Adventures in Supercomputing with R

Lecture 5: Shared Memory (continued)

George Ostrouchov

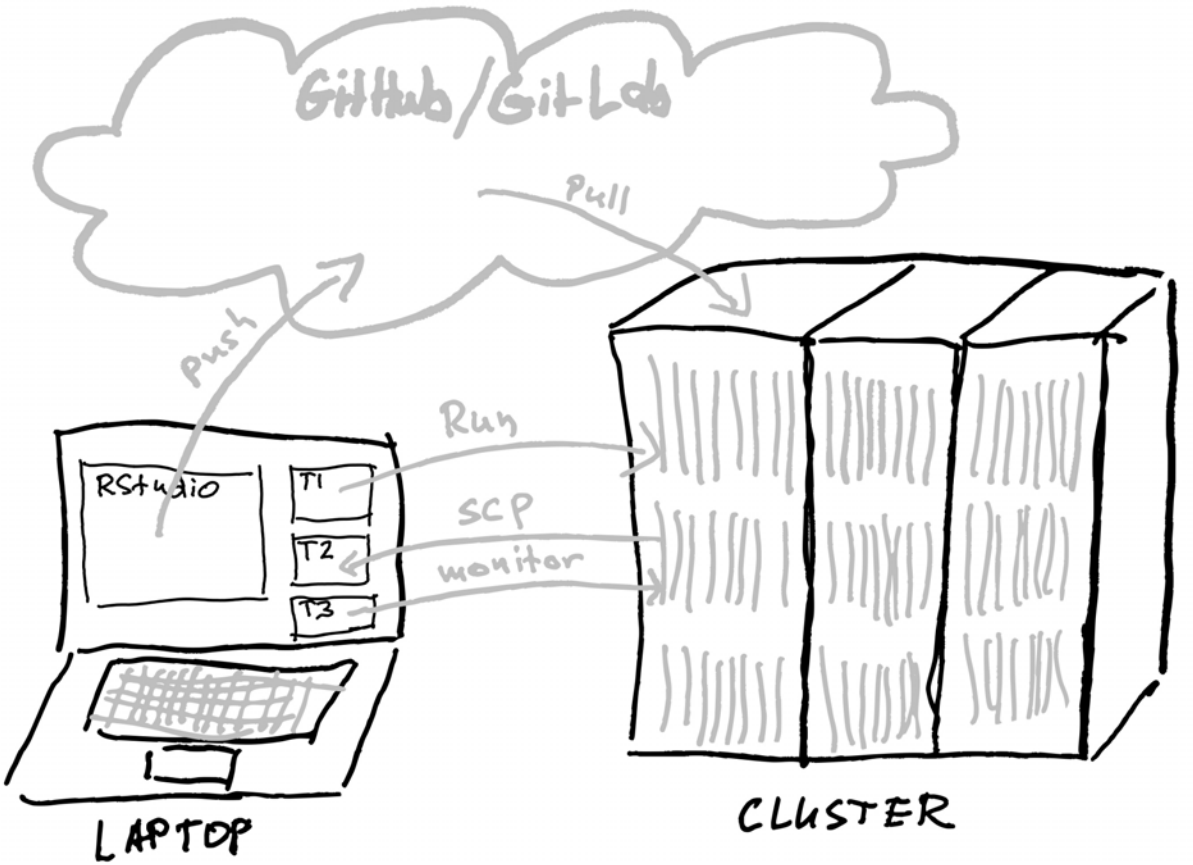
Oak Ridge National Laboratory and University of Tennessee

2022/02/20 (updated: 2022-03-16)

Review

- Bootstrap, Bagging, Boosting, and Crossvalidation
- Random forest variable subset size crossvalidation
- Matrix libraries introduction

Typical Workflow from Laptop to Cluster



GitHub Authentication on IT4I

Internet Protocols

A set of rules and formats governing network communication. Messages are split into packets have a specific set of binary formats.

ssh protocol

Secure Shell Protocol is a network protocol for secure services over an unsecured network, especially login and command line shell interface.

http (https) protocol

Hypertext Transfer Protocol (Secure) is an application layer network protocol for media information systems, especially as the basis for the World Wide Web.

GitHub Authentication on IT4I

- clone via **https**
  - public repository does not require authentication
  - private repository requires authentication
- clone via **ssh**
  - requires authentication but can be handled with ssh keys:
    1. Check your `.ssh` directory on IT4I for `id-rsa.pub`
    2. Add the public key to your GitHub SSH keys (Settings → SSH and GPG keys)
    3. Check that it works with `ssh -T git@github.com`
    4. Clone or re-clone your private repository using ssh protocol
       - `git clone git@github.com:your-github-username/your-private-repo.git`
    5. You should now be able to `git pull` without authentication
- clone via **gh CLI**
  - GitHub command line interface

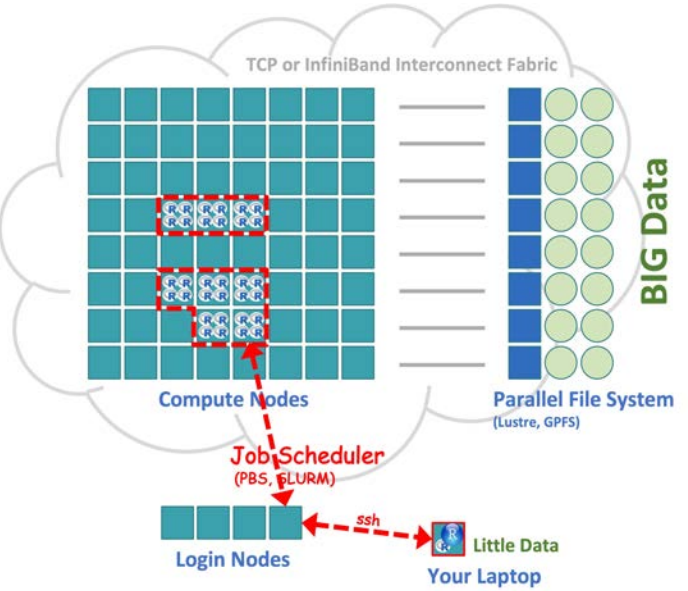
Job Information on IT4I

```
qsub your-script.sh

qstat -u your-user-id

check-pbs-jobs --check-all
```

The `check-pbs-jobs` gets compute node name. To monitor CPU and memory use from a login node:

```
ssh compute-node-name
top -u your-user-id
```

Can be done in a separate login window.



IT4I Job Management link

Matrix Libraries ...

R-LAPACK-BLAS

- BLAS: Basic Linear Algebra Subroutines - A matrix multiplication library
  - `%*%`, `crossprod()`, `sweep()`, `scale()`, and many more
- LAPACK: dense and banded matrix decomposition and more
  - `svd()`, `La.svd()`, `prcomp()`, `princomp()`, `qr()`, `solve()`, `chol()`, `norm()`, and many more
  - But not `lm()`, careful with `qr(x, LAPACK = TRUE)`: column pivoting
- Implementations: OpenBLAS, Intel MKL, Nvidia nvBLAS, Apple vecLib, AMD BLIS, Arm Performance Libraries
- **FlexiBLAS**: A BLAS and LAPACK wrapper library with runtime exchangeable backends

NETLIB

Netlib is a collection of mathematical software, papers, and databases.

- netlib.org maintained at ORNL and UTK
- Contains Reference BLAS and LAPACK (v 3.10.0, June 28, 2021) Fortran libraries
    - Default BLAS and LAPACK libraries in R
- IT4I FlexiBLAS: "NETLIB" backend

Jack J Dongarra and Eric Grosse. 1987. Distribution of mathematical software via electronic mail. Commun. ACM 30, 5 (May 1987), 403–407. DOI

OpenBLAS

OpenBLAS is an optimized BLAS library based on GotoBLAS2 (2010, Kazushige Goto).

- openblas.net
- Optimizes algorithm to chip architecture details such as cache size
- IT4I FlexiBLAS: "OPENBLAS" backend

Wang Qian, Zhang Xianyi, Zhang Yunquan, Qing Yi, AUGEM: Automatically Generate High Performance Dense Linear Algebra Kernels on x86 CPUs, In the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13), Denver CO, November 2013.

FlexiBLAS

`flexiblas_setup.r`

```
library(flexiblas)
flexiblas_avail()
flexiblas_version()
flexiblas_current_backend()
flexiblas_list()
flexiblas_list_loaded()

setthreads = function(thr, label = "") {
  cat(label, "Setting", thr, "threads\n")
  flexiblas_set_num_threads(thr)
}
setback = function(backend, label = "") {
  cat(label, "Setting", backend, "backend\n")
  flexiblas_switch(flexiblas_load_backend(backend))
}
```

https://github.com/Enchufa2/r-flexiblas
https://cran.r-project.org/package=flexiblas

Benchmarks with FlexiBLAS

`flxiblas_bench.r`

```r
source("flexiblas_setup.r")

x = matrix(rnorm(1e7), nrow = 1e4, ncol = 1e3)
memuse::howbig(1e4, 1e3)
beta = rep(1, ncol(x))
err = rnorm(nrow(x))
y = x %*% beta + err
data = as.data.frame(cbind(y, x))
names(data) = c("y", paste0("x", 1:ncol(x)))

elo = 0
ehi = 7

# lm ------------------------------------
setback("NETLIB", "lm")
system.time((lm(y ~ ., data)))

setback("OPENBLAS", "lm")
for(i in elo:ehi) {
  setthreads(2^i, "lm")
  print(system.time((lm(y ~ ., data))))
}

# qr ------------------------------------
```

Benchmarks with FlexiBLAS

`flxiblas_pbs.sh`

```
#!/bin/bash
#PBS -N fx
#PBS -l select=1:ncpus=128,walltime=00:50:00
#PBS -q qexp
#PBS -e fx.e
#PBS -o fx.o

cd ~/KPMS-IT4I-EX/code
pwd

module load R
echo "loaded R"

time Rscript flexiblas_bench.r
```

Benchmarks with FlexiBLAS