Automatically Testing Correctness of HPC Scientific Applications

Manish Motwani
manish.motwani@oregonstate.edu
mmotwani.com
Why Test HPC Scientific Applications?
Why Test HPC Scientific Applications?

- Improve application quality
- Accuracy & Reliability
- Reproducibility
- Error Prevention
- Verification & Validation
Why Test HPC Scientific Applications?

- Improve application quality
  - Accuracy & Reliability
  - Reproducibility
  - Error Prevention
  - Verification & Validation

- Prevent resource wastage
  - Performance Optimization
  - Resource Efficiency
Why Test HPC Scientific Applications?

**Improve application quality**
- Accuracy & Reliability
- Reproducibility
- Error Prevention
- Verification & Validation

**Prevent resource wastage**
- Performance Optimization
- Resource Efficiency

**Ease software management**
- Debugging and Maintenance
- Code Evolution
- Documentation & Collaboration
- Compliance & Regulation
Challenges in Testing HPC Scientific Applications

- Resource limitations
- HPC Resource Constraints
- Large Datasets
- Time constraints
Challenges in Testing HPC Scientific Applications

Resource limitations:
- HPC Resource Constraints
- Large Datasets
- Time constraints

Inherent code complexity:
- Parallelism & Concurrency
- Non-deterministic behavior
- Heterogeneous computation
Challenges in Testing HPC Scientific Applications

Resource limitations
- HPC Resource Constraints
- Large Datasets
- Time constraints

Inherent code complexity
- Parallelism & Concurrency
- Non-deterministic behavior
- Heterogeneous computation

Oracle absence
- Lack of standard benchmarks
- Lack of domain expertise
- Test oracle problem
Automatically Detecting Memory Bugs in HPC Scientific Applications

HPC defect benchmark (buggy program, fixed program, triggering test)

- Incorrect synchronization between host and device due to API misuse
- Missed synchronization between host and device
- Incorrect copying of data between host and device
- Missed copying of data between host and device
- Accessing device space from host
- Concurrent modification of shared variable from device and host
- Use of stale data
- Data race
Automatically Detecting Memory Bugs in HPC Scientific Applications

- Large Language Model
- Fuzzing
- Differential Testing
- Program Analysis

HPC Test

HPC defect benchmark

fail

Error causing test input

pass
Why Test HPC Scientific Applications?

- Improve application quality
  - Accuracy & Reliability
  - Redundancy
  - Error Prevention
  - Verification & Validation
- Prevent resource wastage
  - Performance Optimization
  - Resource Efficiency
- Ease software management
  - Debugging and Maintenance
  - Code Evolution
  - Documentation & Collaboration
  - Compliance & Regulation

Challenges in Testing HPC Scientific Applications

- Resource limitations
  - HPC Resource Constraints
  - Large Datasets
  - Time constraints
- Inherent code complexity
  - Parallelism & Concurrency
  - Non-deterministic behavior
  - Heterogeneous computation
- Oracle absence
  - Lack of standard benchmarks
  - Lack of domain expertise
  - Test oracle problem

Automatically Detecting Memory Bugs in HPC Scientific Applications

- Incorrect synchronization between host and device due to asynchronous
  - Incorrect synchronization between host and device
  - Incorrect copying of data between host and device
  - Incorrect copying of data between host and device
  - Data race

Automatically Detecting Memory Bugs in HPC Scientific Applications

- Large Language Model
- Fuzzing
- Differential Testing
- Program Analysis

manish.motwani@oregonstate.edu
mmotwani.com