A Scalable Simulation Framework for Evaluating Thermal Management Techniques and the Lifetime Reliability of Multithreaded Multicore Systems

Ming-yu Hsieh
Scalable Computer Architectures Organization
Sandia National Laboratories
Albuquerque, NM
myhsieh@sandia.gov

Abstract—It has become increasingly challenging to understand supercomputers behavior and performance as they grow. New hurdles in scalability, programmability, power consumption, reliability, cost, and cooling are emerging. This paper introduces the integrated power, area, temperature, reliability modeling framework in the open, modular, multiscale, parallel Structural Simulation Toolkit (SST) to help evaluate new technologies and guide design of future computers. In this study, the simulation framework is used to evaluate different dynamic thermal management techniques, in terms of power, temperature and reliability, on multicore systems running multithreaded and more irregular applications. Simulation results shed some new light on application-aware, performance/power efficient thermal and reliability management policies of multithreaded multicore systems.

Keywords-reliability; power management; multicore system; simulation

I. INTRODUCTION
The advancement of a range of scientific and technical challenges ranging from basic science to climate prediction to weapons design requires development of large, capability-class super computers. The task of building these computers is becoming increasingly difficult as the High Performance Computing (HPC) community reaches towards exascale. In addition to the traditional challenges of raw performance and scaling, fundamental challenges in performance, power consumption [1], cost, reliability [2], programmability [3], and cooling arise. Simulations can guide design, program and operate future computers.

Currently, a variety of simulators exist for individual system components, but no unified framework allows them to act in concert. The Structural Simulation Toolkit (SST) [4] aims to address this problem. The SST couples parameterizable models (components) for processors, memory, and network subsystems. All these components have access to a uniform interface to a variety of power/thermal/reliability modeling libraries. The SST allows parallel simulation of large machines at scale to understand their performance, power consumption, temperature and reliability.

This paper introduces the technology interface in SST, the core of integrated power, temperature and reliability simulation. It receives usage counts from SST components and calculates the power dissipation, temperature change and mean-time-to-failure (MTTF). In the current implementation, the interface includes the McPAT [5], IntSim [6], ORION [7] power modeling libraries, the HotSpot [8] thermal modeling library and the reliability models from [9].

Advancements in technology enable integration of a large number of cores on a single silicon die to enable the scaling of performance. Thermal effects on multicore (CMP) systems are still prominent issues. One typical thermal effect is the thermal-aware lifetime reliability, which has become a serious concern. In this study, we illustrate the utilization of the modeling framework by applying it to study the effects of different dynamic thermal management techniques on lifetime reliability of CMP systems running multithreaded and more irregular (graph-based) applications.

Graph-based applications are data-intensive and are believed to dominate high performance computing over the next decade [10]. These kinds of applications include cyber security, medical informatics, data enrichment, social networks, and symbolic networks. Managing performance, energy, and reliability in the presence of temperature and load imbalance can lead to interesting and non-intuitive effects. This will become increasingly true as applications become more irregular (e.g. graph-based codes with hotspots) and architectures more flexible (e.g. migration threads into memory system, between processors, or across the network). We show that having a fully integrated model, including a reliability model that accounts for all the major causes of temperature-induced hard failures, sheds some new light on power-efficient, thermal-aware reliability management of multithreaded multicore (CMT) systems.

The main contributions of this work are:

• We extend the technology interface in SST with the reliability model. The integrated simulation framework can guide design future thermal management techniques for CMT systems.

U.S. Government work not protected by U.S. copyright
We present the parallelization techniques of power, feedback leakage, and thermal estimations with McPAT and HotSPot. This allows SST to guide design and optimize the supercomputers of tomorrow using the supercomputers of today.

To the best of our knowledge, this is the first work to evaluate different thermal management techniques on CMT systems running irregular and parallel applications where threads are dependent to each other.

The rest of this paper is organized as follows. Section 2 discusses prior related works. Section 3 describes the scalable and parallel design of SST and its key interfaces for power/thermal/reliability modeling. In Section 4 we use the framework to evaluate different dynamic thermal management techniques on CMP systems running parallel and irregular applications. The paper is concluded in Section 5.

II. RELATED WORK

There has been much research in the area of computer architectural performance and power modeling. The M5 simulator [11] 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 [9]. 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 [12] and has an improved network model, GARNET [13]. However, it can only model power dissipation of the network components, not the entire system. Recently, Lis et al. present HORNET, a parallel, highly-configurable, cycle-level manycore simulator with support for power and thermal modeling [14]. However, it does not have a modular design to allow integration of other architectural models not shipped with the package. Moreover, it uses ORION [7] for power estimation which only provides power modeling for network components. In general, most simulation efforts focus on a given piece of the system. As noted in [15], readily available simulators tend to lack the ability to pull together high fidelity simulations of all system components into a single framework. The modular design of SST eases integration of existing simulators to a parallel, scalable, and open-source framework.

Reliability of CMP systems is another area of increasing concern. A good summary of research contributions that combine performance and reliability measures is given in [16]. Srinivasan et al. [17] presented an architectural-level model to evaluate processors lifetime reliability. In [9], Coskun et al. proposed a novel CMP simulation framework to simulate thermal dynamics over far longer time periods to show how job scheduling and power management policies affect lifetime of systems running with single-threaded applications. An analytical model for the lifetime reliability of homogeneous manycore systems was proposed in [18]. To our best knowledge, this work is the first to evaluate different thermal management techniques, in terms of power, performance, and reliability, on CMP systems running multithreaded and graph-based applications.

III. THE MODELING FRAMEWORK

Effective supercomputer design and evaluation requires a simulation environment for quickly simulating large HPC systems in a variety of ways. This kind of simulation framework needs to meet some requirements, including scalable, parallel and multiscale simulation. The SST simulation framework allows parallel simulation of large (tens to hundreds of thousands of nodes or more) machines at multiple levels of detail (from cycle-accurate execution-driven instruction-based to abstract message-driven simulation).

The SST is comprised of a simple simulation core that contains a parallel discrete event simulator and supports services for simulation. Components, representing hardware systems such as processors, network switches, or memory devices, interface with the simulation core to communicate and operate with a common notion of timeframe. The simulator core provides simulation configuration and startup, the parallel model of computation, checkpointing and common interfaces to technology models and statistics gathering. The technology interface is integrated with several technology models supporting power, temperature, area, and reliability estimation for components. The introspection interface provides a standard method of retrieving statistics so that external programs can access the simulation results without requiring knowledge of simulator structures.

The most important class in SST is Component, the base class from which all simulation components 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.

A. The Technology Interface

The technology interface is the core of power, temperature and reliability simulation. It is currently integrated with various power, thermal and reliability modeling libraries to compute run-time energy dissipation, temperature and MTTF, and stores these analyses in a central database.

The technology interface gets usage counts of each SST component at user-specified period or condition. It then uses the well-known Watch method [20] to calculate the dynamic energy by multiplying these count values with dynamic energy per access statically calculated by power models. 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 to calculate the new temperature which is then fed back to the power model. 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]. Both the instantaneous power and temperature values are stored in the central database. When a component requests reliability modeling, the interface feeds the temperature values from the database to the reliability model. This on-the-fly power estimation, to thermal model for temperature calculation and to...
reliability models for reliability analysis enables not only the usual average and peak value reporting of power/temperature/reliability for the entire chip but also percentile and per-time-period reporting.

The technology interface currently adopts the reliability model defined in [9]. It models the most commonly studied temperature-induced intrinsic hard failure mechanisms, which are electromigration (EM), time dependent dielectric breakdown (TDDB), and thermal cycling (TC) [22, 23]. The EM failure rate ($\lambda_{EM}$), based on Black’s model [24], is given in (1).

$$\lambda_{EM} = C_{EM} e^{(-E_a/kT)}$$  \hspace{1cm} (1)

In this equation, $C_{EM}$ is a constant (an average technology and circuit dependent value), $E_a$ is the activation energy, $k$ is the Boltzmann’s constant, and $T$ is the temperature. In our experiment, $E_a = 0.7$ and $C_{EM} = 7.39125 \times 10^{-10}$. The TDDB failure rate is defined in (2).

$$\lambda_{TDDB} = C_{TDDB} e^{(-E_b/kT)}$$  \hspace{1cm} (2)

Similar to the EM failure equation, $C_{TDDB}$ is a constant and $E_b$ is the activation energy. In this work, $E_b = 0.75$ and $C_{TDDB} = 1.36334 \times 10^{-10}$. The TC failure rate model is based on the Coffin-Manson equation and is formulated as in (3).

$$\lambda_{TC} = C_{TC} \Delta T^{-q} f$$  \hspace{1cm} (3)

In the equation, $C_{TC}$ is a material dependent constant, $\Delta T$ is the temperature cycling range, $q$ is the Coffin-Manson exponent, and $f$ is the frequency of thermal cycles. In this study, $C_{TC} = 1.52 \times 10^{-5}$ and $q = 4$. The frequency, $f$, is determined by using the method in [9]. We use the sum-of-failure-rates model [25] to calculate the system level reliability. The model assumes the core is a series failure system and as such can be represented as the sum of failure rates of individual failure mechanisms which are independent from each other. Therefore the average MTTF is computed by $1/\lambda$, where $\lambda$ is the average failure rate observed throughout the simulation.

Fig. 1 presents the flow chart of power, thermal and reliability modeling with consideration of leakage feedback using SST. First, the SST component feeds usage count information into a power library (McPAT in this study) through the technology interface to calculate the power consumption with consideration of leakage feedback. The power values are stored in the central database of the technology interface. If the component requests thermal modeling, the interface feeds the power values from the database to the thermal model to calculate temperature. These new temperatures are also stored which can be further fed into the reliability model when the component requests to compute MTTF. Finally, the introspection interface gathers simulation results including power dissipation, temperature, MTTF, delay and area from the component for further analysis.

B. Scalable Parallel Simulation

The SST uses a parallel component-based discrete event simulation (DES) model layered on top of Message Passing Interface (MPI). Parallelism is transparent to the component writer. To achieve better performance, the SST uses a conservative (i.e. no rollback) distance-based [26] optimization. At the start of the simulation, the system topology is represented by a graph with components as nodes and links between them as edges, with each edge labeled with the minimum latency between the connected components. The Zoltan [27] library is then used to partition components across the MPI tasks (ranks) with the goal of balancing the load and partitioning across the highest latency links.

Figure 1. Flow chart of power, thermal and reliability modeling with consideration of leakage feedback

When a simulation is partitioned, each MPI rank has both a power library instance (e.g. McPAT) and a thermal library instance (e.g. HotSpot), that hold partial power information of the entire system. That is, each MPI rank only knows the power consumption of the components that are assigned to it. To calculate the temperature of the entire system, a HotSpot instance needs to know the power consumption of all components, including the ones on its own MPI rank and the ones on other MPI ranks. We accomplish this by parallelizing the technology interface to HotSpot and use MPI Allreduce to make the power values stored by each MPI rank available to all MPI ranks (illustrated in Fig. 2). In the example below, there are four components in the simulated chip, where Router$_0$ and CPU$_0$ are in floorplan$_0$ and Router$_1$ and CPU$_1$ are in floorplan$_1$. After partitioning, Router$_0$ and CPU$_1$ are on MPI rank$_0$ (blue)
IV. POWER AND RELIABILITY MANAGEMENT OF MULTITHREADED MULTI-CORE SYSTEMS

We illustrate the utilization of the integrated modeling framework by applying it to evaluate different thermal management techniques of CMT systems. We study the effects of Dynamic Power Management (DPM) and Dynamic Voltage/Frequency Scaling (DVFS) on system power, temperature, MTTF and the energy-delay product (EDP). We also examine the need of novel management techniques to address the particular characteristics of graph-based applications.

A. Experimental Setup

We develop a CMT component linked with a memory component in SST to model a 16-core multiprocessor connected to a memory system. The CMT component models cores, caches, directories, on-chip network and memory controller with a supply voltage of 1.2 V and a base frequency of 2 GHz. For the dynamic voltage/frequency scaling, we vary the supply voltage from 1.2 V (100%) to 1.187 V (95%) and 1.16 V (85%), and vary the processor frequency from 2 GHz (100%) to 1.9 GHz (95%) and 1.7 GHz (85%). We run selected benchmarks from PARSEC [28] and graph-based benchmark suite MTGL [29] on the simulator along with McPAT. HotSpot and the reliability models provided by the technology interface to study power management of CMT systems running parallel and irregular applications. We fast-forwarded PARSEC and MTGL applications to the beginning of the parallel execution, and then simulate 1.5 billion cycles. Simulation results, such as power, temperature, and MTTF, are gathered every 10 ms. Results shown in the figures are normalized with respect to the default case, where there is no power management.

One of the power management techniques we investigate is Dynamic Power Management (DPM). For each core, the DPM waits for a timeout period when the core is idle, and then turns off the core to save energy. We assume the timeout period to be 50 ms (Section IV.B), the sleep state power value to be 0.05 W [9]. When a core switches state between sleep and active, there are penalties such as transition power of 10W and wakeup delay of 25 ms [9]. This is to ensure we do not turn off cores for very short idle times. The simulation parameter values are summarized in Table I.

B. Effect of Dynamic Power Management (DPM)

Power management is a feature of electrical devices, which attempts to turn off or place certain or all portions of the device in some lower-power state when inactivity is present. In the experiment, we assume each core has two power states, active and sleep. Let r be the dynamic power rate when a core is active, B be the transition power, T be the sleep cycles and s be the sleep state power value. It is straightforward that DPM can reduce processor power if

$$rT > B + sT$$  \hspace{1cm} (4)

From simulations we observe that when running PARSEC applications on the 16-core multiprocessor simulator, about 12 cores are idle at a time and an idle core becomes active at an average of every 80 ms. The effect of DPM on the system power, performance, MTTF and EDP is shown in Fig. 3, which shows these effects are adverse. This is because the characteristics of the PARSEC applications. In simulations, the DPM puts a core to sleep when it has been idle for at least 50 ms, which results in a core becomes active soon after it is put to sleep. Therefore, the sleep cycles (T) is short and the advantage of DPM putting idle cores in low-power state is not taken.

Equation (4) suggests that both transition power (B) and sleep cycles (T) play main roles on the effectiveness of DPM. We run another set of simulations with lower transition power of 5 W and results are shown in Fig. 4. Fig. 4 shows that DPM reduce system power by 20% at average. This is trivial since in this simulation B = 5, s = 0.05, r = 2.2, and from (4) we can see that DPM can reduce system power as long as T is greater than 2 cycles (20 ms). Since cores have an average idle interval of 80 ms (8 cycles) and cores are put to sleep if they are idle for more than 50 ms (5 cycles), each core has an average sleep cycles of 3 (>2). Besides, results also show that DPM has both

<table>
<thead>
<tr>
<th>TABLE I. PARAMETER VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural Parameters</td>
</tr>
<tr>
<td>CPU clock rate</td>
</tr>
<tr>
<td>Technology</td>
</tr>
<tr>
<td>Vdd</td>
</tr>
<tr>
<td>Core count</td>
</tr>
<tr>
<td>Total threads</td>
</tr>
<tr>
<td>DVFS settings</td>
</tr>
<tr>
<td>Delay and Power Model Assumptions</td>
</tr>
<tr>
<td>Transition power</td>
</tr>
<tr>
<td>Wakeup delay</td>
</tr>
</tbody>
</table>

Figure 3. Effects of DPM on energy, MTTF, performance and EDP

Equation (4) suggests that both transition power (B) and sleep cycles (T) play main roles on the effectiveness of DPM. We run another set of simulations with lower transition power of 5 W and results are shown in Fig. 4. Fig. 4 shows that DPM reduce system power by 20% at average. This is trivial since in this simulation B = 5, s = 0.05, r = 2.2, and from (4) we can see that DPM can reduce system power as long as T is greater than 2 cycles (20 ms). Since cores have an average idle interval of 80 ms (8 cycles) and cores are put to sleep if they are idle for more than 50 ms (5 cycles), each core has an average sleep cycles of 3 (>2). Besides, results also show that DPM has both
positive and negative effects on system power. For example, DPM increases the power consumption of *swaptions* by 4% comparing to the default case, and this is because *swaptions* has shorter idle interval at average. These findings shed some light on new thermal management techniques for multi-core systems running multi-threaded simulations. Based on the characteristics of parallel applications that cores have short idle interval (80 ms) and most of the cores (12 out of 16) are idle at a time, new management approach can be proposed to reduce core state-switching frequency and prolong core idleness.

![Figure 4. Effect of active-sleep state transition power on effectiveness of DPM](image)

**Table II. Effect of DPM on Temperature**

<table>
<thead>
<tr>
<th></th>
<th>Average Temperature (K)</th>
<th>Variance</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>331.39</td>
<td>127.84</td>
<td>11.31</td>
</tr>
<tr>
<td>DPM</td>
<td>329.90</td>
<td>173.02</td>
<td>13.15</td>
</tr>
</tbody>
</table>

We also examine the effect of DPM on system temperature. Table II shows that DPM reduces average temperature but also increases temperature variance. This is consistent with previous findings [9] that DPM can lead to greater thermal cycling which has adverse effect on reliability.

**C. Reliability-Aware Scheduling**

Next we examine the effect of dynamic voltage and frequency scaling (DVFS) together with DPM on power, performance, MTTF and EDP of CMT systems. We apply the DVFS-location technique [9] which sets the V/f for each core to a fixed value based on the location of the core. The four cores in the center of the floorplan have the 85% setting because the center cores tend to heat up more quickly. The corner cores use the 100% V/F setting because they are usually the coolest cores. The rest of the cores in the flooeplan have the 95% setting.

![Figure 5. Effects of DVFS with DPM on energy, MTTF and EDP](image)

Fig. 5 shows the effect of the DVFS-location on multi-core systems running PARSEC benchmarks. DVFS improves the system power by 7% and the MTTF by 36% at average while has a 15% decrease in EDP. This is because cores running with PARSEC applications have short idle interval and when the timeout period is 50 ms, performance is decreased by delay due to frequent waking up of cores. We expect that when new power management technique is applied to prolong core idleness, DVFS will have more positive impact on such systems while sacrifices very little in performance and EDP.

**D. Graph-Based Applications**

Graph-based applications have unstructured and intensive communications and poor load balancing, which can further lead to thermal hotspots. Therefore, managing power and reliability of systems running irregular graph-based applications is even more non-intuitive. We first study the characteristics of selective MTGL benchmarks. Results show that unlike PARSEC benchmarks, when the 16-core multiprocessor is running MTGL, all the cores are active at a time. Therefore, DPM has little effect on such system because cores are rarely idle.

Next we study the effects of DVFS on systems running MTGL. We adopt the DVFS policy where the hottest four cores (the central ones) have the 85% setting, the four coolest cores (the corner ones) have the 100% setting and the rest of the cores have the 95% setting. The setting for each core is fixed and does not change at runtime. Fig. 6 shows that DVFS improves system power by 17% and achieves 2.33 times improvement in MTTF due to its ability to reduce temperature even when cores are fully utilized. Moreover, DVFS results in a minimal performance loss (4%). We expect new DVFS policy that is especially designed based on the irregular characteristics of graph-based applications can further improve system power and reliability with little performance degradation.

![Figure 6. Energy and MTTF effects of DVFS on irregular applications](image)

**V. CONCLUSIONS AND FUTURE WORK**

In this work, we implement a framework for integrated power, thermal and reliability simulation of HPC systems and data gathering. The framework is the key interface in the Structural Simulation Toolkit (SST) that provides power/area/temperature/reliability modeling for SST components. Like the SST, the framework has a fully modular design and provides a parallel simulation environment. These capabilities provide SST a high level of performance and the ability to look at large systems, such as hardware/software co-design of future exascale systems. We illustrated the utilization of the framework by applying it to evaluate different dynamic thermal management techniques, in terms of power, performance and reliability, on multicore systems running parallel and graph-based applications.
The results in this paper shed some light on the design of future thermal management policies:

- Characteristics of applications need to be considered when designing new job scheduling and temperature management policies. For example, systems running PARSEC benchmarks have low utilization (12 of 16 cores are idle at a time) and cores have short idle interval. DPM and DVFS can be more effective on such systems if they are applied together with thread migration and job scheduling methods that reduce frequent core state switching.

- Systems running graph-based MTGL applications have high utilization and cores are active all the time. A simple location-based DVFS policy improves system power and reliability with small performance degradation; while the DMP fixed timeout policy is not effective in such systems. Better DVFS and temperature management policies can be designed that address the particular characteristics of graph-based applications, such as poor load balance and thermal hotspots.

- DPM can have both positive and negative effects on system power and reliability depending on the transition power of core state switching and the length of the idle interval. It is critical to combine DPM with intelligent migration and scheduling policy that prolong core idleness.

Our future work is to create novel management techniques with consideration of application characteristics, which perform thread migration, job scheduling and DVFS, to optimize application performance, energy consumption, and reliability of multithreaded multicore systems. Thread migration will look at migration within a chip, and between chips (either processor to processor-in-memory or between processors connected with a NIC).

ACKNOWLEDGMENT

The author would like to thank Arun Rodrigues, Chad Kersey, Ioannis Dougalis, and all of the anonymous reviewers for their help in improving the paper. Support from the Architectures management, James Ang, has also been invaluable.

REFERENCES


[22] Failure mechanisms and models for semiconductor devices, JEDEC publication JEP122C.


