

# Servicing Seismic and Oil Reservoir Simulation Data through Grid Data Services

Sivaramakrishnan Narayanan, Tahsin Kurc,  
Umit Catalyurek and Joel Saltz

**Multiscale Computing Lab**

**Biomedical Informatics Department**

**The Ohio State University**

<http://www.bmi.osu.edu>

<http://www.multiscalecomputing.org>

## **Multiscale Computing Lab**

<http://www.multiscalecomputing.org>

**Joel Saltz**

**Gagan Agrawal**

**Umit Catalyurek**

**Shannon Hastings**

**Vijay S Kumar**

**Tahsin Kurc**

**Steve Langella**

**Scott Oster**

**Tony Pan**

**Benjamin Rutt**

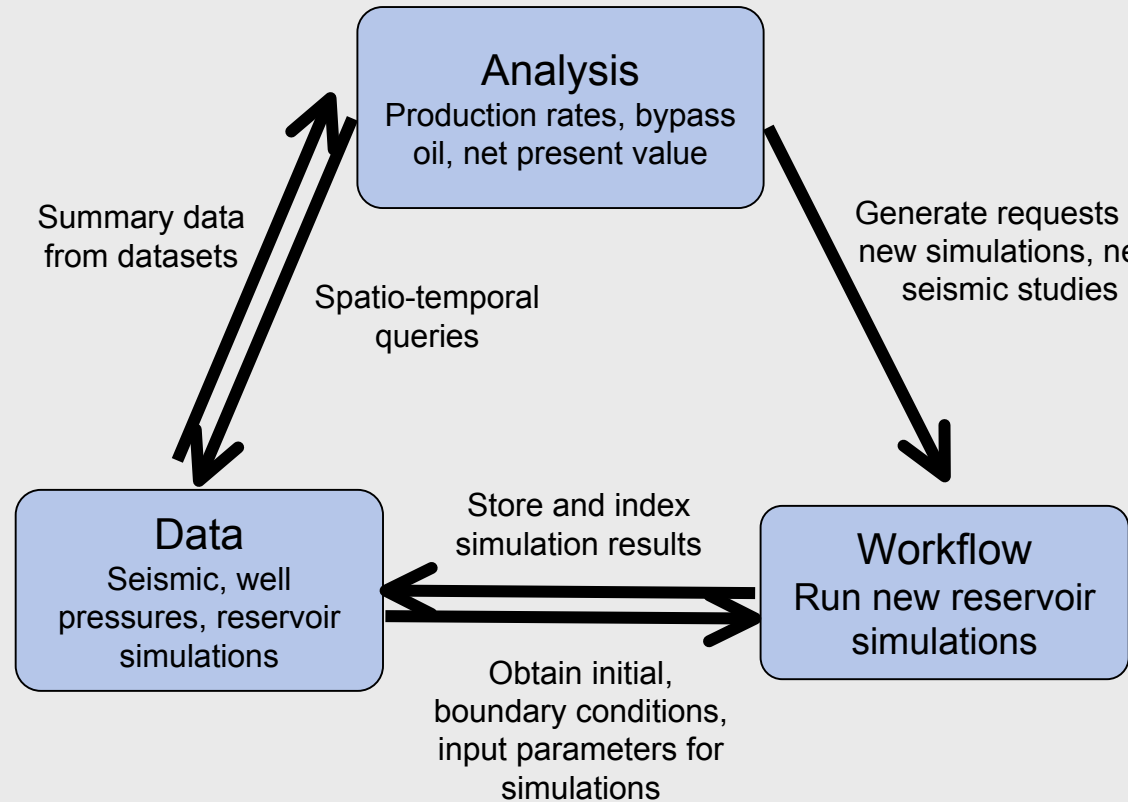
**Narayanan Sivaramakrishnan,**

**Li Weng**

**Michael Zhang**

# Implementing effective oil and gas production

- Simulate multiple realizations of multiple geostatistical models and production strategies
- Evaluate **geologic uncertainty** and **production strategies** simultaneously
- Enable on-demand exploration and comparison of multiple scenarios
  - Integration of a robust, Grid-based computational and data handling infrastructure
  - **Distributed databases of reservoir and geophysical data**
  - **Storage and computing resources at multiple institutions**



# Characteristics and Issues

- Spatio-temporal datasets
  - Simulations carried out/data captured on 3D meshes over many time steps
  - Multiple data attributes per data point (gas pressure, oil saturation, seismic traces, etc).
- Very large datasets
  - Tens of gigabytes to 100+ TB data
- Lots of simulation runs
  - Up to thousands of runs for a study are possible
- Data can be stored in distributed collection of files
- Distributed datasets
  - Data may be captured at multiple locations by multiple groups
  - Simulations are carried out at multiple sites
- Common operations: subsetting, filtering, interpolations, projections, comparisons, frequency counts

# Data Management, Access and Integration

- Tracking of metadata associated with data
  - Metadata defining simulation parameters, mesh description, files associated with simulations, etc.
  - Metadata defining seismic measurements (location, year, files storing data, etc.)
- Support for data subsetting and filtering on file-based, distributed datasets
- Support for on-demand data product generation
  - Track metadata associated with data analysis workflows
- Grid data services and distributed querying
  - Make data and data products available through Grid service interfaces

# Data Virtualization

Applications developers generally prefer storing data in files

Support high level queries on multi-dimensional distributed datasets

Many possible data abstractions, query interfaces

- Grid virtualized object-relational database or XML database

- Grid virtualized objects with user defined methods invoked to access and process data

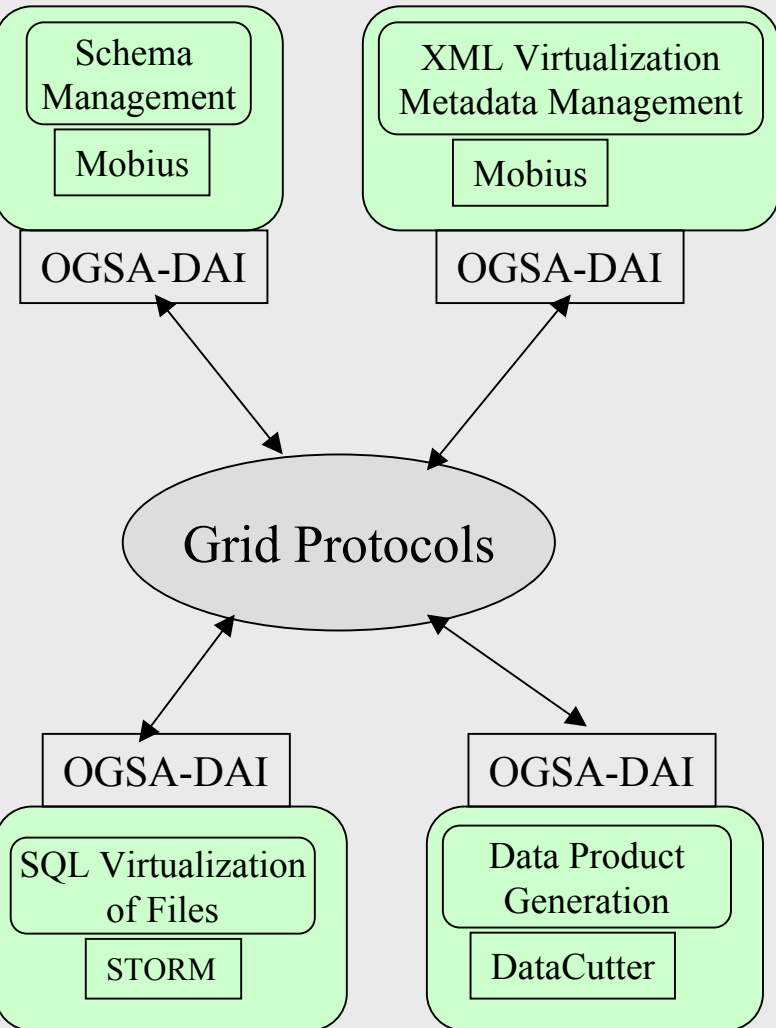
Our Approach

- Support a basic SQL Select query with a virtual relational table view or a virtual XML database view
- A lightweight layer on top of datasets
  - Runtime middleware carries out query execution, query planning

# Middleware Support

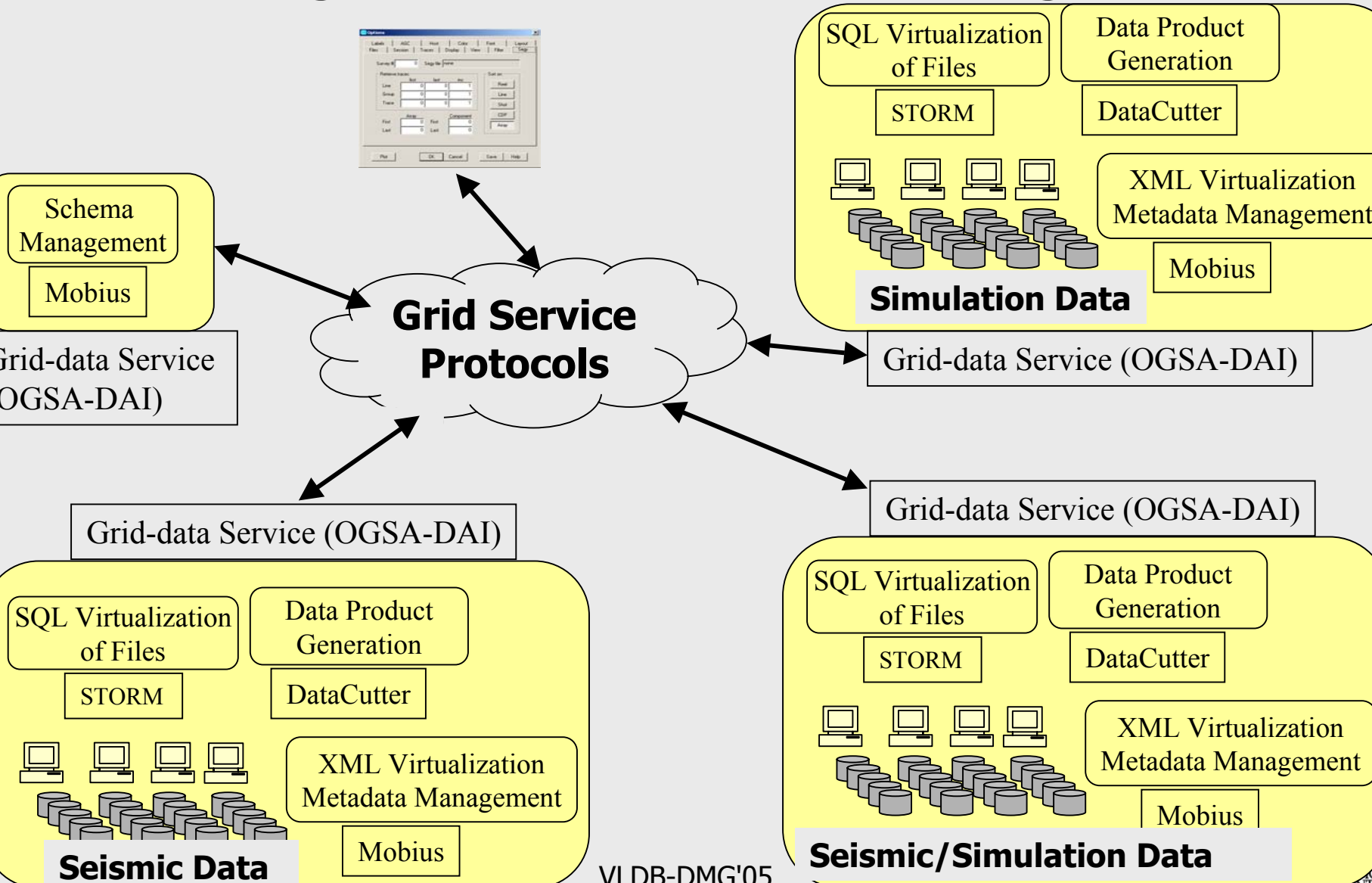
- **Data Virtualization: STORM**
  - Large data querying capabilities, layered on DataCutter
  - Distributed data virtualization
  - Indexing, Subsetting, Data Cluster/Deccluster, Parallel Data Transfer
- **Data Analysis/Processing Workflows: DataCutter**
  - Component Framework for Combined Task/Data Parallelism
  - Filtering/Program coupling Service: Distributed C++ component framework
  - On demand data product generation
- **Distributed Metadata and Data Management: Mobius**
  - Create, manage, version data definitions
  - Management of metadata and data instances
  - Data integration
- **Grid Data Services (OGSA-DAI)**
  - Defines services and interfaces that can be used by clients to specify operations on data resources and data

# Data Management, Access, Integration



- Grid-level data services via OGSA-DAI
- Management of data definitions and metadata, XML virtualization via Mobius
- Object-relational virtualization and subsetting of file based datasets via STORM
- On-demand data product generation via DataCutter
- STORM, Mobius, DataCutter support data operations on heterogeneous collections of storage and compute clusters

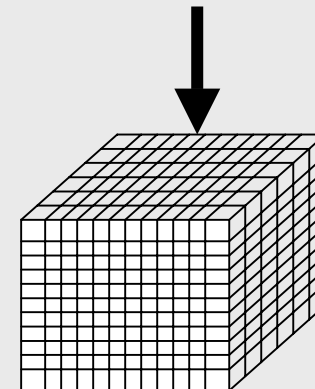
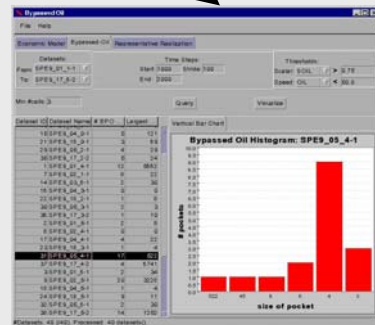
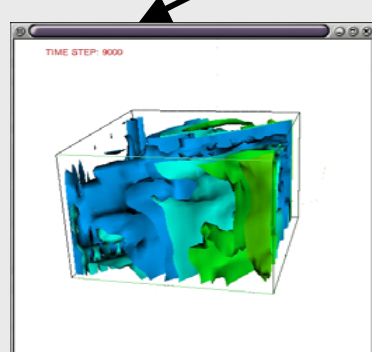
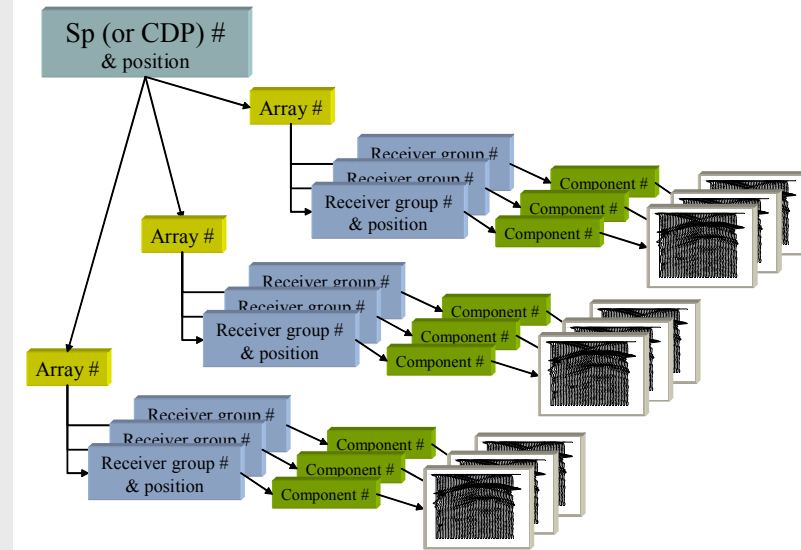
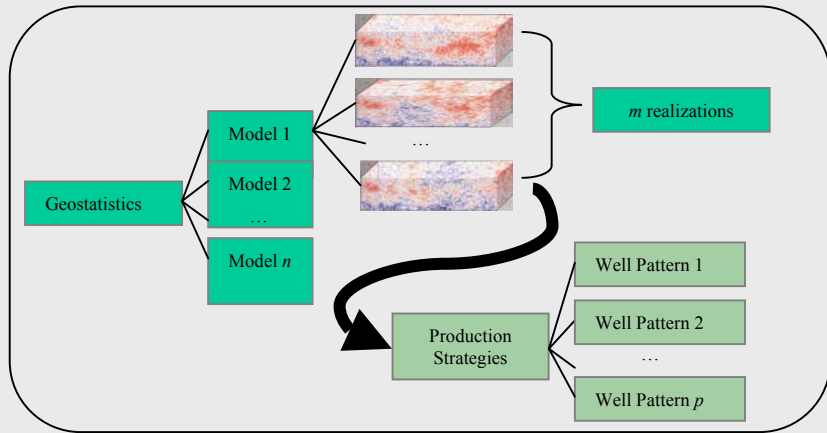
# Data Management, Access, and Integration



# Data Querying and Processing

## Seismic Data

### Reservoir Simulations



# STORM

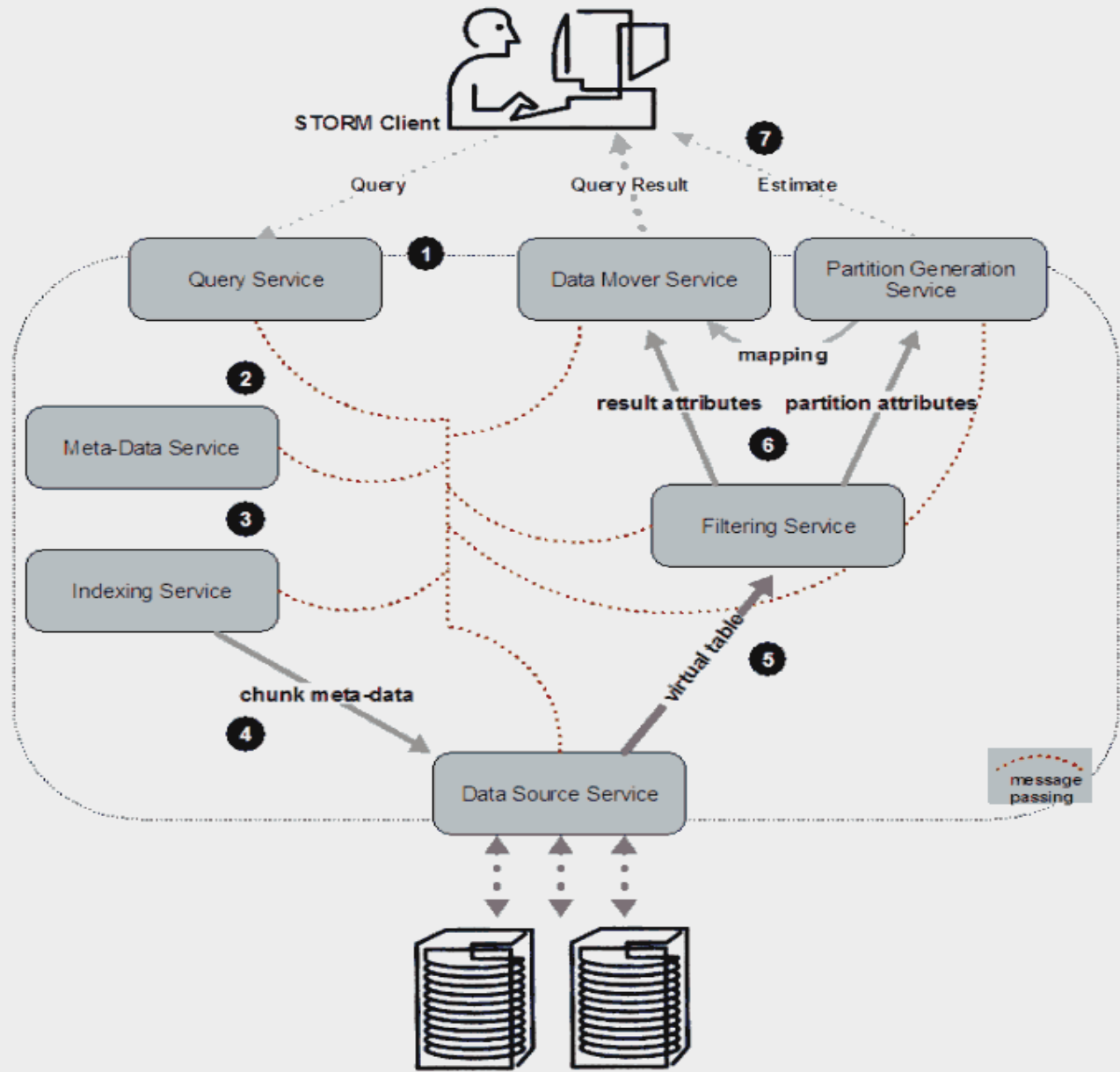
*Support efficient selection of the data of interest from distributed scientific datasets and transfer of data from storage clusters to compute clusters*

- Data Subsetting Model
  - Virtual Tables
  - Select Queries
  - Distributed Arrays

```
SELECT <DataElements>  
FROM Dataset-1, Dataset-2, ..., Dataset-n  
WHERE <Expression> AND <Filter(<DataElement>)>  
GROUP-BY-PROCESSOR ComputeAttribute(<DataElement>)
```

## STORM Services

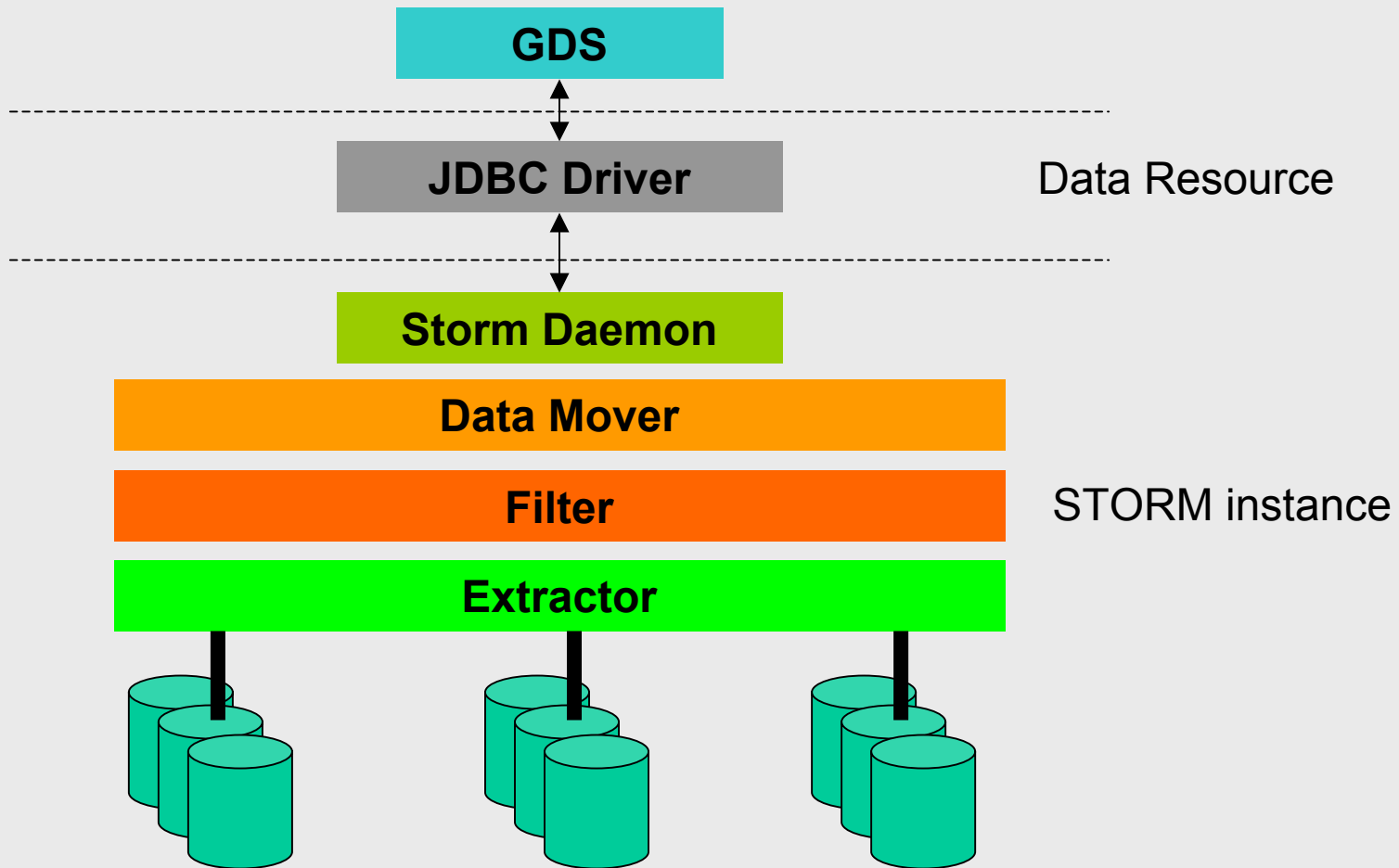
- Query
- Meta-data
- Indexing
- Data Source
- Filtering
- Partition Generation
- Data Mover



# Grid Data Resource

- Grid has emerged as an integrated infrastructure for distributed computation
- OGSA-DAI initiative is to deliver high level data management functionality for the Grid.
  - Defines services and interfaces that can be used by clients to specify operations on data resources and data
- OGSA-DAI services can be configured to expose a specific database management system.
- To be a GDS, a service must accept perform documents and return results
  - Interpretation of perform documents is open to interpretation
  - Traditionally wrap SQL queries

# STORM Data Resource



# Experimental Setup

mob	8 nodes	Dual 1.4 GHz AMD Optron	8 GB memory	1.5 TB local disk
Xio	16	2 Xeon 2.4 GHz	4 GB memory	7.3 TB FASTT600 disk array

<b>Dataset</b>	<b>Attributes</b>	<b>Record Size</b>	<b>Records (millions)</b>	<b>Dataset (GB)</b>	<b>Cluster, Num nodes</b>
Oil Reservoir	21	84 bytes	3,840	315	Mob,03
Seismic	16	4240 bytes	247	1,056	Xio,16
TXm	6	24 bytes	X	24 * X / 1M	Mob,01

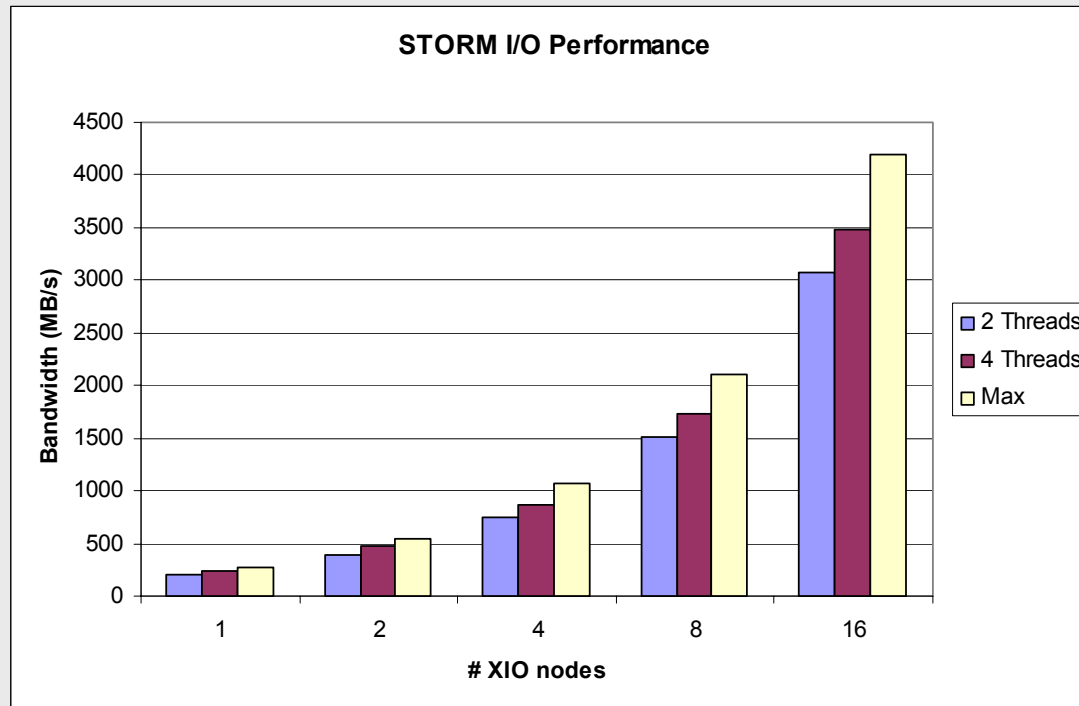
- All nodes running linux
- Gigabit switch

# STORM Results

## Seismic Datasets

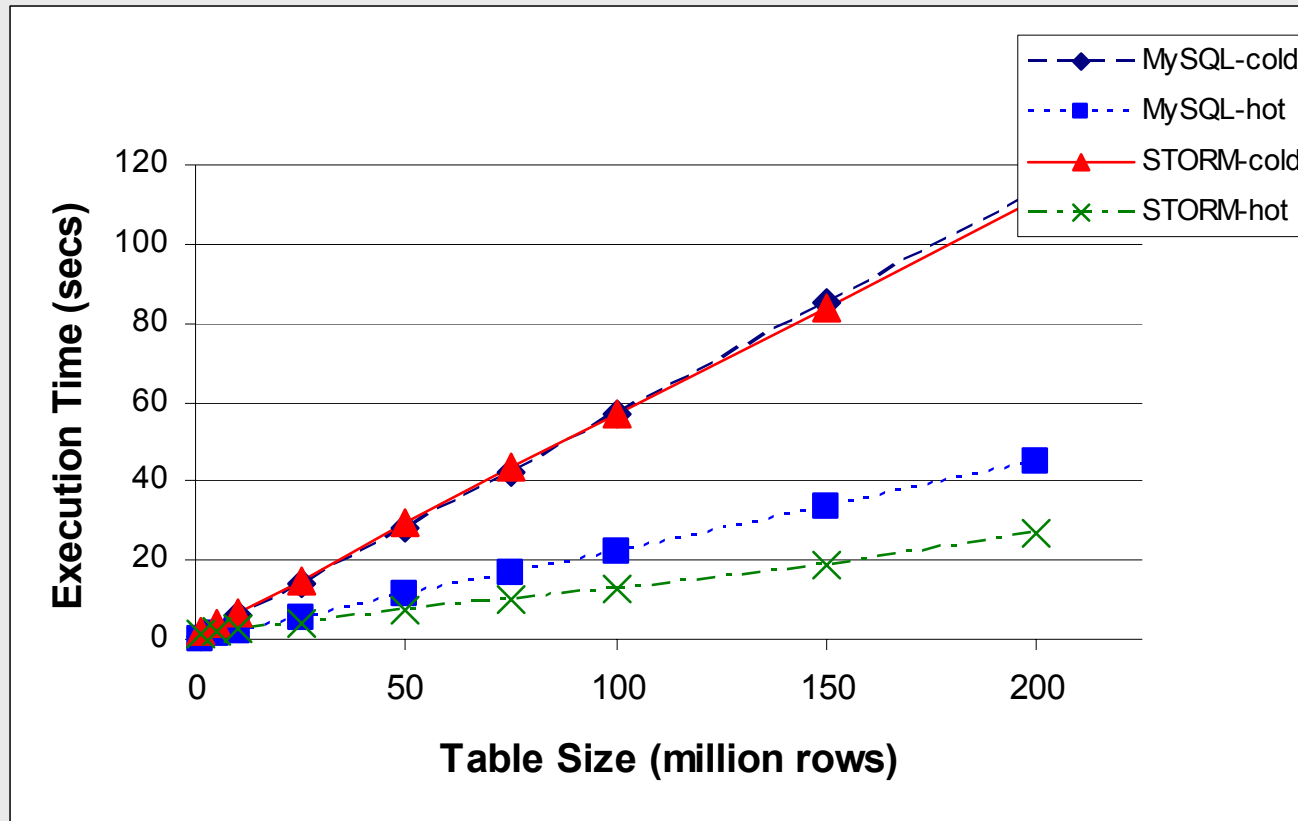
10-25GB per file.

About 30-35TB of Data.



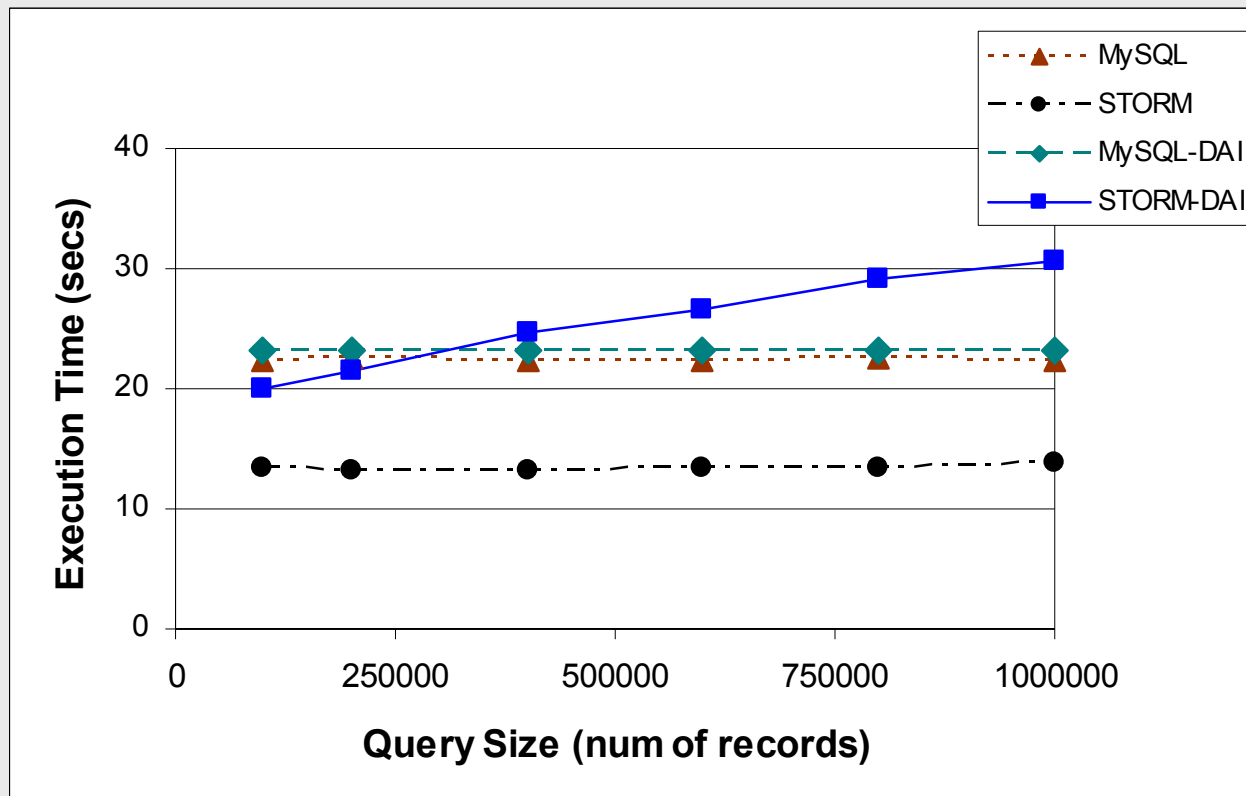
# Comparison with MySQL - 1

- Varying table size.
- Per tuple cost is lesser



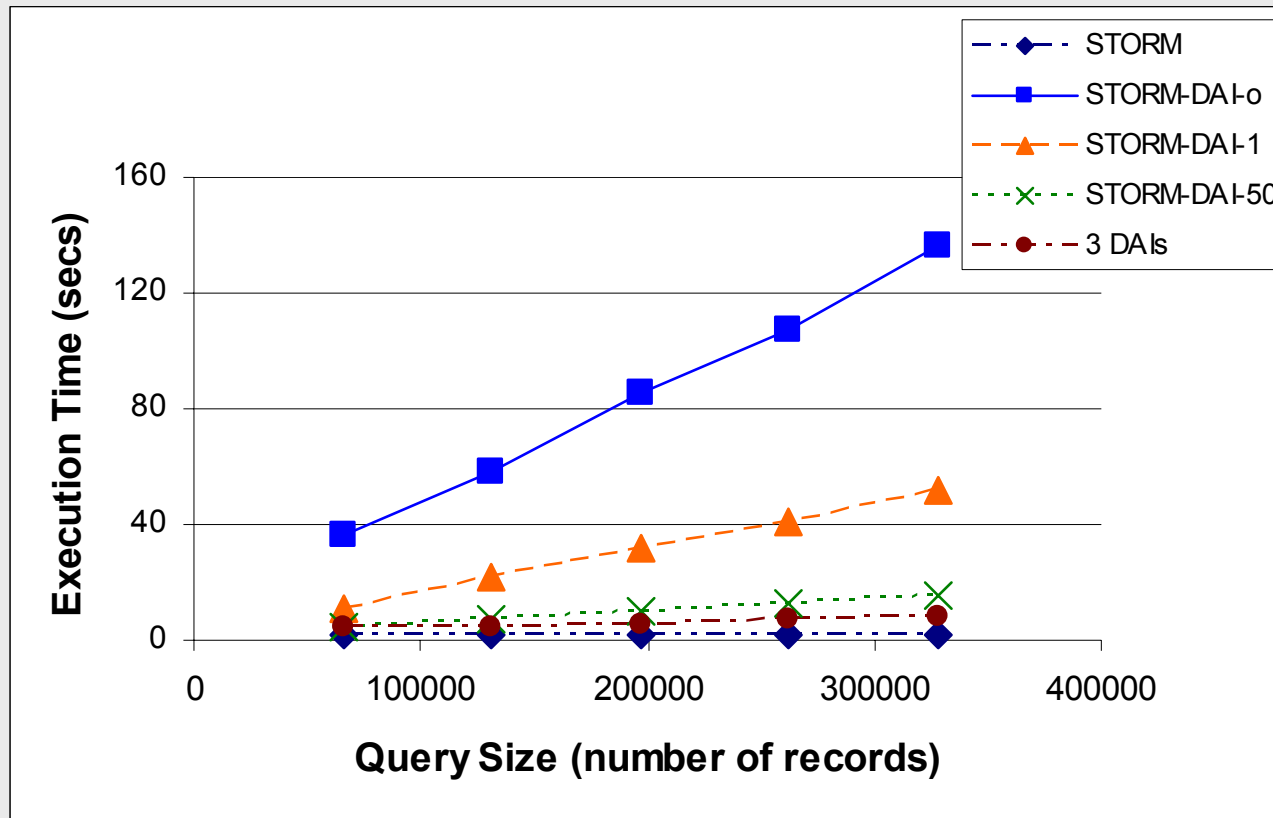
# Comparison with MySQL - 2

- Varying query size
- Also compare them as data resources



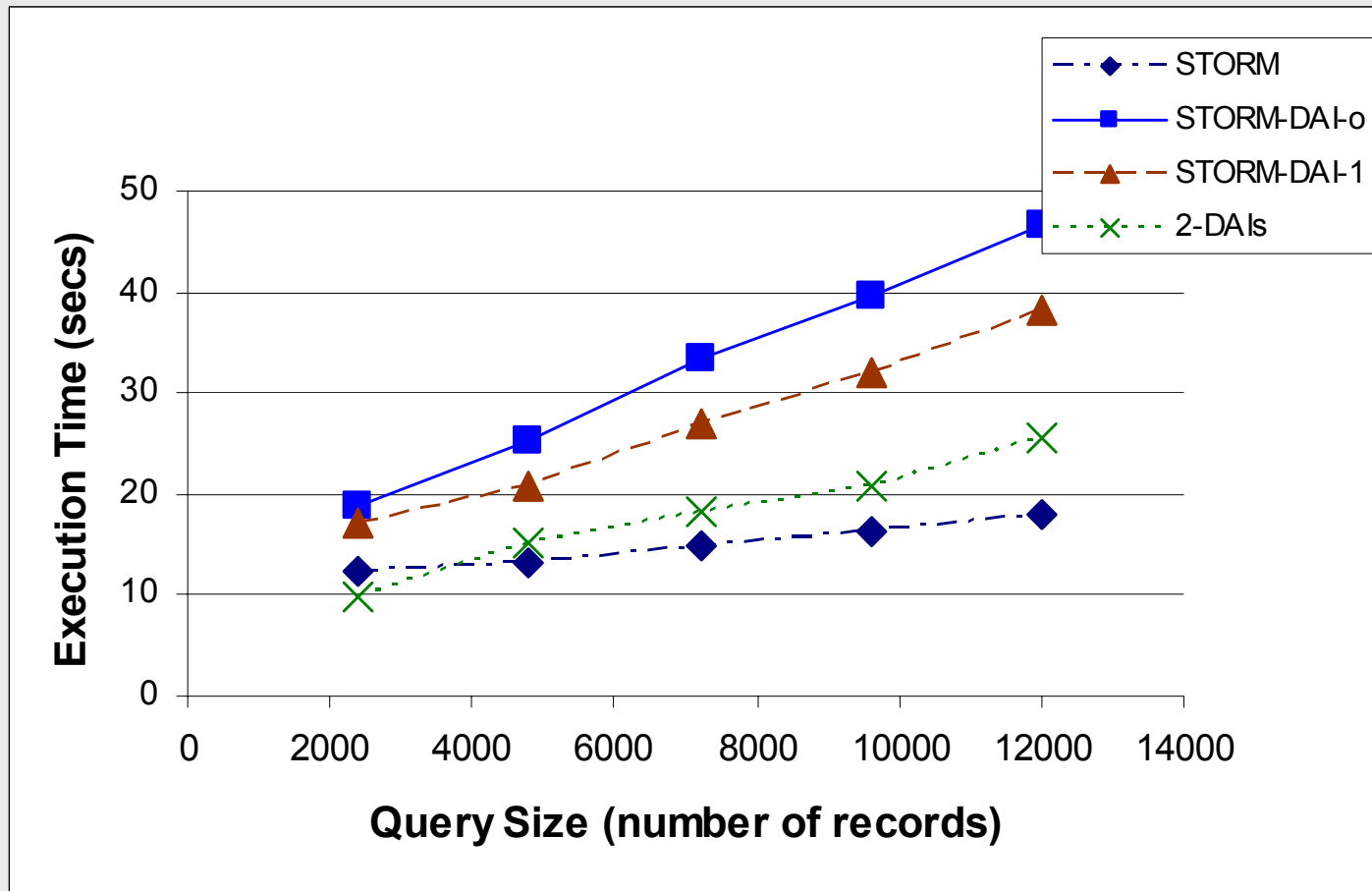
# Oil Reservoir Data Results

- Improvements due to: treating records as array of bytes, combining results at client



# Seismic Data Results

- 96 x 11GB files on 16 nodes



# Conclusions

- Overview of work related to Large Scale Scientific Data Management at Multi-Scale Computing Lab
- Exposed STORM as a Grid Data Service
  - Results on use case: Oil reservoir management
  
- For more info / to download STORM, DataCutter, Mobius
  - <http://www.multiscalecomputing.org>
  - or
  - <http://www.bmi.osu.edu>