LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
IBE MEDIZINISCHE FAKULTÄT
INSTITUT FÜR MEDIZINISCHE INFORMATIONSVERARBEITUNG BIOMETRIE UND EPIDEMIOLOGIE

State-of-the-art in Parallel Computing with R

Markus Schmidberger
(schmidb@ibe.med.uni-muenchen.de)

The R User Conference 2009
July 8-10, Agrocampus-Ouest, Rennes, France

useR!

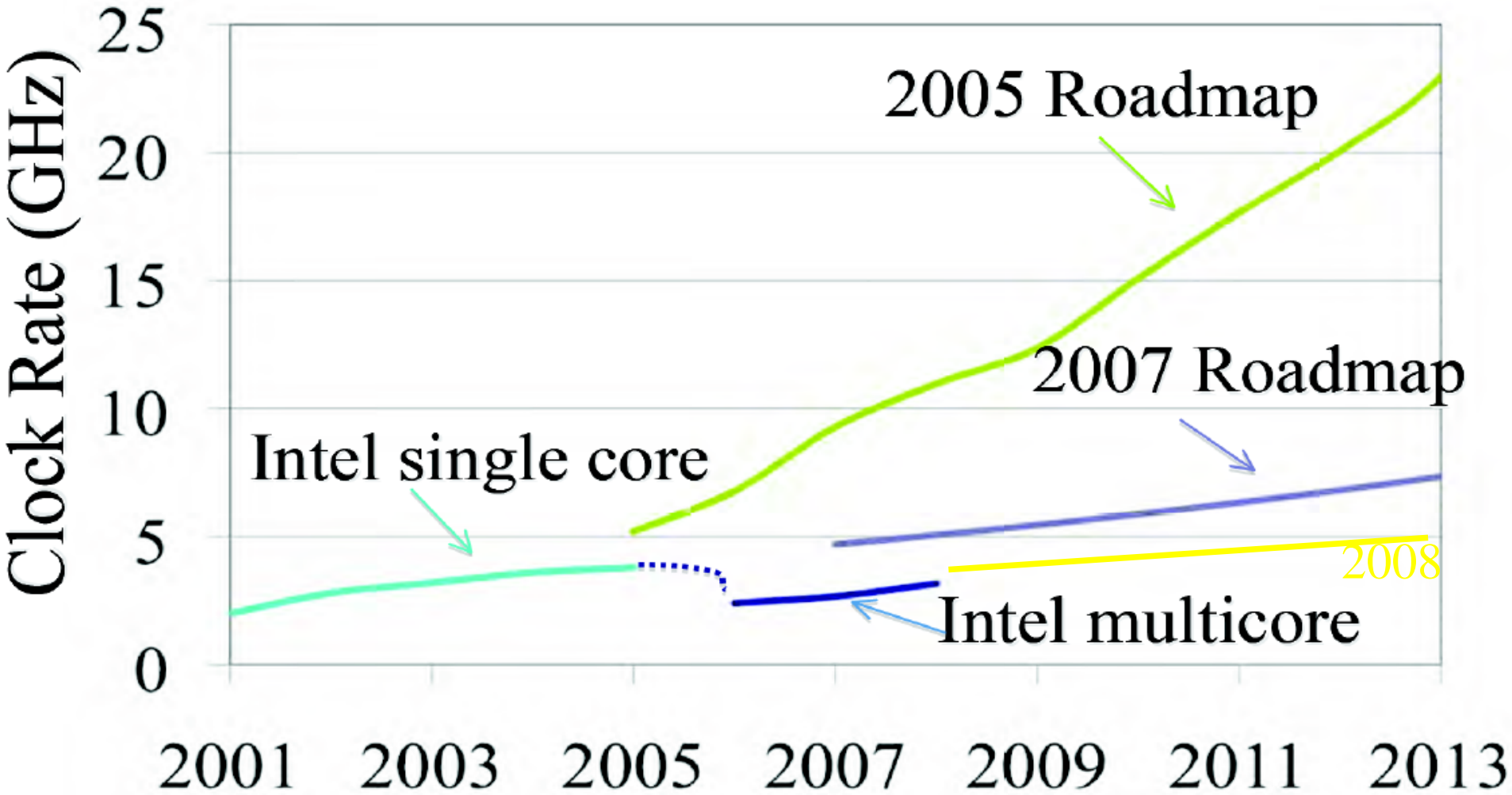
The Future is Parallel

Prof. Bill Dally, Nvidia, 01-2009

Thilo Kielmann, University of Amsterdam, 12-2008

ITRS Roadmap 2005 and 2007

International Technology Roadmap for Semiconductors



[D. Patterson, USENIX 2008 keynote]

New Paper

submitted in December – State-of-the-art at the end of 2008

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

State-of-the-art in Parallel Computing with R

Markus Schmidberger
Ludwig-Maximilians-Universität
München, Germany

Martin Morgan
Fred Hutchinson Cancer
Research Center, WA, USA

Dirk Eddelbuettel
Debian Project,
Chicago, IL, USA

Hao Yu
University of Western Ontario,
ON, Canada

Luke Tierney
University of Iowa,
IA, USA

Ulrich Mansmann
Ludwig-Maximilians-Universität
München, Germany

- State of development
- Technology
- Fault-Tolerance & Load balancing
- Usability
- Acceptance
- Performance

Preprint: http://epub.ub.uni-muenchen.de/8991/

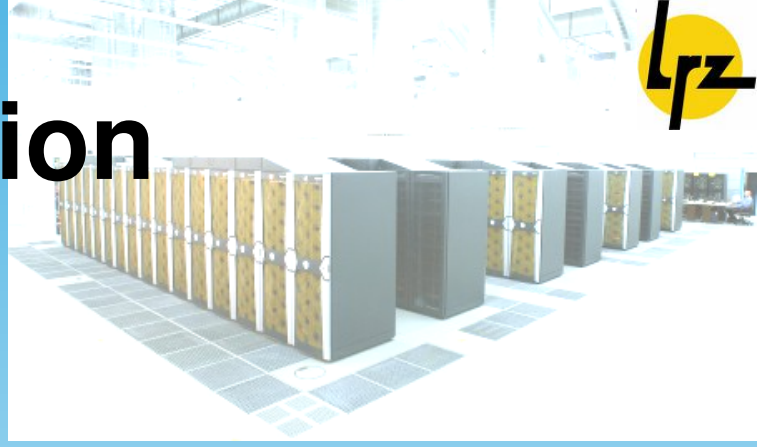
Parallel Program Design

- Convert serial programs into parallel programs
  - compiler or pre-processor
  - prefer manual to automatically parallelization
    - wrong results may be produced,
    - performance may actually degrade,
    - much less flexible than manual parallelization,
    - code is too complex for automatical parallelization, etc..

- Very manual process of identifying and implementing parallelism

- Analysing the serial Code
  - understand serial code
  - profilers and performance analysis tools exist
  - identify program's hotspots or bottelnecks.
  - In the R language:
    - profile R code for memory use and evaluation time
    - ?Rprof

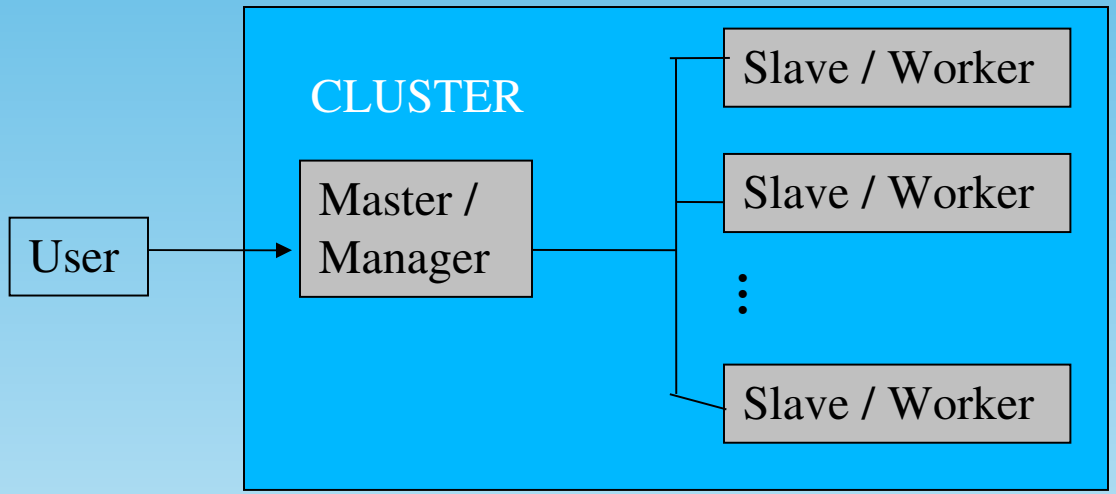
Parallelization



- Multiprocessors
  - the use of two or more central processing units (CPUs) within a single computer system
  - Today: Two/Four-processors are becoming a standard for workstations

- Multicomputers
  - different parts of a program run simultaneously on two or more computers that are communicating with each other over a network
  - Computer, network, software
  - Cluster, Grid

Master-Slave Architecture

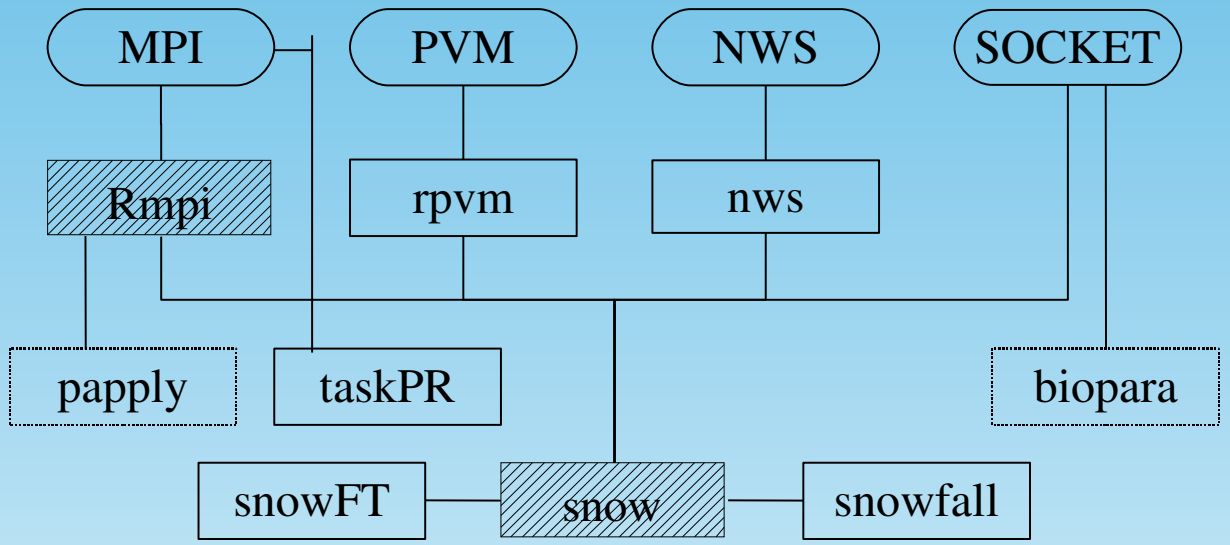


- Works on computer clusters, on multiprocessor machines and in grid computing
- You need underlying technology for communication
  - MPI: Message Passing Interface
  - PVM: Parallel Virtual Machine
  - socket, ssh
  - (NWS: NetWorkSpace)

Parallel R Packages

Package	Version	Websites	Technology
Computer Cluster			
Rmpi	0.5-6	http://cran.r-project.org/web/packages/Rmpi http://www.stats.uwo.ca/faculty/yu/Rmpi	MPI
rpvm	1.0.2	http://cran.r-project.org/web/packages/rpvm http://www.biostat.umn.edu/~nali/SoftwareListing.html	PVM
nws	1.7.0.0	http://cran.r-project.org/web/packages/nws http://nws-r.sourceforge.net	NWS and socket
snow	0.3-3	http://cran.r-project.org/web/packages/snow http://www.cs.uiowa.edu/~luke/R/cluster	Rmpi, rpvm, nws, socket
snowFT	0.0-2	http://cran.r-project.org/web/packages/snowFT	rpvm, snow
snowfall	1.60	http://cran.r-project.org/web/packages/snowfall http://www.imbi.uni-freiburg.de/parallel	snow
papply	0.1	http://cran.r-project.org/web/packages/papply http://math.acadiau.ca/ACMMaC/software/papply.html	Rmpi
biopara	1.5	http://cran.r-project.org/web/packages/biopara http://hedwig.mgh.harvard.edu/biostatistics/node/20	socket
taskPR	0.31	http://cran.r-project.org/web/packages/taskPR http://users.ece.gatech.edu/~gte810u/Parallel-R	MPI (only LAM/MPI)
Grid Computing			
GridR	0.8.4	http://cran.r-project.org/web/packages/GridR http://www.stefan-rueping.de	web service, ssh, Condor, Globus
multiR	-	http://e-science.lancs.ac.uk/multiR	3 tier client/server architecture
Biocep-R	NA	http://biocep-distrib.r-forge.r-project.org	java 5
Multi-core System			
pnmath(0)	0.2	http://www.cs.uiowa.edu/~luke/R/experimental	openMP, Pthreads
fork	1.2.1	http://cran.r-project.org/web/packages/fork	Unix: fork
R/Parallel	0.6-20	http://www.rparallel.org	C++, file
romp	0.1a	http://code.google.com/p/romp	openMP
multicore	0.1-3	http://cran.r-project.org/web/packages/multicore	fork

Computer Cluster R Packages



XXX R Package

XXX Technology

XXX No longer maintained

Performance evaluation of R packages for computer clusters

	Component 1	Component 2	Component 3
Rmpi	0.91	1.83	3.00
nws	14.87	3.49	3.03
snow			
MPI	10.33	2.00	2.96
PVM	4.13	1.01	2.93
NWS	8.68	1.58	2.98
Socket	3.47	0.93	2.90
snowfall			
MPI	13.99	2.07	2.93
PVM	4.29	0.94	2.93
NWS	8.82	1.62	2.99
Socket	3.77	1.00	2.91

Component 1: Sending Data from the Master to all Slaves (matrix 500 x 500)
Component 2: Distributing a List of Data from the Master to the Slaves (list of matrices 500 x 500)
Component 3: Compute intregral of a three-dimensional function (10.000 points)

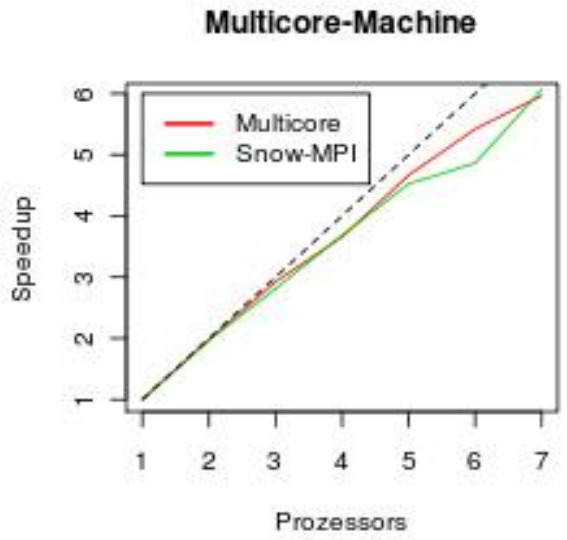
Performance evaluation of R packages for computer clusters

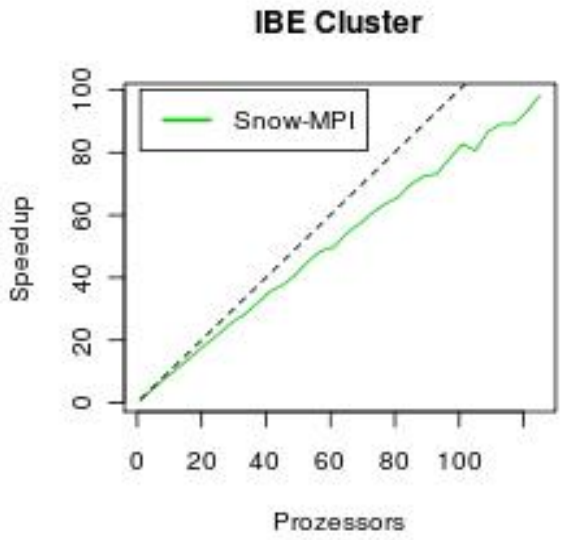
	Component 1	Component 2	Component 3
Rmpi	29.1	18.6	21.9
nws	97.3	34.8	21.2
snow			
MPI	103.2	20.1	20.5
PVM	41.2	10.1	20.5
NWS	86.7	16.0	20.8
Socket	34.8	9.3	20.2
snowfall			
MPI	109.6	20.9	20.5
PVM	43.0	9.9	20.6
NWS	88.0	16.3	20.9
Socket	37.1	9.9	20.3

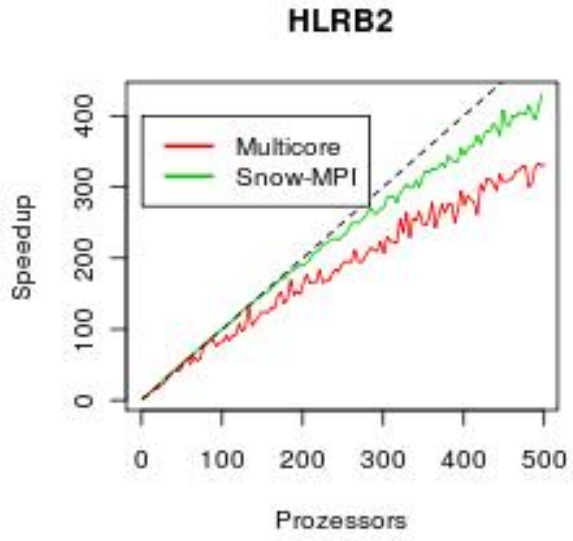
Performance - Sudoku

- R package: **sudoku**_2.2
- Generates, plays, and solves Sudoku puzzles.
- Solve 10.000 Sudokus
- Distribute Sudokus equally to all nodes
- The basic rules of Sudoku are used to fill in missings, then elimination is used to find the TRUE's. If that approach runs out of steam, a guess is made and the program recurses to find either a solution or an inconsistency.

Performance - Sudoku







State of the Art in Parallel Computing with R

- Computer Cluster: **Rmpi** and **snow**
  - acceptable usability, wide spectrum of functionality, good performance
  - Other packages: Usability <-> lower functionality

- Multi-core: in development
  - **Multicore** package ↔ Windows ?
  - external and architecture optimized libraries (PBLAS)
    - bottleneck in statistical computation?
  - Multicomputer packages: **Rmpi** and **snow**
    - every R instance requires its own main memory!

- Grid Computing: early-stage packages

Which package should I use?

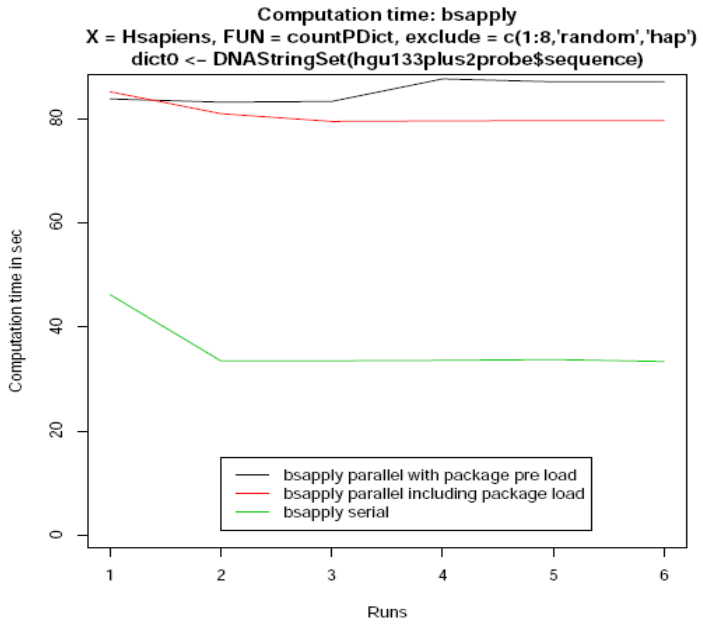
- Depends on your available hardware:

- Multicore machine: **multicore**
- Cluster environment:
  - **Snow**(-**fall**) with the available communication mechanism (MPI mostly used)
  - **NWS**, if you have a lot of global variables
  - **Rmpi**, for excellent programmer and for high end optimazation

- Grid Computing: **gridR**, which statistical application is usefull for grid computing?

Tips for Parallel Computing

- Communication is much slower than computation.
  - functions produce large results, reduce results on the worker before returning.
  - additional function parameters can be huge.



bsapply and countPDict Example

```
R> params <- new("BSParams", X = Hsapiens,
       FUN = countPDict)
R> library(hgu133plus2probe)
R> dict0 <- DNAStringSet(hgu133plus2probe$sequence)
R> pdict0 <- PDict(dict0)
R> bsapply(params, pdict = pdict0)
```

Tips for Parallel Computing

- Random Generators have to used with care; special-purpose packages **rsprng**, **rlecuyer** (and **snow**) are available.

```
R> clusterCall(cl, runif, 3)
[[1]]
[1] 0.4351672 0.7394578 0.2008757
[[2]]
[1] 0.4351672 0.7394578 0.2008757
...
[[10]]
[1] 0.4351672 0.7394578 0.2008757
```

```
R> clusterSetupSPRNG(cl)
R> clusterCall(cl, runif, 3)
[[1]]
[1] 0.014266542 0.749391854 0.007316102
[[2]]
[1] 0.8390032 0.8424790 0.8896625
...
[[10]]
[1] 0.591217470 0.121211511 0.002844222
```

- lexical scoping: requires some care to avoid transmitting unnecessary data to workers
  - Functions used in apply-like calls should be defined in the global environment, or in a package name space.

HELP

- „State-of-the-Art in Parallel Computing with R“; Schmidberger, et.al.; JSS 2009

- CRAN Task View 'High Performance and Parallel Computing'
  - http://cran.r-project.org/web/views/HighPerformanceComputing.html

- R Mailinglist 'R SIG on High-Performance Computing'
  - https://stat.ethz.ch/mailman/listinfo/r-sig-hpc

Conclusion & Future

- Parallel Computing can help improving performance,
  - but first of all improve your serial code (profiling, vectorization, ...).
  - but be careful with communication costs.
- First Parallel Implementations are easy,
  - but there are a lot of stumbling blocks.
- Parallel Computing with R needs to be improved:
  - Teach R useres to think in parallel
  - Integration of R code into multi-core environments
  - Cloud Computing with R
  - Computing power of graphic processing units
- Flexibility of R package system allows integration of many different technologies.

Acknowledgment

Parallel R

Martin Morgan

Dirk Eddelbuettel

Hao Yu

Luke Tierney

Anthony Rossini

LRZ

HPC Team

Ferdinand Jamitzky

200.000 CPUh

AffyPara Package

Ulrich Mansmann

Esmeralda Vicedo

Klaus Rüschstroer

Robert Gentlemans Group

Dipl.-Tech. Math. Markus Schmidberger
schmidb@ibe.med.uni-muenchen.de
http://ibe.med.uni-muenchen.de

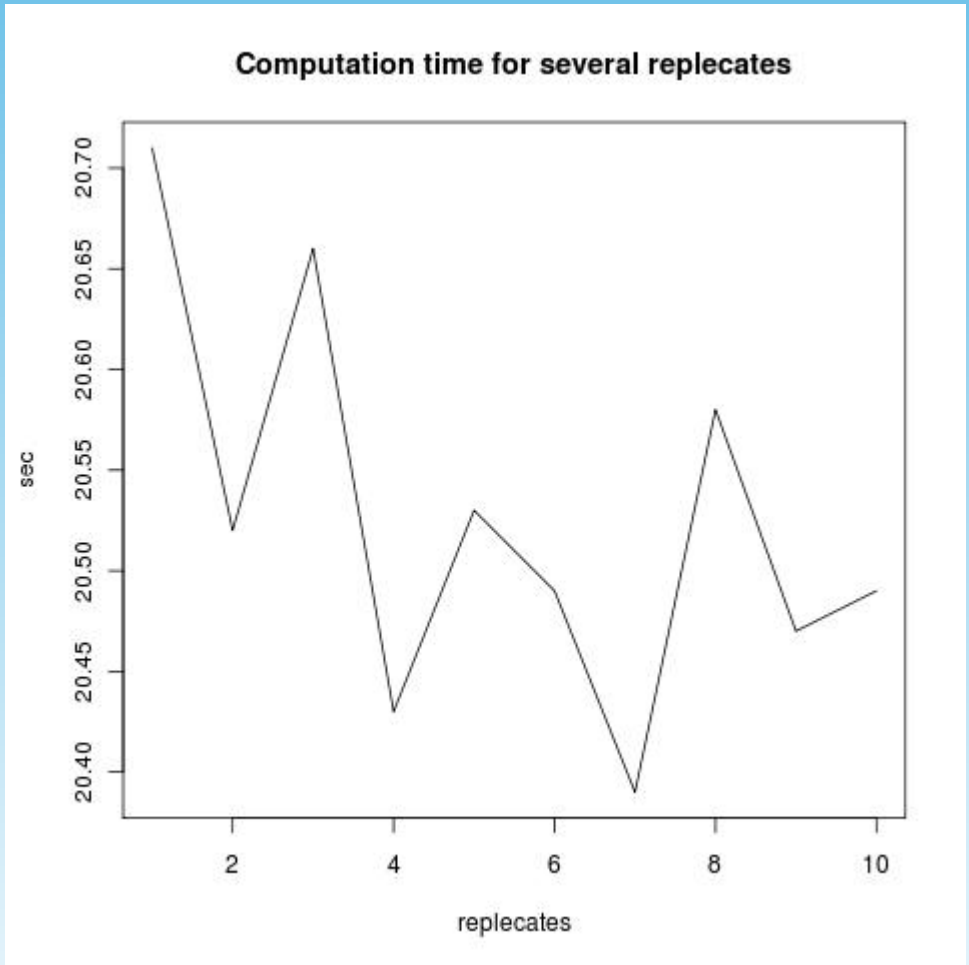
Thanks for your attention

State of the Art in Parallel Computing with R

	Learnability	Efficiency	Memorability	Errors	Satisfaction
Rmpi	+	−−	++	+	+
rpvm	−−	−−	+	−	−−
nws	+	+	++	+	+
snow	+	++	++	+	++
snowFT	+	++	++	+	++
snowfall	++	++	++	+	++
papply	+	+	++	+	0
biopara	−−	−−	0	0	−−
taskPR	+	−	++	0	−

	Learnability	Efficiency	Memorability	Errors	Satisfaction
Grid Computing					
gridR	+	−	+	−	−
R/parallel	0	+	+	−	0
Multi-core System					
pnmath(0)	++	++	++	++	+
fork	+	−	+	−	−
Multicore	+	++	++	0	+

Computation Time for 10 replicates



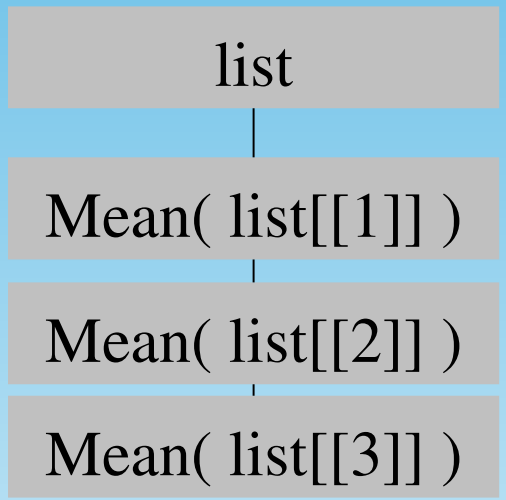
Simple Parallelization

L <- list(a=c(1:10), b=c(2:12), c=c(4:14))

Seriell:

for(i in 1:3) { res[[i]] <- mean(L[[i]]) }

res <- lapply(L, mean)



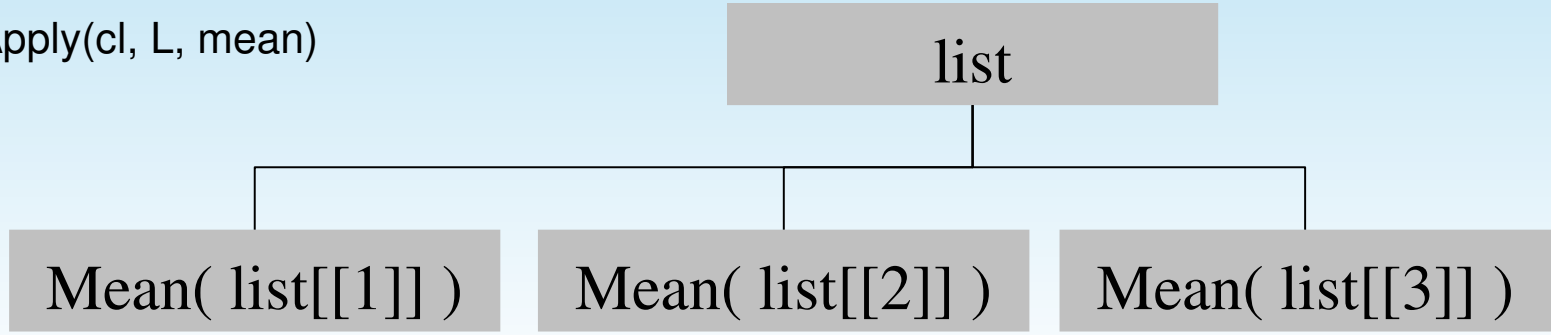
Parallel:

library(**snow**)

cl <- makeCluster(3, type='SOCK')

res <- clusterApply(cl, L, mean)

stopCluster(cl)



Simple Parallelization for Statisticans

- Bootstraping: time-consuming and simple to parallelize
- library(**boot**): generating bootstrap replicates
- Example: generalized linear model fit for data on the cost of constructing nuclear power plants. 999 bootstraps
- Serial: 9.2 sec <-> 3 nodes: 3.1 sec