A Framework for Architecture-Level Power, Area, and Thermal Simulation and its Application to Network-on-chip Design Exploration

Ming-yu Hsieh
Sandia National Laboratories
NM, USA
myhsieh@sandia.gov

Kevin Thompson
Dept. of EE
New Mexico State University
NM, USA
kevint@nmsu.edu

William Song
Dept. of ECE
Georgia Institute of Technology
GA, USA
wjhsong@gatech.edu

Arun Rodrigues
Sandia National Laboratories
NM, USA
afrodri@sandia.gov

Rolf Riesen
Sandia National Laboratories
NM, USA
rolf@sandia.gov

ABSTRACT

In this paper, we describe the integrated power, area and thermal modeling framework in the Structural Simulation Toolkit (SST) for large scale high performance computer simulation. It integrates various power and thermal modeling tools and computes run-time energy dissipation for core, network on chip, memory controller and shared cache. It also provides functionality to update the leakage power as temperature changes.

We Illustrate the utilization of the framework by applying it to explore interconnect options in manycore systems with consideration of temperature variation and leakage feedback. We compare power, energy-delay-area product (EDAP), and energy-delay product (EDP) of four manycore configurations-1 core, 2 cores, 4 cores and 8 cores per cluster. Results from simulation with or without consideration of temperature variation both show that the 4-core per cluster configuration has the best EDAP and EDP. Even so, considering temperature variation increases total power dissipation. We demonstrate the importance of considering temperature variation in the design flow. With this power, area and thermal modeling capability, SST can be used for Hardware/Software co-design of future Exascale systems.

Categories and Subject Descriptors

C.0 [**Computer Systems Organizations**]: General – *modeling of computer architecture.*

General Terms

Performance.

Keywords

NoC, simulation framework, performance modeling, power consumption.

1. INTRODUCTION

With the cost of a megawatt of power for one year ranging from $590K to $1.85 Million [6], the power bill for future many-megawatt machines could outweigh their purchase cost. Additionally, increased power consumption often requires expensive new facilities to deliver and manage the vast amount of electricity, as well as facilities to cool the energy-dissipating computers. This is demonstrated by recent data centers constructed by companies like Google and Apple with costs in the hundreds of millions of dollars [13]. The problem is one of worldwide importance as 2010 server power consumption is estimated at $44.5 Billion [15].

Traditional architectural simulators for High Performance Computing (HPC) narrowly focus on the performance and power dissipation of part of the system. This makes it difficult for system designers to make educated decisions about how to manage the energy for the entire system.

The Structural Simulation Toolkit (SST) [14] aims address this problem. The SST provides a framework for simulating large-scale HPC systems. This simulator allows parallel simulation of large machines at scale to understand both performance and energy consumption. SST couples parameterizable models for processors, memory, and network subsystems. These models all have access to a

uniform interface to power and thermal modeling libraries that estimate power dissipation and update temperature.

This paper introduces the technology interface in SST, the core of integrated energy, power, and temperature simulation. It receives access counts from the SST components and calculates the power dissipation and temperature change. The power and thermal libraries are architecture modeling tools that provide power and temperature calculation methods to the technology interface. In the current implementation, the power library includes McPAT [9], IntSim [16, 17], and ORION [19], and HotSpot [20] is used as a thermal library. The technology interface also provides the functionality to dynamically update the leakage power as temperature changes.

Temperature variation has become a major challenge to future HPC architecture design. It has an exponential relationship with leakage which can cause further exacerbation of power [11]. This problem is becoming more and more critical with the continuous scaling of technology [8]. In this study, we illustrate the utilization of the modeling framework to study the performance and efficiency tradeoffs in clustering of manycore systems with consideration of temperature variation. The communication pattern generator and the router model SST components are used to model the manycore systems and provide component usage counts to the technology interface. The estimated power and performance of the on-chip network (NoC) are gathered by the introspection interface for further evaluation.

The remainder of this paper is organized as follows. In Section, we discuss prior related works. In Section 3, we describe the structure of SST and its key interfaces for power/area/thermal modeling and statistics gathering. Section 4 gives more details on the technology models currently supported by the framework. In Section 5, we use the framework to explore the interconnect options of future manycore processors by varying the degree of clustering and examine the impact of temperature variation on this. The paper is concluded in Section 6.

2. RELATED WORK

There has been much research in the area of performance and power modeling of HPC architectures. Traditional single-core simulators were designed to model out-of-order superscalar processor pipelines. For instance, SimpleScalar [4] was widely used to evaluate uniprocessor systems. It is extended with several power models, such as Wattch [3] and Sim-Panalyzer [18], for examining performance/power trade-offs. However, it runs only user-mode single-threaded workloads, let alone the simulation of multiple processor cores. Full-system simulators are frequently used to evaluate manycore architectures. For example, M5 simulator [2] supports the execution of operating system as well as application code, and is capable of modeling I/O subsystems and multiple networked systems. It has been used together with power models on study of processor lifetime of chip multiprocessors [5]. However, it only supports a shared-bus model to simulate interconnection of manycore processors and the integration with the power model is not available in the standard distribution. GEMS models detailed aspects of cores, cache hierarchy, cache coherence and memory controllers [10] and has an improved network model, GARNET [1]. However, it can only model power dissipation of the network components, not the entire system. The Polaris toolchain provides rapid estimations of power/area/delay of large NoC design space and its power model has been extended to consider temperature variation and leakage feedback [8]. However, it is lack of detailed power modeling of the computation and storage components on the chip. The integrated power, area and timing modeling framework, McPAT, has been used together with a manycore performance simulator to explore the interconnect options of future manycore processors, but the simulator is not publicly available.

3. STRUCTURAL SIMULATION TOOLKIT (SST)

SST is based on a fully modular design and provides a parallel simulation environment using MPI. This enables SST to extensively explore parameters of an individual system without the need to intrusively change the simulator. It also provides a high level of performance and the capability to look at large systems.

3.1 Modeling Framework

The most important class in SST is *Component*, the base class from which all simulation components (e.g. core, router, memory, etc) inherit. *Components* are connected by *Link* to communicate with each other and are partitioned among all ranks to ensure balanced workload and scalability of the simulator.

Components can query the SST's technology interface for power and temperature estimates, and in some cases, chip area and timing information. The *Components* can set technology parameters (such as clock frequency and supply voltage), choose the estimation library it wishes to use, and set architecture parameters like cache size or the number of register file ports. Once the estimates are computed the SST provides *Components* a uniform interface (the introspection interface) for reporting power consumption and gathering statistics.

3.2 Communication Pattern Generation and the Router Model

The communication pattern generator SST component is implemented as a state machine. It simulates compute time by suspending operations until a future event indicates the passing time and the need to transition to another state in the state machine. The communication pattern components use dimension-ordered routing in the NoC. The only communication pattern implemented at the moment is ghost which simulates ghost cell exchanges on a five-point stencil operator where each rank communicates only with its East, West, South, and North neighbor. Implementations of communication patterns for FFT [22], the NAS parallel benchmark integer sort (IS) [12], and master/slave [7] are under way.

The router model SST component allows for arbitrary topologies. Messages are wormhole routed and use source-based routing. If a path from an input port to an output port is available, the message is forwarded without further delays. The router model also models congestion in the network when the input or the output port is busy. The router model maintains a small number of counters to enable statistics on the number of messages coming in and going out of each port, how often congestion occurred and how much delay that caused.

3.3 Technology Interface

The technology interface is the core of power and thermal simulation. It integrates various power and thermal libraries (e.g., McPAT, IntSim, ORION power library, and HotSpot), computes run-time energy dissipation, and stores the data traces. Its implementation is independent of the front-end simulator such that the statistical timing model is replaceable.

3.3.1 The Design

The technology interface is initialized in two folds; chip-level and component-level setups. Chip-level initialization creates silicon layer floor plans and links the simulation chip with a thermal library. The floor plan may include multiple blocks, and the blocks are assumed to be running with the same clock frequency and voltages, and more importantly within the uniform temperature plane. Therefore, the size of the floor plan determines the modeling granularity of the simulation and can vary from a single block to a SST component (e.g., core or main memory). After floor plans are set up, thermal tiles are created for silicon, interface, spreader, and heatsink layers. Thermal tiles are the same size as underlying floor plans but do not need to be on the same layer - only silicon-layer thermal tiles are duplicated in terms of xyz coordination.

Chip-level physical parameters include thermal design parameters, floor plan parameters (e.g. sizing and coordinate information), and technology parameters for each floor plan. Thermal design parameters include chip designs (e.g., chip thickness, thermal RC constants) and the initial temperatures for each thermal tile. Technology parameters include device-level parameters as well as higher-level parameters such as temperature, clock frequency, and supply voltage.

The component-level setup reads architectural parameters of each component from System Description Language (SDL) file and runs the power model to compute the basic information such as dynamic energy per access, leakage energy, and area. Component parameters include core/uncore-level microarchitectural parameters such as number of branch predictors, cache hierarchy, pipeline design, etc.

3.3.2 Runtime Power and Temperature Estimation with Leakage Feedback Control

The technology interface specifies the power into five categories: dynamic, short circuit (or switching), gate leakage, subthreshold leakage, and peak power. The technology interface gets usage counts of each component at user-specified period or condition. It then uses the well-known Wattch method [3] to calculate the dynamic energy by multiplying these count values with dynamic energy per access statically calculated by power models. The leakage power consumption is calculated by power models at an initial temperature. The estimated power is then stored in a central power database which can be accessed for later analysis.

After power dissipation for each component is calculated, the power values for the underlying floor plans are correspondingly updated. Once the power calculation is done for all components in the simulated chip, the technology interface triggers temperature calculation using the thermal library. The thermal library takes instantaneous power values for each floor plan. The new temperature of each floor plan updated by the thermal library is fed back to the blocks grouped within the floor plan. This procedure is called leakage feedback, and the new leakage power is calculated based on the new temperature. These power changes will again affect temperature profile [21]. After several iterations, the difference between each loop will be small enough and power and thermal profile converge [8].

3.4 Introspection Interface

The introspection interface is a unified way to report and record simulation data for analysis and display. It provides a standard method of retrieving statistics so that external programs can access the simulator statistics without requiring knowledge of simulator structures. The SST

Introspector class inherits from the *Component* class and can be created to monitor information from all, or a sub set of other real components. Like *Components*, *Introspectors* are created and parameterized by the SDL specifying which components they wish to monitor and how frequent to query these components to retrieve components state (e.g. power). Unlike *Components*, *Introspectors* are not partitioned and have a copy on every rank. *Introspectors* can exchange components data with introspectors on other ranks via MPI collective calls. Via these collective communications, *Introspectors* can gather information like highest core temperature in the system, and summation of power dissipation of all components, etc.

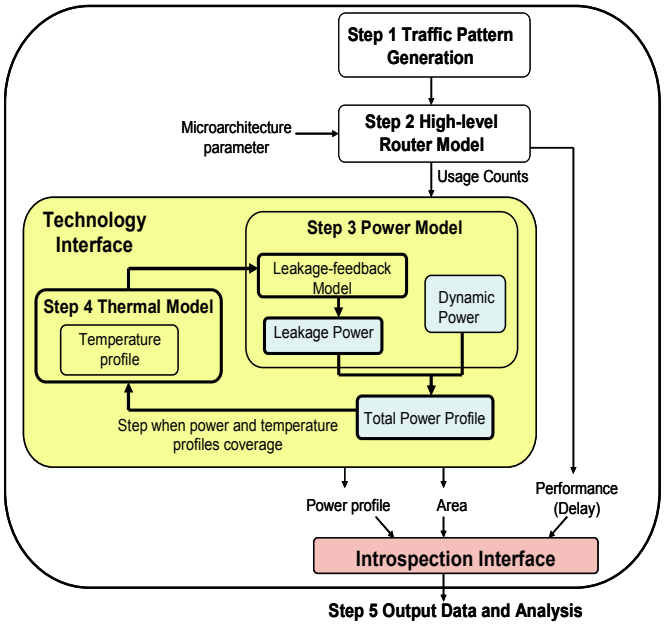

Figure 1. Flow chart of NoC power modeling with consideration of leakage feedback

Figure 1 presents the flow chart of NoC power modeling with consideration of leakage feedback using the modeling framework. The traffic pattern generator component generates a traffic trace which is then fed into the router model component to capture the usage count of the router and calculate the network delay. The usage count information is then fed into a power library (McPAT or ORION in this study) to calculate the power consumption with consideration of leakage feedback and temperature variations. At this step, within each iteration leakage power is calculated with the simulated temperature until the power and thermal profile converged. The power library also calculates the area for the NoC. Finally, the introspection interface gathers simulation results including NoC power dissipation, temperature, delay, and area from all the router model components. It then uses several metrics to explore interconnect options in manycore processors.

PMBS'10, November 15, 2010, New Orleans, Louisiana, USA.

4. TECHNOLOGY MODELS

The following power, area and thermal models are currently integrated with the SST technology interface. They support power/area/thermal analysis of five component types (core, clock, shared cache, memory controller and NoC/router).

4.1 HotSpot

HotSpot is a thermal simulation tool used for estimating chip-wide temperature based on an architectural floor plan and simulated power measurements [20]. The SST technology interface includes the transient thermal simulation kernel of HotSpot, and provides HotSpot with dynamic values for power density and the present temperature of each block. In this paper, we use power densities obtained from McPAT. HotSpot then solves a series of differential equations and returns the new temperature of each block. However, the temperature analysis does not support the case in which power dissipation is dependent on the temperature, which is the situation with leakage [21]. To solve this problem we extended the thermal analysis such that the power consumption at a time step is calculated as the sum of two components: (1) the dynamic power return from the power modeling library (2) the leakage power calculated at the simulated temperature of the previous step.

4.2 McPAT

McPAT is an integrated power, area, and timing modeling framework for multithreaded and multicore and manycore processors. It includes power, area and timing models for the processor cores, NoC, shared caches, memory controllers, and clocking. Its power models account for dynamic, subthreshold leakage, gate leakage, and short-circuit power. McPAT was validated against Niagara, Niagara2, Alpha 21364, and the Xeon Tulsa processors. The difference between its models and the reported data was between 10-20%.

The original McPAT work [9] assumes uniform temperature within the chip and no leakage feedback. We extended this tool to calculate power consumption with consideration of leakage feedback and temperature variation. Leakage feedback is processed in three steps; updating temperature-dependent technology parameters,

reconfiguring McPAT, and retrieving the new leakage power.

4.3 IntSim

IntSim is an interconnect-centric CAD tool that optimizes die size and pitches of different wiring levels for circuit blocks or logic cores. It includes a methodology for co-optimization of signal, power, and clock interconnects and stochastic wiring distribution [16, 17]. The primary IntSim algorithm is an iterative loop that repeatedly calculates the dynamic and leakage power until the presumed power and the calculated power values converge. This model is useful to evaluate the power of logical blocks such as functional units in the microarchitecture.

4.4 ORION

ORION is a power and area library for different router configurations which projects potential circuit structures for each configuration at each technology node [19]. ORION was validated to within 7% and 11% error of the Intel 80-core and SCC chips, respectively. It takes SST router model resource utilization information for each router and link as input activity and returns network dynamic and leakage power.

5. PERFORMACE ANALYSIS OF VARIOUS INTERCONNECT OPTIONS OF MANYCORE PROCESSORS

We illustrate the utilization of the integrated modeling framework by applying it to evaluate different interconnect options of manycore processors with consideration of leakage feedback and within-chip temperature variation. We study the power, the energy-delay product (EDP) and the energy-delay-area product (EDAP) for four manycore system configurations. We also examine how temperature variation impacts these metric values.

5.1 Experimental Setup

We consider a manycore architecture used in McPAT's work for our baseline system and analysis. The manycore architecture consists of multiple clusters connected by a 2D-mesh on-chip network. A cluster has one or more cores and there is one communication pattern component on each core. Routers in the network have local ports that connect to a cluster of cores as well as ports that connect to the neighboring routers. The cores attached to the same router in the NoC are assumed to have a shared L2 cache. Messages between them carry a flag indicating that they should be handled at a lower latency and higher bandwidth by the router model.

We adopt parameters from McPAT and our experiments assume 64-core NoC designs and 22nm processing technology. Our study considers four configurations: 1 core, 2 cores, 4 cores and 8 cores per cluster. Simulations are run with a time step of 150000 ns and the NoC power is estimated every 100 time steps.

5.2 Comparison of Router Power Estimated by McPAT and ORION

The two power models, McPAT and ORION, supported by the modeling framework are popularly used in many power-related architecture research projects. First, we want to compare them in terms of NoC power modeling. We use the parameters in Table 1 to model router power dissipation by both McPAT and ORION and the result is shown in Figure 2.

Table 1. Parameter Values

Technology	65 nm
Clock frequency	1.0 GHz
Supply voltage	1.2 V
Number of input ports	8
Number of output ports	8
Flit bits	128
Virtual Channels per port	2

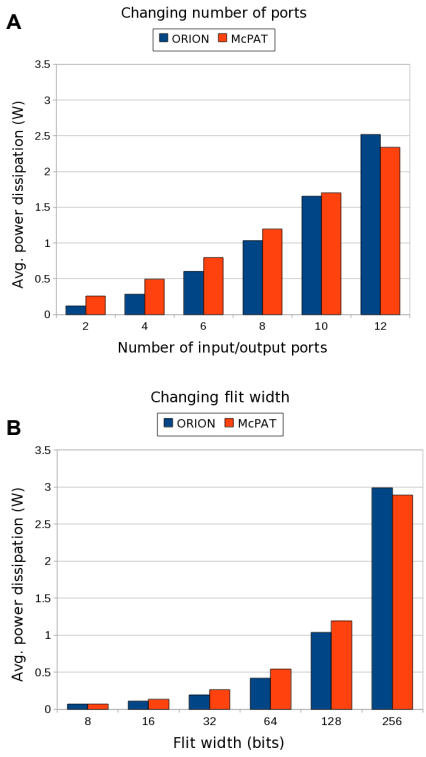

Figure 2. Comparison of McPAT and ORION in estimation of router power

In Figure 2A, the number of ports is varied from 2 to 12 while the rest of the parameters are kept the same. In Figure

2B, the flit width is varied from 8 bits to 256 bits. The figures show that the discrepancy between McPAT's and ORION's power estimations is small (an average of 10%), and they both grow at about the same rate when the number of ports and flit bits are varied.

Each of the two power modeling tools has its advantage over the other. For example, McPAT also provides power models for core, memory controller, and shared cache. One can simply call McPAT to model the overall power dissipation of manycore systems. On the other hand, ORION's NoC parameters are much more customizable than McPAT's and is more flexible to model various router architectures. In addition, ORION is a modeling tool specifically for NoC so it takes less time to initialize while McPAT may have a huge design space to exhaust search for the best value power and area deviation.

5.3 Performance and Efficiency Tradeoff in Clustering

Figure 3 shows the total power of the 4 configurations (1-core, 2-core, 4-core and 8-core per cluster) normalized by the value of the 4-core per cluster configuration. In these simulations, we use McPAT to model the NoC power dissipation with neither leakage feedback nor temperature variation. In general, clustering more cores together reduce the system total power except at the 8-core per cluster configuration.

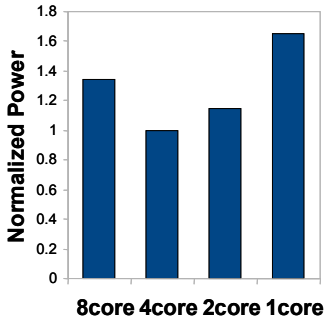

Figure 3. Normalized power of the manycore systems

We use two metrics to evaluate the performance and efficiency tradeoffs in clustering. The energy-delay-area product (EDAP) is of particular interest because the metric includes both an operational cost element (energy) and a capital cost element (area) [9]. Figure 4 shows EDAP and EDP of the 4 system configurations normalized by the values of the 4-core per cluster configuration. Figure 4A shows that clustering using 4 cores gives the best EDAP. This is consistent with McPAT's conclusion that the 4-core per cluster configuration has the best EDAP on all benchmark suites on average [9]. Figure 4B shows the 4-core per cluster configuration also has the best EDP.

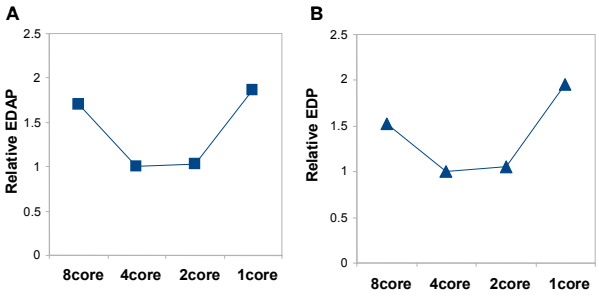

Figure 4. Normalized EDP and EDAP

It has been shown that on average, increasing cluster size improves the system EDP and the effects of clustering on the metric values depend heavily on applications [9]. In our study, the 8-core per cluster configuration has the worse EDP than the 4-core per cluster design. This is because of the characteristics of the communication pattern component we use. In this study, we assume the cores in the same cluster have a shared L2 cache. Figure 5 shows that the communication pattern results in similar numbers of local communications (intra-cluster communications) for the 4- and the 8-core per cluster configurations. Therefore, clustering 8 cores together does not take more benefit from cache sharing comparing to the 4-core per cluster design. On the other hand, the 2-core per cluster design has about 50% less local communications and thus consumes more power in routing messages.

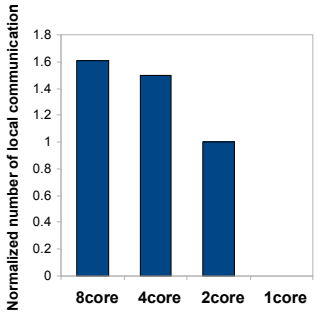

Figure 5. Normalized number of intra-cluster communication

5.4 Effect of Temperature Variation and Leakage Feedback on Performance

In the previous section we simulated the network on chip with a single uniform temperature and no leakage feedback. We now consider temperature variation and leakage feedback in the model and examine how these affect the metric values for the 4 system configurations.

Figure 6 shows the total power consumption of the 4 configurations normalized to the lowest–power NoC configuration. The blue bars indicate estimated NoC power

with no leakage feedback or temperature variation while the yellow bars show the estimated power taking both into consideration. The figure shows that when both leakage feedback and temperature variation are considered, the power consumption for each configuration increases (by about 10%) compared to the no variation case. Besides, the relative power ranking among the 4 configurations remains the same as the one with no variation.

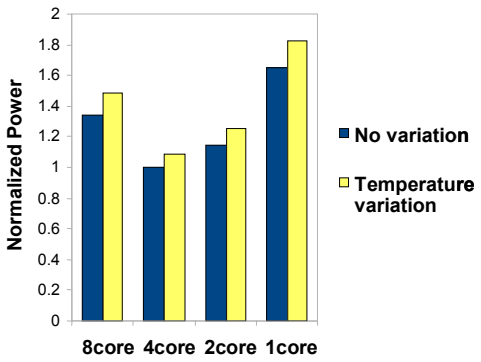

Figure 6. Normalized power with consideration of temperature variation

Next, we examine the impact of considering both leakage feedback and temperature variation on EDAP and EDP. Figure 7 shows the normalized EDAP and EDP of the four configurations with (red line) and without variance (blue line). We can see that considering temperature variation does not change the relative ranking of EDAP or EDP among the four configurations. The figure gives the same conclusion that the 4-core per cluster configuration has the best EDAP and EDP. However, it is still important to consider temperature variation in NoC power modeling because it can introduce substantial differences in total power dissipation. Even though the relative power ranking does not change, a design may have a power constraint that needs to be met. Therefore, total power should be accurately estimated in order to make a correct design decision.

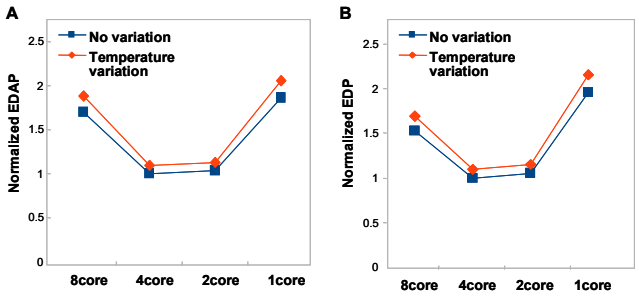

Figure 7. Normalized EDP and EDAP with consideration of temperature variation

6. CONCLUSION

In this work, we implemented a framework for integrated power, area and thermal simulation and data gathering. The framework includes a number of power and thermal models and contains leakage feedback functionality. We illustrated the utilization of the framework on studying performance and efficiency tradeoffs in clustering of manycore processors with consideration of within-chip temperature variation and leakage feedback. Integration of this power, area and temperature estimation capability to the Structural Simulation Toolkit (SST) for HPC Architectural Simulations makes it possible to use SST for Hardware/Software co-design of future Exascale systems.

We first compare the two power models, McPAT and ORION, in NoC power estimation. Our result suggests that there is small discrepancy between the two in terms of power dissipation. We then use a manycore architecture in McPAT's work for our baseline system and analysis. Instead of using a front-end full simulator to model a core like what McPAT does, our communication pattern component of SST allows the generation of network traffic without incurring the processing and memory overhead. We arrive at the same conclusion as McPAT's that configuring a cluster with 4 cores has the best energy-delay-area product. When temperature variation and leakage feedback are taken into consideration, the conclusion remains the same. Even so, considering temperature variation in power estimation is still important because it substantially increases the power dissipation which can change a designer's decision in choosing a network configuration. Moreover, it has been shown that the amount of performance gained by the use of a manycore processor depends very much on the application. When more communication pattern components are implemented, we expect that our simulation results would demonstrate the impact of communication patterns on power, EDAP and EDP in more detail.

7. ACKNOWLEDGMENTS

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

8. REFERENCES

[1] Agarwal, N., Krishna, T., Peh, L., and Jha, K. N., 2009. GARNET: A Detailed On-Chip Network Model inside a Full-System Simulator. In *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software* (Boston, Massachusetts, USA, April 26-28, 2009). ISPASS'09, IEEE Press, Boston, Massachusetts, 33-42.

[2] Binkert, N. L., Dreslinski, R. G., Hsu, L. R., Lim, K. T., Saidi, A. G., and Reinhardt, S. K. 2006. The M5 Simulator: Modeling Networked Systems. *IEEE Micro* 26, 4 (Jul. 2006), 52-60. DOI= http://dx.doi.org/10.1109/MM.2006.82

[3] Brooks, D., Tiwari, V., and Martonosi, M. 2000. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th Annual international Symposium on Computer Architecture* (Vancouver, British Columbia, Canada). ISCA '00. ACM, New York, NY, 83-94. DOI= http://doi.acm.org/10.1145/339647.339657

[4] Burger, D. C., and Austin, T. M. 1997. The SimpleScalar Tool Set. Technical Report CS-TR-1997-1342, University of Wisconsin, Jun. 1997.

[5] Coskun, A. K., Strong, R., Tullsen, D. M., and Simunic Rosing, T. 2009. Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors. In *Proceedings of the Eleventh international Joint Conference on Measurement and Modeling of Computer Systems* (Seattle, WA, USA, June 15 - 19, 2009). SIGMETRICS '09. ACM, New York, NY, 169-180. DOI= http://doi.acm.org/10.1145/1555349.1555369

[6] Energy Information Administration (EIA), Average Retail Price of Electricity to Ultimate Customers http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_b.html, September 2009.

[7] Leary, G., Mehta, K., and Chatha, K. S. 2007. Performance and resource optimization of NoC router architecture for master and slave IP cores. In *Proceedings of the 5th IEEE/ACM international Conference on Hardware/Software Codesign and System Synthesis* (Salzburg, Austria, September 30 - October 03, 2007). CODES+ISSS '07. ACM, New York, NY, 155-160. DOI= http://doi.acm.org/10.1145/1289816.1289856

[8] Li, B., Peh, L., and Patra, P. 2008. Impact of Process and Temperature Variations on Network-on-Chip Design Exploration. In *Proceedings of the Second ACM/IEEE international Symposium on Networks-on-Chip* (April 07 - 10, 2008). NOCS. IEEE Computer Society, Washington, DC, 117-126.

[9] Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. 2009. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM international Symposium on Microarchitecture* (New York, New York, December 12 - 16, 2009). MICRO 42. ACM, New York, NY, 469-480. DOI= http://doi.acm.org/10.1145/1669112.1669172

[10] Martin, M. M., Sorin, D. J., Beckmann, B. M., Marty, M. R., Xu, M., Alameldeen, A. R., Moore, K. E., Hill, M. D., and Wood, D. A. 2005. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. *SIGARCH Comput. Archit. News* 33, 4 (Nov. 2005), 92-99. DOI= http://doi.acm.org/10.1145/1105734.1105747

[11] Mohanty, S., Ranganathan, N., Kougianos, E., and Patra, P. 2008 *Low-Power High-Level Synthesis for Nanoscale CMOS Circuits*. 1. Springer Publishing Company, Incorporated.

[12] Prakash, S., Deelman, E., and Bagrodia, R. 2000. Asynchronous Parallel Simulation of Parallel Programs. *IEEE Trans. Softw. Eng.* 26, 5 (May. 2000), 385-400. DOI= http://dx.doi.org/10.1109/32.846297

[13] Register 2009. http://www.theregister.co.uk/2009/05/26/new_apple_data_center/

[14] Rodrigues, A. F. 2007. The structural simulation toolkit, http:// www.cs.sandia.gov/sst

[15] Scaramella, J. 2010. Worldwide Server Power and Cooling Expense 2006-2010 Forecast. http://www.sun.com/service/eco/IDCWorldwideServerPowerConsumption.pdf

[16] Sekar, D. 2008. *Optimal Signal, Power, Clock, and Thermal Interconnect Networks for High-Performance 2D and 3D Integrated Circuits.* Ph. D. Dissertation, School of ECE, Georgia Tech, Dec. 2008.

[17] Sekar, D. C., Naeemi, A., Sarvari, R., Davis, J. A., and Meindl, J. D. 2007. IntSim: A CAD tool for optimization of multilevel interconnect networks. In *Proceedings of the 2007 IEEE/ACM international Conference on Computer-Aided Design* (San Jose, California, November 05 - 08, 2007). International Conference on Computer Aided Design. IEEE Press, Piscataway, NJ, 560-567.

[18] Sim-Panalyzer library 2003. http://www.eecs.umich.edu/~panalyzer/

[19] Singh, S., Mayfield, C., Mittal, S., Prabhakar, S., Hambrusch, S., and Shah, R. 2008. Orion 2.0: native support for uncertain data. In *Proceedings of the 2008 ACM SIGMOD international Conference on Management of Data* (Vancouver, Canada, June 09 - 12, 2008). SIGMOD '08. ACM, New York, NY, 1239-1242. DOI= http://doi.acm.org/10.1145/1376616.1376744

[20] Skadron, K., Stan, M. R., Sankaranarayanan, K., Huang, W., Velusamy, S., and Tarjan, D. 2004. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.* 1, 1 (Mar. 2004), 94-125. DOI= http://doi.acm.org/10.1145/980152.980157 hotspot

[21] Su, H., Liu, F., Devgan, A., Acar, E., and Nassif, S. 2003. Full chip leakage estimation considering power supply and temperature variations. In *Proceedings of the 2003 international Symposium on Low Power Electronics and Design* (Seoul, Korea, August 25 - 27, 2003). ISLPED '03. ACM, New York, NY, 78-83. DOI= http://doi.acm.org/10.1145/871506.871529

[22] Swarztrauber, P. N., Sweet, R. A., Briggs, W. L., Henson, V. E., and Otto, J. 1991. Paper: Bluestein's FFT for arbitrary N on the hypercube. *Parallel Comput.* 17, 6-7 (Sep. 1991), 607-617. DOI= http://dx.doi.org/10.1016/S0167-8191(05)80051-1