

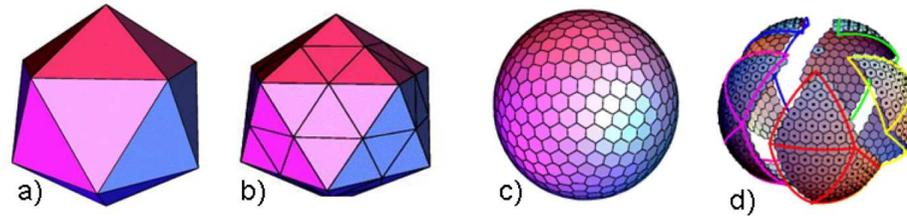
Global Cloud Resolving Model (GCRM) Grid

Karen Schuchardt
GO-ESSP, Oct 2009

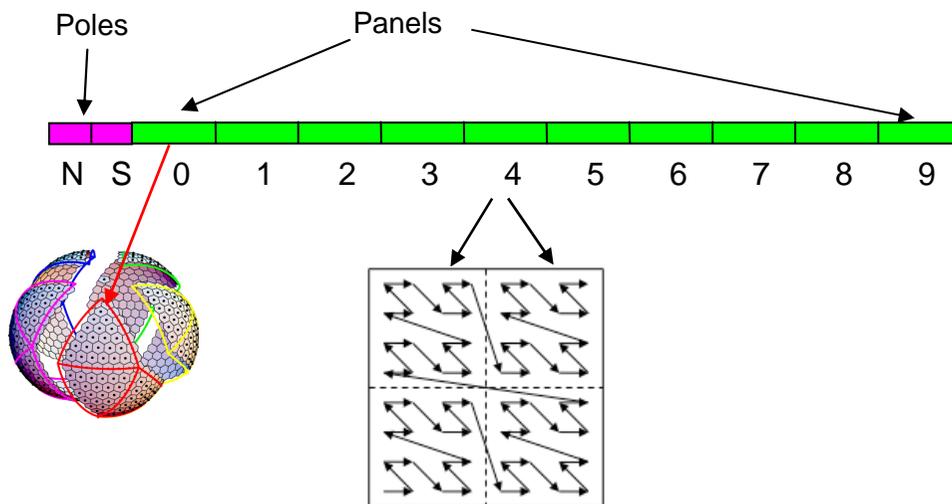
The Global Cloud Resolving Model

- ▶ Dave Randall @ Colorado State University
 - Similar in scale (and grid) to Earth Simulator
- ▶ Two Models: Red Team/Blue Team
- ▶ Data Products:
 - Near term, week long simulations to evaluate model
 - One 1 year run by 2011
 - Within 10 years, used in weather forecasting
 - 10+ years, “time-slice” execution to improve parameterization of coarse grained models.

The Geodesic Grid



- Start with a regular dodecahedron, which is a platonic solid with 20 equilateral triangles as faces.
- Decompose each triangular face into four new triangles.
- Repeat...
- Grid points form a logically rectangular block.



Morton Ordering

- ▶ Blocks within panels can be written as contiguous blocks
- ▶ Order not dependent on number of processors
- ▶ Good data locality for parallel analysis

The Data Challenge

GCRM Horizontal Grid

	#cells	#corners	#edges
~7 km	10M	20M	30M
~4 km	40M	80M	120M
~2 km	160M	320M	480M



We need parallel tools!

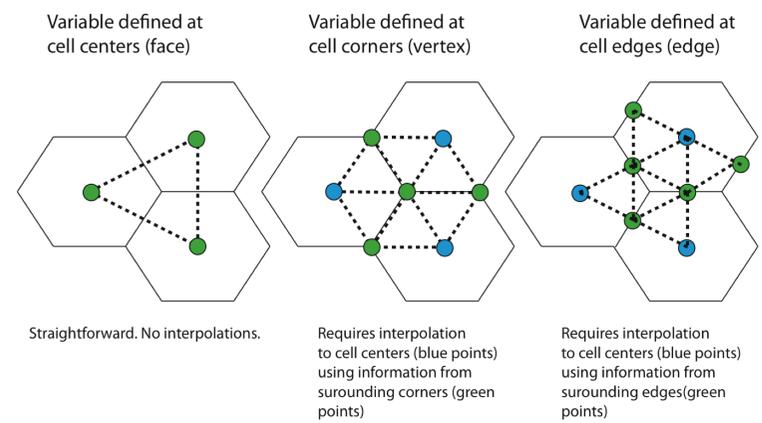
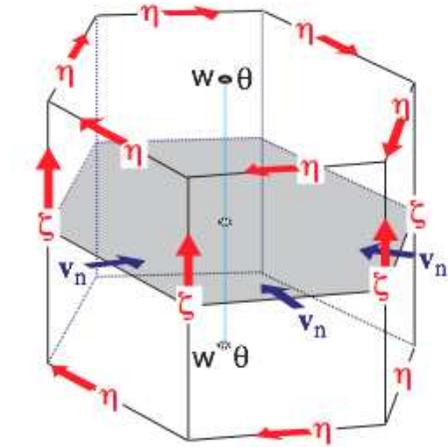
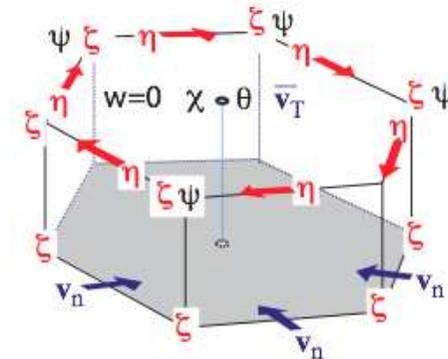
Assuming

- 4km
- 100 vertical levels
- 11 cell (16.8GB each)
- 4 corner (33.6 GB each)
- 5 edge (50.3 GB each)

Duration	Frequency	Size
1 snapshot	NA	571 GB
1 day	3 hourly	4.5 TB
1 week	4 hourly	24 TB
1 month	6 hourly	68.5 TB
1 Year	6 hourly	822 TB

Data Model

- ▶ Vertical layers and interfaces
- ▶ Horizontal data at cells, corners, edges
- ▶ External Grid files
- ▶ Currently drawing from:
 - ▶ CF
 - ▶ UGRID
 - ▶ Gridspec (but not tiles)
 - ▶ Our Extensions = Neighbors+



Data Model – Enhanced

Dimensions

```
time = UNLIMITED ;
cells = 41943042 ;
corners = 83886080 ;
edges = 125829120 ;
layers = 25 ;
interfaces = 26 ;
cellcorners = 6 ;
celledges = 6 ;
cellneighbors = 6 ;
```

Vertical

```
float interfaces(interfaces)
  interfaces:positive = "down";
  interfaces::axis = "Z";

float layers(layers)
  layers:positive = "down";
  layers::axis = "Z";
```

Topology

```
int grid(cells, cellcorners) ;
  grid:cell_type = "hex";
  grid:index_start = 0s;
  grid:standard_name = "connectivity_array";
  grid:edgelist = "cell_edges";
  grid:coordinates_cells = "grid_center_lon grid_center_lat";
  grid:coordinates_nodes = "grid_corner_lon grid_corner_lat";
  grid:coordinates_edges = "grid_edge_lon grid_edge_lat";
  grid:traverse = "counter-clockwise";
int cell_edges(cells, celledges) ;
  cell_edges:traverse = "counter-clockwise";
  cell_edges:standard_name = "edge_array";
int edge_corners(edges, 2);
  edge_corners:standard_name = "TBD";
int cell_neighbors(cells, cellneighbors) ;
  cell_neighbors:traverse = "counter-clockwise";
  cell_neighbors:standard_name = "connectivity";
```

Geometry

```
float grid_center_lat(cells) ;
  grid_center_lat:bounds = "corner_cell_map_lat" ;
float grid_center_lon(cells) ;

float grid_corner_lat(corners) ;
float grid_corner_lon(corners) ;

float grid_edge_lat(edges) ;
float grid_edge_lon(edges);
```

Convenience

```
float corner_cell_map_lon(cells, cellcorners);
float corner_cell_map_lat(cells, cellcorners);

float edge_cell_map_lon(cells, celledges) ;
float edge_cell_map_lat(cells, celledges);
```

Variables

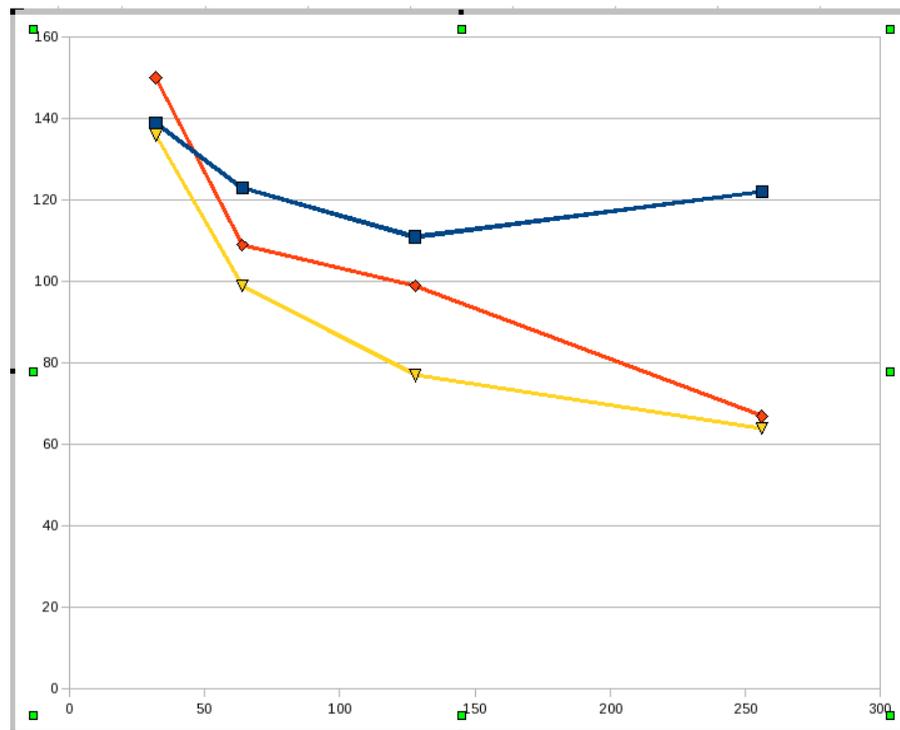
```
float wind(time, edges, layers) ;
  wind:grid = "grid";
  wind:coordinates = "grid_edge_lat grid_edge_lon";

float pressure(time, cells, interfaces) ;
  wind:grid = "grid";
  pressure:coordinates = "grid_center_lat grid_center_lon" ;

float wind_ew(time, corners, interfaces) ;
  wind:grid = "grid";
  pressure:coordinates = "grid_corner_lat grid_corner_lon" ;
```

Parallel Subsetter

- ▶ Based on Global Arrays toolkit to simplify communications
- ▶ Currently binding to PNetCDF but pluggable to NetCDF4
- ▶ Investigating ways to improve IO performance



- N. Hemisphere
- ◆ N. Hemisphere region; collective buffering
- ▼ MJO region; collective buffering

Parallel Data Analysis

▶ Visit

