A ten year radar climatology of rainfall over Portland, Maine.

SCOTT COLLIS*

Environmental Science Division, Argonne National Laboratory

ABSTRACT

This paper describes a Department of Homeland Security, Regional Resiliency Assessment Program study into precipitation over Portland, Maine carried out by Argonne National Laboratory. The primary data source is scanning weather radar which collects information about precipitating particles over a large (250km plus) domain. The Python ARM radar Toolkit was used to retrieve rainfall rates from the local radar (in the Gray township outside of Portland) and map them onto a 250m resolution grid over a 100km domain. Using Argonne's large scale cluster, Blues, the problem was mapped to 10 years worth of data, just over 600,000 ten minute volumes or just over 10TB of data. Each time step was reduced to two numbers: The domain mean and maximum rainfall rate to allow easy indexing of the data set. In addition the diurnal (daily) cycle of precipitation was studied as was the seasonal cycle.

1. Introduction

Quoting directly from the Department of Homeland Security's (DHS) webpage : *The Regional Resiliency Assessment Program (RRAP) is a cooperative assessment of specific critical infrastructure within a designated geographic area and a regional analysis of the surrounding infrastructure. The RRAP is led by the Department of Homeland Security and addresses a range of hazards that could have regionally and nationally significant consequences. Each year, the Department selects these voluntary and non-regulatory RRAP projects with input and guidance from federal and state partners.* Argonne is a partner in the RRAP project over Portland, Maine. One specific aspect we are looking at in critical drainage infrastructure and what will be required in a changing climate scenario. Dynamical downscaling (eg Wang and Kotamarthi (2015)) where a high resolution computer model is used to simulate what happens within the 100km climate model grid cell, is coming into more common usage. However, the key science question being asked by Argonne researchers is: Is the 12km resolution of these models enough to resolve high intensity localized precipitation events?

To this end our team carried out a study that involves using NOAA weather radars, known as Weather Surveillance Radars -1988 Doppler or WSR-88D (Crum and Alberty (1993)) to create high resolution rain maps for ingest into hydrological models. These rain maps are then de-resolved or made coarser and the impact on stream and culvert flow analyzed over the Portland area. This report only deals with the generation of the rain maps and is broken into four sections: This introduction, Background information on the tools used and an introduction to radar meteorology, a discussion of the results obtained and some thoughts as to future work.

2. Backround

Data from the WSR-88D radars are freely available both realtime, archived by the National Centers for Environmental Information and also, recently in the Amazon Cloud. In order to help the reader understand the processing that was performed on the radar data to extract geophysical insight we will first give some basic background in radar meteorology.

a. NOAA weather radars

The National Weather Service operates a network of 160 radars for the monitoring of atmospheric phenomena. These transmit pluses of radiation and monitor the reflections, or backscatter, of objects in the pulse's path. The radar receives these reflected signals and calculates profiles of data along the path of propagation of the pulse. If the radar stayed still users will receive just one "soda straw" view through the target of study (eg a rainstorm) so, in order to map a volume of space the radar is scanned in azimuth (from north) and elevation (from the horizon). This is illustrated in part in fig 1.

A single sweep through 360 degrees of azimuth is carried out before the elevation angle is changed. This is

* *Corresponding author address:* Scott Collis, Argonne National Laboratory, Environmental Science division, 9700 South Cass Ave, Argonne, IL 60514
E-mail: scollis@anl.gov

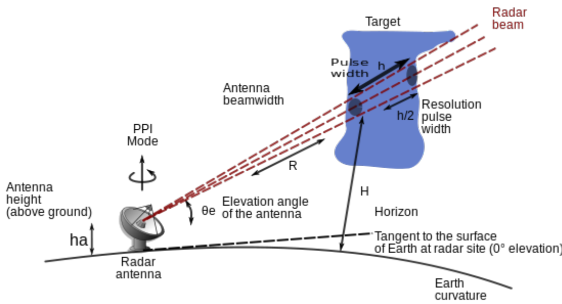
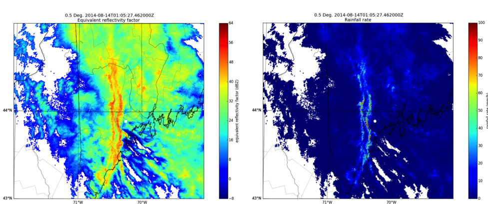


FIG. 1: Cartoon of the propagation of the radar beam through a storm, By Vigneron et Pierre cb - Own work, CC BY-SA 3.0, Link to Wikimedia

called a Plan Position Indicator scan or PPI (a historical hangover from when radar images were viewed on cathode ray tube displays). One primary measurement radars collect is how effectively the media (eg raindrops) reflects radiation back and is called Reflectivity Factor often denoted Z_e. We will skip the derivation and description of this quantity, interested readers are directed to Doviak and Zrnić (1984) for more information. An example of a PPI of reflectivity factor is shown in fig 2a.



(a) Radar Reflectivity. (b) Rainfall rate.

FIG. 2: Data in the radars native coordinate system of azimuth and range from the lowest elevation sweep of 0.5°.

If we assume that the volume of the radar pulse contains a number of rain droplets of varying diameter (D) and can be described as obeying a number distribution N(D) (so for a particular Diameter you get a density of rain drops in m^{-3} of N). The reflectivity of raindrops ends up being:

$$Z_e = \int N(D) D^6 dD$$

or the 6^{th} moment of the drop size distribution. This means a small number of larger drops has a bigger impact than increasing the density of smaller drops.

There have been many papers written on the best way to derive rainfall rates from reflectivity factor and other radar variables. For the purposes of this study where we are more interested in the comparative nature of rainfall patterns we stuck to a simple parametric fit as presented in Gu et al. (2011) of

$$Z_e = 300R^{1.4}. \tag{1}$$

Equation is inverted to give rainfall rate and an example of this is shown in fig 2b.

b. Application chain

The Python-ARM Radar Toolkit (Py-ART, Heistermann et al. (2014)) is the main tool used to perform the steps of: Reading, retrieving rainfall rate, gridding to a cartesian grid and writing to a Community standard file format. Py-ART is a community data-model drive architecture for interactive development of custom processing chains implemented in the Python programming language and makes heavy use of the Scientific Python ecosystem (Jones et al. (2001–)). Py-ART is community based open source software welcoming public contributions on GitHub.

Py-ART has a native reader for multiple WSR-88D file formats. The data is read into the Py-ART common data model. Rainfall rate in native (antenna) coordinates of range from the radar, azimuth from north and elevation above the horizon is calculated. Then, in order to facilitate easy analysis, the rainfall (in addition to other values) is mapped to a regular Cartesian grid with coordinates of meridional and zonal (east/west, north/south) displacement from the radar location. Mapping is achieved using an inverse distance weighted objective analysis system. Each grid point in the destination grid is assigned a radius of influence. Radar gates falling within that radius are averaged to that point using a exponential inverse distance weighting (after Barnes (1964)). Py-ART calculates the optimal variable radius of influence.

Since the radar samples azimuthally the resolution, R, of the data varies as a function of distance given by

$$R = 2d \tan\left(\frac{1}{2}\theta_{fwhm}\right) \tag{2}$$

where θ_{fwhm} is the radar beam width and d is the distance of the range gate from the radar. The WSR-88D radars have a beam width of 1 degree so at a distance of 30km the resolution will be approximately 500m. Given we wish to oversample near the city of Portland which is approximately 20km from the radar we chose a grid spacing of 200m resulting in a grid with 501 by 501 elements and a domain of 100km.

Figure 3 shows a mapping of rainfall rates calculated in fig 2b. This data is then saved to a NetCDF file that complies with the Climate Forecasting conventions (see http://cfconventions.org/).

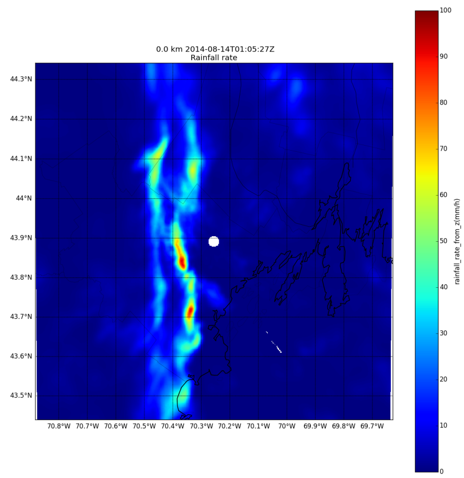


FIG. 3: Rainfall rates as shown in fig 2b mapped to a Cartesian grid around the Portland, Maine area.

c. Data informatics

Section b gave an overview of the application chain for one volume. Given that our data set consists of hundreds of thousands of files we need an effective way to reduce the problem and map to a large scale cluster. Fortunately, due to the granularity (one granule = one file = one volume) the problem is *pleasantly* parallel. Also fortunately the Scientific Python ecosystem has recently been enhanced by project Jupyter (Pérez and Granger (2007)) which includes methods for simply mapping a serialized problem to a list of parameters (eg a file list) as shown in fig 4.

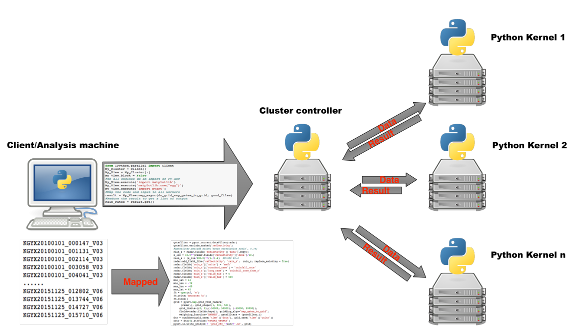


FIG. 4: pictorial representation of a Jupyter cluster being used for Map-Reduce.

Files containing data from the Portland WSR-88D (Code: KGYX) were staged to Argonne's Blues cluster with 310 compute nodes, 64 GB of memory on each node, 16 cores per compute node (Intel Sandy Bridge) 4,960 compute cores available and a theoretical peak performance of 107.8 TFlops. Mapping the 670,353 files that made up the study was performed using several jobs ranging from 64 to 1024 cores taking a wall time of just over 48 hours. Each job produced three output files: a NetCDF file containing the gridded data, plots such as those shown in fig 3 and a simple ascii file with the mean of the rainfall values in the domain and the maximum value in the domain.

3. results

The data u

a. Rainfall look up table

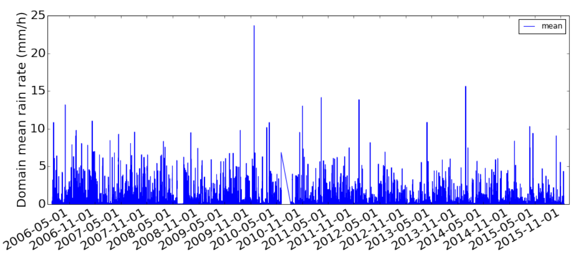


FIG. 5: Domain mean rainfall rate as derived from a reflectivity/rainfall relationship using the KGYX WSR-88D radar.

lup

b. Brief analysis of rainfall variability

lup

4. Future work

The climatology and varianc

Acknowledgments. Scott G
Patricia

References

Barnes, S. L., 1964: A Technique for Maximizing Details in Numerical Weather Map Analysis. *Journal of Applied Meteorology*, **3 (4)**, 396–409, doi:10.1175/1520-0450(1964)003⟨0396:ATFMDI⟩2.0.CO;2, URL http://journals.ametsoc.org/doi/abs/10.1175/1520-0450%281964%29003%3C0396%3AATFMDI%3E2.0.CO%3B2?prevSearch=A+technique+for+maximizing+details+in+numerical+weather+map+analysis&searchHistoryKey=.

Crum, T. D., and R. L. Alberty, 1993: The WSR-88d and the WSR-88d Operational Support Facility. *Bulletin of the American Meteorological Society*, **74 (9)**, 1669–1687, doi: 10.1175/1520-0477(1993)074⟨1669:TWATWO⟩2.0.CO;2, URL http://journals.ametsoc.org/doi/abs/10.1175/1520-0477%281993%29074%3C1669%3ATWATWO%3E2.0.CO%3B2?prevSearch=The+WSR-88D+and+the+WSR-88D+Operational+Support+Facility&searchHistoryKey=.

Doviak, R., and D. Zrnić, 1984: *Doppler radar and weather observations*. Academic Press, URL https://books.google.com/books?id=BzxRAAAAMAAJ.

Gu, J.-Y., A. Ryzhkov, P. Zhang, P. Neilley, M. Knight, B. Wolf, and D.-I. Lee, 2011: Polarimetric Attenuation Correction in Heavy Rain at C Band. *J. Appl. Meteor. Climatol.*, **50 (1)**, 39–58, doi:10.1175/2010JAMC2258.1, URL http://journals.ametsoc.org/doi/abs/10.1175/2010JAMC2258.1.

Heistermann, M., and Coauthors, 2014: The Emergence of Open Source Software for the Weather Radar Community. *Bull. Amer. Meteor. Soc.*, doi:10.1175/BAMS-D-13-00240.1, URL http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-13-00240.1.

Jones, E., T. Oliphant, P. Peterson, and Coauthors, 2001–: SciPy: Open source scientific tools for Python. URL http://www.scipy.org/, [Online; accessed 2016-03-02].

Pérez, F., and B. E. Granger, 2007: IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, **9 (3)**, 21–29, doi:10.1109/MCSE.2007.53, URL http://ipython.org.

Wang, J., and V. R. Kotamarthi, 2015: High-resolution dynamically downscaled projections of precipitation in the mid and late 21st century over north america. *Earth's Future*, **3 (7)**, 268–288, doi:10.1002/2015EF000304, URL http://dx.doi.org/10.1002/2015EF000304, 2015EF000304.