Recap from the last workshop

- Workflows for big data and extreme computing share many characteristics
- Many stages in the computations, different algorithms for each stage
  - Diverse and often conflicting demands from system resources
  - Challenge of Interoperability

- From our Data Intensive Systems Group Lead: Usable software should be the leading topic in this area!
Software Architecture and Frameworks

- We know how to design for clusters
  - Componentization, Composability
  - Achieved at the cost of some performance loss

- We do not know how to design for the future
  - Heterogeneity => need for abstraction lifting
  - Data movement minimization => rethinking computations
  - Higher failure rates => algorithmic/redundancy

Chasm

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY
Why the Urgency?

- A huge array of design choices
- Available benchmarks and/or applications studies focus on one programming abstraction at a time
  - Usually fine-tuned to produce the best possible results to showcase the technology
  - Do not always translate to equivalent performance at production level
  - Almost no data about what happens when you mix them
- Separation of concerns will dictate that different abstractions handle different concerns
  - We need to figure out what the framework that enables this kind of interoperability will look like
  - Then we will need to figure out how to get there from here
  - Build tools and methodologies along the way

There are some efforts to refactor, but in the absence of knowing what should a code look like after refactoring, the developers are operating blind.
Example of Abstractions (SAMR)

- Abstractions in use
  - Stencil DSL for solvers
    - Address heterogeneity and operation fusion for performance
  - Tiling for fine-grain parallelism and memory management
  - Dynamic tasking for eliminating unnecessarily expensive bulk-synchronous parallelism

Logical Vs Physical View of the physical domain