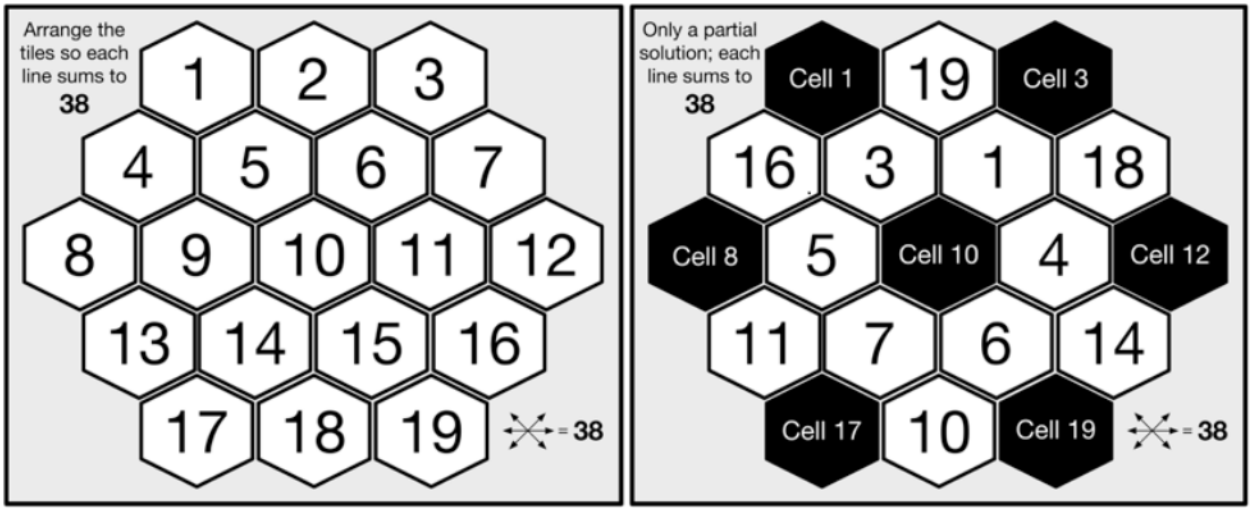
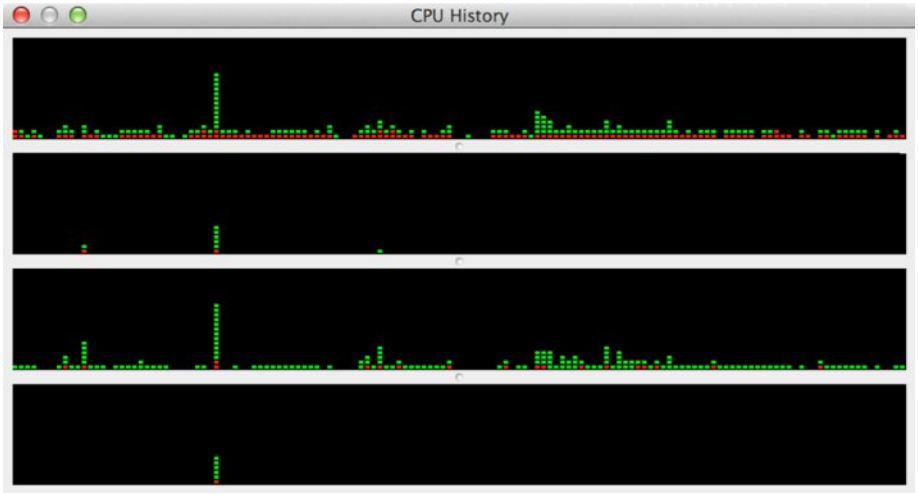
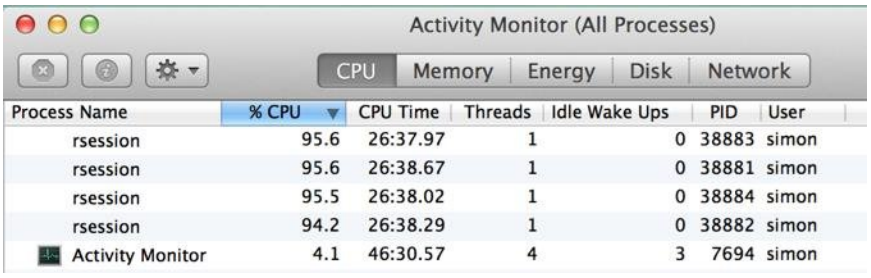
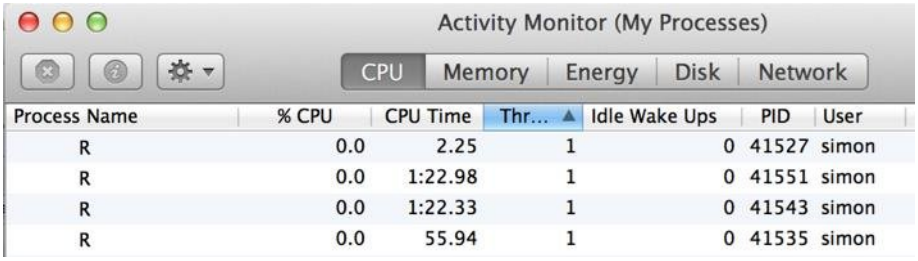
Chapter 1: Simple Parallelism with R



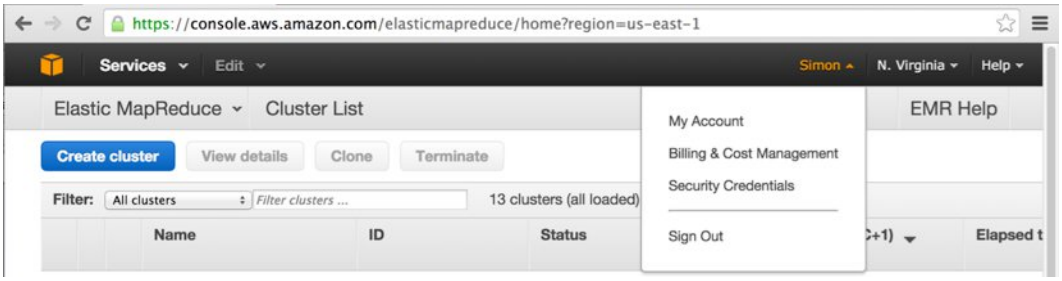


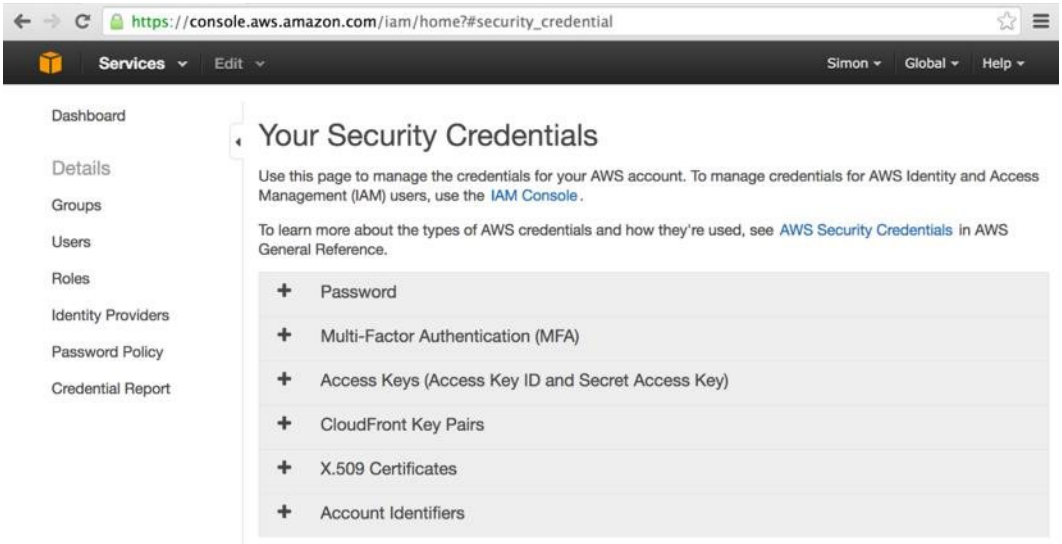




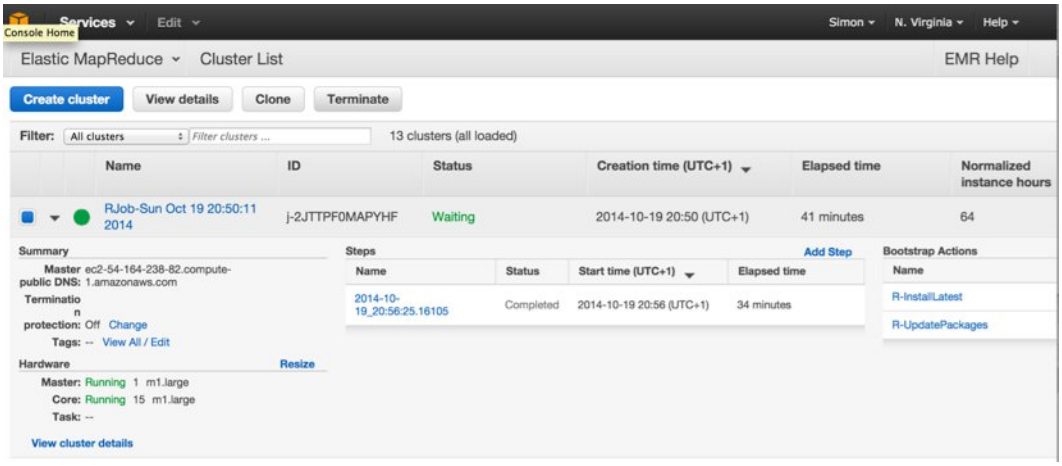
Process Name	% CPU	CPU Time	Thr...	Idle Wake Ups	PID	User
R	0.0	6:34.15	1	0	42720	simon
R	0.0	3:34.99	1	0	42712	simon
R	0.0	8.36	1	0	42704	simon
R	0.0	7:26.65	1	0	42728	simon

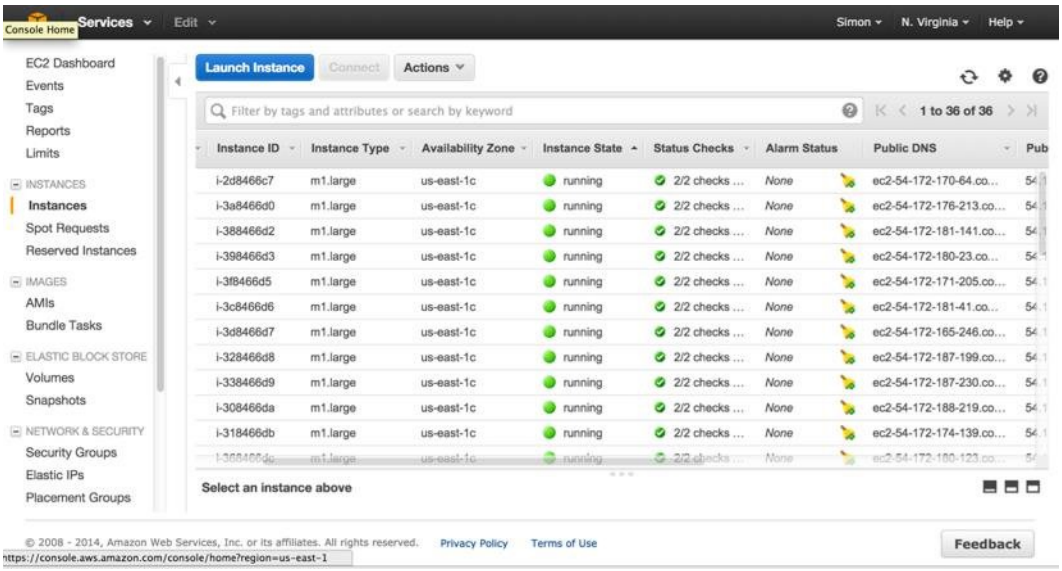
Process Name	% CPU	CPU Time	Thr... ▲	Idle Wake Ups	PID	User
R	0.0	6:51.47	1	0	43092	simon
R	0.0	6:14.22	1	0	43084	simon
R	0.0	6:12.69	1	0	43076	simon
R	0.0	5:14.08	1	0	43100	simon

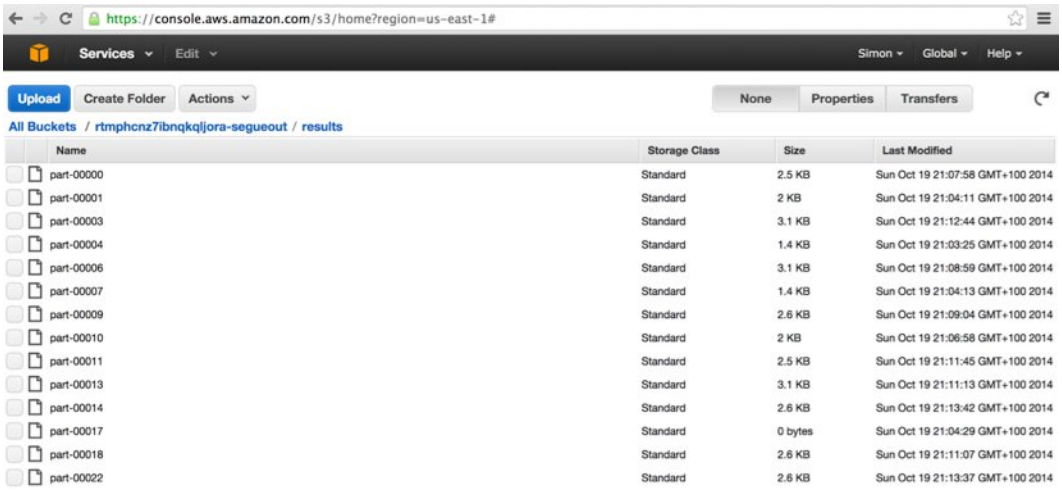


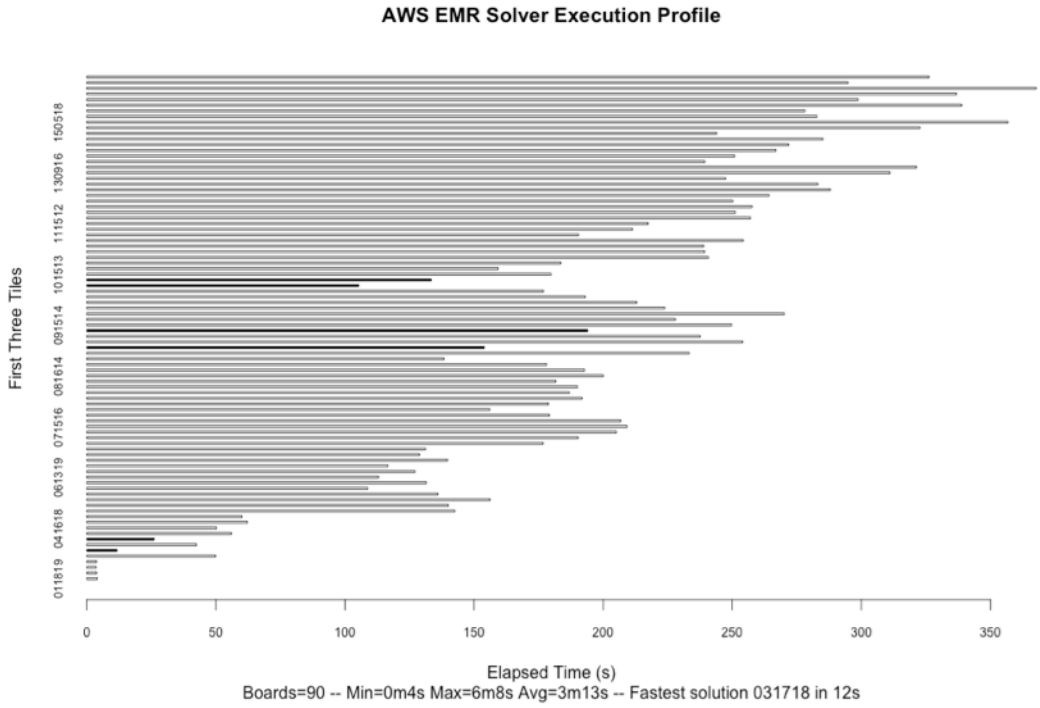



Create Access Key
Your access key (access key ID and secret access key) has been created successfully.
Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.
To help protect your security, store your secret access key securely and do not share it.
Show Access Key
Download Key File
Close

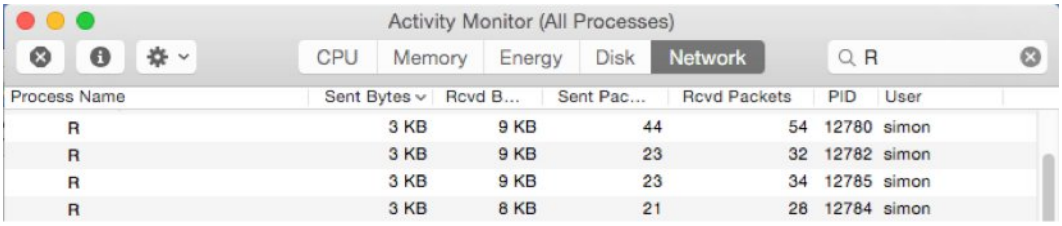

Services
Edit
Simon
N. Virginia
Help
Console Home
Elastic MapReduce
Cluster List
EMR Help
Create cluster
View details
Clone
Terminate
Filter:
All clusters
Filter clusters ...
13 clusters (all loaded)
Name
ID
Status
Creation time (UTC+1)
Elapsed time
Normalized instance hours
RJob-Sun Oct 19 20:50:11 2014
j-2JTTPF0MAPYHF
Waiting
2014-10-19 20:50 (UTC+1)
41 minutes
64
Summary
Master public DNS: ec2-54-164-238-82.compute-1.amazonaws.com
Termination protection: Off Change
Tags: -- View All / Edit
Hardware
Resize
Master: Running 1 m1.large
Core: Running 15 m1.large
Task: --
View cluster details
Steps
Add Step
Name
Status
Start time (UTC+1)
Elapsed time
2014-10-19_20:56:25.16105
Completed
2014-10-19 20:56 (UTC+1)
34 minutes
Bootstrap Actions
Name
R-InstallLatest
R-UpdatePackages

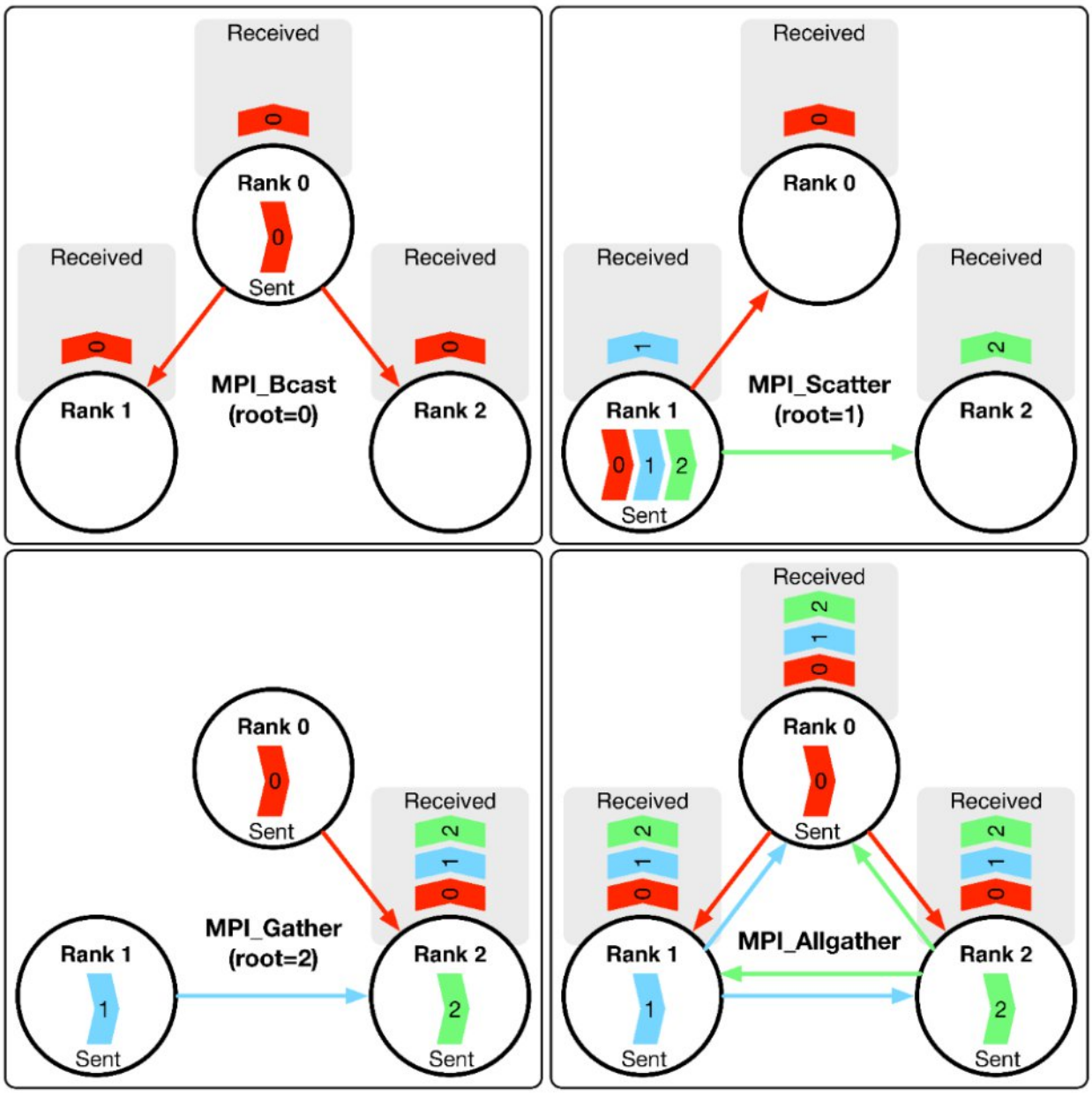

Console Home
Services
Edit
Simon
N. Virginia
Help
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 36 of 36
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS
i-2d8466c7 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-170-64.co...
i-3a8466d0 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-176-213.co...
i-388466d2 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-181-141.co...
i-398466d3 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-180-23.co...
i-3f8466d5 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-171-205.co...
i-3c8466d6 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-181-41.co...
i-3d8466d7 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-165-246.co...
i-328466d8 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-187-199.co...
i-338466d9 m1.large us-east-1c running 2/2 checks ... None ec2-54-172-187-230.co...
i-308466da m1.large us-east-1c running 2/2 checks ... None ec2-54-172-188-219.co...
i-318466db m1.large us-east-1c running 2/2 checks ... None ec2-54-172-174-139.co...
Select an instance above
© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Feedback
https://console.aws.amazon.com/console/home?region=us-east-1


https://console.aws.amazon.com/s3/home?region=us-east-1#
Services
Edit
Simon
Global
Help
Upload
Create Folder
Actions
None
Properties
Transfers
All Buckets / rtmphcnz7ibnqkqljora-segueout / results
Name
Storage Class
Size
Last Modified
part-00000 Standard 2.5 KB Sun Oct 19 21:07:58 GMT+100 2014
part-00001 Standard 2 KB Sun Oct 19 21:04:11 GMT+100 2014
part-00003 Standard 3.1 KB Sun Oct 19 21:12:44 GMT+100 2014
part-00004 Standard 1.4 KB Sun Oct 19 21:03:25 GMT+100 2014
part-00006 Standard 3.1 KB Sun Oct 19 21:08:59 GMT+100 2014
part-00007 Standard 1.4 KB Sun Oct 19 21:04:13 GMT+100 2014
part-00009 Standard 2.6 KB Sun Oct 19 21:09:04 GMT+100 2014
part-00010 Standard 2 KB Sun Oct 19 21:06:58 GMT+100 2014
part-00011 Standard 2.5 KB Sun Oct 19 21:11:45 GMT+100 2014
part-00013 Standard 3.1 KB Sun Oct 19 21:11:13 GMT+100 2014
part-00014 Standard 2.6 KB Sun Oct 19 21:13:42 GMT+100 2014
part-00017 Standard 0 bytes Sun Oct 19 21:04:29 GMT+100 2014
part-00018 Standard 2.6 KB Sun Oct 19 21:11:07 GMT+100 2014
part-00022 Standard 2.6 KB Sun Oct 19 21:13:37 GMT+100 2014

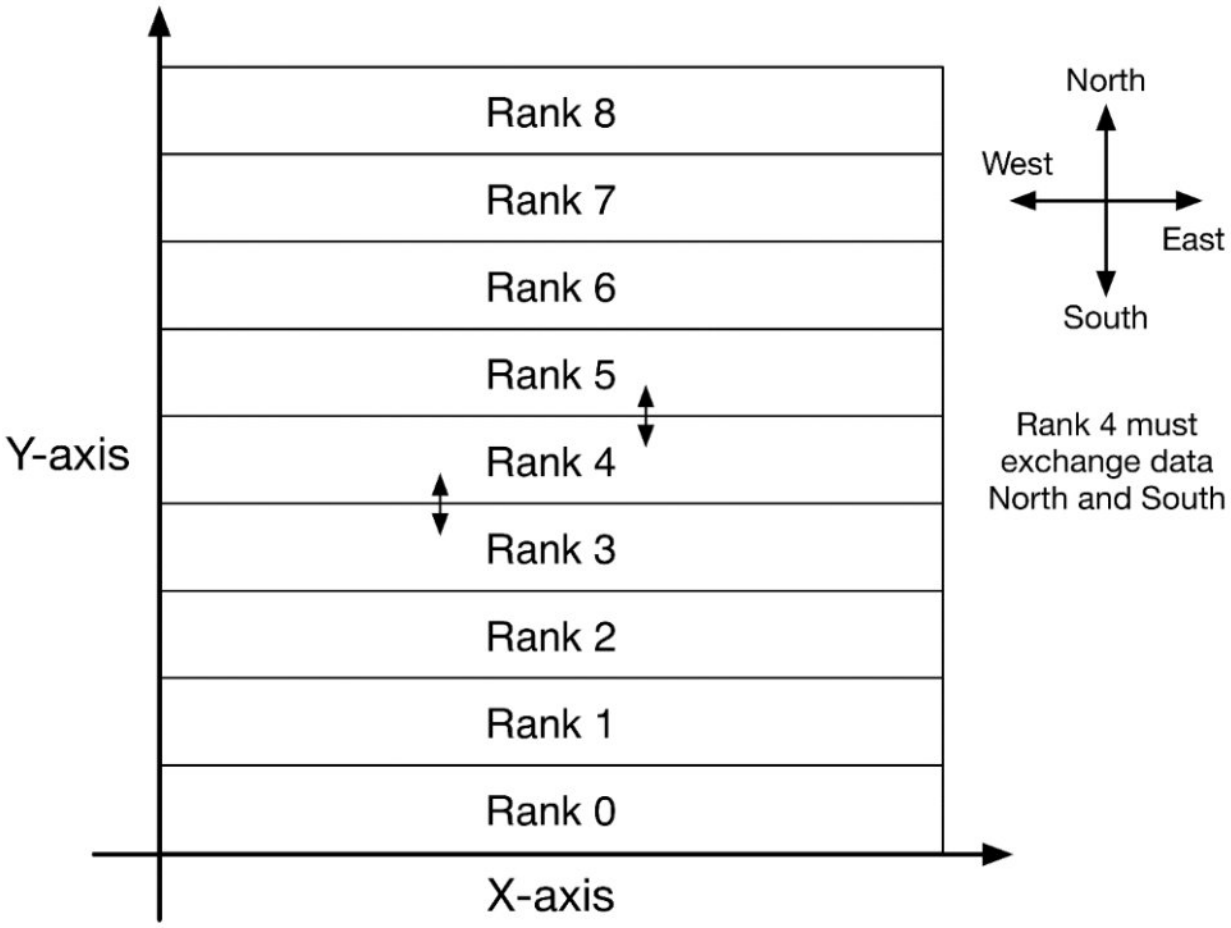

AWS EMR Solver Execution Profile
First Three Tiles
011819
041618
061319
071516
081614
091514
101513
111512
130916
150518
0
50
100
150
200
250
300
350
Elapsed Time (s)
Boards=90 -- Min=0m4s Max=6m8s Avg=3m13s -- Fastest solution 031718 in 12s

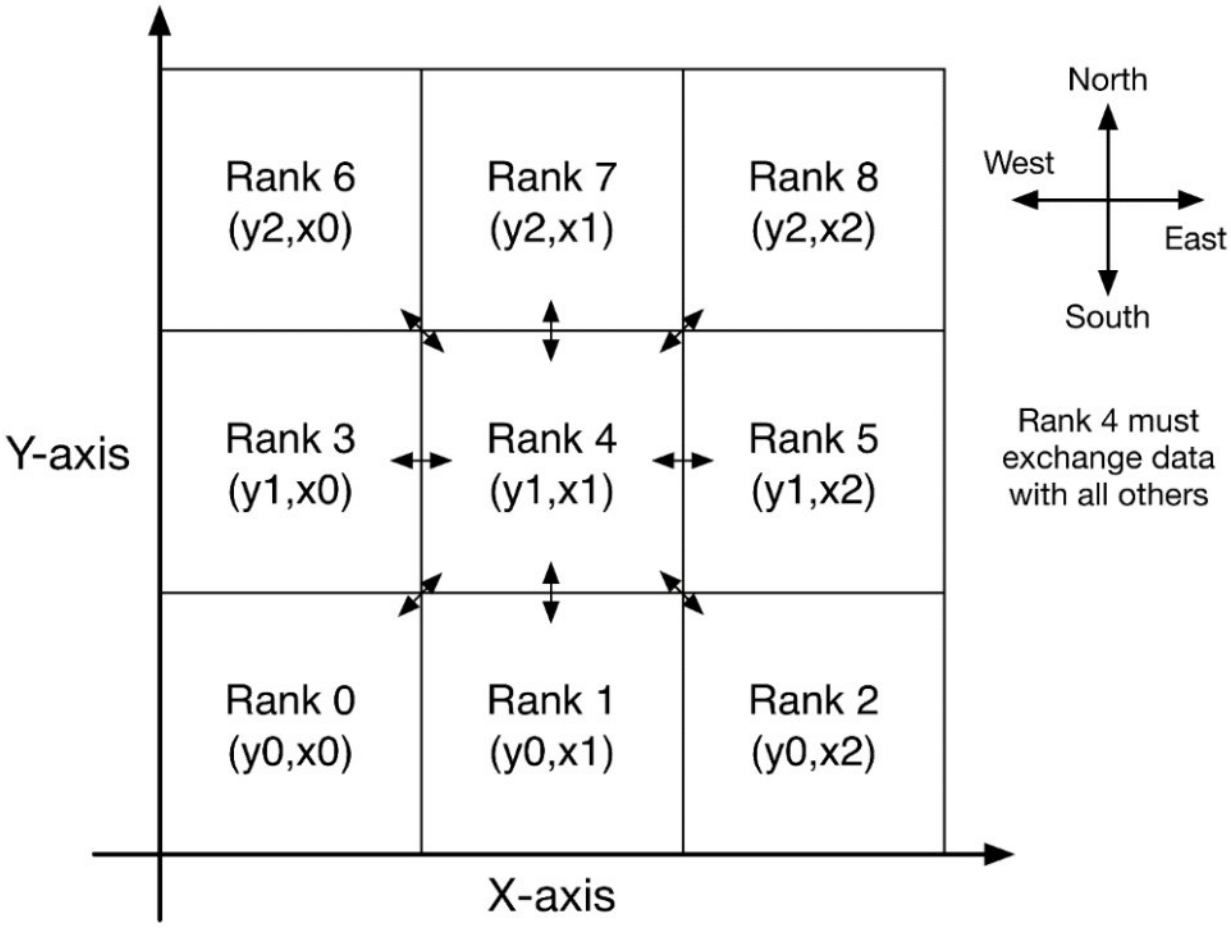
Chapter 2: Introduction to Message Passing

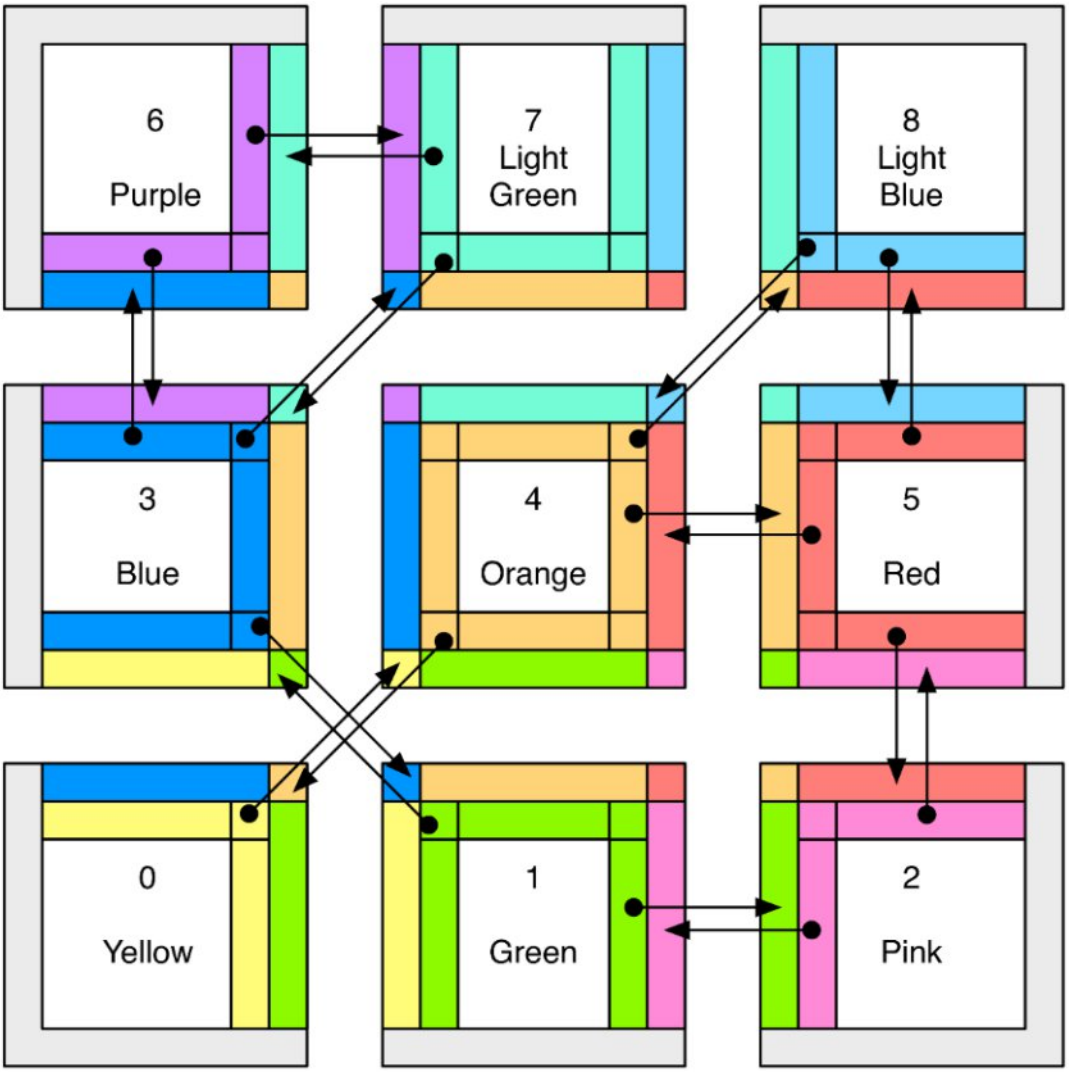


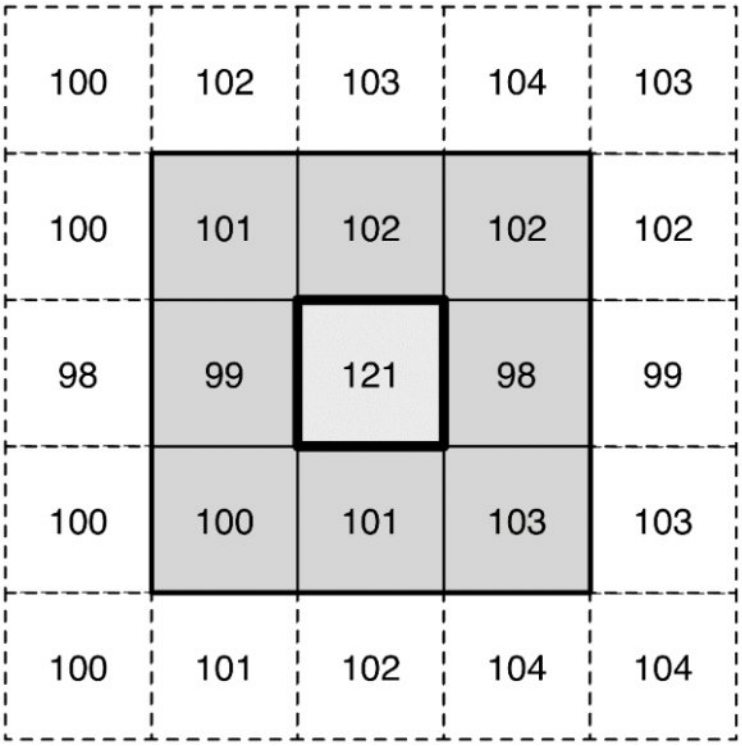


Chapter 3: Advanced Message Passing




Rank 6
(y2,x0)
Rank 7
(y2,x1)
Rank 8
(y2,x2)
Rank 3
(y1,x0)
Rank 4
(y1,x1)
Rank 5
(y1,x2)
Rank 0
(y0,x0)
Rank 1
(y0,x1)
Rank 2
(y0,x2)
Y-axis
X-axis
North
West
East
South
Rank 4 must
exchange data
with all others


6
Purple
7
Light
Green
8
Light
Blue
3
Blue
4
Orange
5
Red
0
Yellow
1
Green
2
Pink

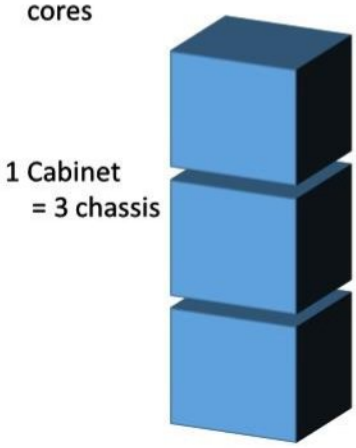


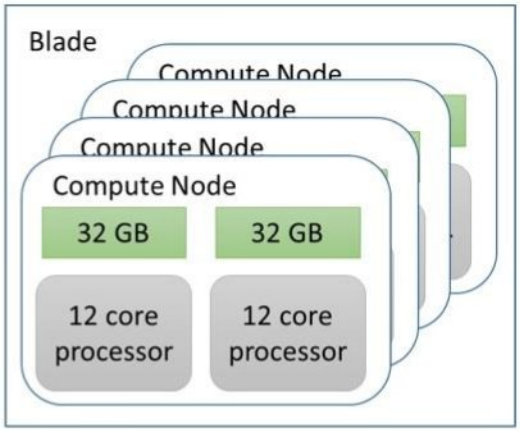
The median replacement value for pixel 121 is 101

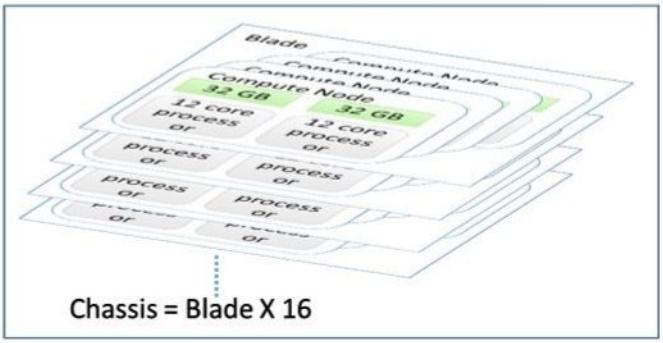
98, 99, 100, 101,
101,
102,102,103,121

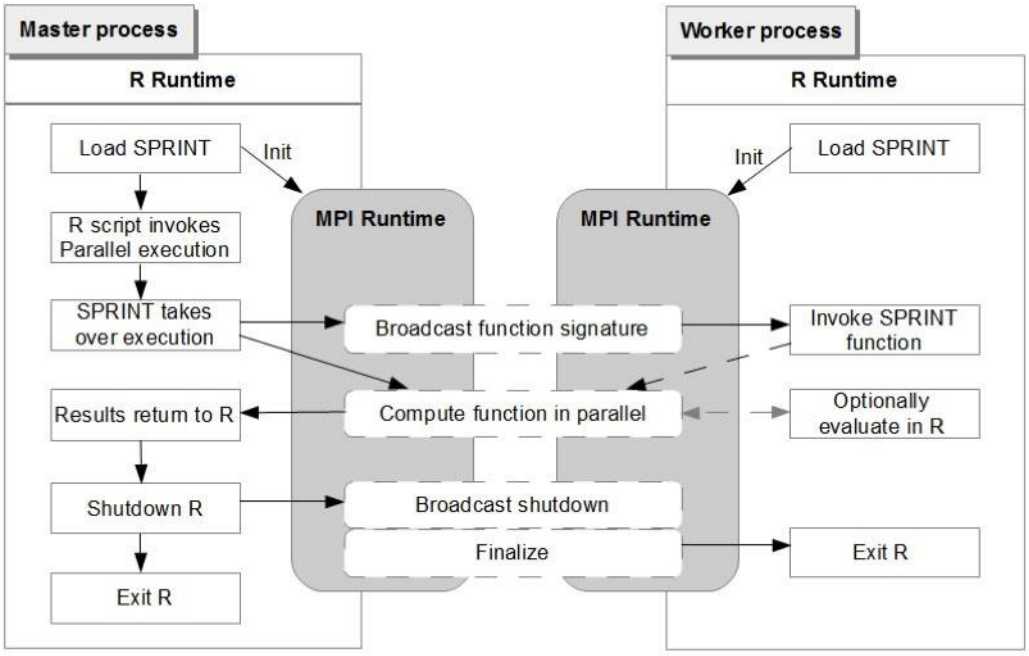
Chapter 4: Developing SPRINT, an MPI-Based R Package for Supercomputers

ARCHER has 4920 nodes in 26 cabinets giving a total of 118,080 cores

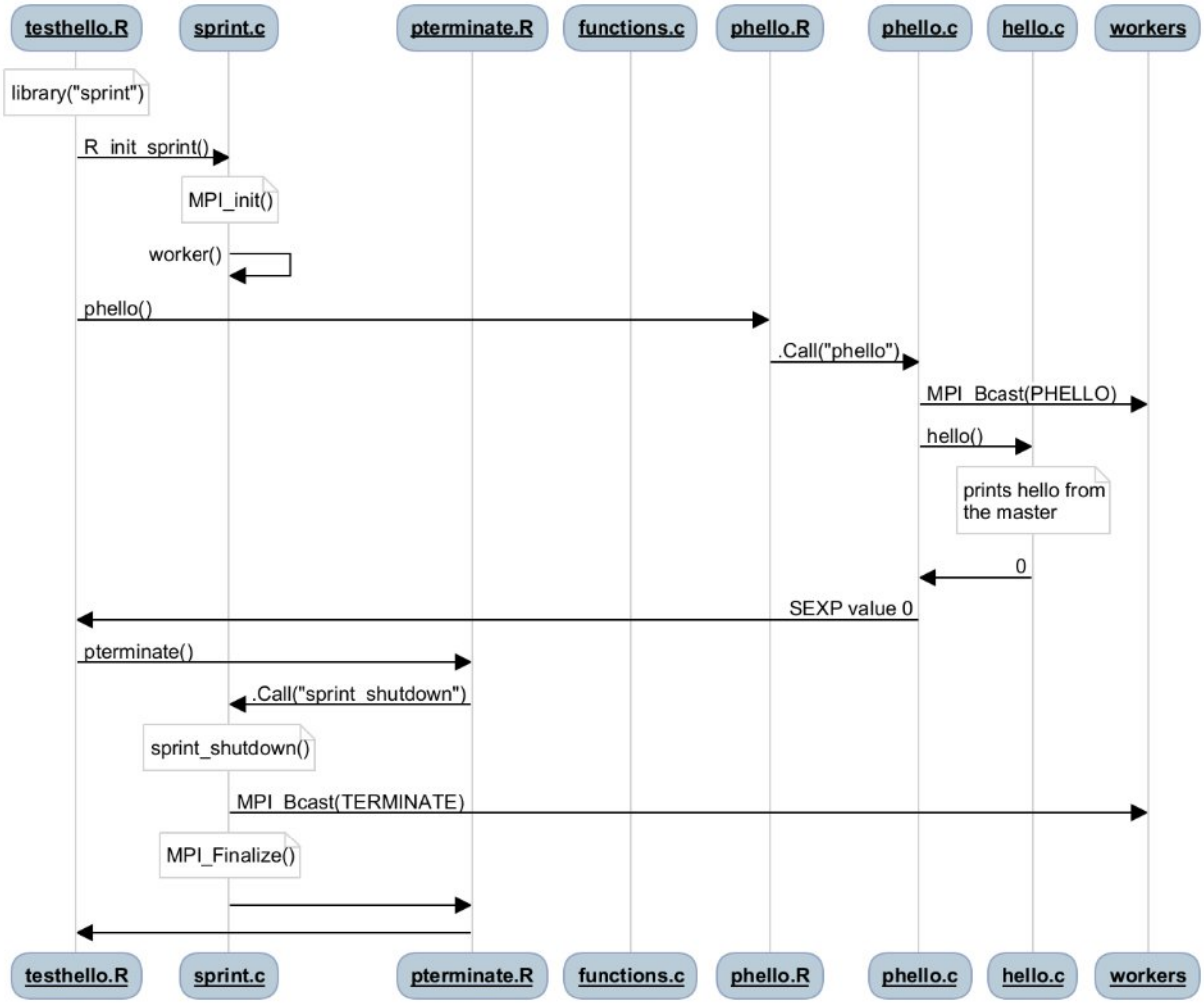




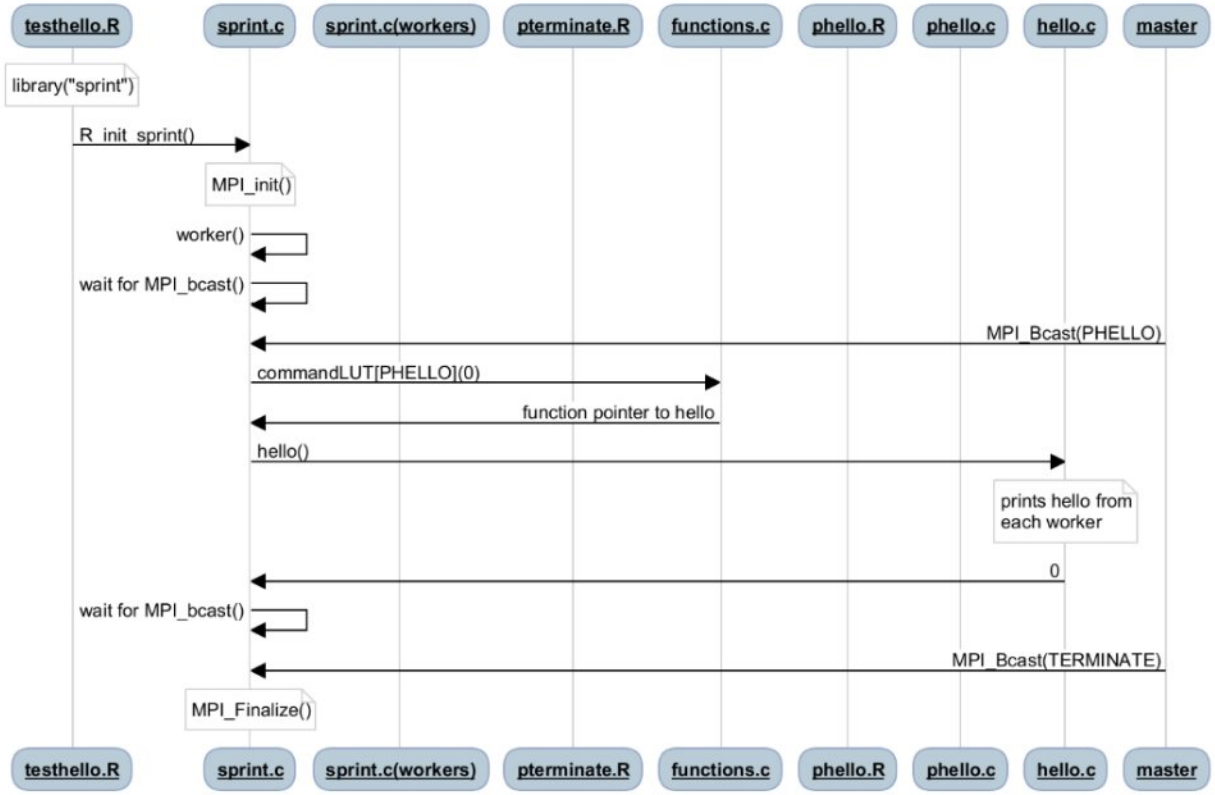


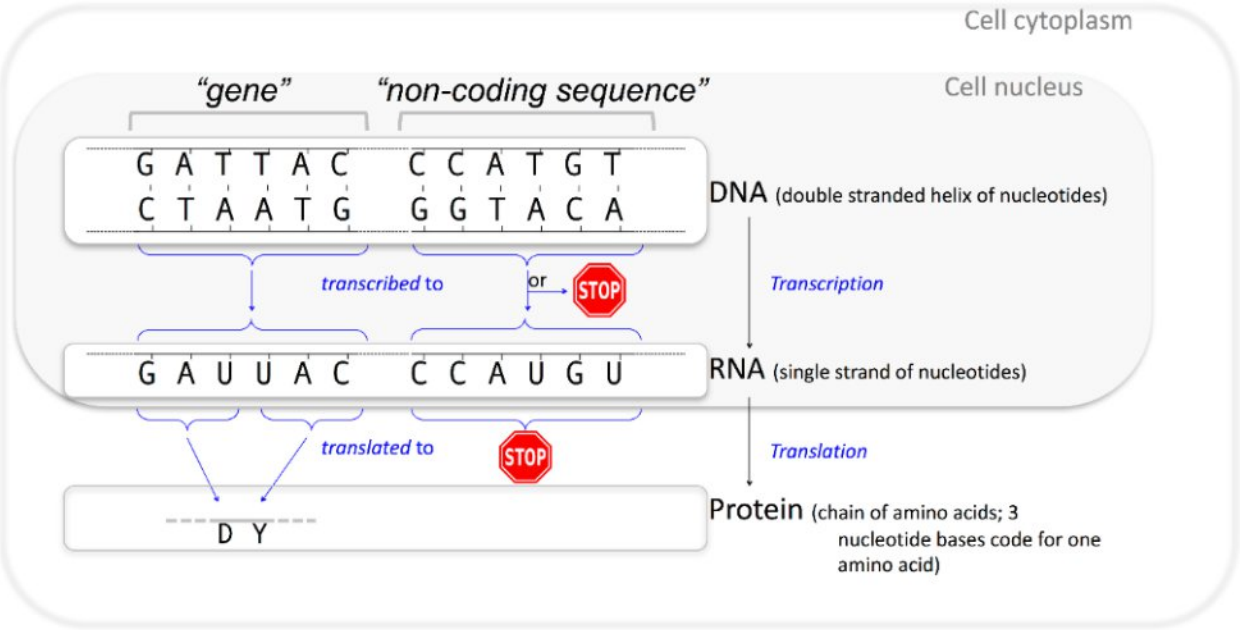

Master process
R Runtime
Load SPRINT
Init
R script invokes Parallel execution
SPRINT takes over execution
Results return to R
Shutdown R
Exit R
MPI Runtime
Broadcast function signature
Compute function in parallel
Broadcast shutdown
Finalize
MPI Runtime
Worker process
R Runtime
Init
Load SPRINT
Invoke SPRINT function
Optionally evaluate in R
Exit R

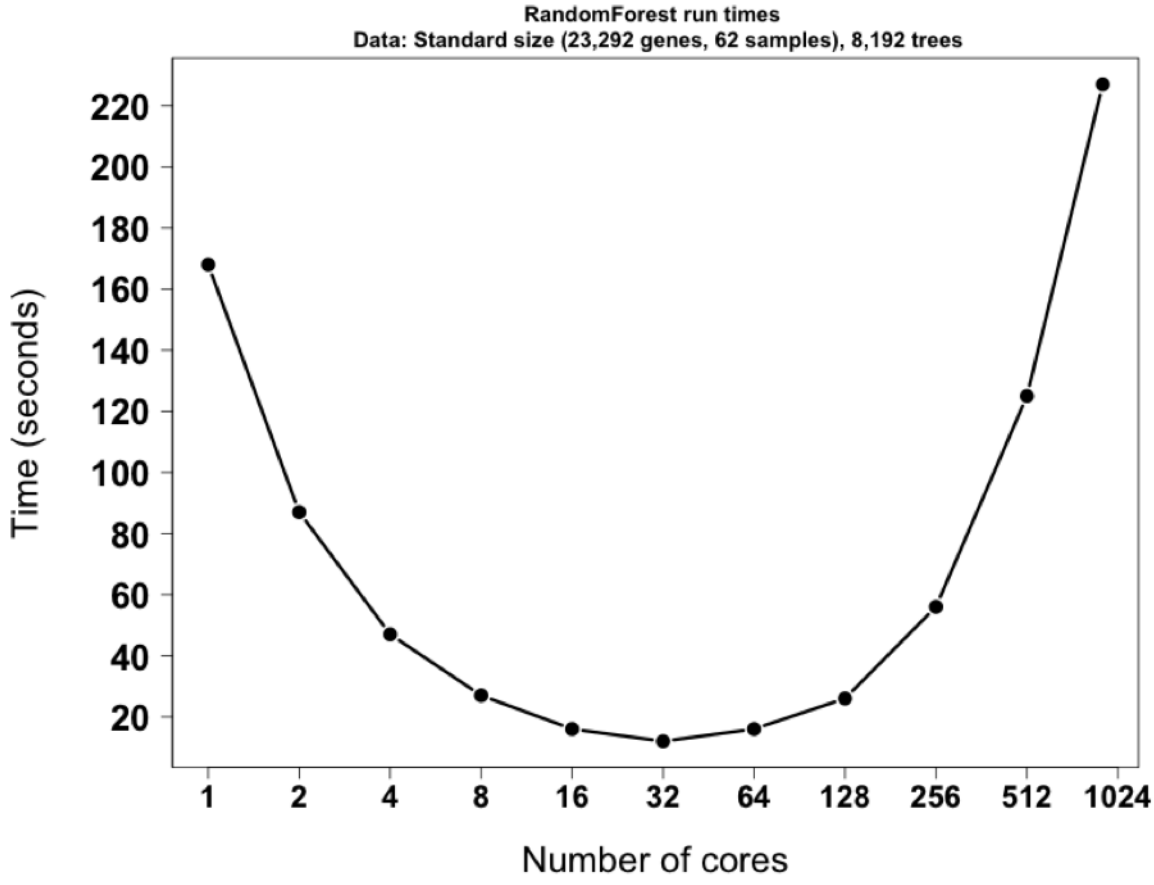
SPRINT hello world master

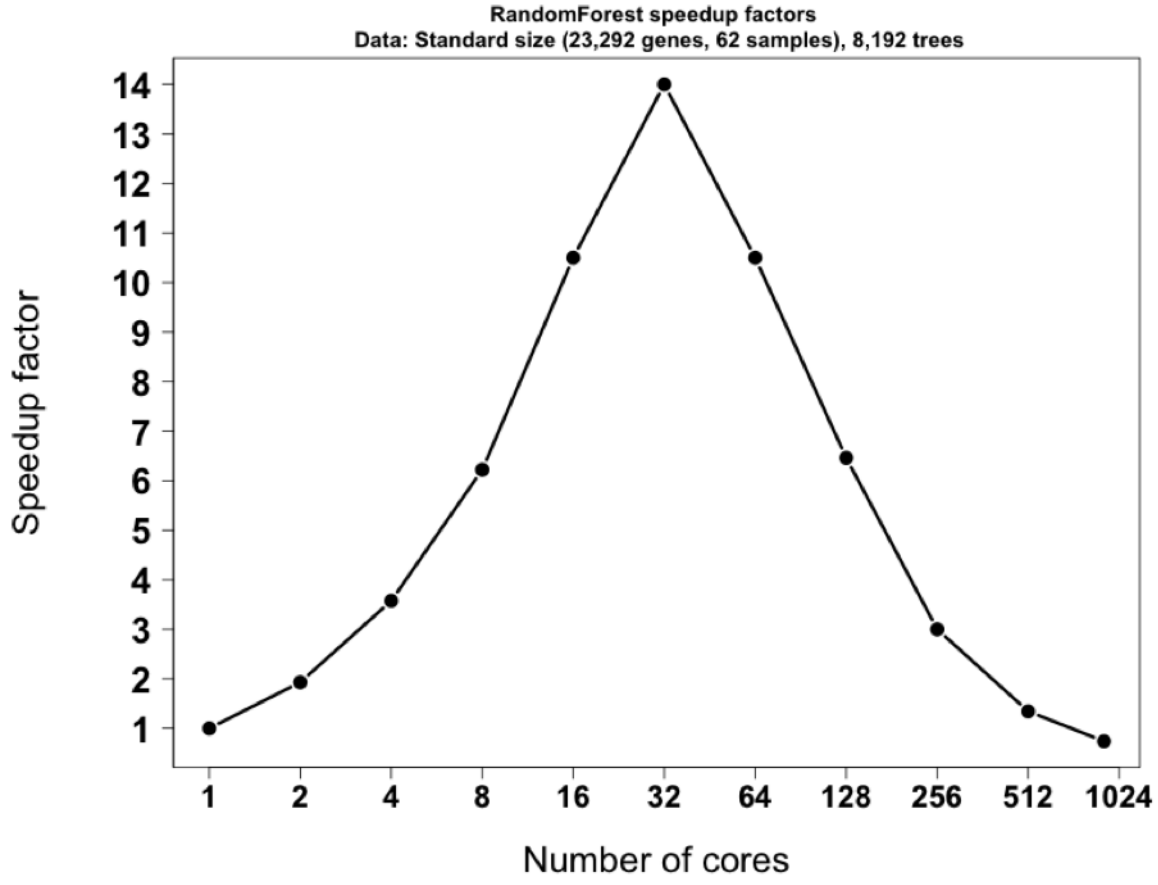


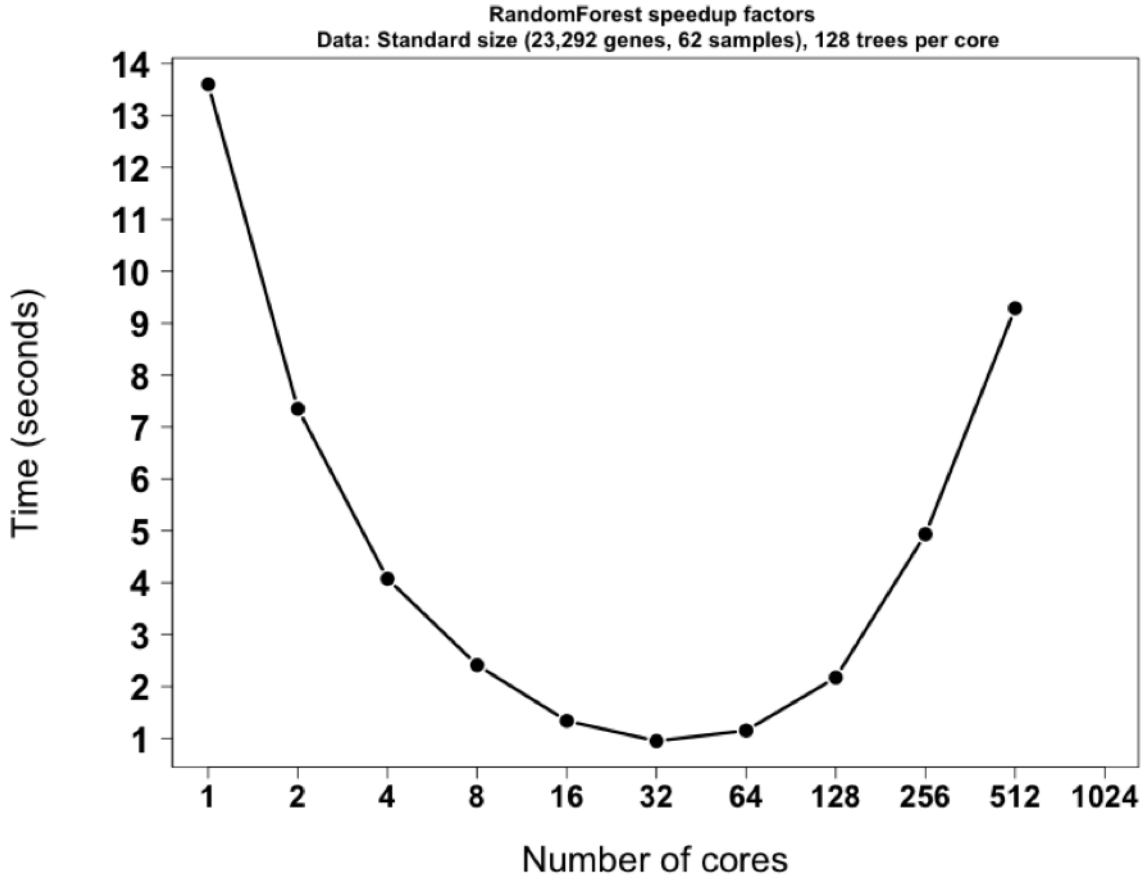
SPRINT hello world worker

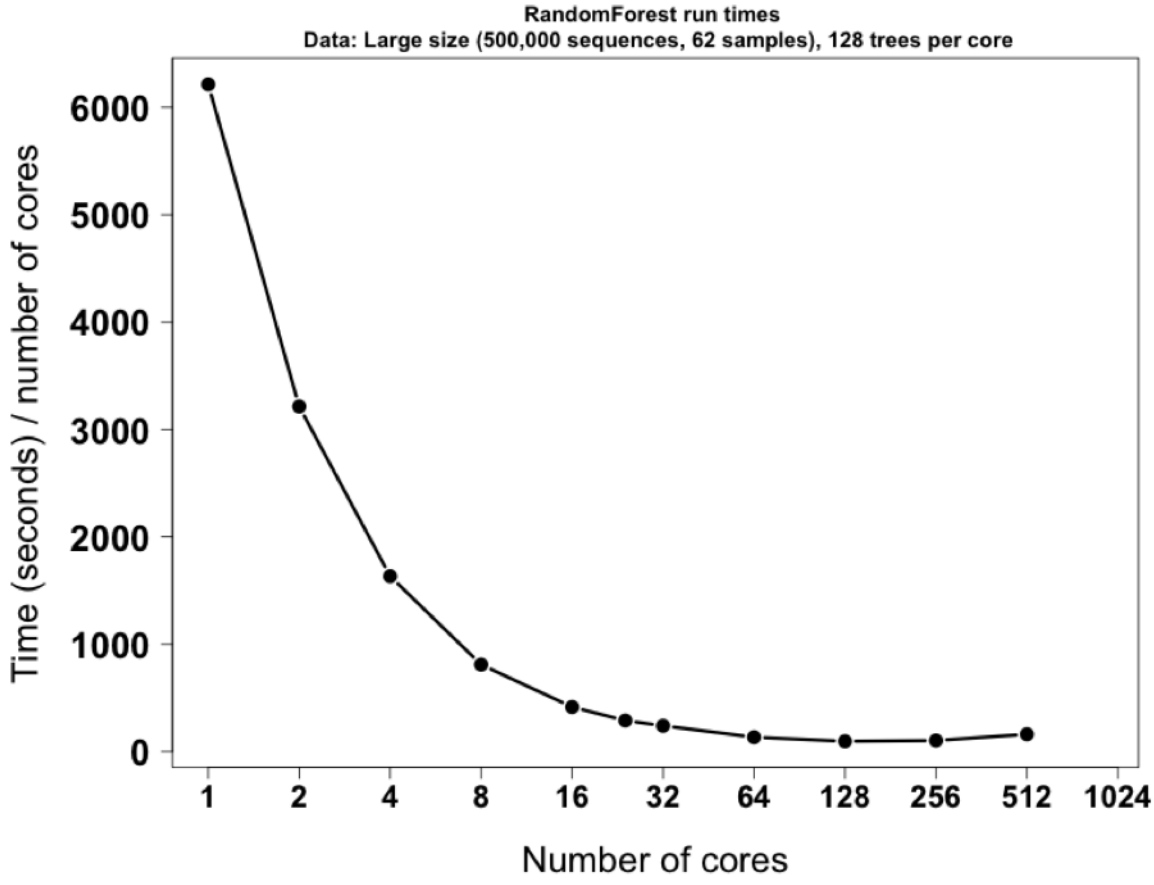


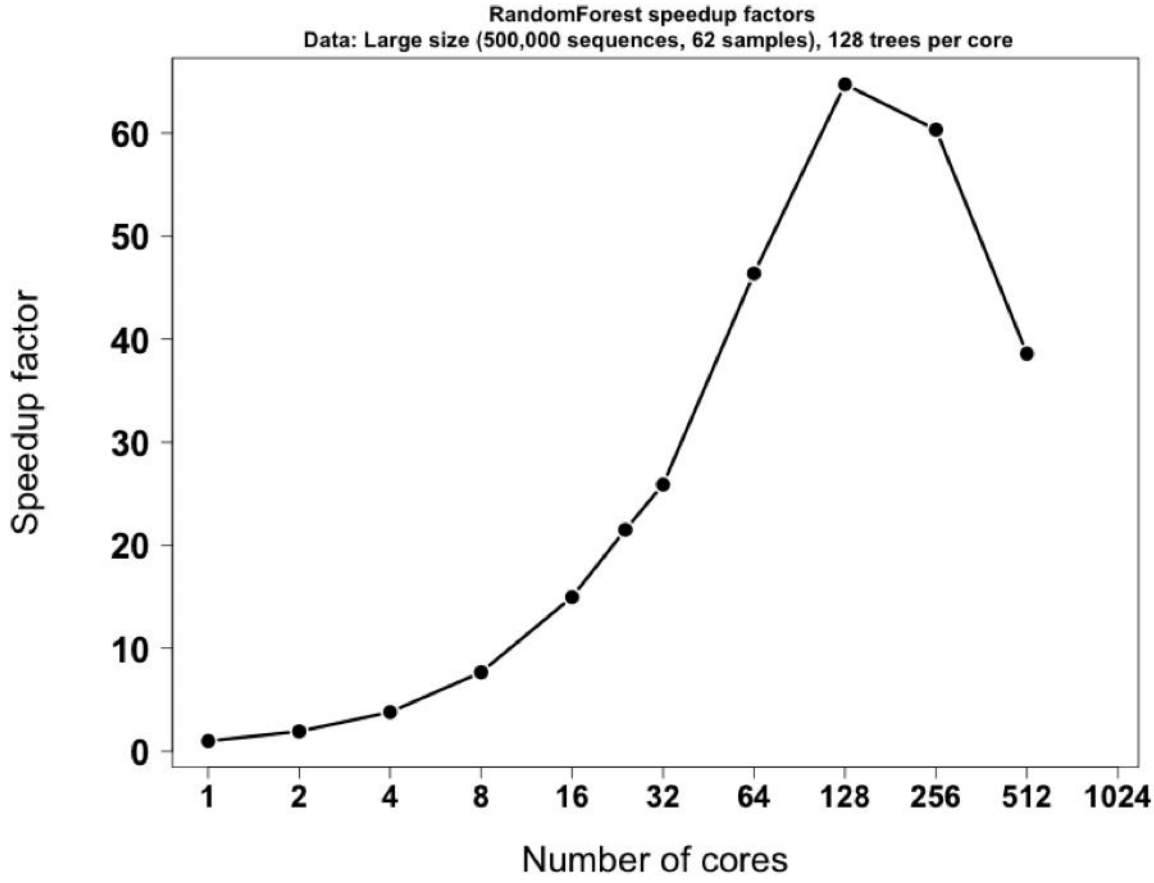

Cell cytoplasm
"gene"
"non-coding sequence"
Cell nucleus
G A T T A C C C A T G T
C T A A T G G G T A C A
DNA (double stranded helix of nucleotides)
transcribed to
or
STOP
Transcription
G A U U A C C C A U G U
RNA (single strand of nucleotides)
translated to
STOP
Translation
D Y
Protein (chain of amino acids; 3 nucleotide bases code for one amino acid)

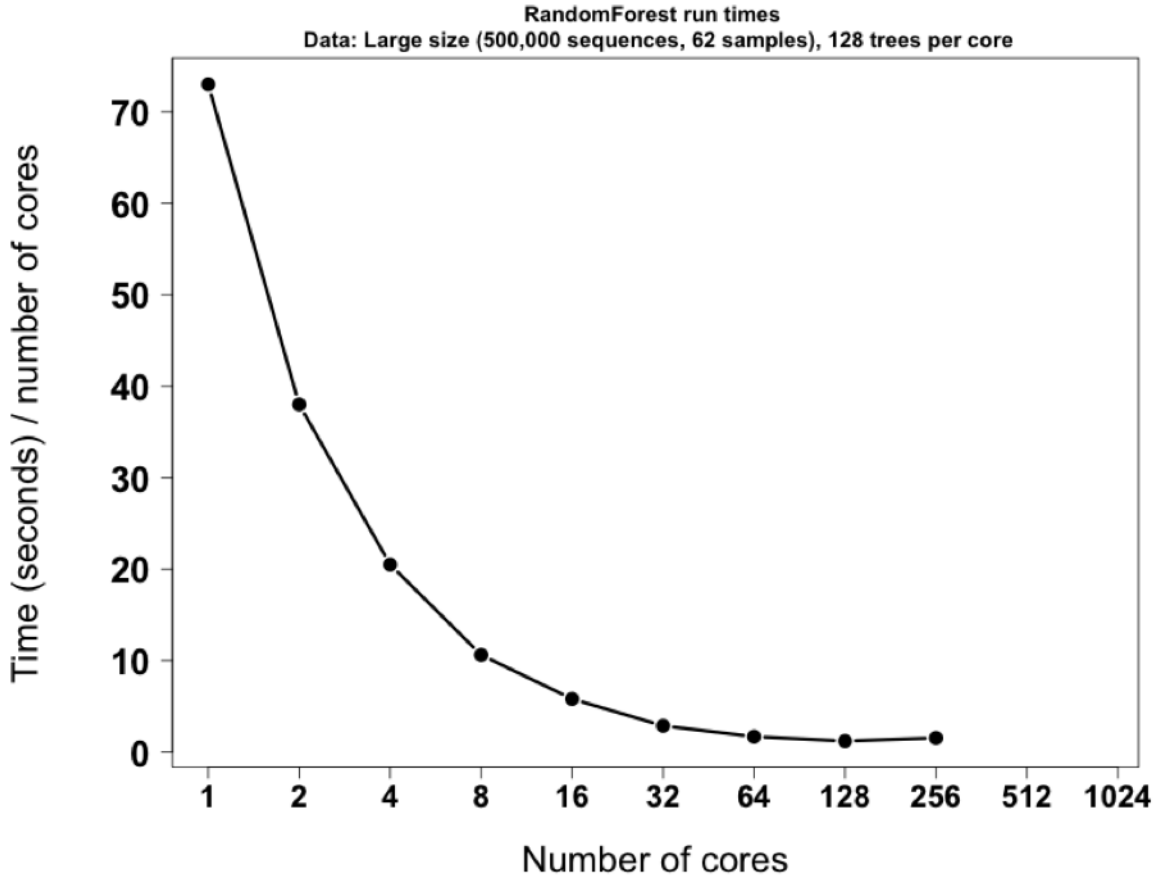

RandomForest run times
Data: Standard size (23,292 genes, 62 samples), 8,192 trees
Time (seconds)
220
200
180
160
140
120
100
80
60
40
20
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

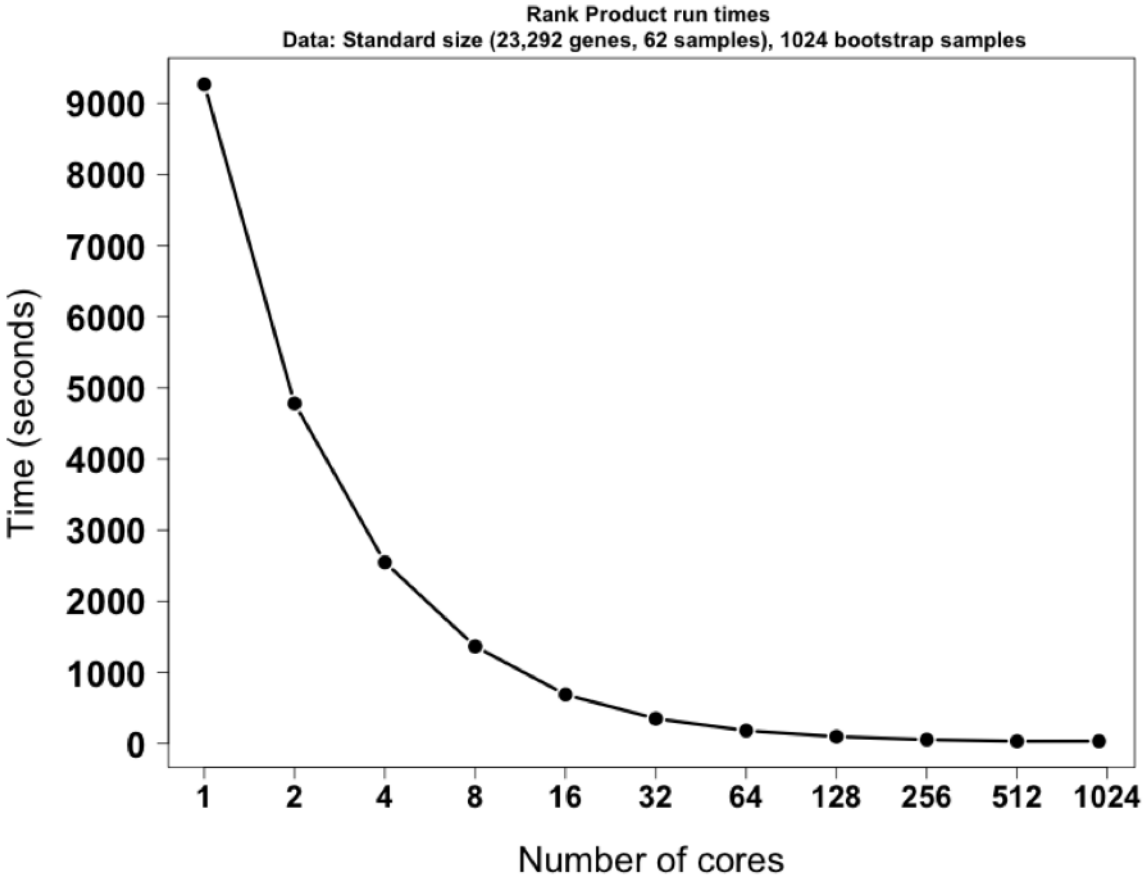

RandomForest speedup factors
Data: Standard size (23,292 genes, 62 samples), 8,192 trees
Speedup factor
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

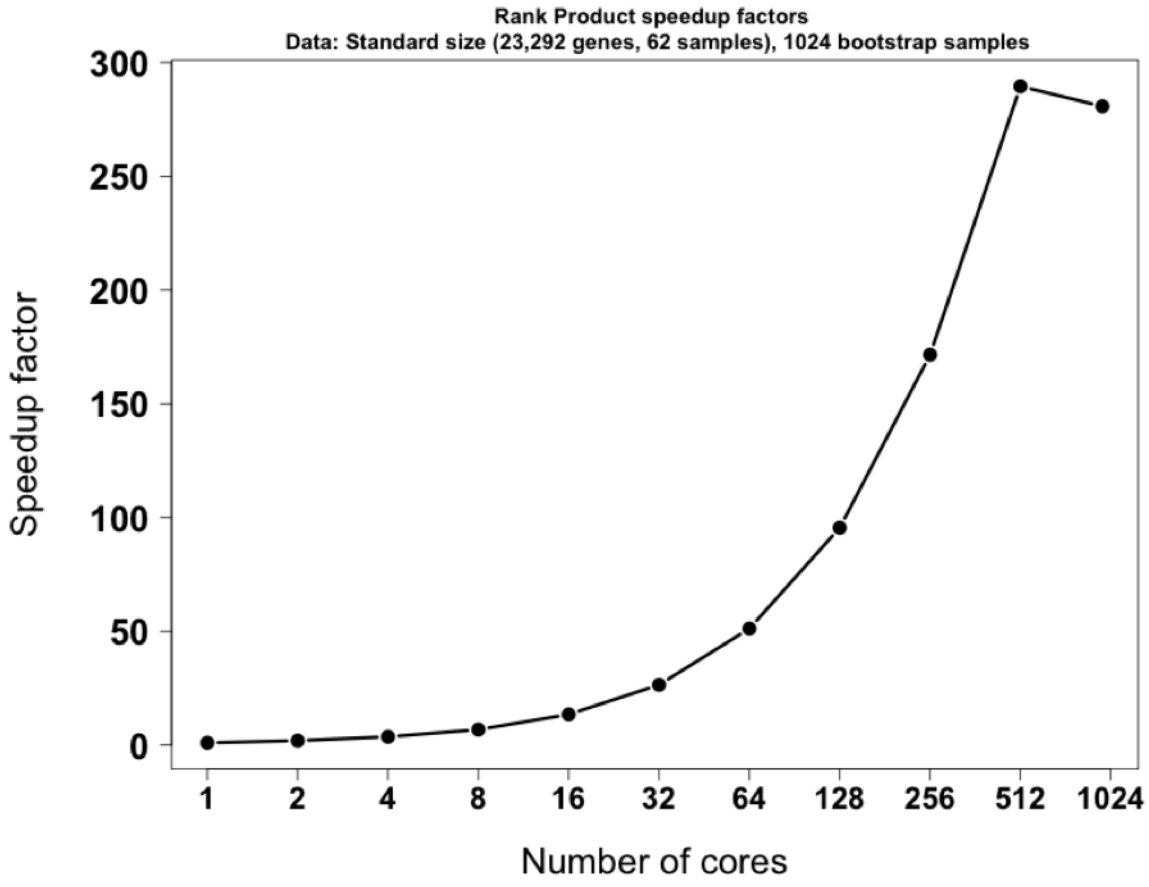

RandomForest speedup factors
Data: Standard size (23,292 genes, 62 samples), 128 trees per core
Time (seconds)
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

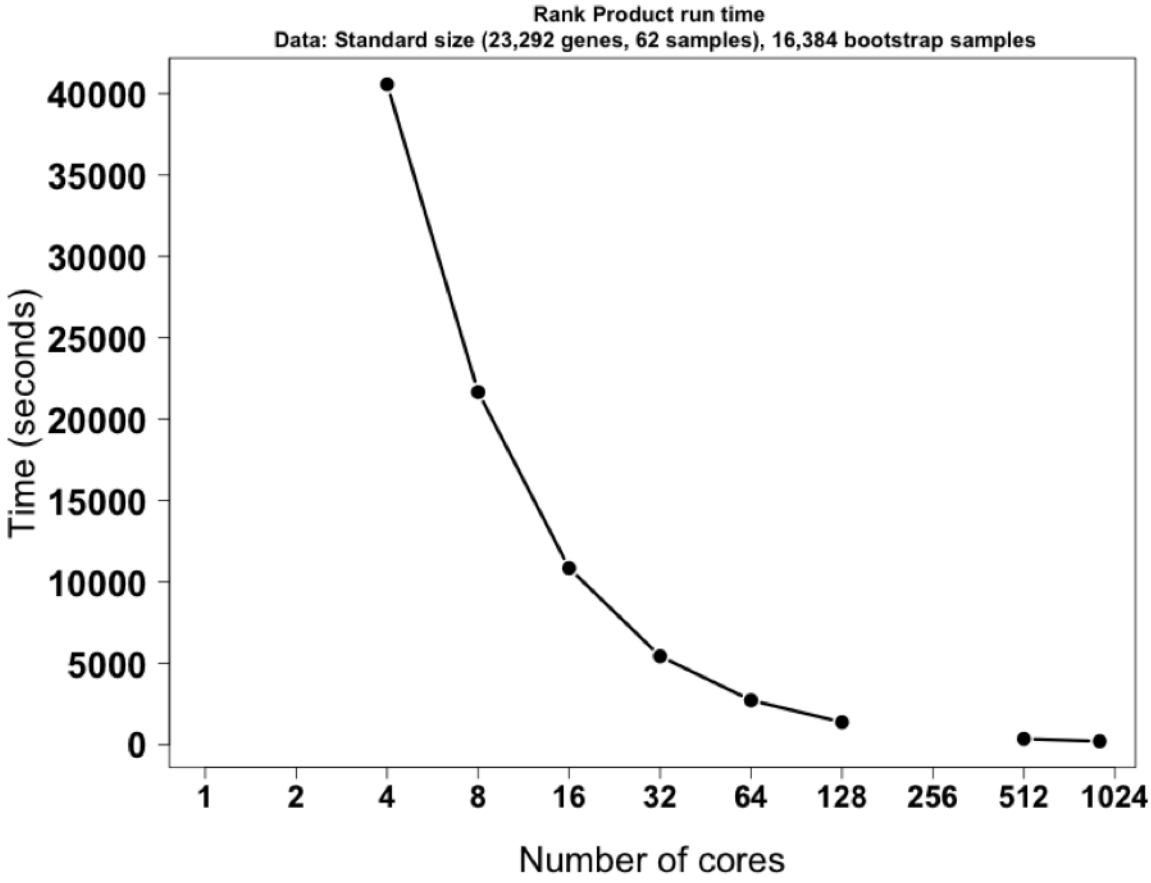

RandomForest run times
Data: Large size (500,000 sequences, 62 samples), 128 trees per core
Time (seconds) / number of cores
6000
5000
4000
3000
2000
1000
0
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

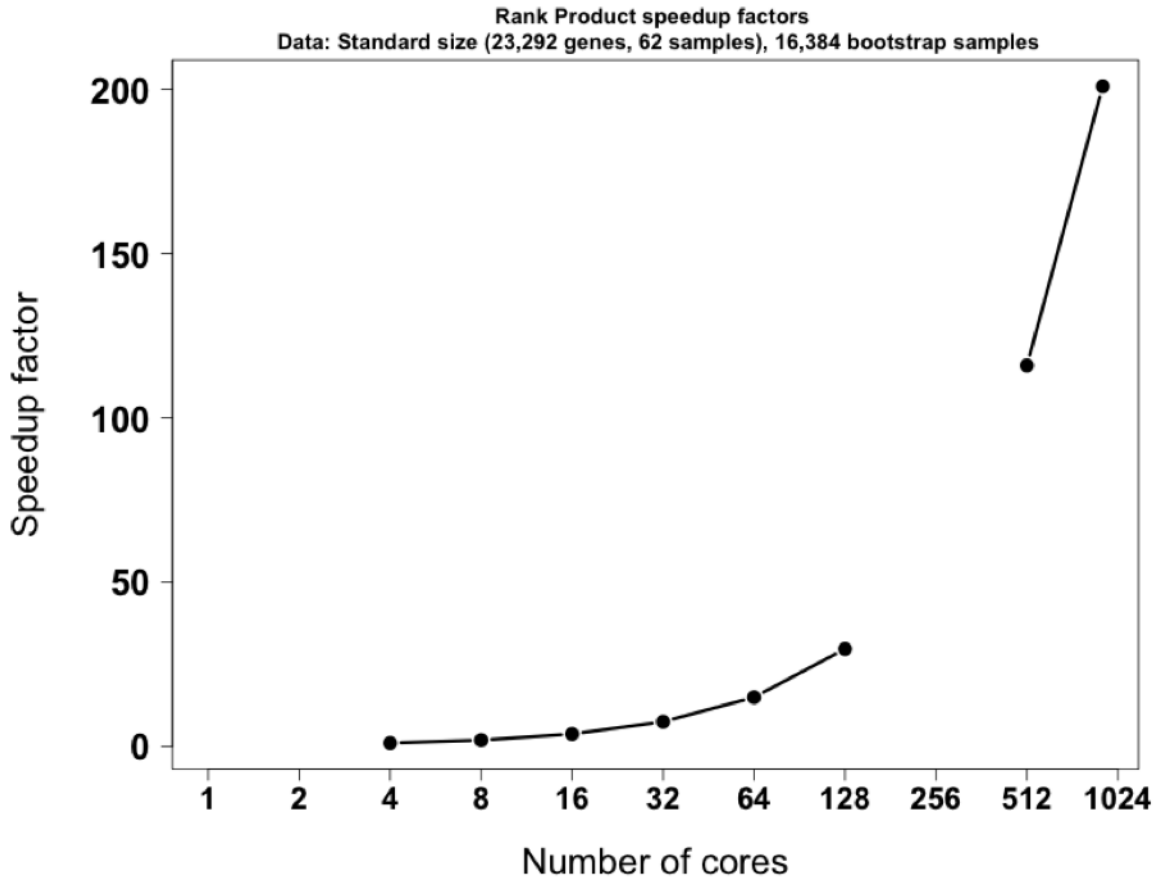

RandomForest speedup factors
Data: Large size (500,000 sequences, 62 samples), 128 trees per core
Speedup factor
0
10
20
30
40
50
60
1
2
4
8
16
32
64
128
256
512
1024
Number of cores


RandomForest run times
Data: Large size (500,000 sequences, 62 samples), 128 trees per core
Time (seconds) / number of cores
0
10
20
30
40
50
60
70
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

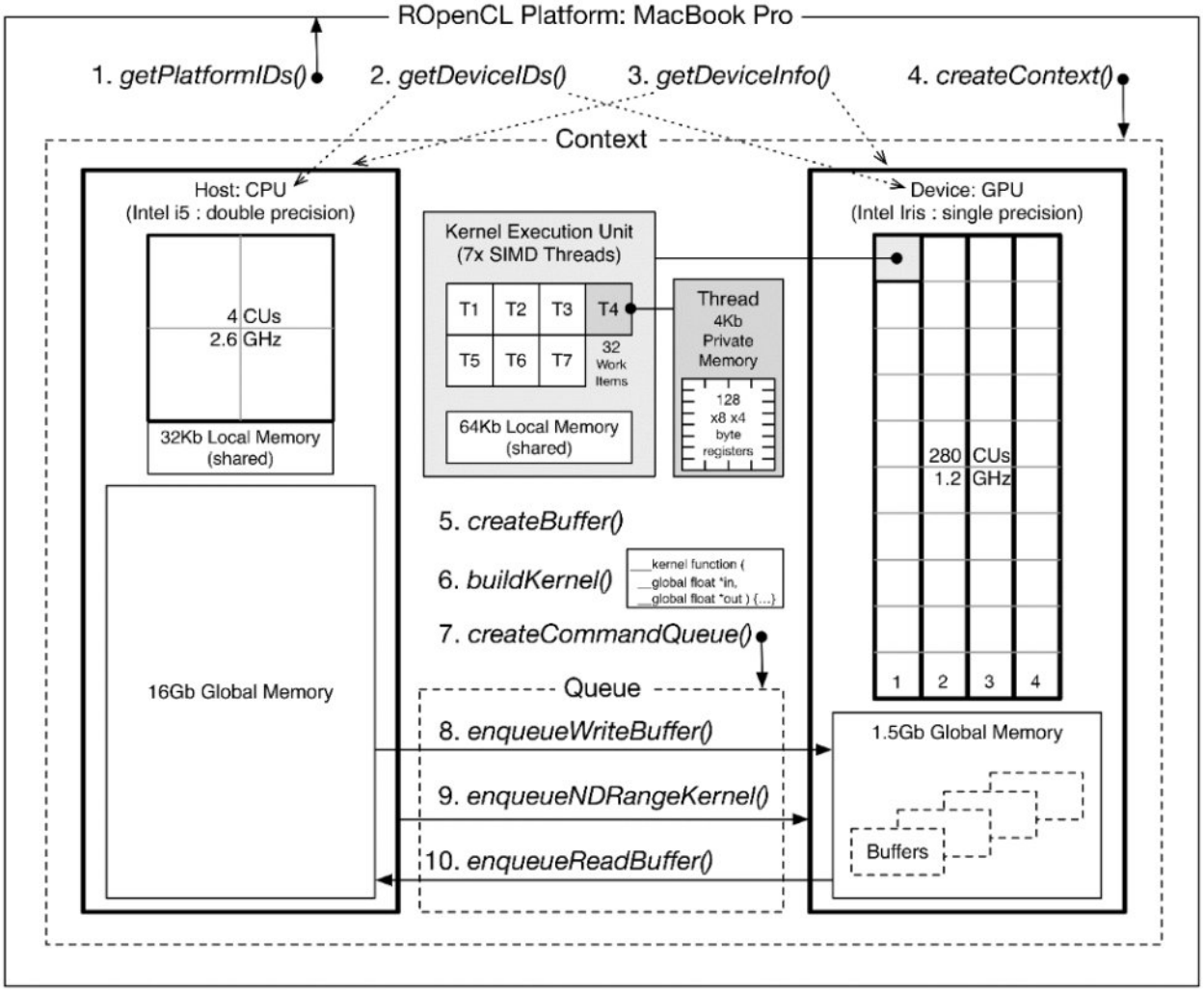

Rank Product run times
Data: Standard size (23,292 genes, 62 samples), 1024 bootstrap samples
Time (seconds)
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1
2
4
8
16
32
64
128
256
512
1024
Number of cores


Rank Product speedup factors
Data: Standard size (23,292 genes, 62 samples), 1024 bootstrap samples
Speedup factor
300
250
200
150
100
50
0
1
2
4
8
16
32
64
128
256
512
1024
Number of cores


Rank Product run time
Data: Standard size (23,292 genes, 62 samples), 16,384 bootstrap samples
Time (seconds)
40000
35000
30000
25000
20000
15000
10000
5000
0
1
2
4
8
16
32
64
128
256
512
1024
Number of cores


Rank Product speedup factors
Data: Standard size (23,292 genes, 62 samples), 16,384 bootstrap samples
Speedup factor
200
150
100
50
0
1
2
4
8
16
32
64
128
256
512
1024
Number of cores

Chapter 5: The Supercomputer in Your Laptop



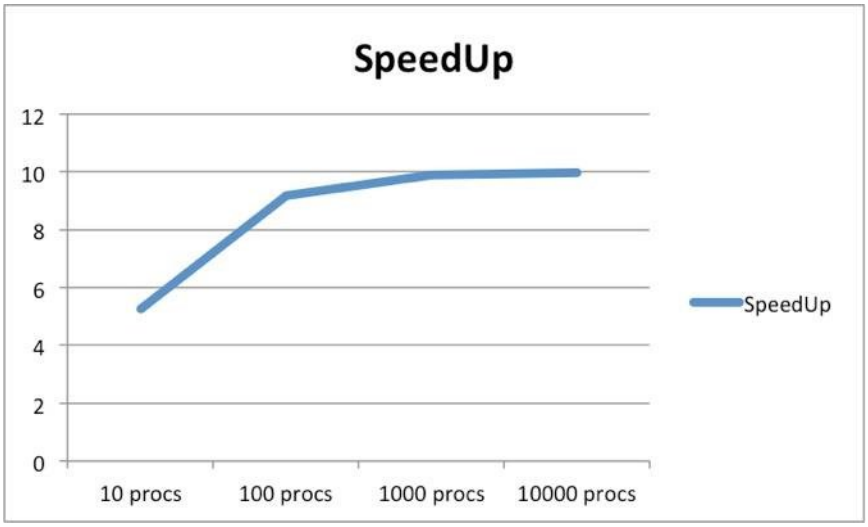
$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{N} \left(A[i] - B[i] \right)^2}$$

$$SpeedUp = \frac{T_{serial}}{T_{parallel}}$$

Perfect parallelism: $T_{parallel\_N} = \frac{T_{parallel\_1}}{N}$

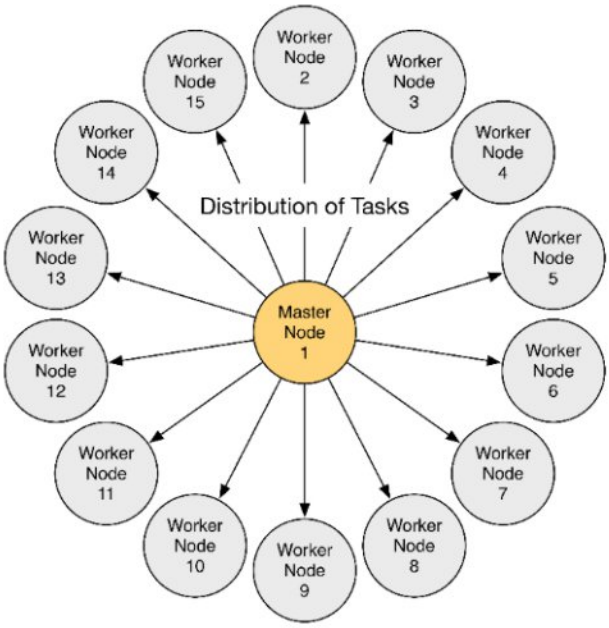
Total time: $T_{overall\_N} = T_{non-parallel} + T_{parallel\_N}$

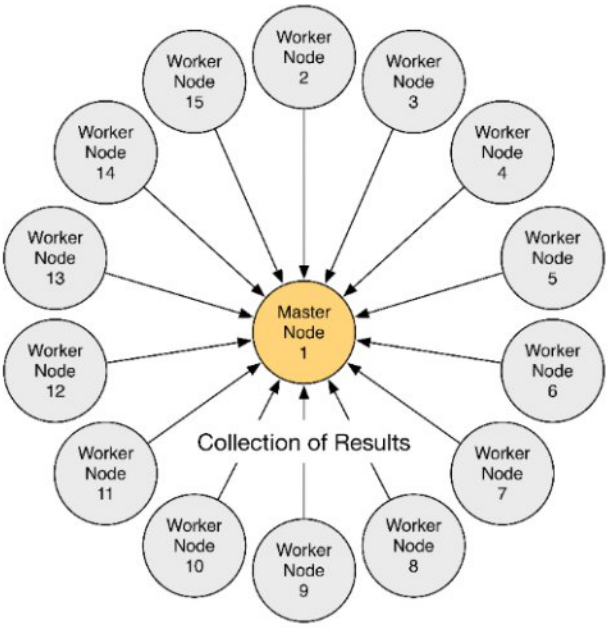
$$SpeedUp(N) = \frac{1}{(1-P) + \frac{P}{N}}$$

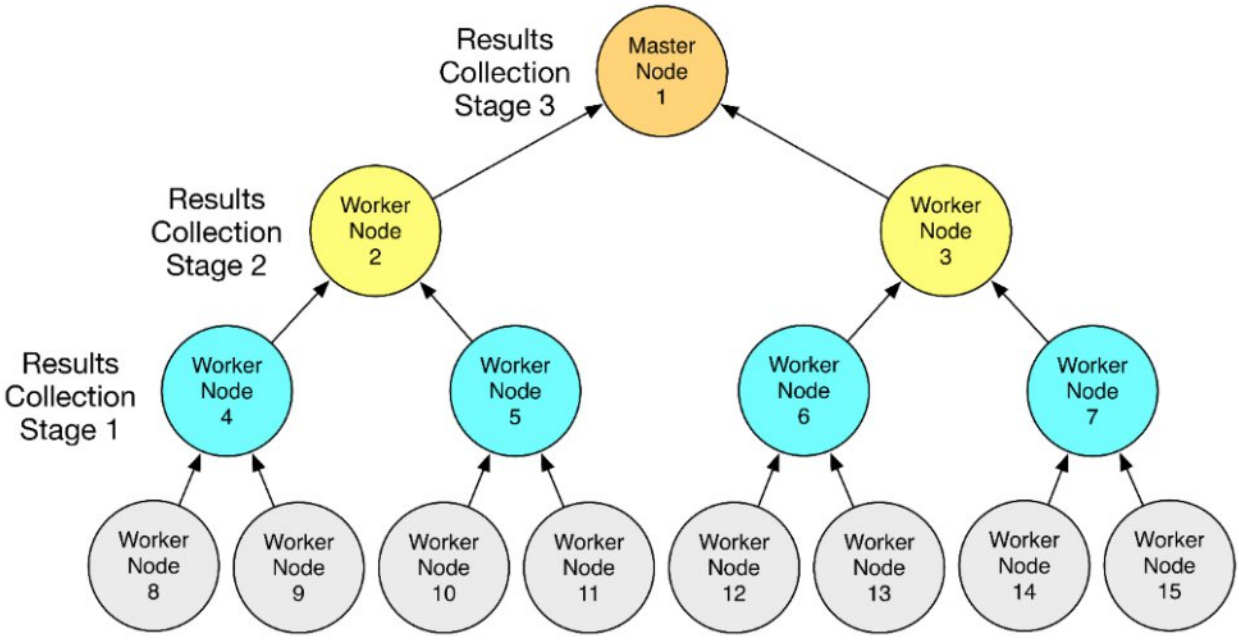


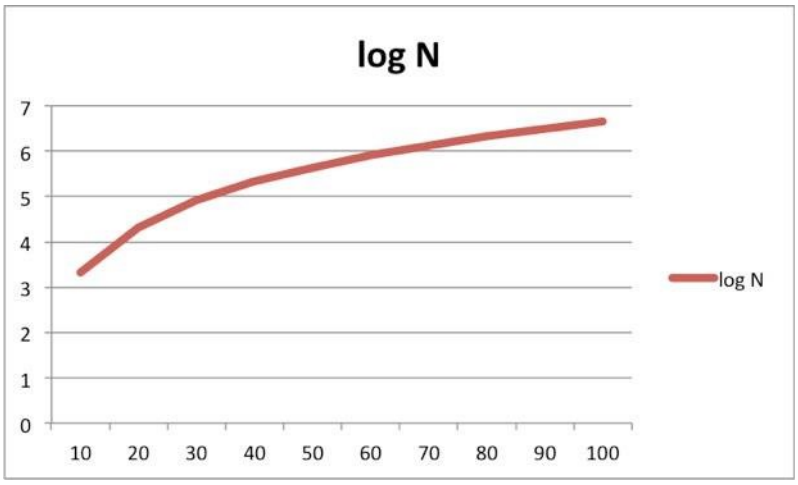
$$P_{estimated} = \frac{\frac{1}{SpeedUp} - 1}{\frac{1}{N} - 1}$$

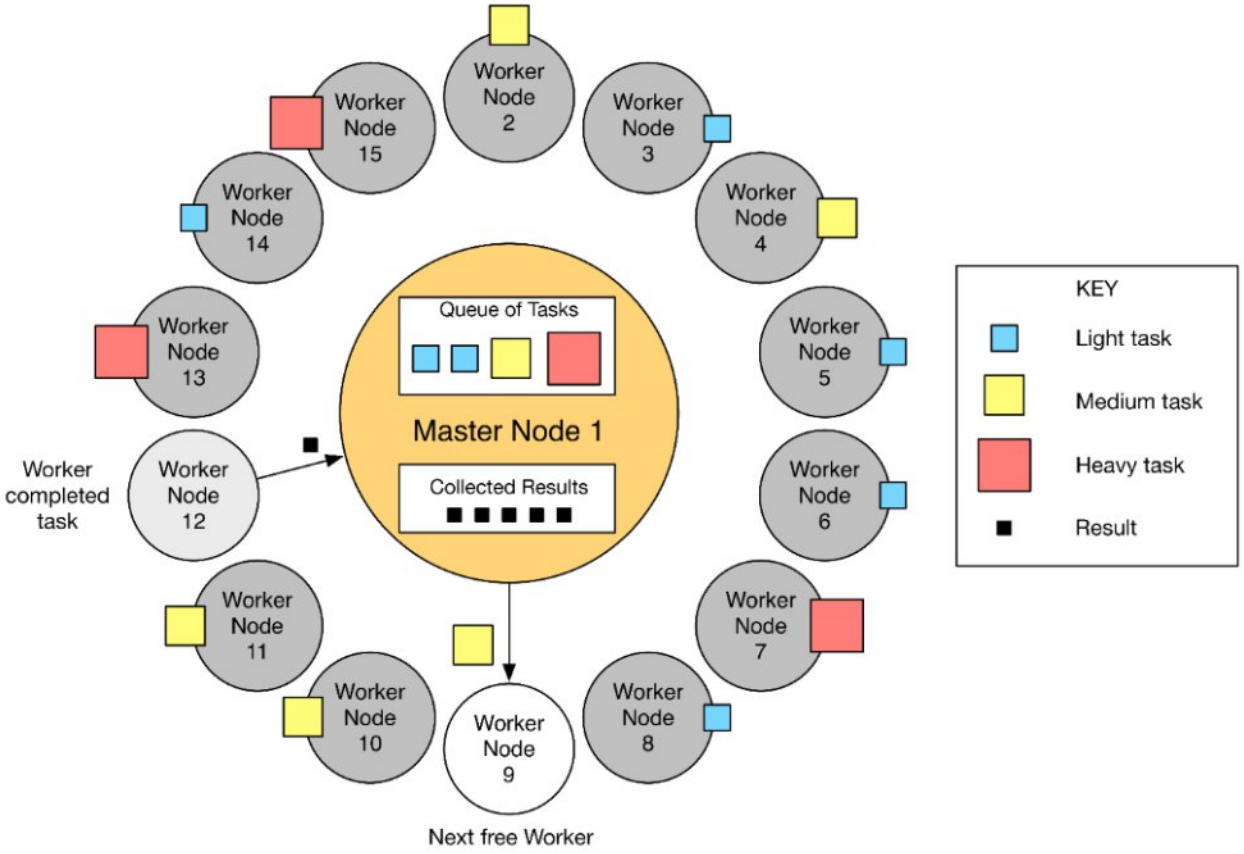
Chapple's Law: $(T_{parallel} \times N) + T_{parallel\_algorithm\_development} << T_{serial} \times N$

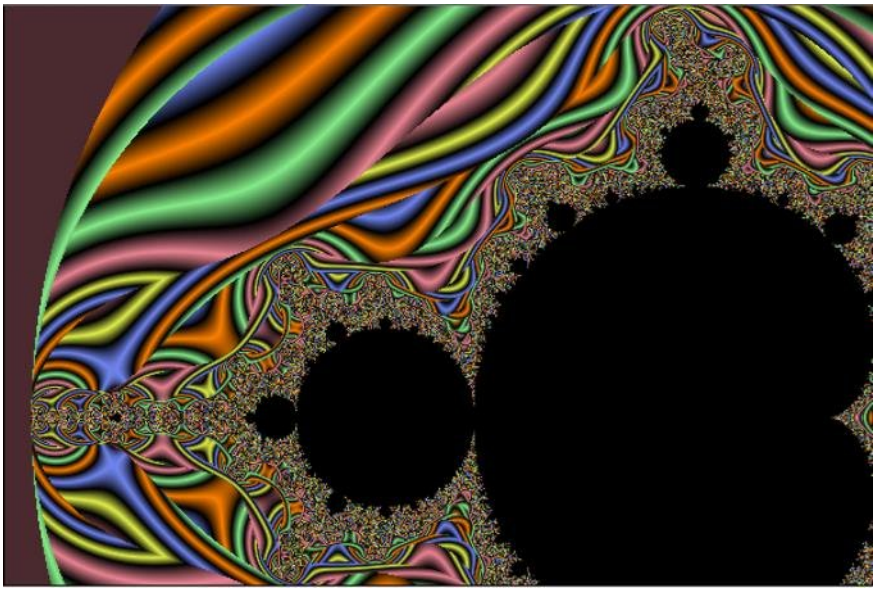

Distribution of Tasks
Master Node 1
Worker Node 2
Worker Node 3
Worker Node 4
Worker Node 5
Worker Node 6
Worker Node 7
Worker Node 8
Worker Node 9
Worker Node 10
Worker Node 11
Worker Node 12
Worker Node 13
Worker Node 14
Worker Node 15

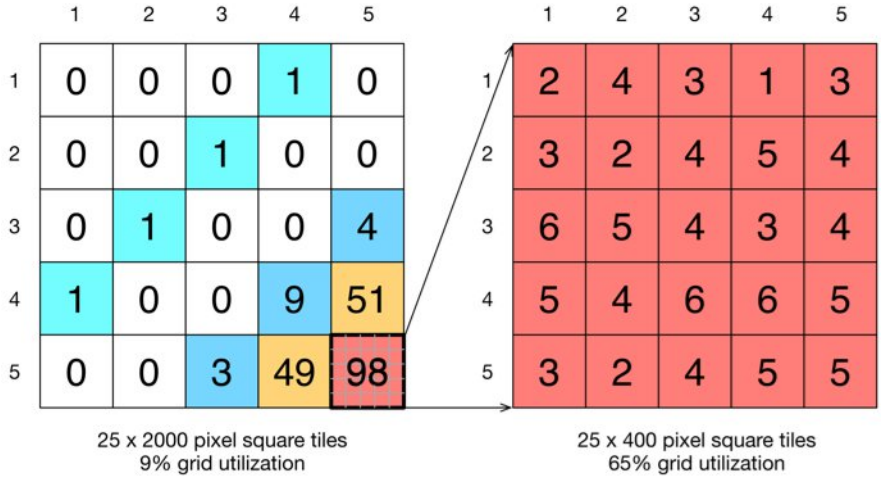

Collection of Results
Master Node 1
Worker Node 2
Worker Node 3
Worker Node 4
Worker Node 5
Worker Node 6
Worker Node 7
Worker Node 8
Worker Node 9
Worker Node 10
Worker Node 11
Worker Node 12
Worker Node 13
Worker Node 14
Worker Node 15


Results Collection Stage 3
Results Collection Stage 2
Results Collection Stage 1
Master Node 1
Worker Node 2
Worker Node 3
Worker Node 4
Worker Node 5
Worker Node 6
Worker Node 7
Worker Node 8
Worker Node 9
Worker Node 10
Worker Node 11
Worker Node 12
Worker Node 13
Worker Node 14
Worker Node 15


log N
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
80
90
100
log N


Worker Node 2
Worker Node 3
Worker Node 4
Worker Node 5
Worker Node 6
Worker Node 7
Worker Node 8
Worker Node 9
Worker Node 10
Worker Node 11
Worker Node 12
Worker Node 13
Worker Node 14
Worker Node 15
Queue of Tasks
Master Node 1
Collected Results
Worker completed task
Next free Worker
KEY
Light task
Medium task
Heavy task
Result




1 2 3 4 5
1 0 0 0 1 0
2 0 0 1 0 0
3 0 1 0 0 4
4 1 0 0 9 51
5 0 0 3 49 98
1 2 3 4 5
1 2 4 3 1 3
2 3 2 4 5 4
3 6 5 4 3 4
4 5 4 6 6 5
5 3 2 4 5 5
25 x 2000 pixel square tiles
9% grid utilization
25 x 400 pixel square tiles
65% grid utilization

Parallel computing: Applications

- The caret package by Kuhn can use various frameworks (MPI, NWS etc) to parallelized cross-validation and bootstrap characterizations of predictive models.
- The maanova package on Bioconductor by Wu can use snow and Rmpi for the analysis of micro-array experiments.
- The pvclust package by Suzuki and Shimodaira can use snow and Rmpi for hierarchical clustering via multiscale bootstraps.
- The tm package by Feinerer can use snow and Rmpi for parallelized text mining.
- The varSelRF package by Diaz-Uriarte can use snow and Rmpi for parallelized use of variable selection via random forests.
- The bcp package by Erdman and Emerson for the Bayesian analysis of change points can use foreach for parallelized operations.
- The multtest package by Pollard et al. on Bioconductor can use snow, Rmpi or rpvm for resampling-based testing of multiple hypothesis.
- The GAMBoost package by Binder for `glm` and `gam` model fitting via boosting using b-splines, the Geneland package by Estoup, Guillot and Santos for structure detection from multilocus genetic data, the Matching package by Sekhon for multivariate and propensity score matching, the STAR package by Pouzat for spike train analysis, the bnlearn package by Scutari for bayesian network structure learning, the latentnet package by Krivitsky and Handcock for latent position and cluster models, the lga package by Harrington for linear grouping analysis, the peperr package by Porzelius and Binder for parallised estimation of prediction error, the orloca package by Fernandez-Palacin and Munoz-Marquez for operations research locational analysis, the rgenoud package by Mebane and Sekhon for genetic optimization using derivatives the affyPara package by Schmidberger, Vicedo and Mansmann for parallel normalization of Affymetrix microarrays, and the puma package by Pearson et al. which propagates uncertainty into standard microarray analyses such as differential expression all can use snow for parallelized operations using either one of the MPI, PVM, NWS or socket protocols supported by snow.
- The bugsparallel package uses Rmpi for distributed computing of multiple MCMC chains using WinBUGS.
- The partDSA package uses nws for generating a piecewise constant estimation list of increasingly complex predictors based on an intensive and comprehensive search over the entire covariate space.
- The dclone package provides a global optimization approach and a variant of simulated annealing which exploits Bayesian MCMC tools to get MLE point estimates and standard errors using low level functions for implementing maximum likelihood estimating procedures for complex models using data cloning and Bayesian Markov chain Monte Carlo methods with support for JAGS, WinBUGS and OpenBUGS; parallel computing is supported via the snow package.
- The pmclust package utilizes unsupervised model-based clustering for high dimensional (ultra) large data. The package uses pbdMPI to perform a parallel version of the EM algorithm for finite mixture Gaussian models.
- The harvestr package provides helper functions for (reproducible) simulations.
- Nowadays, many packages can use the facilities offered by the **parallel** package. One example is pls, another is PGICA which can run ICA analysis in parallel on SGE or multicore platforms.