Eric Paulz

ECE 4930-004

October 8, 2018

Paper Review: Algorithm Based Fault Tolerance for Matrix Operations

Summary

This paper presents a new method of fault tolerance called algorithm-based fault tolerance. The goal of this technique is to reduce the need for hardware redundancy, reduce cost of fault tolerance, and reduce overall overhead of fault tolerance operations. In this paper, the technique is applied specifically to matrix operations. The authors found that their technique was especially useful in multiprocessor systems.

Strengths

As far as I can tell, this seems to be a novel idea that could become very important in the next generation of supercomputers. The authors did a good job of introducing their topic and explaining to the reader why it matters.

Shortcomings

This paper was really painful to read. The introduction and high-level explanation of the concepts were fine, but when they got into the details of implementation I found that most of it was way over my head. Often, I would try to refer to the figures to make some sense of what they were talking about, but I found that very few of those were helpful either. It’s like you’d almost have to create a notes document of your own to keep track of what each equation, theorem, definition, lemma, etc. means. 1/10 would not read again.

Improvements

The conclusion is almost non-existent. After all the math and definitions that are explained, it’d be nice to read a thorough conclusion that explains the results and findings succinctly. This way, even if you had no idea what was going on throughout the paper, you could walk away with a decent understanding of the study.

Question(s) for Presenter

- What is VLSI?
- How can this technique be utilized in applications other than matrix operations?

Additional Questions

- What is the goal of algorithm based fault tolerance (ABFT)?
  - To replace redundancy in hardware with novel software techniques. This reduces cost and overhead.
- Can a full checksum matrix recover from two errors? Explain why or why not?
  - Yes. Each matrix can contain at most one inconsistent row and at most one inconsistent column. The faulty processor can be located at the intersection of the first inconsistent row and column. From there, the set of unreliable elements can be located, and therefore all errors caused by the faulty processor can be corrected.
- How does the overhead of ABFT scale with matrix size?
  - O(1/n) for hardware and $O(\log_2(n)/n)$ for time.