

# Brief notes on setting up semi-high performance computing environments

October 30, 2013

We have two different computing environments for fitting demanding models to large space and/or time data sets.

- 1 A **distributed system** consists of multiple autonomous computers (nodes) that communicate through a computer network. A computer program that runs in a distributed system is called a distributed program. Message Passing Interface (MPI) is a specification for an Application Programming Interface (API) that allows many computers to communicate with one another (implementations in C, C++, and Fortran.)

We have two different computing environments for fitting demanding models to large space and/or time data sets.

- 1 A **distributed system** consists of multiple autonomous computers (nodes) that communicate through a computer network. A computer program that runs in a distributed system is called a distributed program. Message Passing Interface (MPI) is a specification for an Application Programming Interface (API) that allows many computers to communicate with one another (implementations in C, C++, and Fortran.)
- 2 A **shared memory multiprocessing system** consists of a single computer with memory that may be simultaneously accessed by one or more programs running on multiple central processing units (CPUs). The OpenMP (Open Multi-Processing) is an API that supports shared memory multiprocessing programming (implementations in C, C++, and Fortran).

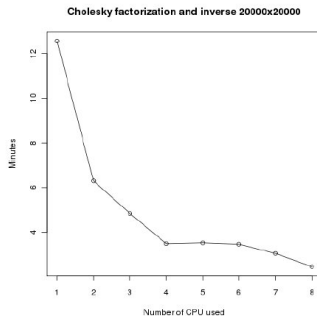
We have two different computing environments for fitting demanding models to large space and/or time data sets.

- Recent work focuses on fitting geostatistical (specifically point-referenced) models using MCMC methods. This necessitates iterative evaluation of a likelihood which requires operations on large matrices.
- A specific hurdle is **factorization** to computing determinant and inverse of large dense covariance matrices.
- We try to model our way out and use tools from computer science to overcome the computational complexity (e.g., covariance tapering, Kaufman et al. 2008; low-rank methods, Cressie and Johannesson 2008; Banerjee et al. 2008, etc.).
- Due to **slow network communication** and transport of submatrices among nodes distributed systems are not ideal for these types of iterative large matrix operations.

- My lab currently favors **shared memory multiprocessing** system.
- We buy rack mounted units (e.g., Sun Fire X4170 Server with 2 quad-core Intel Xeon Processor 5500 Series and 48 GB of RAM  $\sim$ 10-15k) running the Linux operating systems.
- Software includes OpenMP coupled with Intel Math Kernel Library (MKL) <http://software.intel.com/en-us/non-commercial-software-development>. MKL is a library of highly optimized, extensively threaded math routines (e.g., BLAS, LAPACK, ScaLAPACK, Sparse Solvers, Fast Fourier Transforms, and vector RNGs).



So what kind of speed up to expect from threaded BLAS and LAPACK libraries. Mean computing times of dpotrf:



See <http://blue.for.msu.edu/comp-notes> for some simple examples of C++ with MKL and Rmath libraries along with associated Makefile files (I'll add more examples shortly and upon request).

- Many core and contributed packages (including *spBayes*) call Basic Linear Algebra Subprograms (BLAS) and LAPACK (Linear Algebra PACKage) Fortran libraries.
- Substantial computing gains:
  - processor specific threaded BLAS/LAPACK implementation (e.g., MKL or AMD's Core Math Library (ACML))
  - processor specific compilers (e.g., Intel's *icc/ifort*)



## Compiling *R* to call MKL's BLAS and LAPACK libraries (rather than stock serial versions).

```
MKL_LIB_PATH="/opt/intel/composer_xe_2011_sp1.10.319/mkl/lib/intel64"
export LD_LIBRARY_PATH=$MKL_LIB_PATH

MKL="-L${MKL_LIB_PATH} -lmkl_intel_lp64 -lmkl_intel_thread \
      -lmkl_core -liomp5 -lpthread -lm"

./configure --with-blas="$MKL" --with-lapack
```

For many BLAS and LAPACK functions calls from *R*, expect near linear speed up . . .

```

Terminal - andy@darkstar: ... Terminal - andy@darkstar: ~
File Edit View Terminal Go Help

 1 [ .....] 5 [ .....]
 2 [ .....] 6 [ .....]
 3 [ .....] 7 [ .....]
 4 [ .....] 8 [ .....]
Mem [ .....] 896/1594MB Tasks: 102, 142 thr: 1
Swp [ .....] 0/16275MB Load average: 1.56 0.5
Uptime: 01:06:35

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
9765 andy 20 0 790M 270M 17864 R 398. 1.7 1:54.12 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-proc
9770 andy 20 0 790M 270M 17864 R 100. 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-proc
9768 andy 20 0 790M 270M 17864 R 100. 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-proc
9769 andy 20 0 790M 270M 17864 R 99.0 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-proc
1266 root 20 0 168M 21548 10168 S 1.0 0.1 2:26.61 /usr/bin/X :0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswi
9609 andy 20 0 31220 3428 1416 S 0.0 0.0 0:03.50 htop
1959 andy 20 0 1915M 61564 17404 S 0.0 0.4 0:30.10 /home/andy/.dropbox-dist/dropbox
2666 andy 20 0 290M 17712 11472 S 0.0 0.1 0:07.58 /usr/bin/xfce4-terminal
2034 andy 20 0 142M 8756 6968 S 0.0 0.1 0:32.10 /usr/lib/xfce4-systemload-plugin/xfce4/panel-plugins/xfce4-systemload-plugin
1933 andy 20 0 340M 26600 11620 S 0.0 0.2 0:02.14 xfdesktop
1927 andy 20 0 156M 10724 8664 S 0.0 0.1 0:17.44 xfwm4 --replace
1929 andy 20 0 320M 16176 11332 S 0.0 0.1 0:03.04 xfce4-panel
2019 andy 20 0 436M 15104 10892 S 0.0 0.1 0:01.15 /usr/lib/xfce4-notifyd/xfce4-notifyd
9752 andy 20 0 146M 9300 7408 S 0.0 0.1 0:00.04 /usr/lib/xfce4-notifyd/xfce4-notifyd
1984 root 20 0 188M 8856 3016 S 0.0 0.0 0:00.10 /usr/lib/udisks/udisks-daemon
9615 andy 20 0 27540 4584 1728 S 0.0 0.0 0:00.14 bash
1 root 20 0 24472 2392 1368 S 0.0 0.0 0:01.16 /sbin/init
367 root 20 0 17364 640 448 S 0.0 0.0 0:00.05 upstart-udev-bridge --daemon
372 root 20 0 21960 1736 836 S 0.0 0.0 0:00.04 /sbin/udev --daemon
613 root 20 0 21692 1088 388 S 0.0 0.0 0:00.00 /sbin/udev --daemon
614 root 20 0 21956 1268 360 S 0.0 0.0 0:00.00 /sbin/udev --daemon
642 root 20 0 50032 2868 2264 S 0.0 0.0 0:00.00 /usr/sbin/sshd -D
760 syslog 20 0 243M 1656 1128 S 0.0 0.0 0:00.03 rsyslogd -c5
762 syslog 20 0 243M 1656 1128 S 0.0 0.0 0:00.00 rsyslogd -c5
763 syslog 20 0 243M 1656 1128 S 0.0 0.0 0:00.00 rsyslogd -c5
700 syslog 20 0 243M 1656 1128 S 0.0 0.0 0:00.18 rsyslogd -c5

```