ULFM-MPI Implementation of a Resilient Task-Based Preconditioner for 2D Uncertain Elliptic PDEs

SAND2016-3175C

F.Rizzi[†], K.Morris[†], K.Sargsyan[†], P.Mycek[‡], C.Safta[†], O.LeMaitre[‡], O.Knio[‡], B.Debusschere[†]

[†] Sandia National Laboratories, Livermore, CA
[‡] Duke University, Durham, NC

SIAM UQ-2016
– April 2016 –

Supported by the US Department of Energy (DOE)
Advanced Scientific Computing Research (ASCR)

Towards Exascale: Energy & Resilience

Energy is the biggest obstacle to reach exascale.

- Titan, at Oak Ridge: 8.2 MW.
- Scale up to exascale: $\sim 100/200$ MW.
- Cost: $\sim$ a third of billion dollars/year.
- DOE: 20 MW by $\sim$ 2023.
- Reducing power is needed but...
- ...it compromises system resilience.

- Smaller transistors running at lower voltages increase the probability of circuits flipping state spontaneously.
- Powering off every unused chip and turning them on when needed: chips lifetime reduced up to 25 percent (UM 2009).
- Voltage fluctuations throughout the system, too large of a voltage fluctuation cause circuits to switch on or off spontaneously.

Towards Exascale: Energy & Resilience

Energy is the biggest obstacle to reach exascale.

- Titan, at Oak Ridge: 8.2 MW.
- Scale up to exascale: $\sim 100/200$ MW.
- Cost: $\sim$ a third of billion dollars/year.
- DOE: 20 MW by $\sim$ 2023.
- Reducing power is needed but...
- ...it compromises system resilience.

- Smaller transistors running at lower voltages increase the probability of circuits flipping state spontaneously.
- Powering off every unused chip and turning them on when needed: chips lifetime reduced up to 25 percent (UM 2009).
- Voltage fluctuations throughout the system, too large of a voltage fluctuation cause circuits to switch on or off spontaneously.

Towards Exascale: Energy & Resilience

- Do we need resilience?
- Yes? No? Maybe?
- Feb.16, article by Al Geist, ORNL.
- Many factors affecting resilience.

IEEE SPECTRUM

How To Kill A Supercomputer: Dirty Power, Cosmic Rays, and Bad Solder

Will future exascale supercomputers be able to withstand the steady onslaught of routine faults?

By Al Geist
Posted 23 Feb 2016 | 16:00 GMT

"As a child, were you ever afraid that a monster lurking in your bedroom would leap out of the dark and get you?"

"Lack of resilience is a similar monster, hiding in the steel cabinets of the supercomputers and threatening to crash the largest computing machines."

Towards Exascale: Challenges & Needs

- Software challenges are also daunting.
- Checkpoint/restart: won't work indefinitely. Eventually, the interval will become longer than the typical period before the next fault.
- Applications will have to be rewritten to withstand a constant barrage of faults and keep on running.
- Applications will need to make use of new architectures.

Problem description

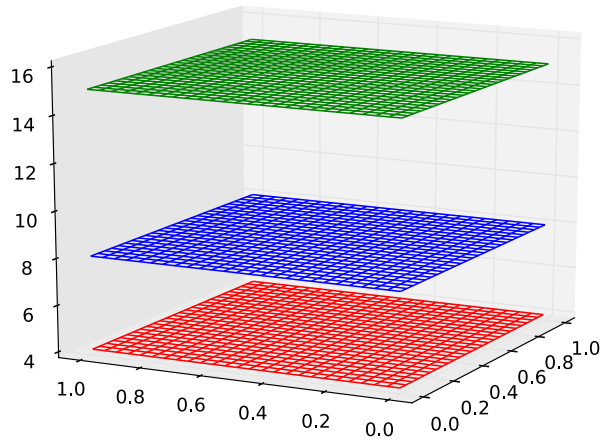
2D Diffusion equation with uncertain diffusivity field κ:

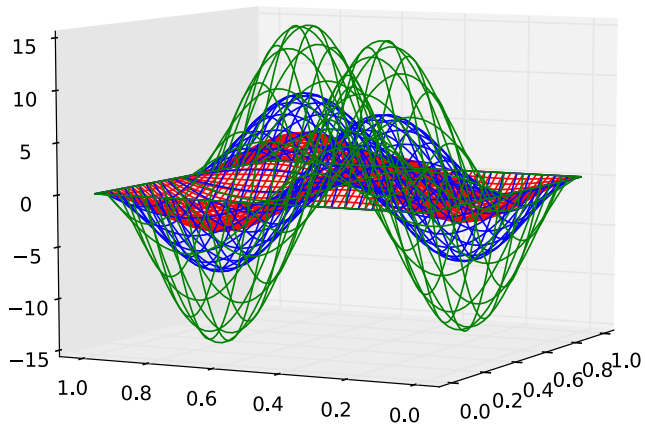
$$\begin{cases} \kappa(\xi)\partial_x^2 u(x,\xi) = g(x_1,x_2) & \text{in } \Omega = (0,1)^2 \\ u(x) = 0, \quad \forall x \in \partial\Omega, \end{cases}$$

where $\xi \sim \mathcal{N}(0,1)$ (Gaussian standard random variable).

Each realization of ξ defines a new diffusivity field $\kappa(\xi)$.

Diffusivity $\Rightarrow$ **Solution**

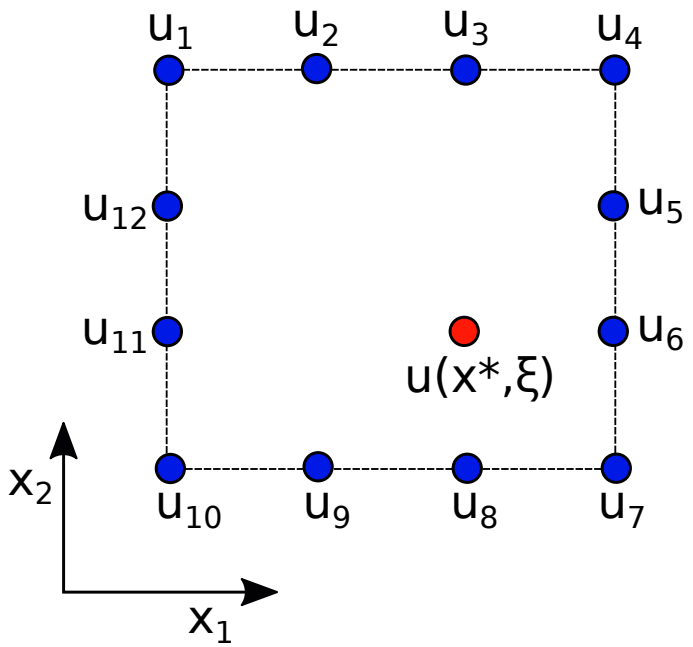




Problem description

- The solution at any internal point x^* **linearly depends** on the boundary conditions:

$$u^* \equiv u(\boldsymbol{x}^*, \xi) = a(\xi) + b(\xi)u_1 + c(\xi)u_2 + \dots$$



- For 2D dimensionality is larger than 1D... (reduction is in progress).

Polynomial chaos expansion (PCe) of coefficients

- Polynomial Chaos Expansion: series representation of a stochastic process through a spectral expansion using orthogonal polynomials.
- Original map: $u(\boldsymbol{x}^*, \xi) = a(\xi) + b(\xi)u_1 + c(\xi)u_2 + \dots$
- Spectral Polynomial Chaos approximation of the map coefficients:

$$a(\xi) \approx \sum_{k=1}^{P} a_k \Psi_k(\xi) \quad b(\xi) \approx \sum_{k=1}^{P} b_k \Psi_k(\xi) \quad \dots$$

  where the Ψ_k are Hermite (orthonormal) polynomials in ξ.
- The map becomes:

$$\begin{aligned} u(\boldsymbol{x}^*, \xi) &= a(\xi) + b(\xi)u_1 + c(\xi)u_2 + \dots \\ &\approx \sum_{k=1}^{P} a_k \Psi_k(\xi) + u_1 \sum_{k=1}^{P} b_k \Psi_k(\xi) + u_2 \sum_{k=1}^{P} c_k \Psi_k(\xi) + \dots \end{aligned}$$

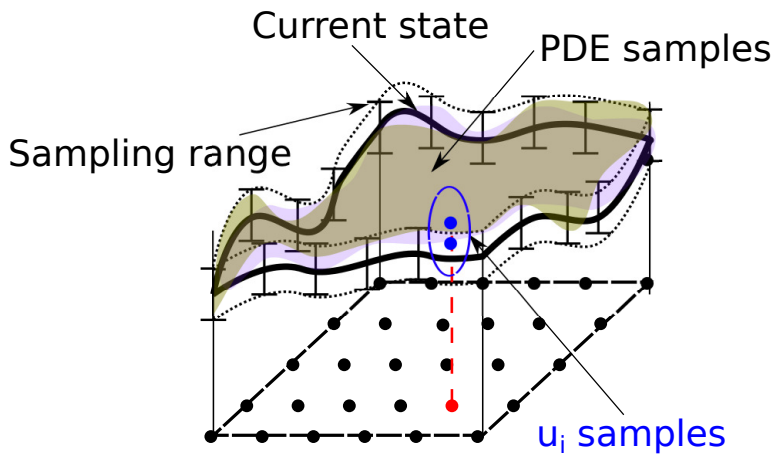
Regression approach to find the map coefficients

The map has the general form:

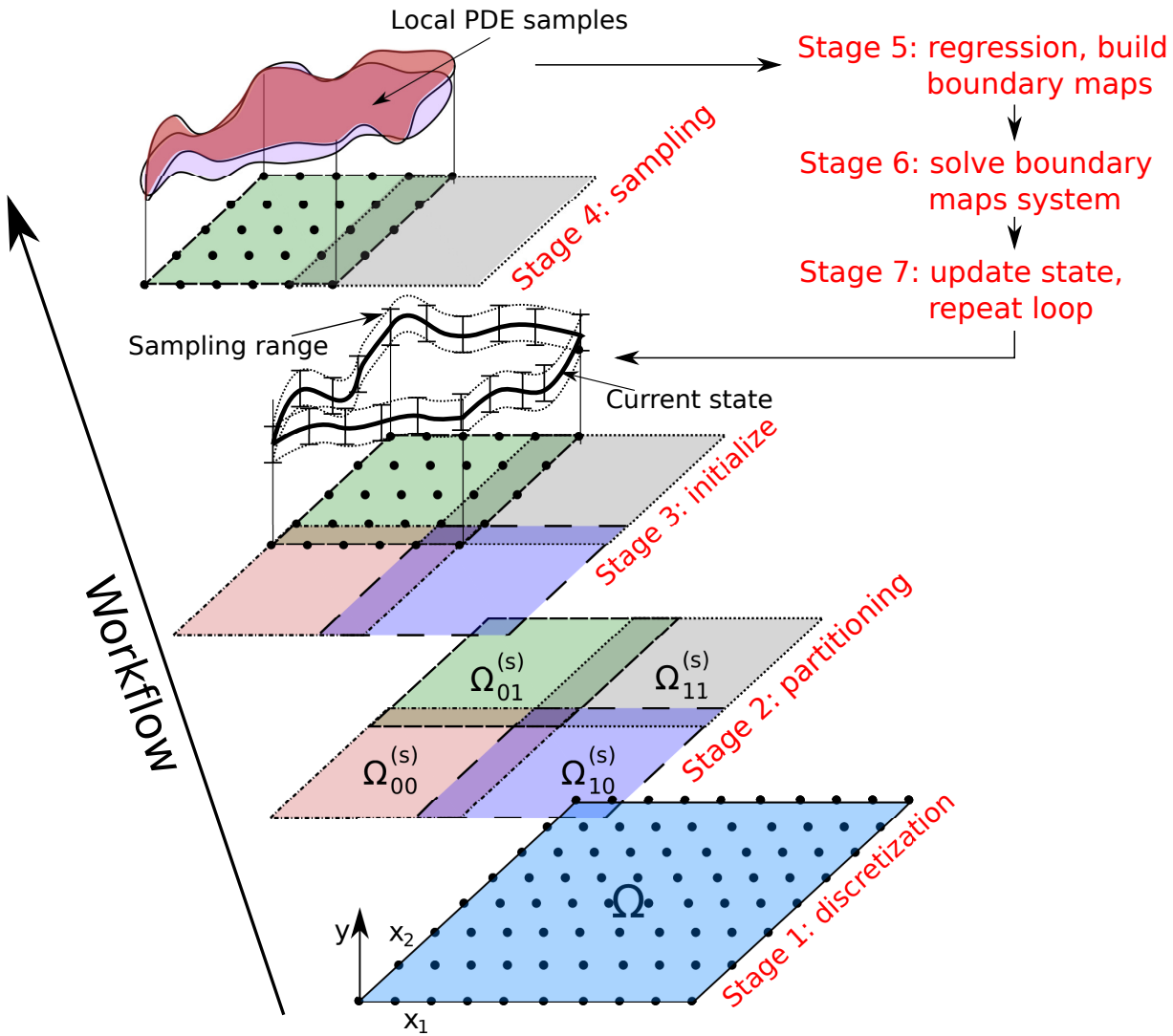
$$u(\boldsymbol{x}^*, \xi) \approx \sum_{k=1}^{P} a_k \Psi_k(\xi) + u_1 \sum_{k=1}^{P} b_k \Psi_k(\xi) + u_2 \sum_{k=1}^{P} c_k \Psi_k(\xi) + \dots$$

Sampling approach:

- Sample the BCs and the $\xi \Rightarrow (u_1^{(i)}, u_2^{(i)}, ..., \xi_i)$
- Collect solution values u_i for each realization $(u_1^{(i)}, u_2^{(i)}, ..., \kappa(\xi_i))$
- Regression problem to compute coefficients.



Extension: Domain Decomposition



Many local problems.

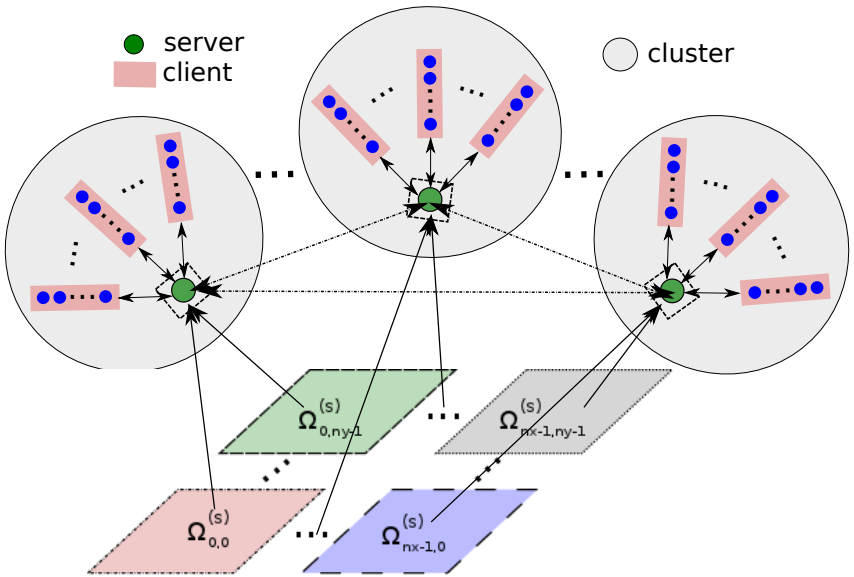
Compatibility conditions at boundaries.

Solve compatibility equations in a weak sense.

Obtain solution at boundaries.

Solution is a PCe capturing dependence on diffusivity.

Server/Client-based Implementation

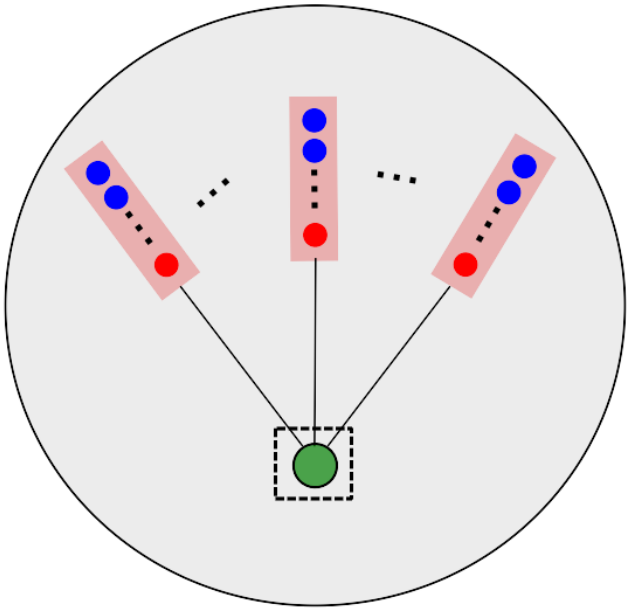


- Cluster: 1 server + *n* clients.
- Servers:
  - Communicate between each other.
  - Safe data/state storage (sandboxed).
- Clients:
  - Independent from one another.
  - Only serve as computing units.

- Separates state from computation: reduces the overall vulnerability.
- Fault-tolerance supported via ULFM-MPI: resilient to clients crashing.
- Resilient to clients crashing because even if tasks are lost, state is safe.
- It aligns with the vision of future exascale architectures involving heterogeneous and hierarchical hardware required to meet energy and cost constraints.
- C++ code, two external dependencies: Trilinos and Boost.

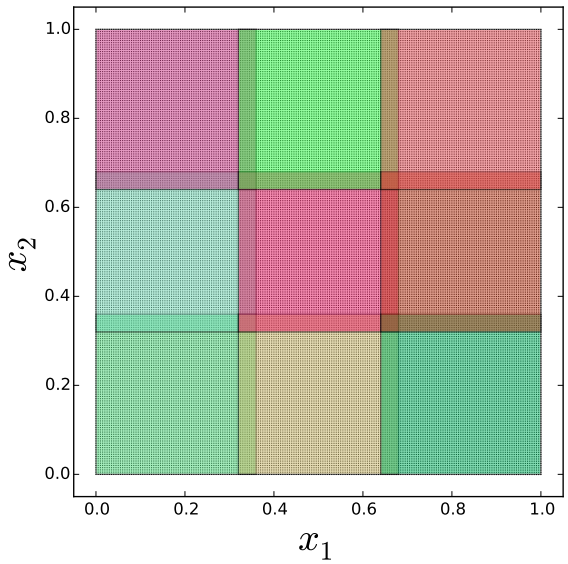
ULFM and SCM: Why is this a good combination?

- Hard faults modeled as clients crashing.
- When a client crashes, the server simply continues the execution using only the clients that are alive.
- Avoid expensive collective procedures of ULFM to rebuild the client and fix any communicator.
- Communicator reconstruction is expensive.



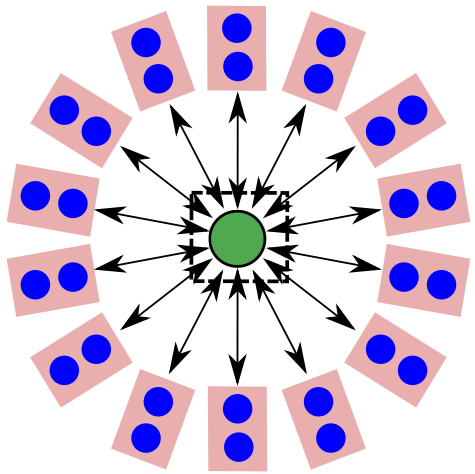
Resilience: Test Problem

- 2D linear elliptic equation.
- 251^2 grid, 3x3 subdomains.



Partitioning

- 1 **server**, 14 clients size 2.
- Faults affect clients only.



SC configuration

Resilience Details: SDC and Hard Faults

Silent Data Corruptions (SDC)

- SDC: transient and do not cause the termination of the application.
- Faults injection: percentage of tasks, $0.25, 0.5, 1\%$.
- Selective reliability: only affect sampling stage.
- Bit-flip model: random bit-flip in binary representation.
- Injection: corrupt all boundary conditions of a task.

Hard Faults

- Permanent, cause the termination of the application.
- Faults injection: $2, 4, 6$ clients crashing.
- Selective reliability: only affect sampling stage.

Resilience Details

Silent Data Corruptions (SDC)

- Resilience condition: out of the samples used in the regression, the number of uncorrupted samples has to be greater than the minimum set needed to have a mathematically well-posed regression problem.
- Oversampling: $\rho > 1$, such that $N = \rho N^s_{nom}$.
- Filter $(-100, 100)$ to eliminate outrageous data (expert opinion).

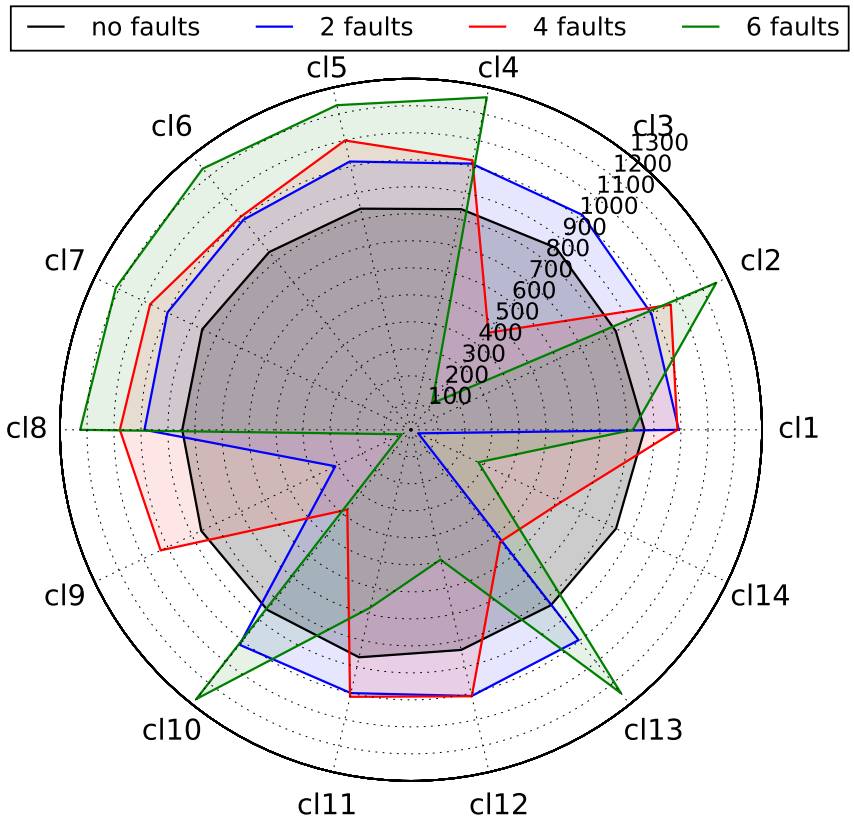
Hard Faults

- No need for ULFM collectives to rebuild broken communicators.
- Server continues the execution using only the clients that are alive.

- Analyze hard faults only.
- Hard and soft faults together.

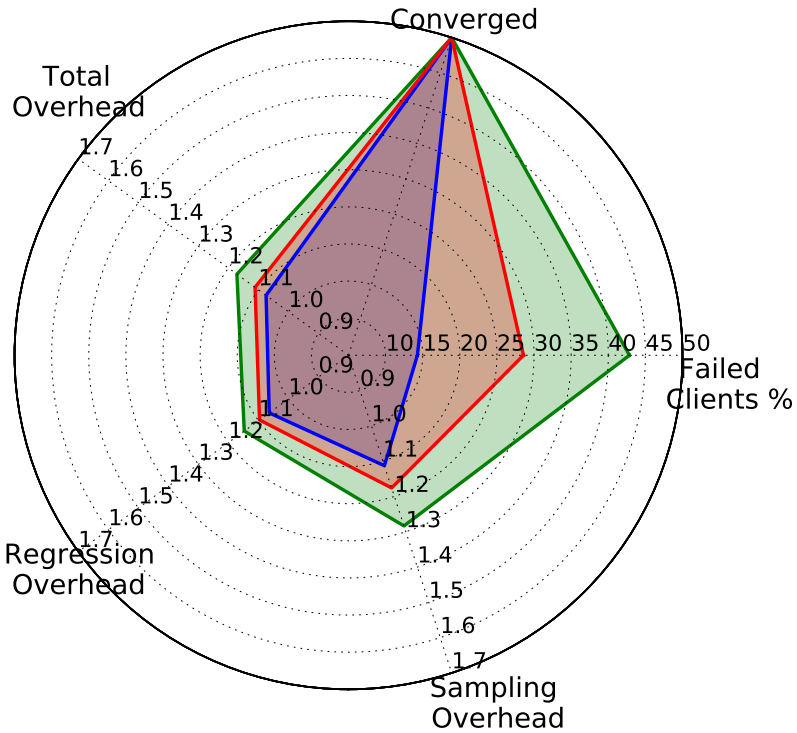
Hard Faults Only

- Hard faults randomly happening during the sampling.
- Angular direction = client name.
- Data = total number of tasks being handled during the simulation.
- No-fault case: workload is fairly uniform
- As expected, increasing the number of faults causes the clients that are alive to handle more and more tasks to compensate for those that are dead.



Clients Workload.

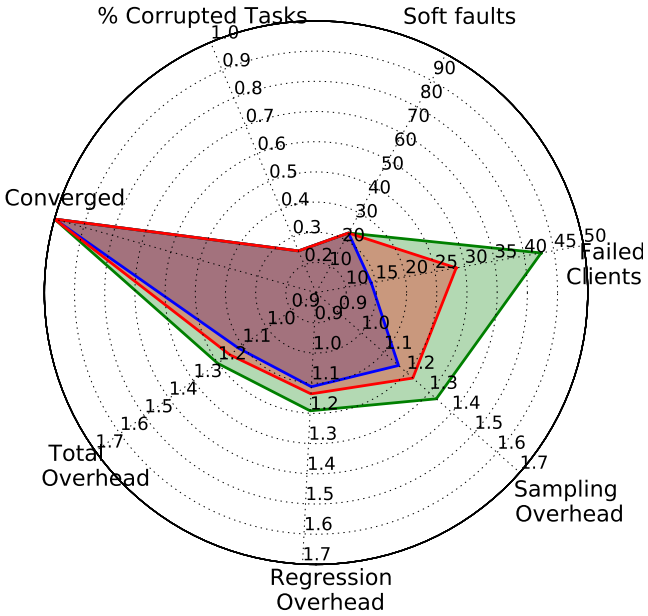
Hard Faults Only



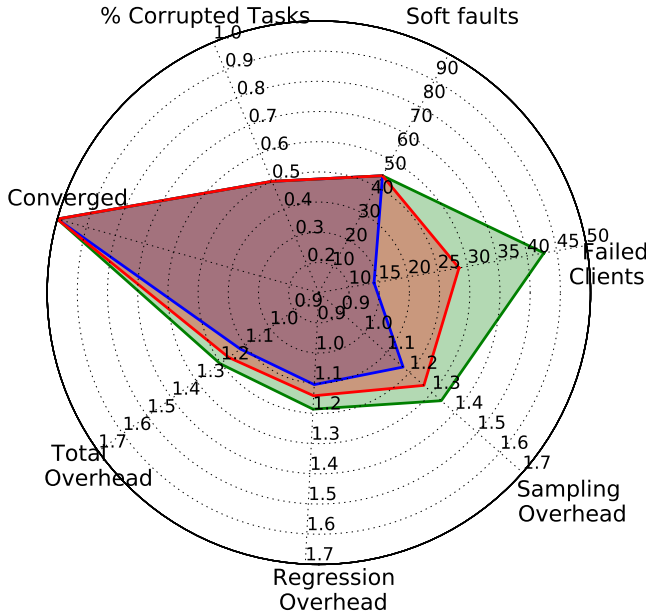
- Runs converge in all cases.
- Losing 14%, 28% and 42% of the clients yields, respectively, a total overhead of 8%, 11% and 18%.
- Resilience ensured with 5% oversampling: measured sampling overhead is larger because clients crash randomly during execution.

Hard Faults and SDC

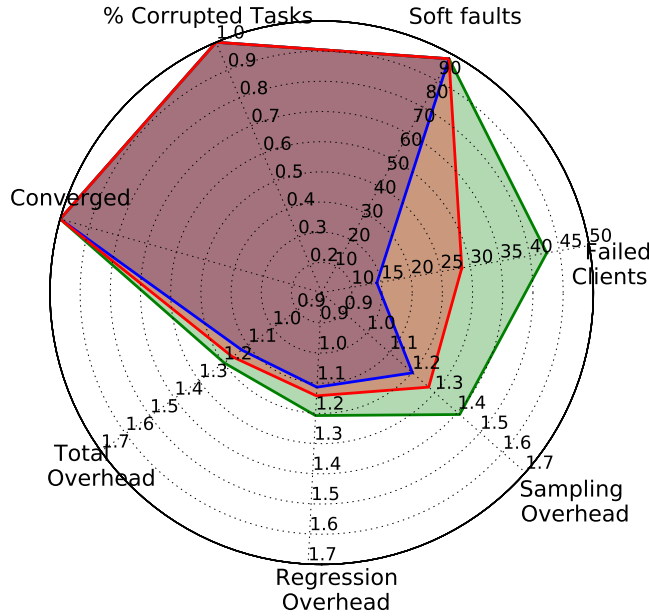
0.25 % SDC



0.5 % SDC



1.0 % SDC



- Consider 4 hard faults (red); *four-fold* increase in SDC from 25 to 98 causes the sampling overhead to only increase from 21% to about 28%.
- Regression overhead only increases from 15% to about 17%.
- This yields the total overhead to only increase from 15% to 19%.
- Resilience ensured with 10% oversampling.

Power Consumption

- Resilience and energy consumption are tightly linked: voltage decrease is linked to higher faults rates. [a]
- Decreasing the energy consumption is possible via variable-voltage CPUs, which can reduce power consumption quadratically at the expense of linearly reduced speed . [b]
- Circuit delay is almost linearly related to $1/V$, where V is the voltage, so for systems to function correctly, the operating frequency needs to decrease linearly when supply voltage decreases [a,b].

---

[a] D. Zhu, R. Melhem, and D. Mosse, "The effects of energy management on reliability in real-time embedded systems", IEEE-ACM International Conference, 2004

[b] D. Zhu, R. Melhem, D. Mosse, and E. Elnozahy, "Analysis of an energy efficient optimistic tmr scheme", ICPADS 2004

Power Consumption

- How to exploit the resilience of the application and the small overhead for energy purposes?
- Idea: lower the energy consumption during the *sampling stage* by means of voltage scaling.
- Compare two scenarios:
  (A) assume that the machine runs at full operational capacity/speed
  (B) decrease the energy consumption of the clients during the sampling by voltage scaling.
- Same problem, same SC configuration, same machine.
- The servers always run at full capacity to keep the state safe.
- This framework can be enabled because of the SC, which allows us to separate state from computation.

Power Consumption

Power consumption and energy over $T = t_2 - t_1$ [a]:

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

$\hat{P}$ = frequency independent active power

C is the switch capacitance

V is the voltage, and f is the frequency.

no sleep power: system always on.

[a] D. Zhu, R. Melhem, and D. Mosse, IEEE-ACM, 2004

Full operational mode

- V_m, f_m
- $T_1 =$ time for one task.
- Energy for N_1 samples:

$$E_1^s = N_1 \left(\hat{P} T_1 + C V_m^2 f_m T_1 \right)$$

Reduced voltage mode

- $V_2 < V_m, f_2 < f_m$
- $T_2 = T_1 \frac{f_m}{f_2}$.
- Energy for $N_2 = \rho N_1$ samples:

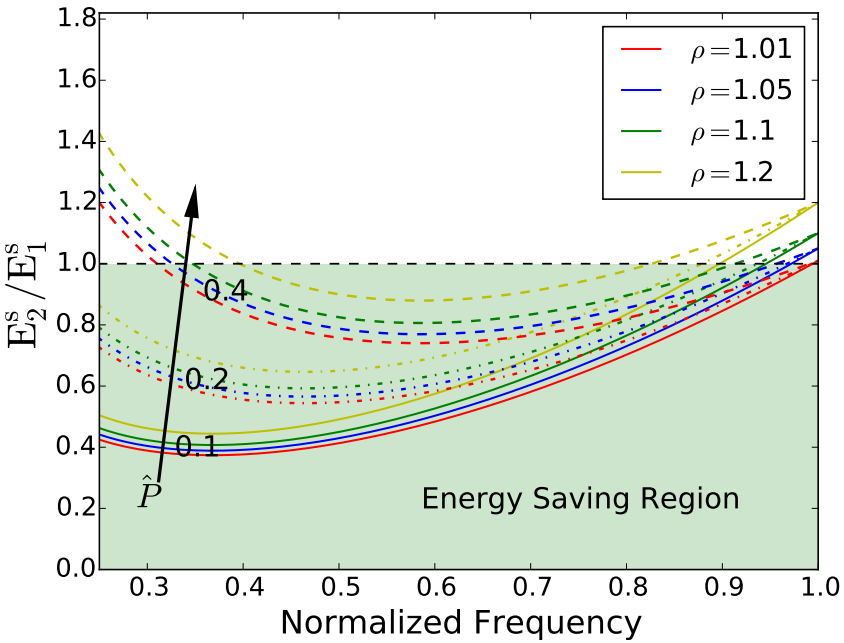
$$E_2^s = N_2 \left(\hat{P} T_1 \frac{f_m}{f_2} + C V_m^2 f_m T_1 \frac{f_2}{f_m} \right)$$

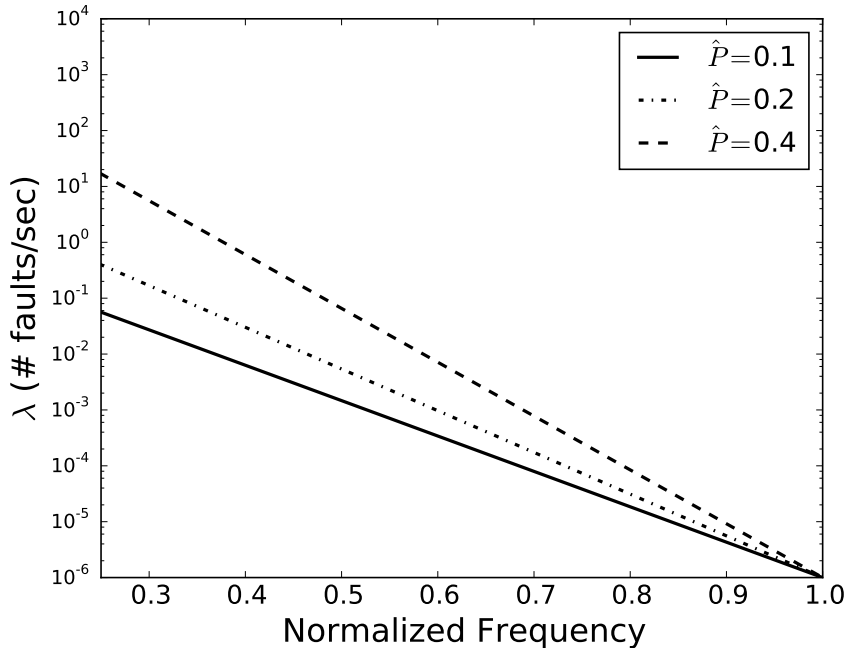
Power Consumption

- Voltages/frequencies normalized; $V_m = 1$ and $f_m = 1$, all within $(0, 1)$.
- Ratio E_2^s / E_1^s as a function of the frequency f_2; recall $\rho = N_2/N_1$.
- Voltage scaling causes fault rates increase exponentially.

$$\lambda = \lambda_0 10^{\left(d \frac{1-f}{1-max\{f_{low}, f_2^*\}} \right)},$$

λ_0 =fault rate corresponding to V_m, f_m, d = sensitivity constant.





Conclusions and Ongoing Work

- Application is resilient to:
  - Silent Data Corruptions during sampling.
  - Missing data due to communication issues or node failures.
- Sampling/decomposition approach provides concurrency/parallelism.
- Convergence is achieved in all cases.
- Scalability is excellent.
- Ongoing work/outlook:
  - Dimensionality reduction.
  - Extension to other types of PDE.
  - Other faults?

Acknowledgments

- This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number 13-016717.
- Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.