An Environment for Systemizing Data Preparation and Machine/ Deep Learning

K-S Kuo, A O Oloso, M L Rilee, T L Clune,
K Doan, H Lin, X Li

Ultimate Goal

An *efficient* and *cost-effective* environment that

- Addresses Big Data challenges;
  - Volume, Variety, Velocity, Veracity, **Value**
- Alleviates scientists from undue data-management burdens associated with data analysis;
- Automates data preparation to systemize machine/deep learning exercises.

Existing State of Business

Existing *file-centric practice* causes

- Massive data download
- Enormous duplication in resource consumption
- Individual analysis algorithm implementation
- Diffusive provenance collection

(Moving data stores to Cloud without changing the file-centric practice only makes it marginally better!)

Variety Scaling is the Key

- While most efforts focus on scaling *volume*, we believe scaling *variety* is the key to better value!
- Earth system science requires *integrative analyses*.
  - Interdisciplinary, collaborative investigations
  - Interoperability in data (i.e. fusion) and processing
- Earth Science data are very heterogeneous!
  - Various geometries and resolutions
  - Good **volume** scaling can be achieved relatively easily with homogeneous data (or a small variety).
  - Good **variety** scaling is needed to achieve optimal value for *integrative analyses*.

Where are the *pain* points?

- Massive volumes of download.
- Need to overcome the "file" barrier.
- Homogenization of the variety.
- Sparsity of parallelization expertise among scientists.

Why can't we scientists compare geophysical quantities obtained by different means like comparing prices of, say, hard drives online?

Solution Technologies/Techniques

- Transition to a *data-centric practice*
  - Access and analyze data directly on server side.
- Unified data representation
  - 1st step in homogenizing variety
- Regridding/Remapping
  - 2nd step in homogenizing variety
- Connected component labeling (CCL)
  - Enable event-based analysis
- Moving Object Database (MODB) capabilities

Data-centric Technologies

- **SciDB** – a distributed, parallel database management system (DBMS) based on array data model.
  - Our technology of choice.
  - Tight coupling between compute and storage.
    - More efficiently than the loosely coupled approaches (Doan et al 2016, *IEEE Big Data*).
  - Leveraging both OpenMP and MPI underneath its own API programming model.
  - A suite of statistical functions/operators built-in! (Think: MS Access or *Excel* on steroid!)
- Spark and Hadoop (MapReduce)
  - Loose coupling between compute and storage.
    - More elastic than the tightly coupled approaches.

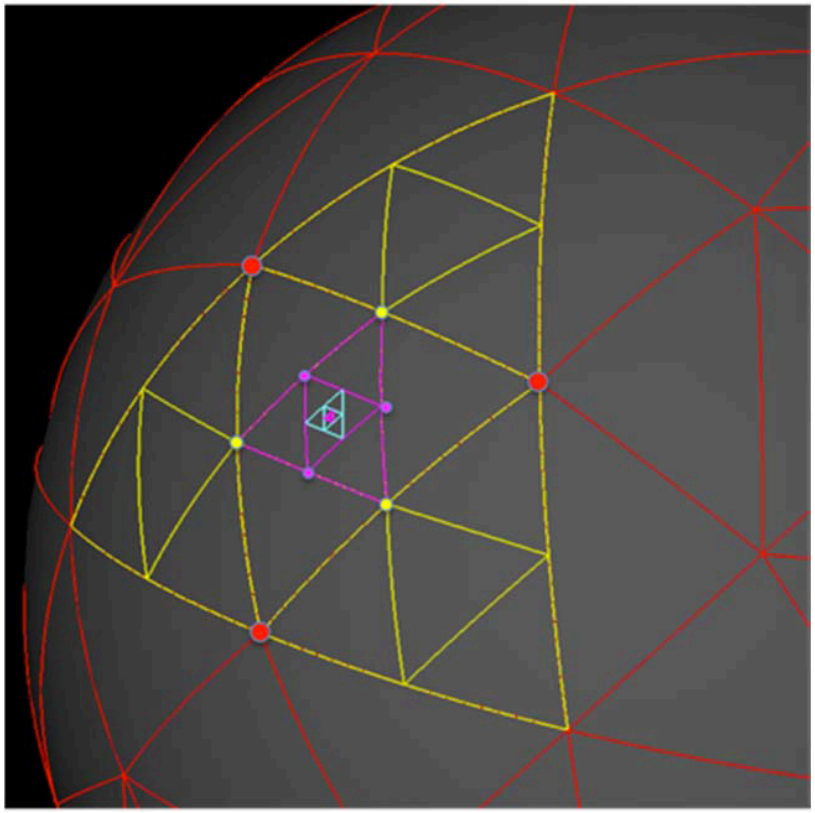
Unified Data Representation

- *SpatioTemporal Adaptive-Resolution Encoding*, STARE
  - Spatial indexing based on Hierarchical Triangular Mesh
    - Quad tree – *adaptive resolution* achieved with noting the quadfurcation level.
    - Each edge of a spherical triangle is a segment of a great circle; think: ***fast*** identification of region membership.
    - Our customized implementation turns spatial set operations into integer interval operations; think: ***fast*** too!
  - Similar indexing in time, i.e. hierarchical (but no binary tree).
- Facilitates *data placement alignment*
  - Same spatiotemporal subsets from all datasets (arrays) reside on the same nodes.
  - Analyses requiring spatiotemporal coincidence (which is a lot) remain embarrassingly parallel; think: ***fast*** again!
- Supports Moving Object Database (MODB) operations
  - E.g. whether an event of phenomenon *A* runs into an event of phenomenon *B* and, if so, where and when?
  - (Event identification and tracking was done previously with an implementation of CCL.)

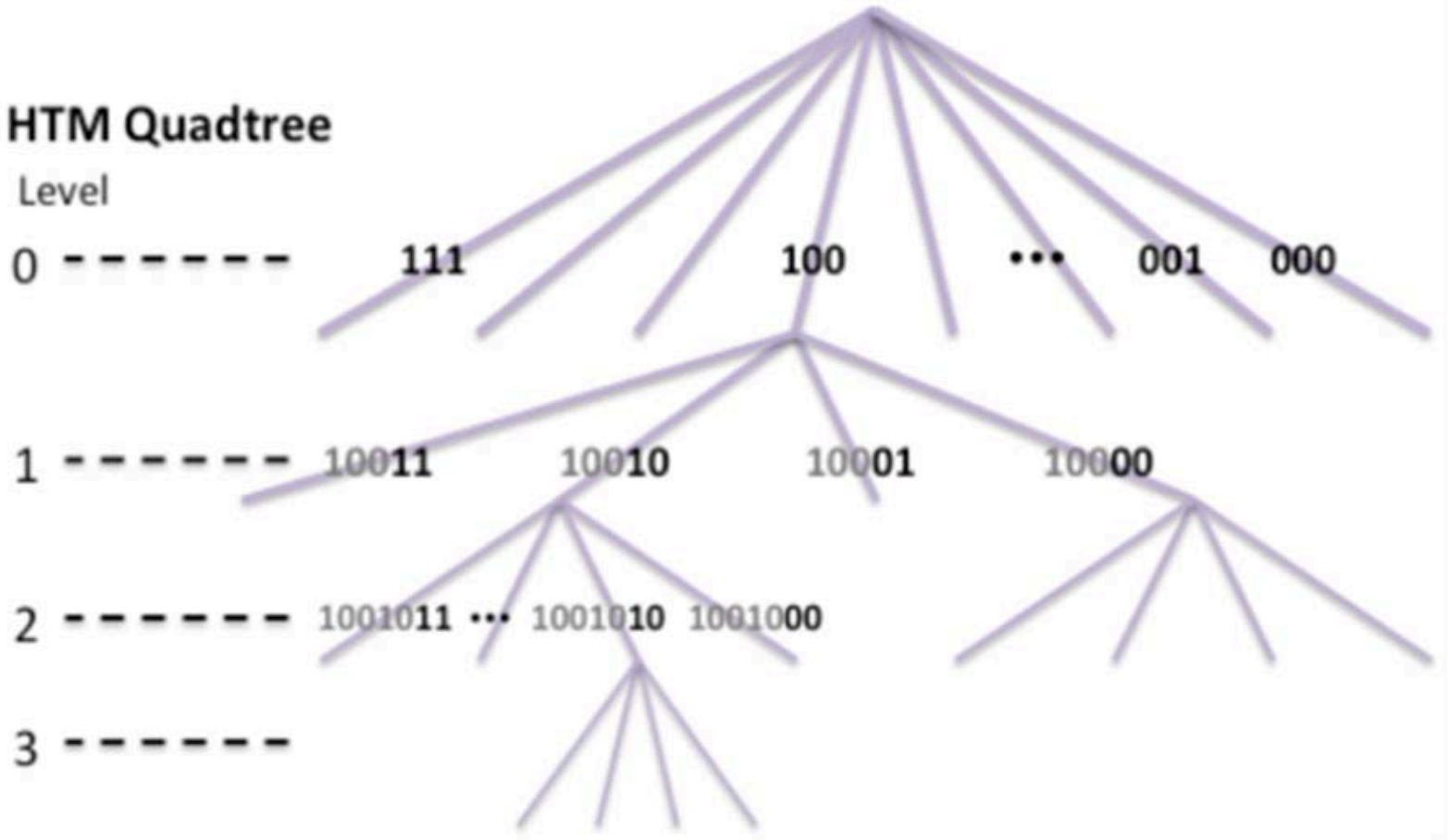
Hierarchical Triangular Mesh – HTM

HTM is a way to index/ address the surface of a sphere using a hierarchy of spherical triangles.

1. Start with an inscribing octahedron of a sphere.
2. Bisect each edge.
3. Bring the bisecting points to inscribe the sphere to form 4 smaller spherical triangles.
4. Repeat



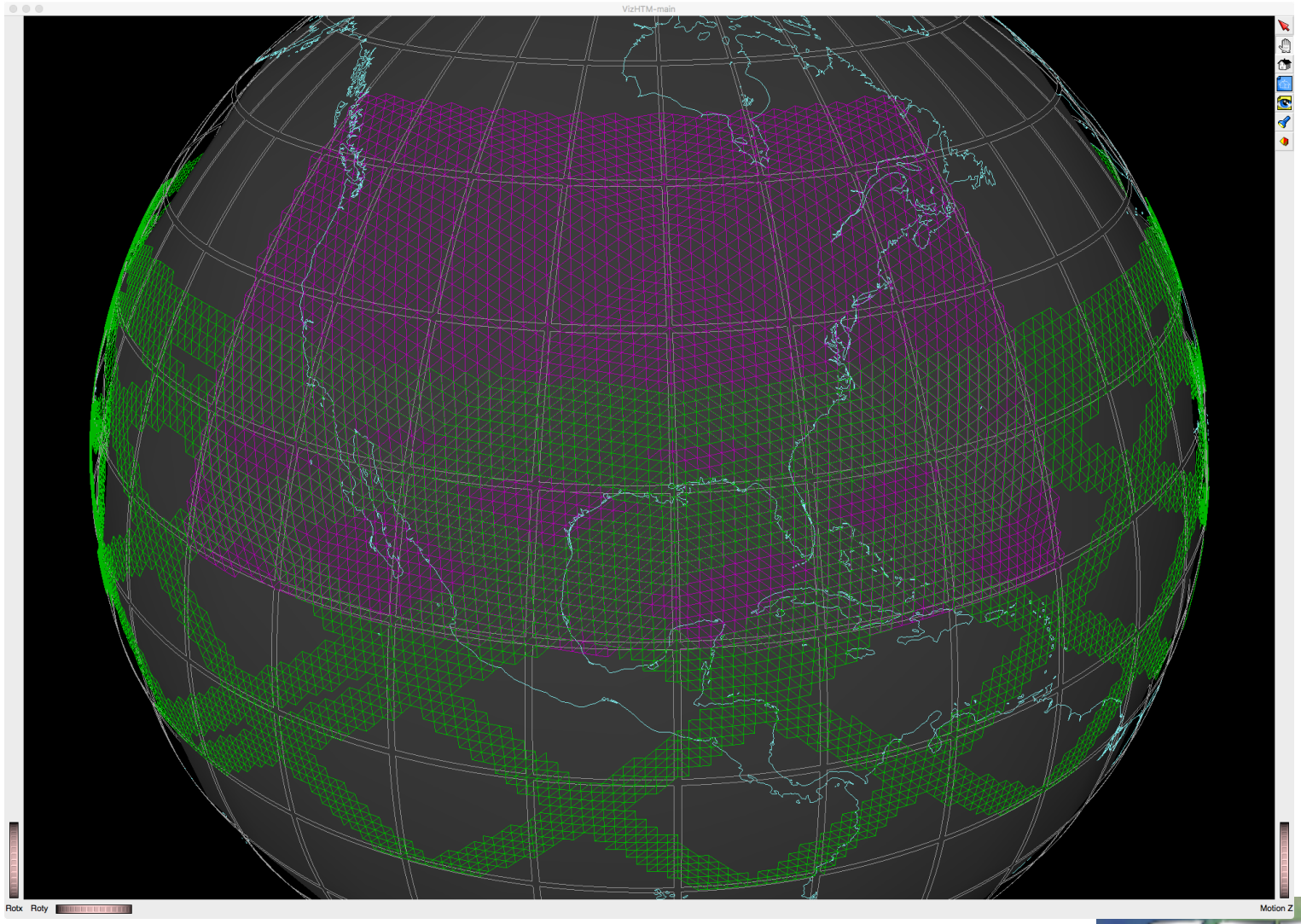
HTM Quad-tree



HTM Geospatial Indexing

- At quadfurcation depth (level) 23, the linear resolution reaches ~1 meter.
  - Each latitude-longitude coordinate is assigned an HTM index with an uncertainty of ~±0.5 meter.
  - Sufficient to index the great majority of Earth Science remote sensing data.
  - Our customized implementation carries the quadfurcation *depth level* to denote approximate resolution of the data.
- Applicable to Grid, Swath, and Point data types.
  - In fact, it is applicable to all geospatial data.

NMQ and TRMM



Temporal Indexing

- Hierarchical as HTM (but not "tree-like")
  - Bit ranges (from LSB to MSB) and their meanings

Range	Starting Bit	Ending Bit	No. Bits	Denoting
0	0	2	3	*Resolution*
1	3	12	10	millisecond
2	13	24	12	Second
3	25	29	5	Hour
4	30	32	3	Day of week
5	33	34	2	Week
6	35	38	4	Month
7	39	48	10	Year
8	49	58	10	Kilo-annum
9	59	62	4	Mega-annum
10	63	63	1	Before/After

STARE

- **S**patio**T**emporal **A**daptive-**R**esolution **E**ncoding.
  - Implemented in SciDB
  - It can be used independently as an indexing scheme, i.e. standalone (modifications are likely needed for optimization)
- Consistent indexing of all dataset arrays greatly simplifies and speeds up integrative analyses!
  - *Flexible subsetting*: spatial (including irregular regions of interest, ROIs), temporal, and/or spatiotemporal
  - *Conditional subsetting*: using query result from one dataset array to subset another dataset array
    - E.g. Wind speed (temperature, pressure *etc.*) for where there is rain.
  - Efficient geo-spatiotemporal set operations.

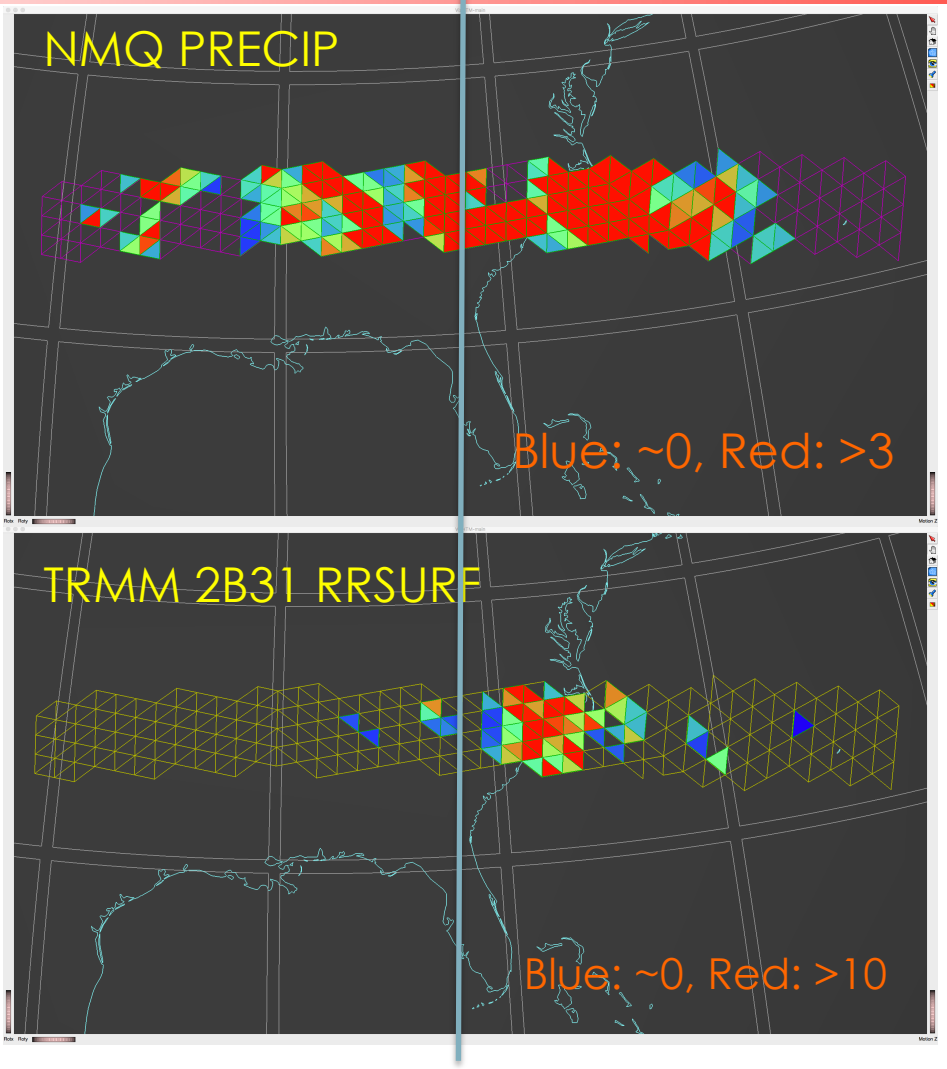
Regridding/Remapping

- *Comparison*, a ubiquitous kind of integrative analysis
  - Observations with observations, observations with model output, and model output with model output…
  - Differences in geometry and resolution complicate comparisons.
  - Requires more than spatiotemporal coincidence.
  - If we cannot do *comparisons* conveniently and efficiently, there is no hope for more sophisticated analyses.
- Two types of regridding implemented
  - Inverse distance weighting (IDW)
  - Flux conservative
  - (We have gained the skill to systematically implement other regridding methods in Climate Data Operators, CDO.)

2009 December 3 0300 | 2009 December 3 0305

These data were obtained by performing a "join" operation based on STARE spatiotemporal indexing in SciDB.

Full resolution data was regridded to a lower resolution for clarity and convenience. Values shown are the actual maximum values within the geographic trixel.



Corresponding TRMM and NMQ data elements share temporal and spatial indexes.

Level 7 trixels (~78km) are used for clarity. Native STARE resolutions are level 11 for TRMM and level 13 for NMQ.

Temporal ID
~5 minute resolution → 1131260126691328 1131260128788480

SciDB Query

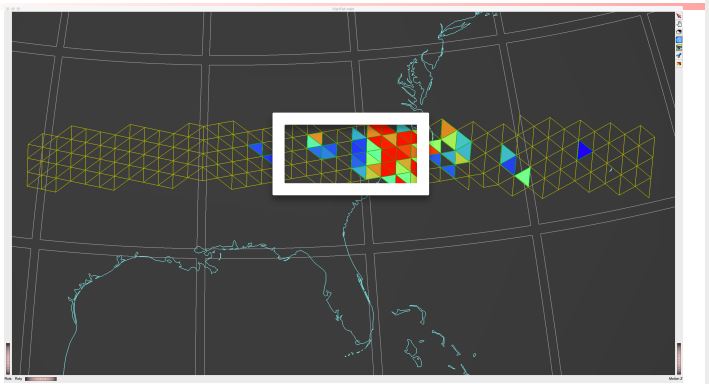
- To spatiotemporally "join" the two datasets with STARE indexing (5-min at level 7, i.e. ~78-km resolution):

```
join( nmq_precip, trmm1_2B31 );     - Magic!
```

- To subset temporally for visualization:

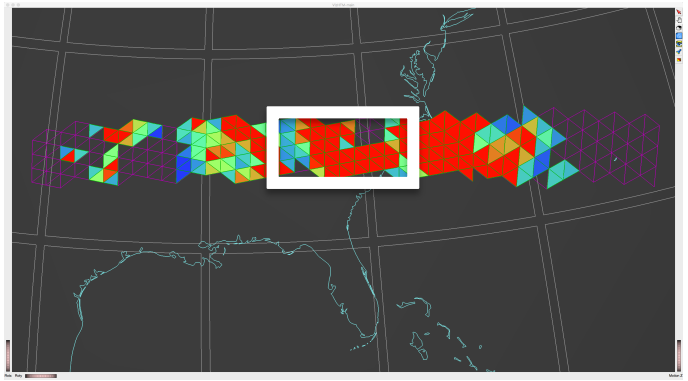
```
select *
  into nmq_trmm_09120303
  from nmq_trmm1_result        - result from previous join
  where
    tIndex= temporalIndexFromString("2009-11-03 03:00:00.000 (00)") or
    tIndex= temporalIndexFromString("2009-11-03 03:04:16.000 (00)");
```

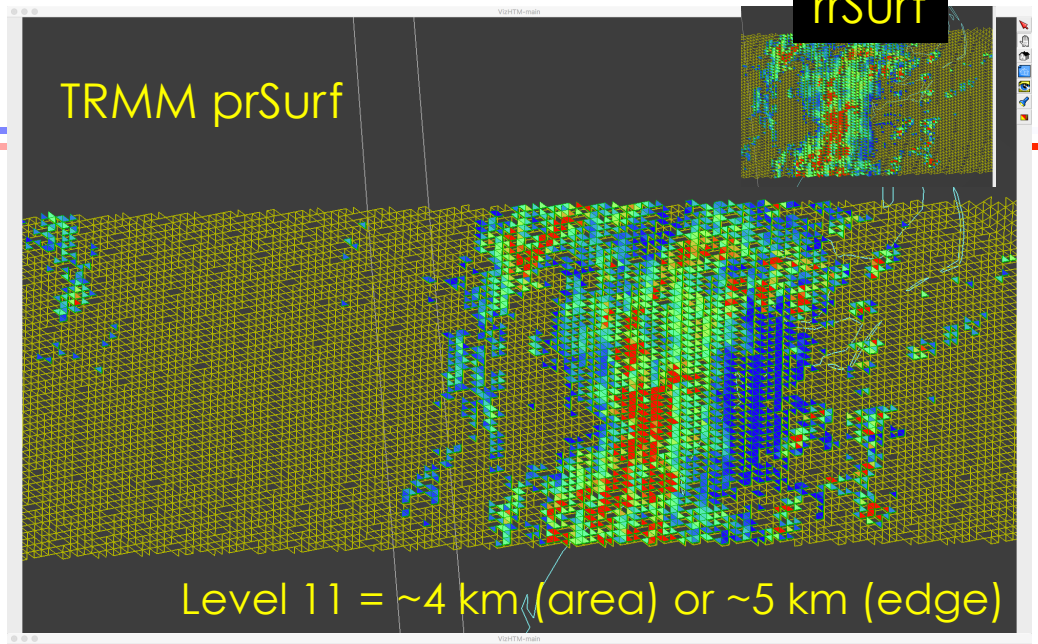
- Joining a week of NMQ and TRMM 2B31 takes less than 1 minute on MAS cluster.
  - NMQ: 2016(=12*24*7) 5-min time slices of 7000×3500 2D floating-point array.
  - TRMM 2B31: 110 orbit granules of ~9000×49 2D floating-point array (multiple attributes)
- No more painstakingly reading and filtering numerous files from various datasets!
  - Operations can be straightforwardly interfaced with a GUI for *Visual Analytics*!
- One remaining performance hurdle is *visualization* – data movement!

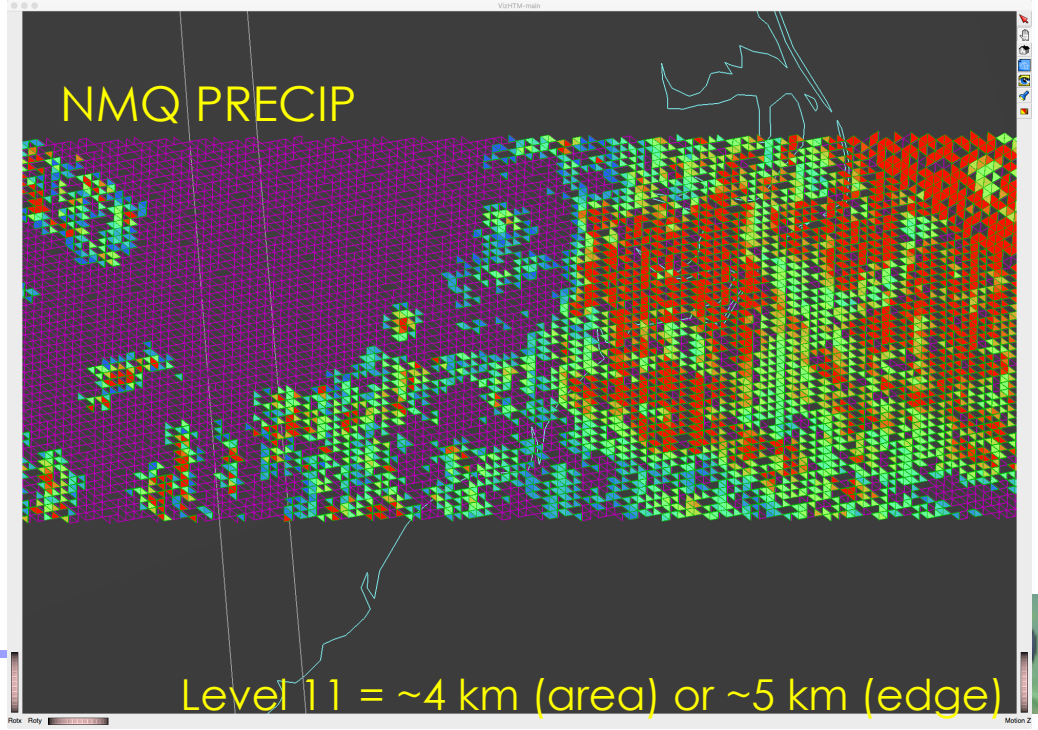


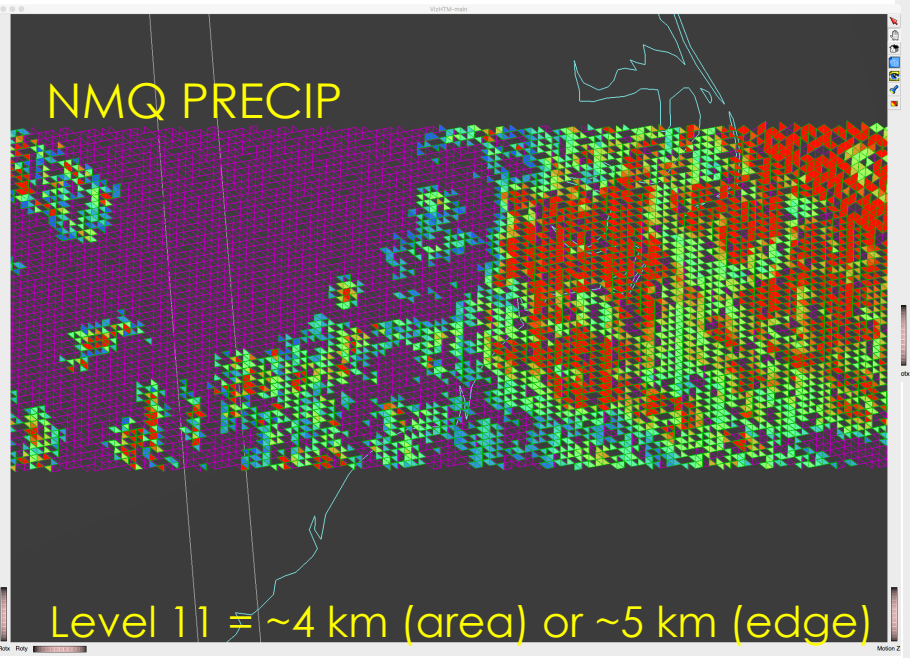
Level 7:

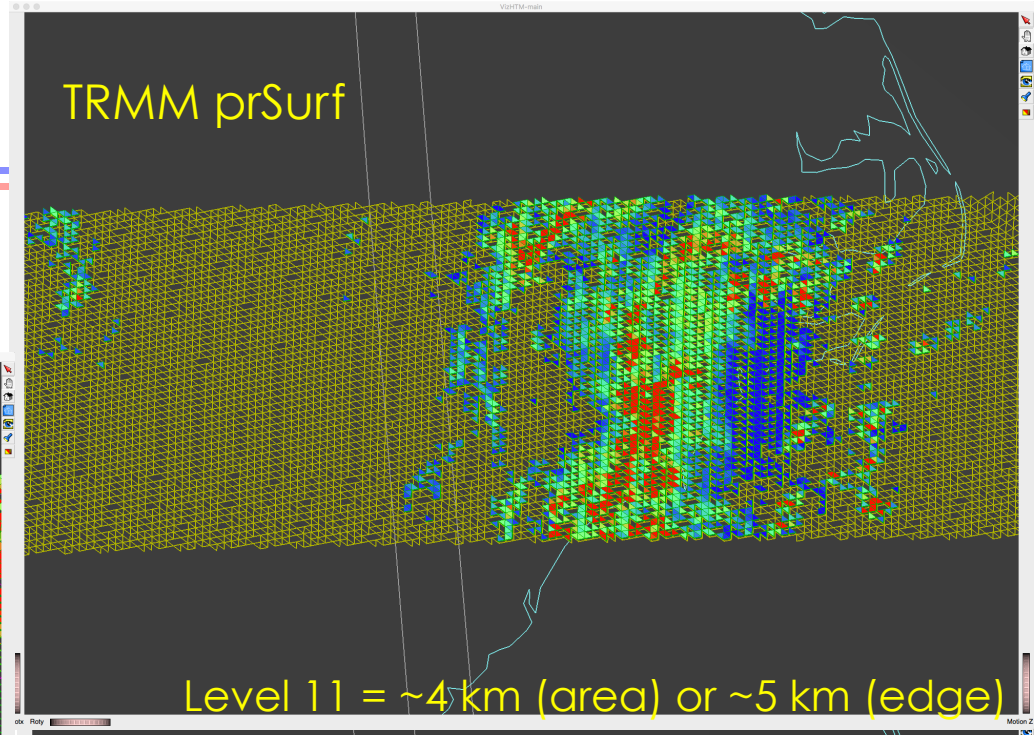
- ~63 km (area) or
- ~78 km (edge)

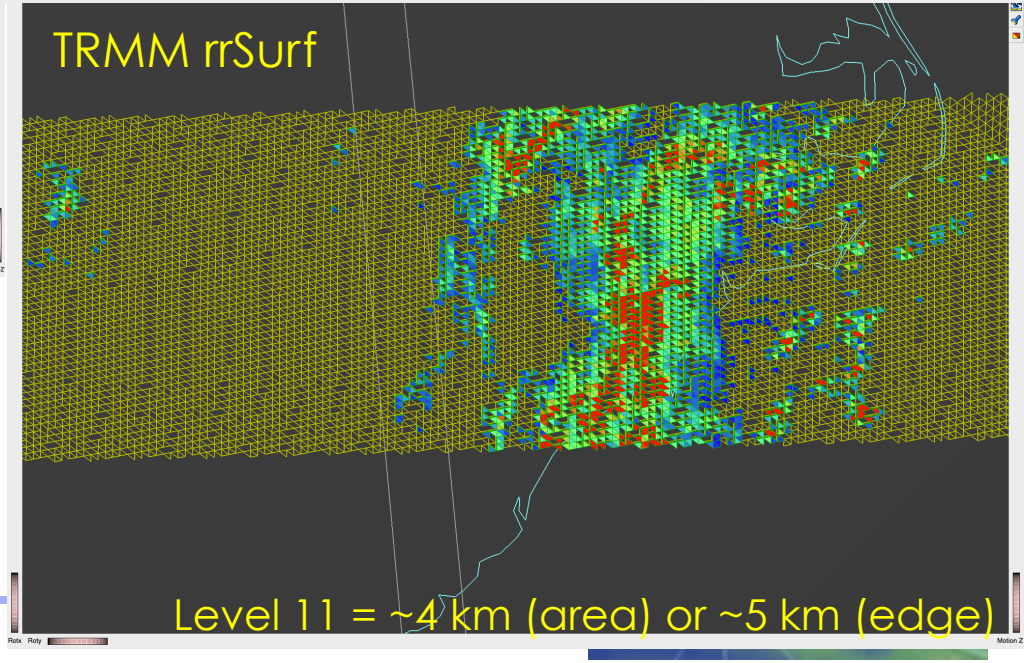




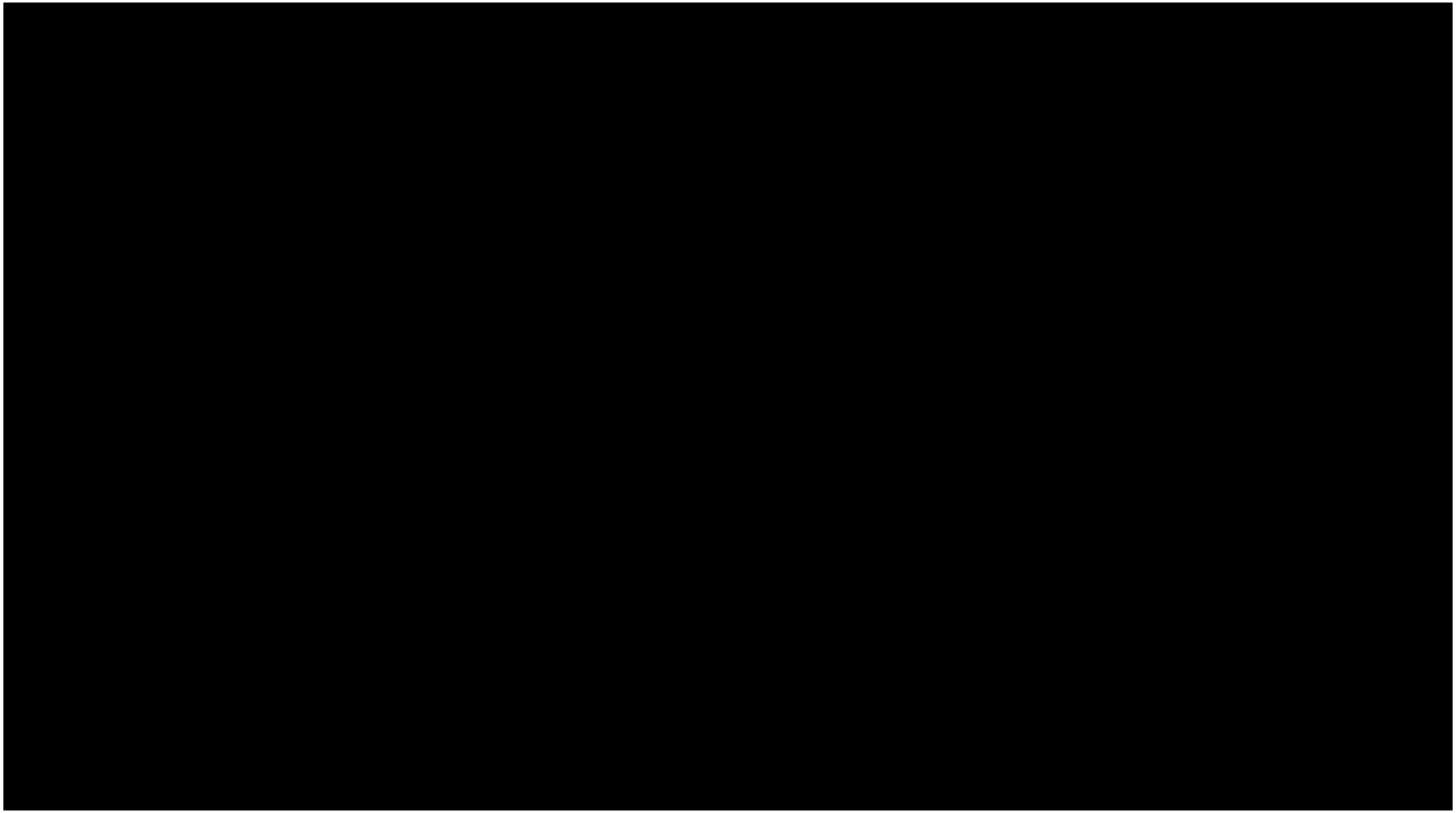



NMQ PRECIP
Level 11 = ~4 km (area) or ~5 km (edge)


TRMM prSurf
Level 11 = ~4 km (area) or ~5 km (edge)


TRMM rrSurf
Level 11 = ~4 km (area) or ~5 km (edge)

Demo Animation
(with NSF EarthCube support)



Conclusions

- Data-centric practice addresses many ills of file-centric practice.
- Volume scaling through parallelization addresses only (less than) half of the problem.
- Variety scaling through indexing/partitioning is the key to achieve better value.
- We implemented CCL in SciDB previously.
- We implemented STARE and regridding in a Big Data technology, SciDB.
- Together they automate data preparation and support *event-based* analyses.
- It constitutes a foundation for systemizing machine learning.
  - E.g., what is the probability of an event of phenomenon *A* occurring at the same location in a week given that an event of phenomenon *B* occurs?

Patent Applications

- Connected component labeling (CCL) implementation in SciDB as a user defined operator (UDO).
- SpatioTemporal Adaptive-Resolution Encoding, STARE.

THANK YOU!

HTM Resolutions

Depth	Resolution	Edge Length	Depth	Resolution	Edge Length
23	~1 m	~1.2 m	11	~4 km	~5 km
22	~2 m	~2.4 m	10	~8 km	~10 km
21	~4 m	~5 m	9	~16 km	~20 km
20	~8 m	~10 m	8	~31 km	~39 km
19	~15 m	~19 m	7	~63 km	~78 km
18	~31 m	~38 m	6	~125 km	~157 km
17	~61 m	~77 m	5	~251 km	~314 km
16	~122 m	~153 m	4	~501 km	~628 km
15	~245 m	~307 m	3	~1003 km	~1,256 km
14	~490 m	~615 m	2	~2005 km	~2,500 km
13	~1 km	~1.2 km	1	~4011 km	~5,000 km
12	~2 km	~2.5 km	0	~8021 km	~10,000 km