

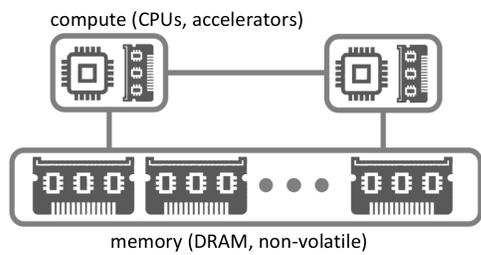
Prodasy:

PRincipled resource Optimization through declarative DATA SYStems

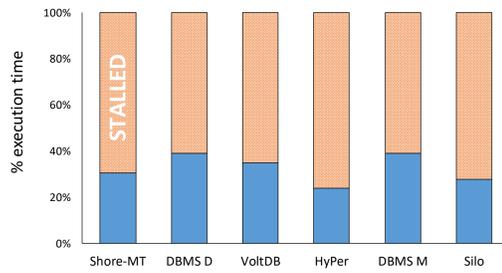
Data-Intensive Applications and Systems Laboratory

Architecture-conscious data systems

Abundant but heterogeneous processing and memory



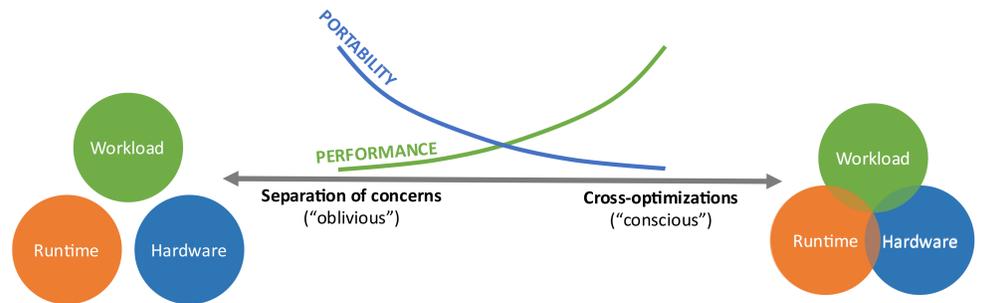
Processors wait for memory most of the time



Hardware evolves rapidly
Decades-old abstractions invalid

Performance – portability tradeoff

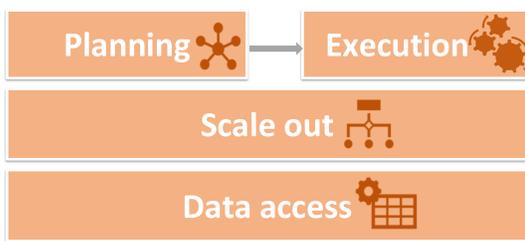
Performance – portability tradeoff



Declarative component interfaces → optimal resource allocation
Cross-optimizations possible *while maintaining separation of concerns*
Maximize portability AND performance

Research Questions

Which abstractions generate hardware-conscious executable code?

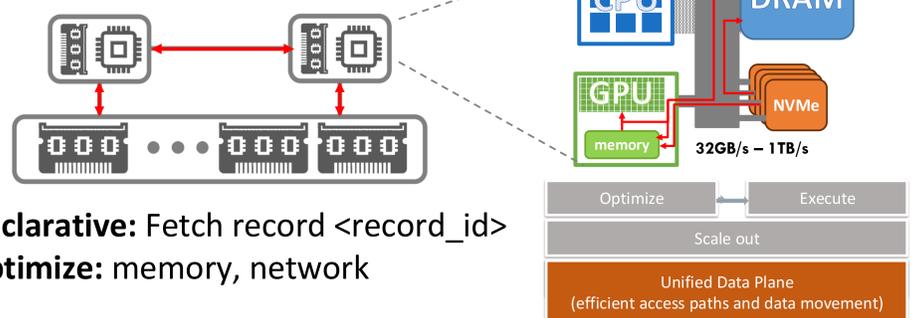


How to ensure scalability when scheduling distributed resources?

How to route and access heterogeneous data efficiently and transparently?

How to move data efficiently and transparently?

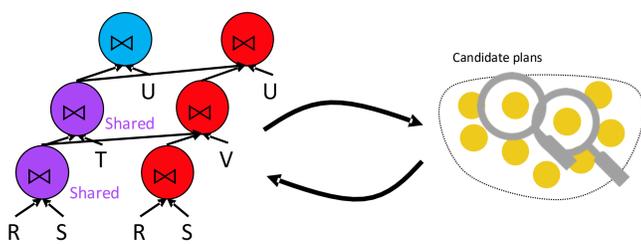
Now: Allocate (size, device...)
Memcpy (8KB, from to, Aligned...)



Declarative: Fetch record <record_id>
Optimize: memory, network

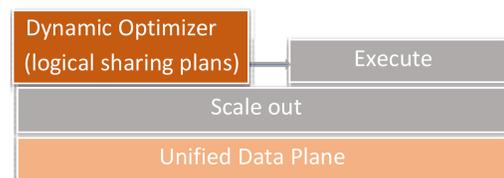
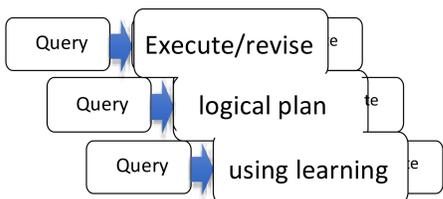
Ideas: dynamic access path generation, cross-system data routing

How to plan resources for concurrent requests?



Now: opportunistic runtime sharing

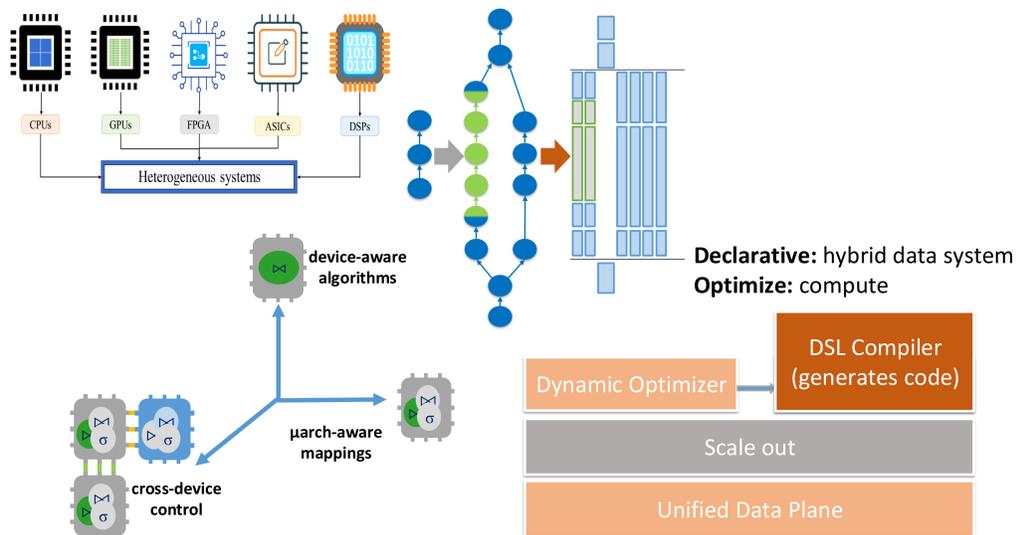
Declarative: continuously optimize global plan
Optimize: compute and memory



Ideas: evaluate plans at runtime, predict cross-task optimization

How to optimize across abstraction boundaries?

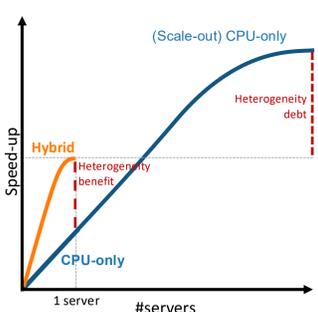
Now: CPU data system or GPU data system...



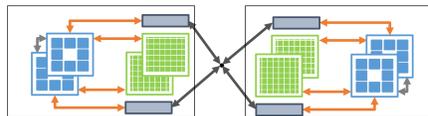
Declarative: hybrid data system
Optimize: compute

Ideas: abstractions to generate hardware-aware code at runtime

How to ensure scalable distributed resources?



Now: Provision: 5VM, machine specs, location, each process 10GB data, ...

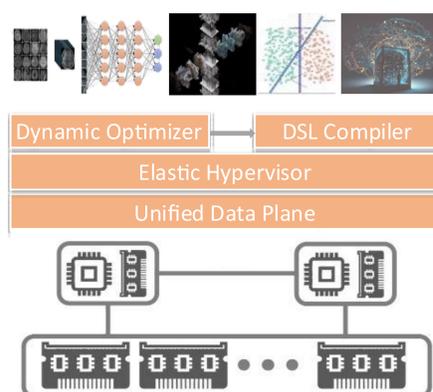


Declarative: Process 50GB, scale as-required
Optimize: compute, memory, network, power



Ideas: dynamic storage partitioning, virtual machine scheduling

PRODASY marks a paradigm shift



- Maintain separation of concerns
- Cross-module optimization at runtime
- Declarative system component interfaces
- Efficient resource allocation and utilization