Outline of the Tutorial Modules

I. Overview on Scientific Data Management (Gertz)
13:30—14:15

II. From Conventional to Scientific Data Integration (Ludaescher)
14:15—15:00

III. From Scientific Data Formats to Data Stream Processing (Gertz)
15:30—16:15

IV. Introduction to Scientific Workflows (Ludaescher)
16:15-17:00

III. From Scientific Data Formats to Data Stream Processing

- Overview
  - Scientific data formats (and data models)
    - HDF, CDF, netCDF, FITS, XSIL, GML
  - What are recent trends? What about XML ?
  - Data processing pipelines
  - Data stream processing

Scientific Data Formats

Data Diversity

- Scientific data come from a variety of sources (remote-sensing instruments, sensors, experiments, simulations...), often in proprietary data formats.
- Typical "raw data" and data products include
  multi-dimensional arrays, images, spectra, vector fields
- Different scientific communities have different data management and data processing needs, and thus use different data formats.

The purpose of specialized scientific data formats is to

- store, manage, exchange, share, and archive data from scientific applications
- provide the base for data management and analysis tools, such as data integration, sharing, visualization, and archiving
- Provide ***complete software solutions*** with support of a variety of programming languages, ranging from Fortran 77 to Java

Scientific Data Formats (2)

There are several key data formats Earth & Space Sciences

- HDF (Hierarchical Data Format)
- CDF/netCDF (Common Data Format)
- FITS (Flexible Image Transport System)
- XSIL (eXtensible Scientific Interchange Language)
- GML (Geography Markup Language)
- GRIB (Grids in Binary)
- CCM (Community Climate Model History Tape Format)

...and many XML-based languages in the Life Sciences...

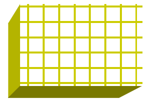
Chemical Markup Language (*CML*) – Molecular Dynamics [Markup] Language (*MoDL*) – MicroArray and Gene Expression Markup Language (*MAGE-ML*) – Genome Annotation Markup Elements (*GAME*) – BIOpolymer Markup Language (*BIOML*) – Numerical Data Markup Language (*NDML*) – Protein Extensible Markup Language (*PROXIML*) – Systems Biology Markup Language (SBML) ...

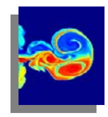
(see also http://xml.coverpages.org/)

Scientific Data Formats (3)

For Earth and Space Sciences, the major driver for scientific data formats is the effective representation and management of

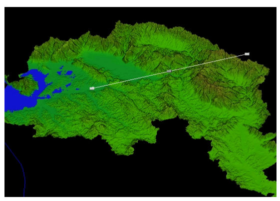
- Multi-dimensional arrays
- Tables of records
- Images
- Metadata associated with the above

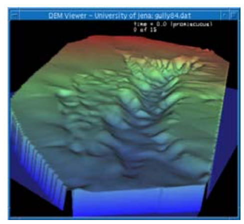


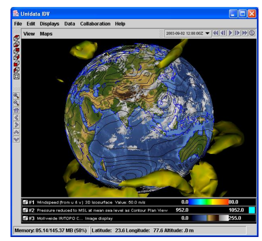


lat	lon	temp
12	23	3.1
15	24	4.2
17	21	3.6

Many of the languages for scientific data formats are data-intensive and come with a variety of APIs.

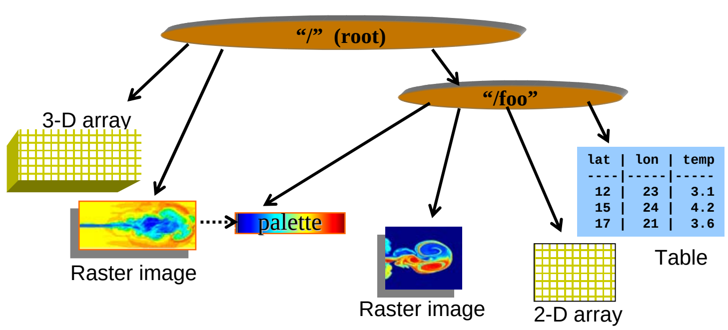






Hierarchical Data Format (HDF)

- Developed by NCSA; HDF4, HDF5; http://hdf.ncsa.uiuc.edu/
- HDF is a self-defining file format for transfer of various types of data
- HDF ***software library*** contains APIs for
  - storing and retrieving (un)compressed raster images with palettes,
  - storing and retrieving multi-dimensional scientific datasets together with information about the data, such as labels, units, formats, and scales for all dimensions.



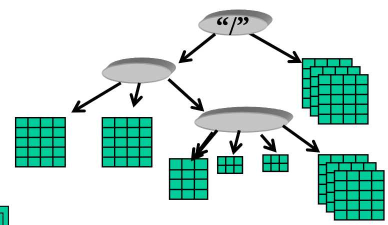
HDF (2)

HDF Data Model

- Primary objects: groups and datasets
- Additional ways to describe data: attributes, sharable objects, storage and access properties

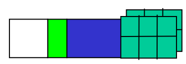
A group is a way to describe collections of related objects

- HDF file starts with a root group
- Similar to Unix directory structure
- Objects are identified and located by their pathname
- Groups and members of groups can be shared



Dataset

- Data arrays of records
- Elements are based on atomic and composite data types



Record

HDF (3)

Metadata

- Data space – rank, dimensions, other spatial info about dataset
- Attribute list – user-defined metadata
- Special storage options – how array is organized

Data spaces

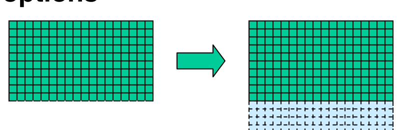
- Spatial info about a dataset: rank and dimension

Attributes

- “name = value” attached to objects

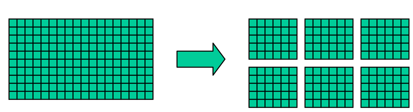
Special storage options

extendible



Arrays can be extended in any direction

chunked



Better subsetting access time; extendible

compressed, external file

Common Data Format (CDF)

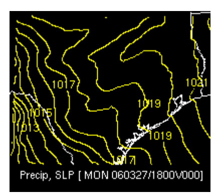
Designed and developed in 1985 by the National Space Science Data Center at NASA Goddard Space Flight Center.

- Conceptual data abstraction for storing, manipulating, and accessing multi-dimensional data sets.
- Basic component: software programming interface that is a device independent view of the CDF data model.
- CDF files created on any given platform can be transported to any other platform onto which CDF is ported and used with any CDF tools or layered applications.
- CDF ***software package*** is used by hundreds of government agencies, universities, and private and commercial organizations as well as independent researchers on both national and international levels.
- Adopted by the International Solar-Terrestrial Physics project as well as the Central Data Handling Facilities (CDHF) as their format of choice for storing and distributing data.

nssdc.gsfc.nasa.gov/cdf/cdf_home.html

Network Common Data Format (netCDF)

- Developed by University Corporation for Atmospheric Research around 1990.
- netCDF library defines a machine-independent format for representing scientific data.
- Together, the APIs for array-oriented data access, library, and format support the creation, access, and sharing of scientific data.



Objectives:

- ***Self-describing***. A netCDF file includes information about the data it contains.
- ***Portable***. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- ***Direct-access***. A small subset of a large dataset may be accessed efficiently,
- ***Appendable***. Data may be appended to a properly structured netCDF file
- ***Sharable***. One writer and multiple readers may simultaneously access file
- ***Archivable***. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

http://www.unidata.ucar.edu/

netCDF (2)

There are many language interfaces to netCDF

– C, Fortran 77/90, C++, R, Ada, Perl, Python, Ruby, Tcl/Tk, Java

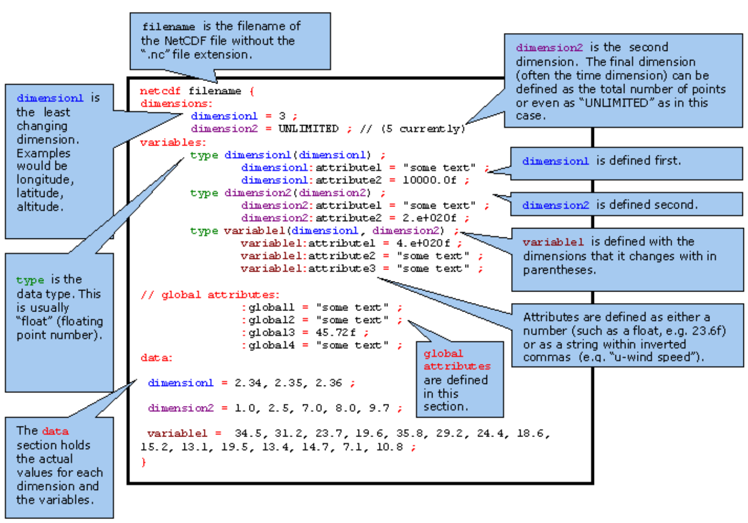
netCDF Operators

– Ncdump: reads a netCDF dataset and prints information from the dataset
  - ncdump –h file.nc -- print header
  - ncdump –v VAR_name file.nc -- print values of the VAR

– Ncgen: generates a netCDF file or a C or Fortran program that creates a netCDF dataset

– Other netCDF operators: NCO – http://nco.sourceforge.net/

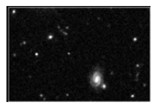
netCDF (3)



Src http://badc.nerc.ac.uk/help/formats/netcdf/index_cf.html

Flexible Image Transport System (FITS)

- Developed 1981 by NASA
- Data format most widely used within astronomy for transporting, analyzing, and archiving scientific data files.
- More than just another image format (such as JPG or GIF)
- Primarily designed to store scientific data sets consisting of multi-dimensional arrays (images) and 2-D tables organized into rows and columns of data.



- FITS file is comprised of segments called Header/Data Units (HDUs)
- First HDU is called the “Primary Array”, which can contain a 1-999 dimensional array of integers or floating point numbers.
- A typical primary array could contain a 1-D spectrum, a 2-D image, or a 3-D data cube.
- Any number of additional HDUs may follow the primary array; these are referred to as FITS “extensions” (Images, ASCII tables, binary tables)

Extensible Scientific Interchange Language (XSIL)

- Developed at Caltech’s Center for Advanced Computing Research
- XML tags for generic scientific data markup; comes with Java software to manipulate and display XSIL documents
- Flexible, hierarchical, extensible transport language for scientific data objects
- Comes with Java object model that is designed to be extensible, so that scientific data and metadata represented in XML is available to Java code.
- Developed in the context of several projects led by CACR, including
  - Laser Interferometer Gravitational Wave Observatory (LIGO)
  - Digital Sky (large-area sky surveys, focuses on multi-wavelength astronomical studies)
  - DigitialPuglia (Synthetic Aperture Radar Atlas); Web-based retrieval mechanism for SAR images)

http://www.cacr.caltech.edu/SDA/xsil/

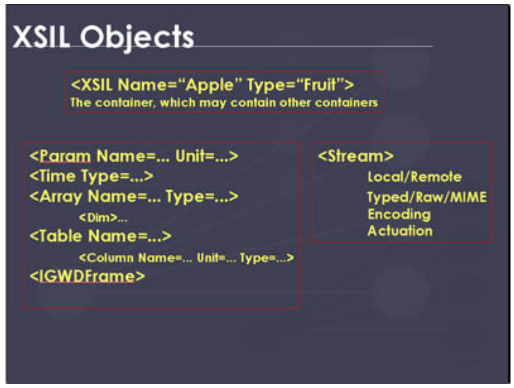
SDM Tutorial, EDBT’06, Gertz, Ludäscher

XSIL (2)

Container objects; container may contain other containers so that object includes hierarchy.

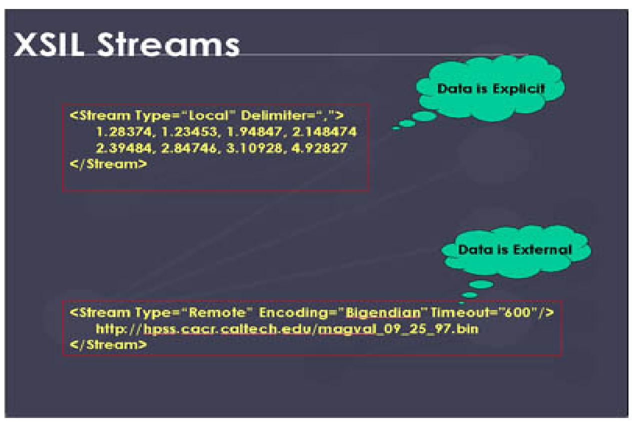
Element objects include Param, an association between keyword and a value (similar to array concept in Fortran or C)

A table is a list of column headers followed by a set of records



Src www.cacr.caltech.edu/SDA/xsil/

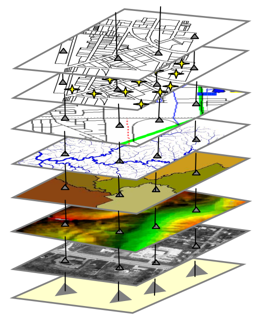
XSIL (3)



Stream object in XSIL is a mechanism to include data either within the XSIL file (local stream) or pointing to an external source (remote stream)

Geography Markup Language (GML)

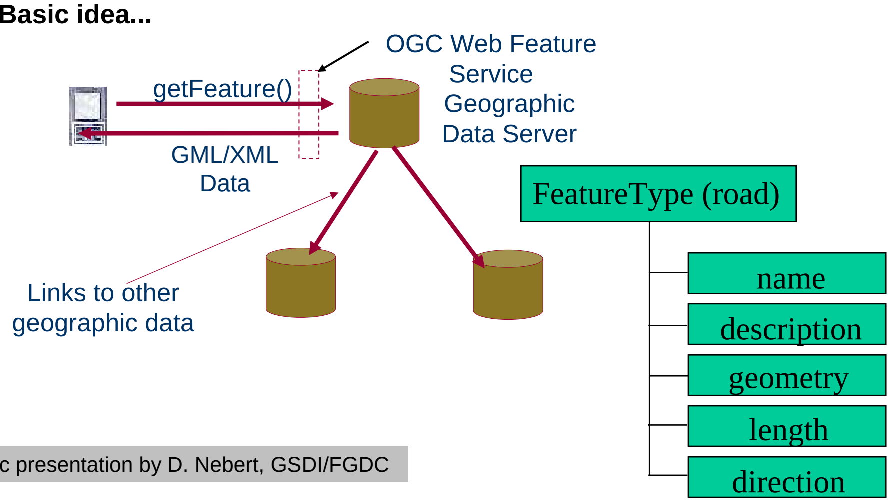
- XML based encoding standard for geographic information developed by the OpenGIS Consortium (OGC).
- Objective: allow Internet browsers the ability to view Web-based mapping without additional components or viewers.
- Application framework for handling geospatial information/data on the Web
- Enables complex features and feature associations among data
- Standard for representing geospatial data, endorsed by 200+ companies and agencies
- New features in GML3 include topology, temporal features, enhanced geometry, units of measure



http://www.opengeospatial.org/

GML (2)

Basic idea...



Src presentation by D. Nebert, GSDI/FGDC

GML (3)

Example instance of GML Feature; features are defined in GML application schemas

```
<exp:Building fid = "st100">
        <exp:noFloors>100</exp:noFloors>
        <exp:use>Residential</exp:use>
        <exp:surfaceArea>100000</exp:surfaceArea>
        <exp:frontsOn>Georgia Street</exp:frontsOn>
        <exp:streetNumber>1150</exp:streetNumber>
        <gml:locationOf>
                <gml:Point srsName = " … ">
                        <gml:coordinates>55661.1454,
                                454656.67
                        </gml:coordinates>
                </gml:Point>
        </gml:locationOf>
</ exp: Building>
```


Why so many? What are the differences....?

Some important observations about Earth Sciences

- Data formats have been designed by different communities, often many years ago, and are still maintained by these communities.
- There was/is nothing like a W3C for the science communities.
- Developments are driven by specific application domains and research objectives.
- Scientific data formats are not just data formats, but they ***come with complete software libraries*** that support a variety of programming languages (Fortran77 ... Java).
- XML? There is no urgent need to exchange data between communities...

In the life sciences, XML is much more prominent. Why?

- More complex structured data, not just large arrays of numerical data
- Standard vocabularies, taxonomies, and ontologies play an important role to semantically enrich data

Trends and Developments

There are many efforts two wrap scientific data represented in the data format XYZ into XML. For example,

– XML-based markup language called CDF Markup Language to describe CDF data and metadata and created the following two utilities in Java:
  - CDF2CDFML - dumps the contents of a CDF file into a XML file that conforms to the CDF DTD or CDF schema.
  - CDFML2CDF - creates a CDF file from an XML file that conforms to the CDF DTD or CDF schema.

– To promote data exchanges among space scientists, the CDF office has developed a Web service called Data Translation Web Service (DTWS) and a client to talk to DTWS via a web browser. The DTWS is a web service based on SOAP. Supported translations include
  - CDF-to-netCDF, CDF-to-FITS, CDF-to-CDFML, netCDF-to-CDF
  - FITS-to-CDF, HDF4-to-CDF

  http://translators.gsfc.nasa.gov

– XSLT is a helper in doing such transformations!

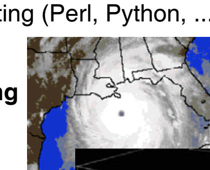
Processing Scientific Data

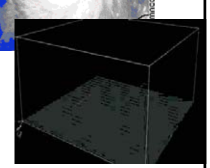
In Earth Sciences and Astrophysics

– Almost exclusively based on a file-processing approach:
  1. obtain files from instrument, experiment, or simulation (ftp, gridFTP,...)
  2. run program on files, typically local
  3. record data product as file and store it either locally or remotely

– Data and metadata go through numerous complex processing steps formulated as (image) pipelines and workflows (→ Module 4)

– Pipelines are assembled manually, often through scripting (Perl, Python, ...)





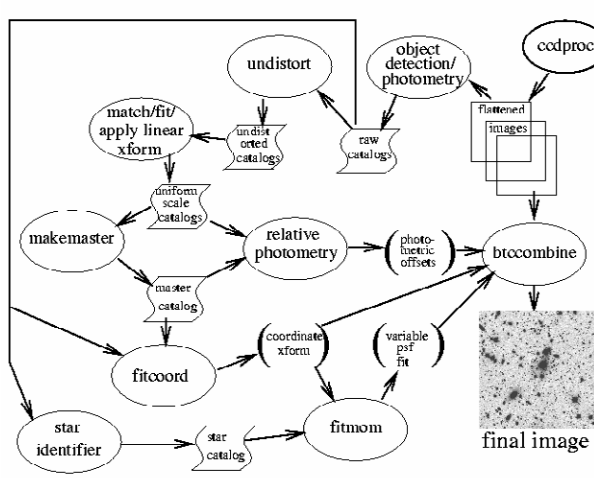
There are many applications settings in which incoming data is delivered in a continuous, streaming fashion but the processing occurs in a file-based approach.

– Weather forecasting, wildfire detection, hurricane tracking...

– In general, there are many types of remote-sensing applications in which real-time processing is crucial.

Data Processing Pipelines

Example data processing pipeline in the context of the Deep Lens Survey (DLS)



— Charge-coupled device

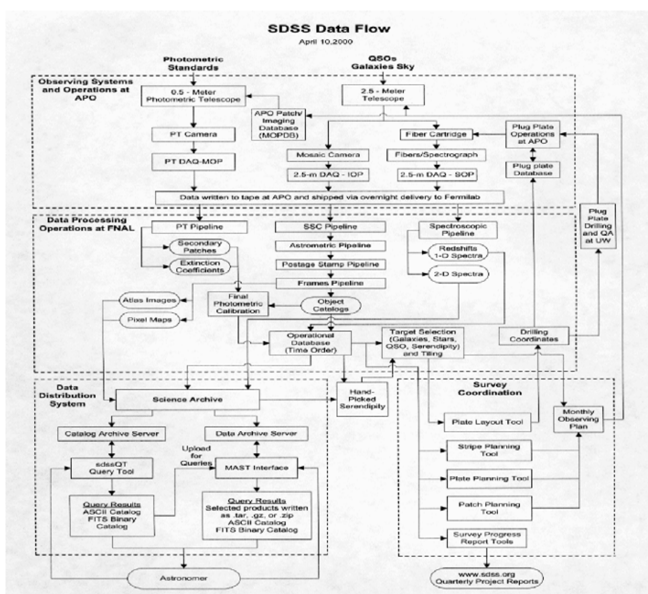
DLS images show roughly ten times as many galaxies per unit area than the Sloan Digital Sky Survey (SDSS)

http://dls.bell-labs.com/

Data Processing Pipelines (2)

Data flow in the Sloan Digital Sky Survey

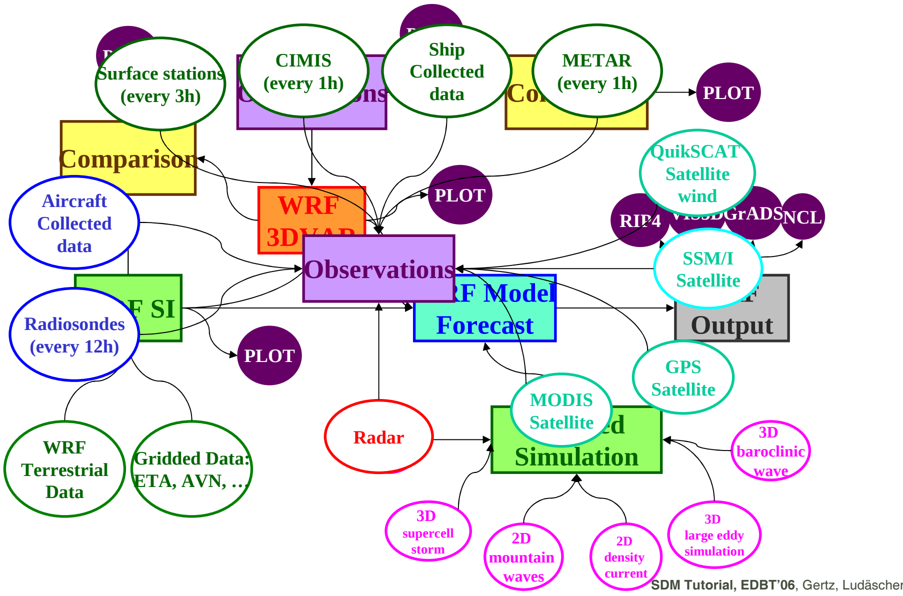
Photometric pipeline **turns the imaging data into information** about the celestial objects



Src: http://cas.sdss.org/dr4/en/sdss/data/data.asp

Data Processing Pipelines (3)

Weather Research & Forecasting Model



Data Stream Processing

Objectives:

– Exploit models, techniques, and concepts developed for traditional (relational and XML) data stream management systems

– Develop stream management system against which users can formulate queries and analytical operations

– ***Move query processing and computation to the data***

How to go about this?

– Get an understanding of the science domain, talk to scientists

– Develop formal data model for scientific data of concern (e.g., multi-dimensional arrays in HDF), streams of such data, and operations on data streams

– Start with the design and implement a stream simulator that takes science data files and converts them into stream

Data Stream Processing (2)

There are some scientific data models out there (developed by the DB community) that look at non-streaming data, e.g.,

- Multi-dimensional arrays (Marathe and Salem, VLDB Journal 2002; Libkin, SIGMOD 1996)
- Array algebra (Baumann, VLDB Journal 1994, RasDaMan)
- Gridded data sets (Howe and Maier, VLDB 2004)

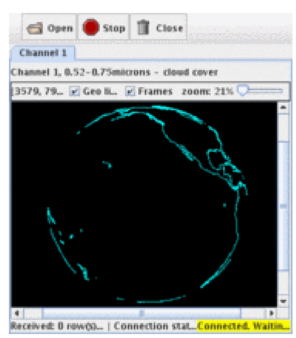
Can't we just take "conventional" data stream processing models, techniques, and architectures?

- It is not a good idea to map hundreds of millions of pixels to relations
- Operations on multi-dimensional data (e.g., raster images) are often much more complex than selection, project, join (e.g., re-projections or affine image transforms)
- Data sets exhibit spatio-temporal characteristics
- Queries are typically formulated using graphical user interfaces

Streaming Geospatial Image Data

A case study:

- GeoStreams: A query processing architecture for streaming remotely-sensed image data http://geostreams.ucdavis.edu
- GOES West, geostationary weather satellite operated by NOAA
- 5 imager bands, 19 sounder channels
- Delivers data in GVAR format at a rate of 2.1Mbits/sec (~22GB/day)

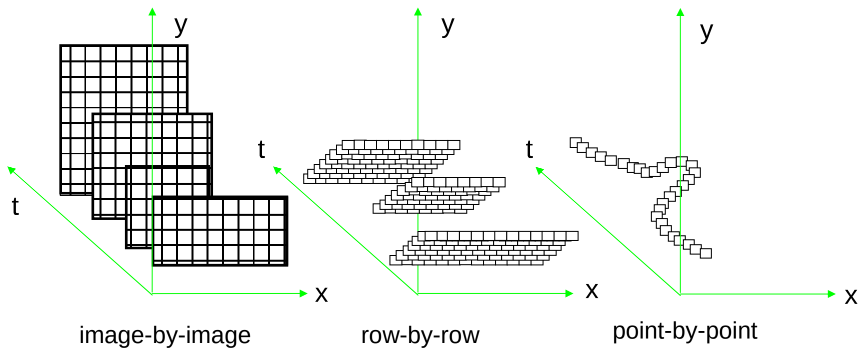


How does the stream data model look like?

- Infinite ***point set***, with points of the form (x,y,t), each point has ***point value*** x,y = spatial location of point (pixel), t = timestamp
- Point set is a ***topological space***, value set is a homogeneous algebra
- Spatial component of point set corresponds to a regularly space lattice and is ***geo-referenced*** (we are talking about field-based data...)

Streaming Geospatial Image Data (2)

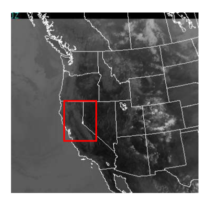
Different remote-sensing instruments deliver image data in different ways, e.g., as ***frames*** that cover different regions of the Earth's surface at different points in time.



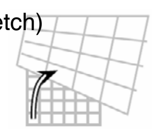
Consecutive points build stream of image data and points have close spatial and temporal proximity!

Streaming Geospatial Image Data (3)

Operations on stream of points (called *GeoStreams*)



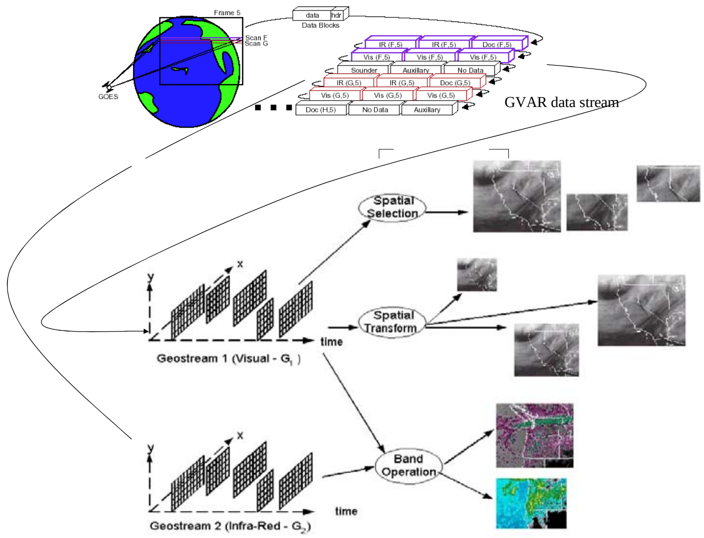
- Spatial, temporal, and value ***selections***
  - spatial selection allows to define regions of interest
  - all selection operations are non-blocking operations
- ***Stream transforms*** map a stream over a given lattice to a stream over another lattice
  - value transforms (e.g., from RGB to grayscale, contrast-stretch)
  - spatial transforms (e.g., zooming, map re-projections)



- ***Stream compositions***
  - A single stream corresponds to one spectral band, and has its own lattice (incl. spatial resolution and value set)
  - Compositions include image addition, subtraction, difference

Stream transforms and stream compositions are potentially blocking operations, e.g., contrast stretch has to read all pixel values of an "image"

Streaming Geospatial Image Data (4)



Summary so far...

- Science and applications specific data formats make perfect sense
- Data formats basically have two components (1) a metadata header, and (2) a big chunk of numerical (multi-dimensional) data
- Scientific data formats come with complete software solutions
- If XML is used, then it is merely a data exchange format
- File-based approach is almost exclusively the preferred data processing approach (data-intensive/compute-intensive)
- Many scientific data come in the form of streams

Team-up with scientists!

What is in for database folks?

- Study scientific data format, data properties, and applications
- Develop a stream model for respective scientific data, e.g., *streaming HDF5*, and study real-time application scenarios
- Algebraic framework (data types and operations) provides a good basis for studying optimization techniques for processing pipelines

References Module III

- Hierarchical Data Format (HDF). http://hdf.ncsa.uiuc.edu/
- HDF: the hierarchical data format. Brand Fortner, DR DOBB'S J SOFTWARE TOOLS PROF PROGRAM. Vol. 23, no. 5, pp. 42, 44-48. May 1998
- Common Data Format (CDF). http://cdf.gsfc.nasa.gov/
- A software package for the data-independent management of multidimensional data L.Treinish, M. Gough, EOS. Vol. 68, pp. 633-635. 14 July 1987
- Network Common Data Format (netCDF). http://www.unidata.ucar.edu/software/netcdf/
- NetCDF: an interface for scientific data access, R. Rew, G. Davis, Computer Graphics and Applications, Jul 1990, 10(4): 76-82
- Flexible Image Transport System (FITS). http://fits.gsfc.nasa.gov/
- FITS - a Flexible Image Transport System, D.C. Wells, E.W. Greisen, International Workshop on Image Processing in Astronomy. Proceedings of the 5th. Colloquium on Astrophysics, held in Trieste, Italy, June 4-8, 1979. Editors, G. Sedmak, M. Capaccioli, R.J. Allen.; Publisher, Osservatorio Astronomico di Trieste, Trieste, Italy, 1979
- eXtensible Scientific Interchange Language (XSIL). http://www.cacr.caltech.edu/SDA/xsil/
- A High-Performance Active Digital Library, Roy Williams, Bruce Sears, Proceedings of HPCN98, Amsterdam, April 1998, eds. L. O. Herzberger and P. M. A. Sloot, Lect. Notes Comp. Sci. (Springer)
- Geography Markup Language. http://www.opengeospatial.org/
- Deep Lens Survey. http://dls.bell-labs.com/
- GeoStreams project. http://geostreams.ucdavis.edu