The STREAM Benchmark

John D. McCalpin, Ph.D.
IBM eServer Performance
2005-01-27

History

- Scientific computing was largely based on the vector paradigm from the late 1970's through the 1980's
  - E.g., the classic DAXPY loop:

```
REAL*8 X(N),Y(N)
DO I=1,N
  X(I) = X(I) + Q*Y(I)
END DO
```

- If all data comes from memory, this requires 12 Bytes per FLOP (=3*8Bytes/2FLOPS)

History (cont'd)

- Vector machines like the Cray X/MP, Cray Y/MP, Cray C90, CDC Cyber 205, ETA-10, and various Japanese vector systems were built to this target of 12 Bytes/second of peak memory bandwidth per peak FLOP
- Easy to sustain 30%-40% of peak performance for vectorizable code.
  - >90% on LINPACK

History (cont'd)

- 1990: The Attack of the Killer Micros
  - HP PA-RISC (aka "snakes")
  - IBM RS/6000 (aka "RIOS", aka "POWER", aka "POWER1"
- These systems provided performance roughly matching Cray vector machines on many applications, but fell very short on others
- Users were confused….

The Genesis of STREAM

- I was an assistant professor at the University of Delaware in 1990
- My research was on large-scale ocean circulation modelling
  - Vectorizable codes that ran well on vector machines
  - RS/6000-320 gave about 4% of Cray Y/MP performance
- The STREAM benchmark was developed as a proxy for the computational kernels in my ocean models

STREAM Design

- STREAM is designed to measure sustainable memory bandwidth for contiguous, long-vector memory accesses
- Four kernels are tested:
  - COPY: a(i) = b(i)
  - SCALE: a(i) = q*b(i)
  - ADD: a(i) = b(i) + c(i)
  - TRIAD: a(i) = b(i) + q*c(i)
- Arrays should all be much larger than caches

STREAM Design (cont'd)

- The code is set up with dependencies between the kernels to try to prevent excessive optimization
- Recent versions (in FORTRAN and C) have self-contained validation code
- Recent versions (in FORTRAN and C) have OpenMP directives for parallelization
  - Code is NUMA-friendly with "first touch" memory placement
- An MPI version is also available (FORTRAN)

STREAM History

- STREAM was maintained at my FTP site at the University of Delaware from 1991-1996
- STREAM web/ftp sites moved to the University of Virginia in 1996 when I moved to Industry (SGI)
- Current run rules have several publication classes
  - "standard" – no source code changes to kernels
  - "tuned" – modified source code
  - "MPI" – uses MPI source code
  - "32-bit" – uses 32-bit data items and arithmetic

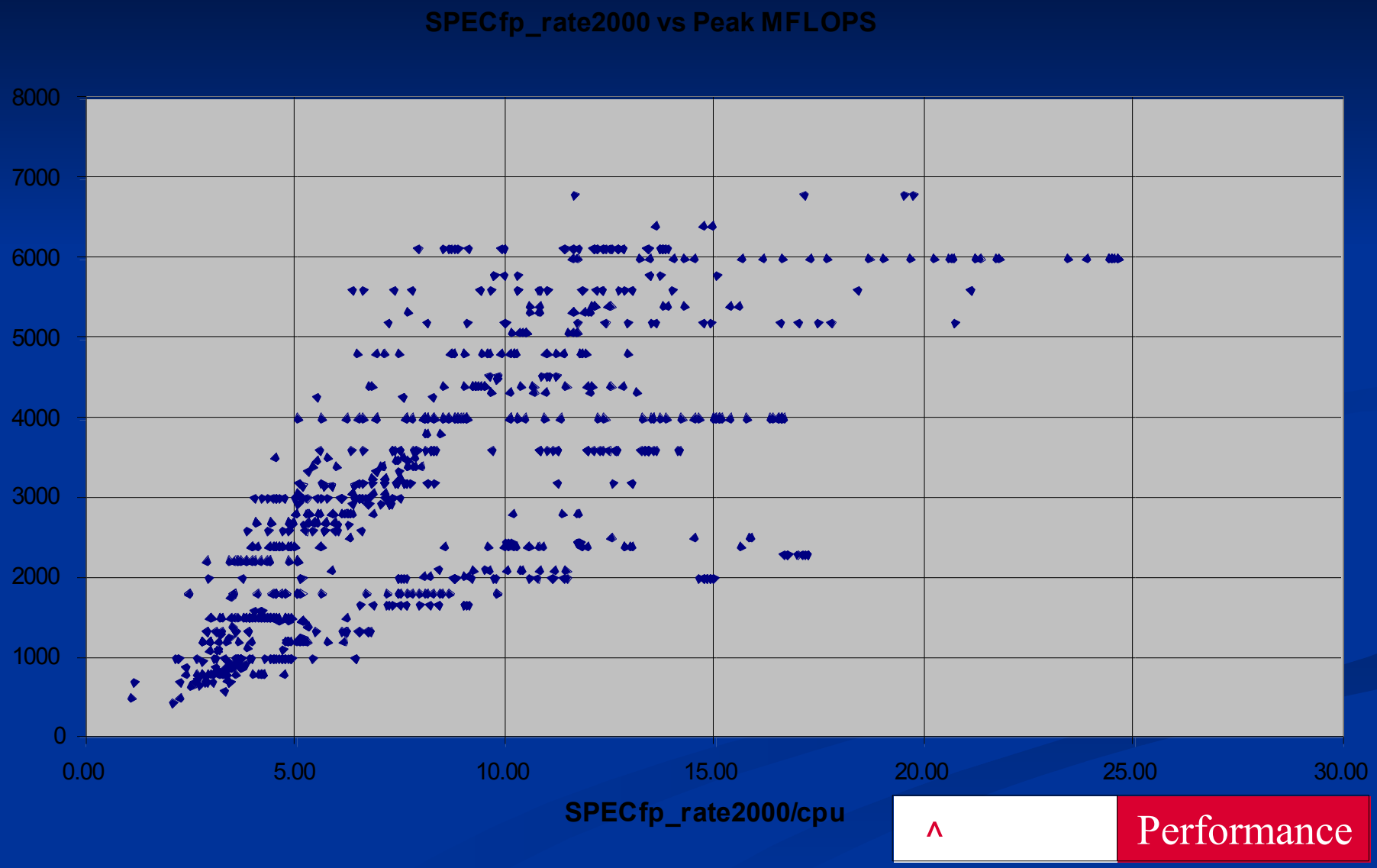
Is STREAM Useful?

- Yes – if its limitations are properly understood
- STREAM is intended as one component of a performance model that includes both processor and memory timings
  - E.g., $T_{total} = T_{cpu} + T_{memory}$
- Such models are intrinsically application dependent
  - Often strongly dependent on cache size

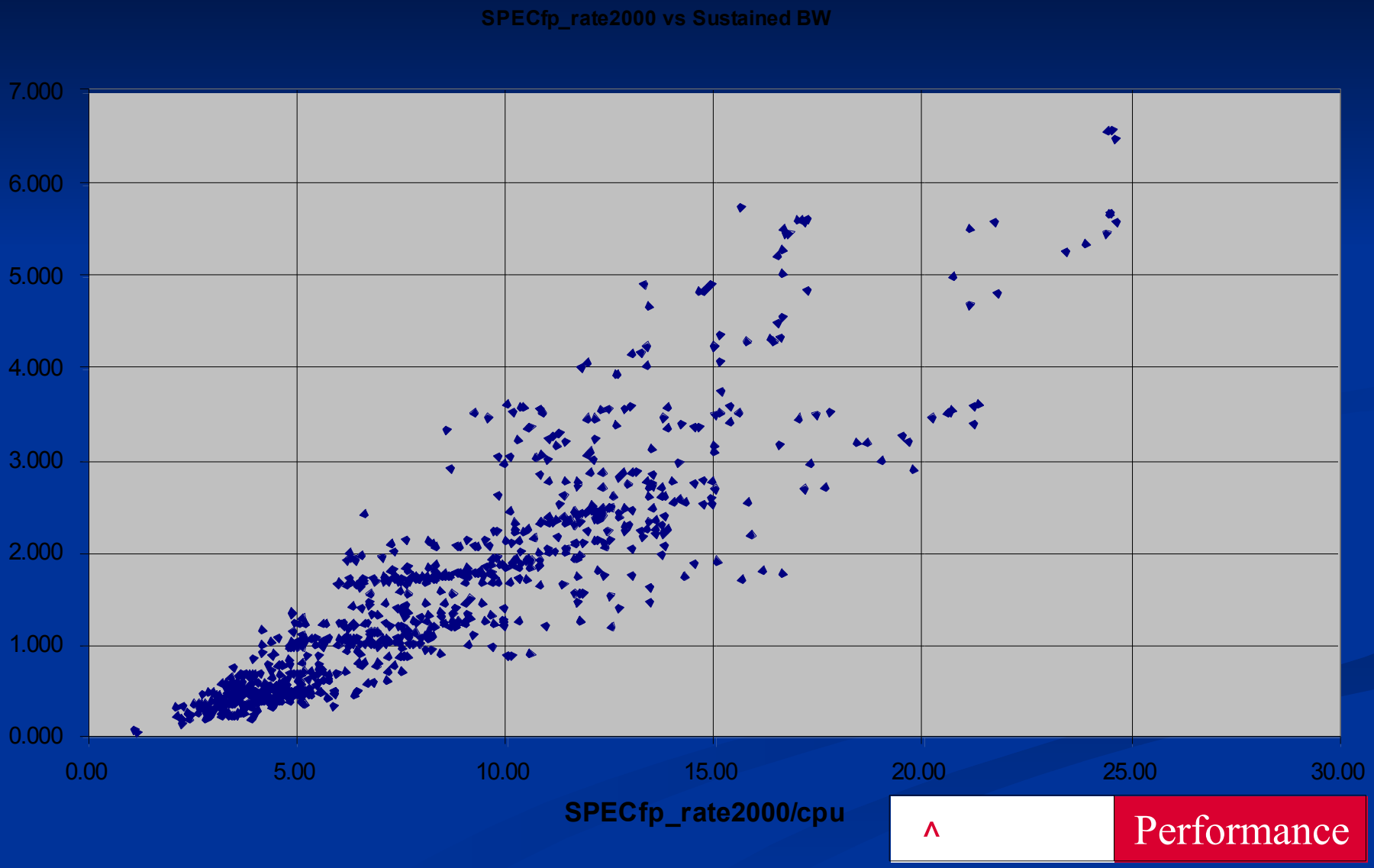
The Big Question

- Q: How should one think about composite figures of merit based on such a collection of low-level measurements?
- A: Composite Figures of Merit must be based on "time" rather than "rate"
  - i.e., weighted harmonic means of rates
- Why?
  - Combining "rates" in any other way fails to have a "Law of Diminishing Returns"

Does Peak GFLOPS predict SPECfp_rate2000?



Does Sustained Memory Bandwidth predict SPECfp_rate2000?

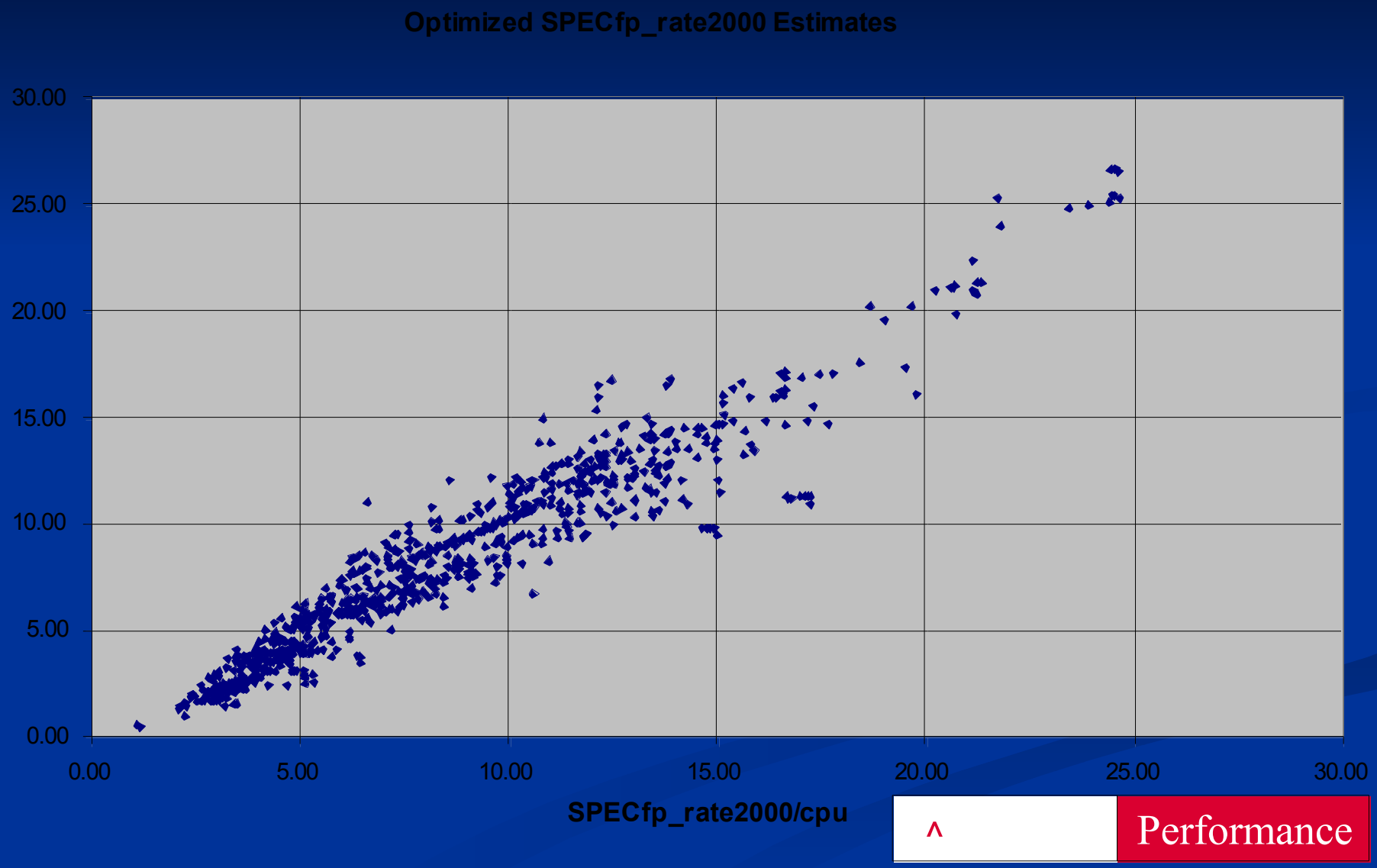


A Simple Composite Model

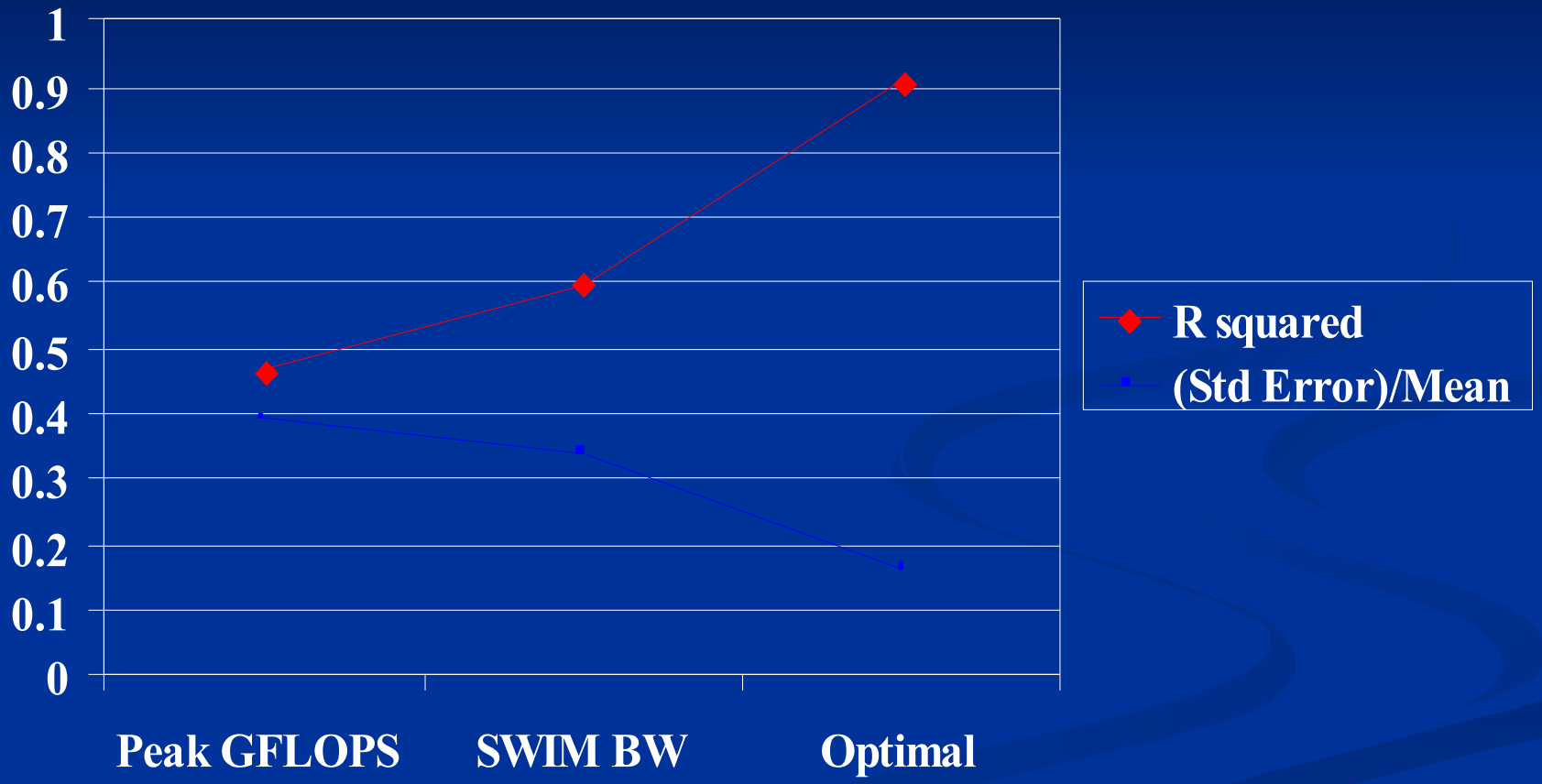
- Assume the time to solution is composed of a compute time proportional to peak GFLOPS plus a memory transfer time proportional to sustained memory bandwidth
- Assume “x Bytes/FLOP” to get:

$$\text{"Balanced GFLOPS"} \equiv \frac{1\ \text{"Effective FP op"}}{\left(\frac{1\ \text{FP op}}{\text{Peak GFLOPS}}\right)+\left(\frac{x\ \text{Bytes}}{\text{Sustained GB/s}}\right)}$$

Does this Revised Metric predict SPECfp_rate2000?



Statistical Metrics

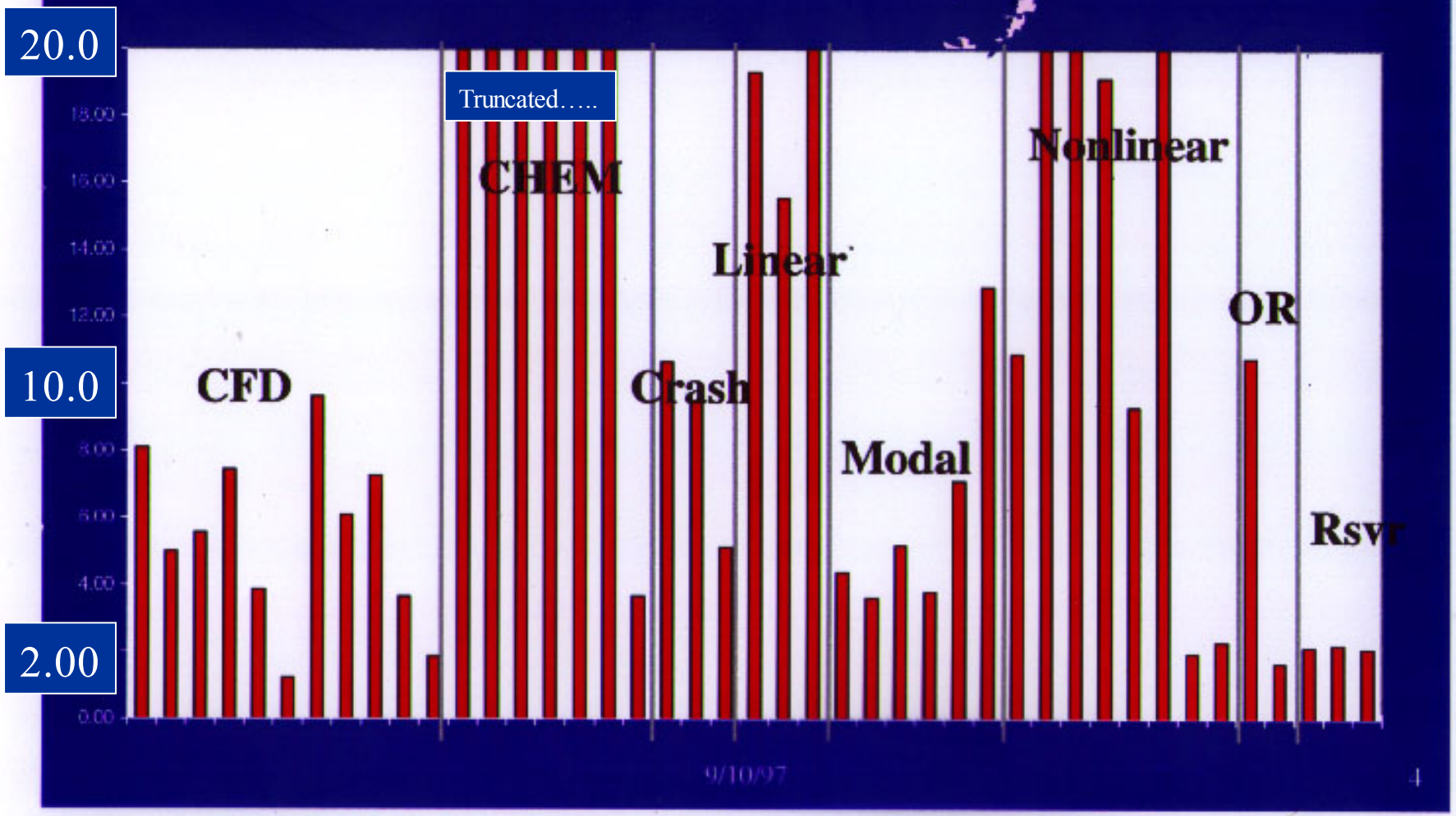


^ Performance

What about other applications?

- Effectiveness of caches varies by application area
- Requirements for interconnect performance vary by application area
  - Some apps are short-message dominated
  - Some apps are long-message dominated
- Composite models can be tuned to specific application areas – if app properties known

BW Reduction due to 4 MB Cache



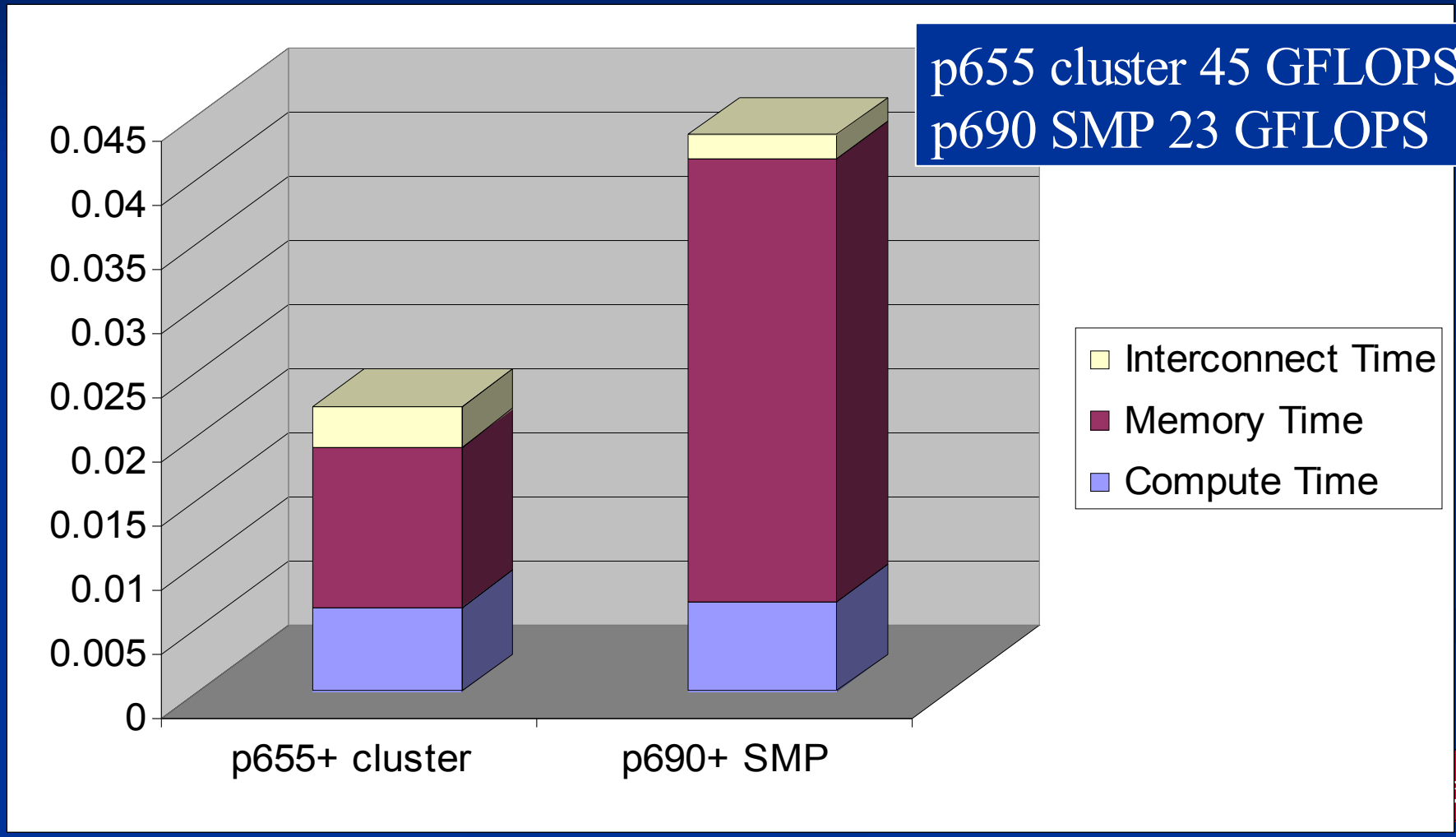
An Example Model tuned for CFD

- Analyze applications and pick reasonable values:

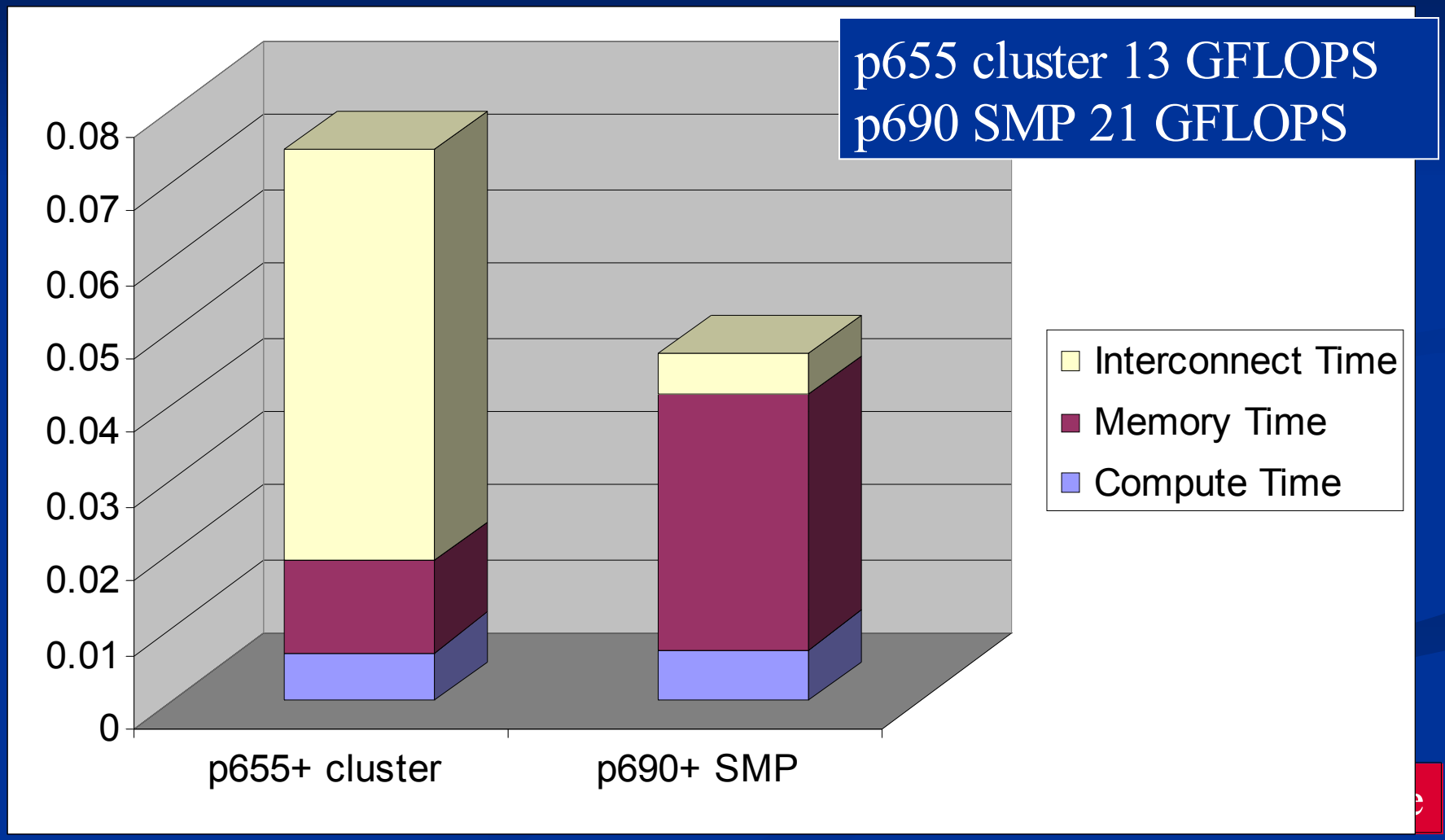
$$\text{"Balanced GFLOPS"} \equiv \frac{1\ \text{"Effective FP op"}}{\left(\frac{1\ \text{FP op}}{\text{LINPACK GFLOPS}}\right)+\left(\frac{2\ \text{Bytes}}{\text{STREAM GB/s}}\right)+\left(\frac{0.1\ \text{Bytes}}{\text{Network GB/s}}\right)}$$

- Two cases tested:
  - Assume long messages (network BW tracks PTRANS)
  - Assume short messages (network BW tracks GUPS)
- The relative time contributions will quickly identify applications that are poorly balanced for the target workload

Comparing p655 cluster vs p690 SMP Assumes long messages



Comparing p655 cluster vs p690 SMP
Assumes short messages



Summary

- The composite methodology is
  - Simple to understand
  - Based on the components of the HPC Challenge Benchmark
  - Based on a mathematically correct model of performance
- Much work remains on documenting the work requirements of various application areas in relation to the component microbenchmarks