Using Machine Learning to Classify Cloud Types

Carlos Barajas[1] Lipi Mukherjee[2] Pei Guo[3] Susan Hoban[4]
Mentors: Dr. Daeho Jin[5] Dr. Aryya Gangopadhyay [3]
Dr. Jianwu Wang [3]

[1]Department of Mathematics and Statistics, UMBC

[2]Department of Physics, UMBC

[3]Department of Inforamtion Systems, UMBC

[4]Joint Center for Earth Systems Technology, UMBC

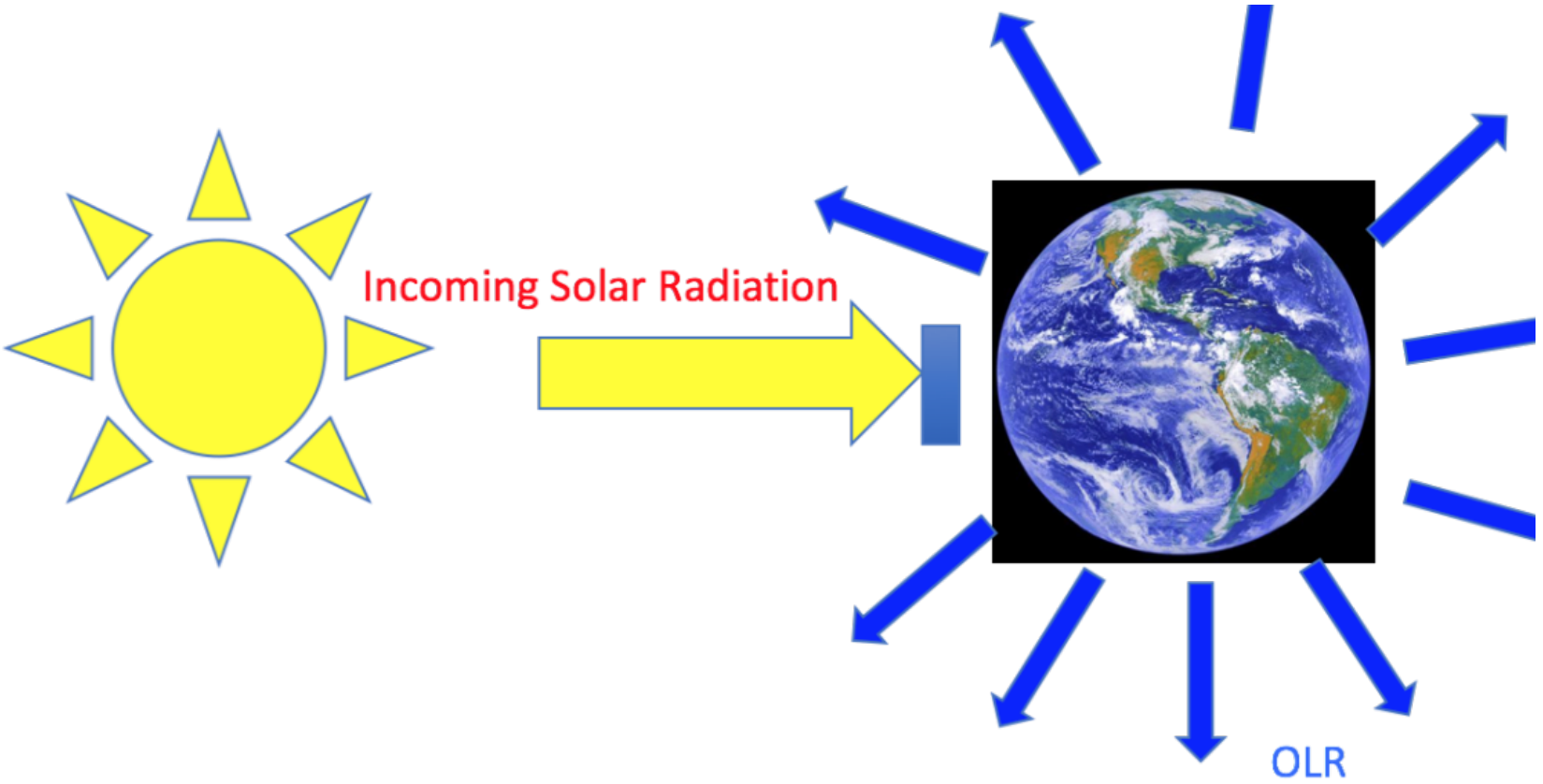
[5]GESTAR, USRA, and NASA GSFC

Cybertraining: Big Data+High-Performance Computing+Atmospheric Sciences
University of Maryland Baltimore County, Spring 2018
cybertraining.umbc.edu
Acknowledgements: NSF, UMBC, HPCF

Overview

- Introduction
- Background
- MPI implementation
- Spark Implementation
- Clustering Results
- Conclusions
- Future Work

Why do we study clouds?



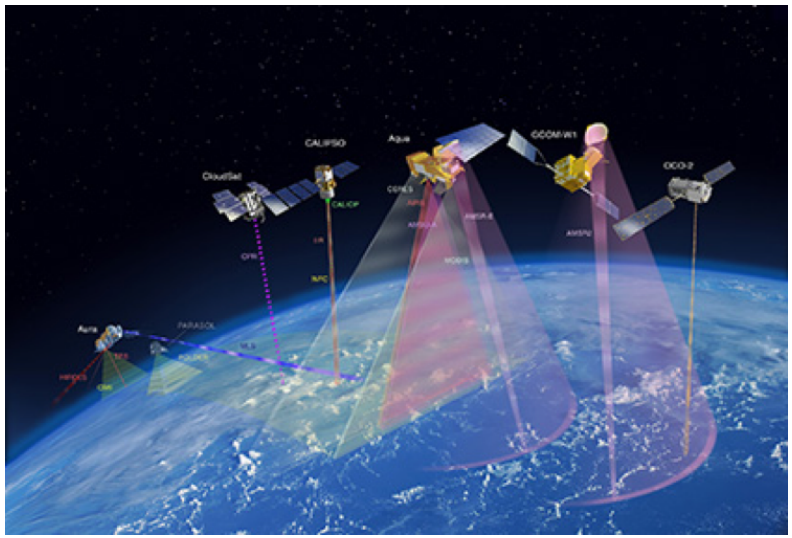
Why do we study clouds?

Cooling effect: Cloud and aerosol particles reflect (by scattering) part of the incoming solar radiation back to space, which have cooling effects on the climate

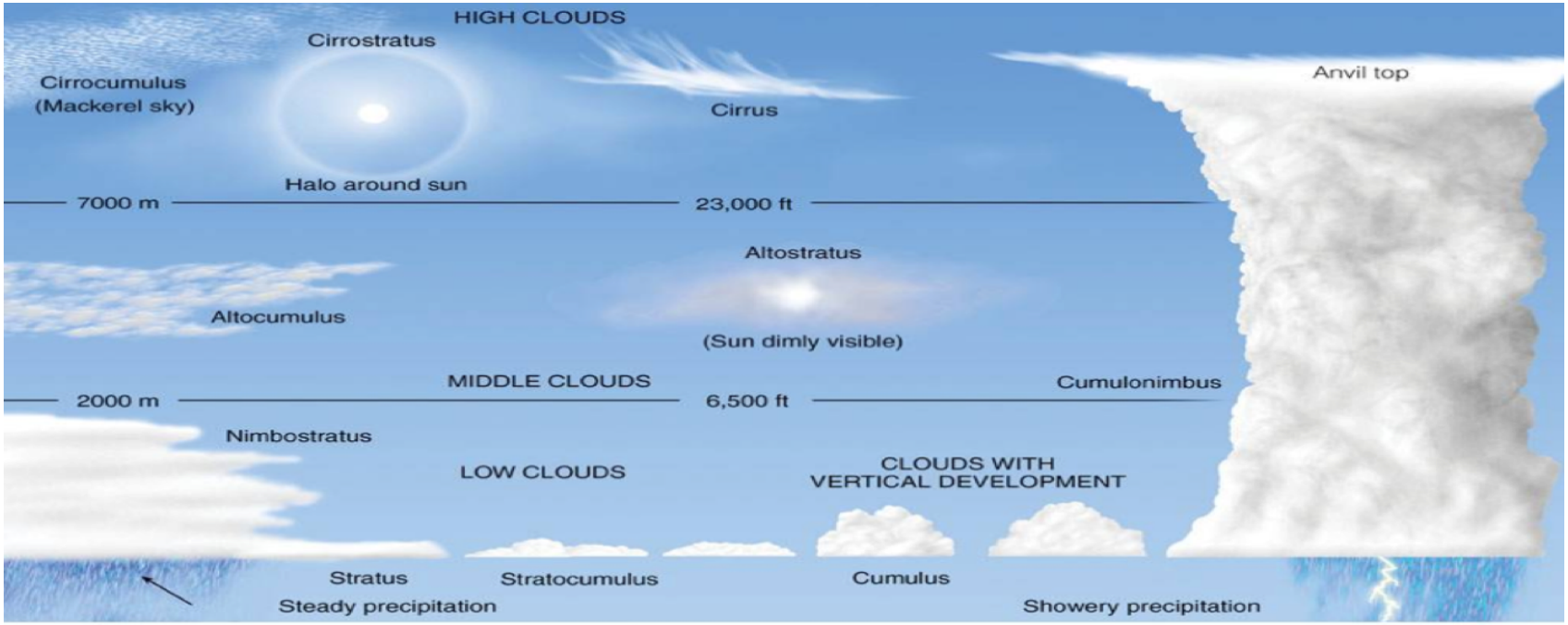
Warming effect: On the other hand, cloud and aerosol particles also absorb the infrared radiation from earth surface/lower atmosphere and re-emit at a lower temperature (i.e., greenhouse effect), which warms the climate.

How do we study clouds?

- NASA satellite data (e.g., MODIS, CALIPSO, CloudSat, CERES)
- Atmospheric modeling

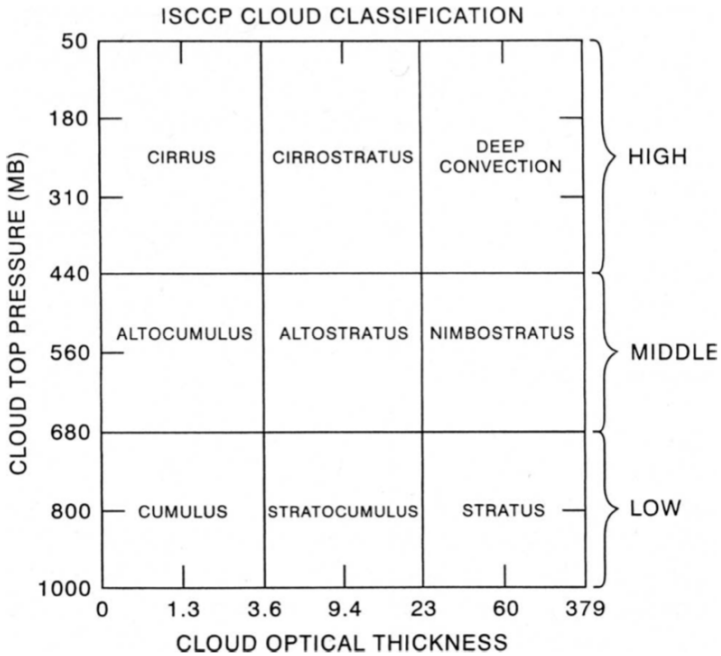


What problem are we trying to solve?

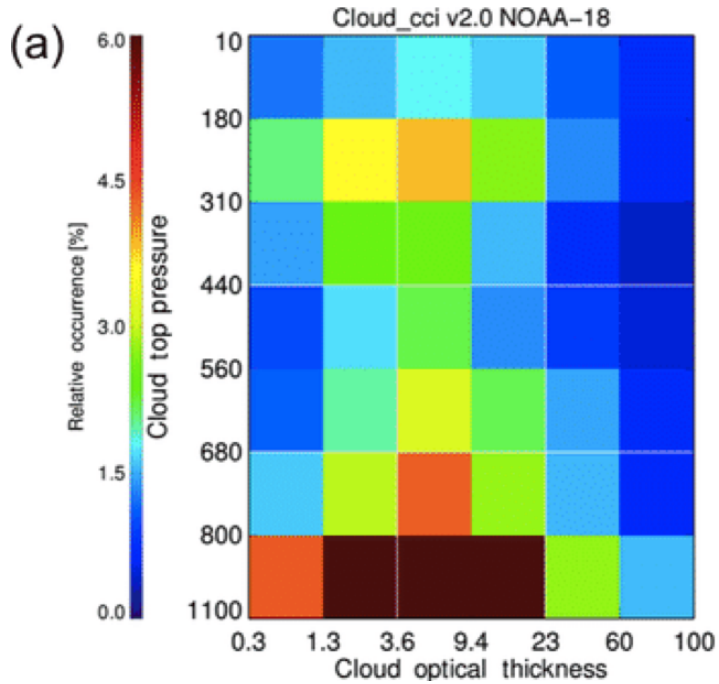


The impact of clouds on the Earth's net cooling and warming effect motivate our need to classify and quantify clouds.

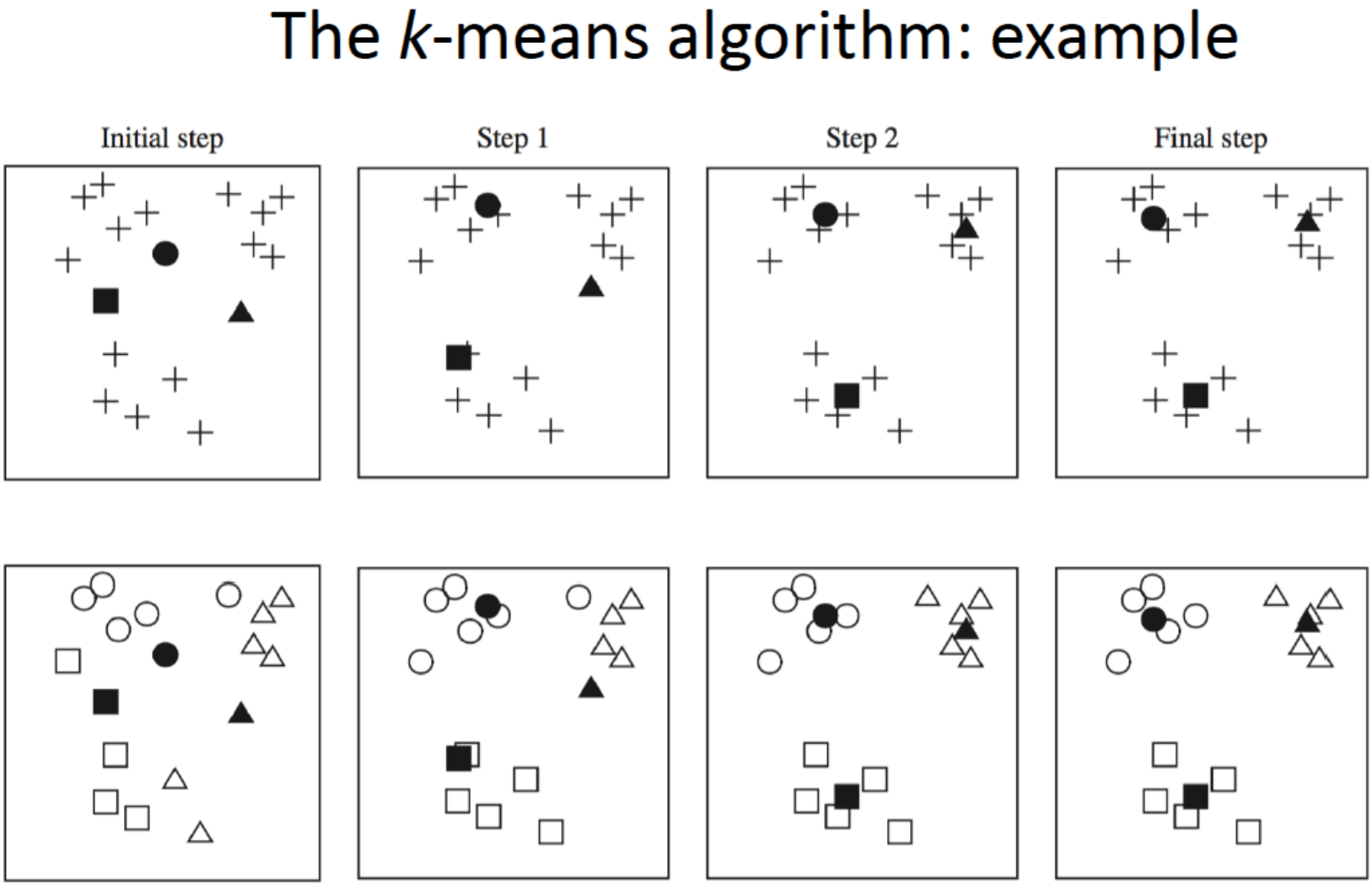
International Satellite Cloud Climatology Project (ISCCP)



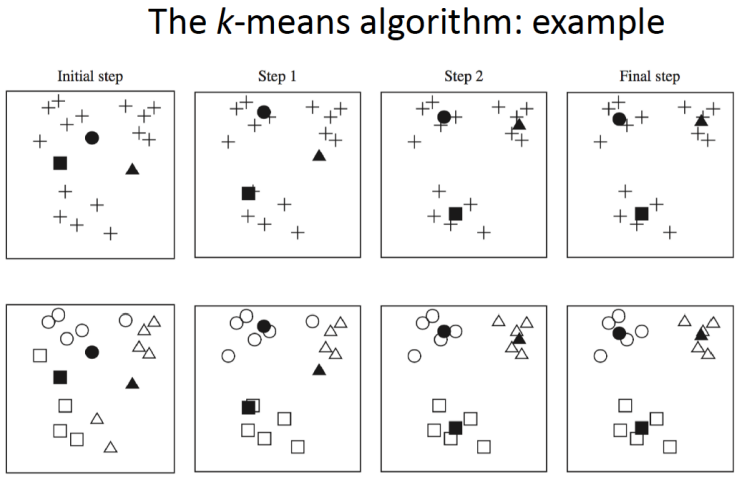
Rossow WB 1999

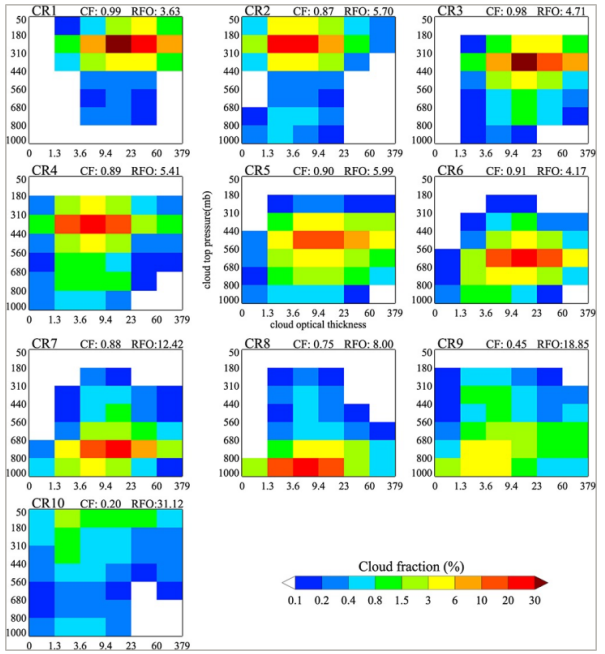


K-Means


The k-means algorithm: example
Initial step
Step 1
Step 2
Final step

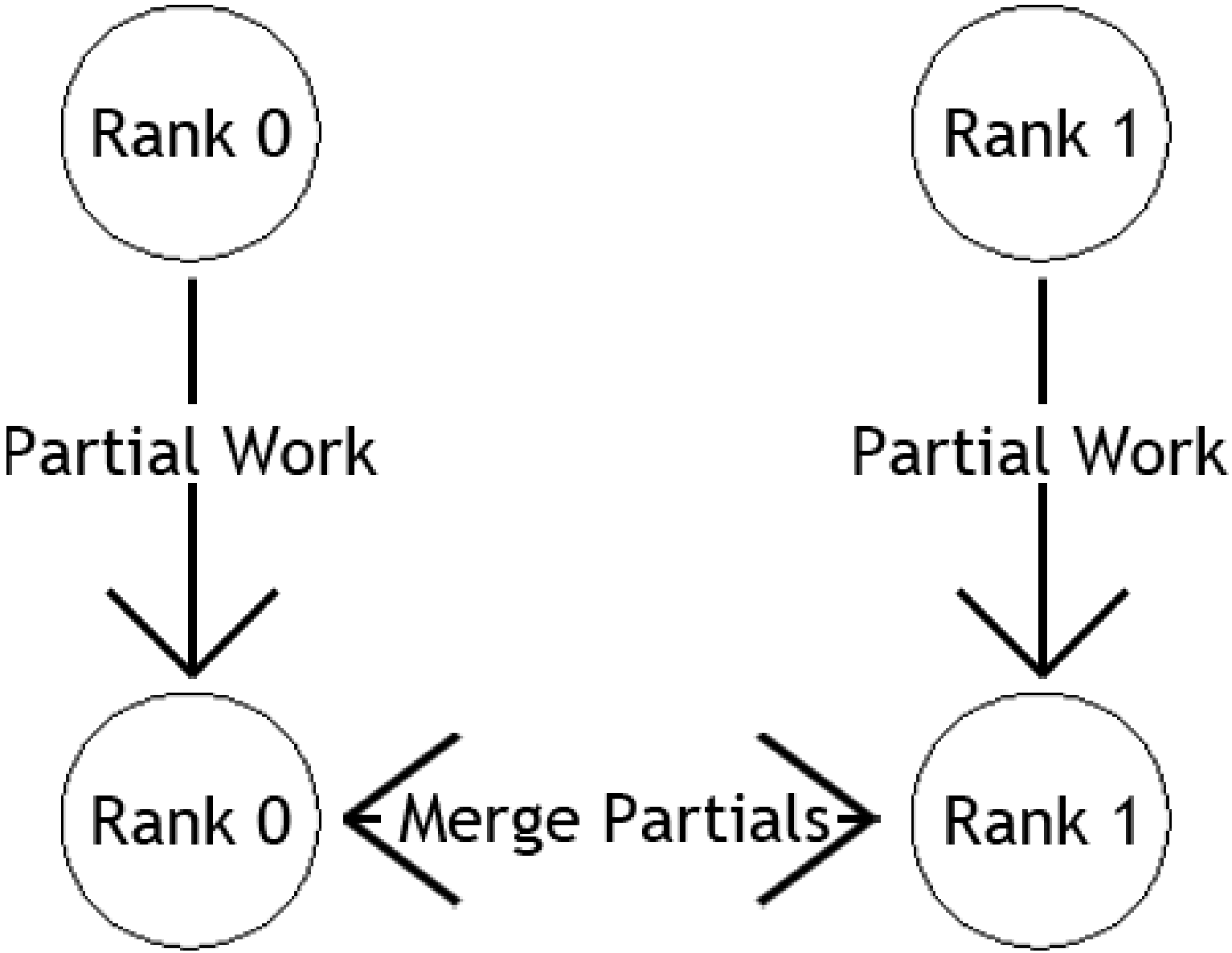
K-Means Application

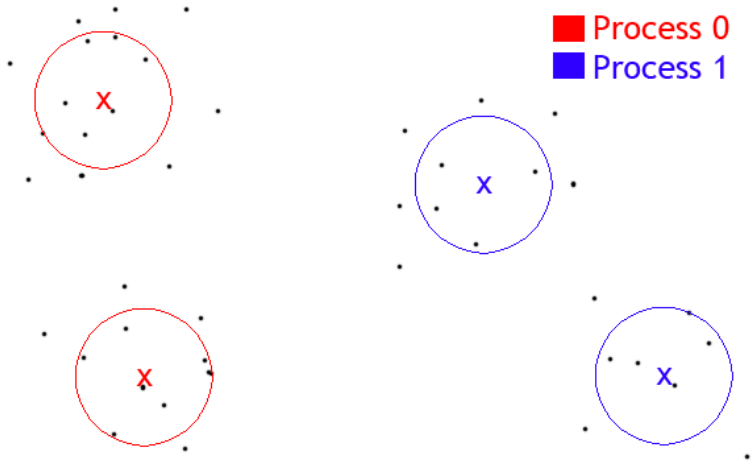




1. By using K-means clustering one can obtain a frequency and a cloud classification based on:
   1. Cloud Optical Thickness
   2. Cloud Top Height

- FORTRAN → Python 3
  - Python is a higher level language making it easier to read and write.
  - Bindings for FORTRAN can be written using `f2py`.
  - Python has a high level support for reading input data and writing output via libraries.

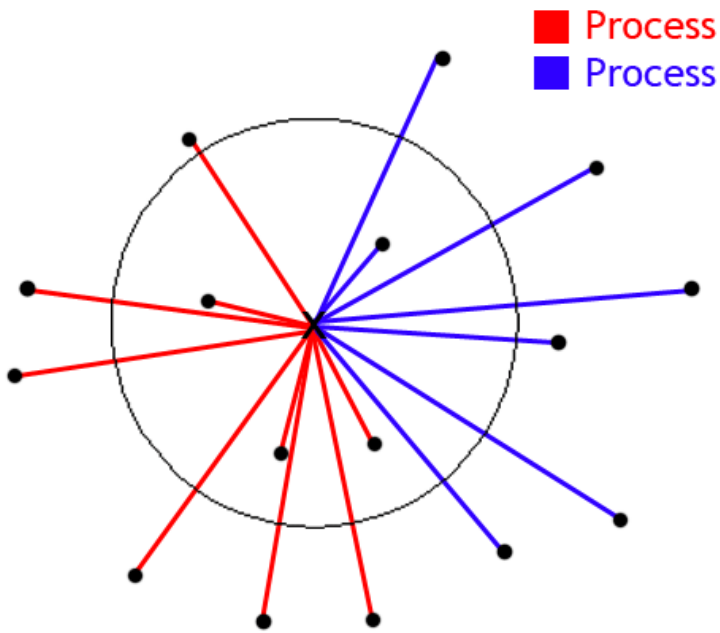

Rank 0
Rank 1
Partial Work
Partial Work
Rank 0
Merge Partials
Rank 1



- High Level Approach
  - Introduce MPI into the Python portion using `mpi4py`
  - Use the independent nature of the clusters for parallelism.
    1. Distribute the some clusters to each process
    2. The initial clusters are calculated on Process 0 to protect against initial cluster calculation changes. (Probabilistic vs Deterministic Initializations).
  - Split all other python loops such that anything that would rely on distributed results only do partial calculations.

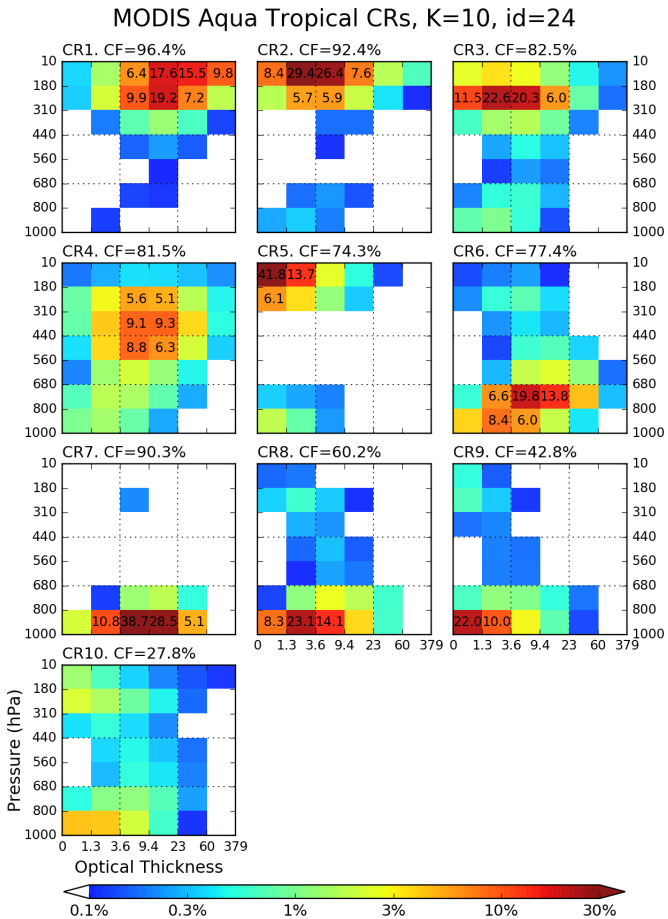
Implementation failed.

1. Parallelism is limited to number of clusters which has been capped to $k = 10$.
2. Concept couldn't scale with the data.
   Parallelism $\propto k$
   Data$\uparrow$ Time$\uparrow$.

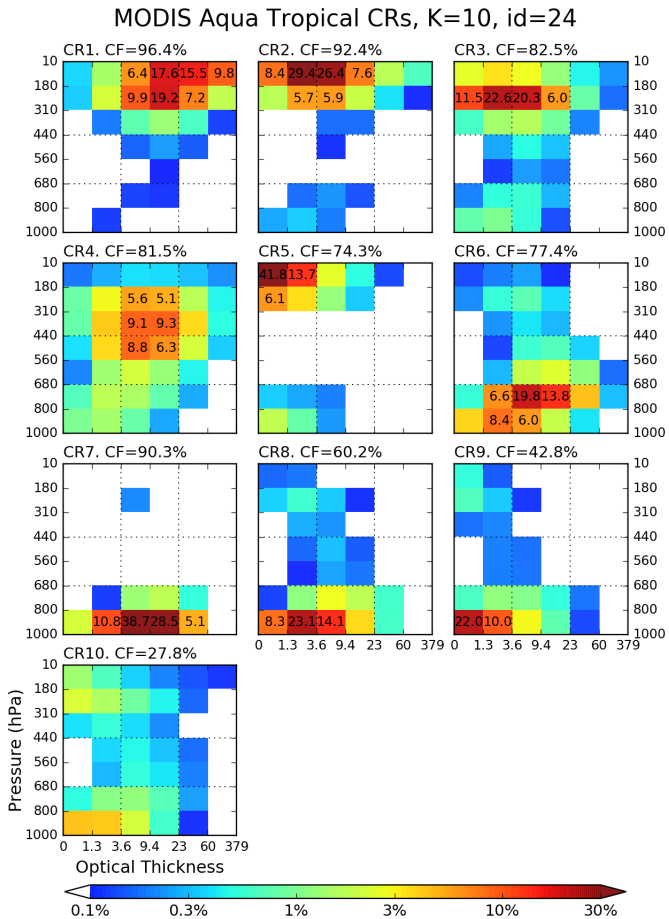


- Parallelism↑ as Data↑ thus Time≈↓
- Texas Advanced Computing Center idea, MPI $\propto$ OpenMP. $p_{ppn}/\text{cpus}_{ppn} = t_{ppn}$
- MPI introduced into the wrapped FORTRAN.
  - Parallelize the "Hot loops" that OpenMP handles with MPI.

MPI Result Plots: K = 10



MPI+OpenMP



OpenMP

Spark Implementation

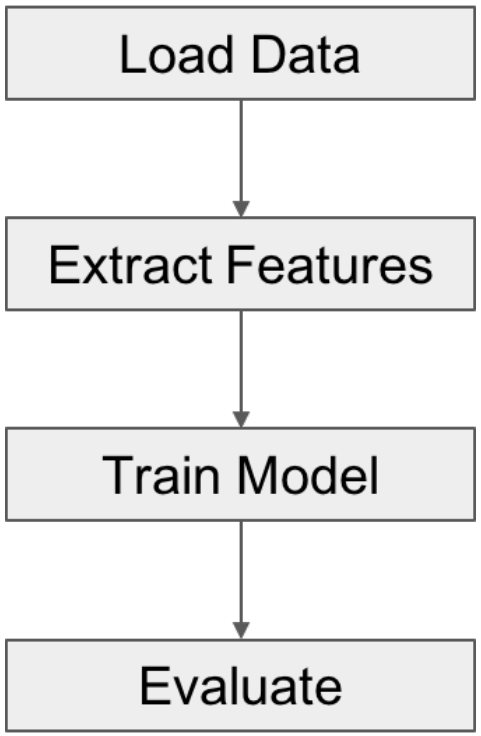
- Apache Spark™ is a unified analytics engine for large-scale data processing.
  - Fast Speed: a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine
  - Easy of Use: support Python, R, Scala, Java, SQL
  - Runs everywhere: Runs on Hadoop, standalone, etc.
  - Access diverse data sources including HDFS, Cassandra, HBase, and S3
- SparkMLlib: Large-scale machine learning on Apache Spark
- MLlib's mission is to make practical machine learning easy and scalable.
  - Easy to build machine learning applications
  - Capable of learning from large-scale datasets

K-Means Clustering in Spark: Scalable K-Means++

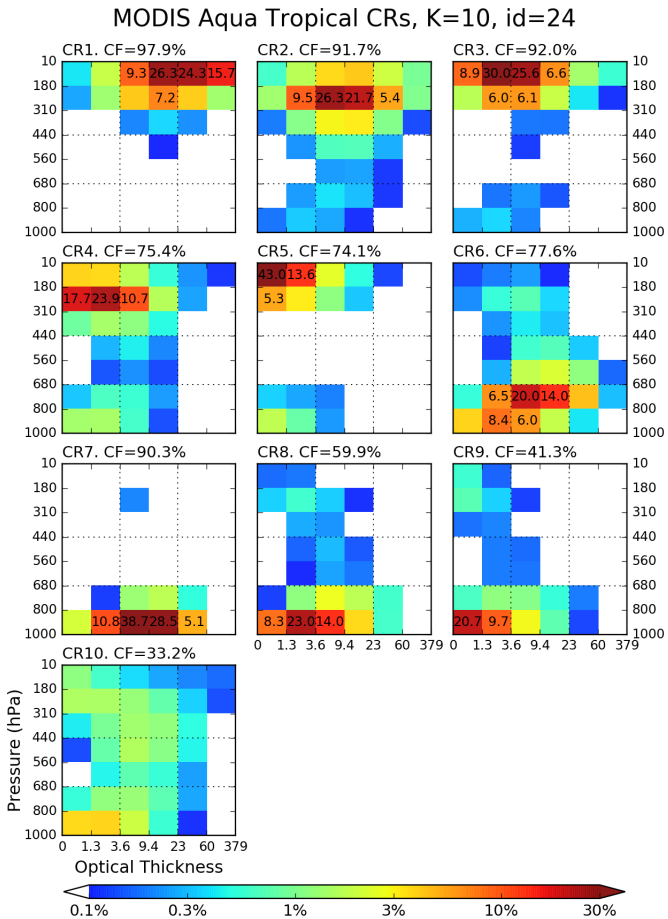
- K-means is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters.
- In data mining, k-means++ is an algorithm for choosing the initial values (or "seeds") for the k-means clustering algorithm.
- The MLlib implementation includes a parallelized variant of the k-means++ method called kmeans||.
- Experimental evaluation on real world large-scale data demonstrates that k-means|| outperforms k-means++ in both sequential and parallel settings.

Spark ML Workflow

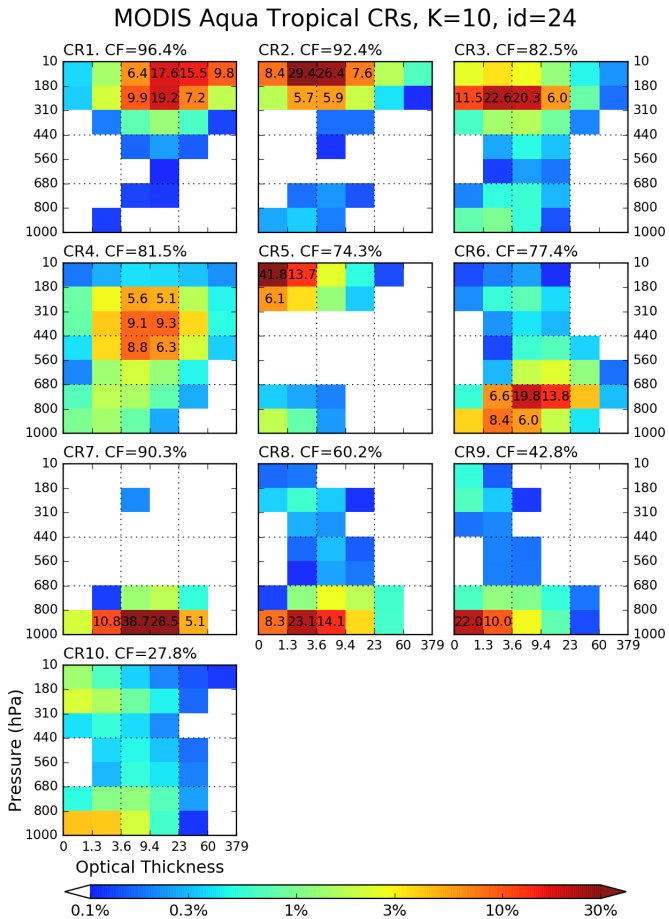
- Load Data: 1 year data (2005) from 15 N to 15 S in binary format.
  - We transformed data to .csv format and load into Spark
  - DataFrame: 42 columns, 3445612 rows.
- Extract Features: 42 columns are assembled as feature vector
- Train Model: Train K-means clustering model in Spark using K = 10.
- Evaluate: Compute euclidean distance



ML Result Plots: K = 10



SparkML



OpenMP FORTRAN

Conclusions

- Dr. Jin's original code could easily be expanded through MPI if it was integrated into the OpenMP Fortran code and then linked back to Python.
- The MPI produced identical results.
- His concept on clustering clouds was easily implemented using SparkML and the K-Means|| implementation.
- SparkML predicted the cloud regime in the tropics well
- We found dominant cloud regime as cumulonimbus found in the tropics.

Future Work

- Split the data across MPI Processes to support more data and re-implement FORTRAN code in Cython for C-performance and native numpy integration with Cython bindings to MPI.
- Use our Spark machine learning parallel program to cluster cloud regime based on more years of data.
- Do analysis of our new results and study the characteristics of clouds in different regions of Earth.

References

- Rossow WB, Schiffer RA (1999) Advances in understanding clouds from ISCCP. Bull Am Meteorol Soc 80:2261-2287. doi:10.1175/1520-0477(1999)080<2261:AIUCFI>2.0.CO;2
- Apache Spark https://spark.apache.org/
- GitHub Repository: https://github.com/AmericanEnglish/K-means_Clustering4CloudHistogram
- https://atrain.nasa.gov/images/historical/A-Train_Mar2003.jpg
- Zhang, Z. (2018). Class Lectures:Big Data+HPC+Atmospheric Science, Module 4: Overview of Earth's Atmosphere and Radiative Energy Budget.
- Oreopoulos L, Cho N, Lee D, Kato S, Huffman GJ.(2014) An examination of the nature of global MODIS cloud regimes. Journal of Geophysical Research: Atmospheres.;119(13):8362-83.
- Oreopoulos L, Cho N, Lee D, Kato S.(2016) Radiative effects of global MODIS cloud regimes. Journal of Geophysical Research: Atmospheres.;121(5):2299-317.