Scalable analytical model of the reliability of multi-core systems-on-chip by interacting Markovian agents

Andrea Bobbio
DiSit - Università Piemonte Orientale
Alessandria, Italy
andrea.bobbio@uniupo.it

Cristiana Bolchini
Deib - Politecnico di Milano
Milano, Italy
cristiana.bolchini@polimi.it

Davide Cerotti
DiSit - Università Piemonte Orientale
Alessandria, Italy
davide.cerotti@unipo.it

Marco Gribaudo
Deib - Politecnico di Milano
Milano, Italy
marco.gribaudo@polimi.it

Antonio Miele
Deib - Politecnico di Milano
Milano, Italy
antonio.miele@polimi.it

ABSTRACT

The reliability of multi-core systems-on-chip has been the object of several studies in recent years since these devices are heavily utilized in modern digital equipment at any level of complexity. This level of integration has caused a reduced time to failure due to the rapid scaling down of the dimension with consequent increase of the cores temperature and current densities. Past studies have utilized discrete event simulation: a technique very difficult to master in this scenario due to the number of components and the rarity of the failure events. The present study proposes an analytical framework based on Markovian Agent Models (MAM), able to capture systems with the big number of cores possible with the today and tomorrow's technology, while at the same time considering the effects caused by the position of the cores (center, border, corner) on the temperature level, and the dynamic redistribution of the workload with the progressive failure of the cores. The paper presents the model, adopting realistic parameters and interaction phenomena taken from the literature.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; • **General and reference** → *Reliability*; • **Hardware** → Temperature simulation and estimation;

KEYWORDS

Modeling, reliability evaluation, multi-core systems

ACM Reference Format:
Andrea Bobbio, Cristiana Bolchini, Davide Cerotti, Marco Gribaudo, and Antonio Miele. 2017. Scalable analytical model of the reliability of multi-core systems-on-chip by interacting Markovian agents. In *Proceedings of ACM Valuetools 2017 (VALUETOOLS'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

VALUETOOLS 2017, December 5–7, 2017, Venice, Italy

ACM ISBN 978-1-4503-6346-4/17/12…$15.00
https://doi.org/10.1145/3150928.3150935

1 INTRODUCTION

In the past decades, the aggressive advances in chip manufacturing process and technology scaling have led to the integration of several processing cores within the same chip. The consequence has been the growing adoption of multi-core and many-cores architectures in all the computing system scenarios, spanning from the embedded applications to the High-Performance Computing Infrastructures.

However, this technological progress has brought also a considerable drawback in terms of devices' lifetime and reliability. As a matter of fact, the extreme transistor shrinking and integration have caused a considerable increase in the current densities, and in turn, in the operating temperatures. As reported in the International Technology Roadmap for Semiconductors (ITRS) in 2011 [13], such temperature increase is the primary cause of an acceleration in device aging and wear-out phenomena, including electromigration (EM), time dependent dielectric breakdown (TDDB), negative bias temperature instability (NBTI) and thermal cycling (TC), that are leading to a higher susceptibility of digital systems to degradation effects, such as timing errors, and eventually to breakdowns. The resulting dramatic decrease of the lifetime of digital computing systems based on multi-core chips has pushed the focus on the performance and reliability of such devices. In fact, the system-level design choices (e.g., mapping and scheduling, workload distribution, utilization levels) have a macroscopic impact on the chip lifetime higher than circuit-level reliability enhancing techniques [24].

In the past, the reliability of multi-core devices has been studied by resorting to simulation [3, 11, 22, 24]. Due to the high number of cores per chip and the rarity of the failure events, simulation experiments require very extensive computation times. This paper has the purpose of investigating a scalable analytical model able to analyze the reliability of a single core as well as of the overall multi-core system taking into account the impact and progression of aging phenomena

in time and the problem of maintaining a maximum power budget, that is connected to the dark silicon areas [9] of CPU. The construction of the analytical model is particularly challenging because the lifetime of any single core is influenced by the operating conditions of all the other cores. The time dependent rate of wear caused by the progressive aging is primarily related to the core temperature and utilization level. The core temperature depends both on the internal power consumption and on the heat conveyed by neighboring cores. The power consumption on each core is determined by runtime resource management strategies but varies in time due to the progressive failures of the cores with consequent redistribution of the workload. The usual assumptions of system failure after the first core failure is too simplistic to fully estimate the system's lifetime [10, 11, 15, 24] in a dynamic working scenario where the workload redistribution policy should be taken into account. We prefer to relate the reliability to the capacity of the system to process an assigned amount of workload.

The proposed analytical model is based on the formalism of Markovian Agent Model (MAM) [1, 4, 5]. A MAM is a collection of Markovian Agents, where an MA is a finite-state stochastic model governed by an infinitesimal generator kernel that contains local transition rates and transition rates induced by the interaction with the other MAs. Furthermore, MAs are located in a defined geographical space, so that their local properties and the interaction induced mechanisms may depend on their relative positions. The MA dependence on the position in the space allows the model to investigate the effect of various geometrical layouts of the cores on the chip. In the rest of the paper, we will focus on illustrating how the MAM formalism is suited to model temperature and workload variations in time and how these variations are influenced by the interaction among the cores, so that the reliability of a multi-core system-on-chip can be evaluated. Preliminary results are presented to illustrate the potentialities of the model.

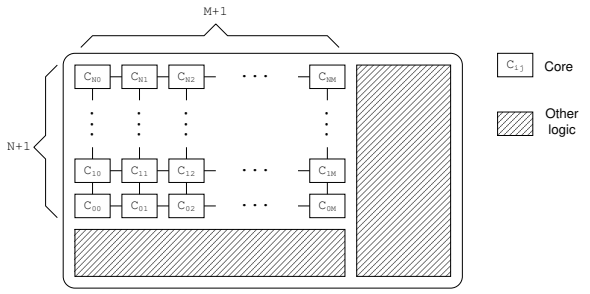


Figure 1: Simplified architecture of a many-core CPU

2 MARKOVIAN AGENT MODEL

Markovian Agents (MA) have a finite number of states over which a transition kernel is defined. We define a Markovian Agent Model (MAM) as an agent-based spatiotemporal analytical formalism composed by a collection of interacting Markovian Agents (MAs). MAMs are intended to model large scale systems of interacting objects, in which the local properties of each MA as well as the interactions among MAs may depend on the spatial localization of the agents.

The transition kernel of an MA is composed of two components a *local transition matrix* and an *induced transition matrix*. The local matrix contains a fixed component that depends on the MA structure and its position in the space but does not depend on the interaction with the other MAs. The induced transition matrix depends on the interaction of an MA with the other MAs. A general view of MAMs is given in [1]. We show that MAMs are suited to model the reliability of multi-core system-on-chip by taking into account that the failure characteristics of each individual core vary in time as a function of the aging level, and depend on the position of the core on the chip (Figure 1), on the operating state of the core and on the interactions with neighboring cores. MAs may belong to different classes that are distinguished by a subscript c; MAs inside each class are characterized by the same governing transition kernel whose parameters are possibly related to the MA position in the space.

The transition kernel governing the stochastic dynamics of an MA of class c is be written as:

$$\boldsymbol{K_c}(t; v; \boldsymbol{\Pi_V}) = \boldsymbol{Q_c}(t; v) + \mathcal{I}_c(t; v; \boldsymbol{\Pi_V})$$

where $\boldsymbol{Q_c}(t; v)$ is the local kernel (infinitesimal generator that depends only on the position v of the MA, $\mathcal{I}_c(t; v; [\boldsymbol{\Pi_V}])$ is the induced kernel that depends on the position v of the MA and on the ensemble $[\boldsymbol{\Pi_V}]$ of the state probability vectors of all the MAs in the space at time t.

By incorporating the interaction term into the global kernel $\boldsymbol{K_c}(t; v; [\boldsymbol{\Pi_V}])$, we can avoid the construction of the combined state space and solve the overall model by building several sub-models, one for each MA, and then solve them separately, by means of the standard equation:

$$\frac{d\,\boldsymbol{\pi}_c(t; v)}{d\,t} = \boldsymbol{\pi}_c(t; v)\,\boldsymbol{K_c}(t; v; [\boldsymbol{\Pi_V}]) \qquad (1)$$

with given initial condition $\boldsymbol{\pi}_c(0; v)$. In Equation (1), $\boldsymbol{\pi}_c(t; v)$ is the state probability vector of a single MA of class c at time t. At any time t, the solution of the MAM implies the solution of an equation of type (1) for any MA of any class c in the space V. The challenge of the modeling problem using MAM consists in deriving the structure of the kernel $\boldsymbol{K_c}(t; v; [\boldsymbol{\Pi_V}])$ for any class c of MAs. Several examples are in [1, 5].

3 SYSTEM MODEL

We evaluate the reliability of a multi-core system-on-chip by defining for each single core two MAs of different classes. The MA of class *Operation mode* has three states: *active (W)*, *idle (I)* and *failed (F)* (Figure 2). The MA of class *Wear-out* has a single state *Aging (A)* (Figure 2) and is used to identify the aging level of the core. The core may alternate between the W and the I modes before reaching the absorbing state F. The transition rates between W and I depend on the

location of the core, on the initial workload of the core and on the policy by which the load is redistributed among the still alive cores, upon a core failure. This is why $\delta(v, \mathbf{\Pi_V})$ and $\gamma(v, \mathbf{\Pi_V})$ are function of the position and of the probability vectors of all the other core states (see Equations from (2) to (5)). The time to failure (TTF) distribution either from state W or I to F, is a function of the aging level of the core that primarily depends on core temperature and utilization level.

We assume that the workload to be processed by the system

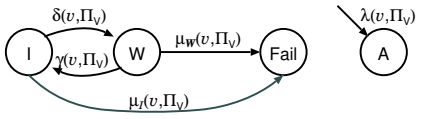


Figure 2: Markovian Agent model of a many-core CPU core

is known and constant in time. The assigned workload takes into account the power budget (that must not be exceeded), and the workload distribution policy assumes a rotation in the activation of the cores so that all the cores are stressed, on average, at the same level with the same aging effects. We identify the average stress level of the core by means of a utilization parameter $0 \leq U \leq 1$. When a core fails, its workload is uniformly redistributed among the alive cores, until the cores reach an utilization $U = 1$. Further core failures cause the multi-core system-on-chip to operate at a degraded level.

The alternation between the W and I states in each core is tuned to the rotation cycle-time at which the different cores are activated in the chip. Since the transition rates $\delta(v, \mathbf{\Pi_V})$ and $\gamma(v, \mathbf{\Pi_V})$ that determine the alternation cycle between W and I are, in any case, several orders of magnitude faster than the failure rates $\mu_W(v, \mathbf{\Pi_V})$ and $\mu_I(v, \mathbf{\Pi_V})$ from W and I to F, respectively, we can assume that the states W and I reach their steady state value at the time scale of the core failure [2]. Hence:

$$U = \frac{\frac{1}{\gamma(v,\mathbf{\Pi_V})}}{\frac{1}{\delta(v,\mathbf{\Pi_V})} + \frac{1}{\gamma(v,\mathbf{\Pi_V})}} = \frac{\delta(v,\mathbf{\Pi_V})}{\delta(v,\mathbf{\Pi_V}) + \gamma(v,\mathbf{\Pi_V})} \quad (2)$$

As the time proceeds, the average number of still alive cores $n_{al}(t)$ at time t can be calculated from:

$$n_{al}(t) = n_0 - \sum_{i=1}^{n_0} \pi_F(t; v_i)$$

where n_0 is the initial number of cores in the multi-core system, v_i is the position of the core of index i and $\pi_F(t; v_i)$ is the failure probability of the core in position v_i at time t. To preserve the global workload of the chip, the utilization of the cores increases in time according with the following law:

$$U(t) = \frac{n_0}{n_{al}(t)} U_0 \quad (3)$$

where U_0 is the initial utilization of the system with n_0 cores. We assume a fixed mean active-idle cycle length L, with:

$$L = \frac{1}{\delta(v,\mathbf{\Pi_V})} + \frac{1}{\gamma(v,\mathbf{\Pi_V})} = \frac{\delta(v,\mathbf{\Pi_V}) + \gamma(v,\mathbf{\Pi_V})}{\delta(v,\mathbf{\Pi_V})\,\gamma(v,\mathbf{\Pi_V})} \quad (4)$$

Equation (4), combined with Equations (3) and (2), allows us to evaluate how the transition rates from W to I change in time:

$$\begin{cases} \gamma(v,\mathbf{\Pi_V}) &= \dfrac{1}{L\,U(t)} \\ \delta(v,\mathbf{\Pi_V}) &= \dfrac{1}{L\,(1-U(t))} \end{cases} \quad (5)$$

3.1 Aging model

Failure rates $\mu_W(v, \mathbf{\Pi_V})$ and $\mu_I(v, \mathbf{\Pi_V})$, and aging rate $\lambda(v, \mathbf{\Pi_V})$ depend on the temperature of the core, which in turns depends on its instantaneous power consumption. As it has been proven in many works, power consumption can be accurately approximated from the utilization, which can be directly derived from the global state of the system. In particular, we first determine the temperature of the core in each state $T_W(v, \mathbf{\Pi_V})$ and $T_I(v, \mathbf{\Pi_V})$, and then we estimate the TTF distribution conditioned to the given temperature $Y(t|T)$, where small t denotes time. However, special care must be taken since: `I`) $Y(t|T)$ is not an exponential distribution, and `II`) $Y(t|T)$ changes with the temperature (and in turn with time). In other words, to be able to capture the time failure events of the system, our model should support non-homogeneous non-exponential distributions.

Given a random variable Y, characterized by a probability density function (pdf) $f_Y(t)$ and Cumulative distribution function (Cdf) $F_Y(t)$, its hazard rate $h_Y(t)$ and accumulated hazard rate $H_Y(t)$ are defined, respectively, as:

$$h_Y(t) = \frac{f_Y(t)}{1 - F_Y(t)} \quad H_Y(t) = \int_0^t h_Y(u)du \quad (6)$$

Following the supplementary variables approach from Cox [7], if the process starts at time $t = 0$, then we can model its non-exponential behavior by a continuous variable Markov process, where at each point in time t, the non-exponential event modeled by Y occurs at rate[1] $\lambda_Y = h_Y(t)$. Let us call A the accumulated age of the process:

$$A = H_Y(t) = \int_0^t h_Y(u)du \quad (7)$$

Since $h_Y(t)$ is a strictly positive function, $H_Y(t)$ is invertible. This allows us to express the failure rate as function of the accumulated age A:

$$\lambda_Y(A) = h_Y\left(H_Y^{-1}(A)\right) \quad (8)$$

Moreover, the evolution of the accumulated age can be expressed in a differential way:

$$\frac{dA}{dt} = \frac{dH_Y}{dt} = h_Y(t) = h_Y\left(H_Y^{-1}(A)\right) = \lambda_Y(A) \quad (9)$$

[1] To simplify the presentation, in this section we have deliberately not included all the dependencies in the variable: for example, we have written $\lambda_Y = h_Y(t)$ instead of $\lambda_Y(t) = h_Y(t)$

It has been proved that when we have changes in the TTF distribution, the *reliability conserves* [16], that is, if at time t', the TTF distribution changes from Y to Y', we must have that

$$R_Y(t') = R_{Y'}(t') \tag{10}$$

This is in general obtained by "time shifting" the hazard rate of the new distribution Y' to match the previously accumulated aging. However, in our model we have that:

$$R_Y(t') = e^{-H_Y(t')} = e^{-A} = e^{-H_{Y'}(t')} = R_{Y'}(t') \tag{11}$$

since A does not depend on t or t'. In other words, the change of variable from t to A that allows us to express the hazard rate as function of the aging also permits to change directly the TTF distribution from Y to Y' without extra steps by automatically guaranteeing the conservation of reliability.

In the special case of the Weibull distribution, characterized by scale parameter α and shape parameter β we have:

$$f_W(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^{\beta}} \; F_W(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}} \tag{12}$$

$$h_W(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \quad H_W(t) = \left(\frac{t}{\alpha}\right)^{\beta} \tag{13}$$

$$H_W^{-1}(A) = \alpha A^{\frac{1}{\beta}} \; \lambda_W(A) = \frac{\beta}{\alpha} A^{\frac{\beta-1}{\beta}} \tag{14}$$

3.2 Thermal model

The temperature of each core is influenced by its physical location, by its operating mode (either W or I) and by the heat exchange with the neighboring cores. To determine this dependency in a realistic way, the system architecture has been characterized by a set of commonly used state-of-the-art tools. The single processing core within each many-core chip has been based on the specifications of the *Niagara2 in-order core* obtained from McPAT [17]; the core has a 0.629mm x 0.629mm area. Physical scaling parameters were extracted from the Lumos framework [23] by considering the 16nm CMOS technology node; idle and active power consumption are approximately 0.5W and 2.2W, respectively. Each chip contains is supposed to contain from 4×4 to 12×12 cores and Thermal Design Power (TDP) was set to 100W; it is worth mentioning that for the sake of simplicity memories and other control/communication modules were not considered. Hotspot [20] was used to analyze the thermal characteristics of the single chip and derive a high-level steady-state temperature model; default configuration of the tool has been adopted. Only the influences of the first eight neighbors - on the four sides and on the four corners - of a core have been considered, and a simple linear regression model has been derived. In particular let us call $\mathcal{W}$ a binary matrix whose element $w_{i,j}$ in position (i,j) is 1 if the core in position (i,j) is in the W state, and 0 otherwise. The temperature $T_c(i,j)$ of a core in position (i,j) is approximated as:

$$\begin{aligned} T_c(i,j) &= c_0 + c_1 \cdot w_{i,j} + \\ &+ c_2 \cdot (w_{i+1,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1}) + \\ &+ c_3 \cdot (w_{i+1,j+1} + w_{i+1,j-1} + w_{i-1,j+1} + w_{i-1,j-1}) \end{aligned} \tag{15}$$

where $c_0 \dots c_1$ are four thermal constants. In this work, we have considered $c_0 = 329$ K, $c_1 = 16$ K, $c_2 = 3$ K, and $c_3 = 1.5$ K.

3.3 Failure time distribution model

Finally, for a proof of concept, the device aging has been characterized as in [3] by considering the electromigration mechanism. In particular, aging is supposed to follow a Weibull distribution, with parameters $\beta = 2$ and $\alpha_{EM}(T)$ is computed as:

$$\alpha_{EM}(T) = \frac{A_{EM}(J - J_{\text{crit}})^{-n} e^{\frac{E_{a_{EM}}}{kT}}}{\Gamma\left(1 + \frac{1}{\beta}\right)} \tag{16}$$

where A_{EM} is a material-dependent constant, J and J_{crit} are the current density and its critical value activating the phenomenon, respectively, n is empirically determined constant, $E_{a_{EM}}$ is the activation energy, K is Boltzman's constant, Γ is the Gamma function, and T is the constant worst-case temperature in Kelvin degrees. In particular, as in [3], $E_{a_{EM}} = 0.48$eV, $K = 8.61673324 \cdot 10^{-5}$eV/K and $n = 1.1$. Moreover, other parameters have been determined in order to have approximately an MTTF= 10 years for a single core working at the constant steady-state temperature of 60°C; thus, $J = 1.5 \cdot 10^6 \text{A/cm}^2$, $J_{\text{crit}} \cong 0\text{A/cm}^2$, $A_{EM} = 3 \cdot 10^5 (\text{h/cm}^2) \cdot (\text{A/cm}^2)^n$.

3.4 Complexity reduction

As seen in Section 3.2 the temperature of a core depends on its working state, and on the working states of its neighbors. Following Section 3.3, a different temperature involves a different failure time distribution. Whenever a component switches from W to I its temperature, and thus its failure rate, changes. Figure 3a and 3b show the evolution of the reliability function supposing that the system performs respectively 10 and 1 switches from I to W in a time frame of 120 months. To emphasize the scenario, we have considered that $T_I = 318$ K and $T_W = 353$ K, for which the reliability at a constant temperature is shown in Figure 3d and 3f, respectively. The time distribution that would be obtained considering the average temperature of the cores $T_A = (T_I + T_W)/2$ is shown in Figure 3e: as it can be seen it is quite different from the one that could be obtained by considering a non-homogenous model where failure time distribution changes according to the state of the system. Instead, by letting the number of switches tend to infinite, we can characterize the aging rate as the average of the aging rates in the two states:

$$\lambda(A) = \frac{\beta}{2} A^{\frac{\beta-1}{\beta}} \cdot \left(\frac{1}{\alpha_{EM}(T_I)} + \frac{1}{\alpha_{EM}(T_W)}\right) \tag{17}$$

The results of the definition given in Equation 17 are shown in Figure 3c: as it can be seen, in this case, the failure time distribution closely matches the TTF of the systems that alternates between the states.

Since the temperature depends on the state of 9 cores, the previous results should be extended appropriately. Let us assume that we are considering a 3×3 chip and focus on its

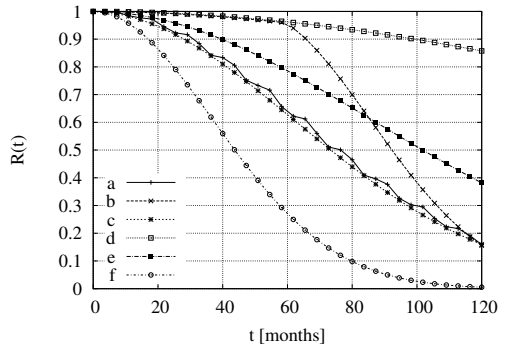


Figure 3: Failure time distribution of a system switching from I to W: a) 10 switches, b) 1 switch, c) the average failure rate model, d) always in I state, e) the average temperature model, f) always in W state

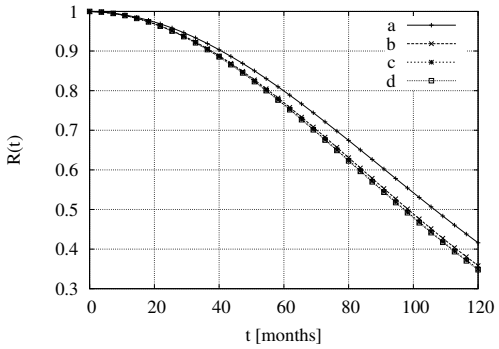


Figure 4: Different approximations of the W and I failure rates: a) the average temperature; b) the W and I states for the local core and the average effect of the neighbours; c) the W and I states for the local core and the ones on the sides, plus the average effects of the neighbours on the corner; and d) the W and I states for all the cores

central core. In particular, if we suppose that the W and I states of neighbor cores are uncorrelated, we can express the probability of being in a given configuration $\mathcal{W}$ as:

$$\pi_{\mathcal{W}}(t) = \prod_{i=1}^{3}\prod_{j=1}^{3}\Big[w_{i,j}\pi_W(t;v_{i,j}) + \big(1-w_{i,j}\big)\big(1-\pi_W(t;v_{i,j})\big)\Big] \tag{18}$$

Let us call $2^{\mathcal{W}}$ the 512 elements set containing all the possible configurations of I and W states for the 9 neighbors cores. The aging rate can then be expressed as:

$$\lambda(A) = \beta A^{\frac{\beta-1}{\beta}} \sum_{\mathcal{W}\in 2^{\mathcal{W}}} \frac{\pi_{\mathcal{W}}}{\alpha_{EM}(T(\mathcal{W}))} \tag{19}$$

where $T(\mathcal{W})$ is Equation 15 is computed with the configuration given in $\mathcal{W}$. Although simple to write, the computation of Equation 19 is quite complex to compute, since it requires more than 4500 iterations which must be repeated for every agent. We have thus resorted to a simplified and approximated model. In particular, we consider the W and I state separately for the local core, but then we consider the average temperature of the neighbor cores. Figure 4 shows the effect of considering: a) the average temperature; b) the W and I states for the local core and the average effect of the neighbors; c) the W and I states for the local core and the ones on the sides, plus the average effects of the neighbours on the corner; and d) the W and I states for all the cores (Equation 19). As it can be seen, the differences between cases 4b, 4c, and 4d are minimal, even if the complexity of 4b is just marginally larger than the computation of the average temperature as in 4a.

To summarize, let us define sum of the utilization of the side ($U_S(i,j)$) and corner ($U_C(i,j)$) neighbors as:

$$\begin{aligned} U_S(i,j) &= \pi_W(t;v_{i+1,j}) + \pi_W(t;v_{i-1,j}) + \\ &+ \pi_W(t;v_{i,j+1}) + \pi_W(t;v_{i,j-1}) \\ U_C(i,j) &= \pi_W(t;v_{i+1,j+1}) + \pi_W(t;v_{i+1,j-1}) + \\ &+ \pi_W(t;v_{i-1,j+1}) + \pi_W(t;v_{i-1,j-1}) \end{aligned}$$

We approximate the $T_W(i,j)$ and $T_I(i,j)$ temperatures in the W and I states as:

$$\begin{aligned} T_W(i,j) &= c_0 + c_2\cdot U_S(i,j) + c_3\cdot U_C(i,j) \\ T_I(i,j) &= c_0 + c_1 + c_2\cdot U_S(i,j) + c_3\cdot U_C(i,j) \end{aligned}$$

And finally, we define $\mu_W(v_{i,j},\boldsymbol{\Pi_V})$, $\mu_I(v_{i,j},\boldsymbol{\Pi_V})$ and $\lambda(v_{i,j},\boldsymbol{\Pi_V})$ as:

$$\begin{aligned} \mu_W(v_{i,j},\boldsymbol{\Pi_V}) &= \big(\pi_A(t;v_{i,j})\big)^{\frac{\beta-1}{\beta}} \frac{\beta}{\alpha_{EM}\big(T_W(i,j)\big)} \\ \mu_I(v_{i,j},\boldsymbol{\Pi_V}) &= \big(\pi_A(t;v_{i,j})\big)^{\frac{\beta-1}{\beta}} \frac{\beta}{\alpha_{EM}\big(T_I(i,j)\big)} \\ \lambda(v_{i,j},\boldsymbol{\Pi_V}) &= \mu_W(v_{i,j},\boldsymbol{\Pi_V})\pi_W(t;v_{i,j}) \\ &+ \mu_I(v_{i,j},\boldsymbol{\Pi_V})\big(1-\pi_W(t;v_{i,j})\big) \end{aligned}$$

Basically, the age accumulates following Equation 17 where the temperature considers the effect of the local W and I states, plus the average contribution of the neighbors. However, the failure rates in the two W and I states, consider only the failure time distribution specific for the corresponding operational temperature.

4 RESULTS

Let us assume that the system initially has n_0 cores, each of them with an utilization of U_0 as introduced in Section 3. When a core fails, the workload distribution policy preserves the total system workload of n_0U_0 by increasing the utilization of the remaining cores. However, when all cores reach the full utilization, further failures lead the system to a degraded operating state. In order to investigate such process, we can define the workload $Wl(t)$ at time instant t as:

$$Wl(t) = \sum_{v} \pi_W(t;v), \tag{20}$$

i.e. the processing capacity of the CPU that corresponds to the sum of the utilisations of the cores. Then, we define the

mean time to degradation $MTTD$ of the system as the first time instant at which the global workload goes below n_0U_0. It can be computed as:

$$MTTD_{sys} = \inf\{t \in [0, +\infty] : Wl(t) < n_0U_0\}, \quad (21)$$

Such performance index allows us to evaluate the effectiveness of the workload distribution policy in extending the fully operational phase of the multi-core system. We have carried on the experiments on a personal computer with an Intel Core i5-2450M CPU at 2.50GHz and 6 GB RAM, all experiments take a few minutes to be completed.

4.1 Scenario 1

We start considering a simple scenario with a uniform distribution of the workload over a system with 16 cores and an initial per-core utilization of $U_0 = 0.95\%$, for a total workload $\Theta = 16 \cdot 0.95 \simeq 15$. Due to the high utilization level of each core, in such a case a short fully operational phase is expected.

However, the same amount of workload can be performed by systems with a larger number of cores reducing their stress level. Indeed, for an architecture with n cores, we can recompute the per-core utilization required to elaborate the workload Θ as $U_0^n = \Theta/n$. For instance, the per-core utilization required for a system with 36 cores will be reduced to $U_0^{36} = 0.416\%$. Figure 5 shows in detail the behavior of a core of such system. After an initial short transient phase, the core reaches its fully operational condition working at the target utilization U_0^{36}. With the passage of time, the failure probability increases thus, to preserve the desired workload, the core must increase its utilization until the value of 1 is reached. After such time, further failures lead the system to work in a degraded condition.

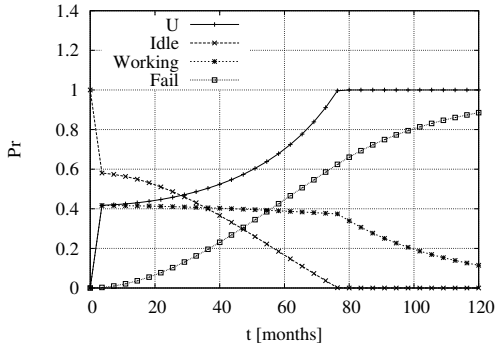


Figure 5: The utilization and state probabilities evolution in a system with 36 cores.

Figure 6 shows the behavior of the workload normalized with respect to Θ for the 16, 36 and 64 architectures. As expected, we can observe how spreading the same constant workload of the 16 cores chip over architectures with a greater number of cores increases the system $MTTD$. In particular, the 16 core chip on average stay in full operating condition for about one year, whereas the 36 CPU lasts slightly more than six years and the 64 chip more than ten years.

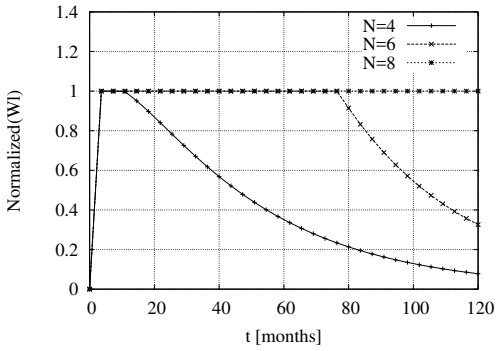


Figure 6: The global workload trend for a chip with 16, 36 and 64 cores.

4.2 Scenario 2

In the second scenario, we consider that cores are partitioned into two sub-classes: the one starts initially active, and the one that starts initially spare. We focus on a 12×12 CPU with 144 cores and we study two different workload redistribution policies:

P_1: the workload lost due to the failures of the primary cores is evenly redistributed to the spare ones only, increasing their utilization level.

P_2: the workload lost is evenly redistributed to the spare ones until the workloads of the primary and spare cores are balanced. Then, the lost workload will be redistributed evenly on all the remaining cores.

We also consider different spatial distributions of the initially active/spare cores, according to the patterns shown in Figure 7.

Results for two different target utilization levels (of the initially active cores) $U_0 = 0.6$ and $U_0 = 0.9$ for the considered policies P_1 and P_2 is shown in Figure 8. First, we can observe that the two policies obtain very similar results, even if P_2 performs slightly better. The pattern distribution has instead a more sensible impact: best results are obtained when the initially working cores are concentrated in a corner of the CPU. It is interesting to note that its dual - when the cores initially active are distributed over an L-shaped pattern, has the worst performances.

To better show the evolution of the system, Figure 9 presents the thermal distribution of the temperature over the chip at three different time instants for policy P_1, target $U = 0.9\%$ and initial core distribution $B3$ of Figure 7. At the beginning of the life of the system, the active cores, which are concentrated in the middle band, are working at a high temperature, while the outer bands are colder. When the MTTD is reached at around five years (Figure 9b), the CPU works at a lower temperature, and the pattern is inverted: indeed the spare cores at the beginning now have to take the

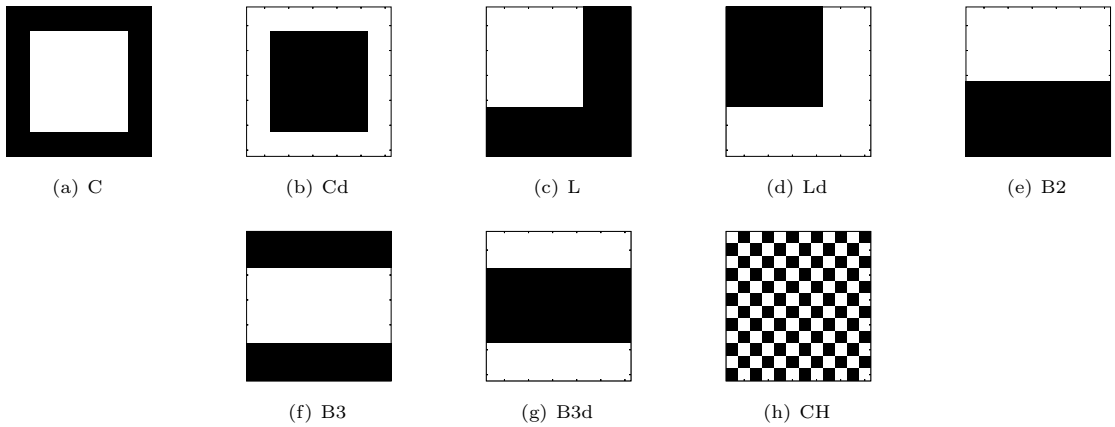


Figure 7: Primary vs spare core patterns. The primary cores are white, the spare black.

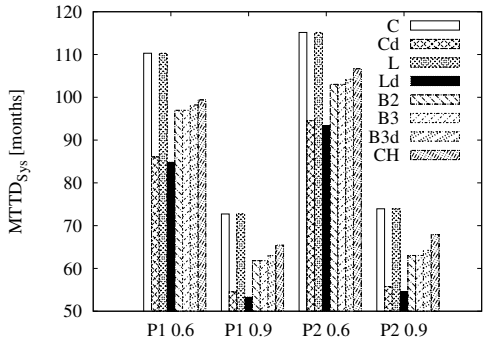


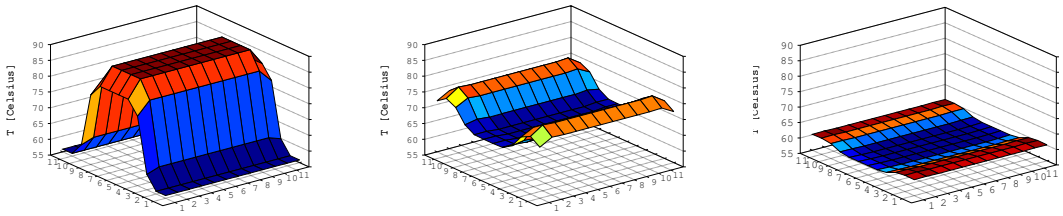
Figure 8: Comparison of the system MTTD achieved by different core patterns varying the load redistribution policy and the initial primary core utilization.

place of the failed ones. After ten years, due to the distribution of the failures over the CPU, the differences between the temperature of the hot and cold area are reduced, and the surviving cores work at a much lower average temperature.

5 RELATED WORKS

System level models for the various aging mechanisms have been widely investigated in the literature. JEDEC Solid State Technology Division has defined the industry standard [14] used for performing accelerated post-production testing to estimate the lifetime of a class of components. Such models generally consider the chip as a single unit, thus considered damaged after the first failure, and assume a single steady-state operation mode at fixed (worst-case) temperature. These two aspects represent a considerable limitation in the early estimation of the lifetime of a modern computing system being composed of various dozens of cores and subject of highly variable workloads. Our proposal allows considering the temporal evolution of the system, providing a more detailed analysis than the worst case. Authors of [21] proposed one of the first works performing a system-level estimation of the lifetime of a single core system. The work combines the aging models defined in [14] with the sum-of-failure-rates (SOFR) approach to take into account several aging mechanisms. Its main limitations are the fact that they consider single-core chips, that is a quite old scenario, and, moreover, it adopts an exponential distribution to model the fault occurrence, that not accurately fits the real distribution of faults. Various subsequent works [6, 15] suffer from the same limitations. Our work instead aims at considering a large number of cores to tackle possible future scenario where the parallelism of a CPU will grow dramatically.

Subsequent works improved the aging model by considering Weibull or lognormal distributions to consider the aging history, as in [11, 24]. In order to make computationally feasible the solution of such model, such works adopted an approach based on the simulation of only a subset of representative workload traces for a reduced period of time with a fine granularity, then extrapolating the average failure rate [11, 24] to be considered during the MTTF computation. Here we presented a novel way to include aging, factorized when possible to avoid the explosion of the computational complexity of the model, but preserved separately for the single cores. System level lifetime reliability models have to take into account also the fact that multi-core systems are actually K-out-of-N:F systems, i.e., the system may tolerate a given number of F failures by remapping the workload from the failed cores to the healthy ones. Such model is based on multiple integrals [18], whose dimension is given by the number of failures the system can tolerate. In practice, its analytical solution is unfeasible. Thus, many approaches [11, 15] consider the entire system not to survive beyond the first failure, while some other one [8] computes the MTTF iteratively by summing at each step the lowest MTTF of the various cores. Both solutions are inaccurate. In this paper we have followed a different approach: instead of considering a K-out-of-N:F system, we have considered to adapt the achitecture in order to provide a target workload. Finally, other numerical



(a) One hour. (b) Five years. (c) Ten years.

Figure 9: Average core temperature in $B3$ configuration at: a) $t = 1$ hour, b) $t = 5$ years and c) $t = 10$ years.

approaches [3, 10, 12, 24] adopt Monte Carlo simulation or the multi-armed bandits approach [19] to quickly estimate the multiple integrals required to study the time to failure in an analytical way. Such technique will become computationally expensive when considering a temporal redistribution of the workload to maintain a target service level: the MAM approach instead provides reasonably accurate solutions in reduced computation times.

6 CONCLUSIONS

In this work, we have presented a MAM to study the reliability of a multi-core CPU. With the proposed technique, we were able to study CPUs composed by a large number of cores, and consider complex non-homogeneous non-exponential time to failure distributions. Future works will focus on the applications, trying to remove the simplifying assumption used both in the thermal and in the aging models.

ACKNOWLEDGMENTS

This work is original and has been partially supported by EUBra-BIGSEA (690116), a Research and Innovation Action (RIA) funded by the European Commission under the Cooperation Programme, Horizon 2020 and the Ministrio de Cincia, Tecnologia e Inovao (MCTI), RNP/Brazil (grant GA-0000000650/04) and has a financial support of the Università del Piemonte Orientale.

REFERENCES

[1] A. Bobbio, D. Cerotti, M. Gribaudo, M. Iacono, and D. Manini. Markovian agent models: A dynamic population of interdependent markovian agents. In E. K. Al-Begain and A. B. (Eds.), editors, *Seminal Contrib. to Modelling and Simulation*, pages 185–203. Springer, 2016.

[2] A. Bobbio and K. Trivedi. An aggregation technique for the transient analysis of stiff Markov chains. *IEEE Trans. on Computers*, C-35:803–814, 1986.

[3] C. Bolchini, M. Carminati, M. Gribaudo, and A. Miele. A lightweight and open-source framework for the lifetime estimation of multicore systems. In *Proc. Int. Conf. Computer Design*, pages 166–172, 2014.

[4] D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo. Markovian agent modeling swarm intelligence algorithms in wireless sensor networks. *Performance Evaluation*, 69:135–149, 2012.

[5] D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo. An intelligent swarm of markovian agents. In J. Kacprzyk and W. Pedrycz, editors, *Springer Handbook of Comp. Intelligence*, pages 1345–1359. Springer Berlin Heidelberg, 2015.

[6] A. Coskun, T. Simunic, K. Mihic, G. D. Micheli, and Y. Leblebici. Analysis and optimization of MPSoC reliability. *J. Low Power Electr.*, pages 56–69, 2006.

[7] D. R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. *Math. Proc. of the Cambridge Philosophical Society*, 51(3):433–441, 1955.

[8] Z. Gu, C. Zhu, L. Shang, and R. P. Dick. Application specific MPSoC reliability optimization. *Trans. VLSI Systems*, 16(5):603–608, 2008.

[9] M. H. Haghbayan, A. M. Rahmani, A. Y. Weldezion, P. Liljeberg, J. Plosila, A. Jantsch, and H. Tenhunen. Dark silicon aware power management for manycore systems under dynamic workloads. In *IEEE 32nd Int. Conf. on Computer Design*, pages 509–512, Oct 2014.

[10] A. Hartman and D. Thomas. Lifetime improvement through runtime wear-based task mapping. In *Int. Conf. Hardware/software codesign and system synthesis*, pages 13–22, 2012.

[11] L. Huang and Q. Xu. AgeSim: A simulation framework for evaluating the lifetime reliability of processor-based SoCs. In *Conf. Design Autom. & Test in Europe*, pages 51–56, 2010.

[12] L. Huang and Q. Xu. Lifetime reliability for load-sharing redundant systems with arbitrary failure distributions. *Trans. Reliability*, 59(2):319–330, 2010.

[13] ITRS. Int. Tech. Roadmap for Semiconductors. http://www.itrs2.net/, 2011.

[14] JEDEC Solid State Tech. Ass. Failure mechanisms and models for semiconductor devices. *JEDEC Publ. JEP122G*, 2010.

[15] E. Karl, D. Blaauw, D. Sylvester, and T. Mudge. Multi-Mechanism Reliability Modeling and Management in Dynamic Systems. *Trans. on VLSI Systems*, 16(4):476–487, 2008.

[16] D. Kececioglu. *Reliab. Eng. Handbook (Vol. 1)*. Prentice-Hall, Upper Saddle River, NJ, USA, 1991.

[17] S. Li, J. H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proc. Int. Symp. on Microarchitecture (MICRO)*, pages 469–480, 2009.

[18] H. Liu. Reliability of a load-sharing k-out-of-n:G system: non-iid components with arbitrary distributions. *Trans. Reliability*, 47(3):279–284, 1998.

[19] C. Ma, A. Mahajan, and B. H. Meyer. Multi-armed bandits for efficient lifetime estimation in MPSoC design. In *Proc. of Design, Automation Test in Europe Conf.*, pages 1540–1545, 2017.

[20] K. Skadron, M. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, pages 94–125, 2004.

[21] J. Srinivasan, S. Adve, P. Bose, and J.A.Rivers. The case for lifetime reliability-aware microprocessors. In *Int. Symp. Comp. Arch.*, pages 276–287, 2004.

[22] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. The impact of technology scaling on lifetime reliability. In *Int. Conf. on Dependable Systems and Networks*, pages 177–186, June 2004.

[23] L. Wang and K. Skadron. Dark vs. Dim Silicon and Near-Threshold Computing Extended Results. In *Univ. of Virginia Dep. of Comp. Science Tech. Rep. TR-2013-01*, 2012.

[24] Y. Xiang, T. Chantem, R. Dick, X. Hu, and L. Shang. System-level reliability modeling for MPSoCs. In *Conf. Hardware/Software Codesign and System Synthesis*, pages 297–306, 2010.