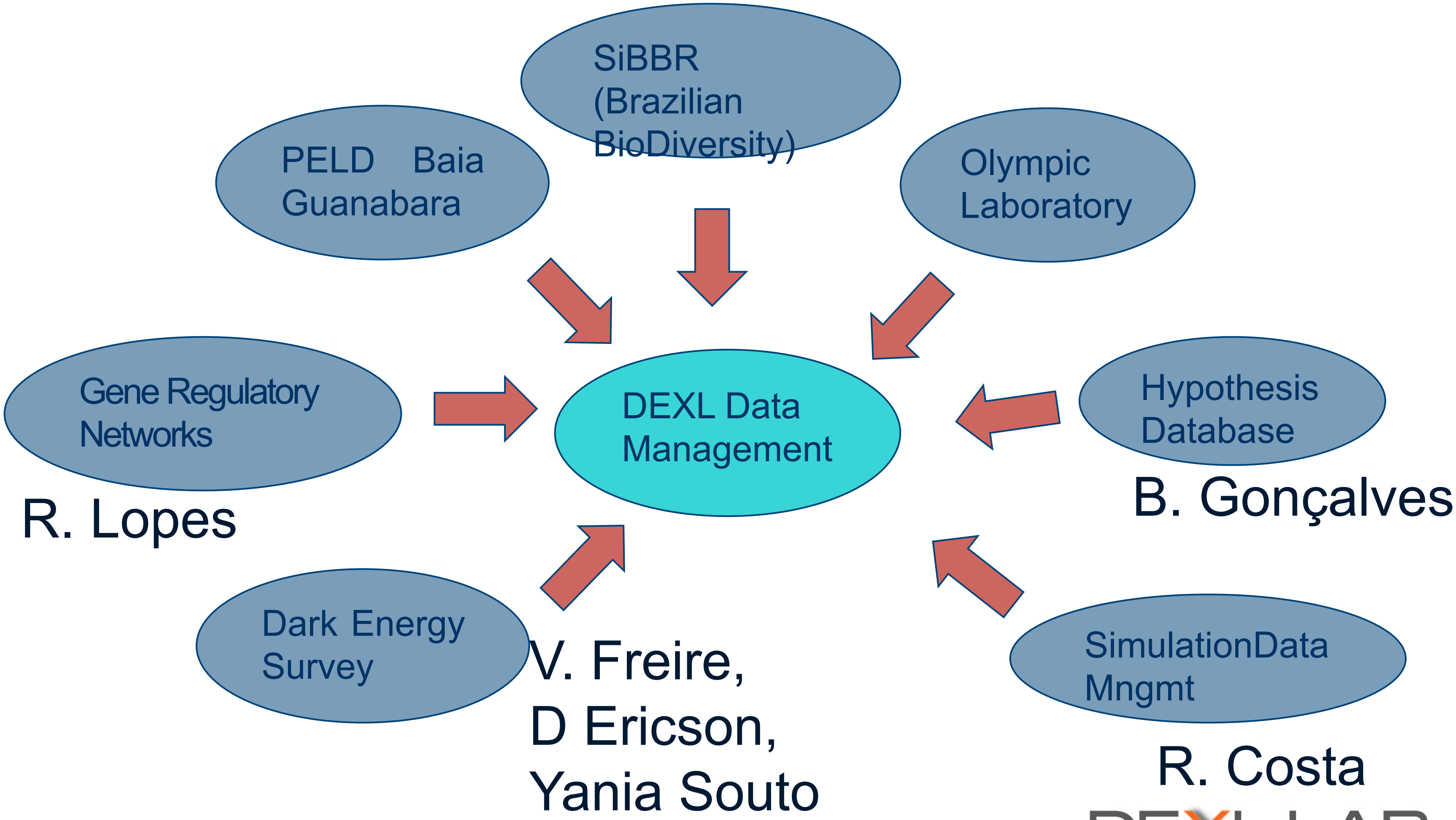
- Introduction
- Previous work in the collaboration
- HaQoop
- Initial experiments
- Final Comments

The Data eXtreme Lab (DEXL) Mission

- To support *in-silico* science with data management techniques;
  - To develop interdisciplinary research with contributions on data modelling, design and management;
  - To develop tools and systems in support to in-silico science;
- Currently
  - 3 researchers
  - 8 PhD students
  - 10 engineers
- Projects
  - Astronomy
  - Medicine
  - Sports Science
  - Biology, Ecology
  - Biodiversity

DEXL LAB
EXTREME DATA LAB

Current projects

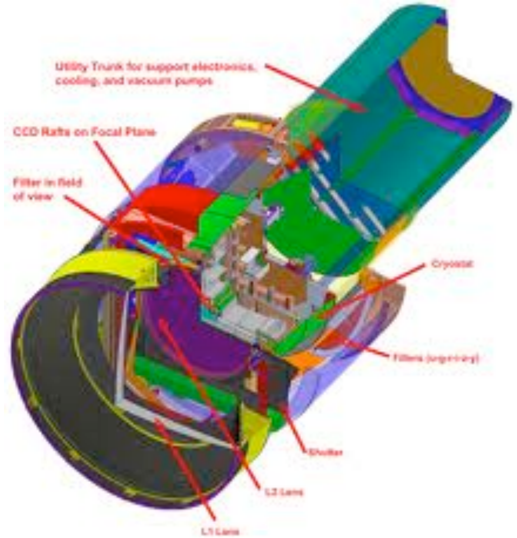


DEXL LAB
EXTREME DATA LAB

BigData for the *10s*

- BigData Processing and Analyses
  - Concerns with Obtaining
    - Volume, Variety, Velocity
  - Concerns with Usage
    - Sparse, infrequent
    - Exploratory, hypotheses driven
- Interested in processing scientific BigData

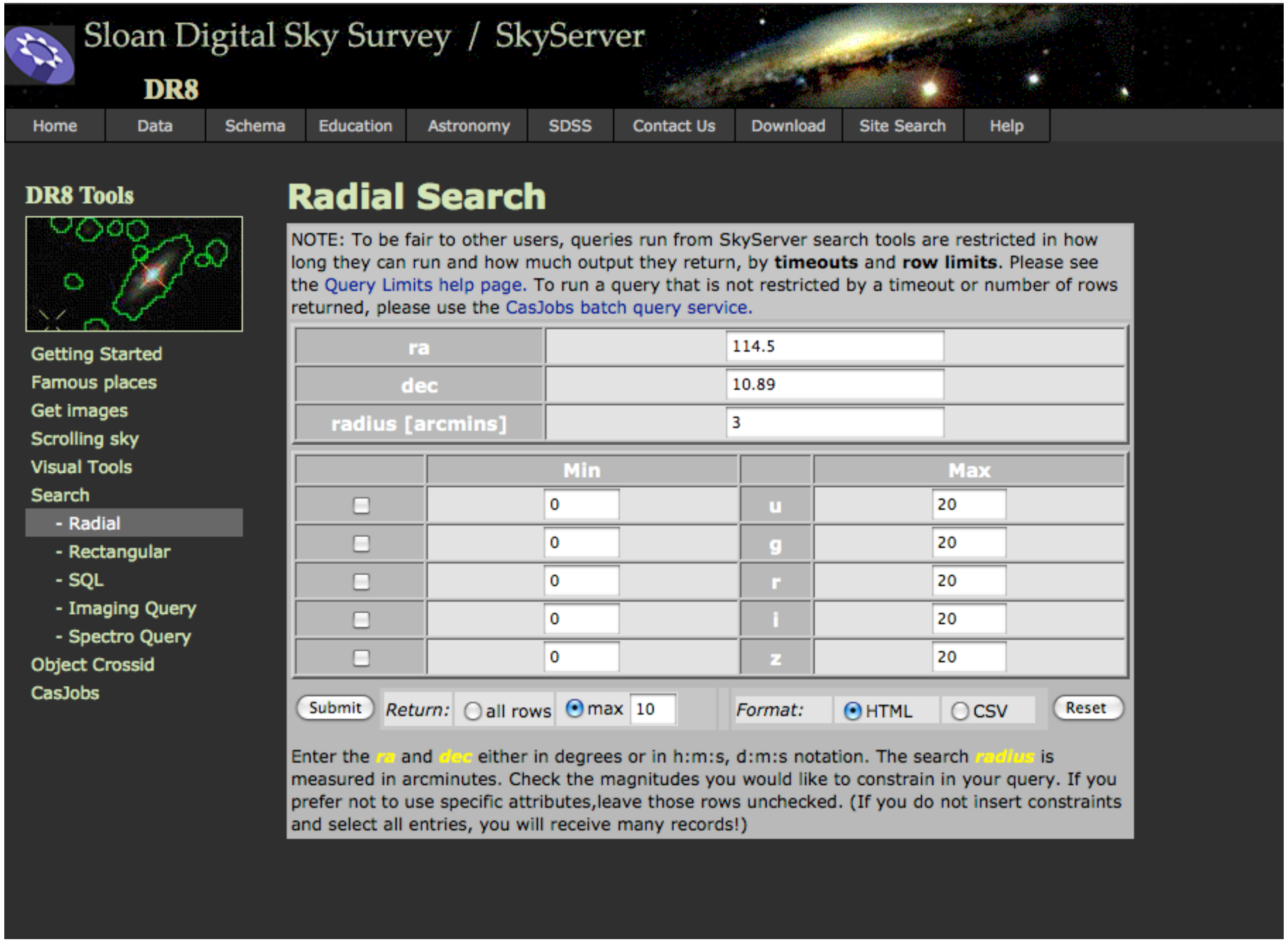
LSST – Large Synoptic Survey Telescope



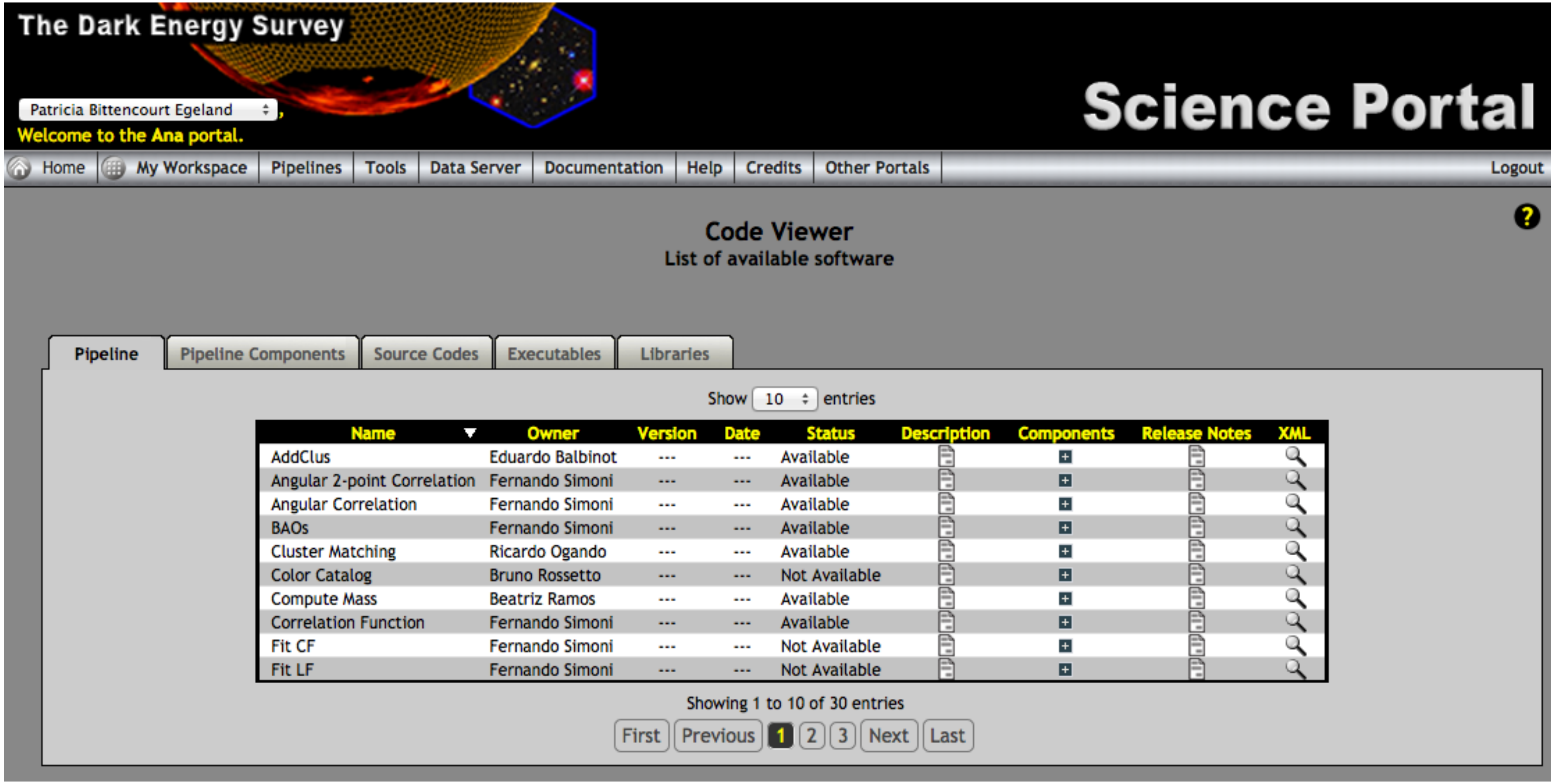
- 800 images p/ night during 10 years !!
- 3D Map of the Universe
- 30 TeraBytes per night
- **100 PetaBytes in 10 years**
  - **10^5 disks of 1 TB**

DEXL LAB
EXTREME DATA LAB

Skyserver – Sloan Project



Dark Energy survey - pipelines



The Dark Energy Survey

Science Portal

Patricia Bittencourt Egeland

Welcome to the Ana portal.

Home | My Workspace | Pipelines | Tools | Data Server | Documentation | Help | Credits | Other Portals | Logout

Code Viewer

List of available software

Pipeline | Pipeline Components | Source Codes | Executables | Libraries

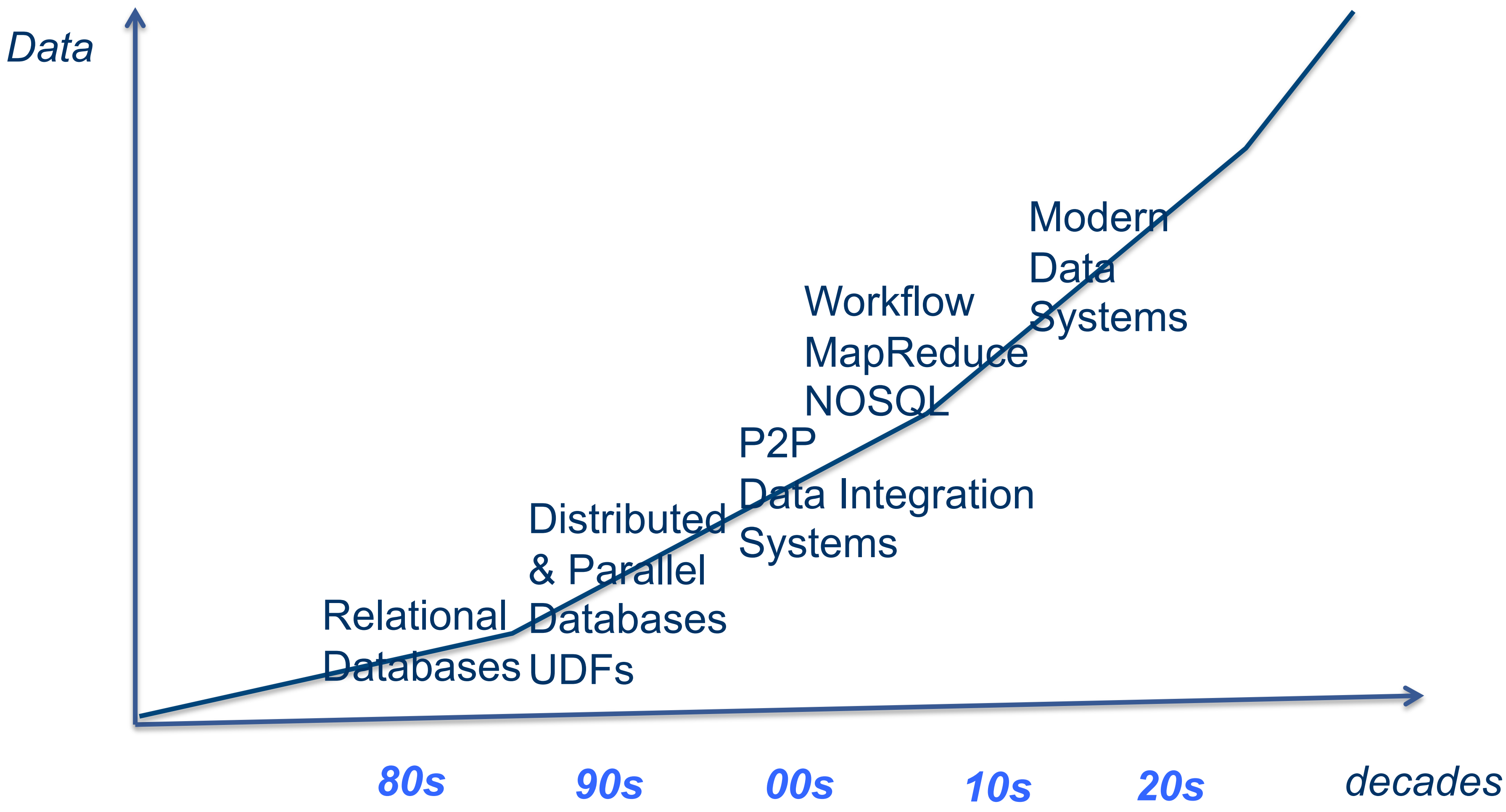
Show 10 entries

Name	Owner	Version	Date	Status	Description	Components	Release Notes	XML
AddClus	Eduardo Balbinot	---	---	Available				
Angular 2-point Correlation	Fernando Simoni	---	---	Available				
Angular Correlation	Fernando Simoni	---	---	Available				
BAOs	Fernando Simoni	---	---	Available				
Cluster Matching	Ricardo Ogando	---	---	Available				
Color Catalog	Bruno Rossetto	---	---	Not Available				
Compute Mass	Beatriz Ramos	---	---	Available				
Correlation Function	Fernando Simoni	---	---	Available				
Fit CF	Fernando Simoni	---	---	Not Available				
Fit LF	Fernando Simoni	---	---	Not Available				

Showing 1 to 10 of 30 entries

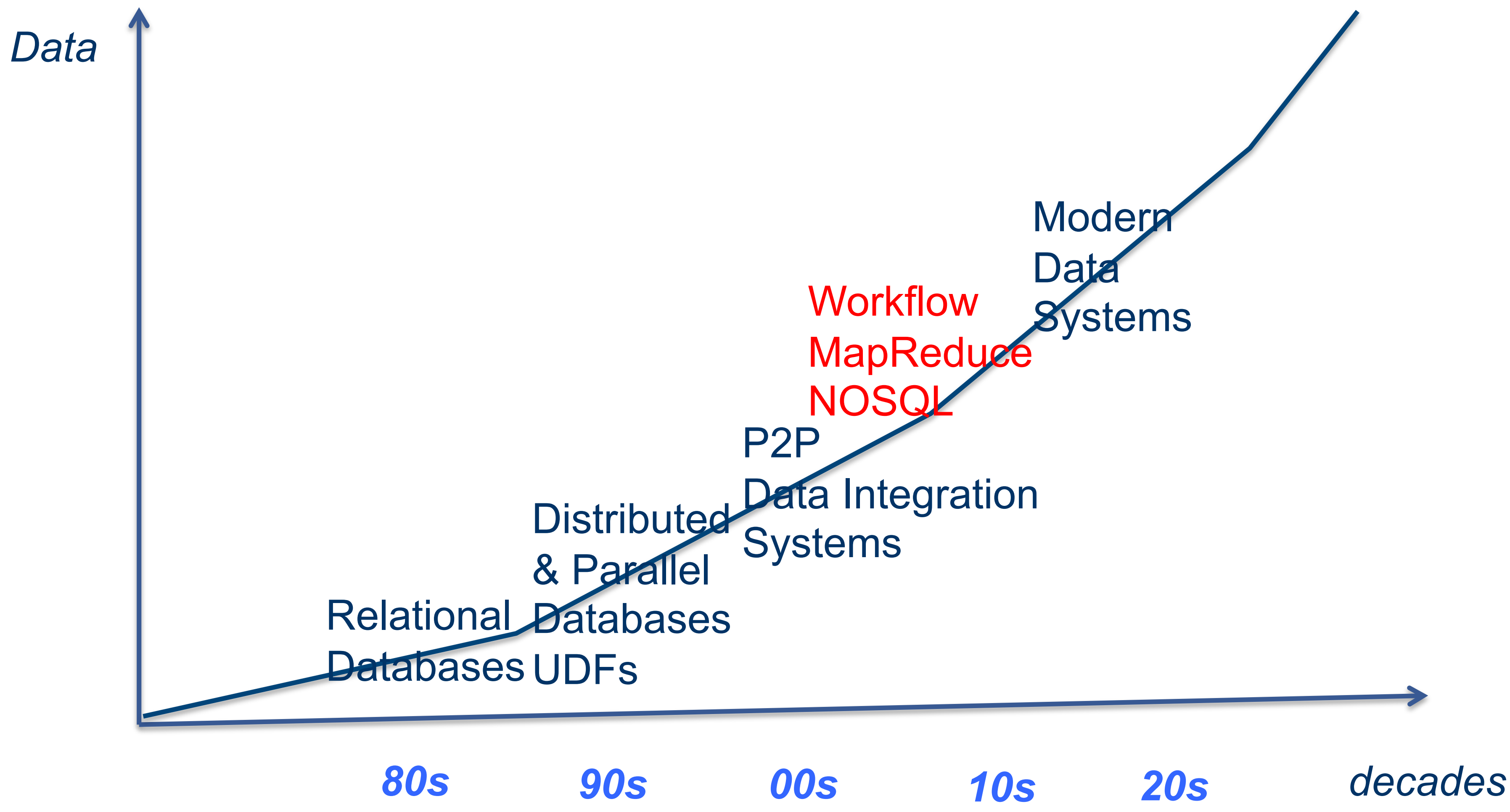
First | Previous | 1 | 2 | 3 | Next | Last

Data Processing Systems: an Evolution



DEXL LAB
EXTREME DATA LAB

Data Processing Systems: an Evolution



DEXL LAB
EXTREME DATA LAB

Data Processing Pillars

- Reduce the number of data retrieval operations
- Efficient iterative processing over elements of sets;
- Parallelism obtained by partitioning data;
  - Or pipelining data trough parallel execution of operators
- Explore the semantics of data operations;
- Automatic decisions based on data statistics;
- Data consumed by humans
- Data of simple structure/semantics

DEXL LAB
EXTREME DATA LAB

General Model

$$R \text{ ---- } f(x) \rightarrow R'$$

$$R_1 \text{ ----- } f(x) \rightarrow R_1'$$

$$R_2 \text{ ----- } f(x) \rightarrow R_2'$$

$$\ldots$$

$$R_n \text{ ----- } f(x) \rightarrow R_n'$$

WHAT CHANGES?

$$R''$$

Processing BigData

- Reduced data is still Big: millions of elements;
  - Access patterns less predictable
- Data may be:
  - Incomplete
  - Uncertain
  - Ambiguous
- Operation semantics are unknown (black box modules)
  - User code implementation
  - Arbitrary f (a workflow)
- Some operations are blocking, with respect to the consumption and production of data
  - Parallel MPI based programs
  - Prevent data-driven parallelism
- Consumption
  - Data analysis

Big Data Model

$$X \text{----} <T_1, M_2, \ldots, V_k> \text{---}\rightarrow Z'$$

$$X1 \text{-----} <T_1, M_2, \ldots, V_k> \text{---}\rightarrow X_1'$$

$$X2 \text{-----} <T_1, M_2, \ldots, V_k> \text{---}\rightarrow X_2'$$

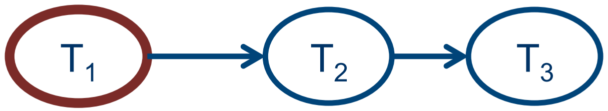
...

$$Xn \text{-----} <T_1, M_2, \ldots, V_k> \text{---}\rightarrow X_n'$$

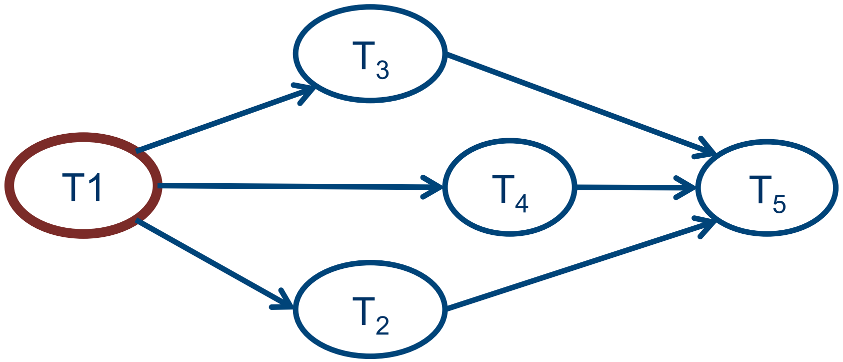
$$U_{i=1,n} X'_i \text{-----} g(y) \text{---}\rightarrow X''$$

DEXL LAB
EXTREME DATA LAB

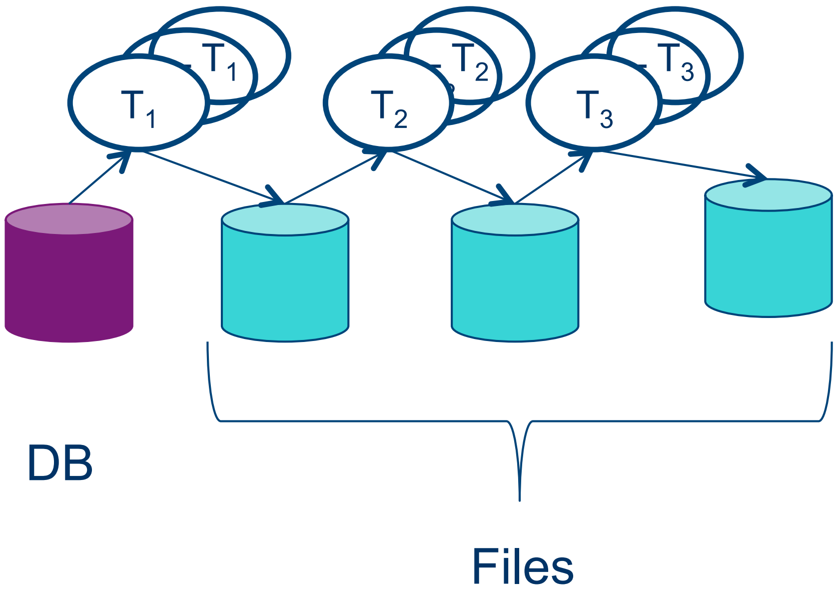
Workflow - Partial Ordering of $\mathcal{T}$



Where each ○ is an activity



Workflow DB – complete picture



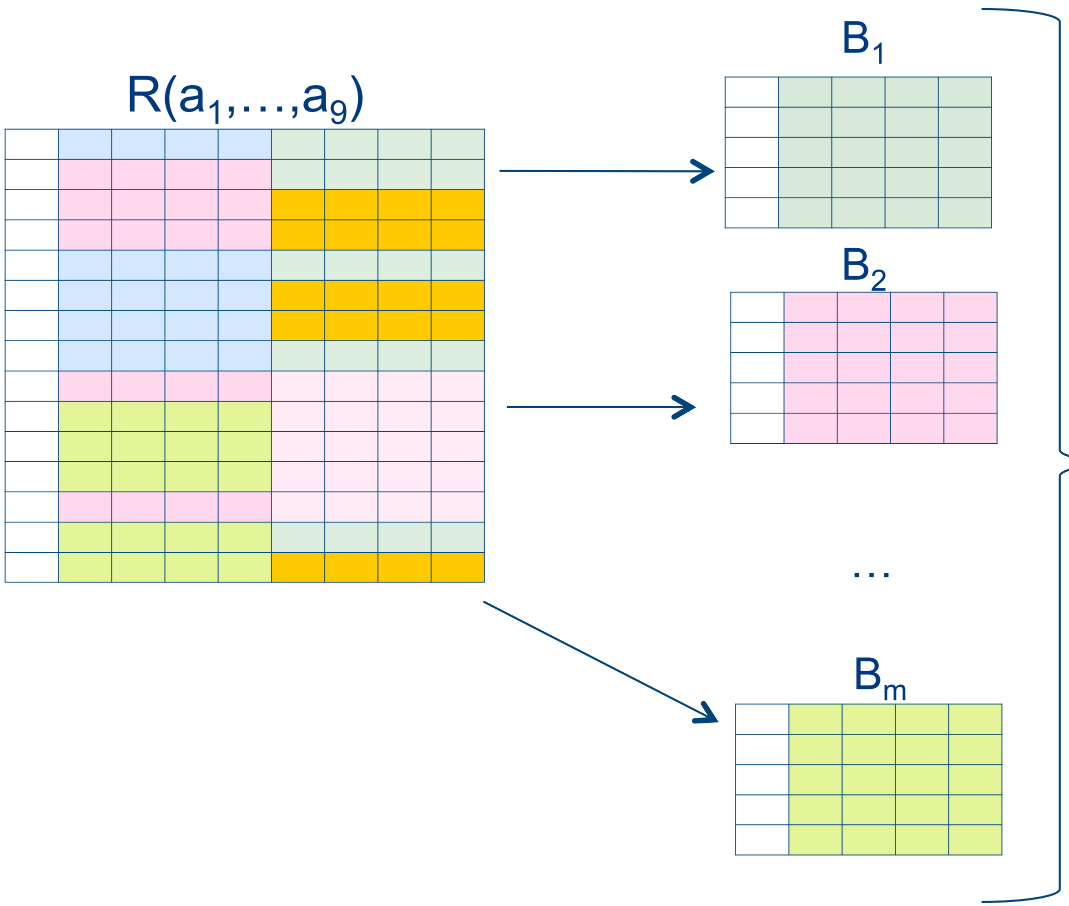
General Problem

- To Conceive an efficient and robust workflow execution strategy that considers data retrieved from databases and files produced in intermediate steps

DEXL LAB
EXTREME DATA LAB

PREVIOUS WORK IN THE COLLABORATION: LNCC, COPPE-UFRJ, INRIA - ZENITH

Partitioning the DB into Blocks Work with: Miguel Liroz-Gistau, Esther, Patrick, Reza



How to compute a partitioning strategy according to a known workload

DEXL LAB
EXTREME DATA LAB

Workflow algebra and optimization Eduardo Ogasawara, Marta Mattoso, Patrick Valduriez

- Scientific workflow definition mapped to a known data model
  - Input/output modelled as relations
  - workflow activities mapped to operators in a generic algebra;
    - Algebra operators describe input/output ratio
      - Enables automatic analysis of workflow definition according to type of applied data transformation
      - Enables automatic workflow transformation

DEXL LAB
EXTREME DATA LAB

Objective

- Processing big data by scientific workflows shall benefit from known data processing techniques
  - Activities semantics
  - Process to data locality
  - Optimize data and files distribution
  - Use generic MapReduce parallelism paradigma

DEXL LAB
EXTREME DATA LAB

Approach

- Use MapReduce paradigm to run scientific workflow
- Define a allocation strategy that considers:
  - The number of database partitions
  - The number of map tasks
  - The input/output semantics of workflow activities
  - The number of reduce tasks

Three scenarios evaluated

- Exploring experimentally variations on $|P|$, $|T|$, $|F|$ as the basis for the model:
  a) $|P| = 1$, $|T| >> 1$
  b) $|P| = |T| >> 1$, D is a distributed database
  c) $|P| \leq |T|$, $|P|$, $|T| >> 1$
- Which data processing parallel strategy leads to best results in workflow execution?

DEXL LAB
EXTREME DATA LAB

Parallel workflow evaluation on BigData

HaDooPDB

APACHE GIRAPH

OOZIE

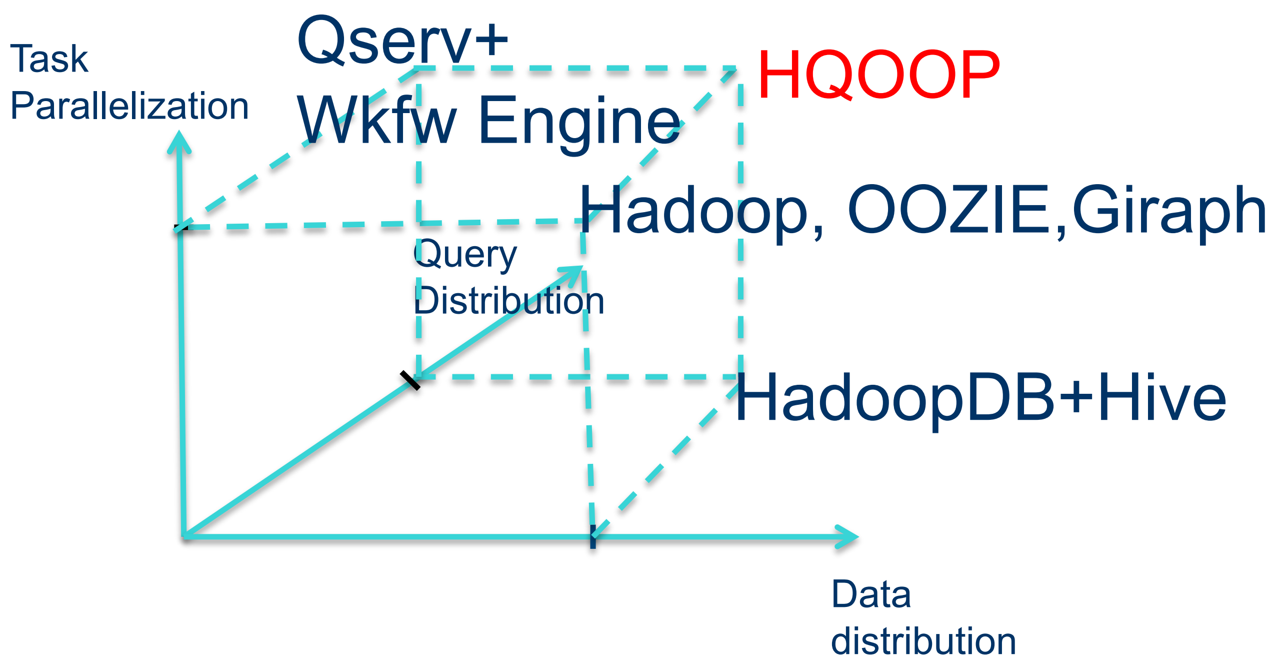
CHIRON

Dryad
LINQ
MS Research

hadoop

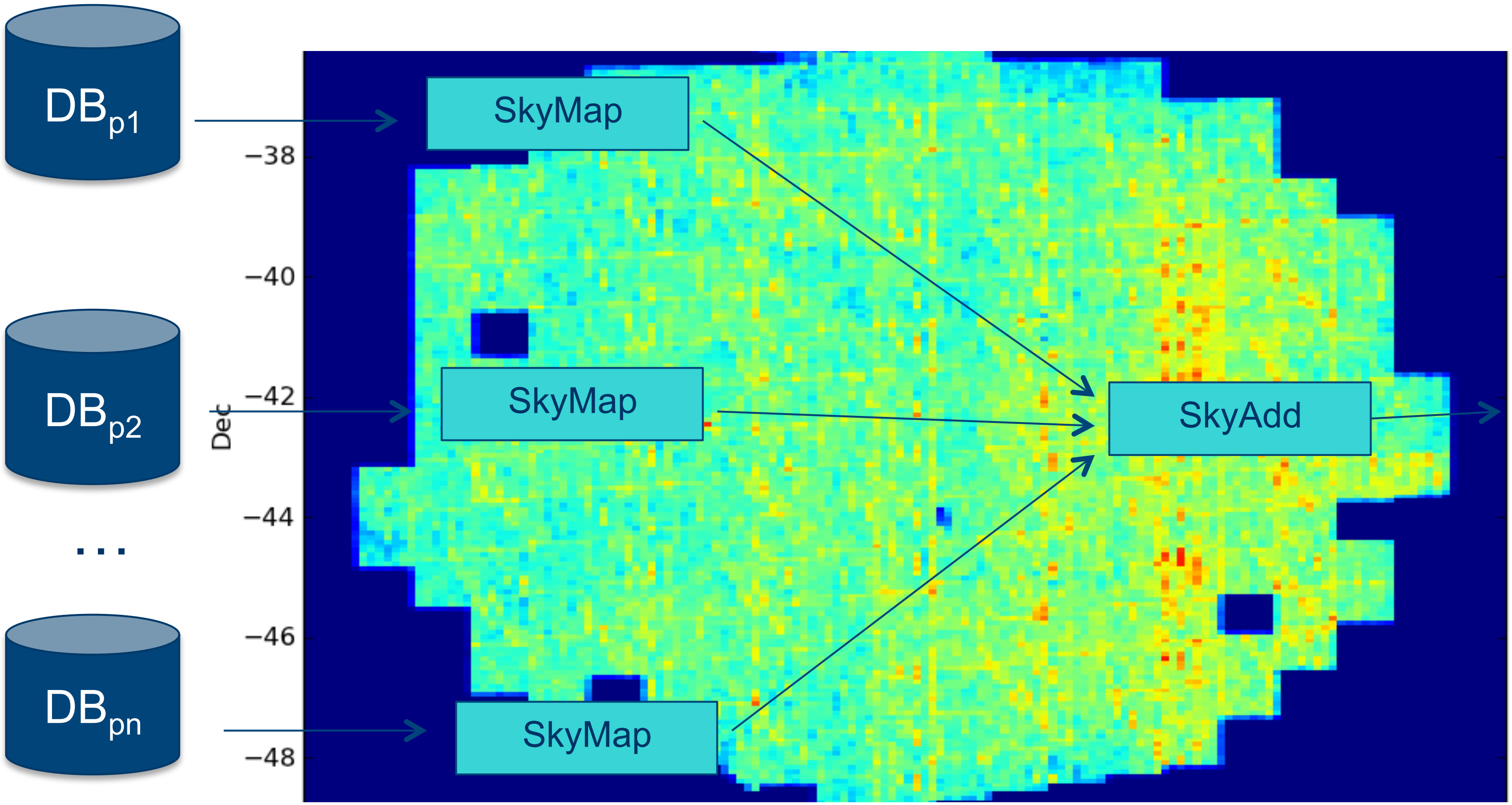
DEXL LAB
EXTREME DATA LAB

Architectural Viewpoint



Parallel workflow execution over Dark Energy Survey Catalog

Partitioned catalogue stored on PostgreSQL



HaQoop

- Hadoop – Open Source apache project
  - A state of the art task parallelization framework for Big Data processing
  - Split computation into two steps
    - Map (remember f ?)
    - Reduce (remember g ?)

- To reuse Hadoop scalability, fall tolerance
- To extend Hadoop with workflow expressions
  - Make f a general workflow engine (QEF)
- Restricted workflow expressions

DEXL LAB
EXTREME DATA LAB

QEF – Data Processing System

QEF
Query Evaluation Framework

- Designed based on principles of modern database query engines;
- Extendable for any user code
- Extendable for any data structure
- Can be downloaded: http://dexl.lncc.br/qef

Main technical characteristics

- Pipeline (iterator execution model)
- Iterations
- Algebraic/control operations
  - Allows both in-memory data exchange as file-based i/o
  - Run in both CPUs and GPUs
  - Push and pull data execution (using control operations)
- Dynamic optimization
  - Block-size computation
- Global and local state
  - Control tuples
- Catalog
  - Environment
  - Statistics
  - Metadata
- Synchronous and asynchronous execution

QEF as a Mappers & Reduce Job on Hadoop

$$\mathcal{X}_1 \text{ ------ } \boxed{<\mathcal{T}_1, \mathcal{M}_2, \ldots, \mathcal{V}_k>} \text{ ---}\rightarrow \mathcal{X}_1'$$

$$\mathcal{X}_2 \text{ ------ } \boxed{<\mathcal{T}_1, \mathcal{M}_2, \ldots, \mathcal{V}_k>} \text{ ---}\rightarrow \mathcal{X}_2'$$

...

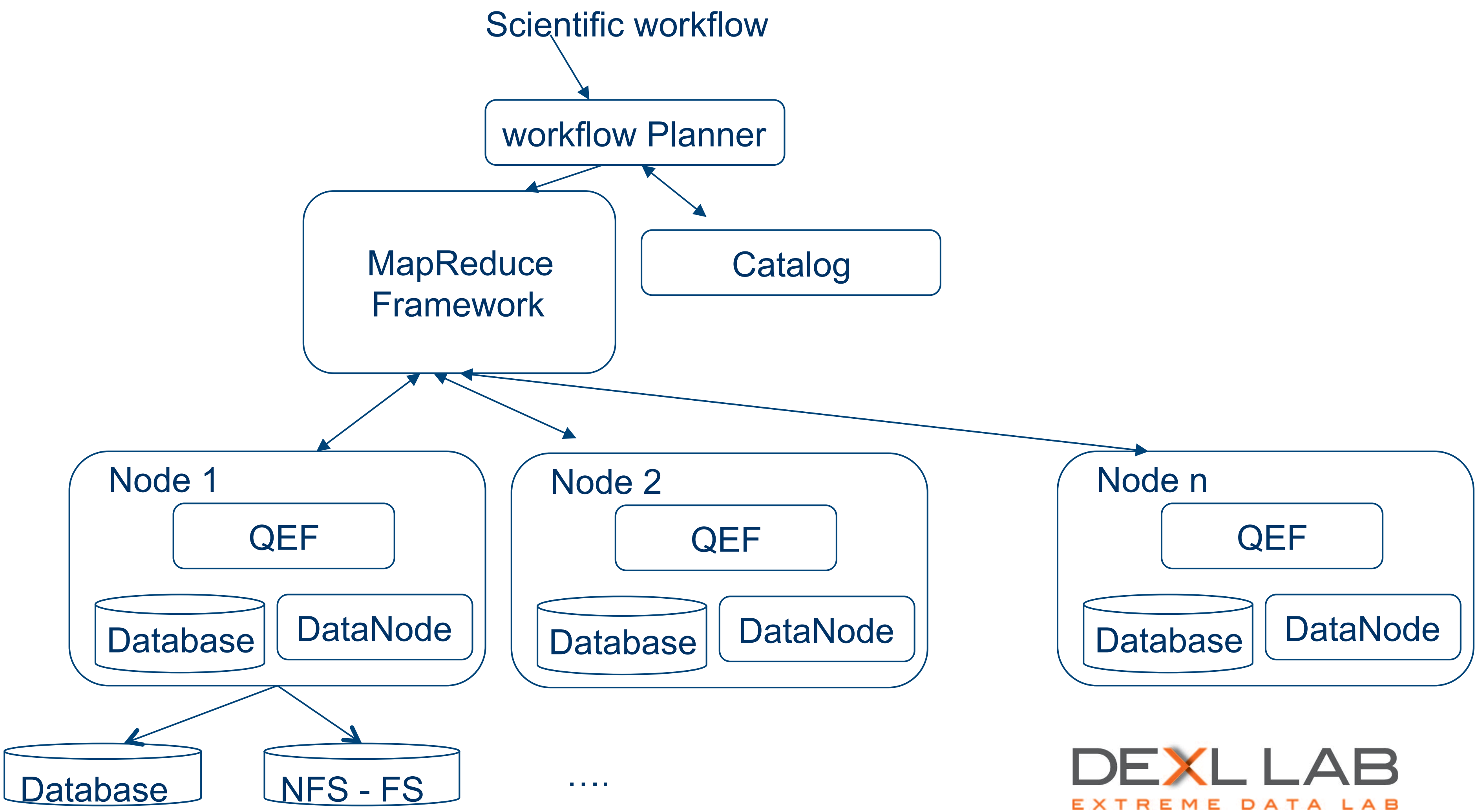
$$\mathcal{X}_n \text{ ------ } \boxed{<\mathcal{T}_1, \mathcal{M}_2, \ldots, \mathcal{V}_k>} \text{ ---}\rightarrow \mathcal{X}_n'$$

$$\mathcal{U}_{i=1,n}\mathcal{X}'_i \text{ ------ } \boxed{g(y)} \text{ ---}\rightarrow \mathcal{X}''$$

DEXL LAB

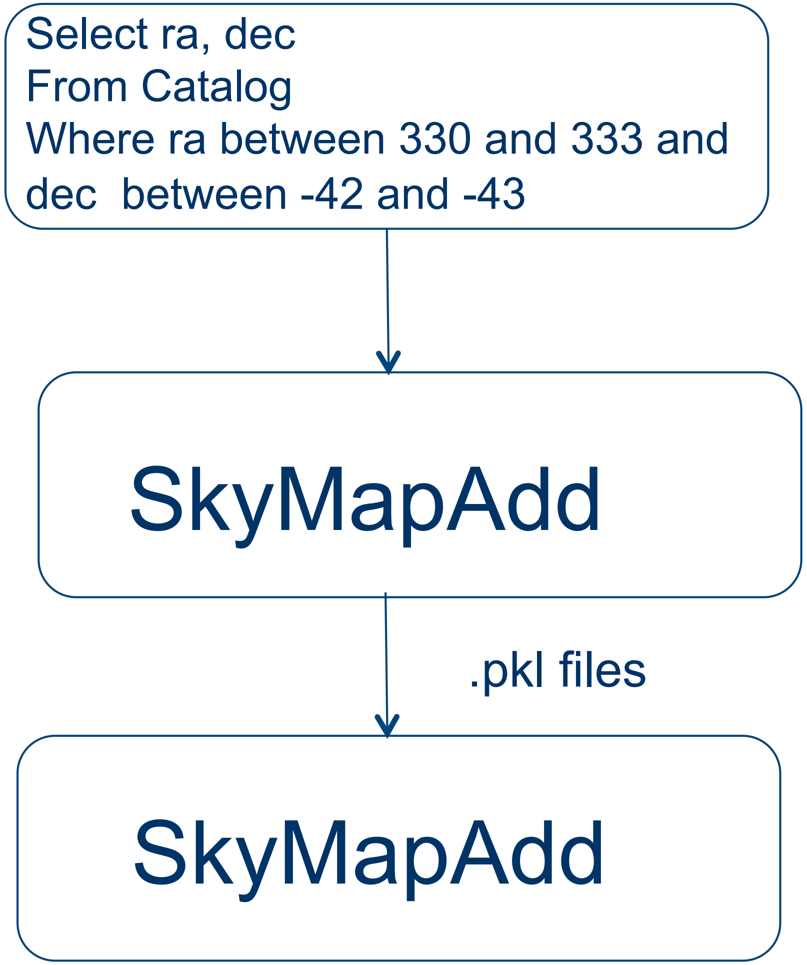
EXTREME DATA LAB

HaQoop architecture



DEXL LAB
EXTREME DATA LAB

Example: SkyMap Workflow



Catalog table
 - query returns 200 million sky objects
 - uniformly distributed through nodes
 - centralized mode
    each tuple is logically partitioned

Example

a) Catalog Table uniformly partitioned

QEF

Map

SCAN

Select ra, dec
From Catalog
Where ra between 330 and 333 and
dec between -42 and -43

Map

SkyMap

Reduce

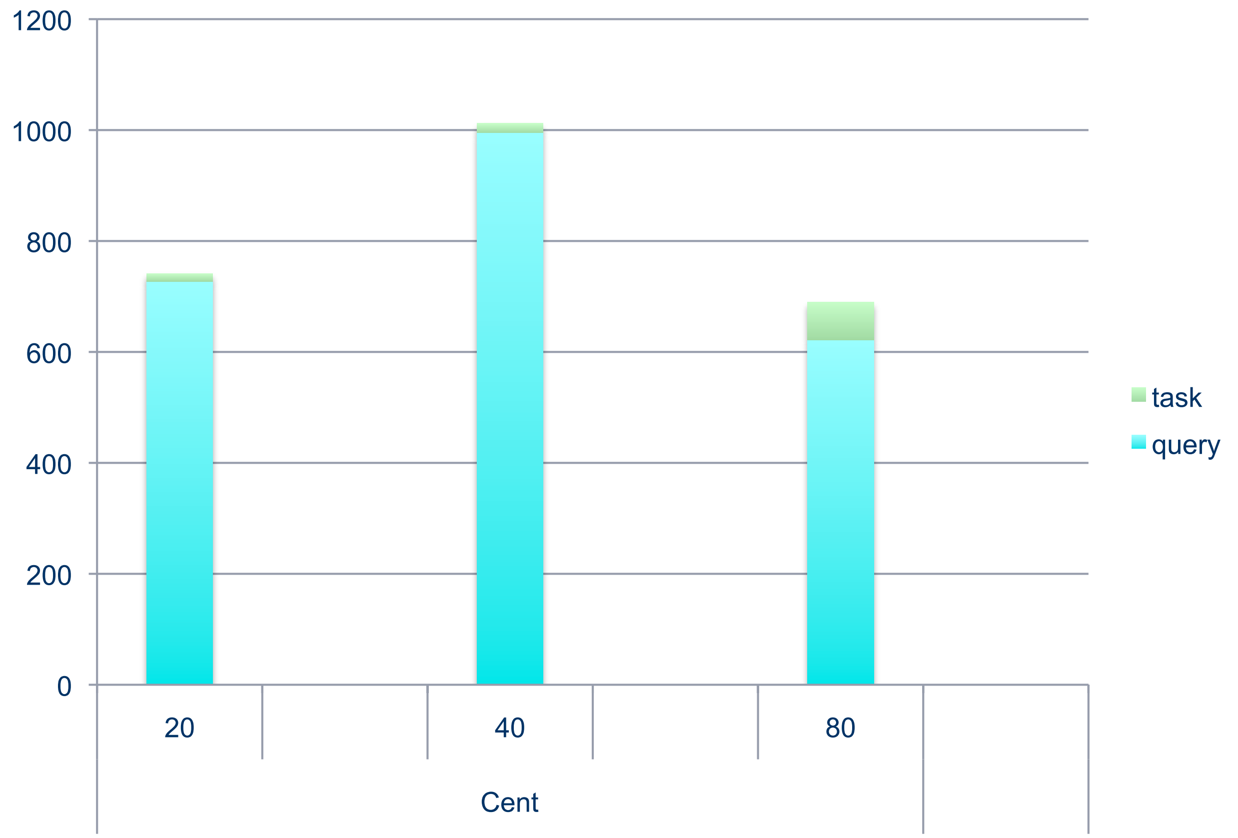
SkyMapAdd

Initial Experiments

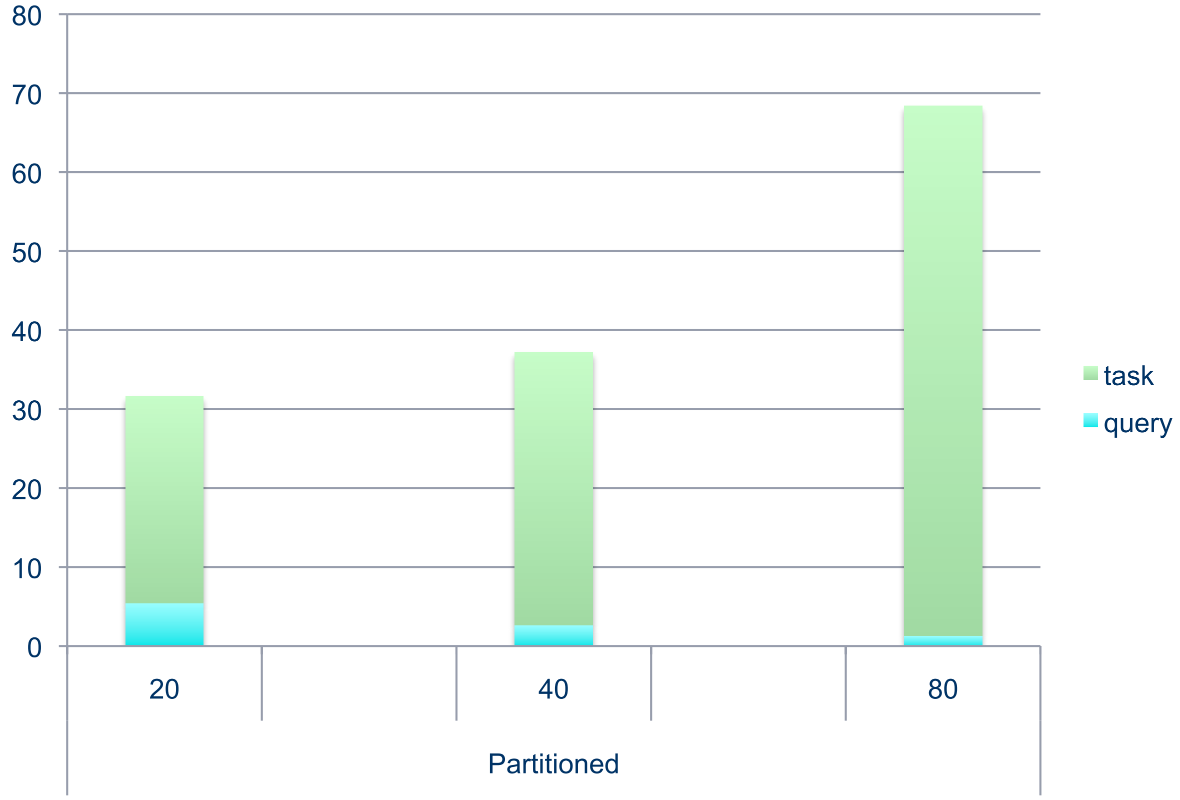
- Initial experiments
  - Skymap scenario;
- Cluster SGI
  - Configurations: 20, 40 and 80 nodes;
  - Each node:
    - 2 proc. Intel Zeon – X5650, 6 cores, 2.67 GHz
    - 24 GB RAM
    - 500 GB HD
- Data
  - DES Catalog DC6B
- Tasks
  - Python
- HAQOOP
- Centralized version
  - PostgreSQL 9.1
- Distributed
  - Pg_pool
- Partirioned
  - Multiple postgreSQL

DEXL LAB
EXTREME DATA LAB

Centralized – Elapsed-time (s)



Partitioned DB – Elapsed-time (s)



DEXL LAB
EXTREME DATA LAB

Final comments

- Collaboration with Zenith-Inria team
- Probable PhD student exchange in 2014

DEXL LAB
EXTREME DATA LAB

MERCI – OBRIGADO

fporto@lncc.br

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

FAPERJ
Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro

COPPE 50
UFRJ

Zenith

informatics mathematics
Inria

ComCiDis
Computação Científica Distribuída

LIneA
Laboratório Interinstitucional de e-Astronomia

DEXL LAB
EXTREME DATA LAB

Processing Scientific Workflows

- Analytical Workflows process a large part of Catalog data
  - Catalogs are supported by few indexes, thus most queries scan tens-to-hundreds of millions of tuples
- Parallelization comes as a rescue to reduce analyses elapsed-time, but
  - Compromise between:
    - Data partitioning and degree of parallelization;
  - Current solutions consider:
    - Centralized files to be distributed through nodes (MapReduce)
      - [Alagianins, SIGMOD, 2012] NoDB – reading raw files without data ingestion;
    - Distributed databases (Qserv) to serve Workflow engines
      - [Wang.D.L,2011], Qserv: A Distributed Shared-Nothing Database for the LSST catalog;
    - Centralized databases to serve Workflow Engine (Orchestration LineA)
    - Partitioned database to serve distributed queries (HadoopDB)

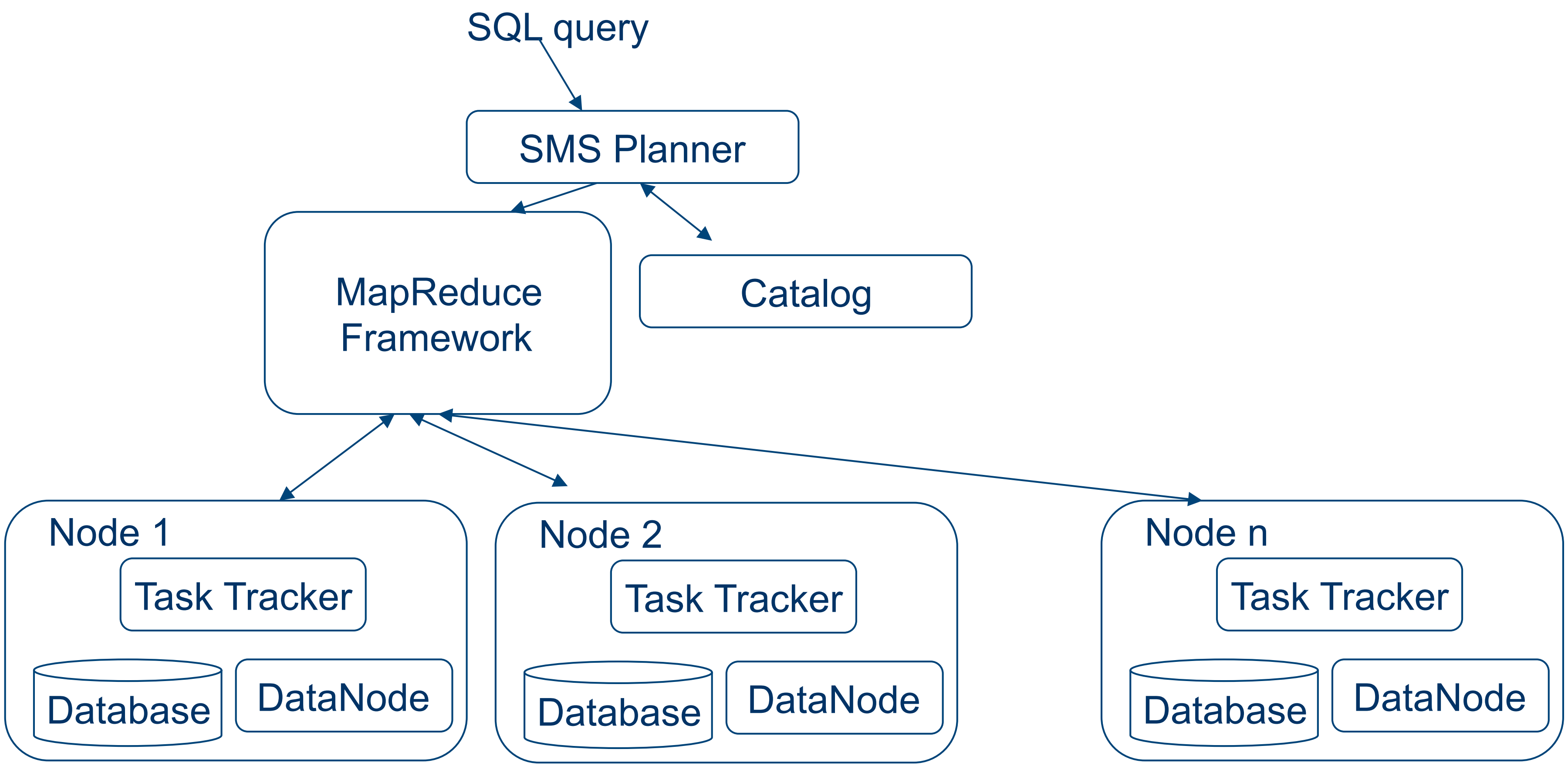
DEXL LAB
EXTREME DATA LAB

HadoopDB - a step in between [Abouzeid09]

- Offers parallelism and fault tolerance as Hadoop, with SQL queries pushed-down to postgreSQL DBMS;
- Pushed-down queries are implemented as Map-reduce functions;
- Data are partitioned through nodes.
  - Partitioning information stored in the catalog
  - Distributed through the N nodes

DEXL LAB
EXTREME DATA LAB

HadoopDB architecture



DEXL LAB
EXTREME DATA LAB

Example

Select year(SalesDate),sum(revenue)
From Sales
Group by year(salesDate)

a) Table partitioned by year(SalesDate)

Map

FileSink Operator

Select Year(SalesDate),
Sum(revenue)
From Sales
Group by year(salesDate)

b) no partitioning by year(SalesDate)

Reduce

FileSink Operator

Sum Operator

Group by Operator

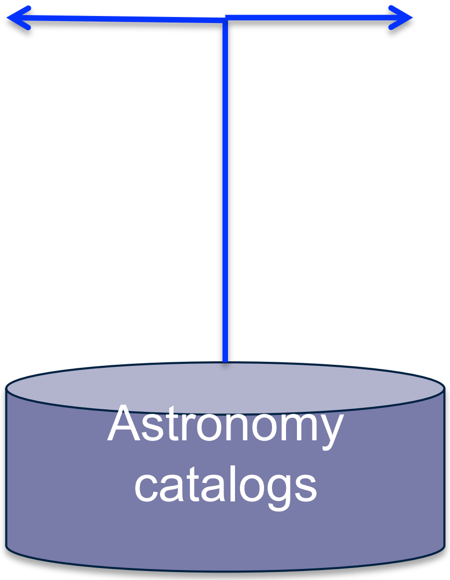
Map

Reduce Sink Operator

Select Year(SalesDate),
Sum(revenue)
From Sales
Group by year(salesDate)

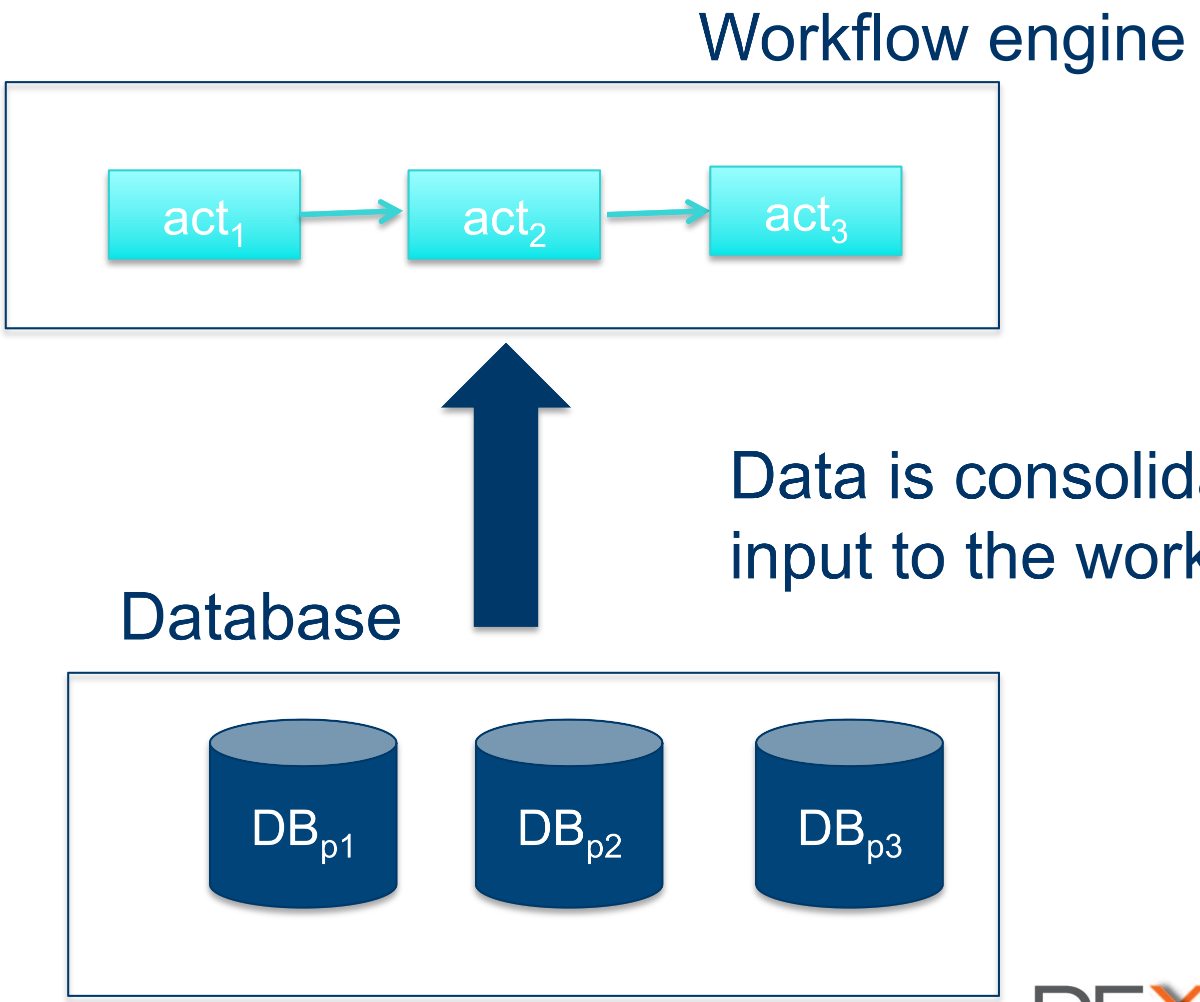
Processing Astronomy data

User access
- Ad-hoc queries
- downloads



Scientific workflows
- Analysis

Traditional WF–Database decoupled architecture



Data is consolidated as input to the workflow engine

Problems

- Data locality
  - Workflow activities run in remote nodes wrt the partitioned data;
- Load Balance
  - Local processes facing different processing time

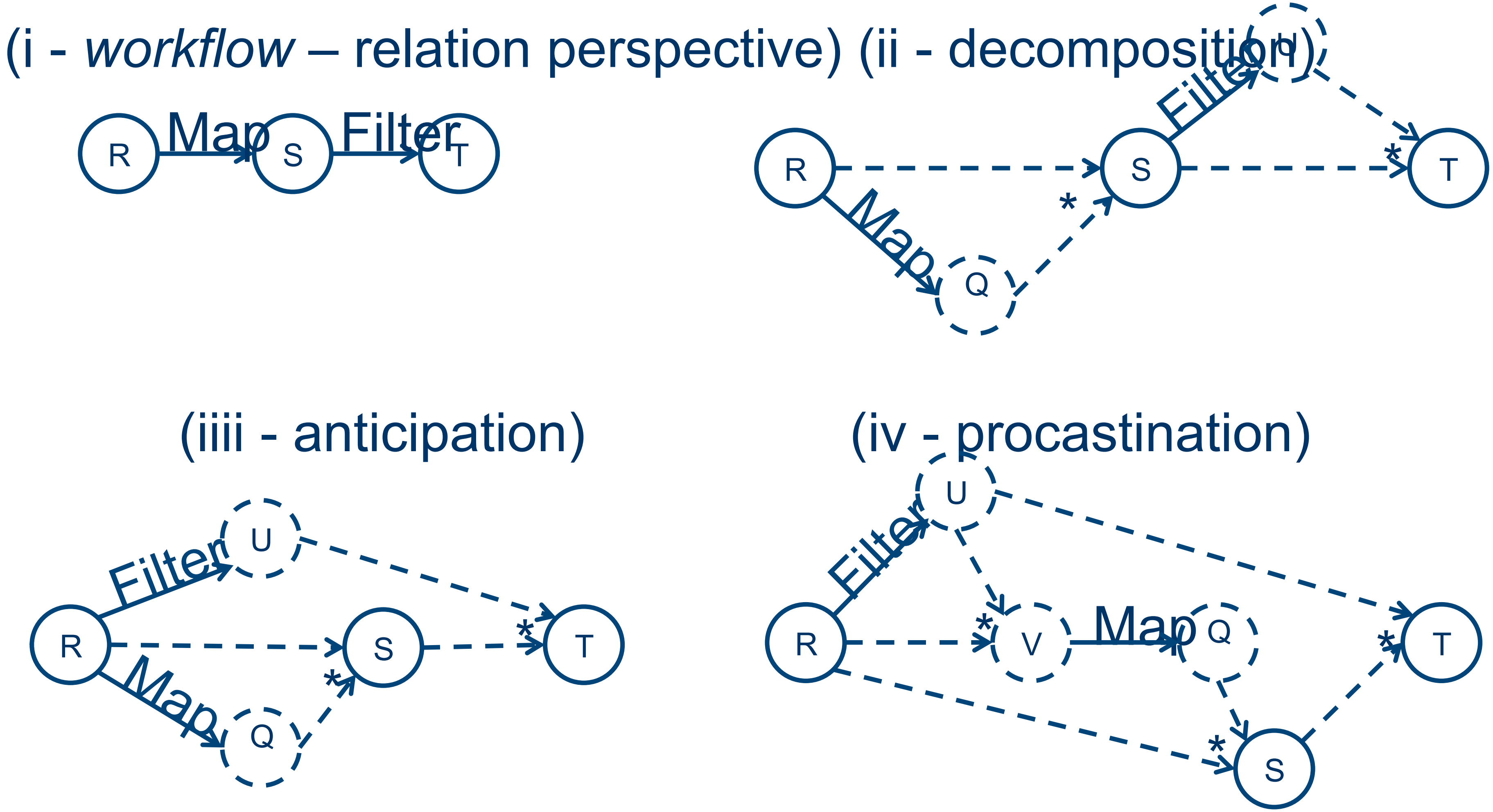
DEXL LAB
EXTREME DATA LAB

Data locality

- Traditional distributed query processing pushes operations through joins and unions so that can be done close to the data partitions;
- Can we “localize” workflow activities?
  - Moving activities in workflows require operation semantics to be exposed
  - Mapping of workflow activities to a known algebra
  - Equivalence of algebra expressions enabling pushing down operations

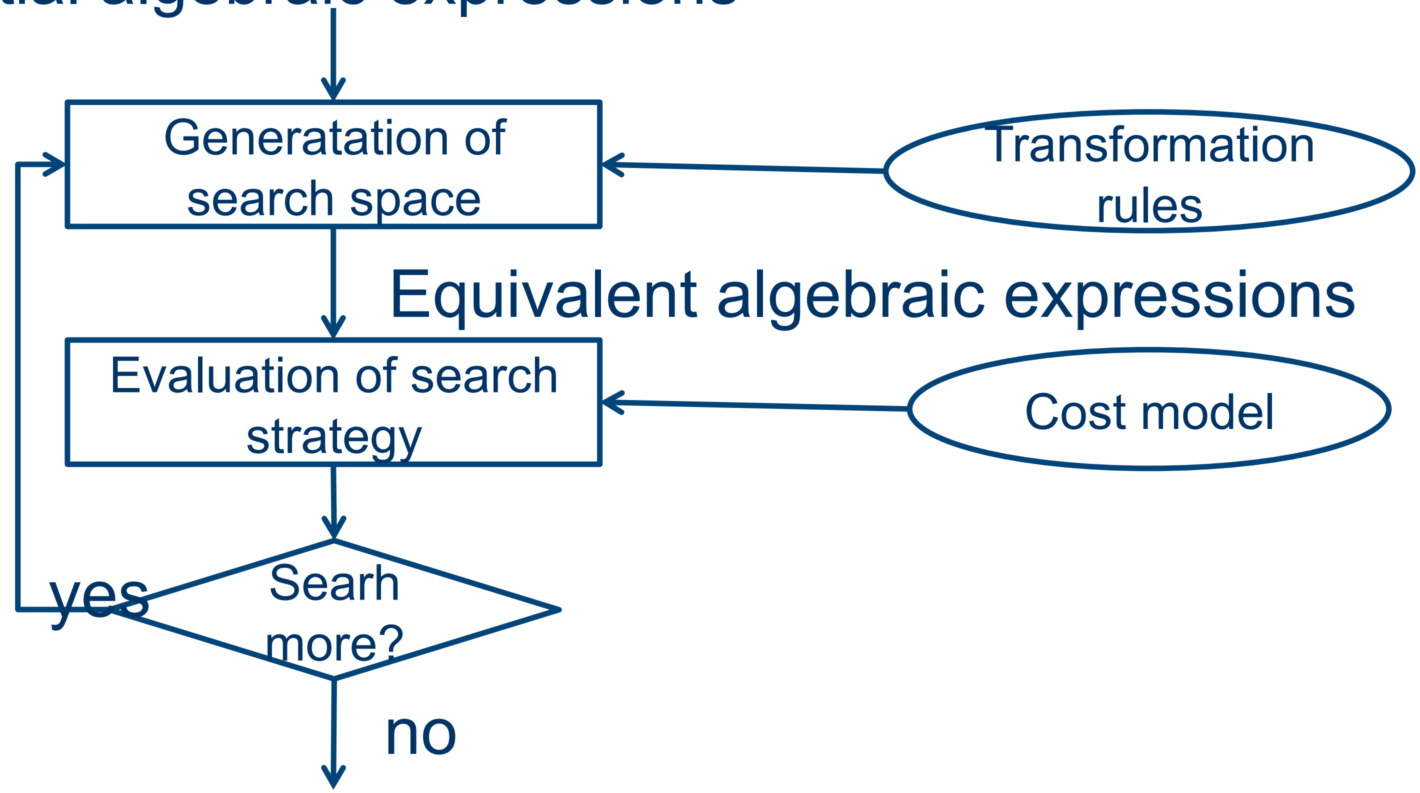
DEXL LAB
EXTREME DATA LAB

Algebraic transformation



Workflow optimization process

Initial algebraic expressions

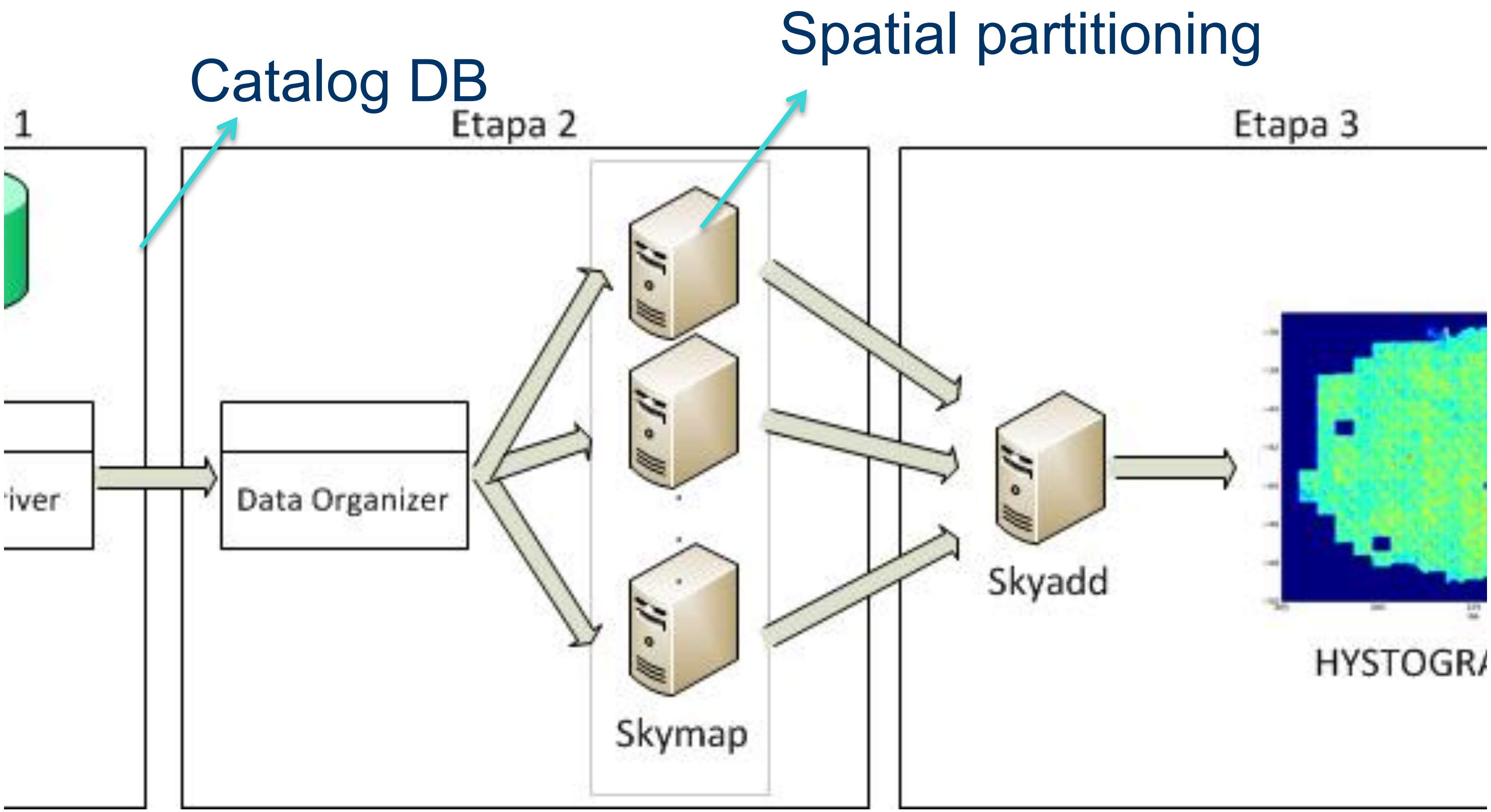


Optimized algebraic expressions

Pushing down workflow activities

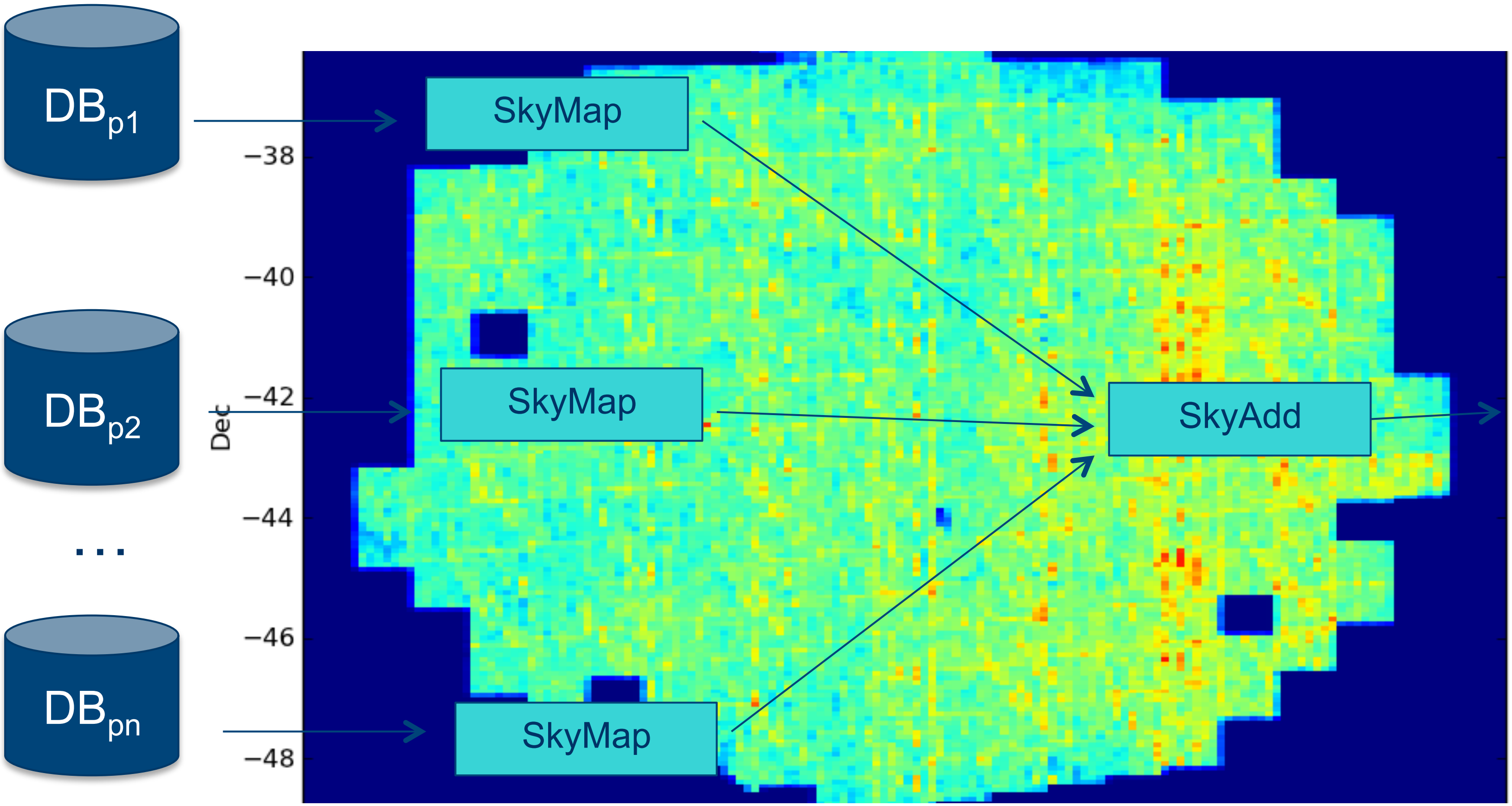
- A first naïve attempt
  - Push down all operations before a Reduce;
- Use a MapReduce implementation where
  - Mappers execute the “pushed-down” operations close to the data

Typical Implementation at LineA Portal



Parallel workflow over partitioned data

Partitioned catalogue stored on PostgreSQL



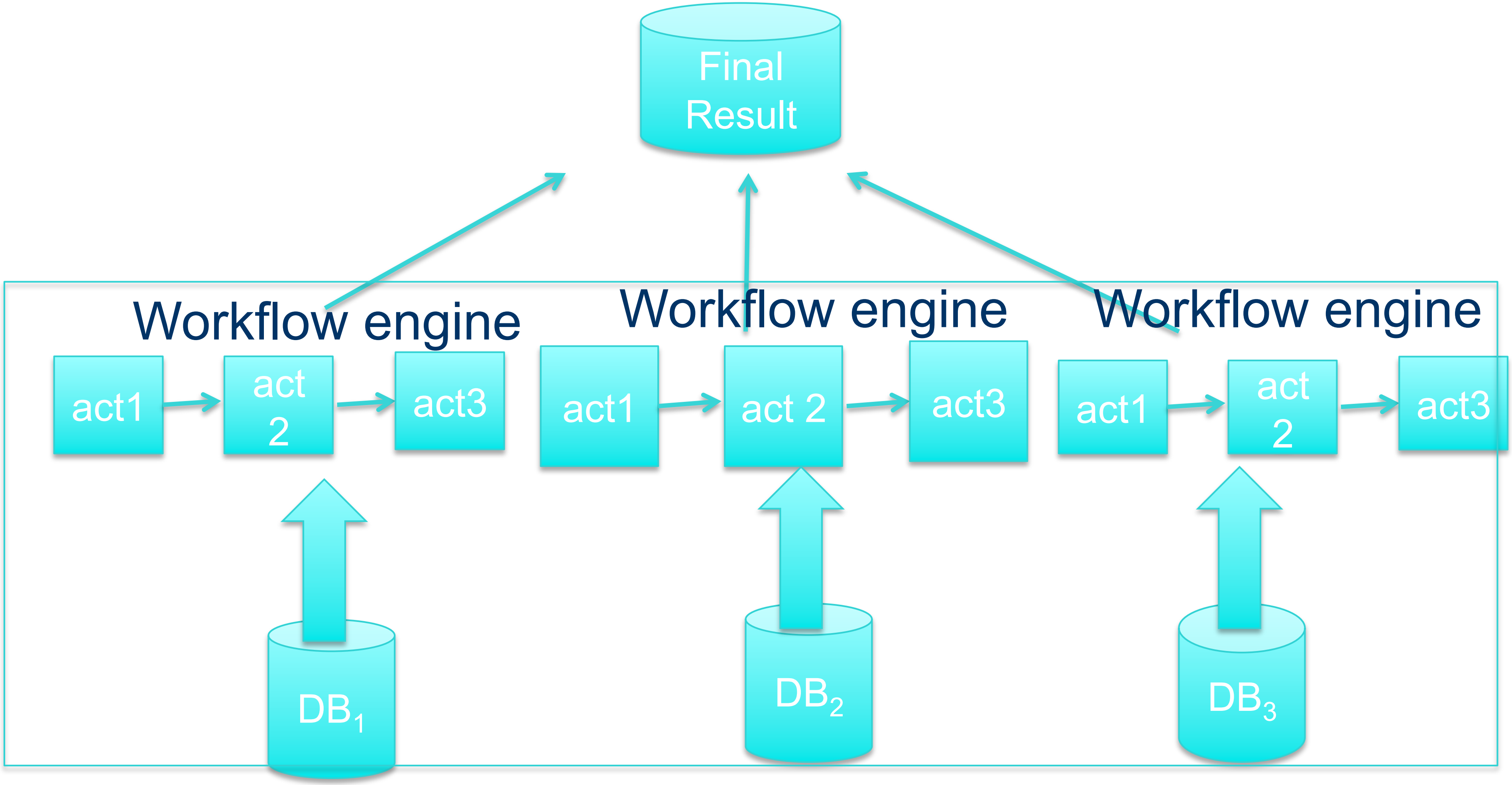
DEXL LAB
EXTREME DATA LAB

HQOOP - Parallelizing Pushed-down Scientific Workflows

- Partition of data across cluster nodes
  - Partitioning criteria
    - Spatial (currently used and necessary for some applications)
    - Random (possible in SkyMap)
    - Based on query workload (Miguel Liroz-Gestau's Work)
- Process the workflow close to data location
  - Reduce data transfer
- Use Apache/Hadoop Implementation to manage parallel execution
    - Widely used in Big Data processing;
    - Implements Map-Reduce programming paradigm;
    - Fault Tolerance of failed Map processes;
- Use QEF as workflow Engine
  - Implements Mapper interface
  - Run workflows in Hadoop seamlessly;

DEXL LAB
EXTREME DATA LAB

Integrated architecture



Experiment Set-up

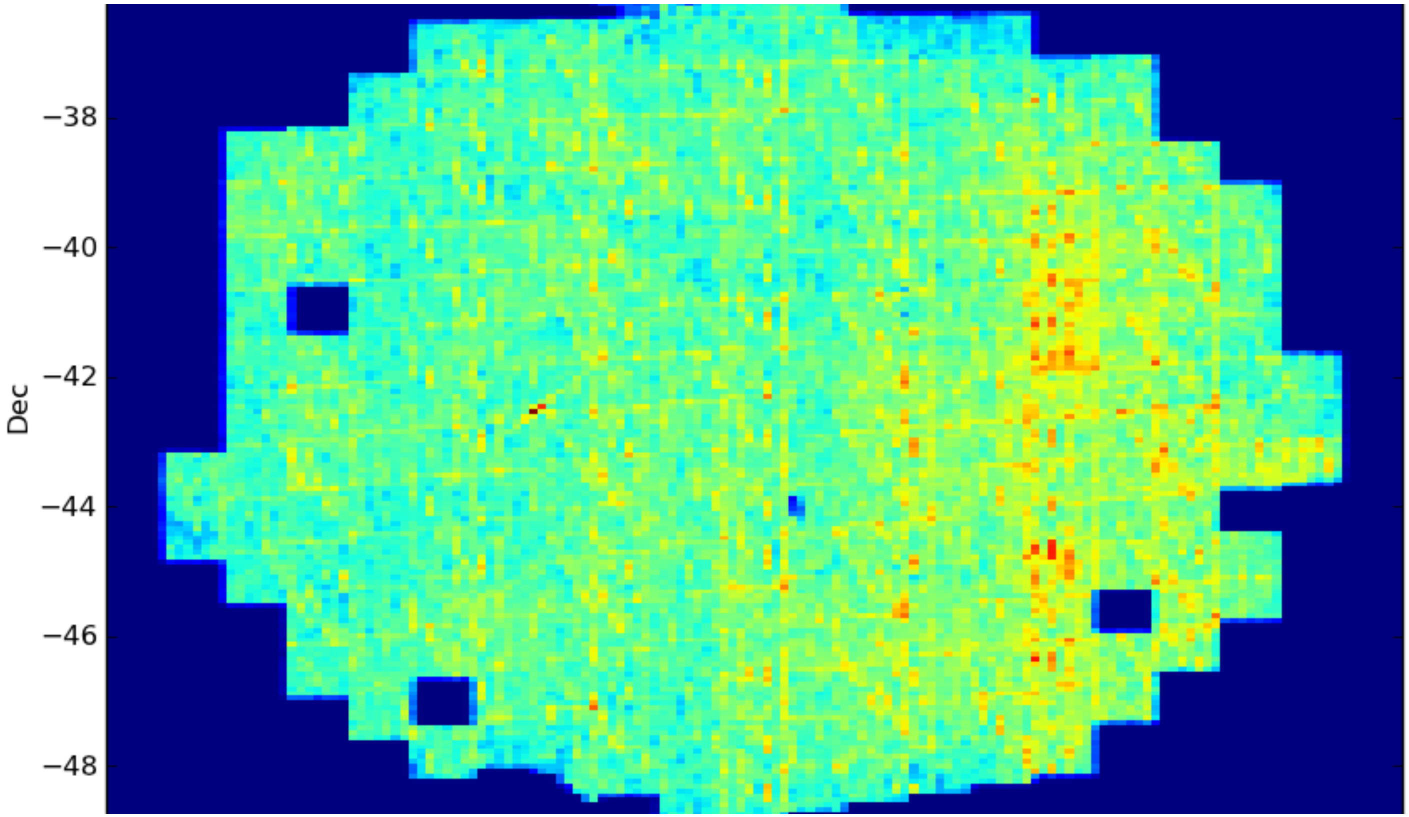
- Cluster SGI
  - Configurations: 1, 47 and 95 nodes;
  - Each node:
    - 2 proc. Intel Zeon – X5650, 6 cores, 2.67 GHz
    - 24 GB RAM
    - 500 GB HD
- Data
  - Catalog DC6B
- Hadoop
  - QEF workflow engine

Preliminary Results

- Preliminary results are encouraging:
  - Baseline Orchestration layer (234 nodes) – approx. 46 min
  - 1 node HQOOP – approx. 35 min
  - 4 nodes HQOOP – approx. 12.3 min
  - 95 nodes (94 workers) HQOOP – approx. 2.10 min
  - 95 nodes (94 workers) Hadoop+Python – approx. 2.4 min

DEXL LAB
EXTREME DATA LAB

Resulting Image



DEXL LAB
EXTREME DATA LAB

Conclusions

- Big data users (scientists) are in Big Trouble;
  - Too much data, too fast, too complex;
- Different expertise required to cooperate towards Big Data Management;
- Adapted software development methods based on workflows;
- Complete support to scientific exploration life-cycle
- Efficient workflow execution on Big Data

DEXL LAB
EXTREME DATA LAB

Collaborators

- LNCC Researchers
  - Ana Maria de C. Moura
  - Bruno R. Schulze
  - Antonio Tadeu Gomes
- PhD Students
  - Bernardo N. Gonçalves
  - Rocio Millagros
  - Douglas Ericson de Oliveira
  - Miguel Liroz-Gistau (INRIA)
  - Vinicius Pires (UFC)

DEXL LAB
EXTREME DATA LAB

Collaborators

- ON
  - Angelo Fausti
  - Luiz Nicolaci da Costa
  - Ricardo Ogando
- COPPE-UFRJ
  - Marta Mattoso
  - Jonas Dias (Phd Student)
  - Eduardo Ogasawara (CEFET-RJ)
- UFC
  - Vania Vidal
  - José Antonio F. de Macedo
- PUC-Rio
  - Marco Antonio Casanova
- INRIA-Montpellier
  - Patrick Valduriez group
- EPFL
  - Stefano Spaccapietra

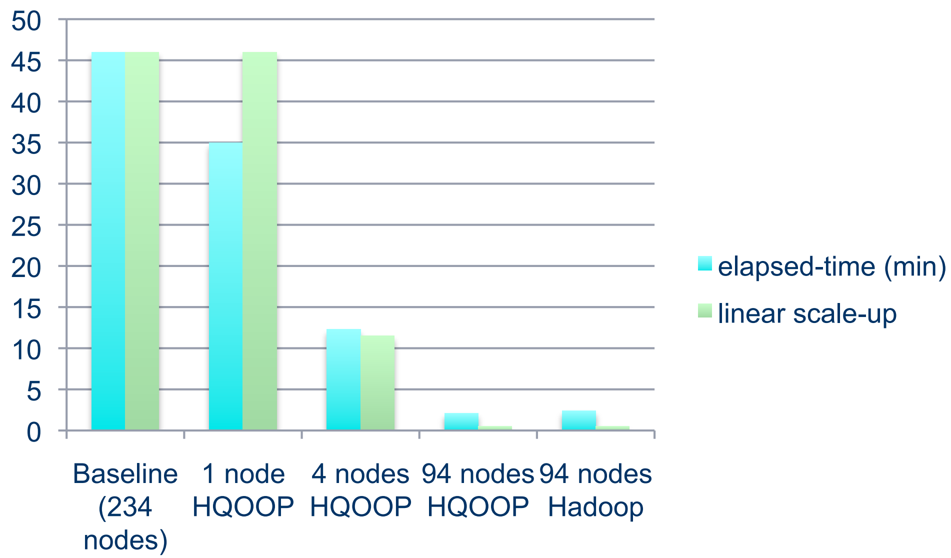
DEXL LAB
EXTREME DATA LAB

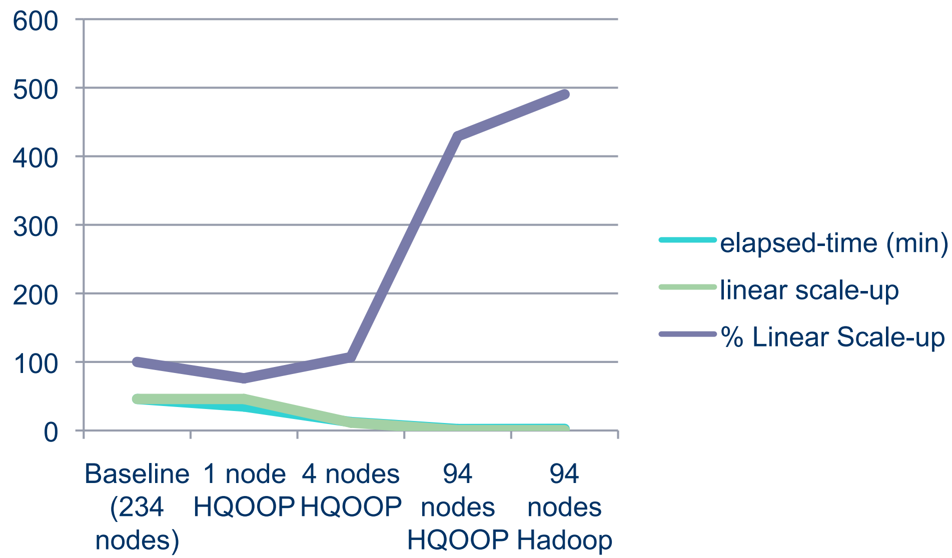
EMC Summer School on BIG DATA – NCE/UFRJ

Big Data in Astronomy

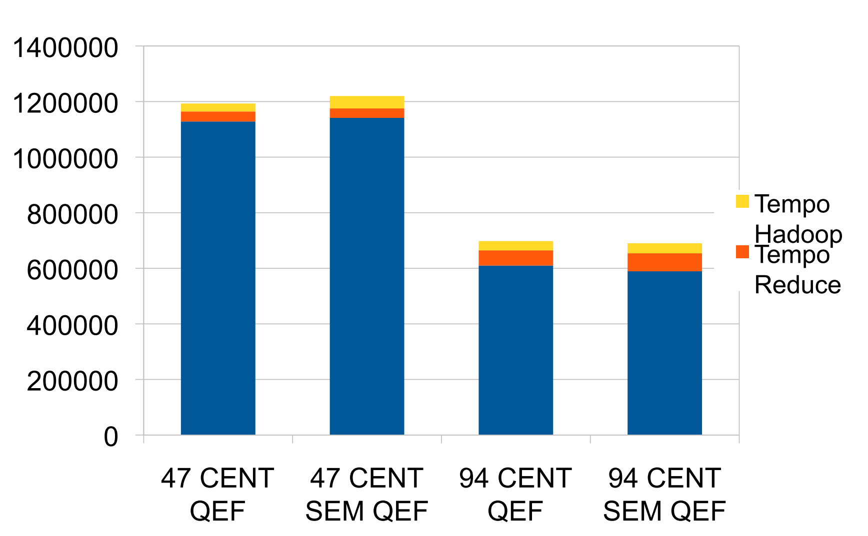
Fabio Porto (fporto@lncc.br)
LNCC – MCTI
DEXL Lab (dexl.lncc.br)

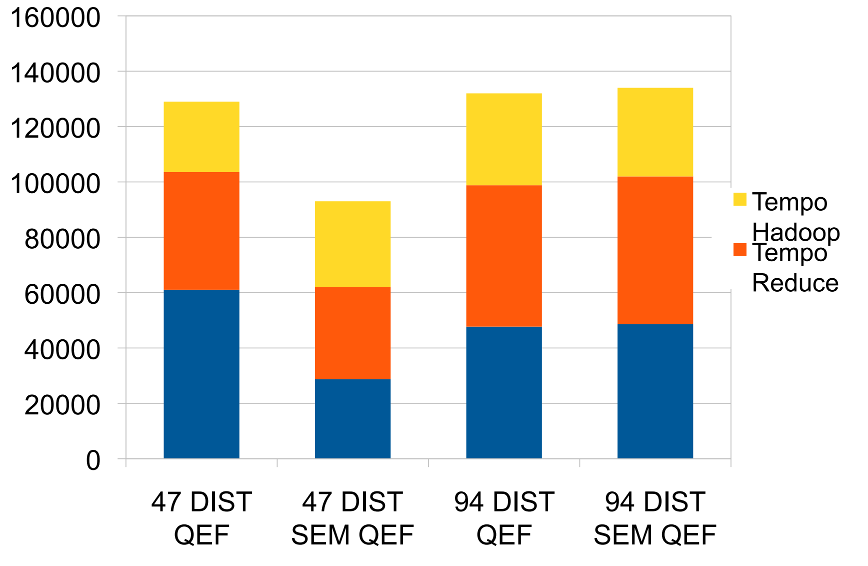
Overall performance





DEXL LAB
EXTREME DATA LAB

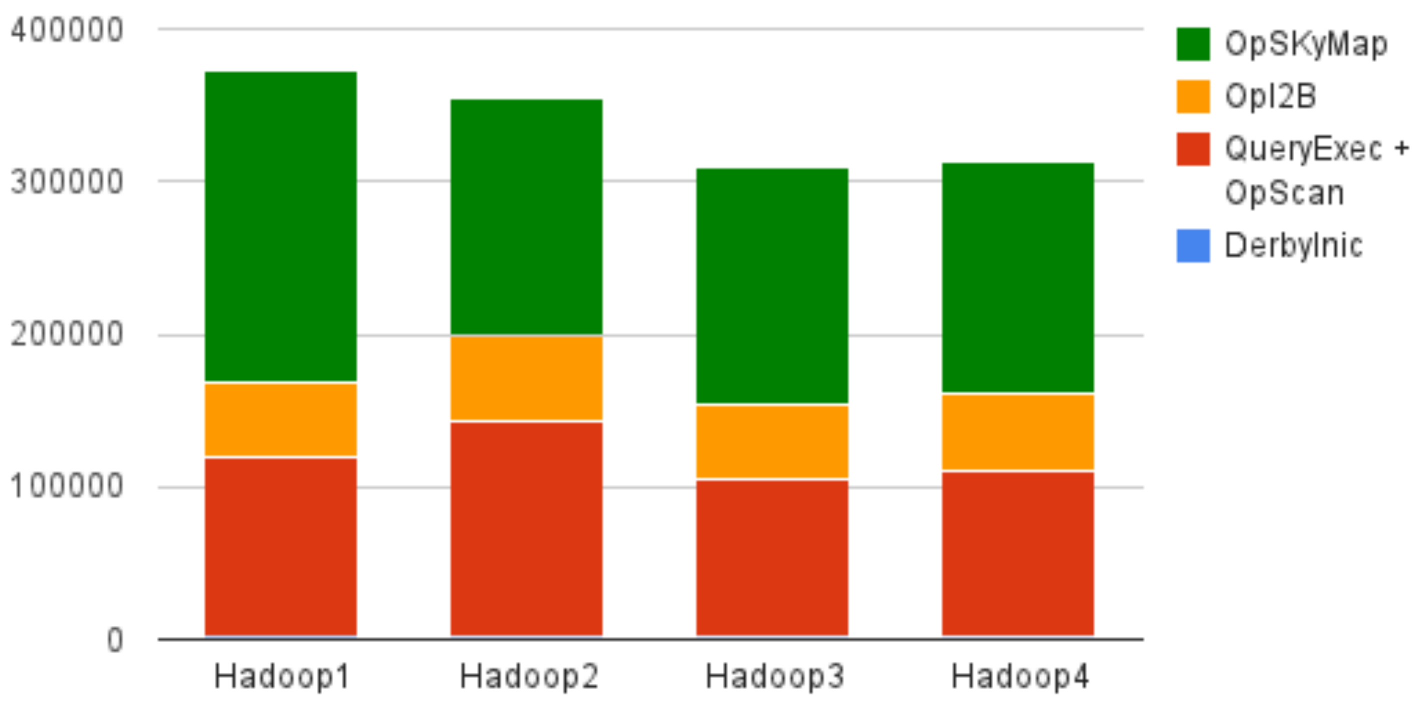




DEXL LAB
EXTREME DATA LAB

Execution with 4 nodes

Elapsed-time total: 11.27 min



QEF

Query Evaluation Framework

Adaptive and Extensible Query Engine

- Extensible to **data types**
- Extensible to **application algebra**
- Extensible to **execution model**
- Extensible to **heterogeneous data sources**

DEXL LAB
EXTREME DATA LAB

Objective

- Offer a query processing framework that can be extended to adapt to data centric application needs;
- Offer transparency in using resources to answer queries;
    - Query optimization transparently introduced
    - Standardize remote communication using web services even when dealing with large amount of unstructured data
    - Run-time performance monitoring and decision

DEXL LAB
EXTREME DATA LAB

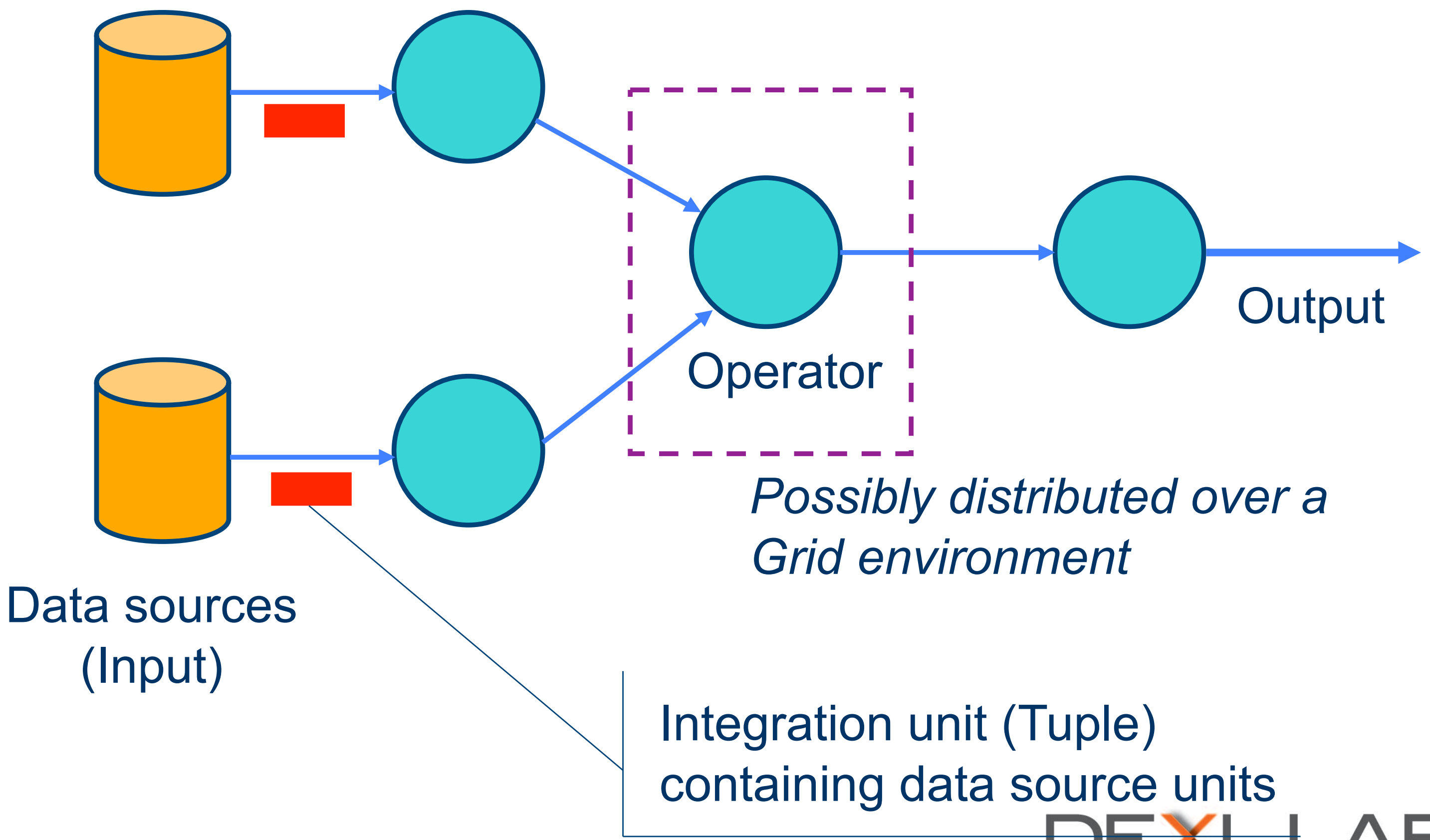
Control Operators

- Add data-flow and transformation operators
- Isolate application oriented operators from execution model data-flow concerns

- parallel grid based execution model:
  - **Split/Merge** - controls the routing of tuples to parallel nodes and the corresponding unification of multiple routes to a single flow
  - **Send/Receive** - marshalling/ unmarshalling of tuples and interface with communication mechanisms
  - **B2I/I2B** - blocks and unblocks tuples
  - **Orbit -** implements loop in a data-flow
  - **Fold/Unfold** - logical serialization of complex structues (e.g. PointList to Points)

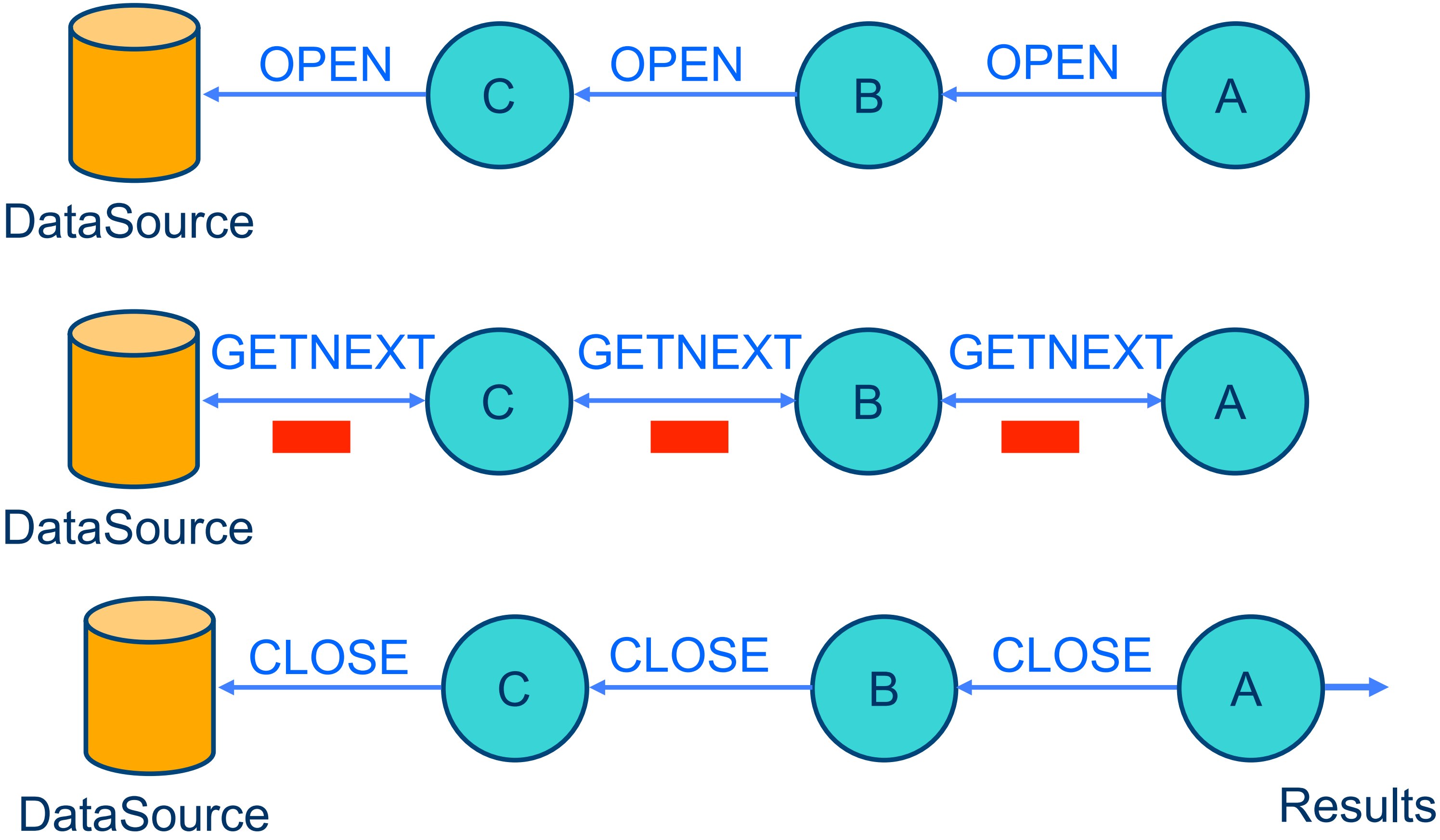
The Execution Model

Example of simple QEF Workflow



DEXL LAB
EXTREME DATA LAB

Iteration Model

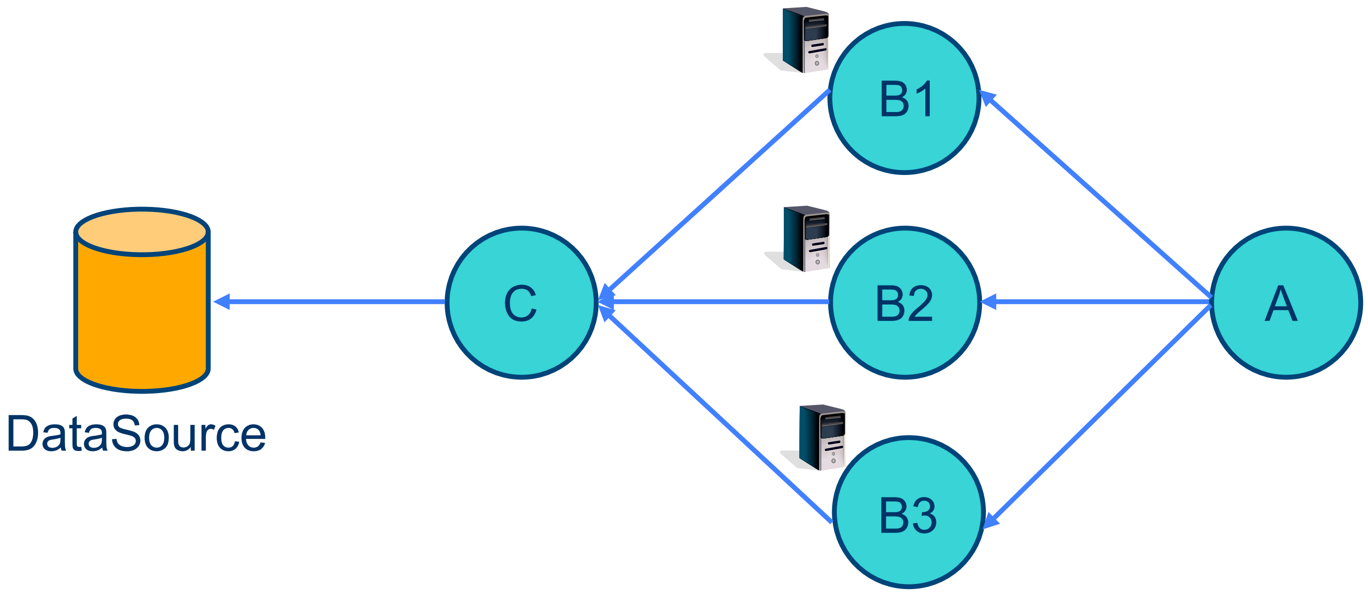


DEXL LAB
EXTREME DATA LAB

Distribution and Parallelization

Operator distribution

A Query Optimizer selects a set of operators in the QEP to execute over a Grid environment.

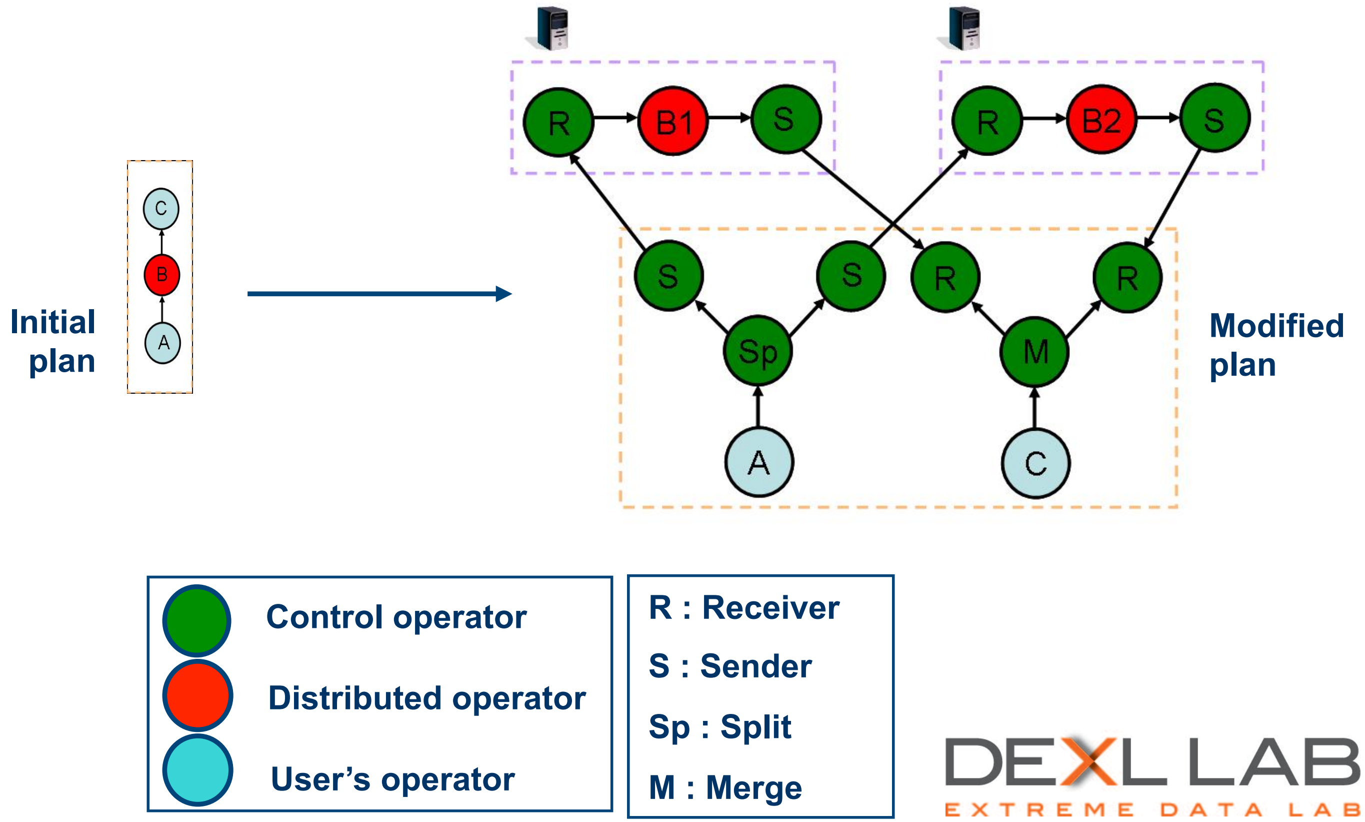


DEXL LAB
EXTREME DATA LAB

General Parallel Execution Model

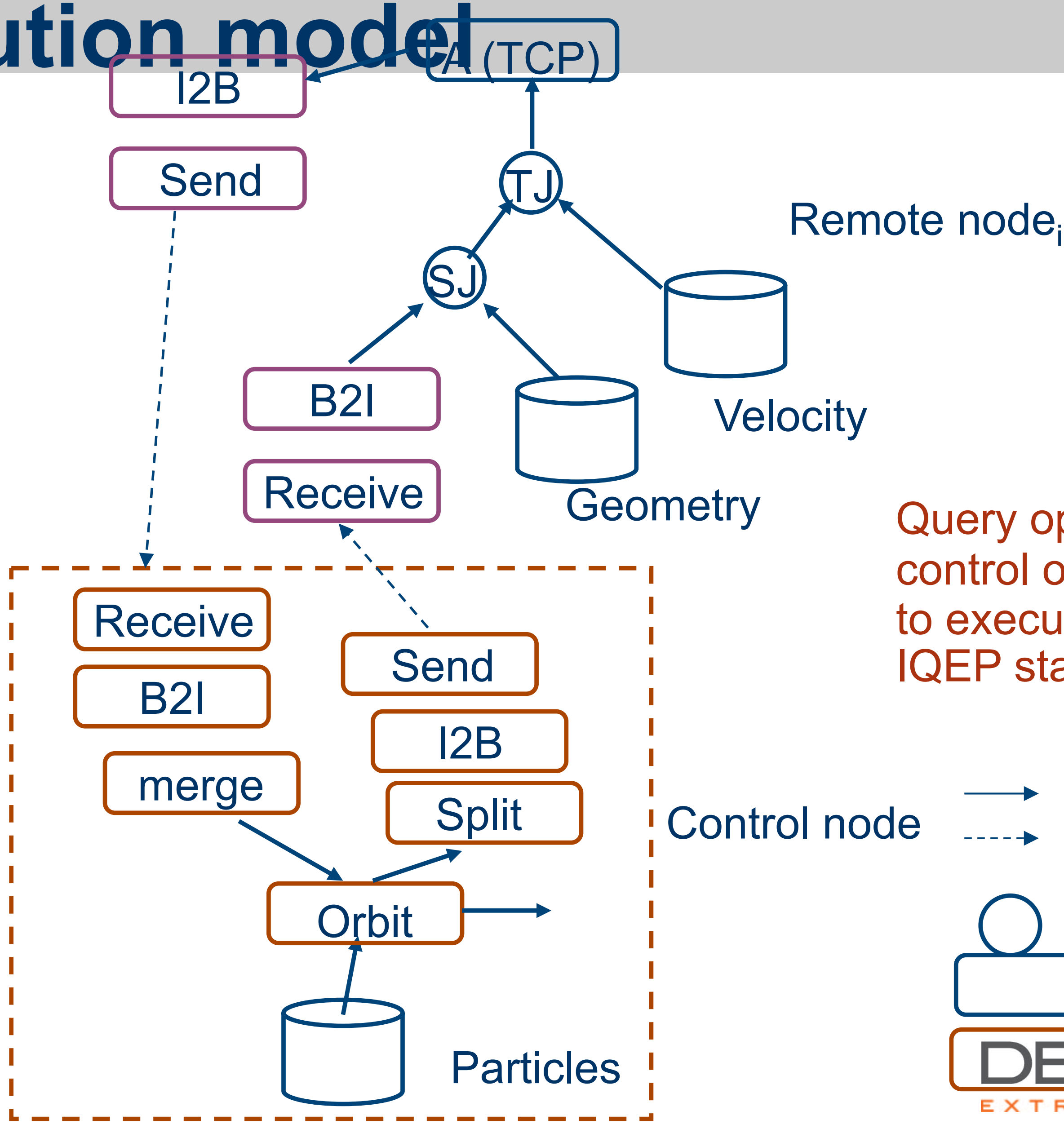
Remote QEP

In order to parallelize an execution, the initial QEP is modified and sent to remote nodes to handle the distributed execution.

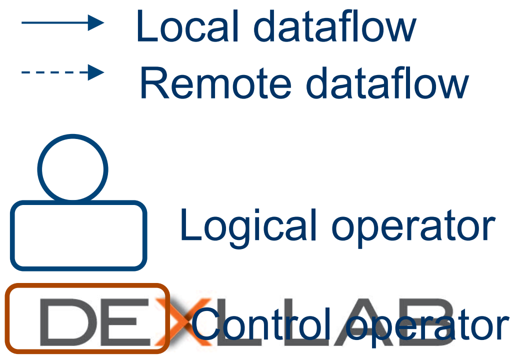


DEXL LAB
EXTREME DATA LAB

Modifying IQEP to adapt to execution model



Query optimizer adds control operators according to execution model and IQEP statistics



DEXLAB EXTREME DATA LAB

Grid node allocation algorithm (G2N)

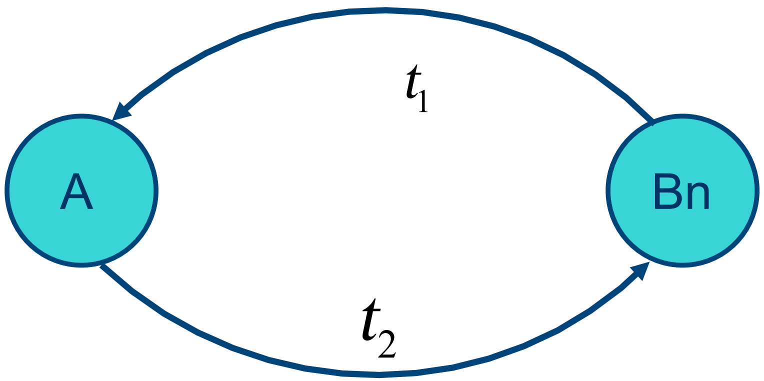
- Introduction
- Principles
- Application
- **Architecture**
- Implem.
- Conclusion

Grid Greedy Node scheduling algorithm (G2N)

- Offers maximum usage of scheduled resources during query evaluation.
- Basic idea : "*an optimal parallel allocation strategy for an independent query operator … is the one in which the computed elapsed-time of its execution is as close as possible to the maximum sequential time in each node evaluating an instance of the operator*".

$\boxed{t_1 + t_2 = t_x(Bn)}$

$t(Bn)$ operator cost on this node



Implementation

- Core development in Java 1.5.
- Globus toolkit 4.
- Derby DBMS (catalog).
- Tomcat, AJAX and Google Web Toolkit for user interface.
- Runs on Windows, Unix and Linux.
- source code, demo, user guide available at:

http://dexl.lncc.br

DEXL LAB
EXTREME DATA LAB

Summing-up

- HadoopDB extends Hadoop with expressive query language, supported by DBMSs
- Keeps Hadoop MapReduce framework
- Queries are mapped to MapReduce tasks
- For scientific applications is a question to be answered whether or not scientists will enjoy writing SQL queries
- Algebraic like languages may seem more natural (eg. Pig Latin)

Pig Latin - an high-level language alternative to SQL

- The use of high-level languages such as SQL may not please scientific community;
- Pig Latin tries to give an answer by providing a procedural language where primitives are Relational albegra operations;
- Pig Latin: A not-so-foreign language for data processing, Christopher Olson, Benjamin Reed et al., SIGMOD08;

DEXL LAB
EXTREME DATA LAB

Example

- Urls (url, category, pagerank)
- In SQL
  - Select category, avg (pagerank)

    from urls where pagerank > 0.2

    group by category

    having count(*) > 10^6
- In PIG
  - Groupurls = FILTER urls by Pagerank > 0.2;
  - Groups= Group good-urls by category;
  - Big-group=FILTER groups BY count(good_urls) > 10^6
  - Output = FOREACH big-groups GENERATE

    category, avg(good_urls_pagerank);

DEXL LAB

EXTREME DATA LAB

Pig Latin

- Program is a sequence of steps
  - Each step executes one data transformation
- Optimizations among steps can be dynamically generated, example:
  - 1) spam-urls= FILTER urls BY isSpam(url);
  - 2) Highrankurl = FILTER spam-url BY pagerank > 0.8;

1 → 2

2 → 1

DEXL LAB
EXTREME DATA LAB

Data Model

- Types:
  - Atom - a single atomic value;
  - Tuple - a sequence of fields, eg.('DB','Science',7)
  - Bag - a collection of tuples with possible duplicates;
  - Map - a collection of data items where for each data item a key is associated

['fanOf' → { 'flamengo' 'music' }
'age' → 20]

DEXL LAB
EXTREME DATA LAB

Operations

- Per tuple processing: Foreach
  - Allows the specification of iterations over bags
    - Ex:
      - Expanded-queries=FOREACH queries generate userId, expandedQuery (queryString);
      - Each tuple in a bag should be independent of all others, so parallelization is possible;
  - Flatten
    - Permits flattening of nested-tuples

(alice, { (Ipod,nano) (Ipod, shuffle) }) flatten → (alice, ipod, nano)
(alice, ipod, shuffle)

Olympic Laboratory