

The PC GAMESS Project at MSU

Alex A. Granovsky

Laboratory of Chemical Cybernetics, M. V. Lomonosov
Moscow State University, Moscow, Russia

November 1999, SC'99 PC GAMESS BOF

What is PC GAMESS?

The **PC GAMESS** is the Intel-specific version of the well-known quantum chemistry (QC) GAMESS (US) program.

What are the PC GAMESS key features?

- The program is **strongly modified** to achieve the maximum possible performance on Intel-based platforms
- **Functionally extended** to those QC methods which are not currently present in the regular GAMESS version
- **Written to support both shared memory** (*via multithreading on SMP systems*) and **distributed memory** (*via MPI on LANs and PC clusters*) parallel models of execution
- **Operational on all popular PC Operating Systems:**
 - ◆ **Win32: NT** (the preferred OS for PC GAMESS) & Win9x
 - ◆ **Linux** (only partial support at present)
 - ◆ OS/2
- **Different executables tuned for Pentium, Pentium Pro, Pentium II and Pentium III CPUs**

Outline

- a brief **historical overview** of the PC GAMESS project and its evolution
- main **directions** of our **current and future** PC GAMESS-related activity
- the results of **several PC GAMESS performance measurement tests**

The PC GAMESS history: Prologue.

Year 1993.

- We got a copy of the GAMESS (US) program (this was probably the first copy of this program in the Russia).
- GAMESS was compiled to run under DOS/DPMI on Intel 386 and 486-based systems.

The PC GAMESS history: Year 1994.

- Pentium-based systems became a serious alternative to the traditional workstations because the available FP performance was dramatically increased.
- We got several Pentium-based systems.
- Start of the PC GAMESS project.

Initial goal: create GAMESS version which will run as fast as possible on Intel-based systems.

First phase of the PC GAMESS project: Years 1994-1996.

- Choice of the appropriate Fortran compiler (Watcom F77) as well as base Operating System (Windows NT);
- Multiple bug fixes;
- Pentium-specific optimization;
- Creation of the low-level library of QC primitives (LQCP);
- Usage of MKL v. 1.0 (Intel's Math Kernel Library which includes highly optimized BLAS routines);
- First PC GAMESS versions (for NT, OS/2, and DOS) used internally at MSU.

This phase was entirely (PC) GAMESS driven.

Second phase of the PC GAMESS project: Year 1997.

- First public PC GAMESS versions;
- Multiple source-level changes to improve performance;
- Further development of the LQCP, P6 family specific optimization;
- Use of MKL v. 2.0. Multiple changes in order to use BLAS level 3 much more extensively;
- First Pentium Pro/Pentium II optimized PC GAMESS version with partial SMP support (based exclusively on MKL-level multithreading);
- Development of the our own QC codes (PC GAMESS specific):
 - ◆ Fast non-Fortran file I/O with large files support;
 - ◆ Real time data packing/unpacking technology;
 - ◆ Fast MP2 energy/energy gradient code which is capable to handle very large systems.

*This intermediate phase was driven both by
the PC GAMESS and by ourselves.*

Third phase of the PC GAMESS project: Year 1998.

- New GAMESS (US) code developed by the Gordon's group at ISU was incorporated into the PC GAMESS;
- Several modules are rewritten completely for a speed;
- Development of the new memory management technology;
- Further development of our own QC codes:
 - ◆ SMP-enabled MP3 energy module;
 - ◆ SMP-enabled MP4(SDQ) energy module.

New goals formulated in this year:

- ◆ to include modern high-level highly-correlated calculation techniques into PC GAMESS;
- ◆ to improve SMP support.

*All PC GAMESS-related activity are now
entirely driven by ourselves.*

Third phase of the PC GAMESS project: Year 1999.

★ Our research project is now supported by Intel.

■ Further development of our own QC codes:

- ◆ SMP-enabled MP4(SDTQ) energy module;
- ◆ Code prototypes for CISD, CCSD, QCISD, and BD energies.

■ Better SMP support:

- ◆ Improvement of SMP scaling properties for MP3/MP4 codes;
- ◆ Changes to switch from the BLAS-level parallelism to the native parallelism;
- ◆ SMP-enabled conventional RHF, ROHF, UHF, and TDHF codes.

New priorities formulated this year:

- ◆ Creation of the parallel PC GAMESS versions;
- ◆ Improved performance on Pentium II/III CPUs;
- ◆ Linux support.

Third phase of the PC GAMESS project: Year 1999 (continued).

■ Parallel PC GAMESS:

- ◆ Parallel (MPI) PC GAMESS version was created for Win32-based LANs and clusters. All the features of the standard PC GAMESS versions are available in the parallel version as well;
- ◆ MP4(SDTQ) module was rewritten to work in parallel mode.

■ Improved performance on Pentium II/III CPUs:

- ◆ From MKL v. 2.1 to the newest MKL v. 3.1;
- ◆ CPU type, L1, and L2 cache sizes autodetection. This information is used for automatic fine-tuning by several time-critical parts of the PC GAMESS;
- ◆ Additional Pentium II/III-specific optimization of LQCP.

■ Linux is supported using the combination of modified WINE and Win32 PC GAMESS executables.

Current PC GAMESS-related activity.

- Development of the efficient CISD, CCSD, QCISD, BD, CCSD(T), QCISD(T), and BD(T) energy code with a good SMP and parallel mode scaling;

Years 2000-2001 plans and priorities.

- Native Linux PC GAMESS versions;
- Development of the efficient parallel algorithms for CISD, CCSD, BD, QCISD, MP4(SDQ) analytical energy gradients as the part of the PC GAMESS;
- Development of the efficient MP2 energy analytical second derivatives module;
- Development of the efficient parallel MRDCI energy module;
- Development of AO-based large CIS energy and energy gradient modules;
- IA64-based (Itanium) PC GAMESS version.

PC GAMESS performance measurement tests.

- The performance of fully parallel MP4(full) algorithm which is the part of the PC GAMESS.
- The same test job is used, only the execution environment is different, namely:
 - ◆ Four-CPUs Pentium III Xeon 550 MHz (1MB L2 cache) system running under Windows NT Server v. 4.0 (Example #1);
 - ◆ Eight-CPUs Pentium III Xeon 550 MHz (2MB L2 cache) Profusion-based system running under Windows NT Server v. 4.0 (Example #2);
 - ◆ Cluster built from four dual-CPU Pentium III Xeon 500 MHz (1MB L2 cache) systems running under Windows NT Workstation v. 4.0 (Example #3).
- What we'll study in our tests are:
 - ◆ The PC GAMESS performance;
 - ◆ SMP and parallel scaling properties on different hardware.

What is the Amdahl's law?

- One of the simplest but very popular model which allows one to relate the speedup (S) with the number of CPUs (n) used:

$$S(n) = 1/(\alpha + (1-\alpha)/n)$$

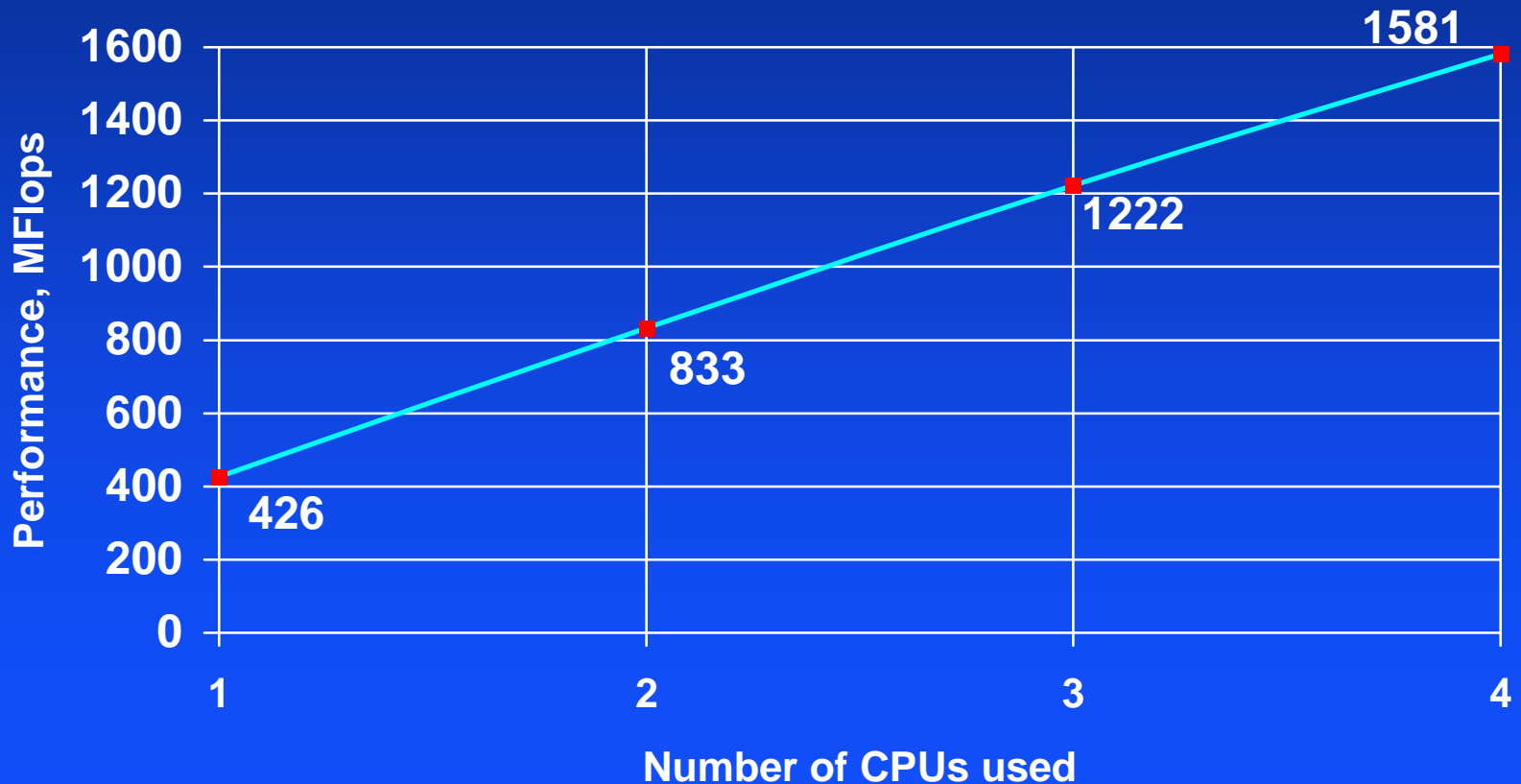
α is the relative part of operations which are performed sequentially using only one CPU.

- We'll use this model to analyze our results.

Example #1. Absolute performance in MFlops vs. number of CPUs used.

MP4(full) SMP scalability testcase

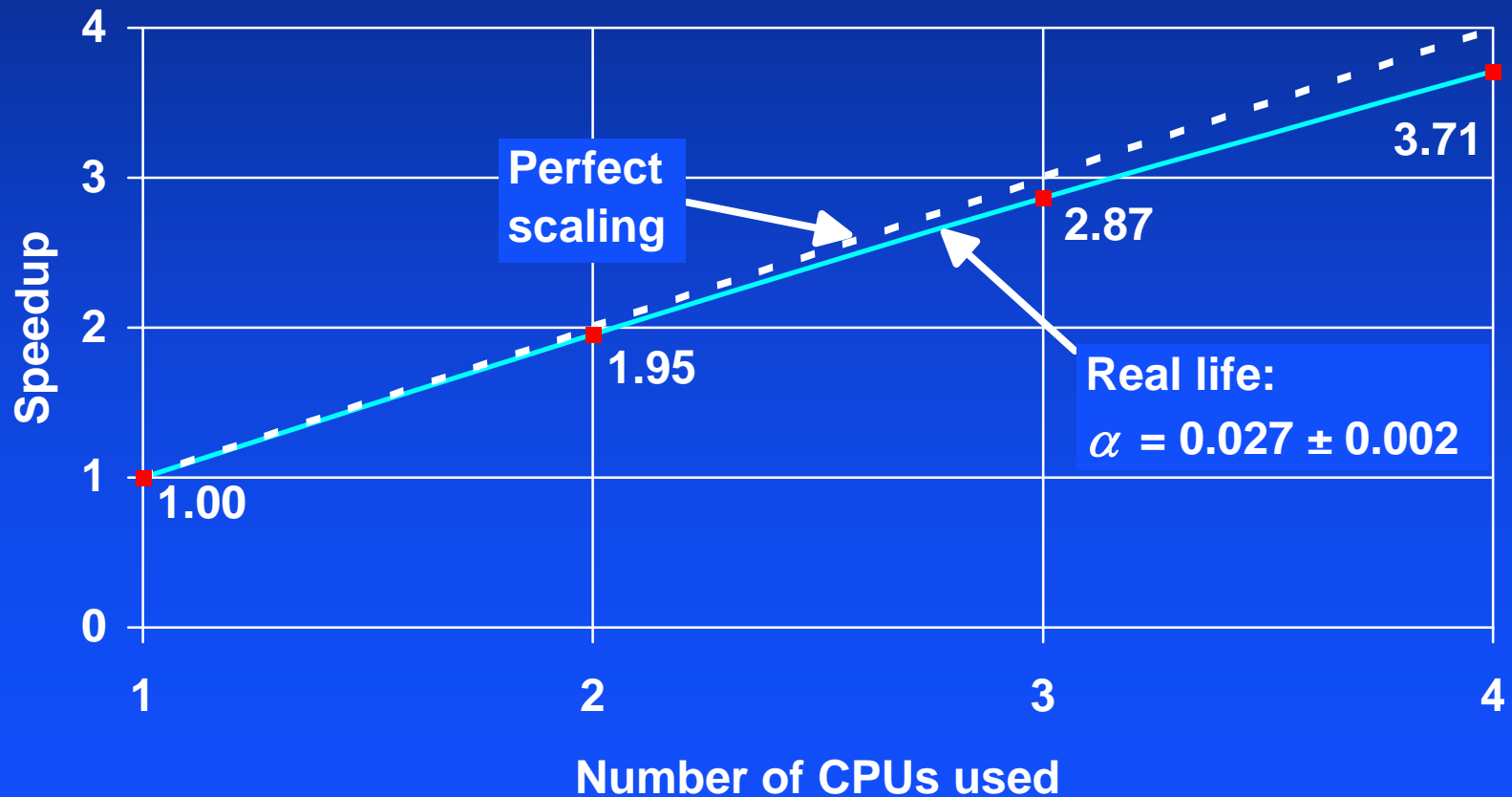
$N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C_1 symmetry group



Example #1. SMP scaling vs. number of CPUs used.

MP4(full) SMP scalability testcase

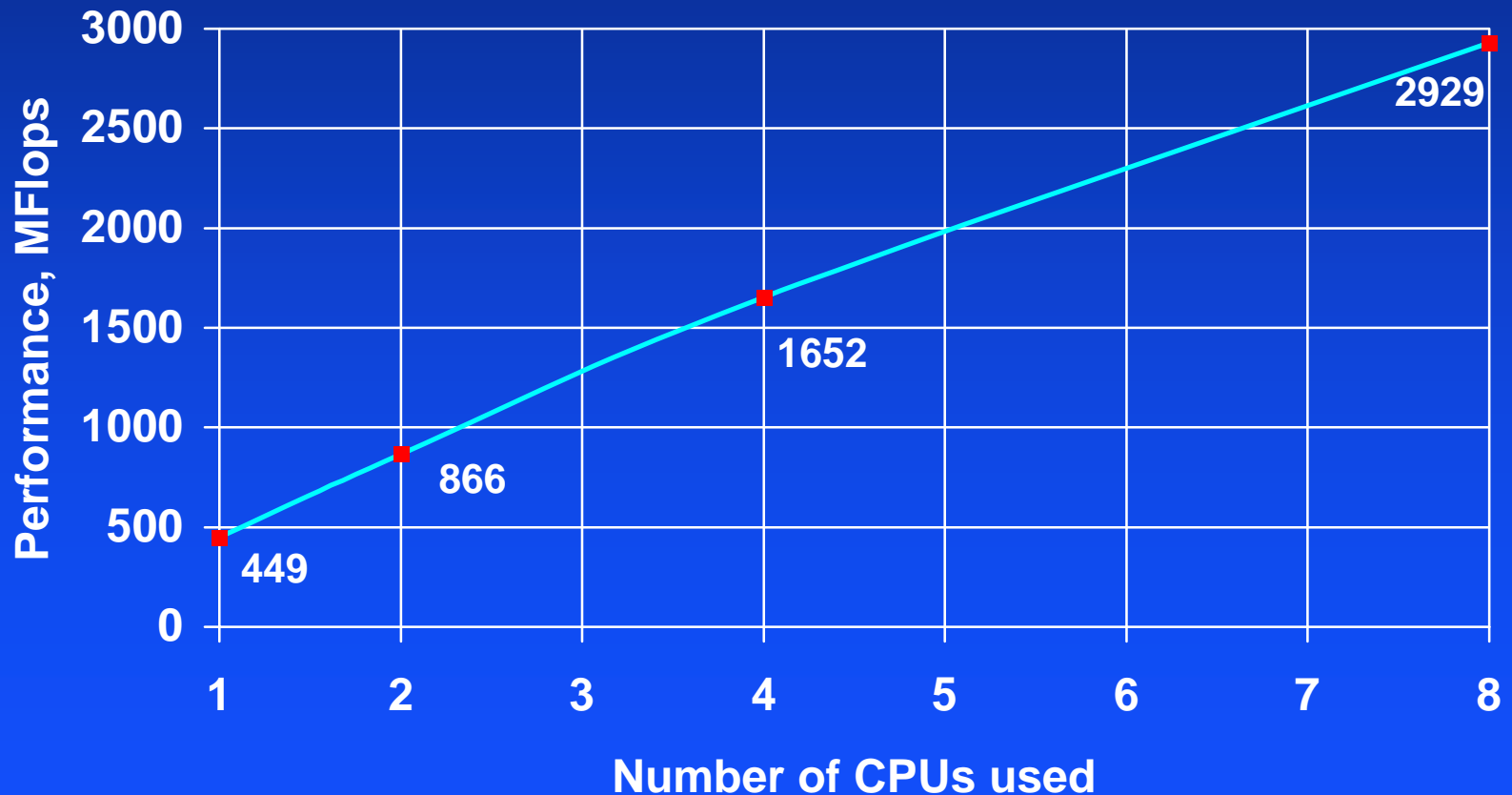
$N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C1 symmetry group



Example #2. Absolute performance in MFlops vs. number of CPUs used.

MP4(full) SMP scalability testcase

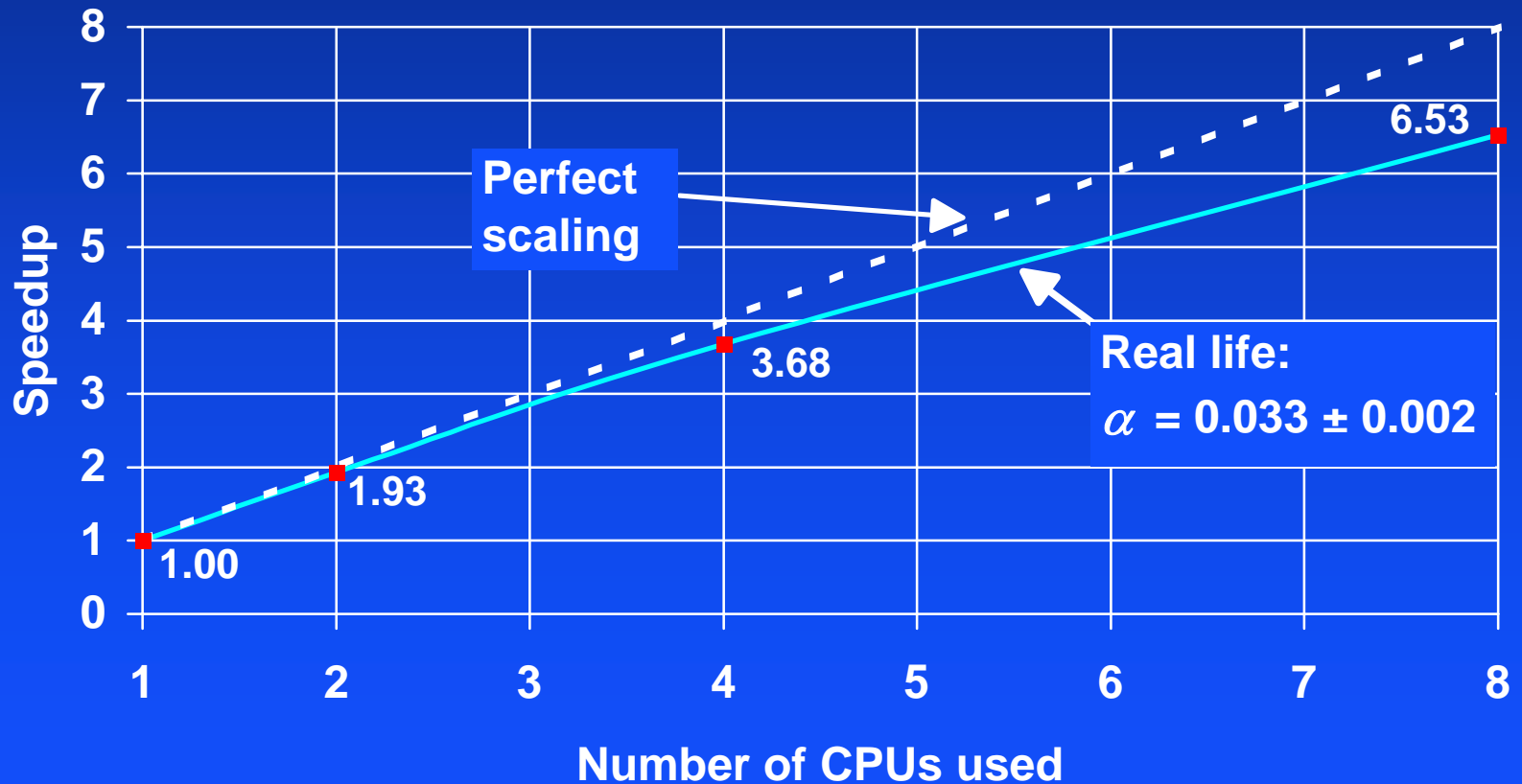
$N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C1 symmetry group



Example #2. SMP scaling vs. number of CPUs used.

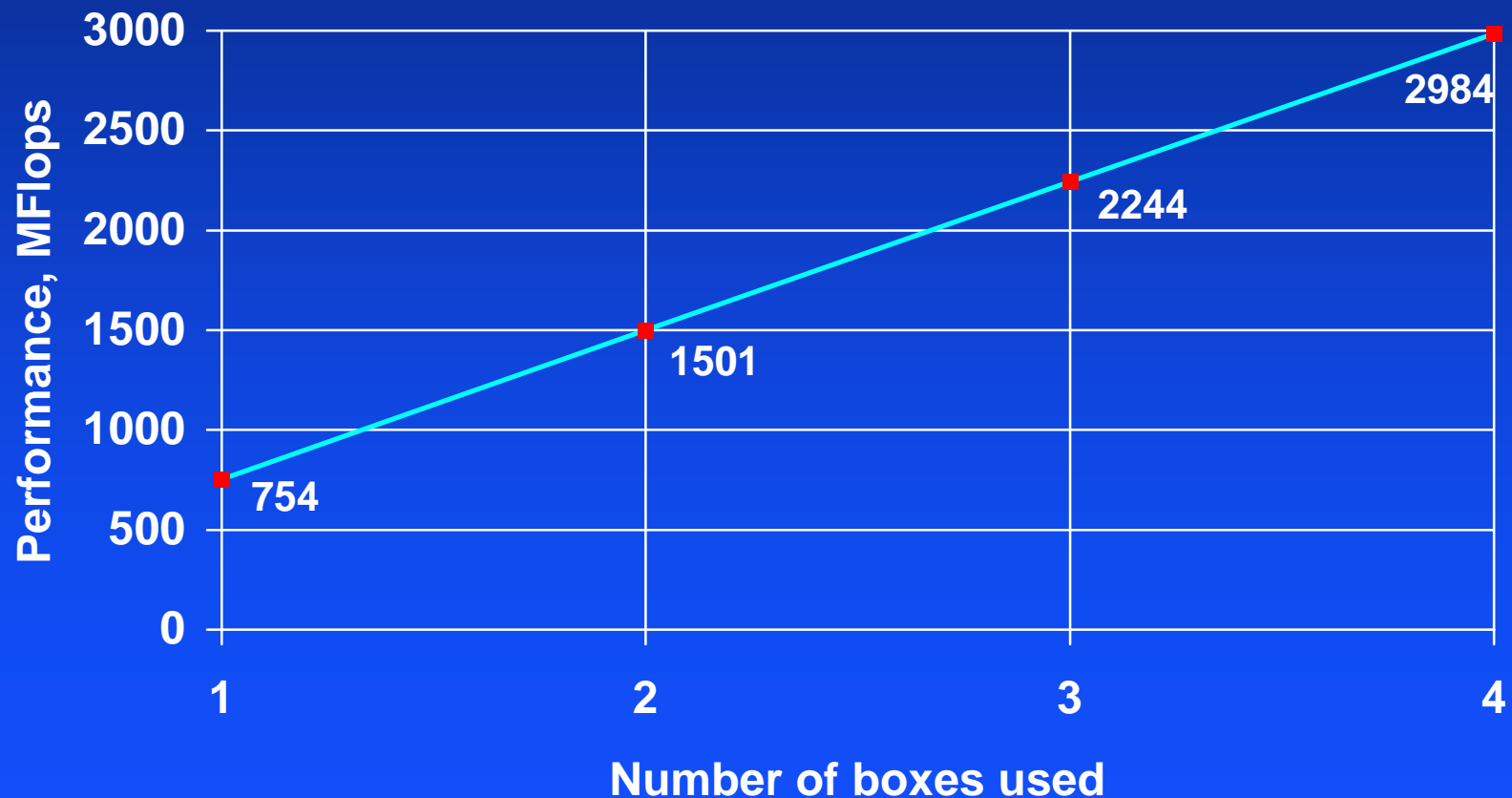
MP4(full) SMP scalability testcase

$N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C1 symmetry group



Example #3. Absolute performance in MFlops vs. number of boxes used. PC GAMESS runs in SMP mode on each box.

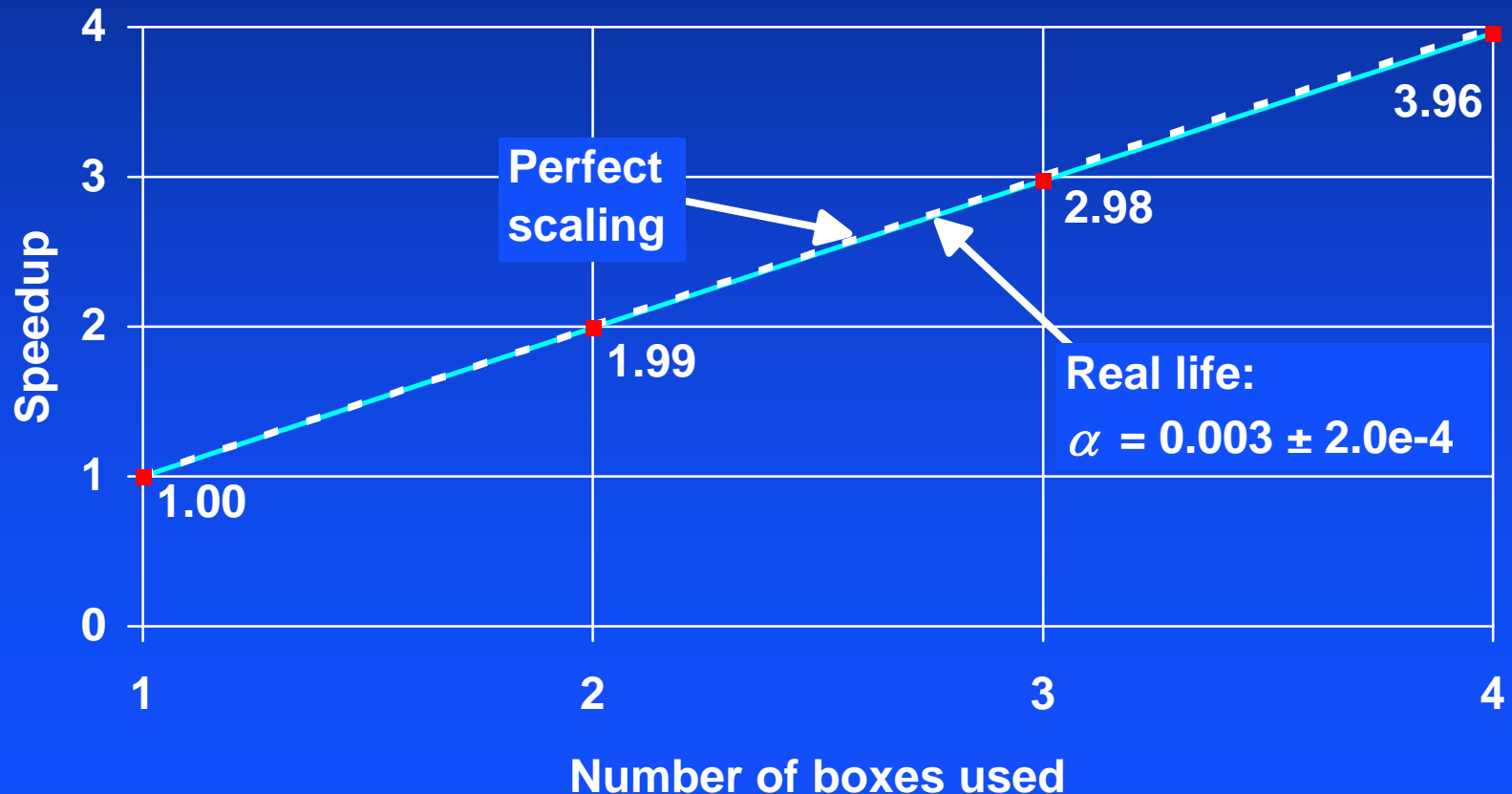
MP4(full) parallel scalability testcase
 $N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C1 symmetry group



Example #3. Parallel scaling vs. number of boxes used. PC GAMESS runs in SMP mode on each box.

MP4(full) parallel scalability testcase

$N_{\text{core}} = 10$, $N_{\text{occ}} = 34$, $N_{\text{virt}} = 193$, $N = 227$, C1 symmetry group



Questions and answers.

- The SMP version of MP4 algorithm is fully parallel. Why the SMP scaling is not perfect, especially on 8-CPU system?
 - ◆ The reason is in the memory access conflicts (different CPUs share the same memory bus).
- Should the scaling properties be better for the larger chemical problems?
 - ◆ Yes, they should be improved linearly with the number of basis functions (N) used.
- Is there any difference in performance if running PC GAMESS on Pentium III Xeon CPUs with 1 MB L2 cache and CPUs having 2 MB L2 cache?
 - ◆ For the MP4 calculations, examples #1 and #2 show that the system with CPUs having 2 MB L2 cache is only about 5% faster.
- Can we expect any serious performance degradation if using much cheaper Pentium III (not Xeon) CPUs?
 - ◆ The performance degradation is only about 5-10% for the case of MP4 calculations (although it can be larger for other tasks).

Conclusions.

1. SMP mode scaling is excellent if only 2 CPUs are used, still good for 4 CPUs, and is not very good for 8 CPUs



the most efficient and flexible solution is to use clusters built from relatively cheap dual-CPU systems.

2. Parallel mode scaling is almost perfect



we can efficiently use large (100 and more PCs) clusters for MP4 and CC-like calculations.

The PC GAMESS on the Web:

- <http://classic.chem.msu.su/gran/gamess>