

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

October 30, 2013

1

Computing environments

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).

2

LWF, ZWFH, DVFFA, & DR IBS: Biometry workshop

Computing environments

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.

3

LWF, ZWFH, DVFFA, & DR IBS: Biometry workshop

Computing environments

- 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 ~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).

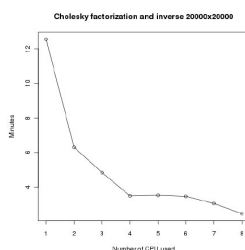


4

LWF, ZWFH, DVFFA, & DR IBS: Biometry workshop

Computing environments

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



5

LWF, ZWFH, DVFFA, & DR IBS: Biometry workshop

Computing environments

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).

6

LWF, ZWFH, DVFFA, & DR IBS: Biometry workshop

- 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/fort*)

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 ...

PID	PPID	USER	CPU	MEM	COMMAND
9765	and	and	20	0	7900 2708 17864 R 398 1.7 154.12 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-prod
9770	and	and	20	0	7900 2708 864 100 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-prod
9768	and	and	20	0	7900 2708 864 100 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-prod
9769	and	and	20	0	7900 2708 864 99.0 1.7 0:24.83 /usr/local/lib/R/bin/exec/R --no-restore --slave --args nextArgpredictive-prod
1266	root	root	20	0	896 1554 0 0 0 1 2:25.61 /usr/bin/X :0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 --novtswitch -ba
9609	and	and	20	0	220 428 416 0 0 0 0:03.50 http
1959	and	and	20	0	1138 1564 7404 5 0 0 0 4 0:30.10 /home/andy/.dropbox-dist/dropbox
2656	and	and	20	0	898 17712 472 5 0 0 0 1 0:07.58 /usr/lib/xfce4/terminal
2034	and	and	20	0	1428 1756 968 5 0 0 0 1 0:32.10 /usr/lib/xfce4/systemload-plugin/xfce4/panel-plugins/xfce4-systemload-plugin
1933	and	and	20	0	1408 26600 620 5 0 0 0 2 0:02.14 xfdesktop
1927	and	and	20	0	1588 1724 664 5 0 0 0 1 0:17.48 xfwm --replace
1929	and	and	20	0	1208 16176 1332 5 0 0 0 1 0:03.04 xfce4-panel
2019	and	and	20	0	1398 15104 8892 5 0 0 0 1 0:01.15 /usr/lib/x86_64-linux-gnu/xfce4/panel-plugins/xfce4-indicator-plugin 5 188744
9352	and	and	20	0	1488 3200 488 5 0 0 0 1 0:00.04 /usr/lib/xfce4/motif/xfce-notifyd
1984	root	root	20	0	1888 3856 016 5 0 0 0 0 0:00.10 /usr/lib/udisks/udisks-daemon
9615	and	and	20	0	540 4584 728 5 0 0 0 0 0:00.14 bash
1	root	root	20	0	472 292 268 5 0 0 0 0 0:01.16 /sbin/init
367	root	root	20	0	364 640 448 5 0 0 0 0 0:00.05 upstart-udev-bridge --daemon
372	root	root	20	0	960 1736 836 5 0 0 0 0 0:00.04 /sbin/udevd --daemon
613	root	root	20	0	692 1088 388 5 0 0 0 0 0:00.00 /sbin/udevd --daemon
614	root	root	20	0	956 268 360 5 0 0 0 0 0:00.00 /sbin/udevd --daemon
642	root	root	20	0	932 868 254 5 0 0 0 0 0:00.00 /usr/sbin/sshd -D
760	syslog	syslog	20	0	1488 656 128 5 0 0 0 0 0:00.03 rsyslogd -cS
762	syslog	syslog	20	0	1438 656 128 5 0 0 0 0 0:00.00 rsyslogd -cS
763	syslog	syslog	20	0	1488 656 128 5 0 0 0 0 0:00.00 rsyslogd -cS
700	syslog	syslog	20	0	1438 656 128 5 0 0 0 0 0:00.18 rsyslogd -cS