Rohan Jhaveri

An Analysis of Resilience Techniques for Exascale Computing Platforms

Daniel Dauwe_, Sudeep Pasricha_y, Anthony A. Maciejewski_ and Howard Jay Siegel_y
_Department of Electrical and Computer Engineering
yDepartment of Computer Science
Colorado State University, Fort Collins, CO, 80523, USA
Email: ddauwe@rams.colostate.edu, sudeep@colostate.edu, aam@colostate.edu, hj@colostate.edu

Since High Performance Computing (HPC), involves large systems performing tightly coupled operations, there is a possibility that an error occurred at one node, can easily propagate to other nodes in just microseconds. In the process of achieving better reliability, by mitigating the effects of failures, many techniques are used. This paper explains different types of resilience techniques that are briefly described below that could be used in developing exa-scale computers.

1. Checkpoint/Restart: The central principle of Checkpoint/Restart is to periodically save the state of the whole system. The time interval between two checkpoints, depends on the system's checkpoint time (TCPFS) and failure rate (λsys).

2. Multilevel Checkpointing: Multi-Level Checkpointing approach, makes use of different types of checkpoints, having different levels of resilience and cost in just a single application run. In Multi-Level Checkpointing several storage technologies i.e. multiple levels are combined to store a checkpoint. Each level offers specific reliability and reliability trade-offs.

3. Message Logging/Parallel Recovery: Message logging attempts to provide resilience to a system by recording messages sent among processes to create snapshots of the system's execution distributed across system memory. When a failure occurs, the failed node can use messages stored in the memory of other system nodes to reduce the amount of rework that is performed by the system when recovering.

4. Redundancy: It executes redundant copies of the same piece of code or some hardware to improve performance of the system.

The authors of this paper considered resource management and scheduling techniques to perform their simulation to analyze performance of the system based on the application type. The scheduling techniques considered were First come first served (FCFS) Technique, Random Technique and Slack-Based Technique. The results of this paper show that parallel recovery gives best performance in terms of efficiency. But the author here fails to provide information about the energy consumption of each resilience techniques. Also, different parameters such as component reliability, bandwidth of the network, latency of the network could impact the performance of the system in terms of both energy and efficiency.