Modeling and Simulation of Extreme-Scale Systems for Resilience by Design

Christian Engelmann (PI), and Rizwan Ashraf – Oak Ridge National Laboratory

U.S. DEPARTMENT OF ENERGY | Office of Science | OAK RIDGE National Laboratory

ACKNOWLEDGEMENTS

Work supported by the Early Career Program of the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research with program manager Lucy Nowell

Motivation

The Paradox of Choice:

Many possible solutions for resilience in extreme-scale high-performance computing systems [hardware, system software, algorithm-based, programming model-based, etc.]

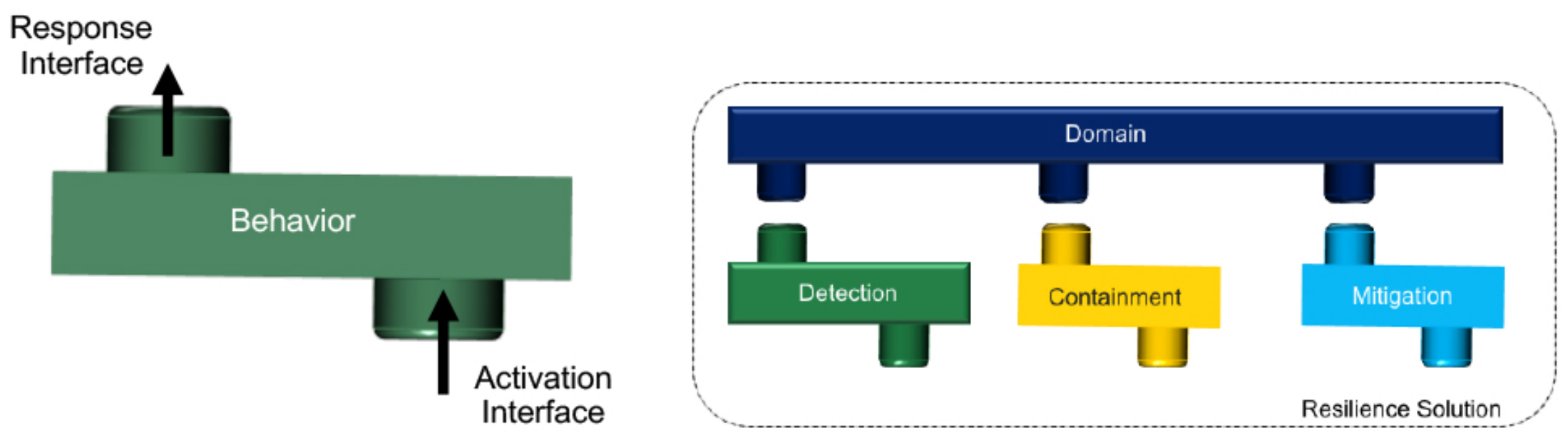
- Incomplete understanding of protection coverage against high probability & high impact vs. less likely & less harmful faults
- No good evaluation methods & metrics that consider
  - Fault impact scope, handling coverage and handling efficiency
  - Performance, resilience and power trade-offs
- No mechanisms and interfaces for coordination for avoidance of costly overprotection
- No resilience portability across architectures and software environments

Design Patterns for Resilience

- A design pattern provides a generalizable solution to a recurring problem
- It formalizes a solution with an interface and a behavior specification
- Design patterns do not provide concrete solutions
- They capture the essential elements of solutions, permitting reuse and different implementations
- *State patterns* provide encapsulation of system state for resilience:
  - *Persistent State, Dynamic State, Environment State* and *Stateless* patterns
- Behavioral patterns provide encapsulation of detection, containment and mitigation techniques for resilience:
  - *Strategy, Architecture,* and *Structural patterns*

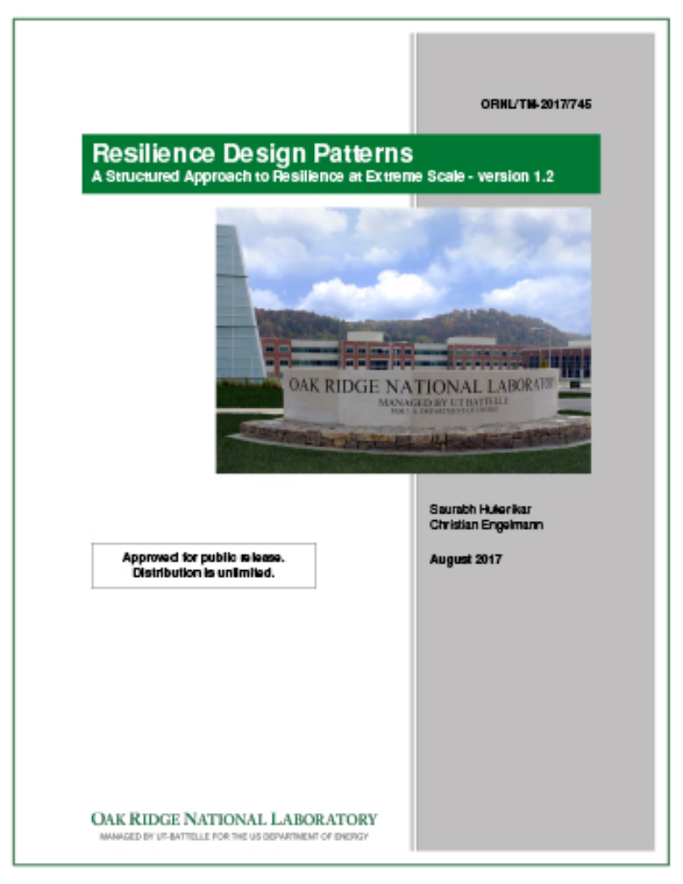
Anatomy of a Resilience Design Pattern

- A resilience design pattern is defined in an event-driven paradigm
- Instantiation of pattern behaviors may cover combinations of *detection*, *containment* and *mitigation capabilities*
- Enables writing patterns in consistent format to allow readers to quickly understand context and solution



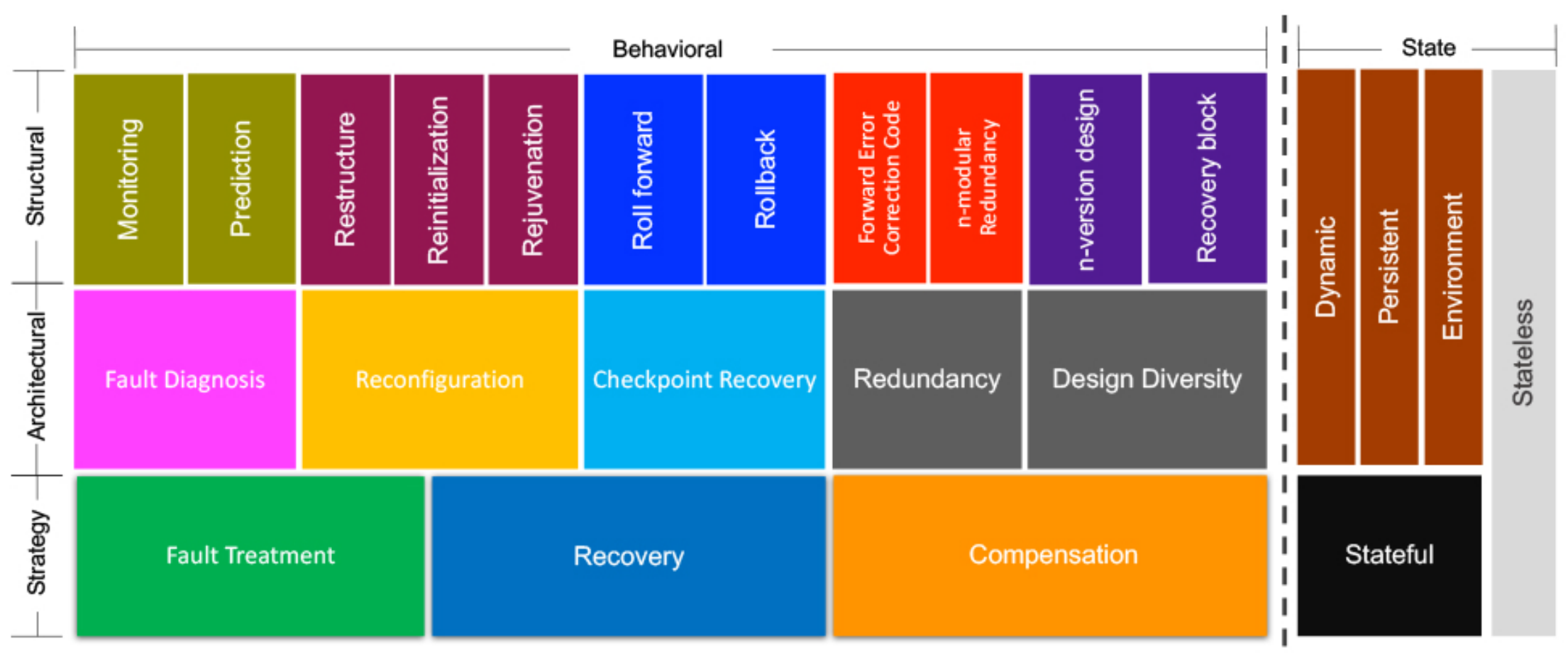
Resilience Design Patterns Specification v1.2

- Taxonomy of resilience terms and metrics
- Survey of resilience techniques
- Classification of resilience design patterns
- Catalog of resilience design patterns
  - Uses a pattern language to describe solutions
  - 3 strategy patterns, 5 architectural patterns, 11 structural patterns, and 5 state patterns
- Case studies using the design patterns
- A resilience design spaces framework

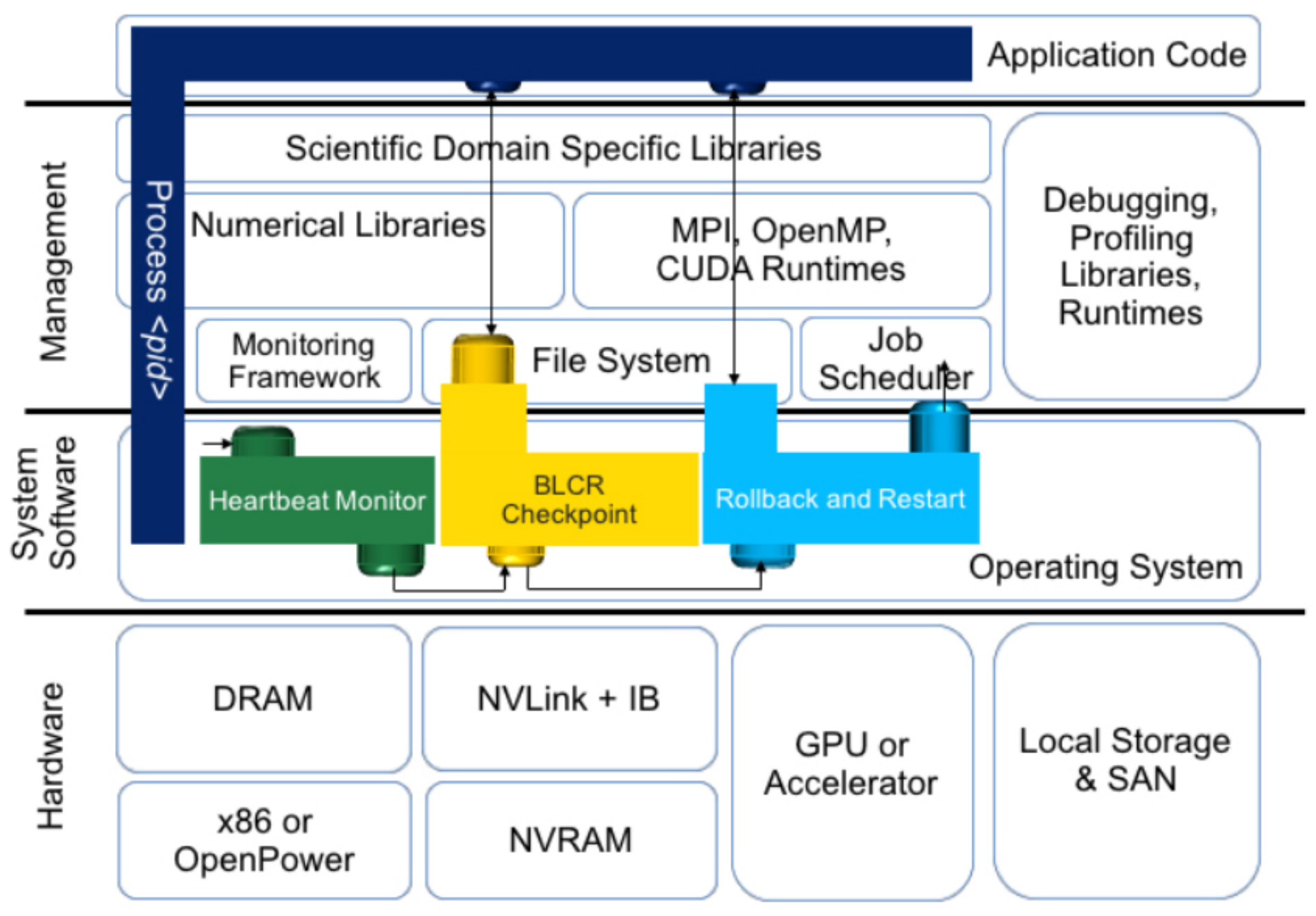

Resilience Design Patterns
A Structured Approach to Resilience at Extreme Scale - version 1.2
ORNL/TM-2017/745
Approved for public release. Distribution is unlimited.
Saurabh Hukerikar
Christian Engelmann
August 2017
OAK RIDGE NATIONAL LABORATORY

Saurabh Hukerikar and Christian Engelmann. **Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale (Version 1.2)**. Technical Report, ORNL/TM-2017/745, Oak Ridge National Laboratory, Oak Ridge, TN, USA, August, 2017. DOI: 10.2172/1436045

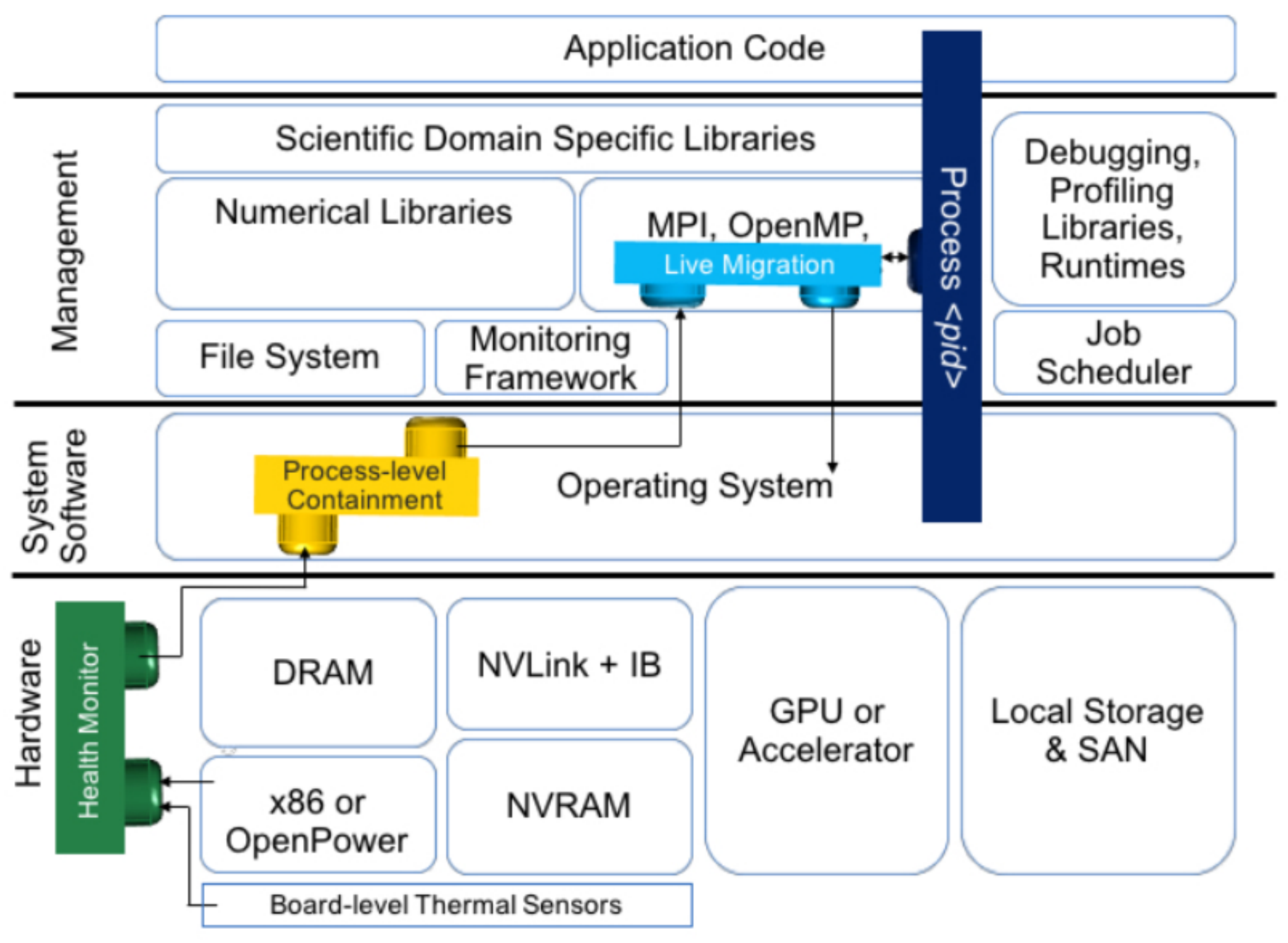
Resilience Design Patterns Classification



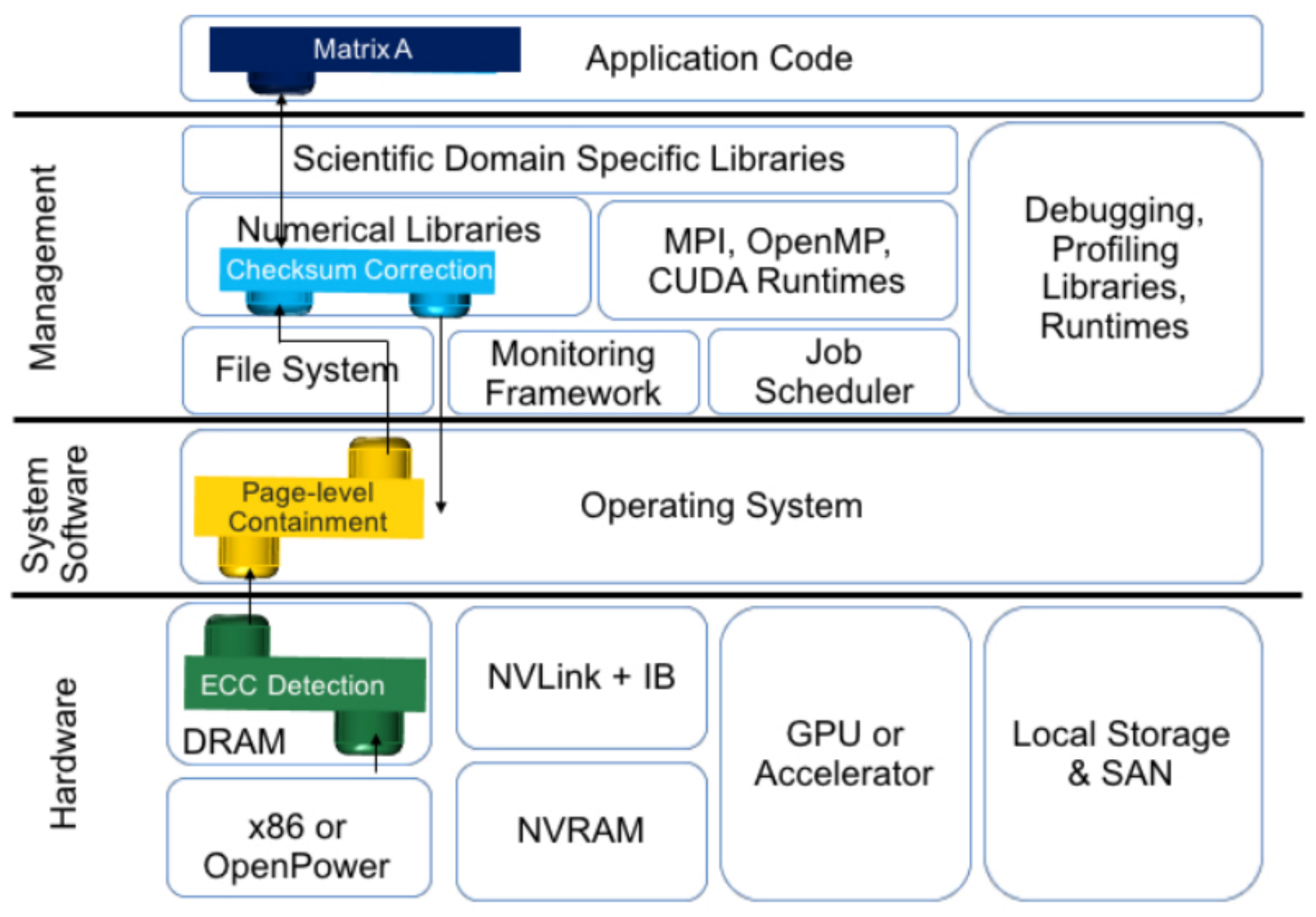
Case Study: Checkpoint Recovery with Rollback



Case Study: Proactive Process Migration

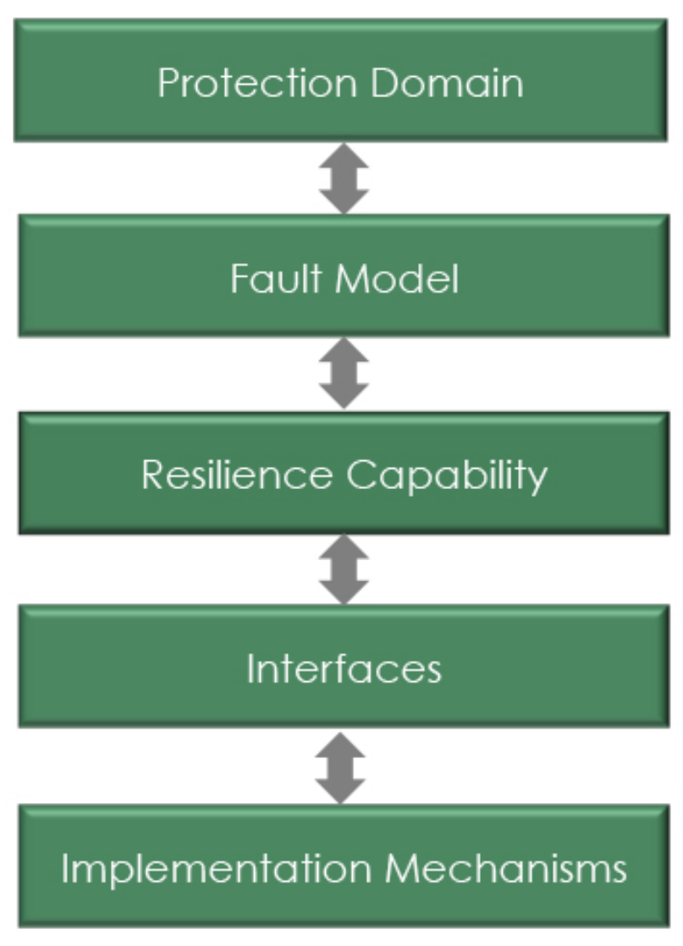


Case Study: Cross-Layer Hardware/Software Hybrid Solution



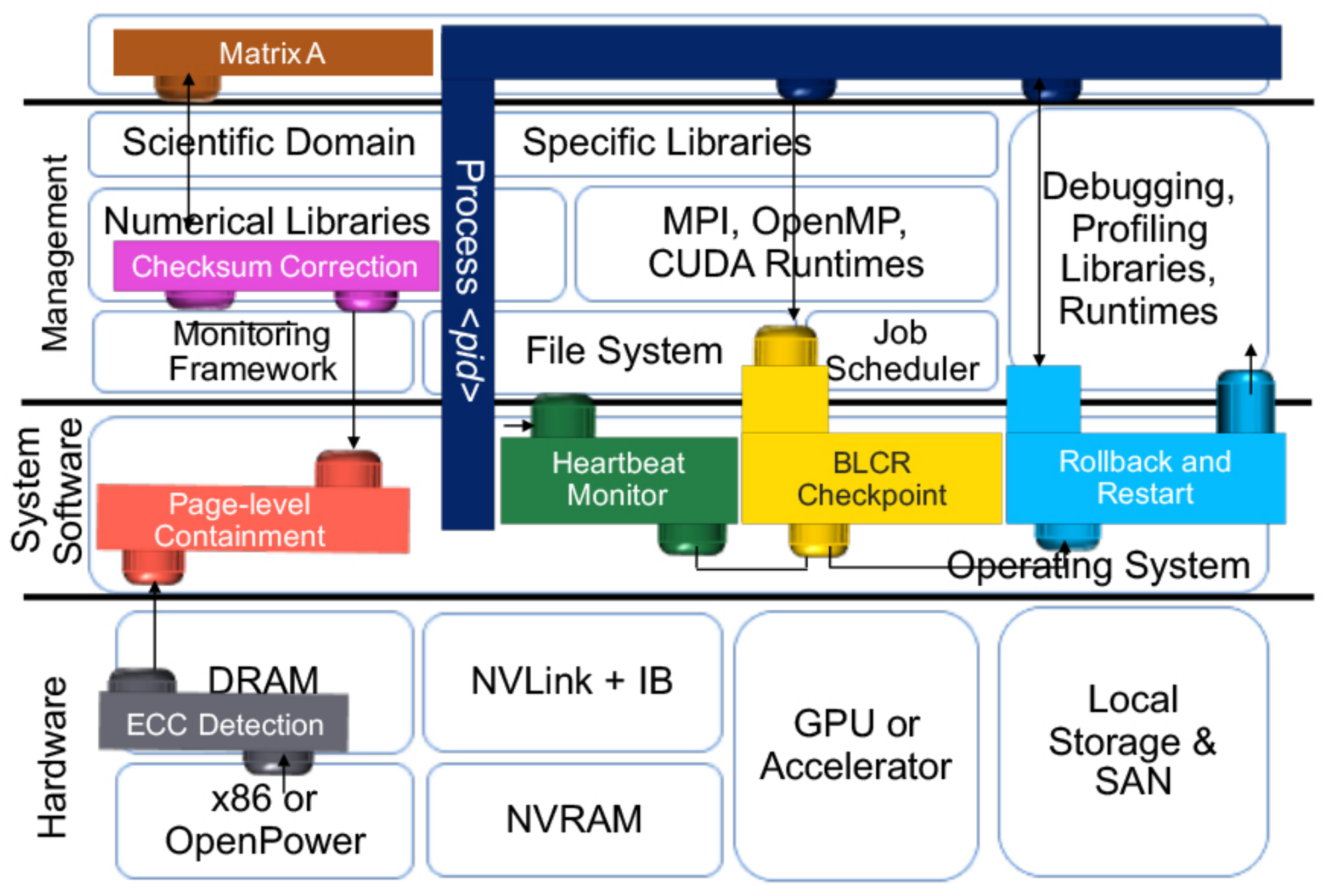
Resilience Design Spaces Framework

- Design for resilience can be viewed as a series of refinements
- The design process is defined by 5 design spaces
- Navigating each design space progressively adds more detail to the overall design of the resilience solution
- A single solution may solve more than one resilience problem
- Multiple solutions often solve different resilience problems more efficiently



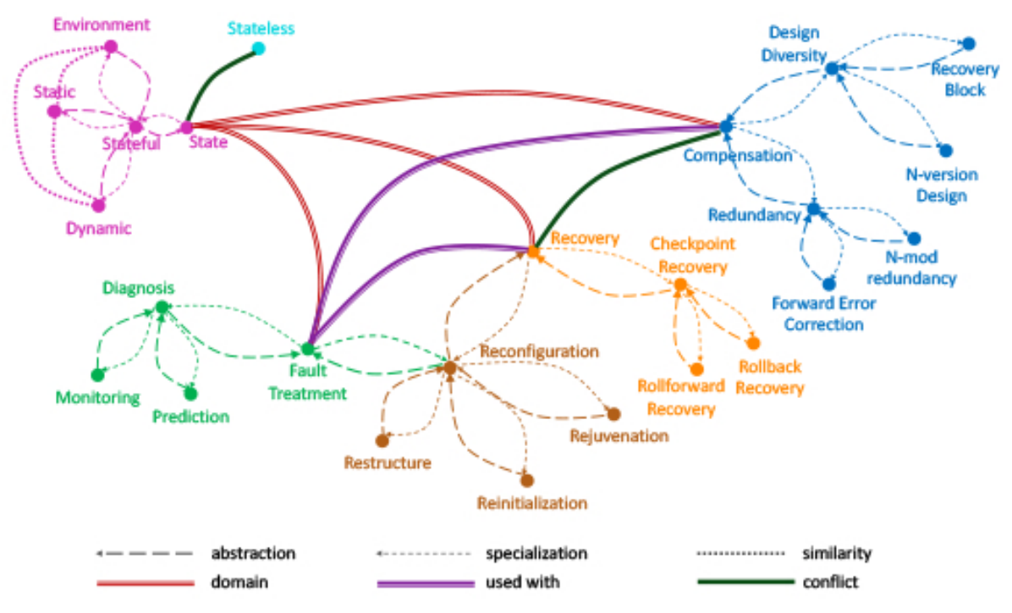
Design Space Exploration for Resilience

- Vertical and horizontal pattern compositions describe the resilience capabilities of a system
- Pattern coordination leverages beneficial and avoids counterproductive interactions
- Pattern composition optimizes the performance, resilience and power consumption trade-off



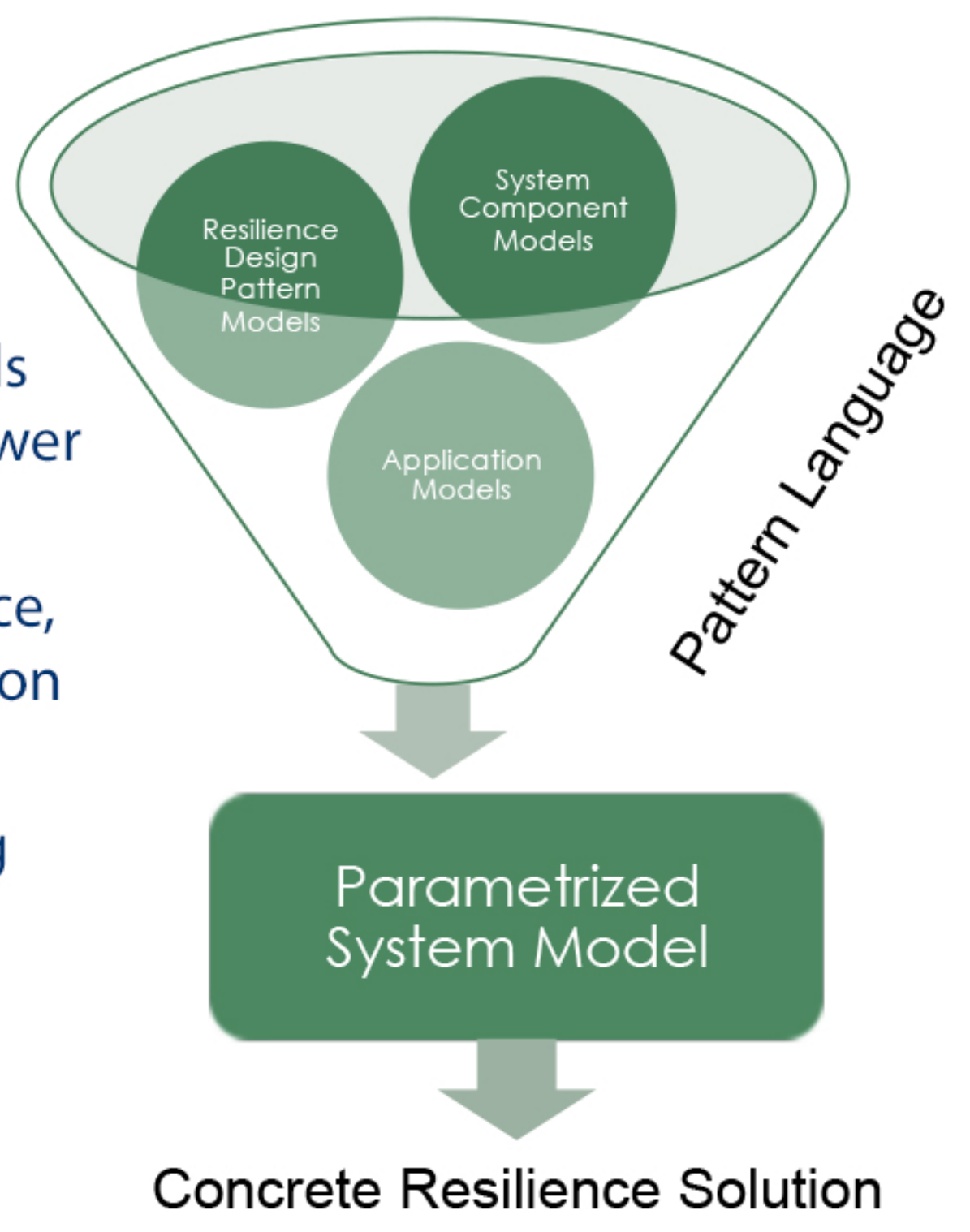
Resilience Design Pattern Language

- Identifies relationships between patterns
  - Abstraction vs. specialization
  - Used with vs. conflict
  - Similarity
  - Domain
- Uses graph representation
- Enables structured analysis



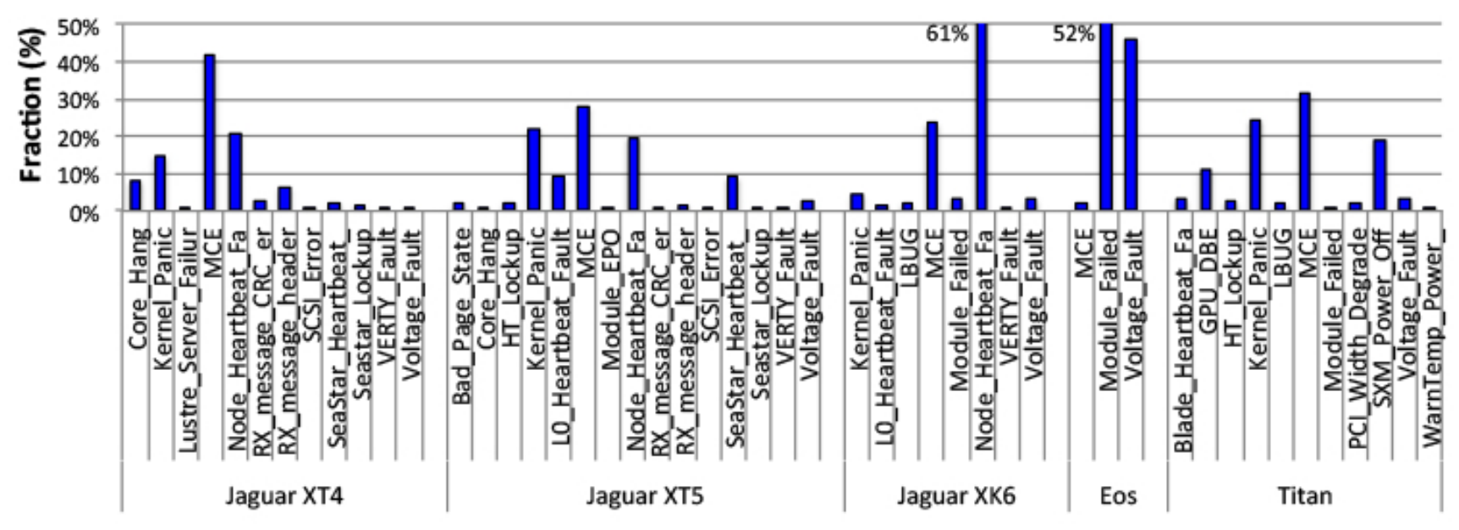
Systematic Modeling & Design of Resilience Solutions

- Abstract the system with models:
  - System component models (performance, resilience and power consumption models)
  - Resilience design pattern models (performance, resilience and power consumption models)
  - Application models (performance, resilience and power consumption models)
- Evaluate solutions using modeling and simulation
- Discover suitability of pattern combinations for system-specific resilience problems
- Predict behavior on different hardware architectures and in different software environments

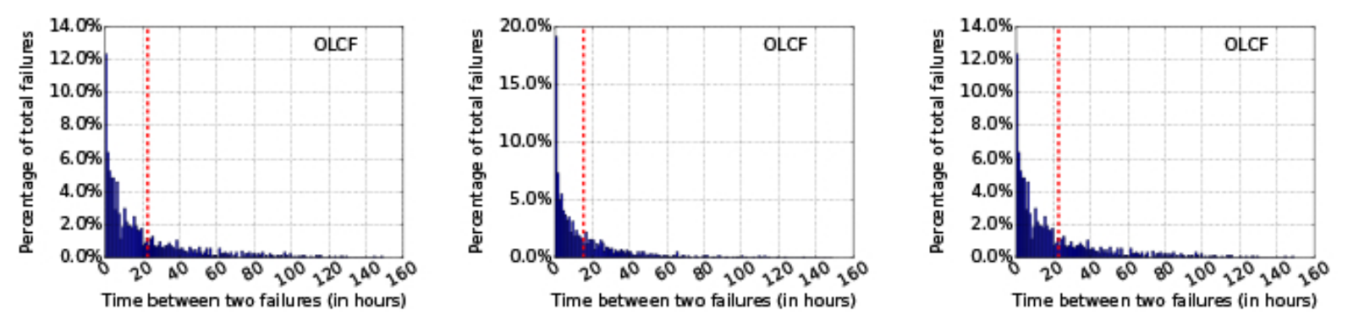

Resilience Design Pattern Models
System Component Models
Application Models
Pattern Language
Parametrized System Model
Concrete Resilience Solution

System Component Models

- Already extensive work by ORNL, ANL and LLNL in analyzing DOE systems
- The *Catalog* project identifies, categorizes and models the fault, error and failure properties of DOE systems
  - Fault, error and failure types
  - Probability distributions
  - Temporal and spatial locality and correlation
  - Propagation paths and detection latency



Fraction of each failure type on 5 ORNL systems



Failure inter-arrival time for 3 ORNL systems (MTBF as red vertical line)

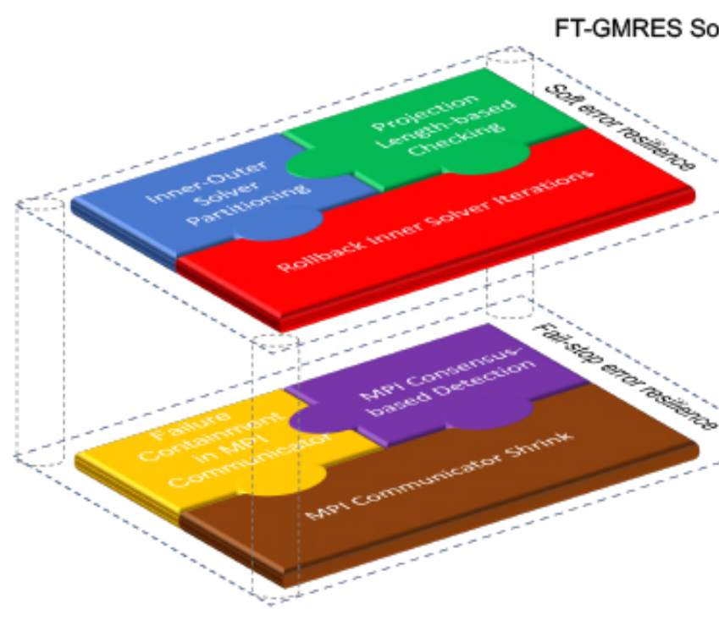
Resilience Design Pattern Models

- Preliminary mathematical reliability and performance models for each pattern
  - Take into account detection latency and performance loss due to repair and/or system degradation
- Ongoing work in outcome-based metrics considers value and performance efficiency
  - Correctness and time to solution
- Preliminary power consumption models are still work in progress

Pattern	Performance Model
Fault Diagnosis	$T_{system} = T_0 + \sum_{k=1}^{n} t_{inference}/\eta$
Reconfiguration	$T_{system} = T_{FF} + (1 - T_{FF}).\frac{n-1}{n} + T_R$
Rollback	$T_{system} = (T_{FF} + \gamma)/\eta$ where $\mathrm{T}_{FF} = \mathrm{o} + \delta/\mathrm{r}$
Roll forward	$T_{system} = o + \delta/r$
Redundancy	$T_{system} = T_{SER}.((1 - \mathcal{A}) + \beta.\mathcal{A})) + T_{MV}$

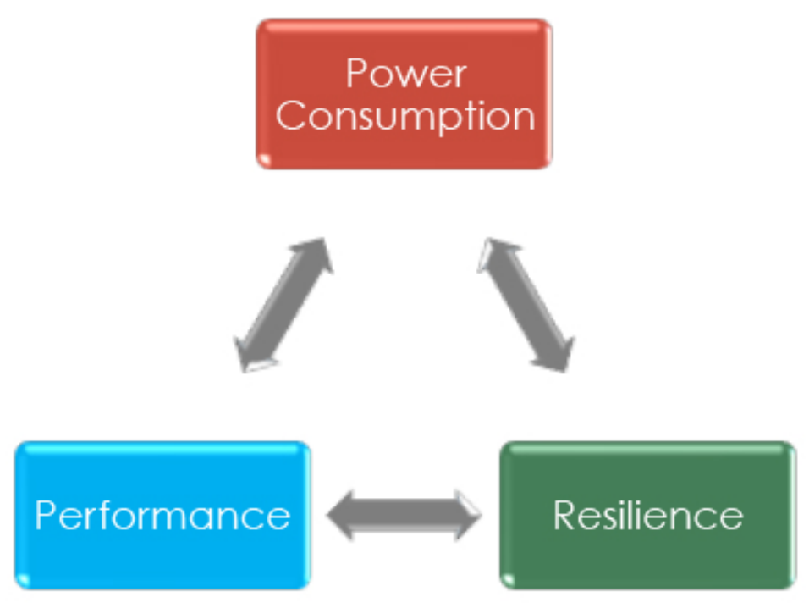
Application Models

- Significant amount of existing work in application performance models
- Some amount of existing work in application reliability models
  - Application vulnerability studies
  - Error propagation patterns
  - Resilient solvers
  - *More work is needed*
- Some amount of existing work in application power consumption models
  - *More work is needed*



Modeling and Simulation for Design Space Exploration (Future Work)

- Model the performance, resilience, and power consumption of an entire system
- Start at compute-node granularity with
  - System component models
  - Resilience design pattern models
  - Application models
- Simulate dynamic interactions between the system, resilience solutions and applications
- Move to finer-grain resolution to include on-node communication, computation and storage
- Build upon prior work with the Extreme-scale Simulator (xSim)



Web site: https://ornlwiki.atlassian.net/wiki/spaces/RDP, PI: Christian Engelmann, engelmannc@ornl.gov

ORNL 2018-G00870/DLR