Composite Metrics for System Throughput in HPC

John D. McCalpin, Ph.D.
IBM Corporation
Austin, TX

SuperComputing 2003
Phoenix, AZ
November 18, 2003

Overview

- The HPC Challenge Benchmark was announced last night at the TOP500 BOF

Overview

- The HPC Challenge Benchmark was announced last night at the TOP500 BOF
- The HPC Challenge Benchmark consists of
  - LINPACK (HPL)
  - STREAM
  - PTRANS (transposing the array used by HPL)
  - GUPS
  - and some low-level MPI latency & BW measurements

Overview

- The HPC Challenge Benchmark was announced last night at the TOP500 BOF
- The HPC Challenge Benchmark consists of
  - LINPACK (HPL)
  - STREAM
  - PTRANS (transposing the array used by HPL)
  - GUPS
  - and some low-level MPI latency & BW measurements
- No single figure of merit is defined

The Big Question

- How should one think about composite figures of merit based on such a collection of low-level measurements?

The Big Answer

- How should one think about composite figures of merit based on such a collection of low-level measurements?
- 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"

Performance $\propto$ 1/Time

- Time = Work/Rate
- Repeat for each component: $T_i = W_i/R_i$

Performance ∝ 1/Time

- Time = Work/Rate
- Repeat for each component: $T_i = W_i/R_i$

- Big Issues:
  - Where do we get the W_i's?
  - Can we understand the R_i's well enough to be useful?
  - How do we combine the T_i's?

Performance ∝ 1/Time

- Time = Work/Rate
- Repeat for each component: $T_i = W_i/R_i$

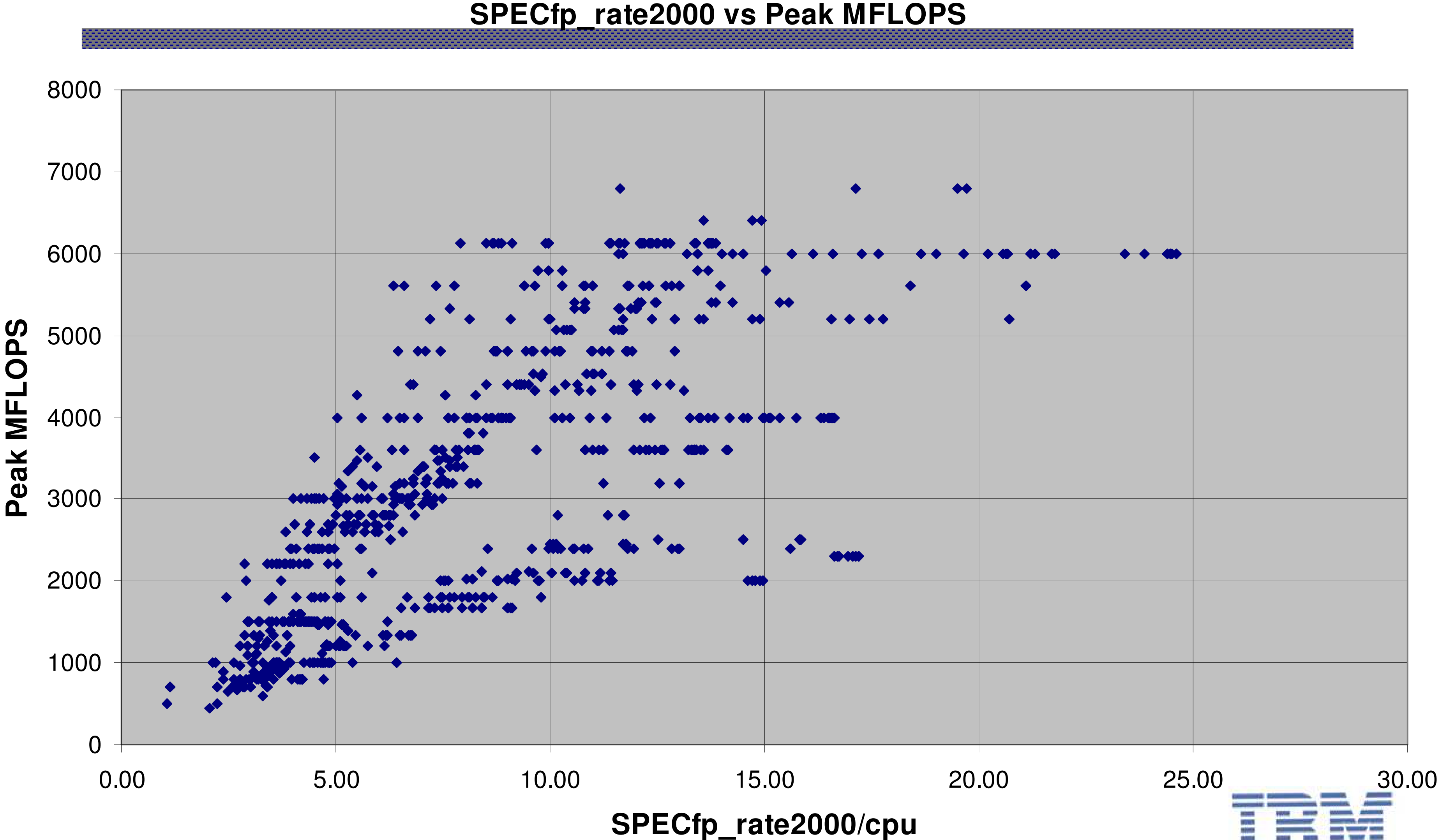
- Big Issues:
  - Where do we get the W_i's?
  - Can we understand the R_i's well enough to be useful?
  - How do we combine the T_i's?

- This talk will mostly address the first issue

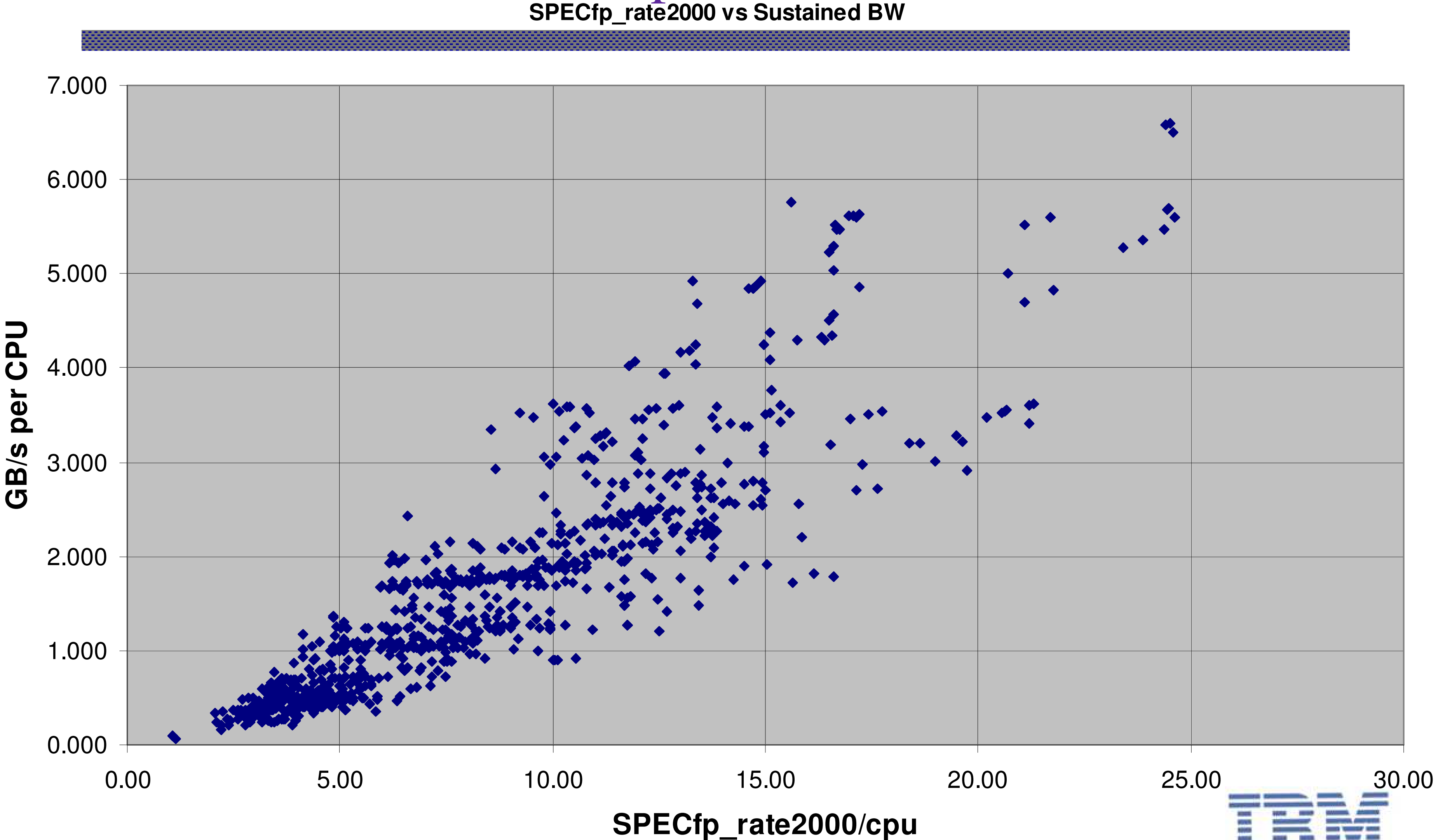
An Example of a Composite Figure of Merit for a Particular Workload

- The target workload is SPECfp_rate2000
  - All 939 published values as of September 14, 2003
  - Duplicates not removed (I am lazy)

Does Peak GFLOPS predict SPECfp_rate2000?



Does Sustained Memory Bandwidth predict SPECfp_rate2000?



An Example of a Composite Figure of Merit for a Particular Workload

- The target workload is SPECfp_rate2000
  - All 939 published values as of September 14, 2003
  - Duplicates not removed (I am lazy)
- Assume that FP arithmetic is the primary bottleneck

An Example of a Composite Figure of Merit for a Particular Workload

- The target workload is SPECfp_rate2000
  - All 939 published values as of September 14, 2003
  - Duplicates not removed (I am lazy)
- Assume that FP arithmetic is the primary bottleneck
- Add memory bandwidth as the secondary bottleneck

An Example of a Composite Figure of Merit for a Particular Workload

- The target workload is SPECfp_rate2000
  - All 939 published values as of September 14, 2003
  - Duplicates not removed (I am lazy)
- Assume that FP arithmetic is the primary bottleneck
- Add memory bandwidth as the secondary bottleneck
- No W_i's were measured
  - model values were obtained *a posteriori* by modifying the parameters of a simple analytic model to minimize the RMS error of the projections

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

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:

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

A Simple Composite Model

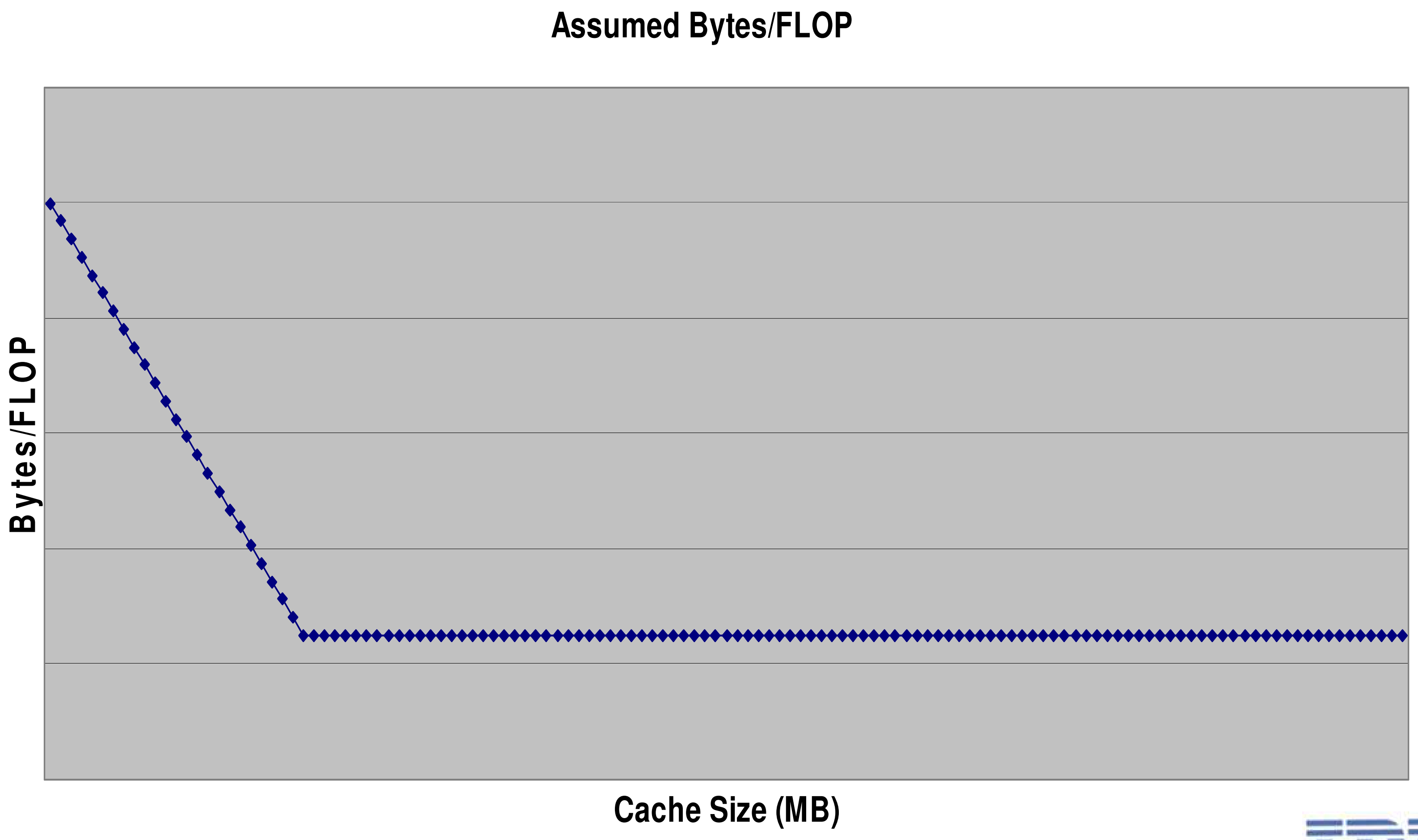
- Assume the time to solution is composed of a compute time proportional to peak GFLOPS plus a memory transfer time proportional to sustained memory bandwidth
- Assume "x Bytes/FLOP" to get:

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

- Use performance of 171.swim from SPECfp_rate2000 as a proxy for memory bandwidth

Sustained BW = (420 GB * (# of copies)) / (run time for 171.swim)

Make "Bytes/FLOP" a simple function of cache size



Make "Bytes/FLOP" a simple function of cache size

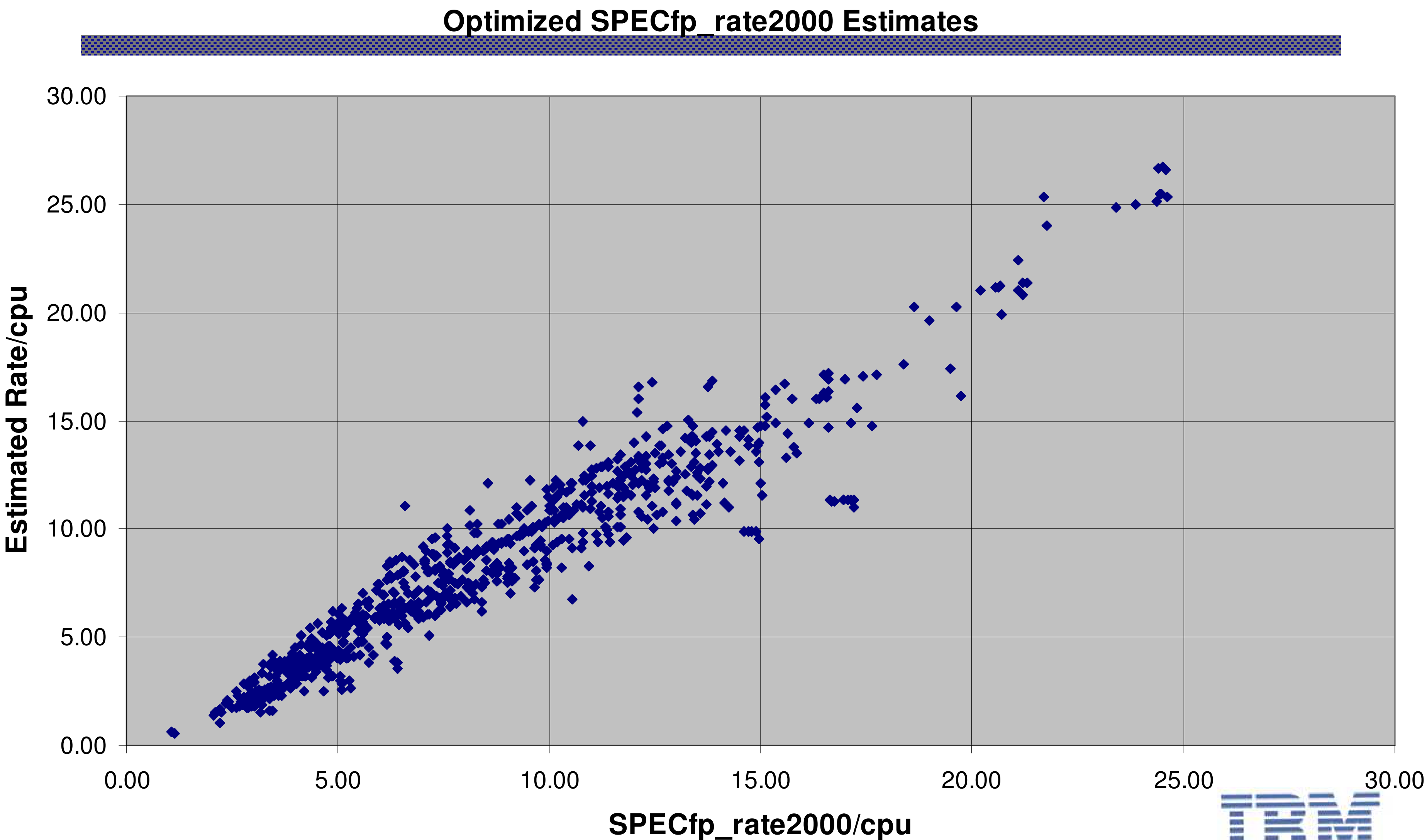
- Minimize RMS error to calculate the four parameters:
  - Bytes/FLOP for large caches
  - Bytes/FLOP for small caches
  - Size of asymptotically large cache
  - Coefficient of best-fit to SPECfp_rate2000/cpu

Make "Bytes/FLOP" a simple function of cache size

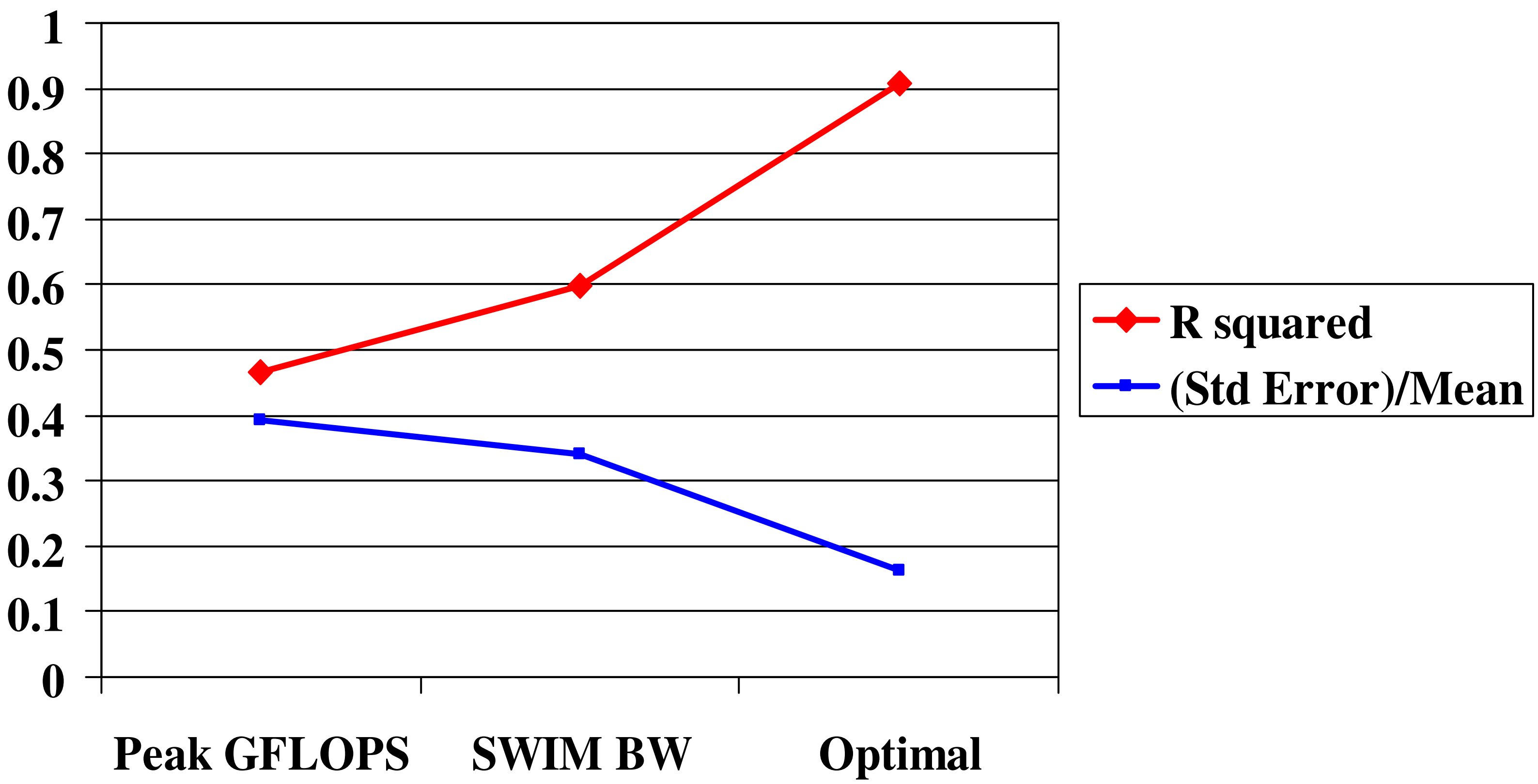
- Minimize RMS error to calculate the four parameters:
  - Bytes/FLOP for large caches
  - Bytes/FLOP for small caches
  - Size of asymptotically large cache
  - Coefficient of best-fit to SPECfp_rate2000/cpu
- Results (rounded to nearby round values):
  - Bytes/FLOP for large caches === **0.16**
  - Bytes/FLOP for small caches === **0.80**
  - Size of asymptotically large cache === **~12 MB**
  - Coefficient of best fit === ~6.4
  - The units of the coefficient are

    SPECfp_rate2000 / Effective GFLOPS

Does this Revised Metric predict SPECfp_rate2000?



Statistical Metrics



Comments

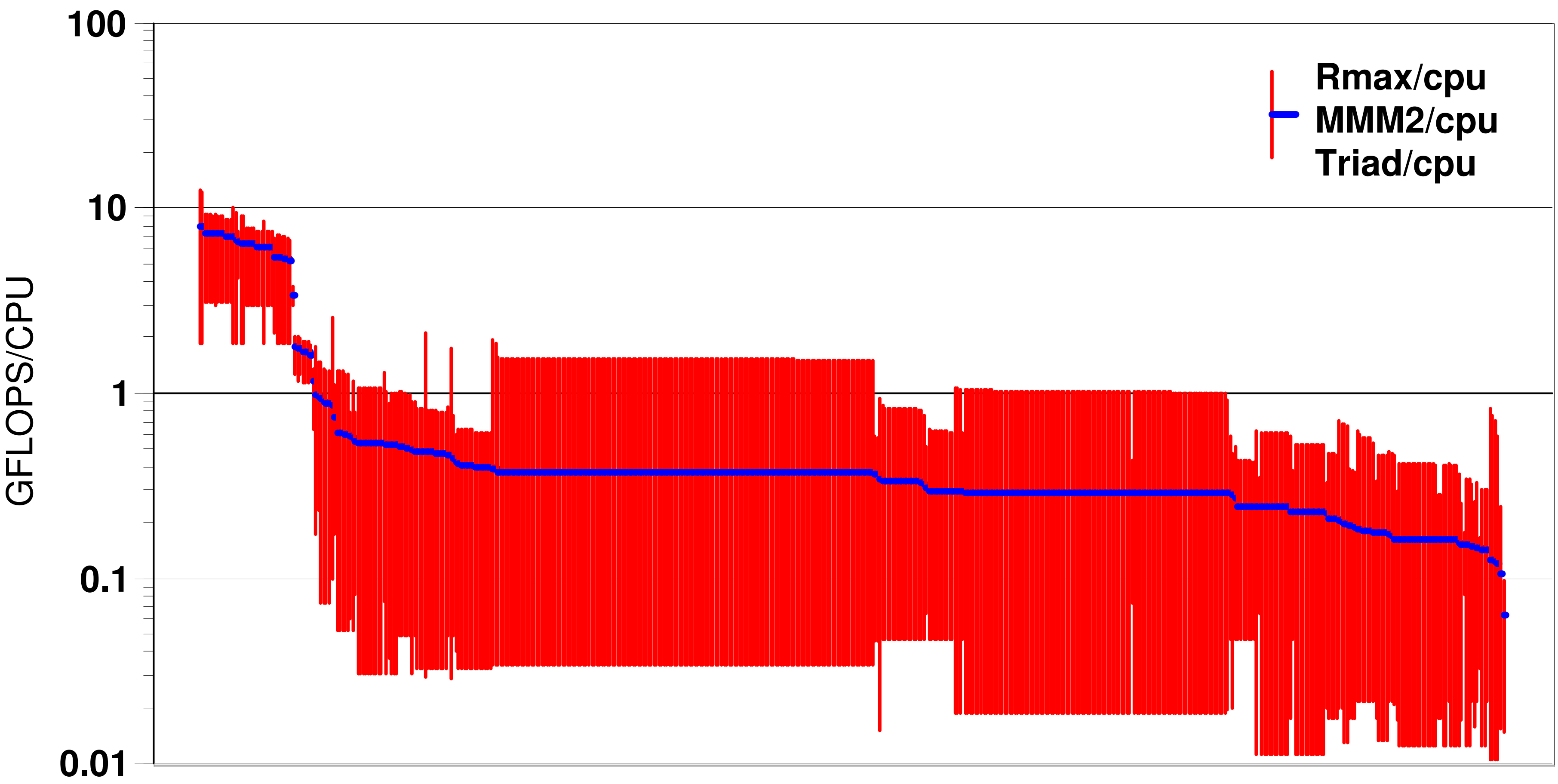
- Obviously, these coefficients were derived to match the SPECfp_rate2000 data set, not a "typical" set of supercomputing applications

- However, the results are encouraging, delivering a projection with 16% accuracy (one sigma) using a model based on only **one measurement** (sustainable memory bandwidth), plus specification of several architectural features

One more demonstration….

- I applied the preceding methodology to the November 2002 TOP500 list
- I estimated cache sizes and STREAM Triad bandwidth for all 500 systems
- I used the Bytes/FLOP parameters from a previous round of the SPECfp_rate2000 study
  - 1 B/F for small caches
  - 0.33 B/F for large caches
  - 6 MB is the cut-off for "large" caches

Performance Ranges per CPU



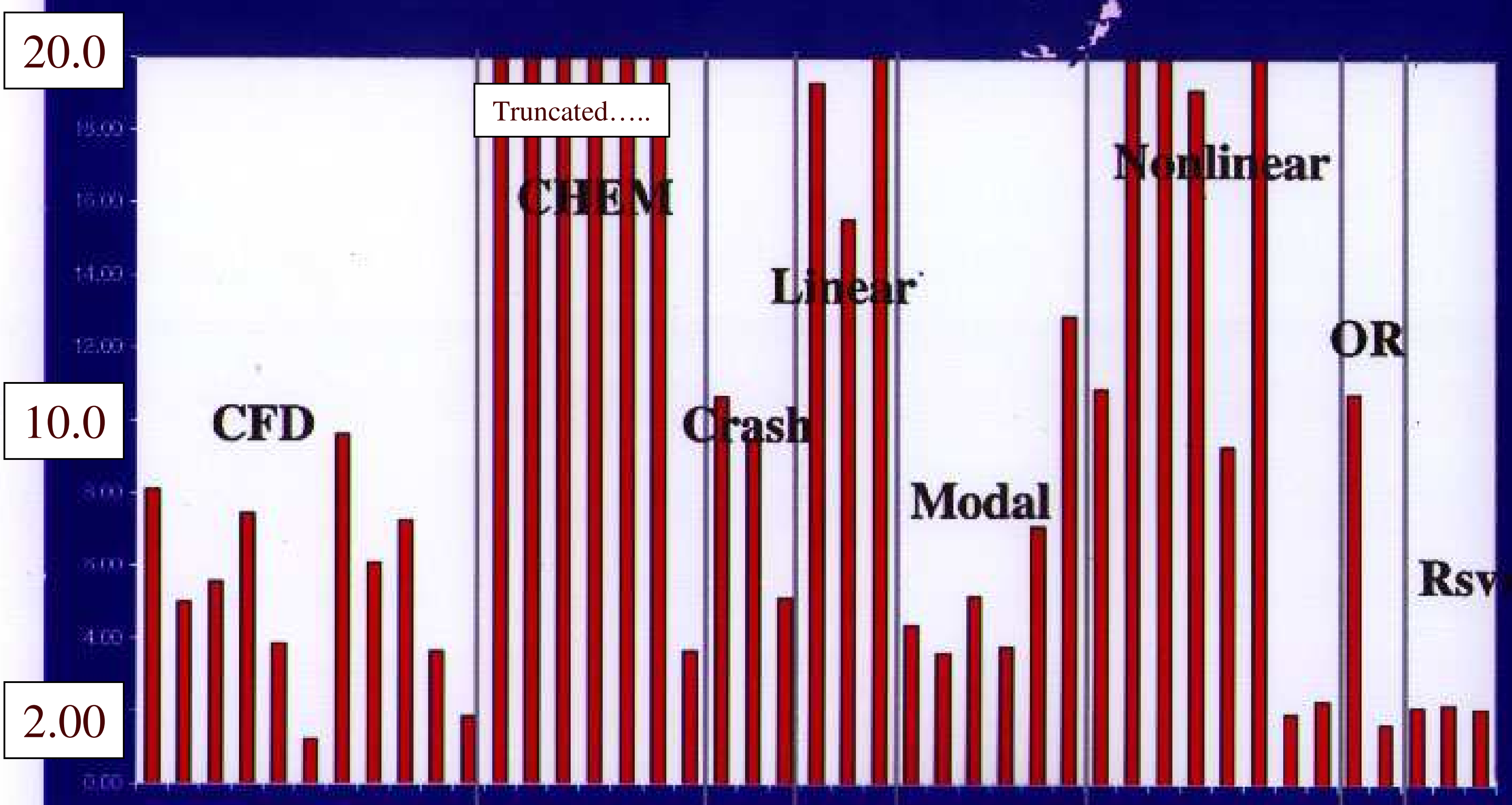
Comments

- Results shown per cpu
  - Earth Simulator is at position #30
- Sorted by "Balanced GFLOPS"
- Lower bound is STREAM Triad MFLOPS
  - Equal STREAM Triad MB/s divided by 12 Bytes/FLOP
- Upper bound is LINPACK Rmax

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



Using HPC Challenge Benchmark Components

- Pick an application area, e.g., CFD
- Pick a “typical” cache re-use factor for the cache size of the target system, e.g. 4
- Assume 8 Bytes/FLOP required from memory hierarchy
- Divide by re-use factor to get 2 Bytes/FLOP from main memory
- Assume 0.1 Bytes/FLOP using long messages on interconnect

An Example Model tuned for CFD

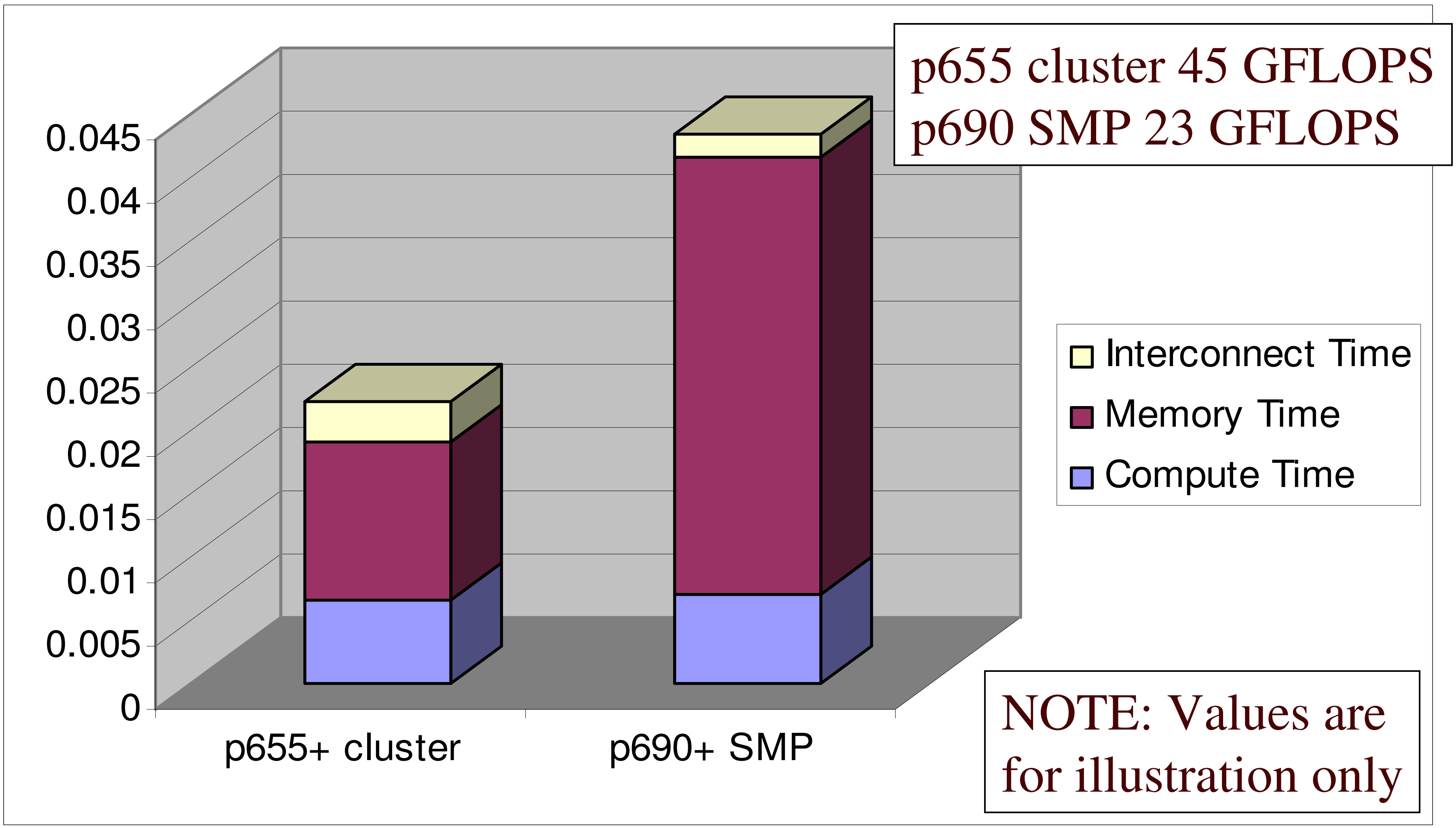
- Analyze applications and pick reasonable values:

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

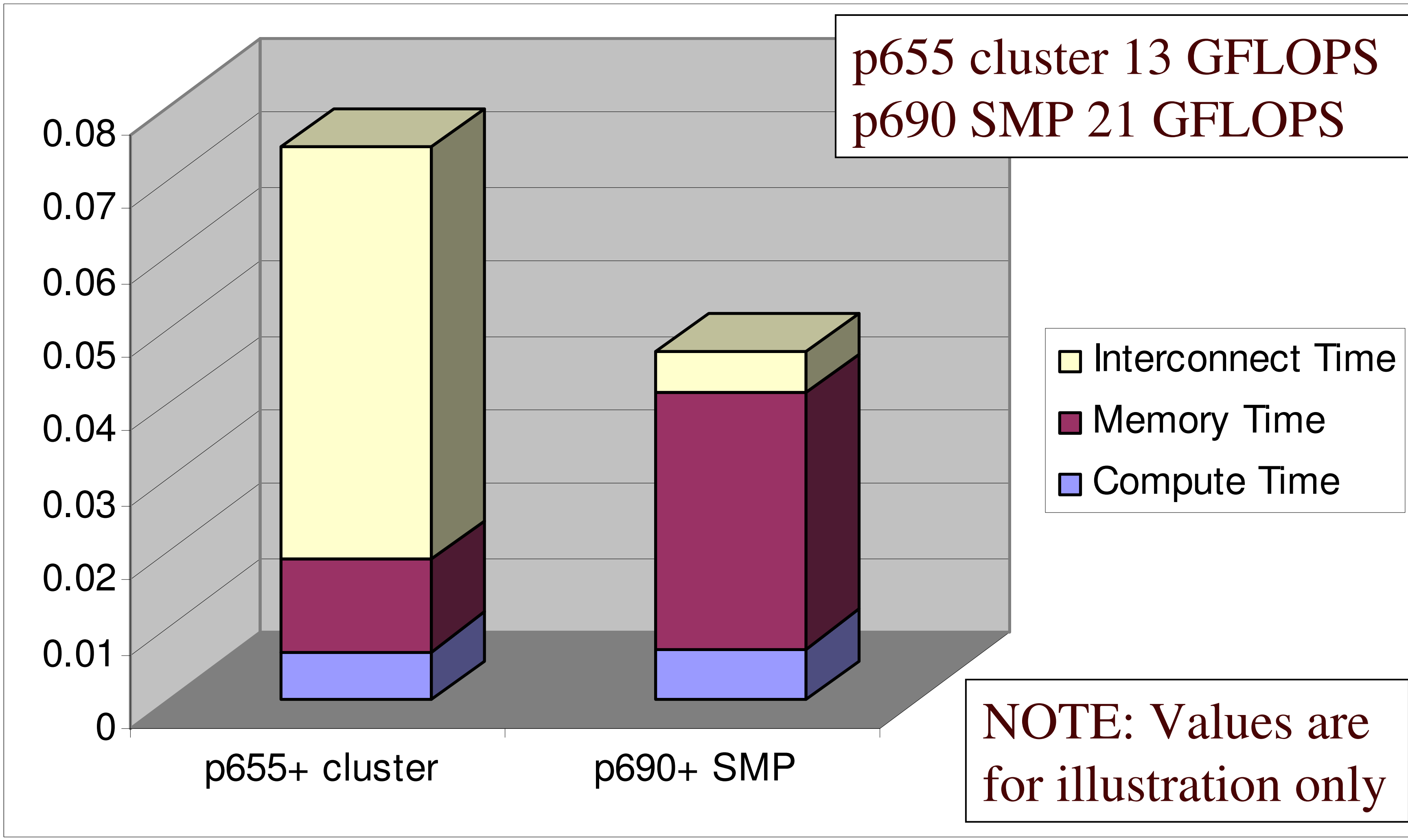
- Two cases: (values are representative, not measured!)
  - Assume long messages (network BW tracks PTRANS)
  - Assume short messages (network BW tracks GUPS)

- 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
  - Simple to measure
  - 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