Introduction to Querying Multivariate Ensembles

Jian Huang
University of Tennessee, Knoxville
SciDAC Institute of Ultrascale Visualization

Extreme Complexity → Ultrascale

- Scientific visualization faces problems more complex than ever before by orders of magnitude
  - Complexity: carbon, biogeochemical, evolution, coupling
  - Number of variables: >100
  - Temporal resolution + span: every 3 min, 1000 years (1.75e8)
  - Spatial resolution: 22km →1km
  - Size of ensemble runs : 50 → 1000

What can you show me?

- A critical gap:
  - What do you want to see?
  - Show me what ever you find then.

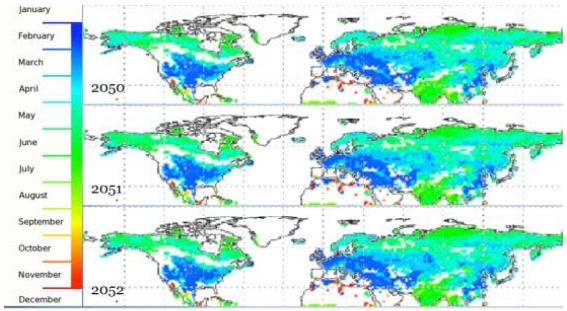
- Too many variables to look at side by side
- Too many time steps to examine one by one
- Too many models/run – to compare/contrast

What can you show me?

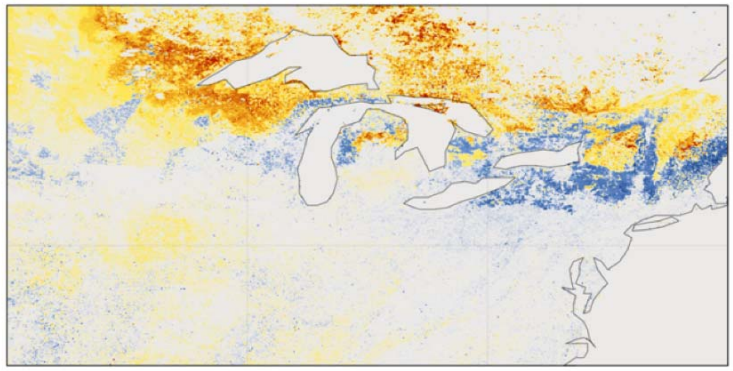
- A critical gap:
  - Often scientists know what they want to see
  - But cannot provide a formal quantitative description

User Concepts

- Qualitative user concepts:
  - When does the growing season start?
- Domain specific programming language methods
  - Specify events in an expressive, concise and powerful way
- Any persistent trends of event changes
  - Has the beginning of the growing season shifted in time in recent decades? How are different locations affected?



"Green-Up": Northern Hemisphere colored by month of event in variable ELAI
Query: ELAI:[-.4-.4]*T[.4-max]?*



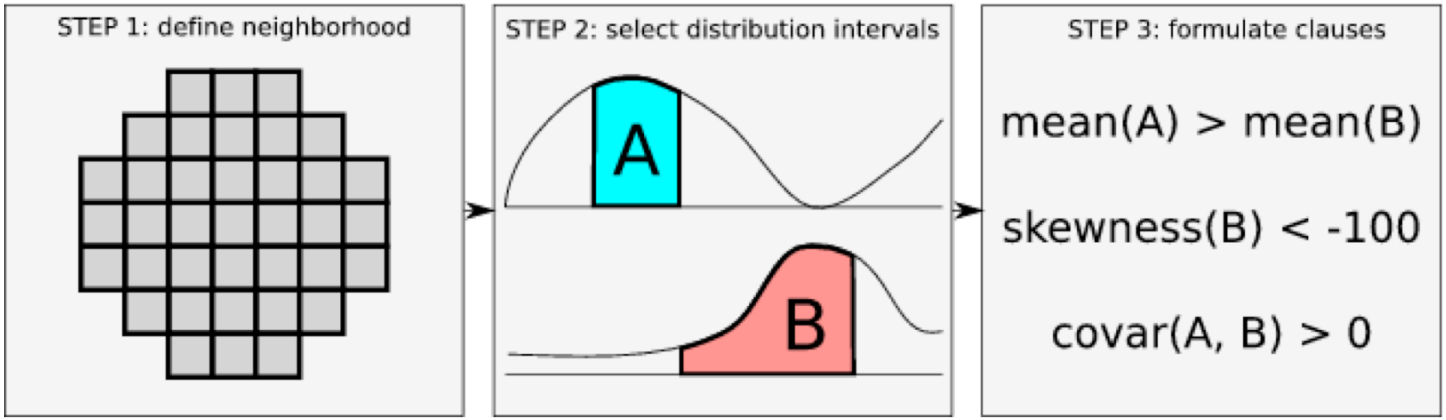
Concept-Driven Visualization

- As visual summaries, the benefits are:
  - Data reduction
  - Semantic meaning
  - Focus
  - Easily multivariate and temporal
  - Iteratively refined and recorded

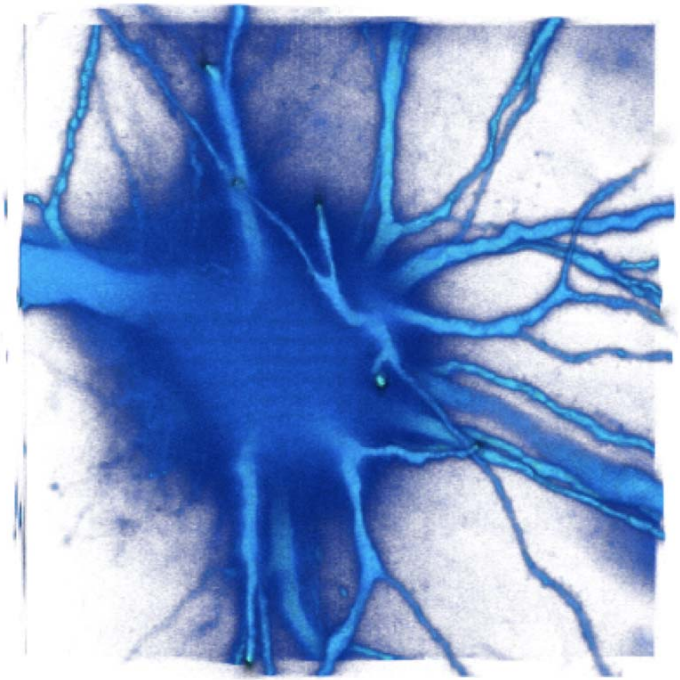
- Require infrastructural support:
  - Parallelism
  - Scalable data structures
  - Optimal use of parallel I/O

Relational Patterns in Local Distribution

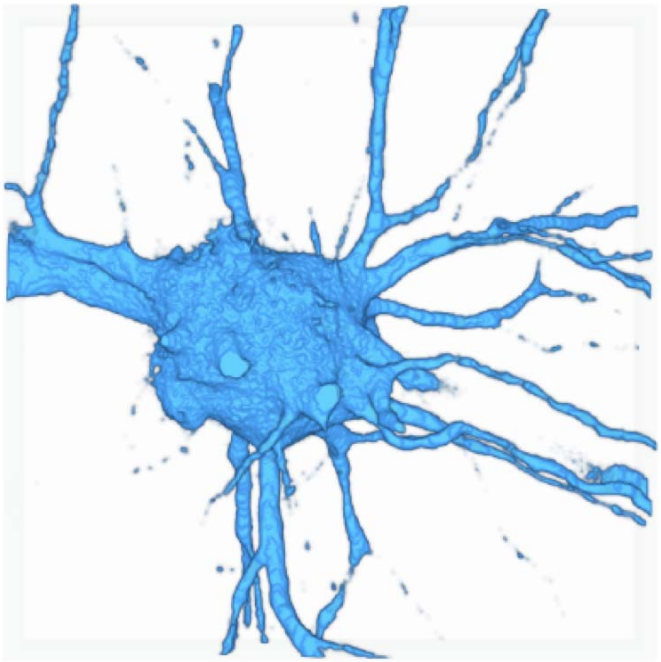
1. Define neighborhood
2. Establish relevant data ranges
3. Draw up clauses



Spatial Neighborhood Query

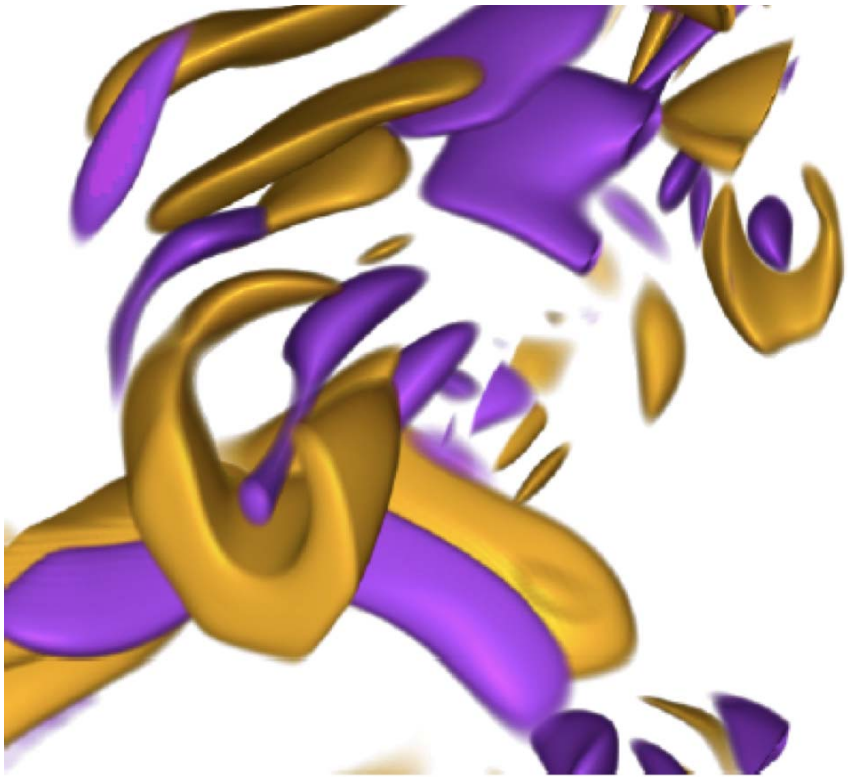


With typical 1D transfer function



Neighborhood Query
freq(nonbackground) > freq(background)

Temporal Neighborhood Query



Positive and negative covariance between two timesteps

Fuzzy Matching

We want to show locations that:

– match to a degree (score → opacity)

– match a subset of inequalities (combination → color)

Evaluating a Query

- For each location
  - For each clause
  - If TRUE

    score = 1

    tag bit set
  - Otherwise

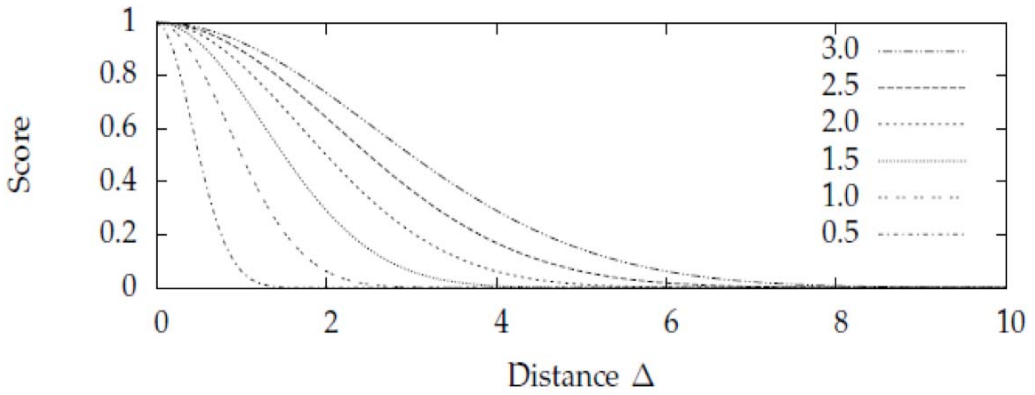
    score = f(distance) < 1

```
bin 0 {
   attribute 0
   range 0.9 to 1.0
}
bin 1 {
   attribute 1
   range < 0.5
}
bin 2 {
   attribute 4
   range >= 0.9
}
```

(a) Bins

```
clause 0 {
   freq(bin 0) > 50%
   weight 50%
   dropoff 0.1
}
clause 1 {
   var(bin 2) < var(bin 1)
   weight 25%
   dropoff 10
}
clause 2 {
   freq(bin 2) > 2 * freq(bin 1)
   weight 25%
   dropoff 10
}
```

(b) Clauses



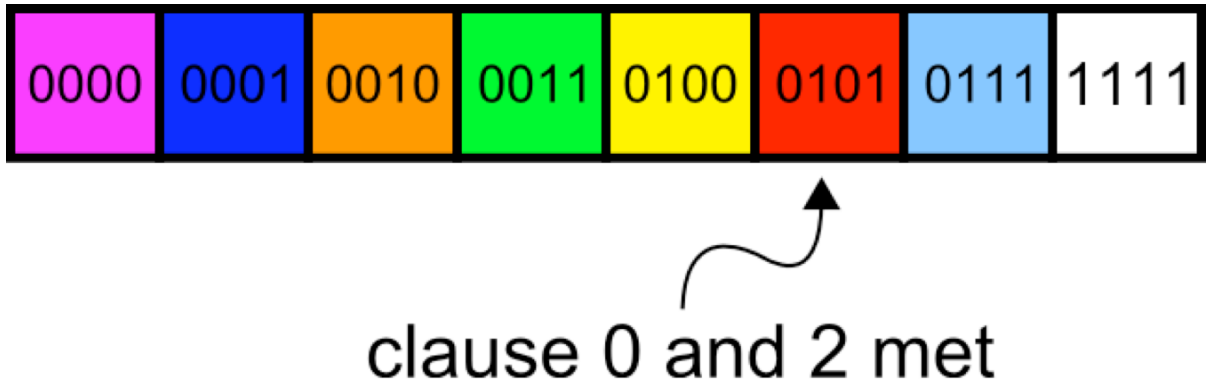
SciDAC Scientific Discovery through Advanced Computing

SciDAC Institute for Ultrascale Visualization

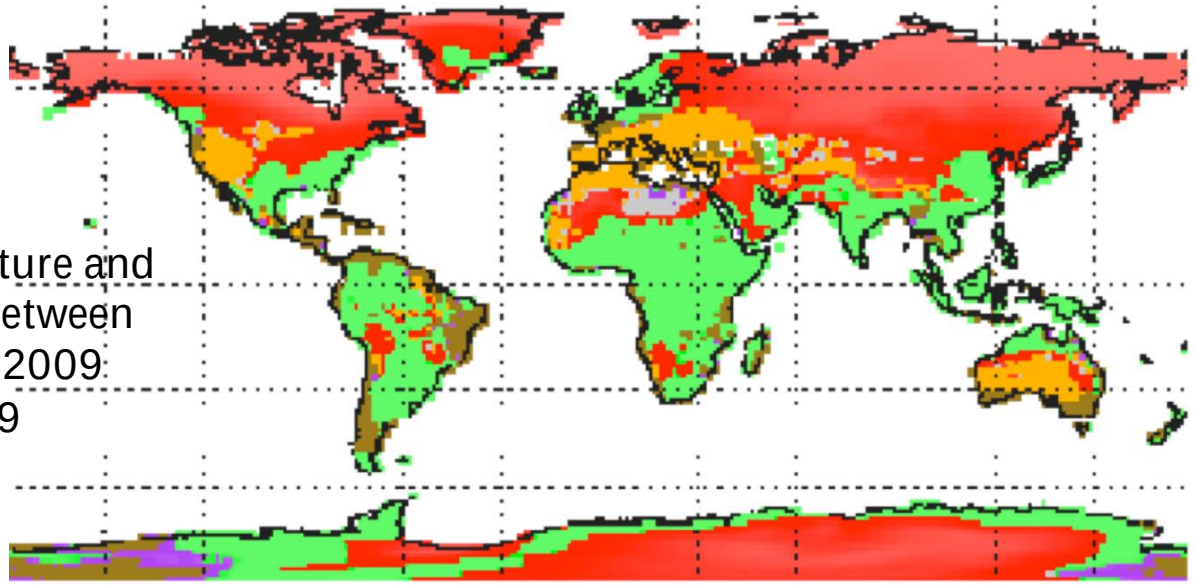
Oak Ridge National Laboratory

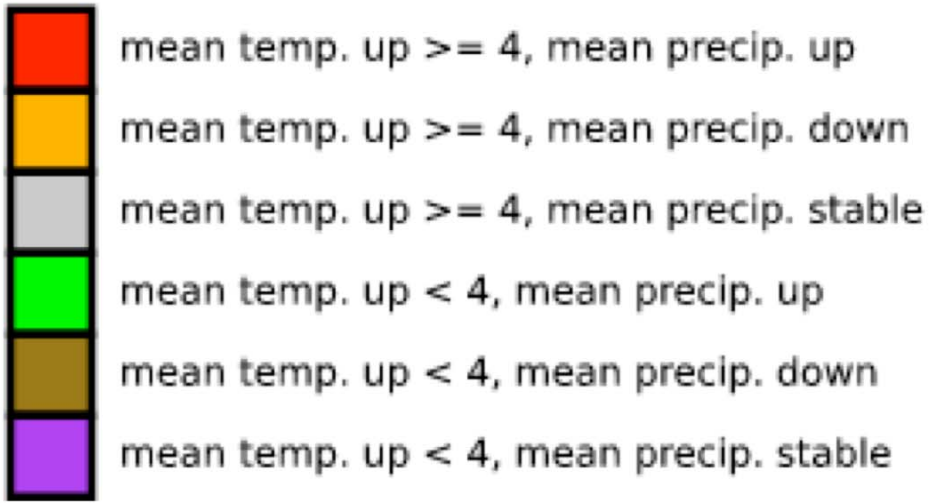
Visualizing a Query

- *All* locations scored and the rendered
- Sum of clause scores → opacity
- Clause bitfield → color
  - Bitfield indexes into colormap on the GPU
  - $2^{|clauses|}$ possible bitfield configurations



mean temperature and precipitation between decades 2000-2009 and 2090-2099





Specifying Temporal features

- User concepts about temporal events are often “story” like
- Uncertainty expressed via regular expression
  - *.mp3, %sale%, img[0-3][0-9].png
- Modeled after *regex*, but need to answers where and when an event occurs

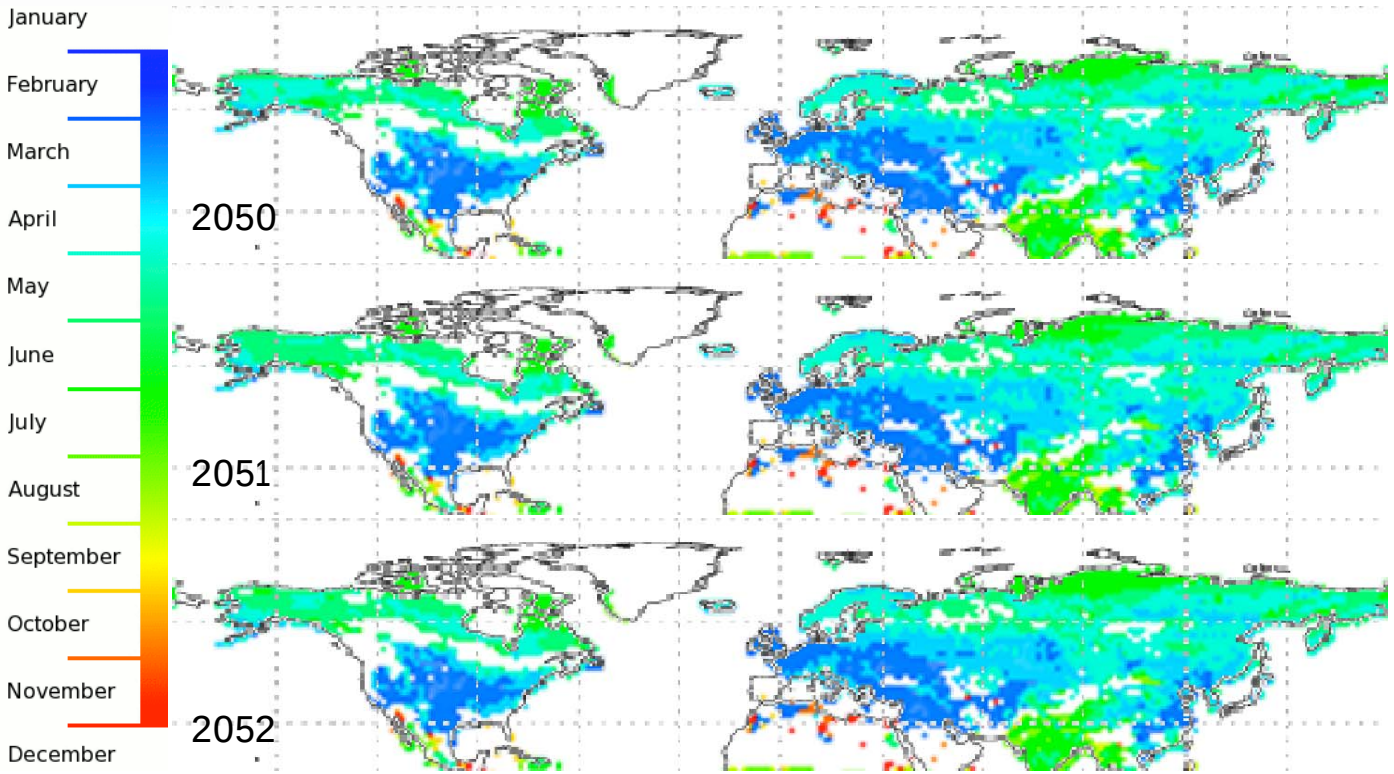
TimeMarks

For example: `[-.4,.4]*T[.4, max]?*`

– For each location, find time step T sandwiched between zero or more changes in [-40%, 40%] and at least one change of more than 40%

- T – TimeMark: when event occurs
- Automatic expansion into substantiated queries
- Combine primitives in time sequence

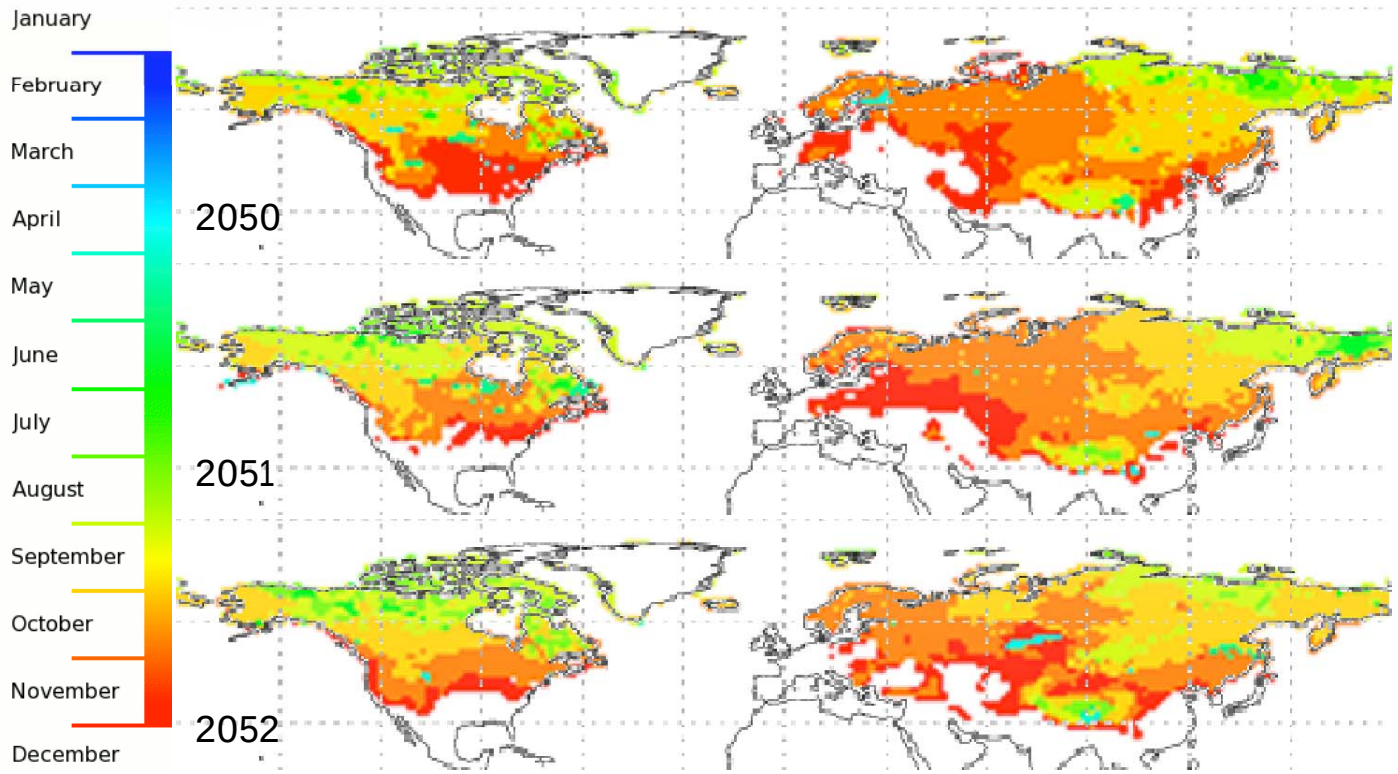
Meta-Queries



“Green-Up”: Northern Hemisphere colored by month of event in variable ELAI.

```
[-.4,.4]*T[.4, max]?*
```

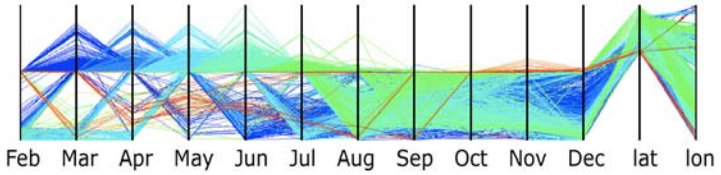
Meta-Queries



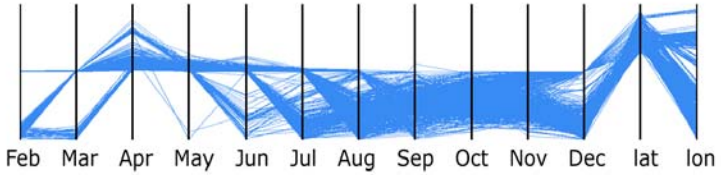
“First Snow”: Northern Hemisphere colored by month of event in variable FSNO

???[min, 0.7]*T[0.7, max]?*

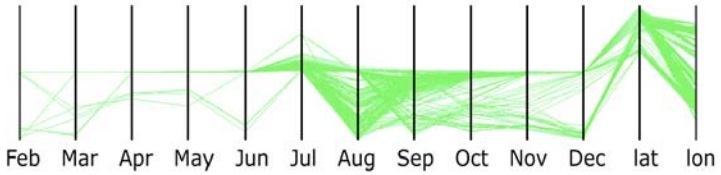
Another look in parallel coordinates



(a) Year 2050 of ELAI, all discrete queries

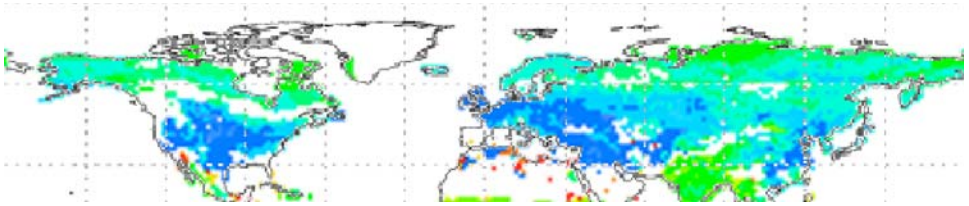


(b) Year 2050 of ELAI, discrete query #3



(c) Year 2050 of ELAI, discrete query #6

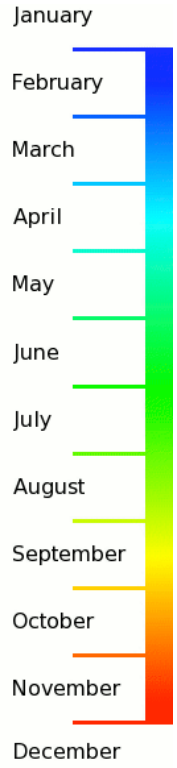
(d) Legend of discrete query numbers



"Green-Up" in 2050

Northern Hemisphere colored by month of event in variable ELAI:

[-0.4,.4]*T[.4, max]?*



SciDAC Scientific Discovery through Advanced Computing

SciDAC Institute for Ultrascale Visualization

Oak Ridge National Laboratory

Complexity of Meta Queries

- Many cases could lead to exponential problem spaces
- Fortunately, the data access patterns are not random (except in rare cases)

ID	Query	# Queries generated	# Data blocks accessed	# Voxels returned	Running time (secs)
1	[0-10][0]?[0-1e10]*	1	5	3	3.168
2	[0-99][0]?[0-1e10]*	1	41	3	26.522
3	[40-60]??[0-1e10]*	1	10	342	6.067
4	[50]??*?*	71	1	3,615,888	7.454
5	[70]?[-5-1e10]*[5--1e10]*	71	1	6,584	7.485
6	[0-20]??[0-1e10]*?*	71	10	3,593,696	159.97
7	[80]?[-100-1e10]*[100--1e10]*[-100-1e10]*	2,556	1	16,994,091	248.87
8	[80-82]?[-100-1e10]*[100--1e10]*[-100-1e10]*	2,556	3	50,565,859	757
9	[0-99]?[-100-1e10]*[100--1e10]*[-100-1e10]*	2,556	≈ 117,500	≈ 1,685,529,000	≈ 72,500

Concept-Driven Visualization

- As visual summaries, the benefits are:
  - Data reduction
  - Semantic meaning
  - Focus
  - Easily multivariate and temporal
  - Iteratively refined and recorded

- Require infrastructural support:
  - Parallelism
  - Scalable data structures
  - Optimal use of parallel I/O

Backend Technical Requirements

- Underlying data structures and management need to be optimized for common data types in scientific research.
  - Time-varying, multi-dimensional, multi-variate, potentially non-uniform grids.
- Data management systems (DMS) for massive data sets must ...
  - incur small storage costs,
  - provide ad hoc query support,
  - exhibit reasonable latency and throughput performance.
- Implications of these requirements are ...
  - no unnecessary data duplication,
  - a transparent, self-explanatory query structure,
  - use of sophisticated underlying data structures and algorithms.

Backend Technical Requirements

- Simplistic queries are not sufficient to describe features / subsets.
- Many features can generally be described as local events, i.e. spatially and temporally limited regions with characteristic properties in value space.
- Scientists know what they are looking for in their data, but may be unable to formally or definitively describe their concept, especially when based on partially substantiated knowledge.
- Scientists need to query and extract such features or events directly without having to rewrite their hypothesis into an inadequately simple query language.
- A more sophisticated feature-oriented query language is required.

Related Work

- Large data management in visualization
  - Data partitioning (blocks, “bricks”)
  - Efficient searching using tree-based data structures:
    - Interval tree, k-d tree quad-tree, octree, etc.
    - Bitmap indexing
  - Relational Database Management Systems (RDBMS)
- (Programming) languages in visualization:
  - More versatile and flexible compared to GUIs.
  - Alter GPU shader programs on the fly: “Scout”
  - VTK provides Tcl/Tk and Python bindings.

Data Organization

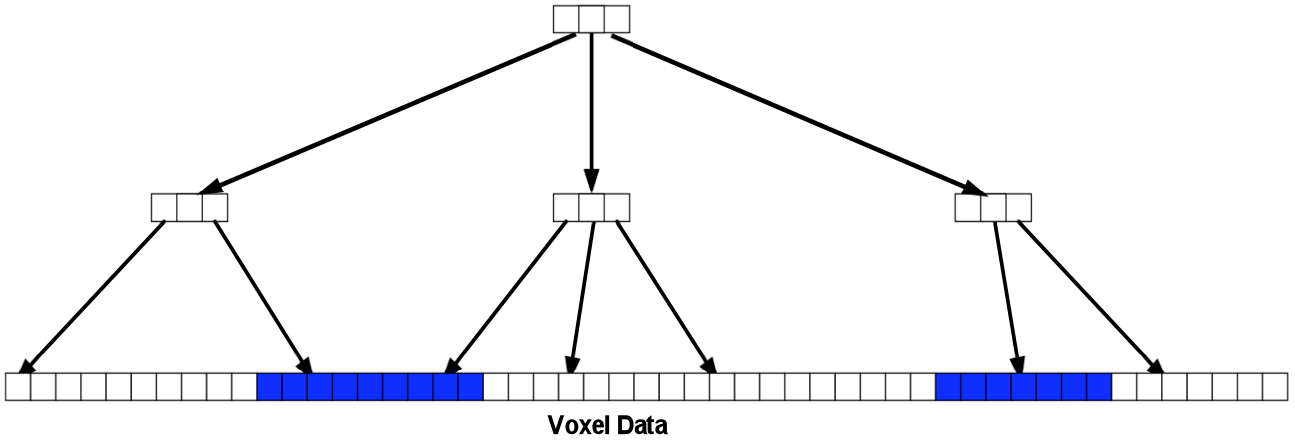
- Large data sets need to be partitioned for data distribution and load-balancing.
- Break up data set into data items containing
  - spatial and temporal location (x,y,z,t),
  - a value for each data variable.

e.g. `{x=1; y=2; z=3; t=10; density=2.7; entropy=.7}`

- Implications
  - Yields increase in total data size!
  - Number of data items can be enormous!
  - **But**: Load-balancing can be applied on the level of data items.

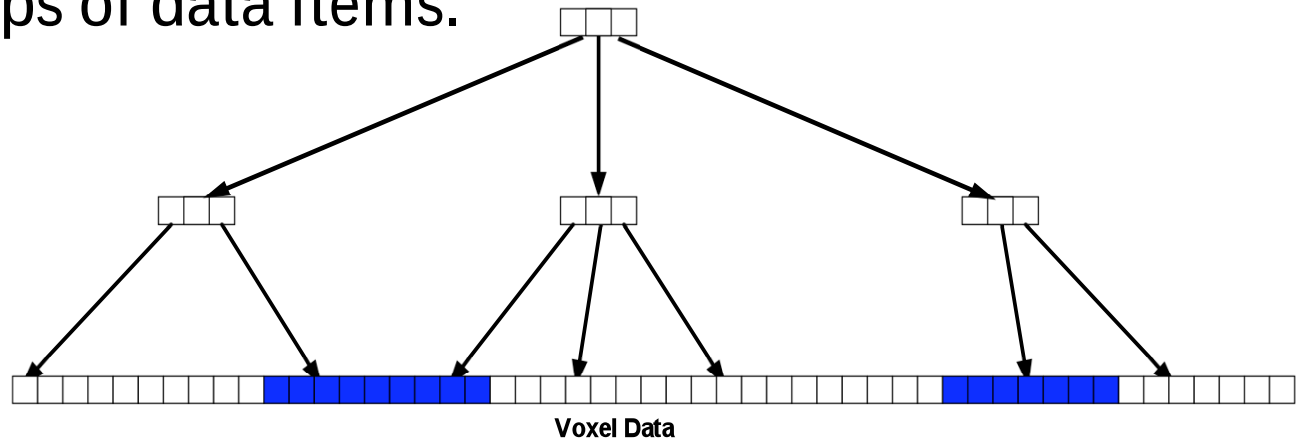
Query-Driven Visualization

- **Load-balancing** by breaking up locality within the dataset.
- Optimized data access by using **a B-tree like structure** to skip irrelevant data items on top of a linear search.
- **Discard unwanted data items** upon distribution (data items are independent of any structural meta-information)
- **Compress blocks of data items** to trade memory space vs. access time, decompress on access.



Data Selection

- Each data server hosts a portion of the data set as data items in a *sorted* list.
- On top, a complete M-ary search tree of depth N << M (e.g. M = 256, N = 3) indexes into the list of data items.
- Search: Find first matching data item and initialize a *linear search* from it. Use search tree to skip irrelevant groups of data items.



Enhancements

- Observation: Data items are independent of any structural meta-information (e.g. a grid).
  - Unwanted data items can be deleted before distribution to data servers.
  - This counterbalances the increase of data set size.

- Compress the linear list of data items.
  - Trade-off: memory space vs. access time
  - Blocks of data items are decompressed on the fly.
  - Since linear list is sorted, high compression rates (20:1) are possible in many cases.

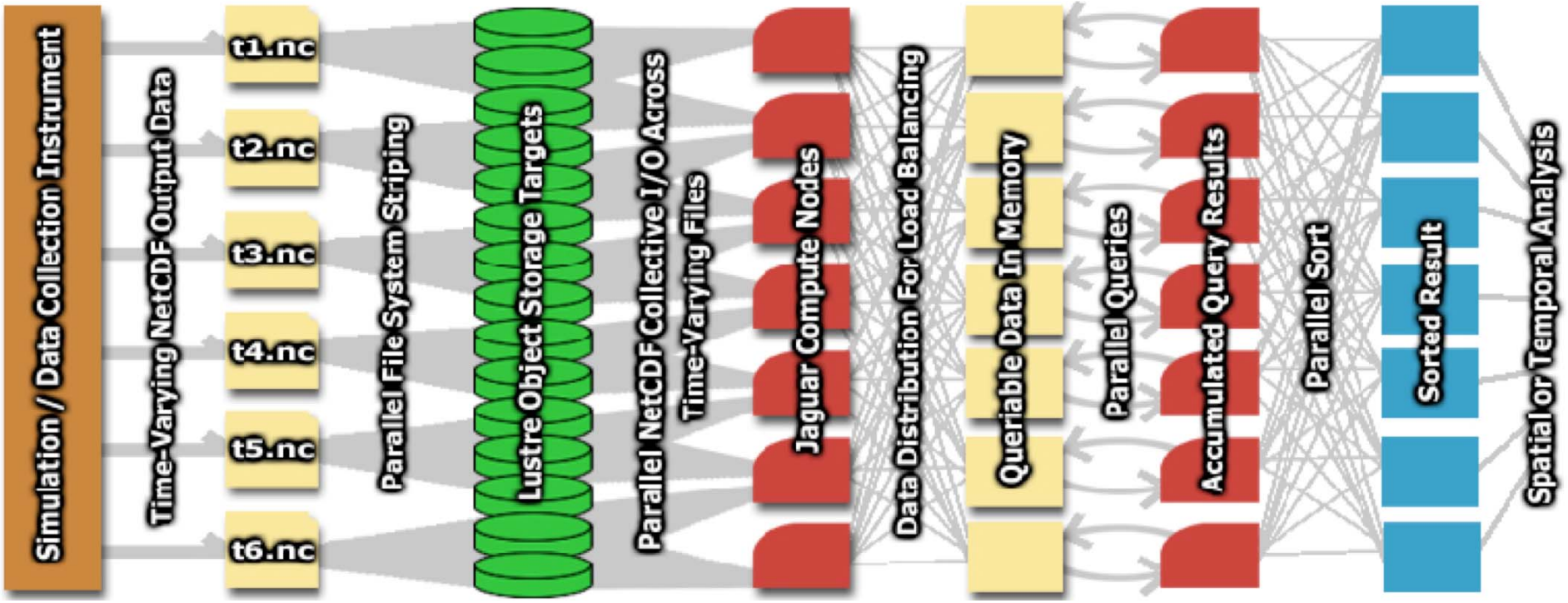
Scalability Tests: the data

- NASA's Moderate Resolution Imaging Spectro-radiometer (MODIS) database, continuously updated
- Use 417 timesteps, 8-day interval, 02/2000 to 02/2009
- 500 meter resolution sampling of North and South America, creating a 31,200x21,600 grid
- Compute variables from 7 wavelength bands
- Use MRT toolkit to reproject from sinusoidal grid to equirectangular grid
- Total data used for scalability tests amount to 1.1TB

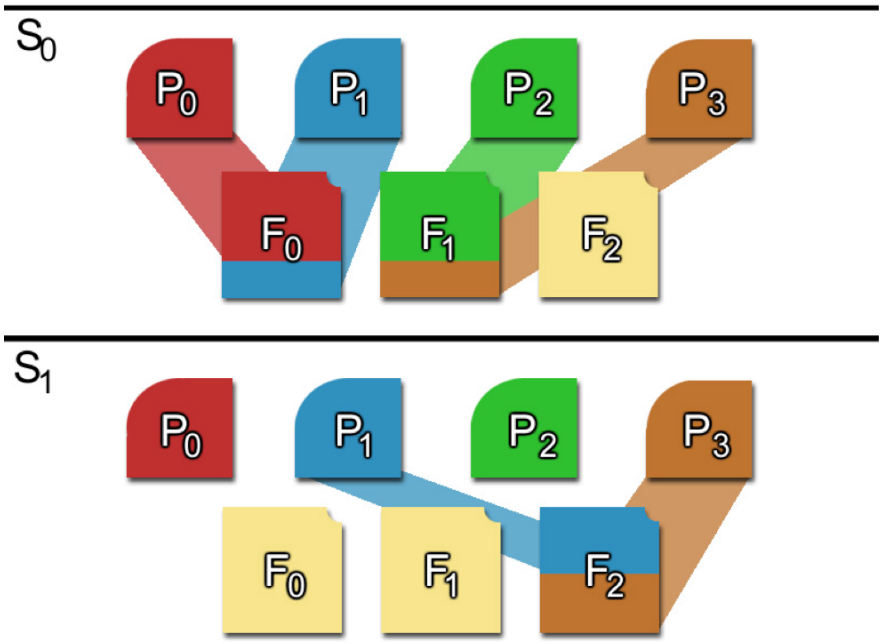
Scalability Tests: the machine

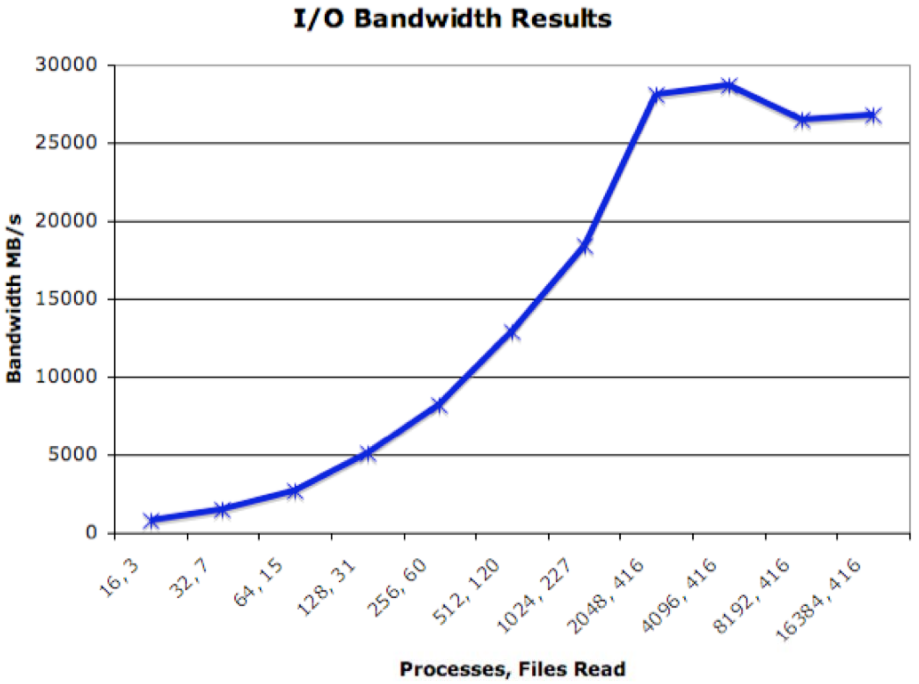
- Jaguar, ORNL
- Cray XT4 consisting of 7,832 quad-core 2.1 GHz AMD Opteron processors with 8 GB of memory.
- 31,328 cores with over 60 TB of main memory.
- Lustre parallel file system. One meta data server (MDS), 72 OSSs (I/O nodes), 144 OSTs (physical disk systems)

Infrastructural Diagram

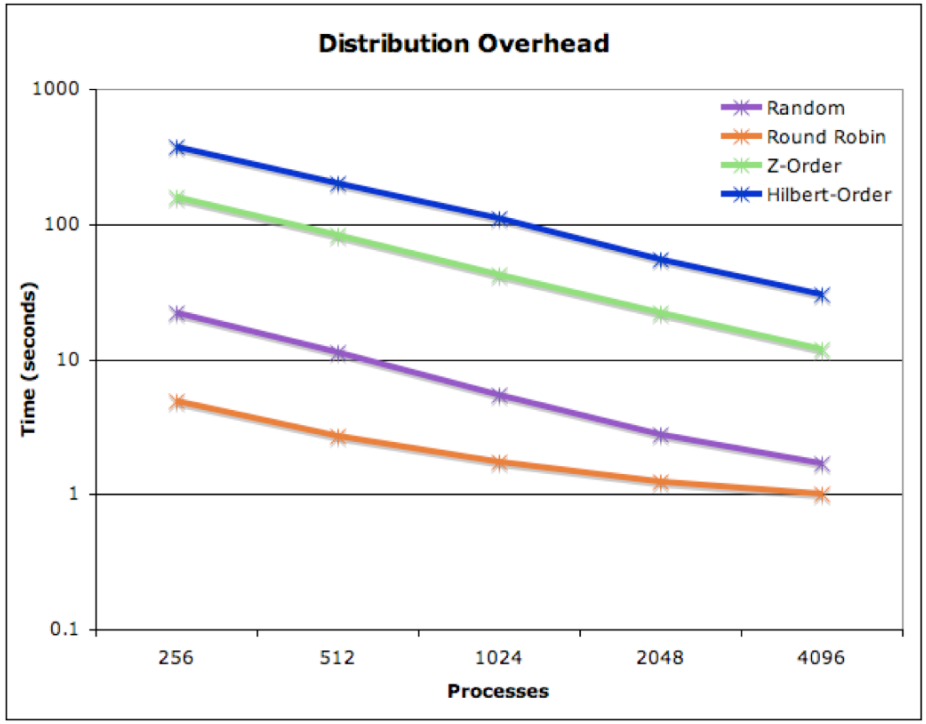


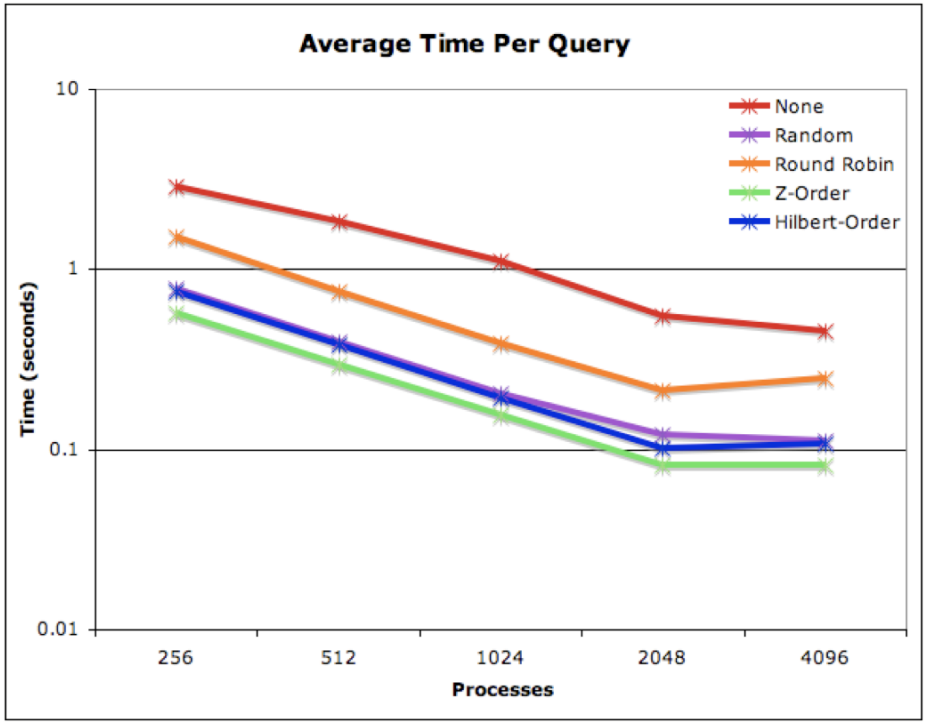
Measured I/O Bandwidth





Distribution and Query Overheads





Climate Science - Drought Assessment

- NDVI and NDWI can be good indicators of drought, and NDDI (Normalized Difference Drought Index) can be computed by

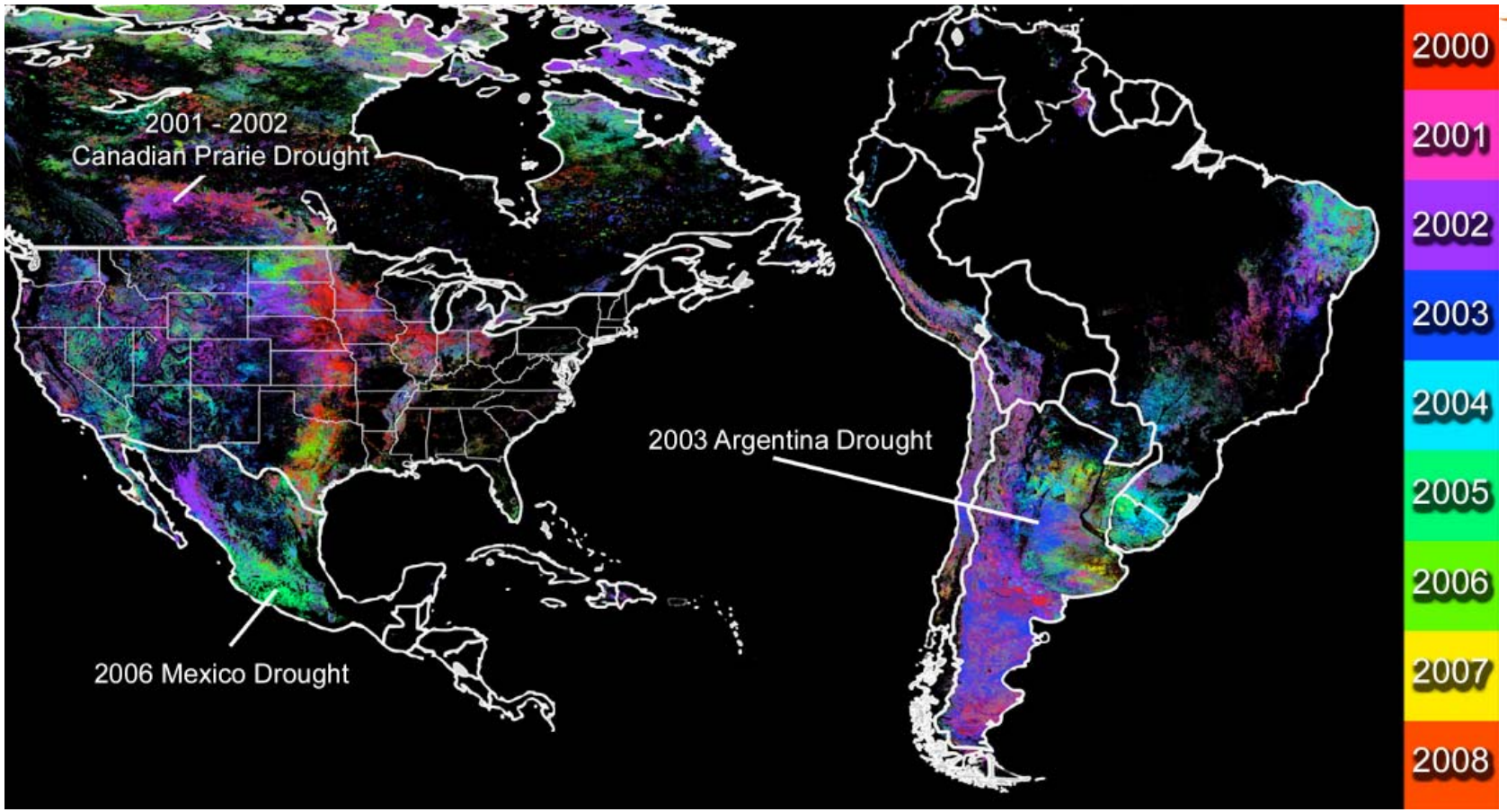
  NDDI = (NDVI - NDWI) / (NDVI + NDWI)

- We use a similar method of drought assessment by querying for:

  NDVI < 0.5 and NDWI < 0.3

- After queries are issued, result is sorted in spatial order
- Temporal overlap can then be computed
  - look for 0.5 < NDDI < 1 for at least 4 timesteps (1 month) in a row
- We also placed one more restriction and throw out the areas where the event happened more than one time, finding only the areas where abnormal drought conditions occur

Drought Assessment



Processes	Read	Filter	Redistribution	Query	Sort	Analysis	Write	Total Time
4096	60.12	6.15	12.24	2.25	2.26	0.35	4.84	85.44
8192	47.28	3.66	9.76	1.46	3.09	0.38	4.5	70.13
16384	50.35	1.82	7.01	2.13	4.01	0.45	7.0	72.77

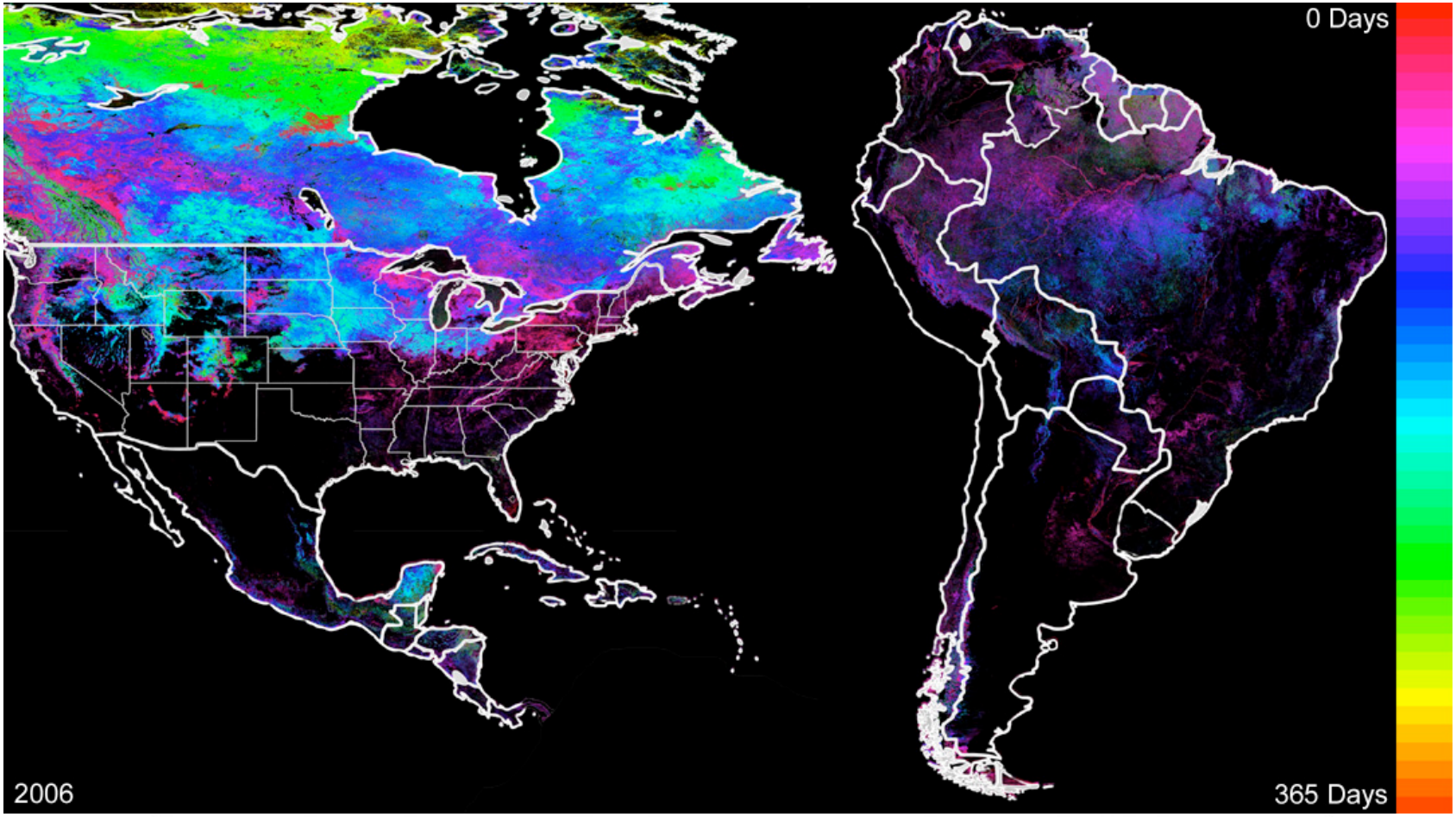
Climate Science: Time Lag Analysis

- Studying time lag is important for obtaining a better understanding of how variables like NDVI are affected by other conditions
- Studying past and present droughts in relation to these conditions could enhance the capability to develop early warning systems
- In our example, we compute the time lag between when NDWI first occurs in the 0.7 - 0.9 range (first snow) and when NDVI first occurs in the 0.4 - 0.6 range (vegetation green up)

example of time lag of first snowfall and vegetation green up

SciDAC Scientific Discovery through Advanced Computing
SciDAC Institute for Ultrascale Visualization
Oak Ridge National Laboratory

Time Lag Analysis



Processes	Read	Filter	Redistribution	Query	Sort	Analysis	Write	Total Time
4096	50.12	6.16	12.46	0.40	0.40	0.10	3.66	73.30
8192	67.28	3.70	10.00	0.20	1.21	0.14	4.51	87.03
16384	52.35	1.93	7.13	0.04	3.41	0.26	7.00	72.11

SciDAC Scientific Discovery through Advanced Computing
UltraVis SciDAC Institute for Ultrascale Visualization
ornl Oak Ridge National Laboratory

Conclusion

- Creating single images as summarizing visualization of a high-level event to study climate change
- Feature specification that empowers “eye-balling” should be studied in depth, in addition to feature extraction and rendering
- Programming language type of methods have offered encouraging results
- It is crucial to have truly scalable parallel infrastructure for visualizing terascale data and beyond

References

- Wesley Kendall, Markus Glatter, Jian Huang, Tom Peterka, Robert Latham and Robert Ross, *Terascale Data Organization for Discovering Multivariate Climatic Trends*, SC'09, November 2009, Portland, OR.

- C. Ryan Johnson and Jian Huang, *Distribution Driven Visualization of Volume Data*, IEEE Transactions on Visualization and Computer Graphics, 15(5):734-746, 2009.

- Markus Glatter, Jian Huang, Sean Ahern, Jamison Daniel, and Aidong Lu, *Visualizing Temporal Patterns in Large Multivariate Data using Textual Pattern Matching*, IEEE Transactions on Visualization and Computer Graphics, 14(6):1467-1474, 2008.

- Markus Glatter, Colin Mollenhour, Jian Huang, and Jinzhu Gao, *Scalable Data Servers for Large Multivariate Volume Visualization,* IEEE Transactions on Visualization and Computer Graphics, 12(5):1291-1299, 2006.

Acknowledgements

- Current Students:
  - Wesley Kendall
- Graduated students:
  - Dr. Markus Glatter, Dr. C. Ryan Johnson, Dr. Rob Sisneros, Dr. Josh New
  - Brandon Langley, Colin Mollenhour
- Collaborators DOE SciDAC Ultravis Institute (www.ultravis.org)
  - Rob Ross, Kwan-Liu Ma, Han-Wei Shen, Ken Moreland, John Owens
- Collaborators at Oak Ridge National Laborartory
  - Sean Ahern, Forrest Hoffman, David Erickson.

- Our funding were provided by DOE SciDAC, DOE Early Career PI Award, NSF ACI and CNS programs.