64-Bit versus 32-Bit CPUs in Scientific Computing

Axel Kohlmeyer

Lehrstuhl für Theoretische Chemie
Ruhr-Universität Bochum
<axel.kohlmeyer@rub.de>

March 2004

Outline

- 64-Bit and 32-Bit CPU Examples
- 64-Bit and Scientific Computing
- What is the best Machine (for me)?
- Summary

What is a 64/32-Bit CPU?

- CPUs have (several, separate) sub-units: integer, floating-point, SIMD/vector, load/store, caches/buffers
- Size of integer registers usually defines “bitness”
- there are “pure” and “hybrid” 64/32-Bit CPUs.
- IEEE754 double precision floating-point is 64-bit.

Example: DEC/Compaq/HP Alpha

- oldest widely available 64-bit platform (RISC)
- medium clock rates, high floating-point performance
- up to 256-bit wide memory interface
- SMP via elaborate crossbar switches
- typical “Workstation” CPU

Example: Intel Itanium

- completely new 64-bit design (EPIC)
- low clockrate, very high floating-point speed
- 32-bit via emulation only
- architecture offloads optimizations to the compiler
- SMP via processor bus

Example: Intel Pentium 4 / Xeon

- most common 32-bit platform (CISC)
- instructions to access up to 64GB memory
- extremely high clock rates
- SMP via processor bus
- SSE unit for floating point vector operations

AMD Athlon similar: lower clock, twin floating-point

Example: AMD Athlon64/Opteron

- evolutionary 64-bit design (extension to 32-bit)
- memory controller in CPU integrated
- SMP via point-to-point connect
- hybrid 32-/64-bit use in hardware possible

Summary: Architectures

- different architecture strategies
- each platform has different characteristics
- CPU only part of the platform
- 64-bitness only minor aspect
- simple comparison of CPU only not useful

Demands of Scientific Computing

- Ex.: Fluid Dynamics, Molecular Dynamics, Quantum Chemistry/Physics, . . .
- mainly floating-point intensive
- high memory bandwidth for large datasets
- Linear Algebra, Fast-Fourier-Transform
- Parallelization (MPI, OpenMP, Multi-threading)

Impact of change to 64-bit CPU

- o intrinsic floating-point performance bitness independent
- o programming language mostly bitness independent
- \+ larger address space $\rightarrow$ compute larger problems
- \+ new architecture $\rightarrow$ no legacy support required
- – larger registers $\rightarrow$ more cache contamination
- – new compiler, new performance libraries needed, new tuning tricks need to be learned

Benchmarking

- absolutely <u>essential</u>
- needs to be done by people who know the application, the hardware, and the operating system
- needs several representative benchmarks
- careful with artificial benchmarks or marketing myths

Some Benchmark Results

- Car-Parrinello Molecular Dynamics Simulation
- linear algebra (matrix multiplication) + FFT $\Rightarrow$
  - ⋆ high memory bandwidth
  - ⋆ medium to large memory
  - ⋆ MPI-parallel (MIMD) + OpenMP
  - ⋆ high network throughput for MPI-parallel runs
- uses BLAS/LAPACK a lot

CPMD Serial Runs (10 Ryd)

Machine	Wall Time / s
AMD Athlon XP1600+, 1.4GHz, PC133	545
AMD Athlon MP1600+, 1.4GHz, PC266-ECC	443
Compaq Alpha EV6, 600MHz, XP1000	435
HP SuperDome 32000, HPPA 8700,750MHz	388
AMD Athlon XP2500+, 1.83GHz, PC333	361
Compaq Alpha EV67, 677MHz, ES40	284
AMD Opteron, 1.6GHz, 32-bit	287
AMD Opteron, 1.6GHz, 64-bit	254
Intel P4 Xeon, 2.4GHz, PC266	236
Compaq Alpha EV68AL, 833MHz, DS20	234
Intel Itanium2, 900MHz, HP zx6000	206
AMD Athlon64 3200+, 2.0GHz, PC333, 64-bit	173
IBM Power4+ 1.7 GHz, Regatta H+	171

CPMD Serial Runs (30/50 Ryd)

Machine	Wall Time / s
AMD Athlon XP1600+, 1.4GHz, PC133	2878
HP SuperDome 32000, HPPA 8700, 750MHz	2672
Compaq Alpha EV6, 600MHz, XP1000	2624
AMD Opteron, 1.6GHz, PC266 memory, 64-bit	1292
Intel Pentium 4 Xeon, 2.4GHz,	1275
AMD Opteron, 1.6GHz, PC266 memory, 32-bit	1157
IBM Power4+ 1.7 GHz, Regatta H+	997
AMD Athlon XP1800+, 1.53GHz, PC266	5878
AMD Athlon XP2500+, 1.83GHz, PC333	5196
AMD Athlon XP2500+, 1.83GHz, dual-channel PC333	3848
Intel Itanium2, 900MHz, HP zx6000	3145
AMD Opteron, 1.6GHz, PC266 memory, 32-bit	3143
AMD Athlon64 3200+, 2.0GHz, PC333, 64-bit	3134
IBM Power4+ 1.7 GHz, Regatta H+,	2259

SMP Overhead

one serial run per CPU simulataneously
relative speed compared to a single cpu run
does not account for multi-threading latencies

Machine	relative SMP speed
Quad Pentium 4 Xeon, 2.4GHz	31%
Dual Pentium 4 Xeon, 2.4GHz	61%
Dual AMD Athlon MP1800+, 1.53GHz	69%
Dual AMD Athlon MP1600+, 1.4GHz	73%
Dual Compaq Alpha EV68AL, 833MHz, DS20	88%
Dual Intel Itanium2, 900MHz, HP zx6000	89%
Quad Compaq Alpha EV67, 667MHz, ES40	90%
AMD Opteron, 1.6GHz, PC266, 32-bit,	98%
AMD Opteron, 1.6GHz, PC266, 64-bit	98%

Program Differences: CPMD vs.CMB2

CPMD: Machine	Wall Time / s
AMD Athlon XP1600+, 1.4GHz, PC133	2878
Compaq Alpha EV6, 600MHz, XP1000	2624
AMD Athlon XP1800+, 1.53GHz,	2136
Compaq Alpha EV68AL, 833MHz, DS20	1519
AMD Opteron, 1.6GHz, 64-bit	1292
Intel Itanium2, 900MHz, HP zx6000	1158
AMD Opteron, 1.6GHz, 32-bit	1157

CMB2: Machine	Wall Time / s
AMD Athlon XP1600+, 1.4GHz, PC133	1914
AMD Athlon XP2500+, 1.83GHz, PC333	1782
Compaq Alpha EV6, 600MHz, XP1000	1740
Intel Itanium2, 900MHz, HP zx6000	1266
Compaq Alpha EV68AL, 833MHz, DS20	1254
AMD Opteron, 1.6GHz, 64-bit	1194
AMD Opteron, 1.6GHz, 32-bit	1080

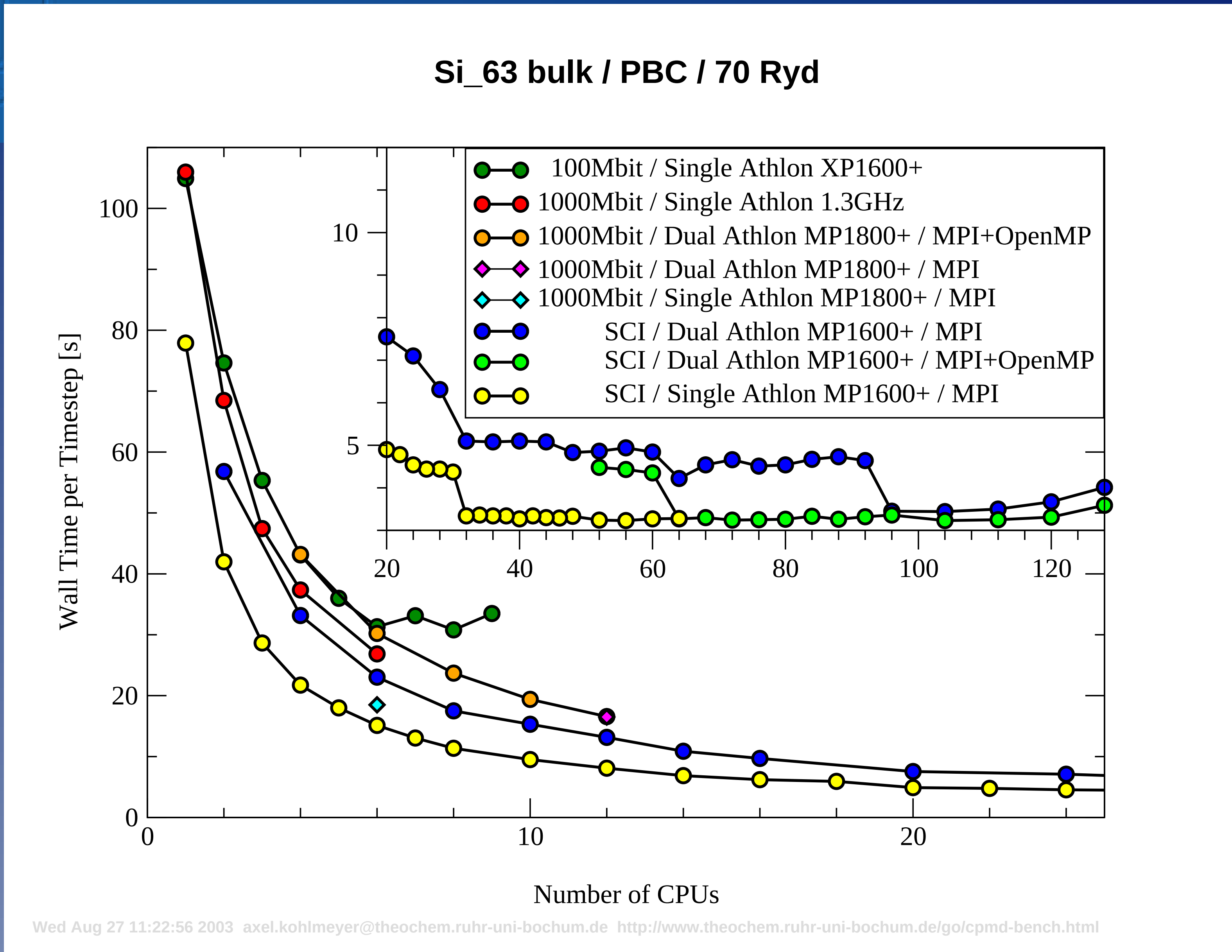
Library Optimizations

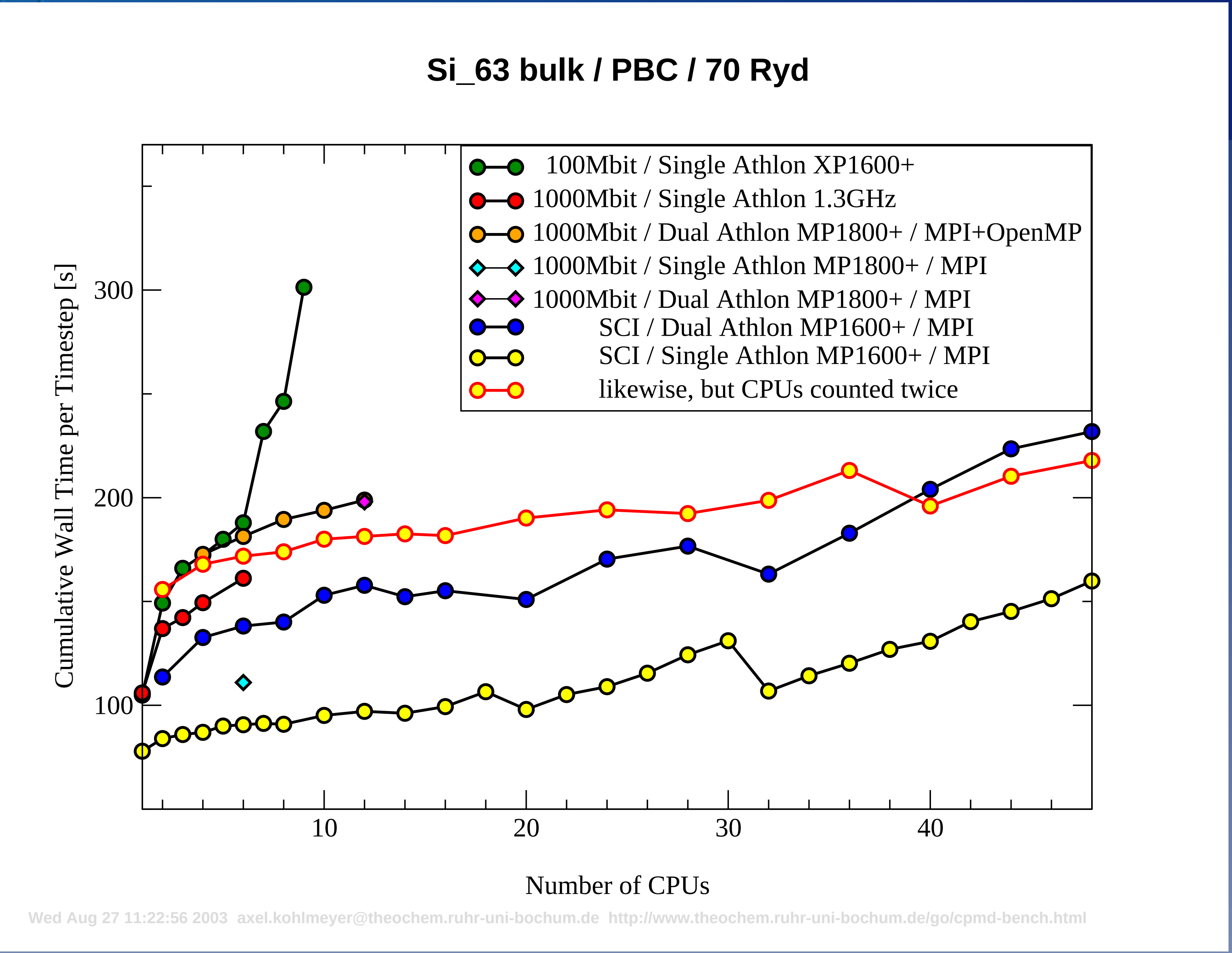
100 steps CP-MD: 63 Si-Atoms, 10Ryd

Machine	BLAS	generic ATLAS	specific ATLAS
Athon XP1800+	950 s	428 s	378 s
	251%	113%	100%
Pentium IV 2GHz	765 s	493 s	441 s
	173%	112%	100%
P4 Xeon 2.4GHz	471 s	316 s	276 s
	171%	118%	100%
Pentium M 900MHz	716 s	430 s	-

Networking Impact

- scalability limit of the application
- scalability limit of the network
- throughput / peak performance considerations
- price / performance ratio considerations
- I/O performance limits
- SMP performance limits


Si_63 bulk / PBC / 70 Ryd
100Mbit / Single Athlon XP1600+
1000Mbit / Single Athlon 1.3GHz
1000Mbit / Dual Athlon MP1800+ / MPI+OpenMP
1000Mbit / Dual Athlon MP1800+ / MPI
1000Mbit / Single Athlon MP1800+ / MPI
SCI / Dual Athlon MP1600+ / MPI
SCI / Dual Athlon MP1600+ / MPI+OpenMP
SCI / Single Athlon MP1600+ / MPI
Wall Time per Timestep [s]
Number of CPUs
0
20
40
60
80
100
10
20
5
10
40
60
80
100
120
Wed Aug 27 11:22:56 2003 axel.kohlmeyer@theochem.ruhr-uni-bochum.de http://www.theochem.ruhr-uni-bochum.de/go/cpmd-bench.html


Si_63 bulk / PBC / 70 Ryd
100Mbit / Single Athlon XP1600+
1000Mbit / Single Athlon 1.3GHz
1000Mbit / Dual Athlon MP1800+ / MPI+OpenMP
1000Mbit / Single Athlon MP1800+ / MPI
1000Mbit / Dual Athlon MP1800+ / MPI
SCI / Dual Athlon MP1600+ / MPI
SCI / Single Athlon MP1600+ / MPI
likewise, but CPUs counted twice
Cumulative Wall Time per Timestep [s]
100
200
300
10
20
30
40
Number of CPUs
Wed Aug 27 11:22:56 2003 axel.kohlmeyer@theochem.ruhr-uni-bochum.de http://www.theochem.ruhr-uni-bochum.de/go/cpmd-bench.html

I/O Performance

Conventional SCF Quantum Chemistry Program

creating 2 * 9 = 18 GByte Integral Files.

Machine	Disk	Cputime	Wall Time
Athlon64 2.0GHz	SCSI 10k rpm	74.0 min	82.5 min
Athlon64 2.0GHz	IDE 7.2k rpm	73.5 min	81.0 min
Athlon64 2.0GHz	IDE RAID-0	74.0 min	75.0 min
AthlonXP 1.53GHz	IDE RAID-0 (old)	105 min	123 min
Athlon 650MHz	IDE RAID-0	254 min	255 min

I/O Performance

Conventional SCF Quantum Chemistry Program

16 SCF Iterations using the Integral Files.

Machine	Disk	Cputime	Wall Time
Athlon64 2.0GHz	SCSI 10k rpm	58.5 min	160.0 min
Athlon64 2.0GHz	IDE 7.2k rpm	58.5 min	128.0 min
Athlon64 2.0GHz	IDE RAID-0	60.5 min	77.0 min
AthlonXP 1.53GHz	IDE RAID-0 (old)	114 min	148 min
Athlon 650MHz z	IDE RAID-0	249 min	266 min

What else does matter?

- reliability: 2nd/3rd fastest components more reliable.
- availability of OS updates, compiler and optimized libraries
- optimize for the common case, not to fit all demands best
- (small) special machine for special uses only
- get (and pay for) real Hardware support
  or do-it-yourself and get a larger machine

Code Optimizations

- Fortran90 vs Fortran77 + BLAS (C++ vs C)
  $\Rightarrow$ ease of use vs. absolute speed
- prefer generic optimizations to specific
- search for better algorithms
- identify performance bottlenecks
- check accuracy (danger of overoptimizing)

Summary

- Bigger is not always better!
- 32-bit or 64-bit does not really matter, unless you need the address space
- determine requirements of the dominant application(s).
- representative benchmarks to find best architecture
- ’weakest link in chain’ determines total performance
- always consider the whole architecture (cpu, memory, i/o).
- factor in future software support requirements (compiler, optimized libraries, parallization support).

Thanks

- RRZE Erlangen
- FZ-Jülich
- Ruhr-Universität Bochum
- Prof. Domink Marx, Prof. Volker Staemmler, Dr. Bernd Meyer, Dipl.-Chem. Holger Langer

---

http://www.theochem.rub.de/~axel.kohlmeyer/
axel.kohlmeyer@rub.de