

# Fault Detection and Profiling Algorithms for Exascale Computing Systems

Nageswara S. V. Rao

Oak Ridge National Laboratory, raons@ornl.gov

## I. INTRODUCTION

Exascale computing systems are expected to consist of millions of components, and the current engineering and manufacturing practices cannot guarantee their fault-free operation during code executions lasting for hours. It is important to detect the faults as they occur, and also to develop their statistical profiles to support: i) fault-tolerant applications in setting their parameters, such as replication and check-point levels, and ii) facility operations in removing the faulty processors from scheduler pools and physically replacing the failed nodes. Such approaches have been used for fault detection in processors and digital circuits [2]. However, the exascale systems require a new class of fault detection and profiling algorithms that are optimized to their architecture, scale and component failure statistics, since a deep fault coverage of even a single processor is NP-hard. Furthermore, due to the inherent stochastic nature of the component failures in exascale systems, outputs of such algorithms are non-deterministic and must be quantified with statistical confidences. The overall goal here is to detect failures using fast algorithms and utilize their outputs to build robust statistical state estimates and profiles.

The fault detection and profiling computations constitute a special class of algorithms, and are much different from application computations: they “seek out” failures whereas the latter attempt to avoid or account for them. In this white paper, we outline a novel detection and profiling approach for the exascale systems based on fault detection using chaotic maps and state estimation using finite sample statistics. A complete development of this approach requires mathematical advances in a number of areas, including Lyapunov exponent design for fast diagnosis, optimal pipelining methods for fault coverage, and the design of statistical probes for robust profile estimates.

## II. CLASS OF PROFILING ALGORITHMS

Let  $\mathcal{S}_i, i = 0, 1, \dots$ , denote the (random) state of an exascale system at time step  $i$ ; it is a large vector representing components such as processor cores, communication links or memory elements. Let  $\mathcal{S} \odot \mathcal{A}(I)$  denote the output of algorithm  $\mathcal{A}$  executed on the system in state  $\mathcal{S}$  with input  $I$ . A profiling algorithm  $\mathcal{P}$  is specifically designed to estimate  $\hat{\mathcal{S}}_i = \mathcal{S}_i \odot \mathcal{P}(I_{\mathcal{P}})$  of  $\mathcal{S}_i$ . From an application perspective, its objectives is to support (a) design of customized application algorithms  $\mathcal{A}_{\hat{\mathcal{S}}_i}$  optimized to  $\hat{\mathcal{S}}_i$ , and (b) estimation of confidence measures for the output  $\mathcal{S}_i \odot \mathcal{A}_{\hat{\mathcal{S}}_i}(I)$ .

### A. Fault Detection Using Chaotic Maps

We propose a detection approach that performs the same computation on all nodes and compares their outputs by exploiting two factors: (i) the computations can be performed in

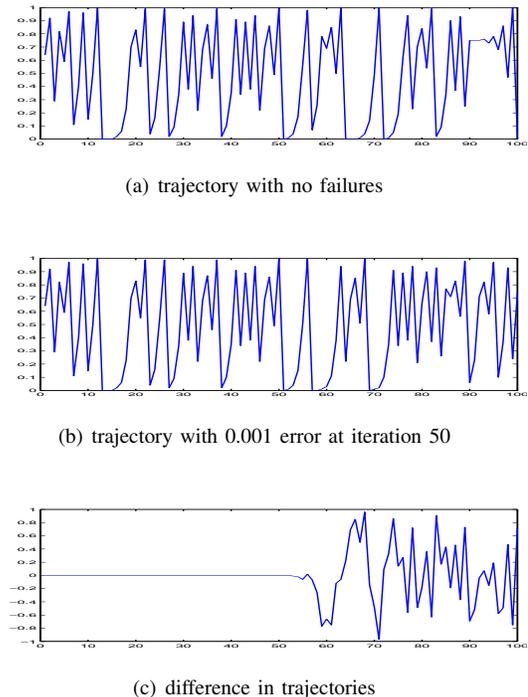


Fig. 1. Chaotic map trajectories.

parallel, and (ii) only a few (non-majority) of the components fail during computations. If there are no faults, all computation outputs are the same; otherwise, the nodes with failures can be determined by their deviations from the majority.

An efficient detection of such deviations is critical to this approach and can be carried out by utilizing chaotic maps [3]. A *Poincare map*  $M : \mathbb{R}^d \mapsto \mathbb{R}^d$  specifies the trajectory, of a real-vector state  $X_i \in \mathbb{R}^d$  that is updated at each iteration  $i$  such that  $X_{i+1} = M(X_i)$  [1]. These trajectories may exhibit complex profiles, even when  $M$  is computationally simple. In Figure 1, we show the trajectories of the logistic map  $M_{L_a}(X) = aX(1 - X)$ , which requires two multiplications and one subtraction; they exhibit chaotic dynamics for  $a = 4$  as shown in Figure 1(a). The trajectories of a chaotic Poincare map exhibit the *exponential divergence*, which implies that trajectories that start from states that are only slightly different from each other rapidly diverge in a few steps, as shown in Figure 1(a) and (b). This property is utilized as a mechanism to rapidly amplify the errors in computations caused by factors such as bit flip in memory elements or stuck-at fault in an Arithmetic and Logic Unit (ALU) operation. Two chaotic trajectories computed on two different processor cores,  $P_{s_1}$  and  $P_{s_2}$  are identical if no fault occurs in either. But, a fault

in one will lead to quick divergence of the outputs, leading to different  $X_n$  values, which are detectable for different values of  $n$  based on  $M(\cdot)$  and type of fault. The faults that could lead to trajectory divergence range from those in arithmetic and logical operations performed by the ALU, or faults in registers and memory, but are limited to the operations used in the computation of  $M(\cdot)$ . By prepending chaotic map updates with mathematical operations, other ALU failure can be detected. Similarly, assignment operations with memory allocated in different hierarchies can be used to diagnose memory elements, memory bus and interconnects such as hypertransport [3].

These detection strategies can be combined into chains that execute various operations to catch faults, and utilize chaotic maps to amplify the difference for quick detection [3]. By pipelining these chains and strategically guiding their executions, errors in various parts of the system can be detected quickly. The exact fault detection times depend on the *Lyapunov exponent* defined as  $\mathcal{L}_M = \ln \left| \frac{dM}{dX} \right|$  and the nature of faults. These works show the basic feasibility of the chaotic map approach for fault detection, but a detailed analysis of the errors and Lyapunov exponents is essential to assess the performance of this method. Rigorous analysis and design methods must be developed to optimize the diagnosis pipelines and Lyapunov exponents, which could lead to a more general theory of diagnosability of exascale systems.

### B. Performance of Profiling Algorithms

We propose a general framework where the profiling algorithm is provided an input vector  $I_{\mathcal{P}}$  to guide the probing of components that are prone to more frequent errors by a suitable choice of the probability distribution  $\mathbf{P}_{I_{\mathcal{P}}}$ . The size of  $I_{\mathcal{P}}$  is denoted by  $|I_{\mathcal{P}}|$  such that a larger size involves testing more components or running longer and is more likely to detect errors in more components. The error of the state estimate  $\hat{S}_i(I_{\mathcal{P}}) = \mathcal{S}_i \odot \mathcal{P}(I_{\mathcal{P}})$ , is given by

$$\mathcal{E}(\mathcal{S}_i, \mathcal{P}, I_{\mathcal{P}}) = \|\mathcal{S}_i - \hat{S}_i\|_{\mathcal{P}, I_{\mathcal{P}}} = (\mathcal{S}_i \odot \mathcal{P}(I_{\mathcal{P}})) \otimes \mathcal{S}_i,$$

where  $\|\cdot\|_{\mathcal{P}, I_{\mathcal{P}}}$  represents the error between the predicted and actual state, and  $\otimes$  is its representation in the output space. The expected errors of  $\mathcal{P}$  are

$$\bar{\mathcal{E}}_{\mathcal{S}}(\mathcal{P}, I_{\mathcal{P}}) = \int \mathcal{E}(\mathcal{S}_i, \mathcal{P}, I_{\mathcal{P}}) d\mathbf{P}_{\mathcal{S}_i}$$

$$\bar{\mathcal{E}}(\mathcal{P}) = \int \mathcal{E}(\mathcal{P}, I_{\mathcal{P}}) d\mathbf{P}_{I_{\mathcal{P}}}.$$

We are also interested in estimating the performance of the profiling algorithm for the state  $\mathcal{S}_i$  given by

$$\bar{\mathcal{E}}(\mathcal{S}_i, \mathcal{P}) = \int (\hat{\mathcal{S}}_i \otimes \mathcal{S}_i) d\mathbf{P}_{I_{\mathcal{P}}}.$$

Then we have

$$\bar{\mathcal{E}}(\mathcal{P}) = \int \bar{\mathcal{E}}(\mathcal{S}_i, \mathcal{P}) d\mathbf{P}_{\mathcal{S}_i} = \int \left[ \int_{I_{\mathcal{P}}} (\hat{\mathcal{S}}_i \otimes \mathcal{S}_i) d\mathbf{P}_{I_{\mathcal{P}}|\mathcal{S}_i} \right] d\mathbf{P}_{\mathcal{S}_i},$$

which shows that the performance of profiling algorithm can be improved by customizing its input  $I_{\mathcal{P}}$  to  $\mathcal{S}_i$ . Since  $\mathbf{P}_{\mathcal{S}_i}$  depends on the error distributions of components and their correlations, it is mostly unknown and complex; for example, an overheated shelf of an exascale system will likely lead to simultaneous faults in several components housed in it.

We outline a method to estimate  $\bar{\mathcal{E}}(\mathcal{P})$  by running  $\mathcal{P}$  under different configurations with actual or induced errors. Now consider that we executed the profiling algorithm  $\mathcal{P}$  with fixed  $I_{\mathcal{P}}$  on the machine with errors and measured its output  $\hat{S}_{i,j}$  in state  $\mathcal{S}_{i,j}$ , for  $j = 1, 2, \dots, l$ . Consider the empirical error committed by  $\mathcal{P}$  given by

$$\hat{\mathcal{E}}(\mathcal{P}) = \frac{1}{l} \sum_{j=1}^l (\hat{S}_{i,j} \otimes \mathcal{S}_{i,j}).$$

We can utilize this empirical error as a measure of the performance of  $\mathcal{P}$  with the following guarantee

$$\mathbf{P} \left\{ \left| \bar{\mathcal{E}}(\mathcal{P}) - \hat{\mathcal{E}}(\mathcal{P}) \right| > \epsilon \right\} \leq 2e^{-2\epsilon^2 l},$$

which improves as more measurements are collected. This bound is based on Hoeffding's inequality, and is valid under certain statistical independence conditions.

Now consider that we record the sizes of inputs used by  $\mathcal{P}$  in different runs, given by  $I_{\mathcal{P}_1}, I_{\mathcal{P}_2}, \dots, I_{\mathcal{P}_l}$ . We fit a regression *profile function*  $r(|I|)$  to the measured  $\hat{\mathcal{E}}(\cdot)$  values that estimates the error as function of the size of input  $I$  to  $\mathcal{P}$ . We assume that  $r(\cdot)$  is a non-increasing function in that the probability of detecting errors does not degrade as  $\mathcal{P}$  tests more components. Then based on the Vapnik-Chervonenkis theory [4], we have the performance bound

$$\mathbf{P} \left\{ \max_{|I|} |r(|I|) - \bar{\mathcal{E}}(\mathcal{S}_i, I)| \right\} \leq 8\epsilon l e^{-\epsilon l/4},$$

on the performance  $r(x)$  of the profiling algorithm  $\mathcal{P}$  for input of size  $x$  (for which measurements may not be available). This performance bound is in general weaker than the above guarantee on  $\bar{\mathcal{E}}(\mathcal{P})$ , but provides us qualitative information about executing the profiling algorithm on longer inputs.

These results show the feasibility of developing profiling algorithms with probabilistic guarantees on the error for exascale systems. Rigorous methods are needed to extend this approach to take into account the complex component failures and their correlations that lead to statistical non-independence conditions. Designing the optimal inputs and their distributions for  $\mathcal{P}$  requires solutions to two classes of optimization problems:  $\max_{I_{\mathcal{P}} \in \mathcal{D}} \bar{\mathcal{E}}_{\mathcal{S}}(\mathcal{P}, I_{\mathcal{P}})$  and  $\max_{\mathbf{P}_{I_{\mathcal{P}}} \in \mathcal{P}(\mathcal{D})} \bar{\mathcal{E}}(\mathcal{P})$ ,

where (i)  $I_{\mathcal{P}}$  is optimized over the set  $\mathcal{D}$  of detection strategies that includes as elements different diagnosis pipelines, subsets of processors and interconnects, and their combinations, and (ii)  $\mathbf{P}_{I_{\mathcal{P}}}$  is optimized over a set of distributions on  $\mathcal{D}$ . The components outlined in this white paper could be parts of a more general theory of fault diagnosis and profiling algorithms for exascale computing systems.

### REFERENCES

- [1] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag Pub., Reading, MA, 1996.
- [2] N. K. Jha and S. Gupta. *Testing of Digital Systems*. Cambridge University Press, 2003.
- [3] N. S. V. Rao. Fault detection in multi-core processors using chaotic maps. In *3rd Workshop on Fault-Tolerance for HPC at Extreme Scale (FTXS 2013)*, 2013.
- [4] V. N. Vapnik. *Statistical Learning Theory*. John-Wiley and Sons, New York, 1998.