STATS 531
Introduction to R and pomp on Great Lakes

Dr Charles J Antonelli, LSA IT ARS
John Thiels, LSA IT ARS
March, 2020

Preliminaries

Using Great Lakes

- Three basic requirements:
  A Great Lakes user login
  [https://umichprod.service-now.com/sp?id=sc_cat_item&](https://umichprod.service-now.com/sp?id=sc_cat_item&sys_id=5dd92f33db57af40f0f52b35ca961974)
  [sys_id=5dd92f33db57af40f0f52b35ca961974](https://umichprod.service-now.com/sp?id=sc_cat_item&sys_id=5dd92f33db57af40f0f52b35ca961974)
  A Duo app on your smartphone or tablet
    http://its.umich.edu/two-factor-authentication
  A Slurm account
    `stats531w20_class`

- Logging in to Great Lakes
  `ssh -XY `*`uniqname`*`@greatlakes.arc-ts.umich.edu`
  Campus wired or MWireless
  VPN
  `ssh -XY `*`uniqname`*`@login.itd.umich.edu` first
    See Appendix to verify you have a login there

Connecting to Great Lakes (macOS)

Task: Connect to Flux from macOS

1. If there is a Terminal icon in the Dock , double-click it
    Otherwise:

    a. Type Command-Spacebar

    b. Enter "Terminal" without the quotes into Spotlight Search

    c. Double-click on the Terminal application

2. This creates a Terminal window on your local computer with a local command shell running inside it

3. Type
    `ssh -XY `*`uniqname@greatlakes.arc-ts.umich.edu`*
    where *uniqname* is your uniqname

4. Login with Kerberos password, and Duo

5. This creates a remote command shell on Great Lakes

Connecting to Great Lakes (Linux)

Task: Connect to Flux from Linux

1. Open a terminal window

2. Type
   `ssh -XY `*`uniqname@greatlakes.arc-ts.umich.edu`*
   where *uniqname* is your uniqname

3. Login with Kerberos password, and Duo

4. This creates a remote command shell on Great Lakes

Connecting to Great Lakes (Windows)

Task: Connect to Flux from Windows

1. Get the PuTTY/WinSCP installer for your flavor of Windows:
    Compute at the U: http://its.umich.edu/computing/computers-software/compute
    Select + under Get Going
    Get U-M PuTTY (UM_PuTTY_WinSCP.zip)
    Execute the installer (UM_PuTTY_WinSCP.exe)
    Accept all defaults

2. Double-click "UM Internet Access Kit" icon on Desktop
    Double-click PuTTY application within

3. In the Putty Application that appears:
    Enter `greatlakes.arc-ts.umich.edu` in the Host Name box
    Select the + preceding SSH in Category box on left
    Click X11 that appears below Auth below SSH
    Check the "Enable X11 forwarding" checkbox on the right
    Click "Open" at bottom
    Click Yes at bottom of the PuTTY Security Alert popup if it appears

4. In the terminal window that appears:
    Login with uniqname, Kerberos password, and Duo

5. This creates a remote command shell on Great Lakes

Connecting to Great Lakes (Windows)

To display graphics generated by applications running on Great Lakes on your local machine, perform this step as well:

6. If you don't see this icon in your system tray: or
   Install the Xming X Server for Windows
   https://sourceforge.net/projects/xming/files/Xming/6.9.0.31/
   Get the Xming installer (Xming-6-9-0-31-setup.exe)
   Execute the installer
   Accept all defaults

   Optional:
   https://sourceforge.net/projects/xming/files/Xming-fonts/7.7.0.10/
   Get the XMing fonts installer (Xming-fonts-7-7-0-10-setup.exe)
   Execute the installer
   Accept all defaults

Typing at the shell prompt

- Basic input line editing commands
  - Backspace erases previous character
  - Left and right arrow move insertion point on the line
  - Control-U erases the line to the insertion point, so you can start over
  - Enter executes the line you typed
  - Control-C interrupts whatever command you started and returns you to the shell prompt (usually)
  - Up and down arrow will access your *command history*
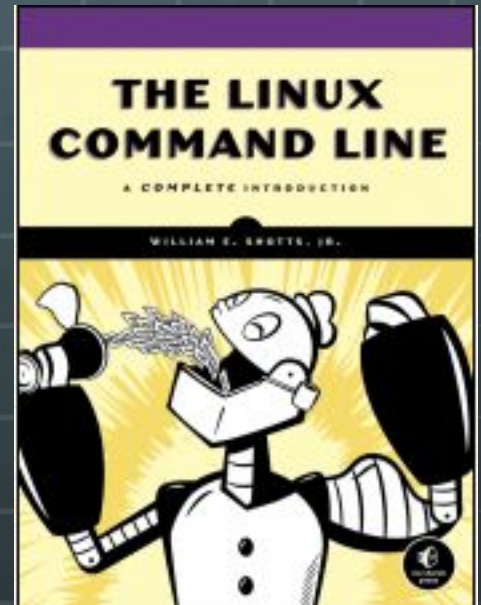 - Type "exit" without the quotes to exit the shell

Command Line Reference

William E Shotts, Jr.,
"The Linux Command Line: A Complete Introduction,"
No Starch Press, January 2012.
http://linuxcommand.org/tlcl.php .

Download Creative Commons Licensed
version at
http://downloads.sourceforge.net/project/
linuxcommand/TLCL/13.07/TLCL-13.07.pdf

Cluster batch workflow

- You create a batch script and submit it as a job

- Your job is scheduled, and it enters the queue

- When its turn arrives, your job will execute the batch script

- Your script has access to all applications and data

- When your script completes, anything it sent to standard output and error are saved in files stored in your submission directory

- You can ask that email be sent to you when your jobs starts, ends, or fails

- You can check on the status of your job at any time, or delete it if it's not doing what you want

- A short time after your job completes, it disappears

Basic batch commands

Submit a job

```
sbatch sample.sbat
```

Query job status

```
squeue -j jobid
squeue -u uniqname
```

Delete a job

```
scancel jobid
```

Check a job script and estimate its start time

```
sbatch --test-only sample.sbat
```

11

Useful batch commands

`sacct -u user`   *show recent job history*

`seff jobid`   *show cpu utilization for jobid*

Parallel R and pomp

Random numbers

Serial pseudorandom sequence
    Usually based on linear recurrences modulo m
    Initialize the generator with some seed s
    Generate stream of pseudorandom numbers

Parallel pseudorandom sequence
    Performance, Reproducibility, Serializability

Install pomp

Task: Install pomp package

- `module load R`

- `R`
  `install.packages("pomp",repos="https://repo.miserver.it.umich.edu/cran")`
  (answer y to "Would you like to use a personal library instead?" and "Would you like to create a personal library")
  `install.packages("ggplot2",repos="https://repo.miserver.it.umich.edu/cran")`

  List of mirrors: https://cran.r-project.org/mirrors.html

Multicore pomp (1)

```
# Rpomp-par.R
rm(list=ls())
library(doParallel)
set.seed(2020,kind="L'Ecuyer")
cores <-
as.numeric(Sys.getenv('SLURM_NTASKS_PER_NODE',
unset='8'))
cl <- makeCluster(cores)
registerDoParallel(cl)
```

Multicore pomp (2)

```
# Rpomp-par.R continued...
trials = 100
system.time(
  r <- foreach(icount(trials),
  .inorder=FALSE,
  .options.multicore=list(set.seed=TRUE)
  ) %dopar% {
   library(pomp)
   for (i in 1:100) {gom=gompertz()}
  }
)
stopCluster(cl)
save(r,file="sims.Rda")
```

Lab: Submit a batch job

Submit a job to Great Lakes, check its status and output

Copy sample code:

```
cp -rp /sw/examples/workshops/IntroGreatLakes/R-examples ~
cd ~/R-examples
```

Execute these commands:

```
sbatch Rpomp-par.sbat      # submit the job
squeue -j jobid            # check status of job
less slurm-jobid.out       # look at script output
less Rpomp-par.out         # look at R output
```

Resources

- The [Great Lakes Slurm HPC Cluster](#)

- Great Lakes [User Guide](#) (start here!)

- Two-page [Cheat Sheet](#)

For assistance: hpc-support@umich.edu

- Read by a team of people including unit support staff
- Cannot help with programming questions, but can help with operational and basic usage questions
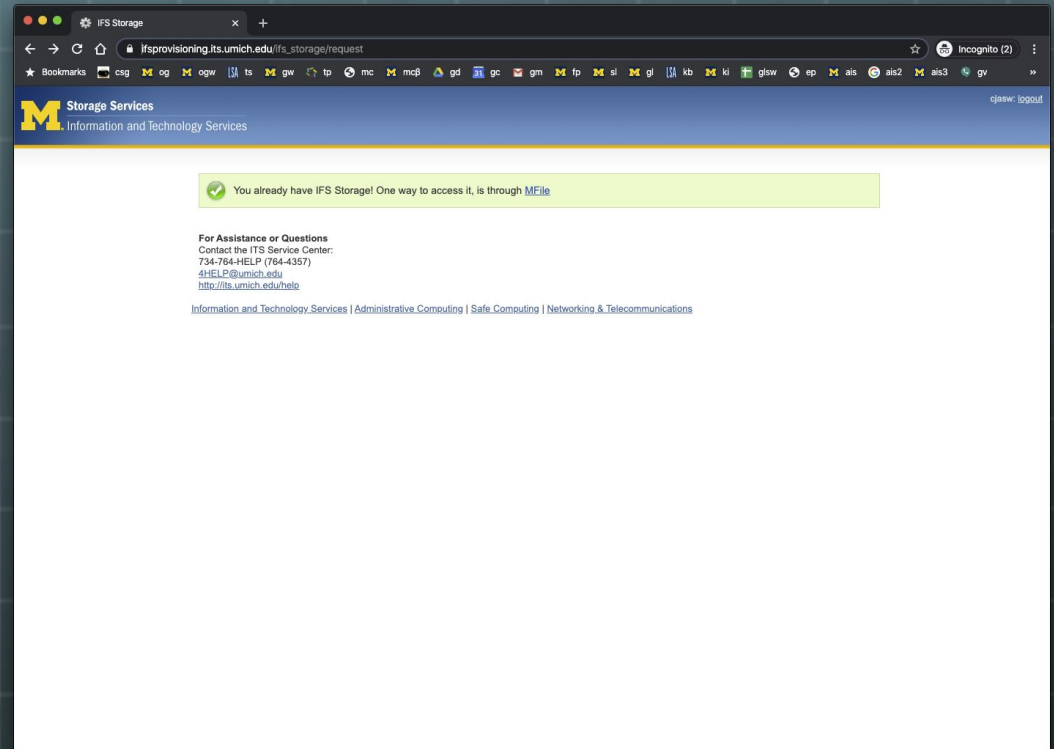
Appendix

Connecting to login.itd.umich.edu

If you are unable to ssh to `login.itd.umich.edu`, follow these steps to make sure you have a login there

1. Visit the following URL
   https://ifsprovisioning.its.umich.edu/ifs_storage/request
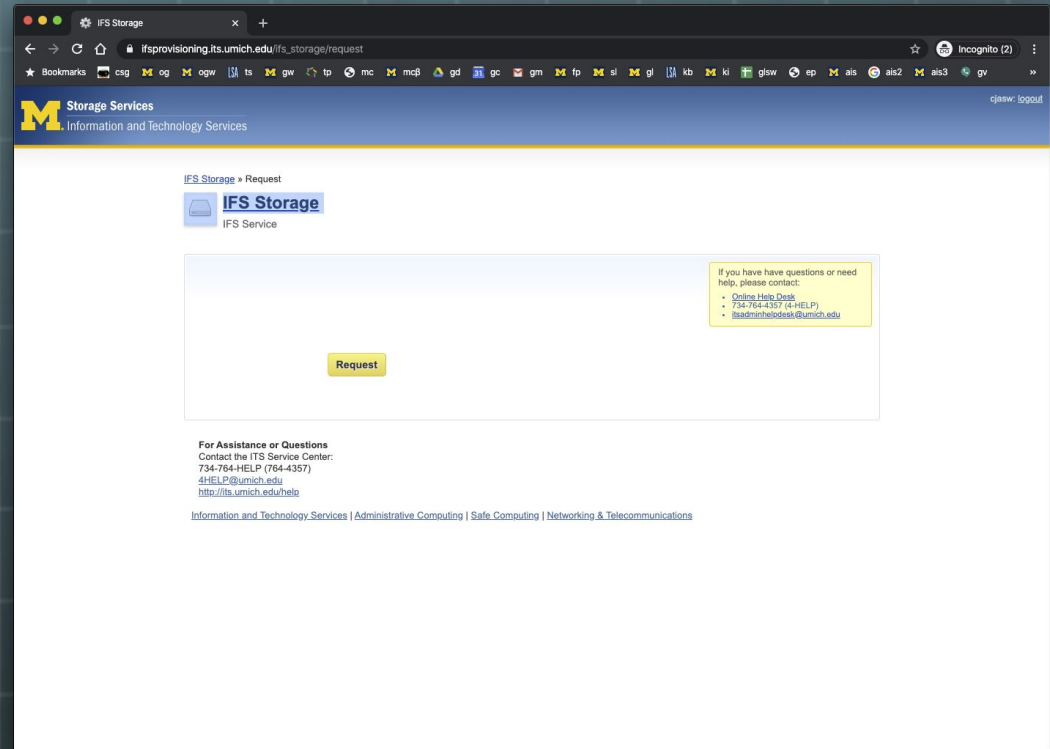
Connecting to login.itd.umich.edu

2. If you see this,
   you already have
   a login on
   login.itd.umich.edu

Connecting to login.itd.umich.edu

3. If you see this, you do not have a login on login.itd.umich.edu

4. Click the Request box

Connecting to login.itd.umich.edu

5. You should see this page

6. After about 20 minutes, you will have a login on login.itd.umich.edu