Comparison of Crystallography Program Execution Times (sec) on UNIX/Linux Computers Available to Johnson Lab[1]

	SGI Altix Cluster 1.3Ghz Itanium 2	Linux Cluster 3.4GHz Xeon	Linux Cluster 2.4Ghz Xeon	Apple PowerMac Dual 2.5Ghz G5--XLF	Apple PowerMac Dual 2.5Ghz G5	Apple Xserve Dual 2Ghz G5 -- XLF	Apple Xserve Dual 2Ghz G5	Apple PowerMac Dual 2Ghz G5--XLF	Apple PowerMac Dual 2Ghz G5	Apple PowerMac Dual 1.8Ghz G5	Apple PowerMac 1.8Ghz G5	Apple PowerBook 1.5Ghz G4
GLRF[2]	297	**249**	564	288	367	320	414	325	408	449	517	535
i2f #1	43	28	N.P.	**20**	35	**24**	49	**25**	50	49	49	96
i2f #2	35	24	N.P.	**14**	24	**17**	34	**17**	33	32	33	77
CCP4 #1	632	**520**	935	597	N.P.	N.P.	772	677	786	878	965	1176
CCP4 #2	71	**57**	116	69	N.P.	N.P.	88	74	89	96	106	140
CCP4 #3	**31**	**26**	48	**26**	N.P.	N.P.	34	**28**	34	36	39	60
CNS	208	238	374	**153**	274	N.P.	332	203	337	369	403	555
CNS #2	839	900	1690	**653**	1330	N.P.	1588	806	1614	1726	1862	2291
RAVE/ CCP4	22016[3] (18.3 min/cycl)	**5891** (4.9 min/cycl)	8515 (7.1 min/cycl)	6834[4] (5.7 min/cycl)	N.P.	N.P.	8315 (6.9 min/cycl)	7932[4] (6.6 min/cycl)	8350 (7.0 min/cycl)	9054 (7.6 min/cycl)	9673 (8.0 min/cycl)	11643 (9.7 min/cycl)

[1] Shortest time (multiple if within 5 sec of each other) is highlighted in grey. N.P. - not performed. XLF designates the programs were compiled with the IBM XLF v8.1 Fortran compiler. Reporting only user seconds from UNIX 'time' utility (not sure if this is the best way, but seemed consistent). No 'walltime' or system time included. OSX does not have a native Fortran compiler. GNU g77 versions 3.3 to 3.4 (freeware) was installed via Fink for compiling Fortran code, and the IBM XLF v8.1 compiler (commercial) demo was used as installed on the Xserve G5. No code porting or compiling to take advantage of the G4 or G5 vector processing unit except for CCP4, which uses Apple's optimized Lapack and Blas libraries. CCP4 was compiled on the Linux and Altix clusters with their native compilers. The XLF version of CCP4 was downloaded from the CCP4 web site (v5.0.2). No porting or compiling to take advantage of Apple G5 64-bit processor architecture (the G5 is running programs compiled for the G4) except for perhaps the XLF created executables. GLRF and i2f were compiled with g77 and XLF using high optimization on the Apple machines, with g77 on the 2.4GHz Linux cluster using high optimization, with Intel v7 efc using high optimization on the SGI Altix cluster, and with Intel v8 ifc using high optimization on the 3.4GHz Linux cluster. CNS was compiled on the Apple's with g77 via Fink (various optimizations), and on the other platforms with their native compilers. The XLF compiled CNS for Apple machines was downloaded from http://www.sbgrid.org/osx.php, and is "Modified to support multithreading and fast FFT calculations on G4/G5 systems. Select functions have been pipelined for the G5. G4/G5 and G5 only builds. G5 only version takes advantage of additional G5 hardware features" – the results certainly show it! No knowledge of how RAVE programs were compiled since they were downloaded as executables. All results on the Apple computers were compared with those from the other systems to verify the programs had run correctly and accurately. All tasks given as near 100% processor time as possible (i.e. benchmarks were run when machines were otherwise idle, full processor allocated on servers). No benchmark times were accepted unless 'time' reported at least 98% processor utilization over the course of the task. The XLF compiled CNS jobs utilized up to 140% of processor time on dual processor machines, but identical speed gains were seen on a single processor G5 machine (not shown). All applications and data were accessed from a local disk on the system.

[2] Apple benchmarks for GLRF updated to reflect significant gain in execution times when compiled with g77 v3.4 (via Fink) and PowerPC options. All boxes running OS X 10.3.4. The other Apple benchmarks had various OS X and g77 revisions at time of benchmarking.

[3] Program lx_ave of the RAVE package is not running well on the SGI Altix. It is 3x slower than on the Linux cluster or G5. It is only available as a generic binary download for all linux systems, so it probably needs to be compiled specifically for the Altix. Typically the 'ave' portion of each cycle takes the most time to complete.

[4] Program osx_ave of the RAVE package may not have been compiled using XLF for OS X. It is only available as a binary download. Typically the 'ave' portion of each cycle takes the most time to complete.

Task details:

N.P.: Test not performed.

GLRF (v3.5): Providence Virus 5-fold search from 10-5Å stepping 2 degrees over entire hemisphere. Note: the time results from this test can vary quite a bit (up to 5-10%). The algorithm modifies the search grid size on the fly (RF cut off) and can change between runs. Time given is the shortest time observed in a run with an RF cut off value of 0.3-0.5.

i2f #1 (v2.6.1): (Jeff's program) Read and analyze 1.25 million Scalepack format reflections split into 500 equal reciprocal space volume bins, write new reflection file in CNS format.

i2f #2 (v2.6.1): Read and analyze 1.25 million Scalepack format reflections split into 500 equal reciprocal space volume bins, no reflection file output. Shows relative contribution of hard disk performance by comparison to test i2f #1.

CCP4 #1 (v4.2.2): Compute Fcalc for half Prv particle using SFALL, then scale with RSTATS. Version 4.2 for 1 GHz Powerbook, version 4.1 for others. No updates to SFALL or RSTATS reported in v4.2. XLF compiled version is 5.0.2, but no updates reported to SFALL or RSTATS.

CCP4 #2 (v4.2.2): FFTBIG map generation using Fo, Acalc for Prv from 33-3.8Å using grid size (663, 440, 420). Produces 500Mb (Fo, Acalc) map. Took 470Mb of RAM. XLF compiled version is 5.0.2, but no updates reported for FFTBIG.

CCP4 #3 (v4.2.2): FFTBIG map generation using Fo, Acalc for Prv from 33-3.8Å using grid size (520, 342, 330). Produces 235Mb (Fo, Acalc) map. Took 250Mb of RAM. XLF compiled version is 5.0.2, but no updates reported for FFTBIG.

CNS (v1.1): 30 cycles of Providence Virus rigid body refinement from 15-8Å resolution using 30-fold NCS. CNS is known to have diminished performance when compiled with g77.

CNS #2 (v1.1): Christoph Weber's standard benchmark run, which consists of slowcooled torsion molecular dynamics starting at 500K and stepping 100K, and 30 cycles of gradient minimization.

RAVE (v010122/5.0)/CCP4 (v4.2.2): Twenty cycles of Providence Virus 30-fold averaging from 33-3.8Å in a P1 unit cell (663, 440, 420). Allocated 1Gb of RAM on the SGI and Linux clusters.

Computer details:

Apple PowerMac G5 Dual 2.5GHz: Two 2.5Ghz PowerPC G5 processors and 4.5Gb of RAM. Tested with g77 v3.4 and OSX 10.3.5-10.3.9.

Apple Xserve G5 Dual 2GHz: Two 2.0Ghz PowerPC G5 processors and 3Gb of RAM. Tested with g77 v3.4 and OSX 10.3.4.

Apple PowerMac G5 Dual 2GHz (rev 1): Two 2.0Ghz PowerPC G5 processors and 4Gb of RAM. Tested with g77 v3.4 and OSX 10.3.2-10.3.9.

Apple PowerMac G5 Dual 1.8GHz (rev 2): Two 1.8Ghz PowerPC G5 processors and 1.25Gb of RAM. Tested with g77 v3.4 and OSX 10.3.4-10.3.9.

Apple PowerMac G5 1.8GHz (rev 1): Single 1.8Ghz PowerPC G5 processor and 1Gb of RAM. Tested with g77 v3.4 and OSX 10.2.8-10.3.9.

Apple PowerBook 1.5GHz G4 (Al): 17 inch Aluminum PowerBook with single 1.5Ghz PowerPC G4 processor and 1.5Gb of RAM. Tested with g77 v3.3 and OSX 10.3.3-10.3.9.

Linux Cluster (2.4Ghz): Linux 32-bit server (RedHat) with 256 Intel Xeon processors, 64Gb of RAM, and 5Tb RAID.

Linux Cluster (3.4GHz): Demo Linux 36-bit (?) server (RedHat) with 4 Intel Xeon processors, 8Gb of RAM, and 73Gb disk.

SGI Altix 3700 Supercluster: Linux 64-bit server (RedHat) with 128 1.3Ghz Intel Itanium-2 processors, 128Gb of RAM, and 1Tb RAID.

Acknowledgements:

- Professor William G. Scott for sharing all his tremendous work on installing crystallography programs under OS X. He has put this great wealth of knowledge on his well organized web site for all to use: http://chemistry.ucsc.edu/~wgscott/xtal/.

- The Fink group for providing an invaluable resource for organizing and automatically installing programs ported to the Darwin environment. They make it work far easier than it looks. You'll find them at: http://fink.sourceforge.net/index.php.

- The SBGrid group for sharing an amazingly fast CNS compile for G4/G5 Macs, and especially Dave Gohara for providing binaries compiled with increased parameter sizes for virus crystallography work. You can see more of their work at: http://www.sbgrid.org/index.php.

- Christoph Weber, Brant Kelley, Bill Young, Matt Barnas, David Braun, David Laing and Kim Renner of The Scripps Research Institute's Research Computing department (Michael Denisevich, Director) for their interest in OS X as a scientific computing platform. They provide the Linux servers, and back them up with great technical and user support.