Resilience by Design (and not as an Afterthought)

Christian Engelmann

Oak Ridge National Laboratory

1). What is one of the greatest resilience challenge facing future leadership-class systems?

- Nobody expects the Spanish Inquisition!
- Nobody expects ***significant*** reliability issues during operation, but ***they do happen***!
  - See Titan GPU failures as an example (SC'18 paper)
- ***We should expect significant reliability issues during operation!***
- ***We should design the HPC hardware/software ecosystem to be able to deal with high error and failure rates!***

2). What are the factors contributing to this challenge?

- Shock, Disbelief and Denial
  - Past systems were reliable, current systems are
  - DOE isn't going to buy an unreliable system
- Bargaining and Guilt
  - Vendors/manufacturers are going to solve this
- Acceptance
  - Thinks break during operation and potentially at a high rate
    - Bad solder, dirty power, unexpected early wear-out, etc.
    - Process technology uncertainties, ever-growing system scale and acquisition/operating cost constraints don't make things easier
    - Expected extreme heterogeneity adds complexity
    - Non-von Neumann architectures create new questions:
      - Correctness? Determinism? Reliability?

3). What is a proposed solution and its known cost and benefits?

- Resilience needs to be holistically provided by the HPC hardware/software ecosystem with:
  1. Wide-ranging resilience capabilities in hardware, system software, programming models, libraries, and applications
  2. Interfaces and mechanisms for coordinating resilience capabilities across diverse hardware and software components
  3. Appropriate metrics and tools for assessing performance, resilience, and energy
  4. An understanding of the performance, resilience and energy trade-off that eventually results in well-informed HPC system design choices and runtime decisions
- We don't understand costs today beyond checkpoint/restart (what we do today) and full redundancy (what nobody wants)!
  – Assumptions for those cost estimates are questionable

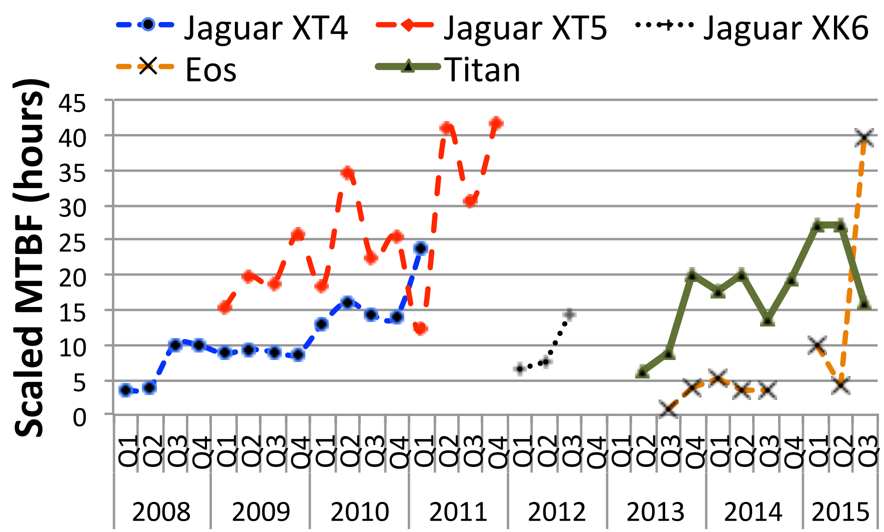
Understanding the Problem is Key

What does actually fail, why and how? Are our assumptions correct?

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Reliability of HPC systems: Large-term Measurement, Analysis, and Implications (1/3)

- Analyzed 1.2 billion node hours of logs from 5 different OLCF supercomputers
- Combined information from different logs and created a consistent log format for analysis
- Used standard and created new methods to model the temporal and spatial behavior of failures
- Analyzed the evolution of temporal and spatial behavior over the years
- Analyzed the correlation of different failure types
- Compared the mean-time between failures of the 5 systems

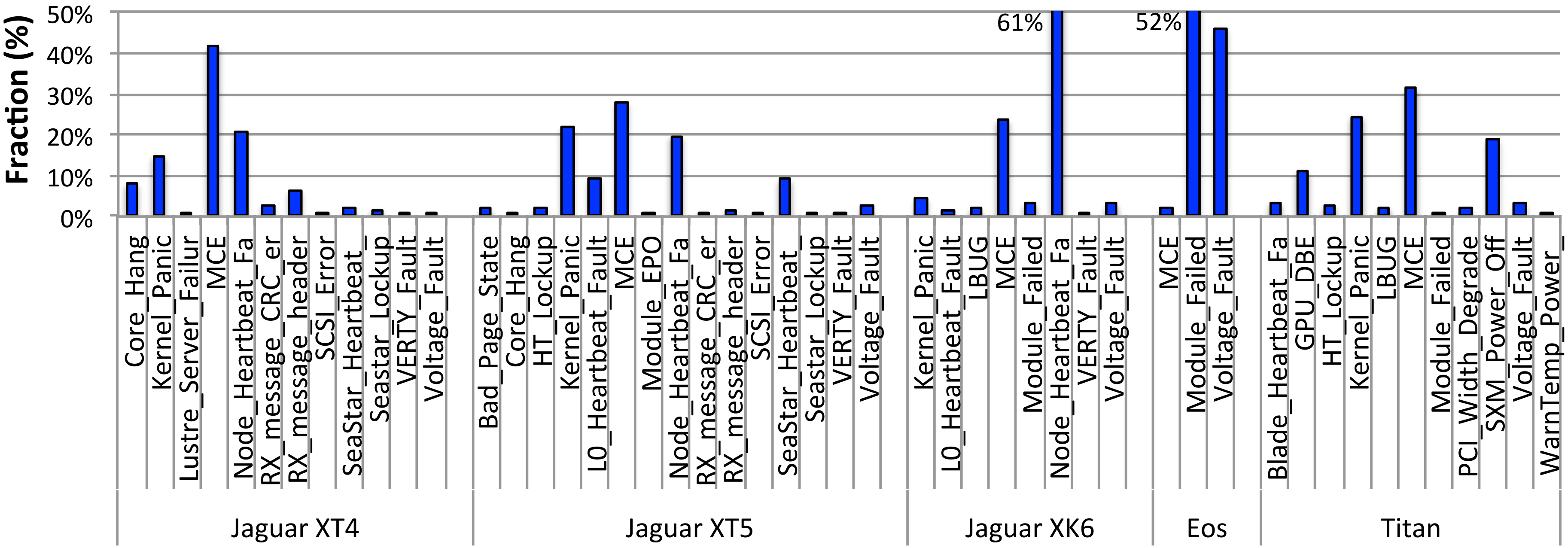


Scale-normalized MTBF of each system over time (averaged quarterly)

$$\text{Scale-Normalized MTBF} = \frac{\text{MTBF} \times \text{Num of Nodes in the System}}{\text{Max Number of Nodes across all Systems}}$$

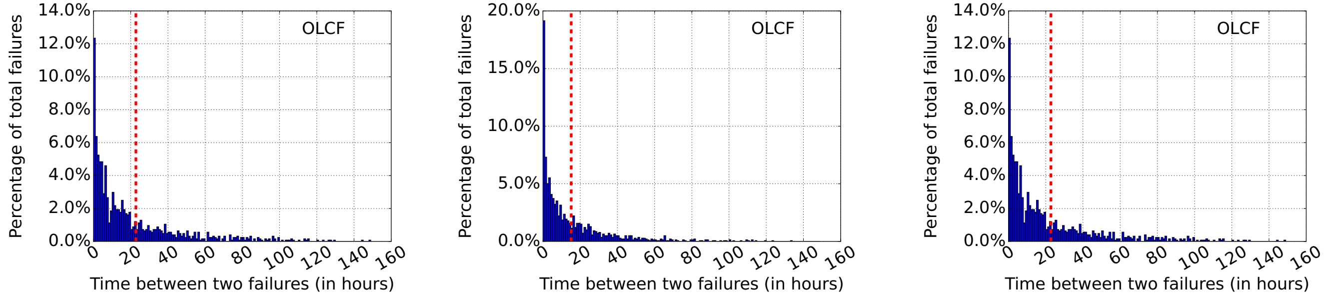
Saurabh Gupta, Devesh Tiwari, Tirthak Patel, and Christian Engelmann. **Reliability of HPC systems: Large-term Measurement, Analysis, and Implications.** SC'17.

Reliability of HPC systems: Large-term Measurement, Analysis, and Implications (2/3)



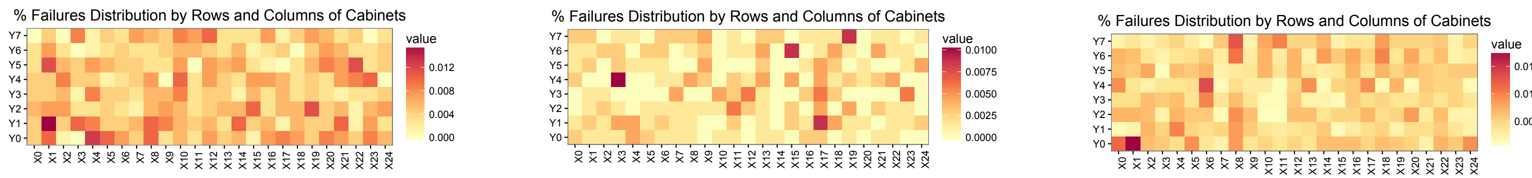
Fraction of each failure type on the studied systems

Reliability of HPC systems: Large-term Measurement, Analysis, and Implications (3/3)



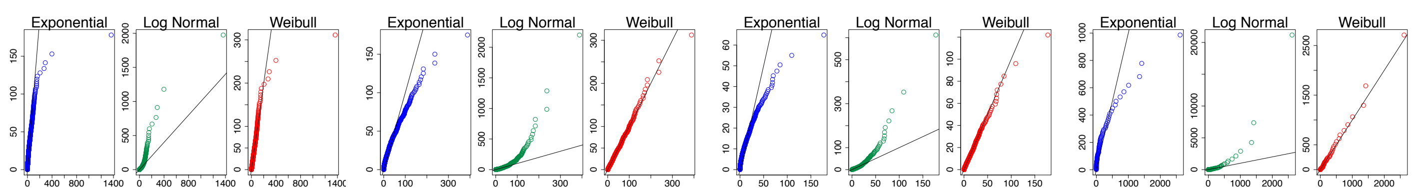
(a) Jaguar XT4 **(b) Titan** **(c) Eos**

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



(a) Jaguar XT5 **(b) Jaguar XK6** **(c) Titan**

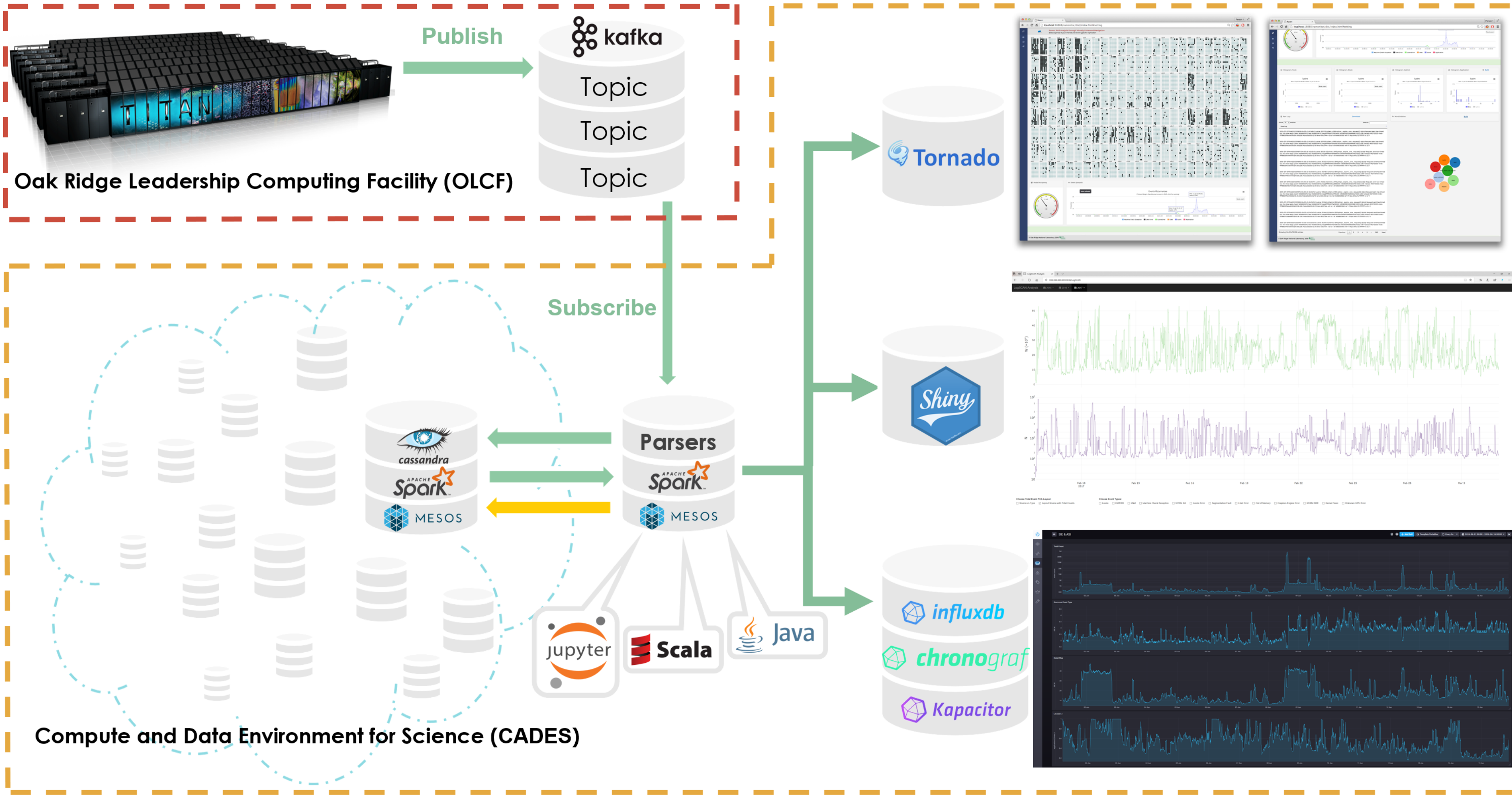
Spatial distribution of failures among cabinets for 3 studied systems



(a) Jaguar XT4 (b) Jaguar XT5 (c) Jaguar XK6 (d) Eos

QQ-plots showing goodness of fit for the failure inter-arrival times for 4 studied systems with different failure probability density functions (Weibull fits best)

LogSCAN Real-Time Processing Architecture



Y. Hui, R. Ashraf, B. H. Park, and C. Engelmann. **Real-Time Assessment of Supercomputer Status by a Comprehensive Informative Metric through Streaming Processing.** Poster at IEEE BigData'18.

New Metrics to Evaluate HPC System Health

3D

General Form of Data Table

	Feature 1	Feature 2	…	…	Feature N
Record 1					
Record 2					
…					
…					
…					
…					
Record M					

2D

Variance Distribution of Principal Components

$$\xi_i = \frac{\sigma_i}{\sum_1^k \sigma_i}$$

1D

Shannon Entropy

$$H = -\sum_1^k \xi_i log_b(\xi_i)$$

Entropy: in a general "b-ary" form

2D

Principal Components in Feature Space

$$\text{SVD} \Longrightarrow \sigma_i$$

σ_i: *i*-th variance out of k eigenvalues of the SVD decomposition

1D

System Information Entropy (SIE)

$$W(t) = b^{H(t)}$$

b: the logarithmic base used in calculating H. In our analysis, b = 10.

System Reliability Event Counts

$$\vec{A} = [a_1\ a_2 \cdots a_M]$$

a_i: total event counts for the application "i"

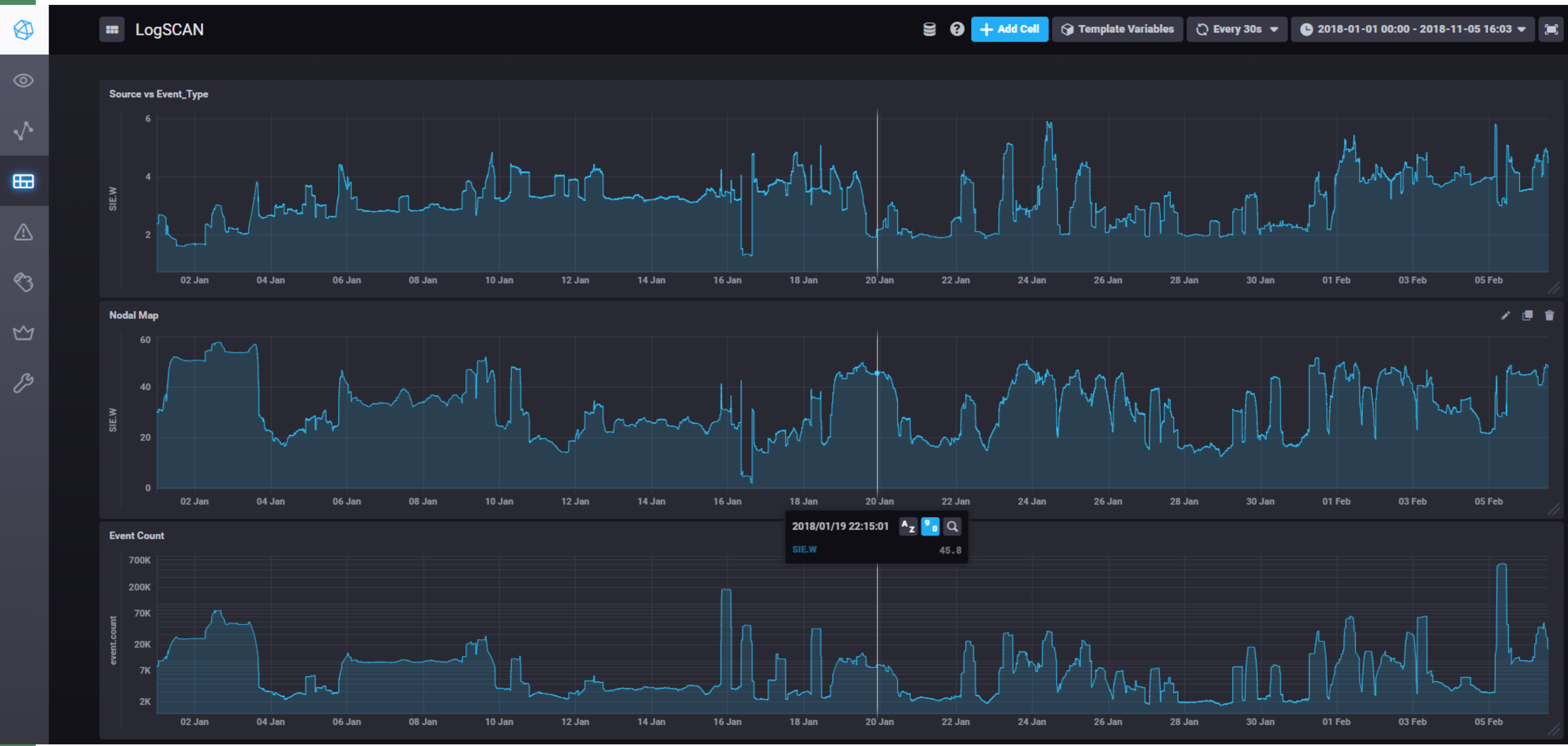
Application System Impact (ASI)

$$\text{ASI} = \frac{\|\vec{A}\|_{l_2}}{\|\vec{A}\|_{l_1}} = \frac{\sqrt{\sum_1^M a_i^2}}{\sum_1^M a_i}$$

$\| \|_{l_1}$ and $\| \|_{l_2}$ represent the L_1- and L_2-norm applied on $\vec{A}$, respectively.

The value of ASI is limited to the range (0, 1). When ASI approaches 1, it represents high sparsity or a time interval in which only a few applications are generating most of system reliability events and vice versa.

Real-Time Analysis of System Information Entropy



SIE with Source Type layout (top),
SIE with Nodal Map layout (middle), and
Total event count (bottom)

Coordinating Multiple Solutions is Key

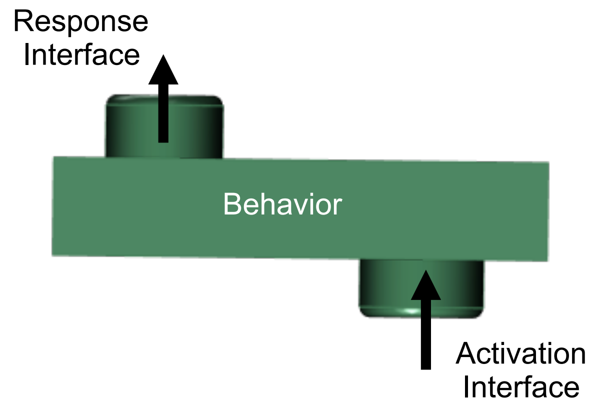
Why do we abort and restart an entire job when 1 out of 27,648 GPUs has an error? Why don't we just re-run the single failed GPU execution?

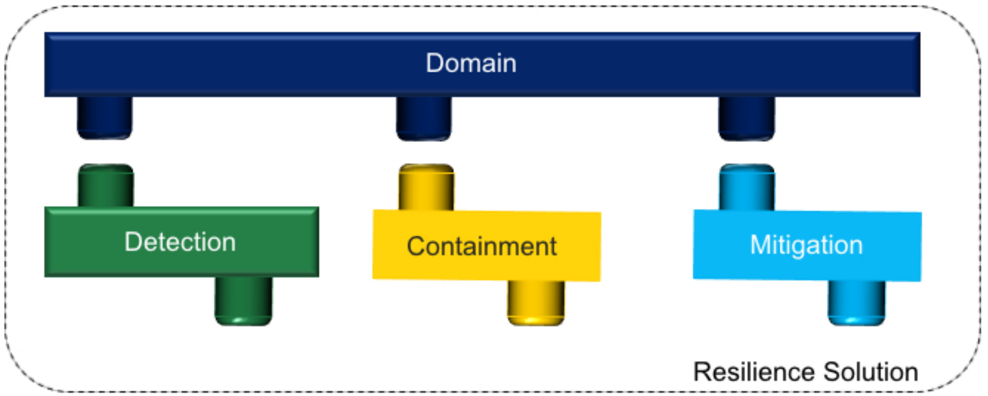
Novel Solution: Design Patterns for Resilience

- 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
- Behavioral patterns provide encapsulation of detection, containment and mitigation techniques for resilience

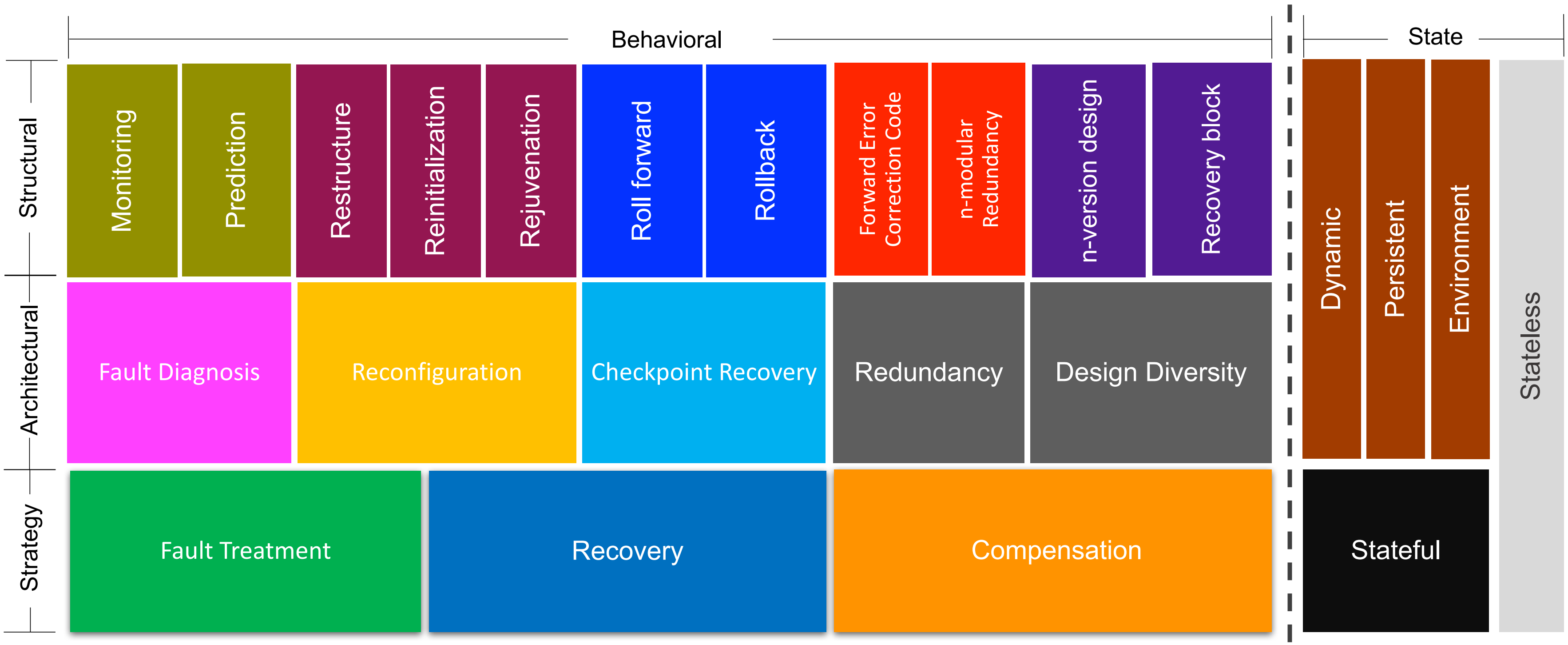
Anatomy of a Resilience Design Pattern

- 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



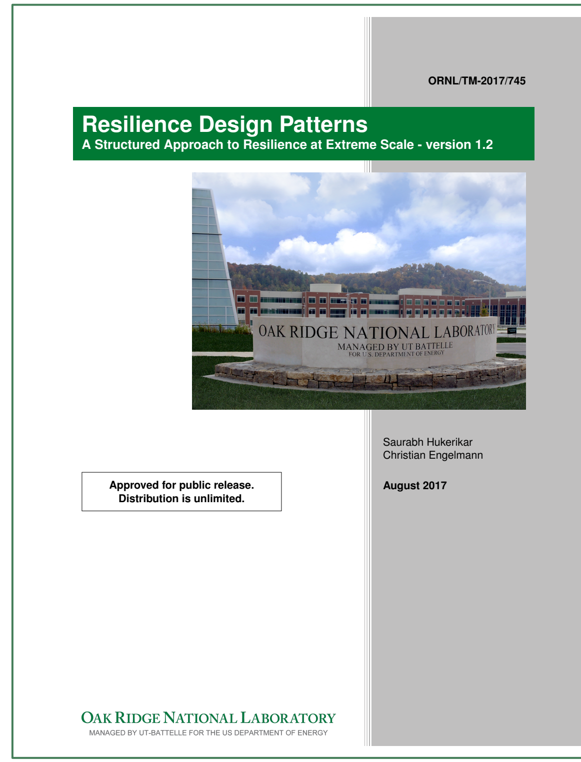


Resilience Design Patterns Classification



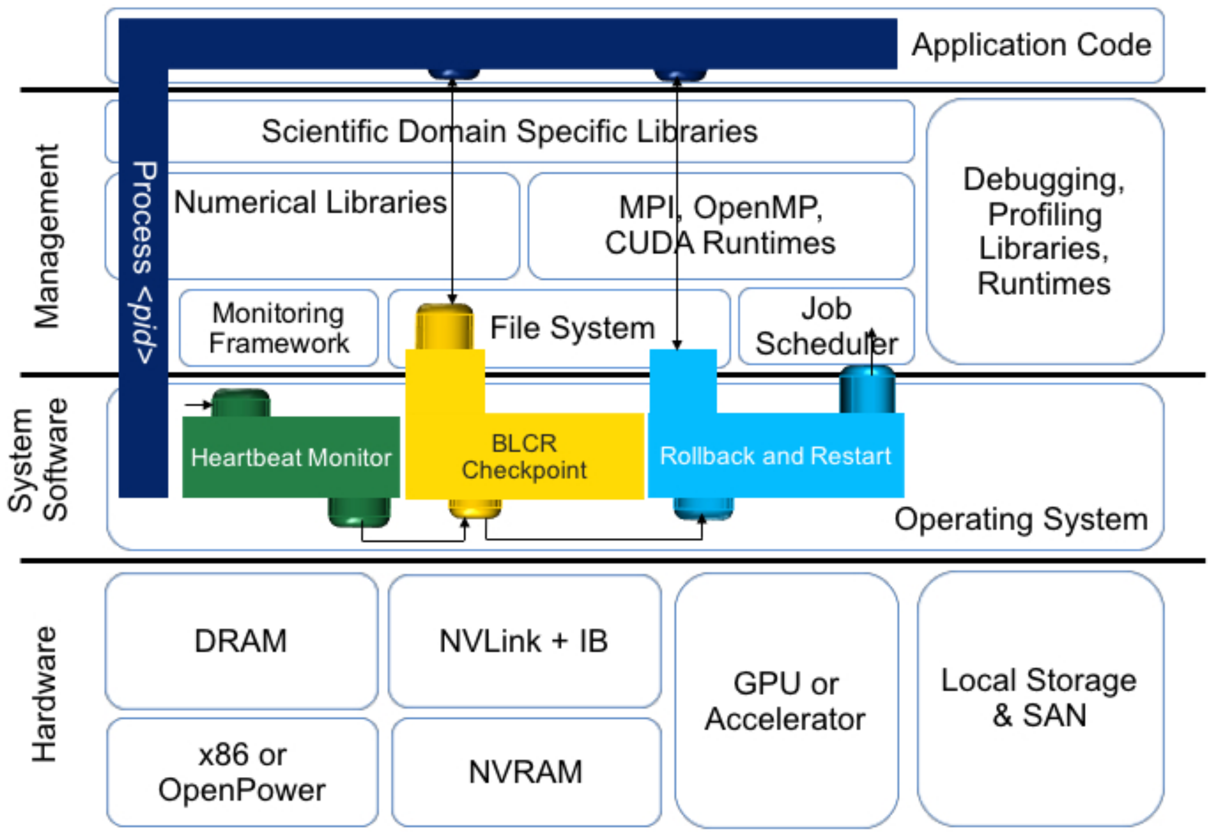
Resilience Design Patterns Specification v1.2

- 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

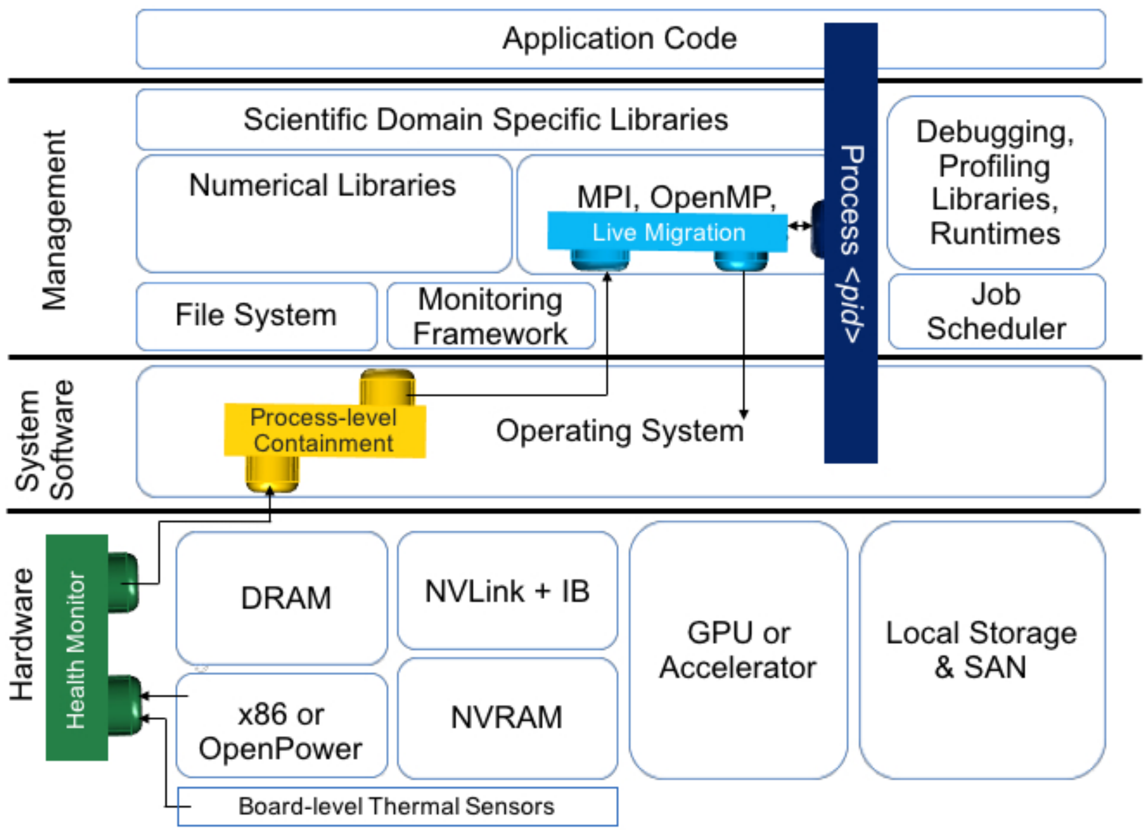


Saurabh Hukerikar and Christian Engelmann. **Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale (Version 1.2)**. Technical Report, ORNL/TM-2017/745, Oak Ridge National Laboratory, Oak Ridge, TN, USA, August, 2017. DOI: 10.2172/1436045

Case Study: Checkpoint Recovery with Rollback

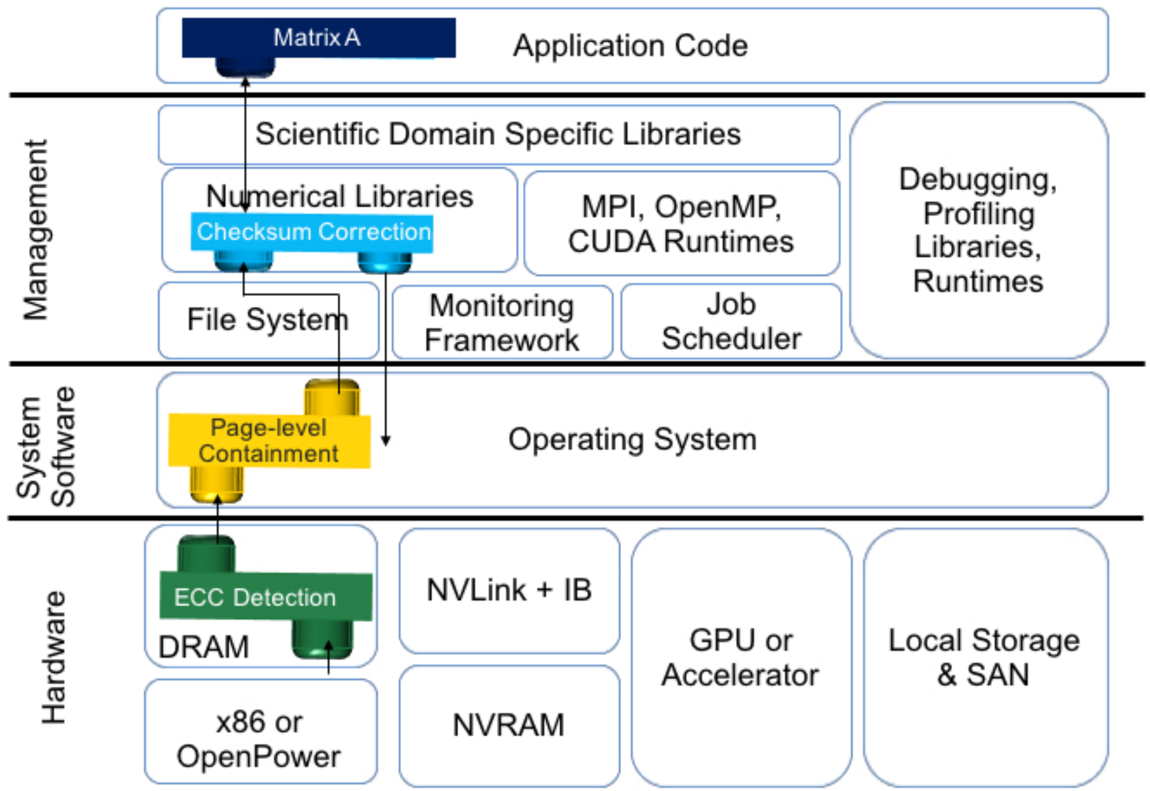


Case Study: Proactive Process Migration



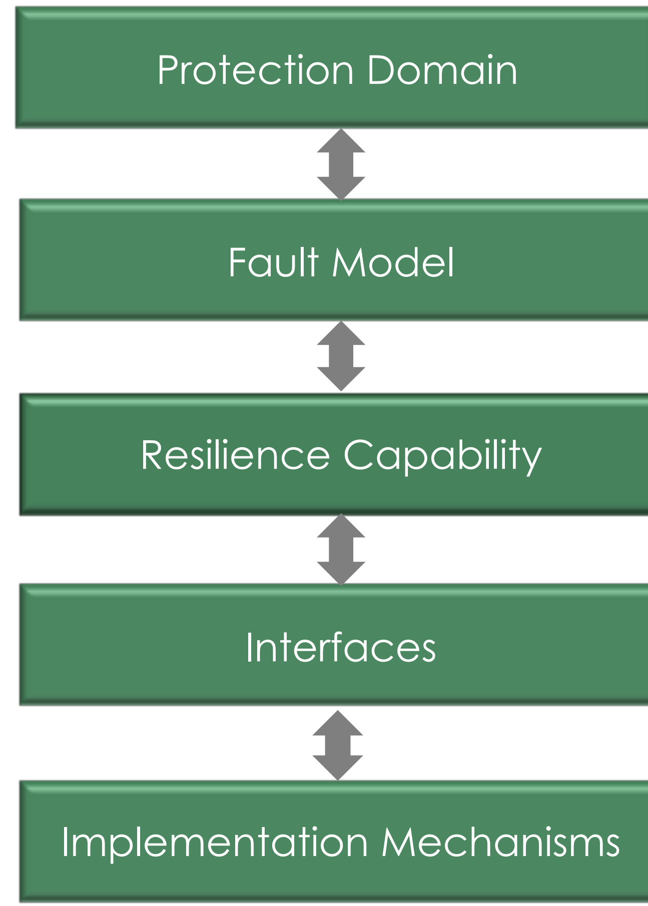
Oak Ridge National Laboratory

Case Study: Cross-Layer Hardware/Software Hybrid Solution



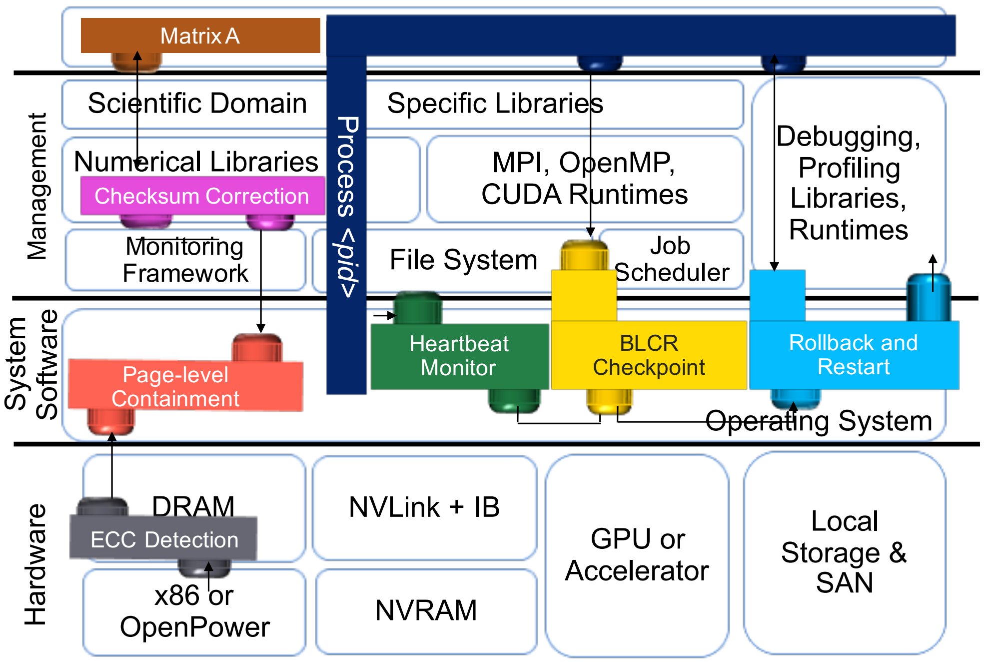
Resilience Design Spaces Framework

- 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

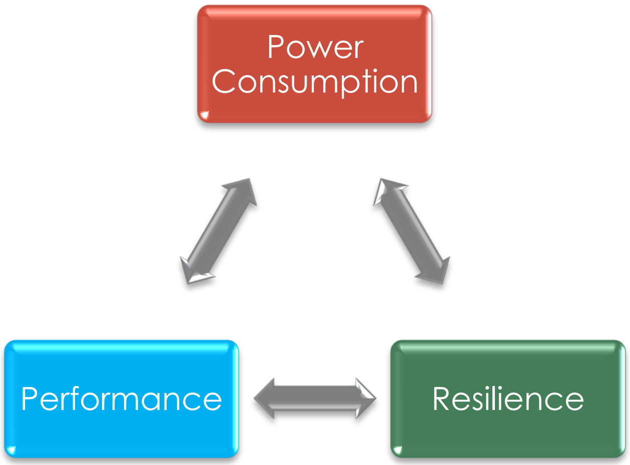
- Vertical and horizontal pattern compositions describe the resilience capabilities of a system
- Pattern coordination leverages beneficial and avoids counterproductive interactions
- Pattern composition optimizes the performance, resilience and power consumption trade-off



Oak Ridge National Laboratory

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



Resources and Contact

- Catalog: Characterizing Faults, Errors, and Failures in Extreme-Scale Systems
  - https://ornlwiki.atlassian.net/wiki/spaces/CFEFIES
- Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale
  - https://ornlwiki.atlassian.net/wiki/spaces/RDP

- Christian Engelmann, Oak Ridge National Laboratory
  - engelmannc@ornl.gov