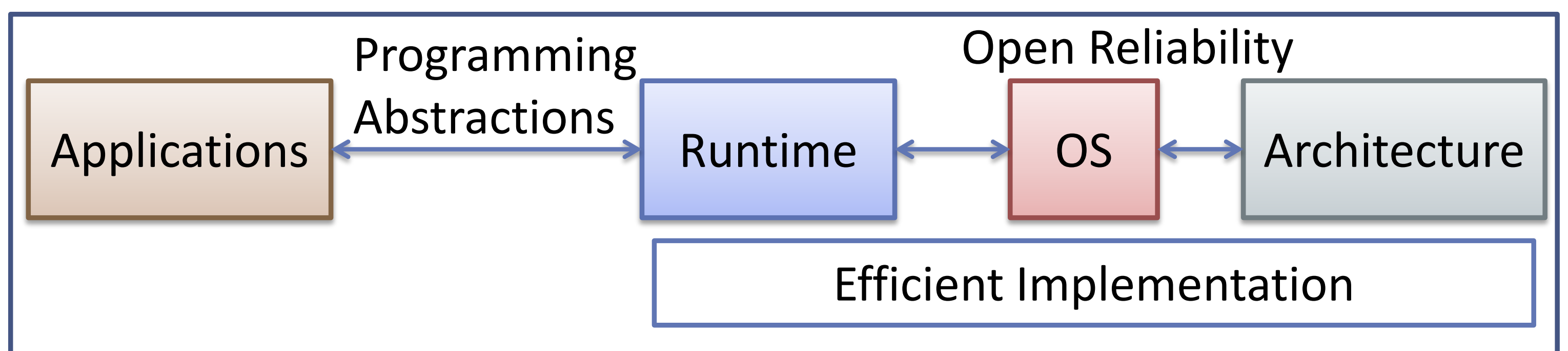
Exploiting Global View for Resilience (GVR)

Andrew A. Chien (PI), Hajime Fujita, Guoming Lu, and Zachary Rubenstein, *University of Chicago*; Pavan Balaji (co-PI), Pete Beckman, James Dinan, Jeff Hammond, Kamil Iskra, *ANL*; Robert Schreiber, *Hewlett-Packard Labs*

Background

- Resilience: a critical exascale challenge
- Examples of resilient large-scale systems
  - Scalable internet services
  - Batch internet-scale data processing
  - Internet
- Key features for reliable internet-scale systems
  - A foundation of reliable data
  - Programmer-managed, non-uniform reliability
  - Application-managed consistency

Goals

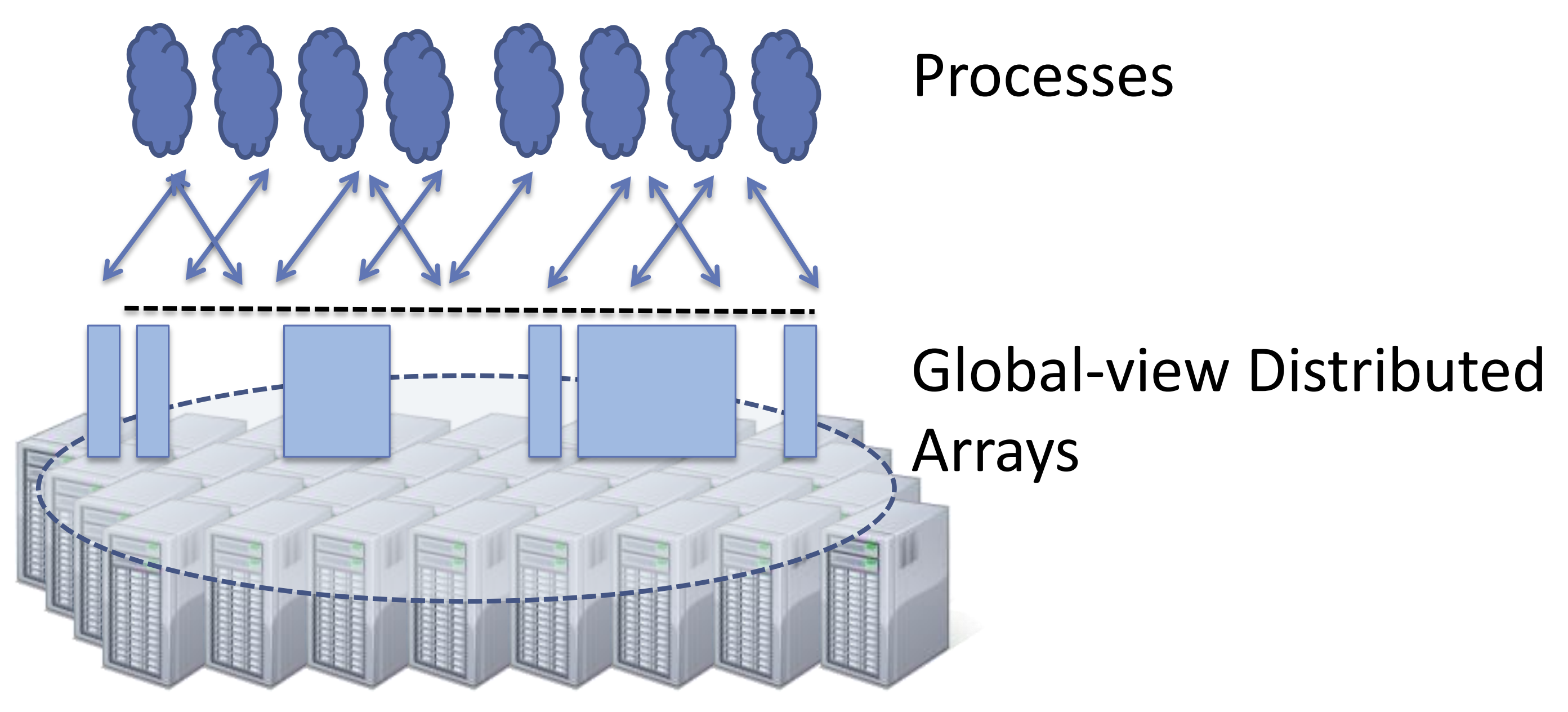


- Understand and create application-system partnership for flexible resilience
- Explore efficient implementation of resilient and multi-version data
- Create empirical understanding of GVR's effectiveness and performance requirements

Approach

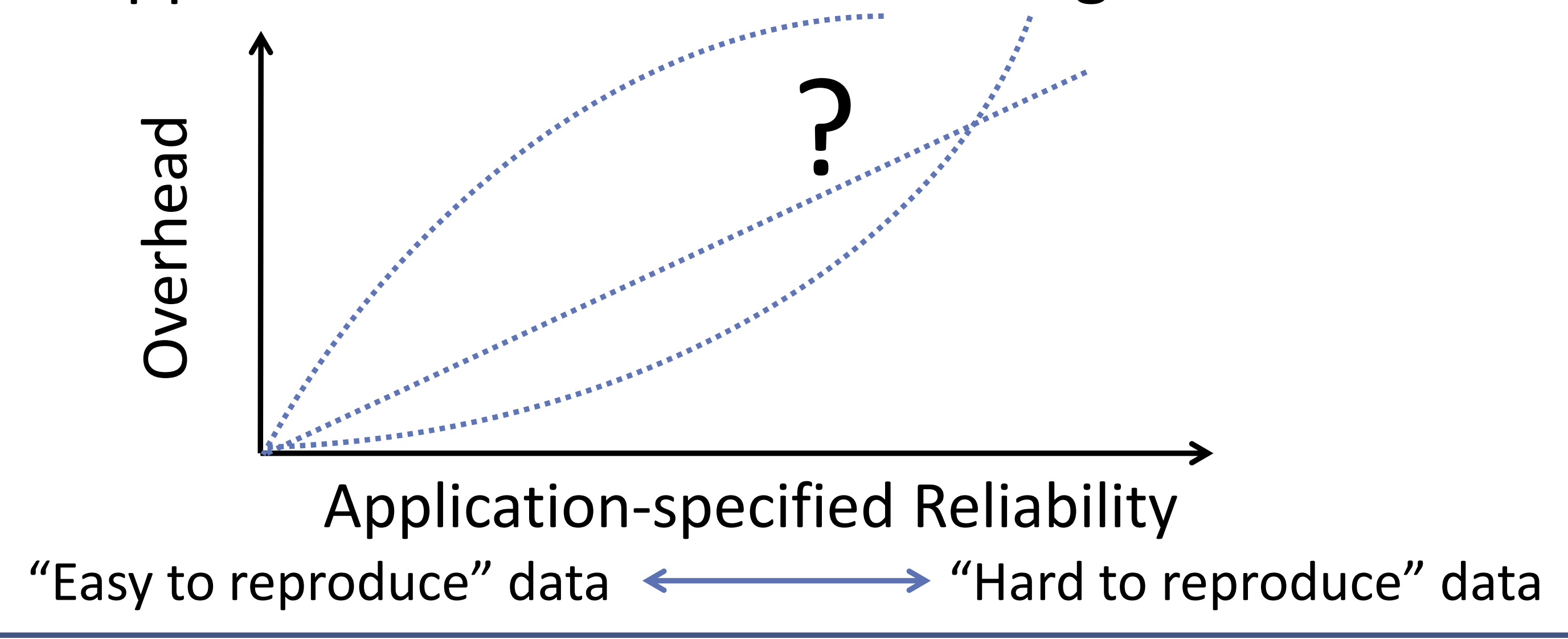
GVR (Global View for Resilience)

- Exploits a global-view data model, which enables irregular, adaptive algorithms and exascale variability
- Provides an abstraction of data representation which offers resilience and seamless integration of various memory/storage hierarchy
- Adds reliability to globally visible distributed arrays



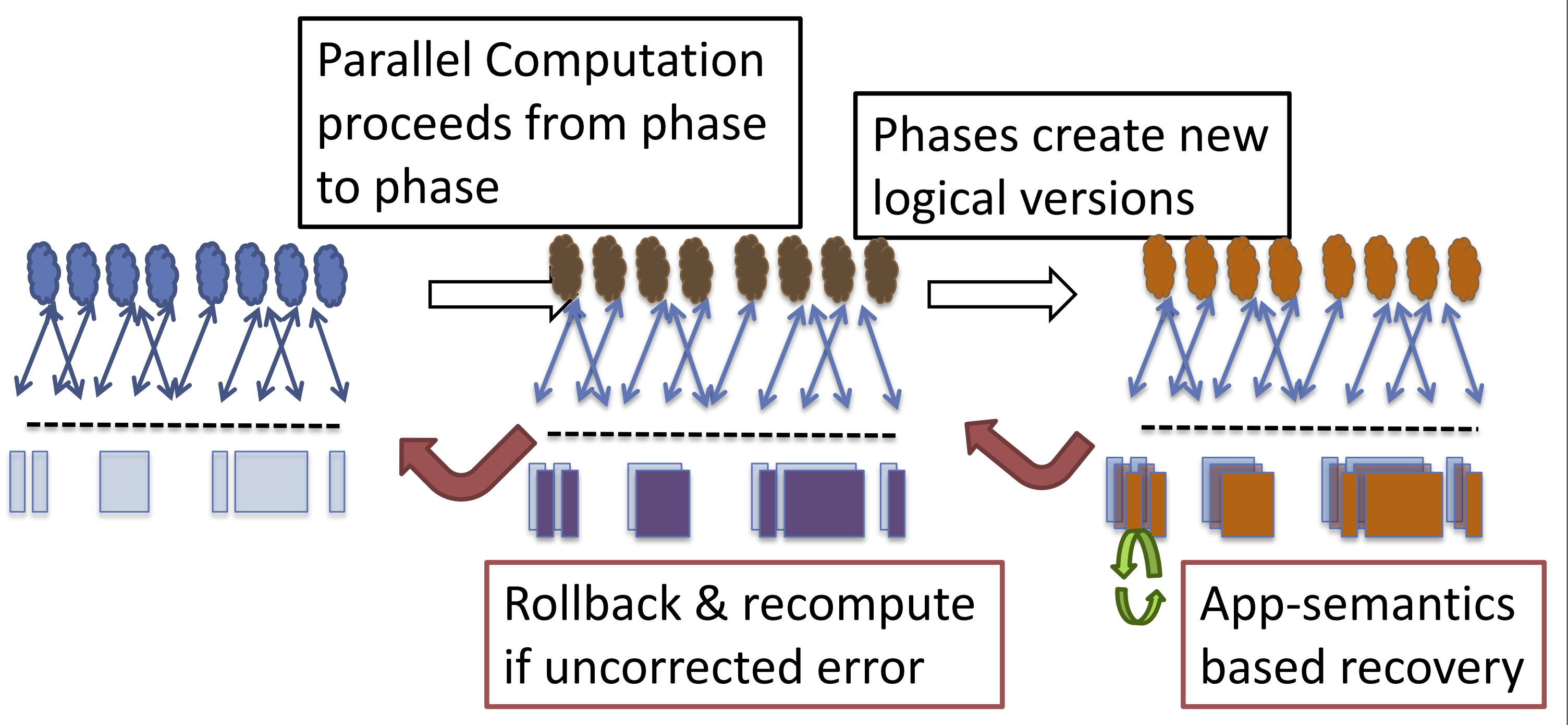
Reliability Priorities Specified by Applications

- Applications can specify which data are more important to protect so that they can manage reliability overheads
- Application-based error checking



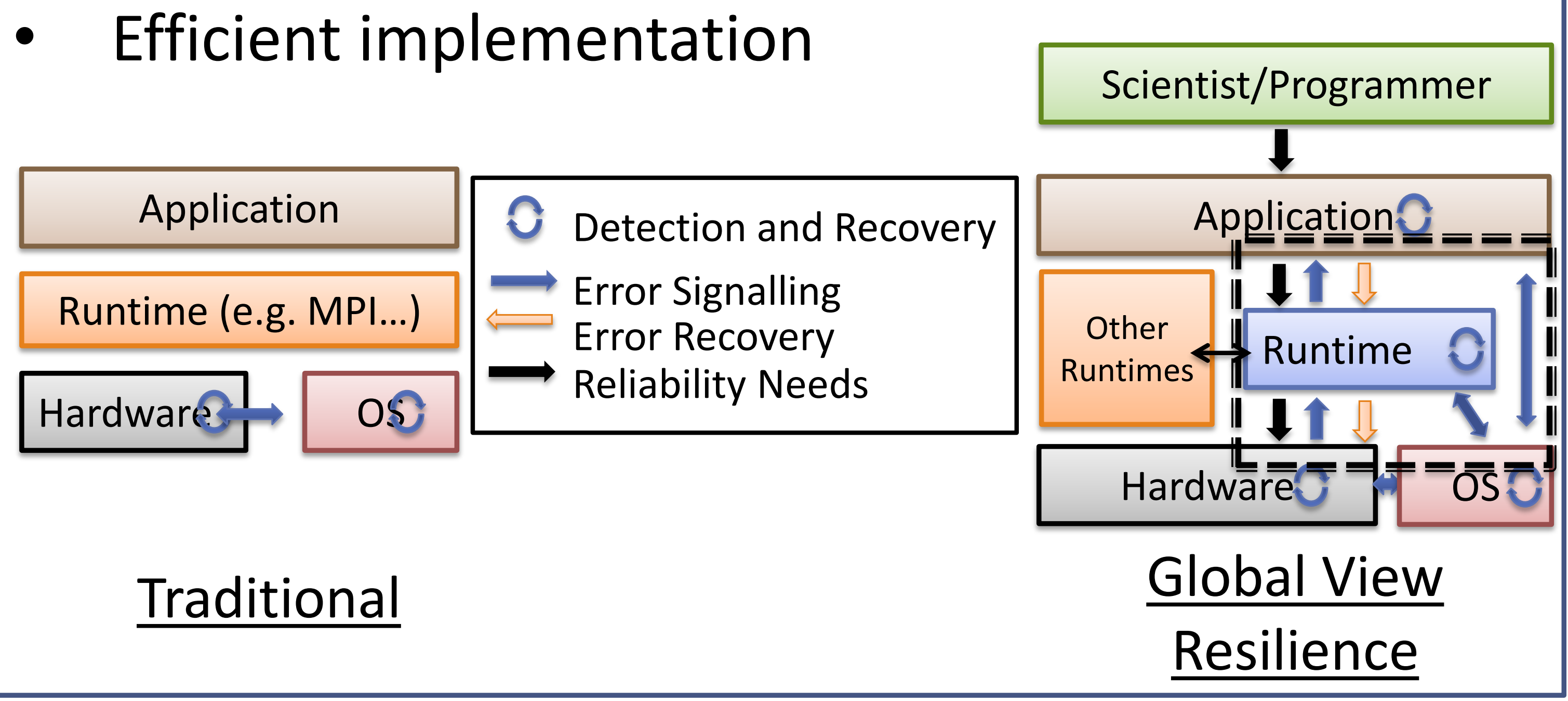
Multi-version Memory

- Computation phases form different "versions" of data
- A program can obtain and recover from earlier versions if needed


Parallel Computation proceeds from phase to phase
Phases create new logical versions
Rollback & recompute if uncorrected error
App-semantics based recovery

Cross-layer Partnership (App, Runtime, OS, Architecture)

- Rich error check
- Efficient implementation



Impact

- Incremental, portable approach to resilience for large-scale applications
- Flexible, application-managed cost and coverage for resilience

Research Challenges

- Understand application needs for flexible, portable resilience and performance
- Design of API suitable for use by application/library developers and tools
- Achieve efficient GVR runtime implementation for multi-version memory and flexible resilience
- Understand architecture support and its benefits
- Explore new opportunities created by GVR abstractions and its implementation technologies

Research Products and Artifacts

- Design of GVR API for flexible resilience and multi-version global data
- Research prototype software developed as a library; target of backend
- Assessment of opportunities and quantitative benefits of architecture support for GVR

U.S. DEPARTMENT OF ENERGY Office of Science
ASCR X-Stack Awards DE-SC0008603/57K68-00-145