

Fortress istream Benchmark Implementation and Results 2008-02-28

Implementation Details

1. Polymorphic version, works for three types (ZZ32—32bit int, ZZ64—64bit int and RR64—double). Currently RR32 (float) not available as a 'Number', is supported according to the user manual.
2. Instrumented with timers nanoTime()
3. LSHIFT (<<) not available for RR64
4. Performance poor compared to gcc -O3 istream.c
5. checkSTREAMresults do not match (using sequential loops).

Results

```
$fortress istream.fss
Parsing files: istream.fss
Parsing istream.fss: 392 milliseconds
The data type is 32-bit int
This system uses 4 bytes per Element being tested
Array size = 200000, Offset = 0
Total memory required = 2.288818359375 MB.
Function  Rate (MB/s) Avg time
Copy  0.1690630723359879  9.463923599
Scale  0.16171250999495498  9.894101576
Add  0.20419489079040956  11.753477233
Triad  0.1984296419517062  12.094967145
Sscale  0.15844711352225047  10.098006612
*****

The data type is 64-bit int
This system uses 8 bytes per Element being tested
Array size = 200000, Offset = 0
Total memory required = 4.57763671875 MB.
Function  Rate (MB/s) Avg time
Copy  0.33725174241204936  9.488460985
Scale  0.3227982297160011  9.91331335
Add  0.41265021868714274  11.632127605
Triad  0.3889396119171864  12.341247466
Sscale  0.32702139624585047  9.785292451
```

```

*****
The data type is double
This system uses 8 bytes per Element being tested
Array size = 200000, Offset = 0
Total memory required = 4.57763671875 MB.
Function   Rate (MB/s) Avg time
Copy  0.34459258219792543  9.286328741
Scale  0.3221221964027066  9.934118281
Add   0.4035468109557772  11.894530869
Triad  0.39303445280566834  12.21266982
*****
Program execution: 247268 milliseconds

```

```

$ gcc -O3 -o istream istream.c -DIBY8
$ ./istream
-----
VSTREAM version $Revision: 1.0 $
-----
The data type is long long
This system uses 8 bytes per Element being tested.
-----
Array size = 2000000, Offset = 0
Total memory required = 45.8 MB.
Each test is run 10 times, but only
the *best* time for each is used.
-----
Printing one line per active thread...
-----
Your clock granularity/precision appears to be 2 microseconds.
Each test below will take on the order of 8949 microseconds.
(= 4474 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-----
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-----
Function   Rate (MB/s) Avg time   Min time   Max time
Copy:      2730.8889   0.0117   0.0117   0.0117
Scale:     2767.4329   0.0116   0.0116   0.0116
Add:       2917.9459   0.0165   0.0164   0.0165
Triad:     2934.8755   0.0164   0.0164   0.0164
SScale    2763.8427   0.0116   0.0116   0.0116
-----

```

Productivity Issues

1. Usually very little or no response from developers.
2. Low productivity attributed to incomplete implementation of language constructs and mismatch between language spec document and current release.
3. Not always clear what works and does not work and why. Useless error messages, most cases java trace dump.

To do

1. Experiment with more language constructs e.g. traits
2. Checkstreamresults
3. UTS implementation