

ismxc, ismsr, ismaltx の 各 環境

2006 6 月現在 (ismxc 2006/10 改定)

Ei-ji Nakama

nakama@com-one.com

株式会社 COM-ONE

並列 R 環境

並列  環境
と
必要な設定及び環境変数

全環境共通 - 環境変数設定

- bsh 系
source /usr/local1/bin/env_local1.sh
- csh 系
source /usr/local1/bin/env_local1.csh

~/.bash_profile や ~/.cshrc に追加する場合は、
ismsr, ismaltx では mpi 関連のコマンドが LAM-MPI
になる。ismxc は hp-mpi を利用するので差異は無
い。ことに注意

ismxc - R の選択方法

~/.Rconf ファイルに設定^a

```
RVER=2.2.1           # R Version.
RCOMPILER=path       # PathScale 2.2.1
#RCOMPILER=intel     # intel compiler 9.0
#RCOMPILER=gcc       # gcc-3.4.5
#RBLAS=              # normal R BLAS
RBLAS=gotop          # GotoBLAS 1.02 pthread
                    #      (1 から 2 並列まで)
```

^a/usr/local1/bin/R で処理

ismsr - R の選択方法

~/.Rsrconf ファイルに設定^a

```
RVER=2.2.1           # R version
RCOMPILER=gcc        # intel8 compiler
#RBLAS=              # normal RBLAS
RBLAS=gotop          # GotoBLAS 1.02 pthread
                     # (1 から 16 並列まで)
```

^a/usr/local1/bin/R で処理

ismaltx - R の選択方法

~/.Rconf ファイルに設定^a

```
RVER=2.2.1           # R version
RCOMPILER=intel8     # intel8 compiler
#RCOMPILER=gcc       # gcc-3.4.6
#RBLAS=               # normal RBLAS
#RBLAS=atlas         # ATLAS 3.6.0
#RBLAS=atlaspthread # ATLAS 3.6.0 pthread
#                   # (1 から 8 並列まで)
RBLAS=gotop         # GotoBLAS 1.01
#                   # (1 から 8 並列まで)
```

^a/usr/local1/bin/R で処理

コンソールから



コンソールからの対話的利用方法

コンソールから Windows

- ssh クライアント (PuTTY) などのインストール

<http://hp.vector.co.jp/authors/VA024651/>

- *セキュリティの関係で鍵の生成が必要な場合

<http://www.chiark.greenend.org.uk/%7esgtatham/putty/download.html>

PuTTYgen は本家のモジュールに入っています

コンソールから R 起動

```
[xxxxxxx@ismaltx xxxxxx]$ R
```

```
R : Copyright 2006, The R Foundation for Statistical Computing  
Version 2.2.1 Patched (2006-02-10 r37327)  
ISBN 3-900051-07-0
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.
```

```
Type 'q()' to quit R.
```

コンソールから R 終了

```
[xxxxxxx@ismaltx xxxxxxx]$ R  
> pdf("hoge.pdf")      # PDF デバイスのオープン  
> plot(1:10)           # PDF デバイスに作図  
> dev.off()            # PDF デバイスのクローズ  
> q("no")
```

コンソールから キーバインド 1

M/C	Key	Rgui	Rterm	Terminal
	Home	行頭へ	行頭へ	行頭へ
	End	行末へ	行末へ	行末へ
	PageUp	表示バッファ単位で遡る		履歴の先頭
	PageDown	表示バッファ単位で進める		履歴の末尾
	Up	履歴を遡る	履歴を遡る	履歴を遡る
	Down	履歴を進める	履歴を進める	履歴を進める
	Left	カーソルを 1 文字左へ	カーソルを 1 文字左へ	カーソルを 1 文字左へ
	Right	カーソルを 1 文字右へ	カーソルを 1 文字右へ	カーソルを 1 文字右へ
	Delete	カーソル上の文字を削除	カーソル上の文字を削除	カーソル上の文字を削除

コンソールから キーバインド2

M/C	Key	Rgui	Rterm	Terminal
Ctrl	Home	スクロールバッファ先頭へ		
Ctrl	End	スクロールバッファ末尾へ		
Ctrl	Up	1 行上スクロール		
Ctrl	Down	1 行下スクロール		
Ctrl	Left	5 カラム左スクロール		
Ctrl	Right	5 カラム右スクロール		
Ctrl	Delete	カーソル以降を削除		

コンソールから キーバインド3

M/C	Key	Rgui	Rterm	Terminal
Ctrl	A	行頭へ	行頭へ	行頭へ
Ctrl	B	1文字左へ	1文字左へ	1文字左へ
Ctrl	C	カット	Kill	Kill
Ctrl	D		Quit	Quit
Ctrl	E	行末へ	行末へ	行末へ
Ctrl	F	1文字右へ	1文字右へ	1文字右へ
Ctrl	G			Bell
Ctrl	H	後退	後退	後退
Ctrl	I			補間

コンソールから キーバインド4

M/C	Key	Rgui	Rterm	Terminal
Ctrl	J	改行	改行	改行
Ctrl	K	カーソル以降を削除	カーソル以降を削除	カーソル以降を削除
Ctrl	L	改頁して再描画	改行	改頁して再描画
Ctrl	M	改行	改行	改行
Ctrl	N	履歴を進める	履歴を進める	履歴を進める
Ctrl	O	上書きモード	上書きモード	上書きモード
Ctrl	P	履歴を遡る	履歴を遡る	履歴を遡る
Ctrl	R			履歴の検索

コンソールから キーバインド5

M/C	Key	Rgui	Rterm	Terminal
Ctrl	T	1 文字前と入替	1 文字前と入替	1 文字前と入替後に 右移動
Ctrl	U	現在行削除	現在行削除	カーソル以前削除
Ctrl	V	ペースト		
Ctrl	W		前方単語の削除	前方単語の削除
Ctrl	X	カット		
Ctrl	Y	ペースト		
Ctrl	Z	Quit	Quit	バックグラウンドへ

コンソールから キーバインド 6

M/C	Key	Rgui	Rterm	Terminal
Alt	B			カーソルを単語単位で左へ
Alt	C			単語の先頭を大文字 それ以外は小文字
Alt	D			後方単語削除
Alt	F			カーソルを単語単位で右へ
Alt	L			単語の小文字化
Alt	T			単語の入替
Alt	U			単語の大文字化

簡単な並列処理の利用方法



簡単な並列処理の利用方法 参考 URL

<http://www.stat.uiowa.edu/~luke/R/cluster/cluster.html>

<http://www.stat.uiowa.edu/~luke/R/cluster/uiowasnow.html>

ex1. snow4Rmpi 共通

R のコード

```
> require(snow)
> cl <- makeCluster(4, type = "MPI")
> myfunc <- function() {
+   c(Sys.info()[c("nodename", "machine")], Sys.getpid())
+ }
> result <- clusterCall(cl, myfunc)
> do.call("rbind", result)
> stopCluster(cl)
```

ex1. snow4Rmpi on ismxc

ShellScript

```
#!/bin/sh
#BSUB -n 10 # CPU 数 (親 1+子 4) * GOTO_NUM_THREADS(2)
#BSUB -o snow4Rmpi.out
#BSUB -e snow4Rmpi.err
#BSUB -ext "SLURM[nodes=5]" # いくつのノードを使うか
export GOTO_NUM_THREADS=2
source /usr/local1/bin/env_local1.sh
mpirun -spawn -srun -n 1 R CMD BATCH --no-save snow4Rmpi.R
# -srun の後ろの '-n 1' は srun のオプション
```

CommandLine

```
$ bsub -q q16t < snow4Rmpi.ismxc.sh
```

Result

```
[1,] "ismxc65" "x86_64" "1234"
[2,] "ismxc66" "x86_64" "32494"
[3,] "ismxc67" "x86_64" "32422"
[4,] "ismxc68" "x86_64" "32065"
```

ex1. snow4Rmpi on ismxc

ShellScript

```
#!/bin/sh
#BSUB -n 5 # CPU数 (親 1+子 4) * GOTO_NUM_THREADS(1)
#BSUB -o snow4Rmpi.out
#BSUB -e snow4Rmpi.err
#BSUB -ext "SLURM[nodes=3]" # ノードをいくつ使うか
export GOTO_NUM_THREADS=1
source /usr/local1/bin/env_local1.sh
mpirun -spawn -srun -n 1 R CMD BATCH --no-save snow4Rmpi.R
# -srun の後ろの '-n 1' は srun のオプション
```

CommandLine

```
$ bsub -q q8 < snow4Rmpi.ismxc.sh
```

Result

```
[1,] "ismxc58" "x86_64" "20819"
[2,] "ismxc59" "x86_64" "20919"
[3,] "ismxc59" "x86_64" "20921"
[4,] "ismxc60" "x86_64" "21013"
```

ex1. snow4Rmpi on ismsr

ShellScript

```
#!/bin/bash
#@class=P4
#@job_type=parallel
#@node=1
#@output=snow4Rmpi_$(jobid).out
#@error=snow4Rmpi_$(jobid).err
#@environment=COPY_ALL
#@queue
source /usr/local1/bin/env_local1.sh
mpiexec -boot N R CMD BATCH --no-save snow4Rmpi.R
```

CommandLine

```
$ llsubmit ./snow4Rmpi.ismsr.sh
```

Result

```
[1,] "srnd04" "001020034C00" "569482"
[2,] "srnd04" "001020034C00" "208946"
[3,] "srnd04" "001020034C00" "442492"
[4,] "srnd04" "001020034C00" "426214"
```

ex1. snow4Rmpi on ismaltx

ShellScript

```
#!/bin/bash
#PBS -q q8
#PBS -j oe
#PBS -l ncpus=8
export GOTO_NUM_THREADS=1
source /usr/local1/bin/env_local1.sh
mpiexec -boot N R CMD BATCH --no-save snow4Rmpi.R
```

CommandLine

```
$ qsub ./snow4Rmpi.ismaltx.sh
```

Result

```
[1,] "ismaltx3" "ia64" "8106"
[2,] "ismaltx3" "ia64" "8104"
[3,] "ismaltx3" "ia64" "8105"
[4,] "ismaltx3" "ia64" "8103"
```

ex2. 物理乱数 altix4 R

```
> # rxxxx 系の全ての関数が物理乱数を利用します
> N<-8000                                # サイズ
> library(RNDMaster)                     # ライブラリロード
> system.time(runif(N^2)) # MT 生成速度
> kind.old<-RNGkind()                    # MT の保存
> init.RNDMaster()                       # 物理乱数開始
> system.time(RNDMaster(N^2))# 物理乱数直接生成
> system.time(runif(N^2)) # 物理乱数 RNG 経由生成
> finish.RNDMaster(kind.old=kind.old)
```

ex2. 物理乱数 altix4 JOB

ShellScript

```
#!/bin/bash
#PBS -q q1r
#PBS -j oe
#PBS -l ncpus=1
export GOTO_NUM_THREADS=1
source /usr/local1/bin/env_local1.sh
R CMD BATCH --no-save RNDMaster.R
```

CommandLine

```
$ qsub ./RNDMaster.ismaltx.sh
```


ex2. 物理乱数 altix4 結果

```
> N<-8000 # サイズ
> library(RNDMaster) # ライブラリ読込
> system.time(runif(N^2)) # MT 生成速度
[1] 8.0722656 0.3310547 8.4091797 0.0000000 0.0000000
> kind.old<-RNGkind() # MT の保存
> init.RNDMaster() # 物理乱数開始
Initialize Random Master.
> system.time(RNDMaster(N^2))# 物理乱数直接生成
[1] 0.07617188 0.10351562 1.37890625 0.13574219 1.91015625
> system.time(runif(N^2)) # 物理乱数 RNG 経由生成
[1] 13.46680 1.53125 20.31738 0.00000 0.00000
> finish.RNDMaster(kind.old=kind.old)
Finisd Random Master.
[1] 0
>
> proc.time()
[1] 22.6689453 2.5781250 32.1064453 0.1376953 1.9150391
```

ex3. 物理乱数 ismxc R

altix との違いは乱数サーバ IP を指定する

```
> # rxxxx 系の全ての関数が物理乱数を利用します
> N<-8000                                # サイズ
> library(RNDMaster)                    # ライブラリロード
> system.time(runif(N^2))                # MT 生成速度
> kind.old<-RNGkind()                   # MT の保存
> init.RNDMaster("172.22.0.129") ##### 乱数ノード指定
+                                       # 物理乱数開始
> system.time(RNDMaster(N^2))# 物理乱数直接生成
> system.time(runif(N^2))               # 物理乱数 RNG 経由生成
> finish.RNDMaster(kind.old=kind.old)
```

ex3. 物理乱数 ismxc JOB

ShellScript

```
#!/bin/bash
#BSUB -n 1
#BSUB -o RNDMaster.out
#BSUB -e RNDMaster.err
export GOTO_NUM_THREADS=1
R CMD BATCH --no-save RNDMaster.R
```

CommandLine

```
bsub < RNDMaster.ismxc.sh
```

ex3. 物理乱数 ismxc 結果

```
> N<-8000 # サイズ
> library(RNDMaster) # ライブラリ読込
> system.time(runif(N^2)) # MT 生成速度
[1] 5.38 0.34 5.74 0.00 0.00
> kind.old<-RNGkind() # MT の保存
> init.RNDMaster("172.22.0.129") ##### 乱数ノード指定
> # 物理乱数開始
Initialize Random Master.
> system.time(RNDMaster(N^2))# 物理乱数直接生成
[1] 0.03 0.76 3.12 0.00 0.00
> system.time(runif(N^2)) # 物理乱数 RNG 経由生成
[1] 4.09 1.07 15.19 0.00 0.00
> finish.RNDMaster(kind.old=kind.old)
Finisd Random Master.
[1] 0
>
```

ex4. 並列乱数 SPRNG

RScript

```
> require(snow)
> if(""==Sys.getenv("LAMHOME")){
+   cl <- getMPIcluster()
+ } else {
+   cl <- makeCluster(3, type = "MPI")
+ }
> do.call("rbind",clusterCall(cl, runif, 3))
> clusterSetupSPRNG(cl)
> do.call("rbind",clusterCall(cl, runif, 3))
> stopCluster(cl)
```

execute ismxc

```
$ bsub -q q8 -n 5 mpirun -srun \  
> RMPISNOW CMD BATCH --no-save sprngsnow.R
```

ex4. 並列乱数 結果

```
:  
> do.call("rbind",clusterCall(c1, runif, 3))  
      [,1]      [,2]      [,3]  
[1,] 0.5642345 0.9489323 0.8303016  
[2,] 0.5642345 0.9489323 0.8303016  
[3,] 0.5642345 0.9489323 0.8303016  
[4,] 0.5642345 0.9489323 0.8303016  
> clusterSetupSPRNG(c1)  
> do.call("rbind",clusterCall(c1, runif, 3))  
      [,1]      [,2]      [,3]  
[1,] 0.01426654 0.7493919 0.007316102  
[2,] 0.83900320 0.8424790 0.889662451  
[3,] 0.27247416 0.5006002 0.428113404  
[4,] 0.72963698 0.8640611 0.746583761  
> stopCluster(c1)
```

Tips - 自己パッケージの導入 1

Rから

自己パッケージのディレクトリの作成
コンパイラ単位で別けた方が良い

```
$ mkdir ~/.R-2.2.1.gcc
```

自己パッケージも対象となるよう設定

```
$ cat ~/.Rprofile  
.libPaths("~/R-2.2.1.gcc")
```

Rからインストール

リポジトリの設定を行い、インストールする

```
$ R -q  
> options(repos=c(CRAN="http://cran.md.tsukuba.ac.jp"))  
> install.packages("snow", "~/R-2.2.1.gcc")
```

Tips - 自己パッケージの導入 2

コマンドラインから

自己パッケージのディレクトリの作成

コンパイラ単位で別けた方が良い

```
$ mkdir ~/.R-2.2.1.gcc
```

コマンドラインからインストール

.Rconf 又は .Rsrconf の内容を確認する

ここで hoge_0.2-2.tar.gz は scp 等で転送しておく

```
$ R CMD INSTALL --library ~/.R-2.2.1.gcc hoge_0.2-2.tar.gz
```

自己パッケージも対象となるよう設定

```
$ cat ~/.Rprofile  
.libPaths("~/R-2.2.1.gcc")
```


Tips - インストール関連

- パッケージ名とバージョンの一覧取得
`installed.packages()[,c("Version", "LibPath")]`
- パッケージの詳細な情報
`read.dcf(file = system.file("DESCRIPTION", package = "base"))`

ismsr, ismxc には多くのパッケージが導入済みです。

Rビルド環境参考資料



ビルド環境参考資料

ismxc - コンパイラ 1 GNU

command	version	prefix	comment
gcc g++ g77	3.2.3	/usr	アライメント処理 の不具合塵を計算
gcc-3.4 g++-3.4 g77-3.4	3.4.5	/usr/local1/gcc-3.4.5	3.4 系の最終版-1

ismxc - コンパイラ 2 intel

command	version	prefix	comment
icc	8.1.028	/opt/intel/cce/8.1	対象より除外
icpc			
ifort	8.1.023	/opt/intel/fce/8.1	
icc	9.0.021	/opt/intel/cce/9.0	
icpc			
ifort	9.0.021	/opt/intel/fce/9.0	

ismxc - コンパイラ 3 pgi

command	version	prefix	comment
pgic pgicc pgif77	5.1-6	/opt/pgi/pgi5.1-6/linux86-64/5.1	構築不能
pgic pgicc pgif77	5.2-4	/opt/pgi/pgi5.2-4/linux86-64/5.2	構築不能
pgic pgicc pgif77	6.0-8	/opt/pgi/pgi6.0-8/linux86-64/6.0	構築不能

ismxc - コンパイラ 4 pathscale

command	version	prefix	comment
pathcc	2.2.1	/opt/pathscale	lattice でのみエラー
pathCC			adhoc なパッチで対処
pathf95			

ismrsr - コンパイラ xls,f90,GNU

command	version	prefix	comment
xlc			C++が無い I/F が異なる
f90			
gcc-4.0	4.0.3	/usr/local1/gcc-4.0.3	4.0 系の最新版
g++-4.0			
gfortran-4.0			

ismaltx - コンパイラ

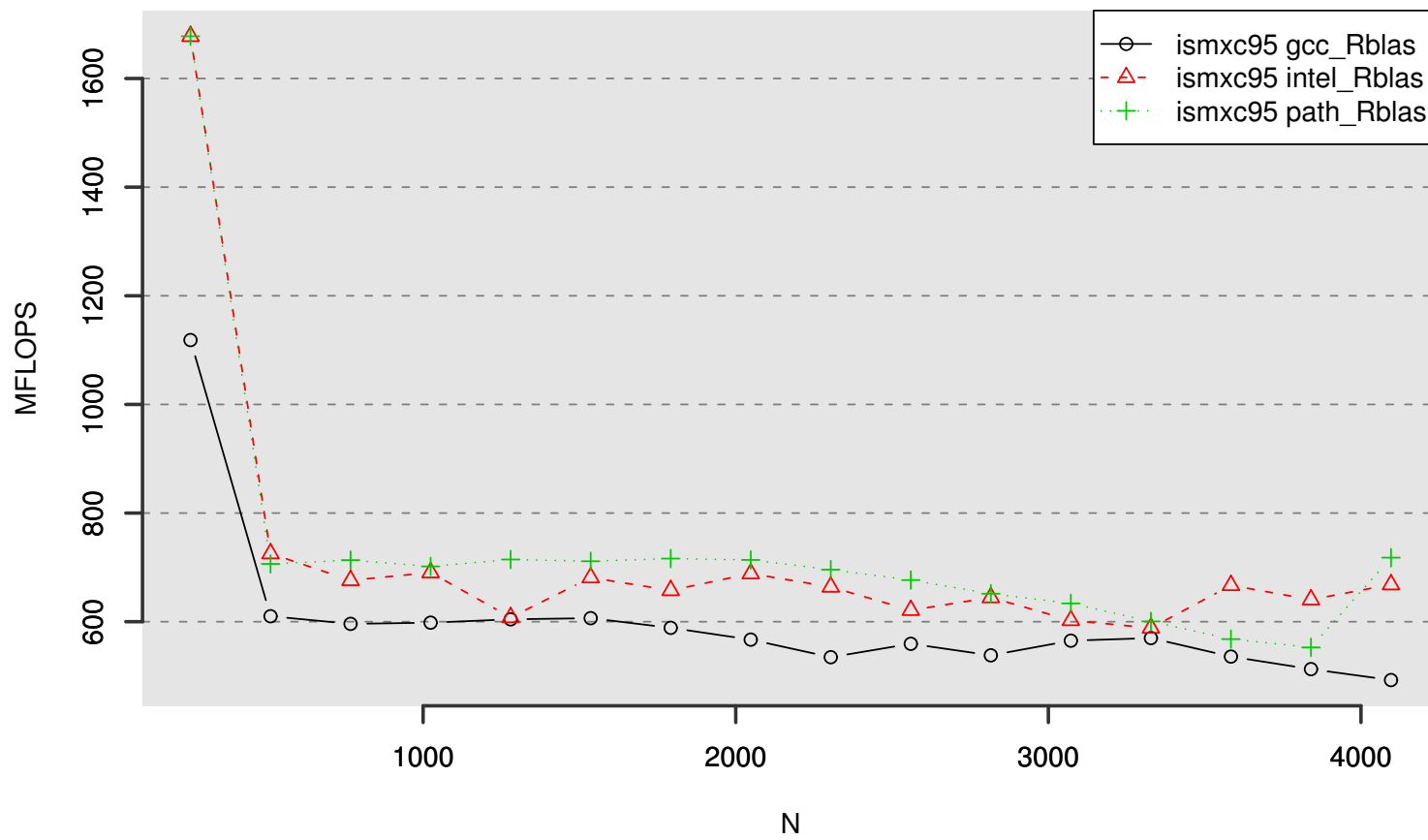
command	version	prefix	comment
gcc g++ g77	3.2.3	/usr	IA64 アライメント処理 の不具合塵を計算
gcc-3.4 g++-3.4 g77-3.4	3.4.6	/usr/local1/gcc-3.4.6	3.4 系の最終版
icc icpc	8.1.026	/opt/intel_cc_80	
ifort	8.1.023	/opt/intel_fc_80	

R 選択参考資料



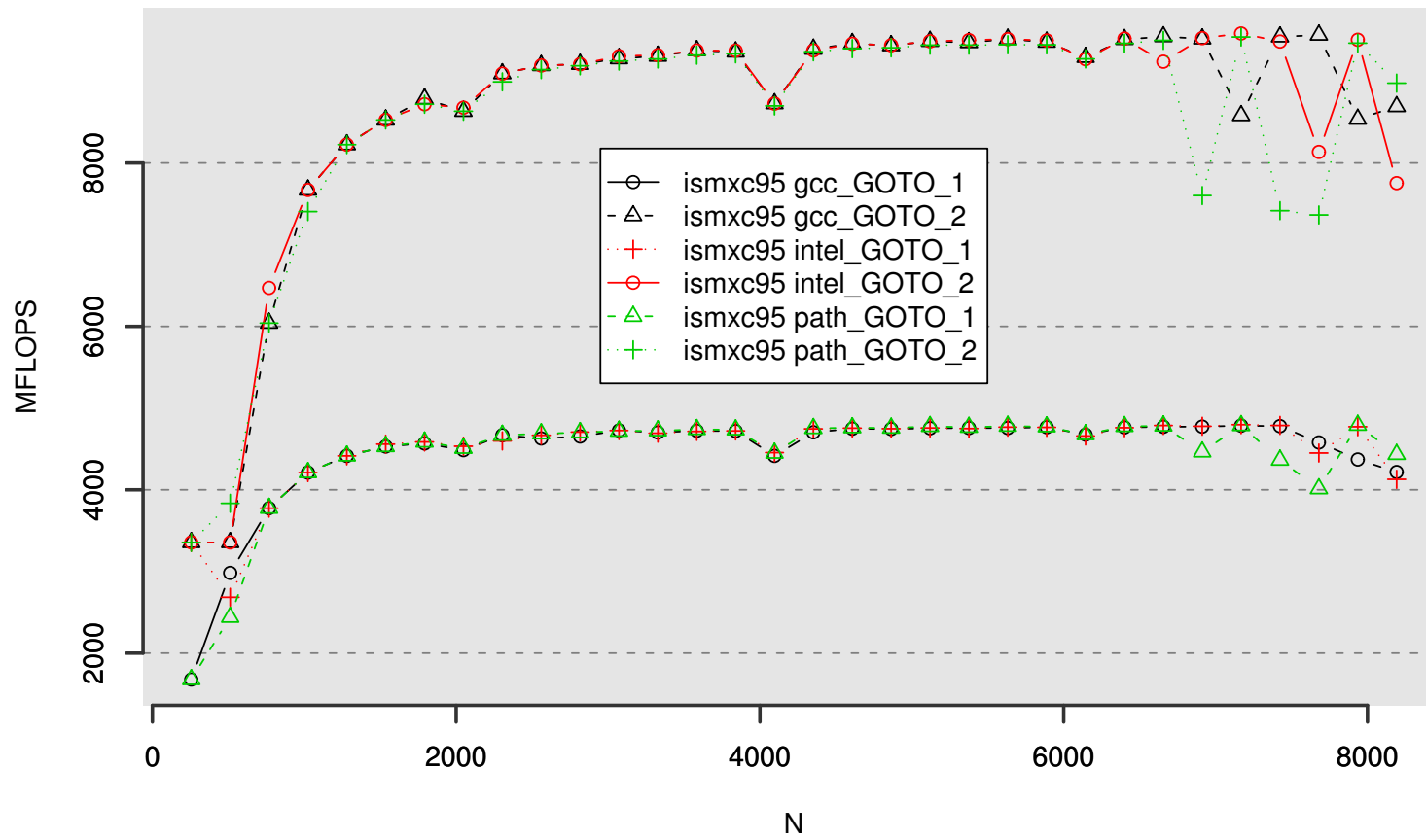
ismxc - R 行列積 RBLAS

R-2.2.1 build for gcc,intel and pathscale compiler on amd64 Linux



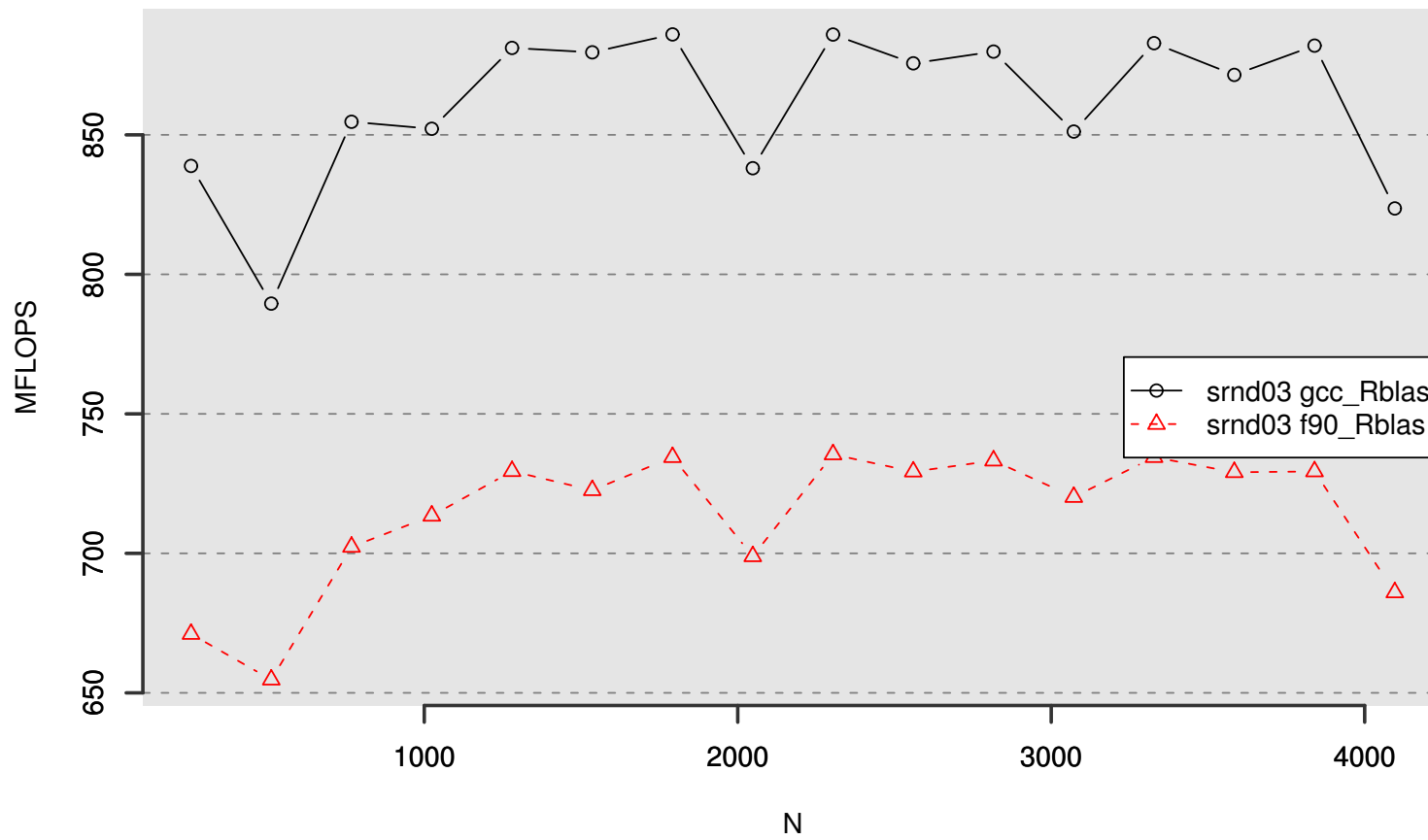
ismxc - R 行列積 GotoBLAS-1.02

R-2.2.1 build for gcc,intel and pathscale compiler + Goto on amd64 Linux



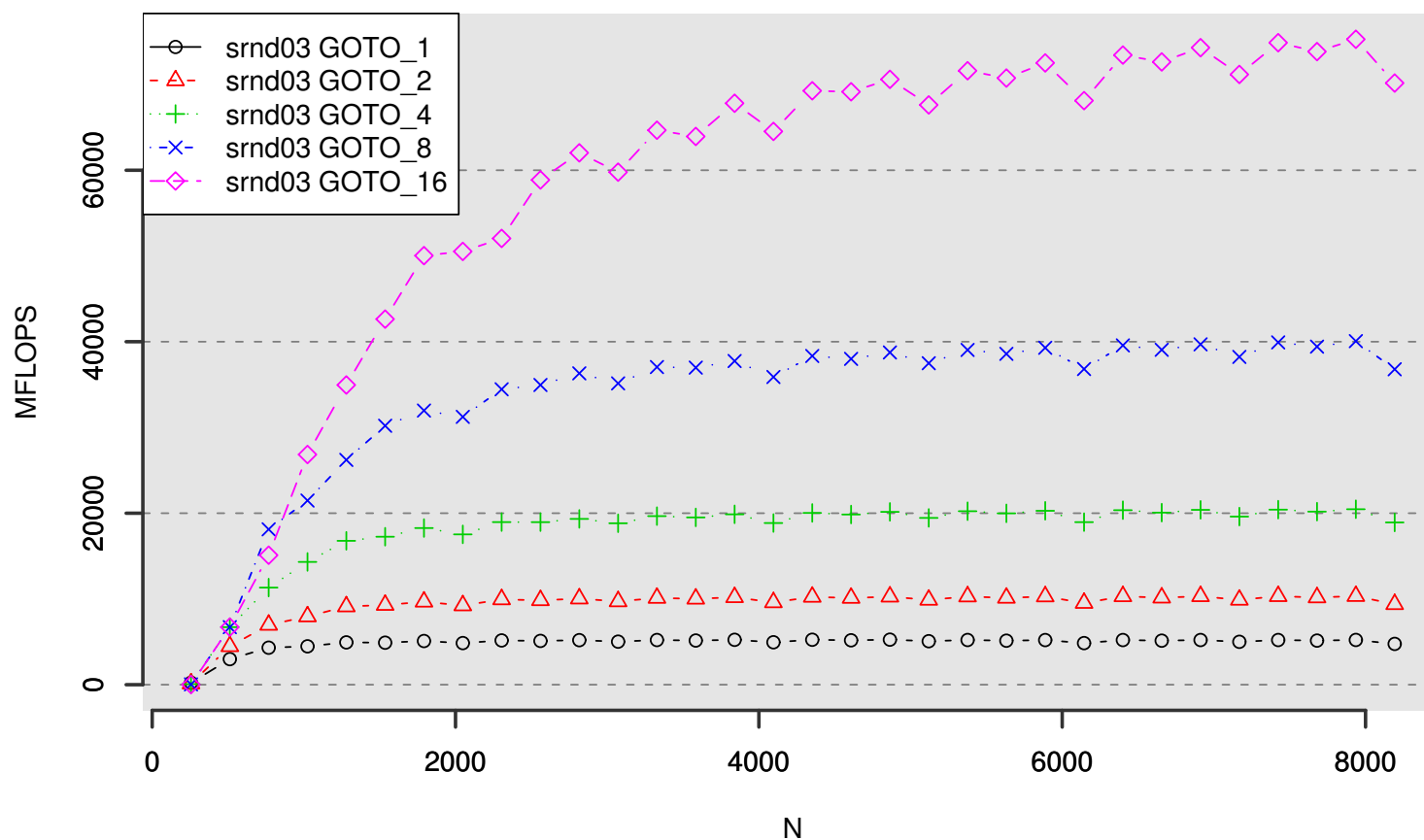
ismsr - R 行列積 RBLAS

GNU Compiler 4.0.3 + R-2.2.1 and xlc + f90 R-2.0.1 on AIX5.2L



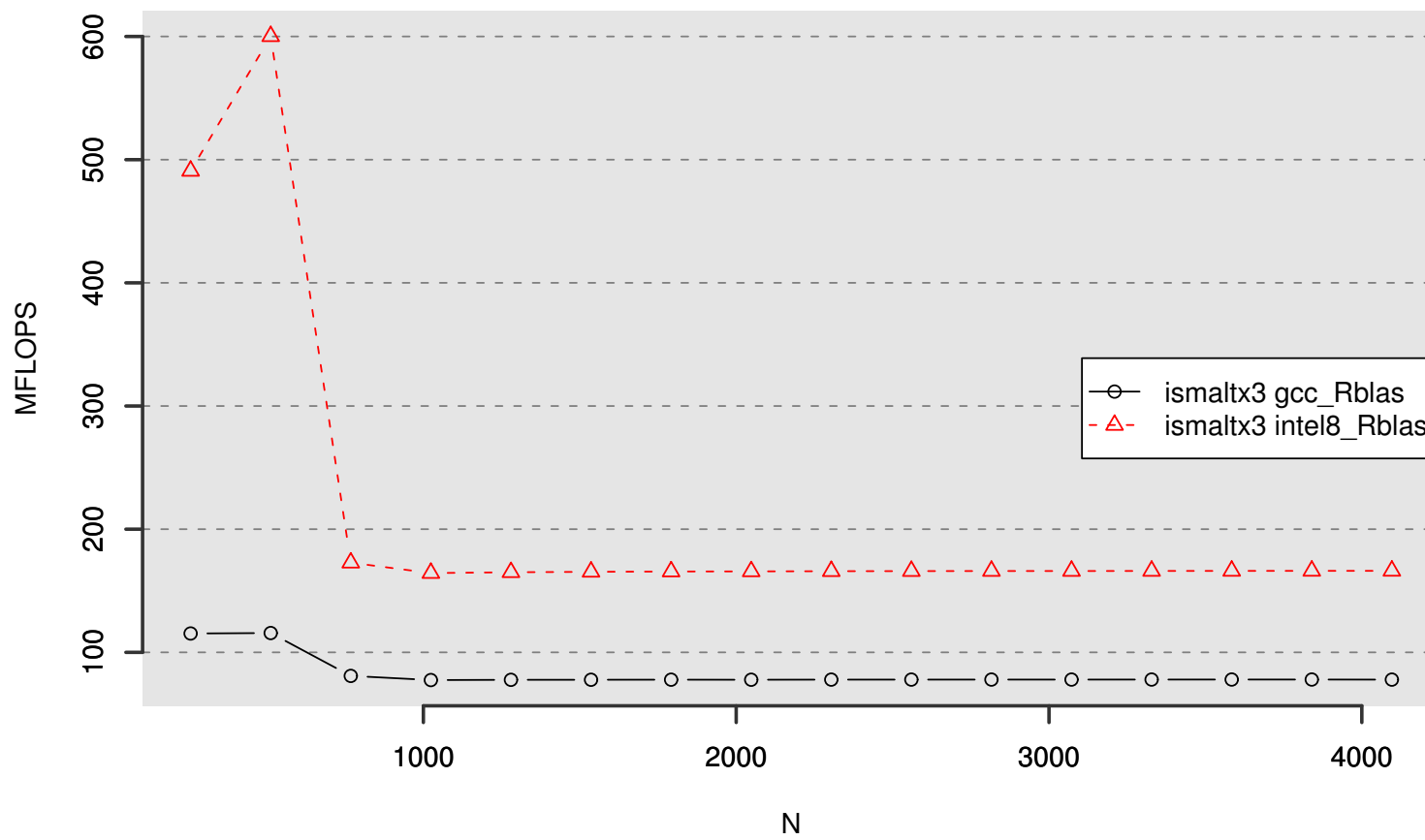
ismrs - R 行列積 GotoBLAS-1.02

R-2.2.1 build for gcc-4.0.3 + GotoBLAS-1.02 on AIX5.2L



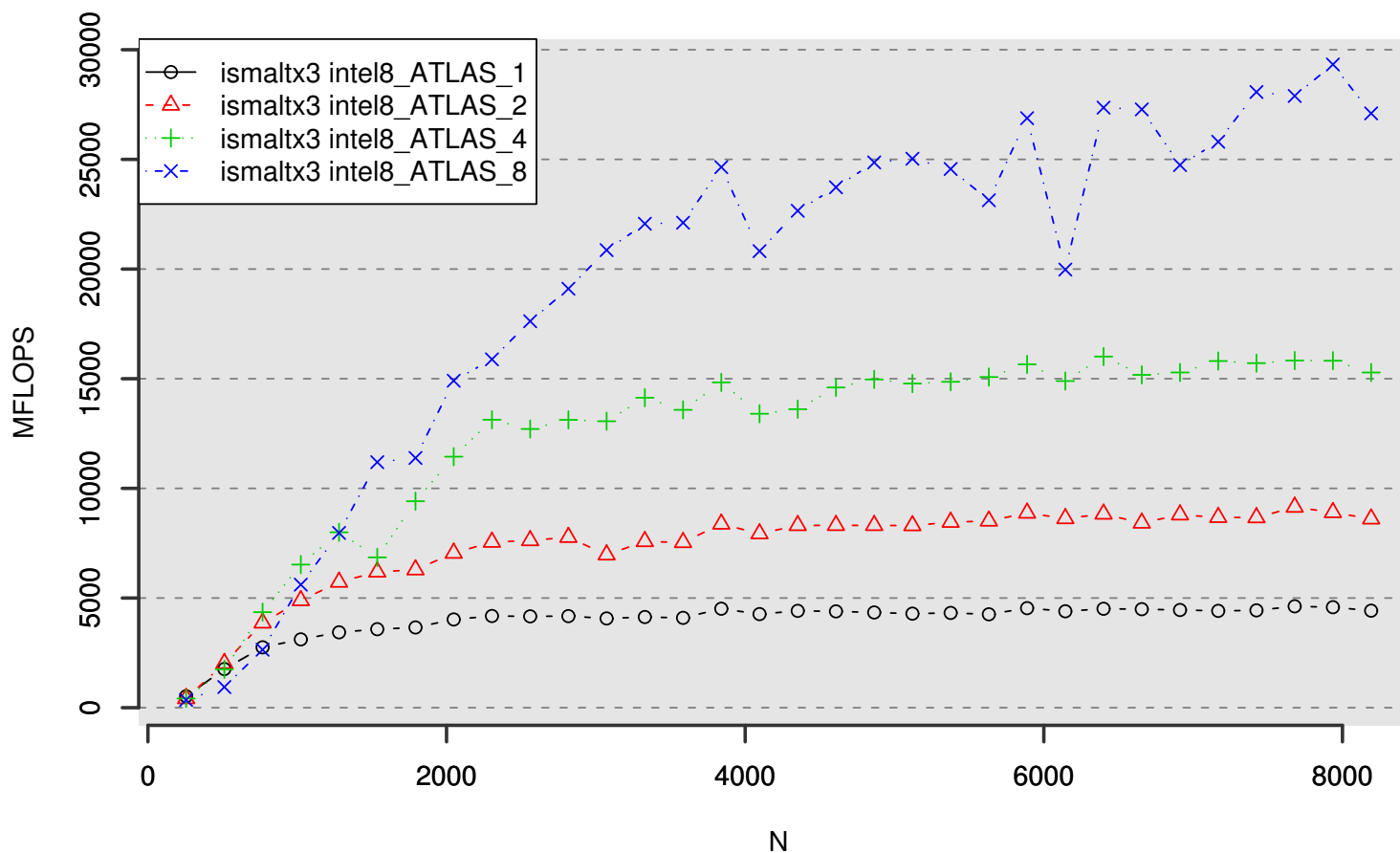
ismaltx - R 行列積 RBLAS

R-2.2.1 build for gcc-3.4.5 and Intel8 Compiler on IA64 Linux



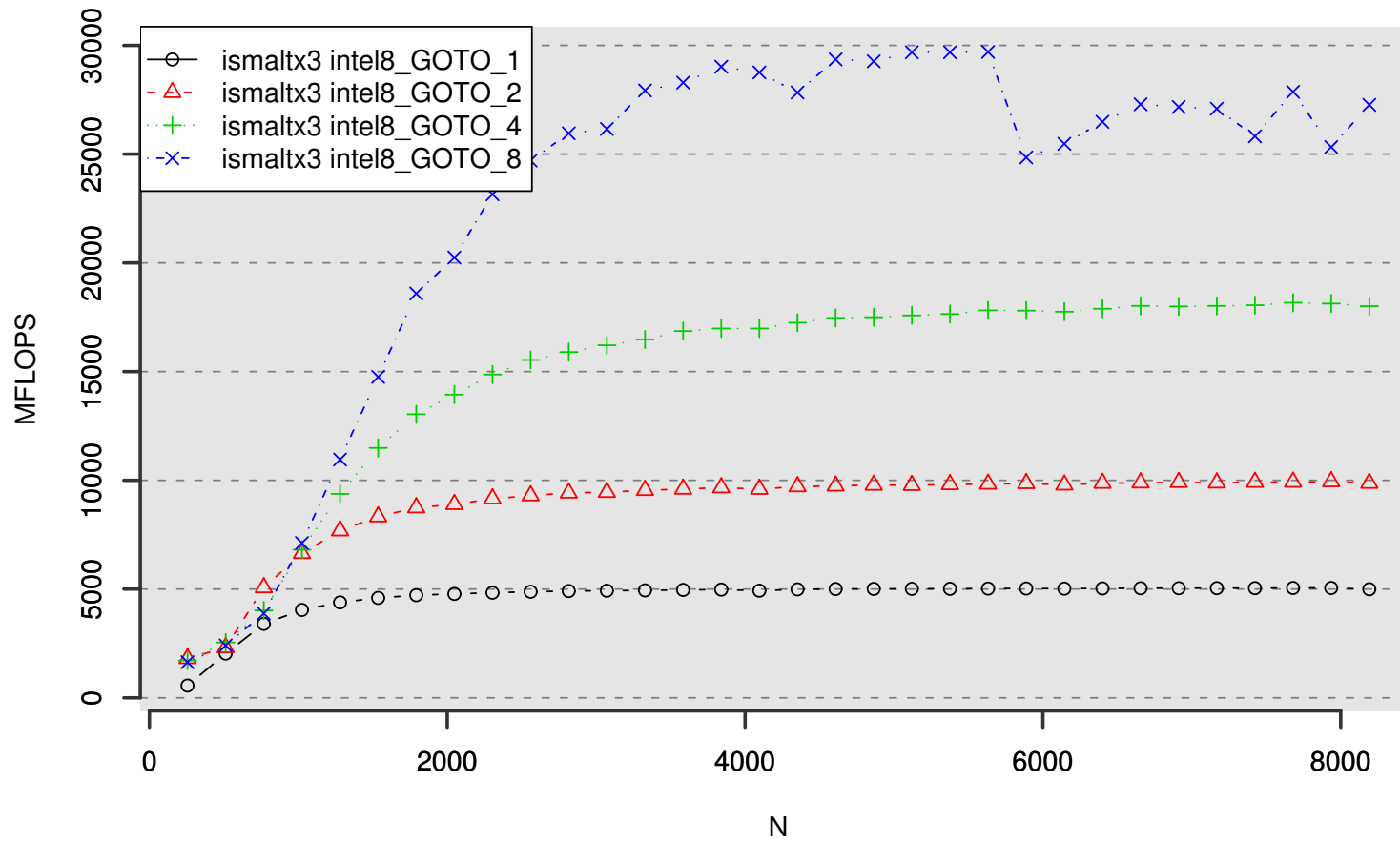
ismaltx - R 行列積 ATLAS-3.6.0

R-2.2.1 build for Intel8 Compiler + ATLAS-3.6.0 on IA64 Linux



ismaltx - R 行列積 GotoBLAS-1.01

R-2.2.1 build for Intel8 Compiler + GotoBLAS-1.01 on IA64 Linux



ism スーパーコンピュータ R FAQ

- 物理乱数は sample 等でも使えますか？
ismxc, ismaltx 上の R の物理乱数モジュールは user_unif_rand を定義していますので、rnorm, sample 等の R の RNG 経由の乱数を扱う全ての関数が物理乱数を利用します。