Numerical Performance and Throughput Benchmark for Electronic Structure Calculations in PC−Linux Systems with New Architectures, Updated Compilers, and Libraries

Jen-Shiang K. Yu,*[,†,‡] Jenn-Kang Hwang,[†] Chuan Yi Tang,[‡] and Chin-Hui Yu[§]

Department of Biological Science and Technology, National Chiao Tung University, No. 75, Po-Ai Street, Hsinchu 300, Taiwan, and Departments of Computer Science and Chemistry, National Tsing Hua University, No. 101, Kuang-Fu Road Sec. 2, Hsinchu 300, Taiwan

Received September 18, 2003

A number of recently released numerical libraries including Automatically Tuned Linear Algebra Subroutines (ATLAS) library, Intel Math Kernel Library (MKL), GOTO numerical library, and AMD Core Math Library (ACML) for AMD Opteron processors, are linked against the executables of the Gaussian 98 electronic structure calculation package, which is compiled by updated versions of Fortran compilers such as Intel Fortran compiler (**ifc**/**efc**) 7.1 and PGI Fortran compiler (**pgf77**/**pgf90**) 5.0. The **ifc** 7.1 delivers about 3% of improvement on 32-bit machines compared to the former version 6.0. Performance improved from **pgf77** 3.3 to 5.0 is also around 3% when utilizing the original unmodified optimization options of the compiler enclosed in the software. Nevertheless, if extensive compiler tuning options are used, the speed can be further accelerated to about 25%. The performances of these fully optimized numerical libraries are similar. The double-precision floating-point (FP) instruction sets (SSE2) are also functional on AMD Opteron processors operated in 32-bit compilation, and Intel Fortran compiler has performed better optimization. Hardware-level tuning is able to improve memory bandwidth by adjusting the DRAM timing, and the efficiency in the CL2 mode is further accelerated by 2.6% compared to that of the CL2.5 mode. The FP throughput is measured by simultaneous execution of two identical copies of each of the test jobs. Resultant performance impact suggests that IA64 and AMD64 architectures are able to fulfill significantly higher throughput than the IA32, which is consistent with the SpecFPrate2000 benchmarks.

INTRODUCTION

It has been proved that extensively tuned numerical libraries are able to enhance the performance of number-crunching codes in complicated matrix operations.[1] The Automatically Tuned Linear Algebra Subroutines[2] (ATLAS library) has become one of the most frequently used libraries by scientific researchers to support a variety of numerical applications. The Intel Math Kernel Library (MKL) version 6.0 has been released formally, and the combination of 64-bit MKL working with the 64-bit Intel Fortran compiler (**efc**) is currently the most recommended numerical solution to the IA64 architecture.[3] Alternatives to the two aforementioned optimized mathematical libraries are available. The high-performance Basic Linear Algebra Subprograms (BLAS) library developed by Kazushige Goto[4] (the GOTO library), has been extensively tuned and is based on assembly kernels. Using this library, the Multiprogrammatic Capability Cluster[5] (MCR), which is composed of 1152 Linux nodes (2304 CPUs of 2.4 GHz Pentium4/Xeon), in the Lawrence Livermore National Laboratory has been able to achieve 7.634 TeraFLOPs running the HPC LINPACK[6] suite, and ranked third in the TOP500 supercomputer list in June 2003.[7] The GOTO library features complete BLAS level 1, 2, and 3 subroutines for most PC processors, including both 32-bit and 64-bit implementations for Intel and AMD processors running Linux operating system. This library is available for free downloading; moreover, a subroutine **xerbla.f** is provided, which allows users to compile their own target programs with different Fortran compilers without recompiling this numerical library.

Designed for its own 64-bit Opteron processors, AMD has released the AMD Core Math Library[8] (ACML), which incorporates fully tuned BLAS level 1 to level 3 kernels, as well as a completely implemented Linear Algebra PACKage (LAPACK), plus a comprehensive suite of Fast Fourier Transforms (FFTs) in single-, double-, single-complex, and double-complex data types. The ACML supports both 32-bit and 64-bit applications, in conjunction with the PGI as well as other GNU-compatible compilers. Different from ATLAS, GOTO and ACML are available in the form of precompiled shared libraries, which spare their users recompilation of applications when newer revisions of the libraries are available.

The Fortran language is indispensable in numerical applications and therefore advances in Fortran compilers always receive distinguished attention. In addition to newly incorporated functions, Fortran compilers aim at optimization for specific hardware. Fro example. in the SpecFP2000 benchmark[9] list, the floating-point (FP) performance of Intel IA32 architecture has been enhanced by 2% with the revisions of Fortran compiler using identical compiling options, whereas

* Corresponding author fax: +886-3-5729288; e-mail: jsyu@mail.nctu.edu.tw.
† National Chiao Tung University.
‡ Department of Computer Science, National Tsing Hua University.
§ Department of Chemistry, National Tsing Hua University.

10.1021/ci034210h CCC: $27.50
Published on Web 02/07/2004

under the same condition, the performance advantage for IA64 has been 5%. The SpecFP2000 benchmark suite has shown good correlation with real applications in quantum chemistry, which indicates possibilities of further acceleration by updated compiler versions.

There is potential for hardware-level tuning to improve computational efficiency, especially for setup options inside the BIOS menu. One of the various adjustments is the timing of the random access memory (RAM), which tends to affect applications demanding high memory bandwidth. Technical reviewers[10] have addressed that proper setting of Column Address Strobe (CAS) latency of the RAM modules can deliver higher memory transfer rates for synthetic benchmark suites. However, a similar effect for scientific packages remains to be quantified.

To evaluate the computational performance, various measurements exist. One of them is to reckon the expedition at which a processor completes one single task, which is a speed measurement. Such evaluations for electronic structure computation have been continually reported in recent years.[1,11,12] Another strategy is to gauge how many tasks one machine can accomplish within a certain period of time. The latter is the rate measurement, or more terminologically, the throughput.[13] Traditionally this vocabulary represents the comparative effectiveness of computers which execute multiple programs concurrently. To computational scientists the efficiency of running multiple calculations in a multi-user environment that apportions the resources is of more interest.

We present in detail the advances among numerical libraries and Fortran compiler revisions, as well as the hardware-level tuning for the electronic structure calculations. To be consistent with and comparable to former reports,[1] Gaussian 98 A11.3[14] remained the target for benchmark although a new major release is available. The detailed description of each test file has been addressed in Table 3 in ref 1. Analysis of the FP performance was quantified by correlation with the SpecFP2000 benchmark. The throughput was measured by means of the performance multiplication while the test jobs were concurrently performed in duplicate under dual-processor systems.

SYSTEM DETAIL

The following hardware architectures were utilized to run the benchmark: three motherboards loaded with Intel E7501 and E7505 chipsets and Xeon processors (533 MHz Front-side bus), a Hewlett-Packard zx6000 workstation (HP zx1 chipset with 900 MHz 64-bit Itanium-2 CPU, the IA64 architecture) as well as AMD Opteron 244 (AMD 8131 chipset with 1800 MHz 64-bit processor, the ×86_64 or AMD64 architecture). All of these machines were equipped with dual processors. An IBM P690 workstation consisting of 32 POWER4 CPUs of 1.3 GHz was additionally benchmarked. Test jobs described in our previous report[1] were used to provide meaningful comparison among the performances. Unmodified G98 source code fails to handle a single scratch file sized over 2 GB in a 32-bit Linux system, and therefore the scratch file in the second part of test415 was split into files less than the file-size limitation. In addition, Intel Pentium III-based and AMD Athlon family CPUs lack the double-precision FP instruction sets of the second generation Streaming SIMD (single-instruction, multiple data) Extension (known as SSE2) that have been proved to be beneficial to quantum mechanical calculations, especially in post-Hartree−Fock computations,[1] and hence benchmarking on these processors without SSE2 is ignored.

Benchmarked compilers consisted of Intel Fortran compiler version 7.1 (build 20030307Z) and PGI Fortran compiler version 5.0. The two compilers were tested on both 32-bit and 64-bit systems. The numerical libraries used included ATLAS 3.4.1 and ATLAS 3.5.7 (only for AMD Opteron), Intel Math Kernel Library version 6.0 (for both IA32 and IA64), and the GOTO library. These libraries were linked to generate the executables. The GOTO library has several variations: multi-threaded **libgoto_p4_512p−r0.6.so** was chosen for systems with E7501 and E7505 chipsets and dual Xeon processors with 512 kB L2 cache, whereas multi-threaded **libgoto_it2p−r0.7.so** was installed in the HP zx6000 system. At the time of submission of this paper, the multi-threaded GOTO library for 32-bit compilation with AMD Opteron processors was not yet released, and therefore was not included in the test. The hardware and software configurations of the benchmarked systems are listed in detail in Table 1. For the Iwill DP533 system (E7505), it is possible to set the DRAM CAS latency (CL) to CL2.5 or CL2, and both of the settings were tested.[15] The Linux kernels used with these platforms were of 2.4.19 or 2.4.20 depending on the architecture.

The single CPU performance was the focus of the first part of the benchmark, and therefore all of the computations have been performed with one processor (nonparallel) on all of the platforms with multiple processors installed. In the second part of the evaluation, the stress was on multitasking performance. Several configurations in the E7505, K8-32, and zx6000 systems were chosen to experiment the throughputs. Two identical copies of each test job were executed simultaneously in these three systems in order to observe the speed impact by multitasking. The outcomes have been then analyzed in correlation to the SpecFPrate2000 benchmark.

Nevertheless, it is necessary to stress again that utilization of nondefault combination of compilers and numerical libraries is not supported by the software vendor, and the results require careful examination. Furthermore, out-of-specification settings over unqualified hardware components may lead to system instability as well as damage when the machines are manually tuned. These configurations are discouraged for those without technical experience.

BENCHMARK SETUP

Activation of the SSE2 double-precision FP optimization by the new version 5.0 of **pgf77** compiler has been attempted; the compiling switches of "-fastsse -time -Mreentrant -Mrecursive -Mnosave -Minfo -Mneginfo -Mscalarsse -Mvect=assoc,recog,prefetch,sse,cachesize:**L2-cache-size**"[16] have been employed in addition to the default "-tp p7 -O2 -Munroll" options in the Intel Xeon systems. Generation of 32-bit binaries in the AMD64 system using PGI compilers needs several special compiling and linking options because the default system setting is to produce 64-bit executables. The architectural tuning option of PGI compiler should be set to "-tp k8-32", whereas the option of "-m32" is required to specify 32-bit compilation with GNU compilers. Further-

Table 1. Detailed Hardware Specifications and Software Configurations of the Tested Platforms

	Alpha500	E7505	E7501-S	E7501-T	K8-32	zx6000	IBM P690
CPU type	Alpha 21264A	Intel P4 Xeon[a]	Intel P4 Xeon[a]	Intel P4 Xeon[a]	AMD Opteron 244	Intel Itanium2	IBM RS6000
architecture	21264A	IA32	IA32	IA32	AMD64	IA64	POWER4
L1 cache size	64kB(I)+ 64kB(D)	12kμ−ops(I)+ 8kB(D)	12kμ−ops(I)+ 8kB(D)	12kμ−ops(I)+ 8kB(D)	64kB(I)+ 64kB(D)	16kB(I)+ 16kB(D)	64kB(I)+ 32kB(D)
L2/L3 cache size	4MB/0	512kB/0	512kB/0	512kB/0	1MB/0	256kB/ 1.5MB	1440 kB/ 128MB
CPU frequency	500 MHz	2.4 GHz	2.4 GHz	2.4 GHz	1.8 GHz	900 MHz	1.3 GHz
installed CPUs	2	2	2	2	2	2	32
motherboard	API DP264	Iwill DP533	Supermicro x5DPE-G2	Tyan S2723GNN	Rioworks R3140	HP zx6000	
M/B chipset	Compaq 21272	Intel E7505	Intel E7501	Intel E7501	AMD 8131/8111	HP zx1	
RAM size[b]	2048 MB 128 MB× 16	2048 MB 512 MB× 4	2048 MB 512 MB× 4	2048 MB 512 MB× 4	8192 MB 1024 MB× 8	6144 MB 512 MB× 12	128 GB
RAM type	83 MHz SDRAM 256bit bus, 72bit ECC	DDR266 ECC Registered CL2	DDR266 ECC Registered CL2.5	DDR266 ECC Registered CL2.5	DDR333 ECC Registered	DDR266 ECC Registered	
theoretical bandwidth	2650 MB/s	4.3 GB/s	4.3 GB/s	4.3 GB/s	5.3 GB/s per processor	8.5 GB/s	12.8 GB/s
OS version[c]	Tru64 v4.0F	Linux 2.4.20 (S)	Linux 2.4.20 (R9)	Linux 2.4.20(S)	Linux 2.4.19 (U)	Linux 2.4.20 (R2)	AIX 5L V5.1
C compiler	DEC C V5.9−010	gcc 2.95.3	gcc 3.2.2	gcc 2.95.3	gcc 3.2.2	Intel v7.1	xlc v8.0
Fortran compiler[d]	Digital Fortran v5.2	PGI/IFC	PGI/IFC	PGI/IFC	PGI/IFC	EFC	xlf90 v8.0
math library[e]	CXML(DXML)	AT/M/G	AT/M/G	AT/M/G	AT/M/AC	AT/M/G	ESSL 3.3
SpecFP2000[f]	383/422	934/944 (W)			1093/1168[g](L)	1139/1139 (L)	1202/1266

[a] All of the Intel P4/Xeon CPUs are socket 604 and operating at 533 MHz Front Side Bus (FSB). [b] All banks of memory modules of every tested platform are filled. [c] Linux (R9), RedHat 9.0; (S)m Slackware 8.1; (U)m SuSE SLES 8; (R2)m RedHat Advanced Workstation 2.1. [d] PGI, Portland Group Inc. Fortran Compiler v5.0-1 for AMD64 and v5.0-2 for IA32; IFC, Intel Fortran Compiler v7.1, Build 20030307Z for IA32; EFC, Intel Fortran Compiler v7.1, Build 20030307Z for IA64. [e] AT, ATLAS 3.4.1/3.5.7; M, Intel MKL 6.0; G, GOTO library 0.6p for P4/Xeon and 0.7p for Itanium 2; AC, AMD ACML v1.0 for AMD64 (32 bit). [f] Published SpecFP2000 base/peak marks at http://www.spec.org/cpu2000 tested under native OS of Alpha and IBM, and Linux (L) and MS Windows (W) for Intel and AMD machines. [g] For 32-bit binary.

more, the "-melf_i386" option is necessary to link the executables as 32-bit ELF format (the native Linux binary format) at the linking stage, and to pass the above options to the linker, the options of "-Wl,-melf_i386" should be used if the linking is to be done by the compilers (**gcc**, **g77**, and **ifc**, etc.) rather than by **ld**.

For 32-bit Linux distributions that incorporate the new native POSIX threading library, such as RedHat 9.0, an undefined reference to "__ctype_b" may appear and can be solved by the "-i_dynamic" option at the linking stage using **ifc** with MKL.[17] In the case of linking against GOTO library, options of "-lpthread -lsvml" are required to resolve other undefined references.

In the AMD64 system, complains of undefined references to "e_wsfe", "s_wsfe", and "do_fio" appear at the linking stage when using **ifc** in combination with the ACML gnu32 library, and it is cleared up by linking the object file of GOTO's **xerbla.f**, which is recompiled by **ifc** with "-c" option. On the other hand, to successfully link binaries against ACML pgi32 library, **pgf90** must be used instead of **pgf77** to eliminate various undefined references because the pgi32 version of ACML is built with Fortran90 rather than Fortran77. In contrast, using **ifc** to generate 32-bit executables for AMD64 simply requires the "-Wl, -melf_i386" options at the linking stage. Optimization options of "-tpp7 -axW" are to activate the SSE2 support for the double-precision FP acceleration as **ifc** is able to treat the AMD64 hardware as a Pentium4 compatible derivative while performing the 32-bit compilation.

RESULTS AND DISCUSSION

The benchmark results are displayed in Tables 2 through 4. The second part of test419 can normally finish by splitting the RWF file into scratch files less than 2 GB in size.

Compilers. From Table 2, it is observed that the revision of **ifc** from 6.0 to 7.1 can improve the double-precision FP performance by about 3.0% (828 vs 804 min) in the Xeon system with SSE2 support. This observation is close to that in the SpecFP2000 benchmark list, of which the acceleration is 2%. In Table 3, the improvement by **pgf77** from revisions 3.3 to 5.0 is also about 2.9% (1150 vs 1118 min) using an identical numerical library and default G98 tuning options. In Table 4, more extensive tuning options of "-fastsse -time -Mreentrant -Mrecursive -Mnosave -Minfo -Mneginfo -Mscalarsse -Mvect=assoc,recog,prefetch,sse,cachesize:**L2-cache-size**" implemented in **pgf77** 5.0 are able to improve the efficiency by approximately 25.1% (1066 vs 852 min) and 24.8% (1103 vs 884 min), respectively, for the K8-32 system employing ATLAS and ACML. On the other hand, the above-mentioned lengthy tuning options incorporated by **pgf77** failed to execute test364 and caused infinite loops during structural minimization in the first part of test439 for the Xeon system (Table 3). In the K8-32 system using either simply "-fastsse" or the aforementioned complex tuning, the benchmarks are both about 25% faster for the binaries linked against ATLAS 3.4.7 and ACML than the default G98 compilation. However, such simple and complicated SSE2 activations of PGI compilers have generated unstable ex-

Table 2. CPU Time Consumption (in Minutes) of Each Test Job by the Alpha500 Machine and Intel Xeon Systems with CL2.5

system:	G98[a]	Alpha500	E7505				E7501-S			E7501-T	
compiler:		f90	ifc 6.0	ifc 7.1	ifc 7.1	ifc 7.1	ifc 7.1	ifc 7.1	ifc 7.1	ifc 7.1	ifc 7.1
arch. opt.:[b]		−tune ev6	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)
library:		DXML	ATLAS 3.4.1	ATLAS 3.4.1	MKL 6.0	GOTO 0.6p	ATLAS 3.4.1	MKL 6.0	GOTO 0.6p	ATLAS 3.4.1	MKL 6.0
322-1	5.17	7.87	2.52	2.47	2.20	2.22	3.28	2.37	2.63	2.78	2.49
322-2	142	154	58	57	57	56	67	70	64	61	59
322-3	236	225	82	82	81	80	86	86	104	84	84
338	35	36	14	14	13	13	18	15	16	15	15
339	44	43	16	15	15	15	17	20	18	17	16
364	44	48	23	23	23	23	28	24	24	24	24
397	684	658	233	228	223	223	275	251	271	256	244
415-1	29	24	10	10	9	9	13	11	11	10	10
415-2	62	52	21	21	21	21	27	22	25	23	23
420-1	131	113	49	48	47	47	62	52	55	55	49
420-2	153	136	48	46	47	46	50	60	55	53	49
424-1	89	76	29	29	27	27	37	30	33	32	29
424-2	89	75	29	29	26	27	34	32	30	32	28
438	52	75	17	17	16	14	23	22	17	18	17
439-1[c]	134(2)	123(2)	45(2)	77(4)	42(2)	42(2)	96(4)	56(2)	50(2)	87(4)	45(2)
439-2	102	90	35	34	33	33	41	39	39	39	35
447-1	194	214	74	69	70	73	95	84	89	75	77
447-2	116	121	39	35	35	33	41	38	42	37	39
559-4	31	35	10	9	10	9	10	11	10	11	13
560-4	35	40	11	10	12	11	12	12	11	12	15
561-4	81	90	27	26	26	26	31	31	30	27	29
sum	2354	2313	828	804	793	788	970	912	947	884	857

[a] The CPU time in the g98 packages under $g98root/g98/tests/sgi directory. [b] The global option to **ifc** except the architectural optimization is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". [c] The numbers in the parentheses are the step numbers required in the geometrical optimization calculations; this CPU time is excluded from sum.

Table 3. CPU Time Consumption (in Minutes) of Each Test Job by the Intel Xeon Systems with CL2 as Well as the IA64 System

system:	E7505[a]								zx6000			
compiler:	ifc 7.1	ifc 7.1	ifc 7.1	PGI 3.3−1	PGI 5.0−2	PGI 5.0−2	PGI 5.0−2	PGI 5.0−2	efc 7.1	efc 7.1	efc 7.1	efc 7.1
arch. opt.:[b]	(a)	(a)	(a)	(b)	(b)	(b)	(c)	(c)	−tpp2	−tpp2	−tpp2	−tpp2
library:	ATLAS 3.4.1	MKL 6.0	GOTO 0.6p	ATLAS 3.4.1	ATLAS 3.4.1	GOTO 0.6p	ATLAS 3.4.1	GOTO 0.6p	ATLAS 3.4.1	MKL 6.0	GOTO 0.7p	BLAS
322-1	2.46	2.13	2.17	2.99	3.61	2.69	2.71	2.54	2.80	2.38	2.59	2.18
322-2	55	55	55	94	94	92	75	failed	58	58	58	58
322-3	81	80	80	129	119	121	107	-	88	88	88	87
338	14	13	13	20	20	20	17	17	14	14	14	14
339	15	15	15	23	23	22	21	19	18	17	17	18
364	23	23	23	31	30	30	failed	failed	15	15	15	14
397	219	216	217	334	316	316	269	271	222	222	223	220
415-1	9	9	9	12	12	12	11	11	11	11	10	11
415-2	21	20	21	25	25	25	23	23	23	21	21	22
420-1	47	46	44	62	58	58	51	51	40	40	39	41
420-2	45	45	45	61	59	59	52	51	47	48	46	48
424-1	28	26	26	34	33	33	31	failed	28	26	23	32
424-2	28	26	27	34	33	32	31	-	28	25	23	32
438	17	16	14	21	21	18	20	17	23	19	18	22
439-1[c]	76(4)	42(2)	42(2)	69(4)	65(2)	67(2)	−(∞)	57(2)	44(2)	43(2)	43(2)	45(2)
439-2	33	32	32	54	52	51	-	44	32	32	32	32
447-1	69	68	71	104	101	106	86	88	68	66	65	70
447-2	34	35	33	51	51	48	44	failed	41	40	37	42
559-4	9	10	10	10	13	12	10	-	10	11	12	11
560-4	10	11	11	12	15	14	11	-	11	14	14	13
561-4	25	25	26	36	39	38	33	-	31	32	33	31
sum	784	773	774	1150	1118	1110	-	-	811	801	791	820

[a] DRAM CAS latency set to 2 in the BIOS menu. [b] The global option of **ifc** is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". The global optimization to **pgf77** is set to "-Munroll -O2", plus: (b) "-tp p7 -Mvect=cachesize:524288"; (c) "-fastsse -time -Mreentrant -Mrecursive -Mnosave -Minfo -Mneginfo -Mscalarsse -Mvect=assoc,recog,prefetch,sse,cachesize:524288". The global optimization of **efc** is set to "-O3". [c] The numbers in the parentheses are the step numbers required in the geometrical optimization calculations; this CPU time is excluded from sum.

ecutables which have failed in test364 and the fourth part of test560. For the K8-32 system, the most efficient 32-bit binaries are produced by **ifc** with vectorizations of "-tpp7 -axW -ipo -ipo_obj", which simply sees the Opteron processor as a Pentium4/Xeon clone. The required number of steps to converge the geometrical optimization in the first part of test439 is two, no matter which of the numerical libraries is being linked against. The PGI-compiled binaries need four steps, however, to converge the same optimization utilizing ATLAS.

Table 4. CPU Time Consumption (in Minutes) of Each Test Job by the AMD Opteron System and IBM P690

system:	K8-32[a]										P690
compile:	ifc 7.1	ifc 7.1	ifc 7.1	pgf77 5.0-1	pgf77 5.0-1	pgf90 5.0-1	pgf77 5.0-1	pgf90 5.0-1	pgf77 5.0-1	pgf90 5.0-1	xlf 8.0
arch. opt.:[b]	(a)	(a)	(a)	(b)	(b)	(b)	(c)	(c)	(d)	(d)	g98 default
library:	ATLAS 3.4.7	MKL 6.0	ACML 1.0	BLAS	ATLAS 3.4.7	ACML 1.0	ATLAS 3.4.7	ACML 1.0	ATLAS 3.4.7	ACML 1.0	ESSL
322-1	2.10	1.92	2.45	2.66	2.72	3.44	2.36	3.1	2.36	3.1	2.08
322-2	49	49	49	72	72	70	62	60	62	60	38
322-3	76	74	74	124	125	123	94	93	94	93	90
338	12	12	12	20	19	19	15	15	15	15	12
339	14	14	14	23	22	23	17	18	17	18	15
364	17	17	17	24	24	25	18	19	18[d]	19[d]	15
397	196	196	197	297	297	296	232	231	232	231	246
415-1	9	10	10	14	13	14	11	12	11	12	9
415-2	20	21	22	30	26	29	22	25	23	25	17
420-1	43	43	43	64	62	61	49	49	49	49	47
420-2	43	43	43	64	62	61	49	49	49	49	56
424-1	27	27	29	52	33	39	30	36	30	35	25
424-2	26	27	29	52	33	39	30	36	30	35	25
438	18	18	22	36	22	38	19	37	19	36	18
439-1[c]	39(2)	39(2)	39(2)	68(2)	64(2)	64(2)	89(4)	48(2)	89(4)	49(2)	91(4)
439-2	30	31	30	49	49	49	38	33	38	33	39
447-1	64	64	65	109	98	102	74	77	74	78	72
447-2	40	41	41	74	60	62	47	49	47	49	38
559-4	8	9	8	8	8	9	8	8	8	8	13
560-4	9	10	9	9	9	10	9[d]	9[d]	9	9	15
561-4	25	25	25	30	29	31	25	27	25	27	38
sum	728	733	741	1154	1066	1103	851[d]	886[d]	852[d]	884[d]	830

[a] The 32-bit compilation under 64-bit operating system. [b] The global option of **ifc** is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". The global optimization to **pgf77**/**pgf90** is set to "-Munroll -O2", plus: (b) "-tp k8-32 -Mvect=cachesize:1048576"; (c) "-tp k8-32 -Mvect=cachesize:1048576 -fastsse"; (d) "-fastsse -time -Mreentrant -Mrecursive -Mnosave -Minfo -Mneginfo -Mscalarsse -Mvect=assoc,recog,prefetch,sse,cachesize:1048576". [c] The numbers in the parentheses are the step numbers required in the geometrical optimization calculations; this CPU time is excluded from sum. [d] Estimated.

Table 5. Performance Correlation between the SpecFP2000 Benchmark and GAUSSIAN 98 Results

system name:	Alpha500	P690	E7505(CL2)		E7501-S	E7501-T	zx6000		K8-32	
FORTRAN compiler:	f90	xlf 8.0	ifc 7.1	PGI 5.0-2	ifc 7.1	ifc 7.1	efc 7.1	efc 7.1	ifc 7.1	ifc 7.1
architectural optimizations:	−tune ev6 −arch ev6	−power4	−tpp7 −axW	-tp p7	−tpp7 −axW	−tpp7 −axW	−tpp2	−tpp2	−tpp7 −axW	−tpp7 −axW
numerical library and version:	DXML	ESSL	MKL 6.0	ATLAS 3.4.1	MKL 6.0	MKL 6.0	BLAS	GOTO 0.7p	ATLAS 3.4.7	MKL 6.0
sum of CPU time[a]	2313	830	773	1118	912	857	820	791	728	733
inverse ratio of CPU time sum [b]	0.35	0.99	1.06	0.73	0.90	0.96	1.00	1.04	1.13	1.12
geometric mean ratio [b]	0.36	1.04	1.08	0.76	0.92	0.96	1.00	1.04	1.15	1.12
published SpecFP2000 base	383	1202	934				1139		1093	
measured SpecFP2000 base[c]	−[e]	1044	941				1157		1007	
measured SpecFP2000 base ratio [d]	0.33	0.90	0.81				1.00		0.97	
published SpecFP2000 peak	422	1266	944				1139		1168	
measured SpecFP2000 peak[b]	−[e]	1083	947				1157		1043	
measured SpecFP2000 peak ratio [d]	0.36	0.94	0.82				1.00		1.05	

[a] The summation over CPU time includes all the test files listed except the first part of test439. [b] Inverse CPU time ratio computed by (T_{ref}/T) and geometric mean ratio calculated by $\sqrt[n]{\prod_i^n (T_{refi}/T_i)}$, where the reference system is zx6000/BLAS. [c] SpecFP2000 benchmark done by the authors. The marks differ from the published values due to the variations of compilers, operating systems. and environmental settings. [d] All SpecFP ratios are relative to the measured values in the zx6000 system (base = peak = 1157). [e] Measurements of SpecFP2000 cannot be performed because of absence of KAP C and Fortran compilers.

Numerical Libraries. In Table 2, these Xeon systems running at CL2.5 show diverged performance with executables compiled by **ifc**. The MKL has the best efficiency among the three libraries, possibly because it is a native product by the hardware manufacturer. The GOTO library ranks second, and ATLAS 3.4.1 ranks last. Moreover, the performance differs more significantly in the systems with the E7501 chipset, whereas the differences are closer in the E7505 system. Running at CL2 mode and using **ifc**, the E7505 system tends to perform equally well with the MKL and GOTO libraries, and overtakes the efficiency of ATLAS by about 1.3% (773 and 774 vs 784 min). In the identical system using **pgf77** the advantage of the GOTO library is ignorable when compared to that by ATLAS (0.7%).

The numerical solution recommended by most software vendors for the IA64 architecture is MKL. In Table 3, the GOTO library performed the best among all of the libraries. In the HP zx6000 system the ATLAS performs relatively

Table 6. CPU Time Consumption (in Minutes) of Each Test Job Concurrently Executed in Duplicate in the E7505 (CL2) System

compiler:	ifc 7.1 (a)						PGI 5.0-2 (b)		
arch. opt.:[a]									
library:	BLAS			MKL 6.0			GOTO 0.6p		
tasking:	single	dup. 1	dup. 2	single	dup. 1	dup. 2	single	dup. 1	dup. 2
322-1	2.15	3.37	3.38	2.13	3.34	3.35	2.69	3.70	3.68
322-2	56	68	68	55	74	73	92	113	113
322-3	80	99	100	80	95	95	121	136	136
338	13	17	17	13	16	16	20	24	24
339	15	19	18	15	18	18	22	27	27
364	23	24	24	23	24	24	30	47	47
397	220	315	325	216	339	340	316	415	423
415-1	10	12	12	9	12	11	12	16	16
415-2	22	28	28	20	27	27	25	33	34
420-1	46	68	70	46	80	81	58	80	80
420-2	47	75	72	45	69	69	59	76	76
424-1	32	45	46	26	35	35	33	47	48
424-2	32	43	44	26	34	33	32	48	49
438	17	21	20	16	18	18	18	22	24
439-1[b]	42(2)	53(2)	53(2)	42(2)	56(2)	55(2)	67(2)	74(2)	76(2)
439-2	32	43	43	32	42	42	51	59	60
447-1	73	95	95	68	93	92	106	127	127
447-2	36	42	43	35	41	41	48	55	55
559-4	9	11	11	10	13	13	12	16	16
560-4	10	12	13	11	14	14	14	18	18
561-4	25	27	27	25	28	28	38	42	failed[c]
sum	800	1067	1079	773	1075	1073	1110	1405	

[a] The global option of **ifc** is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". The global optimization to **pgf77** is set to "-Munroll -O2", plus: (b) "-tp p7 -Mvect=cachesize:524288". [b] The numbers in the parentheses are the step numbers required in the geometrical optimization calculations; this CPU time is excluded from sum. [c] Reproducible failure while running two copies of test516 concurrently.

Table 7. CPU Time Consumption (in Minutes) of Each Test Job Concurrently Executed in Duplicate in the zx6000 and K8-32 Systems

system:	zx6000									K8-32					
compiler:	efc 7.1 −tpp2 −O3									ifc 7.1 (a)					
arch. opt.:[a]															
library:	intrinsic BLAS			MKL 6.0			GOTO 0.7p			BLAS			ATLAS 3.4.7		
tasking:	single	dup. 1	dup. 2	single	dup. 1	dup. 2	single	dup. 1	dup. 2	single	dup. 1	dup. 2	single	dup. 1	dup. 2
322-1	2.18	2.33	2.31	2.38	2.40	2.43	2.59	2.75	2.73	1.90	2.13	2.10	2.10	2.17	2.16
322-2	58	59	59	58	59	59	58	59	59	49	49	50	49	50	49
322-3	87	88	88	88	89	89	88	89	89	73	73	73	76	74	74
338	14	15	14	14	14	14	14	15	15	13	13	13	12	12	12
339	18	18	18	17	18	18	17	17	17	14	14	14	14	14	14
364	14	15	14	15	15	15	15	15	15	16	17	17	17	17	17
397	220	226	226	222	228	227	223	229	228	195	206	219	196	197	197
415-1	11	12	11	11	11	11	10	11	11	10	10	10	9	10	10
415-2	22	23	23	21	22	22	21	21	21	23	23	23	20	20	20
420-1	41	43	43	40	42	42	39	42	42	43	46	49	43	45	44
420-2	48	52	53	48	50	50	46	49	50	43	47	52	43	45	44
424-1	32	36	36	26	27	27	23	24	24	41	41	41	27	27	28
424-2	32	35	36	25	27	27	23	24	24	40	41	41	26	28	27
438	22	23	23	19	19	19	18	18	18	26	26	26	18	18	18
439-1[b]	45(2)	45(2)	45(2)	43(2)	43(2)	43(2)	43(2)	43(2)	44(2)	39(2)	42(2)	54(2)	39(2)	39(2)	38(2)
439-2	32	32	33	32	33	33	32	32	33	29	28	30	30	30	30
447-1	70	71	71	66	67	67	65	65	65	71	70	71	64	65	65
447-2	42	43	43	40	40	40	37	38	38	47	47	47	40	41	41
559-4	11	12	12	11	13	13	12	13	13	8	8	8	8	9	9
560-4	13	15	15	14	15	15	14	15	15	9	9	9	9	10	10
561-4	31	32	32	32	33	33	33	33	33	25	25	26	25	26	25
sum	820	852	852	801	824	823	791	812	813	777	795	821	728	740	736

[a] The global option of **ifc** is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". [b] The numbers in the parentheses are the step numbers required in the geometrical optimization calculations; this CPU time is excluded from sum.

slower than MKL by 1.2%, yet the difference is rather small. In Table 4, it is suggested that the ATLAS has produced the fastest 32-bit executables under 64-bit operating system in the AMD64 architecture. Combined with **ifc** the efficiency of G98 linked against ATLAS is ahead of the binaries linked against MKL and ACML by 6.9 and 17.8%, respectively. Using PGI compilers the speed of the executables linked against ATLAS outperforms those linked against ACML by 4.1 and 3.5%, respectively, with or without the "-fastsse" option.

Memory Latency. It is noticeable that among the Xeon systems the E7501 chipset has inferior performance com-

pared to that of the E7505 chipset. It is understandable, though, as these two chipsets are designed for different demands. The former chipset targets at the server platforms, enabling larger memory addressing in maximum and stronger I/O functions (multiple PCI-X interfaces), whereas the latter focuses on the workstation level, and has more features such as better graphic bandwidth (AGP 8x). In addition, the E7505 chipset supports CL2 DRAM timing, which is able to reduce the latency and accelerate the performance by 2.6% (804 vs 784 min) using **ifc** with ATLAS. The resulting acceleration in the numerical application is more significant than that reported by the synthetic benchmark suite (0.85%).[10]

Speed and Throughput. Because the SpecCPU2000 benchmark emphasizes on the processors, the memory architecture, and the compilers, the effects of numerical libraries are absent in the SpecFP2000 marks. It is demonstrated in Table 3 that in the zx6000 system the FP performances are close regardless of the numerical libraries, and thereupon the software benchmark results obtained by the intrinsic BLAS library (820 min) is taken as the reference (T_{ref}). All of the sums of CPU time (T) in other systems are represented as the ratios relative to the sum of CPU time in this zx6000 reference system (T_{ref}/T). Another measure is the geometric mean of the run time ratio of each test job (T_i), calculated by

$$\text{geometric mean} = \sqrt[n]{\prod_{i}^{n}(T_{refi}/T_i)} \qquad (1)$$

These correlations are tabulated in Table 5. The published SpecFP2000 marks in the SPEC website utilize similar albeit different hardware from our benchmark systems, and therefore we have provided the SpecFP2000 results metered in our own systems with identical configuration files downloaded from the SPEC website. These measured values are also listed in Table 5 and normalized to the reference zx6000 system. The FP performance of the computational application retains acceptable correlation with the SpecFP2000 benchmark, except for the IA32 architectures in which the numerical applications always outperform the SpecFP2000 benchmark by the aid of extensively tuned numerical libraries.

The CL2-set E7505 system using [**ifc** + MKL] and [**pgf77** + GOTO], K8-32 system with [**ifc** + ATLAS], and zx6000 system equipped with intrinsic BLAS and MKL have been chosen to measure the throughput by means of two copies of each identical test job executed simultaneously. During the tests, the scratch files of each duplication have been written to different physical disks to reduce possible I/O competition. Although each machine has two or more processors installed, they may share common system buses. Therefore processing more than one task concurrently tends to affect the performance. The timing results are shown in Tables 6 and 7, with increased CPU time compared to that by single-tasking. It is observed that, in the zx6000 and K8-32 systems, simultaneous execution of two duplications of identical jobs has a relatively small performance impact to the efficiency. In the IA32 architecture, however, the interference is quite obvious. Converting the CPU time to throughput, the correlation with the SpecFPrate2000 mark measured in these systems is tabulated in Table 8. According

Table 8. Throughput Correlation between the SpecFP2000rate Benchmark and GAUSSIAN 98 Results

system:	E7505 (CL2)								zx6000						K8-32			
compiler:	ifc 7.1 (a)				PGI 5.0-2 (b)				efc 7.1 −tpp2 −O3						ifc 7.1 (a)			
arch. opt.:[a]																		
library:	BLAS		MKL 6.0		GOTO 0.6p				intrinsic BLAS				MKL 6.0		GOTO 0.7p		ATLAS 3.4.7	
duplication:	single	first	second	single	first	second	single	first	second	single	first	second	single	first	second	single	first	second	single	first	second
sum of CPU time[b]	800	1067	1079	773	1075	1073	1110	1405	-	820	852	852	801	824	823	791	812	813	728	740	736
inv. CPU time ratio[c]	1.03	0.77	0.76	1.06	0.76	0.76	0.74	0.58	-	1.00	0.96	0.96	1.02	0.99	1.00	1.04	1.01	1.01	1.13	1.11	1.11
geometric mean ratio [c]	1.05	0.81	0.81	1.08	0.83	0.83	0.79	0.62	-	1.00	0.95	0.96	1.03	0.99	0.99	1.04	1.01	1.01	1.15	1.11	1.12
processors used	1	2								1	2								1	2	
published SpecFPrate2000 base	10.8	14.7								13.2	23.9								12.7	23.3	
measured SpecFPrate2000 base	11.0	13.0								13.6	24.6								11.7	22.0	
measured base SpecFPrate2000 ratio[d]	0.81	0.48								1.00	0.90								0.86	0.81	
published SpecFPrate2000 peak	11.0	14.9								13.2	23.9								13.5	24.9	
measured SpecFPrate2000 peak	11.0	14.1								13.6	24.6								12.1	22.7	
measured peak SpecFPrate2000 ratio[d]	0.81	0.52								1.00	0.90								0.89	0.83	

[a] The global option of **ifc** is set to "-unroll -O2", plus: (a) "-tpp7 -axW -ipo -ipo_obj". The global optimization to **pgf77** is set to "-Munroll -O2", plus: (b) "-tp p7 -Mvect=cachesize:524288". [b] The summation over CPU time includes all the test files listed except the first part of test439. [c] Inverse CPU time ratio computed by (T_{ref}/T) and geometric mean ratio calculated by $\sqrt[n]{\prod_{i}^{n}(T_{refi}/T_i)}$, where the reference system is zx6000/BLAS. [d] 1 processor: (measured SpecFPrate2000 mark)/13.6; 2 processors: (measured SpecFPrate2000 mark)/13.6/2.

to the SpecFP2000rate benchmark carried out by two processors in the E7505 system with CL2 setting, the performance of each CPU is only 59.3% (0.48:0.81) of the original speed, whereas in the zx6000 and K8-32 systems the efficiency factors are 90.0% (0.90:1.00) and 94.2% (0.81:0.86), respectively. With the improvement of **ifc**-compiled BLAS and MKL, the E7505 system is able to accelerate the performance of each CPU to, respectively, 77.1% (0.81:1.05) and 76.9% (0.83:1.08) of the original speed while doing concurrent executions of two duplications of the test jobs in each configuration. In addition, the upper-bounded sums of execution time by the binaries linked against different libraries might imply the saturation in system bandwidth under multitasking. Benchmarking the IA64 and AMD64 architectures with the same strategy, the throughput efficiencies have been improved to 96.1% (zx6000/**efc** + MKL), 97.1% (zx6000/**efc** + GOTO) and 97% (K8-32/**ifc** + ATLAS) of the original single-CPU performance. The aforementioned saturation is not observed in these 64-bit systems, and therefore it is concluded that the IA64 and AMD64 architectures are able to load multiple copies of concurrent numerical computations without significant performance impact compared to the IA32 architecture, possibly benefited by the larger memory bandwidth offered by the advanced hardware implementations.

CONCLUSION

The revisions of both Fortran compilers deliver about 3% of performance advantage. For 32-bit executables, the **ifc** can equally accelerate the performance of the processors with SSE2 instruction sets regardless their manufacturers, and can generate better-performing binary codes. The improvements by the optimized numerical libraries, in terms of ATLAS, GOTO, and MKL, are nearly identical, with differences less than 2% in the E7505 system. For the Intel platforms, MKL with Intel Fortran compilers represents the optimal combination. When software cost is of concern, however, ATLAS and GOTO libraries can serve as alternatives. For the AMD64 architecture running 32-bit application in 64-bit Linux OS, **ifc** can tune binaries as if on Pentium4 clones and invariably accelerate the double-precision FP operations. Although a new PGI compiler with "-tp k8-32" plus its own SSE2 optimization can enhance the FP performance, a few test jobs have failed and require further examination. Significant speed variations between the numerical libraries are observed in the AMD64 platform. Among currently available choices, ATLAS 3.4.7 offers the highest performance regardless of the compiler used. The 0.8-beta version of the single-threaded 32-bit GOTO library for Opteron has generated too many errors in our preliminary tests and therefore the results are not reported. The 32-bit version of multi-threaded GOTO library for Opteron has not been released yet (as of the day of submission of this manuscript), and its performance is to be expected. In addition, compilers have more significant effects on efficiency than the numerical libraries do. Adjusting the CAS latency to CL2 in the E7505 system can additionally accelerate the speed by 5% compared to the default setting of CL2.5. The performance of the electronic structure calculation package retains good correlation with the SpecFP2000 mark with new hardware as well as software technologies, and it continues to be feasible to use this benchmark to estimate the performance of real computational chemistry packages for new architectures. The IA64 and AMD64 machines are more efficient to perform multiple computations concurrently than the IA32 architecture, probably due to these machines' larger memory bandwidths.

ACKNOWLEDGMENT

This work has been supported by the National Science Council of the Republic of China under Grants NSC-91-2321-B-007-002, NSC-92-3112-B-009-001, and NSC-92-2113-M-007- 040.

REFERENCES AND NOTES

(1) Yu, J.-S. K.; Yu, C.-H. Recent Advances in PC−Linux Systems for Electronic Structure Computations by Optimized Compilers and Numerical Libraries. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 673−681.
(2) Whatley, R. C.; Petitet, A.; Dongarra, J. J. Automated Empirical Optimization of Software and the ATLAS Project, dated Sept. 19, 2001. Paper located at http://www.netlib.org/lapack/lawns/lawn147.ps. The ATLAS source code resides at http://math-atlas.sourceforge.net.
(3) From the Spec CPU benchmark, it is observed that IA64 hardware with Linux/Intel Fortran runs faster than an identical platform equipped with HPUX/HP Fortran. See also Reference 9.
(4) Goto, K.; van de Geijn, R. On Reducing TLB Misses in Matrix Multiplication. *FLAME Working Note #9, The University of Texas at Austin, Department of Computer Sciences. Technical Report TR-2002-55*. Nov. 2002. The library can be downloaded at http://www.cs.utexas.edu/users/flame/goto/.
(5) http://www.llnl.gov/linux/mcr/.
(6) http://www.netlib.org/benchmark/performance.ps. The package is available at http://www.netlib.org/benchmark/hpl/.
(7) http://www.top500.org/list/2003/06/.
(8) http://www.developwithamd.com/appPartnerProg/acml/docs/releasenotes.txt. The ACML library can be downloaded from http://www.amd.com/us-en/Processors/DevelopWithAMD/0,,30_2252_2282,00.html.
(9) Henning, J. L. SPEC CPU2000: Measuring CPU Performance in the New Millennium. *COMPUTER* **2000**, July, 28−35. See http://www.spec.org/osg/cpu2000.
(10) This action is performance tuning rather than overclocking the system. See *Perfect Timing: DDR Performance Analysis*, http://www.tomshardware.com/motherboard/20020507/index.html.
(11) Nicklaus, M. C.; Williams, R. W.; Bienfait, B.; Billings, E. S.; Hodošček, M. Computational Chemistry on Commodity-Type Computers. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 893−905.
(12) Yu, J.-S. K.; Yu, C.-H. Benchmarks of the PC−Unix Computer with Electronic Structure Calculation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1111−1114.
(13) See Q12 of http://www.spec.org/cpu2000/press/faq.html.
(14) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery, J. A., Jr.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudin, K. N.; Strain, M. C.; Farkas, O.; Tomasi, J.; Barone, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B. G.; Chen, W.; Wong, M. W.; Andres, J. L.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A. *Gaussian 98*, revision A.11.3; Gaussian, Inc.: Pittsburgh, PA, 1998.
(15) The rest three timing options in the BIOS menu are set as: Act-to-Precharge=5, RAS-to-CAS=2, and RAS Precharge=2, during all of the tests using CL2.5 and CL2.
(16) The default compiling options using PGI compilers for Gaussian 03.
(17) Refer to Dr. A. Kohlmeyer's CPMD online tips, subtitled *Using the Intel IFC Compiler with/without Intel MKL* in the section of *CPMD v.s. Linux: Tips and Downloads*, at http://www.theochem.ruhr-uni-bochum.de/~axel.kohlmeyer/cpmd-linux.html.

CI034210H