29

Elementary benchmarks of the NWT computer system and program performance

by
Shigeki HATAYAMA
National Aerospace Laboratory, 7-44-1 Jindaijihigashi-machi Chofu-shi Tokyo, Japan 182

Abstract

The NWT computer system of the NAL comprises two system administrators, 140 processing elements and a crossbar network, which operates as a distributed-memory message-passing MIMD computer. Each processing element itself is a vector computer. This paper presents measurements of the elementary characteristic parameters of the NWT with SIMD computing and with MIMD computings in the local and global memory access, and measurements of the maximum actual performance obtained when we execute programs for computation of the incompressible viscous flow in two- and three-dimensional lid-driven cavities in parallel on the 128-processing element system.

1. Introducing remarks

Most of the overheads incurred in the parallel processing arise from the time spent in system software routines supporting user programs. Hence the obtained results in this report apply only to the NWT system software available at the NAL during the period April to June 1993 which we call v1, and to the NWT system software during the period April to May 1994 which we call v2. This report indicates a comparison between the actual program performance of the NWT for v1 and one for v2, and the degree of the improvement , I_{21}, which is the ratio of the start-up time for v1 to the start-up time for v2. Hence the start-up time for v2 means an I_{21}-fold decrease of one for v1. Furthermore the dyadic operation in Sections 2, 3 and 5 means the multiplication of matrices (A=B*C), and the triadic operation the multiplication and addition of matrices (A=B*C+D). By the way all the benchmark programs were run with other users on the system.

2. The $(r_\infty, n_{1/2})$ benchmark

The characteristic parameters for a dyadic and triadic operation on a long-vector of length, n, with SIMD computing are given as follows: for dyads and v1

$$r_\infty = 3.897824 * 10^2 \ (Mflop/s), \tag{1}$$

$$n_{1/2} = 2.885078 * 10^2 \ (flop), \tag{2}$$

$$n_b = 2.473941 * 10^1 \ (flop); \tag{3}$$

for dyads and v2

$$r_\infty = 3.928646 * 10^2 \ (Mflop/s), \tag{4}$$

$$n_{1/2} = 3.014385 * 10^2 \ (flop), \tag{5}$$

$$n_b = 2.562819 * 10^1 \ (flop); \tag{6}$$

and for triads and v1

$$r_\infty = 5.514041 * 10^2 \ (Mflop/s), \tag{7}$$

$$n_{1/2} = 4.447699 * 10^2 \ (flop), \tag{8}$$

$$n_b = 4.197182 * 10^1 \ (flop); \tag{9}$$

for triads and v2

$$r_\infty = 5.603720 * 10^2 \ (Mflop/s), \tag{10}$$

$$n_{1/2} = 4.473542 * 10^2 \ (flop), \tag{11}$$

$$n_b = 4.149311 * 10^1 \ (flop), \tag{12}$$

where r_∞ is the maximum rate, $n_{1/2}$ the half-performance length and n_b the vector breakeven length which is the vector length above which the vector processing takes less time to perform the operation on a vector than the scalar processing. Hence the actual performance, r, with SIMD computing can be computed from

$$r = r_\infty/(1 + n_{1/2}/n) \ (Mflop/s). \tag{13}$$

Fig.1 shows a comparison between v1 and v2 of actual processing rates as a function of vector length with SIMD computation on a single processing element. Futhermore, since the start-up time t_0 is $t_0 = t_\infty^{-1} n_{1/2}$, I_{21}=0.9647 (dyads) and I_{21}=1.0104 (triads) (see [1],[3] for details). Finally we note the followings:

(1) For a dyadic and triadic operation on the short-vector length and v2, we obtain $r_\infty = 4.286826 * 10^2$ (Mflop/s,dyads) and $r_\infty = 5.624414 * 10^2$ (Mflop/s,triads).

(2) The theoretical peak performance of each processing element derived from its architecture is 2-fold for dyads and 3-fold for triads as many as the maximum rate.

(3) Hence we can estimate that the hardware performance per one arithmetic pipeline of the processing element is $8.505137 * 10^2$ (Mflop/s) on the average.

3. The $(r_\infty, s_{1/2})$ benchmark

When the efficiency of scheduling is perfect, the characteristic parameters for a dyadic and triadic operation on an amount of computational work, s, with MIMD computing in the local memory access are given as follows: for dyads and v1

$$r_\infty = 3.858962 * 10^2 pe\ (Mflop/s), \quad (14)$$

$$s_{1/2} = -2.497182 * 10^3 + 1.325439 * 10^4 pe\ (flop), \quad (15)$$

$$s_b = (1.325439 * 10^4 pe - 2.497182 * 10^3)/(pe-1)\ (flop); \quad (16)$$

for dyads and v2

$$r_\infty = 3.850414 * 10^2 pe\ (Mflop/s), \quad (17)$$

$$s_{1/2} = -2.441883 * 10^3 + 1.092176 * 10^4 pe\ (flop), \quad (18)$$

$$s_b = (1.092176 * 10^4 pe - 2.441883 * 10^3)/(pe-1)\ (flop); \quad (19)$$

and for triads and v1

$$r_\infty = 5.468374 * 10^2 pe\ (Mflop/s), \quad (20)$$

$$s_{1/2} = -3.451546 * 10^3 + 1.878137 * 10^4 pe\ (flop), \quad (21)$$

$$s_b = (1.878137 * 10^4 pe - 3.451546 * 10^3)/(pe-1)\ (flop); \quad (22)$$

for triads and v2

$$r_\infty = 5.445832 * 10^2 pe\ (Mflop/s), \quad (23)$$

$$s_{1/2} = -3.496416 * 10^3 + 1.565948 * 10^4 pe\ (flop), \quad (24)$$

$$s_b = (1.565948 * 10^4 pe - 3.496416 * 10^3)/(pe-1)\ (flop), \quad (25)$$

where pe is the selected number of processing elements, $s_{1/2}$ the half-performance grain size and s_b the breakeven grain size above which it is faster to suffer the synchronization overhead and split the job between the pe processing elements than to avoid synchronization altogether by using a single processing element. Hence the actual performance, r, with MIMD computing in the local memory access can be computed from

$$r = r_\infty/(1 + s_{1/2}/s)\ (Mflop/s). \quad (26)$$

Fig.2 shows a comparison between v1 and v2 of actual processing rates as a function of the amount of arithmetic operations with MIMD computation in the local memory access when $pe = 64$. Futhermore, since the start-up time is $t_0 = t_\infty^{-1} s_{1/2}$, $I_{21} = 1.1656 \sim 1.2804$, 1.2128 on the average (dyads) and $I_{21} = 1.1876 \sim 1.2610$, 1.2176 on the average (triads) (see [1],[3] for details).Finally we note that the values of r_∞ well coincide with ones in Section 2 when $pe = 1$.

4. A variation of the pingpong benchmark

The characteristic parameters on a long-message of length, n, with the data transfer between the global and local memory spaces are given as follows: for v1

$$r_\infty = 7.554390 * 10^2 pe\ (Mbyte/s), \quad (27)$$

$$n_{1/2} = -3.315542 * 10^4 + 1.834119 * 10^5 pe\ (byte); \quad (28)$$

and for v2

$$r_\infty = 7.444648 * 10^2 pe\ (Mbyte/s), \quad (29)$$

$$n_{1/2} = -1.372388 * 10^4 + 5.285085 * 10^4 pe\ (byte), \quad (30)$$

where r_∞ is the maximum bandwidth and $n_{1/2}$ the half-performance message length. Hence the actual performance, r, with the data transfer between the global and local memory spaces can be computed from

$$r = r_\infty/(1 + n_{1/2}/n)\ (Mbyte/s). \quad (31)$$

Fig.3 shows a comparison between v1 and v2 of actual data transfer rates as a function of message length when $pe = 1 \sim 16$. Futhermore, since the start-up time is $t_0 = (1.024^2 t_\infty)^{-1} n_{1/2}$, $I_{21} = 3.4212 \sim 3.8765$, 3.6082 on the average (see [2],[3] for details).

5. The $(\hat{r}_\infty, \hat{s}_{1/2}, f_{1/2})$ benchmark

The characteristic parameters for a dyadic and triadic operation on an amount of computational work, s, and a computational intensity, f, with MIMD computing in the global memory access are given as follows: for dyads and v1

$$r_\infty = 5.857613 * 10^1 pe\ (Mflop/s), \quad (32)$$

$$s_{1/2} = 469.8217 + 30817.60pe + 73.00503pe^2\ (flop), \quad (33)$$

$$f_{1/2} = 1.932356\ (flop/I/O\ word); \quad (34)$$

for dyads and v2

$$r_\infty = 5.824027 * 10^1 pe\ (Mflop/s), \quad (35)$$

$$s_{1/2} = -2387.097 + 13299.69pe + 56.36769pe^2\ (flop), \quad (36)$$

$$f_{1/2} = 1.972983\ (flop/I/O\ word); \quad (37)$$

for triads and v1

$$r_\infty = 8.619480 * 10^1 pe\ (Mflop/s), \quad (38)$$

$$s_{1/2} = -1021.786 + 49481.97pe + 121.3604pe^2\ (flop), \quad (39)$$

$$f_{1/2} = 2.783590\ (flop/I/O\ word); \quad (40)$$

for triads and v2

$$r_\infty = 8.784405 * 10^1 pe\ (Mflop/s), \quad (41)$$

$$s_{1/2} = -5098.177 + 23233.29pe + 95.54521pe^2\ (flop), \quad (42)$$

$$f_{1/2} = 2.790488\ (flop/I/O\ word), \quad (43)$$

and

$$\hat{r}_\infty = r_\infty(1 + f_{1/2}/f)\ (Mflop/s), \quad (44)$$

$$\hat{s}_{1/2} = s_{1/2}(1 + f_{1/2}/f)\ (flop), \quad (45)$$

where $\hat{r}_\infty$ is the peak maximum rate, $\hat{s}_{1/2}$ the peak half-performance grain size and $f_{1/2}$ the half-performance intensity. Hence the actual performance, r, with MIMD computing in the global memory access can be computed from

$$r = \hat{r}_\infty/(1 + \hat{s}_{1/2}/s + f_{1/2}/f)\ (Mflop/s). \tag{46}$$

Fig.4 shows a comparison between v1 and v2 of actual processing rates as a function of the amount of arithmetic operations with MIMD computation in the global memory access when $pe = 64$. Futhermore, since the start-up time is $t_0 = t_\infty^{-1} s_{1/2}$, $I_{21} = 1.9266 \sim 2.8392$, 2.4259 on the average (dyads) and $I_{21} = 1.8540 \sim 2.6953$, 2.2709 on the average (triads) (see [2],[3] for details). Finally we note the followings:

(1) The degree of degradation of peak maximum rate due to communication overheads when $s \to \infty$ is 1/6.797068 (v1), 1/6.918949 (v2) for dyads and 1/6.56718 (v1), 1/6.580976 (v2) for triads, because data transferred from the global memory to the local memory uses only once, i.e., $f = 1/3$ for dyads and $f = 1/2$ for triads.

(2) The values of $\hat{r}_\infty$ for v1 and v2 computed from (44) when $f = 1/3$ for dyads and $f = 1/2$ for triads well coincide with ones of r_∞ in Section 3.

(3) The number of reference of data transferred to the local memory to be required to reach 90% of the peak maximum rate when $s \to \infty$ can be computed from (46), and is 53 for dyads and 50 for triads both v1 and v2.

6. Program performance

We consider an incompressible viscous flow in two- and three-dimensional cavities by a uniformly moving upper surface, and measure the maximum actual rate obtained when we execute programs to compute the flow in the cavity in parallel on the 128-processing element system of the NWT.

6.1 Square cavity problem

Let H be the depth of cavity, L the width of cavity, and $H = L = 1$. The dimensionless steady-state stream-function vorticity conservation form of the two-dimensional incompressible Navier-Stokes equations is as follows:

$$\psi_y \zeta_x - \psi_x \zeta_y = \frac{1}{Re}(\zeta_{xx} + \zeta_{yy}), \tag{47}$$

$$\psi_{xx} + \psi_{yy} = -\zeta, \tag{48}$$

where ζ is the vorticity, ψ the stream-function, Re the Reynolds number, and x and y the axtial and normal coordinates, respectively.

The boundary conditions on (48) for flow in a lid-driven cavity with the upper surface translating to the right with uniform velocity $u = 1$ and with no flow at the other boundaries are at the upper surface

$$\psi_y = 1,\ \psi_x = 0,\ \psi = 0, \tag{49}$$

and at the bottom, left and right surfaces

$$\psi_y = 0,\ \psi_x = 0,\ \psi = 0. \tag{50}$$

The boundary conditions on (47) are obtained by applying the boundary conditions for the stream-function at the solid boundaries as follows:

$$\zeta = -(\psi_{xx} + \psi_{yy}),\ at\ x = 0, 1\ and\ y = 0, 1. \tag{51}$$

When we could obtain the steady-state solutions ζ and ψ, the primitive variables can be computed from the following equations:

$$u = \psi_y,\ v = -\psi_x, \tag{52}$$

$$p_{xx} + p_{yy} = (v\zeta)_x - (u\zeta)_y = \sigma, \tag{53}$$

where p, u and v are the total pressure, velocity component in the x-direction and velocity component in the y-direction, respectively.

The boundary conditions on (52) are at the upper surface

$$u = 1,\ v = 0, \tag{54}$$

and at the bottom, left and right surfaces

$$u = 0,\ v = 0. \tag{55}$$

The following Neumann boundary conditions on (53) are obtained by applying the momentum equations at the solid boundaries:

$$p_x = v\zeta - \frac{1}{Re}\zeta_y,\ at\ x = 0, 1, \tag{56}$$

$$p_y = -u\zeta + \frac{1}{Re}\zeta_x,\ at\ y = 0, 1. \tag{57}$$

Solutions to (53) with (56) and (57) are unique within an arbitrary constant, which can be determined by using the relation

$$\int_{y=0}^{1}\int_{x=0}^{1} p dx dy = constant. \tag{58}$$

The existence of a solution for (53) with (56) and (57) requires the satisfaction of the following compatibility condition:

$$\int_{y=0}^{1}\int_{x=0}^{1} \sigma dx dy = \int p_n dS, \tag{59}$$

where n is the outward normal to the boundary contour S, enclosing the solution domain.

In order to obtain numerical solutions, we adopt finite-difference approximations for (47)-(59) on non-staggered and uniform grids. All partial derivatives are approximated using second order accurate formulas. We note that the finite-difference approximation for (51) are obtained from (48)-(50) by enforcing reflection at the boundaries, and that for the compatibility condition (59) to be exactly satisfied on a non-staggered grid, we must use the consistent finite-difference approximations for (53), (56) and (57) [4].

We use the following four iterative methods to obtain the steady-state solutions of ζ, ψ and p:

(1) Jacobi's method,
(2) red-black ordering,
(3) conjugate gradient (CG) method,
(4) alternating direction implicit (ADI) method.

Futhermore the solutions of u and v are computed from (52) after obtained ψ.

A large amount of numerical information on the Reynolds number range of $Re = 10^{-6} \sim 10^{5}$ has been collected during this investigation. See [5] for details. Fig.5 shows the square cavity flow at $Re = 20,000$.

6.2 Cubical and 3-D narrow cavity problem

Let L be the length of cavity in the x-direction, H the height of cavity in the z-direction, and W the breadth of cavity in the y-direction. For a cubical cavity, $L = H = W = 1$ and for a 3-D narrow cavity, $L = H = 1$ and $\lambda = W/L < 1$. The dimensionless divergence form of the three-dimensional incompressible Navier-Stokes equations is as follows:

$$u_x + v_y + w_z = 0, \tag{60}$$

$$u_t + uu_x + vu_y + wu_z = -p_x + \frac{1}{Re}(u_{xx} + u_{yy} + u_{zz}), \tag{61}$$

$$v_t + uv_x + vv_y + wv_z = -p_y + \frac{1}{Re}(v_{xx} + v_{yy} + v_{zz}), \tag{62}$$

$$w_t + uw_x + vw_y + ww_z = -p_z + \frac{1}{Re}(w_{xx} + w_{yy} + w_{zz}), \tag{63}$$

and

$$(p_x)_x + (p_y)_y + (p_z)_z = -D_t - Q, \tag{64}$$

where

$$D = u_x + v_y + w_z, \tag{65}$$

$$\begin{aligned} Q &= (uu_x + vu_y + wu_z)_x + (uv_x + vv_y + wv_z)_y \\ &\quad + (uw_x + vw_y + ww_z)_z, \end{aligned} \tag{66}$$

and Re is the Reynolds number, p the static pressure, t the time, u, v and w the velocity components in the x-, y- and z-directions, respectively.

The boundary conditions on the velocity for flow in a lid-driven cavity with the upper surface translating to the right with uniform velocity $u = 1$ and with no flow at the other boundaries are at the upper surface

$$u = 1, \ v = 0, \ w = 0, \tag{67}$$

and at the other surfaces

$$u = 0, \ v = 0, \ w = 0. \tag{68}$$

The following Neumann boundary conditions on the pressure are obtained by applying the momentum equations at the solid boundaries: at the $y = 0$ and $y = 1$ surfaces

$$-p_x = uu_x + vu_y + wu_z - \frac{1}{Re}(u_{xx} + u_{yy} + u_{zz}), \tag{69}$$

at the $x = 0$ and $x = 1$ surfaces

$$-p_y = uv_x + vv_y + wv_z - \frac{1}{Re}(v_{xx} + v_{yy} + v_{zz}), \tag{70}$$

and at the $z = 0$ and $z = 1$ surfaces

$$-p_z = uw_x + vw_y + ww_z - \frac{1}{Re}(w_{xx} + w_{yy} + w_{zz}). \tag{71}$$

The momentum equations (61)-(63) are solved for the velocity components by marching in time. Equation (64) is a second order elliptic partial differential equation of the Poisson type, which is explicitly independent of the Reynolds number, because the diffusion terms are eliminated by the continuity equation (60). The governing equations (60)-(64) for the primitive variables are not independent. The continuity equation (60) is eliminated from the system of equations, and is iteratively satisfied through the solution of the pressure equation (64) as following: we can approximate the unsteady term D_t for the dilation in (64) by $(D^{n+1} - D^n)/\Delta t$, where the superscripts n and $n+1$ refer to the time levels t and $t + \Delta t$, respectively. In order to attempt to be satisfied (60), we set $D^{n+1} = 0$.

Solutions to (64) with (69)-(71) are unique within an arbitrary constant, which can be determined by using the relation

$$\int_{z=0}^{1}\int_{y=0}^{\lambda}\int_{x=0}^{1} p\,dxdydz = constant. \tag{72}$$

The existence of a solution for (64) with (69)-(71) requires the satisfaction of the following compatibility condition:

$$\int_{z=0}^{1}\int_{y=0}^{\lambda}\int_{x=0}^{1} -(D_t + Q)dxdydz = \int\int p_n d\partial V, \tag{73}$$

where n is the outward normal to the boundary surface ∂V, enclosing the solution domain.

In order to obtain numerical solutions, we adopt finite-difference approximations for (61)-(73) on non-staggered and uniform grids. All the other partial derivatives except for the time are approximated using second order accurate formulas. We note that for the compatibility condition (73) to be exactly satisfied on a non-staggered grid, we must use the consistent finite-difference approximations for (64) and (69)-(71) of which method is described in [6] with respect to two-dimensional case. Futhermore we use the four iterative methods mentioned in Subsection 6.1 to obtain the pressure solution.

A large amount of numerical information on the Reynolds number range of $Re = 10^{1} \sim 10^{4}$ has been collected during this investigation. Fig.6 shows the cubical cavity flow at $Re = 100$.

6.3 Measurements of program performance

(1) Definition of the symbols used below

r_1 = actual rate of the part of the solution method in programs,

r_2 = actual rate of programs in case of including the computation part of convergence decision,

r_3 = actual rate of programs in case of not including the computation part of convergence decision,

r_4 = actual rate of programs in case of intentionally suppressing execution of the explicit memory access statements,

r_5 = actual rate of programs in case of using the coalescing technique,

$d_{ma} = r_4/r_3$ = degree of degradation of actual rate due to the memory access bottleneck.

(2) Maximum program perfoermance

Table 1 shows maximum actual rates for each solution method of programs to compute the square cavity flow with respect to the v1 and v2 system software when $pe = 128$. Table 2 and 3 show maximum actual rates for each solution method of programs to compute the cubical and 3-D narrow cavity flow with respect to the v1 and v2 system software when $pe = 128$.

(3) Remarks

We remark here the followings to obtained results: (a)To apply ADI method to 2-D problem is not advisable, because of incurring a large amount of communication overheads by the global memory access. (b)The reason that actual rates of the red-black method are inferior to ones of the other methods is for the amount of works in each parallel section to become half the others. (c)The reason that the values of d_{ma} for Jacobi method and 3-D problem are larger than the others is because dimensional computer variables are stored to and retrieved from memory each iteration. (d)For 3-D problem, actual rates for narrow cavity problem rise than for cubical cavity problem except ADI, but vice versa for shallow cavity problem. (e) Finally we mention that the solution method shown larger actual rate is not always advantageous to obtain convergence solutions, and that it is a fact to find difficulty in being converged to solutions when Re grows larger if we did not use the Jacobi-red-black or CG-red-black combination as a solution method.

7. Concluding remarks

(1) The degree of the improvement on the start-up times of v2 to v1 is 1.18 to 1.28-fold decrease with MIMD computing in the local memory access, 3.42 to 3.88-fold decrease with the data transfer, and 1.86 to 2.84-fold decrease with MIMD computing in the global memory access. In other words, synchronization overheads have decreased about 1.2 to 1.3-fold, and communication overheads about 5.0 to 5.7-fold.

(2) The parallel compiler can stand improvement on the long-vector length performance with SIMD computing, and more decrease of all sorts of the start-up times in the global memory access.

(3) Saying from the real state of the operating and operational system, more speedup of the operating system by trace scheduling, more improvement on the traffic control function of the operational system, and introduction to the distributed process for the trafic are desired.

References

[1] S. Hatayama, The characteristic parameters of the NWT computer system in the local memory access, *Parallel Computing* (submitted for publication).

[2] S. Hatayama, The characteristic parameters of the NWT computer system in the global memory access, *Parallel Computing* (submitted for publication).

[3] S. Hatayama, On an improvement on the NWT system software, *Parallel Computing* (submitted for publication).

[4] S. Abdallah, Numerical solution for the pressure Poisson equation with Neumann boundary conditions using a non-staggered grid I, *Journal of Computational Physics*, 70, 182-192(1987).

[5] S. Hatayama, Computation of incompressible viscous flow in a lid-driven square cavity and its program performance on the NWT computer system, *Parallel Computing* (submitted for publication).

[6] S. Abdallah, Numerical solution for the incompressible Navier-Stokes equation in primitive variables using a non-staggered grid II, *Journal of Computational Physics*, 70, 193-202(1987).

Table 1 Program performance for a square cavity problem.
(1) $pe = 128$ and v1.

method	grid	r_1(Gflop/s)	r_2(Gflop/s)	r_3(Gflop/s)	r_4(Gflop/s)	d_{ma}
Jacobi	16384^2	103.938	91.202	100.828	133.873	1.328
red-black	16384^2	74.108	59.703	68.854	76.721	1.114
CG	12000^2	100.916	88.901	100.005	105.547	1.055
ADI	12001^2	3.323	3.391	3.114	3.253	1.045

(2) $pe = 128$ and v2

method	grid	r_1(Gflop/s)	r_2(Gflop/s)	r_3(Gflop/s)	r_4(Gflop/s)	d_{ma}	r_5(Gflop/s)
Jacobi	16384^2	107.313	94.166	106.274	142.363	1.340	17.766
red-black	16384^2	73.684	59.933	66.144	79.160	1.162	5.224
CG	12000^2	92.123	82.001	103.110	110.366	1.069	16.591
ADI	12001^2	3.174	3.302	3.672	3.931	1.071	1.680

Table 2 Program performance for a cubical cavity problem.
(1) $pe = 128$ and v1

method	grid	r_1(Gflop/s)	r_2(Gflop/s)	r_3(Gflop/s)	r_4(Gflop/s)	d_{ma}
Jacobi	601*601*601	63.599	—	36.425	100.362	2.820
red-black	602*602*602	19.656	—	26.607	63.936	2.235
CG	551*551*551	49.649	—	41.654	—	—
ADI	561*561*561	36.973	—	39.641	70.509	1.779

(2) $pe = 128$ and v2

method	grid	r_1(Gflop/s)	r_2(Gflop/s)	r_3(Gflop/s)	r_4(Gflop/s)	d_{ma}
Jacobi	601*601*601	63.974	27.076	42.914	100.895	2.534
red-black	602*602*602	25.653	22.555	34.366	63.673	1.659
CG	551*551*551	59.520	32.910	47.668	—	—
ADI	561*561*561	42.649	31.796	40.885	71.200	1.741

Table 3 Program performance for a 3-D narrow cavity problem (pe = 128 and v2).

method	grid	r_1(Gflop/s)	r_2(Gflop/s)	r_3(Gflop/s)
Jacobi	1280*90*1280	92.211	33.718	63.618
red-black	1280*90*1280	34.332	30.942	50.839
CG	1280*70*1280	84.063	46.540	73.466
ADI	1280*70*1280	18.038	21.030	24.075

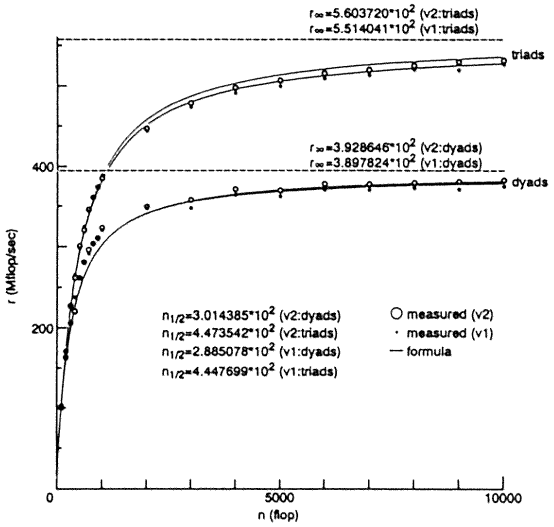


Fig.1 The $(r_\infty, n_{1/2})$ benchmark (actual processing rates with SIMD computing).

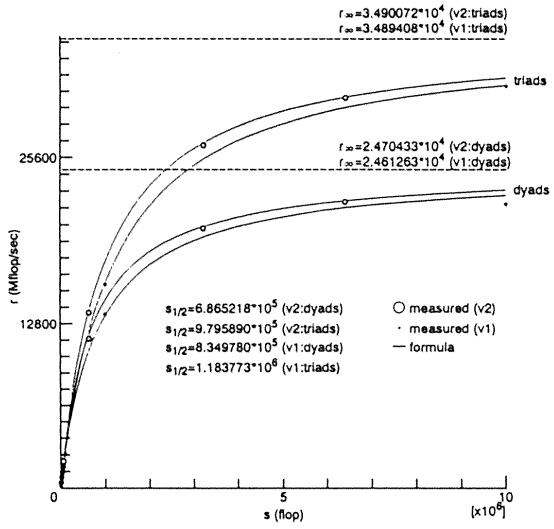


Fig.2 The $(r_\infty, s_{1/2})$ benchmark (actual processing rates with MIMD computing in the local memory access when $pe = 64$).

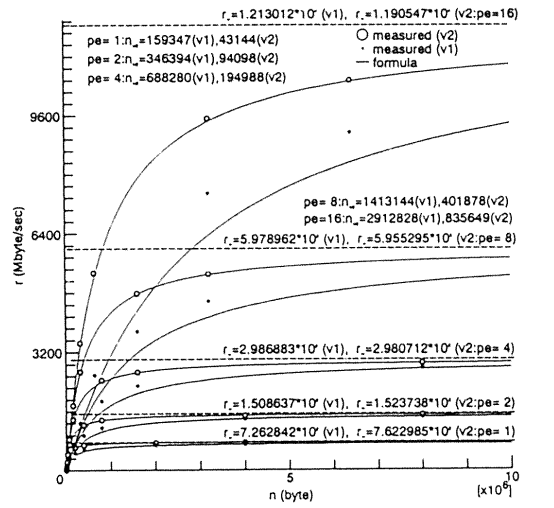


Fig.3 The pingpong benchmark (actual rates with data transfer between the local and global memory spaces when $pe = 1 \sim 16$).

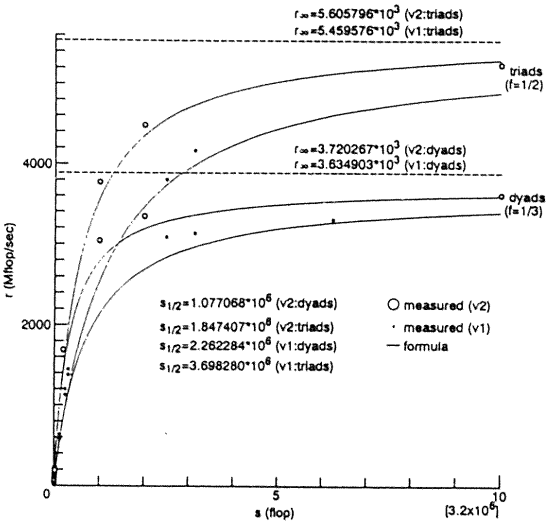
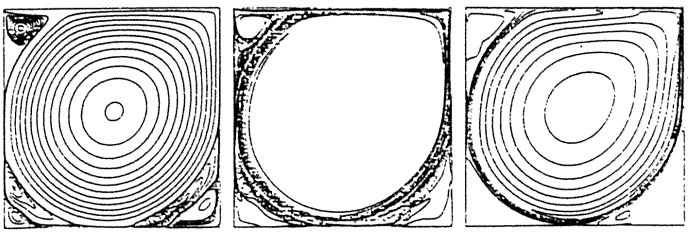
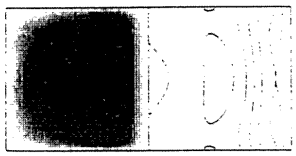


Fig.4 The $(\hat{r}_\infty, \hat{s}_{1/2}, f_{1/2})$ benchmark (actual processing rates with MIMD computing in the globalmemory access when $pe = 64$).

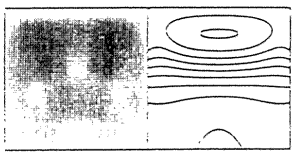


Stream-function Vorticity Pressure

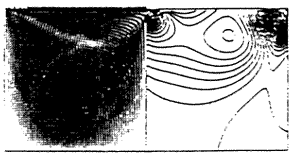
Fig.5 The square cavity flow at $Re = 20,000$.



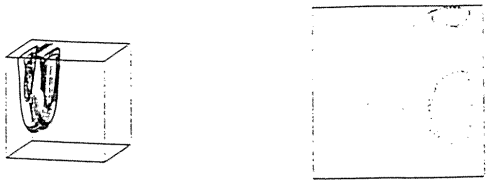
(u,v) and p at xy-cross section (z=.56)



(v,w) and p at yz-cross section (x=.55)



(u,w) and p at xz-cross section (y=.55)



Streamline started at (0.5,0.7,0.97). Locations that flow-particles appear on the right yz-cross section of $x = 0.5$.

Fig.6 The cubical cavity flow at $Re = 100$.