Cloud-Computing for Large Geospatial Datasets

Lauren Mabe - Geography Graduate Group

Agenda

1. About me/my research
2. Parallel computing
3. Practice using doParallel package in R
4. What is is cloud computing/Microsoft Azure
5. Demonstration of doAzureParallel package

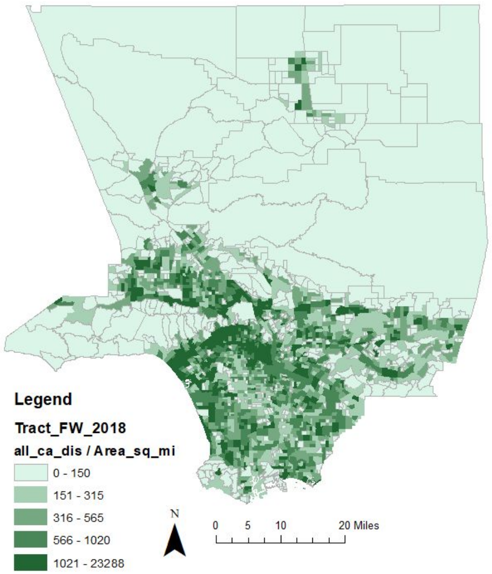
My research

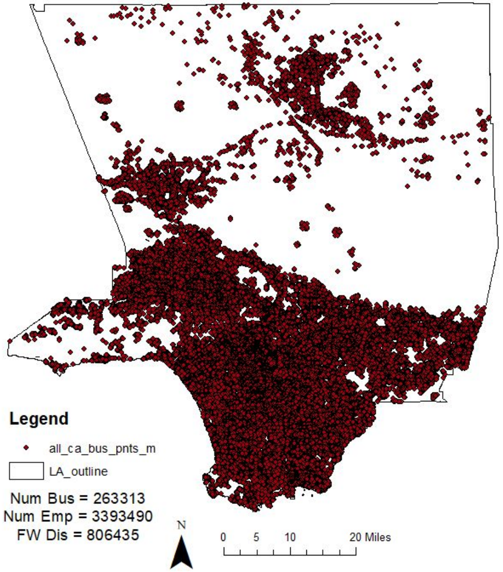
My research - *“Spatial optimization of food waste recycling infrastructure in California”*

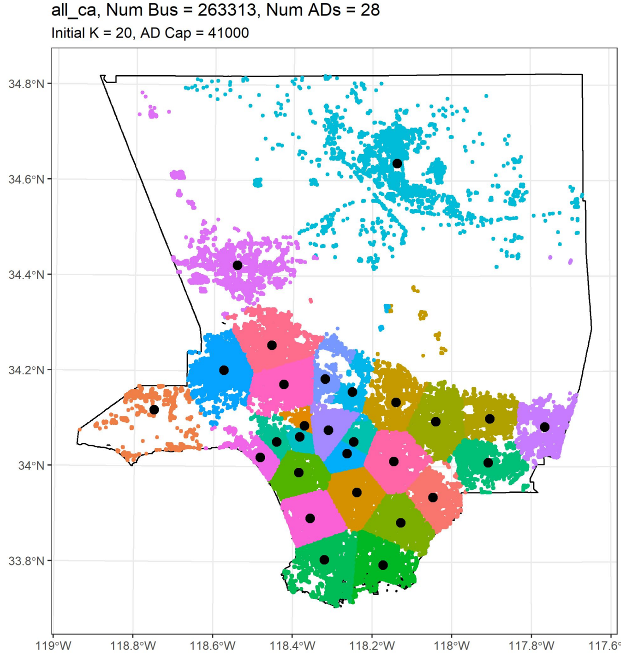
- In response to California SB 1383 which requires a 70% reduction in organic waste in landfills by 2025
- Current infrastructure is inadequate to treat the increased amount organic waste diverted from landfills
- Small scale, containerized anaerobic digesters can be used to convert food waste to biogas and a nutrient-rich digestate that can be used as fertilizer
- This research proposes a decentralized network of ADs to treat this food waste, using the spatial characteristics of the waste generation itself to optimize the network to reduce GHG

AD Optimization Model

After estimating locations of food waste generators (businesses), the model first places a series of ADs using a modified k-means clustering method







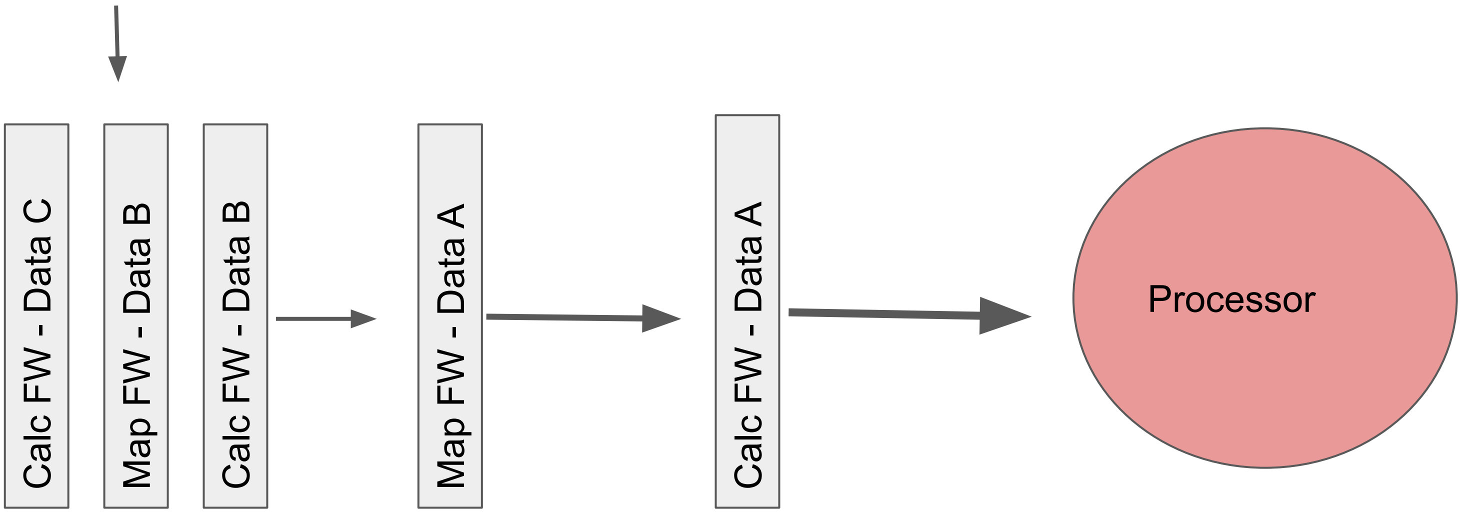
Why Am I Teaching This Workshop?

- K means clustering is an unsupervised method that is dependent on an initial random allocation of groups.
    - To test the operation of an unsupervised model, a Monte Carlo simulation is necessary.
    - With over 263,000 businesses to be clustered, this takes a long time.
- There are methods to speed up the processing of code:
    - Code optimization - synthesizing code to make it consume fewer resources
        - difficult, requires deep evaluation of your code. Possibly have to rewrite code :(
    - Parallel computing - adding more compute resources to run code faster
        - Learn new skills, advance computer knowledge, fun!
- Mainly, want to share this knowledge and help others accomplish their research goals

What is Parallel Computing? - Serial Computing

```
myList <- c(A,B,C)
For (data in myList) {
   CalcFW(data)
   MapFW(data)
}
```

- Serial computing uses one processor (core)
- Lines of code are sent to processor in sequence
- Each line of code must finish before processor moves on to the next



Calc FW - Data C

Map FW - Data B

Calc FW - Data B

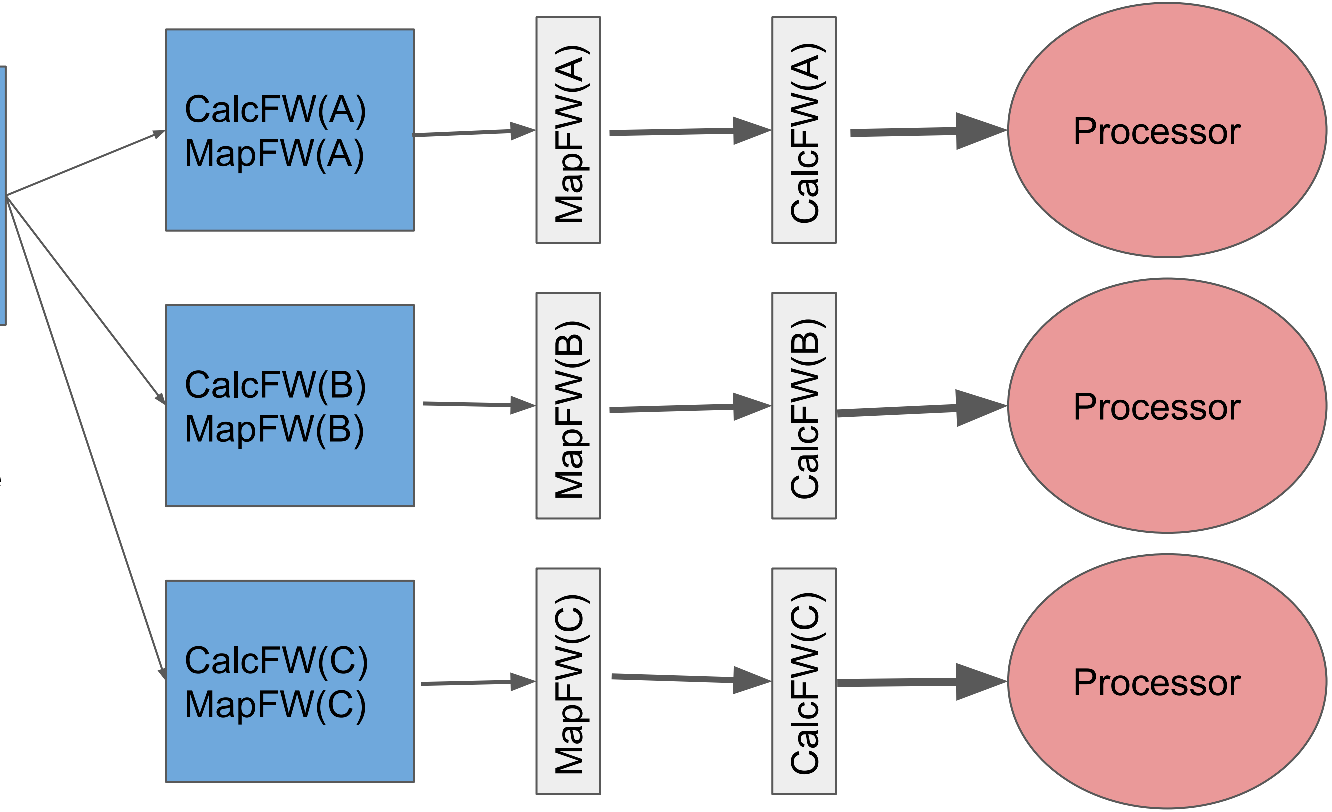
Map FW - Data A

Calc FW - Data A

Processor

What is Parallel Computing

```
myList <- c(A,B,C)
For (data in myList) {
   CalcFW(data)
   MapFW(data)
}
```



- Parallel processing takes the sequence of instructions and breaks them apart across multiple processors

Types of Parallel Computing

- Local Parallel Computing
  - Most modern computers (including laptops) have multiple processors
  - These are sometimes called cores (EX: Intel i7 Core Processor)
  - Code is split up among the different cores in your computer
- Cluster/Network Computing
  - Code is split up among multiple computers connected in a network
  - These clusters are available through a number of departments at UC Davis
- Cloud Computing
  - Code is sent to the “cloud” through the internet
  - This code is split up among many nodes in the services servers
  - Amazon Web Services, Google Cloud, Microsoft Azure

Why do we use parallel computing?

1. Speeds up processing time of large datasets by devoting more computer resources to the problem
2. Local parallel computing takes full advantage of your expensive laptop
3. Cluster/cloud versions speed up workflow even more
   a. Devotes more resources to the computing problem
   b. Allows you to send data to cloud while you work on other parts of your code

When do we use parallel computing?

Parallel computing is used in any field that requires the processing of large datasets

The computing problem should be:

1. Able to be broken down into discrete pieces of work
   a. loops, lapply situations
2. The process can be executed at once
   a. Iterations are not dependent on the results of previous one
3. Can be solved quicker using parallel resources
   a. There is overhead time to set up parallel/cloud resources

doParallel Package in R

- The doParallel package is a “parallel backend” for the foreach package
    - It allows the user to run foreach loops in parallel on your local computer

Setup is very simple:

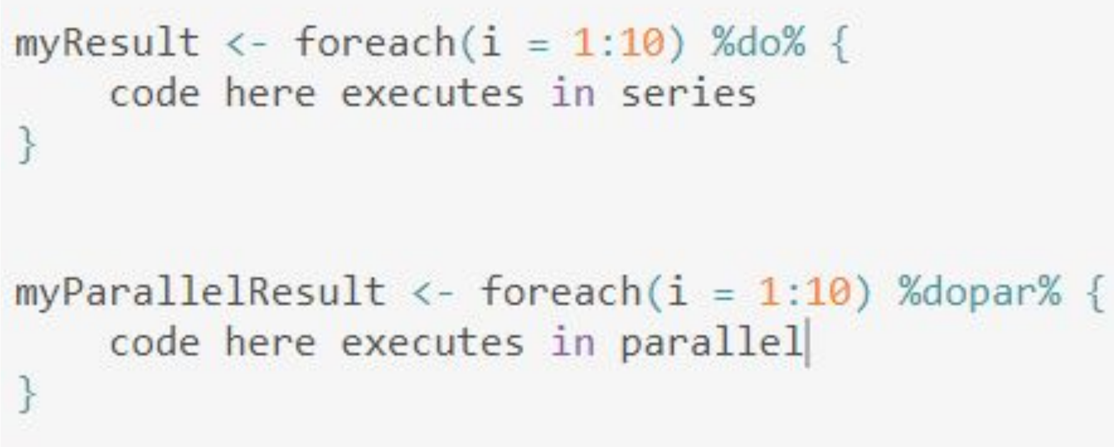
1. Create a cluster - makeCluster()
    a. this is specifying which of your computer's cores are designated for parallel computing
    b. Cores specified as part of the cluster are called workers
2. Register the parallel backend - registerDoParallel()
    a. this “connects” the cluster to R

Using the foreach loop - the basics

The foreach loop makes it easy to switch between series and parallel execution

The foreach loop operates much like a function, returning the final line as the result.

By default, this result is a list



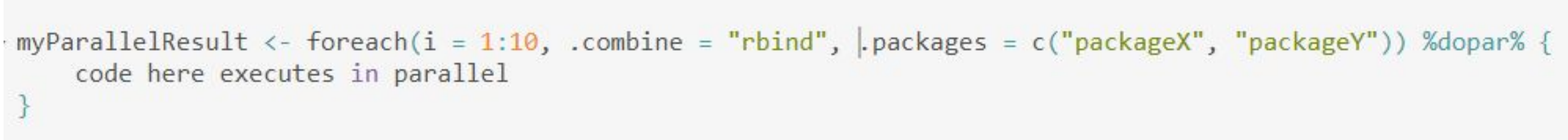
```
myResult <- foreach(i = 1:10) %do% {
    code here executes in series
}


myParallelResult <- foreach(i = 1:10) %dopar% {
    code here executes in parallel
}
```

Using the foreach loop - medium

By specifying certain parameters within the foreach loop, we can control its function and results

The two most important ones are .combine and .packages



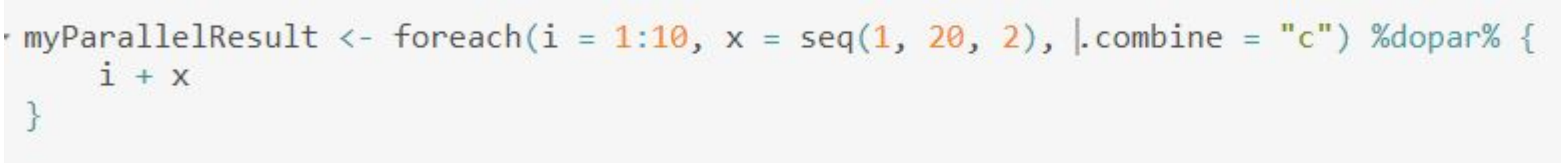
```
myParallelResult <- foreach(i = 1:10, .combine = "rbind", .packages = c("packageX", "packageY")) %dopar% {
    code here executes in parallel
}
```

.combine controls the way the results of the loop are returned

.packages installs necessary packages onto each core

Using the foreach loop - advanced

- Multiple iteration variables (i & x) can be used within the code itself
- Both of these variables are changing at the same time
- However, the first iteration variable (i) is the one controlling the loops iterations



```
myParallelResult <- foreach(i = 1:10, x = seq(1, 20, 2), .combine = "c") %dopar% {
    i + x
}
```

There are also more combine arguments: .multicombine, .maxcombine, .inorder

Let’s practice!

Cloud Computing in Microsoft Azure Batch

- Microsoft Azure Batch is a paid parallel cloud computing service that connects to R using the doAzureParallel package
- doAzureParallel is based on the doParallel package and runs similarly
- Why are we using Microsoft Azure? Why not Amazon/Google/etc?
  - The doAzureParallel package makes it easy to switch between local and cloud computing
  - Allows you to use your local RStudio with the cloud service- no virtual machine needed
  - It’s the one I figured out first - the tutorial is easy to follow

A note on the costs of Azure Batch: There is a 1 month free trial available, afterwards you pay ~$0.10/hour. Lower cost “low-priority” VMs are available

doAzureParallel package in R

The doAzureParallel package sets up the “cloud backend” for the foreach package

Setup is similar to doParallel, but requires a couple more steps

1. Create Microsoft Batch and Storage Accounts
    a. These are free
2. Register your credentials
    a. This tells R that you have valid Batch/Storage accounts
3. Configure your cluster
4. Create your cluster
    a. This creates space for your code in the cloud. (This takes a while!)
5. Register your cluster
    a. This connects the cloud to your RStudio

Lets Practice!

(Kinda)

Thank you!

If you have any questions in the future feel free to contact me at lmabe@ucdavis.edu