A Comparison of the Cray-2 Performance Before and After the Installation
of Memory Pseudo-Banking

Ronald D. Schmickley
David H. Bailey

NASA

A Comparison of the Cray-2 Performance Before and After the Installation
of Memory Pseudo-Banking

Ronald D. Schmickley
David H. Bailey
Sterling Federal Systems, Inc.
1121 San Antonio Road
Palo Alto, CA   94303

N/\S/\

A Comparison of the Cray-2 Performance Before and After the Installation of Memory Pseudo-Banking

Ronald D. Schmickley
and
David H. Bailey

ABSTRACT

A suite of thirteen large Fortran benchmark codes were run on a Cray-2 configured with "memory pseudo-banking" circuits, and floating point operation rates were measured for each under a variety of system load configurations. These were compared with similar "flop" measurements taken on the same system before installation of the pseudo-banking. A useful "memory access efficiency" parameter was defined and calculated for both sets of performance rates, allowing a crude quantitative measure of the improvement in "efficiency" due to pseudo-banking. Programs were categorized as either highly scalar (S) or highly vectorized (V) and either memory-intensive or register-intensive, giving four categories: S-memory, S-register, V-memory, and V-register. Using flop rates as a simple quantifier of these four categories (S-memory corresponds to low Mflops and V-register corresponds to high Mflops), a scatter plot of "efficiency" gain vs Mflops roughly illustrates the improvement in floating point processing speed due to pseudo-banking. On the Cray-2 system tested this improvement ranged from 1% for S-memory codes to about 12% for V-memory codes. No significant gains were made for V-register codes, which was to be expected.

A Comparison of the Cray-2 Performance Before and After the Installation of Memory Pseudo-Banking

Ronald D. Schmickley and David H. Bailey
June 16. 1986

In a previous paper, "A Performance Comparison of the Cray-2 and the Cray X-MP", March 14, 1986, megaflop rates were presented for a suite of thirteen Fortran floating-point intensive benchmark programs run on the Cray-2 under UNICOS and on the Cray X-MP/12 under COS. The main information from that paper was summarized in a table of performance measures for each program, consisting of four sets of megaflop rates:

- Cray-2 Stand-Alone: The performance of the Cray-2 running a single benchmark program on a single CPU with other CPUs idle.

- Cray-2 Simultaneous: The average performance of the four Cray-2 CPUs simultaneously running the same benchmark program.

- Cray-2 Normal: The performance of the Cray-2 running a single benchmark program on a single CPU with a typical daytime load of other jobs running in the other three CPUs.

- Cray X-MP Normal: The performance of the Cray X-MP/12 with a normal amount of swapping with other jobs.

In addition, for each benchmark program, the ratio of the Cray-2 normal load performance to the Cray X-MP/12 normal performance was presented. Those performance figures are reproduced in Table 1.

Since the original acquisition of those data, the hardware of the Cray-2 has been enhanced by the addition of memory pseudo-banking circuits. This pseudo-banking is designed to speed up program execution rates by reducing average memory contention between programs competing for access to the same memory bank. Subsequently, all thirteen benchmark programs were run again on the Cray-2 in Stand-Alone, Simultaneous, and Normal load modes. The results of those runs are presented in Table 2.

A direct comparison of Cray-2 megaflops for each run mode before and after the installation of pseudo-banking shows definite performance improvement for each program. Since significant improvement is to be expected only for vectorized, memory-intensive programs, it is useful to correlate each program's improvement with how "much" vectorization and memory access it generates during operation. Because there are currently no simple quantitative measures for such properties, the programs have been divided into four design categories, ranging from minimum vectorization and memory access to maximum vectorization and memory access.

1

Program Name	Cray-2 Stand-Alone	Cray-2 Simultaneous	Cray-2 Normal	Cray X-MP Normal	C2/XMP (Percent)
ARC3	42.91	26.98	30.11	51.35	58.64
ATRAN3S	12.68	10.81	10.43	21.10	49.43
BL3D	44.98	37.65	37.81	51.10	73.99
DERTRA	18.51	15.63	15.78	20.97	75.25
F3D	32.51	24.70	26.46	33.71	78.51
INS3D	54.55	38.93	41.35	52.75	78.39
LES	90.36	53.21	55.34	83.37	66.38
LLOOPS	9.58	9.28	9.01	14.89	60.50
MATEST	394.55	231.01	244.31	192.48	126.93
NASKERN2	94.17	53.72	57.28	91.21	62.80
PITEST	165.05	161.16	146.52	131.20	111.68
PNS3D	5.76	5.24	5.04	10.77	46.83
SUNSX	3.99	3.75	3.57	9.56	37.33
Average:					71.28

Table 1: Cray-2 and Cray X-MP Performance before Pseudo-Banking (MFLOPS)

Program Name	Cray-2 Stand-Alone	Cray-2 Simultaneous	Cray-2 Normal	Cray X-MP Normal	C2/XMP (Percent)
ARC3	47.72	33.67	35.04	51.35	68.24
ATRAN3S	14.05	12.52	11.80	21.10	55.92
BL3D	46.00	42.12	40.33	51.10	78.93
DERTRA	19.22	17.42	16.88	20.97	80.47
F3D	33.06	27.02	27.31	33.71	81.01
INS3D	59.74	47.44	45.68	52.75	86.60
LES	93.95	66.40	60.59	83.37	72.68
LLOOPS	10.20	10.02	9.62	14.89	64.63
MATEST	404.02	278.30	279.69	192.48	145.31
NASKERN2	98.86	63.66	66.43	91.21	72.84
PITEST	167.13	163.93	154.28	131.20	117.59
PNS3D	6.11	5.83	5.47	10.77	50.82
SUNSX	4.14	3.97	3.76	9.56	39.37
Average:					78.03

Table 2: Cray-2 and Cray X-MP Performance after Pseudo-Banking (MFLOPS)

Program Name	Cray-2 Stand-Alone (Percent)	Cray-2 Simultaneous (Percent)	Cray-2 Normal (Percent)	Program Category
ARC3	11.21	24.80	16.37	V-Memory
ATRAN3S	10.80	15.82	13.14	Partial-V
BL3D	2.27	11.87	6.67	V-Memory
DERTRA	3.84	11.45	6.94	Partial-V
F3D	1.69	9.39	3.20	Partial-V
INS3D	9.51	21.86	10.47	V-Memory
LES	3.97	24.79	9.49	V-Memory
LLOOPS	6.47	7.97	6.81	Partial-V
MATEST	2.40	20.47	14.48	V-Memory
NASKERN2	4.98	18.50	15.98	V-Memory
PITEST	1.26	1.72	5.30	V-Register
PNS3D	6.08	11.26	8.60	Scalar
SUNSX	3.76	5.87	5.42	Scalar
Average:	5.25	14.29	9.45	

Table 3: Cray-2 Performance Improvement due to Pseudo-Banking

Table 3 presents the figures for performance improvement in each run mode due to pseudo-banking, with a fourth column displaying a categorization of the program design. These four design categories are:

- Scalar: Insignificant percentage of run-time vector operations.

- Partial-V: Significant percentage of run-time vector operations.

- V-Register: Very high percentage of run-time vector operations, but register intensive only.

- V-Memory: Very high percentage of run-time vector operations that are memory intensive.

Note that the greatest increases are for "V-memory" programs running in the Simultaneous mode. But the PITEST "V-Register" program barely increases at all, since it generates little memory activity compared to the amount of computation.

An even more interesting statistic, the "memory access efficiency", can be defined as the ratio of a program's performance speed in Simultaneous run mode to its performance speed in Stand-Alone mode. Programs that are highly vectorized and memory intensive

Program Name	Efficiency Before (Percent)	Efficiency After (Percent)	Efficiency Gain (Percent)	Program Category
ARC3	62.88	70.56	7.68	V-Memory
ATRAN3S	85.25	89.11	3.86	Partial-V
BL3D	83.70	91.57	7.86	V-Memory
DERTRA	84.44	90.63	6.19	Partial-V
F3D	75.98	81.73	5.75	Partial-V
INS3D	71.37	79.41	8.05	V-Memory
LES	58.89	70.68	11.79	V-Memory
LLOOPS	96.87	98.24	1.37	Partial-V
MATEST	58.55	68.88	10.33	V-Memory
NASKERN2	57.05	64.39	7.35	V-Memory
PITEST	97.64	98.09	0.44	V-Register
PNS3D	90.97	95.42	4.45	Scalar
SUNSX	93.98	95.89	1.91	Scalar
Average:	78.27	84.20	5.93	

Table 4: Cray-2 Memory Access Efficiency Gain from Pseudo-Banking

can generate a lot of memory access contention when run simultaneously, and will thus have a "low" memory access efficiency. Therefore, for all except V-register programs, a low memory access efficiency generally indicates a fast program – so fast that memory access contention with other programs is a potential bottleneck! Therefore, the effect of pseudo-banking in increasing memory access efficiency can be demonstrated very dramatically by gains in the memory access efficiency statistics for these fast programs. These efficiencies are presented in Table 4, which shows memory access efficiency both before and after the installation of pseudo-banking, along with the efficiency gain due to pseudo-banking.

These figures are even more illuminating when translated from an alphabetic listing by program name to a scatter plot of efficiency gains versus program performance. Such a plot is presented in Figure 1, with the program performance represented as the logarithm (base 2) of the Stand-Alone megaflop values (after pseudo-banking), which is a reasonable measurement of "relative vectorization" among the programs. If one ignores the PITEST program, which is very highly vectorized and register intensive, then there is a clear, broad band of efficiency gains rising from less than 1% efficiency gain at the Scalar end to about 12% efficiency gain at the V-memory end. Although there is a lot of variation, this tends to confirm the positive effect of pseudo-banking, and even gives a rough quantitative measure of this effect, which lies between 7% and 12% gain in the "memory access efficiency" statistic for V-memory programs.
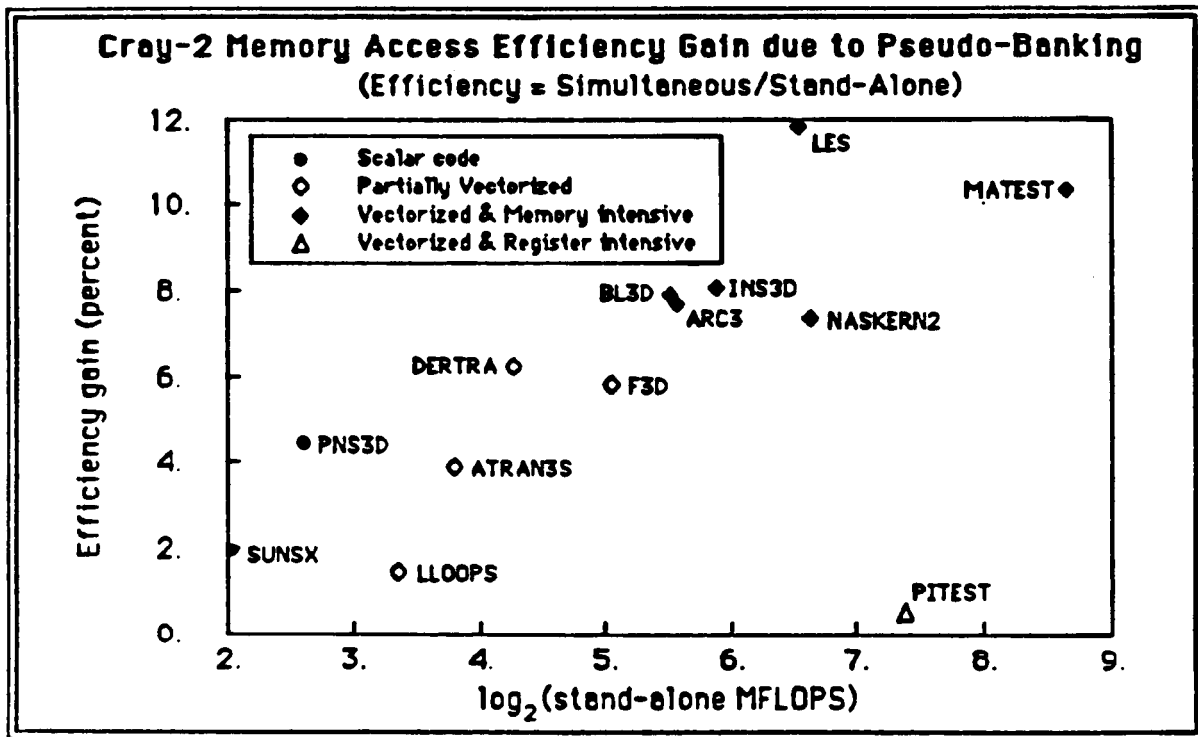
Figure 1: Gains in Memory Access Efficiency due to Pseudo-Banking

1. Report No. CR 177462	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A Comparison of the Cray-2 Performance Before and After the Installation of Memory Pseudo-Banking		5. Report Date June 16, 1987
		6. Performing Organization Code
7. Author(s) Ronald D. Schmickley and David H. Bailey		8. Performing Organization Report No. TN-86-7104-939-13
9. Performing Organization Name and Address Sterling Federal Systems, Inc. 1121 San Antonio Road Palo Alto, CA 94303		10. Work Unit No. K1707
		11. Contract or Grant No. NAS2-11555
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		
		14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Robert Carlson
MS-233-15
Ames Research Center
Moffett Field, CA 94035   FTS 448-6036   (415) 694-6036

16. Abstract

A suite of thirteen large Fortran benchmark codes were run on a Cray-2 configured with "memory pseudo-banking" circuits, and floating point operation rates were measured for each under a variety of system load configurations. These were compared with similar "flop" measurements taken on the same system before installation of the pseudo-banking. A useful "memory access efficiency" parameter was defined and calculated for both sets of performance rates, allowing a crude quantitative measure of the improvement in "efficiency" due to pseudo-banking. Programs were categorized as either highly scalar (S) or highly vectorized (V) and either memory-intensive or register-intensive, giving four categories: S-memory, S-register, V-memory, and V-register. Using flop rates as a simple quantifier of these four categories (S-memory corresponds to low Mflops and V-register corresponds to high Mflops), a scatter plot of "efficiency" gain vs Mflops roughly illustrates the improvement in floating point processing speed due to pseudo-banking. On the Cray-2 system tested this improvement ranged from 1% for S-memory codes to about 12% for V-memory codes. No significant gains were made for V-register codes, which was to be expected.

17. Key Words (Suggested by Author(s)) Cray-2, Pseudo-Banking, Performance Benchmarks	18. Distribution Statement Unclassified, Unlimited Star Category - 60

19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 5	22. Price