BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

# Fault Tolerance Using Lower Fidelity Data in Adaptive Mesh Applications

Anshu Dubey
Lawrence Berkeley National Laboratory

Prateeti Mohapatra and Klaus Weide
University of Chicago

FTXS June 18, 2013

Flash Center for Computational Science

The University of Chicago

# Background and Motivation

- Much of the focus in applications fault tolerance is on checkpoint-restart
- Most large scale scientific applications already use that for other purposes
  - Batch queues and a simulation needing more than one slot to complete
  - Need for in-flight change in parameters
  - Exploring more than one path in the middle of evolution
  - Recovery in relatively rare instances at the moment
- The recovery part is expensive relative to the rest of the simulation cost
  - There is no way of telling whether the recovery was triggered by a non-critical fault or a critical one
- It should be possible to do in-flight recovery from a non-critical fault

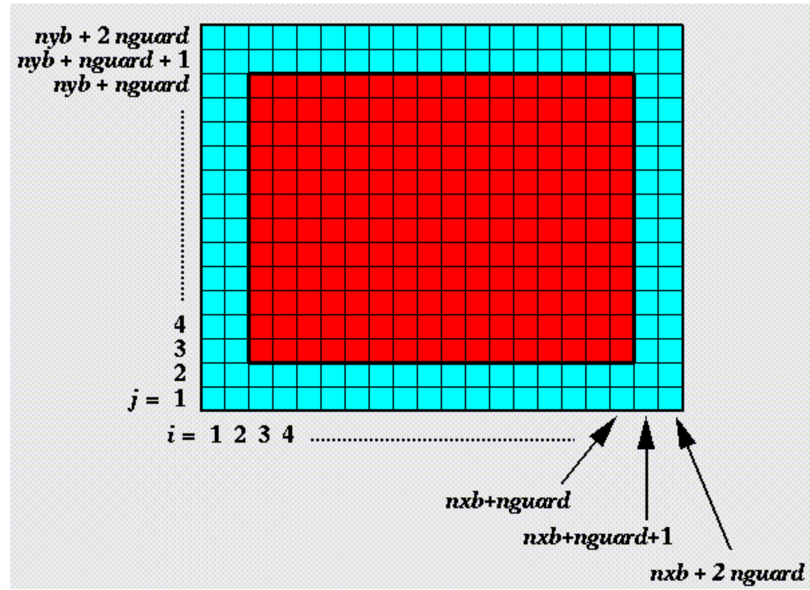
This work explores whether there is a class of faults that can be classified as non-critical in the context of AMR applications, and if so can one recover from them in-flight

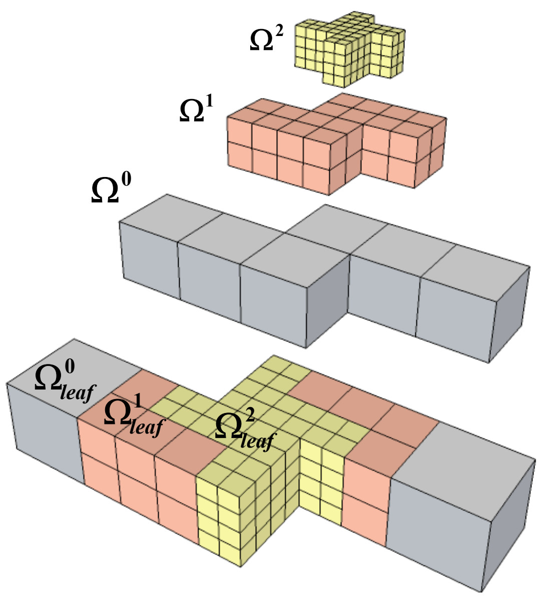
# Outline

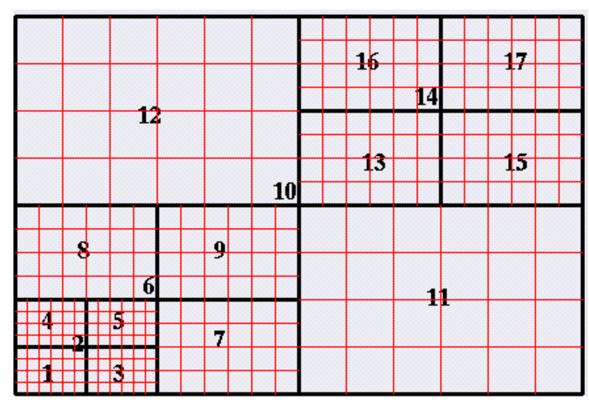
- **Block Structured AMR**
- Algorithm
- Test Cases
- Results
- Future Directions

# Block Structured AMR

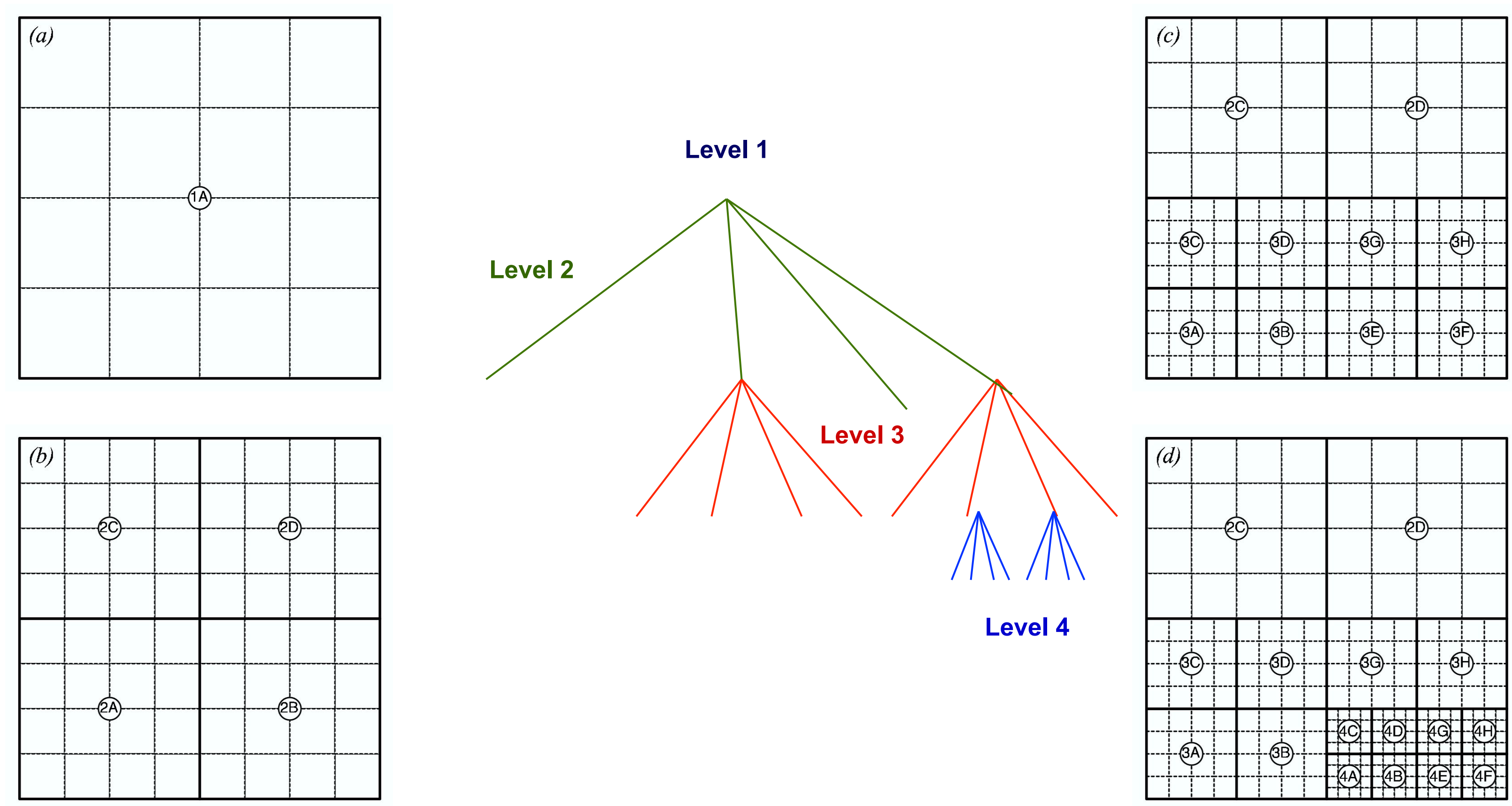
- The grid is composed of blocks
- Cover different fraction of the physical domain.
- In AMR blocks at different levels of refinement have different grid spacing.



# Oct-tree AMR 2D Block Map




- All blocks have same dimensions
- Global block numbers based on Morton order (space filling behavior)
- Blocks have parent-child relation

# AMR: Characteristics

- Useful in applications where different sections of the domain have different characteristics
  - Smooth flow -> low resolution – lower level leaf block
  - Shocks or other structures -> high resolution – higher level leaf block
- The mesh changes dynamically
  - A block may refine – generate 2^d children blocks
  - Blocks may derefine – children blocks go out of existence and their parent becomes the new leaf block
- Applications may choose to evolve on all blocks or only the leaf blocks
  - The union of leaf blocks covers the entire domain
  - When there is no subcycling evolution on parents is not necessary

# AMR: Machinery

Interpolation needed at various points

- Ghost cell filling at fine-coarse boundaries
  - One coarse cell corresponds to two fine cells along each dimension
  - All nearest neighbors of a block are known locally
  - The parent-child relationships for all blocks on a processor are also known locally
- Refinement step
  - Maps and utilities exist for
    - restriction (from higher to lower resolution) from child to parent
    - prolongation (from lower to higher resolution) from parent to child

# Outline

- Block Structured AMR
- **Algorithm**
- Test Cases
- Results
- Future Directions

# The Data Fidelity

Assertion : For every child block there is a parent block that either does, or if not, can be made to hold lower fidelity data for the same computation that the child block is doing

- In patch based meshes the lower fidelity data is already there because solution evolves on all patches
- In octree meshes, where it is optional to evolve on parents two options exist –
  - force evolution on parent blocks
    - amounts to extra 12.5% work in 3D, 25% in 2D and 50% in 1D
  - restrict to the parent blocks at the end of every time step or operator computation
    - less expensive computationally, more expensive communication wise, but definitely more accurate than evolution

# The Algorithm

- We opt for the restriction option, which could be applied after every expensive physics operator
  - For this work we restricted at the end of every time-step

```
restrict_by_one_level // populate parents
get_list_of_blocks(leaf)
forall blocks
  apply operator_1
  .
  .
  apply operator_2
endfor
if(any_block_had_error)
   prolong_from_parents
   forall faulty blocks
     apply operator_1
     .
     .
     apply operator_2
endif
```

# Outline

- Block Structured AMR
- Algorithm
- **Test Cases**
- Results
- Future Directions

# Testing

- We did not focus on fault injection and detection
  - the implementations for them are very simplistic
    - read in a range of blocks and a range of time steps in which to introduce faults, and then randomly pick
    - introduce a fault tag in the block metadata
- More interested in being able to control where to introduce faults when
  - to see whether there are faults that are more critical than others in our applications
    - whether the region of the faults matters more or the timing
    - also whether several faults bunched together had greater impact

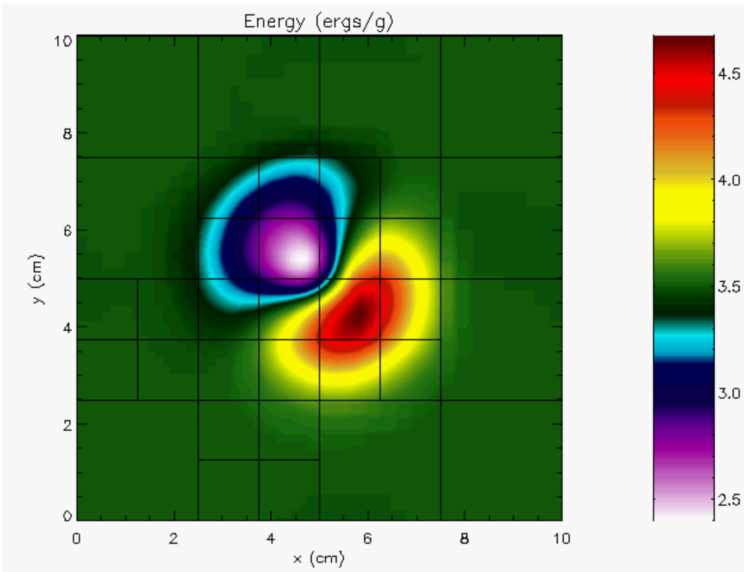
# Selected Problems

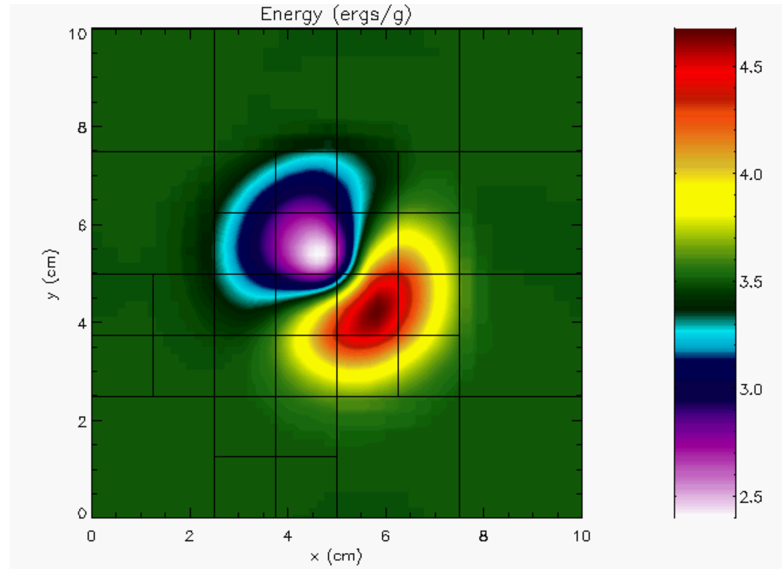
- Isentropic Vortex
  - Smooth flow field with no shocks
  - Vortex at a specified location
  - Faulty blocks placed at the vortex, near the vortex and away from the vortex
- Sod Shock Tube
  - 1D shock discontinuity problem, has multidimensional effects when the shock is placed at an angle
  - Faulty blocks placed very near the shock, in the rarefaction and compression waves and away from the shock

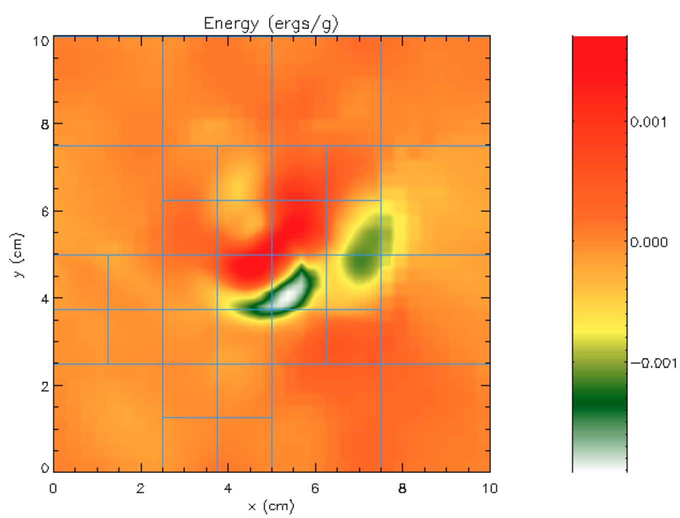
# Outline

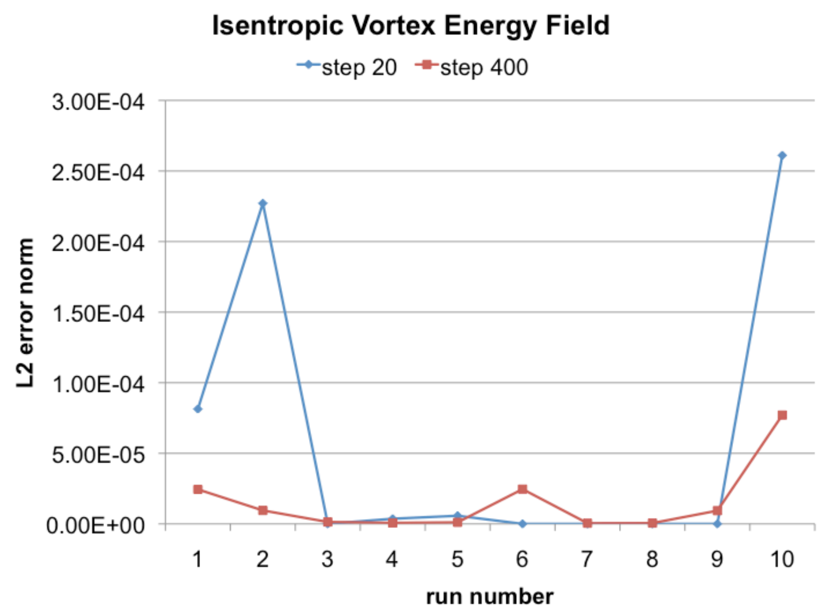
- Block Structured AMR
- Algorithm
- Test Cases
- **Results**
- Future Directions

# Isentropic Vortex Results

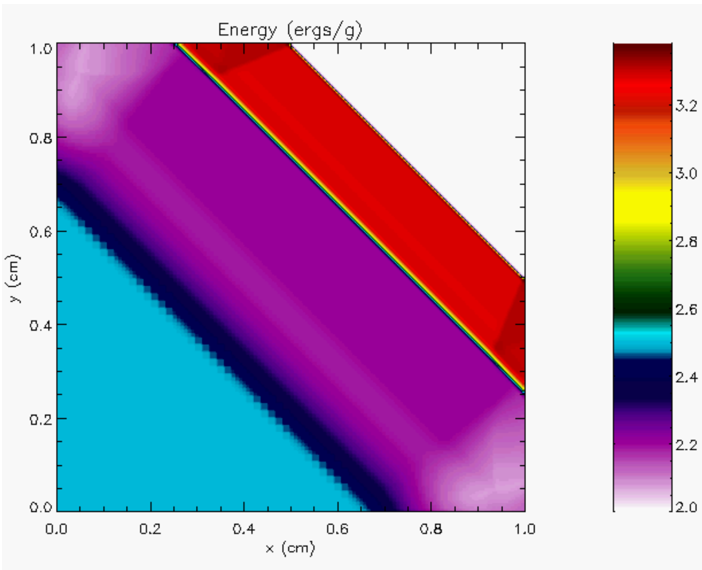


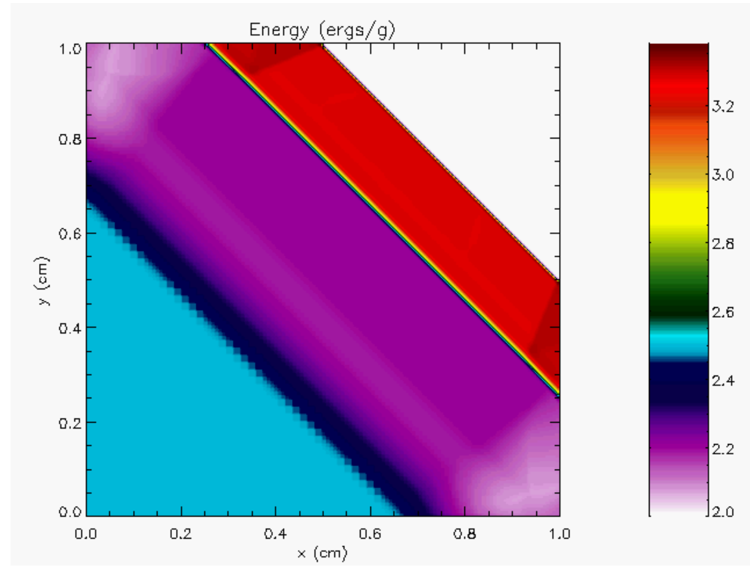



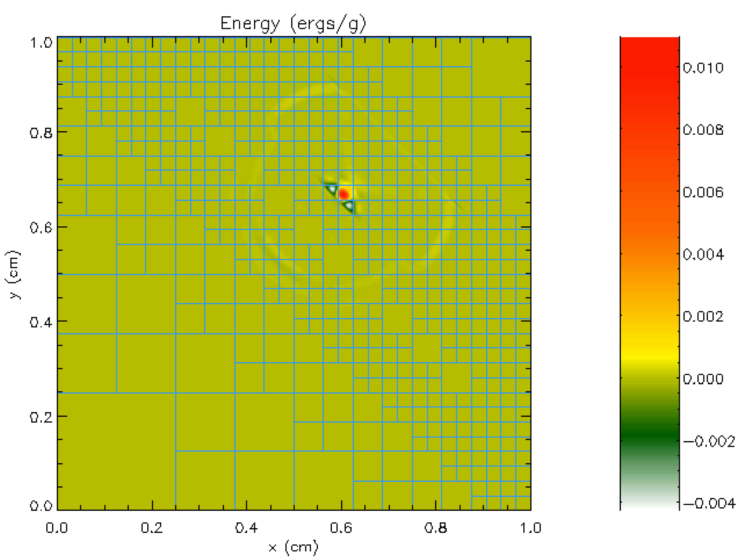



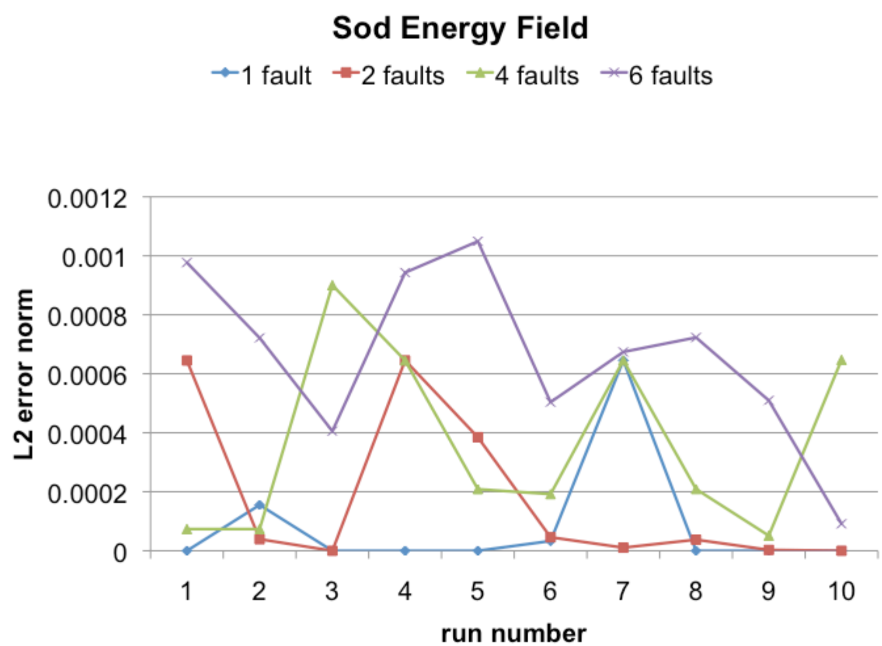



# Sod Results


Energy (ergs/g)
y (cm)
x (cm)


Energy (ergs/g)
y (cm)
x (cm)


Energy (ergs/g)
y (cm)
x (cm)


Sod Energy Field
1 fault
2 faults
4 faults
6 faults
L2 error norm
run number

# Findings

- Completely unexpectedly, the problem with shock proved to have robust recovery, even when the fault was placed right at the shock
- In isentropic vortex the impact was higher soon after the fault occurred, but smoothed out later
  - Implication is that it is within the order of accuracy of the numerics
- The impact is not clearly correlated with the placement of fault, but also with count
  - Though it is possible that the higher count implies greater possibility that one of the faulty block is in the more critical region

# Outline

- Block Structured AMR
- Algorithm
- Test Cases
- Results
- **Future Directions**

# Future Work

- The approach proved to be far more robust than expected
- We have to explore the mathematical arguments for the robustness of the recovery mechanism
- Work on fault injection and detection methods
- Look at rearranging space filling curves and block/ patch distribution
  - Make sure that parent and child blocks are not on the same node
  - Quantify the trade-off in computation cost for resiliency