HPC Challenge Benchmarks Panel: Base versus Optimized

Jeffrey Vetter

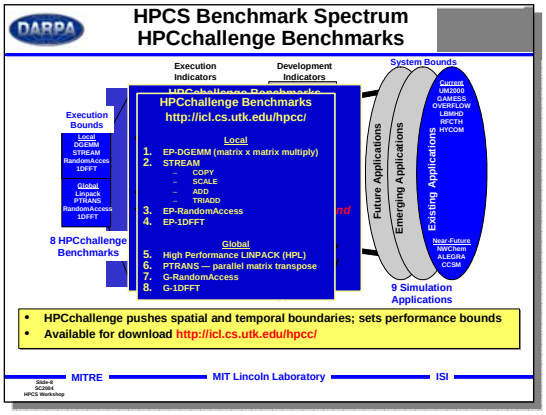
Oak Ridge National Laboratory
vetterjs@ornl.gov

Joint work with Cray and Nathan Wichmann.

ornl

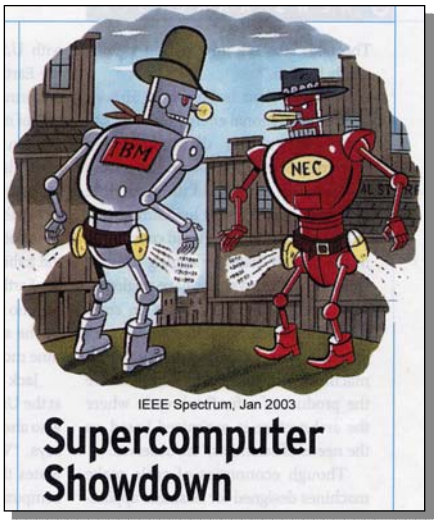
HPC Challenge Benchmarks

- **Don't lose sight of the entire spectrum of benchmarks, kernels, and applications**
- **Offers two versions**
  - Base
  - Optimized



Why two versions?

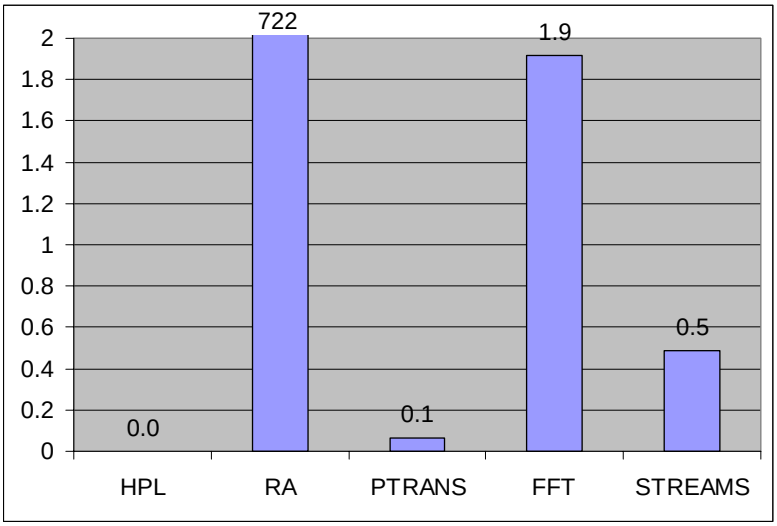
- **TOP500 LINPACK**
  - The Wild, Wild West! ☺
- **Base**
  - Portable: Functional across 'most' platforms
  - Represents legacy codes (or otherwise immutable)
- **Optimized**
  - Match application characteristics to architectural strengths
- **Both versions are valid**
- **Energy required to move from *Base* to *Optimized* proportional to system's productivity?**



Future Architectural Diversity

- **Have you been to an architecture conference recently?**
  - Multicore processors (w/ variable clock speed)
  - Multithreading
  - Co-processors
  - Various memory models
  - SIMD
  - Power, power, power
- **Systems are becoming much more complex!**
- **Cray X1**
  - Vector processors
  - Globally addressable shared memory
  - Programming models
    - Streams
    - Vectors
    - MPI
    - OpenMP
    - Co-array Fortran
    - UPC
    - SHMEM

Speedup on X1 (64 MSPs)



One example Random Ring – Not only improved performance, but improved productivity?

MPI:

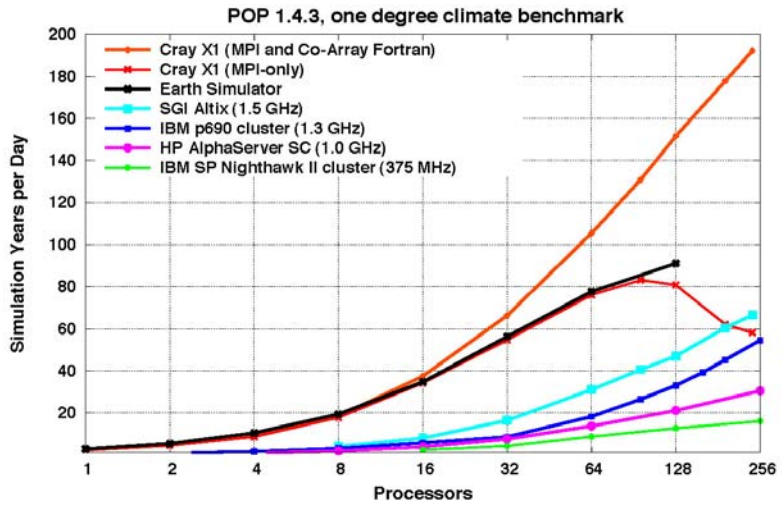
```
MPI_Sendrecv( sndbuf_right, msglenw, MPI_LONG, right_rank,
  TO_RIGHT, rcvbuf_left, msglenw, MPI_LONG, left_rank, TO_RIGHT,
  MPI_COMM_WORLD, &(statuses[0]) );
MPI_Sendrecv( sndbuf_left, msglenw, MPI_LONG, left_rank, TO_LEFT,
  rcvbuf_right, msglenw, MPI_LONG, right_rank, TO_LEFT,
  MPI_COMM_WORLD, &(statuses[1]) );
```

UPC:

```
upc_barrier;
for(i = 0; i < msglenw; i++ ){
  upc_recvbuf_left[i][right_rank] = sndbuf_right[i];
  upc_recvbuf_right[i][left_rank] = sndbuf_left[i]; }
upc_barrier;
```

Impact on Applications

- Augmented POP w/ CAF to improve performance and scaling



Summary

- Two versions of HPCC benchmarks represent two different uses of target systems
  - Base
  - Optimized
- Our measurements on the X1 reveal differences
  - UPC/MPI difference on RandomAccess is 722x!
- Systems are growing more complex
  - To exploit this complexity, we will need to use the proper languages, runtime systems, optimizations, etc.