LABORATORY FOR PHYSICAL SCIENCES

# Project 38

*Innovative Architectures for High-Performance Computing Systems*

Eric Cheng

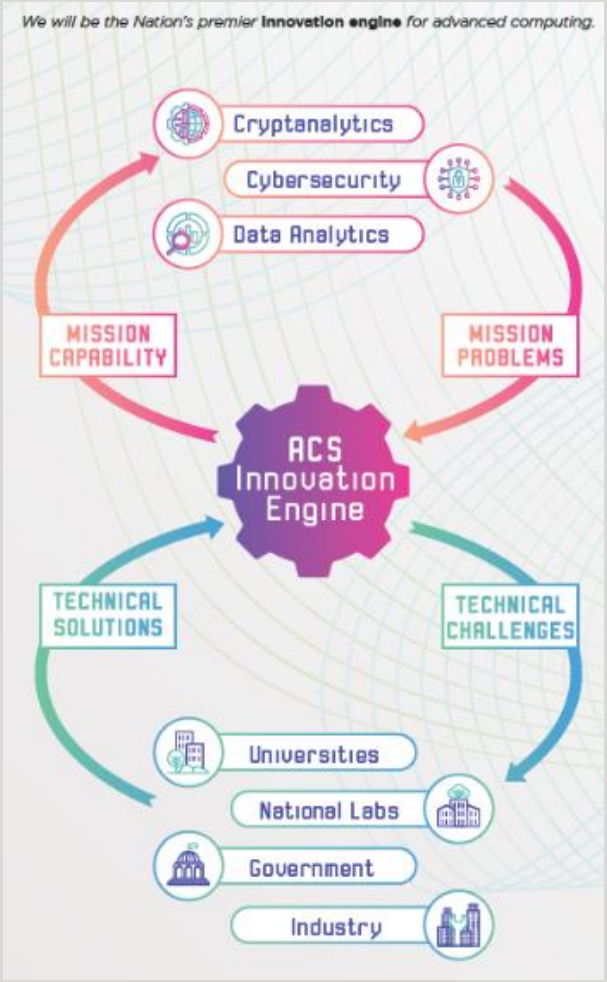
eccheng@lps.umd.edu

# Advanced Computing Systems (ACS) Research Program

We will be recognized, both internally and externally, as the **nation's premier innovation engine for advanced computing**.

We conduct **exploratory research** that combines *algorithms, architectures and technologies* to demonstrate and/or develop **advanced computing systems that provide asymmetric advantage for agency mission**.

Our innovation engine is built upon **mission oriented participatory research.**




Resilience

Productivity

Modeling and Simulation

Computer Architecture

Neuromorphic Computing

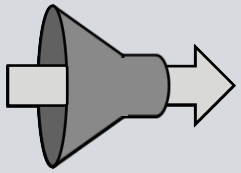
Energy Efficiency

# Computer Architecture and Computer Engineering

**Enabling Technologies**

- Big Global Address Space
- DWM, PeFET, RRAM
- Intelligent memory
- Memory benchmarking

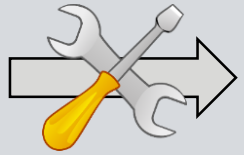
*feed*



**Rapid Development**

- 3D integration
- High-level synthesis
- Memory emulation
- Wafer-scale integration

*facilitate*



**Purpose-built Designs**

- Photonic architecture
- Project 38
- RRAM architecture
- Specialized compute

Mission: deliver innovative technologies/architectures to advance high-performance computing (HPC) systems

# Project 38

# Quick Facts

**Goals**

- Technical collaboration between NSA, DOE Office of Science, National Nuclear Security Administration (NNSA)
- Develop enduring capability for joint NSA/DOE technology innovations, explorations, and design
- Phase 1: quantify value/costs for specific architecture features against subset of NSA/DOE apps
- Phase 2: dedicated vendor engagements to accelerate transfer of technologies into roadmaps/ecosystems

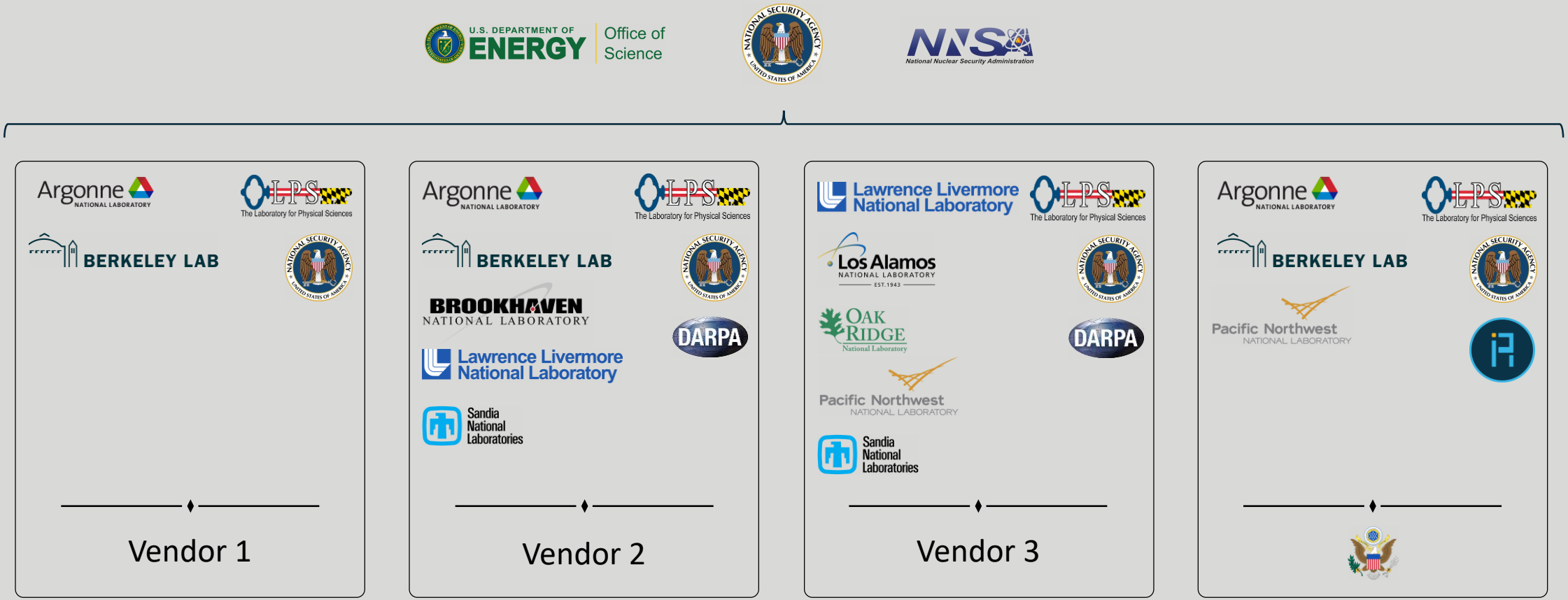
**Value of Strategic Investment**

- Buy: broaden adoption/market availability of targeted architectures/technologies
- Build: steer ecosystem towards purpose-built designs
- Use: increase toolchain, software, application support

**More Information**

- https://www.nitrd.gov/Presentations/files/HPC-Performance-Improvements-Project-38.pdf

# Project 38 “Org Chart”

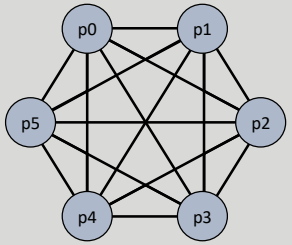


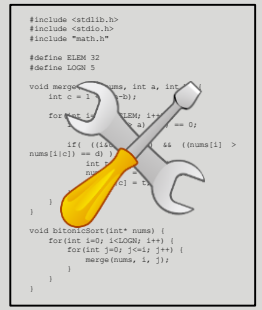

Parallel tracks: vendor engagements + innovative USG design

# US Government (USG) Investment Crucial

**Unique Requirements**

- Computation vs. communication balance
- Software productivity challenges



**Market Pressures**

- Commercial drivers (e.g., Google, Facebook, etc.)
- Narrowing domains (e.g., Cerebras)

[Google]



[Cerebras]

**Foreign Investment**

- Fugaku
- TaihuLight

[RIKEN]

[Xinhua: NSC]

# Quantifying Value and Impact for Select Features

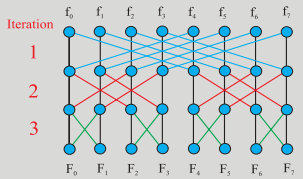
# Initial Drivers (Manageable Scope)

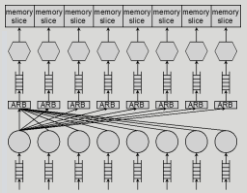
**Mix of applications**

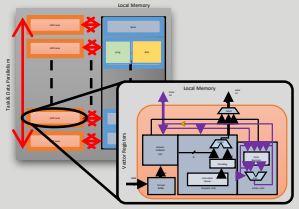
- Fast Fourier Transform (FFT)
- High-performance Geometric Multigrid (HPGMG)
- HipMER (bioinformatics)
- Kripke (radiation transport)
- Mini-tensor Contraction (MTC)
- Sparse matrix trisolve

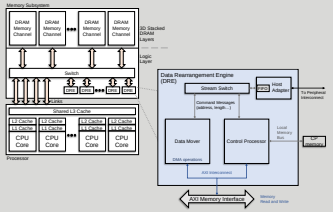
**Mix of hardware features**

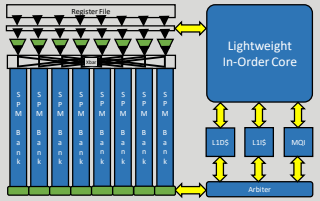
- Fixed function accelerators
- Message queues
- Recode engine
- Scatter/gather memory controller
- Word-granularity scratchpads







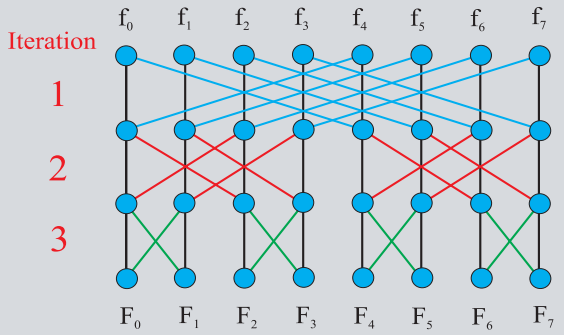



Impact: 40% to over 10× performance improvement

# Fixed Function Accelerators

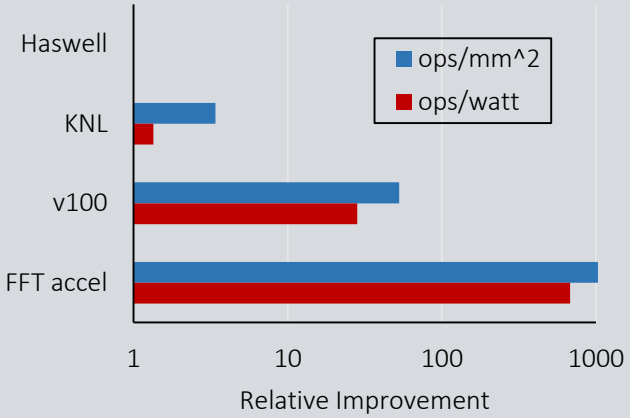
## Enabling Technology

- FFT accelerator
- Bioinformatics accelerator
- Compact, energy-efficient compute engines



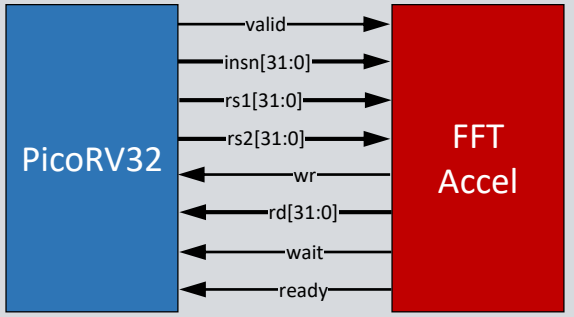

## Impact

- FFT evaluation:
  - 56× performance vs. GPU
  - 126× performance vs. CPU




## Integration

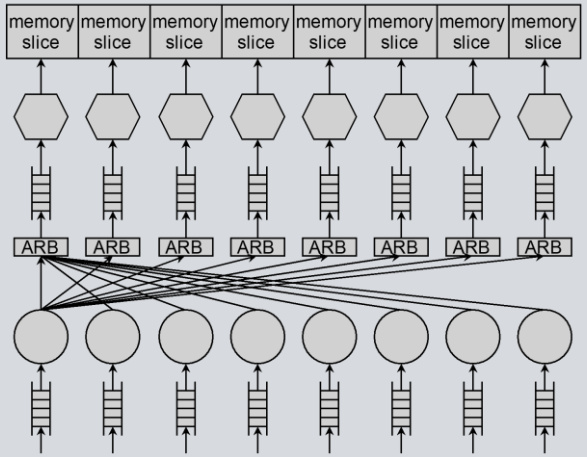
- Generators (e.g., SPIRAL)
- Leverage well-known interfaces / APIs
- Possible ISA extensions



# Message Queues

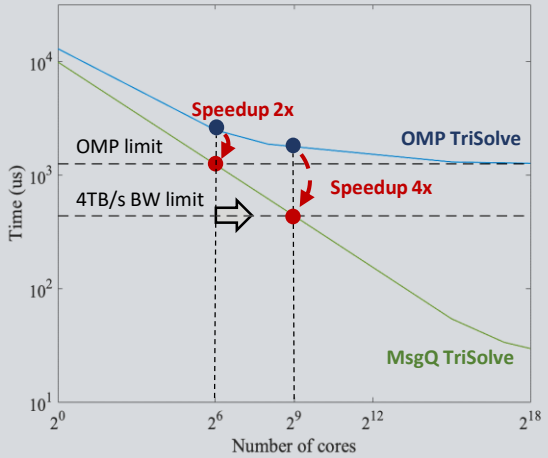
## Enabling Technology

- Efficient inter-processor communication
- Reduce need for mutex




## Impact

- 2-8× performance
- 5-12× latency reduction




## Integration

- Hardware IP available
- Straightforward intrinsics interface

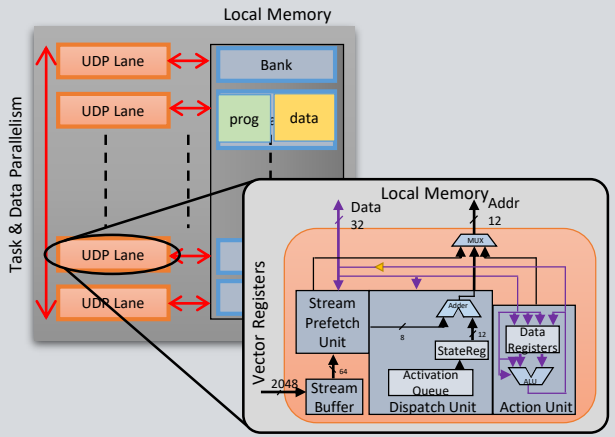
```
1. Initialization: x := b;  lsum := 0
2. while (I have work to do) do
4.     if (pollQ() > 0)
5.         getQ(&vfunc, &K)
3.         fmod(K) = fmod(K) − 1 or fmod(K) = −1 by &vfunc
26.    endif
4.     if (fmod(K) = −1 and I own this column)
5.         for (each of my L(I, K) ≠ 0, I > K)
6.             lsum(I, myid) = lsum(I, myid) + L(I, K)x(K)
7.             putQ(&id, &vfunc, &I) for id on diagonal block
18.        end for
19.    else if (fmod(K) = 0 and I own diagonal block)
20.        x(K) = x(K) − lsum(K, id) for each id in row K
23.        x(K) = L(K, K)^−1 x(K)
7.         putQ(&id, &vfunc, &K) for each id in column K
26.    endif
27. end while
```

Algorithm 2: Lower triangular solve $Lx = b$ with TaskQ.

# Recode Engine

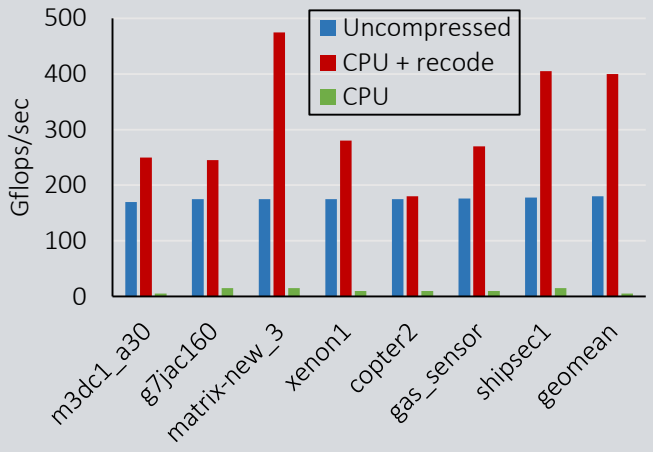
## Enabling Technology

- High-speed, fine-grained data recoding
- Programmable transforms



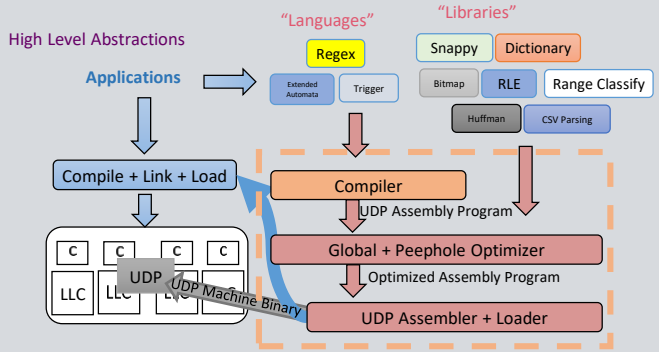

## Impact

- SpMV evaluation:
  - 5× performance increase
  - 8× bandwidth reduction
  - 63% power reduction




## Integration

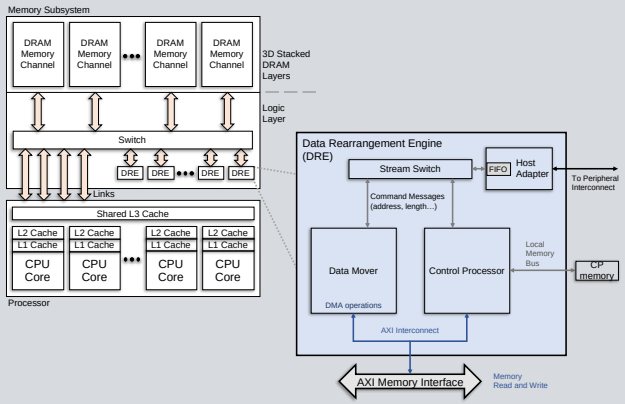
- Synthesizeable RTL
- Emulation infrastructure
- LLVM compiler support
- Rich library support



# Scatter/Gather Memory Controller

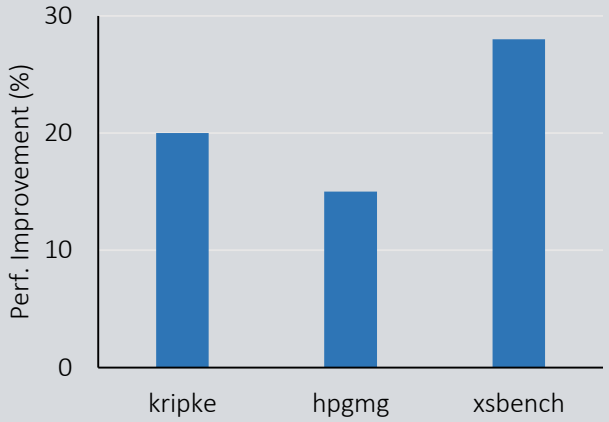
## Enabling Technology

- Data Rearrangement Engine (DRE)
- Near-memory scatter/gather



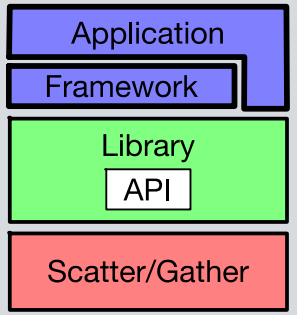

## Impact

- 15-28% application execution time improvement
- 18-50% cache miss reduction




## Integration

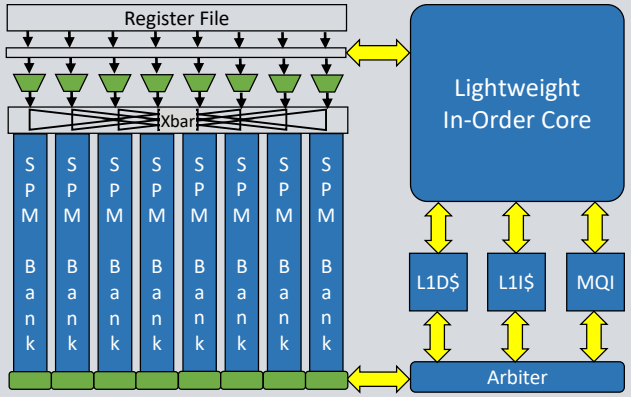
- Portability frameworks (Kokkos, Raja)
- Emulation infrastructure (LiME)
- Programming model (setup/fill/drain)



# Word-granularity Scratchpads

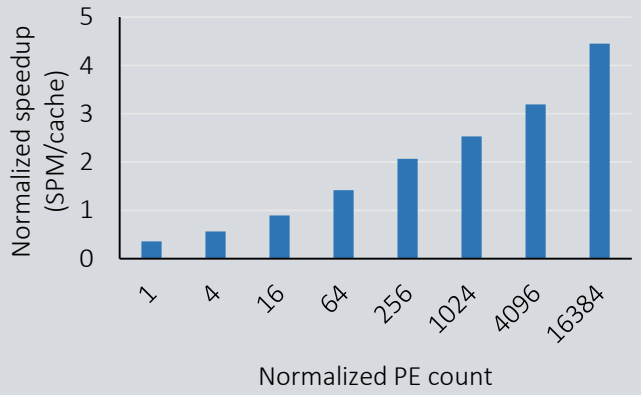
## Enabling Technology

- On-chip scatter/gather
- Avoid memory coherence protocol overhead



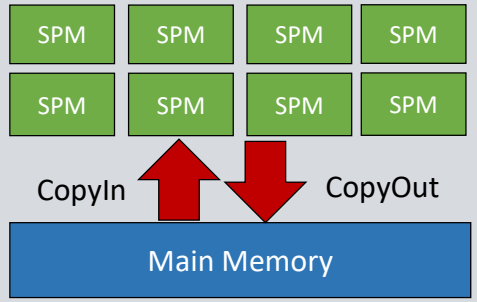

## Impact

- 3× larger capacity
- HPGMG: 4.5× performance
- MTC: 458% data movement reduction




## Integration

- Existing HW IP
- Pragmas / modified library
- Rose compiler support
- Possible side-by-side cache/SPM options



# An Innovative USG Design for HPDA / Graph Analytics

# High-performance Data Analytics (HPDA) System

**Mission Focused**

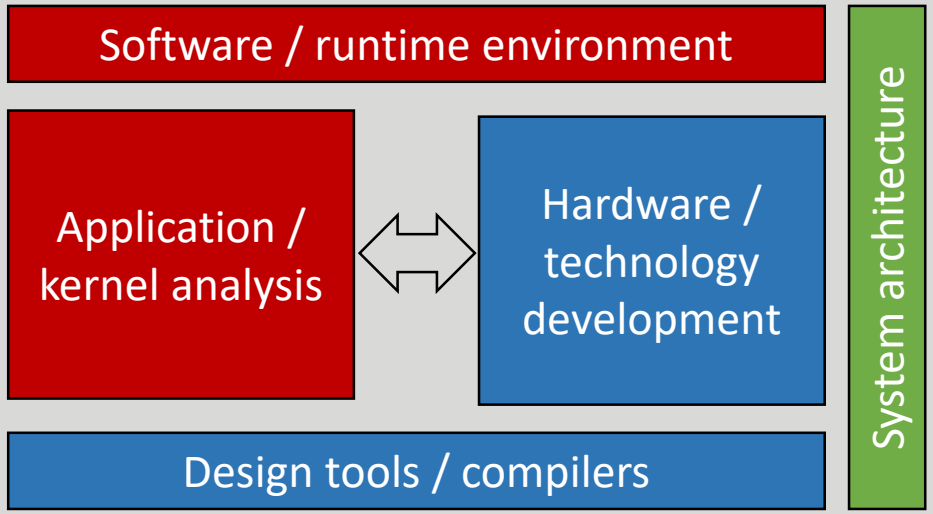
- High-performance Data Analytics (HPDA)
- Graph analytics at scale
- Data science at scale

**Key Drivers**

- User productivity
- Scale, performance, energy-efficiency, cost
- Emerging technologies

**Synergies**

- IARPA AGILE
- PNNL Data Model Convergence




Goal: system-level design for complete end-to-end workflows

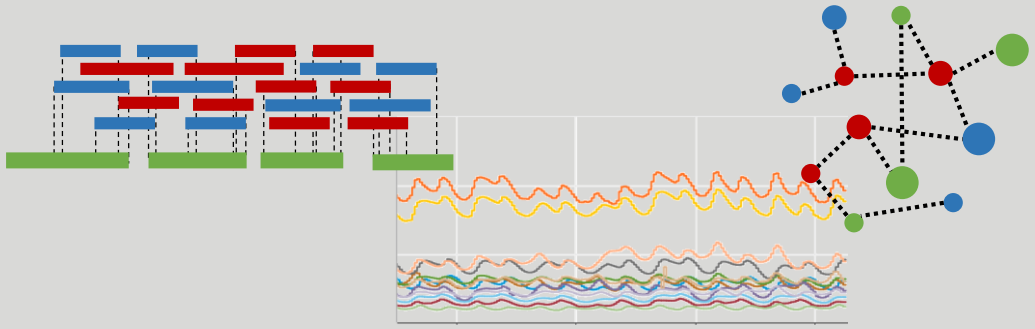
# Convergence of Applications

**Challenge Problems**

- Breadth First Search (BFS)
- Triangle counting
- Jaccard similarity
- PageRank
- Global RandomAccess (GUPS)
- Connected components
- k-truss decomposition

**Overlapping Domains**

- Bioinformatics
- Knowledge graphs
- System and event pattern detection
- Sequence data
- Cyber-physical systems



[eia]

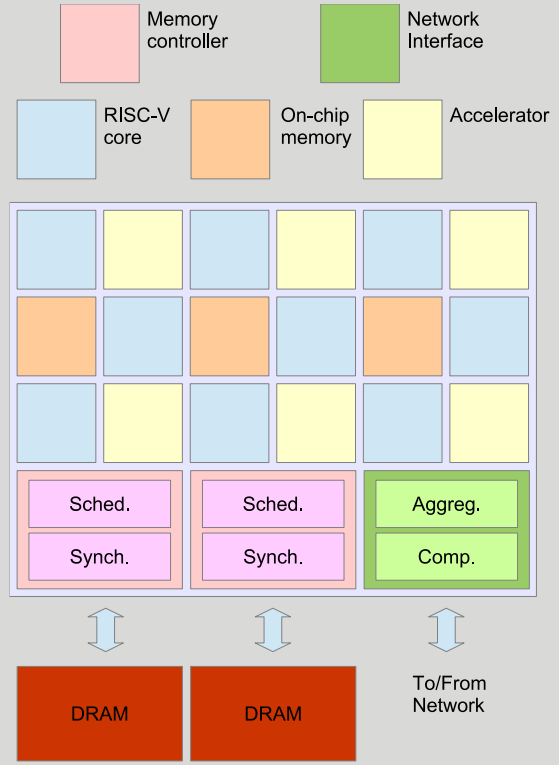
# Convergence of Architectures / Technologies

**System Architecture "Fabric"**

- Compute (processors, accelerators)
- Network, interconnect
- Memory and controllers
- …

**Technologies**

- Processing-in-memory (PIM)
- Fabric-attached memory (FAM)
- Photonics
- Emerging memory technologies
- …



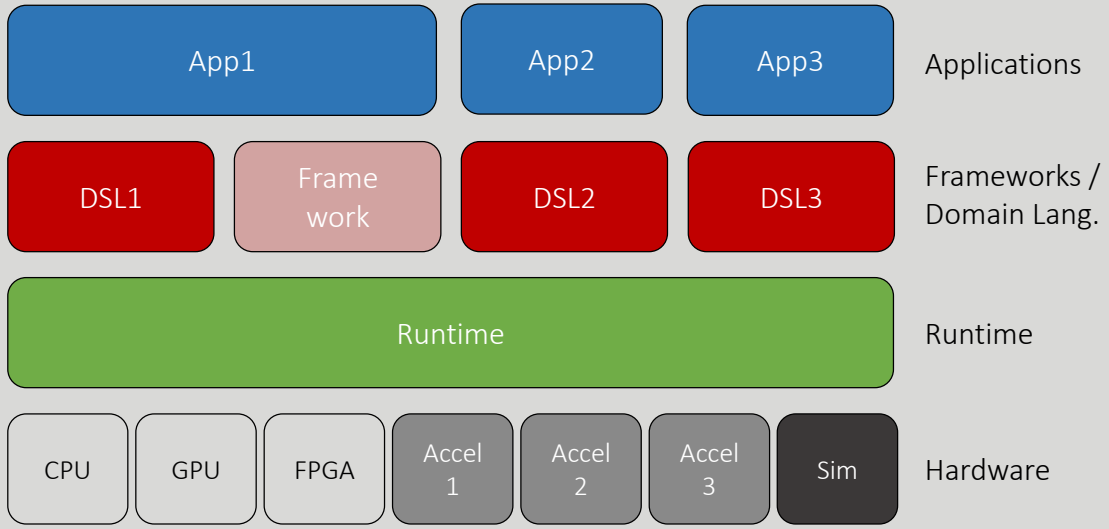
# Convergence of Tools

## Hardware Design Tools

- High-level Synthesis (HLS)
- Accelerator generators
- Open-source RTL / ISAs
- Open-source EDA / design libraries

## Software Infrastructure

- Compiler advancements
- Frameworks / abstractions
- Dynamic runtimes / libraries
- New representations



# Come Work With Us

# LPS Hiring Areas

- Energy efficiency for HPC and new architectures
- HPC system software – runtime and application development
- Math & Computer Science for novel computing
- Computer Architecture
- High Performance Data Analytics
- EE, CE, CS focus, but STEM in general

# Parting Thoughts

**What**

- Cross-agency effort (NSA, DOE SC, NNSA)
- Architecture innovations for HPC

**Why**

- Broaden capabilities (buy)
- Reduce cost (build)
- Increase productivity (use)

**Contact**

- eccheng@lps.umd.edu
- https://www.nitrd.gov/Presentations/files/HPC-Performance-Improvements-Project-38.pdf