Power-Aware Predictive Models of Hybrid (MPI/OpenMP) Scientific Applications on Multicore Systems

Charles Lively III*, Xingfu Wu*, Valerie Taylor*, **Shirley Moore+**, Hung-Ching Chang^, Chun-Yi Su^, and Kirk Cameron^

*Department of Computer Science & Engineering, Texas A&M University

+Electrical Engineering and Computer Science, University of Tennessee-Knoxville

^Department of Computer Science, Virginia Tech

Introduction

- Current trends in HPC put great focus on constraining power consumption without decreasing performance.

- Multicore systems are hierarchical and can consist of heterogeneous components.

- Understanding the mapping of scientific applications onto multicore and heterogeneous systems is necessary to optimize performance and power consumption.

- Goal: Accurate models for performance and power consumption of scientific applications on multicore and heterogeneous systems

Approach and Research Questions

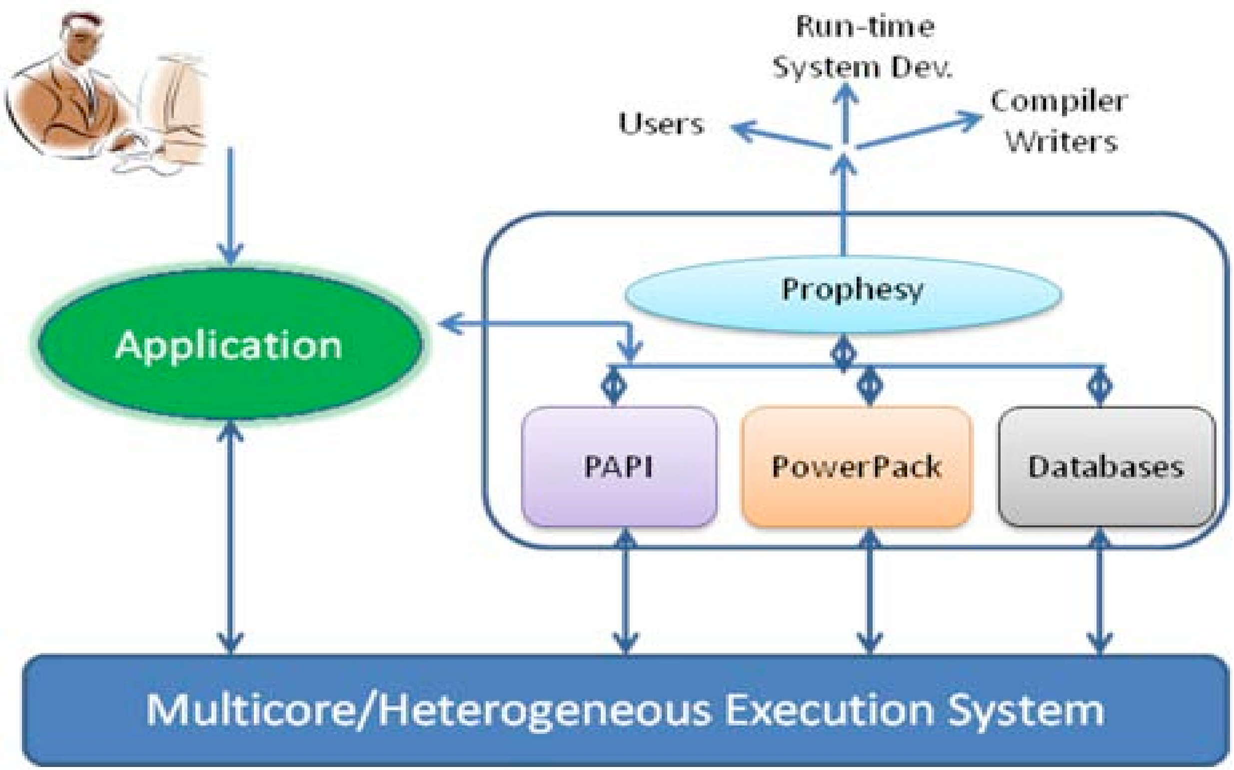
- **Application-specific models** are used to explore common and different characteristics of hybrid (MPI+OpenMP) scientific applications.

1. Which combination of performance counters should be used to model performance and power consumption of each component?
   - System, CPU, memory
2. Which application and system characteristics most affect runtime and power consumption?
3. Which aspects of hybrid applications and systems need to be optimized to improve power-performance on multicore systems?

General Methodology

- Explore which application characteristics (via performance counters) affect power consumption of system, CPU, and memory
- Develop accurate models based on hardware counters for predicting power consumption of system components
- Develop different models for each application class (Previous work used same set of performance counters across all applications).
- Validate predictions using actual power measurements

MuMMI Framework



Multiple Metrics Modeling Infrastructure (MuMMI)
http://www.mummi.org/

SystemG

Configuration of SystemG	
Mac Pro Model Number	MA970LL/A
Total Cores	2,592
Total Nodes	324
Cores/Socket	4
Cores/Node	8
CPU Type	Intel Xeon 2.8Ghz Quad-Core
Memory/Node	8GB
L1 Inst/D-Cache per core	32-kB/32-kB
L2 Cache/Chip	12MB
Interconnect	QDR Infiniband 40Gb/s

- Largest power-aware compute system in the world
- Over 30 power and thermal sensors per node
- http://scape.cs.vt.edu/

Modeling Methodology

- Training Set: 5 training execution configurations
  - 1x1, 1x2, 1x3, 1x8, and 2x8
- 16 larger execution configurations are predicted.
  - 1x4, 1x5,…3x8, 4x8, 5x8, …..16x8
- 40 performance counter events are captured.
- Performance counter events are normalized per cycle.
- Performance-Tuned Supervised Principal Component Analysis Method is utilized to select combination of performance counters for each application.

Performance-Tuned Supervised PCA

1. Compute Spearman's rank correlation for each application and system component

1. Eliminate counters with low correlation

2. Compute regression model based upon performance counter event rates

3. Eliminate performance counters with negligible regression coefficients

4. Compute principal components of reduced performance counter space

5. Use performance counters with highest PCA vectors to build multivariate linear regression model

Repeat the process for each application/system component pair.

Performance-Tuned Supervised PCA

1. Compute Spearman's rank correlation.
2. Eliminate counters with low correlation, based on β_{ai} threshold.

Example: BT-MZ correlation values for runtime

Hardware Counter	Correlation Value
PAPI_TOT_INS	0.9187018
PAPI_FP_OPS	0.9105984
PAPI_L1_TCA	0.9017512
PAPI_L1_DCM	0.8718455
PAPI_L2_TCH	0.8123510
PAPI_L2_TCA	0.8021892
Cache_FLD	0.7511682
PAPI_TLB_DM	0.6218268
PAPI_L1_ICA	0.6487321
Bytes_out	0.6187535

Performance-Tuned Supervised PCA

3. Compute regression model based upon counter event rates.
4. Eliminate counters will negligible regression coefficients.

Hardware Counter	Regression Coefficient
PAPI_TOT_INS	0.04183
PAPI_FP_OPS	-0.04219
PAPI_L1_TCA	0.00165
PAPI_L2_TCH	0.01875
PAPI_L2_TCA	0.100187
Cache_FLD	-0.71548
PAPI_TLB_DM	0.008418

Hardware Counter	Regression Coefficient
PAPI_L1_ICA	-0.000048
Bytes_out	0.00085

Performance-Tuned Supervised PCA

5. Compute principal components of reduced performance counter space.

   – Determine the variance of each principal component

   – Use the principal components containing at least 90% of data variance
     • Typically first 2 principal components

   – Select counters with significant PCA coefficients

5. Use performance counters with highest PCA vectors to build multivariate linear regression model:

$$y=\beta_0+ \beta_1 * r_1+ \beta_2 r_2+ \beta_3 * r_3 \ldots\ldots\ldots + \beta_n * r_n$$

Performance Counter Events

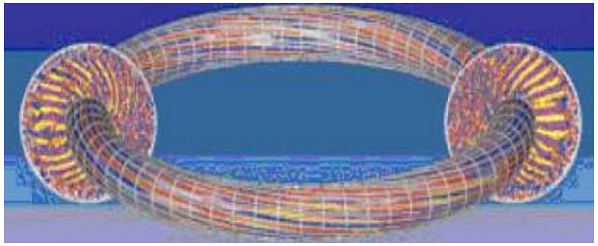
- 15 performance counters used in this Work

Counter	Description
PAPI_TOT_INS	Total instructions completed
PAPI_TLB_ DM	TLB misses
PAPI_L1_TCA	L1 cache total accesses
PAPI_L1_ICA	L1 instruction cache accesses
PAPI_L1_TCM	L1 total cache misses
PAPI_L1_DCM	L1 data cache misses
PAPI_L2_TCH	L2 total cache hits
PAPI_L2_TCA	L2 total cache accesses
PAPI_L2_ICM	L2 instruction cache misses
PAPI_BR_INS	Branch instructions completed
PAPI_RES_STL	System stalls on any resource
Cache_FLD_per_instruction	L1 writes/reads/hits/misses
LD_ST_stall_ per_cycle	Load/stores stalls per cycle

Applications

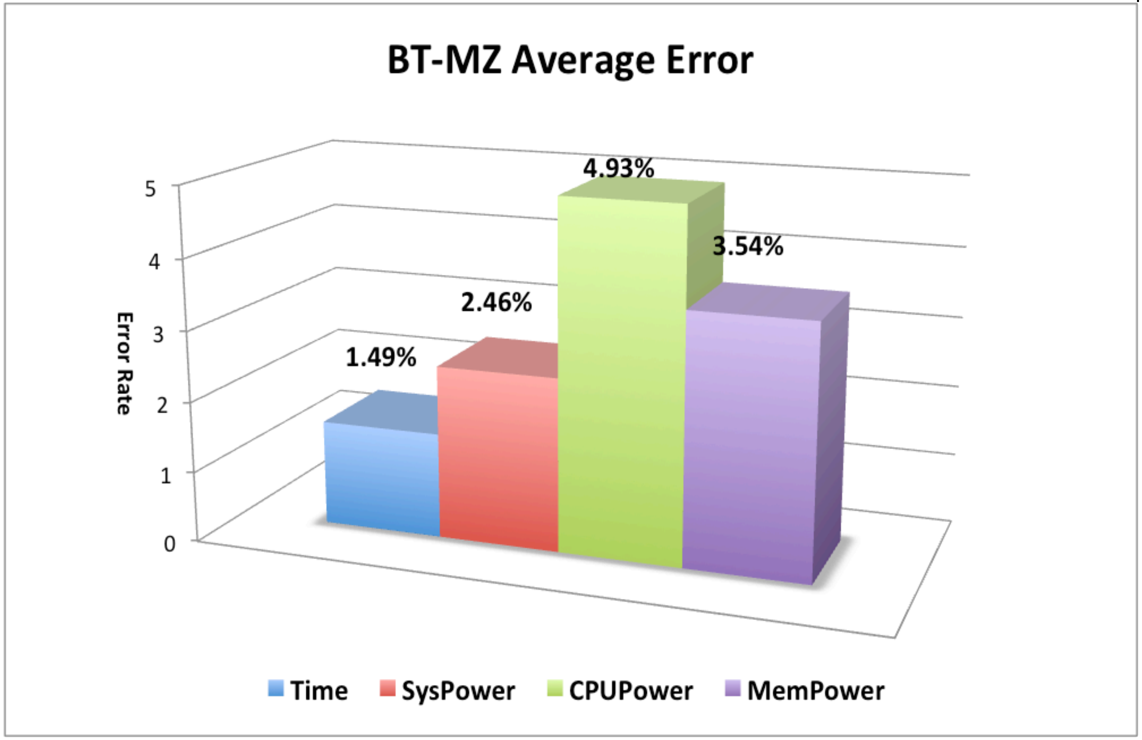
- NAS Multizone Benchmark Suite
  - written in Fortran
  - Uses MPI and OpenMP for communication
  - **Block Tri-diagonal algorithm (BT-MZ)**
    - represents realistic performance case for exploring discretization meshes in parallel computing
  - **Scalar Penta-diagonal algorithm (SP-MZ)**
    - representative of a balanced workload
  - **Lower-Upper symmetric Gauss-Seidel algorithm (LU-MZ)**
    - coarse-grain parallelism of LU-MZ is limited to 16 MPI processes

- Large-Scale Scientific Application
  - **Gyrokinetic Toroidal code (GTC)**
    - 3D particle- in-cell application
    - Flagship SciDAC fusion microturbulence code
    - written in Fortran90
    - Uses MPI and OpenMP for communication



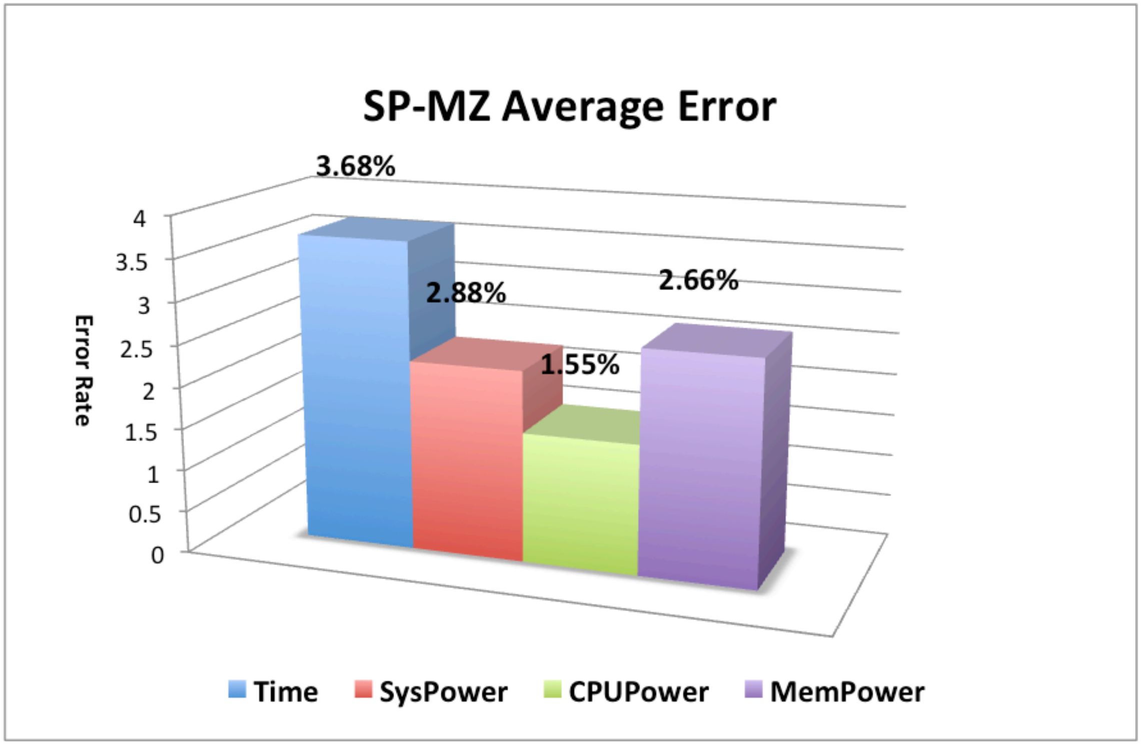
BT-MZ Results

	Time		System Power		CPU Power		Memory Power	
BT-MZ	Cache_FLD	-1.611	PAPI_L2_TCH	-1.6769	PAPI_L1_TCM	3.5432	PAPI_L1_TCA	0.0763
	PAPI_TOT_INS	0.0967	PAPI_L2_TCA	1.5967	PAPI_L2_TCH	-3.9389	PAPI_L1_DCM	4.0496
	PAPI_L2_TCH	0.2992	PAPI_RES_STL	0.0803	PAPI_RES_STL	0.3967	PAPI_L2_TCH	-1.9443
	PAPI_L2_TCA	1.2152					PAPI_L2_TCA	2.1806



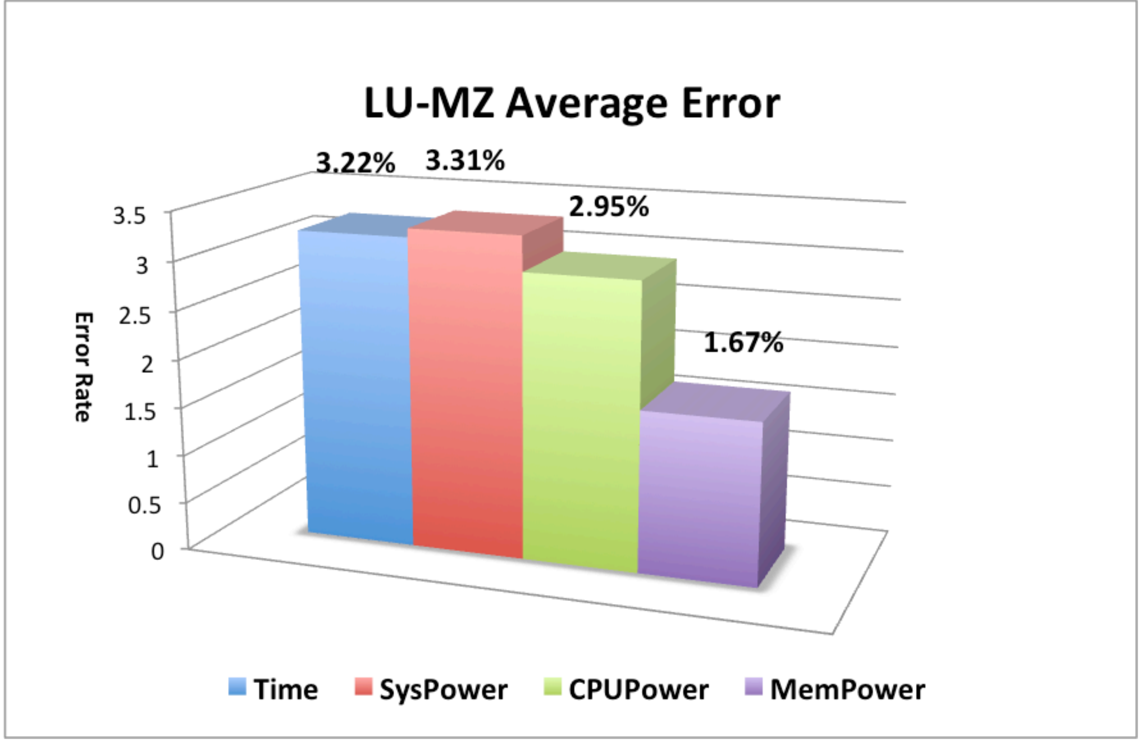
SP-MZ Results

	Time		System Power		CPU Power		Memory Power	
SP-MZ	PAPI_TOT_INS	0.1818	PAPI_L1_ICA	0.355	LD_ST_stall	0.1917	Cache_FLD	0.4563
	PAPI_L1_TCA	0.0744	PAPI_L2_TCH	-1.3452	PAPI_L1_TCM	1.5008	LD_ST_stall	0.0192
	PAPI_L2_TCH	-1.2834	PAPI_L1_TCM	0.9911	PAPI_L2_TCH	-1.6914	PAPI_L2_TCH	-3.5895
	PAPI_L1_TCM	1.1761					PAPI_L2_TCA	3.1151



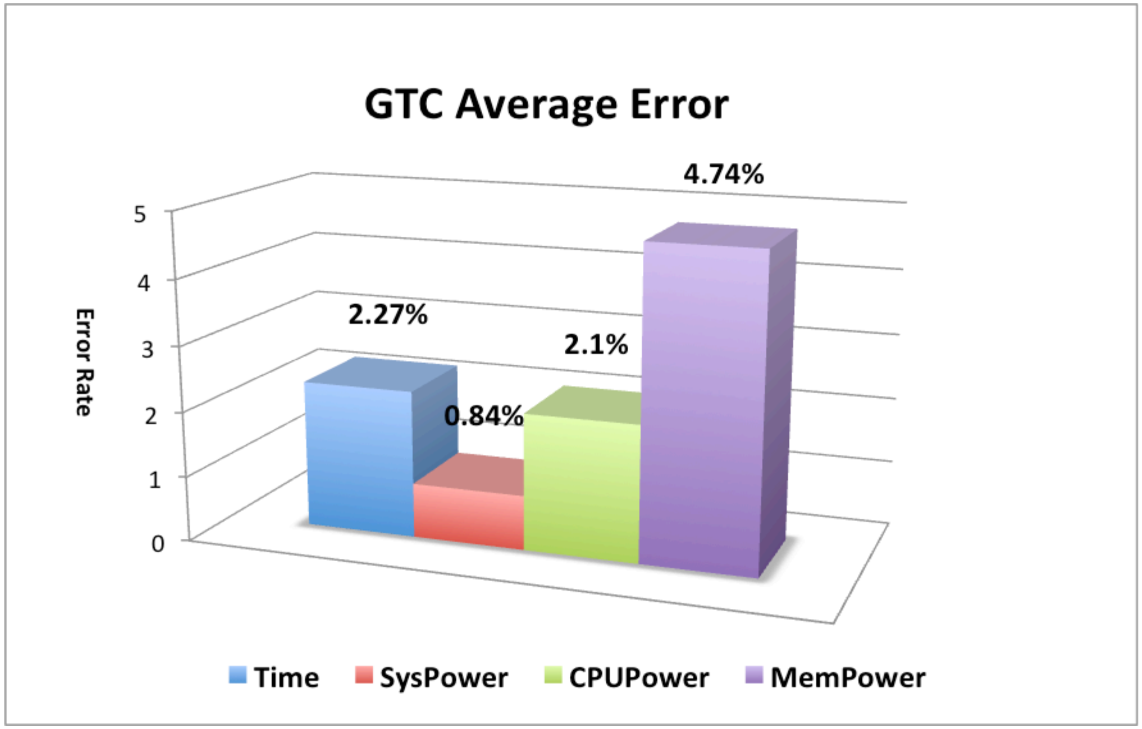
LU-MZ Results

	Time		System Power		CPU Power		Memory Power	
LU-MZ	Cache_FLD	-0.0006	LD_ST_stall	0.0166	LD_ST_stall	0.0869	PAPI_L1_TCA	0.27923
	PAPI_TOT_INS	0.0011	PAPI_L2_TCH	-0.9886	PAPI_L2_TCH	-8.0003	PAPI_L2_TCH	-3.9574
	PAPI_TLB_DM	3.9085	PAPI_L2_TCA	1.0411	PAPI_L2_TCA	7.9137	PAPI_RES_STL	-0.29141
	PAPI_L2_TCH	-0.0591	PAPI_RES_STL	0.025				



GTC Results

	Time		System Power		CPU Power		Memory Power	
GTC	PAPI_TOT_INS	0.0006	PAPI_RES_STL	1.5689	PAPI_RES_STL	0.9261	PAPI_TOT_IN	0.169617
	PAPI_L2_TCH	-1.8976	PAPI_L2_TCH	-3.2505	PAPI_TOT_IN	0.2663	PAPI_L2_TCH	-2.881
	PAPI_L2_TCA	1.9351	PAPI_L1_TCA	1.6916	PAPI_L1_TCA	0.0816	PAPI_L2_ICM	2.7119
	PAPI_BR_INS	-0.0381			PAPI_L2_TCH	-1.2640		

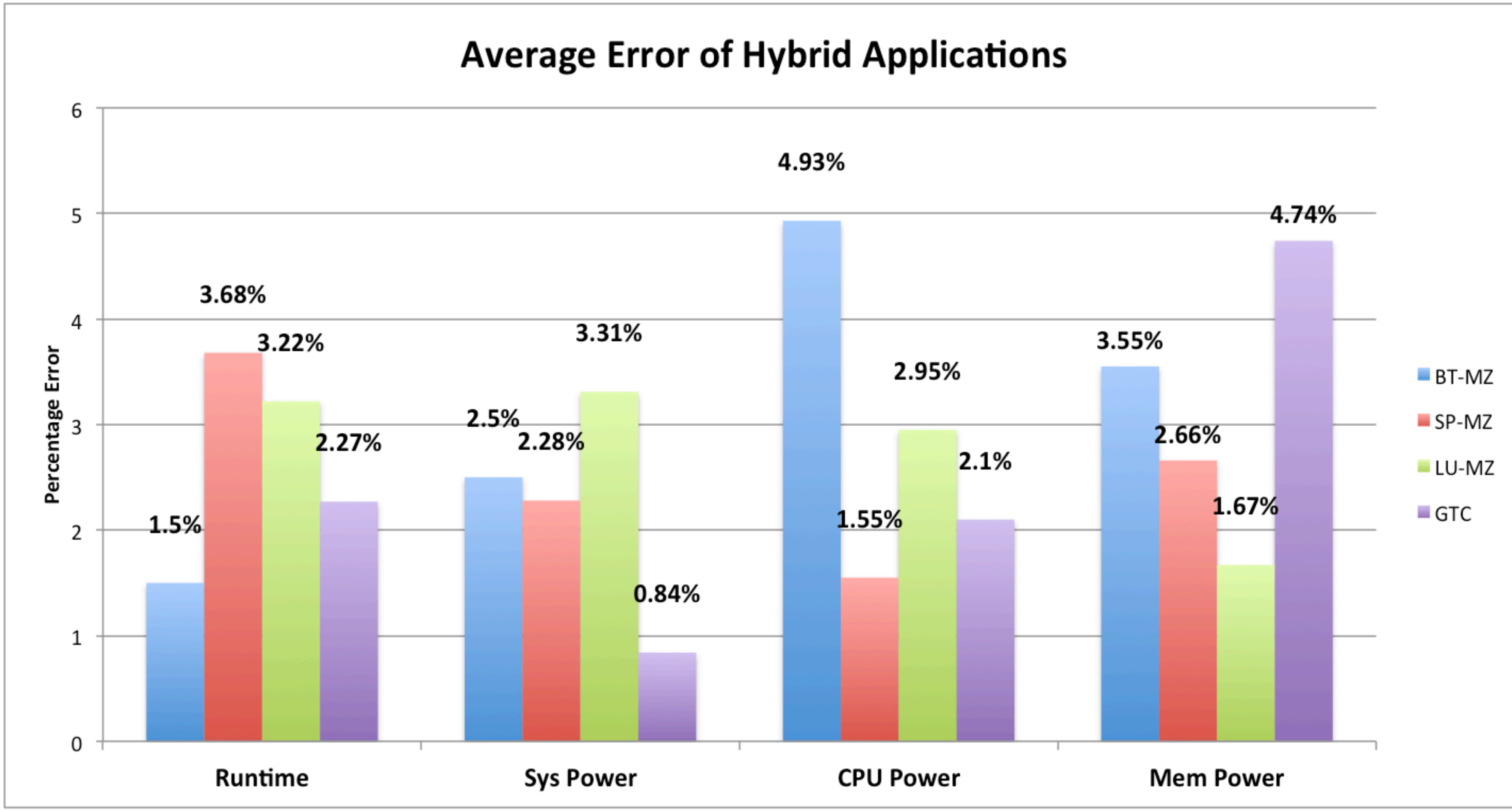


Application-specific Modeling

- Multivariate regression coefficients

	Time		System Power		CPU Power		Memory Power	
BT-MZ	Cache_FLD	-1.611	PAPI_L2_TCH	-1.6769	PAPI_L1_TCM	3.5432	PAPI_L1_TCA	0.0763
	PAPI_TOT_INS	0.0967	PAPI_L2_TCA	1.5967	PAPI_L2_TCH	-3.9389	PAPI_L1_DCM	4.0496
	PAPI_L2_TCH	0.2992	PAPI_RES_STL	0.0803	PAPI_RES_STL	0.3967	PAPI_L2_TCH	-1.9443
	PAPI_L2_TCA	1.2152					PAPI_L2_TCA	2.1806
SP-MZ	PAPI_TOT_INS	0.1818	PAPI_L1_ICA	0.355	LD_ST_stall	0.1917	Cache_FLD	0.4563
	PAPI_L1_TCA	0.0744	PAPI_L2_TCH	-1.3452	PAPI_L1_TCM	1.5008	LD_ST_stall	0.0192
	PAPI_L2_TCH	-1.2834	PAPI_L1_TCM	0.9911	PAPI_L2_TCH	-1.6914	PAPI_L2_TCH	-3.5895
	PAPI_L1_TCM	1.1761					PAPI_L2_TCA	3.1151
LU-MZ	Cache_FLD	-0.0006	LD_ST_stall	0.0166	LD_ST_stall	0.0869	PAPI_L1_TCA	0.27923
	PAPI_TOT_INS	0.0011	PAPI_L2_TCH	-0.9886	PAPI_L2_TCH	-8.0003	PAPI_L2_TCH	-3.9574
	PAPI_TLB_DM	3.9085	PAPI_L2_TCA	1.0411	PAPI_L2_TCA	7.9137	PAPI_RES_STL	-0.29141
	PAPI_L2_TCH	-0.0591	PAPI_RES_STL	0.025				
GTC	PAPI_TOT_INS	0.0006	PAPI_RES_STL	1.5689	PAPI_RES_STL	0.9261	PAPI_TOT_IN	0.169617
	PAPI_L2_TCH	-1.8976	PAPI_L2_TCH	-3.2505	PAPI_TOT_IN	0.2663	PAPI_L2_TCH	-2.881
	PAPI_L2_TCA	1.9351	PAPI_L1_TCA	1.6916	PAPI_L1_TCA	0.0816	PAPI_L2_ICM	2.7119
	PAPI_BR_INS	-0.0381			PAPI_L2_TCH	-1.2640		

Overall Prediction Accuracy



Related Work

- SoftPower: Power Estimations (Lim, Porterfield, & Fowler)
  - Goal: Develop a surrogate power estimation model using performance counters on the Intel Core i7
  - Use Spearman's rank correlation and robust regression analysis for training runs to derive small set of counters and correlation coefficients
  - Evaluation shows less than 14% error (median 5.3% error)
- Power Estimation &Thread Scheduling (Singh, Bhadhauria, & McKee)
  - Goal: Use hardware counter model to predict power consumption on a system
  - Use Spearman's rank correlation to choose top counter from each of four categories: FP, memory, stalls, instructions retired
  - Derive piecewise linear function for estimating core power
- Reducing Energy Usage with Memory & Computation-Aware Dynamic Frequency Scaling (Laurenzano, Meswani, Carrington, Snavely, Tikir, & Poole)
  - Application signatures characterize execution regions
  - Signatures matched with set of benchmarks intended to form a covering set (machine characterization of expected power consumption over space of execution patterns and clock frequencies
  - Derive dynamic application frequency management strategy

Conclusions

- Predictive performance models for hybrid MPI+OpenMP scientific applications.
  - Execution time
  - System power consumption
  - CPU power consumption
  - Memory power consumption

- 95+% accuracy across four hybrid (MPI+OpenMP) scientific applications

Future Work

- Explore use of microbenchmarks and application classes to derive application-centric models

- Finer-granularity analysis of large-scale hybrid scientific applications
  - Do set of hardware counters and coefficients vary with application region?

- Modeling and prediction across different application input sizes and frequency settings
  - Can hardware counter measurements drive a dynamic frequency scaling strategy?

Acknowledgments

- This work is supported by NSF grants CNS-0911023, CNS-0910899, CNS-0910784, CNS-0905187.
- The authors would like to acknowledge Stephane Ethier from Princeton Plasma Physics Laboratory for providing the GTC code.

Questions?