Computer-Aided Engineering

Advantages of Supercomputers For Engineering Applications

The supercomputer computes faster and has more central memory than conventional computers. Its high performance can be utilized to solve complex computation, design, and simulation problems instantly.

Mark A. Stadtherr and James A. Vegeais, Univ. of Illinois, Urbana, Ill. 61801

A supercomputer is a general-purpose machine that computes faster and has more central memory capability than conventional ones. The best known examples are the Cray-1, the CDC Cyber 205, and the Cray X-MP. Table 1 shows a comparison of the efficiencies of the Cray-1, a personal computer (IBM PC), a superminicomputer (VAX 11/780), and a conventional mainframe computer (CDC Cyber 175). The figures for peak speed are approximate and meant only for order of magnitude comparisons. A machine will rarely perform at this peak speed for any sustained period. The significantly higher speed of the supercomputer can be realized only if the program and algorithm take advantage of the supercomputer architecture.

Table 2 shows a comparison of the speed of several machines in an actual application—solving dense systems of linear equations of order 100 using LINPACK, a widely used set of Fortran subroutines for linear algebra problems. The data (*1*) are for the case of full precision—approximately 64-bit arithmetic. In these problems, most of the floating point operations are done in the basic linear algebra subroutines (BLAS). For the supercomputer runs, the BLAS were coded in assembly language or their Fortran code was modified to take advantage of the supercomputer architecture. No attempts were made to otherwise change the Fortran code for LINPACK to exploit the supercomputer architecture.

Speeds for the conventional machines can be improved somewhat by coding the BLAS in assembly language. These speeds vary slightly depending on the operating system and compiler used. All the speeds in Table 2, especially those for the supercomputers, may be somewhat higher on larger problems, as the asymptotic execution rates are approached. The figure for the Cray X-MP is for the case of a single processor.

Table 1. Peak speed and typical central memory capability.

Computer	Peak Speed, MFLOPS*	Central Memory, Bytes
IBM PC	0.05	256
VAX 11/780	0.3	1M
Cyber 175	10	1M
Cray-1	240	8–32M

* MFLOPS = millions of floating-point operations per second

Table 2. Performance of computers in actual applications (*1*).

Supercomputers	BLAS* Assembly Language	BLAS* Modified Fortran
Cray X-MP	33 MFLOPS	21 MFLOPS
Cyber 205	25 MFLOPS	8.4 MFLOPS
Cray-1S	23 MFLOPS	12 MFLOPS

Other Computers	BLAS* Unmodified Fortran
Cyber 175	2.1
IBM 3081	2.0
FPS-164	1.3
VAX 11/780	0.13
IBM PC/AT	0.0091

* BLAS = basic linear algebra subroutines

For steady-state process flowsheeting and optimization, supercomputing will provide not only the ability to more realistically handle units involving complex phenomena, but also truly interactive design capabilities—most likely involving sophisticated graphics interfaces for input and output.

Concurrent vs. sequential

The most significant architectural differences between the conventional machine and the supercomputer (2) is the ability of the supercomputer to perform several computational steps concurrently or in parallel. Conventional "sequential" computers must carry out steps one at a time, though the fastest sequential computers usually include some rudimentary form of "pipelining."

Machines such as the Cray-1, Cyber 205, and Cray X-MP are often called vector machines , because they use a form of parallelism known as *vector processing*. The basic design idea of vector machines is to speed up the execution of vector operations. This includes use of vector instructions in the instruction set used by the instruction processor. This allows one to replace loops by providing single instructions that perform operations on whole vectors, thus eliminating some of the overhead associated with a loop. More important, however, is the use of *multiprocessing* and *pipelining*; both involve provisions for performing parts of the overall vector operation concurrently.

Pipelining operates like an assembly line (2). A floating point operation involves several steps. Without pipelining, all the steps needed to complete one operation would be performed before starting the first step of the next operation; thus, the computer works on one operation at a time. For a particular operation in a highly pipelined computer, however, there are several "stations," each of which performs only one part of the overall operation. Since, as in an assembly line, all stations are working concurrently, the computer can work on more than one similar operation at the same time. The Cray-1 and Cyber 205 are both highly pipelined machines, though each operates somewhat differently. For instance, the Cyber 205 operates most efficiently on very long vectors, while the Cray-1 can operate efficiently even on relatively short vectors.

Multiprocessing basically involves putting several processors in parallel. For instance, The Cray X-MP will ultimately employ up to 16 processors in parallel, each highly pipelined. Thus one could conceivably run 16 separate programs simultaneously or divide a single program among the processors. Techniques for doing the latter are still relatively undeveloped. It is generally agreed, however, that future advances in supercomputing technology will rely significantly on the use of some form of multiprocessing, and several possible configurations of processors are being studied. While multiprocessing can be used for vector processing, it is more versatile than pipelining, and can be used to speed up computations that do not *vectorize*, i.e., computations not amenable to vector processing.

Even though microcomputer technology has constantly improved in speed and memory, this has generally not been seen on the other end of the computer size spectrum. Machines ranging from superminis, through mainframes, to supercomputers have been on a relatively constant plateau of speed and memory size for several years, despite advances in chip technology. This trend is now changing, at least with respect to supercomputing, Table 3. Supercomputers two orders of magnitude faster than current ones seem possible by the end of the decade, though still at very high cost—typically in excess of $10 million.

Perhaps of even more impact is that the technology now exists to mass-produce supercomputers comparable to the current Cray-1 for under $1 million each. One of the first examples along these lines is the recently announced Convex

Table 3. Advances in state-of-the-art supercomputers.

Supercomputer	Date	Approx. Peak Throughput (Mflops)	Memory (Mwords)
Cray-1	1976	240	1–4
Cray X-MP (1 Processor)	1983	400	4
Cray X-MP (16 Processor)	198x	6400	32
Cray-2 (1 Processor)	1985	1000	128
Cray-2 (4-Processor)	198x	4000	

Table 4. Performance of minisupercomputer (*6*)

Computer	Approx. Cost ($M)	Approx. Peak Speed (MFLOPS)
Cray-1	5–15	240
C-1	0.5	60
VAX 11/780	0.1	0.3

> **While off-line simulations can be used to improve control system design and plant operability, supercomputers can also be used to do realistic on-line simulations of complex processes in real time. When incorporated in a control system, such real-time simulations may provide significantly better control.**

C-1. This is a machine based on supercomputer architecture, with a peak speed advertised as one-fourth that of the Cray-1, but at a cost of only $500,000, Table 4. Nobel laureate Kenneth Wilson (*3*) expects that competition will cause prices for such machines to eventually drop to not much higher than those of the widespread superminis. An explosion in supercomputing technology seems to be on the horizon.

Supercomputers for modeling and analysis

The availability of supercomputers presents a number of opportunities to the chemical engineering community, both industrial and academic. Supercomputers provide the following opportunities:

1. To solve problems involving the modeling and analysis of complex physical phenomena that were previously intractable or at least computationally infeasible.
2. To greatly increase engineering design productivity in areas requiring large-scale computation.
3. To use complex models in real-time applications.

Supercomputer applications involving modeling and analysis of complex physical phenomena include:

- Reservoir simulation
- Weather forecasting
- Nuclear weapons research
- Molecular dynamics
- Astrophysics
- Quantum chemistry
- High-energy physics
- Molecular biology
- Turbulent flow phenomena
- Combustion
- Reaction kinetics
- Rheology of coating flows
- Prediction of physical properties
- Reaction injection molding

These problems typically require computationally intensive solution methods such as Monte Carlo simulations or the discretization of partial differential equations by finite difference or finite element techniques. Problem-solving with the available computing power may mean limiting the problem in terms of dimensionality, resolution, or the physical assumptions made. The supercomputer is capable of relaxing these limitations. Thus, one can solve problems in more dimensions, using higher resolution (e.g., a finer mesh in PDE problems), and using more complex models that more accurately represent the true physics and chemistry of the phenomena. Even on a supercomputer, such problems as long-range weather forecasting could require several hundred hours of computation time and still not provide the resolution desired.

Many of the current and future applications of this type in chemical engineering involve problems in which rapid changes occur with respect to position or time, especially when combinations of phenomena occur, e.g. simultaneous chemical reaction, fluid flow, heat transfer, and/or mass transfer. The supercomputer provides the chemical engineer the opportunity to study such phenomena in much more detail than previously possible.

Supercomputers increase design productivity

Supercomputer applications in design include:

- *Nuclear engineering*: design of reactor, and safety and emergency systems.
- *Automotive engineering*: aerodynamics (drag reduction), CAD, structural analysis, and combustion.
- *Electrical engineering*: VLSI design.
- *Aerospace engineering*: CAD, aerodynamics, structural analysis, and propulsion.
- *Chemical process engineering*: Relatively little use so far.

Our discussion on the supercomputer applications to engineering design focus on increasing design productivity, not so much on solving problems that previously were computationally infeasible. Many CAD problems can be solved without a supercomputer; but the supercomputer helps the design engineer get very rapid feedback on his design changes and easily view the results by its very fast and high–resolution graphics.

Supercomputers have not been used much in chemical process engineering. An obvious application is in the design of single process units involving complex physical phenomena discussed earlier. Relatively cheap supercomputing is expected to be available; thus, the supercomputer will have a major impact in the areas of process flowsheeting and optimization, and in process control, provided that practitioners take advantage of its architecture.

For steady-state process flowsheeting and optimization, supercomputing will provide not only the ability to more realistically handle units involving complex phenomena, but also truly interactive design capabilities—most likely involving sophisticated graphics interfaces for input and output. The engineer will be able to make a design change in a complex process and almost immediately see the simulated result. In terms of actual wall-clock time, such complex simulations today often require several minutes even on a relatively fast conventional machine and perhaps several hours on an engineer's personal microcomputer. Supercomputing will greatly increase the productivity of the design engineer.

The design engineer will work with a powerful local workstation networked to a supercomputer. Most routine work will be done entirely on the workstation, but for a computationally intensive problem the supercomputer will take over. The supercomputer will also have a major impact on the ability to do realistic simulations of process dynamics. Realistic dynamic simulations of complex process units or entire plants can be used in process operability studies, simulations of process safety, and the design of control systems. Even though supercomputers are used only in few chemical engineering applications along these lines, applications are likely to expand.

Real-time computing

Real-time computing applications are not well developed, but may ultimately provide a very large market for supercomputer power. Applications of this type include:

- Avionics (aircraft control and instrumentation).
- Robotics (robot vision and control).
- Speech recognition.
- Signal processing.
- Process control.

In chemical process engineering, the application of most interest is process control. While off-line simulations can be used to improve control system design and plant operability, supercomputers can also be used to do realistic on-line simulations of complex processes in real time. When incorporated in a control system, such real-time simulations may provide significantly better control. Real-time simulations of entire plant complexes using supercomputers provide a powerful tool for on-line optimization of complete plant operations.

Fully utilize supercomputer architecture

Supercomputers have the potential to considerably increase computational speed relative to the conventional machine. The amount by which speed can be increased depends on the problem to be solved: how the problem is formulated, what strategies are used to solve the problem, and how the computer code is written. Some problems, formulations, algorithms, and codes are able to take better advantage of the supercomputer architecture than others. If this architecture is not well exploited, a doubling of speed is all that might be expected; if fully exploited, speed may be increased by a factor of 20 or much more. Choosing a problem formulation and a solution algorithm, and providing a code that fully exploits the supercomputer may represent considerable challenges.

Most supercomputer codes are written in Fortran. However, a code that has been running on a conventional machine will not usually run too much faster on a supercomputer. Existing Fortran compilers for supercomputers will make some attempts to produce vectorized object code to take advantage of the supercomputer architecture, but at present better results can usually be obtained if the user vectorizes his source code. Much better vectorizing compilers can be expected in the future. However, for some algorithms and problem formulations, little vectorization is possible no matter how the code is written. In such cases, the challenge is to provide a problem formulation or solution algorithm that better exploits the supercomputer.

Some programs work well on conventional machines, but may not be the best on a supercomputer. When the ASPEN flowsheeting system was implemented on a Cray-1, Duerre and Bumb (*4*) found that it ran only two to three times faster than on an IBM 370, which indicates that little advantage was taken of the supercomputer architecture. This is not unexpected since little attempt was made to vectorize the code or optimize input-output operations. Duerre and Bumb suggest optimizing the code, but it is unclear to what extent the performance of the program can be improved in this way.

In the equation-based approach to process flowsheeting, a key computational step is the solution of a large sparse system of linear equations. In general, it requires the use of a general-purpose sparse linear equation solver; however, the algorithms used on conventional machines in such general-purpose routines are not easily vectorizable. Therefore, it is difficult to take advantage of the vector processing capability of the supercomputer. This is clearly seen in some comparisons by Duff and Reid (*5*).

A general-purpose *full* matrix routine, compiled and executed on a Cray-1, was about 20 to 30 times faster than the same routine compiled and executed on an IBM 3033. On the other hand, when the same comparison was made using a general-purpose *sparse* matrix routine, it was only about two times faster on the Cray-1, simply because the scalar speed of the Cray-1 is about twice that of the IBM 3033.

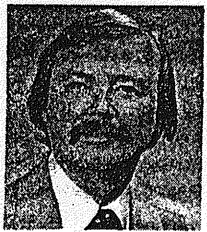
Clearly very little vectorization of the sparse matrix routine was possible, largely because of the amount of indirect addressing in sparse matrix codes. "This is very disappointing since there is no easy fix that can give us better vectorization. We are, therefore, forced to rethink our sparse matrix algorithms with the Cray architecture in mind (*5*)." To make the most effective use of supercomputers in chemical engineering computing, we must rethink the problem-solving strategies used.

In conclusion

The availability of supercomputers such as the Cray-1, Cyber 205, and Cray X-MP presents a number of opportunities to the chemical engineering community, both industrial and academic. Among these are: the opportunity to solve once intractable problems involving the modeling and analysis of complex physical phenomena; the opportunity to greatly increase engineering productivity in areas requiring large-scale computation; and the opportunity to use complex models in real-time applications. The challenges include the development of computational algorithms and codes, and problem formulations that fully exploit the special architecture of these machines. #

Literature cited

1. Dongarra, J. J., "Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment," Technical Memorandum No. 23, Argonne National Laboratory (1984).
2. Levine, R. D., "Supercomputers," *Scientific Amer.*, 246(1), p. 118 (Jan., 1982).
3. Wilson, K., "Supercomputing: Cooperation with Industry," *SIAM NEWS*, 18(5), p. 6 (Sept., 1984).
4. Duerre, K. H., and A. C. Bumb, "Implementing ASPEN on the Cray Computer," Report LA-UR-81-3528, Los Alamos National Laboratory, Los Alamos, N.M. (1981).
5. Duff, I. S., and J. K. Reid, "Experience of Sparse Matrix Codes on the Cray-1," *Comput. Phys. Comm.*, 26, 293 (1982).
6. Davis, D. B., "Bridging the Gap between Supermini and Supercomputer," *High Technol.*, p. 72 (Feb., 1985).



M.A. Stadtherr holds a B.Ch.E. degree from the Univ. of Minnesota and a Ph.D. degree in chemical engineering from the Univ. of Wisconsin. At present, he is on the chemical engineering faculty at the Univ. of Illinois at Urbana-Champaign. He has authored or co-authored several articles on process flowsheeting and optimization.

J.A. Vegeais earned his B.S.Ch.E. degree at the Univ. of Illinois at Chicago Circle and his M.S.Ch.E. degree at the Univ. of Illinois at Urbana-Champaign. At present, he is completing his Ph.D. degree in chemical engineering at the Univ. of Illinois at Urbana-Champaign.