Virtual Microscope: Databases and Systems Software for Multi-Scale Problems

Joel Saltz

Johns Hopkins Medical Institutions

Pathology Department

University of Maryland College Park

Computer Science Department

Data Storage Architectures

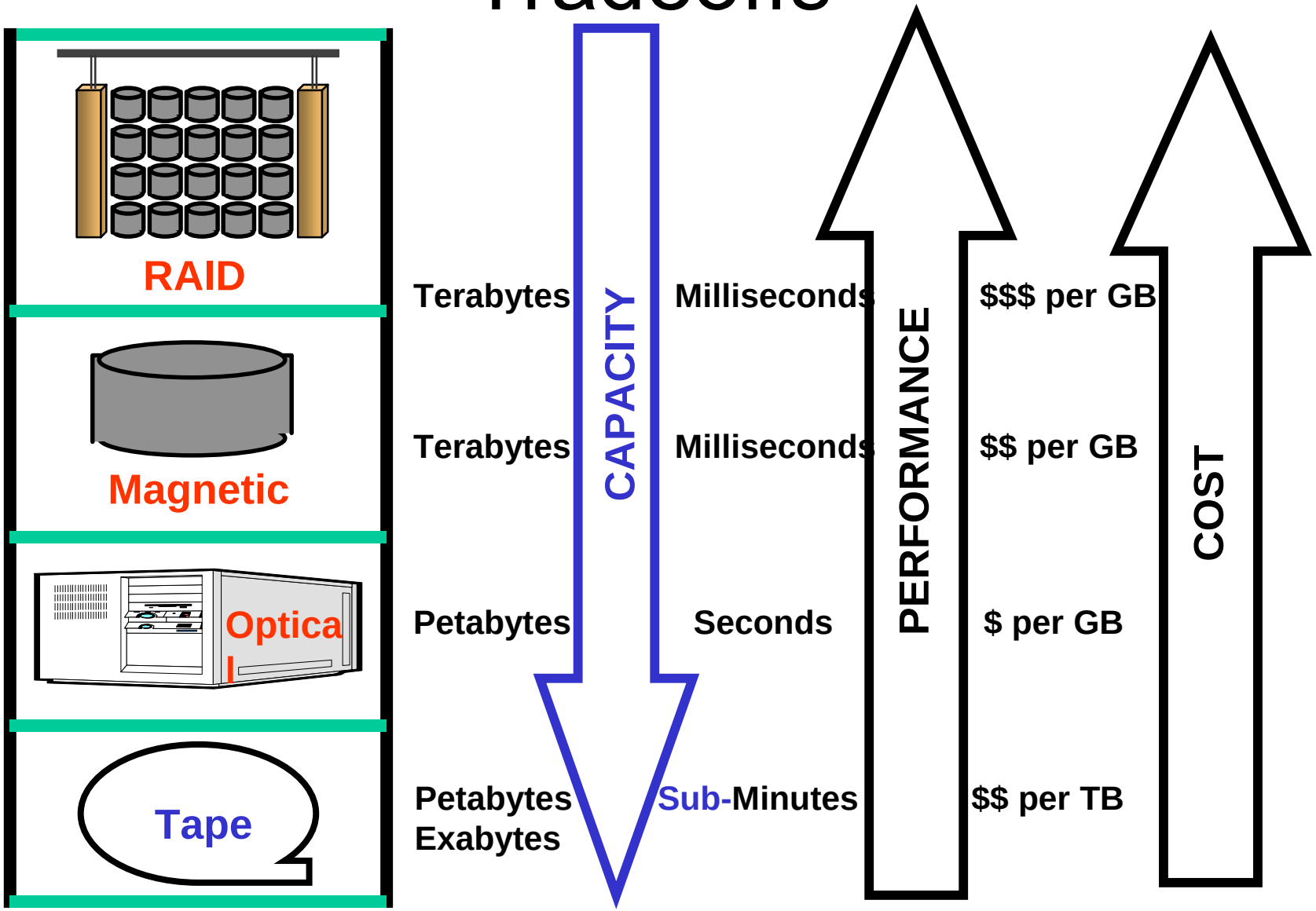
Storage Hierarchy

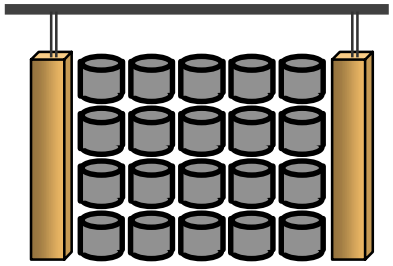
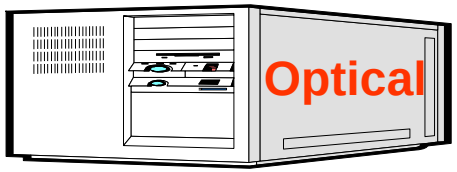
- Disk prices plummeting by roughly 70% per year
- Can build architectures out of many PCs and disks
- 15-25K per terabyte for 1 TB disk storage
- Plummeting disk prices putting downward pressure on other storage hierarchy components (optical, tape)

Deep Storage Hierarchies

- Tapes, low end disk farms, high performance disk caches all needed as part of system configuration
- Software to manage deep storage hierarchies
  - storagetek
  - supercomputer centers
  - DOE laboratories
  - web servers

Capacity, Cost & Performance Tradeoffs



Storage Media	Cost
 RAID	100% of Disk Dollar
 Optical	43% of Disk Dollar
Tape	7% of Disk Dollar

Computational Grid

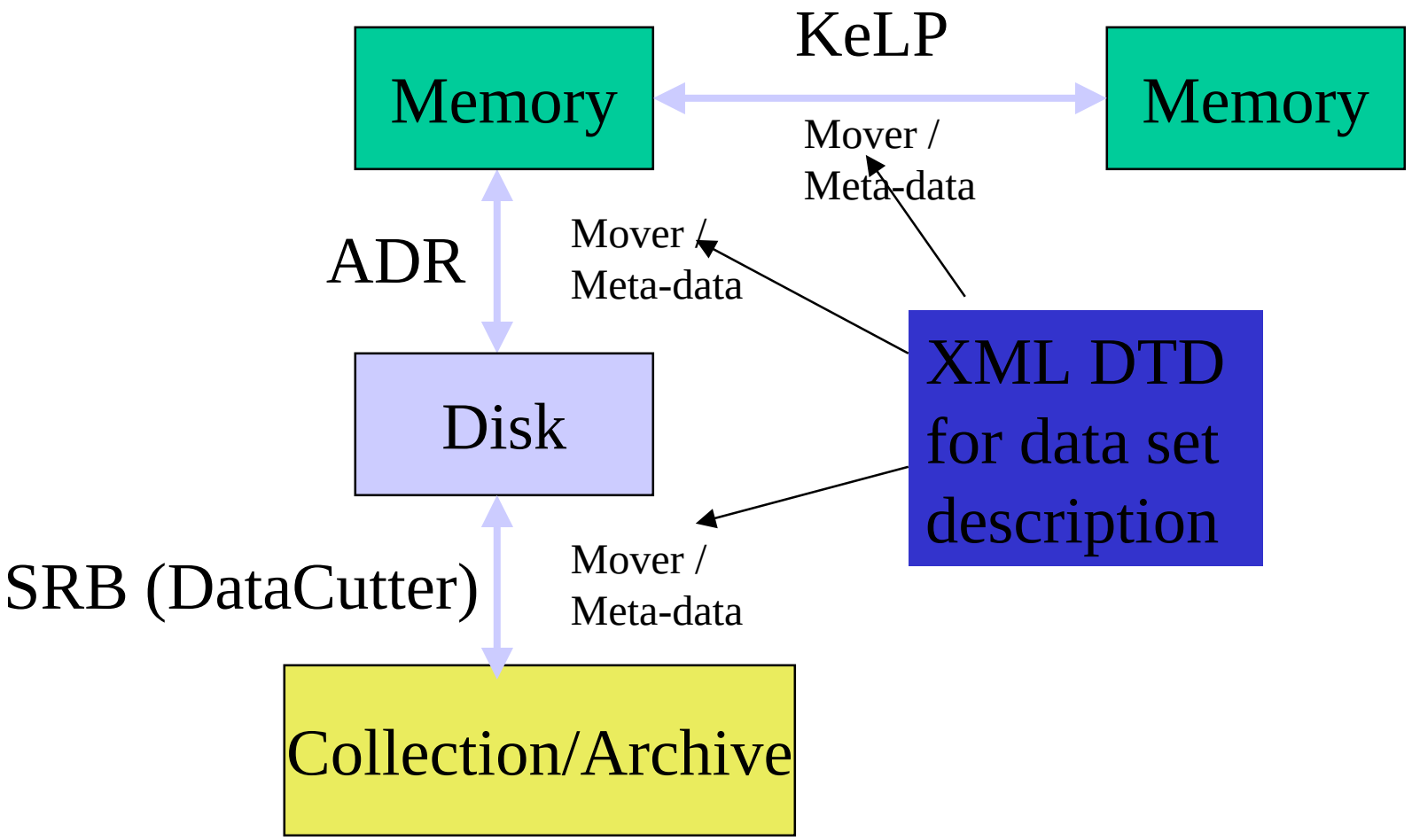
- Many compute servers
- Many data servers
- Rich network interconnections
- User accesses these distributed resources with ease and marshals appropriate resources to handle application at hand
- Analogy to electric power grid
- Active Computational Science Community Effort

Software for Very Large Data Collections

- Multi-petabyte distributed data collections
  - pathology, radiology data, earth 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

Data Handling Integration

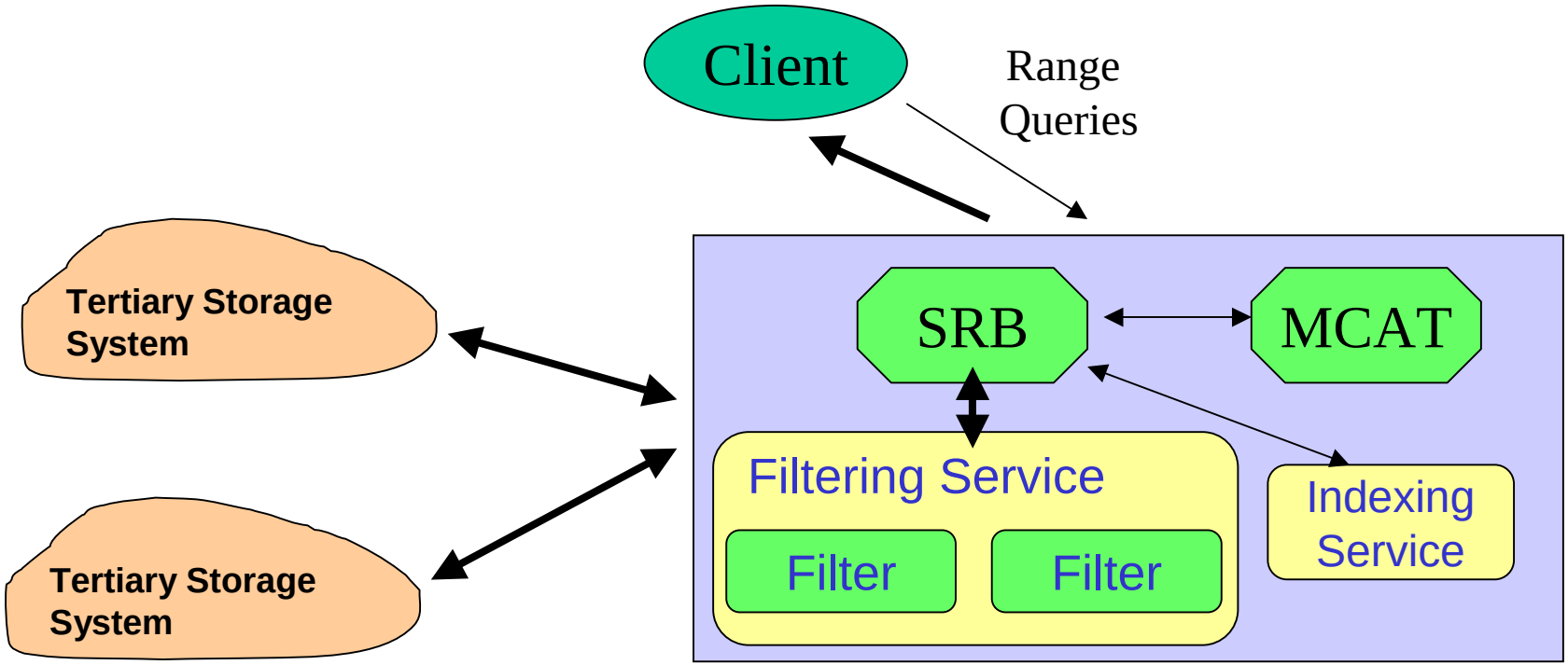
National Partnership for Advanced Computational Infrastructure (NPACI)



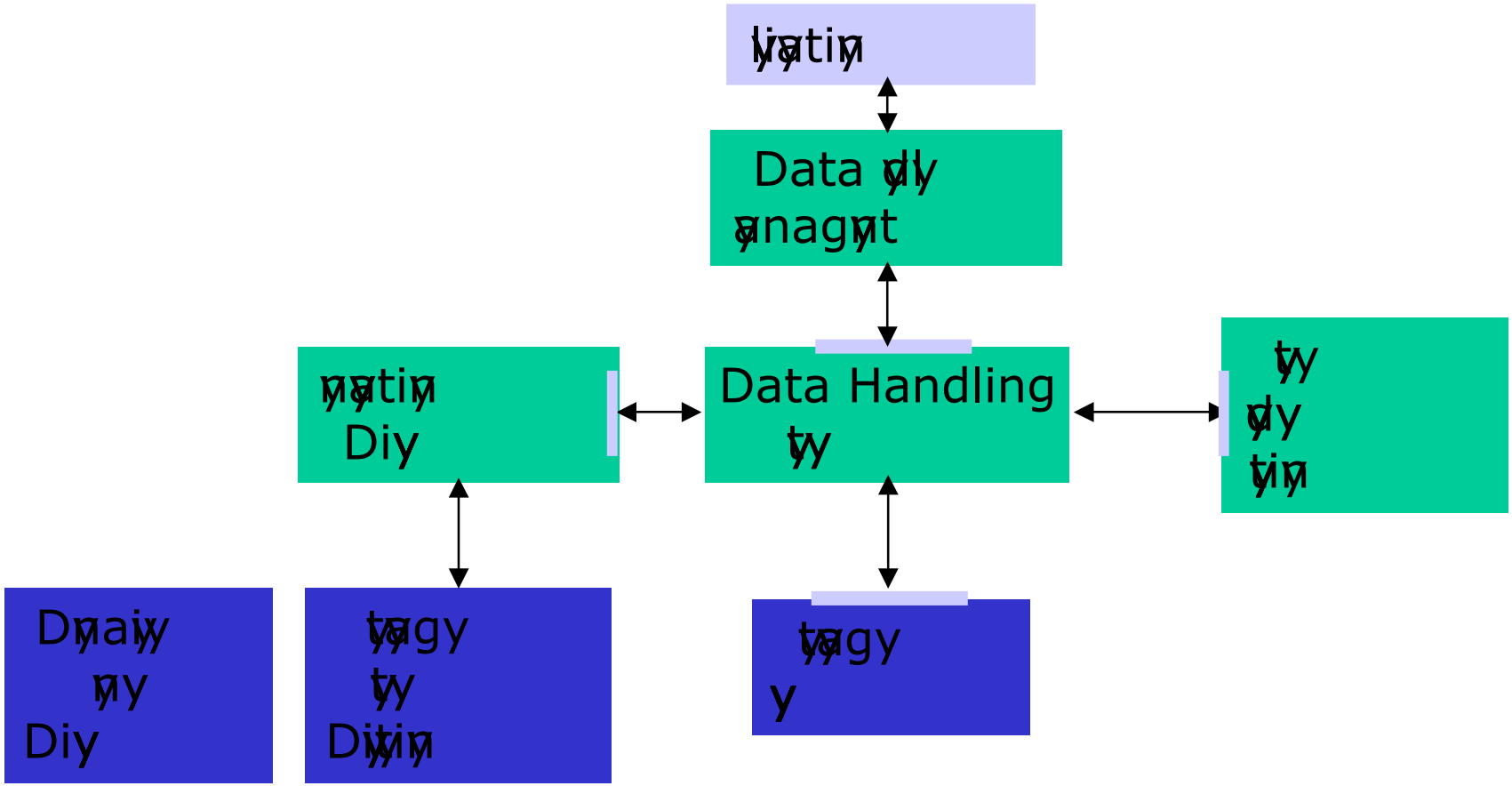
Grid Issues

- Management of deep, distributed storage hierarchy
- Security, authentication
- Finding information
  - datasets, objects, portions of objects
- Remote execution of programs
- Invocation of multiple remote services

Storage Resource Broker/DataCutter - Prototype Implementation



Open Grid Architecture to Encourage Interoperability



Current NPACI Data Collections

Collection Site	Project	Archive (TB)	Database (GB)
Caltech	Digital Sky images / NPACI-DICE	2-20	2
SDSC	CEED / ESA	1	2
SDSC	PDB	0.5	2
SDSC	NARA – USPTO patents	0.3	70
SDSC	Human Brain Project / NPACI-Alpha NS	1	10
SDSC	Molecular Structures / NPACI-Alpha MS	1	10
SDSC	Visualization image collection	0.5	5
SDSC	SRB Production system / NPACI-DICE		75
UCB	Elib flora collection / NPACI-DICE	1	60
UCLA	Human Brain Project / NPACI-NS	1	2
UCOP	Art Museum Image Consortium / CDL	1.5	30
UCSB	Alexander Digital Library / NPACI-DICE	2	2
UCSC	REINAS / NPACI-ESS	0.1	1
U Maryland	HPSS federation / NPACI-DICE	1	1
U Michigan	UMDL / NPACI-DICE	0.1	30
U Wisconsin	Digital Insight / NPACI-EOT	10-20	5
Washington U	Human Brain Project / NPACI-NS	1	10
UCSD	PRDLA - proposal	0.5	10
UCOP	CDL backup - proposal	5	10

New Types of Media

- Revolutionary change
  - No doubt very important in the long run, but with 70% annual price/performance improvement, evolution is working well
- Evolutionary change
  - Active Disks -- coming within next year
    - Each disk comes with a processor
    - Execute algorithms on configuration of processors and disks
    - In some cases, leads to dramatic drop in I/O requirements, performance increases
    - Mustafa Uysal - recent PhD thesis

Motivation: Active Disks

- Opportunity: technology inflection point
  - Powerful, inexpensive embedded processors
  - DRAM: capacity, MB/$, bandwidth
- Proposal: Active Disks
  - Processor and memory integrated into disk
  - Execute **application code** in disk drives

Why Active Disks?

- Processing scales with increasing datasets
  - New processor is added with every disk
- Better Price/Performance
  - Reduce data movement through I/O interconnect
  - Processing offloaded to embedded disk processors
  - Share cabinetry, cooling fans, power supplies
- Processing evolves as disk drives evolve
  - Faster drives, faster processors, more memory

Streams

- Webster's Dictionary:
  - stream, n: [2b.] a constantly renewed supply
- Streams deliver data
  - data delivery as long as supplies last
  - to/from: disklets, disk media and host-resident code
- Why stream-based?
  - Large datasets: processing as data moves
  - More info revealed to OS for optimization
  - Natural for pipelining

Stream-based Programming Model

- Host-resident and Disk-resident code
  - *Disklets* perform bulk of the processing
  - Host-resident code manages/coordinates disklets
  - All data access/communication via streams
- Coarse grain interactions with disklets
- Restricted execution environment on disk
  - Thin operating system layer (DiskOS)

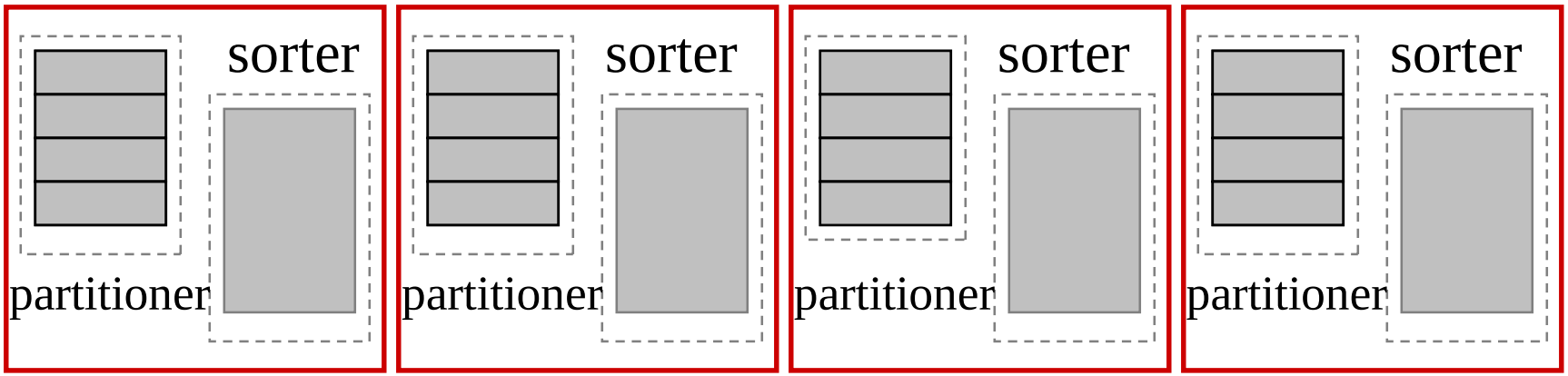
Stream Programming Model

- Webster's Dictionary:
  - stream, n: [2b.] a constantly renewed supply
- Streams deliver data
  - data delivery as long as supplies last
  - to/from: disklets, disk media and host-resident code
- Why stream-based?
  - Large datasets: processing as data moves
  - More info revealed to OS for optimization
  - Natural for pipelining

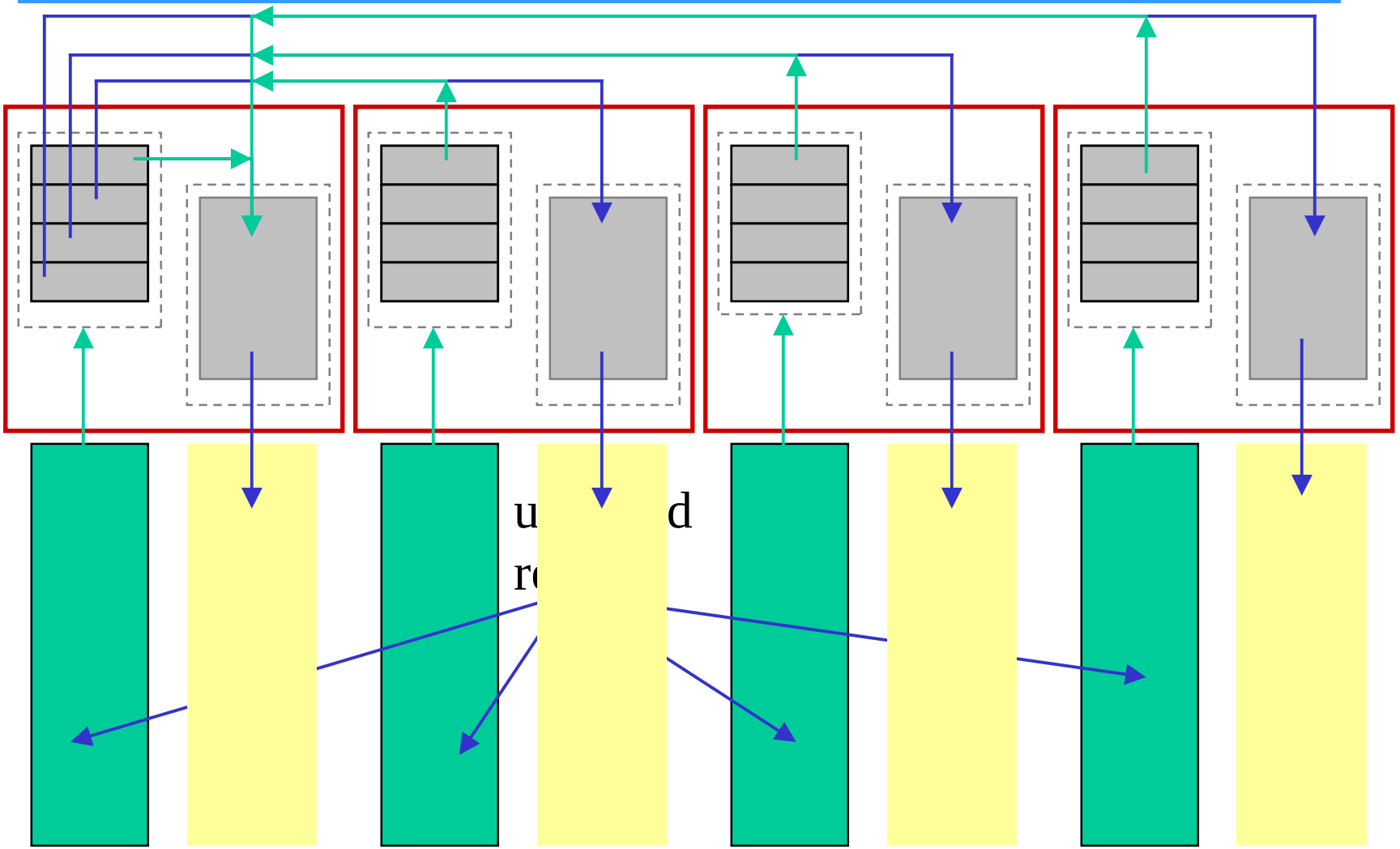
Active Disk Algorithms

- Distributed memory parallel algorithms
  - limited memory: careful staging/pipelining
  - restricted execution environment
- Goals:
  - leverage from existing algorithm base
  - limit code modifications

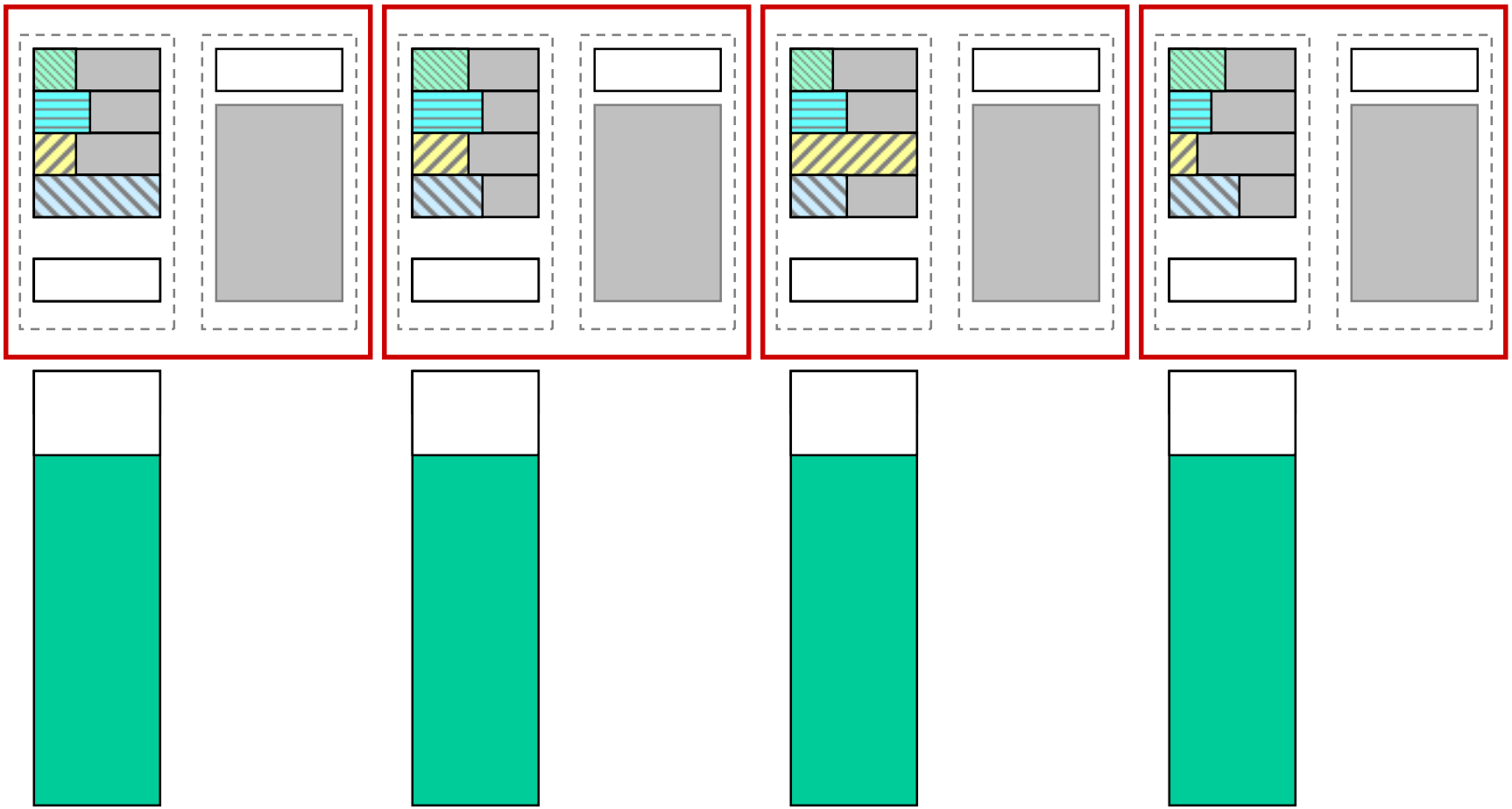
Example: External Sort



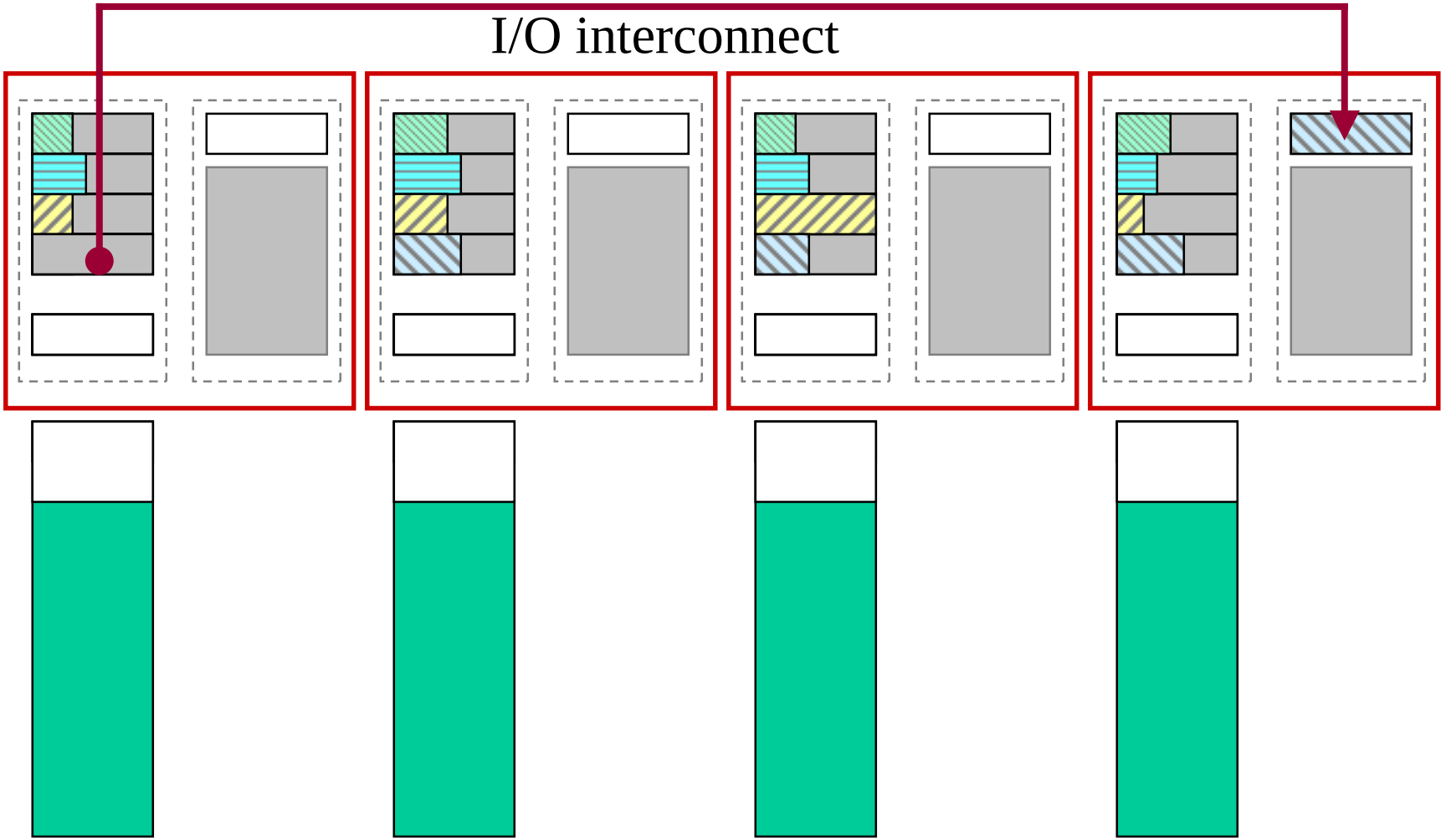
Example: External Sort



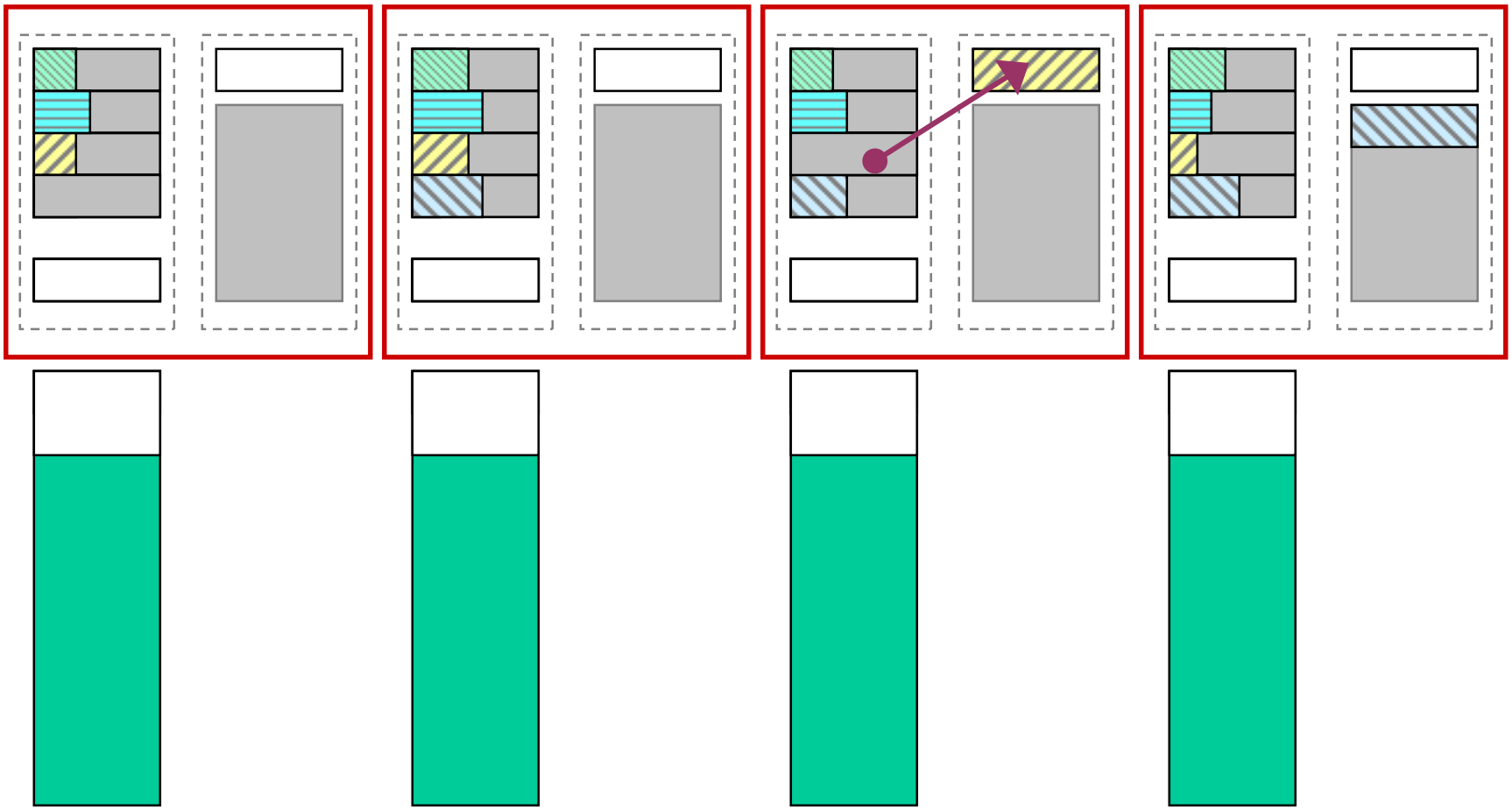
Example: External Sort



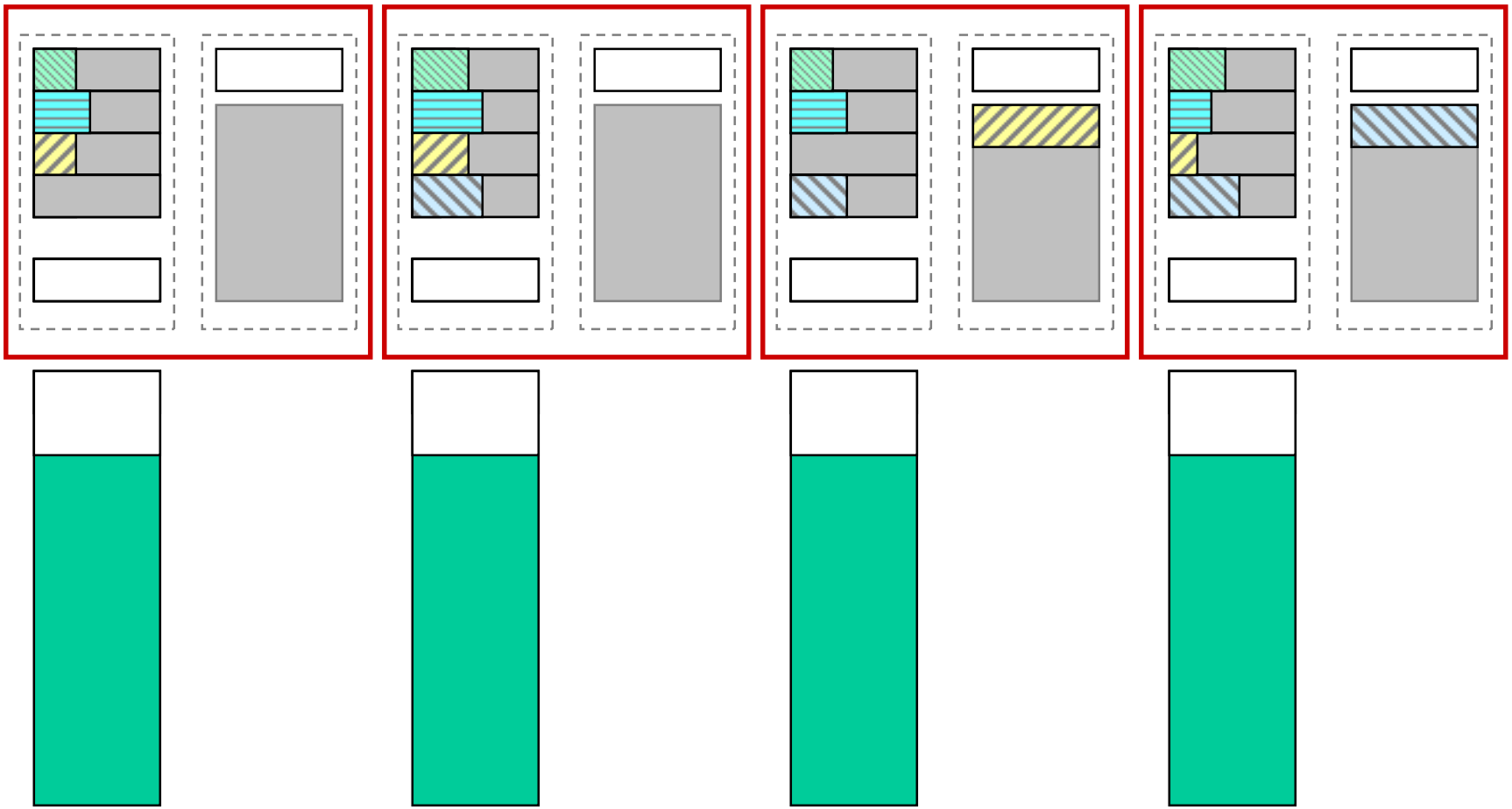
Example: External Sort



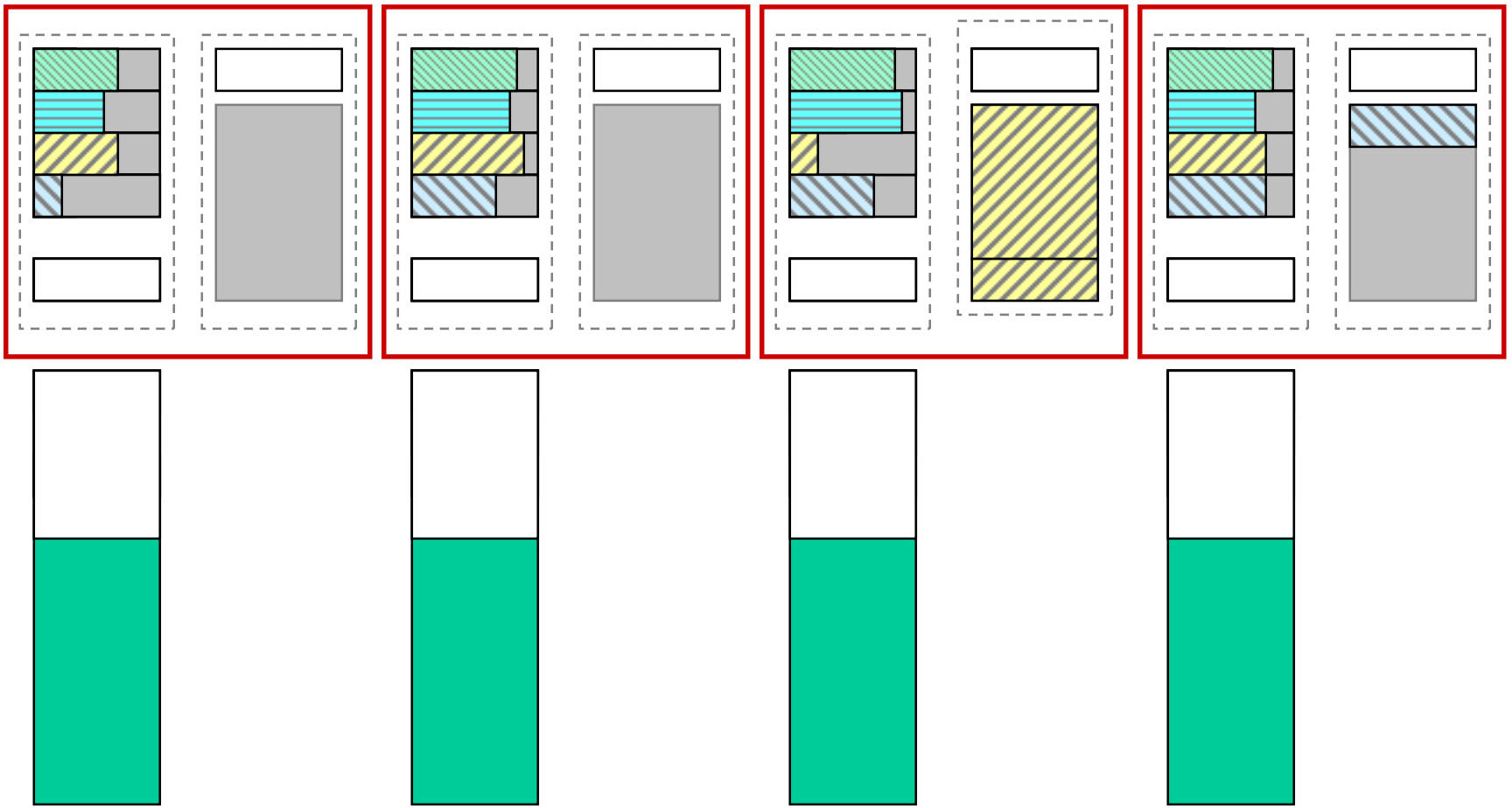
Example: External Sort



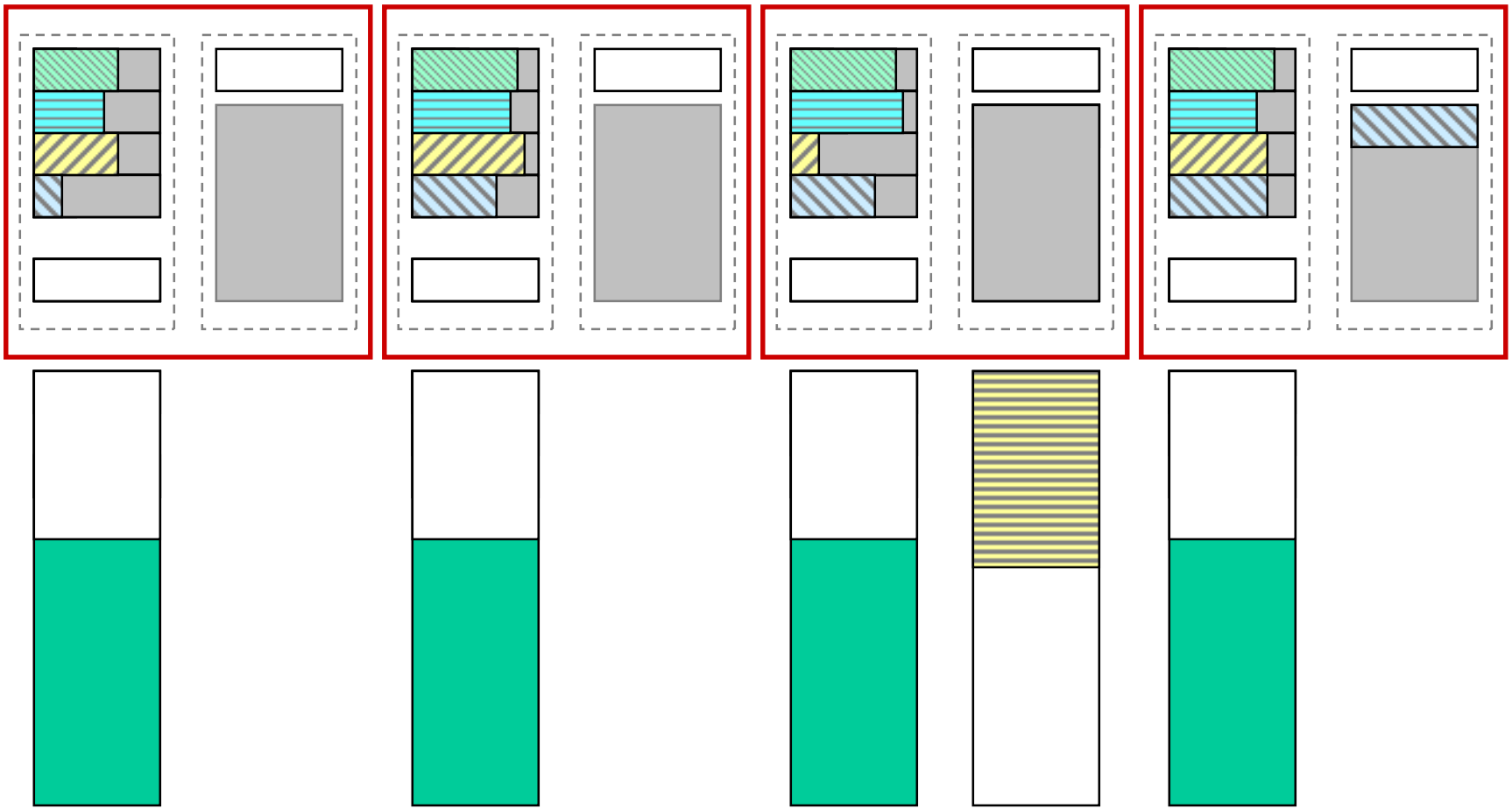
Example: External Sort



Example: External Sort



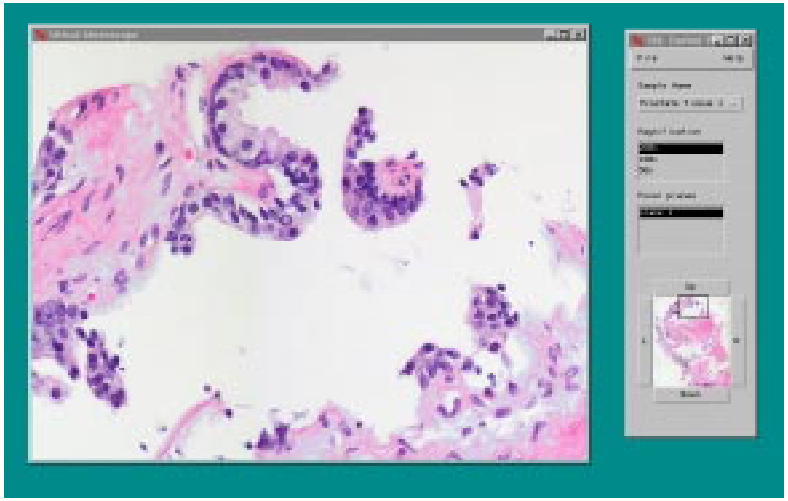
Example: External Sort



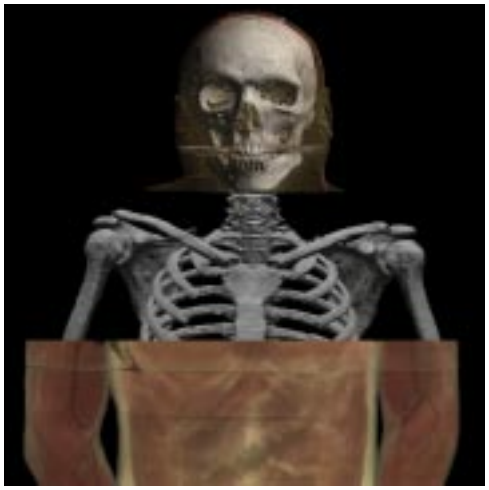
Characterization of Algorithms

Algorithm	***Data reduction***	***Computation/byte***
Select/aggregate	large	Small (19 / 6)
Group-by [Graefe '93]	large	Small (16)
Datacube [SIGMOD '97]	large	Small (23)
Sort (1st pass) [SIGMOD '97]	very small	Moderate (40)
Sort (2nd pass)	total	Small (16)
Join (1^{st}+2^{nd} pass) [Graefe '93]	very small	Moderate (51)
Join (3rd pass)	total	Moderate (96)
Dmine [SPAA '97]	large	Moderate-Large (116)
Mview [SIGMOD '96, PDIS '96]	large	Moderate (49)
Earth [NASA Goddard]	moderate	Large (2009)
Convolution [Johns Hopkins]	none	Large (1041)

Application Scenarios

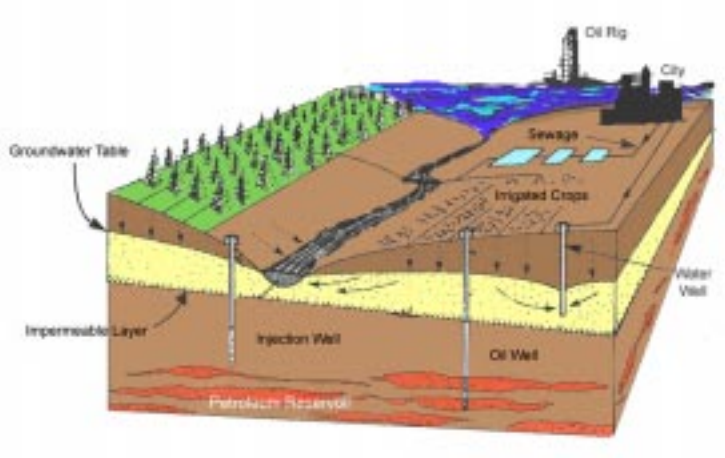


Pathology

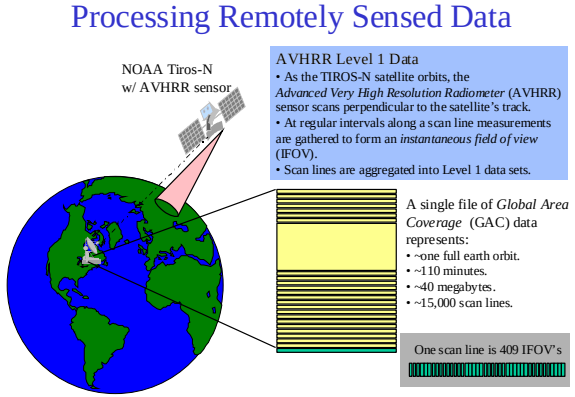


Volume Rendering

Applications



Surface/Groundwater Modeling



Satellite Data Analysis

Digital Pathology

- 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.

Data Requirements

- 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*

Anatomic Pathology

- Ideal:
  - User should be able to obtain any information that would be available were a slide physically present
  - Annotate portions of digitized slide used support diagnosis
  - Electronic capture of slide examination process used in resident training and CME

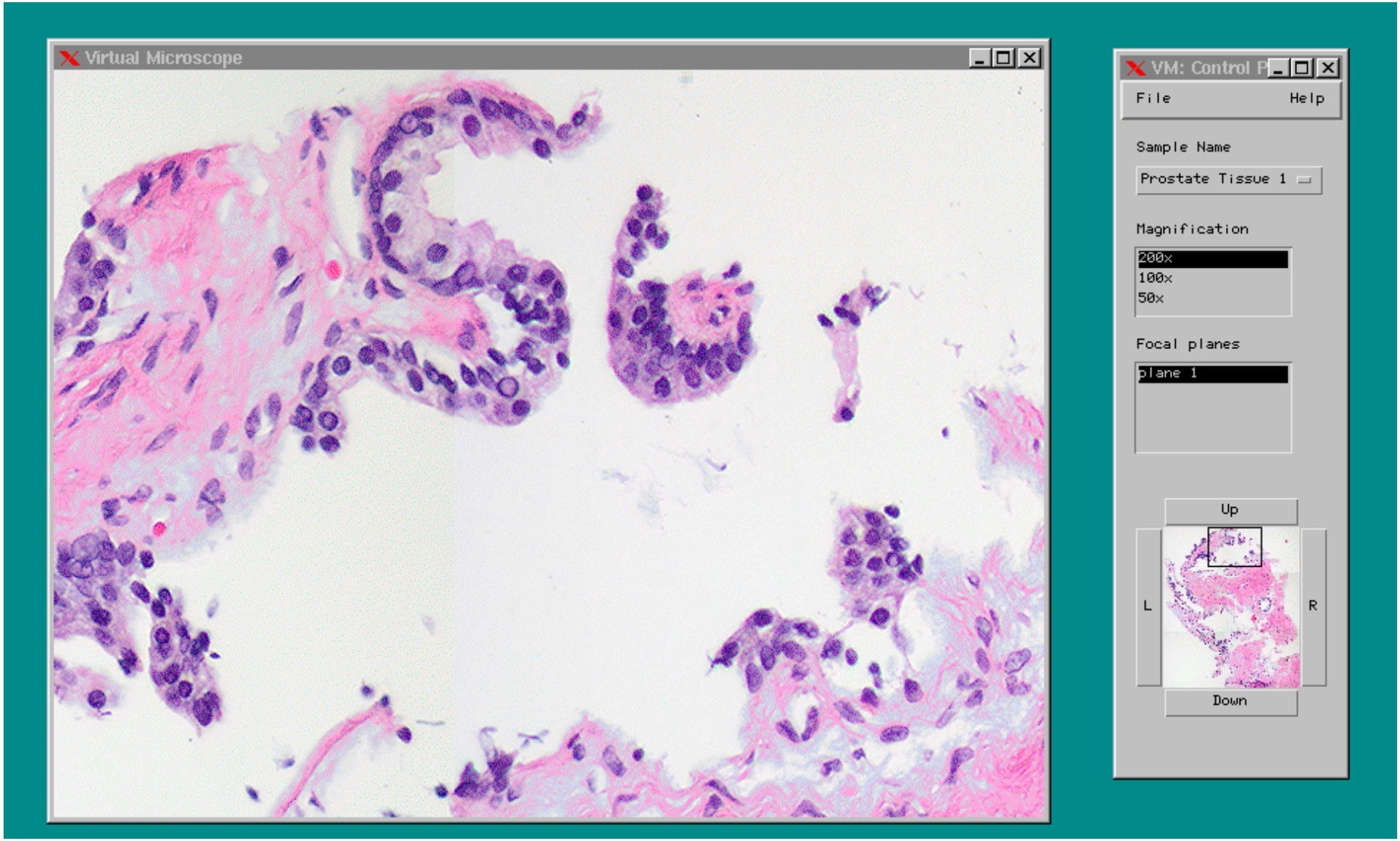
Computations Carried out on Datasets

- Screen for cancer
- Categorize images for associative retrieval
  - which images look like this unknown specimen (e.g. Content Based Image Retrieval)
- Visualize and explore dataset
- 3-D reconstruction

Virtual Microscope Application

- Interactive software emulation of high power light microscope for processing/visualizing image datasets
- 3-D Image Dataset (100MB to 5GB per focal plane)
- Client-server system organization
  - clients can access data over wide-area network
  - co-located clients (e.g., training environment)
- [Chen] Distributed Telemicroscopy System
  [Gu] Virtual Telemicroscope
  [Yagi] Virtual Microscopy
  Baccus Virtual Microscope

Virtual Microscope Client



Data Acquisition

- Can acquire data from microscope with robotic stage and digital camera
  - currently many hours to capture entire slide -BLISS Imaging Workstation
  - better slide scanning technology on its way
- Interim solution -- Video capture from microscopes used in signout
  - capture low power images to form map
    - use robotic stage
  - correlate high power images with low power map
  - archive all data or data near “interesting” findings

Multimedia Reports

- Capture, store, query and manipulate multimedia data associated with the practice of pathology.
  - Light microscopy, electrophoresis (PEP, IFE), blood smears, cytogenetics
- Multimedia medical record
  - Allow pathologists to generate a multimedia interpretive reports
    - free text
    - links to image data
    - links to laboratory data
    - menu driven report of findings (particularly for DI and Special Coag)

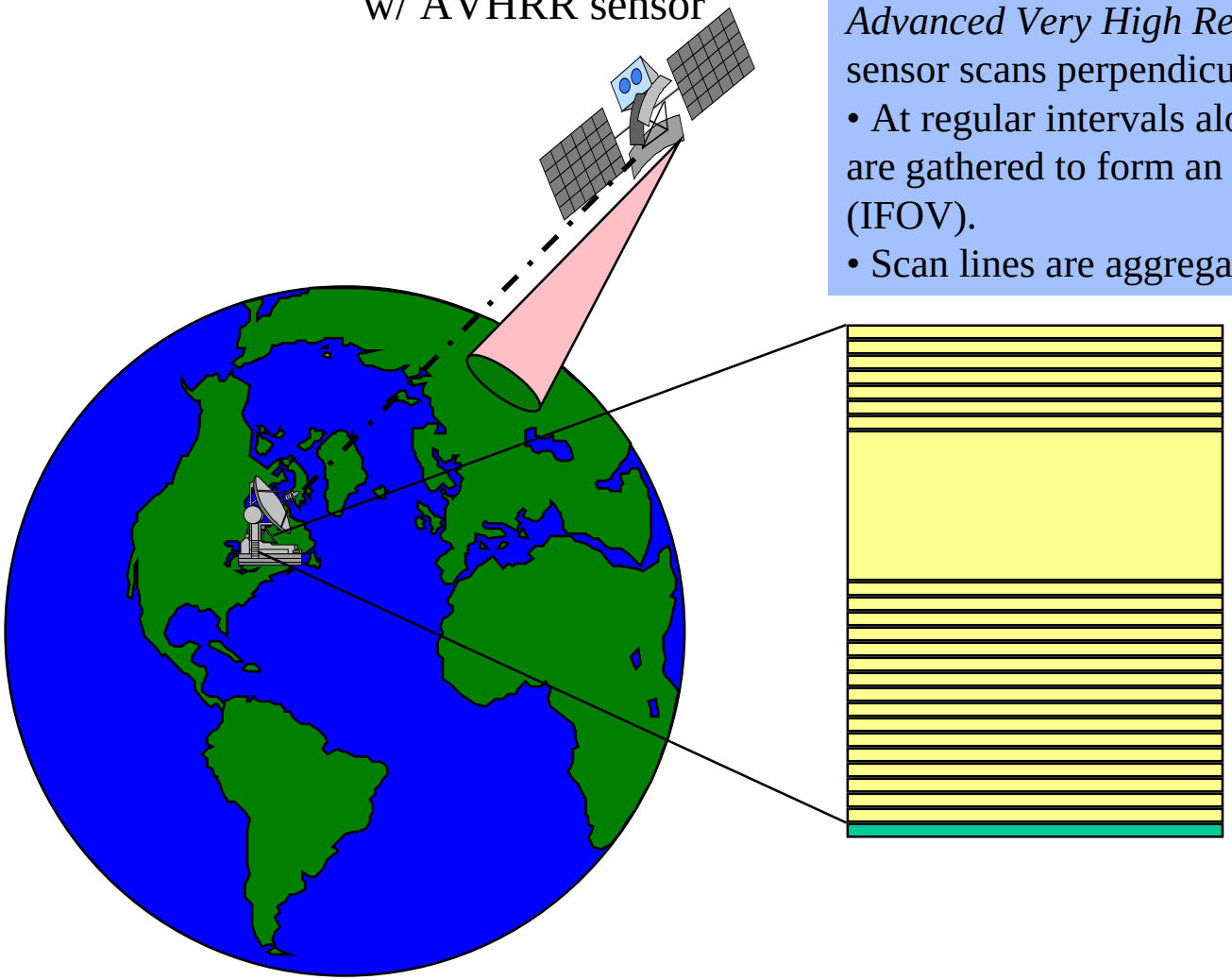
Electronic Support for Consultation

- Referring institution sends slides to consulting institution.
- Consulting institution captures all microscopy data examined during signout
  - generate report with links to portions of image data that support findings
  - user can either examine photomicrographs or can make use of a viewer to look at annotated electronic slide
  - Becich's Information Therapy!

Other Applications that Leverage the Same Infrastructure

Processing Remotely Sensed Data

NOAA Tiros-N
w/ AVHRR sensor



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.

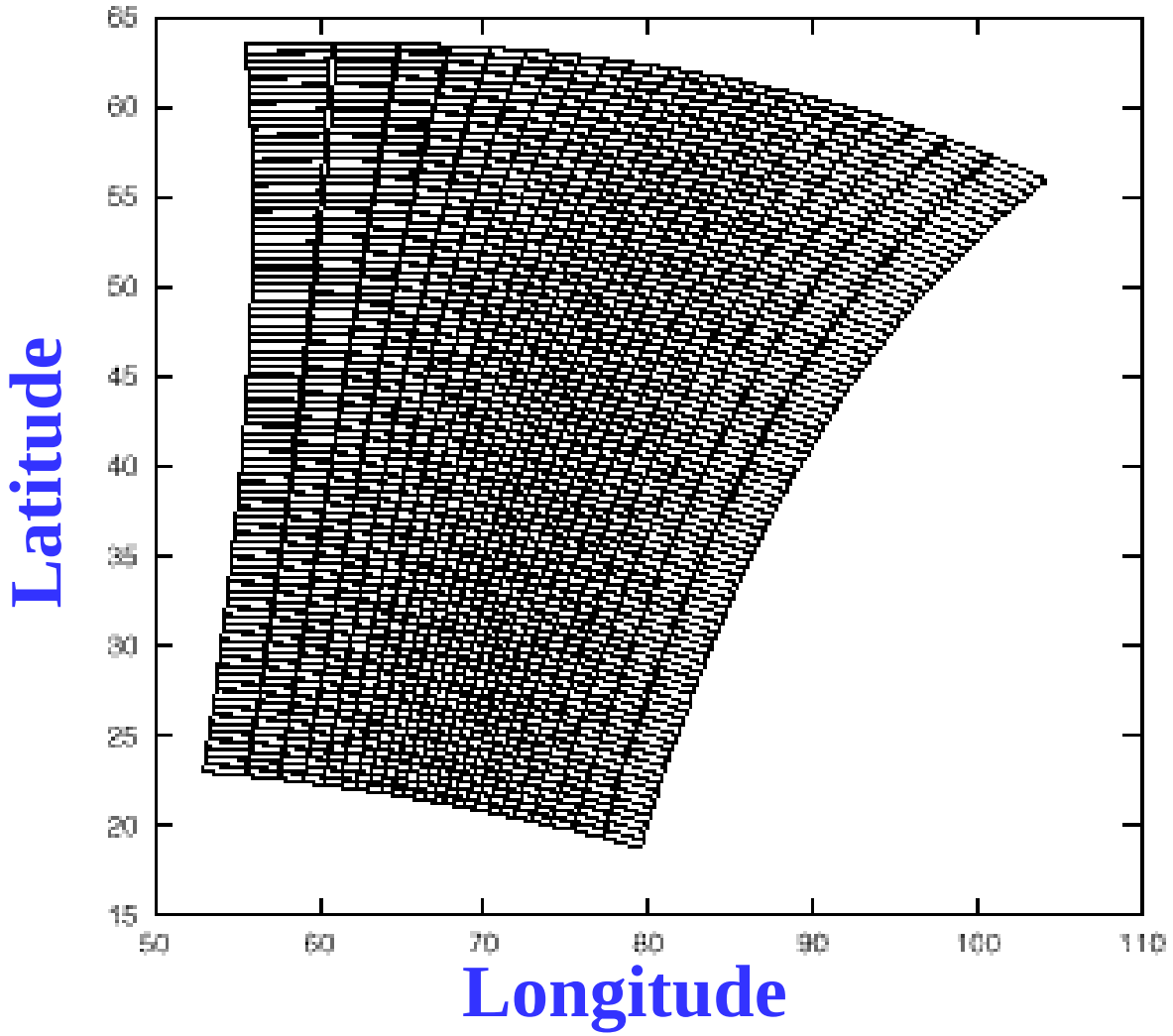
A single file of *Global Area Coverage* (GAC) data represents:

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

One scan line is 409 IFOV's

Spatial Irregularity

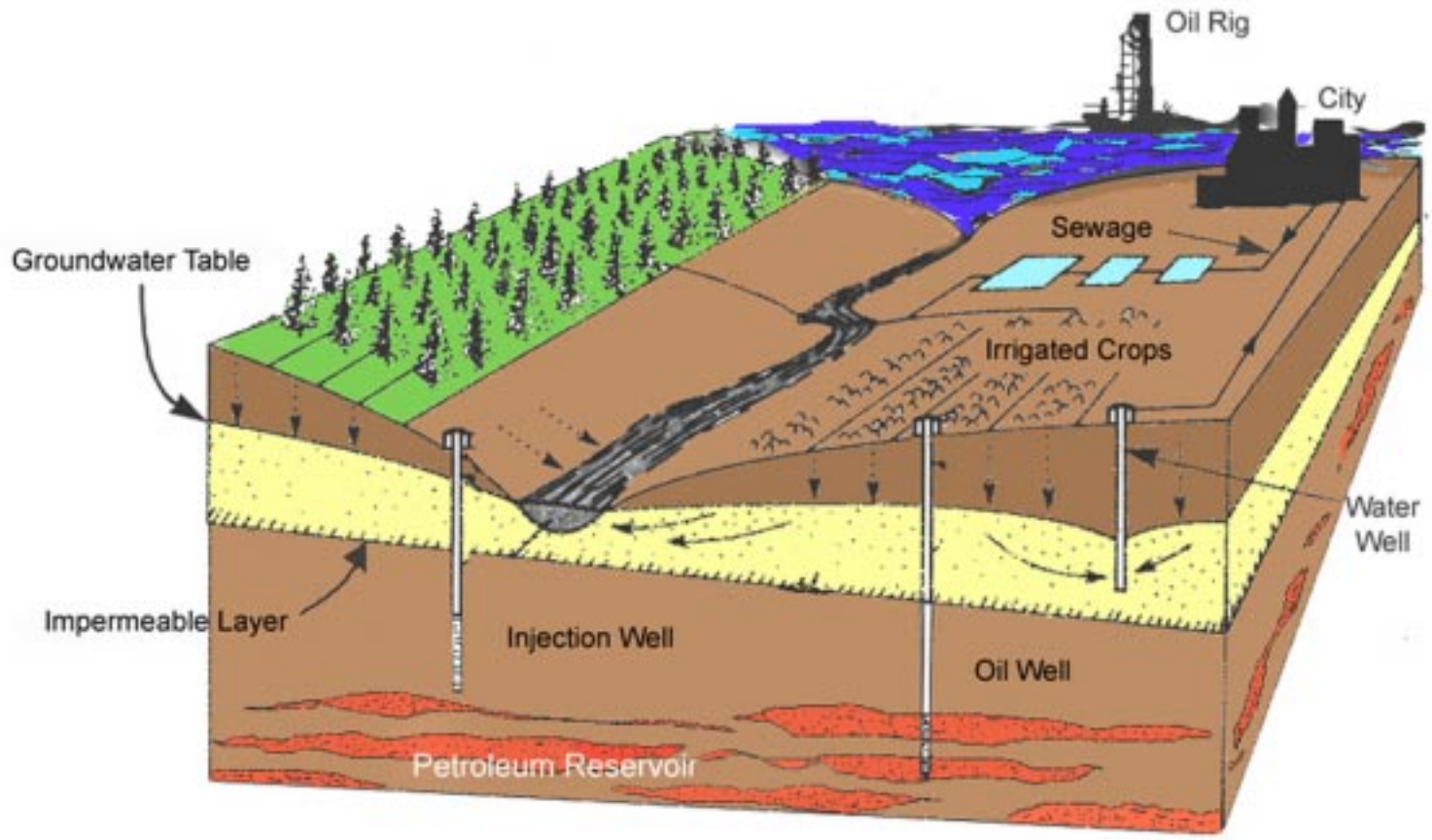
AVHRR Level 1B NOAA-7 Satellite 16x16 IFOV blocks.



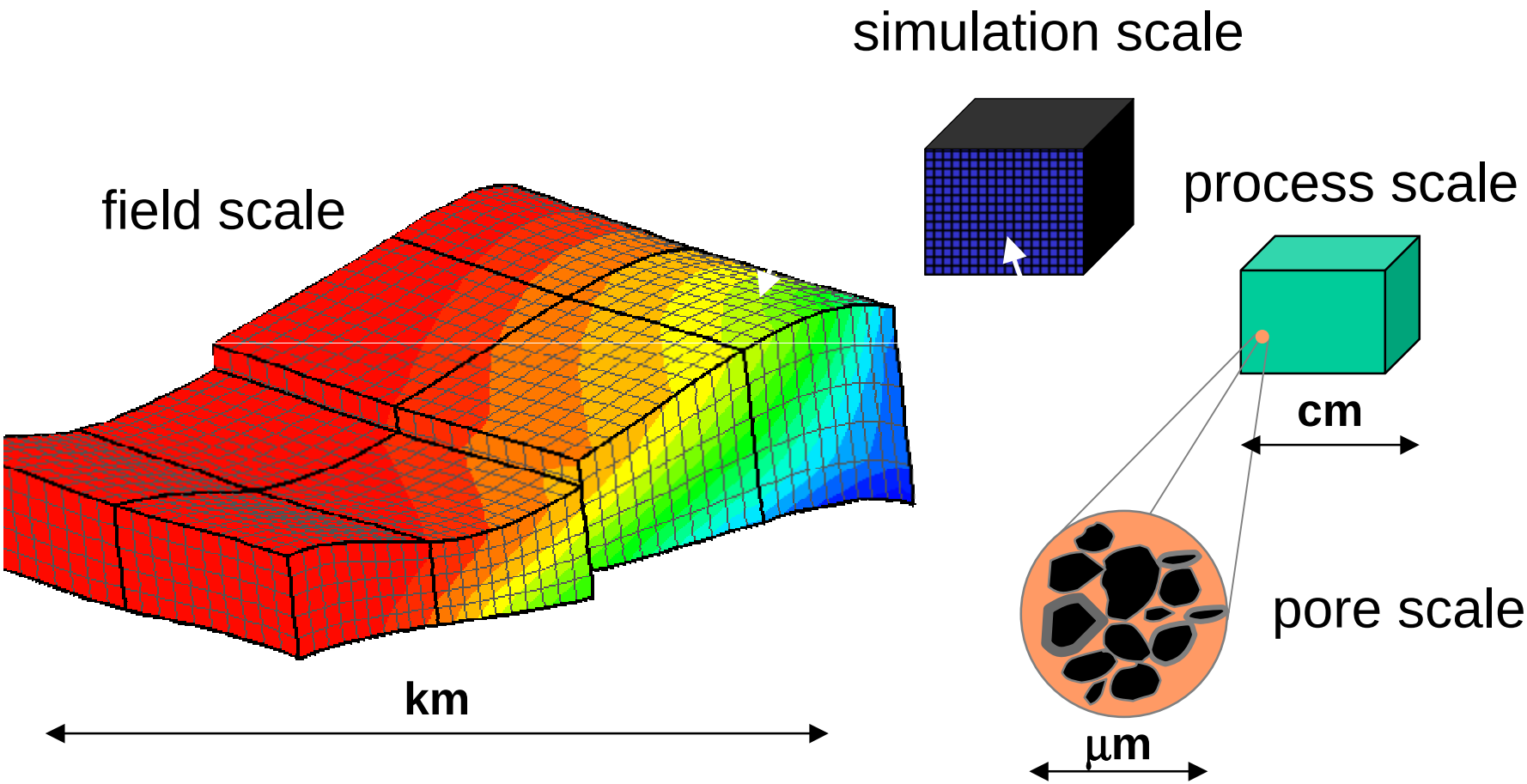
Processing

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

Coupled Ground Water and Surface Water Simulations



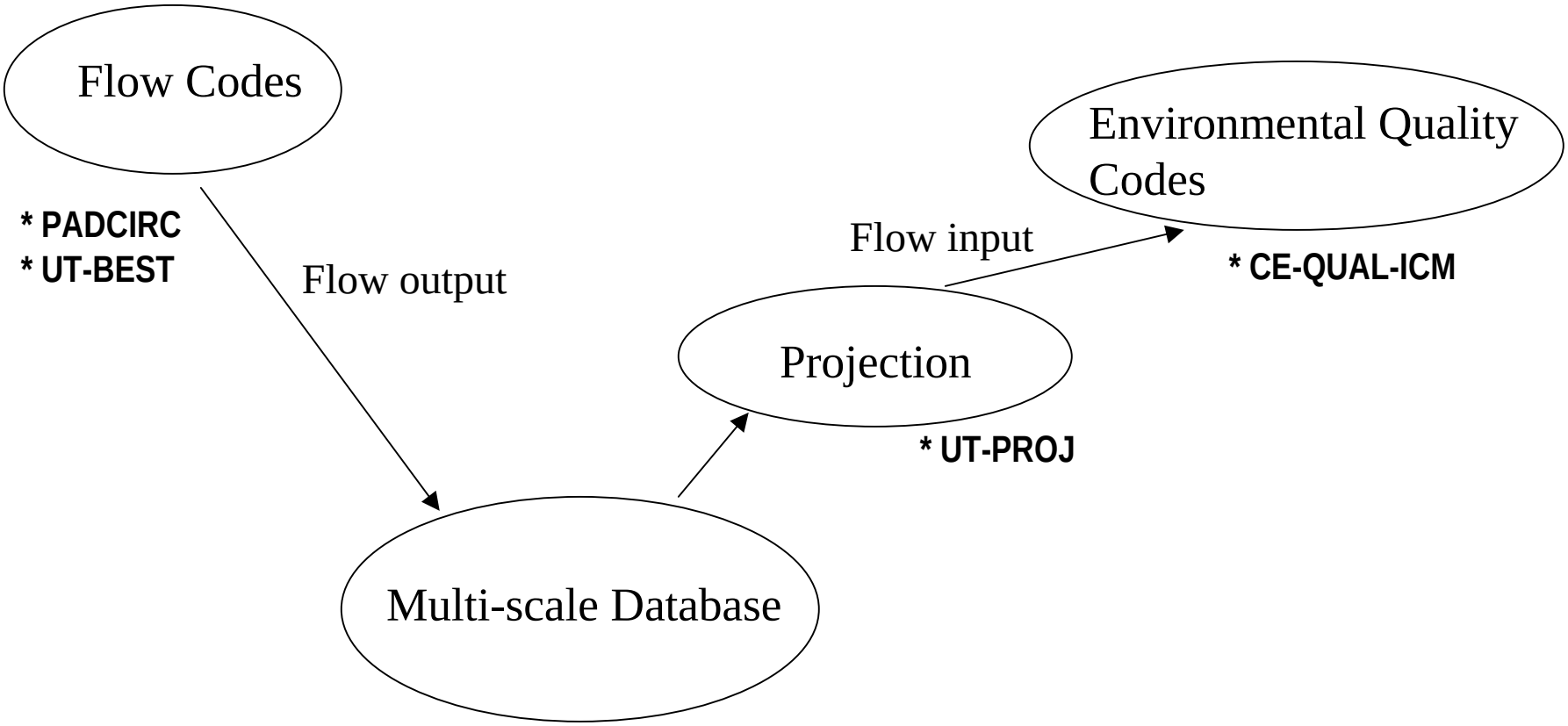
The Tyranny of Scale



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)



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

Attributes common to these applications

Multi-dimensional Data

- Has an underlying multi-dimensional attribute space
  - spatial/temporal coordinates
  - temperature, velocity, magnetic field, …
- Generated from sensors or simulation runs
- Accessed through range queries
- Output data size often significantly smaller than input data size
  - process data near where the data is stored
- Correctness of output usually does not depend on the order the input data items are aggregated
  - source of parallelism

Components of System Software Architecture

- Active Data Repository: On Demand Data Product Generation from Disk Caches
  - C++ toolkit targets SP, clusters
  - Compiler front end
    - Object Relational User defined functions
- DataCutter: Spatial Queries and filtering on distributed data collections
  - Spatial subset and filter
  - Load disk caches with subsets of huge multi-scale datasets

Application Processing Loop

O ← Output dataset, I ← Input dataset

$[S_I, S_O] \leftarrow$ *Intersect*(I, O, R_{query})

foreach o_e **in** S_O **do**

    $a_e \leftarrow$ *Initialize*(o_e)

foreach i_e **in** S_I **do**

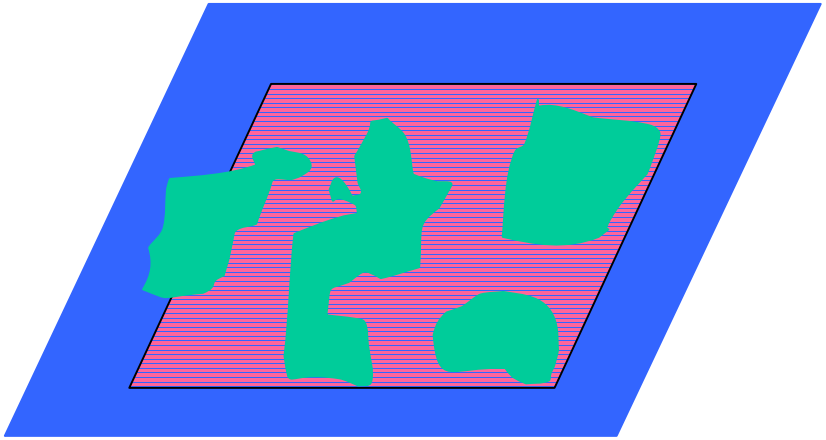
    $S_A \leftarrow$ *Map*(i_e) ∩ S_O

    foreach a_e **in** S_A **do**

        $a_e \leftarrow$ *Aggregate*(i_e, a_e)

foreach a_e **in** S_O **do**

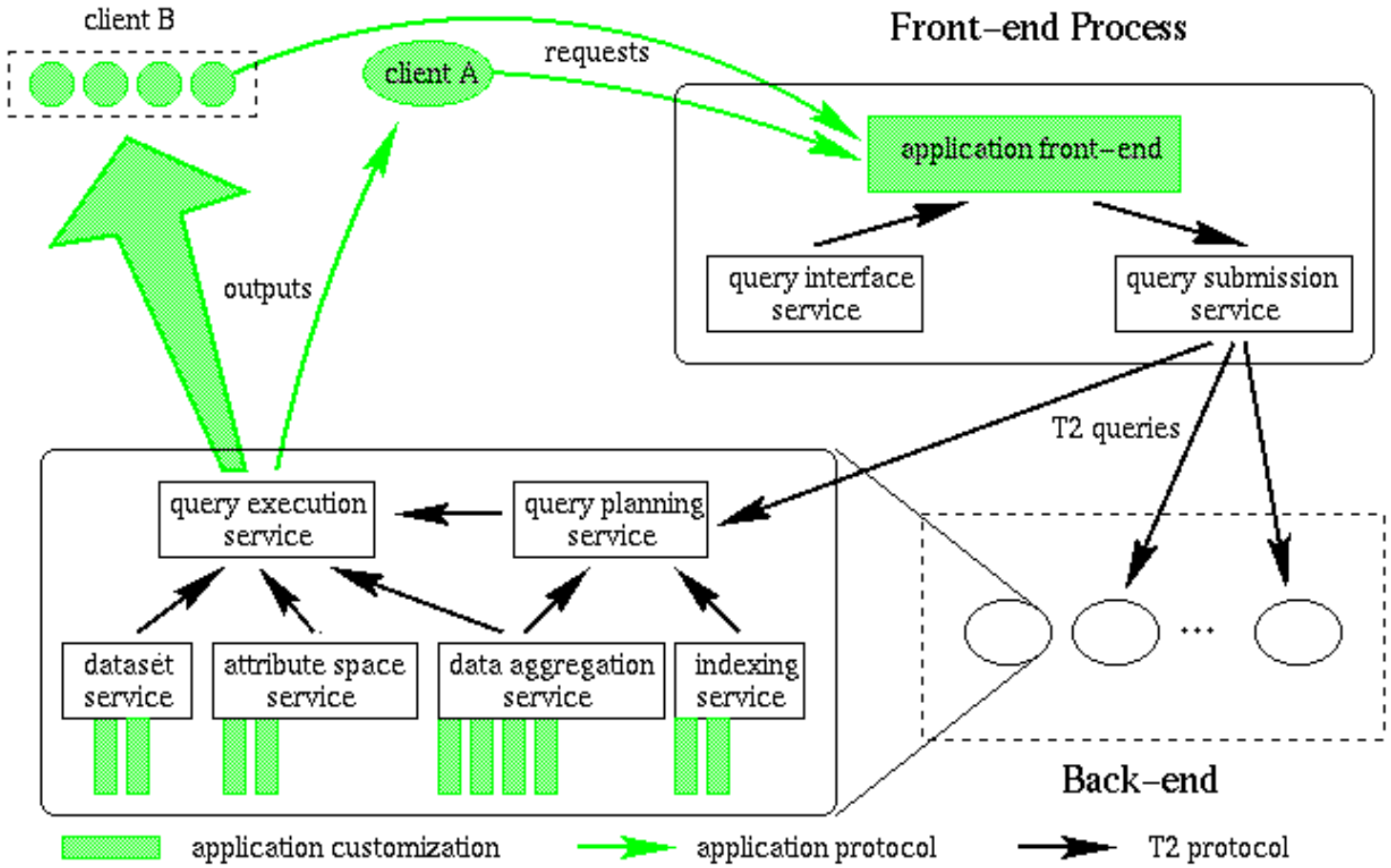
    $o_e \leftarrow$ *Output*(a_e)



ADR Design Objectives

- Support optimized associative access and processing of persistent multi-dimensional data
- Customizable for various applications
- Integrate and overlap a wide range of order-independent user-defined aggregation operations with data retrieval functions
- Target a shared-nothing architecture with disk farm

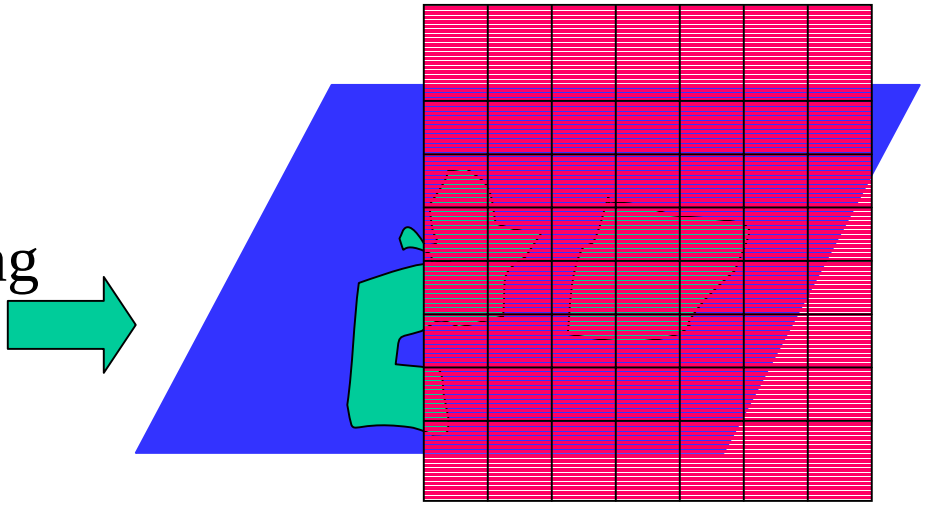
System Architecture



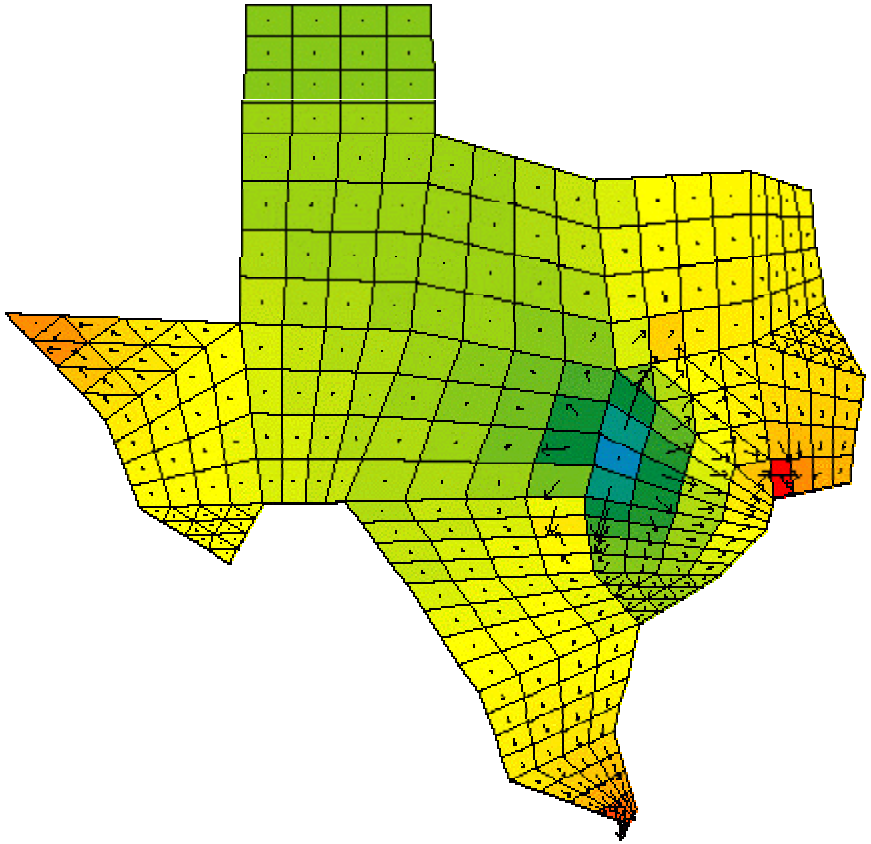
Typical Query

Output grid onto which a projection is carried out

Specify portion of raw sensor data corresponding to some search criterion



Implications for Dataset Structure

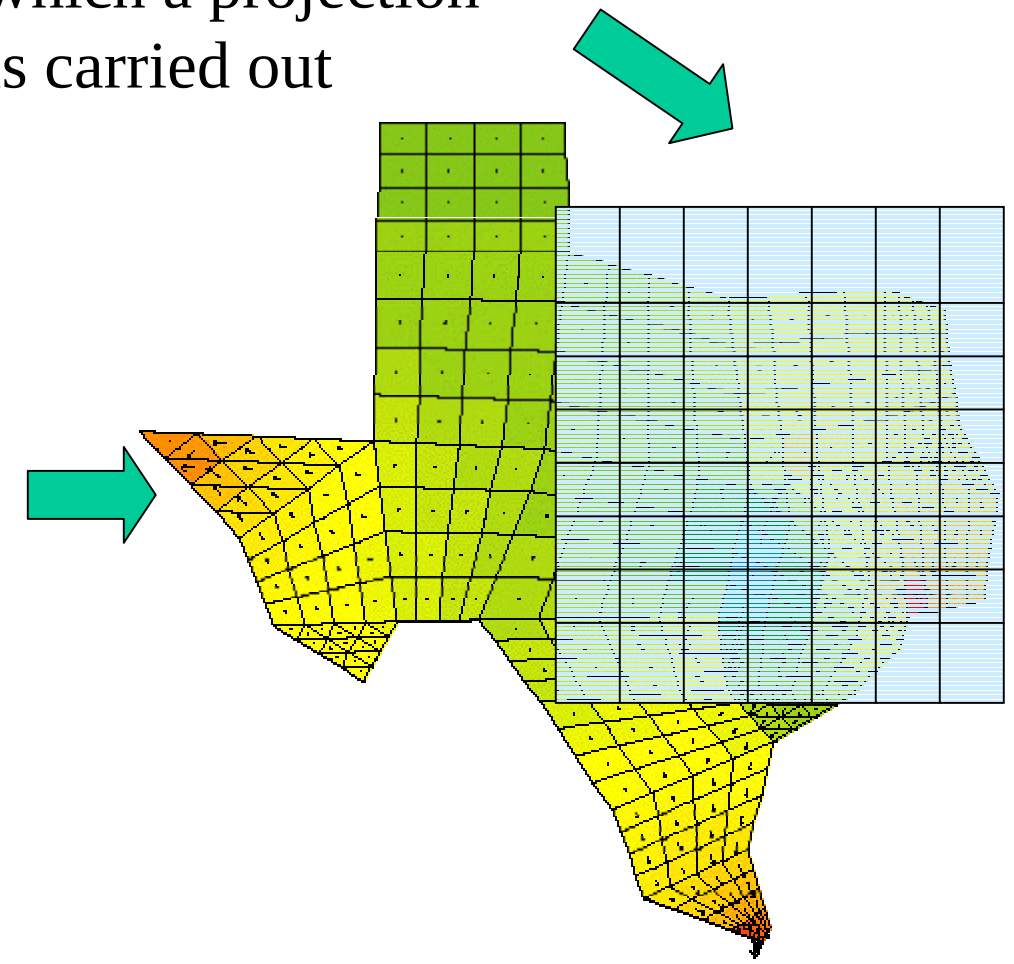


- Spatial and temporal resolution may depend on spatial location
- Physical quantities computed and stored vary with spatial location

Processing Irregular Datasets
Example -- Interpolation

Output grid onto which a projection is carried out

Specify portion of raw sensor data corresponding to some search criterion

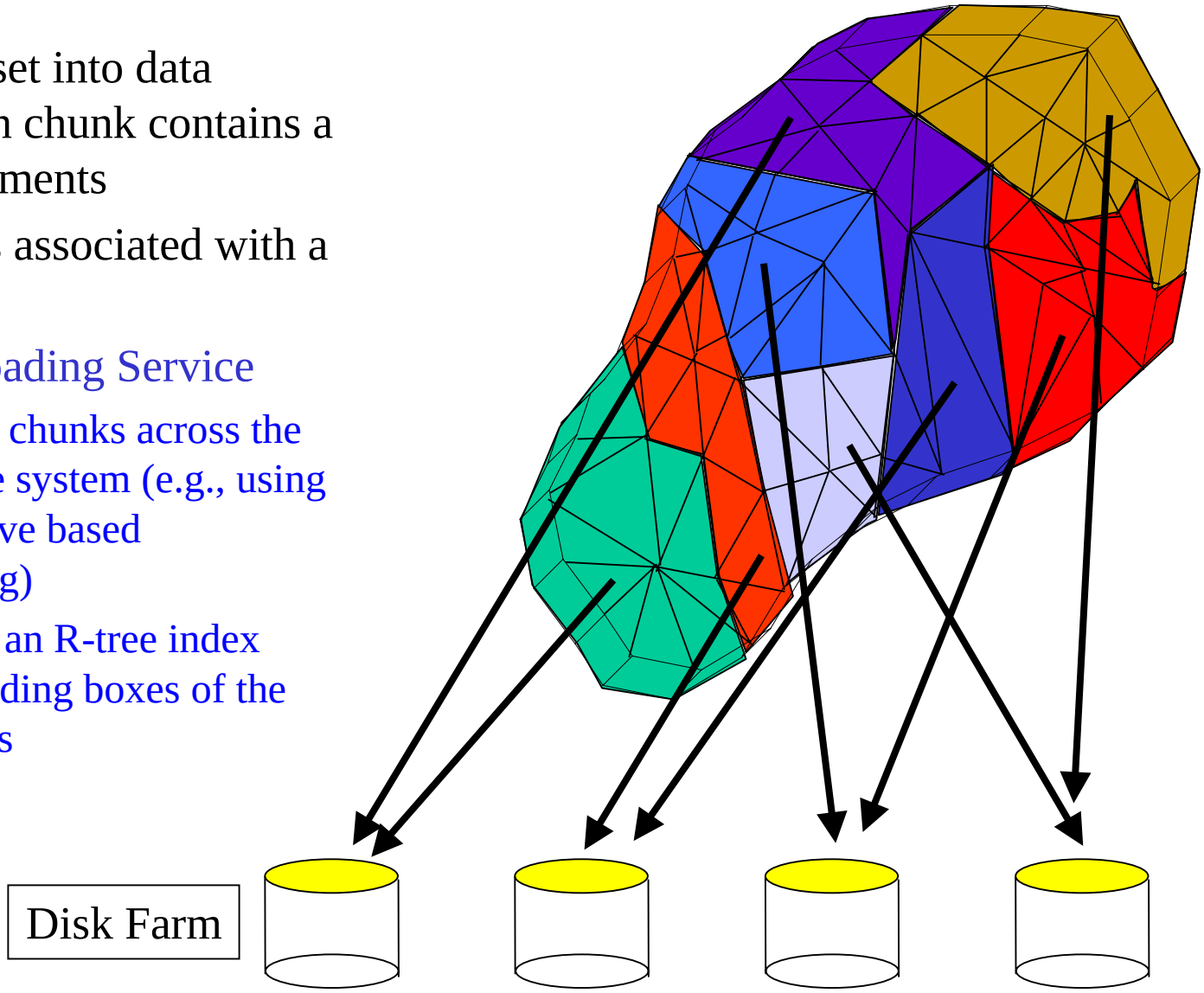


Executing Queries

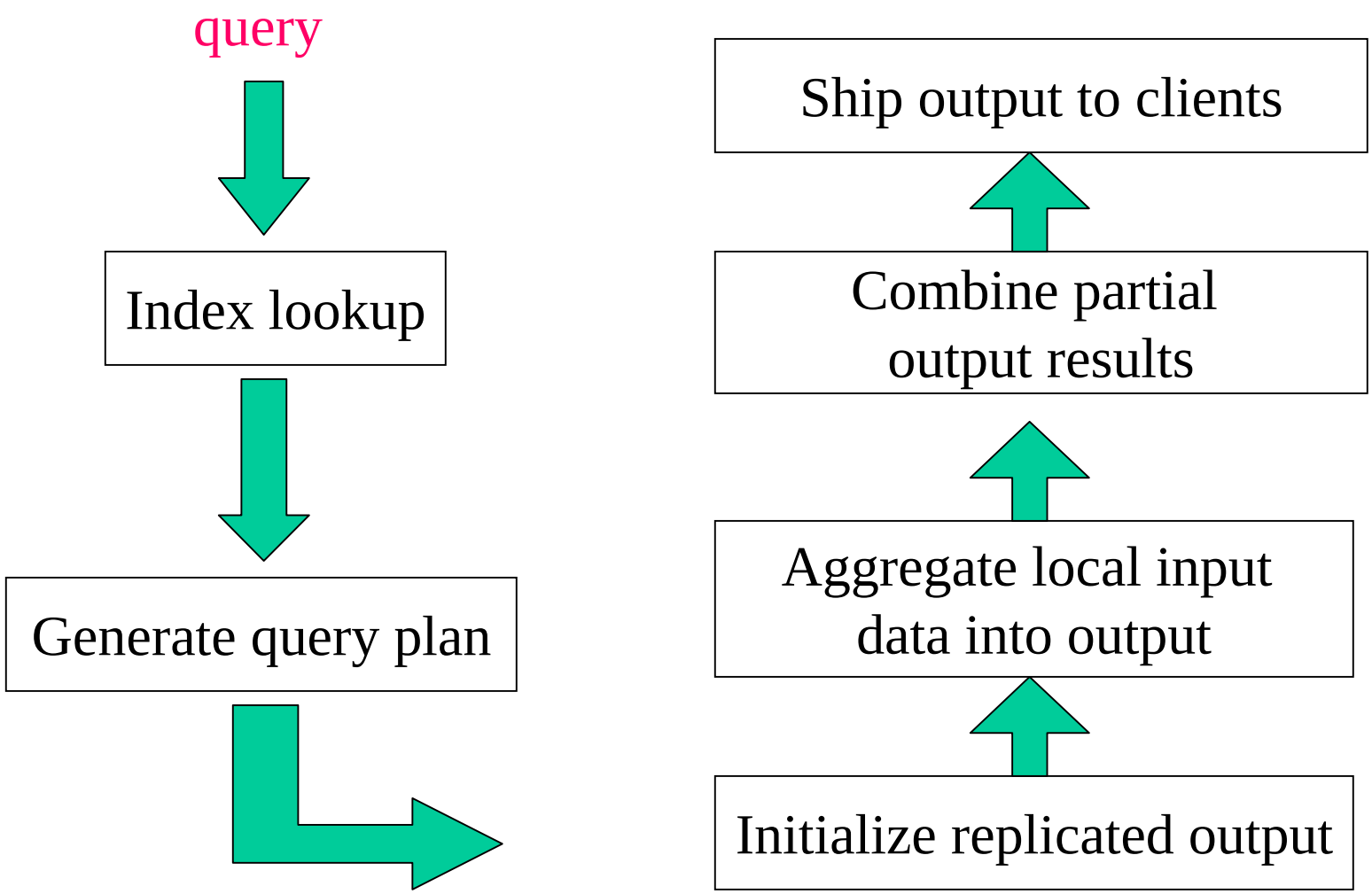
- Very large input, output datasets
- Clustered/declustered across storage units (Analysis of clustering, declustering algorithms -- PhD B. Moon)
- Datasets partitioned into “chunks”
  - Each chunk has associated minimum bounding rectangle
- Processing involves
  - spatial queries
  - user defined projection,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

Loading Datasets into ADR

- Partition dataset into data chunks -- each chunk contains a set of data 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



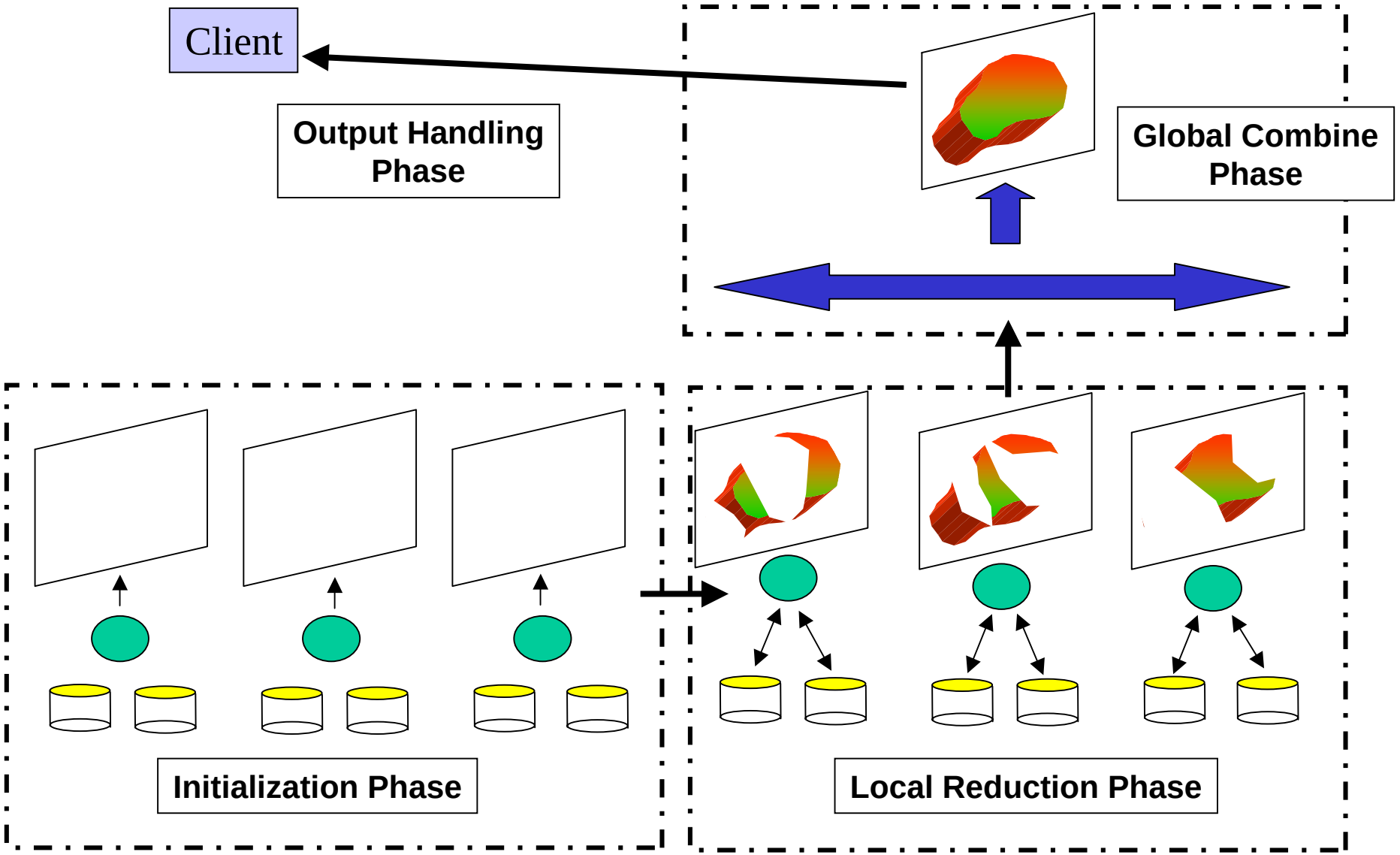
ADR Back-end Processing



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

ADR Back-end Processing



Query Planning Strategies

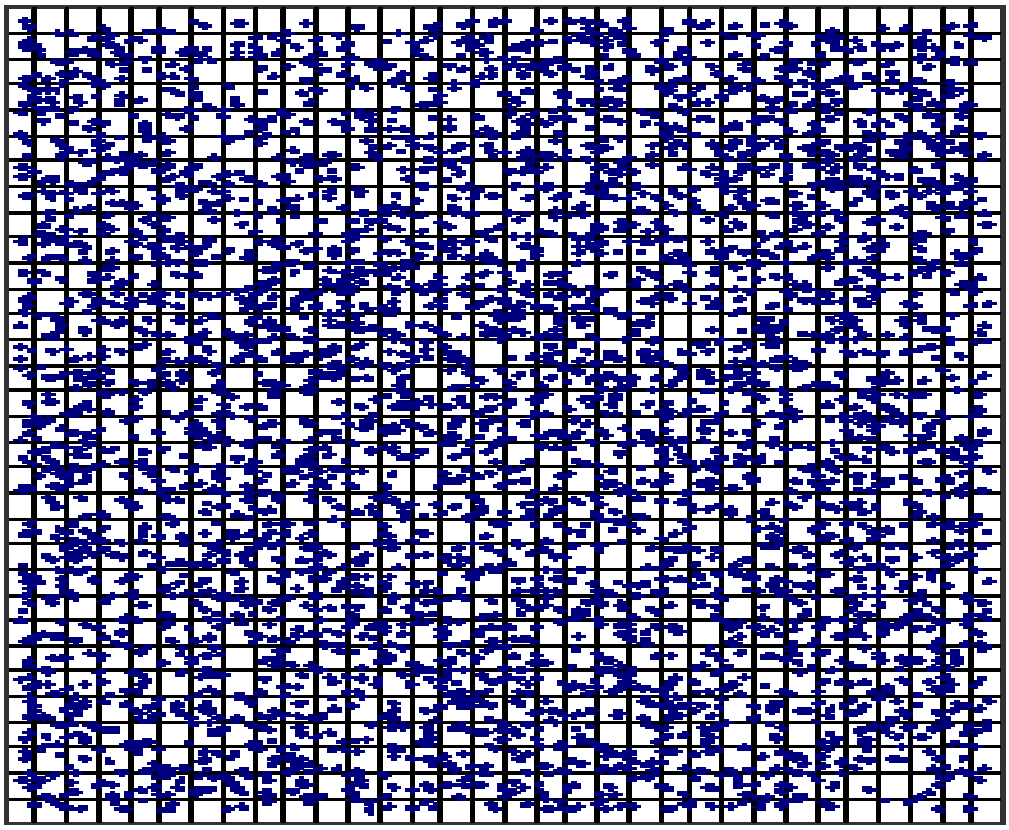
- Fully replicated accumulator strategy
  - Partition accumulator into tiles
  - Each tile is small enough to fit into single processor's memory
  - Accumulator tile is replicated across processors
  - Input chunks living on disk attached to processor P is accumulated into tile on P
  - Global combine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

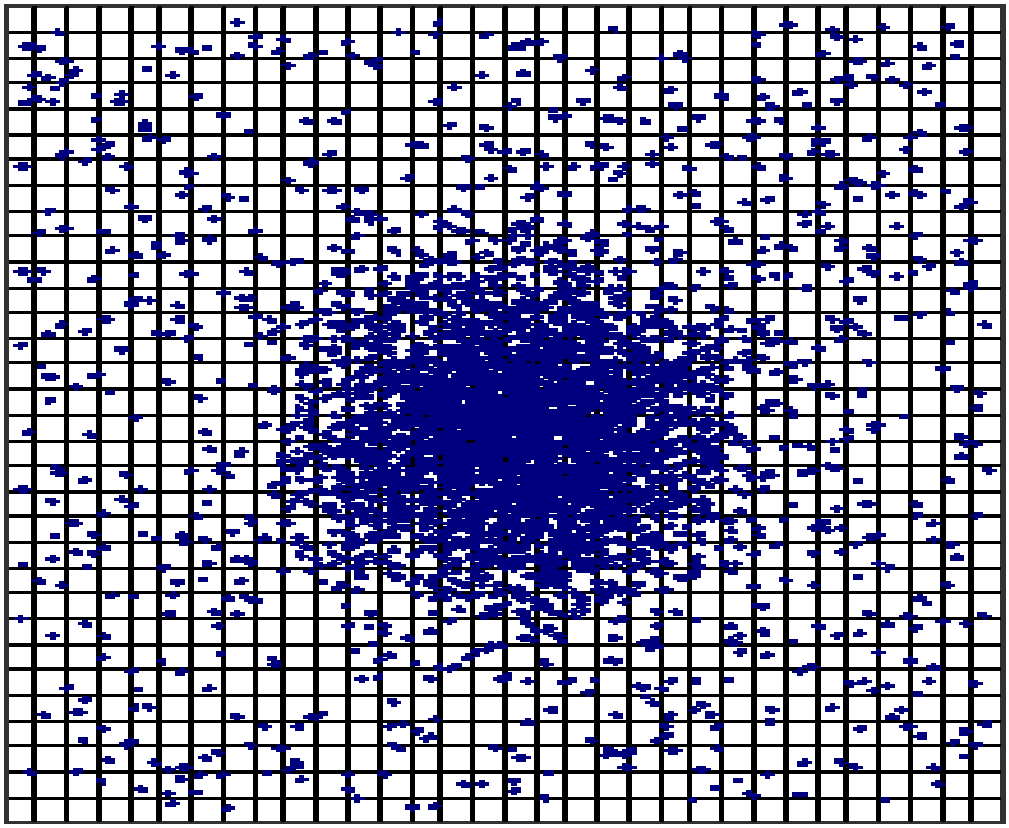
Query Planning Strategies

- 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

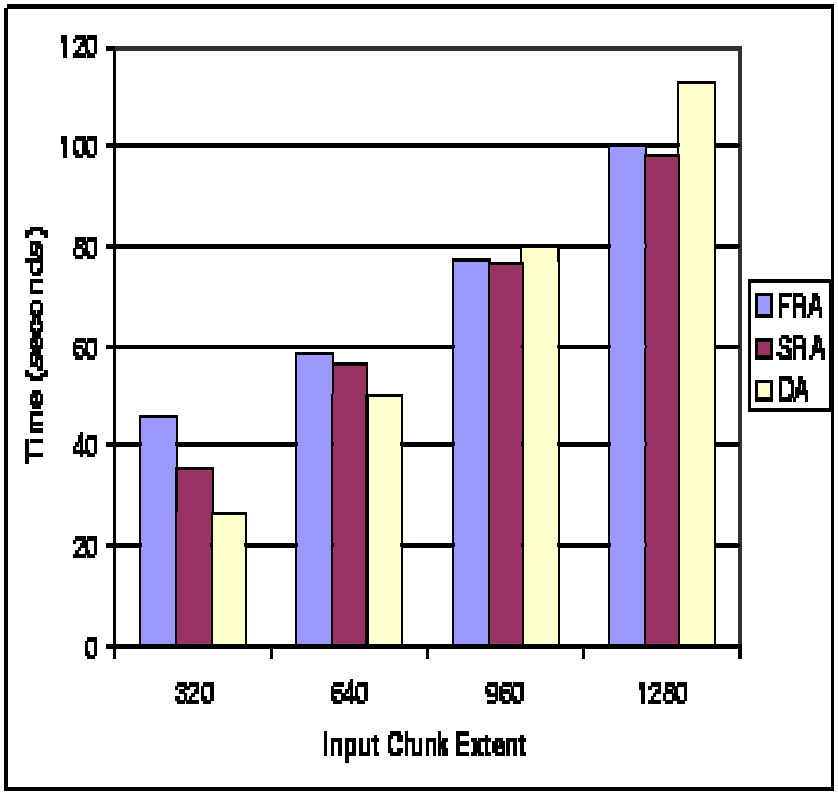
Studies to evaluate query processing strategies

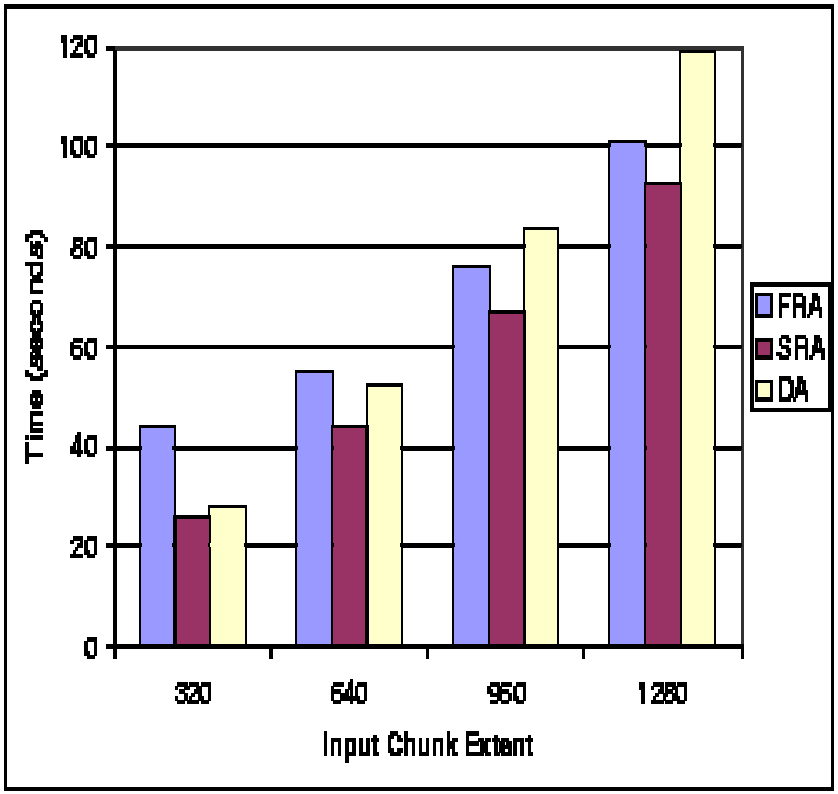
- Projection of 3-D datasets onto 2-D grid
- Query windows of various sizes directed at synthetic datasets with uniform, skewed data distributions
- Sparse replicated accumulator wins when there is a high degree of fan-in -- communication can be saved by local accumulation of multiple chunks
- Distributed 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

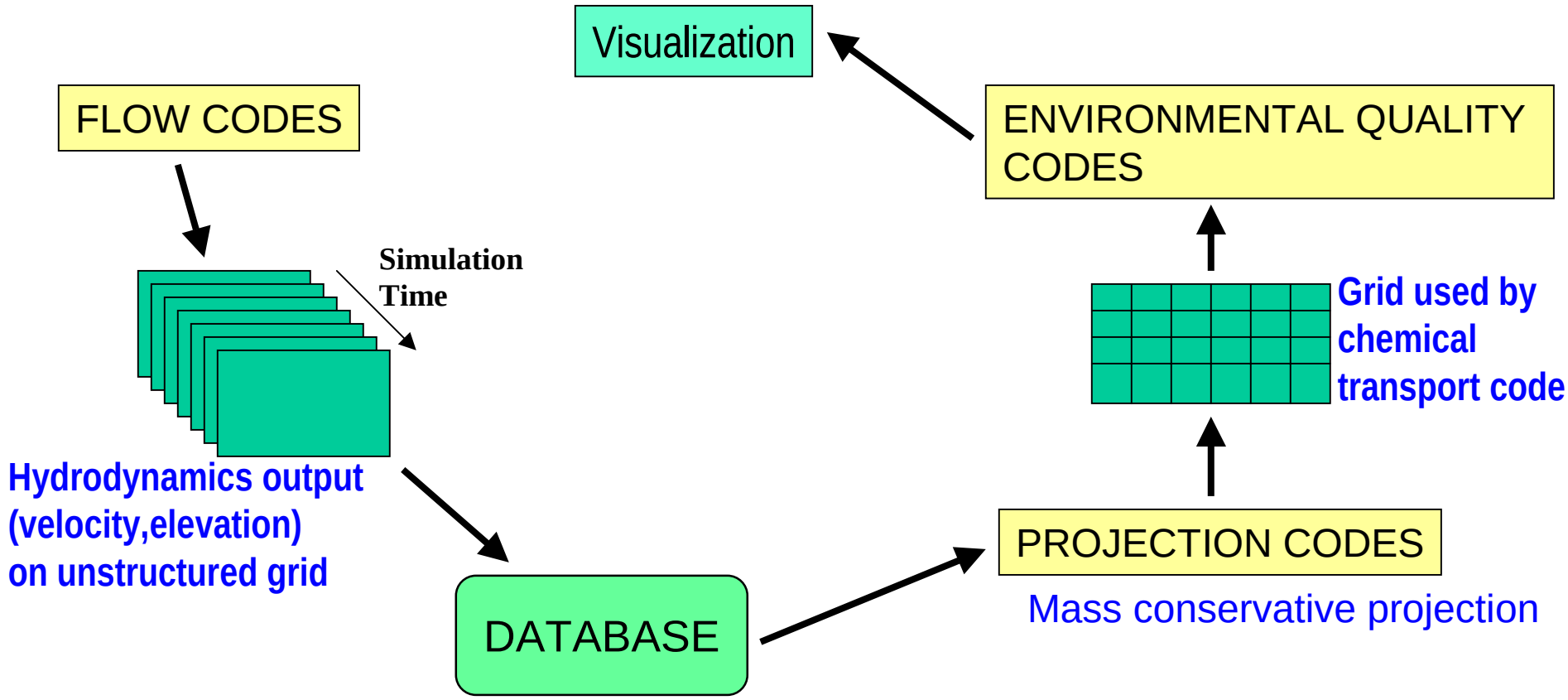




Current Status

- Core ADR services implemented
  - software hardening
- APIs for customization and among services defined
  - under revision for better flexibility and performance
- ADR customizations for applications
  - done for remote sensing (Titan), Virtual Microscope, bay/estuary simulation, large scale data visualization (ray casting), system analysis of CT data in material science
  - HUBS project will fund support of CBIR and other image classification algorithms

Water Contamination Studies

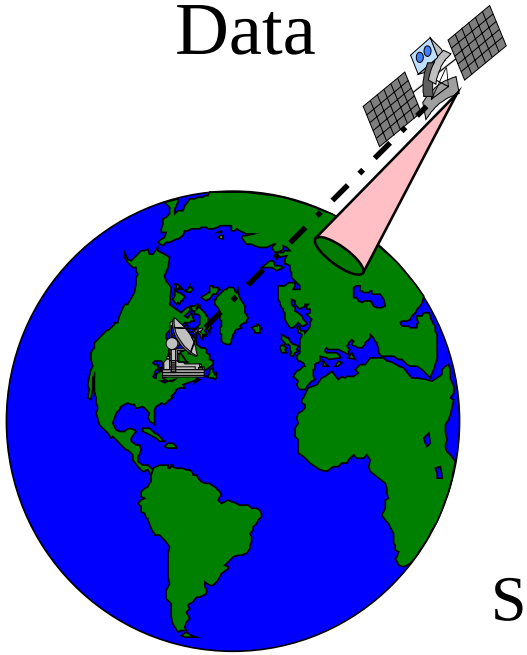


Data Access and Filtering from Distributed Data Collections

- Applications
  - processing and exploring large datasets over a wide-area network (WAN) is recent challenge to be solved
  - largely due to apps that generate large datasets (sensors, scientific simulations)
- Trends
  - heterogeneous systems (power, capacity, etc.)
  - able to build large cheap disk farms (1TB < $15k)
  - needed data accessible over WAN

Generating Data Subsets

Petabytes of Sensor Data



Generate initial conditions for climate model

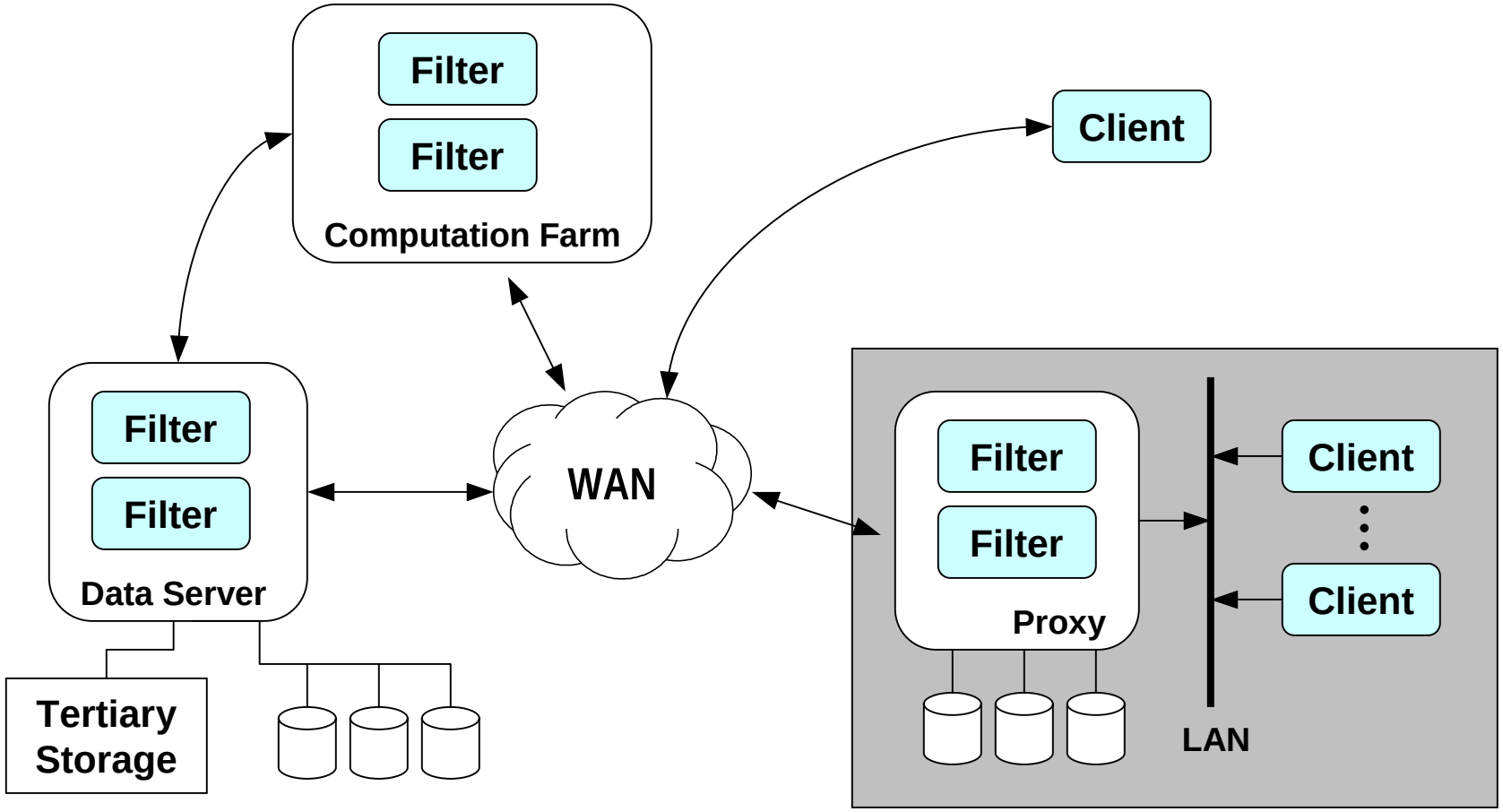
Database: Disk Cache

Spatial Subset: AVHRR North America 1996-1997

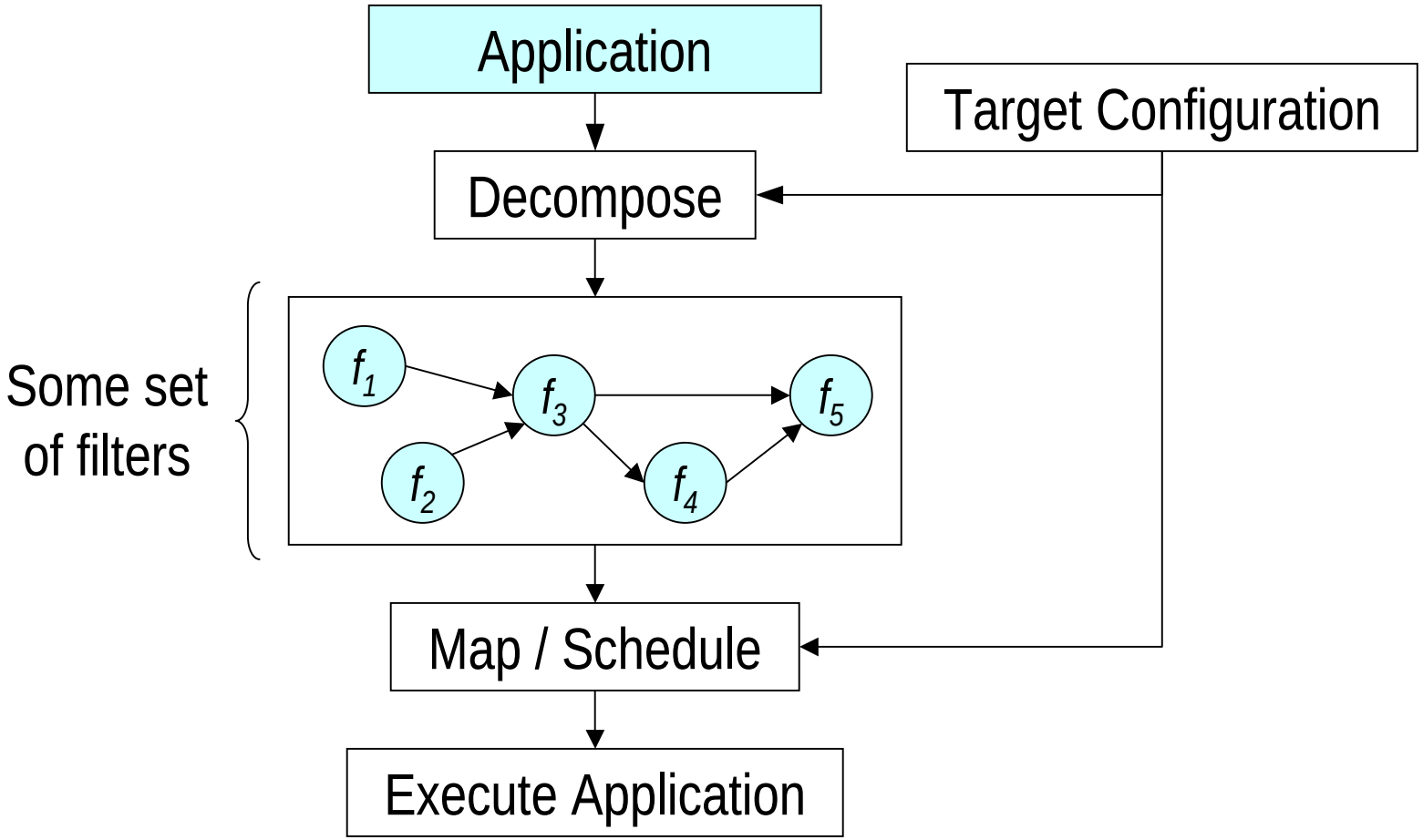
Generate Data Products

Visualize

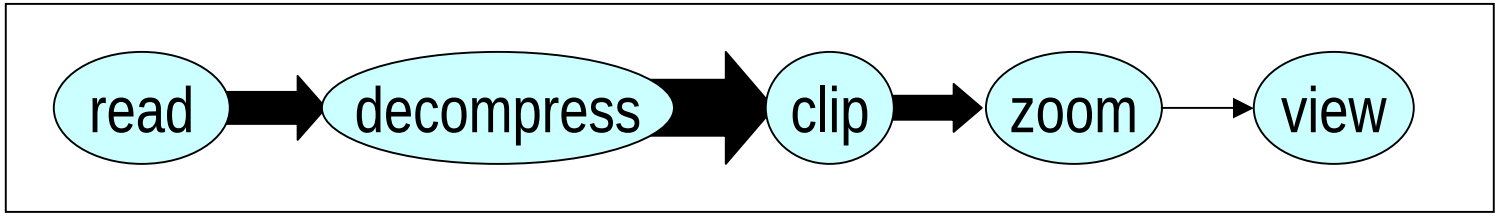
Target Architecture



Idea: Specify Application as Filters Extension of Disklets!!

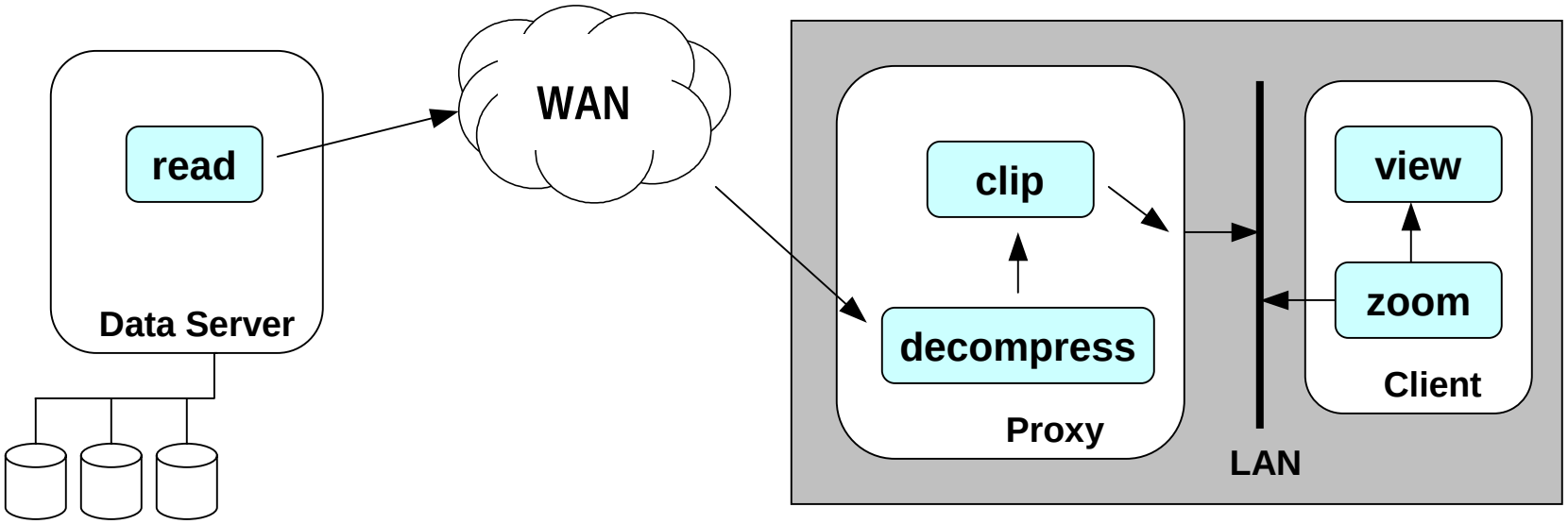


Possible Filters



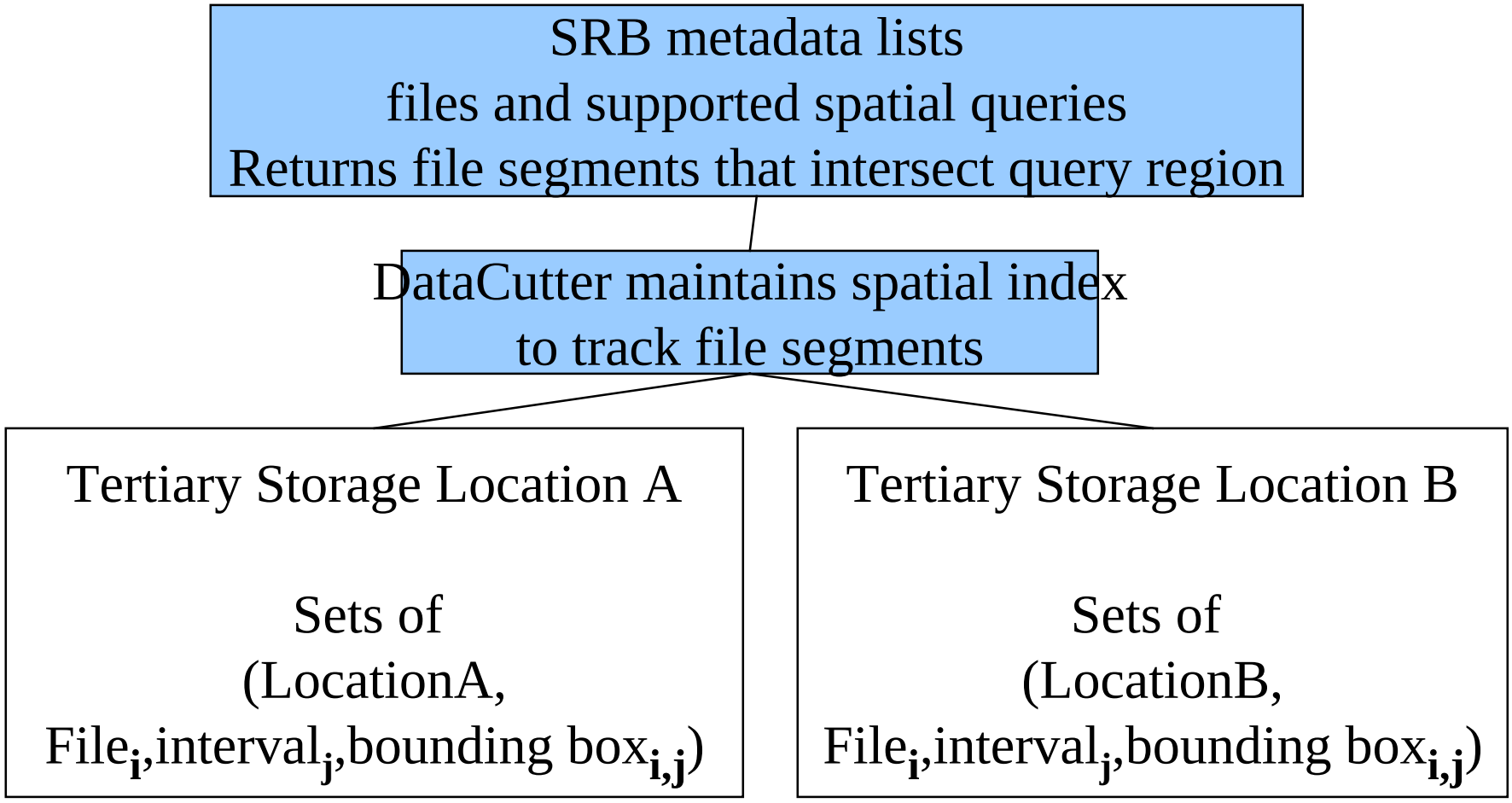
- Graph (chain) of filters connected by streams
  - read: read image chunks that overlap the query.
  - decompress: convert jpeg image chunks into RGB pixels.
  - clip: only keep portion of chunk inside the query region.
  - zoom: sub-sample to the required magnification.
  - view: stitch chunks together and display image.

Sample Mapping



- Sample Goal: match data volume and link characteristics
  - streams with large data volume 🕓 fast low latency links
  - streams with small data volume 🕓 slow WAN link

DataCutter Architecture I
National Partnership for Advanced Computational Infrastructure (NPACI)



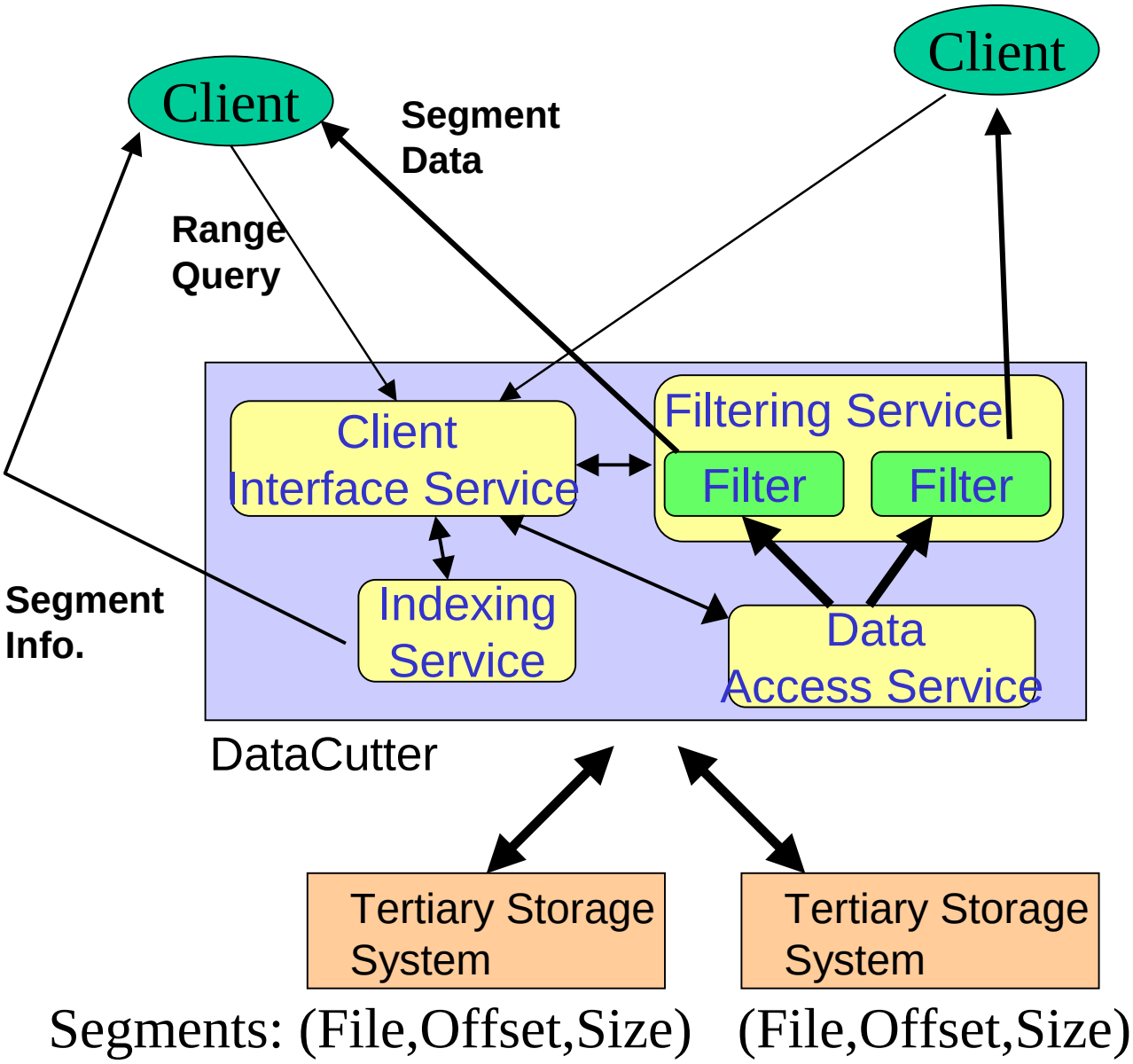
DataCutter Architecture II

- Proxy processes (filters) process 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

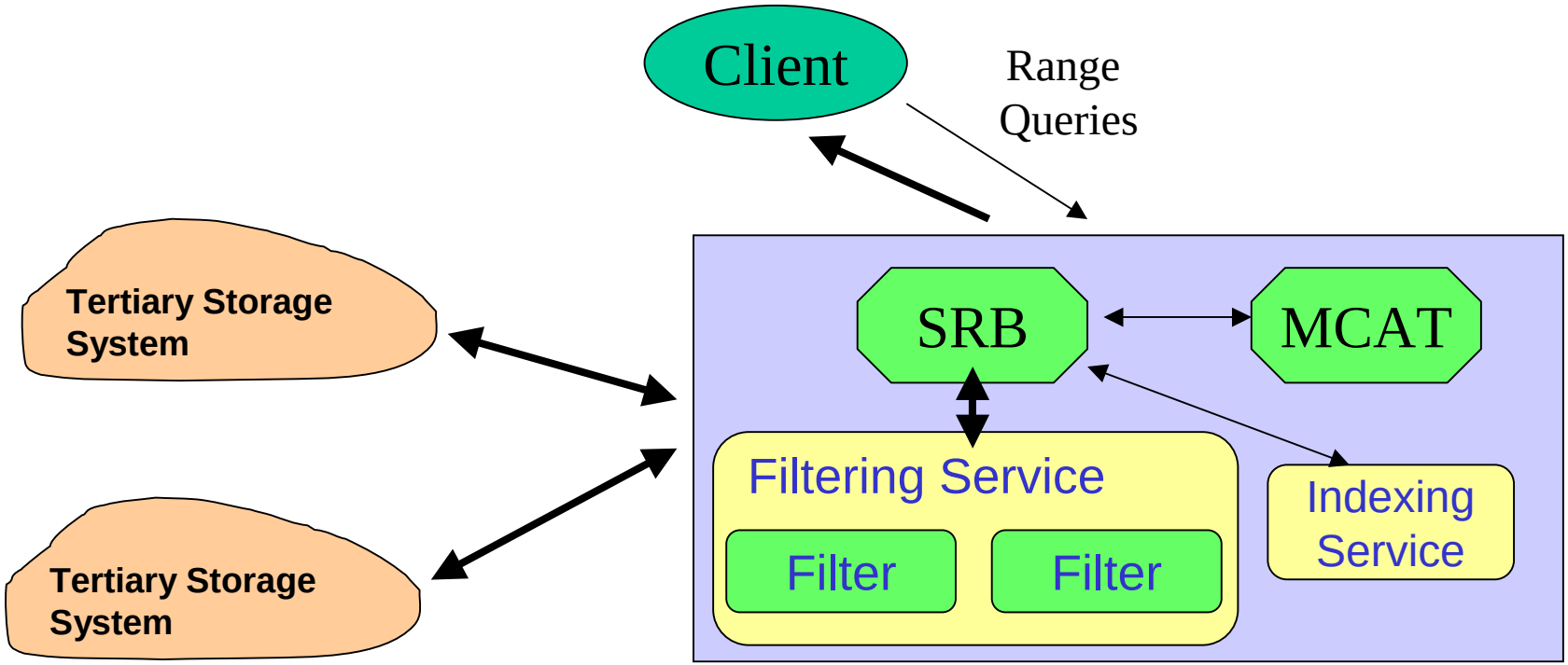
Database operations supported by Filter Algorithms

- 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]

DataCutter


Client
Client
Segment
Data
Range
Query
Client
Interface Service
Filtering Service
Filter
Filter
Indexing
Service
Data
Access Service
Segment
Info.
DataCutter
Tertiary Storage
System
Tertiary Storage
System
Segments: (File,Offset,Size)
(File,Offset,Size)

SRB/DataCutter - Prototype Implementation at San Diego Supercomputer Center



Conclusion

- Deep storage hierarchies and computational grid are coming to Pathology
- We can leverage off of work done in the broader application community
- Talk has focused on Hopkins/UMD/NPACI work but this is just a fraction of the excellent work going on in this area
- No mature grid applications -- Pathology can lead the way!!

Research Group

- Alan Sussman, Mike Beynon, Tahsin Kurc, Charlie Chang, Renato Ferraria, Robert Miller, Mustafa Uysal -- Johns Hopkins Medical School/ University of Maryland Computer Science Department
- Work done in collaboration with National Partnership for Applied Computational Infrastructure