Optimal Checkpointing for Inter-dependent parallel processes in Volunteer PC grids

Mohammad Tanvir Rahman, Dr Jaspal Subhlok, Dr Gopal Pandurangan

The Problem

- To minimize the completion time for inter-dependent parallel processes running in a Volunteer environment.

Key idea: By adopting optimal checkpoint interval considering any level of replication of the clients.

Key Contribution:

- Designing a mathematical model for finding optimal check point interval
- Implementing the algorithm and evaluating the performance.

What is Volunteer Environment

- A network of idle computers
- Can be in different physical location and connected through a volunteer framework.
- Provides free CPU power
- Frequent failure
- Checkpoint and replication needed for successful parallel execution

Motivation

- Checkpoint and replication needed for effective parallel execution
- Optimal Checkpoint Interval needed for minimizing completion time and increase collective throughput
- Too frequent checkpoints: waste a lot of time and resources
- Infrequent checkpoints: increase a work-loss in case of client failure

Challenges in Finding Optimal Checkpoint Interval:

- Highly failure characteristics of volunteer nodes
- Replication of processes

Experiment Setup

Testbed: Around 300 volunteer nodes. Roughly 25% on our university campus and 75% distributed worldwide

Test Application: Replica Exchange Molicular Dynamics (REMD)

Parameters:

- Number of processes: 16, 32
- Level of replication: 1,2,3
- Checkpoint interval: 12 to 24000 sec
- Checkpoint size: 50KB and 5MB
- Host selection policy: None
- Number of Run: 10 times

Algorithm

Input:

- Success distribution of a single process, p = f(t) $= e^{-\lambda t}$
- Number of processes, n
- Number of replica for each process, r
- Time to create a checkpoint, T_s

Output:

Optimal Checkpoint interval, T_c

For n processes each with r replica, Optimal checkpoint interval, $T_{c(op)}$ =

$$\min_{T_c = 1 \ldots \infty} \frac{1}{(1-(1-f(T_c)^r)^n)} + \frac{T_s}{T_c}$$

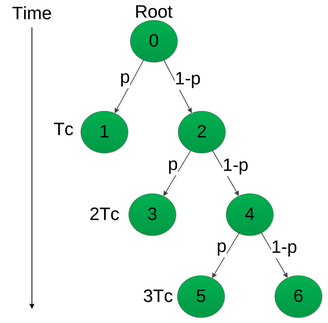

Fig 1: Tree structure to calculate avg time to reach next checkpoint

Final equation calculated by our algorithm for finding

Optimal Checkpoint Interval, $T_c^{opt} = \frac{2W\left(\frac{1}{2}\lambda n \sqrt{\frac{1}{\lambda n T_s}} T_s\right)}{\lambda n}$

Where, W(z) is the Lambert W function

Result

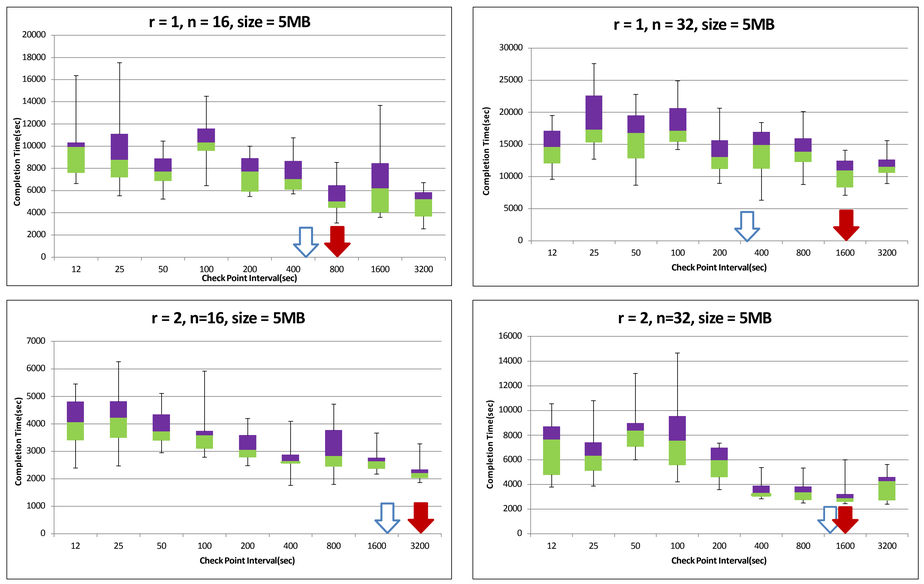

Fig 2: Completion times for varying checkpoint intervals in different execution scenarios with 5MB checkpoints. Blue Blank Arrow indicates Our predicted Optimal CPI, Red Solid Arrow indicates CPI for measured lowest execution time

- Sometimes our predicted checkpoint interval is not close to the actual optimal checkpoint interval, However
- Job completion time with our predicted checkpoint interval is very close to lowest completion time

Performance

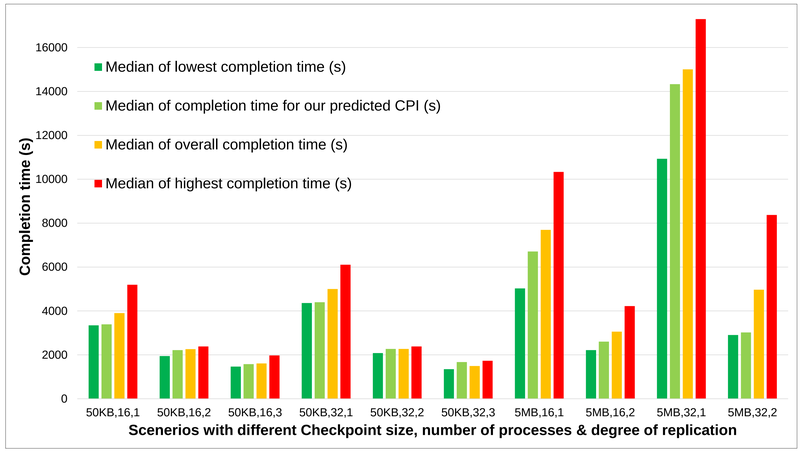

Fig 3**:** Comparison of completion time with different checkpoint interval

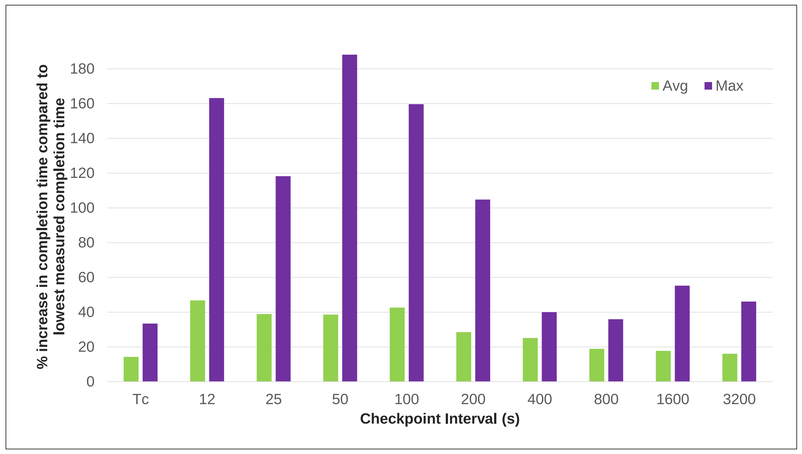

Fig 4: Performance with always using a predefined fixed checkpoint interval versus the performance using our predicted optimal checkpoint interval (T_c).

Conclusion

- Our predicted checkpoint interval (Tc) minimizes job completion time.
- True optimal checkpoint interval is unknown and using a predefined fixed checkpoint interval can result 188% increase in completion time compared to lowest completion time.
- On average, completion time with our predicted checkpoint interval has only a 15% increase in completion time compared to lowest completion time.

Publication: M. Rahman, H. Nguyen, J. Subhlok, and G. Pandurangan, "Checkpointing to minimize completion time for interdependent parallel processes on volunteer grids," *in CCGrid 2016: The 16th IEEE/ACM International Symposium on Clusters, Cloud and Grid Computing*, Colombia, May 2016.