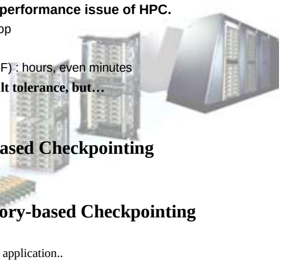
ILLINOIS INSTITUTE OF TECHNOLOGY
Transforming Lives. Inventing the Future.

An Implementation and Evaluation of Memory-based Checkpointing

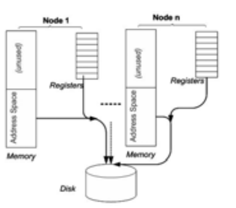
Hui Jin Xian-He Sun Bing Xie Yong Chen
Illinois Institute of Technology

Problem: I/O Bottleneck for Checkpointing

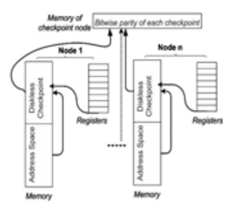


- Fault tolerance becomes a vital performance issue of HPC.
  - Computing Power: Petaflop to Exaflop
  - System scale: 1,000+ / 10,000+ nodes
  - Mean Time between Failures (MTBF) : hours, even minutes
- Checkpointing is widely used for fault tolerance, but…
  - Suffers from I/O overhead.
  - Generates I/O bursts.
- An alternative: Memory-based Checkpointing
  - Fast network.
  - Affordable and sufficient memory.
- Potential Issues with Memory-based Checkpointing
  - Reliability: Memory is volatile..
  - Memory Abuse: Cannot affect the host application..

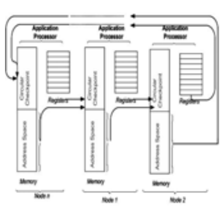
Four Chechpointing Schemes



Disk-based Checkpointing



Parity-based Checkpointing



Neighbor-based Checkpointing

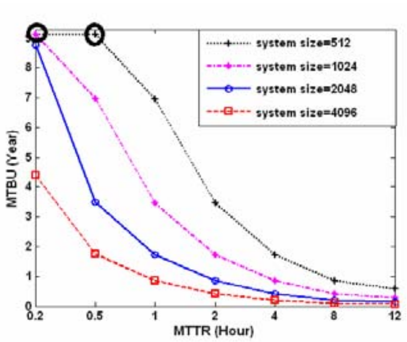
Mutually Paired Checkpointing

Centralized v.s. Decentralized
Reliability v.s. Memory Usage
Complexity v.s Transparency

Reliability Analysis

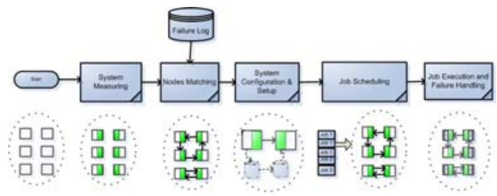
Assumptions and Analysis:

- Failure Arrival $p_f(X \le x) = 1 - e^{-x/\lambda_f}$
- Failure Repair $p_r(X \le x) = 1 - e^{-x/\mu_r}$
- Un-recoverability $p_{unrecover} = \int_{r=0}^{\infty} p_r(X = r) \times (1 - p_f{}^2(Y > r)) = \int_{r=0}^{\infty} p_r(X = r) \times (1 - (1 - p_f(Y \le r))^2)$
- Mean Time between Un-recoverability $MTBU = \frac{\lambda_f / n}{p_{unrecover}}$



MTBU with MTTR and System Size (MTBF=7884 Hours)

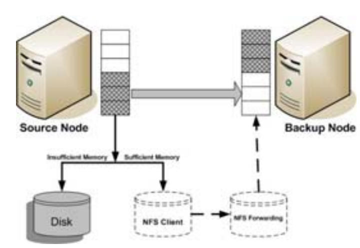
System Overview



Implementation

Virtual memory file system + NFS Protocol:

1) Backup node mounts its virtual memory as a general file system;
2) NFS Forwarding: The virtual memory file system is exported as an NFS server to the source node;
3) The source node mounts the remote memory from backup node as an NFS client.

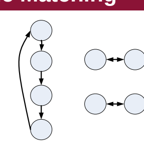


Flexible Switch between Disk-based and Memory-based Checkpointing

Failure-aware Node Matching

Topology Selection:

- Ring Topology: $\frac{C^k_{n-k+1} - C^k_{n-k+1}}{C^k_n}$
- Mirror Topology: $\frac{C^k_{n/2} 2^k}{C^k_n}$



Paired Failures on LANL Failure Trace

Physical Info-based Node Matching

- Nodes from one component (rack, power, etc) cannot be paired

Failure-aware Node Matching

- Match node based on the failure log.

Objective: Match nodes with different reliability patterns to each other

Begin:

1) Calculate the reliability of each node.
2) Sort nodes with their the reliability
3) Match the first and last node in the sorted array until all the nodes are paired

End

ID	Size	Time Span (days)	NO. of Single Failures	NO. of Paired Failures
9	256	678	280	**0**
10	256	665	237	**0**
14	256	514	125	**0**
12	510	677	258	**0**
20	510	1359	2478	**1**
11	578	668	267	**0**
18	1024	1220	3997	**5**
19	1024	1056	3284	**3**

Experimental Environment

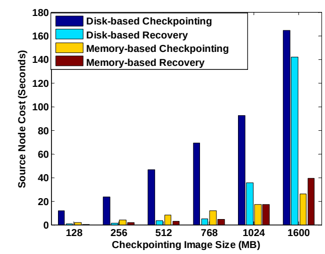
-Platform:

- Sun cluster with 20 Sunfire V210R compute nodes.
- Each node is equipped with two 1GHz CPUsand 2GB Memory.
- All the compute nodes are connected with a Gigabits Ethernet.
- The storage is a disk array with six SCSI hard disks.

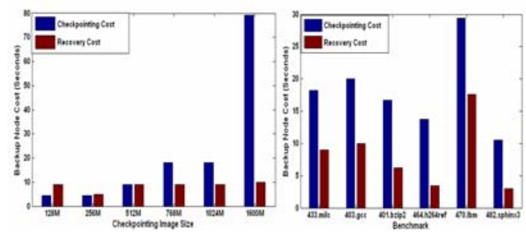
-Software:

- Operation System: SunOS 5.9.
- Checkpointing System: Libckpt
- Application: Matrix Multiplication and CPU2006.

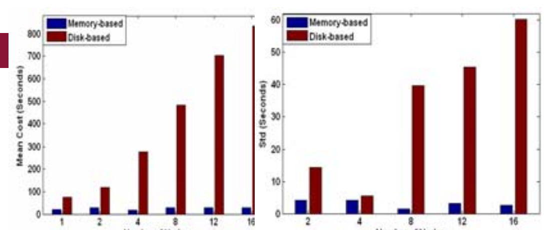
Source Node Performance



Backup Node Performance



Scalability Performance



Conclusion and Future Work

- Review of the state of art of memory-based checkpointing.
- Reliability Analysis of memory-based checkpointing
- Failure-aware Node matching
- Design and Implementation
- Flexible combination between disk- and memory-based ckpt.
- Comprehensive Evaluation.
- Future work
  - Implementation on other checkpointing system.
  - Implementation on coordinated Checkpointing.
  - Dynamic node matching with predicted memory usage, job, etc.
  - RES: Reliable, Efficient, Scalable Checkpointing Environment.

Acknowledgements

This work was supported in part by National Science Foundation under NSF grants CNS-0751200, CCF 0702737, CNS-0834514.

http://www.cs.iit.edu/~scs
Hui. Jin, Xian–He Sun, Bing Xie and Yong Chen
{hjin6,sun,bxie3,chenyon1}@iit.edu