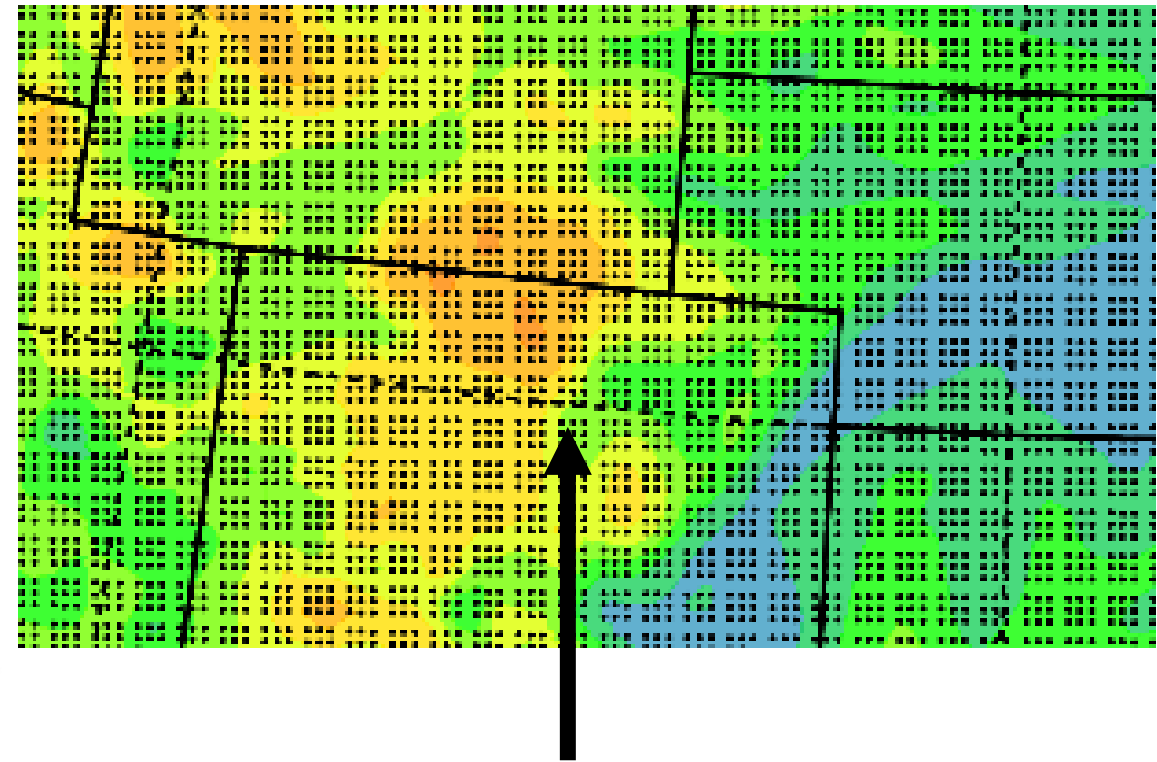
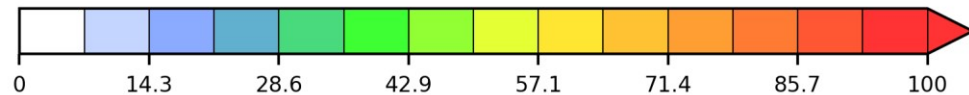
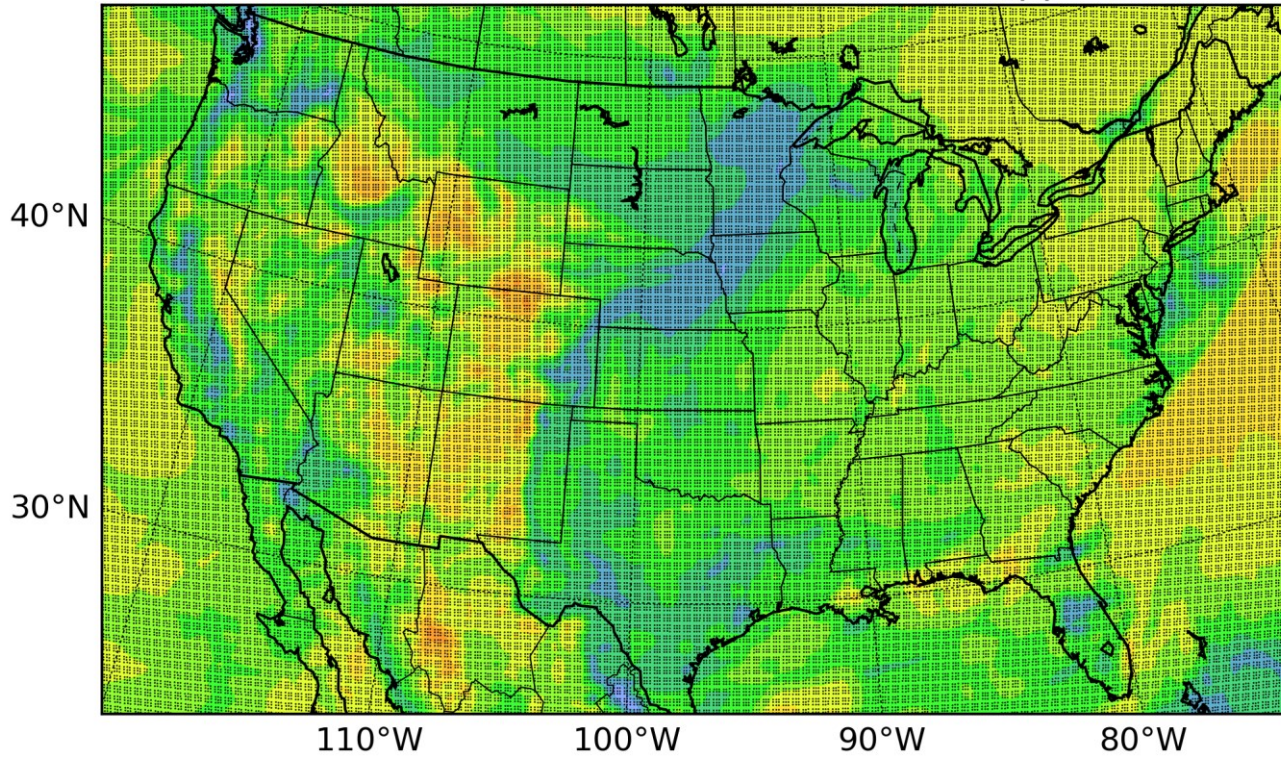


The CESM Regionally Refined Grid . . .

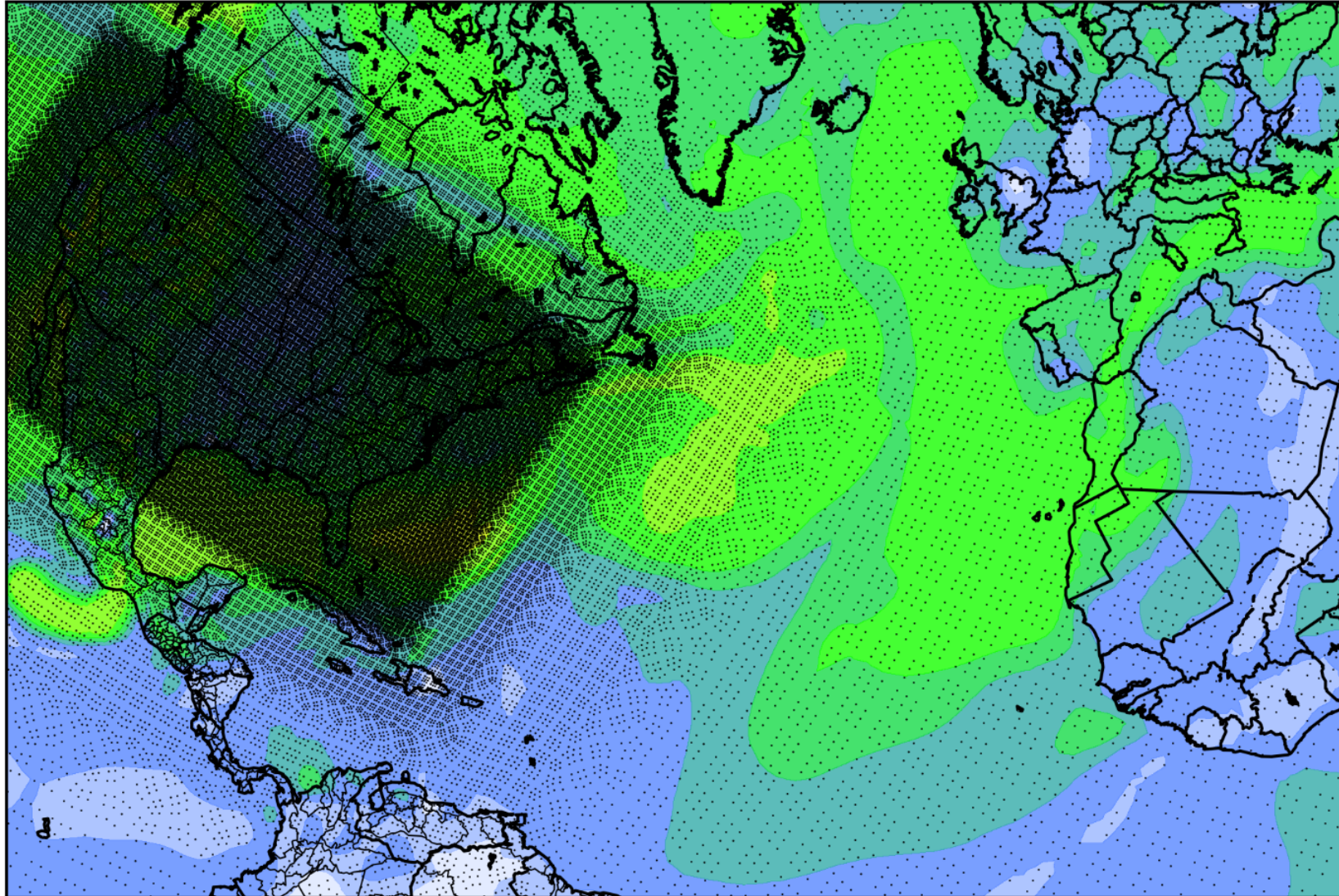
CESM RR O3 at Sfc 2020-04-18 16:00 (ppbv)



Boulder

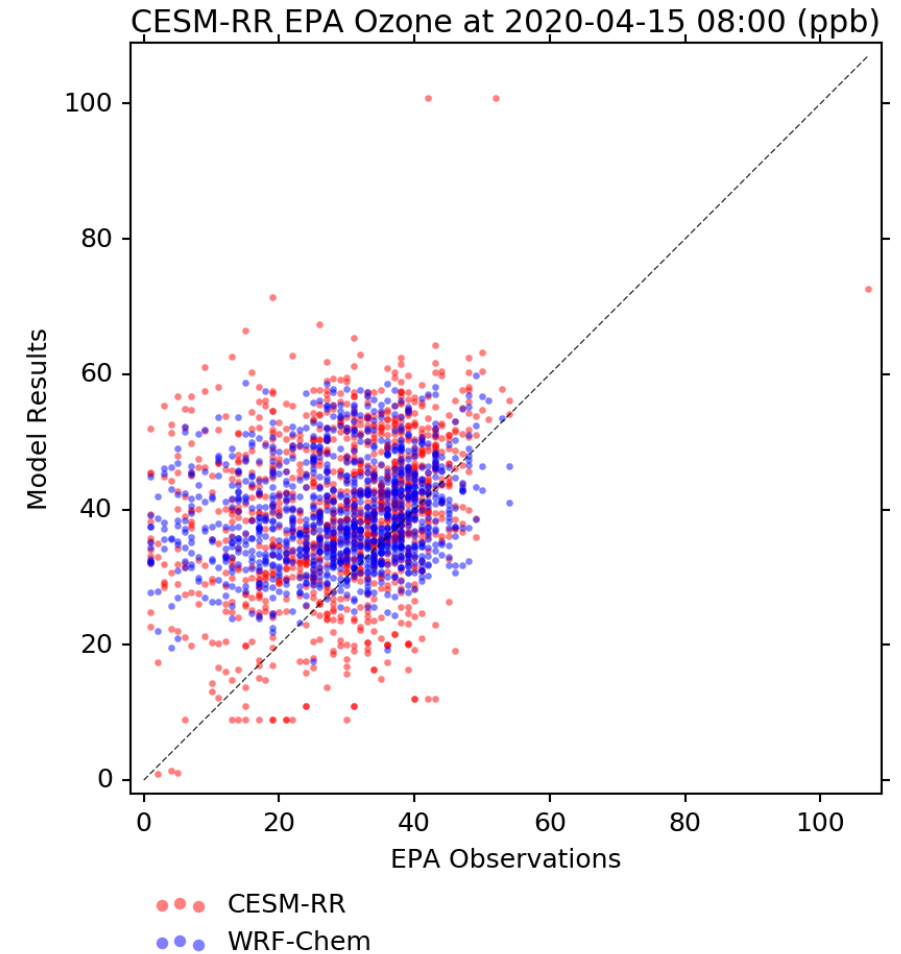
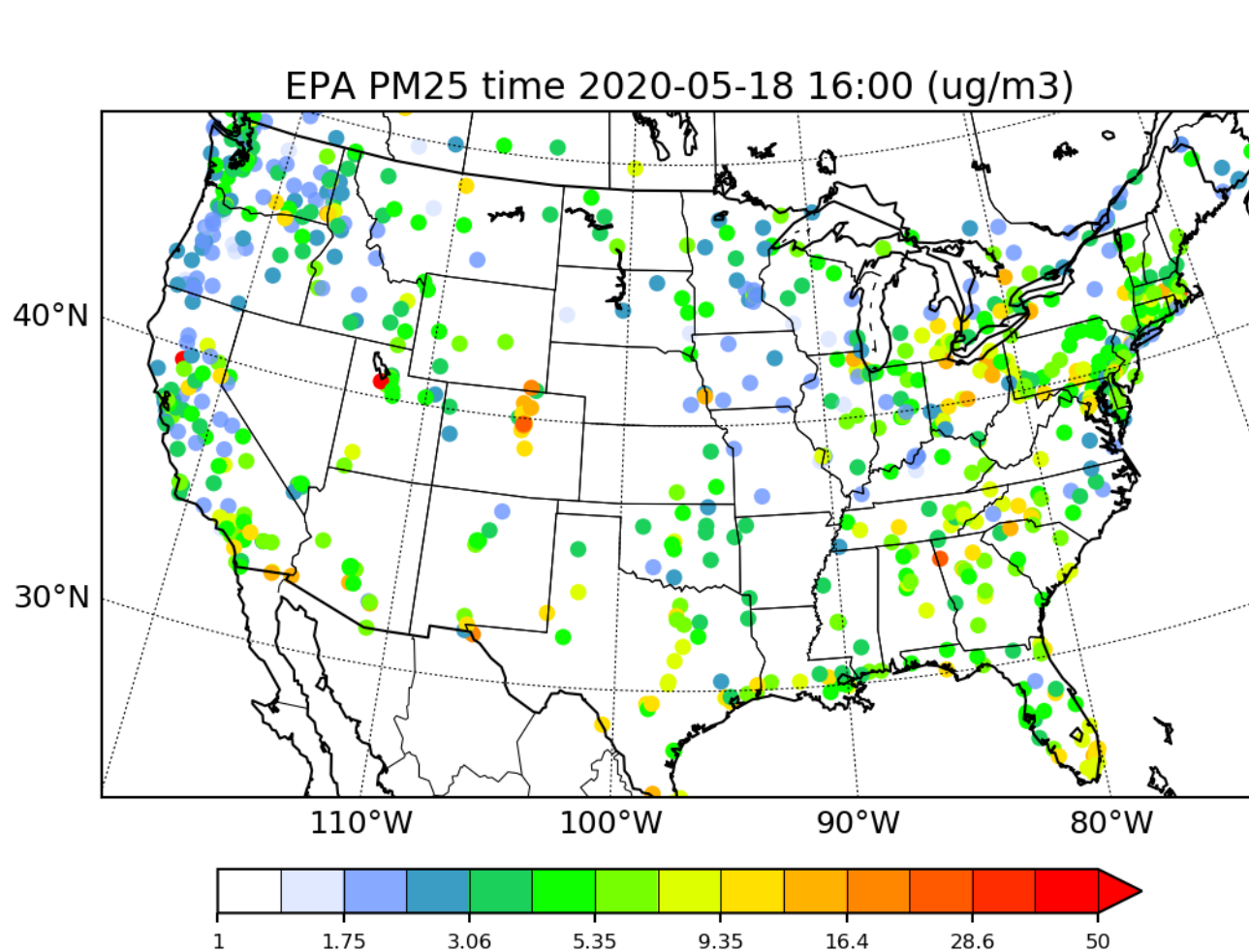
Carl Drews: June 17, 2020 presentation to CESM Workshop - Software Engineering Working Group.

. . . is fascinating!



- 174,098 grid points.
- Clusters of 9 points (3x3).
- Dense over CONUS.
- Grid expands in steps outward.
- Global grid.
- “Regular” grid, but rotated farther from CONUS.
- Designed for computation across scales (MUSICA).

The Problem



Look up CESM chemical value at 1,161 EPA stations, mostly within CONUS. Compare on a scatter plot. ***Do this efficiently.*** Remember that CESM-RR is not a lat-lon grid; grid points are organized in **columns**.

Baseline algorithm

1. Look up **P** EPA stations across **N** CESM grid points.
2. Find the closest grid point by lat-lon coordinate.
3. Return the CESM value at that point.
4. Plot EPA observations vs. model output.

Baseline algorithm is $\text{Order}(P * N)$.

Lookup by latitude-longitude will be a common use case: stations, flight tracks, curtain plots, sections, profiles, EPA regions by state . . .

Baseline Performance with Order(P * N)

EPA calculation takes **152 seconds**: Python on modeling1 (ACOM machine).

Distance calculation requires cosine() to adjust for poleward convergence of meridians, and then hypotenuse() for diagonal distance from EPA points to grid points.

I suspect that other UCAR staff are working to improve this performance.

Fast Access Algorithm for Grid Lookup

cesmRR algorithm is $\text{Order}(N \log(N) + P)$.

Algorithm performs a load phase then a lookup phase.

For the EPA problem $N=174,098$ and $P=1,161$:

Load time = 3.7 seconds; lookup time = 0.094 seconds.

Speedup is **40x** overall, **1500x** for lookup only.

How the algorithm works

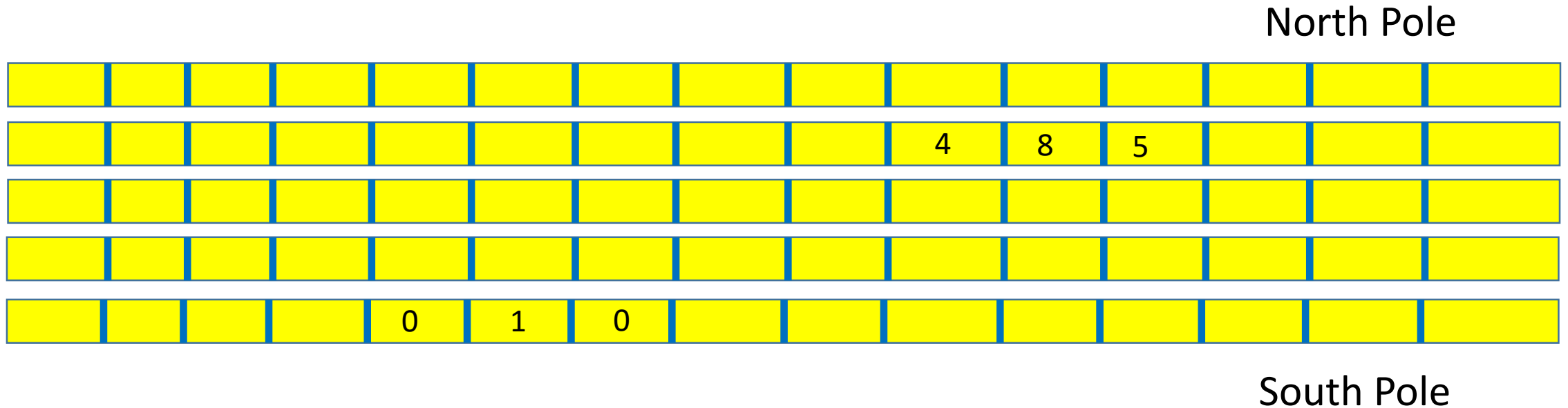
- Ignore all organization of CESM RR grid points. Ignore 3x3 clusters. Ignore focus on CONUS. Treat columns as random points thrown onto the globe. (Sorry!)
 - This technique will work for any collection of model points: triangular mesh, radial, Lambert conformal, multiple areas of high density, etc.
- Sort points into west-east bands by latitude.

North Pole



South Pole

Sort latitude bands into longitude stripes



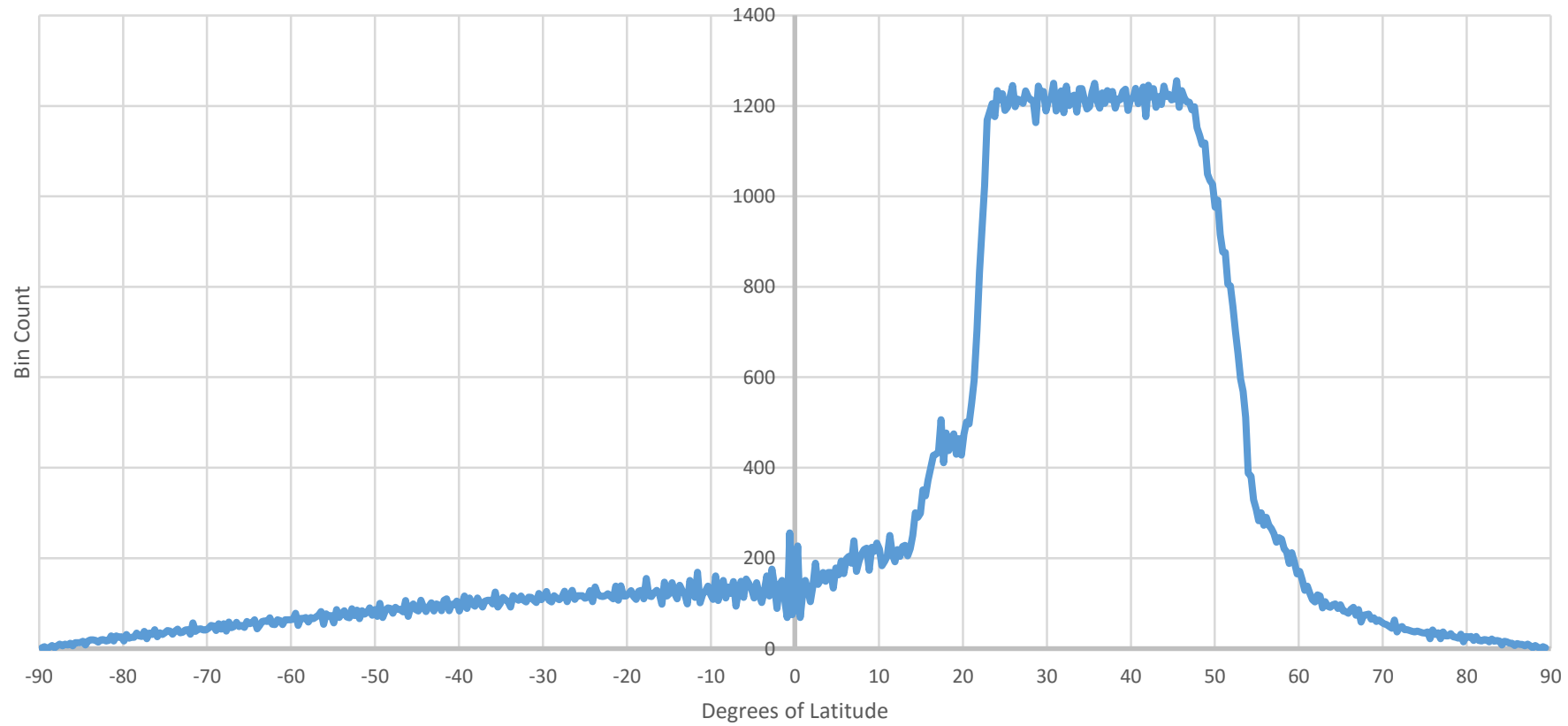
Now we have a regular-spaced lat-lon grid containing all CESM RR grid points.

We know that grid cells over CONUS will contain multiple points, and many grid cells over the Southern Ocean will be empty.

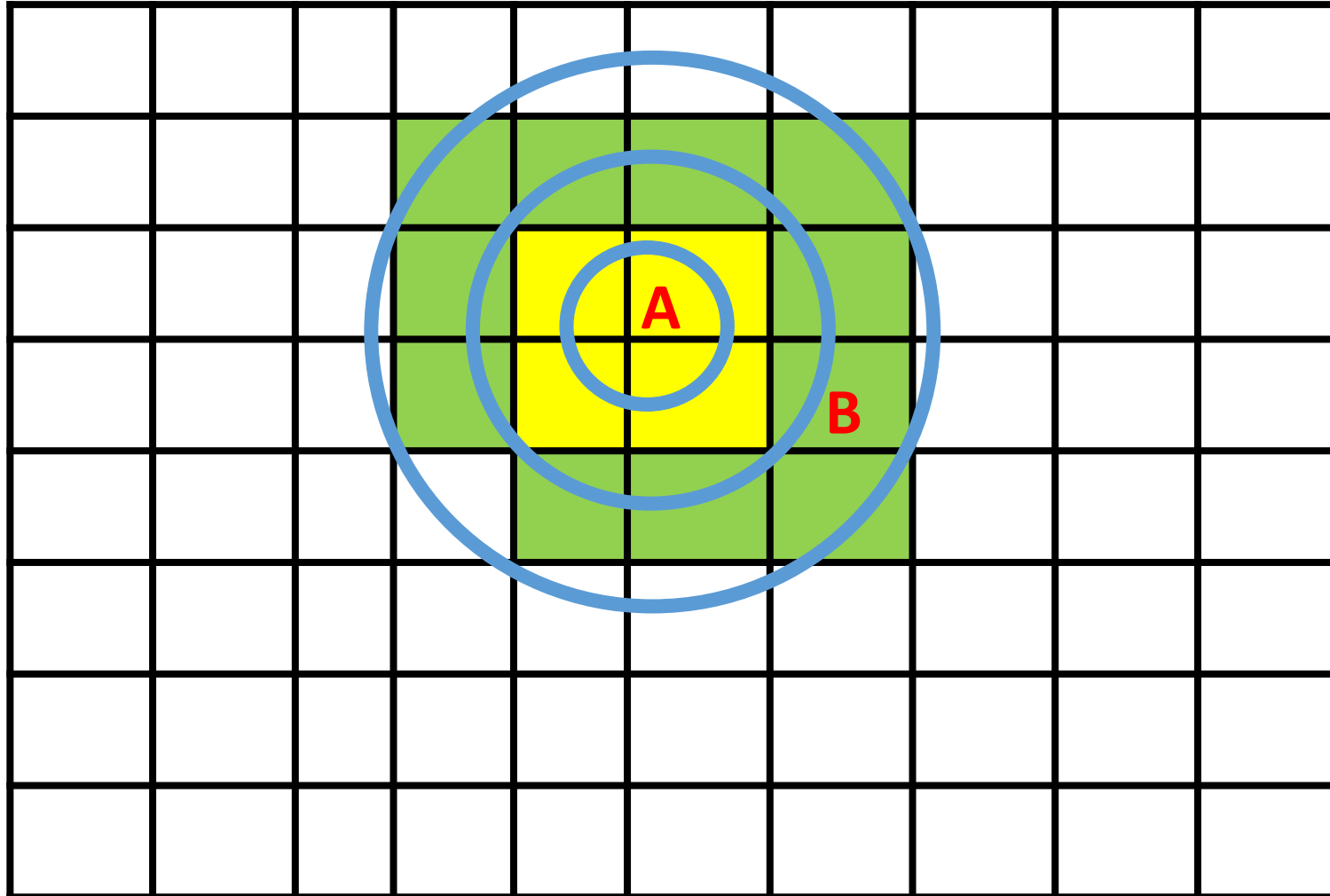
Maximum points in any one cell turns out to be 9. Why?

Algorithm Diagnostics

Number of Grid Points by Latitude Band



Search Within Single Grid Cells



Increase search radius to find point B closest to A.

Each grid cell contains < 10 points.

Sparse factor.

Maximum Search Radius

EPA station at [63.5086 North, -116.0078 East] requires search radius 4. Surely that must be a mistake! Look it up . . .



. . . nothing there, looks like a mistake . . .



. . . nope! It's real EPA station.



Accuracy decreases outside the CONUS domain, but cesmRR algorithm still works.

Software Engineering

- 592 Lines of Python code in module `cesmRR.py`, including unit tests and diagnostics.
- **Future: Create standard Python module for handling CESM RR grids, like Python `wrf.py` module.**
- Use throughout NCAR and beyond.
- `cesmRR` algorithm uses variable-length lists. Other programming languages?

Conclusion

We all like to speed up code by a factor of 40x – 1500x!
Post-processing, plotting, lookup, comparison . . .

Questions?

cesmRR Grid Parameters

Loading cesmGrid with 174098 points:

Latitude range [-89.66428131686655, 89.66428131686655]

Longitude range [0.0035776311085919002, 359.99909607128831]

Aspect ratio of domain = 2.0074633575

Raw grid dimensions are 294 rows x 593 columns.

Sparse factor 2 produces 588 rows x 1186 columns.

Latitude stride = 0.3049975555 and longitude stride = 0.303545968331

Maximum points in any cell = 9, at [21.905062998321295, 239.80501944268408].