Databases and Systems Software for Multi-Scale Problems

Joel Saltz

University of Maryland College Park

Computer Science Department

Johns Hopkins Medical Institutions

Pathology Department

NPACI

Vision

- Multi-petabyte distributed data collections
  - sensor measurements, scientific simulations, media archives
- Subset and filter
  - load small subset of data into disk cache or client
- Tools to support on-demand data product generation, interactive data exploration

Overview

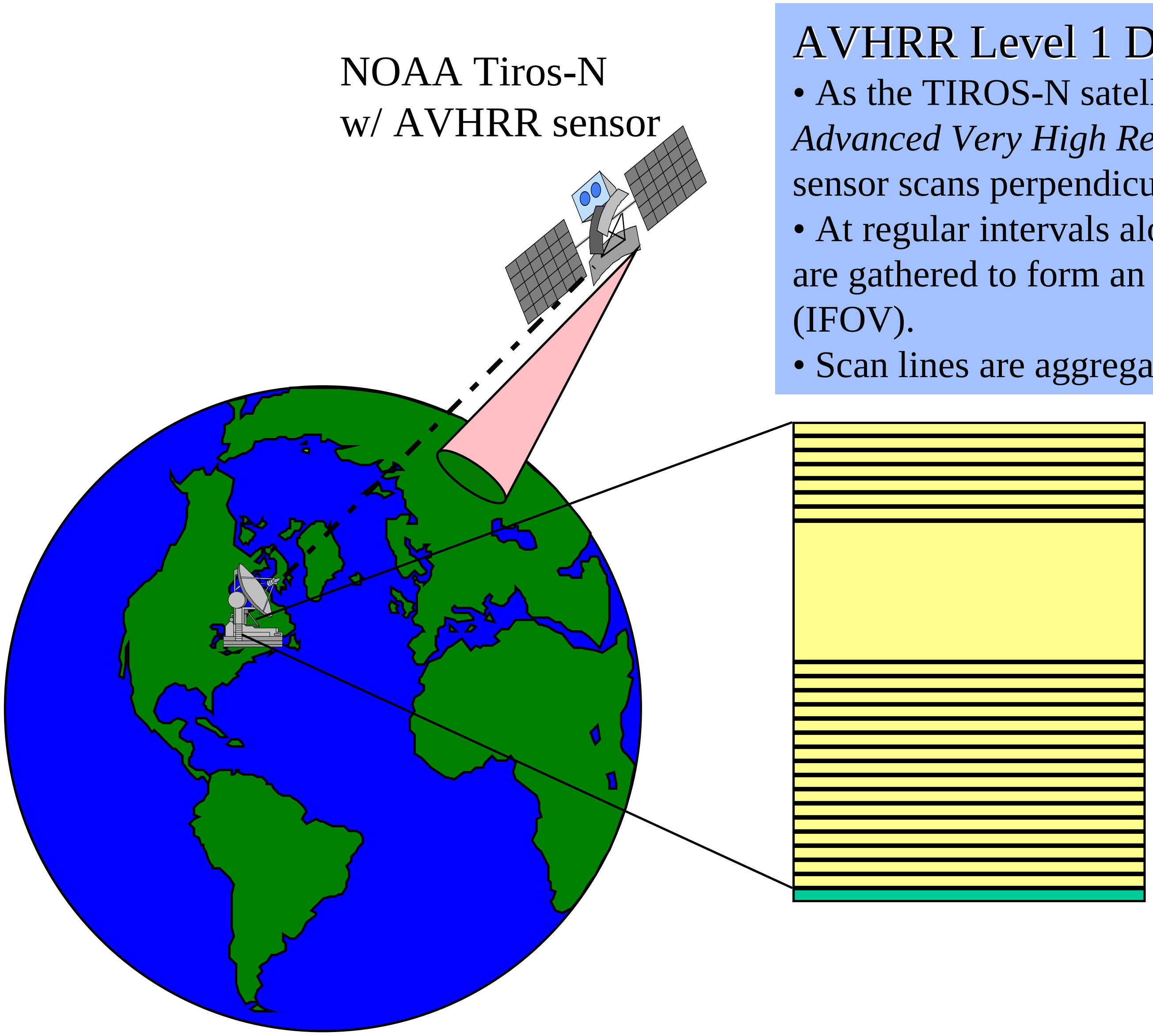
- Application Domain: Multi-scale Data Intensive Applications
- Overview of System Software Architecture
- Active Data Repository -- Design and Query Planning
- Overview of Performance Engineering Methodology
- Conclusions

Application Scenarios

Processing Remotely Sensed Data

AVHRR Level 1 Data

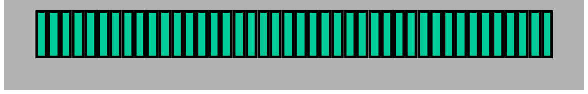
- As the TIROS-N satellite orbits, the *Advanced Very High Resolution Radiometer* (AVHRR) sensor scans perpendicular to the satellite’s track.
- At regular intervals along a scan line measurements are gathered to form an *instantaneous field of view* (IFOV).
- Scan lines are aggregated into Level 1 data sets.



A single file of *Global Area Coverage* (GAC) data represents:

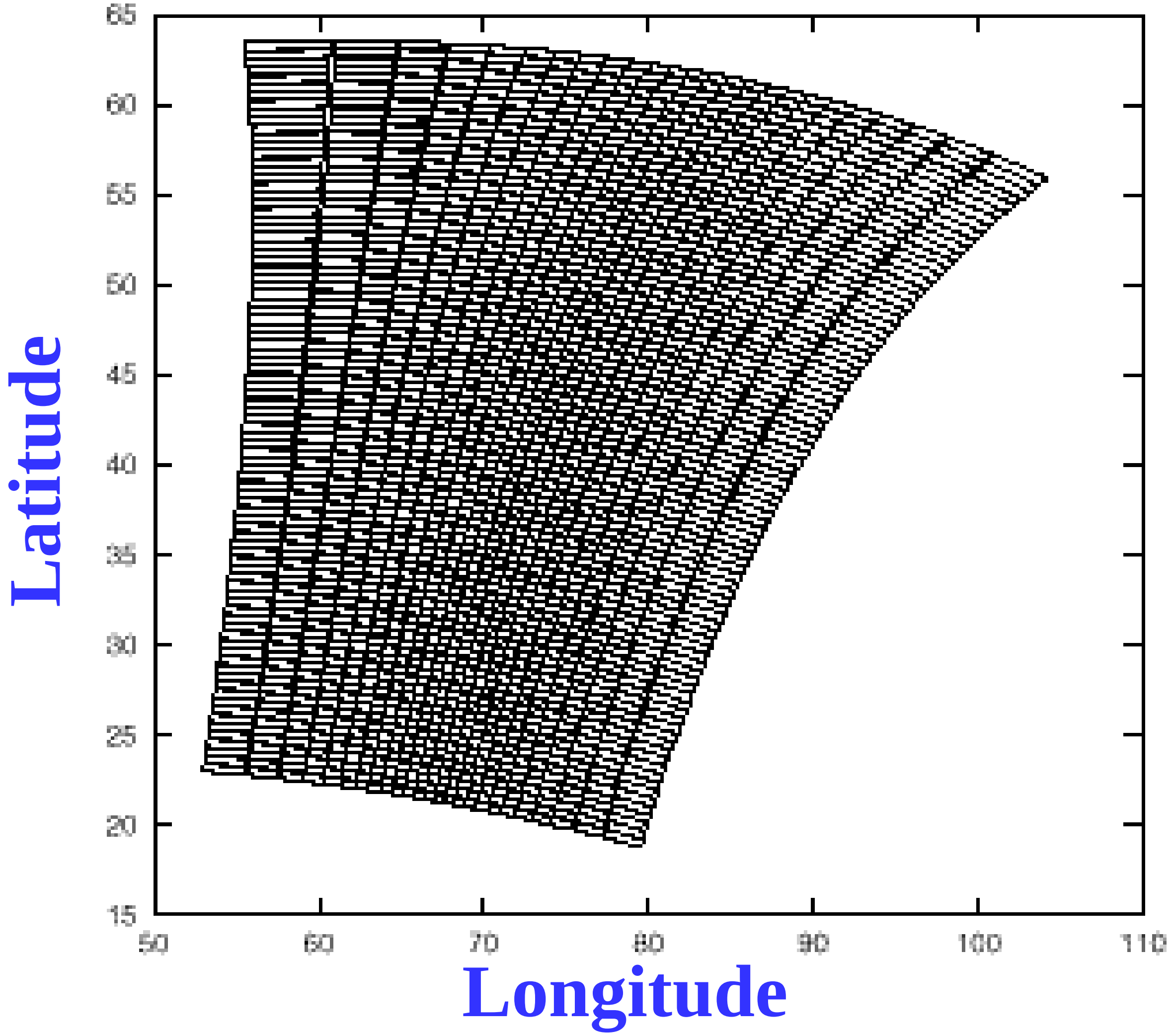
- ~one full earth orbit.
- ~110 minutes.
- ~40 megabytes.
- ~15,000 scan lines.

One scan line is 409 IFOV’s



Spatial Irregularity

AVHRR Level 1B NOAA-7 Satellite 16x16 IFOV blocks.



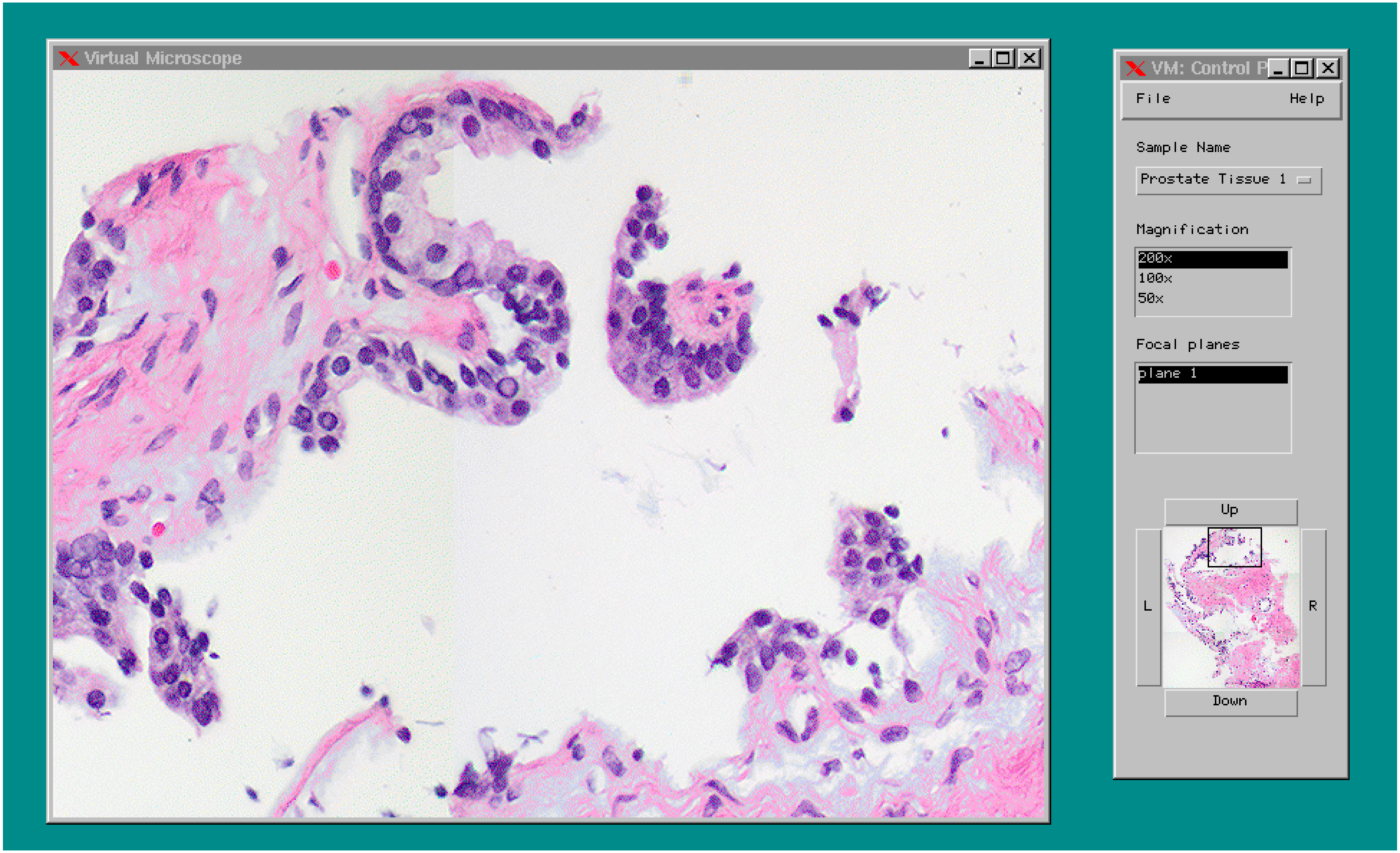
Processing

- Characterize changes in land cover
- Assimilate into weather and climate models
- Assimilate into ecological models
- Visualize
- Identify structures, vehicles

Pathology Application Domain

- Automated capture of, and immediate worldwide access to all Pathology case material
  - light microscopy, electrophoresis (PEP, IFE), blood smears, cytogenetics, molecular diagnostic data,clinical laboratory data.
- Slide data -- .5-10 GB (compressed) per slide -- Johns Hopkins alone generates 500,000 slides per year
- *Digital storage of 10% of slides in USA -- 50 petabytes per year*

Virtual Microscope Client



Sample Name
Prostate Tissue 1

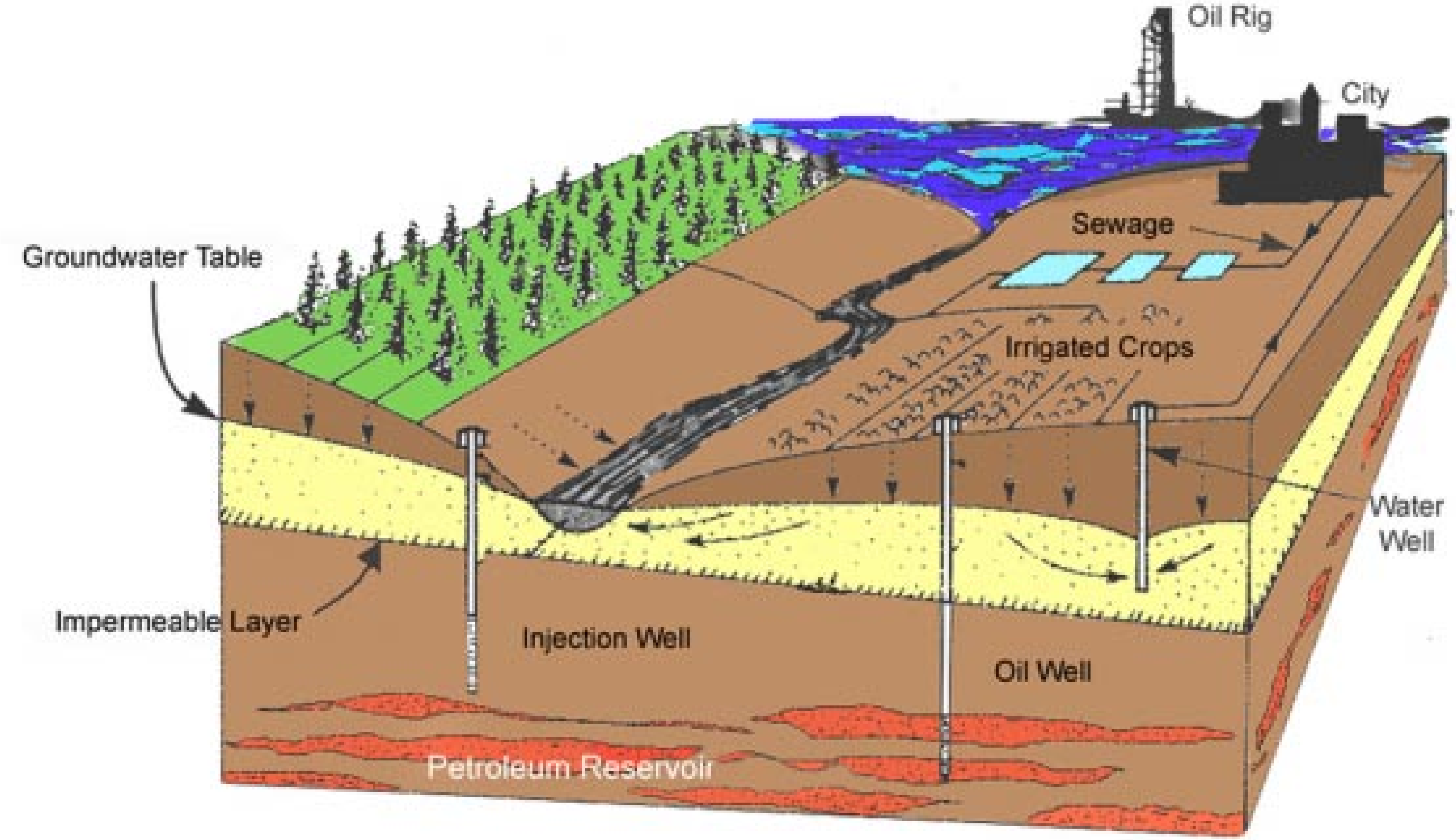
Magnification
200x
100x
50x

Focal planes
plane 1

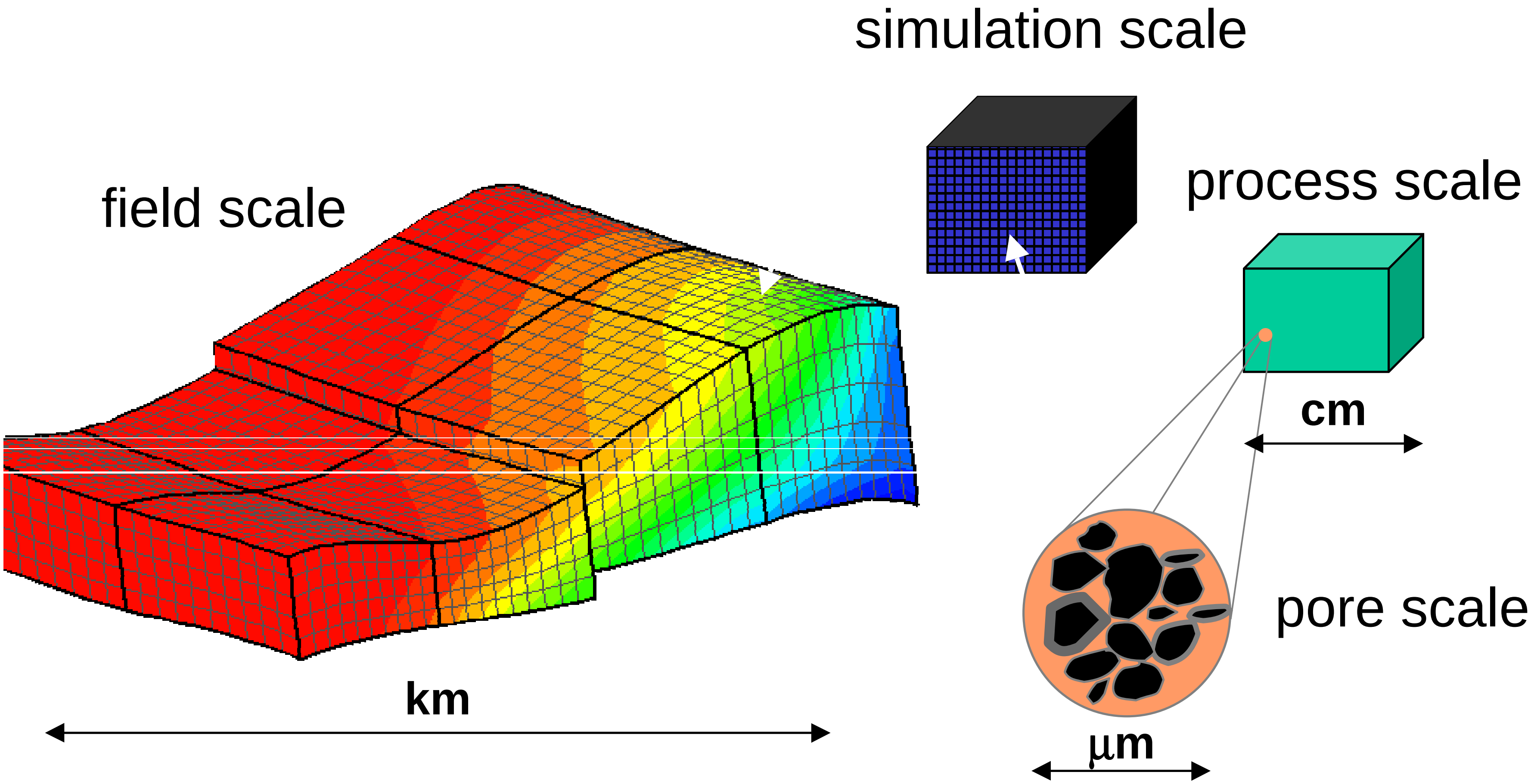
Computations

- Screen for cancer
- Categorize images for associative retrieval
  - which images look like this unknown specimen
- Visualize and explore dataset
- 3-D reconstruction

Coupled Ground Water and Surface Water Simulations



The Tyranny of Scale



simulation scale

field scale

process scale

cm

pore scale

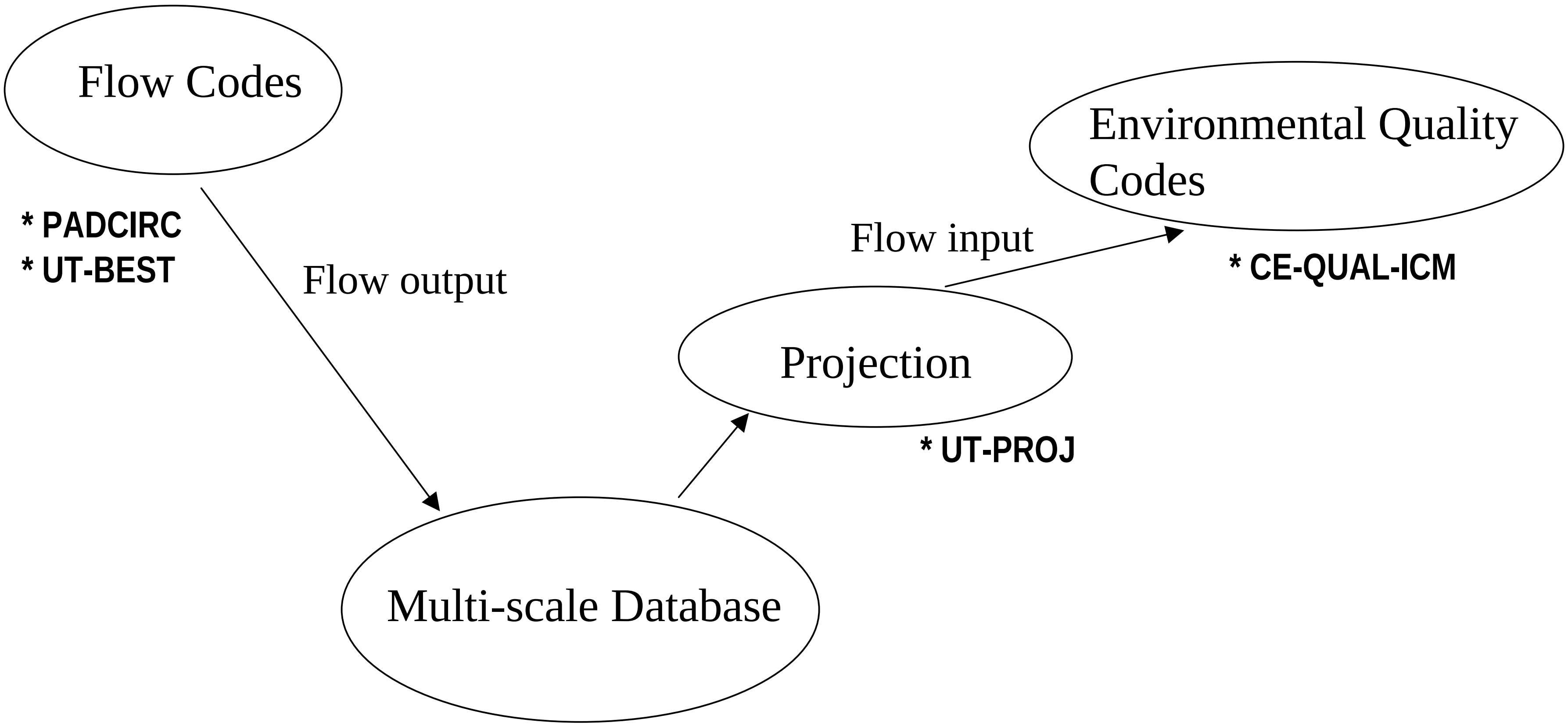
km

µm

Computations

- Spread of pollutants
- Chemical and biological reactions in waterways
- Estimate spread of contamination in ground and surface water
- Best and worst case oil production scenarios (history matching)

Database Couples Programs
(Coupling of Flow Codes with Environmental Quality Codes)


Flow Codes

* PADCIRC
* UT-BEST

Flow output

Multi-scale Database

Projection

* UT-PROJ

Flow input

Environmental Quality Codes

* CE-QUAL-ICM

* *Storage, retrieval, processing of multiple datasets from different flow codes*

Attributes common to these applications

Common Themes

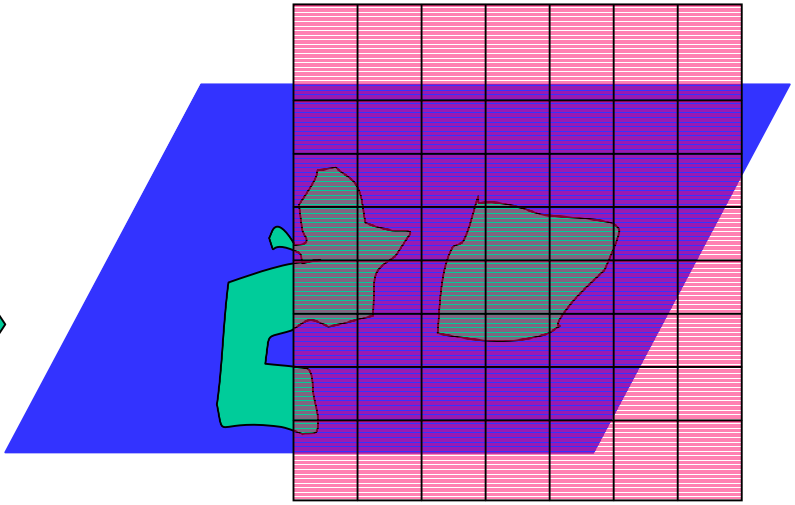
- Spatial/multidimensional multi-scale, multi-resolution datasets
- Multiple spatio-temporal queries
- Complex preprocessing
- Dataset exploration or program coupling

Querying Irregular Multidimensional Datasets

- Irregular datasets
  - Think of *disk based* unstructured meshes, data structures used in adaptive multiple grid calculations
    - indexed by spatial location
  - Iterator specified by spatial query
    - computation aggregates data - data product size smaller than results of range query

Typical Query

Output grid onto
which a projection
is carried out



Specify portion of raw
sensor data corresponding
to some search criterion

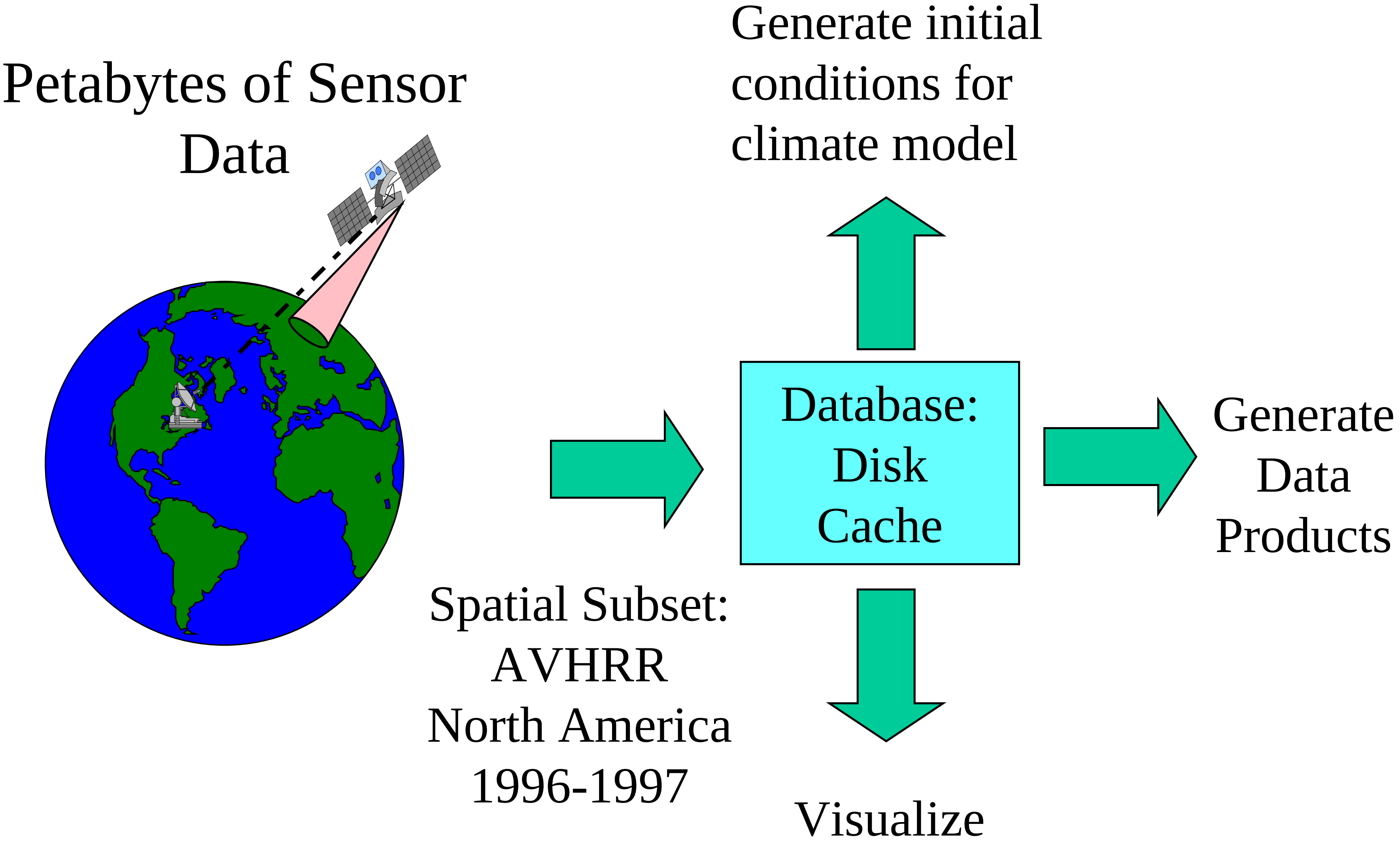
Overview

- Application Domain: Multi-scale Data Intensive Applications
- Overview of System Software Architecture
- Active Data Repository -- Design and Query Planning
- Overview of Performance Engineering Methodology
- Conclusions

Components of System Software Architecture

- Spatial Queries and filtering on distributed data collections
  - Spatial subset and filter (ADR')
  - Load disk caches with subsets of huge multi-scale datasets
- Toolkit for producing data product servers
  - C++ toolkit targets SP, clusters
  - Compiler front end
    - extension of inspector/executor

Generating Data Subsets



Petabytes of Sensor Data

Spatial Subset:
AVHRR
North America
1996-1997

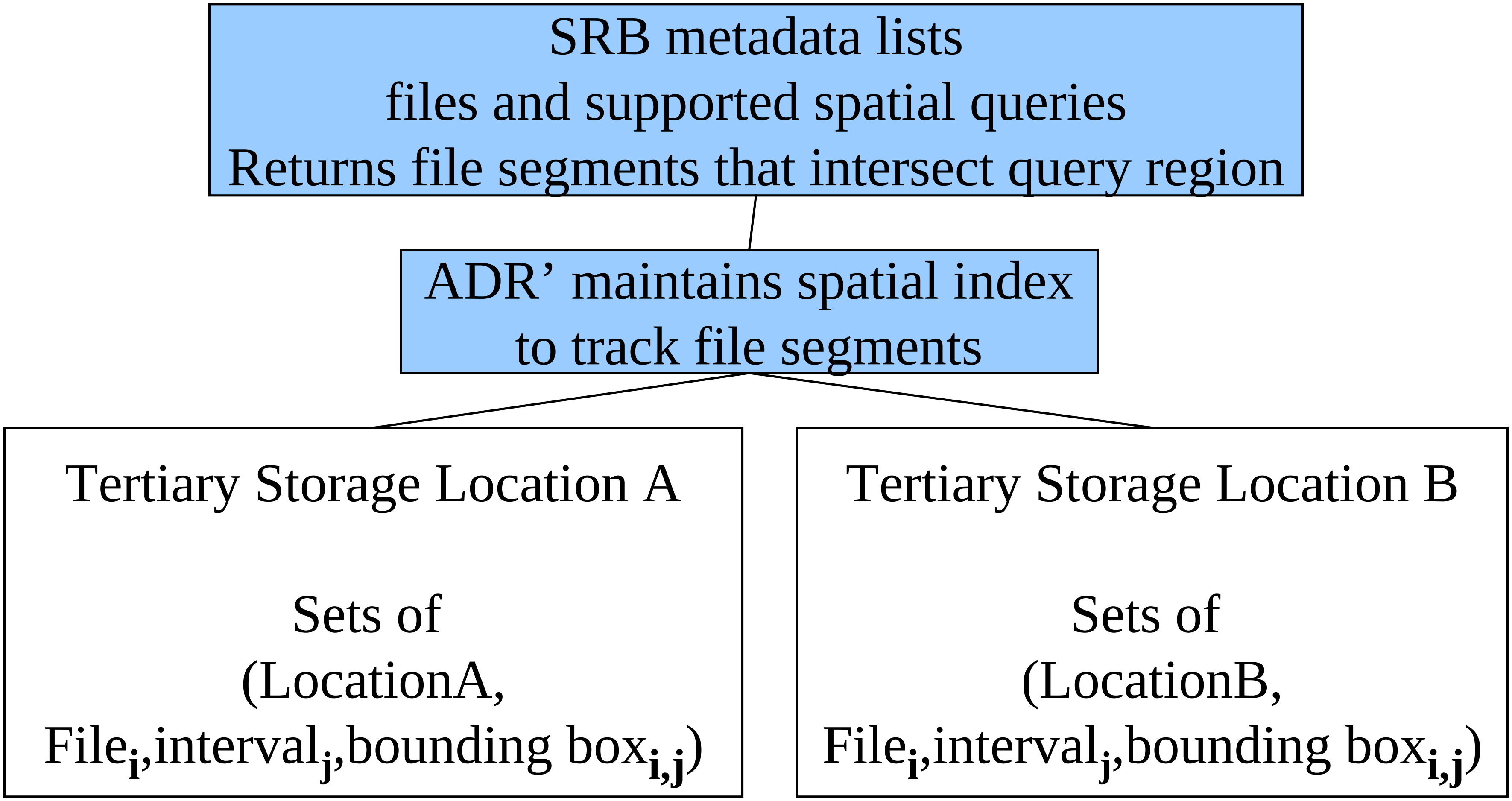
Database:
Disk
Cache

Generate initial conditions for climate model

Generate Data Products

Visualize

Current ADR’ Architecture



SRB metadata lists
files and supported spatial queries
Returns file segments that intersect query region

ADR’ maintains spatial index
to track file segments

Tertiary Storage Location A

Sets of
(LocationA,
File$_i$,interval$_j$,bounding box$_{i,j}$)

Tertiary Storage Location B

Sets of
(LocationB,
File$_i$,interval$_j$,bounding box$_{i,j}$)

Future ADR’ Architecture

- Proxy processes (disklets) filter data as it is extracted from tertiary storage
- File segment partitioned into chunks, disklets extract necessary data from each chunk
- Early data filtering reduces data movement and data transfer costs
- Can be generalized to extend beyond filtering --
  - Uysal has developed algorithms that use fixed amount of scratch memory to carry out selects, sorts, joins, datacube operations

Database operations supported by Disklet Algorithms

- SQL select + aggregate
- SQL group-by [Graefe - Comp Surveys’93]
- External sort [NowSort - SIGMOD’97]
- Datacube [PipeHash - SIGMOD’96]
- Frequent itemsets [eclat- SPAA’97]
- Sort-merge join
- Materialized views [SIGMOD’96,PDIS’96]

Overview

- Application Domain: Multi-scale Data Intensive Applications
- Overview of System Software Architecture
- Active Data Repository -- Design and Query Planning
- Overview of Performance Engineering Methodology
- Conclusions

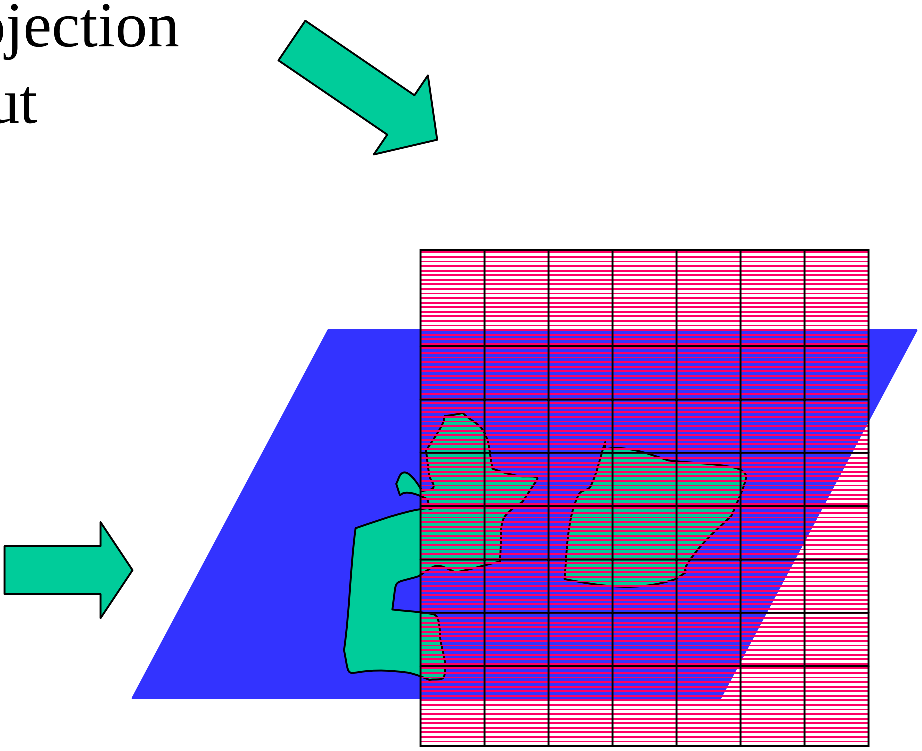
Database Software
Active Data Repository

- *Optimized associative access and processing of multiresolution disk based data structures*
- User-defined projection and aggregation functions
- Targets *parallel and distributed architectures* that have been configured to support high I/O rates
- Modular services implemented in C++
- Satellite sensor data; Virtual Microscope Server, Bay and Estuary Simulation

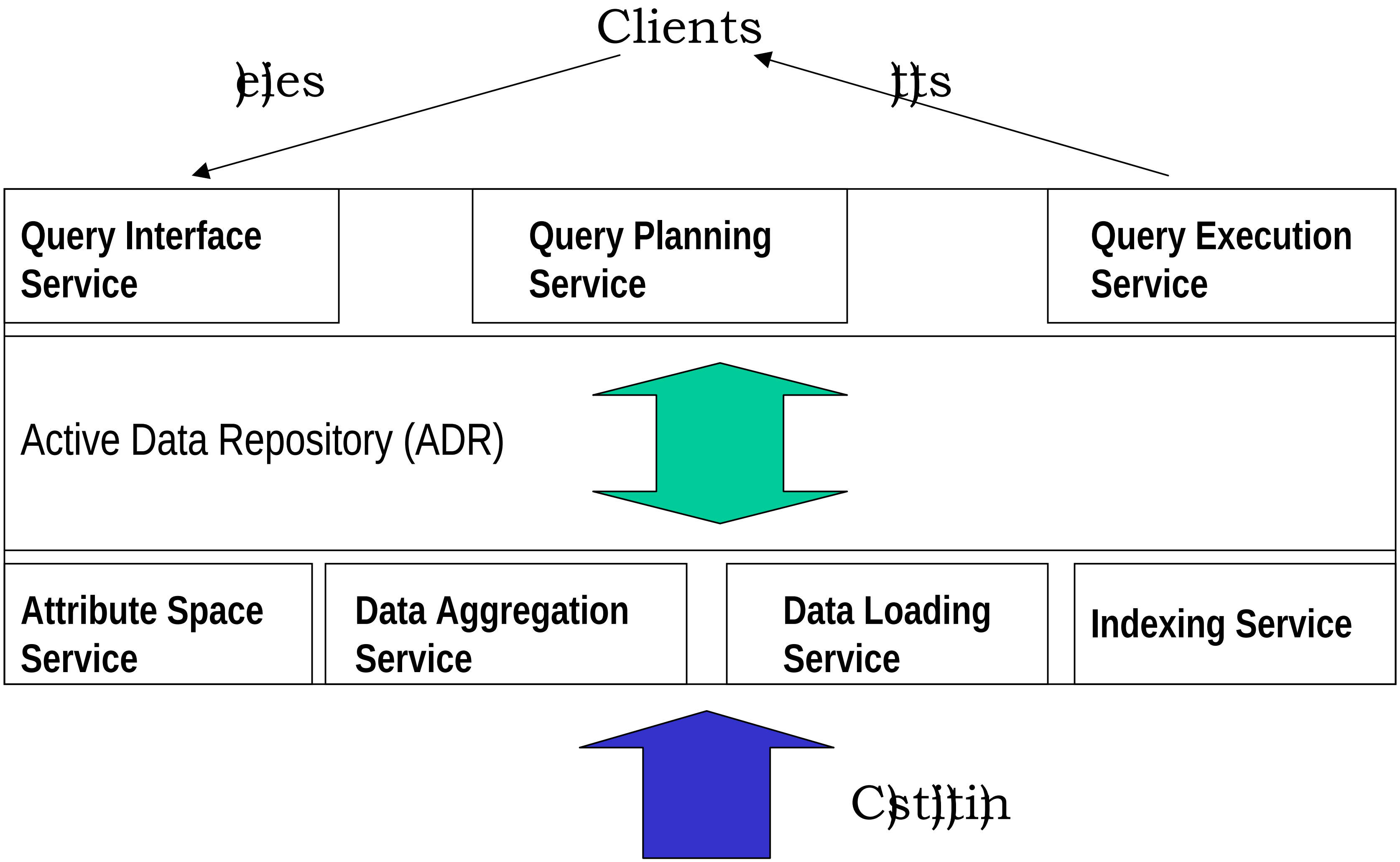
Typical Query

Output grid onto
which a projection
is carried out

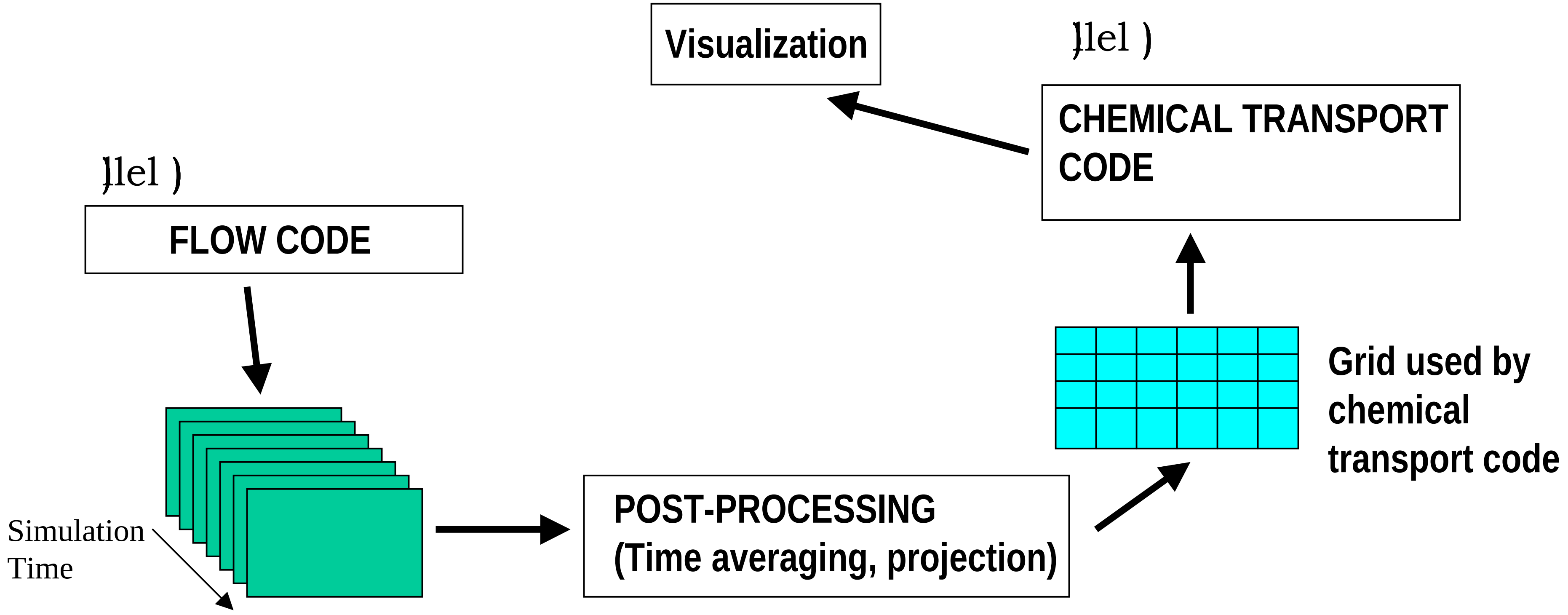
Input
dataset (e.g. raw
sensor data)



Architecture of Active Data Repository



Water Contamination Studies



Visualization

llel)

CHEMICAL TRANSPORT CODE

llel)

FLOW CODE

Grid used by chemical transport code

Simulation Time

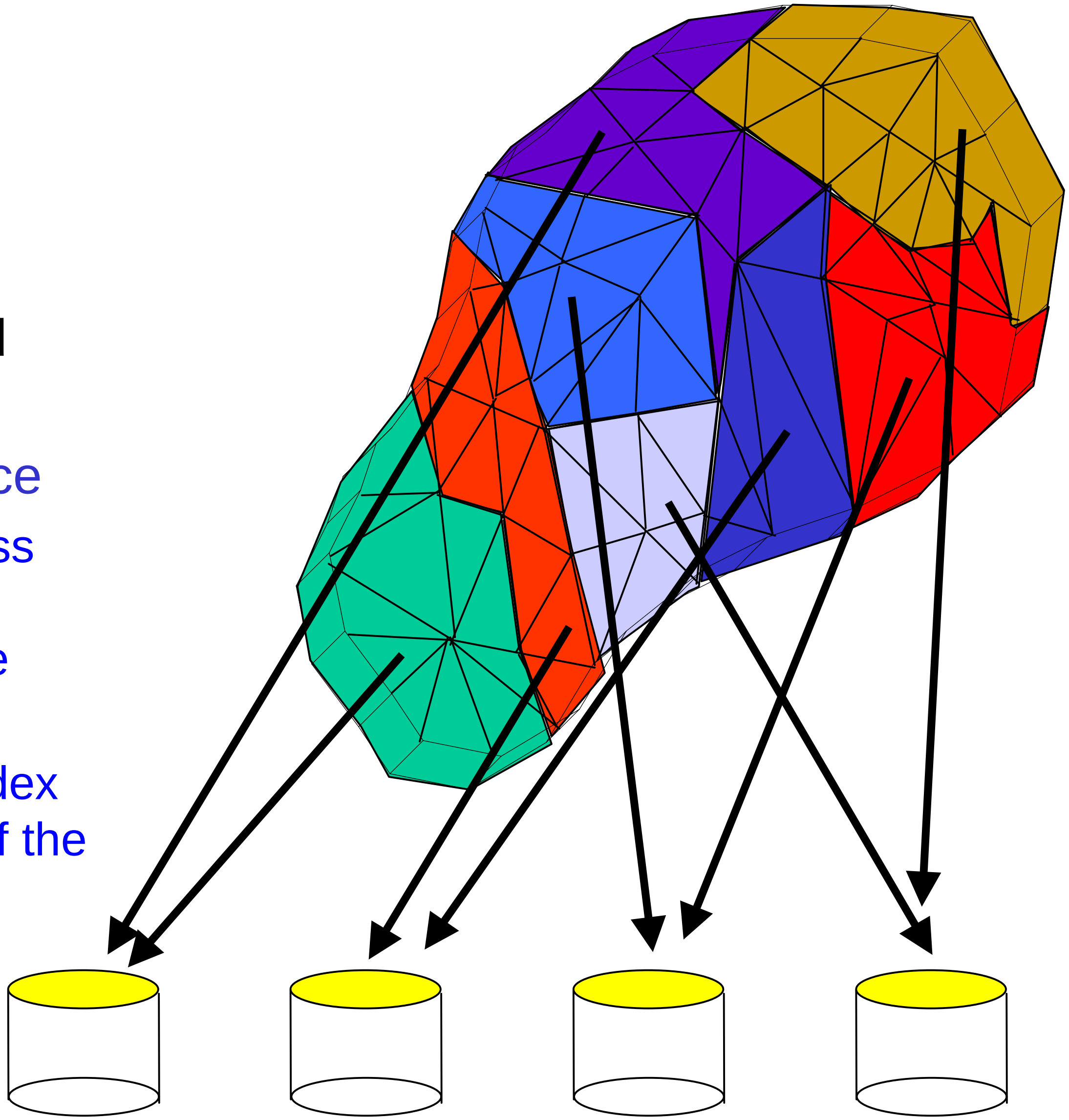
POST-PROCESSING
(Time averaging, projection)

Hydrodynamics output
(velocity,elevation)
on unstructured grid

* Locally conservative projection
* Management of large amounts of data

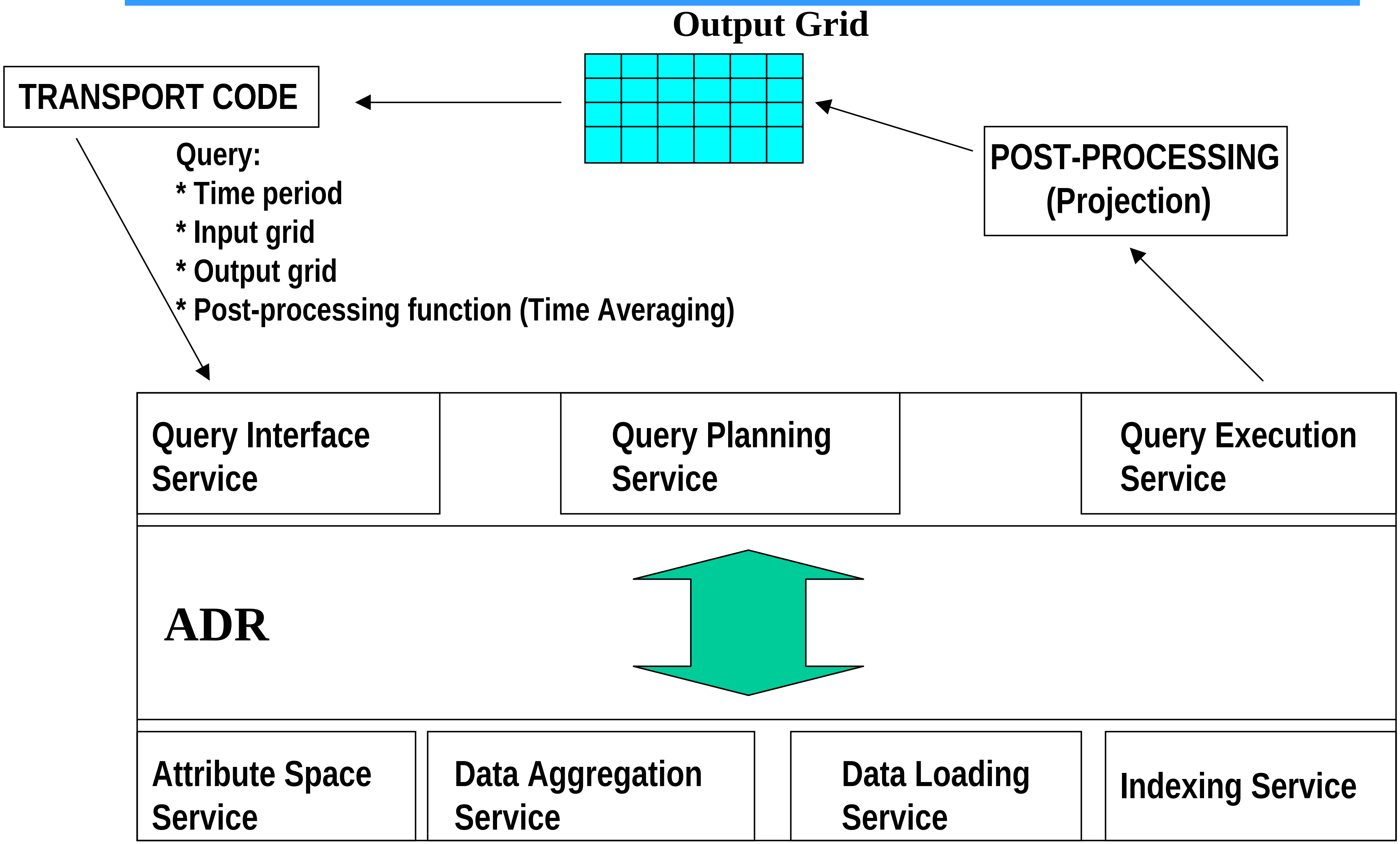
Loading Grids into ADR

- Partition grid into data chunks -- each chunk contains a set of volume elements
- Each chunk is associated with a bounding box
- ADR Data Loading Service
  - Distributes chunks across the disks in the system (e.g., using Hilbert curve based declustering)
  - Constructs an R-tree index using bounding boxes of the data chunks



Disk Farm

Water Contamination Studies



Output Grid

TRANSPORT CODE

Query:
* Time period
* Input grid
* Output grid
* Post-processing function (Time Averaging)

POST-PROCESSING (Projection)

Query Interface Service

Query Planning Service

Query Execution Service

ADR

Attribute Space Service

Data Aggregation Service

Data Loading Service

Indexing Service

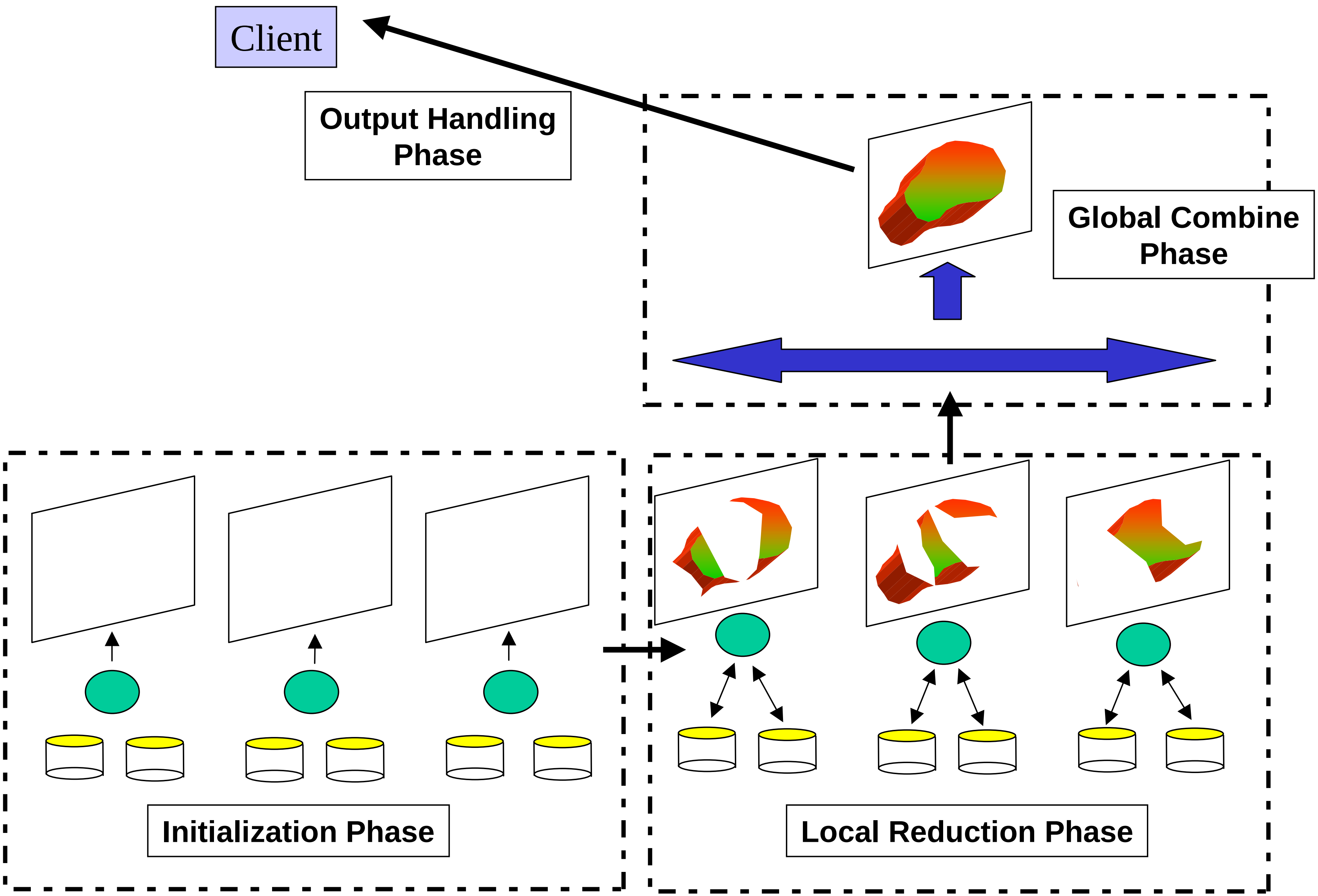
Executing Queries

- Very large input, output datasets
- Clustered/declustered across storage units (Analysis of clustering, declustering algorithms -- PhD B. Moon)
- Datasets partitioned into “chunks”
  - Each chunk has associated minimum bounding rectangle
- Processing involves
  - spatial queries
  - user defined projection, aggregation functions
  - accumulator used to store partial results
  - accumulator tiled
- Spatial index used to identify locations of all chunks

Query Execution

- For each accumulator tile:
  - Initialization -- allocate space and initialize
  - Local Reduction -- input data chunks on each processor’s local disk -- aggregate into accumulator chunks
  - Global Combine -- partial results from each processor combined
  - Output Handling -- create new dataset, update output dataset or serve to clients

Query Processing



Query Planning Strategies

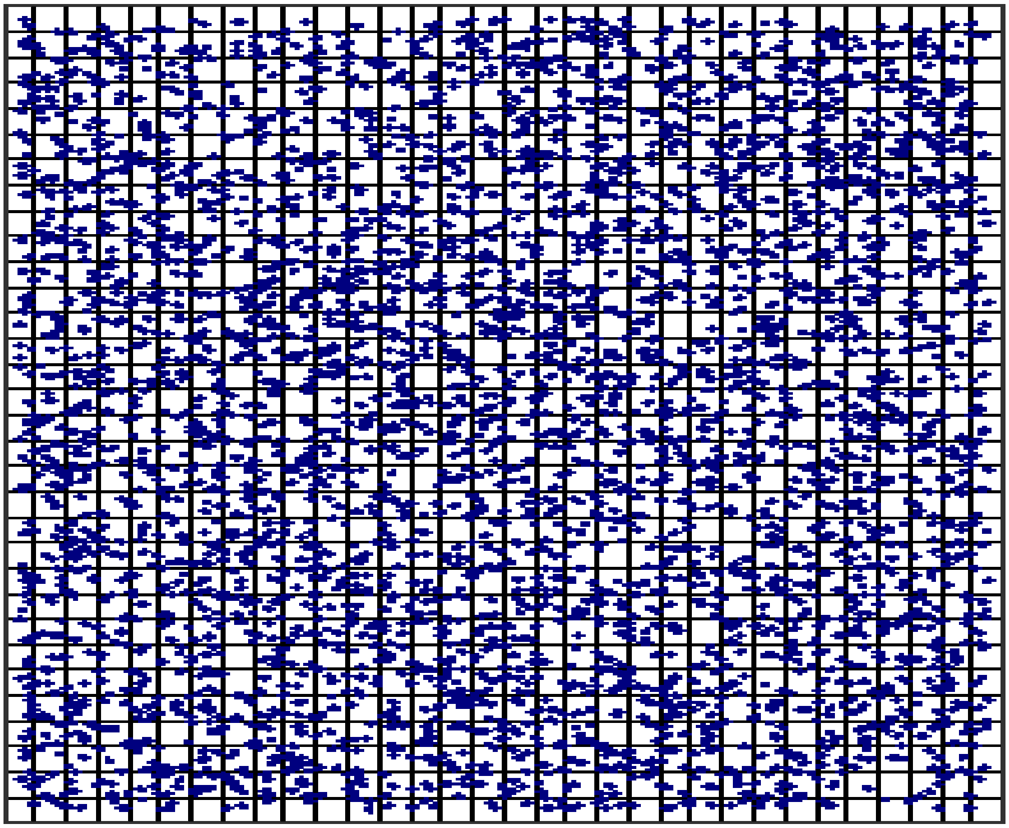
- Fully replicated accumulator strategy
  - Partition accumulator into tiles
  - Each tile is small enough to fit into single processor's memory
  - Accumulator tile is replicated across processors
  - Input chunks living on disk attached to processor P is accumulated into tile on P
  - Global combine employs accumulation function to merge data from replicated tiles

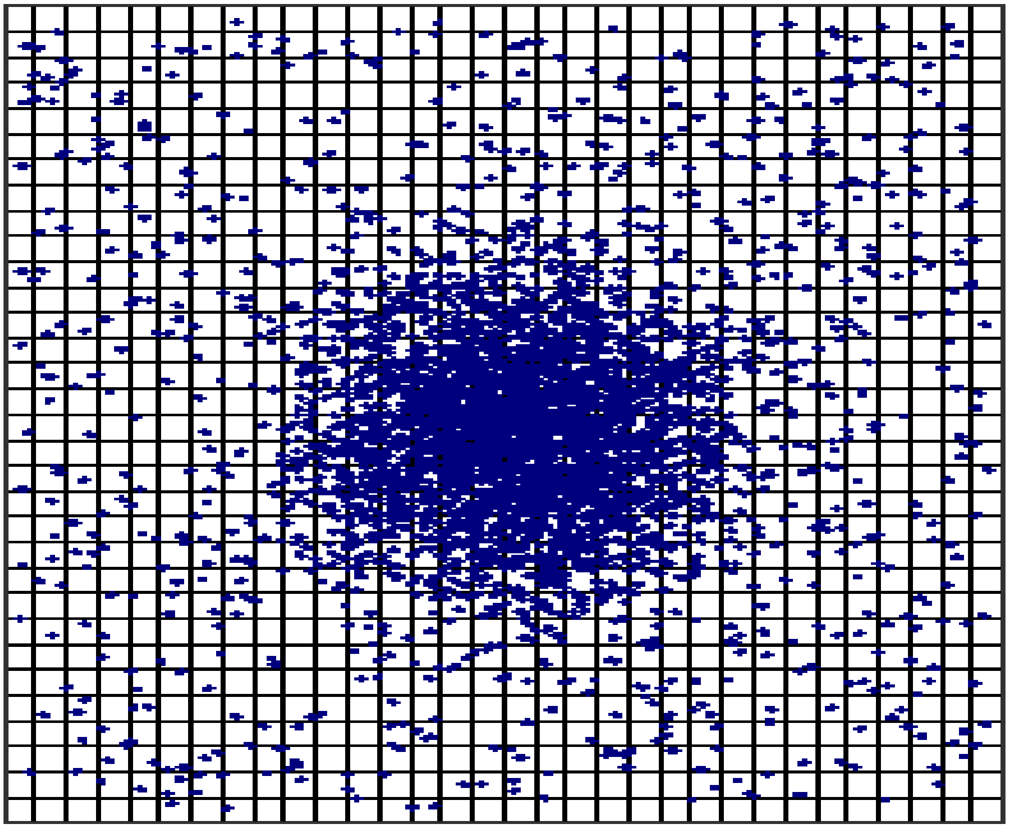
Query Planning Strategies

- Sparsely replicated accumulator strategy
  - Sparse data structures are used in chunk accumulation
- Distributed Accumulator Strategy
  - Partition accumulator between processors
  - Single processor “owns” accumulator chunk
  - Carry out all accumulations on processor that owns chunk

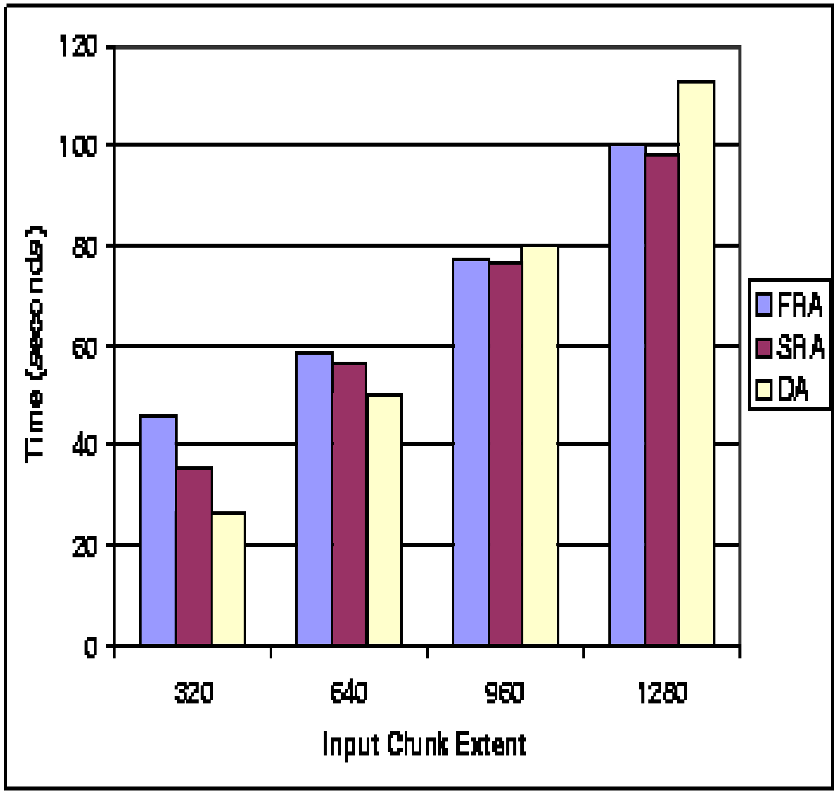
Studies to evaluate query processing strategies

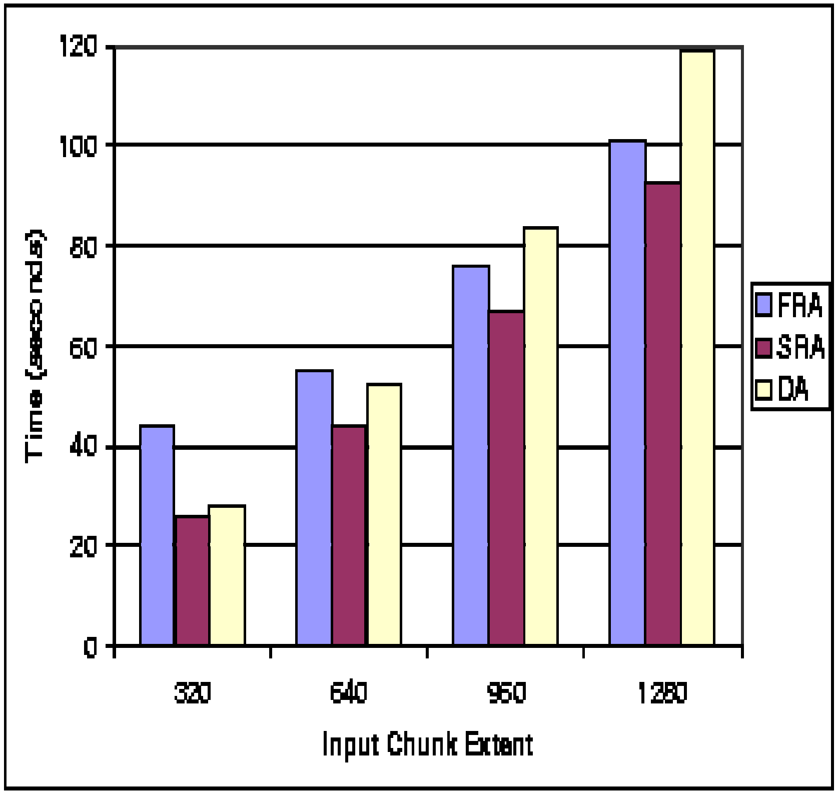
- Projection of 3-D datasets onto 2-D grid
- Query windows of various sizes directed at synthetic datasets with uniform, skewed data distributions
- Sparse replicated accumulator wins when there is a high degree of fan-in -- communication can be saved by local accumulation of multiple chunks
- Distributed accumulator wins when there is a low degree of fan-in
  - avoids overhead arising from computation and datastructure manipulations arising from both local accumulation and subsequent combining stage
  - minor decrease in I/O due to bigger tiles





Effect of Accumulator Strategy on Performance





Conclusion

- ADR, ADR' support several applications
- Plans to incorporate as part of NPACI data handling infrastructure
- Challenges:
  - Scaling up
  - Efficient querying and and processing in very large data collections
  - High level language interface -- ADR as database extender
    - Extend past irregular compilation and interprocedural analysis work to generate optimized queries

Research Group

- Alan Sussman, Tahsin Kurc, Charlie Chang, Renato Ferraria, Mustafa Uysal -- University of Maryland
- Work done in collaboration with National Partnership for Applied Computational Infrastructure