Optimizing Asynchronous Multi-Level Checkpoint/Restart Configurations with Machine Learning

Tonmoy Dey†, Kento Sato†2, Jian Guo†2, Bogdan Nicolae†3, Jens Domke†2, Weikuan Yu†, Franck Cappello†3, Kathryn Mohror†4

† Florida State University, USA

†2 RIKEN Center for Computational Science(R-CCS), Japan

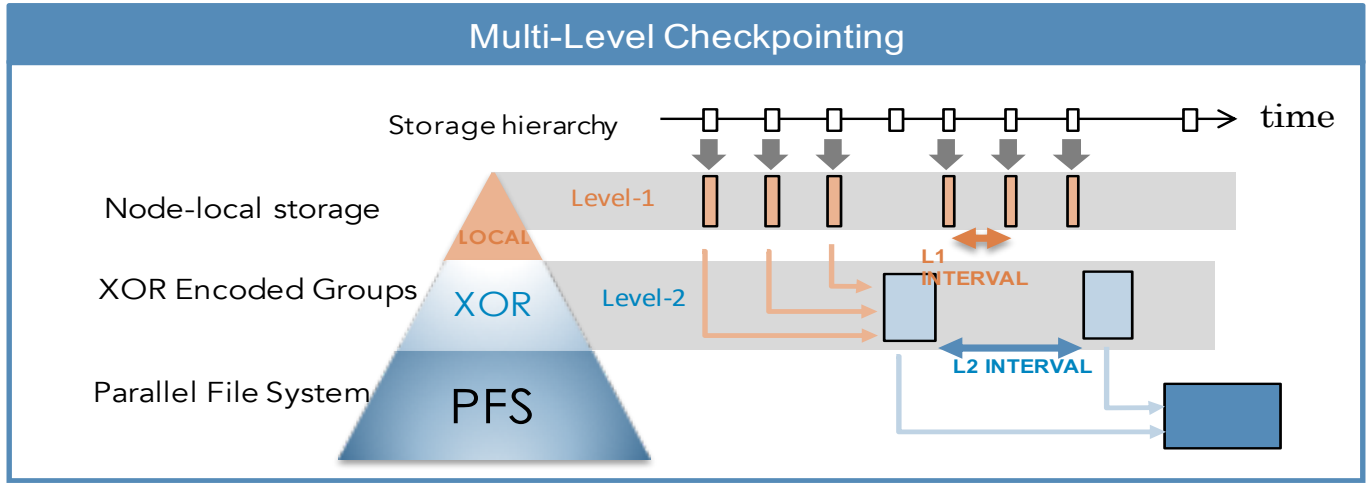
†3 Argonne National Laboratory, USA

†4 Lawrence Livermore National Laboratory, USA

Introduction

Checkpoint/Restart (C/R)

- Checkpoint-and-Restart is a commonly used technique for large-scale applications running for long time that:
  - Writes a snapshot of an application at fixed intervals and
  - On a failure, the application can restart from the last checkpoint
- With emergence of fast local storage, Multi-Level Checkpointing (**MLC**) has become a common approach with hierarchically written checkpoints



Background and Motivation

Optimal Checkpoint Configuration

- Determining the optimal checkpoint configuration is very crucial for efficient checkpointing. However, finding this optimal configuration for efficient checkpointing is complicated

- There exists a tradeoff for finding the optimal configuration:
  - **Frequent checkpoint**: Spends more I/O time for checkpointing
  - **Infrequent checkpoint:** Lose more useful computation on a failure

more resilient but …
huge overhead

Low overhead but …
less resilient

Background and Motivation

Approaches to Determine Optimal Configuration

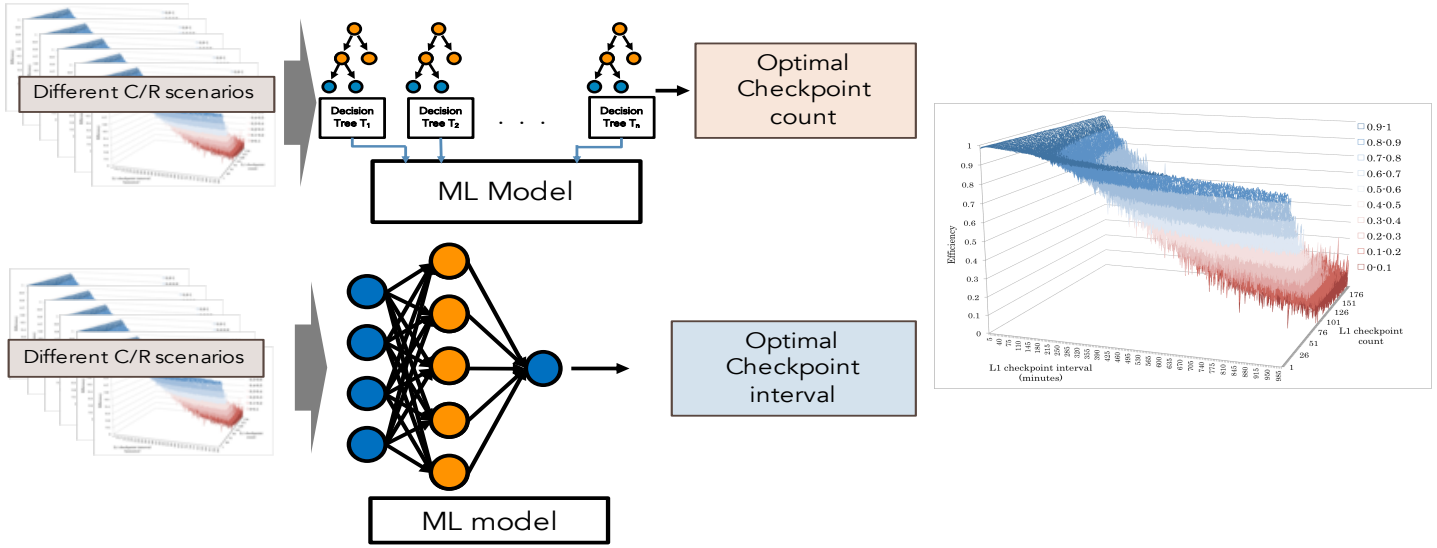
- There are existing two approaches to determine checkpoint configuration
  - **Approach 1: Modeling checkpointing behaviors**
    - Execution states are categorized into compute, checkpoint and recovery state. This approach works well for simpler checkpoint models, but is significantly difficult to implement for complex systems
  - **Approach 2: Simulation for optimal checkpointing**
    - Simulation approach is much more accurate than Modeling approach, however, it takes very long time to find optimal checkpoint configurations

- **In this paper, we try to obtain the optimal checkpoint configuration for a given HPC system using the effectiveness and accuracy of the simulation approach and combine it with machine learning models to avoid the the time taken by simulation to obtain the optimal result.**

Design and Implementation

Combine simulation with Machine Learning

- Apply various AI techniques to learn checkpoint schemes given different C/R scenarios. There are two distinct ways to achieve it:
  - **Machine Learning (ML) Model**: Use existing machine learning models on the simulated dataset to see how well it learns.
  - **Neural Network (NN) Model**: Build our own neural network to see how well it can learn and predict the optimal configuration.

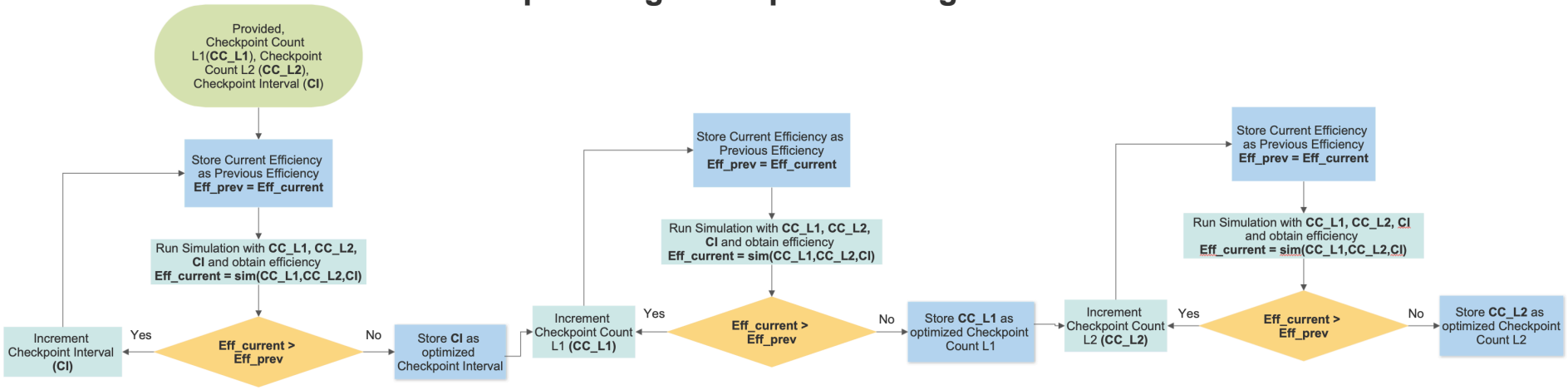


Design and Implementation

Simulation

- The simulator has been developed to replicate the behavior of real-world scenarios when using three-level checkpoint for large scale systems.
  - The simulator is provided with three critical parameters for each level, checkpoint overhead, checkpoint restart time, and failure rates.
  - The parameters are used by the simulator to provide the user with elapsed time and the efficiency (% of time utilized by useful computations) of the system.

Optimizing Checkpoint Configuration

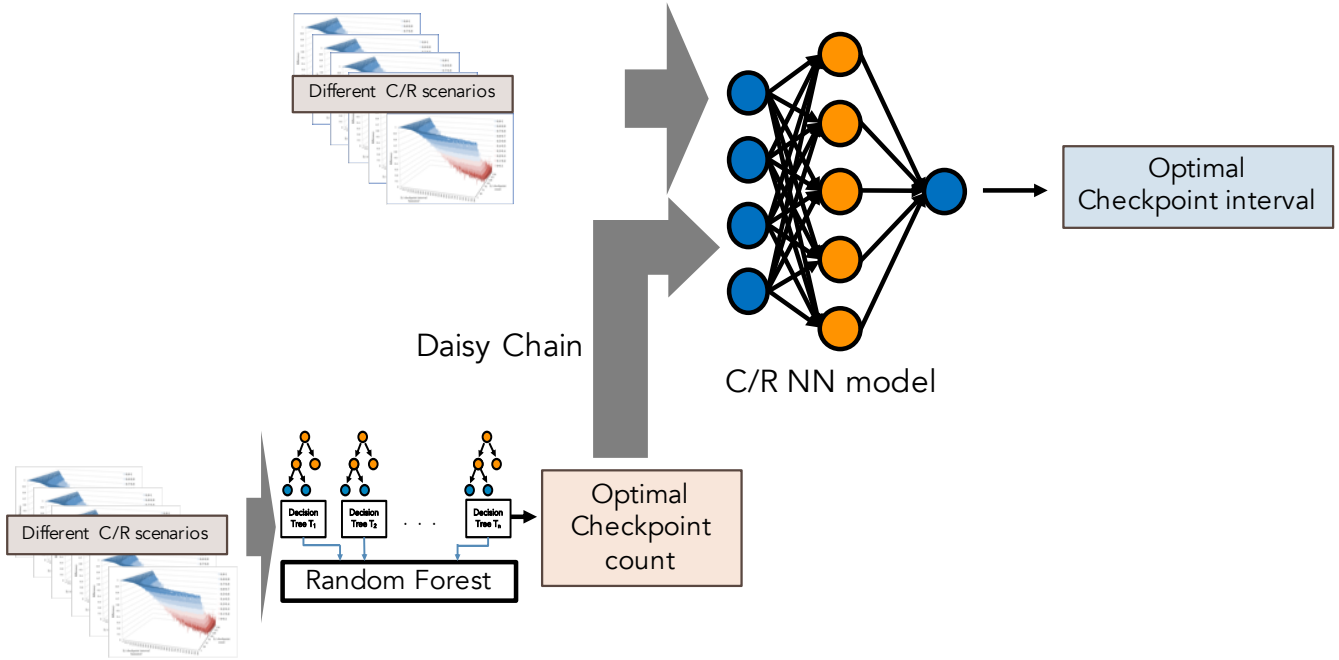


Design and Implementation

Model Optimization

- **Daisy Chaining**: Feed the output from Checkpoint Count prediction as an input to the Neural Network for Checkpoint Interval prediction
- **Parameter Optimization/Reduction**: Remove interdependent, redundant parameters

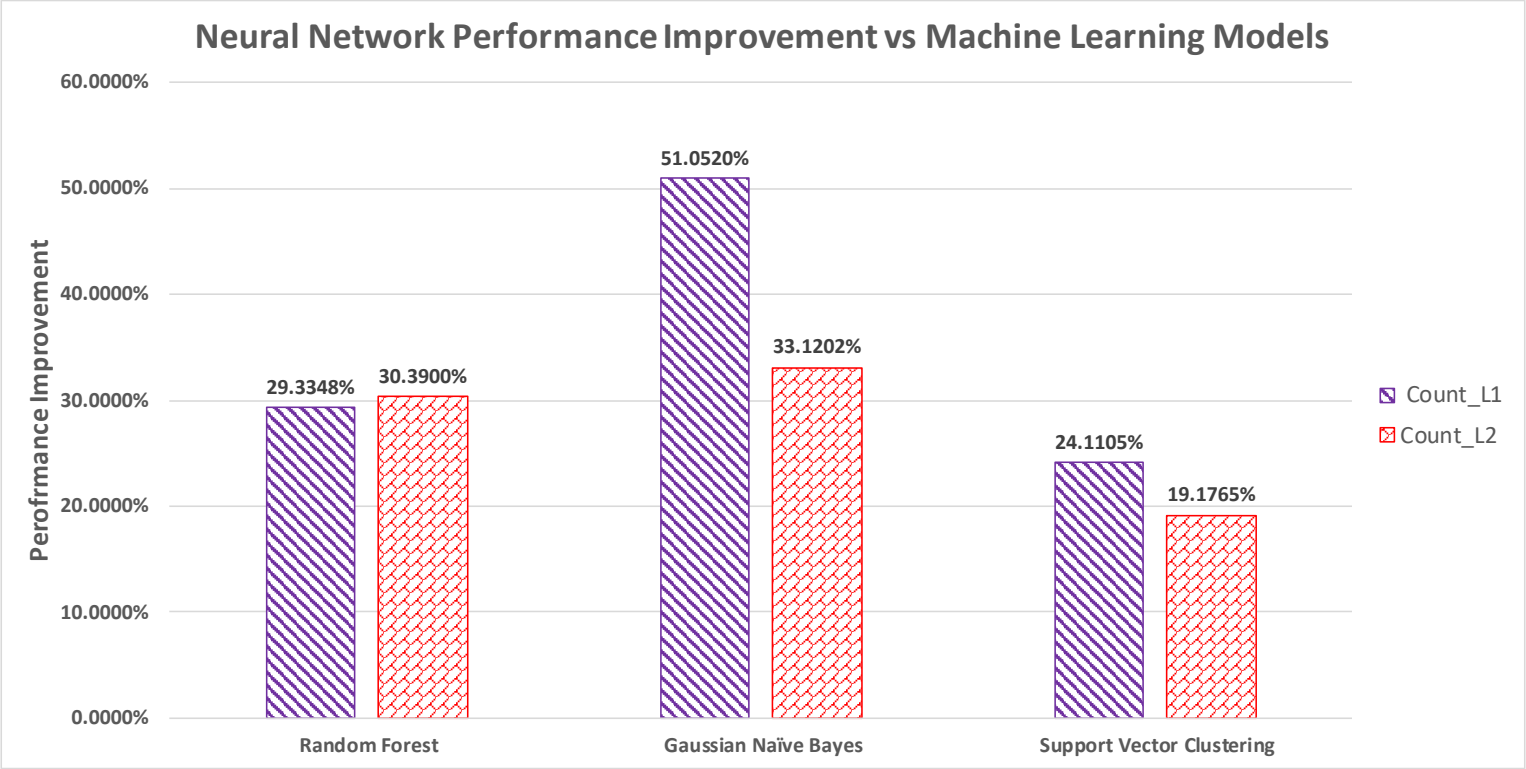
Daisy Chaining



Evaluation

Neural Network vs Machine Learning

- For a three-level checkpoint model, the **neural network** showed better performance with an **improved accuracy between 19 to 51%** in comparison to the **machine learning models.**



Conclusion

- We present an idea to combine the simulation approach with machine learning models to determine the optimized parameter values for different configurations of C/R.
- We show that our models can predict the optimized parameter values when trained with the simulation approach
- We have also demonstrated that using techniques such as neural networks can improve the performance over the machine learning models with neural network sometime exceeding the performance of a machine learning model by 50%.

Contact Information

Name	Contact
Tonmoy Dey	td18d@my.fsu.edu
Kento Sato	kento.sato@riken.jp
Bogdan Nicolae	bnicolae@anl.gov
Jian Guo	jian.guo@riken.jp
Jens Domke	jens.domke@riken.jp
Weikuan Yu	yuw@cs.fsu.edu
Franck Cappello	cappello@mcs.anl.gov
Kathryn Mohror	mohror1@llnl.gov