Parallel R

David Walling

Data Group

Texas Advanced Computing Center

walling@tacc.utexas.edu

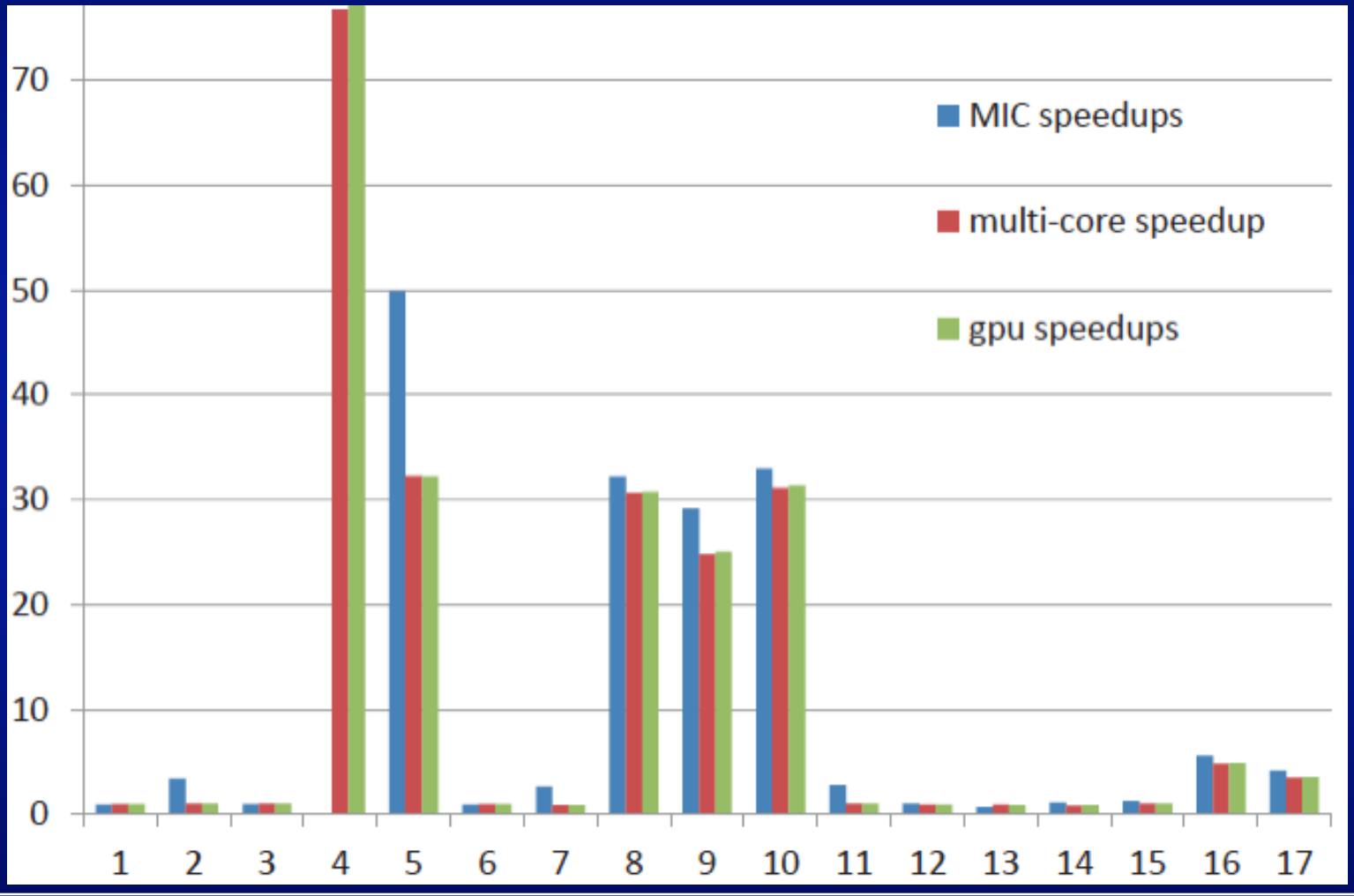
What to do if the computation is too big for a single desktop

- A common user question:
  - I have an existing R solution for my research work. But the data is growing too big. Now my R program runs days to finish or simply runs out of memory.
- 3 strategies
  - Move to more powerful hardware
  - Advanced libraries
  - Implement code using parallel packages

Intel MKL

- Rstats is compiled against Intel Math-Kernel-Library (MKL)
- Optimized calculations for:
  - –Linear algebra
  - –Fast Fourier Transforms (FFT)
  - –Neural Networks
- Many 3rd party packages wrap C++/Fortran code that can utilize this
- Example: Linear Regression
  - –$B = (X^TX)X^TY$

R-2.5 benchmark performance with automatic hardware acceleration



TACC

Snow/Rmpi

Developed based on Rmpi package, but also supports socket connections.

Simplify the process to initialize parallel process over cluster.

While sockets are possible, must use a dirty hack to properly set your environment

Recommend using RMPISNOW instead to launch job

Snow/Rmpi

```
wrangler$ cat SimpleSNOW.R
library(Rmpi)
library(snow)

cluster <- getMPIcluster()

# Print the hostname for each cluster member
sayhello <- function()
{
      info <- Sys.info()[c("nodename", "machine")]
      paste("Hello from", info[1], "with CPU type", info[2])
}

names <- clusterCall(cluster, sayhello)
print(unlist(names))

# Stop cluster will also
# call mpi finalize
# no need for mpi.exit
stopCluster(cluster)
```

Snow/Rmpi

```
wrangler$ cat run-SNOW_Rmpi.slurm
#!/bin/bash
#SBATCH -J SNOW_Rmpi-R
#SBATCH -o SNOW_Rmpi.out%j
#SBATCH -p normal
#SBATCH -t 00:30:00
#SBATCH -A TRAINING-HPC
#SBATCH -e SNOW_Rmpi.err%j
#SBATCH -N 2
#SBATCH -n 10

#set environment
module purge
module load TACC
echo "say hello"
ibrun  RMPISNOW < SimpleSnow.R
echo "done"
```

Snow/Rmpi

```
wrangler$ sbatch run-SNOW_Rmpi.slurm

wrangler(246)$ squeue -u train###
             JOBID   PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           5867739      normal SNOW_Rmp  walling CG       0:13      2 c402-[301-302]

wrangler(389)$ cat SNOW_Rmpi.out5867843
> print(unlist(names))
[1] "Hello from c252-136.wrangler.tacc.utexas.edu with CPU type x86_64"
[2] "Hello from c252-136.wrangler.tacc.utexas.edu with CPU type x86_64"
[3] "Hello from c252-136.wrangler.tacc.utexas.edu with CPU type x86_64"
[4] "Hello from c252-136.wrangler.tacc.utexas.edu with CPU type x86_64"
[5] "Hello from c252-137.wrangler.tacc.utexas.edu with CPU type x86_64"
[6] "Hello from c252-137.wrangler.tacc.utexas.edu with CPU type x86_64"
[7] "Hello from c252-137.wrangler.tacc.utexas.edu with CPU type x86_64"
[8] "Hello from c252-137.wrangler.tacc.utexas.edu with CPU type x86_64"
[9] "Hello from c252-137.wrangler.tacc.utexas.edu with CPU type x86_64"
>
……
```

Memory Considerations

R pipelines often memory bound

Ex. 100 parallel tasks, each task takes 2 GB of memory, have 20 cores and and 20 GB memory -> Only utilize 10 cores at a time

Debugging/Profiling

a. Packages: profmem, lineprof, others
b. ssh to compute node, run ‘top’

top

```
top - 08:40:20 up 100 days, 20:06,  2 users,  load average: 9.45, 4.59, 3.42
Tasks: 1196 total,  98 running, 1098 sleeping,   0 stopped,   0 zombie
%Cpu(s): 72.5 us, 15.5 sy,  0.0 ni,  8.3 id,  3.6 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 13161971+total, 12299456+free,  6576920 used,  2048224 buff/cache
KiB Swap:  1048572 total,  1040164 free,     8408 used. 12305089+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
24422 root      20   0       0      0      0 R 100.0  0.0 144095:28 vpthread-0-1
16096 root      20   0       0      0      0 R  93.9  0.0  22402:47 vpthread-1-1
35179 walling   20   0  560420 258652  11772 S  68.0  0.2   0:05.34 R
35188 walling   20   0  560420 248208   1328 R  59.5  0.2   0:01.84 R
35191 walling   20   0  560420 248204   1324 R  58.6  0.2   0:01.81 R
35201 walling   20   0  560420 248204   1324 R  57.9  0.2   0:01.79 R
35187 walling   20   0  560420 248208   1328 R  55.7  0.2   0:01.72 R
35198 walling   20   0  560420 248204   1324 D  54.7  0.2   0:01.69 R
35186 walling   20   0  560420 248208   1328 R  54.4  0.2   0:01.68 R
35208 walling   20   0  560420 248204   1324 R  53.1  0.2   0:01.64 R
35194 walling   20   0  560420 248204   1324 R  52.8  0.2   0:01.63 R
35218 walling   20   0  560420 248208   1328 R  52.8  0.2   0:01.63 R
35199 walling   20   0  560420 248204   1324 R  52.4  0.2   0:01.62 R
35203 walling   20   0  560420 248204   1324 R  52.4  0.2   0:01.62 R
35204 walling   20   0  560420 248204   1324 R  52.4  0.2   0:01.62 R
35185 walling   20   0  560420 248208   1328 R  52.1  0.2   0:01.61 R
35193 walling   20   0  560420 248204   1324 R  51.8  0.2   0:01.60 R
35200 walling   20   0  560420 248204   1324 R  51.8  0.2   0:01.60 R
35215 walling   20   0  560420 248204   1324 R  51.5  0.2   0:01.59 R
35195 walling   20   0  560420 248204   1324 R  51.1  0.2   0:01.58 R
35206 walling   20   0  560420 248204   1324 R  51.1  0.2   0:01.58 R
35211 walling   20   0  560420 248204   1324 R  51.1  0.2   0:01.58 R
35225 walling   20   0  560420 248208   1328 R  50.5  0.2   0:01.56 R
```