The Surprising Effectiveness of Non-Overlapping, Sensitivity-Based Performance Models

John D. McCalpin, PhD
mccalpin@tacc.utexas.edu
Outline

• Motivation
• History of Sensitivity-Based Modeling
• Model Review
• New Results
• Analysis
Motivation

• Understanding the performance of full-scale applications on modern HPC clusters is challenging

• Detailed analysis by experts is not scalable to the broad set of important application workloads at shared supercomputing centers

• Hardware performance counters are poorly documented and unreliable
  – Tools built on top of counters cannot fix this!
History of this Modeling Effort

• Approach was developed using proprietary system settings & information while the author was working in HW development at SGI, IBM, and AMD

• Philosophy:
  – Start simple and add complexity only as needed
  – Stay connected with the “physics”
  – Model must have correct asymptotic properties
1st try: Peak CPU throughput

SPECfp_rate2000 vs Peak MFLOPS

\[ y = 268.96x + 855.59 \]

\[ R^2 = 0.5237 \]
2\textsuperscript{nd} Try: Sustained Memory Bandwidth

\[ y = 0.2493x - 0.4759 \]

\[ R^2 = 0.79612 \]
Optimal No-Overlap CPU Time + BW Time

Optimum Harmonic Combination (no overlap) vs SPECfp_rate2000

$y = 0.1599x - 0.0923$

$R^2 = 0.87929$
Part 1

REVIEW OF MODEL ASSUMPTIONS
Overview (1)

- Model is based on additive (non-overlapping) performance components
  - Time = Work / Rate
  - \( T_{total} = \sum T_i = \sum \frac{W_i}{R_i} \)

- The “Rate” components are known (or measured) constants for each hardware configuration
- The total time is measured for each configuration
- The “Work” components are the unknowns
Overview (2)

• Work coefficients determined by least-squares fit to the data
  – Overdetermined systems are less sensitive to noise
  – Deviations from linearity point to limitations of model

• Performance components are based on whatever can be varied by machine reconfiguration
  – CPU frequency, number of cores used, memory frequency, number of DRAM channels populated, etc.
Overview (3)

• Specific Models
  - \( T_{total} = T_{cpu} + T_{memory\ bandwidth} \)
  - \( T_{total} = T_{cpu} + T_{memory\ bandwidth} + T_{memory\ latency} \)

• Interpretation:
  - Compute does not overlap with memory access
  - Contiguous Memory Accesses overlap with other Contiguous Memory Accesses about as well as they do in the STREAM benchmark
  - Exposed memory latencies do not overlap with either Contiguous Memory Accesses or Compute
Application to SPECfp_rate

- In 2007, I mined the SPEC results database for all Opteron system results on Linux using the PathScale compiler for all SPECfp benchmarks (CPU2000 & CPU2006)
- E.g., for SPECfp2006 this included
  - 29 published result sets
  - 13 different Hardware + Benchmark configurations
    - Varying number of copies of the benchmark run concurrently
    - Varying CPU frequency
    - Varying number of sockets (changes idle memory latency)
  - Execution time varied by 1.5x to 1.9x across results
SPECfp2006 (cont’d)

• Separate models were built for each of the 17 benchmarks
• Two-term models could not achieve <10% errors on most of the benchmarks
• Three-term model results vs input data:
  – 2 of 17 benchmarks showed ~4% RMS error
  – 7 of 17 benchmarks showed 2%-3% RMS error
  – 3 of 17 benchmarks showed 1%-2% RMS error
  – 5 of 17 benchmarks showed <1% RMS error
SPECfp2006 (cont’d)

• For any hardware configuration the model allows computing the times associated with each component and therefore the time breakdown

• Reference System
  – 2-Socket AMD Opteron (Revision F)
  – 2.8 GHz dual-core
  – DDR2/667 (2 DIMMs per channel)
SPECfp_rate2006 run-time contributions on late 2006-era reference system
New Results

• SPEC benchmarks are no longer useful for these experiments (for many reasons)
• New benchmarks chosen from TACC’s workload
• Single-node runs on Xeon E5-2660 v3 (Haswell EP)
  – WRF (mesoscale weather) – today’s main topic
  – FLASH4 (forced 3D turbulence) – similar to WRF
  – NAMD (molecular dynamics) – very different
WRF (conus 12km)

- Tests run on a Xeon E5-2660 v3 (Haswell EP) with HyperThreading disabled
- **CPU** Frequency varied from 1.2 to 2.9 GHz
- **DRAM** rate varied from 1.333 to 2.133 GT/s
- 1 core, 10 core (1 socket), 20 core
- These cases were MPI-only, no OpenMP and no more than 1 MPI task per physical core
- 32 20-core configurations tested – data used is median timing from a set of 3 consecutive runs
WRF (conus 12km) Model

• Model 1: \[ T_{obs} = T_{cpu} + T_{bw} = \frac{W_{cpu}}{R_{cpu}} + \frac{W_{bw}}{R_{bw}} \]
  - \( R_{cpu} \) = CPU GHz
  - \( R_{bw} \) = STREAM Triad Bandwidth per core in GB/s

• Model 2: add a constant time to account for IO
  - IO time is expected to be approximately independent of both CPU Frequency and Memory Bandwidth

• Work values derived by “best fit” to total time
Models 1 & 2: Projected vs Observed Time for WRF on Haswell EP

\[ y = 1.0572x - 8.8856 \quad R^2 = 0.99237 \]

\[ y = 0.9933x + 1.0183 \quad R^2 = 0.99455 \]
What about “reality”

• Ever since my 2007 presentation I have been curious about whether the $W_{BW}$ and (optional) $T_{const}$ terms bear any relation to reality.

• Last week I added memory controller performance counters to these runs to see…
What about “reality”

- Model 2 fit says $W_{bw} = 7075$ GB
- DRAM Counters report 7027 GB
  - Less than 0.7% difference
- Model 2 fit says $T_{const} = 13.6$ seconds
- WRF reports IO time of 12.2 seconds
  - Difference is about 1% of total execution time
- Not all results are this good, but this is OK
FLASH4

• Similar to WRF, but mostly compute-bound rather than bandwidth-bound

• Slightly larger range of timings
  – CPU frequencies can be varied over a larger range than DRAM frequencies
NAMD

• Performance is almost perfectly linear in CPU frequency
  – CPU only: model error ~1.5%
  – CPU + bandwidth: model error ~0.7%
  – CPU + constant: model error ~0.2%

• NAMD has interesting dependencies on the instruction set, but that is a topic for another day…
Analysis

• Results across a variety of hardware configurations can be used to derive robust bounds on the coefficients of the models if the assumption of non-overlapping execution time components is relaxed.

• These bounds are typically rather weak, but the technique still provides excellent fits to the data.
Analysis (continued)

• A more general statement of bounds in total execution time for a component based model is:

\[ \max_i T_i \leq T_{total} \leq \sum_i T_i \]

• The lower bound is full overlap
• The upper bound is no overlap
• I usually see answers near the upper bound
Analysis (continued)

• What can we say about bounds on the work components?
• Assume (temporarily) that the $R_i$ values are valid
• For a single experiment we have the trivial bound:

\[ T_i \leq T_{obs} \quad \text{or} \quad W_i \leq T_{obs} \times R_i \quad \text{for all } i \]

• This is a very weak bound, so it is not particularly useful
Analysis (continued)

• For multiple tests we can get tighter upper bounds
• For example, if one Rate component is changed and the execution time changes, then this can be used to derive a lower bound on the corresponding Work component
• The algebra is not particularly enlightening, but an example is illustrative…
Formal bounds on $W_i$ estimates for 2:1 Work ratio and 2:1 and 4:1 Rate ratio experiments

- Lower Bound
- Actual $W$
- Upper Bound
Analysis – Summary

• The formal analysis shows very weak bounds on the ability to estimate work components from modest variations in hardware rates
  – For multicore processors using most or all cores, the models are extremely effective
  – (Not shown today) When running on a single core, the models usually derive a $W_{bw}$ term that is much too small
    • These cases are probably seeing overlap of $T_{cpu}$ and $T_{bw}$
    • Similar anomalies have shown up (rarely) in SMP scaling studies
• For a fixed architecture, this simple additive execution time modeling methodology can be extremely accurate
  – Both in prediction of total execution time and in deriving IO time and memory traffic
• Data collection and model building can be largely automated – suitable for modest workload surveys
Ongoing & Future Work

• Easy/Low Risk projects
  – multi-node runs with varying InfiniBand network rates
  – instruction issue throttling

• Potential implementation difficulty
  – Varying MPI short-message latency and/or overhead

• High Risk (but necessary)
  – Continuing to perform and evaluate cross-platform projections, e.g., Haswell to Knights Landing
BACKUP SLIDES
What about machine changes?

• Can I bridge from 10-core Xeon E5 v3 to 12-core Xeon E5 v3 with a different DRAM configuration?
  – The larger cache on the 12-core resulted in reduced $W_{bw}$ when using 10 cores – outside direct scope of model
  – The reduced memory bandwidth of the single-rank DIMM configuration was reflected in a ~10% reduction in STREAM bandwidth (and hence, $R_{bw}$)
  – Detailed analysis shows that the effective bandwidth penalty in this WRF test case is ~14%-15%
  – Currently attempting to model this using measured DRAM page conflict rates, but the errors are only 3%-4%, so….