Subject: Re: Optimising A = B+C?
Posted by davis on Sun, 02 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

On Fri, 31 May 1996 02:24:03 GMT, Karl Glazebrook <kgb@aaoepp.aao.gov.au>
wrote:
: Does anyone know how IDL optimises A=B+C where
: A,B and C are arrays?
:
: I did a test a while ago and it was several times faster
: than C code along the lines of:
:
: i=n;
: while (i--)
: *a-- = *b-- + *c--

Although I have no proof, I believe that the above can be coded to run
faster, e.g.,

```
for (i = 0; i < n; i++) a[i] = b[i] + c[i];
```

Also, if IDL arrays are allocated with an even number of elements (with the
last one padded for odd sized arrays), this is faster:

```
for (i = 0; i < n; i++)
  {
    a[i] = b[i] + c[i];
    i++;
    a[i] = b[i] + c[i];
  }
```

:
: (This was on a 2048x2048 array and everything fiited into
: physical memory.)
:

It also depends upon how your two-dimensional array was implemented. One
common C approach is to use, e.g.,

```
double **a;
unsigned int i;

a = malloc (2048 * sizeof (double *));
for (i = 0; i < 2048; i++)
  a[i] = malloc (2048 * sizeof(double));
```

Now consider adding such arrays together. One might code this as

```
  for (i = 0; i < 2048; i++)
   {
     for (j = 0; j < 2048; j++)
  {
    a[i][j] = b[i][j] + c[i][j];
  }
   }
```

I am not sure how many compilers will optimize this. For that reason, I never code loops like the above. Instead, I always do

```
  for (i = 0; i < 2048; i++)
   {
     double *ai, *bi, *ci;
ai = a[i]; bi = b[i]; ci = c[i];
     for (j = 0; j < 2048; j++)
  {
    ai[j] = bi[j] + ci[j];
  }
   }
```

Finally, if you simply create such arrays via:

```
   double a[2048][2048];  /* this is just a 2048*2048 block of doubles */
```

then you might be tempted to perform the addition as

```
  for (i = 0; i < 2048; i++)
   {
     for (j = 0; j < 2048; j++)
  {
    a[i][j] = b[i][j] + c[i][j];
  }
   }
```

The problem with this is that it is not very friendly to your CPU cache because the loop over j does not deal with neighboring values in the array. As a result, it is best to calculate the sum for these types of arrays as:

```
  double *aa, *bb, *cc;
  unsigned int i;

  aa = (double *) a;
  bb = (double *) b;
  cc = (double *) c;

  for (i = 0; i < 2048*2048; i++)
   {
```

```
    aa[i] = bb[i] + cc[i];
  }
```

--

John E. Davis Center for Space Research/AXAF Science Center
617-258-8119 MIT 37-662c, Cambridge, MA 02139
http://space.mit.edu/~davis
