Technical Note

REEVALUATING AMDAHL'S LAW

JOHN L. GUSTAFSON

At Sandia National Laboratories, we are currently engaged in research involving massively parallel processing. There is considerable skepticism regarding the viability of massive parallelism; the skepticism centers around *Amdahl's law*, an argument put forth by Gene Amdahl in 1967 [1] that even when the fraction of serial work in a given problem is small, say, s, the maximum speedup obtainable from even an infinite number of parallel processors is only $1/s$. We now have timing results for a 1024-processor system that demonstrate that the assumptions underlying Amdahl's 1967 argument are inappropriate for the current approach to massive ensemble parallelism.

If N is the number of processors, s is the amount of time spent (by a serial processor) on serial parts of a program, and p is the amount of time spent (by a serial processor) on parts of the program that can be done in parallel, then Amdahl's law says that speedup is given by

$$\begin{aligned} Speedup &= (s+p)/(s+p/N) \\ &= 1/(s+p/N), \end{aligned}$$

where we have set total time $s + p = 1$ for algebraic simplicity. For $N = 1024$ this is an unforgivingly steep function of s near $s = 0$ (see Figure 1).

The steepness of the graph near $s = 0$ (approximately $-N^2$) implies that very few problems will experience even a 100-fold speedup. Yet, for three very practical applications (s = 0.4–0.8 percent) used at Sandia, we have achieved speedup factors on a 1024-processor hypercube that we believe are unprecedented [2]: *1021* for beam stress analysis using conjugate gradients, *1020* for baffled surface wave simulation using explicit finite differences, and *1016* for unstable fluid flow using flux-corrected transport. How can this be, when Amdahl's argument would predict otherwise?

The expression and graph both contain the implicit assumption that p is independent of N, which *is virtually never the case*. One does not take a fixed-sized problem and run it on various numbers of processors

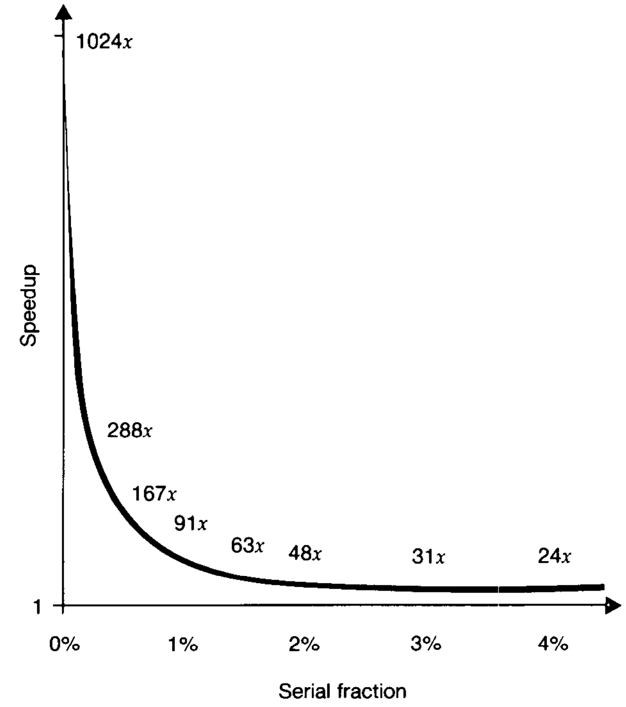

FIGURE 1. Speedup under Amdahl's Law

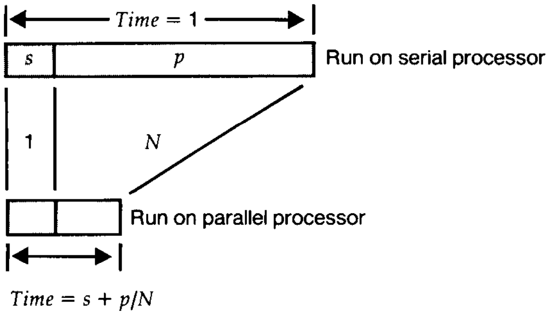

FIGURE 2a. Fixed-Sized Model for *Speedup* **= 1/(s + p/N)**

except when doing academic research; in practice, *the problem size scales with the number of processors.* When given a more powerful processor, the problem generally expands to make use of the increased facilities. Users have control over such things as grid resolution, number of time steps, difference operator complexity, and other parameters that are usually adjusted to allow the program to be run in some desired amount of time. Hence, it may be most realistic to assume *run time*, not *problem size*, is constant.

As a first approximation, we have found that it is the *parallel* or *vector* part of a program that scales with the problem size. Times for vector start-up, program loading, serial bottlenecks, and I/O that make up the s component of the run do *not* grow with problem size. When we double the number of degrees of freedom in a physical simulation, we double the number of processors. But this means that, as a first approximation, the amount of work that can be done in parallel *varies linearly with the number of processors.* For the three applications mentioned above, we found that the parallel portion scaled by factors of 1023.9969, 1023.9965, and 1023.9965. If we use s and p to represent serial and parallel time spent on the *parallel* system, then a serial processor would require time $s + p \times N$ to perform the task. This reasoning gives an alternative to Amdahl's law suggested by E. Barsis at Sandia:

$$\begin{aligned}\textit{Scaled speedup} &= (s + p \times N)/(s + p)\\ &= s + p \times N\\ &= N + (1 - N) \times s.\end{aligned}$$

In contrast with Figure 1, this function is simply a *line*, and one with a much more moderate slope: $1 - N$. It is thus much easier to achieve efficient parallel performance than is implied by Amdahl's paradigm. The two approaches, fixed sized and scaled sized, are contrasted and summarized in Figure 2a and b.

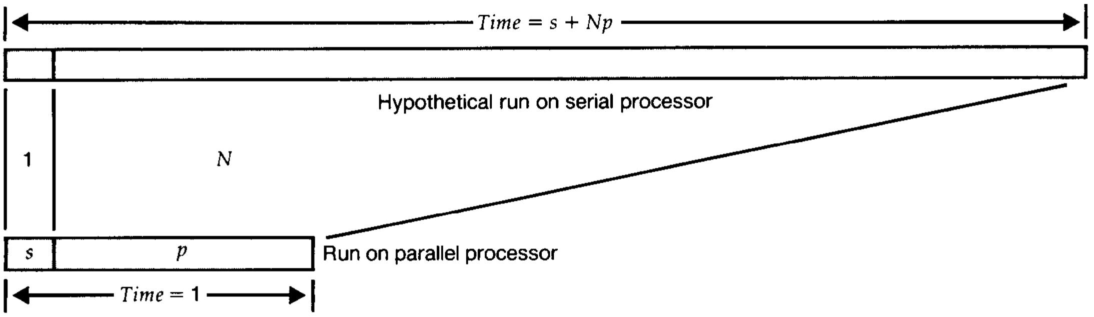
Time = s + Np
Hypothetical run on serial processor
1
N
s
p
Run on parallel processor
Time = 1

FIGURE 2b. Scaled-Sized Model for *Speedup* **= s + Np**

Our work to date shows that it is *not* an insurmountable task to extract very high efficiency from a massively parallel ensemble, for the reasons presented here. We feel that it is important for the computing research community to overcome the "mental block" against massive parallelism imposed by a misuse of Amdahl's speedup formula; speedup should be measured by scaling the problem to the number of processors, not by fixing problem size. We expect to extend our success to a broader range of applications and even larger values for N.

REFERENCES

1. Amdahl, G.M. Validity of the single-processor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings*, vol. 30 (Atlantic City, N.J., Apr. 18–20). AFIPS Press, Reston, Va., 1967, pp. 483–485.
2. Benner, R.E., Gustafson, J.L., and Montry, R.E. Development and analysis of scientific application programs on a 1024-processor hypercube. SAND 88-0317, Sandia National Laboratories, Albuquerque, N.M., Feb. 1988.

CR Categories and Subject Descriptors: C.1.2 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)—*parallel processors*

General Terms: Theory

Additional Key Words and Phrases: Amdahl's law, massively parallel processing, speedup

Author's Present Address: John L. Gustafson, Sandia National Laboratories, Albuquerque, NM 87185.

APPENDIX: Pearson Correlation Coefficients ($N = 209$)

	minimum memory	maximum memory	cache memory	minimum channels	maximum channels
1/cycle time	.695 $P = .000$	.656 $P = .000$	.485 $P = .000$	.490 $P = .000$	.240 $P = .000$
minimum memory		.758 $P = .000$	.535 $P = .000$	.517 $P = .000$	.267 $P = .000$
maximum memory			.538 $P = .000$	.561 $P = .000$	.527 $P = .000$
cache memory				.582 $P = .000$	.488 $P = .000$
minimum channels					.548 $P = .000$

REFERENCES

1. Datapro Research Corporation. *Datapro 70*. Delran, N.J., 1981.
2. Ein-Dor, P. Grosch's law re-revisited: CPU power and the cost of computation. *Commun. ACM 28*, 2 (Feb. 1985), 142–151.
3. Ein-Dor, P., and Feldmesser, J. Attributes of the performance of central processing units: A relative performance prediction model. *Commun. ACM 30*, 4 (Apr. 1987), 308–317.

ONCE AGAIN, AMDAHL'S LAW

In his technical note, "Reevaluating Amdahl's Law," in the May 1988 *Communications* (pp. 532–533), John L. Gustafson makes some interesting points about how Amdahl's law can be misapplied, leading to some unnecessarily pessimistic predictions of the potential efficiency of parallel computers. Amdahl's law gives the expected speedup that would result from using a given number of processors on a problem having a given percentage of serial work. This law is sometimes interpreted as placing severe limits on attainable efficiency when using a very large number of processors. On the other hand, very high speedups and efficiencies have been reported in recent work at Sandia National Laboratories [1], prompting the mistaken impression in popular press accounts that Amdahl's law has been violated. The interpretation of Amdahl's law as an absolute limit on attainable speedup is valid for problems in which the percentage of serial work is fixed, but Amdahl's law says nothing about how the percentage of serial work might vary with problem size, which would be highly problem dependent. The pessimistic conclusion that Gustafson decries is based on the common practice of observing the declining efficiency that results from solving a problem of fixed size, with a fixed percentage of serial work, on an increasing number of processors.

Gustafson's principal observation is that one usually solves ever larger problems as more processors and memory become available, and that for many problems the percentage of serial work decreases as the problem size increases. Such circumstances indeed permit high efficiencies, but not in violation of Amdahl's law. If the problem is made large enough, and this causes the serial percentage to become small enough, then the speedup (as given by Amdahl's law) can be made arbitrarily close to the number of processors used. The notion that Amdahl's law has somehow been violated is encouraged by an error in Gustafson's paper. He states that the applications at Sandia have serial percentages of 0.4 to 0.8 percent, which by Amdahl's law would limit speedup to no more than 250, yet he reports speedups as high as 1021 on 1024 processors. The serial percentages stated are apparently for smaller versions of the problems, not for the much larger versions used for the runs that attained the reported speedups. These much larger problems had appropriately small serial percentages [1], as required by Amdahl's law. In a sense, Gustafson fixes the problem size, which he does not want to do.

As an alternative to fixing the problem size, Gustafson proposes that a more appropriate practice would be to fix the execution time. The rationale is that as users are given more processors they will solve larger problems, but will want to maintain the same turnaround time for the solution. This scaling of problem size as the number of processors grows may seem natural and appealing, but it is not what was actually done in the reported work at Sandia: in attaining the very high efficiencies reported, the size of their problems grew even faster than the number of processors, so the total execution time showed an overall increase as the number of processors increased [1]. Indeed, recent theoretical work of Worley [2] shows that for problems of the type used in the Sandia work, if only a fixed amount of execution time is allowed, then the problem size cannot grow without bound, regardless of the number of processors used. Thus, just as with fixing the problem size, fixing the execution time ultimately places limits on attainable efficiency. These limits did not affect the high speedups reported in [1], since neither the problem size nor the execution time was fixed, but rather the amount of memory used per processor. The Sandia results using explicit methods for finite-difference discretizations of time-dependent partial differential equations are a case in point: in keeping the number of grid points per processor fixed as the number of processors grows, the spatial resolution of the grid becomes finer, which in turn requires a smaller time step (due to stability constraints), so that the total execution time to reach the same simulation time grows.

One can easily imagine problems having a natural, fixed size that cannot be easily or arbitrarily increased (e.g., stability analysis of a bridge or truss of a given structure, the U.S. census, etc.). One can also easily imagine problems for which a fixed limit on execution time is appropriate (chess moves, weather prediction, tracking incoming missiles, etc.). One can also imagine problems for which neither of these constraints is reasonable. In view of the diversity of problems to which multiprocessors may be applied, there seems to be no single "best method or fixed set of assumptions for characterizing potential efficiency, but Amdahl's law, when correctly applied, continues to provide helpful insight."

Michael Heath and Patrick Worley
Oak Ridge National Laboratory
P.O. Box 2009
Oak Ridge, TN 37831-8083

AUTHOR'S RESPONSE

The main thrust of the technical note was that Amdahl's law lacks *predictive value.* We deplore the popular press accounts that have claimed that the law has been violated. In the late 1800s, the law of gravity was used by learned scientists to "prove" that heavier-than-air craft could never fly. Yet, the Wright Brothers flew without the benefit of a waiver for the laws of physics. After a time, people realized that the law of gravity was not an insurmountable obstacle to flight, and never had been.

When we ran our parallelized applications on a *single processor,* we did see serial components ranging from 0.4 to 0.8 percent. This included the overhead of message passing, program startup, I/O, etc., that some authors place in a separate term in the formula. Now suppose that we wish to use this to predict how well 1024 processors would run the application. Amdahl's law only tells us what will happen for a fixed problem size, and we want to know what would happen in the practical situation where the problem size grows with available computing power. Since we have a 1024-processor computer at Sandia, we can measure the time and hence the speedup. From the speedup, say, of 1020 on the Wave Mechanics problem, one can work backward using Amdahl's law to find that the serial component s must have been three parts per million. I do not see how Amdahl's law could have told me this would be the case before I did the experiment, especially since the serial run would have required four months and single-processor access to 300 MBytes of memory. The alternative formula proposed allows predictions when the problem size is varied, and we think that is useful. I did not "trip over my own point," since the act of stating the uniprocessor serial percentage does *not* fix the problem size in our scaled speedup paradigm.

The reported Sandia speedups were based on problem scaling instead of fixed time. We recognize *three* paradigms at Sandia: fixed sized problem, fixed time problem, and scaled problem. Our work thoroughly explored (and published) the two-parameter space of problem size versus number of processors, and all three paradigms exist as linear subsets of that space. For fixed size problems, our speedups were 502, 519, and 637. For fixed time problems, the speedups were 987, 988, and 1009. For scaled problems, the speedups were 1009, 1019, and 1020 (the last three numbers differ from those originally given in the technical note because of improved accounting for best-serial versus best-parallel algorithmic work). For many problems, such as fast Fourier transforms, particle-in-cell methods, sorting, and multigrid solvers, the operation count only grows slightly faster than the required storage, so fixed time and scaled paradigms are almost indistinguishable. For the three applications mentioned, the operation count grows as the 3/2 power of the number of variables, so the number of variables per processor shrinks by a factor of $1024^{1/3} \approx 10$ if we fix execution time as processors are added. This implies a limit, since we cannot have less than one variable per processor. For the Sandia applications, this limit is reached at 50 trillion processors. So we are aware of the difference between scaled speedup and fixed time speedup, but thought the discussion was beyond the scope of the technical note.

Lastly, I disagree with Heath and Worley's concluding paragraph on both its specific and its general statements. Neither the stability analysis of a bridge nor the U.S. census have a natural, fixed size. Stability analysis of a bridge can range from a ball-and-stick model using simple arithmetic to a full three-dimensional dynamic nonlinear finite element model, depending on the computing power available. The U.S. census is a perfect example of a fixed time model, since it must be finished before the next census is begun (every 10 years). It was in fact this fixed time constraint that inspired Herman Hollerith to create his tabulating machine in the late 1800s. Now that we have fast computers for doing the census, the census has expanded from a simple head count to a complicated survey that assesses income levels, housing conditions, and ethnic makeup of the population. Yes, there probably are problems with a fixed size, but they tend to be the trivial ones instead of those on the leading edge of scientific and engineering research. Generally, I cannot agree that "Amdahl's law, when correctly applied, continues to provide helpful insight" when the variables in the formula can only be obtained after the experiment has been done.

John Gustafson
Sandia National Laboratories
Albuquerque, NM 87185

CR Categories and Subject Descriptors: C.1 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)—*parallel processors*

General Terms: Theory

Additional Key Words and Phrases: Amdahl's law, massively parallel processing, speedup

REFERENCES

1. Gustafson, J.L., Montry, G.R., and Benner, R.E. Development of parallel methods for the a 1024-processor hypercube. *SIAM J. Sci. Statist. Comput.* 9, 4 (July 1988), 609–638.
2. Worley, P.H. Information requirements and the implications for parallel computation, Ph.D. dissertation, Computer Science Dept., Stanford University, Stanford, Calif., June 1988.

Calls for Papers

(*continued from p. 284*)

May 15
Supercomputing 89, Reno, Nev., November 13–17. For paper requirements, contact Gary Johnson, GA Technologies, Inc., P.O. Box 85608, San Diego, CA 92138; (619) 534-5180, johnsong@sdscsds.arpa.

May 30
1989 ACM Conference on LISP, Implementation, Semantics, and Practice/History of LISP (LISP/HOL), Cambridge, Mass., November 28–December 1. For paper requirements, contact Guy L. Steele, Thinking Machines, Inc., 245 First Street, Cambridge, MA 02142-1214; (617) 876-1111, gls@think.com.

May 31
5th International Conference on Statistical and Scientific Database Management, Charlotte, N.C., April 3–5, 1990. Submit 5 c. of complete papers to Conference Co-Chair, Dr. Doron Rotem, Lawrence Berkeley Laboratory, University of California, One Cyclotron Road, Berkeley, CA 94720 (USA) or Dr. H. Hinterberger, ETH Zurich, Institut Informatik, RTH-Zentrum, CH-8092 Zurich, Switzerland (other countries).

June 1
1989 ADCIS Conference: Hypermedia in Education. Sponsor: Hypermedia SIG of Association for the Development of Computer-Based Instructional Systems. Submit manuscript (20 double-spaced pages maximum) with a 100-word abstract to Mark Chignell, Department of Industrial and Systems Engineering, University of Southern California, GER-240, MC-0193, University Park, CA 90089-0193; (213) 743-2705, chignell%mizar.usc.edu@oberon.usc.edu.

June 16
4th International Workshop on High-Level Synthesis, Shawmut Inn, Kennebunkport, Maine, October 15–18. Sponsors: ACM/SIGDA and IEEE/DATC in coop. with IEEE/CANDE and IFIP/WG10.2. Submit 12 c. of extended summary (1,500 words maximum) to Technical Program Chairperson, Raul Camposano, IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598; (914) 945-3971, raulc@ibm.com.