CSE 599c
Scientific Data Management

Magdalena Balazinska and Bill Howe

Spring 2010

Lecture 5 Part 2 – SciDB

References

- **Requirements for Science Data Bases and SciDB.** M. Stonebraker et. al. CIDR Perspectives 2009

The Problem…

- Sciences are increasingly data rich
  - Simulations on server clusters produce lots of data
  - Improved instrumentation collects more data
  - Automated experiments produce more data
- Scientists need effective tools to manage data
  - Storage
  - Analysis
  - Organization
  - Sharing
- Existing DBMSs do not meet scientists needs

What are the Challenges

- "Big science" very unhappy with RDBMS
- Reason 1: Main data type is not a relation!
  - Many sciences need arrays
  - Others want graphs, sequences, or meshes
- Reason 2: Many required features are absent
  - Provenance
  - Uncertainty
  - Version control
- Reason 3: Operations are wrong
  - Regrid – not join

What is the State of Affairs?

- Roll-your-own on the bare metal
  - Larger Hadron Collider (LHC) [http://lhc.web.cern.ch/lhc/]
  - NASA Mission to Planet Earth: [http://www.hq.nasa.gov/office/nsp/mtpe.htm]

- Or put up with a horrible kludge on RDBMS
  - With mountains of application logic
  - And copying the world to application space

So What is SciDB?

- A new type of DBMS for data intensive analytics
- Addresses the above limitations
- Community supported, open-source project
- Fundamental idea: build an array-based DBMS!

How Can We Build an Array DBMS?

- By storing arrays as tables inside of a DBMS
  - Advocated by Greenplum [http://www.greenplum.com/]
  - And MonetDB [http://monetdb.cwi.nl/]
  - But can lead to terrible performance

- Using BLOBs (binary object types) in a DBMS to represent arrays
  - Implemented in RasdaMan [http://www.rasdaman.com/]

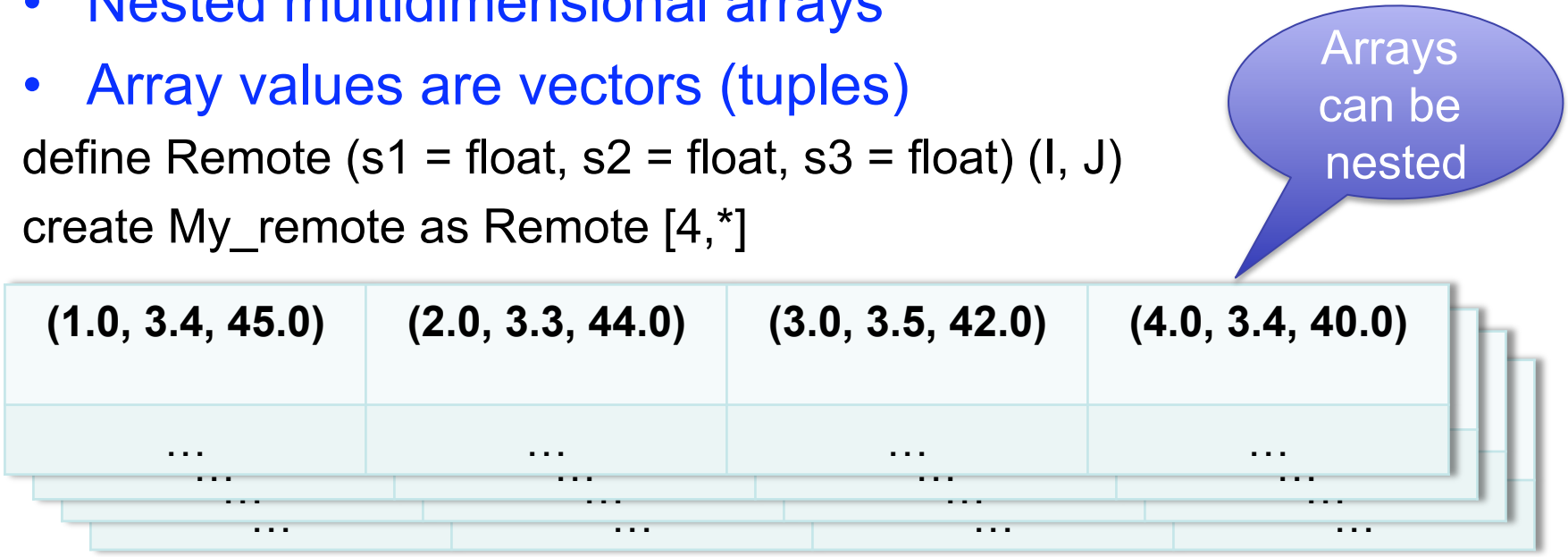
- A from-the-ground-up native array system: SciDB

SciDB Data Model Basics

- Nested multidimensional arrays
- Array values are vectors (tuples)

define Remote (s1 = float, s2 = float, s3 = float) (I, J)

create My_remote as Remote [4,*]



- Updates to arrays logged in a history dimension

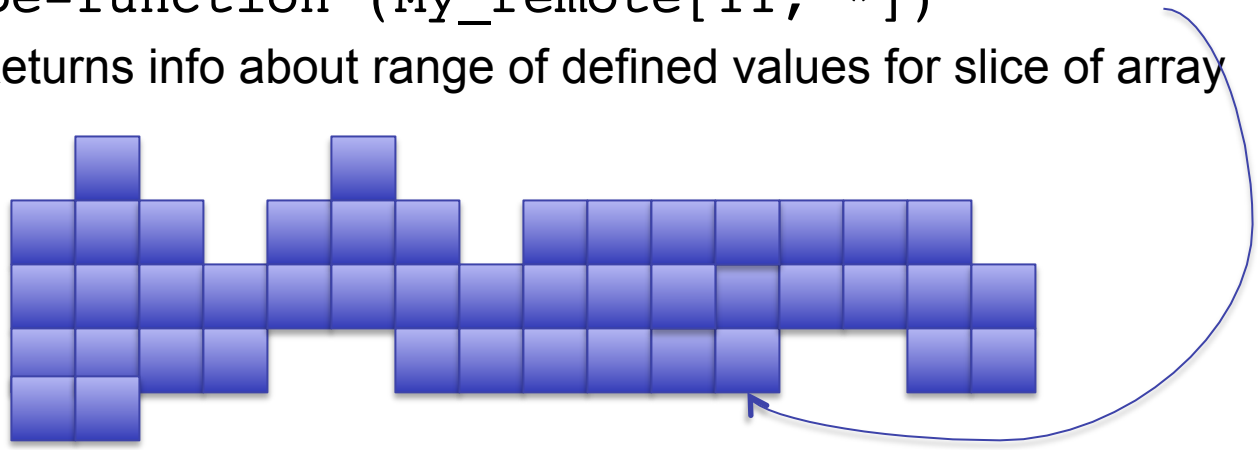
define updatable

Remote_2 (s1=float, s2=float, s3=float) (I, J, history)

SciDB Data Model Advanced Features

- Arrays can be augmented with co-ordinate systems
  - `My_remote[x,y]`, where (x,y) is the cell in the array
  - `My_remote{latitude,longitude}`
  - Coordinates need not be integer-valued nor contiguous

- Arrays can be augmented with "shape" functions
  - To define irregular arrays
  - `shape-function (My_remote[11, *])`
    - Returns info about range of defined values for slice of array



Operators

- Structural Operators
  - Subsample: Subsample (F, even(X))
  - Reshape: not in current system
  - Structured join or SJoin: example in Figure 1

- Content-Dependent Operators
  - Filter
  - Aggregate
  - Cjoin

- But, extensibility is key! Support for user-defined functions

SciDB Query Language

- Declarative queries
- SQL-like and array operators
  - Filter, trim, project, add-dim, slice, join, regrid, etc.
- With a binding to
  - MatLab
  - C++
  - Python
  - There may be more….

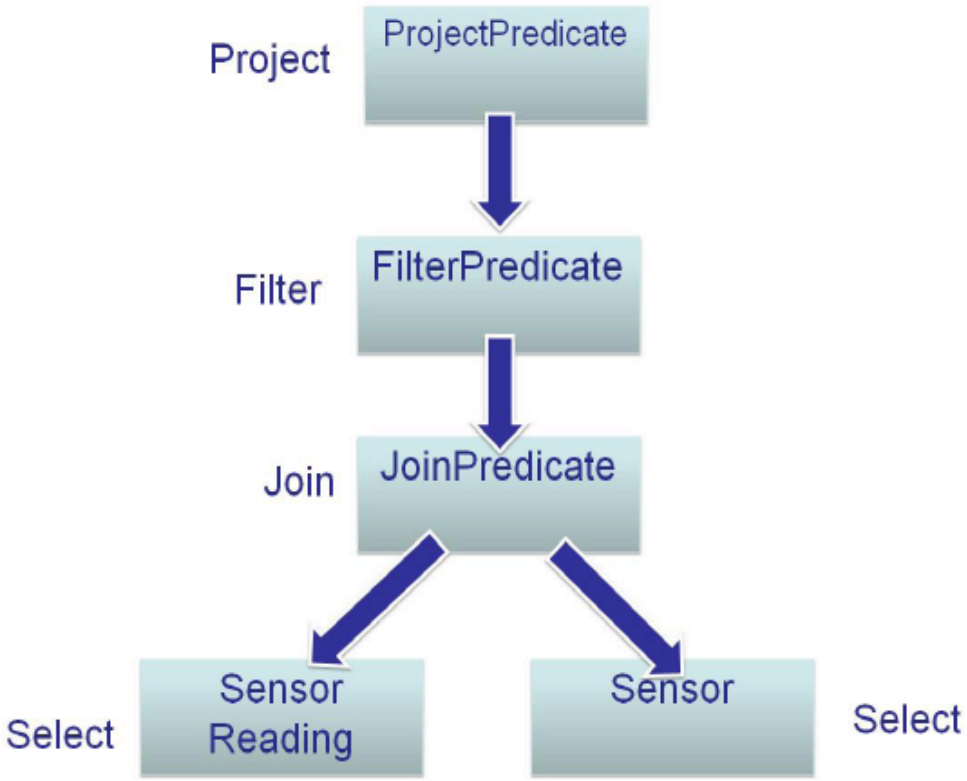
Language Binding Example

Filter pixels with flux values between x1 and x2 and display resulting observations

```
string filterPredicate = "xAstrom BETWEEN @x1 AND @x2

   AND psfFlux > @flux AND flagForDetection = @flag";

DBArray dba("@arrayName");

DBArray dba2 = dba.filter(filterPredicate);

typedef Observation T;

for (DBArrayIterator<T> it = dba2.begin<T>(); it != dba2.end<T>(); ++it){

     Observation observ= (*it);

     cout << observ.ToString();

}
```

Algebra Tree

- Code is translated into algebra tree



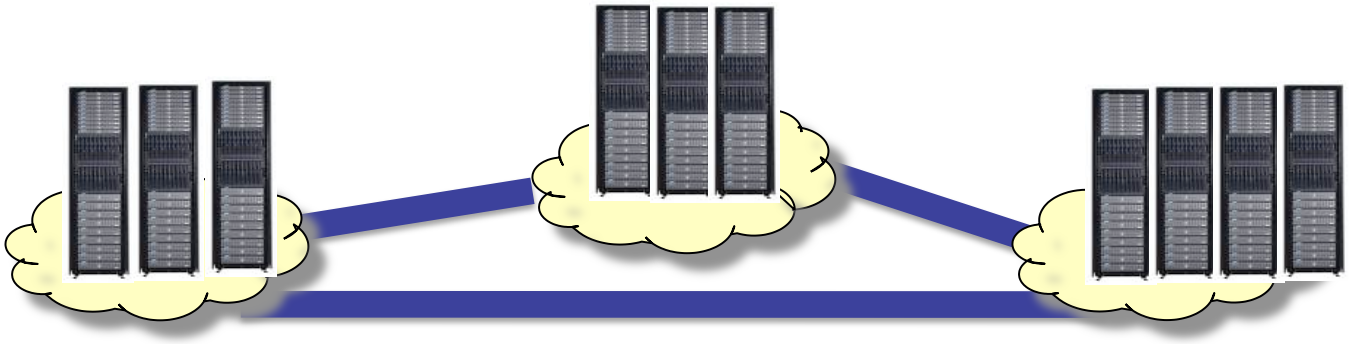
Shipping Query to SciDB

- Algebra tree is serialized to xml format

```
<store ref="q4_res" overwrite="true">
  <filter pred=" xAstrom BETWEEN @x1 AND @x2
   AND psfFlux > @flux AND flagForDetection = @flag ">
   <array ref="@arrayName"/>
  </filter>
</store>
```

Environment and Storage

- Store hundreds of Petabytes of data
  - Tables with trillions of rows
- Extendable cloud
  - N clusters of machines spread over WAN
  - Each cluster is a grid of machines on a LAN



- Built-in high availability, failover, disaster recovery

Storage manager

- Splits large arrays into series of multidimensional chunks
  - Fixed stride
    - Easy to index
    - But blocks may be highly variable in size
  - Or variable stride
    - Need an R-tree
    - But packing can be more uniform
- Each attribute stored in a separate physical array
- Vertica-style compression
- Replication by multiple copies with different partitioning
- Supports overlap

“In Situ” Data

- Can we process data without a load phase?
  - Loading requires that the user define a schema
  - Loading can fail due to errors
  - Loading is time consuming

- SciDB will try to process data without loading it first
  - Of course, only subset of features will be supported on such data
  - And performance will likely be worse

Additional Features

- No overwrite data model: Instead use updatable arrays
- Named versions
- Provenance
  - Need to log all operations
  - Want to trace “back” and “forward” in time
- Uncertainty