**PROJECT GOALS**

- Provide performance bounds in locality space using real-world computational kernels
- Allow scaling of input data size and time to run according to the system capability
- Verify the results using standard error analysis
- Allow vendors and users to provide optimized code for superior performance
- Make the benchmark information continuously available to the public in order to disseminate performance tuning knowledge and record technological progress over time
- Ensure reproducibility of the results by detailed reporting of all aspects of benchmark runs

**FEATURE HIGHLIGHTS OF HPCC 1.4.1**

- New variants of RandomAccess that use Linear Congruential Random Number Generator
- New order of tests makes HPL run last so users may abort execution early if necessary
- Initialization of the main array of RandomAccess is no longer timed
- Global reduction is used for error calculation in MPI FFT to achieve more accurate error estimate
- Updated documentation
- New initialization and finalization routines allow proper setup of external software/hardware components without changing the rest of the HPCC testing harness.
- Fixed memory leaks in G-RandomAccess and FFT driver routines.
- Better interface to 64-bit versions of FFTW such as Intel's MKL.

**LOCALITY SPACE OF MEMORY ACCESS IN APPLICATIONS**

- **Spatial Locality**
- **Temporal Locality**

**SUMMARY OF HPCC AWARDS**

**CLASS 1: Best Performance**

- Best in G-HPL, EP-STREAM-Triad per system, G-RandomAccess, G-FFT
- There will be 4 winners (one in each category)

**CLASS 2: Most Productivity**

- One or more winners
- Judged by a panel at SC11 BOF
- Stresses elegance and performance
- Implementations in various (existing and new) languages are encouraged
- Submissions may include up to two kernels not present in HPCC
- Submission consists of: code, its description, performance numbers, and a presentation at the BOF
**HPCC BENCHMARKS**

**HPL**
This is the widely used implementation of the Linpack TPP benchmark. It measures the sustained floating point rate of execution for solving a linear system of equations.

**STREAM**
A simple benchmark test that measures sustainable memory bandwidth (in GB/s) and the corresponding computation rate for four vector kernel codes.

**RandomAccess**
Measures the rate of integer updates to random locations in large global memory array.

**PTRANS**
Implements parallel matrix transpose that exercises a large volume communication pattern whereby pairs of processes communicate with each other simultaneously.

**FFT**
Calculates a Discrete Fourier Transform (DFT) of very large one-dimensional complex data vector.

**b_eff**
Effective bandwidth benchmark is a set of MPI tests that measure the latency and bandwidth of a number of simultaneous communication patterns.

**DGEMM**
Measures the floating point rate of execution of double precision real matrix-matrix multiplication.

---

**HPCC AWARDS CLASS 1: PERFORMANCE**

**G-FFT**
![Graph showing performance of G-FFT benchmark over years 2005 to 2010.](image)

**G-RandomAccess**
![Graph showing performance of G-RandomAccess benchmark over years 2005 to 2010.](image)

**G-HPL**
![Graph showing performance of G-HPL benchmark over years 2005 to 2010.](image)

**EP-STREAM-Triad**
![Graph showing performance of EP-STREAM-Triad benchmark over years 2005 to 2010.](image)

---

**FIND OUT MORE AT** [http://icl.eecs.utk.edu/hpcc/](http://icl.eecs.utk.edu/hpcc/)