Using Multiprocessing in Python to Decrease Map Production Time

Jeff Nothwehr

National Drought Mitigation Center
University of Nebraska-Lincoln

Overview

- About multi-processing
- How it works
- Implementation
- Results
- Potential issues and solutions
- Other tips

What is Multiprocessing?

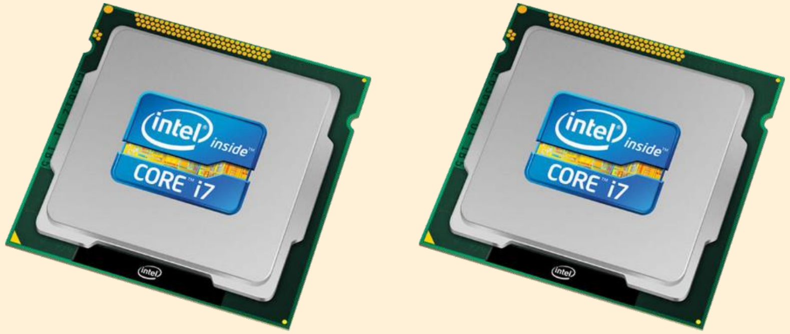
What is Multiprocessing?

Well, first of all, you need a machine with…

…a multi-core CPU

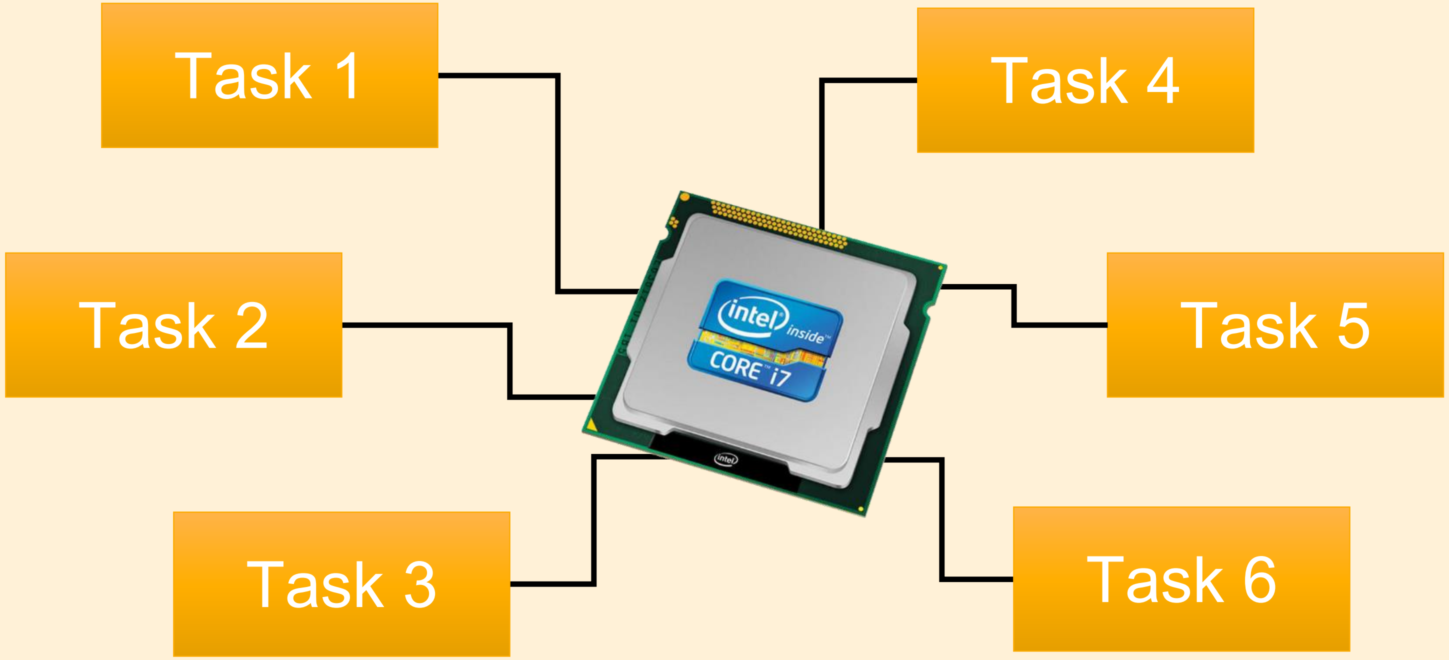
and/or

…multiple CPUs



What is Multiprocessing?

Multi-processing runs multiple tasks in the code at the same time (one per thread)



How does Multiprocessing Work?

How does Multiprocessing Work?

Python contains a multiprocessing library

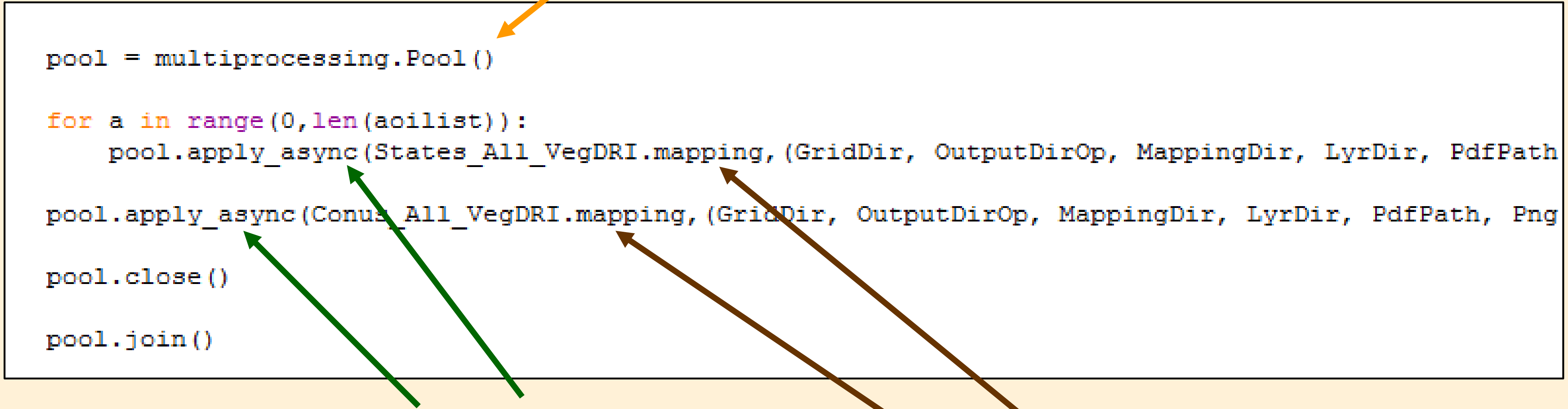
Can create a pool of worker processes

Assign one task to each worker in the “pool”

Tasks executed simultaneously assuming there are processor threads available

How does Multiprocessing Work?

Define worker pool

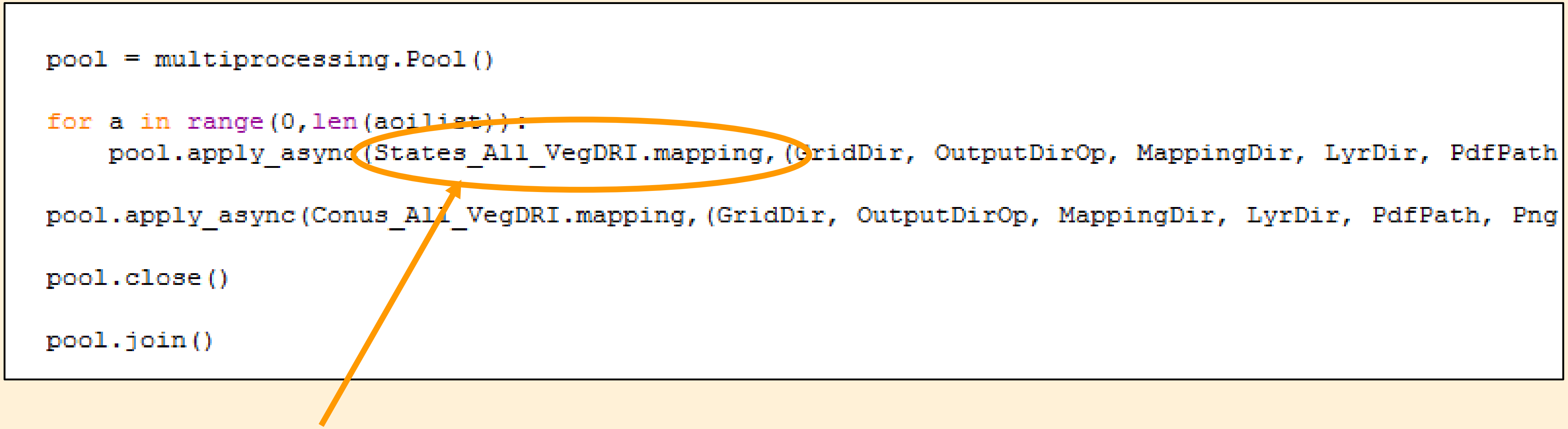


```
pool = multiprocessing.Pool()

for a in range(0,len(aoilist)):
    pool.apply_async(States_All_VegDRI.mapping,(GridDir, OutputDirOp, MappingDir, LyrDir, PdfPath

pool.apply_async(Conus_All_VegDRI.mapping,(GridDir, OutputDirOp, MappingDir, LyrDir, PdfPath, Png

pool.close()

pool.join()
```

Assign tasks to workers

Function (inputs)

How does Multiprocessing Work?

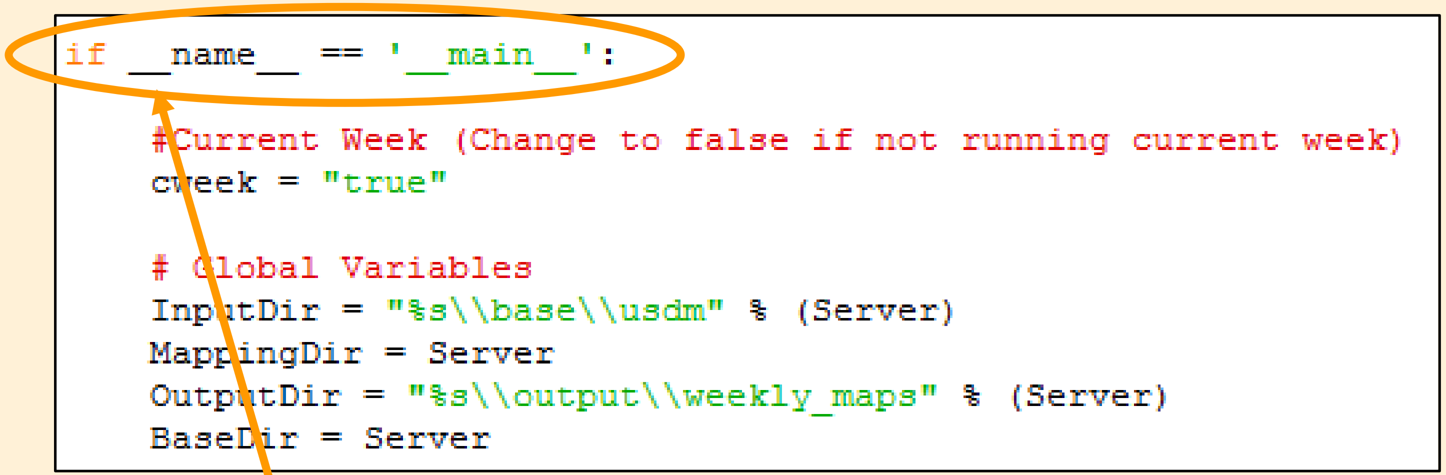


```
pool = multiprocessing.Pool()

for a in range(0,len(aoilist)):
    pool.apply_async(States_All_VegDRI.mapping,(GridDir, OutputDirOp, MappingDir, LyrDir, PdfPath

pool.apply_async(Conus_All_VegDRI.mapping,(GridDir, OutputDirOp, MappingDir, LyrDir, PdfPath, Png

pool.close()

pool.join()
```

One instance of this subproccess is run by each processer thread

How does Multiprocessing Work?

Must be run in the main thread



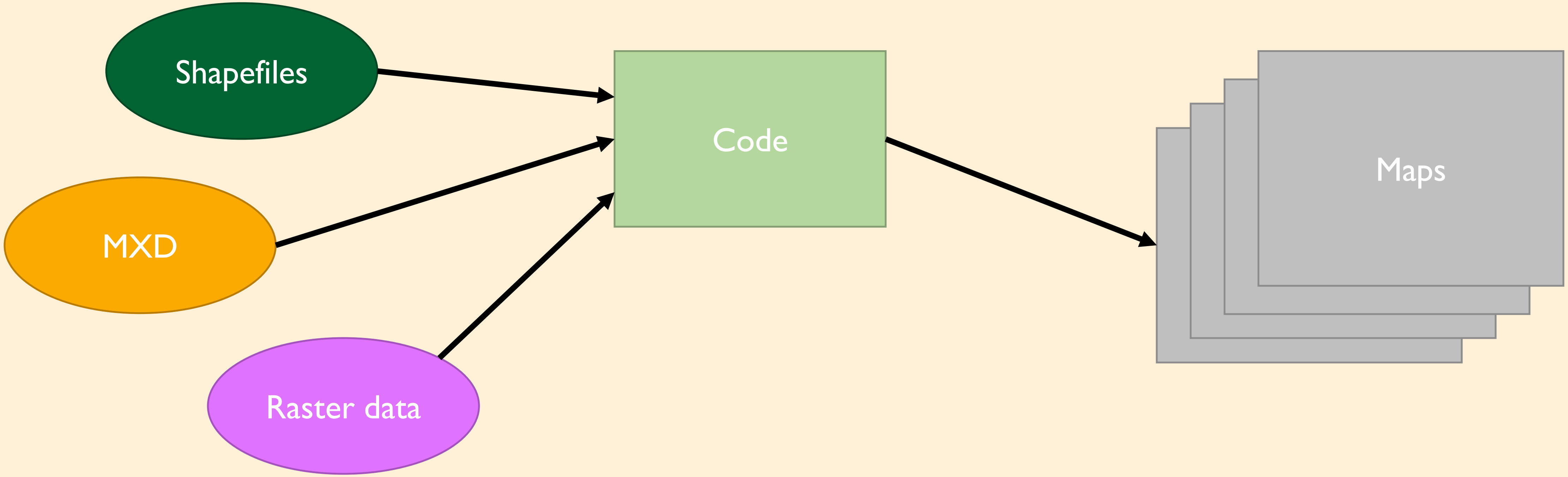
```
if __name__ == '__main__':

    #Current Week (Change to false if not running current week)
    cweek = "true"

    # Global Variables
    InputDir = "%s\\base\\usdm" % (Server)
    MappingDir = Server
    OutputDir = "%s\\output\\weekly_maps" % (Server)
    BaseDir = Server
```

Indicates to run code if part of the main thread

Implementation

Implementation

Can use scripts to produce maps in ArcMap



Implementation

Example: The U.S. Drought Monitor weekly production

1006 PNGs

932 PDFs

933 JPGs

12 other formats

=

A lot of maps!

(2883 to be exact)

Implementation

Currently takes 2 hours to produce

Only 15 hours to finish entire process before public release

Implementation

U.S. Drought Monitor processes that run:

- Shapefile processing
- Statistics
- Change maps

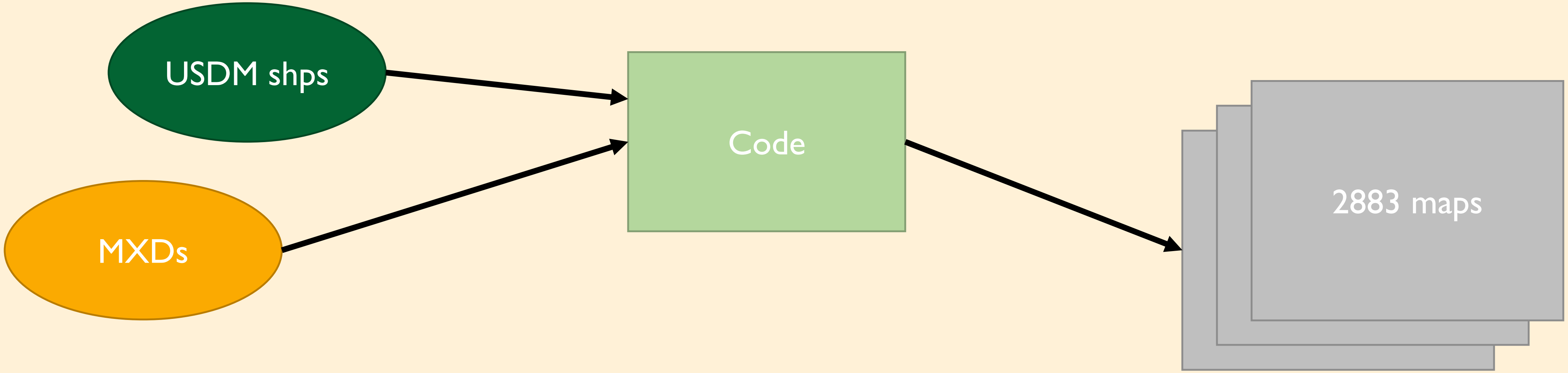
Implementation

Increase efficiency

Decrease production time

Implementation

Applied multiprocessing to map production process



Results

Results

Decreased U.S. Drought Monitor production time:

2 hours → **20 minutes**

Exact same number of maps produced

Results

Decreased U.S. Drought Monitor production time:

83% time savings

Exact same number of maps produced

Results

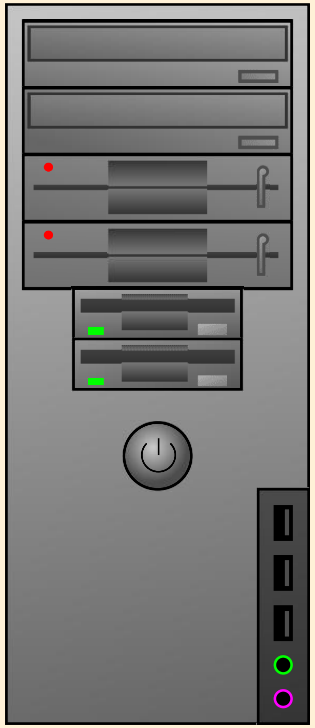
Machine specs for this trial:

Intel Core i7 4790 Processor

4 cores, 8 threads

16 GB RAM

Standard desktop



Results

Increase number of areas mapped:

Over 30,000 maps per week

(over 10 times as many)

Results

Decreased U.S. Drought Monitor production time:

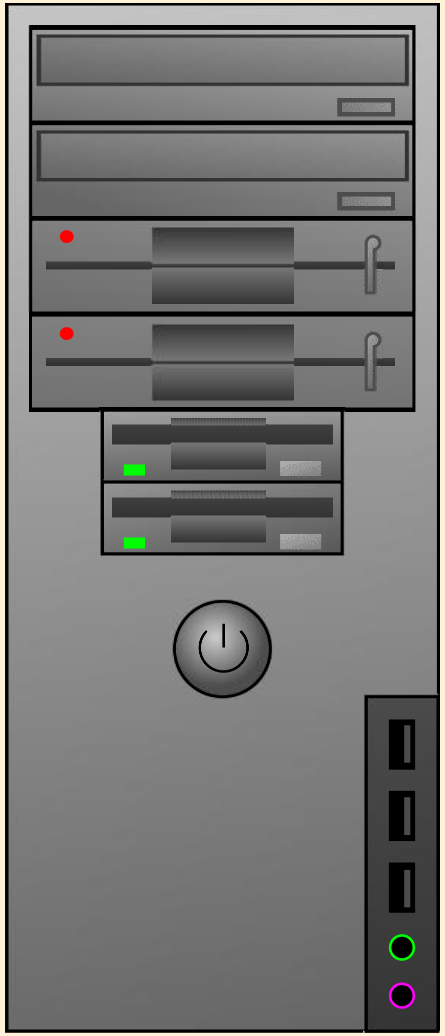
16 hours → **2 hours**

Would not be possible without multiprocessing

Results

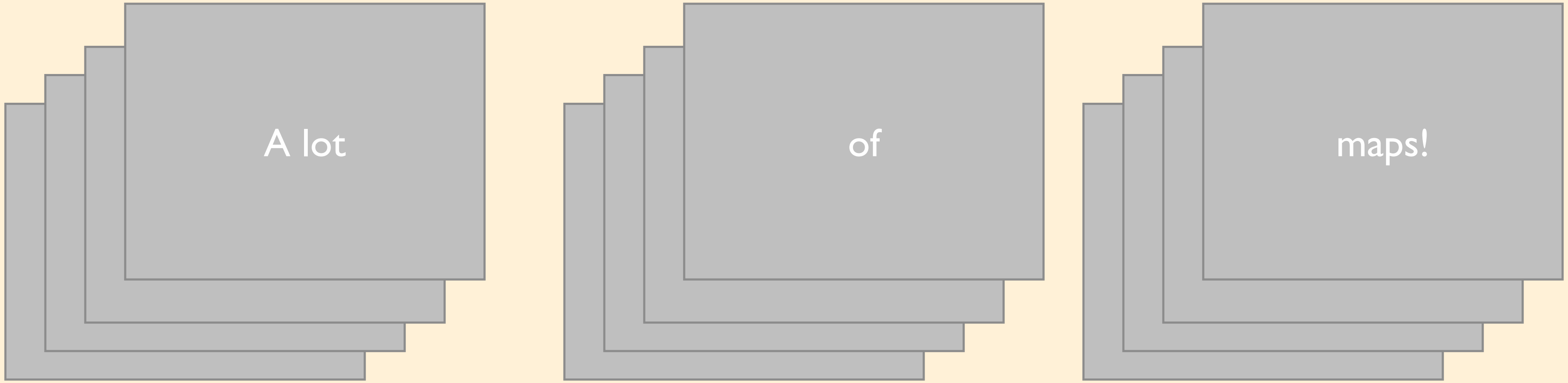
Implement this into production on a dedicated server:

- **Multiple processors**
  - **12 threads each, 24 total**
- **32 GB RAM**



Results

Start implementing phase two of map production (30,000+ maps per week)



A lot

of

maps!

Potential Issues and Solutions

Potential Issues and Solutions

Issues with locked datasets

Cannot edit a dataset that is in use by another process

Potential Issues and Solutions

Break datasets into multiple parts whenever possible

Start a second worker pool after the first one finishes

Make a copy of the dataset in memory and edit the copy

Other Tips

Other Tips

Break input datasets into multiple parts whenever possible

Try to run operations of similar time intervals at the same time:

Reduces amount of down time

Other Tips

Make all code is in main thread:

Code outside of main thread will get run again by each worker

Other Tips

Use a program other than IDLE to run the code:

IDLE won’t print out statements inside of worker processes

Pool worker processes run in background if you close the window during processing

Other Tips

Alternatives?

Run in a command window

Third party software such as PyCharm Community Edition

Other Tips

Monitor your processor cores while testing:

- Can tell you if processes are still running

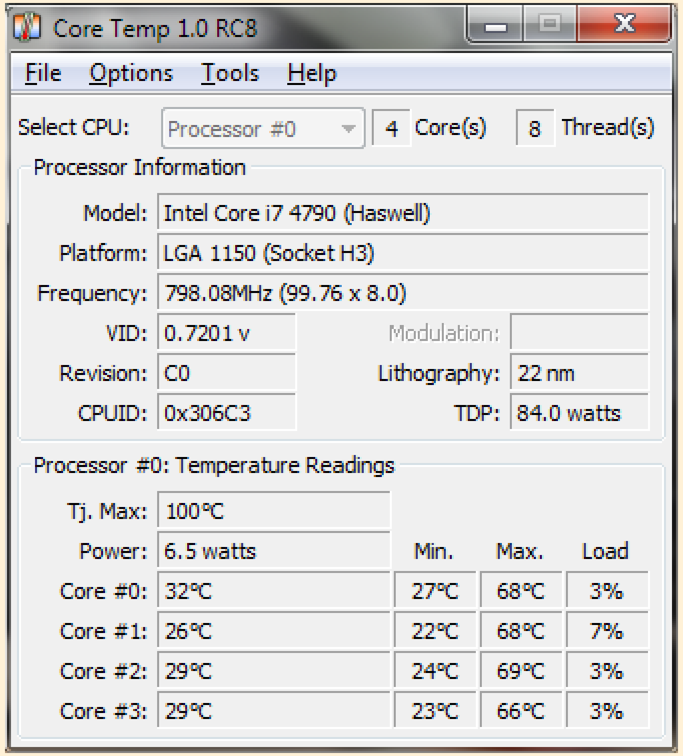
- Can see how many threads/cores are in use

Other Tips

How to monitor?

Windows Task Manager

Third party software (Core Temp)



Select CPU: Processor #0 | 4 Core(s) | 8 Thread(s)

Processor Information

Model: Intel Core i7 4790 (Haswell)

Platform: LGA 1150 (Socket H3)

Frequency: 798.08MHz (99.76 x 8.0)

VID: 0.7201 v | Modulation:

Revision: C0 | Lithography: 22 nm

CPUID: 0x306C3 | TDP: 84.0 watts

Processor #0: Temperature Readings

Tj. Max: 100°C

		Min.	Max.	Load
Power:	6.5 watts			
Core #0:	32°C	27°C	68°C	3%
Core #1:	26°C	22°C	68°C	7%
Core #2:	29°C	24°C	69°C	3%
Core #3:	29°C	23°C	66°C	3%

Other Tips

ESRI blog post about multiprocessing:

https://blogs.esri.com/esri/arcgis/2011/08/29/multiprocessing/

Questions

Ask now or contact:

Jeff Nothwehr
jnothwehr2@unl.edu