Big Data and Big Analytics

• Introducing SciDB
  – Open source, massively parallel DBMS and analytic platform
  – Array data model (rather than SQL, Unstructured, XML, or triple-store)
  – Extensible micro-kernel architecture
  – Reliable storage platform

• Analytic Example
  – Problem Description: Big Data, Big Analytics
  – Compare and Contrast SciDB with SQL and Map/Reduce

• Data, Queries and Scalability Graphs
  – What does SciDB look like from a user’s perspective
  – Cloud deployment (EC2) up to 64 nodes
Introducing SciDB

• Open Source
  – Released under GPL-3
  – Sponsored by Paradigm4 Inc.
  – Contributors from the US, Russia, India, and Europe

• Array Data Model
Why Arrays?

• Formal Mathematical Foundations
  – Closed algebra of Array ➔ Array operations
  – Formal proofs of correctness for re-writes
  – Mathematical Model has broad applications

• Computer Science Pedigree
  – APL – *A Programming Language*
  – Lots of work on algorithms to compute Array ➔ Array ops
  – Lots of work on efficient storage and data manipulation

• Ongoing Area of Research and Development
  – BLAS, LAPack, ScaLAPack – open-source tools
  – NAG – Numerical Algorithmics Group – proprietary platforms
  – R, SaS, etc - each have an efficient Linear Algebra Library inside
Introducing SciDB data definition

CREATE ARRAY Simple_Array <
    v1 : double,
    v2 : int64,
    v3 : string >
[ I = 0:* , 5, 0, J = 0:9, 5, 0 ];

Attributes    Dimensions    Dimension size    Chunk size    Chunk overlap
v1, v2, v3    I, J          * indicates unbounded
2 Programming Interfaces

Array Functional Language AFL

aggregate(
    filter ( Simple_Array, 
        v3 = 'Odd'),

    I,
    avg ( Simple_Array.v1 )
);

Array Query Language AQL

SELECT avg ( S.v1 )
    FROM Simple_Array S
WHERE S.v3 = 'Odd'
GROUP BY S.I;
SciDB: Analytic Features

• Versioned (no-overwrite) Storage System
  – UPDATE operations are append only
  – The state of the system can be reconstructed at a specified time

• Provenance
  – Log of all queries to reconstruct how results were derived

• Uncertainty and Statistical Reasoning
  – Types with error bars
  – Functions to perform statistical tests and analytics

Goal is to meet the requirements for scientific data management identified in Stonebraker, M, et al
“Requirements for Science Data Bases and SciDB”, CIDR 2009
## SciDB vs. RDBMSs
on storage efficiency & complex computations

### Relational Database

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>32.5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>90.9</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>42.1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>96.7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>46.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>35.4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>35.7</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>41.3</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>81.7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>35.9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>35.3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>89.9</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>53.6</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>86.3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>45.9</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>27.6</td>
</tr>
</tbody>
</table>

### SciDB

<table>
<thead>
<tr>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.5</td>
</tr>
<tr>
<td>90.9</td>
</tr>
<tr>
<td>42.1</td>
</tr>
<tr>
<td>96.7</td>
</tr>
<tr>
<td>35.4</td>
</tr>
<tr>
<td>35.9</td>
</tr>
<tr>
<td>41.3</td>
</tr>
<tr>
<td>81.7</td>
</tr>
<tr>
<td>35.7</td>
</tr>
<tr>
<td>35.3</td>
</tr>
<tr>
<td>89.9</td>
</tr>
<tr>
<td>53.6</td>
</tr>
<tr>
<td>86.3</td>
</tr>
<tr>
<td>45.9</td>
</tr>
<tr>
<td>27.6</td>
</tr>
</tbody>
</table>

### Comparison

- **16 cells**

- **48 cells**

- Speeds up data access in a distributed database
- Dramatic storage efficiencies as # of dimensions & attributes grows
- Facilitates drill-down & clustering by like groups
- Math functions like linear algebra run directly on native storage format
SQL vs. SciDB: Data Definition

CREATE TABLE USNOB (  
    Locn  point               not null,  
    -- RA double, DECL double  
    ...  
    B_mag  double precision    not null,  
    ...  
    R_mag  double precision    not null  
    ...  
);  

CREATE ARRAY USNOB <  
    B_Mag  :  double,  
    ...  
    R_Mag  :  double  
>  
[ RA(double)=*,72000,720, DECL(double)=*,36000,360 ];
SQL vs. SciDB: Data Manipulation

SELECT U1.Locn, COUNT(*)
FROM USNOB AS U1, USNOB AS U2
WHERE box(U1.Locn, U1.Locn) &&
  box(point(U1.Locn[0] - 0.001,
           U1.Locn[1] + 0.001),
      point(U1.Locn[0] + 0.001,
           U1.Locn[1] - 0.001))
GROUP BY U1.Locn;

window (USNOB,
        0.001, 0.001,
        count(*))
);

SciDB

SQL
SciDB Storage: Arrays and Chunks

[ l=0:9,5,2, J=0:6,2,1 ]
SciDB Storage: Arrays and Chunks

\[
[ I=0:9,5, J=0:6,2, 1 ]
\]
Shared Nothing MPP Architecture

SciDB Coordinator Node

SciDB Node

SciDB Node

SciDB Node

SciDB Node

SciDB Client (iquery, Python)

SciDB Engine

Local Store

SciDB Engine

Local Store

SciDB Engine

Local Store

SciDB Engine

Local Store

SciDB Inter-Node Communication

PostgreSQL Persistent System Catalog Service

PostgreSQL Connections

SciDB Worker Nodes

Copyright Paradigm4 Inc. 2011 all rights reserved.
SciDB is not Map/Reduce

- Build on the Best of Map/Reduce’s many good ideas
  - Job scheduling over large numbers of nodes
  - Use k-copy replication of chunks to implement reliable storage

- SciDB’s Specialized Storage Model Conveys Advantages
  - Block distribution and inter-node operations
  - Local parallelism for individual block to block operations
SciDB is not Map/Reduce

• Usage Model supports ad hoc queries
  – Use case lessons: Analysts like ‘R’ and dislike programmers
  – Interactive, conversational, explorative model

• Linear Algebra: hard to implement as a Map and a Reduce
  – Linear Algebraic operations are the foundations of statistical analysis
  – Linear Algebra operators are ‘not embarrassingly parallelizable’
  – multiply ( transpose ( A ), A ) is an O ( N^3 ) algorithm
  – inverse ( A ) is best implemented as an iterative algorithm
  – Cumulative sum, multi-dimensional WINDOW

• Deeply integrated data management & analytics has many advantages
  – Data is updatable
So, how do I query the database?

It's not a database. It's a key-value store!

Ok, it's not a database. How do I query it?

You write a distributed map reduce function in Erlang!

Did you just tell me to go fuck myself?

I believe I did, Bob.
Brief Performance Study

• Three Queries in our AFL Query Language

  multiply ( transpose ( Simple_Array ), Simple_Array );

  regrid( Simple_Array, 10, 10, avg (v2) );

  cumsum (  
    filter ( Simple_Array, v1 = 'Odd' ),
    I, v1 );

• Amazon EC2 Cluster of 8 Through 64 Nodes
Mathematical Operations

![Graph showing speed-up vs cluster size. The graph includes lines for pearson, covariance, and regrid/cumsum. The x-axis represents cluster size (k), and the y-axis represents speed-up (T8 / T_k). The graph shows an increasing trend with increasing cluster size.]
The “It’s Beer-o-Clock” Slide

• SciDB
  – Open-source, from the ground up array DBMS and analytic platform
  – [http://www.scidb.org](http://www.scidb.org)

• Paradigm4
  – Contributed all the open-source development to date
  – Will distribute and support enterprise-quality software along with P4 add-ons

• Why SciDB?
  – Query-centric usage model suits scientific / analytic users
  – Arrays support linear algebra and statistical processing
  – Shared-nothing architecture for scalability
  – Building on the shoulders of giants Map/Reduce, RDBMSs, ‘R’
  – Scalable data manipulation and linear algebra