(62) 10/89

Guest Editorial

Comments on Using the Livermore Loops and Harmonic Mean to Measure the Basic Performance of Two Graphics Supercomputers

(Lue and Miyai)

Gordon Bell,
Vice President of R&D
Glen Miranker,
Vice President & Chief Architect
Ardent Computer Company

Editor's Note: With the upgrade to Release 2.1 of the Titan OS, we were able to duplicate the Ardent measurement on LFK loop 13.

The article in the June 89 issue of Vector Register comparing two graphics supercomputers was extremely interesting and insightful. Since one of the numbers obtained for the Titan was low by a factor of two, the harmonic mean was reduced by 20%. This stimulated this note about how reducing one of the 24 loops affects the mean.

Several articles have been written on benchmarking which all who look at performance metrics should understand. Amdahl (1967) originated the concern about performance by comparing uni- and multi-processors for speed-up. Since the harmonic mean is Amdahl's law generalized with an equal number of operations for each loop, the discussion is intended to provide more insight for supercomputer and parallel computer builders and buyers.

Using the mean has implications on the expected performance, capacity, and cost/performance for small (4-16 processor) and large (>100 processor) multi-processors, especially when the processors are used in parallel. This thinking also leads to questioning how computers with multiple processors should be deployed to give the best results in terms of single program speed, system throughput, and performance/cost. Specialized computers like the Connection Machine will be their nature have a disproportionately lower mean than one might imply from their peak (or advertising speed). The mean used alone may automatically cause these new computers to be rejected, unless users understand their workload and can adjust weights accordingly from kernels of other benchmarks. Finally, the authors provide 11 rules about supercomputer design.

The **Ardent Titan** is faster than indicated in the benchmark comparison of the Ardent and Stellar machines (June 1989). The results point out interesting properties of machines as measured by the Livermore Loops (McMahon, 1986) using the harmonic mean. ISR obtained a reading of 0.213 Mflops instead of 0.43 Mflops for loop 13 for Titan 1. Ardent is unable to reproduce the slower speed. By running loop 13 at speed, a Titan 1 (one processor), Release 1.0 provides 1.82 Mflops for vector length 471. Using release 2.0 software the harmonic mean for one processor is 1.84 Mflops and for Titan 2 (two processors), 2.02 MFlops. *(see Editor's Note)*

The **harmonic mean** of the loops (Worlton, 1984) forces system designers to systematically focus on the weakest parts of the machine. It rewards systems for good performance on scalar and hard to vectorize loops and hardly rewards supercomputers with vector capability. Since the loops are uniformly weighted and are not correlated with workload, the difficult, slow loops are likely to give the user a pessimistic indication of performance.

The simple harmonic mean assumes all of the loops carry out an equal number of floating point operations. Harmonic mean is like a swimming, biking, running triathlon where every leg has an identical distance. McMahon's report on the loops provides a set of coefficients for weighting the loops which correspond to actual environments. For an assessment of how a machine will actually perform, users must weight the loops corresponding to their own workload. Smith (1983) explores the harmonic mean as a single metric indicator for a benchmark suite of two or three hypothetical computers to give insight. His rule 3 states "Harmonic mean should be used for **summarizing** performance expressed as a rate."

The **geometric mean** can be a measure of the workload if all the benchmarks are normalized to one of the computers. Smith's rule 2: "Geometric mean should not be used for summarizing performance expressed as a rate or as a time."

The **simple average** is a relatively optimistic measure of machine performance and rewards vector (supercomputer) architectures. Increasing the speed on a few loops by a factor of 25 doubles the performance of the computer. Not doing well on a loop, including not running it, decreases the average by only 4%. In effect, the speed for each of the course legs is just averaged, corresponding more closely to what triathlon race designers try to do when making the three activities spend roughly the same energy (or more closely to the same time). Smith's rule 1: "Arithmetic mean can be used as an accurate measure as performance expressed as time... it should not be used for summarizing performance expressed as megaflops."

Loop 13 and its effect on the Ardent mean

The numbers you obtained are an excellent example of how one slow loop drastically reduces the harmonic mean. For example, we cannot reproduce the result of 0.213 for loop 13. All released versions of the compiler produce 0.43 for this loop. This doubling of speed for the slowest loop increases the harmonic mean from 1.54 to 1.82 Mflops, or 18%. If loop 13 is totally eliminated, Titan would perform at 2.12 Mflops, or another 16% higher. If loop 13 is run infinitely fast, the mean would increase to 2.21 or another 4%. If loop 13 were changed from the slowest loop to be equal to the fastest (12.2 Mflops) the performance would increase from 1.82 to 2.19, or by 20%, illustrating the effect of workload on performance and capacity.

The sensitivity of a slow (deviant) loop on the harmonic mean and arithmetic average is shown in the figure. For highly parallel multiprocessors or very high speed vector computers, the performance of a deviant loop might vary by as much as a factor of 25 in either direction about the norm, causing a loss in performance of a factor of 2 and a gain of only 4%. Even in a balanced supercomputer, **the** lowest performance loop can be a factor of 10 lower than the mean, as observed in the Titan. If the one slow loop that is 0.1 of the norm is not varied by a factor of 2, making it either 0.2 or 0.05 of the norm, then the harmonic mean changes about ± 20%.

Supercomputer Design Rules

These rules illustrate the thinking that went in to improving software and the design of our next generation Titan hardware and software which we invite ISR to test. For a single processor, you should see about 4.5 Mflops for the harmonic mean.

The 11 design rules for a supercomputer are:

◆ 1. Performance, performance, and performance are the three objective criteria for a supercomputer design. Rules 2–6 relate to performance.

◆ 2. Amdahl's law generalized implies that everything matters... a variant of "no chain is stronger that its weakest link", especially when measuring links by harmonic mean.

◆ 3. The scalar speed matters most and a supercomputer must be the fastest of comparable computers in its class... *otherwise the harmonic mean measurement kills it as a supercomputer*.

◆ 4. The vector speed can be arbitrarily high as costs allow. This is the advertised speed of the computer. The past rule of thumb is to have a vector unit which will produce two results per clock tick. Large increases over the scalar speed beyond a hundred provide small benefit except for selected applications, making the computer special purpose (e.g. the Connection Machine). The recent NEC SX-3 announcement for a supercomputer has a peak speed of 16 times the clock. The vector (peak) speed is the speed which the manufacturer guarantees the computer will not exceed on any application.

◆ 5. Allow no holes in the performance space (e.g. arithmetic function, input-output, mass storage) into which a benchmark can step, resulting in large performance losses.

● 6. Provide peaks in the performance space in order that extraordinary performance for a benchmark will result. Use this number to advertise (characterize) the machine and to challenge other machines).

◆ 7. Obey Computer Design Law #1: Provide enough address bits for a decade of constant architecture implementation.

◆ 8. Build at least two generations of the architecture. No first design supercomputer has ever been perfect. Do it again after the first one.

◆ 9. Build on the work of others. Designing a supercomputer is hard. Understand exactly why and how every machine works and move forward using this knowledge and residual software.

◆ 10. Make it easy to use. Have a great compiler and diagnostic tools to aid users in vectorization and parallelization. Training for supercomputers is missing in academe since computer science departments are not oriented to training people to use computers or deal with computers that produce numbers. No computer science text exists about programming a parallel, vector processor (i.e. supercomputers).

◆ 11. Have slack resources when embarking on a supercomputer design. The fatality rate for companies making machines is at least 50%, and even though a design may be good, it has to be re-iterated. Building a new supercomputer costs a minimum of $50 million to get a breadboard.

Implications of the harmonic mean on multiprocessing.

If Titan 2 is always used in a parallel mode (compiler option O3), the performance goes from 1.84 to 2.02 Mflops, an increase of only 10%. Since the cost increment is about 20% on typical configurations, one would not use it for the Livermore loops workload. However, since there is negligible interaction between the two processors when used in a multiprogrammed fashion, the capacity of the two processor case is almost doubled. Titan 2 provides 3.6 Mflops of computing capacity when used by multiple programs, which for a 20% increase in price, increases the performance/price by 67%.

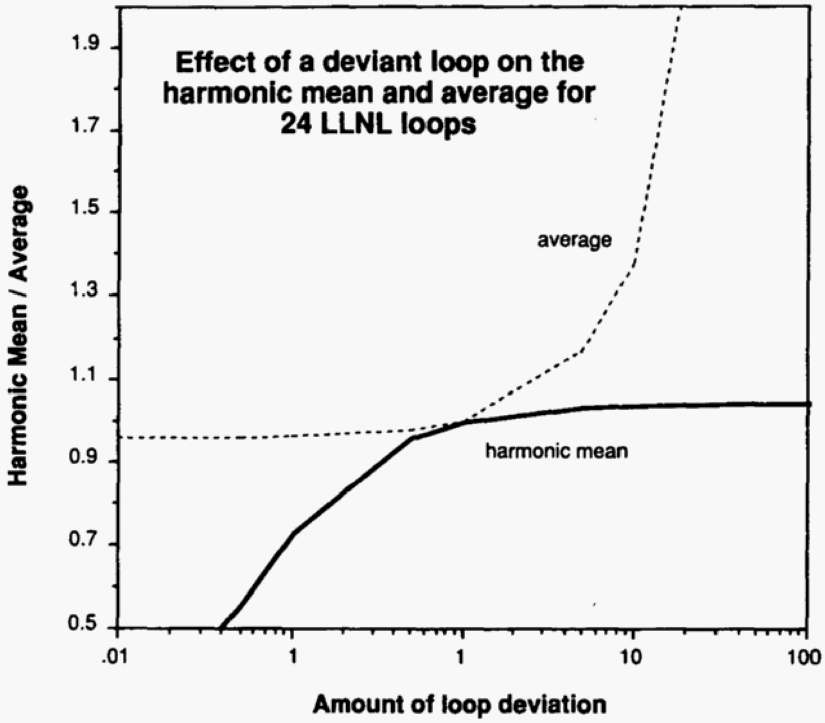
Implications for building fast machines using slower processors or processing elements. Several interesting and potentially important parallel machines are being built with either several hundred or thousands of processing elements (e.g. the Connection Machine). Since the Connection Machine scalar speed is slow, the harmonic mean is likely to be an order of magnitude slower than one would expect based on its peak speed. On the other hand, it's probably ideal when the computation fits the machine and utilizes most of its resources.

Micros have clocks in the 25–30 MHz range, compared to the 300–500 MHz of supercomputers, this, each loop running scalar on one processor will drag the mean down by a factor of 2 versus a supercomputer since the code is likely to run a factor of 10–20 slower. On the other hand, if the large multi's are run in a multiprogrammed fashion, the capacity they would provide for the scalar loop would be many times that of the largest supercomputer. Since their processing capacity, measured in absolute scalar and vector rates is likely to be high, providing the most cost-effective computation, such computers should be even more useful that traditional supercomputers in large computations. Unfortunately, users need a lot more understanding about how benchmarks are related to performance and capacity, otherwise the new parallel structures will be prematurely eliminated from consideration.

Conclusions

Much care must be taken in benchmarking, especially when it goes into a harmonic mean.

Users must know their workload in order to adjust the weights when they look at benchmarks and not just take the "equal-legged triathlon" results when selecting a machine. In other words, if the only race is swimming, it may not be worthwhile selecting an athlete that can bike or run. Without this understanding all the new parallel computers will be, by their numbers, almost automatically rejected.



Bibliography

G. M. Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities", **Proc. AFIPS Comp. Conf.**, Vol 30, 1967.

F. H. McMahon, "The Livermore Fortran Kernels: A Computer Test of the Numerical Performance Range", **Lawrence Livermore National Lab Report UCRL-53745,** December 1986.

J. E. Smith, "Characterizing Computer Performance with a Single Number", **Communications of the ACM**, Vol. 31, No. 10, October 1988, pp. 1202–1206.

J. Worlton, "Understanding Computer Benchmarks", **Datamation**, Vol. 30, No. 4, September 1, 1984, pp. 121–130. ■