A Resilient ULFM-based PDE Solver: Performance Scaling and Energy Analysis

K.Morris[1], F. Rizzi[1], B. Cook[2], K.Dahlgren[1], K. Sargsyan[1], P. Mycek[3], O. Le Maître[3], O. Knio[3], B. Debusschere[1]

[1]Sandia National Laboratories, [2]Lawrence Berkeley National Laboratory, [3]Duke University

Motivation

Addressing exascale challenges:

- Resiliency, power, memory access, concurrency and heterogeneous hardware.
- Codes/algorithms need to be flexible and extensible to unknown architectures.
- Errors and faults will become more frequent compared to current platforms.

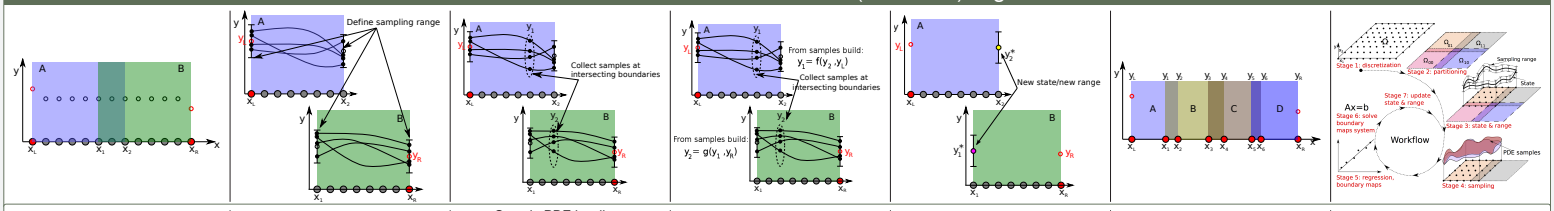
Objectives

- Accurate and robust simulation even in the presence of system faults.
- Assess predictive fidelity of extreme-scale scientific simulations.
- High performance on "uncertain" architectures.
- Portable, extensible and reusable codes.

Technical Approach

- What?
  - Resilient preconditioner for PDEs.
  - To achieve resiliency to both soft and hard faults.
- How?
  - Recast the original PDE as a sampling problem.
  - Focus solely on the information available.
  - State update through resilient data manipulation.

Resilient EXtreme Scale Scientific Simulation (REXSSS) Algorithm Overview



- Initial state
- Partition with overlapping subdomains

- Focus on boundaries
- Define sampling range

- Sample PDE locally
- Collect samples at intersecting boundaries

Regression stage to build maps:
$y_1 \approx a + by_2 + cy_L$
$y_2 \approx d + ey_1 + fy_R$

- y_L, y_R are known BC
- Solve: new state (y_1^*, y_2^*)

Extends to multiple subdomains

2D workflow

Regression stage: Laplace (ℓ_1) vs Gaussian (ℓ_2)

Regression Resilience

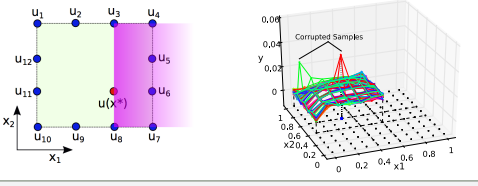
High-dimensional boundary maps: $u(\mathbf{x}^*) = a + bu_1 + cu_2 + \ldots$

Generate samples (some corrupted)

Run regression using collected samples

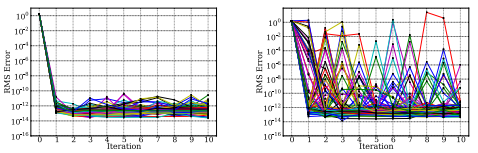
ℓ_1: robust against presence of corrupted data

ℓ_2: highly sensitive to corrupted data



1D Elliptic PDE Test

- Bit-flips inserted during sampling with probability p.
- Laplace likelihood yields resiliency to soft faults.

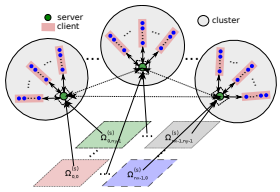


Ensemble runs for ℓ_1 and $p = 0.003$.

Ensemble runs for ℓ_2 and $p = 0.003$.

REXSSS Implementation

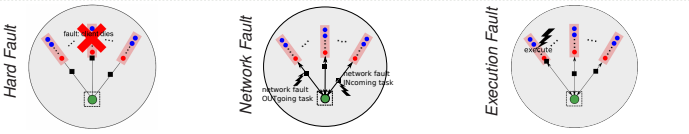
Server-Client Model (SCM)



- Cluster: 1 server + n clients.
- Servers:
  - Communicate between each other.
  - Safe data/state storage (sandboxed).
- Clients:
  - Independent from one another.
  - Only serve as computing units.

- Separates state from computation: reduces the overall vulnerability.
- Fault-tolerance via *User Level Fault Mitigation* ULFM-MPI (`fault-tolerance.org`).
- It aligns with the vision of future exascale architectures involving heterogeneous and hierarchical hardware required to meet energy and cost constraints.

Currently Supported Fault Models



Hard Fault
What? Client dies, lost task.
How? Client ranks terminated.

Network Fault
What? Data corruption during communication.
How? Random bit-flips on task's data.

Execution Fault
What? Data corruption during computation.
How? Random bit-flips on task's data.

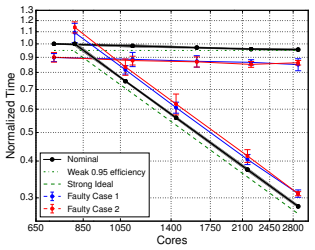
ULFM-MPI and SCM: Why is this a good combination?

- Server simply continues the execution using only the clients that are alive.
- Avoid ULFM-MPI collective procedures to rebuild the client/communicators.
- Servers probe the corresponding cluster communicator using `MPI_ANY_SOURCE` to assess whether a new message is arriving from one of the clients.

Scalability

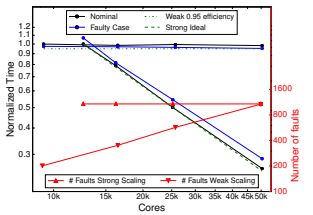
Uncertainty due to SDCs and client failures

- 1% failed clients, SDC=0.1% total tasks (–)
- 2% failed clients, SDC=0.1% total tasks (–)
- Times normalized by smallest nominal case: highlights scaling and overhead.
- Faulty runs show larger variability due to random loss of resources, and additional regression overhead to overcome silent data corruptions (SDCs).
- Red curve has larger overhead because the number of failed clients doubles.



Scaling on Edison (NERSC)

- Of total faults: SDCs=97%, ranks failed=3.0%.
- Constant machine fault rate.
- Times normalized by smallest nominal case: highlights scaling and overhead.
- Excellent scalability with and without faults:
  - ✓ within 95% for weak scaling.
  - ✓ within 90% for strong scaling.
- Overhead with respect to nominal case:
  - a downward shift for weak scaling.
  - an upward shift for strong scaling.



Energy and Resilience

Exploit resilience for energy purposes?

- Idea: lower the energy consumption during the *sampling stage* by means of voltage scaling.
- Compare three scenarios:
  (A) machine running at full operational capacity/speed
  (B) voltage/freq scaling on clients during sampling
  (C) voltage scaling on clients during sampling
- Same problem, SC configuration, and machine.
- Servers run at full capacity to keep the state safe.
- Power consumption (P) and energy (E) over time T:

$$P = \hat{P} + \mathcal{C}V^2 f \qquad E = (\hat{P} + \mathcal{C}V^2 f)T.$$

frequency independent active power ($\hat{P}$), switch capacitance ($\mathcal{C}$), voltage (V), frequency (f).

Energy Budget for Sampling Stage

(A) Full operational mode:
- V_A, f_A, t_A^s = time for one task.
- For N_A samples: $E_A^s = N_A(\hat{P}t_A^s + \mathcal{C}V_A^2 f_A t_A^s)$

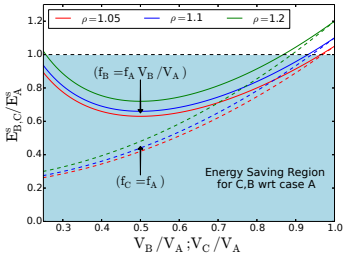
(B) Reduced voltage/frequency:
- $V_B < V_A$, $f_B < f_A$, $t_B^s = t_{A f_B}^{s\, f_A}$.
- For $N_B = \rho N_A$ samples: $E_B^s = \rho N_A(\hat{P}t_{A f_B}^{s\, f_A} + \mathcal{C}V_A^2 f_A t_{A f_B}^{s\, f_B^2})$

(C) Reduced voltage:
- $V_C = \gamma V_A$, with $\gamma < 1$, $f_C = f_A$, $t_C^s = t_A^s$.
- For $N_C = \rho N_A$ samples: $E_C^s = \rho N_A(\hat{P}t_A^s + \mathcal{C}\gamma^2 V_A^2 f_A t_A^s)$.

ρ is the oversampling needed to guard against faults.

Results

Energy ratios for cases B,C wrt to A showing saving regimes for different oversampling.



(1) K. Morris, F. Rizzi, B. Cook, P. Mycek, O. Le Maître, O. Knio, K. Sargsyan, K. Dahlgren and B. Debusschere, *Performance Scaling Variability and Energy Analysis for a Resilient ULFM-based PDE Solver*, Scala16, 2016.
(2) K. Sargsyan, F. Rizzi, P. Mycek, C. Safta, K. Morris, H. Najm, O.Le Maître, O. Knio and B. Debusschere, *Fault resilient domain decomposition preconditioner for PDEs*, SIAM J. Sci. Comp., 37(5), 2015.
(3) F. Rizzi, K. Morris, K. Sargsyan, P. Mycek, C.Safta, O.Le Maître, O. Knio and B. Debusschere, *Exploring the Interplay of Resilience and Energy Consumption for a Task-Based Partial Differential Equations Preconditioner*, PP4REE, co-located with PPoPP16, 2016.
(4) P. Mycek, F. Rizzi, O.Le Maître, K. Sargsyan, K. Morris, C.Safta, B. Debusschere and O. Knio, *Discrete a priori bounds for the detection of corrupted PDE solutions in exascale computation*, SIAM J. Sci. Comp., 39(1), 2017.

E-Mail: knmorri@sandia.gov

U.S. DEPARTMENT OF ENERGY