tails here). The 'speedup' is the factor you gain in speed if you use n processors instead of one to solve a problem. If you have for example 20 workers (processors) working together, they will usually loose some time for communication, *i.e.* exchange of their working pieces (data), and, therefore, reach a 'speedup' of less than 20, let us assume only 12. Such a low number would result if they need to communicate significantly and are often blocking each others way. Again you could 'cheat' by giving each worker (processor) additional work which is not needed, but which forces him to sit longer at his table and do relatively less communication, making the process slower, but the speedup higher!

Finally, we would like to give an example from the real world of quantum chemistry, where people are not 'cheating' their processors, but nevertheless similar effects can be found. *Lüthi et al.* [3] reported results from a calculation on a Cray Y-MP/8-128 supercomputer with the DISCO-program for bis(2,6-dimethylphenyl) carbonate ($C_{17}O_3H_{18}$), a molecule with 38 atoms (314 contracted / 610 primitive basis functions), in which one iteration took about 400 s (this number is different from the one in [3], which was wrong, due to an input error [4]) and a performance of 1531 Mflops was achieved. The speedup for 8 processors was 7.65. *Brode* [5] has carried out a very similar calculation on the same molecule (356 contracted/592 primitive basis functions) with the TURBOMOLE-program on a workstation cluster of 14 machines performing to a maximum rate of 660 Mflops. The speedup was only 11.6. Although the loss in parallelization was higher and the Mflop-rate was much smaller (the formal rate for the 8 Cray processors would even be 2660 Mflops) the time for the first iteration (taking usually most time) was only 524 s, *i.e.* slightly more than with DISCO. Similar experiences have been made by *Vogel et.al.* [6] with a version of DISCO on a network of workstations.

Summarizing, we can state that the Mflop- and the speedup-measure are often not very useful criteria for the real world of vector- and parallel-computers, but that *a comparison between programs solving problems as similar as possible is the best way to estimate the performance of computers for a specific task.* Policies such as that of CSCS in Manno, enforcing that only programs yielding 275 Mflops should run on the NEC, are questionable in view of the difficulty to accurately estimate the efficiency of application programs running on vector processors. An alternative policy would be to supply a supplementary national cluster of workstations or a super parallel computer for codes performing badly on a vector processor making the scientists choose the most efficient facility for their purpose.

We thank Dr. *Stefan Brode*, *BASF* Aktiengesellschaft, Ludwigshafen/Rhein and Dr. *Hans Peter Lüthi*, Interdisziplinäres Projektzentrum für Supercomputing, ETH, Zürich for the example.

[1] Th. Bally, P.-A. Carrupt, J. Weber, *Chimia* **1991**, *45*, 352.
[2] R. Eggenberger, H. Huber, *Chimia* **1992**, *46*, 227.
[3] H.P. Lüthi, J.E. Mertz, M.W. Feyereisen, J.E. Almlöf, *J. Comput. Chem.* **1992**, *13*, 160.
[4] H.P. Lüthi, private communication.
[5] S. Brode, private communication.
[6] S. Vogel, J. Hutter, T.H. Fischer, H.P. Lüthi, *Int.. J. Quantum Chem.*, accepted for publication.

Chimia 47 (1993) 23–24

ISSN 0009–4293

How Computer Science is Taught to Our Students in Chemistry. Part II

Heiner G. Bührer*

In a recent survey [1], the present state of the compulsory introduction courses in computer science taught in different Swiss universities and federal institutes of technology were compared with a report called 'Recommendation for the Introduction of Computer Science in the Chemistry Curriculum' [2]. The survey did not take into account the seven chemistry departments of the Swiss Schools of Engineering (Ingenieurschulen, HTL), which confer about one third of all diplomas in chemistry in Switzerland [3]. In the following *Table* and list of contents this gap has been filled. Some conclusions can be drawn with regard to [1]:

- All Schools of Engineering offer to their chemistry students compulsory courses in the first two semesters. Most of them propose advanced courses in higher semesters. As many schools emphasise on chemical engineering, automation, and electronics are additional elements of training. The School of Engineering at Geneva is different inasmuch as its curriculum lasts five years and is meant for chemical engineering students only.
- The total number of hours per week varies among the different Schools of Engineering. Most of them devote a significant part of time to computer applications in the laboratory or pilot plant (simulations, data comprehension, processing and modelling), which appears only partially in the *Table*.
- A first introduction to computer architecture and programming is often given by a computer specialist or mathematician. Chemical applications are usually taught by a chemist.
- As is the case with university students, PASCAL is here too the preferred programming language. However, many Schools of Engineering tend to reduce the number of lessons devoted to programming for the benefit of computer applications in chemistry.
- In general, the differences between the university students' results [1] and those of students at Schools of Engineering are small. There may be less time for programming for the latter but more for chemical and technical applications; this reflects the difference in interest and requirements of the two groups of chemists.

**Correspondence*: Prof. H.G. Bührer
Technikum Winterthur Ingenieurschule TWI
Postfach 805
CH-8401 Winterthur

tails here). The 'speedup' is the factor you gain in speed if you use n processors instead of one to solve a problem. If you have for example 20 workers (processors) working together, they will usually loose some time for communication, *i.e.* exchange of their working pieces (data), and, therefore, reach a 'speedup' of less than 20, let us assume only 12. Such a low number would result if they need to communicate significantly and are often blocking each others way. Again you could 'cheat' by giving each worker (processor) additional work which is not needed, but which forces him to sit longer at his table and do relatively less communication, making the process slower, but the speedup higher!

Finally, we would like to give an example from the real world of quantum chemistry, where people are not 'cheating' their processors, but nevertheless similar effects can be found. *Lüthi et al.* [3] reported results from a calculation on a Cray Y-MP/8-128 supercomputer with the DISCO-program for bis(2,6-dimethylphenyl) carbonate ($C_{17}O_3H_{18}$), a molecule with 38 atoms (314 contracted / 610 primitive basis functions), in which one iteration took about 400 s (this number is different from the one in [3], which was wrong, due to an input error [4]) and a performance of 1531 Mflops was achieved. The speedup for 8 processors was 7.65. *Brode* [5] has carried out a very similar calculation on the same molecule (356 contracted/592 primitive basis functions) with the TURBOMOLE-program on a workstation cluster of 14 machines performing to a maximum rate of 660 Mflops. The speedup was only 11.6. Although the loss in parallelization was higher and the Mflop-rate was much smaller (the formal rate for the 8 Cray processors would even be 2660 Mflops) the time for the first iteration (taking usually most time) was only 524 s, *i.e.* slightly more than with DISCO. Similar experiences have been made by *Vogel et.al.* [6] with a version of DISCO on a network of workstations.

Summarizing, we can state that the Mflop- and the speedup-measure are often not very useful criteria for the real world of vector- and parallel-computers, but that *a comparison between programs solving problems as similar as possible is the best way to estimate the performance of computers for a specific task.* Policies such as that of CSCS in Manno, enforcing that only programs yielding 275 Mflops should run on the NEC, are questionable in view of the difficulty to accurately estimate the efficiency of application programs running on vector processors. An alternative policy would be to supply a supplementary national cluster of workstations or a super parallel computer for codes performing badly on a vector processor making the scientists choose the most efficient facility for their purpose.

We thank Dr. *Stefan Brode*, *BASF* Aktiengesellschaft, Ludwigshafen/Rhein and Dr. *Hans Peter Lüthi*, Interdisziplinäres Projektzentrum für Supercomputing, ETH, Zürich for the example.

[1] Th. Bally, P.-A. Carrupt, J. Weber, *Chimia* **1991**, *45*, 352.
[2] R. Eggenberger, H. Huber, *Chimia* **1992**, *46*, 227.
[3] H.P. Lüthi, J.E. Mertz, M.W. Feyereisen, J.E. Almlöf, *J. Comput. Chem.* **1992**, *13*, 160.
[4] H.P. Lüthi, private communication.
[5] S. Brode, private communication.
[6] S. Vogel, J. Hutter, T.H. Fischer, H.P. Lüthi, *Int.. J. Quantum Chem.*, accepted for publication.

Chimia 47 (1993) 23–24

ISSN 0009–4293

How Computer Science is Taught to Our Students in Chemistry. Part II

Heiner G. Bührer*

In a recent survey [1], the present state of the compulsory introduction courses in computer science taught in different Swiss universities and federal institutes of technology were compared with a report called 'Recommendation for the Introduction of Computer Science in the Chemistry Curriculum' [2]. The survey did not take into account the seven chemistry departments of the Swiss Schools of Engineering (Ingenieurschulen, HTL), which confer about one third of all diplomas in chemistry in Switzerland [3]. In the following *Table* and list of contents this gap has been filled. Some conclusions can be drawn with regard to [1]:

- All Schools of Engineering offer to their chemistry students compulsory courses in the first two semesters. Most of them propose advanced courses in higher semesters. As many schools emphasise on chemical engineering, automation, and electronics are additional elements of training. The School of Engineering at Geneva is different inasmuch as its curriculum lasts five years and is meant for chemical engineering students only.
- The total number of hours per week varies among the different Schools of Engineering. Most of them devote a significant part of time to computer applications in the laboratory or pilot plant (simulations, data comprehension, processing and modelling), which appears only partially in the *Table*.
- A first introduction to computer architecture and programming is often given by a computer specialist or mathematician. Chemical applications are usually taught by a chemist.
- As is the case with university students, PASCAL is here too the preferred programming language. However, many Schools of Engineering tend to reduce the number of lessons devoted to programming for the benefit of computer applications in chemistry.
- In general, the differences between the university students' results [1] and those of students at Schools of Engineering are small. There may be less time for programming for the latter but more for chemical and technical applications; this reflects the difference in interest and requirements of the two groups of chemists.

**Correspondence*: Prof. H.G. Bührer
Technikum Winterthur Ingenieurschule TWI
Postfach 805
CH-8401 Winterthur

Table

School of Engineering	Compulsory Cours	Hours/Week Lectures and Exercises	Total	Cours given by	Audience	Remarks
Burgdorf	1.+2.+3. semester	4+8	12	mathematician, chemist	chemists	–
Chur	1.+2.+3. semester	2	6	computer specialist	chemists	–
Fribourg	1.+2.+3. semester	2+1	13	computer specialist, chemist	chemists	additional courses in automation
Genève	a)	a)	a)	a)	a)	a)
Muttenz	1.+2.+5. semester	2+3	5	mathematician, chemist	chemists	–
Sion	1. + 2. semester	4	8	computer specialist	chemists, agriculturists, biotechnologists	–
Winterthur	1.+2. semester	2	4	computer specialist, chemist	chemists	new curriculum

a) Not comparable; the general computer training is part of a two year course preceeding chemical engineering studies.

Contents:

Burgdorf: Introduction to computer architecture and WINDOWS user interface, general application software, chemical data bases, molecular modelling

Chur: Introduction to computer architecture, operating system MS-DOS, introduction to programming (PASCAL), numerical methods, data comprehension and processing; general application software (word-processing, spread-sheet, data bases)

Fribourg: Introduction to programming, general application software (WINDOWS programs), data comprehension, processing and modelling, molecular design

Genève: Introduction to computer science and laboratory applications

Muttenz: Introduction to computer architecture and to an operating system, introduction to programming, general application software, overview of computer applications in chemistry

Sion: Introduction to computer science, introduction to programming, Excel

Winterthur: Introduction to computer architecture, networks and operating systems, general application software (chemical word-processing, spreadsheets), introduction to molecular modelling, chemical data bases

[1] *Chimia* **1992**, *46*, 447.
[2] *Chimia* **1988**, *42*, 199.
[3] H. G. Bührer, *Chimia* **1992**, *46*, 6.

ANNOUNCEMENTS

29. Symposium für Theoretische Chemie

Oberwiesenthal (Sachsen), BRD, 28. Sept.–1. Okt. 1993

Organisiert durch *Joachim Reinhold*, Fachbereich Chemie, Universität Leipzig, Talstr. 35, O–7010 Leipzig, Deutschland
EM: reinhold@theorie.chemie.uni-leipzig.dbp.de

European Summer School in Quantum Chemistry (ESQC)

Tjörnarps Kursgård, Sweden, 15.–28. August 1993

Organized by *Björn Roos*, Dept. of Theoretical Chemistry, Chemical Centre, P.O.B. 124, S–221 00 Lund, Sweden
EM: teobor@garm.teokem.lu.se