The Future of HPC at SGI: Early Experience with SGI SN-1

Stephan Seidl, Wolfgang E. Nagel and Holger Brunst

Center for High Performance Computing (ZHR)
Dresden University of Technology
D-01062 Dresden, Germany
E-mail: {seidl | nagel | brunst}@zhr.tu-dresden.de

Abstract

In 1996, SGI introduced the Origin2000, an innovative and powerful computer architecture which enables shared memory applications to run on a very large number of nodes. Based on ccNUMA, today up to 512 CPUs can operate on a global address space. After the merger with Cray Research Inc., SGI had a lot of experience with CRAY T3E, also introduced in 1996. The T3E has been one of the very successful examples where a MPP architecture is used to efficiently solve many problems in the scientific technical market. Even today, roughly half of the 50 fastest computer systems are still CRAY T3E machines (Top500 list from November 1999).

Within SGI, the next generation of supercomputers Origin3000 (code name SN-1) has been announced. This machine looks very much like an Origin2000. Nevertheless, it is promised that this machine also has quite some features from CRAY T3E. At TU Dresden, we have a strong interest in the SN-1, and we were able to run some codes on this new machine; up to 32 nodes in Mountain View at the end of May 2000. Besides kernel and communication tests, we also looked deeply into the run time behavior of real applications from five of our end users.

1 Introduction

In 1998, after the commercial merger, we studied the SGI Origin2000 and the CRAY T3E 600/900 [13], concluding with certain wishes related to the product merger result. SGI, now alone again, just now has announced the Origin3000. Hence, this new machine is to be discussed with respect to our visions of the past. Dresden University of Technology (TU Dresden) will get an Origin3000 in August 2000 with 64 400 MHz MIPS R12k processors and 32 GBytes of memory.

The Origin3000 is a consequent successor of the Origin2000. As promised, Origin2000 binaries run on the Origin3000 without recompiling them and of course with better performance. The most important points inherited from the T3E are a lot of software and the understanding that a supercomputer operating system cannot only be a standard Unix with graphical enhancements. IRIX 6.5, as the common operating system for both generations, has made important steps to control the complicated situation of large shared memory application servers. In most cases, Miser guarantees the high-priority execution of production jobs against interactive loads under UNIX-like resource management. Nevertheless, both IRIX and Miser still have to be further improved to ensure that the current job scheduling system can enforce its plan of action under all circumstances.

2 SN-1 MIPS Architecture Overview

The Origin3000 belongs to the family of multiprocessor distributed shared memory computer systems. Its MIPS variant will have up to 512 cache-coherently working MIPS processors in a global address space. Four processors, with 8 MBytes of second-level cache each, are connected to a so-called Bedrock ASIC. This Bedrock ASIC works as a crossbar between the two CPU pairs, the locally mounted memory, the router entry/exit ports and a full-duplex I/O port. All the described components have their place in a 19 inch unit, this is the C-Brick. Four C-Bricks can be connected to an 8 port router, a 19 inch R-Brick, so that 4 router ports remain available to build up an extended hypercube topology. There are also a lot of other brick types for power supply, disks, and other I/O.

The MIPS R12k processors are similar to their predecessors, the R10k processors. The R12k processors run at 400 MHz. They are able to execute two floating-point operations per cycle. Each processor has a 32 KBytes two-way set associative primary data cache and a 32 KBytes two-way set associative instruction cache. As in the Origin2000, a directory-based protocol is applied for cache coherency, using extra memory hardware which is not accessible to the user.

If one puts all the rates, throughputs and bandwidth data from [15] and [16] together, the following result is obtained. All three have been doubled inside a C-Brick. Additionally, the second-level cache capacity has now been increased from 4 MBytes, in our Origin2000, to 8 MBytes. The only compromise that has been made is with respect to the C-Brick-R-Brick bandwidth. This bandwidth has been left at the same level, i.e. at 400 MBytes/s per CPU and direction. It is difficult to guess whether this compromise disturbs a given application, since a speed reduction is only visible when every CPU tries to access remote memory at the same time. At least the program kernel from [4] can show some performance difference on e.g. 8 processor Origin3000 systems with either 2 CPUs per C-Brick, or with 4 CPUs per C-Brick, respectively.

To provide real-time information, there are also separate wires in the Origin3000, as in the T3E. The method to access real-time is the same as for the Origin2000. The basic idea is to map a 64 bit counter into the user's address space via *mmap()*. During the measurements in Mountain View in May, a real-time timer monotonicity problem occurred occasionally. Detected with the help of VAMPIR [6], it was found that exactly one CPU pair sometimes had a time offset, with a couple of milliseconds difference. The problem stemmed from the PROM code and was fixed immediately. At the same time, one of our old wishes had been fulfilled, the timer resolution had been increased from 800 ns to 50 ns [9].

3 Performance Studies on SN-1 MIPS

3.1 PE Performance

For the per-PE performance, matrix multiplication behavior was studied. The results of the SN-1 are based on IRIX64 6.5.8m with F90 7.3.1.1m and C 7.3.1.1m. All the compilers were invoked with `-O3`, whereat the SGI compilers were under control of `abi=n32`, `isa=mips4` and `proc=r12k`. IEEE-754 Double-Precision was used as the floating-point format. Figures 1 and 2 show the results of the T3E-600, and the are exactly the same as in [13]. Figure 3, which is Fortran on the Origin2000, shows 30 per cent higher values than the ones we measured two

years ago [13]. For five of six index variants, the SN-1 Fortran case reached 90 per cent of the peak performance, i.e. more than 700 MFLOPS. The appropriate results for C, figure 6, could only be obtained with a page size of 4 MBytes, avoiding excessive TLB miss rates.

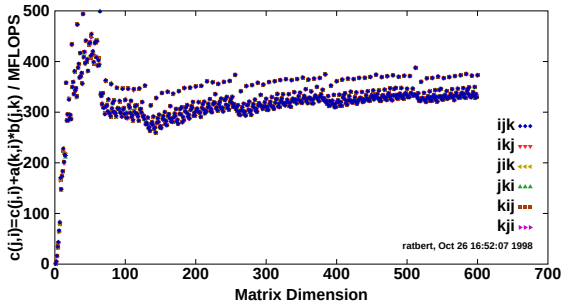

Figure 1: Fortran-coded matrix multiplication on T3E-600 with streams off

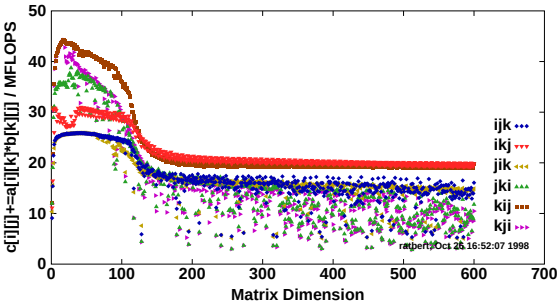

Figure 2: C-coded matrix multiplication on T3E-600 with streams off

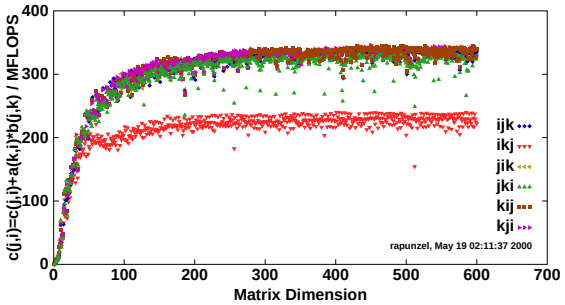

Figure 3: Fortran-coded matrix multiplication on 195 MHz Origin2000

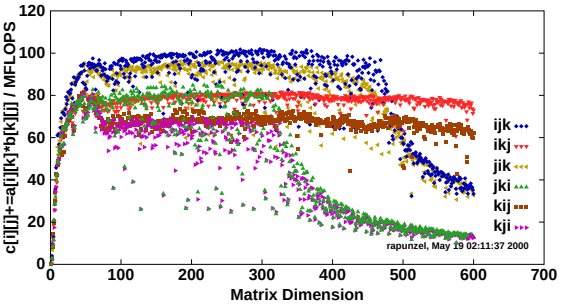

Figure 4: C-coded matrix multiplication on 195 MHz Origin2000

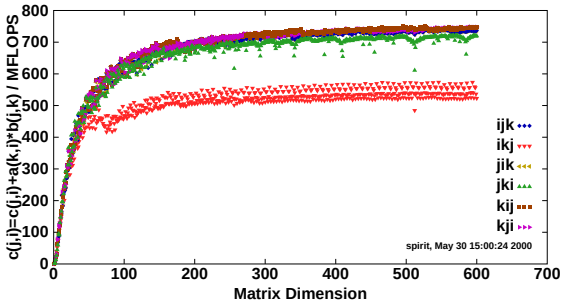

Figure 5: Fortran-coded matrix multiplication on 400 MHz SN-1

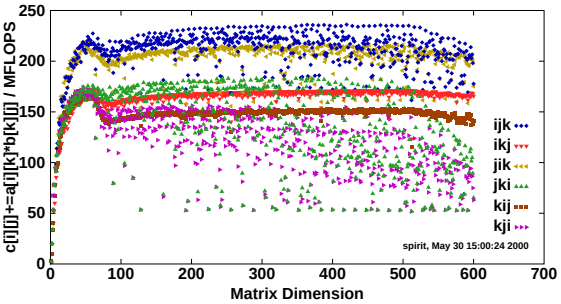

Figure 6: C-coded matrix multiplication on 400 MHz SN-1

3.2 MPI Performance

The SN-1 MPI point-to-point communication measurements are based on IRIX64 6.5.8m with MPI 3.2.0.7 (MPT 1.4) under `abi=64`, `isa=mips4` and `proc=r12k`. For short messages, the Origin2000 results look better than two years ago [13]. The appropriate SN-1 performance is higher than double of the Origin2000 performance, figures 7–10. Furthermore, the code of some collective operations was optimized, in the sense that one copy operation is avoided

when the buffers are symmetrically allocated. Application programmers should take this into account.

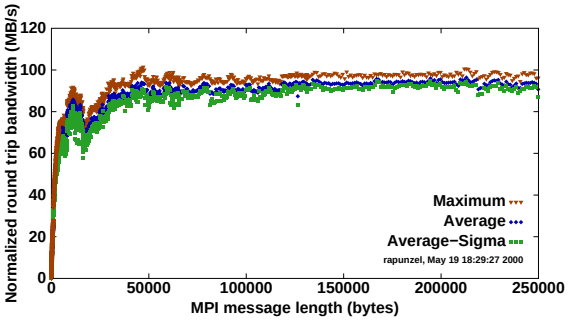

Figure 7: MPI point-to-point communication on 195 MHz Origin2000, long messages with *MPI_Send()*

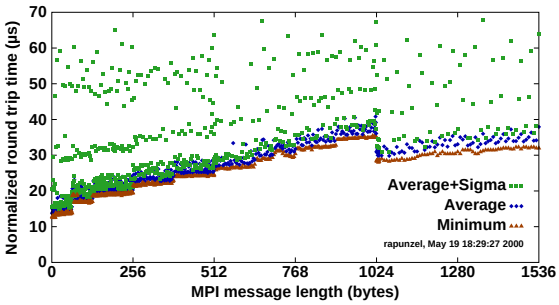

Figure 8: MPI point-to-point communication on 195 MHz Origin2000, short messages with *MPI_Send()*

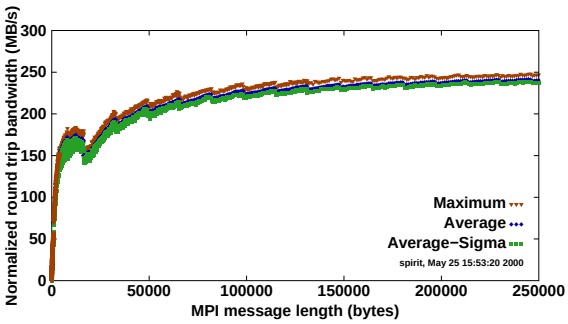

Figure 9: MPI point-to-point communication on 400 MHz SN-1, long messages with *MPI_Send()*

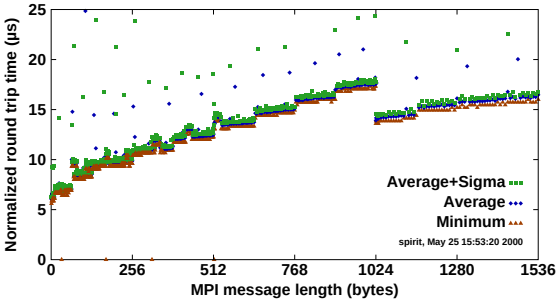

Figure 10: MPI point-to-point communication on 400 MHz SN-1, short messages with *MPI_Send()*

3.3 Performance of Real Applications

To study the run time behavior of real applications, some new software had to be applied. Firstly, the TU Dresden *libtrace.a*, which is only for research purposes, developed the capability to record hardware performance counter values using PCL [1]. The latter library, *libpcl.a*, had been developed by Rudolf Berrendorf from Forschungszentrum Juelich GmbH, Germany. Secondly, to visualize appropriate results, a new prototype of VAMPIR [2], which also handles traces containing counter values, was used.

Especially on SGI systems, we have the situation that the reading of a hardware counter value set needs to execute an *ioctl()* sys-call. On a 195 MHz Origin2000, one such *ioctl()* call takes 20 μs. On SN-1, half of this value was observed. Hence, to keep the measurement overhead low, one has to be very careful while instrumenting the applications. With respect to our examples, this overhead is less than one per cent. Furthermore, all the codes have been compiled with `-TARG:madd=OFF` to ensure that *multiply* and *add* are both counted separately.

Our first user program is MBCP. This is a MPI code which integrates multiple integrals given by polymer physics. If the inner integrand is sufficiently complicated, the parallel efficiency

of the newly developed algorithm [3] is not too small and can be held constant over a wide range of CPU numbers. Most of the communication time is spend in *MPI_Sendrecv_replace()*. Figures 11 and 12 show traditional VAMPIR time lines for 32 processes. On SN-1, this code ran 2.1 times faster than on Origin2000.

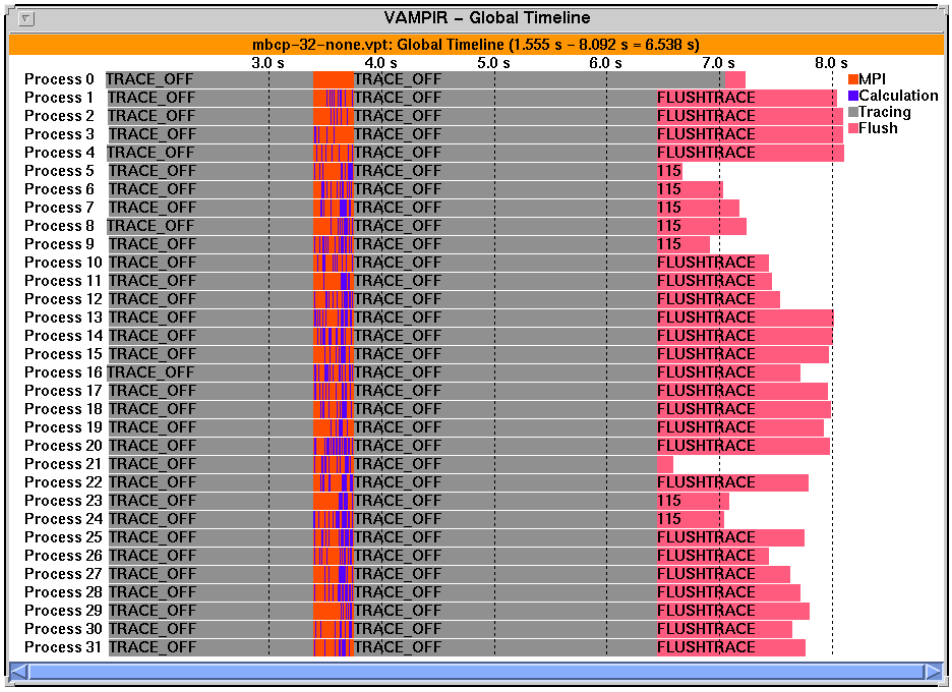

Figure 11: MBCP with 32 CPUs on 195 MHz Origin2000

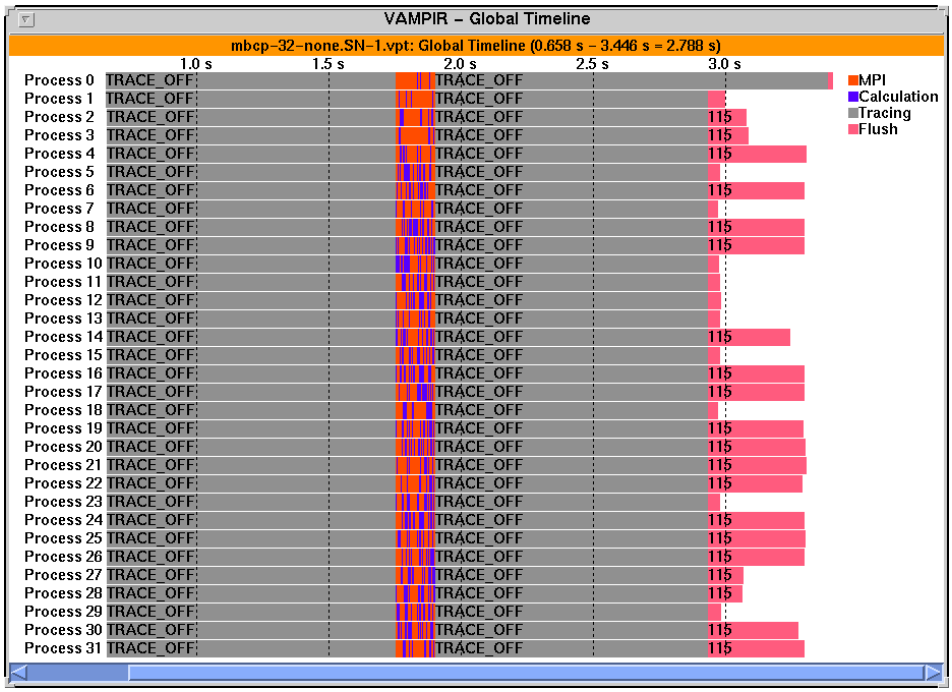

Figure 12: MBCP with 32 CPUs on 400 MHz SN-1

The second example which is presented here is FMC, by the first author. FMC is a parallel code under development, to study solutions of nonlinearly constrained nonlinear optimization problems, with at least piece-wise continuously differentiable functions. Controlled by a master, each process sequentially yields sundry solutions, applying algorithms which have their roots in [5], [8] and [11]. To improve performance, all the critical loops had been moved into so-called *home-brew*–BLASs to be optimized according to [12]. Figures 13–15 show appropriate results. On Origin3000, the sustained floating-point instruction rate equals 200 MFLOPS, and it is less than half this value on the Origin2000 and the T3E-1200. In figure 15, the peaks of 390 MFLOPS come from the level-3 type loop exactly described in [12].

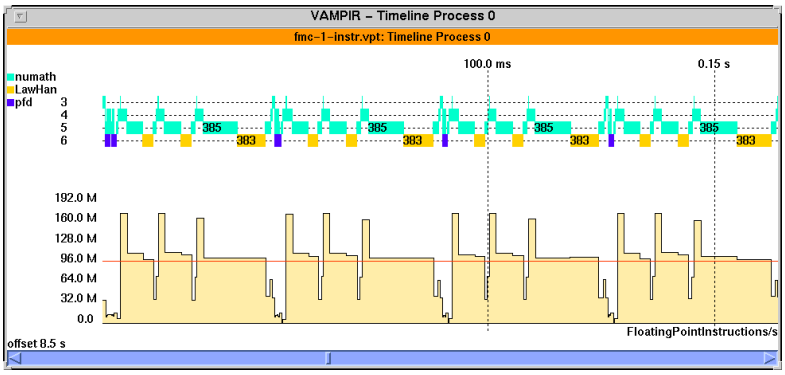

Figure 13: 4 FMC major iterations on 195 MHz Origin2000

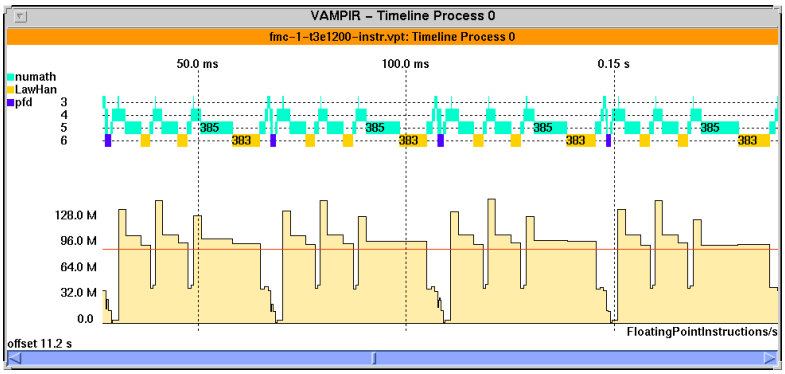

Figure 14: 4 FMC major iterations on T3E-1200

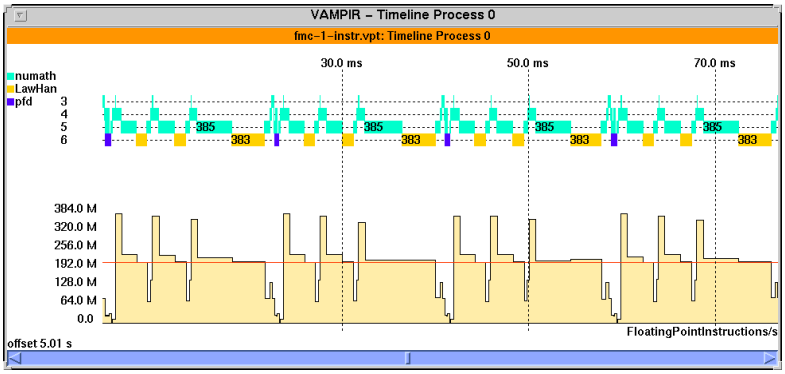

Figure 15: 4 FMC major iterations on 400 MHz SN-1

The third code, SMP, has been taken from stochastic dynamic optimization [10]. After initialization, the solver iteratively converges towards a fix-point. Figures 16 and 17 show the time lines of one process with 3 major iterations. The dominating MPI subroutine is *MPI_ALLGATHERV()*. Carefully expressed, one can say that the communication speed does not scale in the same way as the CPU speed. Involving all the 32 processes, figures 18 and 19 give the same picture. Zooming into the computational part, e.g. figures 20 and 21, the SN-1 is 5 times faster than the Origin2000, but the relatively slow *MPI_ALLGATHERV()* downgrades the SN-1 overall performance gain to a factor of 3. Further investigations should follow to check whether this effect comes from the unchanged C-Brick-R-Brick bandwidth, compared with the Origin2000, or from the changed topology. It should be denoted that a 48 processor Origin2000 can run a 32 process job in quite a balanced manner, while the 64 processor SN-1 only represents a half cube. Furthermore, the Mountain View test machine was an incomplete 64 processor system with slightly more than 32 CPUs.

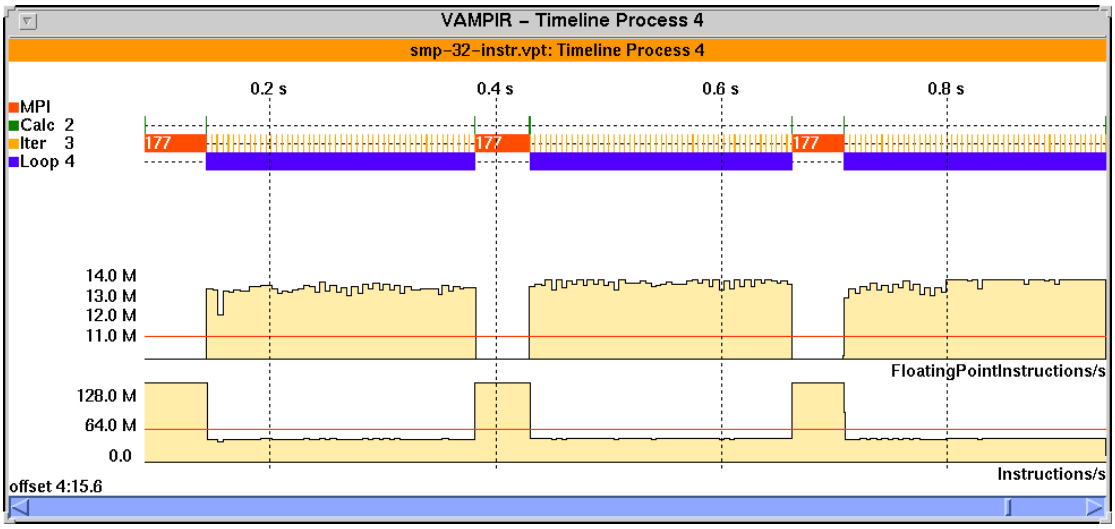

Figure 16: 3 SMP major iterations on 195 MHz Origin2000, one of 32 processes

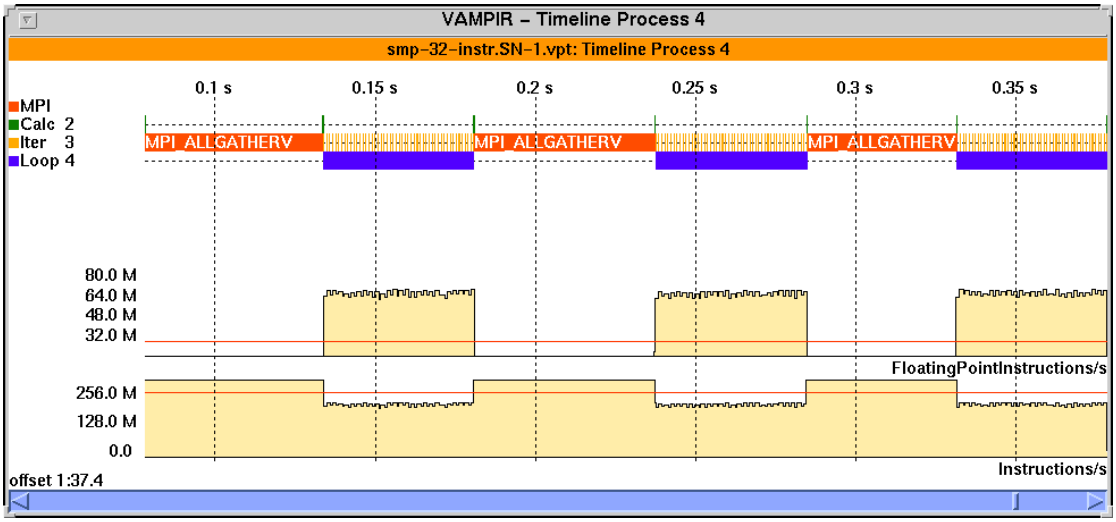

Figure 17: 3 SMP major iterations on 400 MHz SN-1, one of 32 processes

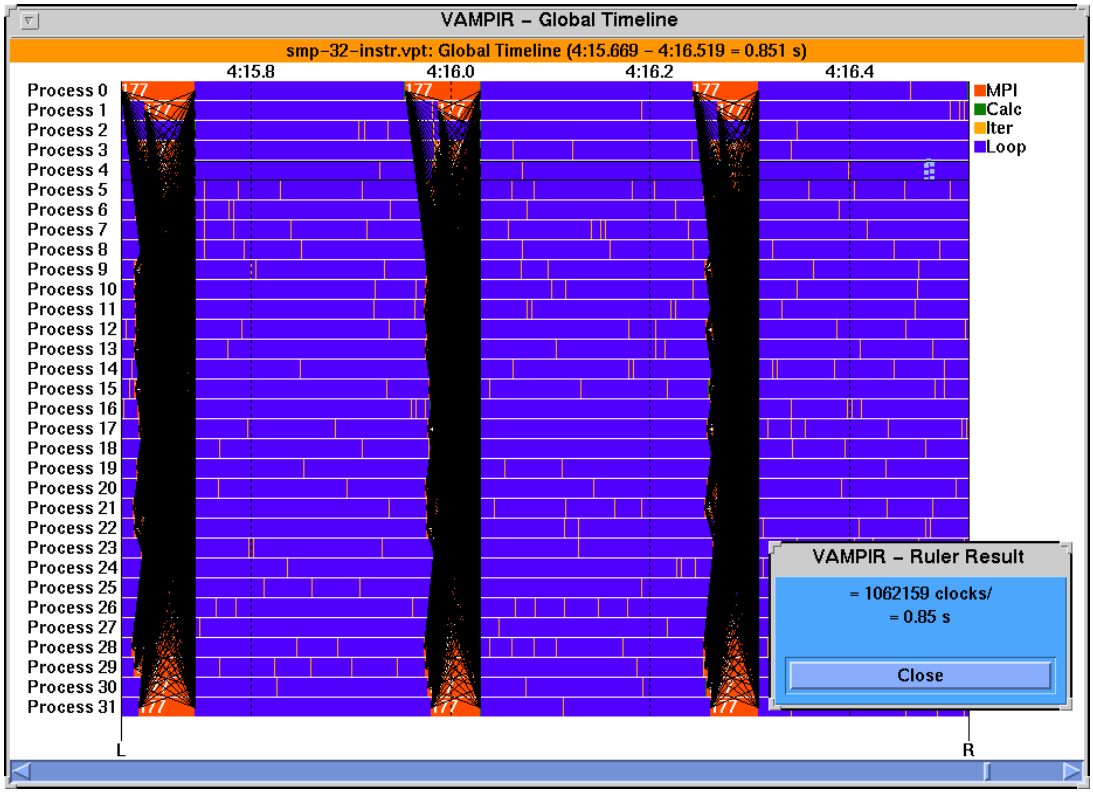

Figure 18: 3 SMP major iterations on 195 MHz Origin2000, all 32 processes

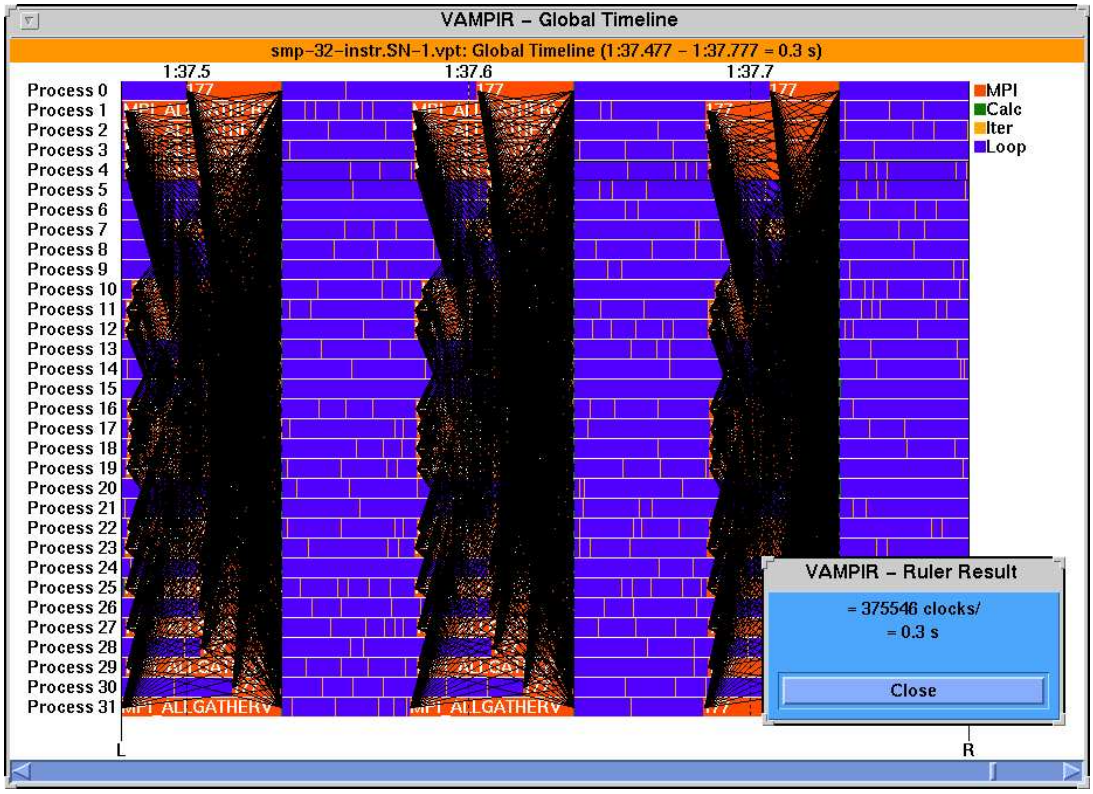

Figure 19: 3 SMP major iterations on 400 MHz SN-1, all 32 processes

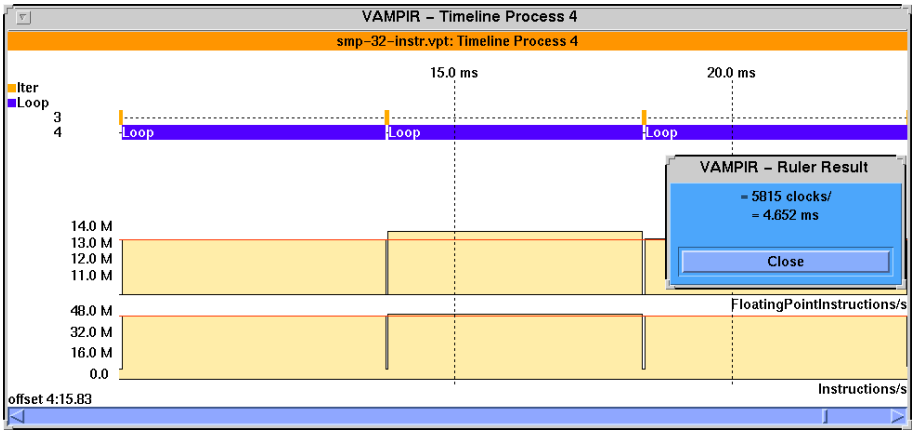

Figure 20: 3 SMP minor iterations on 195 MHz Origin2000, one of 32 processes

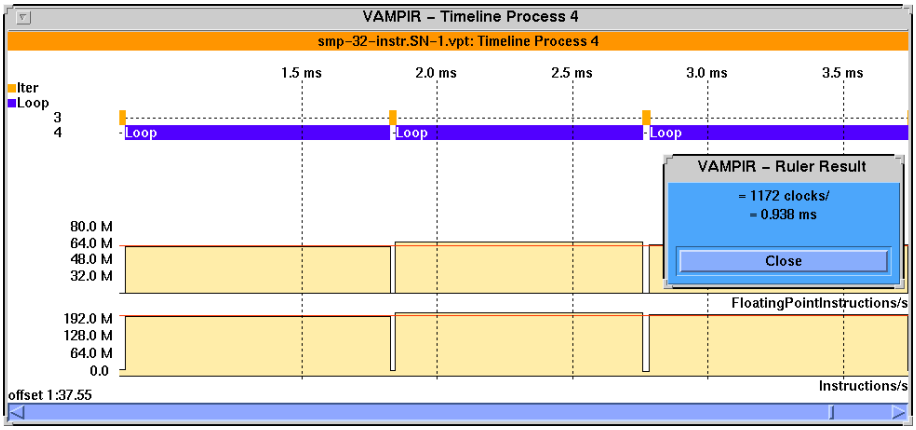

Figure 21: 3 SMP minor iterations on 400 MHz SN-1, one of 32 processes

Two further user codes [7][14] ran on SN-1 with more than double the speed of the Origin2000.

4 Conclusion

As promised, with the SN-1 SGI has given the user community a good new system. As usual, it took some time to develop this machine, and there was quite a delay in the first delivery of such systems. Nevertheless, a lot of software improvements have made the Origin3000 into a well-balanced distributed shared memory computer system. Our performance measurements show good values. In the future, the peak performance will certainly be too low, even so the sustained performance is surprisingly good. On the other hand, we look forward to IA64 and its peak/sustained values.

In 1996, the Origin2000, with its ccNUMA architecture, was a very modern system. Simply because this architecture now functions well, it has not lost any actuality and attractiveness. Perhaps, we will begin to see some more architectures of this type coming into the market soon.

Finally, we should keep in mind: Software will remain the challenge for the future.

Acknowledgments

We wish to thank T. Weselowski, R. Vogelsang and R. Wolff (all from SGI Germany) for supporting early SN-1 performance tests in Mountain View, CA.

References

[1] Berrendorf R., Ziegler H.: *PCL – The Performance Counter Library: A Common Interface to Access Hardware Performance Counters on Microprocessors*. http://www.fz-juelich.de/zam/PCL/.

[2] Brunst H., Nagel W.E. and Seidl S.: *Performance Tuning on Parallel Systems: All Problems Solved?*. Proc. of the PARA2000 Workshop on Applied Parallel Computing (Jun 18-21, 2000, Bergen, Norway).

[3] Friedel P., Bergmann J., Seidl S. and Nagel W.E.: *A New Parallelized Method for Numerical Solution of Multiple Integrals with Recursive Adaptive Step Width Choice*. to be published.

[4] Huedo E., Prieto M., Llorente I.M. and Tirado F.: *Impact of PE Mapping on CRAY T3E Message-Passing Performance*. Proc. of the Euro-Par 2000 (Aug 29 - Sep 1, 2000, Munich, Germany).

[5] Lawson C.L. and Hanson L.R.: *Solving Least Square Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974 (republished by SIAM, Philadelphia, ISBN 0-89871-356-0, 1995).

[6] Nagel W.E., Arnold A., Weber M., Hoppe H-C., and Solchenbach, K.: *VAMPIR: Visualization and Analysis of MPI Resources*. Supercomputer 63, Vol. 12, No. 1, 1996, pp. 69-80.

[7] Posdziech O., Grundmann R., Seidl S. and Nagel W.E.: *Three-dimensional Direct Numerical Simulation of Flow Problems with Electromagnetic Control on Parallel Systems*. Proc. of the ParCo99 Parallel Computing (Aug 17-20, 1999, Delft, The Netherlands), E.H. D'Hollander, G.R. Joubert, F.J. Peters and H.J. Sips, eds., Imperial College Press, London, 2000, ISBN 1-86094-235-0, pp. 176-184.

[8] Powell M.J.D.: *A fast algorithm for nonlinearly constrained optimization calculations*. Numerical Analysis. Proc. of the Biennial Conference (Jun 28 - Jul 1, 1977, Dundee), G. A. Watson, ed., ISBN 3-540-08538-6/0-387-08538-6, 1978, pp. 144-157.

[9] Purdy D.: *Personal SGI Information*. June 2000.

[10] Rudl J.: *Approximative Lösungen diskontierter Semi–Markovscher Entscheidungsprobleme mittels asynchroner Verfahren und Suboptimalitätstests*. Dissertation, TU Dresden, 2000.

[11] Schittkowski K.: *On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function*. Mathematische Operationsforschung und Statistik, K.-H. Elster, ed., Vol. 14, No. 2, 1983, pp. 197–216.

[12] Seidl S.: *Code Crumpling: A Straight Technique to Improve Loop Performance on Cache Based Systems.* Proc. of the Fifth European SGI/Cray MPP Workshop (Sep 9-10, 1999, CINECA, Bologna, Italy) http://www.cineca.it/mpp-workshop/proceedings.htm.

[13] Seidl S., Nagel W.E.: *CRAY T3E and SGI Origin2000: Merging Architectures from the User's Point of View.* Proc. of the Fourth European SGI/Cray MPP Workshop (Sep 10-11, 1998, IPP, Garching, Germany), H. Lederer and F. Hertweck, eds., IPP R/46, 1998, pp. 6-19.

[14] Stiller J. and Nagel W.E.: *MG – A Toolbox for Parallel Grid Adaption and Implementing Unstructured Multigrid Solvers.* Proc. of the ParCo99 Parallel Computing (Aug 17-20, 1999, Delft, The Netherlands), E.H. D'Hollander, G.R. Joubert, F.J. Peters and H.J. Sips, eds., Imperial College Press, London, 2000, ISBN 1-86094-235-0, pp. 391-399.

[15] Silicon Graphics Inc.: *Origin Servers Technical Report.* April 1997.

[16] Silicon Graphics Inc.: *SGI Origin 3000 Series Technical Configuration Owner's Guide.* Document Number 007-4311-001.