Applications and Storage Needs

Joel Saltz

University of Maryland, College Park

Johns Hopkins University

Overview

- Nature of Multi-resolution Applications
  - Sensor measurements, physical simulations
- Tools to Manage Storage Hierarchy
  - Mass Storage:
    - Subset and filter
    - Load small subset of data from tertiary storage into disk cache or client
  - Fast secondary storage
    - Tools for on-demand data product generation, interactive data exploration

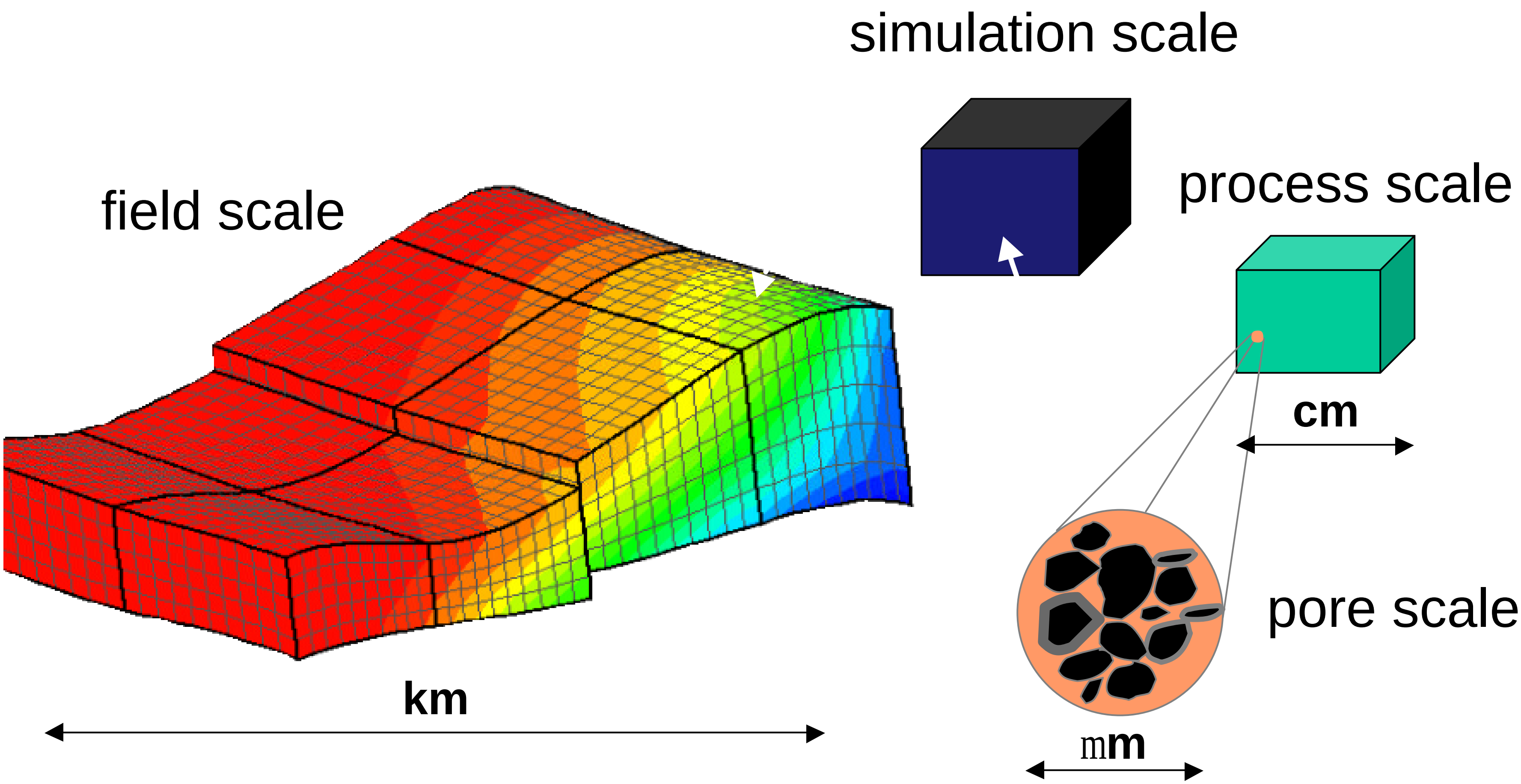
Applications

- Applications that describe or model the physical (or biological) world
  - Why these produce huge datasets and how big will these datasets get?
  - Structural features of these datasets
  - Type of queries and processing required

Why Applications Get Big

- Physical world or simulation results
- Detailed description of two, three (or more) dimensional space
- High resolution in each dimension, lots of timesteps
  - e.g. oil reservoir code -- simulate 100 km by 100 km region to 1 km depth at resolution of 100 cm:
    - 10^6*10^6*10^4 mesh points, 10^2 bytes per mesh point, 10^6 timesteps --- ***10^24 bytes (Yottabyte) of data!!!***

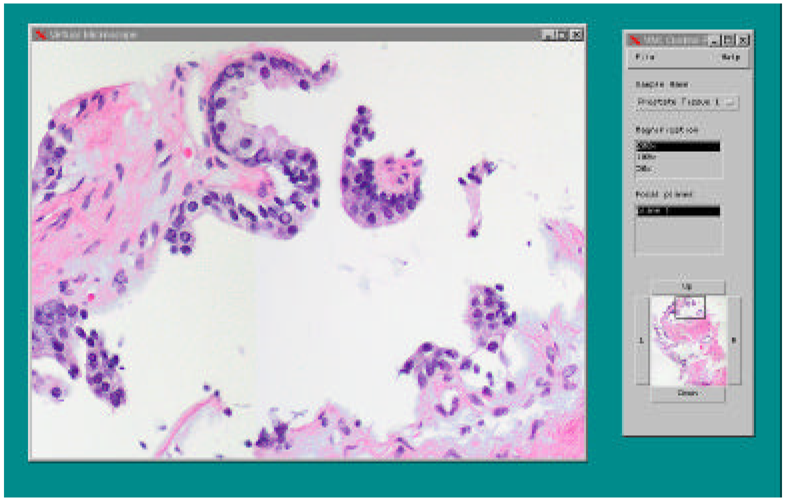
The Tyranny of Scale
(Tinsley Oden - U Texas)



How fast will we produce data?

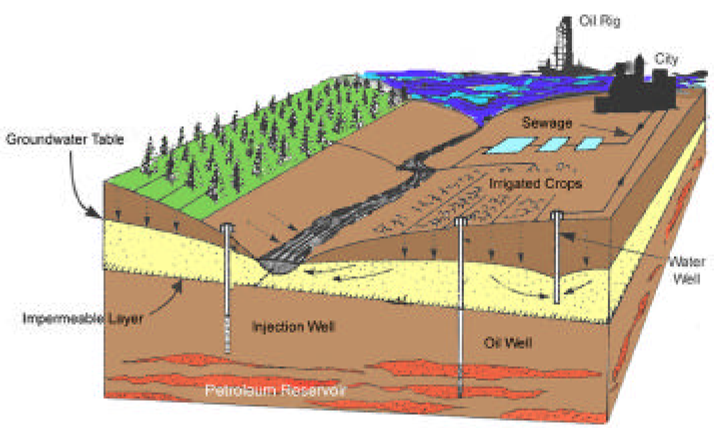
- Current fastest computers -- order of 10^12 floating point operations per year
- 5 years -- 10^15 floating point operations per second (ASCI timetable)
- Assume 10^4 floating point operations produces a byte of data
- 10^11 bytes of data per second, approximately 10^16 bytes (petabyte) per day

Applications

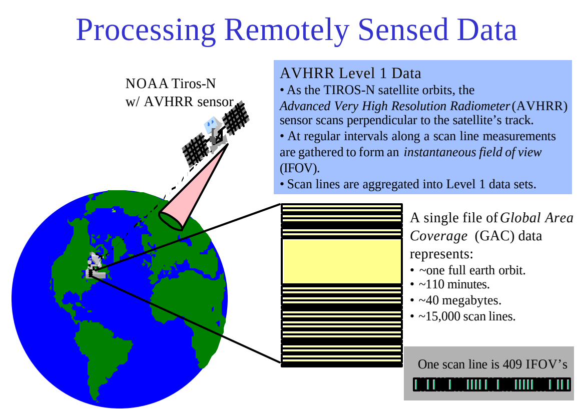


Pathology

Volume Rendering



Surface/Groundwater Modeling

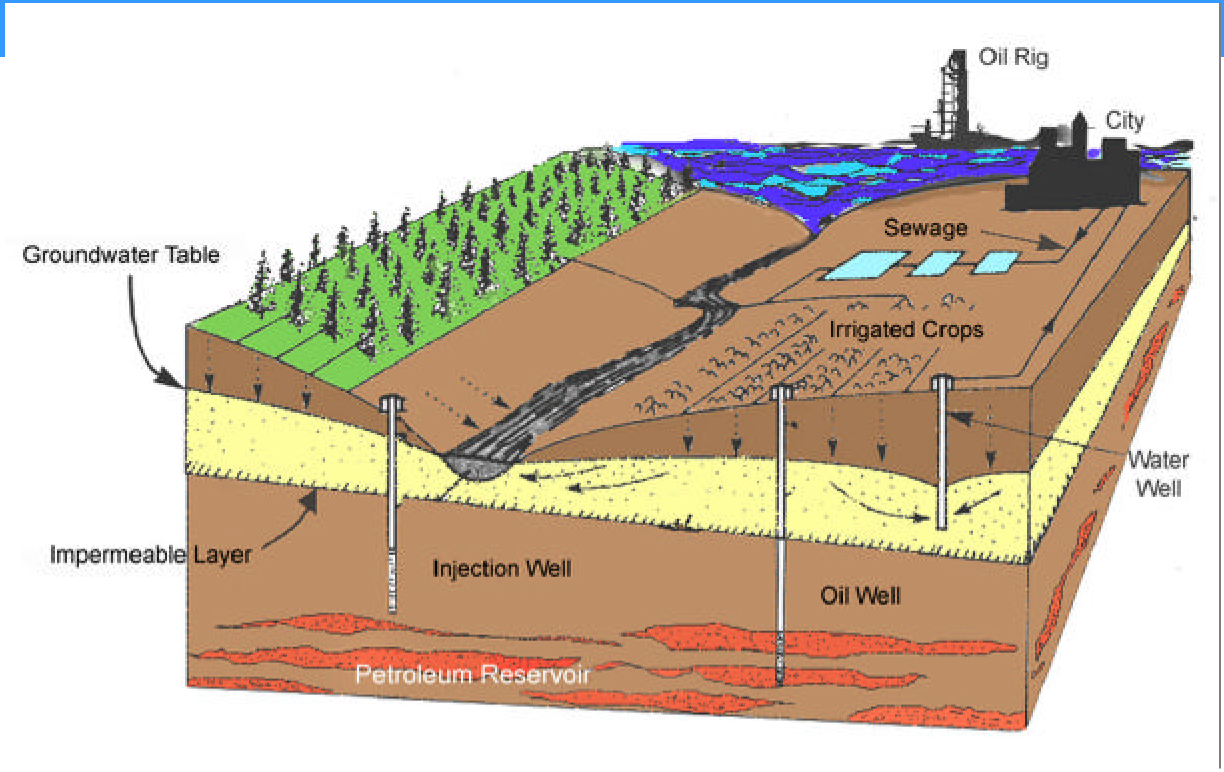


Satellite Data Analysis

Why Multi-Resolution?

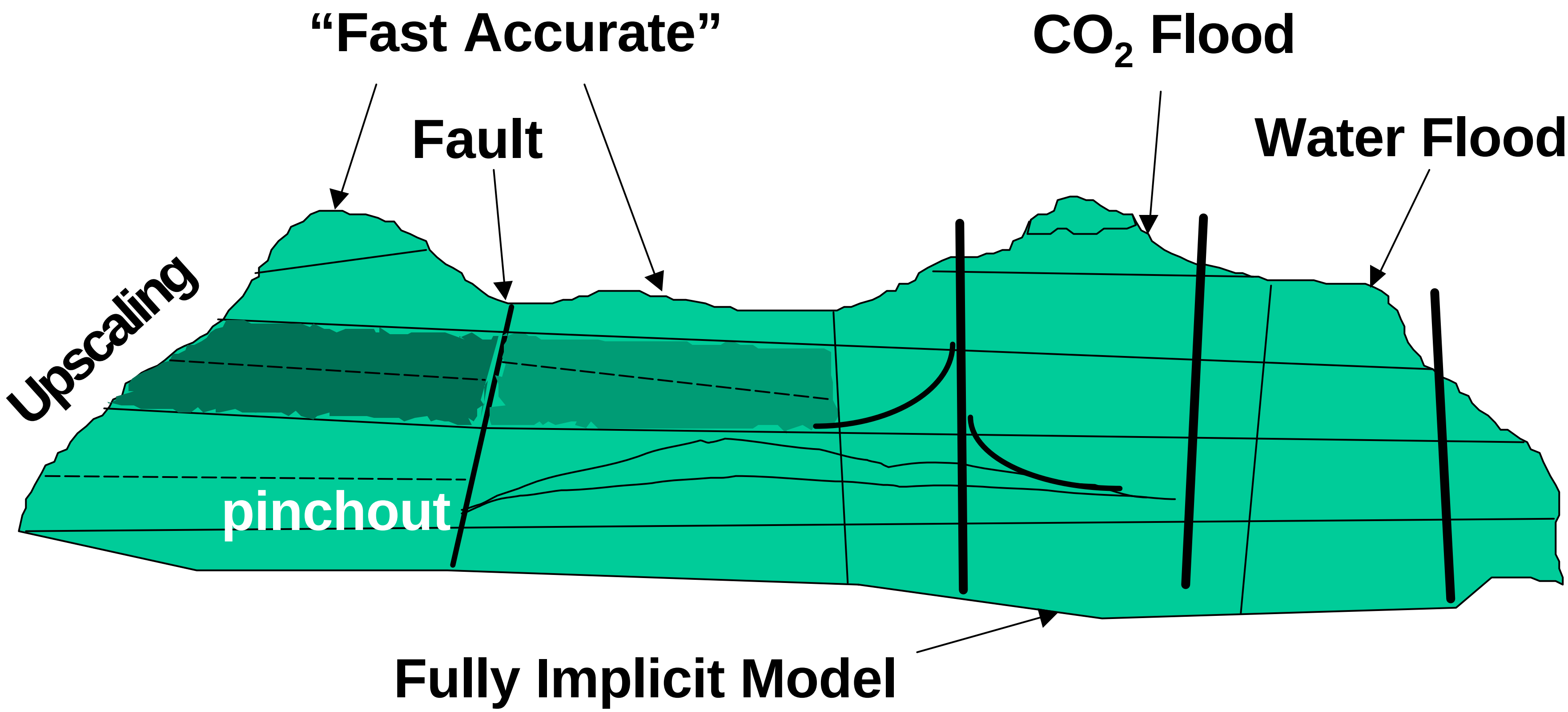
- Producing and managing Yottabyte datasets (10^24 bytes) is a dicey proposition
- Can often use variable resolution methods to reduce required datasize
- Leads to Spatial/multidimensional multi-scale, multi-resolution datasets

Coupled Ground Water and Surface Water Simulations



Multiple codes -- e.g. fluid code, contaminant transport code
Different space and time scales
Data from a given fluid code run is used in different contaminant transport code scenarios

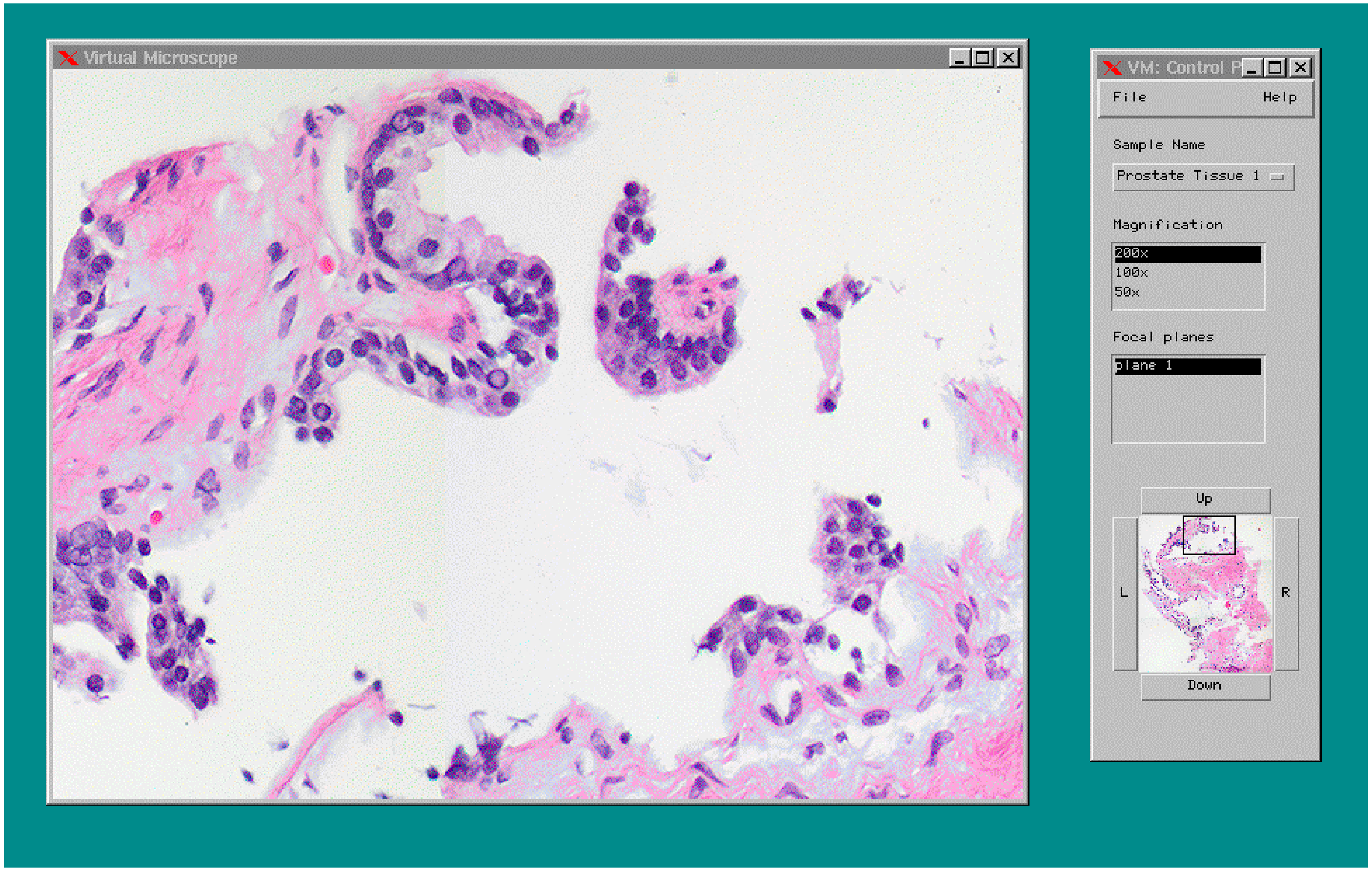
Multiresolution Methods Lead to Irregular Datasets



Pathology Application Domain

- Automated capture of, computer assisted categorization and immediate worldwide access to all Pathology case material
  - microscopy, blood smears, cytogenetics, molecular diagnostic data, lab data
  - Slide data -- .5-10 GB (compressed) per slide Johns Hopkins -- 500,000 slides per year
  - Data acquired and stored in patches at varying magnifications
- *Digital storage of 10% of slides in USA -- 50 petabytes per year*

Virtual Microscope

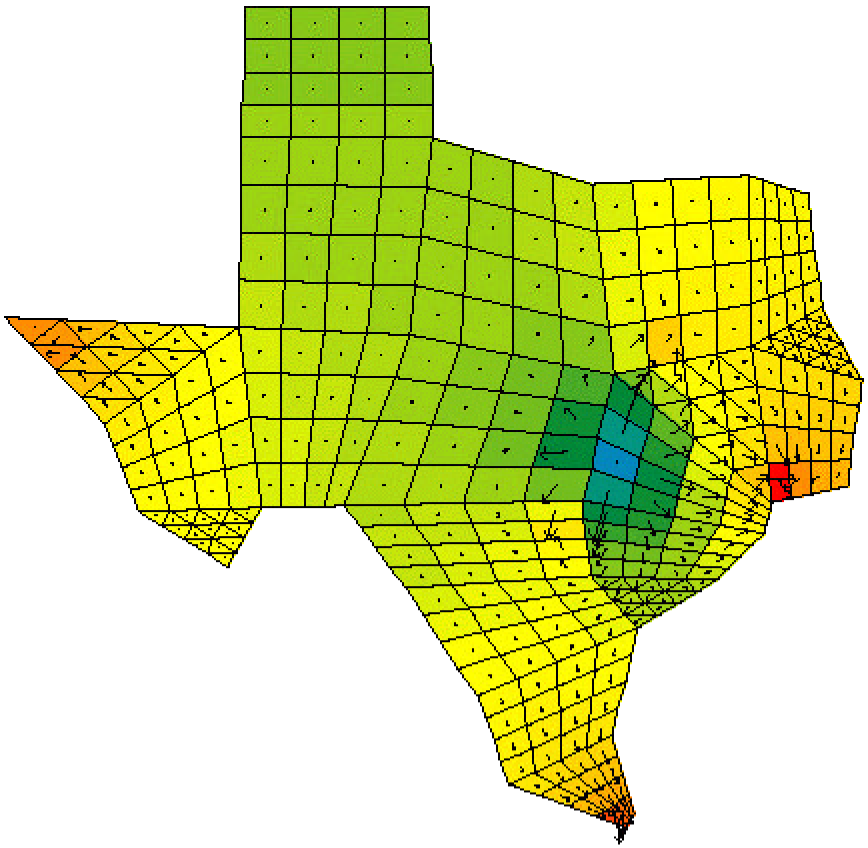


Explore virtual slides

Invoke user image processing functions to grade tumors

Find all data that meet a given criteria (e.g. Gleason Grade)

Dataset Structure

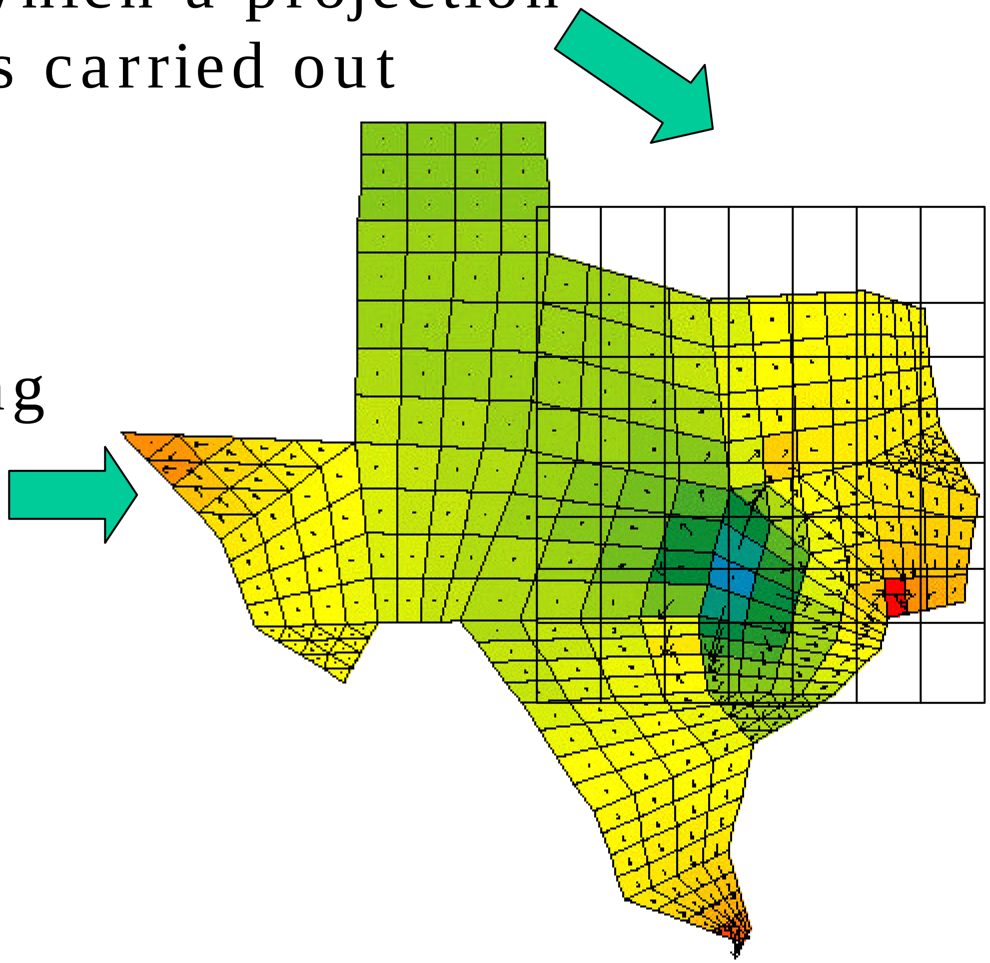


- Spatial and temporal resolution may depend on spatial location
- Physical quantities computed and stored vary with spatial location

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



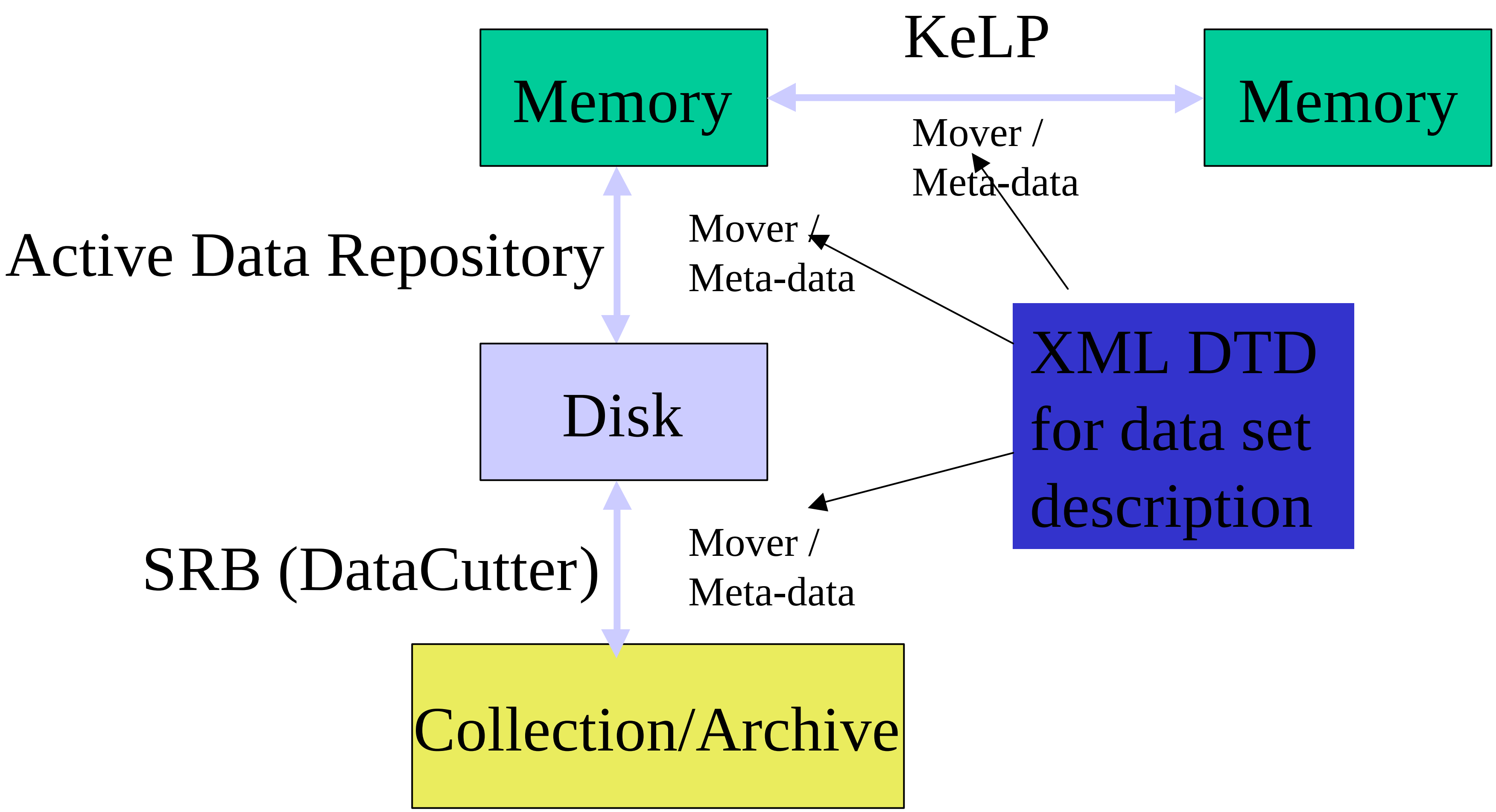
Software for Managing Storage Hierarchy

- Approach shared between all levels of storage hierarchy
- Active Data Repository
  - disk based processing
- DataCutter
  - data collections -- ultimately may subsume Active Data Repository
- Relationship between storage hierarchy management software and active disks

Software for Multi-resolution Data Handling

- *Associative access and processing of multiresolution data structures*
- *Distributed memory, shared nothing, active disk and multiple data collections*
- Users define
  - Objects, each comprised of data elements distributed in space
  - Methods to link each data with a spatial location
  - Spatial subsetting functions
  - Aggregation functions that combine objects within and between coordinate systems

Data Handling and Memory Hierarchy -- NPACI Software

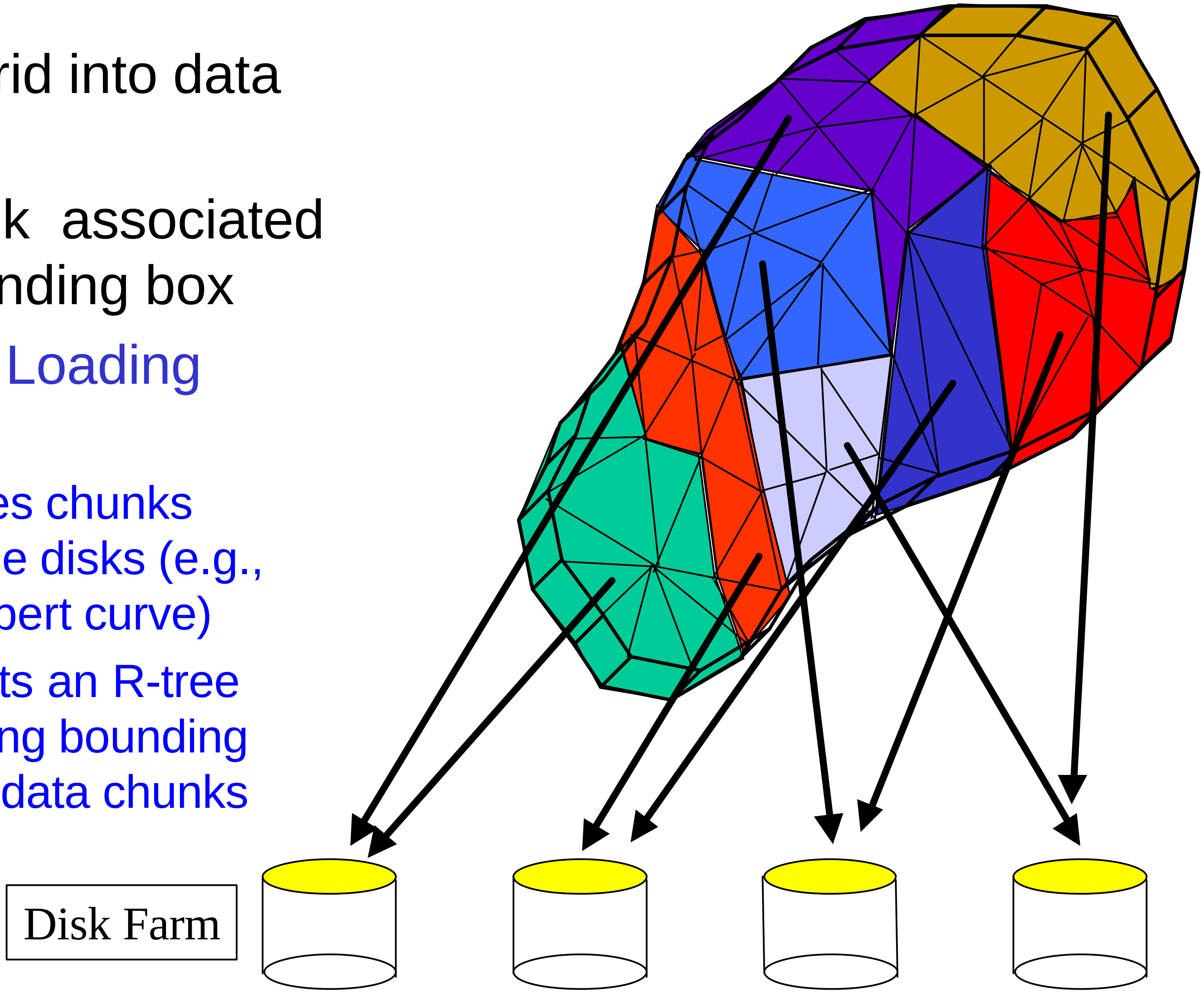


Active Data Repository

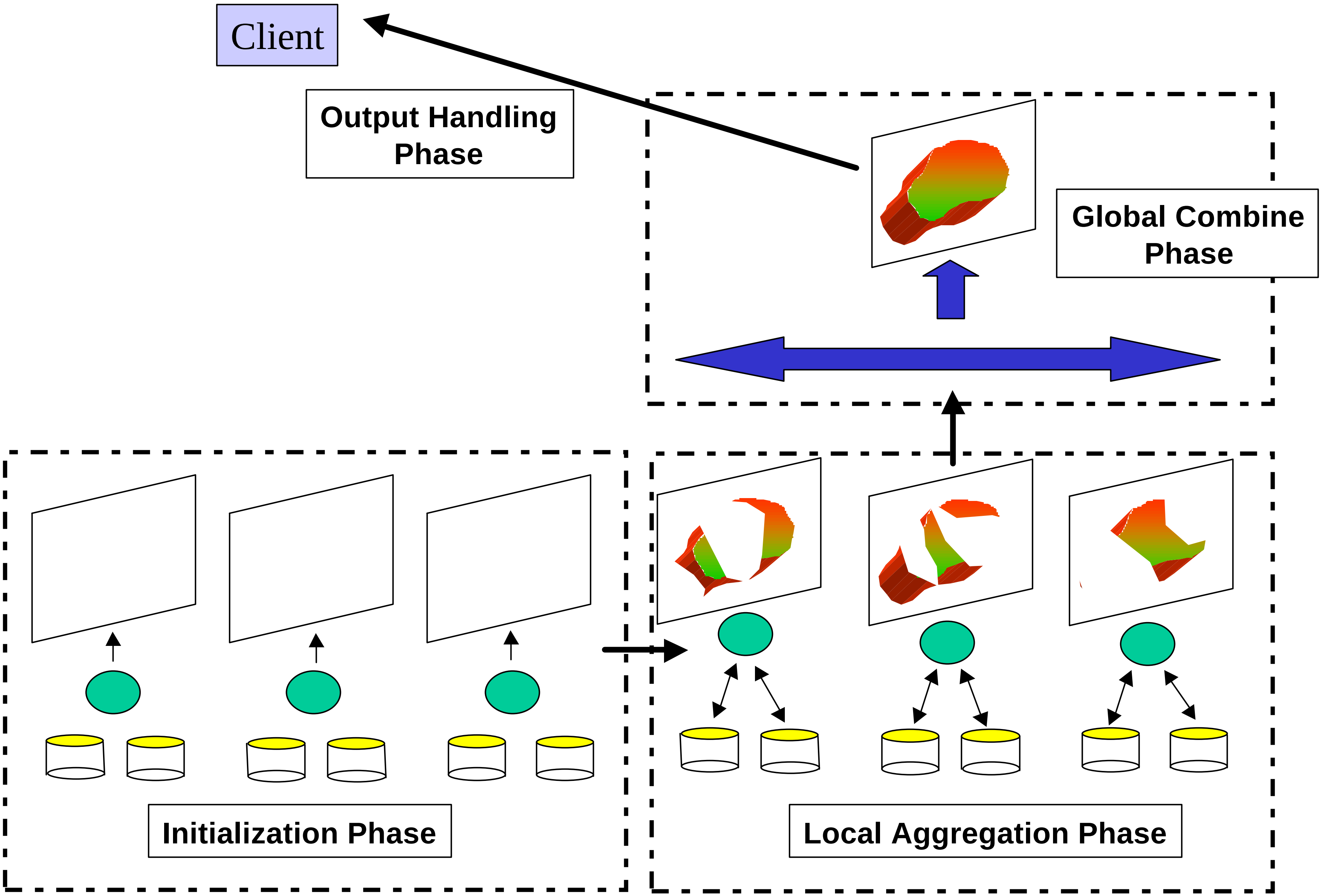
- *Optimized associative access and processing of multiresolution disk based data structures*
- Targets *parallel and distributed architectures, active disk architectures*
- Modular services implemented in C++
  - Ongoing work to develop a compiler to allow users to customize services using user defined functions
- Applications
  - Satellite sensor data -- several types; Virtual Microscope Server, Bay and Estuary Simulation, Large scale data visualization

Loading Grids into ADR

- Partition grid into data chunks
- Each chunk associated with a bounding box
- ADR Data Loading Service
  - Distributes chunks across the disks (e.g., using Hilbert curve)
  - Constructs an R-tree index using bounding boxes of data chunks



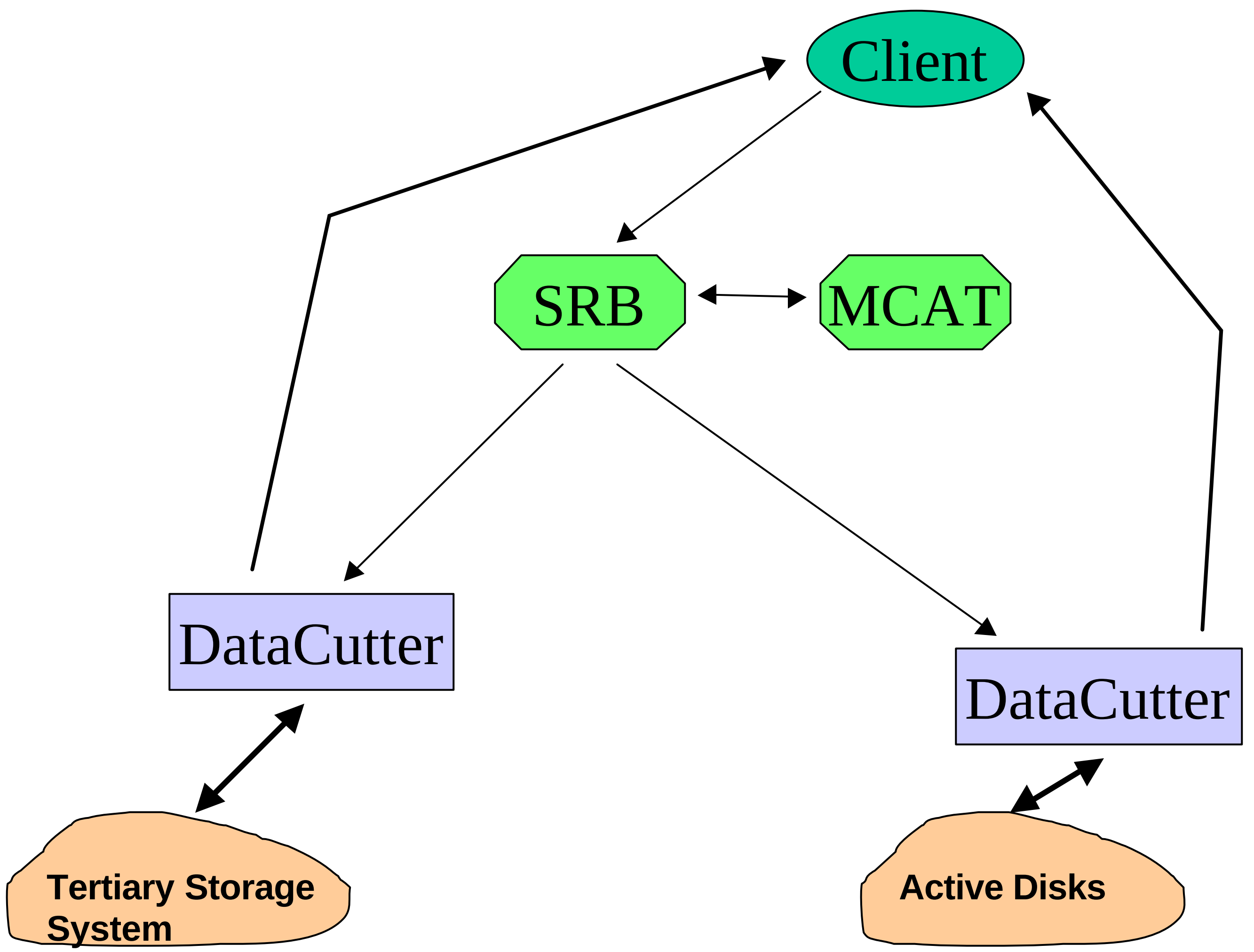
Query Processing



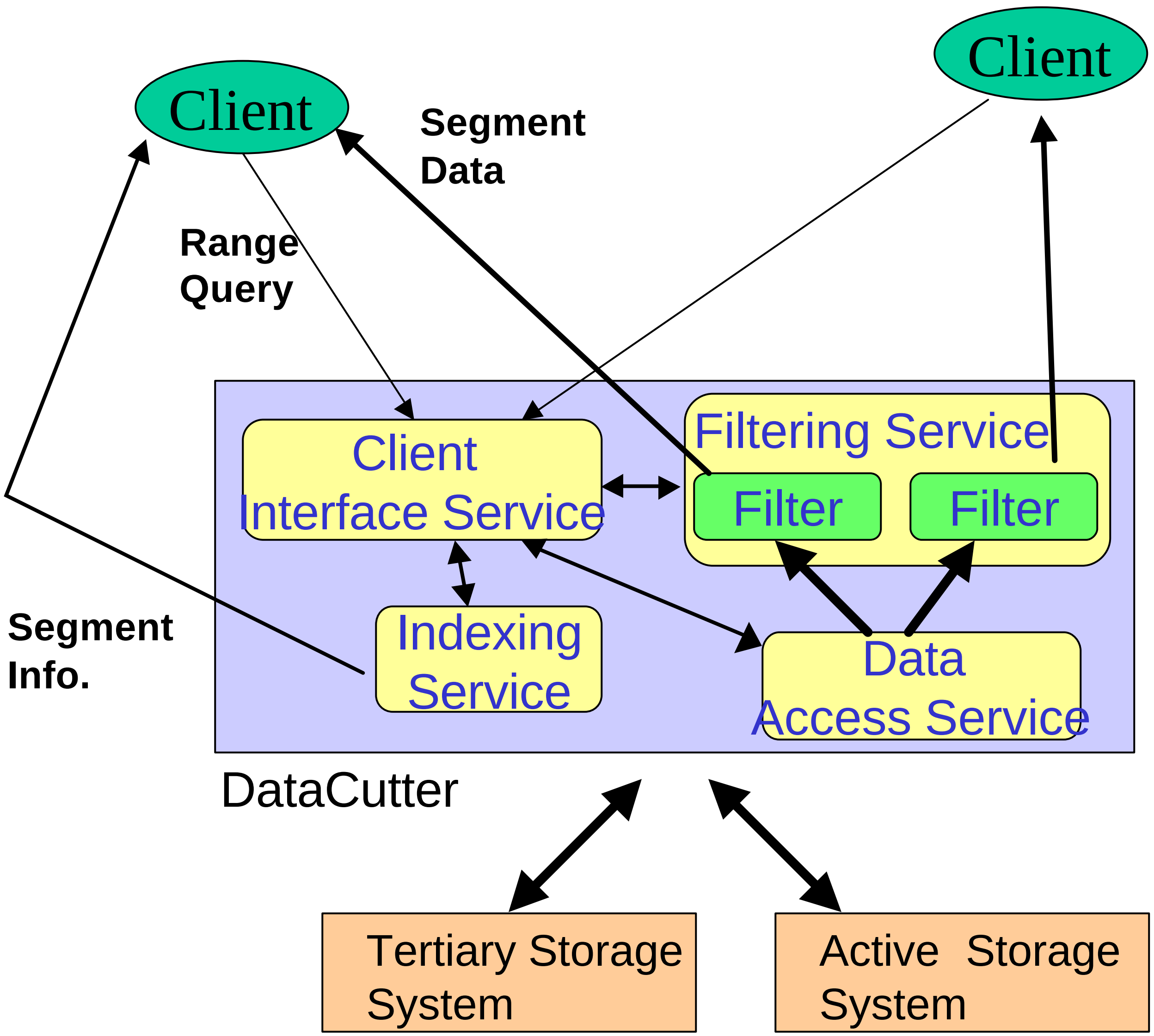
DataCutter Architecture

- Proxy processes (disklets) filter data as it is extracted from storage
  - Active disks -- disklets as proposed by Uysal
  - Data collections -- “disklets” carry out computation close to data
- Early data filtering reduces data movement and data transfer costs
- Uysal has developed algorithms that use fixed amount of scratch memory to carry out selects, sorts, joins, datacube operations

SRB/DataCutter Implementation



DataCutter



Segments: (File,Offset,Size) (File,Offset,Size)

Uysal's 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)

Disklets as Filters

- *Resource requirements associated with remote data processing can be predicted and constrained*
  - Disklets cannot initiate I/O
  - Disklets cannot allocate/free memory
    - long-term scratch-space, stream buffers
    - DiskOS automatically allocates/reclaims memory
- *Easy disklet composition/migration*
  - Disklets cannot determine stream source/sink

Summary

- Where very big datasets come from and inevitable structural aspects
- Example of software architecture for managing storage hierarchy
- Relationship to active storage
  - put processing close to data storage!
  - Some elements of storage hierarchy will probably be active disk based -- software architecture should be able to take advantage of this (ours does!)