HPC Challenge Awards: Class 2 Specification

Jack Dongarra Piotr Luszczek

June 2005

Contents

1 General Guidelines

1. The use of high level languages is encouraged.

2. Calls to tuned library routines could be used in the submission but explicit and "elegant" coding of all aspects of the benchmark is preferred.

3. The entire benchmark could be expressed by using a few built-in operators of an hypothetical programming language. However, such submissions are strongly discouraged as they only show operator overloading and funciton call syntax and say nothing about the language. In particular, how it deals with issues critical to HPC like expressing parallelism and hiding latency.

2 HPL

HPL (High Performance Linpack) is an implementation of the Linpack TPP (Toward Peak Performance) variant of the original Linpack benchmark which measures the floating point rate of execution for solving a linear system of equations.

2.1 Description

HPL solves a linear system of equations of order n:

$$Ax = b; \quad A \in \mathbf{R}^{n \times n}; \; x, b \in \mathbf{R}^n \tag{1}$$

by first computing LU factorization with row partial pivoting of the n by $n+1$ coefficient matrix:

$$P[A,b] = [[L,U],y]. \tag{2}$$

Since the row pivoting (represented by the permutation matrix P) and the lower triangular factor L are applied to b as the factorization progresses, the solution x is obtained in one step by solving the upper triangular system:

$$Ux = y. \tag{3}$$

The lower triangular matrix L is left unpivoted and the array of pivots is not returned.

2.2 Data Size

A is n by n double precision (in IEEE 754 sense) matrix, b is n-element vector. The size of the A matrix ($8n^2$ bytes) should be at least half of the system memory.

2.3 Initialization

Both A and b should contain values produced by a reasonable pseudo-random generator with an expected mean of zero. "Reasonable" in this context means compact, fast, and producing independent and identically distributed elements.

2.4 Timed Region

The timed portion of the code performs steps given by equations (2) and (3) and does not include time to generate A and b.

2.5 Duration

Until solution to (1) is obtained.

2.6 Verification

Correctness of the solution is ascertained by calculating the following scaled residual:

$$r = \frac{\|Ax-b\|_\infty}{\varepsilon(\|A\|_\infty\|x\|_\infty + \|b\|_\infty)n} \tag{4}$$

where ε is machine precision for 64-bit floating-point values and n is the size of the problem. The solution is valid if the following holds:

$$r < 16 \tag{5}$$

2.7 Performance

The operation count for the factorization phase is $\frac{2}{3}n^3 - \frac{1}{2}n^2$ and $2n^2$ for the solve phase thus if the time to solution is t_S the formula for performance (in Gflop/s) is:

$$p_{\mathsf{HPL}} = \frac{\frac{2}{3}n^3 + \frac{3}{2}n^2}{t_S} 10^{-9}. \tag{6}$$

2.8 Alternative Implementations

If an alternative algorithm is chosen it should be able to deal with zeros on the diagonal (some sort of pivoting needs to be used) and the precision of the calculations needs to be preserved.

3 RandomAccess

3.1 Description

Let $T[\cdot]$ be a table of size 2^n.
Let $\{a_i\}$ be a stream of 64-bit integers of length $N_U = 2^{n+2}$ generated by the primitive polynomial over GF(2)[1]:
$x^{63} + x^2 + x + 1$.
For each a_i, set

$$T[a_i\langle 63, 64-n\rangle] \leftarrow T[a_i\langle 63, 64-n\rangle] \oplus a_i \tag{7}$$

where:

- $\oplus$ denotes addition in GF(2) i.e. "exclusive or" (XOR)
- $a_i\langle l,k\rangle$ denotes the sequence of bits within a_i, e.g. $\langle 63, 64-n\rangle$ are the highest n bits.

3.2 Data Size

The parameter $m(=2^n)$ is defined such that:
m is the largest power of 2 that is less than or equal to half of the system memory. Since the elements of the main table are 64-bit quantities, the table occupies $8m$ bytes of memory.

3.3 Initialization

Table elements are set such that:

$$\forall_{0\le i<2^n} T[i] \equiv i \tag{8}$$

3.4 Timed Region

The timed region consists of computation (7). The initialization (8) is not timed.

3.5 Duration

Ideally, 2^{n+2} updates should be performed to the main table ($N_U = 2^{n+2}$). However, the computation can be prematurely stopped after 25% of the time of the HPL run (but not shorter than 1 minute). Thus:

$$N_U \le 2^{n+2} \tag{9}$$

3.6 Verification

The update defined by (7) should be repeated by an alternative method that is safe (does not generate errors resulting from, for example, race conditions in memory updates). If the benchmarked update was correct, the table should return

[1] Galois Field of order 2 – The elements of GF(2) can be represented using the integers 0 and 1, i.e., binary operands.

to its initial state defined by (8). However, 1% of entries may have incorrect values, i.e. given a function:

$$f(i) = \begin{cases} 0 & \text{if } T[i] = i \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

the following should hold:

$$\sum_{i=0}^{N_U} f(i) \leq 10^{-2} N_U \tag{11}$$

3.7 Performance

Let $t_{\text{RandomAcccess}}$ be the time it took to finish the timed portion of the test (including N_U updates) then peroformance (in GUPS: Giga Updates Per Second) is defined as:

$$p_{\text{RandomAcccess}} = \frac{N_U}{t_{\text{RandomAcccess}}} 10^{-9}. \tag{12}$$

3.8 Alternative Implementations

Constraints on the look-ahead and storage before processing on distributed memory multi-processor systems is limited to 1024 per process (or processing element). The pseudo-random number generator that generates sequence $\{a_i\}$ has to be used.

4 Global EP-STREAM-Triad

4.1 Description

EP-STREAM-Triad is a simple benchmark program that measures sustainable memory bandwidth (in Gbyte/s) and the corresponding computation rate for a simple vector kernel operation that scales and adds two vectors:

$$a \leftarrow b + \alpha\, c \tag{13}$$

where:

$$a, b, c \in \mathbf{R}^m; \quad \alpha \in \mathbf{R}.$$

The computation is peformed simultaneously on each computing element on its local data set.

4.2 Data Size

a, b, and c are m-element double precision vectors. The combined size of the vectors ($24m$ bytes) should be at least quarter of the system memory.

4.3 Initialization

Vectors b and c should contain values produced by a reasonable pseudo-random number generator.

4.4 Timed Region

The timed portion of the code should perform operation given by (13) at least 10 times.

4.5 Duration

The kernel operation should be repeated at least 10 times.

4.6 Verification

The norm of the difference between reference and computed vectors is used to verify the result: $\|a - \hat{a}\|$. The reference vector $\hat{a}$ is obtained by an alternative implementation.

4.7 Performance

The benchmark measures Gbyte/s and the number of items transferred is $3m$. The minimum time $t_{\min}$ is taken of all the repetitions of the kernel operation. Performance is thus defined as:

$$p_{\text{EP-STREAM-Triad}} = 24 \frac{m}{t_{\min}} 10^{-9} \tag{14}$$

4.8 Alternative Implementations

5 Global FFT

5.1 Description

FFT measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DFT) of size m:

$$Z_k \leftarrow \sum_j^m z_j e^{-2\pi i \frac{jk}{m}}; \quad 1 \leq k \leq m \tag{15}$$

where:

$$z, Z \in \mathbf{C}^m.$$

5.2 Data Size

Z and z are m-element double precision complex vectors. The combined size of the vectors ($32m$ bytes) should be at least quarter of the system memory. The size m of the vectors can be implementation-specific, e.g. be an integral power of 2.

5.3 Initialization

Vector z should contain values produced by a reasonable pseudo-random number generator. The real and imaginary parts of z should be generated independently. The layout of vectors z and Z should not be scrambled either before or after the computation.

5.4 Timed Region

The computation implied by (15) is timed together with the portion of code that unscrambles (if necessary) the resulting vector data. Timing for computation and unscrambling can be given separately for informational purposes but the combined time is used for calculating performance.

5.5 Duration

Until the transform defined by (15) is obtained.

5.6 Verification

Verification is done by acertainig the following bound on the residual:

$$\frac{\|z-\hat{z}\|_\infty}{\varepsilon \ln m} < 16 \tag{16}$$

where $\hat{z}$ is the result of applying a reference implementation of the inverse transform to the outcome of the benchmarked code (in infinite-precision arithmetic the residual should be zero):

$$\hat{z}_k \leftarrow \sum_j^m Z_j e^{2\pi i \frac{jk}{m}}; \quad 1 \le k \le m \tag{17}$$

5.7 Performance

The operation count is taken to be $5m\log_2 m$ for the calculation of the computational rate (in Gflop/s) in time t:

$$p_{\mathsf{FFT}} = 5\frac{m\log_2 m}{t}10^{-9} \tag{18}$$

5.8 Alternative Implementations

The reference implementation splits the algorithm into computational and communication portions which do not overlap. Valid submsissions may choose other methods that take advantage of language and architectural features.

The number of processors may be implementation-specific, e.g. be an integral power of 2.