Basic Performance of Two Graphics Supercomputers: Stellar GS1000 and Ardent Titan-2

K. Lue
K. Miyai
ISR

The performance of the Ardent Titan-2 and the Stellar GS1000 are analyzed through two sets of standard benchmarks: the Los Alamos Vector Operations Kernels and the Livermore Fortran Kernels. Close scrutiny of the test results shows that basic architectural differences may explain their performance characteristics in these benchmarks. Models are presented to explain these performance characteristics based on the machines' architecture.

The introduction of the graphics supercomputer in 1987 has given the scientific research community a powerful new computational tool. These machines typically come with the computational power of a low-end minisupercomputer (a few tens of MFLOPS) and are distinguished by their integrated high performance graphics capabilities. Most of them are UNIX-based and allow both multiple and single processor configurations. Their excellent cost/performance ratios make them affordable and extremely attractive for people doing scientific visualization and/or medium-size computation.

In this paper, we will look at the Ardent Titan and Stellar GS1000 as compute engines. We will discuss their respective architectures and how architecture affects their benchmark performance, obtained from running the Los Alamos Vector Operation Kernels (Vecops) and twenty-four Livermore Fortran Kernels (LFK). These benchmarks are able to give us some indication of their performance characteristics because they range from basic vector operations to fairly complicated application cores.

Machine Architecture

The Stellar GS1000 is a shared memory computer with a main memory of 16 - 128 MB in increments of 32 MB. The computational engine consists of a multi-stream processor (MSP) and a vector/floating point processor (VFP). The MSP consists of four independent computational streams each capable of executing a common set of machine instructions either independently or in concert with each other. At every 50 ns clock cycle, one instruction can be issued to one of the four streams, for a collective performance of 20 MIPS. This translates to an effective clock rate of 200 ns for each stream.

The vector/floating point unit is shared by all four streams and has the same 50 ns clock. The unit contains 6 vector registers, each with a capacity of 32 64-bit elements. At one floating point calculation per clock cycle, the vector/floating point unit is capable of 20 MFLOPS. On certain calculations, the vector unit can chain add-multiply floating point operations, giving a peak performance of 40 MFLOPS.

The vector/floating point unit is shared by the four computational streams and when one of the streams initiates a vector calculation, the other three are placed in a wait state until the vector operation is completed. The vector unit, therefore, cannot be time-sliced. This allows the GS1000 to achieve a peak of 20 MFLOPS in vector operations even when only one stream is active.

The memory bandwidth is 320 MB/sec, which is sufficient to drive the VFP at peak performance.

The Ardent Titan is also a shared memory computer which uses a multiprocessor instead of multi-stream architecture. In its multiprocessor configurations, the Titan can have up to four physically separate processors. Each processor runs at a clock rate of 125 ns. The Titan is a register-to-register machine with 16-way interleaved memory serviced by a 256 MB/s bus. Each processor unit contains one integer processor and one vector processor (a Weitek chip, consisting of a floating point add, multiply, and logic unit). Each vector processor has a peak performance of 8 MFLOPS for single vector operations and 16 MFLOPS for combined vector multiply-add operations. The integer unit uses a 32-bit MIPS chip with a peak performance of 16 MIPS. Each processor contains 32 vector registers, each capable of storing 32 64-bit elements. The Titan's architecture is somewhat similar to that of the Cray X/MP.

Benchmark Codes

Good benchmarks should reveal the strengths and weaknesses of a computer, from which one may extrapolate the likely performance of the machine under different types of workloads. No one benchmark can fulfill everyone's requirements and because of the wide range of possible applications, a truly universal benchmark is not possible. In this study, we are primarily interested in assessing the fundamental computational functions, involving their ability to do arithmetic operations, strided data operations, multiple array calculations, etc. We have also attempted to correlate results with the machine architecture by presenting models to explain the observed performance. It must be emphasized that the models presented here are only conjectures and we cannot confirm their correctness without more detailed information.

The Vecops kernels are a set of Fortran loops that perform basic arithmetic operations such as scalar-vector adds and vector-vector multiplies. Performance in these loops allows the user to examine raw computational capabilities because of their simplicity. The Vecops kernels also contain test operations with variable vector lengths (Vecskip). These are the same as the Vecops kernels except that they are run with various non-unit strides. Results for Vecskip stride one are identical to Vecops. These different strides help to identify the machine's performance degradation, if any, when non-unit stride memory accesses are used. This is an important aspect of performance, as many physical problems require non-unit stride type memory accesses.

The Livermore Fortran Kernels (LFK) are a set of twenty-four Fortran loops extracted from real applications used at Lawrence Livermore National Laboratory and consolidated as being roughly representative of the LLNL workload. Since they are only the computational cores of actual application codes, extrapolation of these results to predict the performance of any real applications should be done judiciously.

Benchmark Environment

We performed our tests on a two processor Titan with 64 MB of main memory with System V Unix version 1.1 and Fortran compiler version 1.1. The GS1000 tested had 32 MB of main memory and ran Stellix V1.5.1 (the Stellar version of the Unix operating system) with version 4.0 release 11.0 of the Fortran compiler. The tests were performed in February 1989.

Benchmark Results and Analysis

Los Alamos Vector Operations

To understand the basic capabilities of a computational unit, it is best to start with an analysis of its performance on very simple arithmetic operations. We have grouped the Vecops kernels according to number of operations: Group 1 kernels (#1–#4) have only one operation and Group 2 kernels (#4–#9) have two or more. Results from the Vecops kernels are listed in Table 1.

We ran these loops with various vector lengths of 25 to 1000 and the results reported here reflect that of vector length of 1000. Performance improves steadily with increasing vector length, and reaches an asymptotic value around 1000.

In scalar mode, the Titan performs at approximately 1.2 MFLOPS on Group 1 kernels, more than twice the GS1000 speed of around 0.48 MFLOPS. Since the GS1000's four stream architecture means that each stream gets only one of every four clock cycles, when running in scalar mode (single stream), the GS1000 has an effective clock of 200 ns, resulting in lower scalar performance.

In vector mode (stride 1) on the Group 1 kernels, the Titan ran between 5.4 and 7.2 MFLOPS, slightly faster than the GS1000 (between 4.2 and 5.2 MFLOPS). On the more complex Group 2 kernels the Titan peaks at 11.2 MFLOPS for stride 1 data while the GS1000 has a maximum of 9.6 MFLOPS. Below we will present models for the Titan and the GS1000 which attempt to explain the consistency of this data. We shall use 'peak speed' to denote the theoretical hardware peak speed for a given operation and 'adjusted maximum' for the maximum speed possible given the model presented. A 'chime' (short for 'chained vector time') represents the number of clock cycles approximately equal to the vector length, if the vector startup time is ignored. When operations are chained (i.e. do not require an intermediate store to memory) they are executed in the same chime, which produces multiple results every clock period.

Scalar Mode

	Titan				Stellar			
STRIDE	1	2	4	8	1	2	4	8
V=V+S	1.35	1.35	1.35	1.37	**0.49**	**0.35**	**0.35**	**0.33**
V=S*V	1.35	1.35	1.35	1.37	**0.48**	**0.35**	**0.35**	**0.33**
V=V+V	1.27	1.25	1.25	1.26	**0.48**	**0.32**	**0.32**	**0.28**
V=V*V	1.26	1.25	1.25	1.26	**0.49**	**0.44**	**0.44**	**0.37**
V=V+S*V	1.75	1.74	1.74	1.76	**0.88**	**0.60**	**0.60**	**0.54**
V=V*V+S	1.75	1.74	1.74	1.76	**0.75**	**0.54**	**0.54**	**0.48**
V=V*V+V	1.26	1.50	1.50	1.50	**0.88**	**0.48**	**0.48**	**0.42**
V=S*V+S*V	2.63	2.55	2.63	2.63	**0.91**	**0.86**	**0.86**	**0.77**
V=V*V+V*V	1.96	2.11	2.14	2.14	**1.04**	**0.63**	**0.62**	**0.55**

Vector Modc

STRIDE	1	2	4	8	1	2	4	8
V=V+S	6.84	6.61	6.61	4.92	**4.76**	**1.28**	**1.19**	**1.19**
V=S*V	7.18	6.52	6.52	4.92	**5.26**	**1.39**	**1.22**	**1.22**
V=V+V	5.48	5.50	5.79	4.91	**4.26**	**1.19**	**0.96**	**0.89**
V=V*V	5.47	5.53	5.79	4.91	**4.44**	**1.04**	**0.94**	**0.88**
V=V+S*V	11.22	11.06	11.71	9.80	**7.41**	**2.04**	**1.89**	**1.70**
V=V*V+S	10.98	11.45	11.29	9.80	**7.69**	**2.00**	**1.85**	**1.67**
V=V*V+V	6.69	6.43	6.60	5.07	**6.45**	**1.67**	**1.56**	**1.37**
V=S*V+S*V	10.81	10.52	10.58	7.62	**9.68**	**2.94**	**2.73**	**2.50**
V=V*V+V*V	7.03	7.09	6.17	5.89	**8.00**	**2.03**	**1.92**	**1.47**

Note: All Figures in MFLOPS

Table 1

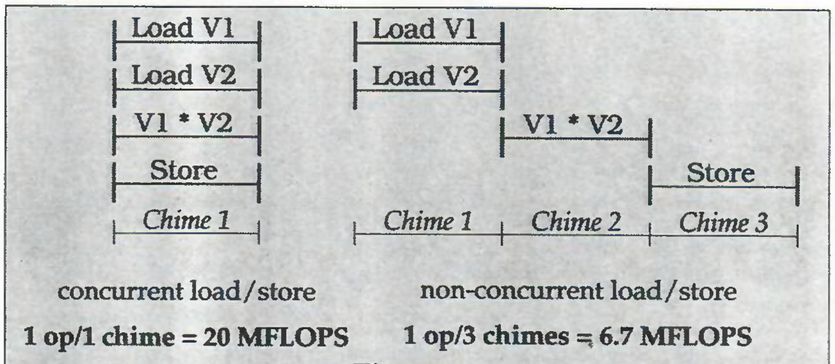


Figure 1

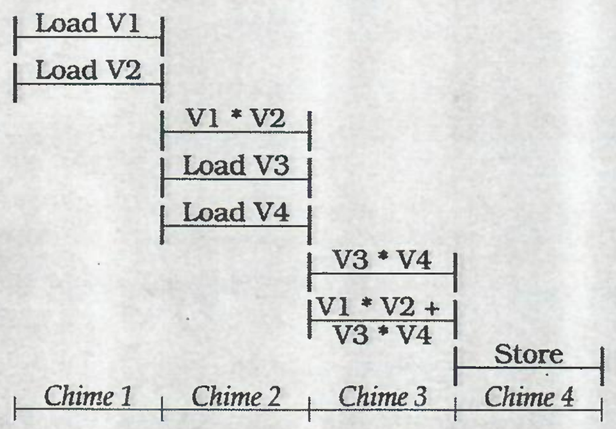
The clock cycles of the GS1000 (50 ns) and Titan (125 ns) result in theoretical peak speeds of 20 and 8 MFLOPS for single vector operations of the form $V = V+V$. For combined multiply/add operations, the theoretical peak speed should double to 40 and 16 MFLOPS. The results reported above are therefore somewhat lower than the peak speeds. For vector-scalar add/multiply operations, the Titan's actual performance is above 80% of peak speed and for vector-vector multiply/add it runs at 67%.

For single vector operations, the GS1000 performs at only a quarter of its peak speed despite the simplicity of these operations, thus suggesting that a bottleneck exists which limits the GS1000's performance. To achieve a rate of 20 MFLOPS, we must assume that the GS1000's supporting architecture is capable of channelling two operands to the vector floating point unit and storing one result to the vector registers all within a single chime. The GS1000's vector load and store paths have sufficient bandwidth to perform the data movement, however, if these loads and stores cannot be completed within the same chime as the vector operation, the overall performance will be adversely affected. In Figure 1, we present a model in the form of a chime sequence for a single vector operation. If load/stores could be chained together with the vector operation, the GS1000 would achieve its peak speed of 20 MFLOPS for single vector operations. In actuality, GS1000 does not seem to have the ability to chain load and store operations, forcing the system to use an two extra chimes for loading and storing. This cuts performance by a factor of three, resulting in an adjusted maximum of 6.7 MFLOPS, which is quite plausible considering we measured from 4.2 to 5.2 MFLOPS in actual execution. Like the Cray-1, the GS1000 appears incapable of chaining load and store operations.

Our analysis can be extended to the more complicated Group 2 kernels. Considering the last kernel, *V=V*V+V*V*, our model shows that the adjusted maximum in this case is 15 MFLOPS rather than 40 MFLOPS (Figure 2). During the first chime, the first two vectors can be loaded and since multiplication cannot take place the unit must wait until the next chime. While performing the first multiplication, the system loads the next two vectors. In the third chime, the full combined vector multiply and add capability of the vector unit is fully realized by chaining the addition to the multiplication. In this chime, the addition can proceed as soon as operands are ready. Since the add and store operation cannot be executed in the same chime for the result vector, a fourth chime is required for the store. This vector operation, therefore, needs four chimes for three arithmetic operations which yields an adjusted maximum of 15 MFLOPS (the measured performance of 8 MFLOPS for this kernel is therefore 53% of this limit, compared to 20% of the peak speed).

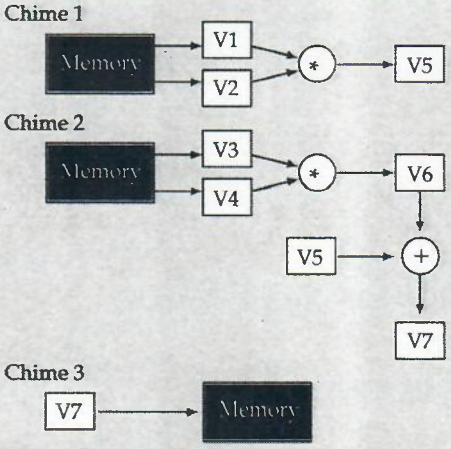
The observed performance on the Titan can be explained by a different kind of bottleneck. The data transfer rate in and out of the vector registers is 512 MB/second, or 8 64-bit words every clock cycle. With the two vector/single scalar operation, V=V*V+S, the effect of this bottleneck is not obvious: we measured 10.9 MFLOPS, or approximately 68% of the peak speed. The vector register bandwidth limitation is more clearly felt in the last kernel, V=V*V+V*V. To understand the effect of the bottleneck in this case, we present a slightly different chime diagram. Figure 3 shows a 'snapshot' within each chime, showing what is happening during the computation cycle.

In the first chime, the arrays V_1 and V_2 are loaded from memory to the vector registers. This takes up two of the 8 words/chime bandwidth. Channelling these operands from the vector registers, performing the multiply, and storing the result back to a third vector register take up another 3 words. At this point V_3 and V_4 cannot be loaded for further operation because the Titan has only two memory load pipes, forcing the next operation to be initiated in a second chime. In the second chime, V_3 and V_4 are loaded and multiplied. With the chaining capability, the temporary result from the previous operation is fetched and used in the addition. At this stage, there is only enough bandwidth to move the result from the adder to a vector register. A further store back to memory cannot be



3 ops/4 chimes = 15 MFLOPS

Figure 2



3 ops/3 chimes = 8 MFLOPS

Figure 3

performed as the 8 word bandwidth of the vector registers has been completely utilized. The last vector register to memory store must be initiated in a completely new chime. In an actual compiler optimized code, the load within each chime might be more evenly divided, e.g. the addition may be moved to the final chime just before the store operation. Therefore three chimes are required to perform 3 arithmetic operations instead of the expected two. This gives an adjusted maximum of 8 MFLOPS (the 7.03 MFLOPS measured is 87.8% of this value).

As pointed out earlier, the Vecskip kernels are computationally the same as that of the Vecops kernels, except that they access memory in non-contiguous strides. Results for these tests are also presented in Table 1.

For scalar operations, the Titan's performance virtually remains the same across all tested kernels with the exception of the *V=V+V*S* kernel which showed a 20 percent increase over unit stride in performance with stride 2, 4, and 8 accesses. In contrast, strided data seemed to have a larger effect on the GS1000, with performance decreasing more than 20 percent for the GS1000 in scalar mode.

The Titan's vector performance on Vecskip does not vary much compared to its unit stride performance. However, a dip in performance is noticeable for stride 8 accesses, due to the way the memory banks are interleaved. The Titan memory bank uses 16-way interleaving and memory conflicts occur with stride eight accesses. The GS1000 is 70 ~ 80% slower on non-unit strides, and the decrease observed is uniform across all kernels. This degradation can be traced to the use of the cache in data transfer. Though the GS1000 has a large data bandwidth, non-unit stride accesses forces it to fetch data that the VFP does not need during non-unit stride operations. In the stride 2 case, every other operand it fetches is unnecessary. Our model in Figure 1 can be extended to take this into account. In essence, stride 2 data accesses mean twice the number of vector loads for the same amount of computational work. The additional cost of vector masking during loading and storing of the operands and results also imposes some overhead. Without knowing the penalty of the masking operations, we can only predict a rough performance limit for the GS1000 in the non-unit stride case. Figure 4 illustrates our model.

The vector load would take twice as long because of the stride and similarly for the store. We have ignored the time required for masking in this model. With this, the operation *V=V+V* takes five chimes to complete, giving an adjusted maximum of 4 MFLOPS. If masking is taken into account, this value would be lower. The performance for stride 2 (1.19 MFLOPS) is approximately 29% of the adjusted maximum and it is therefore highly probable that masking is also required.

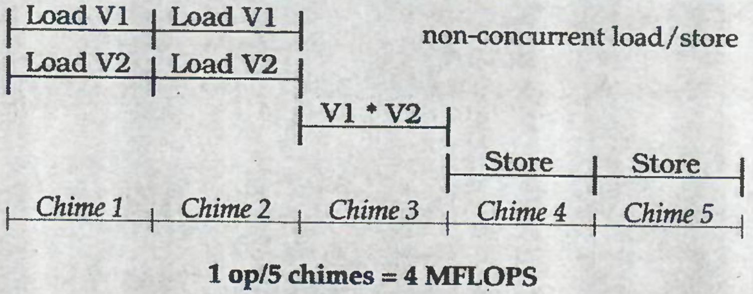


Figure 4

Livermore Fortran Kernels

Proceeding to look at more complicated kernels, we next examine the Livermore Fortran Kernels (LFK). Table 2 is a summary of the scalar and vector results (average vector length=471). The harmonic means of both machines are very similar, with the Titan at 0.811 MFLOPS and the GS1000 at 0.779 MFLOPS (in scalar mode). In vector mode, the GS1000 is slightly better (1.312 MFLOPS) than the Titan (1.253 MFLOPS). The slower performance of the GS1000 in scalar mode is due to a slower 'effective' clock of 200 ns compared to Titan's 125 ns clock, while its better performance in vector mode is related to the poor performance of the Titan on division and intrinsic functions, discussed later.

The various means reported in Table 2 reflect overall performance similarities in both machines. However, since single

Scalar Mode	TITAN	GS-1000
Maximum rate	2.343	2.274
Average rate	1.243	0.957
Geometric rate	1.025	0.862
Harmonic mean	0.811	0.779
Minimum rate	0.213	0.380
Vector Mode		
Maximum rate	11.724	21.026
Average rate	3.633	3.947
Geometric rate	2.174	2.150
Harmonic mean	1.253	1.312
Minimum rate	0.234	0.370

Note: Figures in MFLOPS

Table 2

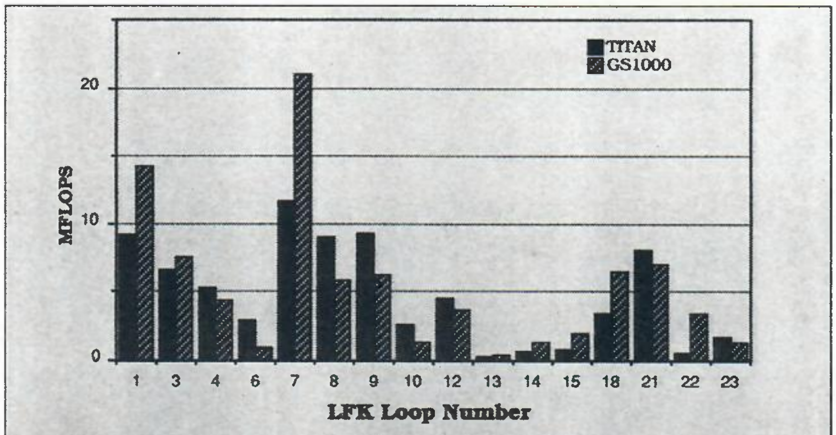


Figure 5

number statistics often smooth out the distinctive system features or weakness which we are trying to identify, we will analyze some of the kernels in an attempt to explain how the GS1000 and Titan function on this benchmark. We will also apply the models discussed in the Vecops section to explain the performance observed.

In vector mode operations, the compiler plays a crucial role in the final performance. The Titan compiler can vectorize kernel 6 where the GS1000 cannot; the reverse case is true with kernel 16. The inability of the GS1000 to vectorize kernel 6 is due to the variable vector length of the innermost loop: the initially short vector length of this loop prevented vectorization.

Performance of kernels vectorized by either system are presented in Figure 5 showing that the GS1000 is faster on kernels 1, 7, 14, 18, and 22, and the Titan is faster on kernels 6, 8, 9, and 10. For these last three kernels, the lower performance of the GS1000 is a result of non-unit stride accesses, explained earlier. In the seven kernels containing strided memory accesses (kernels 4, 8, 9, 10, 11, 13, and 21), the Titan's performance is better with the exception of kernel 13, which contains a MOD function. In this kernel, the Titan's performance edge is offset by its slow execution of the MOD function.

The Titan is also approximately six times slower than the GS1000 in kernel 22. This kernel contains division and exponent calculations which execute much slower on the Titan than on the GS1000.

Another point of interest is the excellent performance of GS1000 on kernels 1 and 7. Recall that the GS1000 is slower than the Titan on simple vector operations, which is explained by its inability to do chained load, compute, and store operations on the same data. However, for more complicated kernels like #1 and #7, the GS1000 is able to compensate for this deficiency by processing different data at different stages of the load-operate-store phase. This can be illustrated by extending our previous analysis on Vecops to LFK #1 in Figure 6a and 6b.

Kernel 1 consists of the following operations: **X** = q + **Y** * (r * **Z1** + t * **Z2**) *(bold face characters indicate vector quantities)*. The GS1000 requires five chimes to complete the operation while the Titan requires only three. Accordingly, the adjusted maxima are:

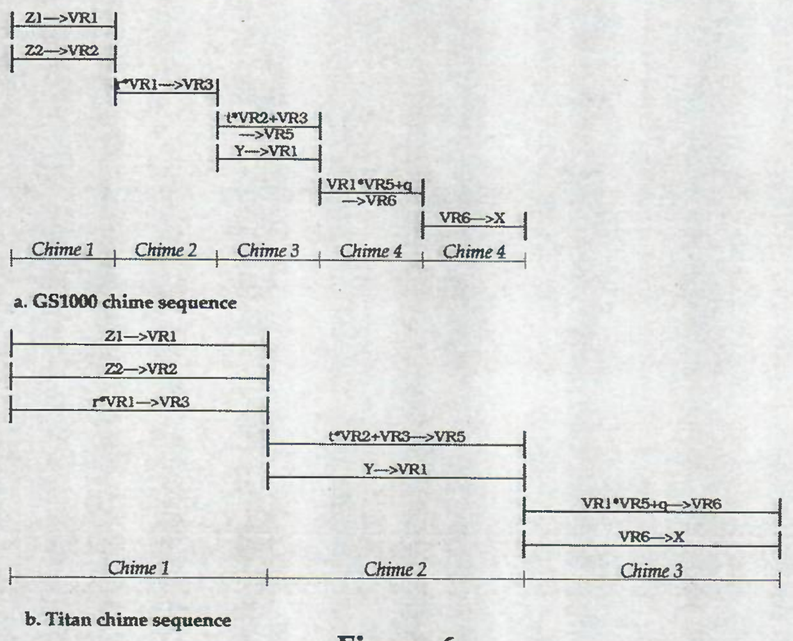
5 operations / (5 chimes x 50 ns) = 20 MFLOPS (Stellar)
[14.26 MFLOPS measured]

5 operations / (3 chimes x 125 ns) = 13.33 MFLOPS (Titan)
[9.17 MFLOPS measured]

By carrying through the same analysis on kernel 7, (X = U1 + r * (Z + r * Y) + t * (U2 + r *(U3 + r* U4) + t * (U5 + r * (U6 + r * U7))), we find that for 16 operations, 11 chimes are needed on the GS1000 while the Titan requires 9 chimes to complete the same operations. The adjusted maxima are therefore:

GS-1000: 29.09 MFLOPS
[21.06 MFLOPS measured]
Titan: 14.22 MFLOPS
[11.72 MFLOPS measured]

Compensation of its inability to chain load/store operations is not the sole reason to GS1000's



a. GS1000 chime sequence

b. Titan chime sequence

Figure 6

speed on kernels 1 and 7. The fact that the VFP unit has a 50 ns clock contributed significantly to its performance. It is also entirely possible that reducing memory accesses through clever use of the vector registers to store intermediate results would further enhance the computational speed.

Summary

We have presented benchmark results of the Ardent Titan and the Stellar GS1000 on two sets of benchmarks with increasing code complexity to examine their basic performance characteristics. For both computers, the speed of the vector floating point unit is not the only consideration which determine final performance. As we have shown, vector register bandwidth, chaining capability, data channel structure, and intrinsic libraries all contribute to overall performance. The models we have presented for both computers are by no means exact and are introduced here merely to help us understand some of the major processes during a computation and to try to explain the observed results.

Acknowledgements

The authors would like to thank Asahi Stellar of Tokyo for the use of their GS1000, Timothy Stewart of Stellar Computer Inc. for his useful discussions on the Stellar GS1000, and Kubota Computer for their helpful assistance.

References

1) E. Misaki, K. Lue, and R. Mendez, Preliminary Study of the CPU Performance of the Titan Compared with that of the Cray X-MP/1, Institute for Supercomputing Research, TR 88-11a.

2) Stellar Graphics Supercomputer System Overview.

3) Titan Technical Overview, Release 1.0.

4) Olaf M. Lubeck, Supercomputer Performance: The Theory, Practice, and Results, Los Alamos National Lab Report LA-11204-MS.

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

6) John M. Levesque and Joel W. Williamson, A Guidebook to Fortran on Supercomputers, Academic Press, Inc., 1989.

7) Sidney Fernbach (Edt), Supercomputers Class VI Systems, Hardware, and Software, North-Holland, 1986. ■

Stardent

Stardent 3000 Visualization Systems and Departmental Supercomputers

Stardent 3000 Visualization Systems and Departmental Supercomputers bring outstanding performance to today's complex mix of scientific problems.

Features

- Peak system performance of 128 MIPS, 192 MFLOPS
- Symmetric multiprocessor architecture
- 2D and 3D graphics and image visualization
- Concurrent integer, scalar floating point, and vector processing
- 32 MHz clock rate
- RISC/CMOS technology
- High-speed system bus
- Automatic vectorizing and parallelizing compilers
- Enhanced UNIX®/X Window environment
- NFS/Ethernet networking
- SCSI and VME I/O

Serious scientific applications demand the performance that comes only from vector processing. Stardent's custom vector unit boosts performance on numerically intensive problems by almost an order of magnitude.

Stardent 3000 Visualization Systems and Departmental Supercomputers bring outstanding performance to today's complex mix of scientific problems. Stardent's advanced, parallel RISC architecture delivers sustainable processing rates of over 100 million instructions per second and 100 million floating point operations per second. In concert with Stardent 3000 graphics capabilities and visualization tools, the 3000 excels in image processing, molecular modeling, computational fluid dynamics, finite element analysis, mechanical computer-aided engineering, industrial design, computer animation, financial and econometric modeling, and mathematical analysis.

Architecture

To deliver the balanced power demanded by real world scientific applications, the Stardent 3000 architecture employs a high-speed system bus, high-performance computational units, a memory subsystem, one or two I/O subsystems, and a graphics subsystem.

Each Stardent 3000 computational subsystem boasts a 32 MHz MIPS R3000 RISC microprocessor for integer processing, a MIPS R3010 scalar floating point co-processor, and a custom, 64-bit vector processing unit for vector and additional scalar floating point processing. Each Stardent 3000 supports one to four computational units.

Serious scientific applications demand the performance that comes only from vector processing. Stardent's custom vector unit boosts performance on numerically intensive problems by almost an order of magnitude. It schedules vector operations, moves vector data, performs arithmetic calculations, and protects the integrity of results through register scoreboarding. With two load pipes, one store pipe, and one operation pipe, the vector unit can complete an operation each clock tick. The result is spectacular performance delivered to end user applications.

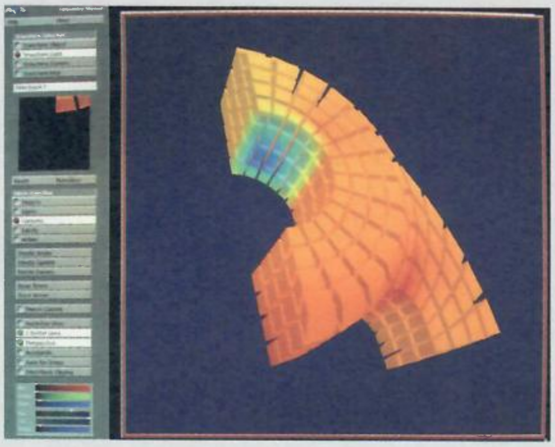
Each Stardent 3000 computational unit delivers performance of 32 million operations per second and 48 million floating point operations per second. That means a system total of 128 MIPS and 192 MFLOPS.

High bandwidth is the key to avoiding bottlenecks in computationally intensive applications. Stardent 3000's 256 megabyte per second bus meets the demand for high bandwidth to and from main memory.

Stardent 3000 offers up to half a gigabyte of main memory to support memory-intensive applications. The performance and reliability of memory resources are guaranteed by the system's 8 or 16-way interleaving and single error correction/multiple error detection.

Stardent 3000 delivers the I/O capabilities expected in today's networked environments. The system offers Ethernet and Cheapernet connections, SCSI and VME interfaces, RS-232 ports, and a Centronics parallel port. Each Stardent 3000 may contain one or two complete I/O subsystems.

Graphics are essential for understanding scientific applications, but not at the expense of machine performance and resource utilization. By splitting graphics processing between the computational units and the graphics subsystem, the Stardent 3000 executes its graphics pipeline with maximum efficiency. Within the computational units, the integer unit executes the graphics display list and the vector unit performs transformations and clipping. The graphics subsystem performs pixel and polygon processing, Z-buffering, image buffering, and display. All graphics processing is done in parallel, meaning that no precious machine resources are wasted.

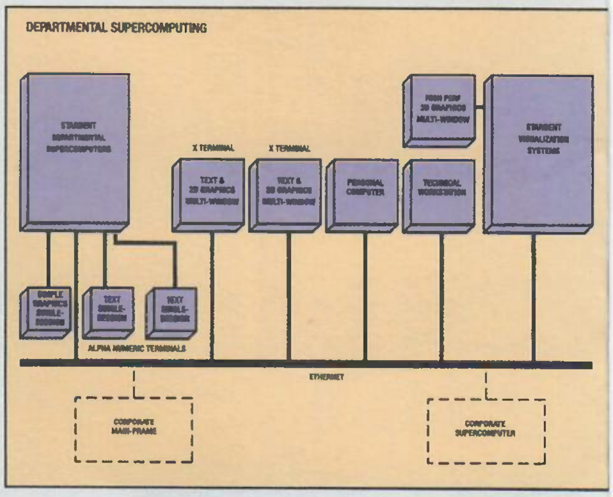


Stardent's AVS™ (Application Visualization System) lets end users build complex visualization applications in little to no time, without user interface programming.

System Software

High performance hardware can only be exploited by high performance system software. The Stardent operating system, based on industry-standard AT&T UNIX System V.3, has been extended for fast, multiprocessor performance. Stardent 3000's symmetric multiprocessing completely avoids the bottlenecks of master/slave multiprocessor designs. Microtasking (lightweight processes) assures efficient use of multiple processors. Other Stardent enhancements to UNIX include a fast file system, disk striping, asynchronous file I/O, and explicit control of application parallelism. The Stardent UNIX system also provides BSD 4.3 extensions including the C-shell, signals, sockets, mail system, and full BSD terminal support. Advanced networking support includes NFS,™ TCP/IP, remote login and execution, and a variety of standard communications protocols.

The Stardent 3000 compilers automatically vectorize and parallelize standard Fortran, C, and Ada code. They also accept directives for optional fine tuning of parallel and vector execution. The compilers are compatible with both VAX- and Sun-developed applications and support both the System V.3 and BSD 4.3 programming environments.


DEPARTMENTAL SUPERCOMPUTING
STARDENT DEPARTMENTAL SUPERCOMPUTERS
X TERMINAL
TEXT & 2D GRAPHICS MULTI-WINDOW
X TERMINAL
TEXT & 2D GRAPHICS MULTI-WINDOW
PERSONAL COMPUTER
HIGH PERF 3D GRAPHICS MULTI-WINDOW
TECHNICAL WORKSTATION
STARDENT VISUALIZATION SYSTEMS
SIMPLE GRAPHICS SINGLE-SESSION
TEXT SINGLE-SESSION
TEXT SINGLE-SESSION
ALPHA NUMERIC TERMINALS
ETHERNET
CORPORATE MAIN-FRAME
CORPORATE SUPERCOMPUTER

Graphics Software

The Stardent Series 3000 offers visualization tools for end users through to graphics programmers. Stardent's AVS™ (Application Visualization System) lets end users build complex visualization applications in little to no time, without user interface programming. Doré, Stardent's popular object-oriented graphics library, is the most complete, easy-to-use, 3D graphics library available today. Figaro+ is available for programmers requiring a standard PHIGS+ graphics interface, and the Direct Graphics Library is offered for image processing and top performance graphics applications. The X Window System is Stardent 3000's standard user interface.

Stardent 3000 Visualization System and Departmental Supercomputer Specifications

Configuration, Model Names

A minimum Stardent 3000 system contains one computational unit, 32 MB memory, a 1/4-inch cartridge tape drive, a 5 1/4-inch disk drive, Ethernet and Cheapernet interfaces, and the features listed under Product Specifications.

The Stardent 3000 product family includes Visualization Systems and Departmental Supercomputers in single or multiple computational unit configurations. Models 3010 to 3040 are one to four processor systems; a Stardent 3000 Departmental Supercomputer is a 3000 system without graphics and is designed for numerically intensive computing across a broad range of applications within a departmental setting. It combines the high-level of computational power offered throughout the 3000 series with a variety of terminal and user interface configurations, including the Stardent X terminal, to create network-based departmental supercomputing. 3000 Visualization Systems are listed in the Stardent Order Book with a T3 prefix; 3000 Departmental Supercomputers are listed with a S3 prefix. Item N-P3/100 is a Stardent 3000 processor module. A Stardent 1500 system can be upgraded to a Stardent 3000 system by replacing P2 processor modules with P3 modules (order book item N-P2/UP3).

Trademarks

Stardent, Doré, and AVS are trademarks of Stardent Computer Inc.
DECnet is a trademark of Digital Equipment Corp.
Ethernet is a registered trademark of Xerox Corp.
NFS is a trademark of Sun Microsystems Inc.
UNIX is a registered trademark of AT&T.
VAX and VMS are registered trademarks of Digital Equipment Corp.
X Window System is a trademark of the Massachusetts Institute of Technology.

The Stardent 3000 product family includes Visualization Systems and Departmental Supercomputers in single or multiple computational unit configurations.

System Bus

Technology	Synchronous, parity checked, disconnect
Width	64-bit data, 32-bit address
Bandwidth	256 MB/second
Arbitration	Single cycle

Computational Units

Number per system	1 to 4
Technology	RISC/CMOS
Floating Point Format	IEEE 754, 32- and 64-bit
Peak integer performance	32 MIPS each; 128 MIPS per system
Peak scalar floating point	16 MFLOPS each; 64 MFLOPS per system
Peak vector performance	32 MFLOPS each; 128 MFLOPS per system
100 x 100 compiled Linpack	8.5 MFLOPS single CPU; 11 MFLOPS dual CPU
Livermore Loops	4.7 MFLOPS (harmonic mean) each
Whetstones	28.6 MegaWhetstones each
Dhrystones	51.7 KiloDhrystones each
Diagnostic console support	VT100-compatible terminal, 9600 or 2400 baud 8-bit characters, 1 stop bit, no parity

Scalar Processors

Integer processor	MIPS 3000
Integer registers	32 @ 32-bits each
Floating point processor	MIPS R3010
Floating point registers	16 @ 64-bits each
Clock rate	32 MHz
Caches	64 KB instruction cache 64 KB write-through data cache

Vector Processor

Processor architecture	Custom
Pipes to memory	2 load pipes; 1 store pipe
Vector registers	256 @ 64-bits wide, 32 elements long
Clock rate	16 MHz; 2 FLOPS each clock tick

Memory Subsystem

Size	32 MB or 128 MB/module; 512 MB max/system
Interleaving	8- or 16-way
Error management	single error correction, multiple error detection

Graphics Subsystem

(Models T3-abcd/e only)

Number per system	1
Color planes	24 or 48, single or double buffered 8-bit pseudo color or 24-bit true color
Other planes	4 overlay, 3 control planes
Z-buffer (depth buffer)	16 bits
Video interface	RS-170A genlock; NTSC/PAL formats
Monitor	19-inch color, single axis curvature 1280 x 1024 resolution, 60 Hz non-interlaced Optional stereo viewer (polarizing filter)

Stardent 3000 Visualization System and Departmental Supercomputer Specifications

I/O Subsystem

Number per system	1 or 2
Bus standards	SCSI; 2 synchronous channels @ 4 MB/second Optional VME; 15 MB/second with 2 internal slots
Network connection	Ethernet and Cheapernet (thin-wire Ethernet)
Serial ports	4 RS-232C, DMA access
Parallel port	Centronics standard
User interface devices	3-button optical mouse; AT-compatible keyboard, optional knob box, digitizing tablet

Internal Mass Storage

Disk	1 to 3 380 MB or 760 SCSI drives
Streaming cartridge tape	¼-inch 120 MB SCSI drive

External Mass Storage

Disk	380 MB, 760 MB SCSI drives (max 22 external) 1 GB SMD drives (to total of 44); requires VME
Tape	½-inch reel-to-reel SCSI drives (6250 GCR) 2 GB 8mm cartridge SCSI drives

Operating System

Kernel	UNIX System V.3
Compliance	SVID standard
Extensions	BSD 4.3 features Multiprocessor support Microtasking (lightweight processes) Fast file system Disk striping Asynchronous file I/O

Compilers

Compiler technology	Automatic vectorization and parallelization
Language standards	C Optional Fortran, Ada
Extensions	VAX/VMS features; Cray directives
Support	High performance libraries Profiler, debugger, post-load technology

Visualization

End-user	Application Visualization System
High-level programmers	Doré, Figaro (PHIGS +)
Graphics programmers	Direct Graphics Library
User interface	X Window

Networking

Data link	Ethernet, Cheapernet Optional FDDI, DR11-W
Transport/Network	TCP/1P, UDP
File System	NFS
Proprietary protocols	Optional DECnet, BISYNC, SNA/RJE

Physical-Main Cabinet

Size	22.5-inch wide, 24-inch deep, 50.5-inch high
Card cage slots	10
Peripheral devices	Space for 4 drawer-mounted SCSI devices
Maximum distance to monitor	200 feet
Power requirement	110V/20A or 220V/16A
Power consumption	Less than 1500 watts

Physical Expansion Cabinet

Size	22.5-inch wide, 36-inch deep, 50.5-inch high
Rack height	23 EIA units
Card Cages	11-slot VME (6U boards)
Peripheral device drawers	8-inch SMD drawer (holds 1 to 2 drives) 5¼-inch SCSI disk drawer (holds 1 to 4 drives)
Power requirement	120V/20A or 240V/16A

Environmental

Temperature	10 to 40°C (operating) -30 to 50°C (non-operating)
Humidity	40 to 80% non-condensing (operating) Maximum 90% non-condensing (non-operating)
Shock	4g, 15ms, 3 axes (non-operating) 30-inch, 5 drops (shipping)
Vibration	Operating: .01-inch @ 5-22 Hz .25g @ 22-500 Hz Non-operating: .1-inch @ 5-10 Hz .5g @ 10-500 Hz Shipping: .25-inch @ 5-11 Hz .15g @ 11-200 Hz
Electrical	90-130 VAC RMS or 180-230 VAC RMS
Frequency	50 ± 3, 60 ± 3 Hz
Acoustical	Less than 63 dbA @ 23°C
Safety/Emissions	UL-478 5th edition CSA 22.2 #154 TUB/VDE 0806 IEC 380 RFI FCC Class A RFI VDE 0871 Level A
ESD	No effect: 2.5/5 KV Machine-recoverable errors: 7.5/10/12.5 KV

HEADQUARTERS

95 Wells Avenue
Newton, MA 02159
(617) 964-1000
(617) 964-8962 FAX

WESTERN OPERATIONS

880 West Maude Avenue
Sunnyvale, CA 94086
(408) 732-0400
(408) 732-2806 FAX

INTERNATIONAL OPERATIONS

Hagenauer Strasse 42
6200 Wiesbaden 1
West Germany
49 61 21-22037
49 61 21-22701 FAX

INTERNATIONAL SUBSIDIARIES

Hohenzollernring/
Friesenplatz No. 1
D-5000 Koln 1
West Germany
49 221-252052
49 221-251730 FAX

6, avenue du Vieil Etang
78180 Montigny
le Bretonneux
France
33 1 30-583322
33 1 30-583008 FAX

15, Frederick Sanger Road
The Surrey Research Park
Guildford Surrey GY2 5YD
United Kingdom
44 483-505388
44 483-505352 FAX

REPRINTED FROM

MARCH 5, 1990

digital review

THE INDEPENDENT NEWSPAPER & TEST LAB OF DEC COMPUTING

PHOTO: TIM DAVIS

A NEW GRAPHICS POWER COMES TO LIGHT

BY DENIS W. HASKIN

Last year's bicoastal merger of Newton, Mass.-based Stellar Computer and Ardent Computer of Sunnyvale, Calif., to form Stardent Computer hasn't slowed the release of new products from these minisupercomputer developers. While the future of products released by the two original companies is still up in the air, the new company has come out with an exceptional minisupercomputer bearing the Stardent name.

Dubbed the Stardent 3000, this new machine comes in two flavors: the Stardent 3000 Visualization System and the Stardent 3000 Departmental Supercomputer. The new models send a clear message that Stardent has selected Ardent's Mips-based hardware over Stellar's proprietary hardware and suggests the company will favor Stellar's software over Ardent's. Stardent has announced that it will merge the Stellar and Ardent product lines by 1991.

We tested the two-processor version of the visualization model, the Stardent 3020 Visualization System, and were thoroughly impressed. The Stardent 3000 turned in an admirable 39.68 MicroVAX II units of processing (MVUPs) on DR Labs' CPU 2 test suite (Fig. 1).

Stardent's 3000 Visualization Systems and Departmental Supercomputers deliver exceptional images

The Stardent 3000 is a direct descendant of the Ardent Titan II, which is now referred to as the Stardent 1500. Much of this new system is identical to its predecessor, with the exception that the 3000 uses Mips Computer Systems' R3000 RISC chip, whereas the Stardent 1500 uses the R2000.

Unlike other vendors of graphics minisupercomputers, including Silicon Graphics, Stardent employs a graphics processor that uses a limited amount of specialized hardware, relying instead on the power of a vector processor and some help from the Mips R3000. The graphics subsystem consists of one or two boards that are tightly coupled with the CPU and memory subsystems (Fig. 2). The optional second board provides additional frame buffers and rasterizers.

The Stardent 3000 Departmental Supercomputer is the Stardent 3000 Visualization System minus the latter's graphics boards and the display hardware that accompanies them.

The 3000's rasterizers transform an image into pixel data and draw it into the frame buffer. Given the very simple task of displaying an image of a straight line, for example, the rasterizers would first determine which pixels need to be turned on to show that line and then turn on those pixels in the frame buffer.

In addition to their drawing operations (which include Gouraud-shading and Z-buffer hidden-surface removal), the rasterizers also perform pixel fill, movement of rectangular blocks of pixels within the frame buffer, and transfer of pixels from memory to the frame buffer and vice versa.

Z buffering is a method of handling hidden-line and surface removal. The Gouraud technique is used to shade solid objects.

Stardent's single-board graphics subsystem has a frame buffer with 24 color planes, and has two rasterizers. The graphics expansion board adds another 24 planes to the frame buffer, along with two additional rasterizers.

The graphics subsystem has two color modes: full-color and pseudo-color. In full-color mode, the system can display 16 million colors simultaneously. In pseudo-color mode, 256 colors may be simultaneously displayed out of a possible 16 million. Each mode has its own color map.

The subsystem also has four overlay planes for cursors, pop-up menus, and similar graphic elements, as well as three control planes used for color mode selection and double-buffering.

Data is transferred from main memory to the graphics subsystem over a 64-bit DMA channel, with a 32-bit DMA channel serving in the reverse direction.

The graphics expansion board includes a genlock interface, allowing users to make broadcast-quality recordings of the displayed graphics. This board provides both NTSC and PAL genlocking and generates RS-170A- or PAL-compatible video signals.

CPU POWER

The CPU structure of the Stardent 3000 is typical of implementations that make use of the Mips RISC chips: it uses the R3000 integer scalar processor along with an R3010 scalar floating-point processor. Each CPU unit also includes two independent 64KB direct-mapped caches—one for instructions and the other for data—and four-deep write buffers.

The Stardent 3000 is a true multiprocessing system that can have from one to four CPUs (Fig. 3). Each CPU incorporates both a scalar and a vector processor. The scalar processor consists of the Mips R3000 and R3010 running at 32MHz. The vector processor is a proprietary product from Stardent.

The scalar processor handles all integer and almost all floating-point scalar operations, and functions as the scene database traverser during graphics operations. (The vector processor incorporates a scalar floating-point processor, but it is rarely used now that the Mips R3010 floating-point proces-

TEST MODEL

LABS

We tested a two-processor version of the Stardent 3000 Visualization System, the Stardent 3020, with 64MB of memory, a graphics expansion board and two 760MB disk drives.

Among the target audience of the Stardent 3000 Visualization System are scientific investigators doing interactive simulation of complex systems, such as the computational fluid dynamics used for weather modeling. Such users need substantial CPU power to perform the involved integer and floating-point calculations such applications require, allowing these users to change the parameters of their model and quickly see how those changes affect their results.

To evaluate the Stardent 3000's ability to satisfy this demand for compute power, we ran DR Labs' CPU 2 benchmark suite on the Stardent 3020. CPU 2 measures a system's ability to handle scientific and engineering applications. Its results reflect the raw speed of a CPU, the efficiency of a system's Fortran compiler, and—to a lesser extent—the speed of memory access. The results are not affected by a system's I/O capabilities.

CPU 2 includes integer calculations, single- and double-precision floating-point calculations, and matrix calculations. Individual kernels measure a system's ability to do vector processing, and some kernels measure the improvement achieved by systems with parallelization capabilities.

TEST BRIEF

LABS

PRODUCT: Stardent 3000

DESCRIPTION: RISC-based multiprocessing and vector processing graphics supercomputer

VENDOR: Stardent Computer Inc.
880 W. Maude Ave.
Sunnyvale, CA 94086
(408) 732-0400

TEST CONFIGURATION: Stardent 3020 Visualization System with two CPUs, 64MB of memory, two 760MB disks and expansion graphics board

PRICE AS TESTED: $176,900

PROS: ■Offers high compute performance ■Has automatic vector and parallelizing compilers ■Has multiprocessing Unix ■Supports multithreaded processes ■Offers high-level graphics environment

CONS: ■I/O bandwidth not as high as other minisupercomputers

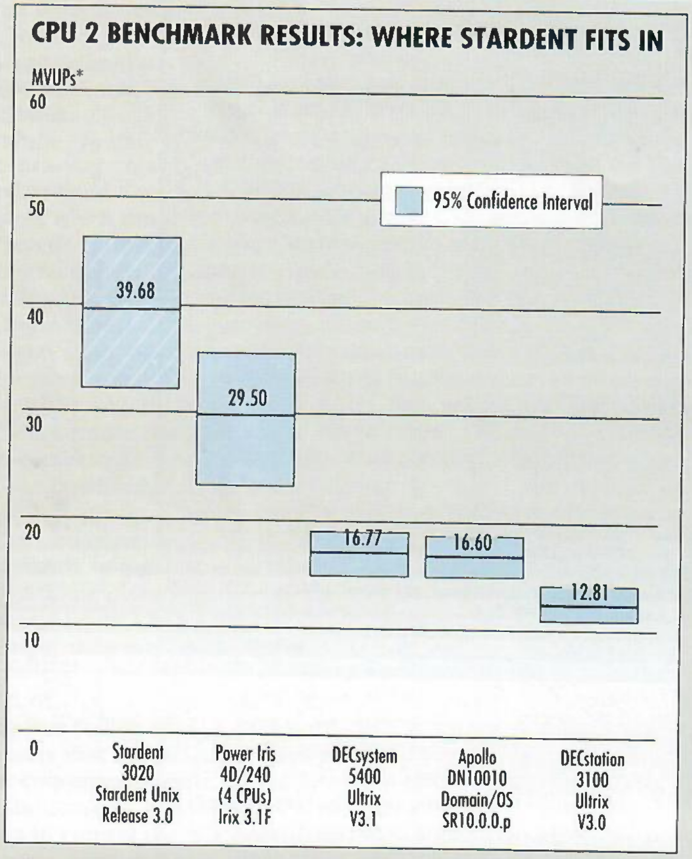


FIGURE 1: *The two-processor Stardent 3020 delivered a strong 39.68 MVUPs. The Silicon Graphics Power Iris, a graphics minisupercomputer targeting much the same applications areas as the Stardent, delivered only 29.50 MVUPs with four CPUs.*

Statistical Analysis of Benchmarks for Stardent 3020

KERNEL NAME	MICROVAX II TIME (SEC.)	3020 TIME (SEC.)	3020 MVUPs*	PSEUDO MVUPs*	SCALED RESIDUALS
FFTR	83.73	2.58	32.41	31.62	1.86
LAHYDR	85.60	1.33	64.56	58.85	–4.60
LABLE	82.39	1.57	52.46	50.71	–2.67
LATRD1	81.24	2.64	30.83	29.63	2.33
LATRD2	81.84	2.45	33.39	32.81	1.58
LAPDE	81.25	1.29	63.21	58.03	–4.40
LADIFF	74.51	1.66	44.86	44.54	–1.20
LAFSUM	80.58	4.36	18.48	8.99	7.23
LA2DPP	84.57	4.27	19.80	11.79	6.56
LA1DPP	82.35	4.23	19.49	11.15	6.71
LA2DHY	75.75	2.12	35.75	35.53	0.93
LAFIMI	81.33	4.18	19.47	11.12	6.72
LAIMCO	81.88	3.24	25.29	21.68	4.22
LAMAPR	79.33	1.29	61.44	56.92	–4.14
LINPAC	80.02	1.57	51.05	49.64	–2.41
AIRREL	81.53	3.02	27.00	24.31	3.59
DWHET	82.38	0.81	102.01	76.62	–8.81
DOUBLE	76.95	0.80	96.22	74.37	–8.28
EGYPT	78.48	2.64	29.70	28.14	2.69
EGYPTD	83.00	2.63	31.55	30.56	2.11
GAUSS	82.88	2.02	41.08	41.06	–0.38
HANOI	76.52	4.33	17.68	7.21	7.65
NEFF	80.79	0.98	82.51	68.41	–6.87
PHILCO	75.96	1.59	47.70	46.97	–1.78
PRIME	73.21	8.50	8.61	-22.38	14.67
RR2	82.58	2.30	35.84	35.63	0.91
RR3	83.87	2.89	29.03	27.21	2.90
RR4	82.44	4.96	16.63	4.70	8.25
SINGLE	75.02	1.09	68.95	61.42	–5.21
SWHET	77.10	0.99	77.54	66.00	–6.29
ALAM18	85.25	1.43	59.82	55.87	–3.89
GAMSIM	91.77	0.65	142.28	89.39	–11.84
LUDD	86.99	1.17	74.59	64.49	–5.94
LUSD	82.65	1.66	49.82	48.68	–2.19
	Geo. Mean 80.96	Geo. Mean 2.04	Geo. Mean 39.68	Jackknife Value 39.46	

Summary of Statistical Analysis

DR Labs Standard MVUPs*: 39.68
Variance: 17.7752
Standard Deviation: 4.2161
95% Confidence Interval: 32.38 – 46.54

* MVUPs = MicroVAX II units of processing.

sor accompanies the R3000.)

Stardent's vector unit, built with components from Bipolar Integrated Technologies, handles all vector operations. During graphics processing it serves as the geometry engine for the system, performing the requested transformations and other graphics operations.

This vector processor is very similar to the one used in the Stardent 1500, but with a few enhancements: the vector instruction set has been increased somewhat, some restrictions on the use of vector registers have been lifted, and there is now better data communication between the vector and scalar processors.

The vector unit uses a vector register file with four banks of 2,048 64-bit words. The divider, pipelined multiplier and pipelined adder of the vector arithmetic unit operate concurrently. A vector data switch handles all transfers of data between the vector unit and memory. It also handles transfers within the vector unit from the vector register file, to and from the vector arithmetic unit.

Stardent's vector processor can work with vectors of any length that start anywhere in the vector register file.

The Stardent 3000's memory system is closely integrated with the bus structure of the machine. The 3000 can have from one to four memory boards, each populated with 8MB, 16MB or 32MB of error checking and correcting (ECC) memory. Depending on its configuration, the memory system can provide either 8-way or 16-way interleaving.

The 3000 uses two buses to communicate between the memory, CPUs, I/O and graphics subsystem (Fig. 4). Both buses have 32-bit address and 64-bit data paths and operate at 16MHz.

One bus, the R-bus, is dedicated to data transfers from memory to the vector processor. The other bus, known as the S-bus, handles all other transfers.

The I/O subsystem of the 3000 is implemented on a single board and supports all of the I/O functions with the exception of graphics display (it does handle the input devices of the graphics display, including the keyboard, mouse and other devices).

This subsystem uses an internal bus operating at 64MB per second as well as an additional "microbus" that runs at 2MB per second. The primary bus supports two SCSI channels, each of which can support as many as seven SCSI devices, and a VME interface. (The VME bus is optional.)

The microbus supports the keyboard and mouse of the graphics workstation and provides four RS-232-C ports (with full modem control), a parallel interface, regular and thin-wire Ethernet, and a sound generator. Other incidental support components include a clock, timer, and nonvolatile RAM for address and configuration information. The I/O subsystem connects only to the S-bus.

RUNNING ON UNIX

The Stardent 3000 runs a fully multiprocessing, multitasking version of Unix that is totally compatible with AT&T System V release 3 and incorporates extensions from Berkeley 4.3.

Stardent's Unix has been modified to yield greater performance on the 3000's multiple CPUs. It offers fully asynchronous multiprocessing: the operating system runs in parallel on all processors.

In a significant departure from some multiprocessing systems, Stardent's Unix supports single-process, multithread microtasking. That is, a single process is divided into threads, each of which can operate in parallel on multiple processors.

Other features of the operating system include the implementation of a "disk farm" device driver to facilitate disk striping, Streams-based device drivers and application-controlled asynchronous file I/O.

To boost system performance, Stardent has modified the file system by organizing files contiguously, using large read sizes to reduce the number of disk accesses and making substantial use of read-ahead I/O.

Like most Unix implementations, Stardent's includes support for the TCP/IP and NFS network protocols, and the company offers optional DECnet emulation. Other DEC compatibility features include strong support for VAX Fortran, an EDT-like text editor and DCL command interface emulator, as well as an ad-hoc collection of LIB$ and SYS$ run-time library routines.

Programming languages currently available on the Stardent 3000 include C and Fortran (a validated Ada compiler is available on the 1500 but has not yet been validated for the 3000). The compilers differ only in their front ends, which compile the source code to an intermediate language. Vectorizing, parallelizing, optimizing and code generation are the same in the two languages. The compilation process includes an instruction scheduler that reorders instructions to improve overlap of floating-point and integer operations and memory access.

The C compiler is based on AT&T System V release 3 C with some extensions from the ANSI C standard. The run-time libraries are based on both System V and BSD libraries. The Fortran used is ANSI Fortran-77 with extensions from VAX Fortran version 3.0, and the compiler accepts many Cray Fortran directives. The Fortran I/O library has also been completely rewritten for improved I/O efficiency.

All of the compilers provide automatic parallelization of code.

The user interface for the Visualization System is Stardent's implementation of the MIT X Window System, termed X+. The Departmental Supercomputer includes the X+ library to allow clients to execute on that node. Most of Stardent's extensions are for the X server and include support for double-buffered windows, true- and pseudocolor windows, backing store (a means of saving part of the window that's obscured), overlay graphics, shaded Z-buffer primitives and non-standard input devices.

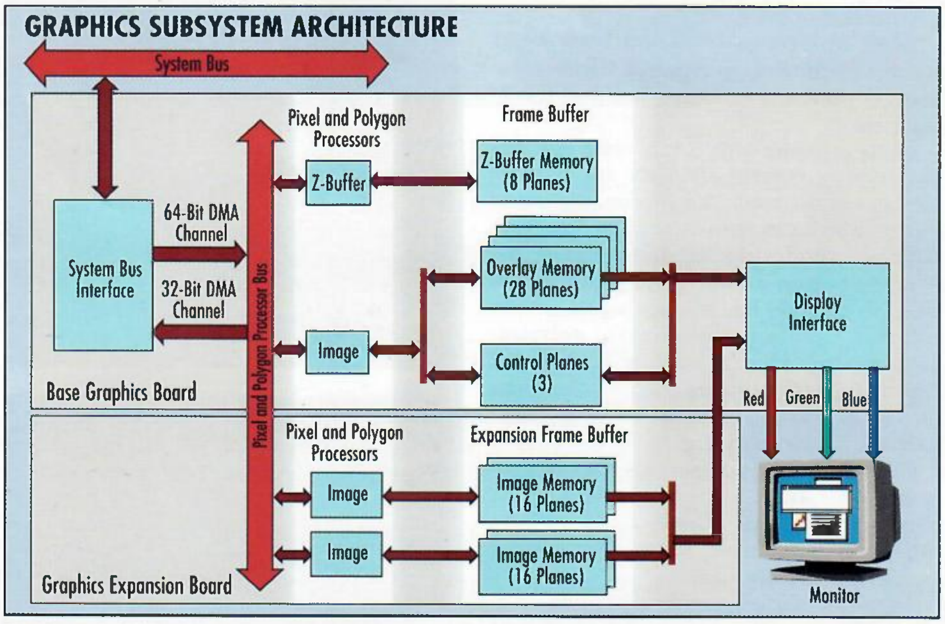


FIGURE 2: *A full graphics subsystem consists of two boards that are tightly coupled to both the CPUs and memory. These boards incorporate rasterizers (pixel and polygon processors), frame buffers, Z buffer, and additional overlay and control planes. The subsystem is coupled to main memory with a 64-bit DMA channel that transfers data from memory to the graphics subsystem and a 32-bit DMA channel that transfers data in the reverse direction.*

Applications that use the X windows' *Xlib* will run under X+.

Graphics software, as one might guess, is extremely important to Stardent. The centerpiece of this software is Stardent's Dynamic Object Rendering Environment (Doré) Library, a library of graphics subroutines used to provide dynamic object rendering.

Doré functions at a very high level, allowing applications programmers to concern themselves with high-level, abstract aspects of a scene instead of attending to every detail.

The library includes bindings for C and Fortran source code.

SPEAKING OBJECTIVELY

Doré is object-oriented, hiding the actual data from the application programmer. An object is a collection of data with an object type associated with it. Methods such as delete, print, pick and reference can be used with each object.

While Doré objects can be collected into hierarchies or groups, Doré does not use

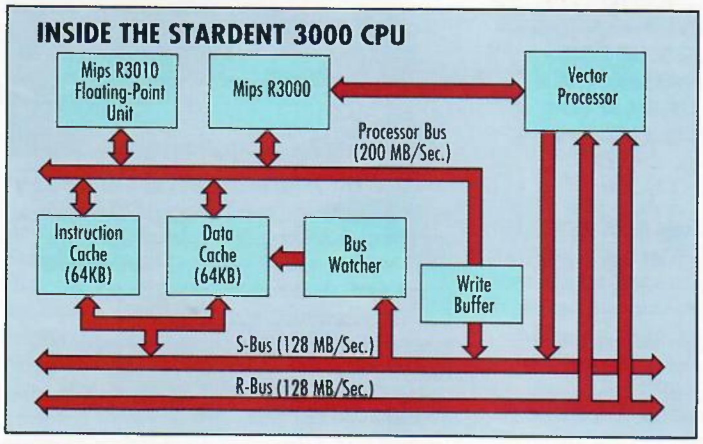


FIGURE 3: *Each Stardent 3000 CPU includes both a scalar and a vector processor. The scalar processor is a Mips R3000 chip running at 32MHz, accompanied by a R3010 floating-point co-processor, and the vector processor is proprietary to Stardent. The scalar processor handles all integer and almost all floating-point scalar operations, and during graphics operations functions as the scene database transverser. There are two independent 64KB direct-mapped caches—instruction and data—and four-deep write buffers. The vector unit performs all vector operations, and during graphics processing serves as the geometry engine.*

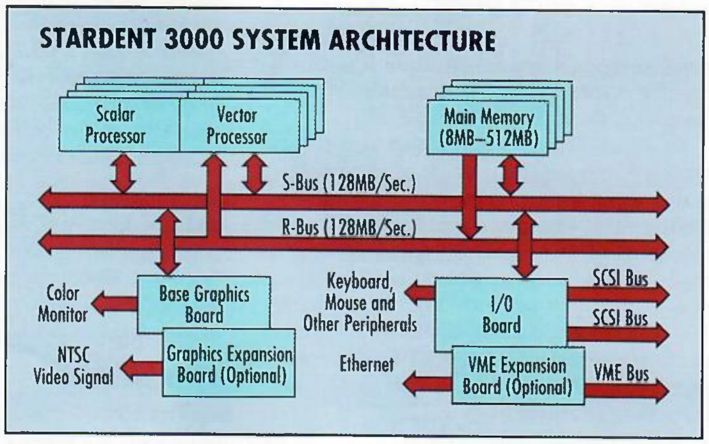


FIGURE 4: *The Stardent 3000 uses two buses for communication between the CPUs, memory, I/O subsystem and graphics subsystem. The buses carry 32-bit addresses and 64-bit data and operate at 16MHz; they are rated at 128MB per second each. The R-bus is dedicated to data transfers from memory to the vector processor and the S-bus handles all other transfers. The system can support a maximum of 512MB of memory.*

message passing or allow "children" objects to inherit their parents' attributes, as do other object-oriented systems.

Doré objects are divided into eight classes: primitives, geometric transformations, attributes, cameras, lights, text, frame and view.

While primitive objects may seem like simple drawing commands, they are actually used to build a scene description, the single copy of which can then be used to draw with different rendering techniques (Fig. 5). Primitive objects include point, marker, line, polyline, triangle list, triangle mesh, simple polygon, simple polygon mesh, polygon, polygon mesh, bilinear patch (e.g., Bezier or Hermite), non-uniform rational B-spline (NURBS) patch, quadric surface (e.g., sphere, cylinder, box), and torus.

Geometric transformations specify geometric operations to be performed on an object or group of objects. They include rotate, translate, scale, shear, look-at-from and matrix transformations.

Atttribute objects modify primitives, cameras and lights. The number of these objects is large; they modify a host of attributes, including color, reflectancy, transparency, field of view, diffuse or ambient lighting, back-face culling (the process of eliminating surfaces that aren't visible) and specularity (the concentrated reflection of light).

The cameras, lights and text classes allow users to control the placement and type of cameras, lights and text.

The frame and view objects are somewhat different from what is found in similar graphics libraries. A view is the result of rendering a set of display groups with a particular camera, set of lights and rendering level. It may be thought of as a single photograph. A frame, however, contains a group of views that are displayed in that frame.

A frame can be displayed on one or more devices. While the frame feature may seem unnecessary, it does provide for an extra measure of device independence; a frame can be viewed on the screen and then output to a printer without changing the objects, view or frame. Multiple frames can then be displayed on a single device simultaneously, if the device is capable of it.

Doré includes two built-in renderers: dynamic and production. A rendering style is associated with a view, not with a frame or device. The dynamic renderer uses the resources of the available hardware to generate images in interactive time—that is, several frames per second. The exact results that are obtained using dynamic rendering may vary from one Stardent model to the next and will likely change over time, according to Stardent.

A Stardent 3000 system that has hardware Gouraud shading will be able to provide such shad-ing at the dynamic level, while systems without such hardware can provide only flat shading.

The production renderer isn't strictly defined either, but currently consists of a constant time ray-tracer that is capable of creating scenes with specularity, environmental reflections and true shadows.

The advantage of defining the rendering styles so loosely is that, over time, as rendering algorithms and rendering hardware improve, Doré applications will take advantage of the advancements without having to make changes to the Doré interface code.

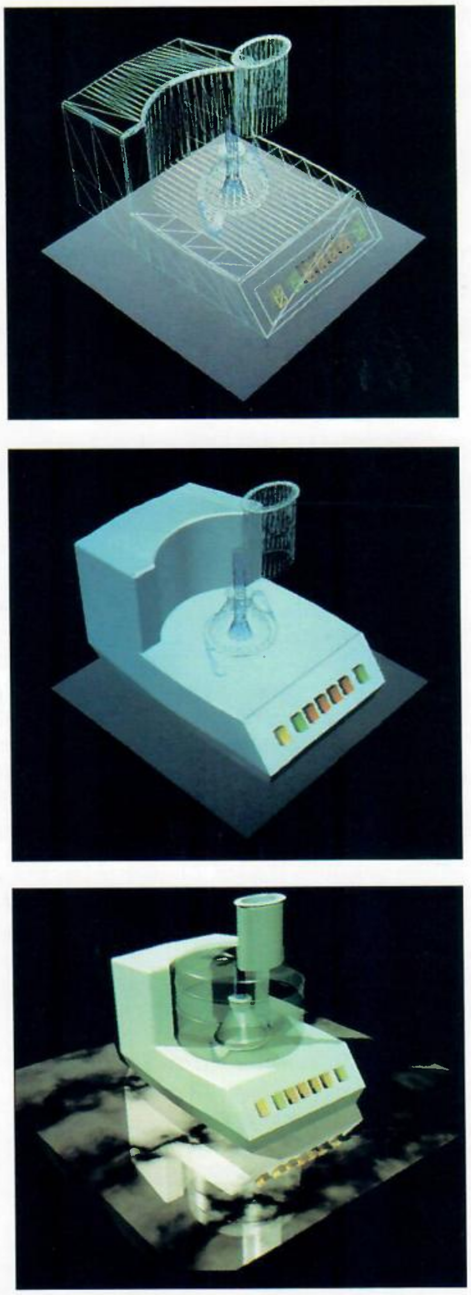


FIGURE 5: DORE'S OBJECT-ORIENTED DESIGN *allows a single scene to be described and then rendered using a variety of methods, for example, wireframe (A), Gouraud shading (B), or a globally shaded image with ray tracing.*

SUPER MINISUPER

The Stardent 3000 is housed in a 22-inch-wide-by-23-inch-deep cabinet that stands 50 inches high (Fig. 6). SCSI or VME expansion can be accomplished via an expansion cabinet whose dimensions and appearance are similar to those of the system cabinet. The 3000 does not require any special cooling equipment and can function in an office environment.

The system enclosure includes a 10-slot backplane, power supply, and room enough for as many as three 5¼-inch SCSI disks and one quarter-inch SCSI cartridge tape drive.

Slot 0 in the backplane is used for either an adapter that accepts two 6-by-9-inch VME boards or an expander board that connects to an external VME card cage; slot 1 holds the I/O subsystem board; slots 2 through 7 are used for memory and processor boards and can include a maximum of four boards of either type (at least one board of each type must be installed); slot 8 contains the base graphics board; and slot 9 is for the expansion graphics board.

The minimum-configuration Stardent 3000 Departmental Supercomputer is priced at $69,000 and includes one CPU, 32MB of memory and one 380MB SCSI disk drive. A minimum-configuration Visualization System with the same CPU, memory and drives that is also capable of single-buffered, true-color graphics or double-buffered, pseudo-color graphics is priced at $89,000.

A moderately sized Visualization System, including two CPUs, 64MB of memory, two 760MB disks, the expansion graphics board, a Fortran compiler and NFS, is priced at $176,900. Finally, a high-end, four-CPU server with 256MB of memory, no graphics, Fortran, NFS and three 760MB disk drives is priced at $384,300.

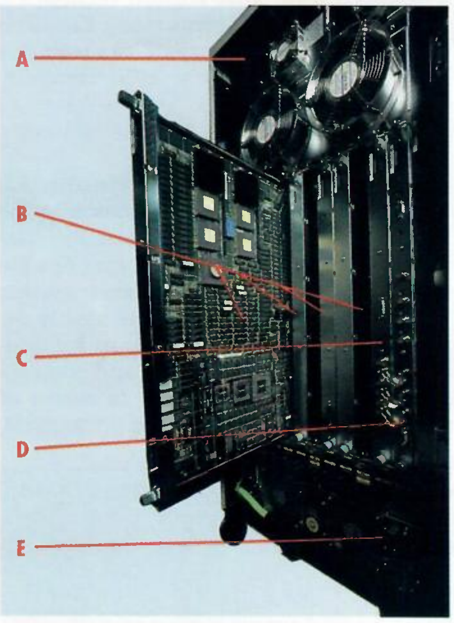


FIGURE 6: THE SYSTEM ENCLOSURE *includes bays (A) for as many as three 5¼-inch SCSI disk drives and one quarter-inch SCSI cartridge tape drive. There is also a 10-slot backplane—shown here with four CPU and/or memory boards (B), a graphics subsystem board (C) and an I/O subsystem board (D)—and the power supply (E).*

Although many industry analysts question whether the graphics minisupercomputer market is big enough to sustain the current group of players, innovations from Stardent Computer suggest that this new company's chances of survival are good. Packing one of the fastest CPUs that DR Labs has ever tested, coupled with a powerful graphics environment, the Stardent 3000 is an excellent choice for users running scientific computing applications.

Application Visualization

Included with every Stardent Visualization System is the Stardent Application Visualization System (AVS). AVS is software that, using the facilities of X windows and the Dynamic Object Rendering Environment (Doré) graphics library (previous AVS versions were based on Phigs+), provides a framework so researchers can take advantage of visualization techniques without having a great deal of graphics programming expertise. By taking care of many of the programming details, AVS allows researchers to devote more time to the problem being studied.

AVS combines three-dimensional interactive graphics with supercomputer performance to provide a real-time environment for analyzing and manipulating data from simulations or experiments. It is not intended for programmers; rather, it builds on the resources of graphics libraries to provide a complete environment for data analysis. It goes beyond rendering and viewing manipulations to include data input and transformation.

Because AVS is built on the X Window System, it uses a familiar point-and-click interface. The use of X windows increases AVS' portability.

AVS is designed for the analysis stage of computational simulation. It consists of a Network Editor and an Application Builder, and Image, Volume and Geometry Viewers.

The Network Editor is a tool for designing a visualization application. With the Network Editor, a user builds a network of independent computing components to form a processing network. This network specifies how data should be brought into AVS, how it might be filtered or otherwise manipulated, and how it should be displayed. The range of modules used in a processing network can include data input, filter, mapper and renderer modules. Additional modules, written in C or Fortran, can be added by the user.

Filters are available for applications from fields such as engineering, fluid dynamics and chemistry, and for commonly used formats such as the Brookhaven Protein Databank, Wavefront or MOVIE.BYU. Mapper modules transform data into geometry. Using one of these modules, users can select a visualization technique to apply to the data. Some of the sample mapper modules include isosurfaces of a 3-D field, 2-D slices of a 3-D volume, and 3-D meshes from 2-D elevation data. Renderer modules display geometry, images and volumes. Multiple rendering modules can be used, including a 3-D geometry renderer, image display renderer and several volume renderers.

The Application Builder allows a visualization application to be rapidly prototyped and delivered as a complete application to other users.

The Image and Volume Viewers provide comprehensive image processing and display capabilities, and extend these capabilities into three dimensions. AVS' image processing works with 8- or 24-bit data and floating-point data. Image display features include real-time pan and zoom, and "flipbook" animation. Filtering may be accomplished via table look-up operations to provide contrast stretching, pseudo-color and histogram balancing. Data resizing functions include interpolation, cropping and real-time warping. The three-dimensional image processing features include a real-time isosurface generator, real-time transparent volume renderer, use of geometric objects as slicing surfaces, dot surfaces and vector nets.

The Geometry Viewer allows data to be displayed with various renderings, for example, wireframe, Gourard or Phong shading. AVS supplies 16 independent point, directional or spotlight light sources. Stardent also allows the use of surface properties such as specularity, transparency, real-time texture mapping or anti-aliasing.

Reprinted from **DIGITAL REVIEW** March 5, 1990