Integrated Fault Tolerance Architecture for MagaScale Computing

RI2N, Cuckoo FT-MPI, and Speculative Checkpointing

MEGASCALE

Objective

To achieve utmost dependability for "MegaScale" computing on a commodity cluster facilitating over 10s of thousands of nodes, various fault tolerance software components must be facilitated and tightly integrated at different layers of the parallel computing software stack, at the interconnect layer (*RI2N*), the message passing layer (*Cuckoo FT-MPI*), and the application checkpointing layer (*Speculative Checkpointing*). Here we present a partial list of our ongoing work:

Cuckoo FT-MPI

A Component-based Fault Tolerant MPI Architecture

Architecture

Component-based, Fault Model-Aware MPI

Portable: components for adapting to different underlying computing environment

Flexible: components for handling different fault and recovery models

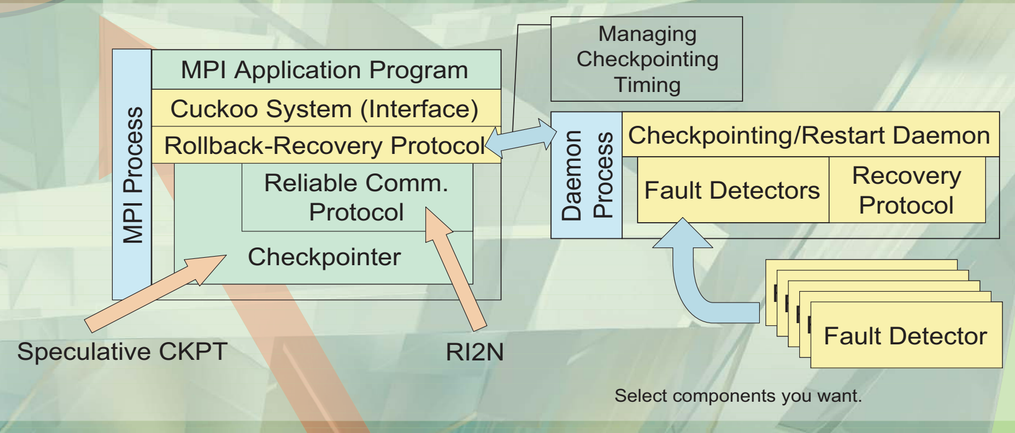
Transparent: transparent to user code; components to handle different execution phases and appropriate recovery

Fault Detection

Fault Detector Components detect various fault instances, and select the appropriate recovery protocol component (e.g.,

Handling Multiple Faults: if a network fault occurs repeatedly, it may NOT be a network fault (e.g., the endpoint node may be flaky)

- Employ a network fault detector w/ fault counter
- If the counter reaches a threshold, delegate to other fault detectors

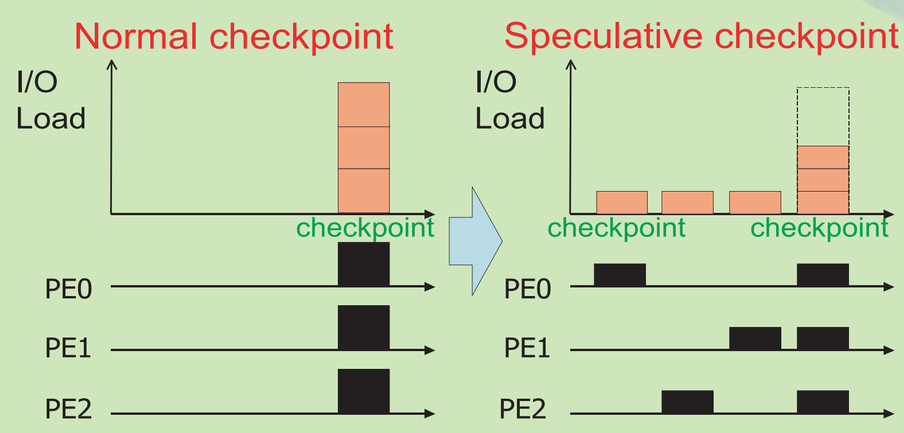


Speculative Checkpointing

Perform speculative & asynchronous checkpoints between synchronous & incremental checkpoints to amortize checkpoint I/O.

- Pages that have changed and speculated not to change till the next synchronous checkpoint is checkpoint ahead-of-time
- If no further changes occur till the next synchronous checkpoint, the subject page need not be checkpointed

Up to 41% speedup in the best case so far due to disk I/O amortization.



RI2N - Redundant Interconnection with

- Utilize multiple links of commodity networks (e.g. GbE) to achieve both high-bandwidth and high-dependability
- Aggregate bandwidth of multiple links via trunking
- Enhanced link failure detection via broadcasting of ack packets
- Completely software-layer implementation; does not depend on IEEE802.3ad thereby avoiding single point failure of switches