# ADAPTIVE AND POWER-AWARE FAULT TOLERANCE FOR FUTURE EXTREME-SCALE COMPUTING

by

**Xiaolong Cui**

Bachelor of Engineering

Xi'an Jiaotong University

2012

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

**Doctor of Philosophy**

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

COMPUTER SCIENCE DEPARTMENT

This proposal was presented

by

Xiaolong Cui

Dr. Taieb Znati, Department of Computer Science, with joint appointment in Telecommunication Program, University of Pittsburgh

Dr. Rami Melhem, Department of Computer Science, University of Pittsburgh

Dr. John Lange, Department of Computer Science, University of Pittsburgh

Dr. Esteban Meneses, School of Computing, Costa Rica Institute of Technology

Dissertation Advisors: Dr. Taieb Znati, Department of Computer Science, with joint appointment in Telecommunication Program, University of Pittsburgh,

Dr. Rami Melhem, Department of Computer Science, University of Pittsburgh

**ADAPTIVE AND POWER-AWARE FAULT TOLERANCE FOR FUTURE EXTREME-SCALE COMPUTING**

Xiaolong Cui, PhD

University of Pittsburgh, 2016

As the demand for computing power continue to increase, both HPC community and Cloud service provides are building larger computing platforms to take advantage of the power and economies of scale. On the HPC side, a race is underway to build the world's first exascale supercomputer to accelerate scientific discoveries, big data analytics, etc. On the Cloud side, major IT companies are all expanding large-scale datecenters, for both private usage and public services. However, aside from the benefits, several daunting challenges will appear when it comes to extreme-scale.

This thesis aims at simultaneously solving two major challenges, i.e., power consumption and fault tolerance, for future extreme-scale computing systems. We come up with a novel power-aware computational model, referred to as Lazy Shadowing, to achieve high-levels of resilience in extreme-scale and failure-prone computing environments. Different approaches have been studied to apply this model. Accordingly, precise analytical models and optimization framework have been developed to quantify and optimize the performance, respectively.

In this work, I propose to continue the research in two aspects. Firstly, I propose to develop a MPI-based prototype to validate Lazy Shadowing in real environment. Using the prototype, I will run benchmarks and real applications to measure its performance and compare to state-of-the-art approaches. Then, I propose to further explore the potential of Lazy Shadowing and improve its efficiency. Based on the specific system configuration, application characteristics, and QoS requirement, I will study the impact of process mapping the viability of partial shadowing.

## TABLE OF CONTENTS

# 1.0 INTRODUCTION

As our reliance on IT continues to increase, the complexity and urgency of the problems our society will face in the future drives us to build more powerful and accessible computer systems. Among the different types of computer systems, High Performance Computing (HPC) and Cloud Computing systems are the two most powerful ones. For both of them, the computing power attributes to the massive amount of parallelism, which is supported by the massive amount of computing cores, memory modules, communication devices, storage components, etc.

Since CPU frequency flattens out in early 2000s, parallelism has become the "golden rule" to boost performance. In HPC, Terascale performance was achieved in the late 90s with fewer than 10,000 heavyweight single-core processors. A decade later, petascale performance required about ten times processors with multiple cores on each processor. Nowadays, a race is underway to build the world's first exascale machine to accelerate scientific discoveries and breakthroughs. It is projected that an exascale machine will achieve billion-way parallelism by using one million sockets each supporting 1,000 cores.

Similar trend is happening in Cloud Computing. As the demand for Cloud Computing accelerates, cloud service providers will be faced with the need to expand their underlying infrastructure to ensure the expected levels of performance, reliability and cost-effectiveness. As a result, lots of large-scale data centers have been and are being built by IT companies to exploit the power and econies of scale. For example, Microsoft, Google, Facebook, and Rackspace have hundreds of thousands of web servers in dedicated data centers to support their business.

Unfortunately, several challenging issues come with the increase in system scale. As today's HPC and Cloud Computing systems grow to meet tomorrow's compute power demand,

the behavior of the systems will be increasingly difficult to specify, predict and manage. This upward trend, in terms of scale and complexity, has a direct negative effect on the overall system reliability. Even with the expected improvement in the reliability of future computing technology, the rate of system level failures will dramatically increase with the number of components, possibly by several orders of magnitude. At the same time, the rapid growing power consumption, as a result of the increase in system components, is another major concern. At future extreme-scale, failure would become a norm rather than an exception, driving the system to significantly lower efficiency with unprecedented amount of power consumption.

## 1.1 PROBLEM STATEMENT

The system scale needed to address our future computing needs will come at the cost of increasing complexity, unpredictability, and operating expenses. As we approach future extreme-scale, two of the biggest challenges will be system resilience and power consumption, both being direct consequences of the increase in the number of components.

Regardless of the reliability of individual component, the system level reliability will continue to decrease as the number of components increases. It is projected that the Mean Time Between Failures (MTBF) of future extreme-scale systems will be at the order of hours or even minutes, meaning that many failures will occur every day. Without an efficient fault tolerance mechenism, faults will be so frequent that the applications running on the systems will be continuously interrupted, requiring the execution to be restarted every time there is a failure.

Also thanks to the continuous growth in system components, there has been a steady rise in power consumption in large-scale distributed systems. In 2005, the peak power consumption of a single supercomputer reached 3.2 Megawatts. This number was doubled only after 5 years, and reached 17.8 Megawatts with a machine of 3,120,000 cores in 2013. Recognizing this rapid upward trend, the U.S. Department of Energy has set 20 megawatts as the power limit for future exascale systems, challenging the research community to provide a 1000x

improvement in performance with only a 10x increase in power. This huge imbalance makes system power a leading design constraint on the path to exascale.

Today, two approaches exist for fault tolerance. The first approach is rollback recovery, which rolls back and restarts the execution every time there is a failure. This approach is often equiped with checkpointing to periodically save the execution state to a stable storage so that execution can be restarted from a recent checkpoint in the case of a failure. Although checkpointing is the most widely used technique in today's HPC systems, it is strongly believed that it may not scale to future extreme-scale systems. Given the anticipated increase in system level failure rates and the time to checkpoint large-scale compute-intensive and data-intensive applications, it is predicted that the time required to periodically checkpoint an application and restart its execution will approach the system's MTBF. Consequently, applications will make little forward progress, thereby reducing considerably the overall system efficiency.

The second approach, referred to as process replication, exploits hardware redundancy and executes multiple instances of the same task in parallel to overcome failure and guarantee that at least one task reaches completion. Although this approach is extensively used to deal with failures in Cloud Computing and mission critical systems, it has never been used in any HPC system due to its low system efficiency. To replicate each process, process replication requires at least double the amount of compute nodes, which also increases the power consumption proportionally.

Previous studies show that neither of the two approaches is efficient for future extreme-scale systems. And unfortunately, neither of them addresses the power cap issue. Achieving high resilience to failures under strict power constraints is a daunting and critical challenge that requires new computational models with scalability, adaptability, and power-awareness.

## 1.2 RESEARCH OVERVIEW

There is a delicate interplay between fault tolerance and power consumption. Checkpointing and process replication require additional power to achieve fault tolerance. Conversely, it has

been shown that lowering supply voltages, a commonly used technique to conserve power, increases the probability of transient faults. The trade-off between fault free operation and optimal power consumption has been explored in the literature. Limited insights have emerged, however, with respect to how adherence to applications desired QoS requirements affects and is affected by the fault tolerance and power consumption dichotomy. In addition, abrupt and unpredictable changes in system behavior may lead to unexpected fluctuations in performance, which can be detrimental to applications QoS requirements. The inherent instability of extreme-scale computing systems, in terms of the envisioned high-rate and diversity of faults, together with the demanding power constraints under which these systems will be designed to operate, calls for a reconsideration of the fault tolerance problem.

In this thesis, our research objective is to simultaneously address the power and resilience challenges for future extreme-scale systems so that both system efficiency and application QoS are guaranteed. To this end, we propose an adaptive and power-aware computational model, referred to as Lazy Shadowing, as an efficient and scalable alternative to achieve high-levels of resilience, through forward progress, in extreme-scale, failure-prone computing environments.

Previously, we have formally defined the computational model, studied possible techniques to realize and optimize the idea, and built analytical models for performance evalution. Next, I propose to continue the study of Lazy Shadowing in two aspects. Firstly, I propose to implement a prototype of Lazy Shadoing in the context of Message Passing Interface (MPI), to validate our computational model as well as measure its performance in real environment. Secondly, I propose to study the possibility of "smart shadowing" which further reduces the cost of shadowing by considering the specific system configuration, application characteristics, and QoS requirement.

### 1.2.1 Lazy Shadowing: a novel fault-tolerant computational model (completed)

The basic tenet of Lazy Shadowing is to associate with each main process a suite of shadows whose size depends on the "criticality" of the application and its performance requirements. Each shadow process is an exact replica of the original main process. To tolerate failures,

the main process and its associated shadow processes will execute in parallel, but on different compute nodes. The shadows initially execute at a reduced rate via Dynamic Voltage Frequency and Scaling (DVFS) to save power. If the main process completes the task succesfully, we will terminate the shadows immediately. If the main process fails, however, we will promote one of the shadow processes to be a new main process and possibly increase its execution rate to mitigate delay. This continues until the task completes.

Given that the failure rate of an individual node is much lower than the aggregate system failure, it is very likely that the main process will always complete its execution successfully, thereby achieving fault tolerance at a significantly reduced cost of power. Consequently, the high probability that shadows never have to complete the full task, coupled with the fact that they initially only consume a minimal amount of power, dramatically increases a power-constrained systems tolerance to failure.

The major challenge in Lazy Shadowing resides in determining jointly the execution rates of all task instances, both before and after a failure occurs, with the objective to optimize performance, resilience, and/or power consumption. As a case study, we develop a reward-based analytical framework to derive the optimal execution rates for minimizing energy consumption under strict completion time constraints.

### 1.2.2 Applying Lazy Shadowing to extreme-scale systems (completed)

Enabling Lazy Shadowing for resiliency in extreme-scale computing brings about a number of challenges and design decisions that need to be addressed, including the applicability of this concept to a large number of tasks executing in parallel, the effective way to control shadows execution rates, and the runtime mechanisms and communications support to ensure efficient coordination between a main and its shadow. Taking into consideration the main characteristics of compute-intensive and highly-scalable applications, we design two novel techniques, referred to as shadow collocation and shadow leaping, in order to achieve high tolerance to failures while minimizing delay and power consumption.

To control the processes' execution rate, DVFS can be applied while each process resides on one core exclusively. The effectiveness of DVFS, however, may be markedly limited by the

granularity of voltage control, the number of frequencies available, and the negative effects on reliability. An alternative is to collocate multiple processes on each core while keeping all the cores executing at maximum frequency. Then time sharing can be used to achieve the desired execution rates for each collocated process. Since this approach collocates multiple processes on a core, it simultaneously reduces the number of compute nodes and the power consumption.

Furthermore, we identify a unique opportunity that allows the lagging shadows to benefit from the faster execution of the mains, without incurring extra overhead. Specifically, when a failure occurs, Lazy Shadowing takes advantage of the recovery time and leaps forward the shadows by copying states from the mains. This technique not only achieves forward progress for the shadows at minimized power and delay, but also reduces the recovery time after each failure.

### 1.2.3 lsMPI: implementation of Lazy Shadowing in MPI (partially completed)

Though Lazy Shadowing has been shown to scale to future extreme-scale systems with our analytical models, a real implementation is still necessary for validation and performance measurement in real systems. We plan to implement a prototype of Lazy Shadowing as a runtime for Message Passing Interface (MPI), which is the de facto programming paradigm for HPC. Instead of a full-feature MPI implementation, the runtime is designed to be a separate layer between MPI and user application, in order to take advantage of existing MPI performance optimizations that numerous researches have spent years on. The runtime will spawn the shadow processes at initialization phase, manage the coordination between main and shadow processes during execution, and guarantees consistency for messages and non-deterministic events. With this implementation, we will perform thorough experiments measuring its runtime overhead as well as performance under failures.

### 1.2.4 Smart shadowing (future)

Lazy Shadowing is a flexible and adaptive computational model that deserves further investigation. Previous studies have shown that different nodes tend to have different failure

probabilities, e.g., 20% of the nodes account for 80% of the failures. The reason is complicated and may attribute to the manufacture process, heterogenous architecture, environment factors (e.g. temperature), and/or workloads. I propose to apply machine learning techniques to learn the heterogenity in failure distributions among a given system's nodes. Then I will study how the mapping from processes to physical cores can impact the performance and cost dichotomy. In addition, I will further consider allocating different number of shadow processes for different tasks to reduce cost while maintaining performance.

## 1.3 CONTRIBUTIONS

This thesis makes the following contributions:

- Development of an adaptive and power-aware computational model for efficient and scalable fault tolerance in future extreme-scale computing systems.
- Comprehensive and accurate analytical models to quantify the expected completion time and energy consumption in both HPC and Cloud Computing systems.
- A fully functional implementation of Lazy Shadowing for Message Passing Interface
- Exploration of Lazy Shadowing's potential to adapt to different environments, workloads, and QoS requirements.

## 1.4 OUTLINE

The rest of this proposal is organized as follow: Chapter 2 introduces fault tolerance and power management in large-scale distributed systems. In Chapter 3, we formally define Lazy Shadowing computational model and build analytical model for performance evaluation. In Chapter 4, we explore techniques to efficiently apply Lazy Shadowing in extreme-scale systems. Implementation issues are discussed in Chapter 5. Adaptivity and smart shadowing are discussed in Chapter 6. Chapter 7 and 8 lists the timeline and concludes the proposal, respectively.

# 2.0 BACKGROUND

# 3.0 LAZY SHADOWING: A NOVEL FAULT-TOLERANT COMPUTATIONAL MODEL

# 4.0 APPLYING LAZY SHADOWING TO EXTREME-SCALE SYSTEMS

# 5.0 LSMPI: IMPLEMENTATION OF LAZY SHADOWING IN MPI

# 6.0 SMART SHADOWING

# 7.0 TIMELINE OF PROPOSED WORK

**Table 7.1:** Timeline of Proposed Work.

| **Date** | **Content** | **Deliverable results** |
|---|---|---|
| Jan - Feb | Explore restoring in approximate computing in Section ?? of Chapter ?? | Pin-based framework for restoring approximation |
| Mar - May | Integrate restoring with information leakage in Section ?? of Chapter ?? | Experimental data of memory performance and security |
| Jun - Sep | Study restoring in Hybrid Memory Cube (HMC) in Section ?? of Chapter ?? | Modified simulator of temperature effect of restoring in HMC |
| Jul - Oct | Thesis writing | Thesis ready for defense |
| Oct - Dec | Thesis revising | Completed thesis |

The proposed works will be undertaken as shown in the Table 7.1. I will start the effort with task (1) to develop Pin-based framework for approximate computing of restoring. This task involves program annotation, chip generation, QoS evaluation, and conventional performance simulation, etc. While multiple complicated subtasks are there, this task has been partially finished, and will not take much time to complete. Afterwards, I'll move to task (2) to study information leakage in restoring scenario, and this task is partially on basis of the previous approximation work. With the completion of task (2), the overall goal of exploring DRAM restoring in application level have been reached, and then I'll start the study restoring's temperature effect in HMC, i.e., task (3). The general infrastructure can be borrowed from my previous HMC work [Zhang et al., 2015]. This task might be performed concurrently with other jobs, and thus might take more time to finish. At the end of task (3), the holistic exploration of DRAM restoring is considered finished, and thus I'll summarize all the tasks into my final thesis.

## 8.0 SUMMARY

DRAM technology scaling has reached a threshold where physical limitations exert unprecedented hurdles on cell behaviors. Without dedicated mitigations, memory is expected to suffer from serious performance loss, yield degradation and reliability decrease, which goes against the demanding of system designs and applications. Among the induced problems, restoring has been an long time neglected issue, and it is likely to impose great constraints to the scaling advancement. As a result, this thesis explores DRAM further scaling from restoring perspective.

Reduced cell dimensions and worsening process variation cause increasingly slow access and significantly more outliers falling beyond the specifications. To alleviate the influences on performance and yield, we propose to manage the timing constraints at fine chunk granularity, and thus more fast regions can be exposed to upper level. In addition, we devise the extra chunk remapping and dedicated rank formation to restrict the impacts of slow cells. However, the exposed fast regions can not be fully utilized for restoring oblivious page allocation; accordingly, to maximize performance gains, we move forward to profile the pages of the workloads, and deliberately allocate the hot pages to fast parts.

In addition, restoring can seek help from the correlated refresh operation, which periodically fully charge the cells. We propose to perform partial restore withe respect to the distance to next refresh; the closer to next refresh, the less needed charge and the earlier the restore operation can be terminated. For ease of implementation, we divide the refresh window into 4 sub ones, and apply an separate set of timings for each. Moreover, compared to refresh, restore contributes more critically to the overall performance, and hence we optimize the partial restore with refresh rate upgrading. More frequent refresh helps to lower the restoring objectives, but at a risk of raising energy consumption. As a compromise, we

selectively upgrade recent touched rows only.

As the most fundamental building block of computer systems, DRAM prolonged restoring operation will ultimately affect upper application level, and thus we further explore in extended scenarios, including approximate computing, information leakage and 3D stacked memory. With our full-stack exploration, we believe the scaling issues can be greatly alleviated.

## BIBLIOGRAPHY

Zhang, X., Zhang Y., and Yang J. DLB: Dynamic lane borrowing for improving bandwidth and performance in hybrid memory cube. In *ICCD*, pages 133–140, 2015.