Investigation of Diskless Checkpointing Algorithms for Multiple Processor and Multicore Failures

Vaithiyanathan Sundaram, V3138147

Abstract—Improving fault tolerance within the clusters has become vital because of the drastic decrease in the Mean Time Between Failures (MTBF) in complex clusters. Checkpointing is one of the robust ways to improve fault tolerance by rolling back from a saved state in the event of a failure in the cluster. Since, checkpointing primarily relies on storage devices for storing the states at regular intervals it will be difficult to accommodate several storage devices to support the ever increasing number of components in the cluster. This performance bottleneck is overcome by an independent checkpointing method called, Diskless Checkpointing. Ever since diskless checkpointing was launched commercially there were several research explorations that led to the division of diskless checkpointing into three namely, neighbor based, parity based and Reed-Solomon code based. All these algorithms were applied for recovering clusters from multiple processor failures. On the other hand, due to space and power constrains the multicore architecture have started to be used widely in the clusters. The future world of supercomputing envisions that most of the upcoming clusters will contain a majority of multicore processors. This shows the necessity for a robust diskless checkpointing algorithm for recovering from failures in multicore architecture. There are robust algorithms to recover multiple processor failures, but there are only few algorithms that can bring back a multicore failure. Thus this paper aims at investigating the existing diskless checkpointing algorithms for multiple processor failures and devising a robust fail-safe algorithm for multicore architecture.

Index Terms—Diskless Checkpointing, Multicore, Multiple processor, Fault Tolerance, Reliable Computing

I. INTRODUCTION

What are the benefits of diskless checkpointing. How diskless checkpointing works Distinction between multiple processor and multicore failures The need for a robust algorithm for multicore failures Summary of following sections

II. RELATED WORKS AND INVESTIGATION

Complete investigation on different types of diskless checkpointing. Recent advancements on diskless checkpointing for multiple processor failures. Few works on multicore failures. Contribution of this paper

III. PROPOSED MODEL

After thorough investigation of the existing schemes, a robust algorithm for multicore failures will be proposed taking into account the similarity of the working of both multicore and multiple processor clusters. The existing algorithms for multicore failures will not work efficiently for recovering multiple processor failures. The proposed algorithm will try to recover multiple processor and multicore failures in unison.

IV. RESULTS

Since the Message Passing Interface in most of the existing algorithms used Berkeley Lab Checkpoint/restart software the proposed model may be implemented at least in a high level. If implemented, results will be used to compare the strengths of using the proposed algorithm.

V. CONCLUSION

Summary of the paper. Future of diskless checkpointing and fault tolerant computing.

VI. REFERENCES

This paper will have around 15-18 papers.