INVITED: Cross-Layer Modeling and Optimization for Electromigration Induced Reliability *

Taeyoung Kim†, Zeyu Sun§, Chase Cook§, Hengyang Zhao§, Ruiwen Li§, Daniel Wong§ and Sheldon X.-D. Tan§

†Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA
§Department of Electrical Engineering, University of California, Riverside, CA 92521, USA
tkim049@cs.ucr.edu, stan@ece.ucr.edu

ABSTRACT

In this paper, we propose a new approach for cross-layer electromigration (EM) induced reliability modeling and optimization at physics, system and datacenter levels. We consider a recently proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks for long-term failures. We show how the new physics-based dynamic EM model at the physics level can be abstracted at the system level and even at the datacenter level. Our datacenter system-level power model is based on the BigHouse simulator. To speed up the online optimization for energy in a datacenter, we propose a new combined datacenter power and reliability compact model using a learning based approach in which a feed-forward neural network (FNN) is trained to predict energy and long term reliability for each processor under datacenter scheduling and workloads. To optimize the energy and reliability of a datacenter, we apply the efficient adaptive Q-learning based reinforcement learning method. Experimental results show that the proposed compact models for the datacenter system trained with different workloads under different cluster power modes and scheduling policies are able to build accurate energy and lifetime. Moreover, the proposed optimization method effectively manages and optimizes datacenter energy subject to reliability, given power budget and performance.

1. INTRODUCTION

Datacenter downtime has become a major concern as every minute equates to money lost. An unplanned outage can easily cost a datacenter $8,000 dollars per minute of downtime and can even reach costs of $16,000 per minute of downtime. The main root causes of unplanned failures are largely attributed to power system failure and human error. Hardware failures, such as server failures, only account for about 4%-5% of unplanned downtime. However, these types of failures are often much more difficult and costly to recover from. As a result, unplanned datacenter outages caused by server failures are responsible for the highest incurred costs, compared to downtimes attributed to other root causes, despite their low rate of occurrence as seen in Fig. 1(a) [1]. This presents much of the motivation behind the work in this paper as we develop a framework for reducing this hardware failure subject to performance constraints.

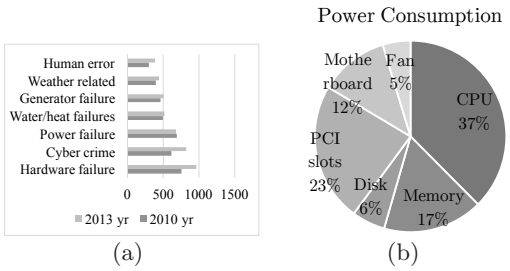


Figure 1: (a) Total datacenter cost by primary causes of unplanned outage (Thousand dollars) (b) Power consumption breakdown for one server

Although the servers consist of multiple components, existing works for datacenter hardware failure research have been mainly focused on the large scale studies in a hard disk [19] and memory failures [20]. However, in a typical server, the processor accounts for the majority of the power consumption at nearly 40% compared to other component such as memory and peripherals [7] in Fig. 1(b). Furthermore, a recent study found that processors are the leading cause of single node hardware failure in high performance computing clusters [16]. This trend is expected to become increasingly common as processor reliability is becoming a limiting constraint in high-performance processor designs due to high failure rates in deep submicron and nanoscale devices. Technology scaling has led to the continuous integration of devices, and processors will have more cores integrated. This growing trend for large scale many-core devices was brought upon by the increase in transistor density and the subsequent breakdown of Dennard Scaling. The result of which is the loss of power distribution scaling with transistor sizes, leading to increased chip temperatures, and the movement from utilization of a single powerful machine to a large cluster of machines which can help distribute workloads. However, large cluster system generates reliability concerns as we no longer can consider the reliability of just a single device or chip. This is especially true as each node in the datacenter begins to utilize highly integrated processors with their own reliability concerns. It is increasingly obvious that single server, or even chip level, reliability needs to be a large factor in how we address the reliability of numerous devices employed on a larger scale. In order to address these concerns, the relationship between datacenter and processor reliability should be examined. The reliability issue for datacenter presents the challenge of correlating processor and datacenter cluster reliability. We examine reliability effects of processors under practical datacenter workloads

*This work is supported in part by NSF grant under No. CCF-1255899, in part by NSF grant under No. 1255899, in part by Semiconductor Research Corporation (SRC) grant under No. 2013-TJ-2417 and in part by DARPA grant under No. HR0011-16-2-0009

DAC '16, June 05-09, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2897937.2905010

and model the effects that the operating parameters of the servers have on the processor reliability.

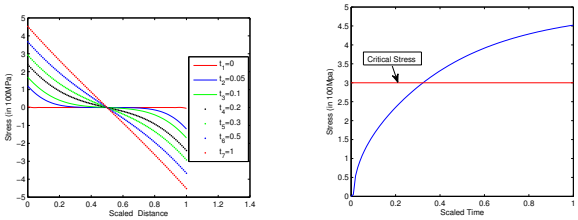
In this paper, we propose a novel cross-layer approach to optimize the energy of a datacenter subject to long-term reliability and performance constraints. We consider a recently proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks for long-term failures. EM has been previously identified as a major contributor to processor reliability in datacenters due to challenges of thermal management [3]. We stress the proposed method is orthogonal to other long-term reliability issues such as NBTI (negative biased temperature instability, TDDB (time-dependent dialectric breakdown), hot carriers etc. We show how the new physics-based dynamic EM model at the physics level can be abstracted at the system level and even at the datacenter level. Our datacenter system-level power model is based on the BigHouse simulator, which is recently proposed and uses a combination of queuing theory and stochastic modeling. To speed up online optimization for energy in a datacenter, we propose a new combined datacenter power and reliability model using a learning based approach in which a feed-forward neural network (FNN) is trained to predict energy and long term reliability for each processor under datacenter scheduling and workloads. To optimize the energy and reliability of a datacenter model, we apply the efficient and adaptive Q-learning based reinforcement learning method. Experimental results show that the proposed compact models for the datacenter system trained with different workloads under different cluster power modes and scheduling policies are able to build accurate energy and lifetime. Moreover, the proposed optimization method effectively manages and optimizes datacenter energy subject to reliability, given power budget and performance.

2. NEW PHYSICS-BASED EM MODELING AND ANALYSIS

EM is a physical phenomenon of the migration of metal atoms along a direction of the applied electrical field. Atoms (either lattice atoms or defects/impurities) migrate toward the anode end of metal wire along the trajectory of conducting electrons. During the migration process, hydrostatic stress will be generated inside the embedded metal wire due to momentum exchange between lattice atoms and conduction electrons and is a prime cause of void and hillock formation at the opposite ends of the wire. Indeed, when metal wire is embedded into a rigid confinement, which is the case with interconnect metallization, the wire volume changes (induced by the atom depletion and accumulation due to migration) create tension at the cathode end and compression at the anode ends of the line. Over time, the lasting unidirectional electrical load will increase hydrostatic stress, as well as the stress gradient which act as counter-forces for atom migrations along the metal line. In some cases, usually when a line is long, this stress can reach a critical level, resulting in a void nucleation at the cathode and/or hillock formation at the anode end of line. Existing Black's model [4] is a semi-empirical model with no physics behind. For instance, it does not consider the impacts of residual stress and wire lengths on the mean time to failure (MTTF) of a wire. Also when the wire reaches its MTTF, the wire is treated as an open circuit, which will over-estimate EM-impacts in circuit reliability.

2.1 Void nucleation and growth phases

Since existing methods cannot scale very well, a more physics-based EM model has been proposed recently for full-chip reliability analysis [12, 22], which is the basis for the proposed work. In this new EM model, the EM development process consists of two phases - the nucleation phase and the growth phase. Development of such analytical formulation was proposed by Korhonen [14]. Stress evolution of nucleation phase based on his model is shown in Fig. 2. Fig. 2(a) shows the EM-induced stress development for a single metal wire over time from the finite element analysis for a given current density and temperature setting. Fig. 2(b) shows the stress evaluation over time. Each time unit here is 10^7 seconds. During this process, tensile stress (positive stress) will be developed at the cathode side (the left node), while compressive stress (negative stress) will be generated at the anode side of the wire (right node). When the tensile stress hits critical stress, a void will be generated, which is called nucleation time (t_{nuc}). The growth phase begins after the void is formed at t_{nuc}. Void starts growing and the wire resistance starts to increase over the time in this phase. As a result, the p/g network becomes a time-varying network and its voltage drops will vary over the time [12].



(a) (b)

Figure 2: (a) EM-stress distribution change over time in a simple metal wire in nucleation phase. (b) EM-stress evaluation on cathode versus time using physics-based model.

2.2 Power grid-level EM reliability model

Because of the concern with the long-term average effects of the current, in EM related work a DC model of the power grid is generally assumed [5]. As a result, we consider only the EM-induced kinetics of the power grid network resistances. In our problem formulation, each mortal wire, subjected to the EM impact, will start to change its resistance value upon achieving the nucleation time. As a result, we end up with the power grid systems, which is a linear, time-varying and driven by the DC effective currents, which is modeled as $G(t)v(t) = I_{eff}$, where, $G(t)$ a $n \times n$ time-varying conductance matrix; I_{eff} is the effective DC current source vector; $v(t)$ is the corresponding vector of nodal voltages and n is the size unknown voltages. In our problem, the time scale is the EM time scale, which can be months or years.

In the new EM-induced reliability analysis algorithm for p/g networks, we compute the voltage drops of the grids at fixed EM time step. The resistance of one or more wires begins to change (increase) starting with their nucleation times. At each time step, we collect new wires whose nucleation times were reached, and compute the new resistance for existing wires in the growth phases and corresponding voltage drops of the whole grids. This process is repeated until the voltage drop of one or more nodes exceeds the critical voltage drops allowed (say 10% of Vdd).

2.3 EM-induced reliability model for a many-core processor

An existing EM model, including the new physics-based model, can only take a constant temperature. Previous study shows that system-level time-to-failure (TTF) or lifetime under different temperature can be approximated by [15]: EM reliability model for individual core can be expressed as follows

$$TTF_{i-core} = \frac{1}{(\sum_{k=1}^{n}(\Delta t_{i,k}\frac{1}{TTF_{i,k}}))/T} \quad (1)$$

where $TTF_{i,k}$ is the actual TTF under the k-th power and temperature settings for Δt_k period, assuming i-th core works through n different power and temperature settings and $T = \sum_{k=1}^{n} \Delta t_{i,k}$. Each $TTF_{i,k}$ will be computed based on the recently proposed physics-based EM model and assessment techniques [12].

A manycore processor lifetime can be defined as the shortest lifetime among the cores [6,23]. The individual core lifetime can be obtained from (1). Recently, one study used *performability* as the ratio of number of non-failure cores over total number of cores [21] to explain chip multiprocessor (CMP). But the specific mechanism was not introduced and is too abstract, so we use the shortest lifetime in this paper, however, our framework easily extends to support performability later.

2.4 EM-induced reliability-aware datacenter model

To evaluate datacenter-level EM-induced reliability, we use the BigHouse simulator [18], a simulation infrastructure for datacenter. BigHouse is based on stochastic queueing simulation, a validated methodology for simulating the power/performance behavior of datacenter workloads. The BigHouse simulator is widely used in academia, as well as in Google datacenter research [17, 18, 24].

BigHouse uses synthetic arrival/service traces that are generated through empirical interarrival and interservice distributions collected from real systems [17, 18]. We evaluate two major workloads, Domain Name Service (DNS) and Apache World Web Service (WWW), provided with the BigHouse simulator. These workloads are modeled by workloads distribution, which represents the average, standard deviation(σ), and coefficient of variation(C_v) for the interarrival and service time distributions of the workloads. The interarrival distribution is used to drive the queueing model, while the service time distribution is used for the service nodes. These synthetic arrival/service traces are fed into a discrete-event simulation of a G/G/k queueing system that models active and idle low-power modes through state-dependent service rates.

During simulation time, measures of interest, such as power consumption and 99-th percentile latency, are obtained by sampling the output of the simulation until each measurement reaches a normalized half-width 95% confidence interval of 5%. The simulation ends when the sample statistics are considered *converged*, that is, once the observed sample size is sufficient to achieve the desired confidence interval of 95%. The sample size required to achieve a certain confidence is given by:

$$N_m = \frac{Z^2_{1-\alpha/2} * \sigma^2}{\epsilon^2} \tag{2}$$

where σ is the standard deviation of the samples, ϵ is the half-width of the desired confidence interval, and $Z_{1-\alpha}$ is the value of the standard normal distribution at the $(1-\alpha/2)^{th}$ quantile. For 95% confidence, this value is 1.96.

To explore the EM effect on datacenter-level reliability, we integrate the EM model into BigHouse simulator. Additionally, we added thermal modeling into BigHouse, and drive the EM model using power, voltage, and temperature measurements. The thermal modeling is achieved using the HotSpot thermal model [11]. This thermal model offers a compact solution with relatively good accuracy and speed. HotSpot is integrated into BigHouse and fed a power trace of each simulated core using the method described above. Each core is then modeled and simulated to produce a thermal trace. It is this thermal trace that is used as the temperature measurements for the EM model.

To explain server-level reliability on datacenters, we use average socket lifetime (Mean-time-to-failure, MTTF). One socket lifetime can be defined as the shortest lifetime among the processors in one server.

We use tail latency as most important service latency for the datacenter since the tail flow completion time (FCT), 99-th or 99.9-th percentile FCT, can be more than 10x larger than the mean FCT. So tail latency is a very crucial performance metric for datacenter as the service response needs to wait for the slowest flow/workload to complete [2].

3. NEW RELIABILITY-CONSTRAINED ENERGY OPTIMIZATOIN FOR DATACENTER

In this section, we introduce new reliability-constrained energy optimization for datacenter. To speed up the online optimization for datacenter energy and reliability, we use a neural network based approach for datacenter power and reliability model. We use feed-forward neural network (FNN), which is trained to predict energy and long-term reliability for each processor under datacenter scheduling and workloads. To further optimize energy and reliability of a datacenter model, we formulate a learning-based optimization method, Q-learning, as minimizing datacenter energy subject to reliability, given power budget and performance.

3.1 Neural networks for datacenter energy and reliability models

3.1.1 Review of feed forward neural network

To build a compact energy and reliability model for datacenters, learning based techniques such as neural network, which is composed of multiple processing layers, can learn representations of data with multiple levels of abstraction. Processor power consumption and EM-induced lifetime/MTTF can be considered as supervised learning in neural networks. One advantage of neural networks is its wide applications for nonlinear systems. The universal approximation capability of feed-forward neural networks (FNN) has been proved to show that any Borel measurable function can be approximated with any arbitrary accuracy by an FNN using squashing activation functions [10].

If we have an input vector $\boldsymbol{u} = \{u_1, u_2, \cdots, u_p\}$ and an output vector $\boldsymbol{y} = \{y_1, y_2, \cdots, y_q\}$, then the layer-wise structured FNN without bias node has the form

$$\begin{aligned} \boldsymbol{a}_1 = \boldsymbol{f}_1(\boldsymbol{u}\boldsymbol{W}^{(\mathrm{IN},1)}), \quad \boldsymbol{a}_2 = \boldsymbol{f}_2(\boldsymbol{a}_1\boldsymbol{W}^{(1,2)}), \quad \cdots, \\ \boldsymbol{a}_i = \boldsymbol{f}_i(\boldsymbol{a}_i\boldsymbol{W}^{(i-1,i)}), \quad \cdots, \quad \boldsymbol{y} = \boldsymbol{a}_k\boldsymbol{W}^{(k,\mathrm{OUT})} \end{aligned} \tag{3}$$

where the activation function $\boldsymbol{f}$ is element-wise squashing operator such as a sigmoid or a hyperbolic tangent function; vector $\boldsymbol{a}_i$ is the intermediate activation result of each layer; $\boldsymbol{W}^{(\cdot,\cdot)}$ is the weighting matrix connecting adjacent layers. FNN with bias node requires each activation result vector $\boldsymbol{a}_i$ to be appended with a fixed unit value before it is fed into next level of calculation, and the dimensions of $\boldsymbol{W}^{(\cdot,\cdot)}$ also need to be adjusted accordingly.

3.1.2 Neural network training for datacenter reliability-aware energy model

As seen in (3), in theoretical aspect, training a neural network is equivalent to the optimization problem to minimize cost function (without bias node or connections, without regulation terms):

$$\boldsymbol{J}\left(\boldsymbol{W}^{(\mathrm{IN},1)}, \boldsymbol{W}^{(1,2)}, \cdots, \boldsymbol{W}^{(k,\mathrm{OUT})}\right) = \sum_{j=1}^{m} \|\boldsymbol{y}_j - \hat{\boldsymbol{y}}_j\| \tag{4}$$

where $\hat{\boldsymbol{y}}_i$ is a neural network output which can be explicitly written in a nested activation form

$$\hat{\boldsymbol{y}} = \boldsymbol{f}_k(\boldsymbol{f}_{k-1}(\boldsymbol{f}_{k-2}(\cdots \boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{u}\boldsymbol{W}^{(\mathrm{IN},1)})\boldsymbol{W}^{(1,2)})\boldsymbol{W}^{(2,3)}\cdots) \boldsymbol{W}^{(k-2,k-1)})\boldsymbol{W}^{(k-1,k)})\boldsymbol{W}^{(k,\mathrm{OUT})} \quad (5)$$

Therefore, the training problem of neural networks can be solved by applying existing optimization methods such as gradient decent, Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [8], and the Quasi-Newton method with the cost function $\boldsymbol{J}$. In practice, an algorithm with lower computational cost, Back-propagation, has been widely used for solving the training problem [9].

3.1.3 Neural network structure and data configuration

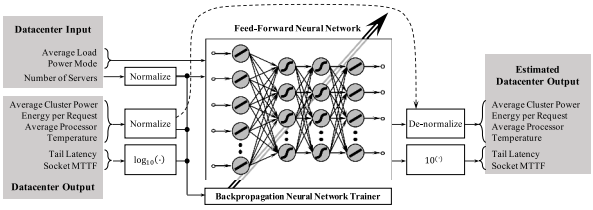


Figure 3: Feed-forward neural network structure and data configuration

As shown in Fig. 3, the feed-forward neural network (FNN) can be constructed to predict energy and long-term reliability for each processor under datacenter scheduling and workloads. We separately construct and train networks for each individual workloads. The inputs to the neural networks are average load rate, power mode (quantified), and number of servers in the datacenter. With these inputs, the neural networks can estimate average cluster power, average processor temperature, tail latency, and average socket MTTF. To train the neural networks more efficiently with less numerical stability issues, the scaling of the inputs is required. Otherwise, it can have a large effect on the quality of final solution. As shown in Fig. 3, the number of servers is normalized. The average load rate can be used without scaling since it already has a good distribution. In the same way, the output data set can be scaled and converted into logarithmic scale since they are served as a part of the training input set in the back-propagation algorithm [9]. We use three hidden layers with sigmoid activation functions, with all layers having 15 nodes respectively. The input and output layer sizes are 3 and 5 respectively.

3.2 Q learning optimization for datacenter

3.2.1 State and action determination

Q-learning is a reinforcement learning method used as a controller to maximize long-term rewards. It can converge close to the optimal result of a state-action function for arbitrary policies while handling problems with stochastic transition [13]. In this case, state(s) used in this work consists of workload model, average load rate, power model, and number of servers. Action (a) is used to describe transitions between two states. Executing an action in a certain state provides a learning agent contained in the model whose goal is to maximize reward (minimize penalty) with updated Q-value by reward/penalty calculation in the Q-table, also known as the state-action table. The environment part is reliability-aware BigHouse model, whose learning agent is Q-learning algorithm. The learning agent can obtain the environment state, calculate penalty function, and finally, decide the next action.

3.2.2 Q-value function and Q-learning process

In the Q-learning process, one critical issue is to define the Q-value function with penalty term. Each state s_i will determine average cluster power $Power(s_i)$, tail latency of datacenter $latency(s_i)$, the average processor temperature, $Temp(s_i)$. $EM_{min}(s_i)$, which is defined as average socket MTTF in datacenter. $E(s_i)$ is energy per request in datacenter. An action, say $a_{i,j}$, can be viewed as the transition from state i to state j. The penalty function Q determines a penalty and a new state which is related to the previous state and selected action. Q-value is updated at every step Δt.

$$Q^{t+1}(s(t),a(t)) = Q^t(s(t),a(t)) + \alpha(t) \times \Big(PT(t+1) + \gamma \min_a(\forall Q^t(s(t+1),a)) - Q^t(s(t),a(t))\Big) \quad (6)$$

where $\alpha(t)$ is learning rate between 0 and 1 which determines the percentage of newly calculated Q-value applied. $s(t+1)$ is determined by action $a(t)$, and $Q^t(s(t+1),a)$ are all possible action's Q-values from future state. The discount factor γ (between 0 and 1) determines the importance of future penalty. $\min(\forall Q^t s(t+1),a)$ is considered to be estimated optimal future value. The difference between old Q-value (Q^t) and learned value ($PT(t+1)+\gamma \min_a(\forall Q^t(s(t+1),a))$) updates the new Q-value (Q^{t+1}) with the learning rate. A penalty term (PT) is shown in (7). PT_E is a penalty term for total datacenter energy, PT_{EM} is a penalty term for average socket MTTF, PT_{power} for average cluster power, PT_{temp} for average processor temperature, and $PT_{latency}$ is tail latency of datacenter. Each penalty term (PT_x) is normalized in (7). This feature scaling method to bring all values between 0 and 1. For instance $PT_E = \frac{E(t+1)-E(t)}{E_{Max}-E_{Min}}$ is for energy related penalty, where $E(t)$ is the energy per request in the previous time t and $E(t+1)$ is energy per request of the datacenter at current time $t+1$. For the EM MTTF, $PT_{EM} = \frac{MTTF(t)-MTTF(t+1)}{MTTF_{Max}-MTTF_{Min}}$ is for EM related penalty where $MTTF(t)$ is the average socket MTTF of the datacenter for EM-induced in the previous time t and $MTTF(t+1)$ is the average socket MTTF of the datacenter at current time $t+1$.

$$PT = PT_E + C \sum_{x=\{EM,power,temp,latency\}} \delta_x PT_x$$
$$\delta_x = \begin{cases} 0 & \text{if } PT_x \le B_x + \Delta_x \\ 1 & \text{if } PT_x > B_x + \Delta_x \end{cases} \quad (7)$$

where δ_x is a binary function to active ($\delta_x = 1$) or inactive ($\delta_x = 0$) of user defined or given constraint bounds, B_{power}, $B_{latency}$, and B_{temp} in the penalty term. They are also normalized average cluster power, tail latency, average processor temperature bounds respectively. Each Δ_x is the difference between each bound and average penalty (PT) for the power, latency, and temperature. Δ_x is positive if the whole datacenter violated the given constraint, otherwise, it is negative which means the system is bounded and working in acceptable condition. Therefore, if the datacenter violated user's constraints in the past, penalty would be significant due to large value for constant C in (7).

The proposed learning-based energy optimization algorithm goes with the following flow: First, all the Q-values in the Q-table are initialized to zero. Current state $s(t)$ finds an action $a(t)$ with the lowest Q^t in (6) and switches to next state corresponding to input values. For every step, average socket MTTF, latency, average processor temperature, average cluster power, and energy per request are evaluated

and thus, all environments can be updated. Then, new corresponding penalty $PT(t+1)$ would be calculated in (7) and Q^{t+1} would be updated (learning process). After the update, the current state could be replaced by a new action and it would iterate with a new updated state. Finally, when all the Q-value changes are less than a certain threshold, the best policy will be chosen based on the result.

3.3 Proposed new datacenter framework for energy and reliability

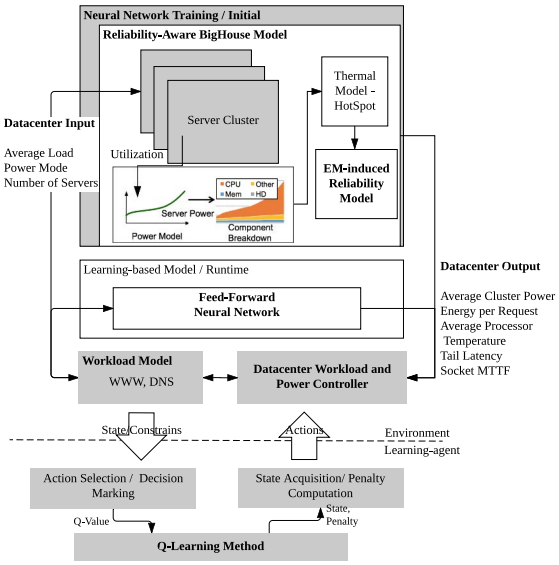


Figure 4: The evaluation platform for datacenter and energy and reliability management algorithms

To evaluate the proposed new reliability-constrained energy optimizatoin for datacenter, we use BigHouse model, which can provide cluster power and performance models with different datacenter scheduling and workloads, such as average load rate, power model (Low, Mid, and High), number of active servers in datacenter. Once BigHouse model generate the performance and power traces of each core in the server of datacenter, HotSpot can generate each core's temperature from the power traces. For the EM-induced reliability, we use the power traces, the thermal traces, and the core's voltage as input to generate the individual core EM-induced lifetime of a manycore processor. As explained in Section 2.3, system-level (a processor) EM-induced lifetime can be calculated. For the datacenter level lifetime, we use average processor MTTF.

The training data can be obtained from BigHouse simulator with all the possible datacenter scheduling and workloads to train the neural network to speed up online optimization for datacenter power and reliability model. With the trained network, Q-learning method can find the optimal policies for datacenter scheduling and workloads to achieve minimizing energy subject to given reliability, power performance constraints as seen in Fig. 4.

4. NUMERICAL RESULT AND DISCUSSIONS

4.1 Experimental setup

The proposed new compact model (FNN-based) and optimization (Q-Learning) for the datacenter framework have been implemented in Python 2.7.9 with the numerical libraries (NumPy 1.9.2 and Scipy 0.15.1). Thermal model (HotSpot 6.0 [11]) to estimate EM-induced lifetime. BigHouse utilizes a simple system-level power model, as shown in figure 4, which takes in a server utilization and outputs the power consumption of each server. Two major workloads (DNS and WWW) have been used to evaluate our proposed models. The server power model is based on a highly energy proportional server (Huawei XH320) derived from reported SPECpower benchmark results [25]. Our EM model requires per core energy. In order to extract per core energy, we instrumented a high energy proportional server to measure per-component power, with component breakdown as shown in Fig. 4.

We instrumented each individual component by intercepting the power rails and measuring the current with LTS 25-NP current sensors. The outputs of the current sensors are sampled at 1kHz using a DAQ and logged using LabView. To measure CPU power, we inserted a current sensor in series with the 4-pin ATX power connector. To measure memory power, we inserted a current sensor in series with pins 10 and 12 of the 24-pin ATX power connector which supplies power to the mother board. To measure the power of the hard drive, we inserted a current sensor in series with the hard drive backplane power connector. We use the per-component power breakdown to derive the per core power from the server.

	Training Error		Validation Error	
	DNS	WWW	DNS	WWW
Tail lagency	3.97%	6.53%	2.83%	9.37%
Avg. cluster power	2.64%	2.45%	3.02%	3.50%
Avg. proc. temp.	0.549%	2.91%	0.497%	2.92%
Avg. proc. MTTF	5.59%	6.78%	5.70%	7.40%
Energy per request	0.671%	0.738%	1.57%	1.20%

Table 1: Accuracy analysis (RMSE) of the feed-forward neural network (FNN) model

	Energy per Request (J)	Energy Saving (%)
Max State (DNS)	67.63	
Case 1 (DNS)	18.76	72.25
Case 2 (DNS)	24.08	64.39
Case 3 (DNS)	35.04	48.18
Max State (WWW)	23.71	
Case 4 (WWW)	8.44	64.37
Case 5 (WWW)	8.44	64.37
Case 6 (WWW)	12.25	49.30

Table 2: Energy optimization for datacenter

4.2 Evaluations of proposed new modeling and optimization

First we evaluate our learning-based datacenter modeling (see Section 2.4) We get normalized root mean square error (RMSE) by calculating $\frac{1}{\max(y_{\text{ref}})-\min(y_{\text{ref}})}\sqrt{\frac{1}{n}\sum(y_{\text{est}}-y_{\text{ref}})^2}$, where y_{ref} and y_{est} are obtained from the reliability-aware BigHouse model (reference) and FNN-based model (estimated), respectively. TABLE 1 shows each training error and validation error of the proposed compact model. In validation phase, both estimations have good accuracy on DNS and Web datacenter workloads, where RMSEs are lower than 10%.

Second, we evaluate our learning-based optimization method (see Section 3.2) by optimizing for energy savings with different sets of average processor MTTF, average cluster power, and tail latency. Table. 2 and Fig. 5 shows the energy savings given constraint for average processor MTTF , average cluster power and tail latency, with DNS and WWW workload on the proposed datacenter framework. As we can see, energy savings for the different constraints have been evaluated in Fig. 5, case 1-3 is DNS workload and case 4-6 is

	Number of servers increase	Power model increase	Average load increase
Average Power	Steeply increase	Decrease	Increase
Tail latency	No trend	No trend	Slightly increase
Energy per request	No trend	Decrease	No trend
Avg socket MTTF	No trend	No trend	Decrease

Table 3: Trends of proposed reliability-aware model

WWW workload with tight MTTF constraints (case 1 and 4) and loose MTTF constraints (case 3 and 6). In Table. 2, our method finds relatively high energy savings for each case.

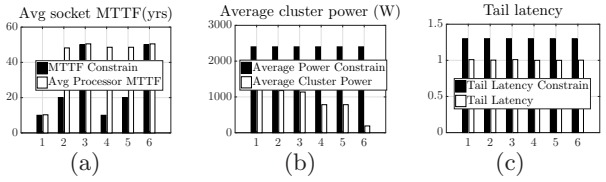


Figure 5: Validating violations with constraint limits (a) Average socket MTTF (b) Average cluster power (c) Tail latency

We show the trends among average power, tail latency, and average socket MTTF (EM-induced) obtained from the proposed compact models as shown in Table 3. Increasing number of servers, average cluster power also increases. No trends were found in tail latency, energy per request, and average socket MTTF for the number of servers. For power model increase (From low to Mid, and High), average cluster power decreased and energy per request decreased as high power can reduce delay, thus energy per request can decrease. With high average load, both average cluster power and tail latency increased. Finally, we observed that there was one obvious trend that the high average load rate of datacenter can significantly impact on datacenter reliability, which can lower socket MTTF in our proposed model.

5. CONCLUSION

We proposed a novel cross-layer approach to optimizing the energy of a datacenter subject to long-term reliability and performance constraints. We considered a recently proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks for long-term failures. We showed how the new physics-based dynamic EM model at the physic level can be abstracted at the system level and even at in a datacenter level. To speed up the online optimization for energy for datacenter, we proposed a new combined datacenter power and reliability model using a learning based approach in which a feed-forward neural network (FNN) was trained to predict energy and long term reliability for each processor under datacenter scheduling and workloads. To optimize the energy and reliability of a datacenter model, we applied the Q-learning based reinforcement learning method. Experimental results showed that the proposed compact models for the datacenter system trained with different workloads under different cluster power modes and scheduling policies are able to build accurate energy and lifetime. Moreover, the proposed optimization method effectively managed and optimized datacenter energy subject to reliability, given power budget and performance.

6. REFERENCES

[1] 2013 cost of data center outages, 2013. http://www.emersonnetworkpower.com.

[2] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker. pfabric: Minimal near-optimal datacenter transport. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 435–446, New York, NY, USA, 2013. ACM.

[3] S. Biswas, M. Tiwari, T. Sherwood, L. Theogarajan, and F. T. Chong. Fighting fire with fire: modeling the datacenter-scale effects of targeted superlattice thermal management. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 331–340. IEEE, 2011.

[4] J. R. Black. Electromigration-A Brief Survey and Some Recent Results. *IEEE Trans. on Electron Devices*, 16(4):338–347, 1969.

[5] S. Chatterjee, M. Fawaz, and N. F. Najm. Redundancy-Aware Electromigration Checking for Mesh Power Grids. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2013.

[6] A. Das, A. Kumar, and B. Veeravalli. Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 689–694, San Jose, CA, USA, 2013. EDA Consortium.

[7] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, pages 13–23, New York, NY, USA, 2007. ACM.

[8] M. T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill, 1997.

[9] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989.

[10] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[11] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, May 2006.

[12] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan. Physics-based electromigration assessment for power grid networks. In *Proc. Design Automation Conf. (DAC)*, June 2014.

[13] T. Jaakkola, M. I. Jordan, and S. P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6(6):1185–1201, Nov. 1994.

[14] M. A. Korhonen, P. Borgesen, K. N. Tu, and C. Y. Li. Stress Evolution Due to Electromigration in Confined Metal Lines. *Journal of Applied Physics*, 73(8):3790–3799, 1993.

[15] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron. Interconnect lifetime prediction under dynamic stress for reliability-aware design. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pages 327–334. IEEE, November 2004.

[16] C. D. Martino, Z. Kalbarczyk, R. K. Iyer, F. Baccanico, J. Fullop, and W. Kramer. Lessons learned from the analysis of system failures at petascale: The case of blue waters. In *Proceedings of the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '14, pages 610–621, Washington, DC, USA, 2014. IEEE Computer Society.

[17] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch. Power management of online data-intensive services. In *International Symposium on Computer Architecture*, 2011.

[18] D. Meisner, J. Wu, and T. F. Wenisch. Bighouse: A simulation infrastructure for data center systems. In *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on*, 2012.

[19] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, FAST '07, pages 2–2, Berkeley, CA, USA, 2007. USENIX Association.

[20] B. Schroeder, E. Pinheiro, and W.-D. Weber. Dram errors in the wild: A large-scale field study. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, pages 193–204, New York, NY, USA, 2009. ACM.

[21] W. Song, S. Mukhopadhyay, and S. Yalamanchili. Architectural reliability: Lifetime reliability characterization and management of many-core processors. *Computer Architecture Letters*, PP(99):1–1, 2014.

[22] V. Sukharev. Beyond Black's Equation: Full-Chip EM/SM Assessment in 3D IC Stack. *Microelectronic Engineering*, 120:99–105, 2014.

[23] S. Wang and J.-J. Chen. Thermal-aware lifetime reliability in multicore systems. In *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pages 399–405, March 2010.

[24] D. Wong and M. Annavaram. Implications of high energy proportional servers on cluster-wide energy proportionality. In *Proceedings of the 19th IEEE International Symposium on High Performance Computer Architecture*, HPCA-19 '14, 2014.

[25] www.spec.org/power_ssj2008/. Specpower_ssj2008, 2012.