Exceptional service in the national interest

Sandia National Laboratories

Resilient Programming Models

Collaborators:
James Elliott
Mark Hoemmen
Keita Teranishi

Michael A. Heroux
Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

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. SAND NO. 2011-XXXXP

Our Luxury in Life (wrt FT/Resilience)

The privilege to think of a computer as a *reliable, digital* machine.

Four Resilient Programming Models

Sandia National Laboratories

- Relaxed Bulk Synchronous (rBSP)
- Skeptical Programming. (SP)
- Local-Failure, Local-Recovery (LFLR)
- Selective (Un)reliability (SU/R)

Toward Resilient Algorithms and Applications
Michael A. Heroux arXiv:1402.3809v2 [cs.MS]

Performance Variability is a Resilience Issue

Sandia National Laboratories

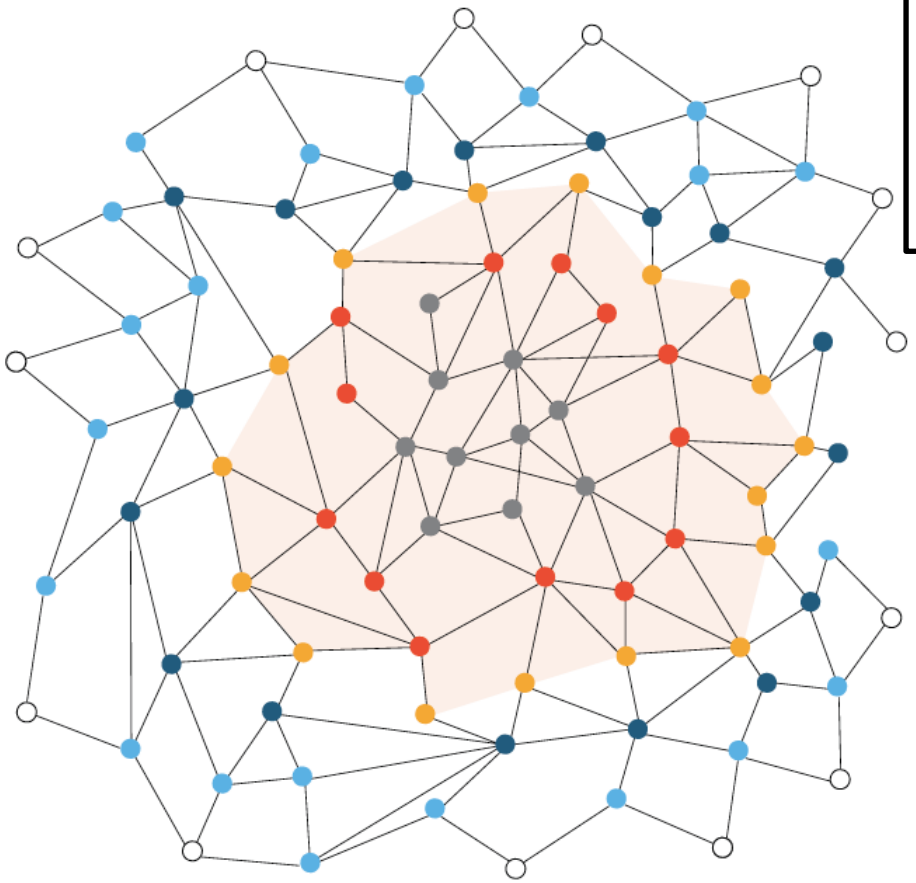
- First impact of unreliable HW?
  - Vendor efforts to hide it.
  - Slow & correct vs. fast & wrong.
  - Variable vs. fast & hot or slow & cool.
- Result:
  - Unpredictable timing.
  - Non-uniform execution across cores.
- Blocking collectives:
  - $t_c = \max_i\{t_i\}$
- Also called “Limpware”:
  - Haryadi Gunawi, University of Chicago
  - http://www.anl.gov/events/lights-case-limping-hardware-tolerant-systems

- Ideal:
  equal work +
  equal data access =>
  equal execution time.
- Reality:
  - Lots of variation.
  - Variations increasing.

rBSP: Reducing synchronization costs "Underlapping" Domain Decomposition

Sandia National Laboratories

Ichitaro Yamazaki, Sivasankaran Rajamanickam, Erik G. Boman, Mark Hoemmen, Michael A. Heroux, and Stanimire Tomov. 2014. Domain decomposition preconditioners for communication-avoiding krylov methods on a hybrid CPU/GPU cluster. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (SC '14). IEEE Press, Piscataway, NJ, USA, 933-944. DOI=10.1109/SC.2014.81 http://dx.doi.org/10.1109/SC.2014.81



(a) Adjacency Graph of Local Submatrix. (b) Local Submatrix

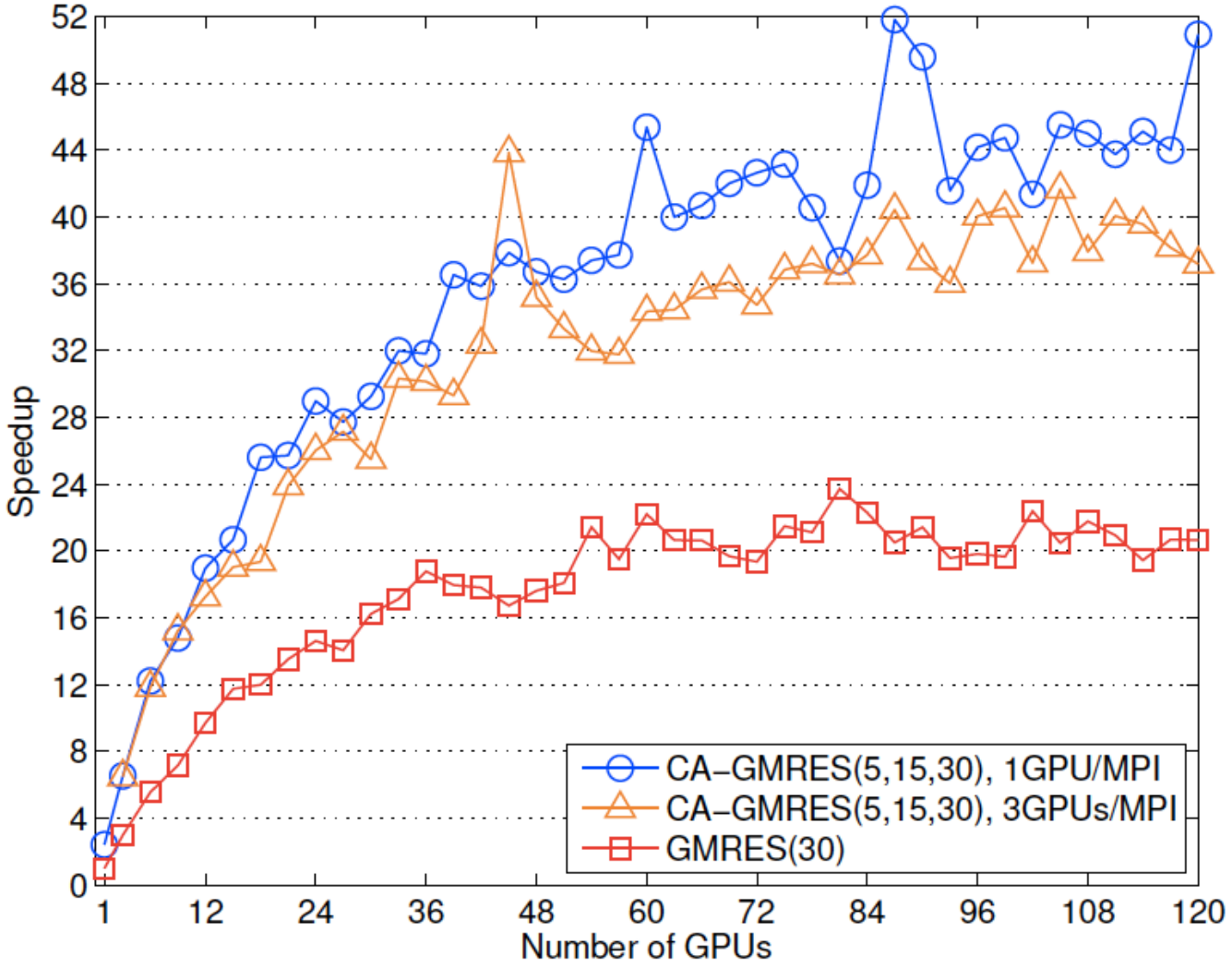
What is Needed to Support Latency Tolerance?

Sandia National Laboratories

- MPI 3 (SPMD):
  - Asynchronous global and neighborhood collectives.
- A “relaxed” BSP programming model:
  - Start a collective operation (global or neighborhood).
  - Do “something useful”.
  - Complete the collective.
- The pieces are coming online.
- With new algorithms we can recover some scalability

Speed up by reducing latency sensitivity

Sandia National Laboratories



Skeptical Programming

I might not have a reliable digital machine

Sandia National Laboratories

- Expect rare faulty computations
- Use analysis to derive cheap "detectors" to filter large errors
- Use numerical methods that can absorb *bounded error*

Algorithm 1: GMRES algorithm

```
for l = 1 to do
    r := b − Ax^(j−1)
    q_1 := r/||r||_2
    for j = 1 to restart do
        w_0 := Aq_j
        for i = 1 to j do
            h_{i,j} := q_i · w_{i−1}
            w_i := w_{i−1} − h_{i,j} q_i
        end
        h_{j+1,j} := ||w||_2
        q_{j+1} := w/h_{j+1,j}
        Find y = min ||H_j y − ||b|| e_1||_2
        Evaluate convergence criteria
        Optionally, compute x_j = Q_j y
    end
end
```

GMRES

Theoretical Bounds on the Arnoldi Process

$$\|\mathbf{w}_0\| = \|\mathbf{A}\mathbf{q}_j\| \leq \|\mathbf{A}\|_2 \|\mathbf{q}_j\|_2$$
$$\|\mathbf{w}_0\| \leq \|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F$$

From isometry of orthogonal projections,

$$|h_{i,j}| \leq \|\mathbf{A}\|_F$$

- $h_{i,j}$ form Hessenberg Matrix
- Bound only computed once, valid for entire solve

Evaluating the Impact of SDC in Numerical Methods
J. Elliott, M. Hoemmen, F. Mueller, SC'13

What is Needed for Skeptical Programming?

Sandia National Laboratories

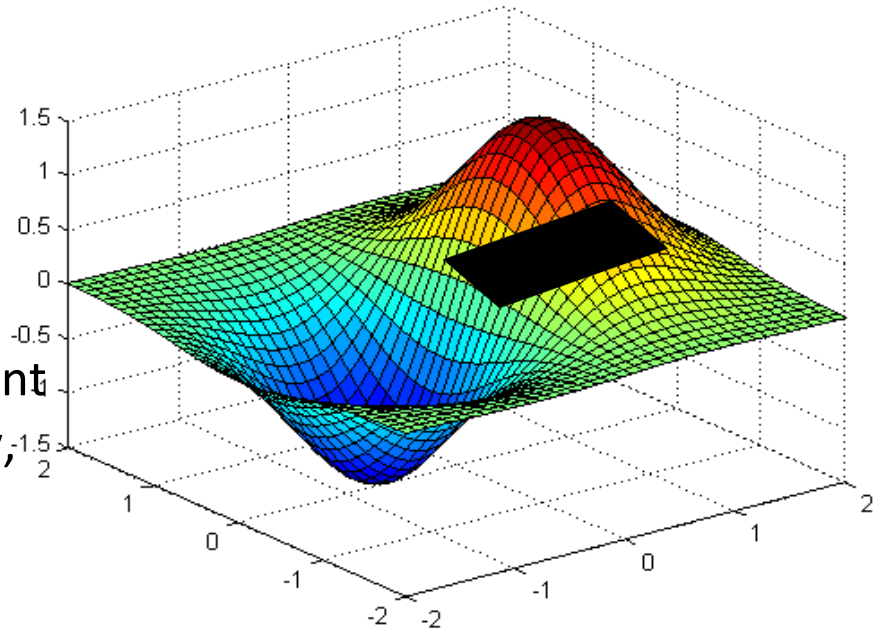
- Skepticism.
- Meta-knowledge:
  - Algorithms,
  - Mathematics,
  - Problem domain.
- Nothing else, at least to get started.

- FEM ideas:
  - Invariant subspaces.
  - Conservation principles.
  - More generally: pre-conditions, post-conditions, invariants.

Enabling Local Recovery from Local Faults

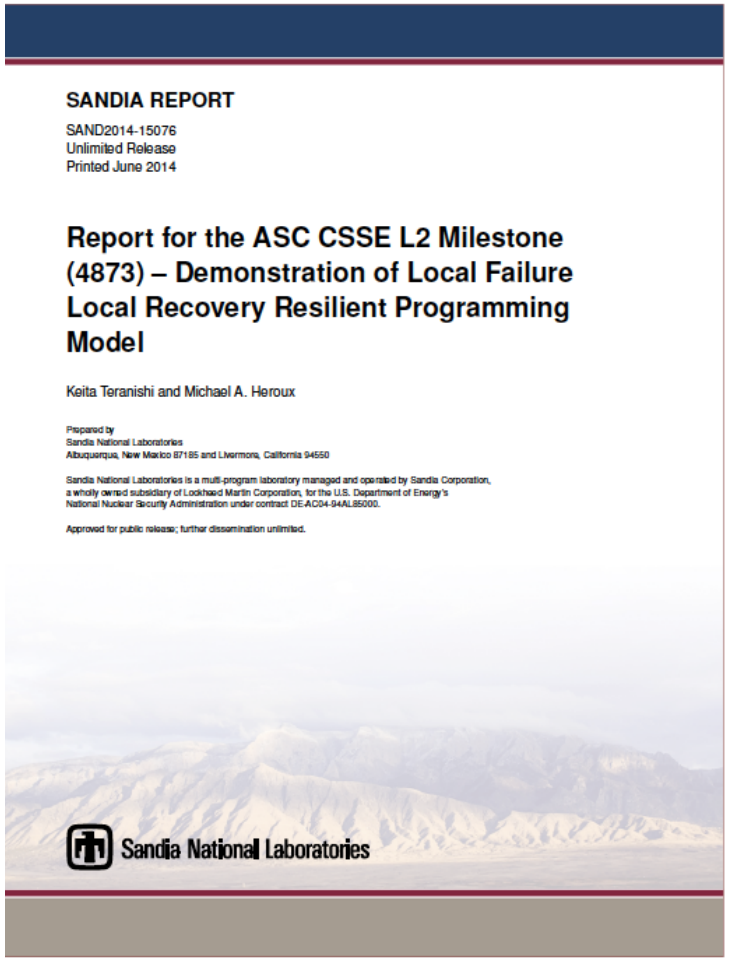
Sandia National Laboratories

- Current recovery model:
  Local node failure,
  global kill/restart.
- Different approach:
  - App stores key recovery data in persistent local (per MPI rank) storage (e.g., buddy, NVRAM),
    and registers recovery function.
  - Upon rank failure:
    - MPI brings in reserve HW, assigns to failed rank, calls recovery fn.
    - App restores failed process state via its persistent data (& neighbors'?).
    - All processes continue.



Motivation for LFLR:

- Current practice of Checkpoint/ Restart is global response to single node (local) failure
  - Kill all processes (global terminate), then restart
  - Dependent on Global File system
  - SCR (LLNL) is fast, but adheres global recovery
- Single node failures are predominant
  - 85% on LLNL clusters (Moody et al. 2010)
  - 60-90% on Jaguar/Titan (ORNL)
- Need for scalable, portable and application agnostic solution
  - Local Failure Local Recovery Model (LFLR)



SANDIA REPORT

SAND2014-15076
Unlimited Release
Printed June 2014

Report for the ASC CSSE L2 Milestone (4873) – Demonstration of Local Failure Local Recovery Resilient Programming Model

Keita Teranishi and Michael A. Heroux

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

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.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Every calculation matters

Description	Iters	FLOPS	Recursive Residual Error	Solution Error
All Correct Calcs	35	343M	4.6e-15	1.0e-6
Iter=2, y[1] += 1.0 SpMV incorrect Ortho subspace	35	343M	6.7e-15	3.7e+3
Q[1][1] += 1.0 Non-ortho subspace	N/C	N/A	7.7e-02	5.9e+5

- Small PDE Problem: ILUT/GMRES
- Correct result:35 Iters, 343M FLOPS
- 2 examples of a single bad op.
- Solvers:
  - 50-90% of total app operations.
  - Soft errors most likely in solver.
- Need new algorithms for soft errors:
  - Well-conditioned wrt errors.
  - Decay proportional to number of errors.
  - Minimal impact when no errors.

Soft Error Resilience

- New Programming Model Elements:
  - SW-enabled, highly reliable:
    - Data storage, paths.
    - Compute regions.
- Idea: *New algorithms with minimal usage of high reliability.*
- First new algorithm: FT-GMRES.
  - Resilient to soft errors.
  - Outer solve: Highly Reliable
  - Inner solve: "bulk" reliability.
- General approach applies to many algorithms.

Fault-tolerant linear solvers via selective reliability,
Patrick G. Bridges, Kurt B. Ferreira, Michael A. Heroux, Mark Hoemmen
arXiv:1206.1390v1 [math.NA]

FT-GMRES Algorithm

Input: Linear system $Ax = b$ and initial guess x_0
  $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, $q_1 := r_0/\beta$
  for $j = 1, 2, \ldots$ until convergence **do**
    Inner solve: Solve for z_j in $q_j = Az_j$
    $v_{j+1} := Az_j$
    for $i = 1, 2, \ldots, k$ **do** ▷ Orthogonalize v_{j+1}
      $H(i,j) := q_i^* v_{j+1}$, $v_{j+1} := v_{j+1} - q_i H(i,j)$
    end for
    $H(j+1,j) := \|v_{j+1}\|_2$
    Update rank-revealing decomposition of $H(1:j, 1:j)$
    if $H(j+1,j)$ is less than some tolerance **then**
      if $H(1:j, 1:j)$ not full rank **then**
        Try recovery strategies discussed in pa
      else
        Converged; return after end of this iteration
      end if
    else
      $q_{j+1} := v_{j+1}/H(j+1,j)$
    end if
    $y_j := \text{argmin}_y \|H(1:j+1, 1:j)y - \beta e_1\|_2$ ▷ GMRES projected problem
    $x_j := x_0 + [z_1, z_2, \ldots, z_j]y_j$ ▷ Solve for approximate solution
  end for

"Unreliably" computed.
Standard solver library call.
Majority of computational cost.

Captures true linear operator issues, AND
Can use some "garbage" soft error results.

What is Needed for Selective Reliability?

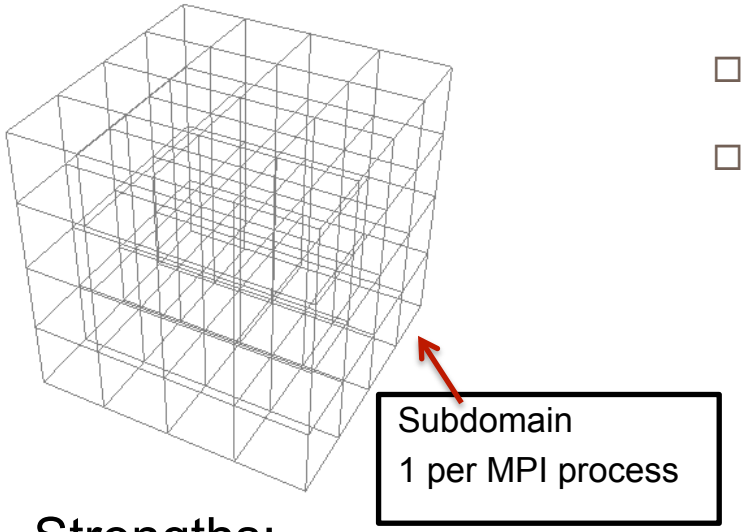
Sandia National Laboratories

- A lot, lot.
- A programming model.
  - Expressing data/code reliability or unreliability.
- Algorithms.
  - Basic approaches:
    - Nest an unreliable algorithm in a reliable version of the same.
    - Dispatch unreliable task subgraph from reliable graph node.
- Lots of runtime/OS infrastructure.
  - Provision of reliable data, paths, execution.
  - Portable interfaces to HW solutions.
- Hardware support?
  - Special HW components that are
    - slower and more reliable or
    - faster and less reliable

Sandia National Laboratories

TASK-CENTRIC/DATAFLOW DESIGN AND RESILIENCE

Classic HPC Application Architecture

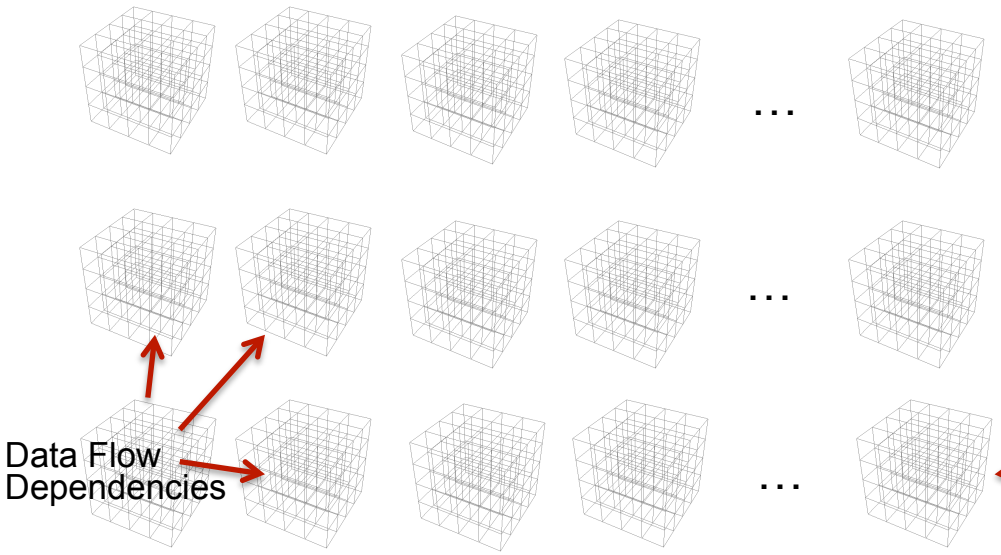


- Logically Bulk-Synchronous, SPMD
- Basic Attributes:
  - Halo exchange.
  - Local compute.
  - Global collective.
  - Halo exchange.

- Strengths:
  - Portable to many specific system architectures.
  - Separation of parallel model (SPMD) from implementation (e.g., message passing).
  - Domain scientists write sequential code within a parallel SPMD framework.
  - Supports traditional languages (Fortran, C).
  - Many more, well known.

- Weaknesses:
  - Not well suited (as-is) to emerging manycore systems.
  - Unable to exploit functional on-chip parallelism.
  - Difficult to tolerate dynamic latencies.
  - Difficult to support task/compute heterogeneity.

Task-centric/Dataflow Application Architecture



- Patch: Logically connected portion of global data. Ex: subdomain, subgraph.
- Task: Functionality defined on a patch.
- Many tasks on many patches.

- Strengths:
  - Portable to many specific system architectures.
  - Separation of parallel model from implementation.
  - Domain scientists write sequential code within a parallel framework.
  - Supports traditional languages (Fortran, C).
  - Similar to SPMD in many ways.

- More strengths:
  - Well suited to emerging manycore systems.
  - Can exploit functional on-chip parallelism.
  - Can tolerate dynamic latencies.
  - Can support task/compute heterogeneity.
  - **Resilience can be applied at task level.**

Resilience & Task-centric/Dataflow

Sandia National Laboratories

- Relaxed Bulk Synchronous (rBSP)
  - Async tasking: Addresses same issues.
  - "Porous barriers":
    - Tasks contribute portion to global collective, move on.
    - Come back later to collect global result.
- Skeptical Programming. (SP)
  - Skepticism applied at task level.
  - Parent task can apply cheap validation test up child's return.
- Local-Failure, Local-Recovery (LFLR)
  - Applied at task level.
  - SSD storage available for task-level persistent store.
- Selective (Un)reliability (SU/R)
  - Parent task (at some level in the task graph) executes reliably.
  - Children are fast, unreliable.
  - Parent corrects or regenerates child task if it times out or SDC detected.

Summary

Sandia National Laboratories

- Resilience will be an issue, really it will.
- Already is: Performance variability is the result.
- Latency tolerant algorithms are a key.
- LFLR approaches are next step.
- Task-centric/dataflow approaches & resilience: synergistic.
- Big concern:
  - Trends in system design: Fewer, more powerful nodes.
  - If node loss is common: Recovery is expensive, hard to do.