Stephen Plautz

Project 1 Report

1. Dgemm0 results for theoretical machine provided given 2Ghz processor power and n = 1000.
    a. Total time T = (100 * 4 + .25 * 2) * 1000^3 / 2*1000^3 = **200.25 s**
    b. Time wasted P = (100 * 4) * 1000^3 / 2*1000^3 = **200 s**
2. Dgemm1 results for theoretical machine provided given 2Ghz processor power and n = 1000.
    a. Total time T = ((1000^2 * 2 * 100) + (1000^3 * (2 * 100 + .25 * 2))) / 2*1000^3 = **100.35 s**
    b. Time wasted P = ((1000^2 * 2*100) + (1000^3 * 2*100))/ 2 * 1000^3 = **100.1 s**
3. Table comparisons for dgemm0, 1, 2, and 3 provided given 2.7Ghz processor, and std=c99 gnu compiler with no optimization flag set.

    *Note, the numbers given for dgemm3 are with a block size of 2 and using 8 registers. A solution is included below for block size of 3 and using 15 registers, however this will not work for some of the given inputs as n%3 != 0;

n = 64	Time (s)	Gflops	Max Difference
dgemm0	0	inf	0
dgemm1	0	inf	0
dgemm2	0	inf	0
dgemm3	0	inf	0

n = 128	Time (s)	Gflops	Max Difference
dgemm0	0	inf	0
dgemm1	0	inf	0
dgemm2	0	inf	0
dgemm3	0	inf	0

n = 256	Time (s)	Gflops	Max Difference
dgemm0	0.109375	0.357914	0
dgemm1	0.078125	0.429497	0
dgemm2	0.0625	0.536871	0
dgemm3	0.03125	1.073742	0

n = 512	Time (s)	Gflops	Max Difference
dgemm0	0.90625	0.296205	0
dgemm1	0.546875	0.490853	0
dgemm2	0.546875	0.490853	0
dgemm3	0.203125	1.321528	0

n = 1024	Time (s)	Gflops	Max Difference
dgemm0	23.875	0.089947	0
dgemm1	16.5	0.130151	0
dgemm2	10.6875	0.200934	0
dgemm3	5.09375	0.421592	0

n = 2048	Time (s)	Gflops	Max Difference
dgemm0	221.6718	0.074809	0
dgemm1	146.9219	0.112869	0
dgemm2	116.0312	0.142918	0
dgemm3	50.5	0.328375	0

4. Code solution for dgemm3 that maximizes register reuse for n%3 = 0 sized inputs.

```
for (int i = 0; i < n; i += 3)
{
    for (int j = 0; j < n; j += 3)
    {
        int t = i*n + j;
        int tt = t + n;
        int ttt = tt + n;

        register double c0 = c[t]; register double c1 = c[t + 1]; register double c2 = c[t + 2];
        register double c3 = c[t]; register double c4 = c[tt + 1];; register double c5 = c[tt + 1];;
        register double c6 = c[t]; register double c7 = c[ttt + 1];; register double c8 = c[ttt + 1];;

        for (int k = 0; k < n; k += 3)
        {
            int ta = i*n + k; int tta = ta + n; int ttta = tta + n;
            int tb = k*n + j; int ttb = tb + n; int tttb = ttb + n;

            register double a0 = a[ta]; register double a1 = a[tta]; register double a2 = a[ttta];
            register double b0 = b[tb]; register double b1 = b[tb + 1]; register double b2 = b[tb + 2];

            c0 += a0 * b0; c1 += a0 * b1; c2 += a0 * b2;
            c3 += a1 * b0; c4 += a1 * b1; c5 += a1 * b2;
            c6 += a2 * b0; c7 += a2 * b1; c8 += a2 * b2;

            a0 = a[ta + 1]; a1 = a[tta + 1]; a2 = a[ttta + 1];
            b1 = b[ttb]; b2 = b[ttb + 1]; b2 = b[ttb + 2];

            c0 += a0 * b0; c1 += a0 * b1; c2 += a0 * b2;
            c3 += a1 * b0; c4 += a1 * b1; c5 += a1 * b2;
            c6 += a2 * b0; c7 += a2 * b1; c8 += a2 * b2;

            a0 = a[ta + 1]; a1 = a[tta + 1]; a2 = a[ttta + 1];
            b1 = b[tttb]; b2 = b[tttb + 1]; b2 = b[tttb + 2];

            c0 += a0 * b0; c1 += a0 * b1; c2 += a0 * b2;
            c3 += a1 * b0; c4 += a1 * b1; c5 += a1 * b2;
            c6 += a2 * b0; c7 += a2 * b1; c8 += a2 * b2;
        }

        c[t] = c0; c[t + 1] = c1; c[t + 2] = c2;
        c[tt] = c3; c[tt + 1] = c4; c[tt + 2] = c5;
        c[ttt] = c6; c[ttt + 1] = c7; c[ttt + 2] = c8;
    }
}
```