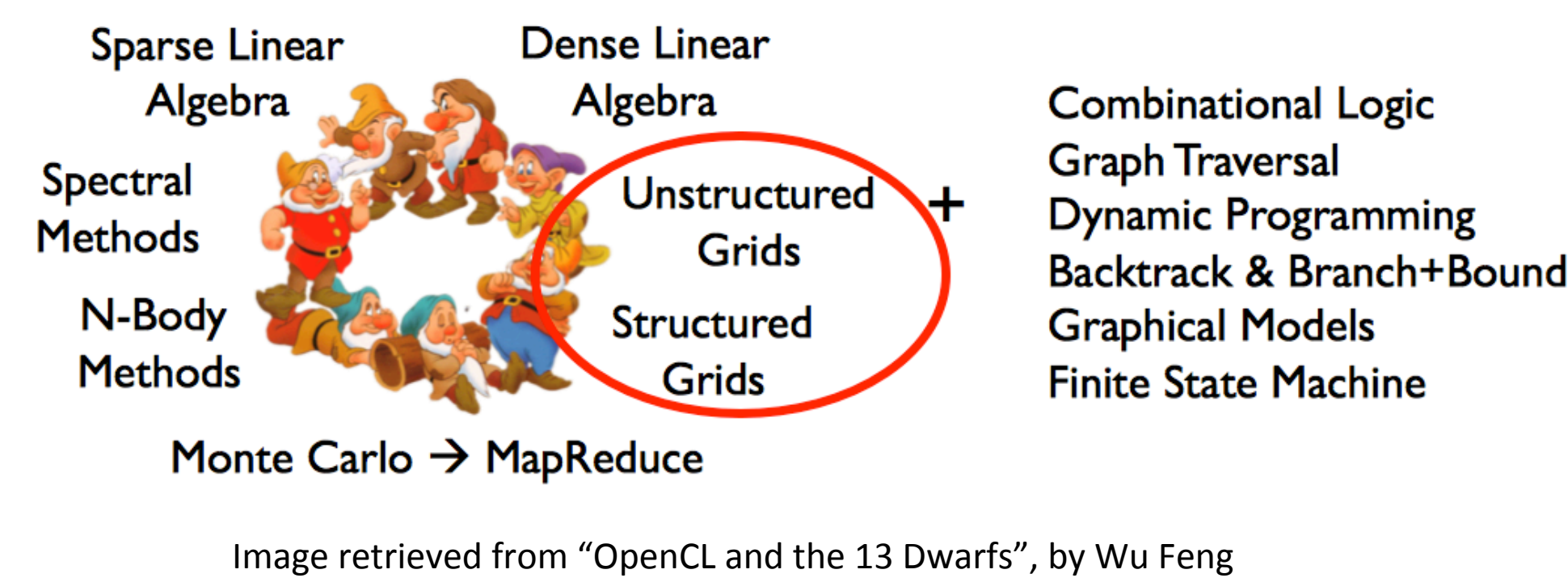


What is CLAMR?

CLAMR is cell based adaptive mesh refinement (AMR) mini-app hydrodynamic simulation modeling shallow water equations. A cylindrical shock is created at the center of the mesh, and symmetry is used in the following time steps to simulate the shock moving towards and away from boundaries. CLAMR was developed at LANL as a test bed for hybrid algorithm development. (<http://github.com/losalamos/CLAMR>)

CLAMR falls into the Unstructured and Structured Grid categories of the original 7 numerical methods important for science and engineering identified by Phil Colella, which have now been expanded to 13.

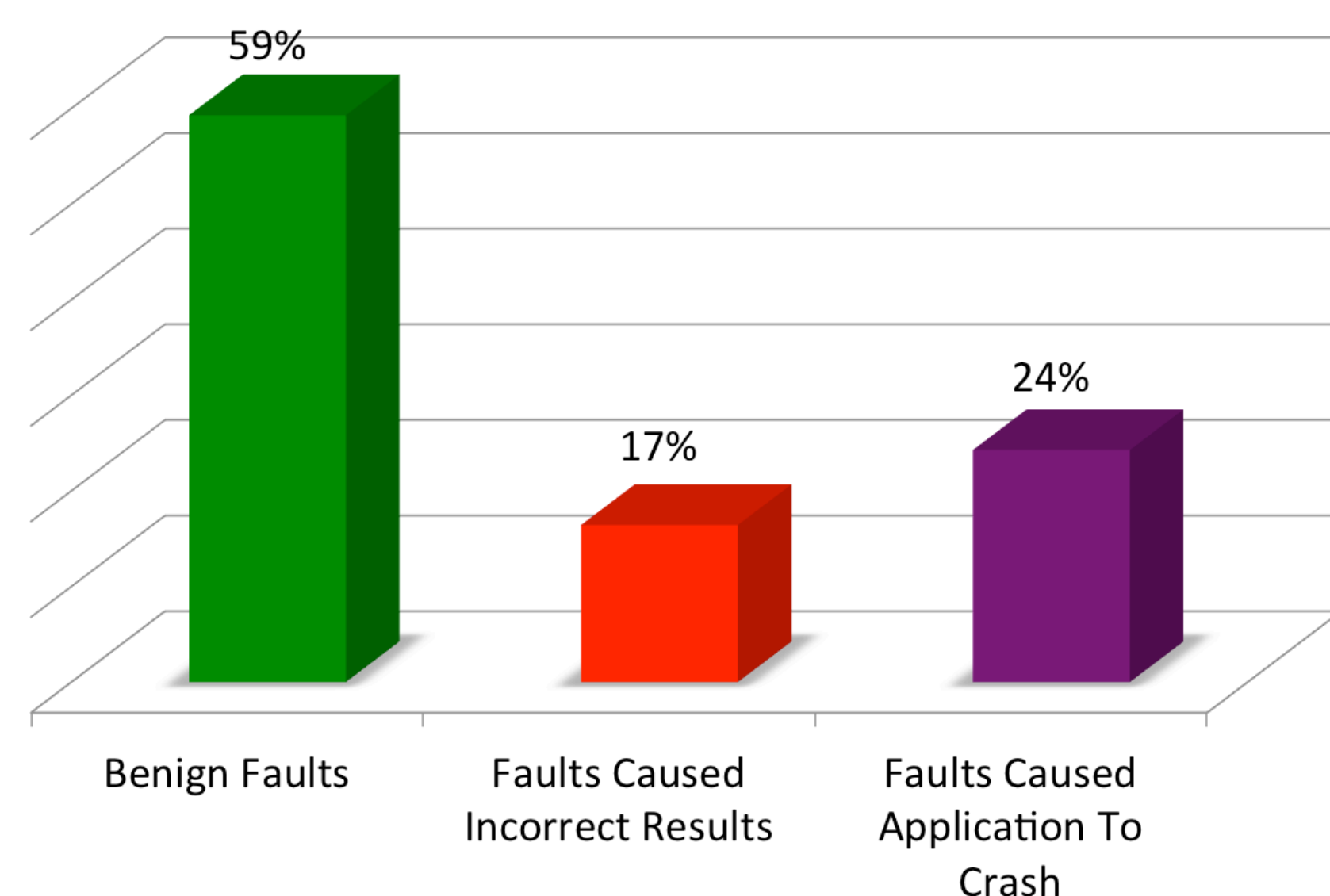


What is F-SEFI?

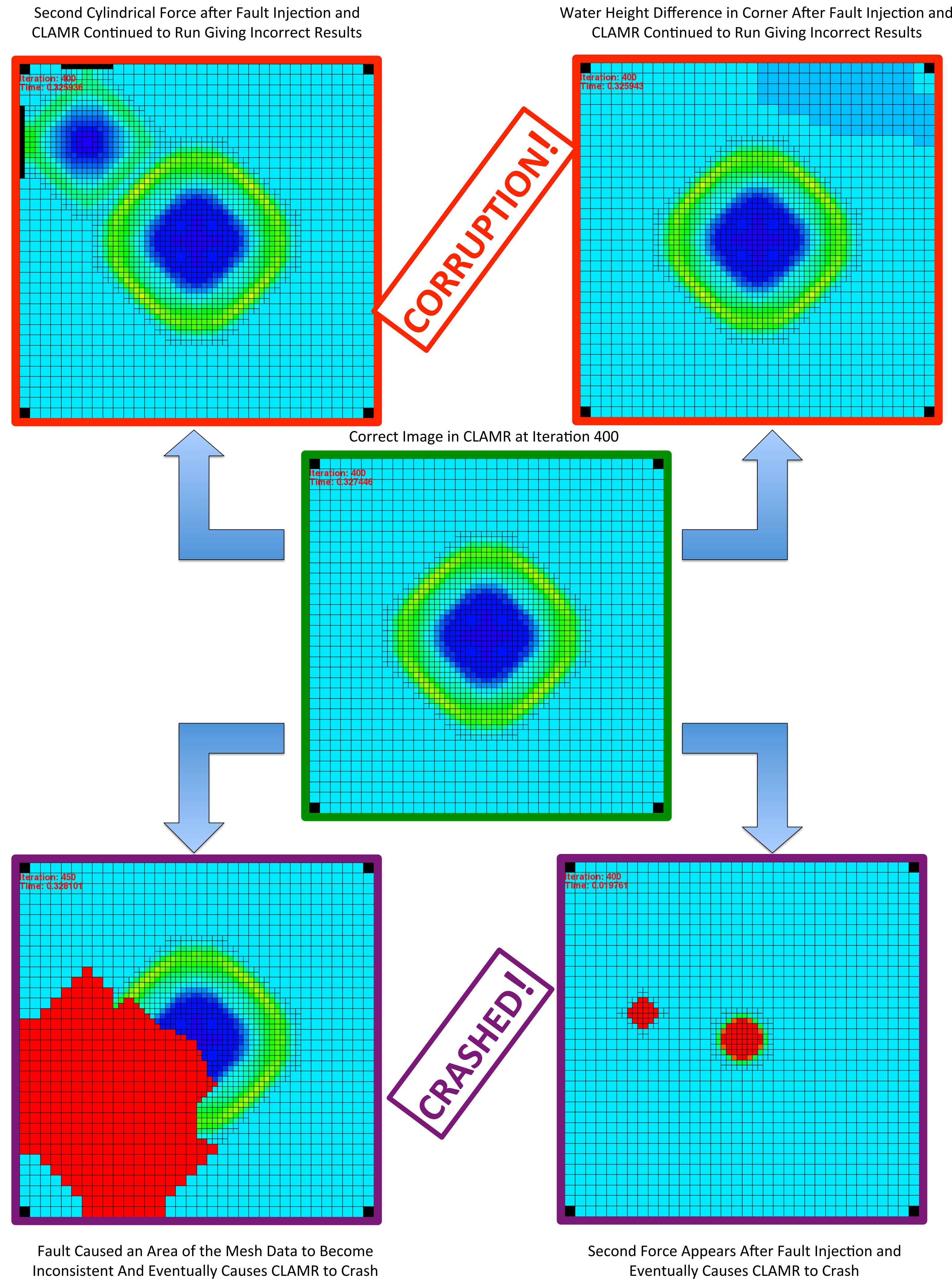
F-SEFI is a software-based fault injector that injects emulated faults into running scientific applications. These faults can be applied to any operation. For this work we focused on floating point operations where injections could target the sign, exponent, or mantissa bit fields. F-SEFI can target specific functions and line numbers of a target application. F-SEFI was designed and developed by USRC, HPC-5, at LANL.

In CLAMR we focused on fault injections into the floating point add operations (FADD) in the exponent bit fields in the calculate finite difference function. This function updates the state variables and mesh values at each time step. The reason we focused on FADD exponent bit fields is because it provided three major cases of fault injection results.

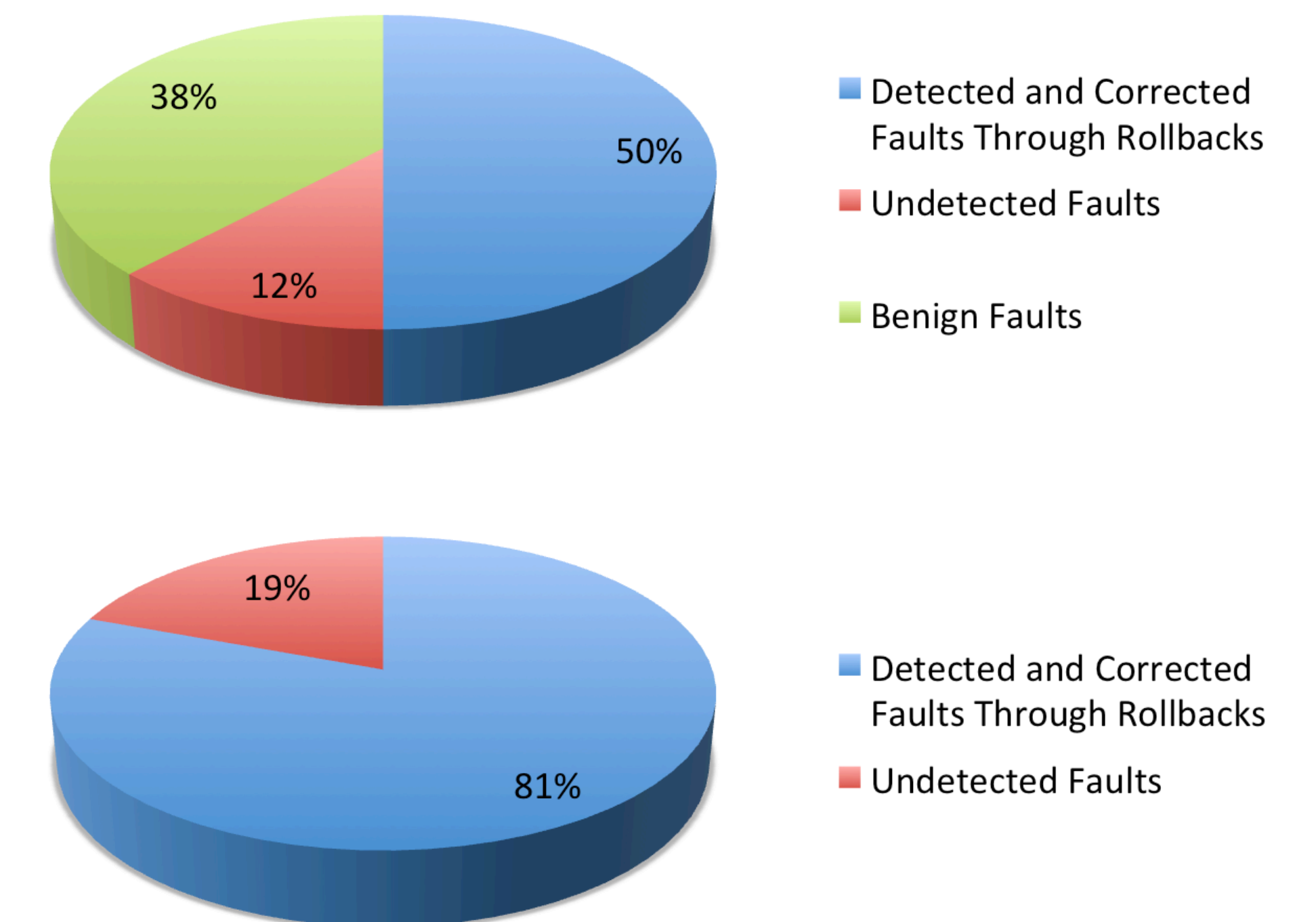
- 1) Benign Faults:** Fault had no impact on the application
- 2) Faults Causing Incorrect Results:** Faults would propagate the simulation continued
- 3) Faults Crashing Application:** Faults led to an inconsistent state within mesh data. These faults eventually caused the CLAMR application to crash.



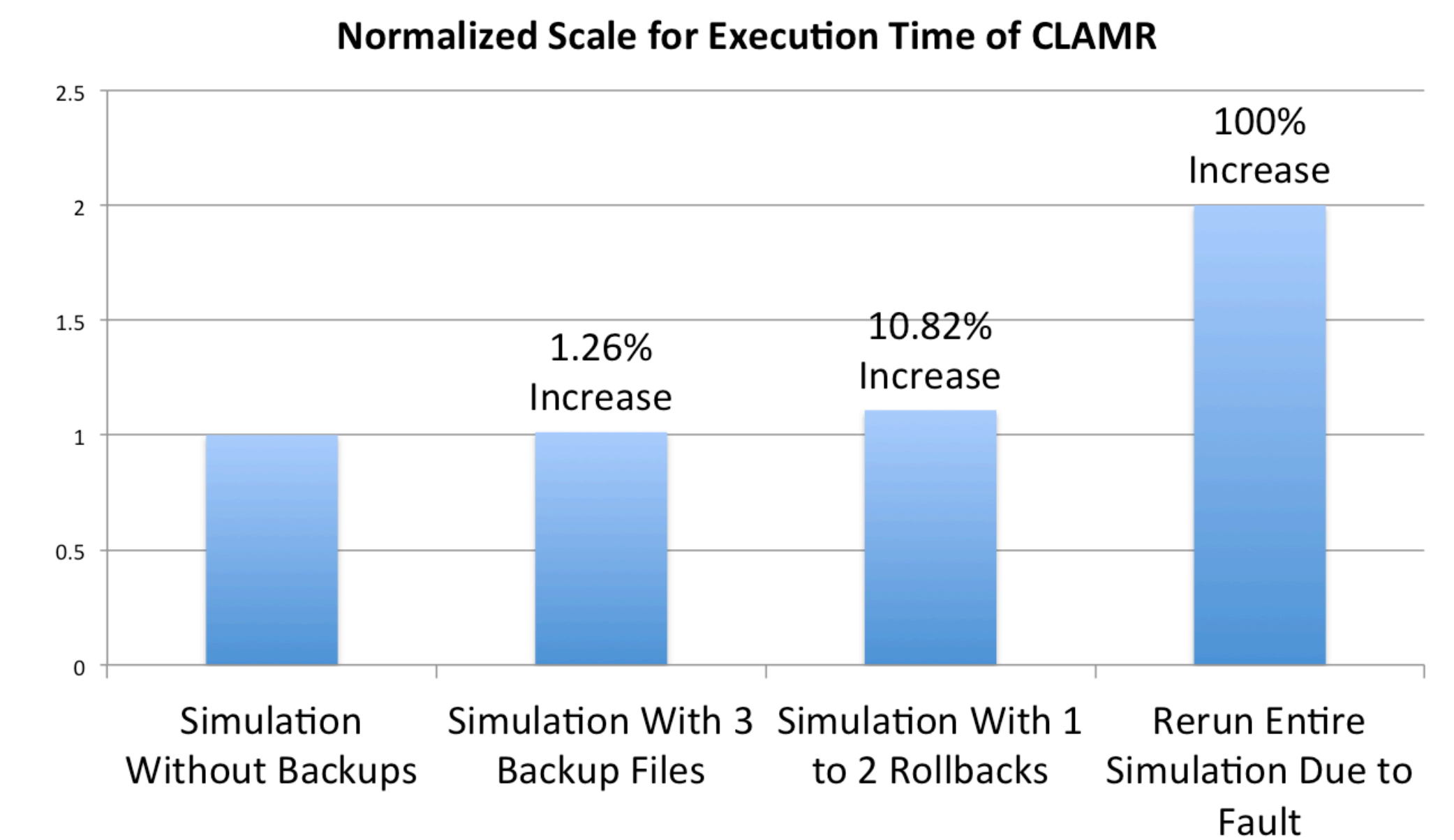
Visualization of Faults in CLAMR



We were able to successfully detect and correct 81% of errors that would have lead to incorrect results or application crashes with the rollback and check pointing features added in CLAMR.



On average, we had only a 1.26% increase to the total execution time of CLAMR with maintaining 3 backup files, created every 100 Iterations, during the simulation. Of cases that required at most 2 rollbacks to recover from a fault, We saw only a 10.82% increase in the total execution time.



Future Work

While we were able successfully recover from 81% of soft error fault injections by using the rollback routines, we believe that we can increase this percentage even higher. This would require another conservation check to better assess the errors that currently eluding the mass conservation check. Also, a better assessment of how many backup files are required based upon the problem size in CLAMR can further reduce the overhead accompanied with creating backups of the state and mesh variables to disk.

Acknowledgments

I would like to thank Nathan DeBardeleben, Qiang Guan, and Sean Blanchard for their guidance in the resilience work using the F-SEFI tool. I would also like to thank Bob Robey for all his help using and understanding CLAMR. This work is supported by Los Alamos Laboratory and the New Mexico Consortium under the Ultrascale Systems Research Center.

Resilience Against Faults

In order to prevent faulty output and application crashes, we implemented check pointing and rollback features into CLAMR's code. These two new capabilities are configurable based on the problem size by the user. The rollback feature is based upon the conservation of mass, inherent to the shallow water equations. If the mass is outside of the allowable percentage difference or has a numerical instability, the rollback feature restarts the application from the last safe checkpoint.

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0$$

Conservation of Mass Equation for Shallow Water

Unclassified