IARPA AGILE Program

ModSim 2023

William Harrod | August 10, 2023

Intelligence Advanced Research Projects Activity

IARPA

Creating Advantage through Research and Technology

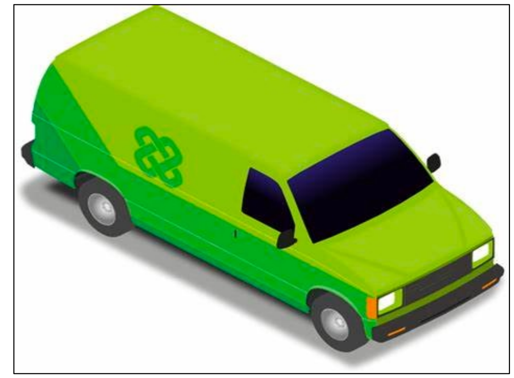
AGILE

VALIDATE DESIGN WORKFLOWS

The Data Problem

What are we trying to accomplish	• **Solve analytic problems that involve at least 10 - 100 times more data** • **Time to solution 10 - 100 times faster** • **Reduce development time for analyst/programmer efforts**
What is wrong with current solutions	• **Traditional systems are CPU/ALU focused designs** • **Current and future planned systems don’t scale for data analytic problems** • **Poor efficiency and productivity**
End user needs	• **Support for multiple programming models and languages** • **<u>Massive data sets that are dynamically changing and streaming</u>** • **Data analytics: graph algorithms, machine learning & more**

AGILE Program Details



Program Objectives:

- Near-real-time solutions for emerging data problems
- Create new generation of data-centric computer architectures

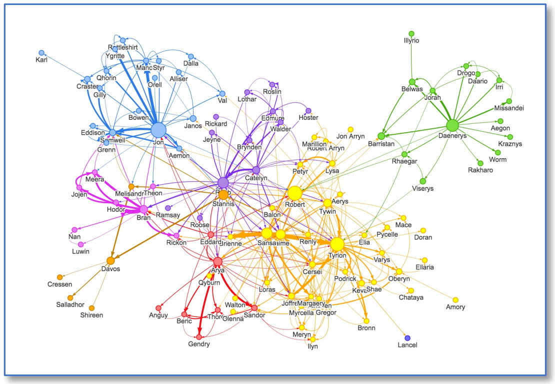
Research Effort:

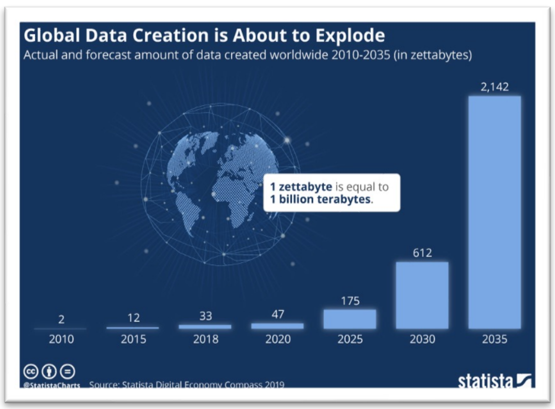
- Develop innovative full-system architectural designs
- Demonstrate achievement of AGILE (data analytic) Target Metrics using model & simulation software.
- Independent test and evaluation (T&E) team will validate results.

Deliverables:

- **Phase 1 (18 months): System-level functional model** of architecture, including runtime.
- **Phase 2 (18 months): Detailed (RTL) design of AGILE system architecture** (foundation for system build)

Why Do We Need AGILE?





- **End of** easy **advancements via Moore's Law/Dennard Scaling**
  - Research on new architectures is the only realistic pathway to performance gains for emerging problems
- **Data explosion:** volume, velocity, variety, complexity
- **Industry not solving the problem;** driven by different market forces
  - Industry is facing a critical technology turning point; advancements are difficult, unpredictable, expensive
- **Need actionable knowledge**
  - Data analytics are used to transform data into actionable knowledge.
  - Data analytics operate on graphs.
  - **Data analytics have minimal data locality, poor data re-use, fine-grain data movement and data driven parallelism – existing systems are not designed for these features**.

High Performance Conjugate Gradients (HPCG) Benchmark

May 2023

Conjugate Gradient (CG) Algorithm – iterative algorithm for solving sparse linear systems

HPCG Rank	HPL (TOP500) Rank	Site	System	Cores	HPL (PFlop/s)	HPCG (PFlop/s)	Efficiency
1	2	Riken/CCS, **Japan**	**Fugaku -** A64FX 48C 2.2GHz, Tofu interconnect D	7,630,848	442.01	16.00	3.6%
2	1	Oakridge National Laboratory, **United States**	**Frontier -** HPE Cray EX235a, AMD	8,699,904	1194.00	[illegible].05	1.2%
3	3	EuroHPC/CSC, **Finland**	**LUMI -** HPE Cray EX2[illegible]	[illegible]	[illegible]	[illegible]	1.1%
4	4	EuroHPC/CINCEA, **Italy**	**Leonardo -** BullSequa[illegible] XH2000, Xeon Platinu[illegible]	[illegible]	[illegible]	[illegible]	1.3%
5	5	Oakridge National Laboratory, **United States**	**Summit -** IBM Power System AC922, NVIDIA Volta	2,414,592	148.60	2.93	2.0%

Dense Algorithm (HPL): 82% of Peak
Sparse Algorithm (HPCG): 3.6% of Peak

HPCG operates on a very large sparse data set. It is a highly optimized code.

Application Characteristics

Results needed in near-real-time

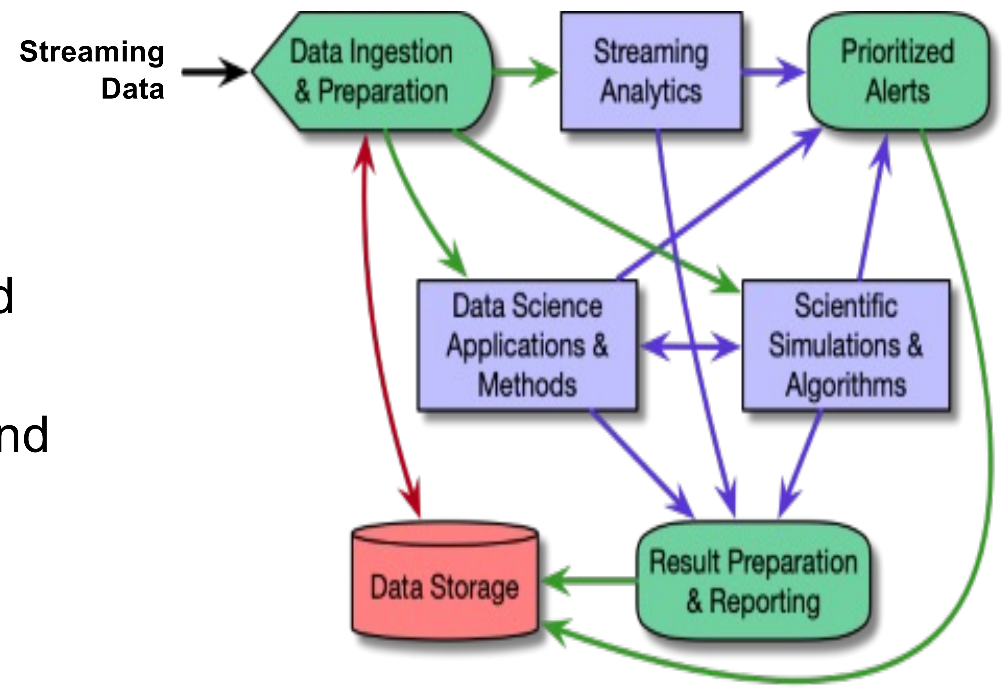
- Near-real-time means minutes to hours

Streaming data causes unpredictable changes to stored data

Extremely fine grain data movement and parallelism

Computations determined by the data and streaming queries

Poor data locality and data reuse



Today's computers are not efficient or productive for these characteristics

Workflows and micro-Kernels

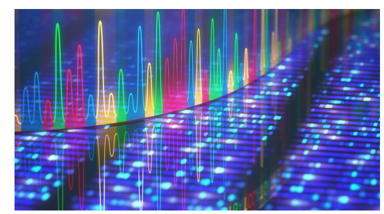
- Given the heterogeneity and complexity of data analytic workloads, kernels that measure individual metrics - FLOPS, TEPS, cache misses, network bandwidth - for a single data type cannot reflect the performance and scalability of full applications
- Only **end-to-end workflows** can reflect the performance and scalability of real-world analytic jobs
  - Ingestion, transformation, and storage of input data can take significant time, energy, and machine resources
  - Prioritization and display of output results can be costly
- **Micro-Kernels** are still valuable when measuring the speeds-and-feeds of individual system components and when systems/tools are too immature to run complete workflows

Driving Applications

Workflows are evaluation tools to measure the performance and scalability of AGILE designs.

They are representative of key AGILE graph computational challenges.

- **Workflow 1 – Knowledge Graphs:** Represent a network of real-world entities—i.e., objects, events, situations, or concepts—and illustrate the relationships among them. Tasks: ID new things, new relationships and indirect connections.
- **Workflow 2 – Detection:** Detect systems and event patterns in a property graph. Tasks: identify exact, approximate and partial match of a target pattern against the graph.
- **Workflow 3 – Sequence Data:** Identify and cluster data sequences using auxiliary data. Tasks: Assemble the correct DNA configuration from its component protein sequences (kmers) plus auxiliary data.
- **Workflow 4 – Networks of Networks:** Represent and analyze cyber-physical systems. Tasks: build a single graph from multiple related graphs, identify influential nodes for the graph, eliminate this node and determine resulting graph properties.



AGILE Co-Design Process

Drive architecture development process with *realistic* AGILE application workflows and datasets – Scaling up a system to achieve performance metrics isn't interesting

Utilize Structural Simulation Toolkit (SST) and FireSim

A multi-phase iterative co-design process is required to design a **well-balanced scalable system**

- ***Performance* –** time to solution estimates for AGILE Applications
- ***Productivity* –** minimize effort required to develop high-performance applications
- ***Efficiency* –** minimize number of resources required to complete task

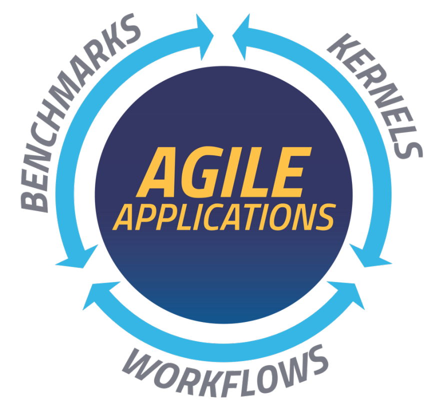
Stage	Quanty Optimized
Performance	Time estimates of AGILE Applications (time to solution)
Productivity	Minimize number of lines of code in comparison to optimized code
Efficiency	Analytic Models: Little's Law, Amdahl's Law, Bottleneck Analysis, M/M/1, Roofline Models, etc.

AGILE Applications

- AGILE Applications (developed at PNNL)
  - Includes workflows, kernels and industry benchmarks
  - Test programs / scripts
  - Data sets or generators
- Reference codes will be written using SHAD
  - Presents a shared–memory view of global memory
  - STL-complaint, thread-safe, distributed data structures
  - Concurrent insert/delete/modify and AMOs on all data structures
  - Asynchronous data and task parallel programming constructs
  - Multithreaded runtime that hides latencies (no data partitioning necessary)
  - Runs on servers and clusters
  - https://github.com/pnnl/SHAD
- **Algorithms can be substituted if they provide the same functionality**

SHAD

SCALABLE
HIGH-PERFORMANCE
ALGORITHMS &
DATA-STRUCTURES



AGILE Applications: Workflows and Benchmarks

BENCHMARKS

Breadth First Search

Counting Triangles

Jaccard Similarity

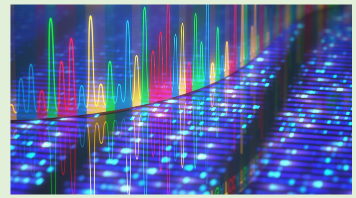
WORKFLOWS

KNOWLEDGE GRAPH

Groups, Relationships & Interests

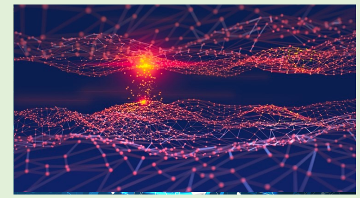
PATTERN DETECTION

System and Event Patterns



SEQUENCE DATA

Identification and Clustering

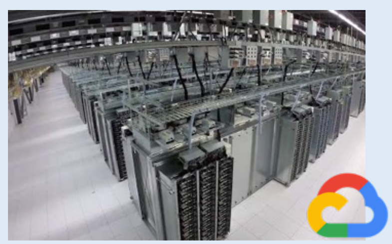


NETWORK

Networks of Networks

STREAMING

Big Data (today)

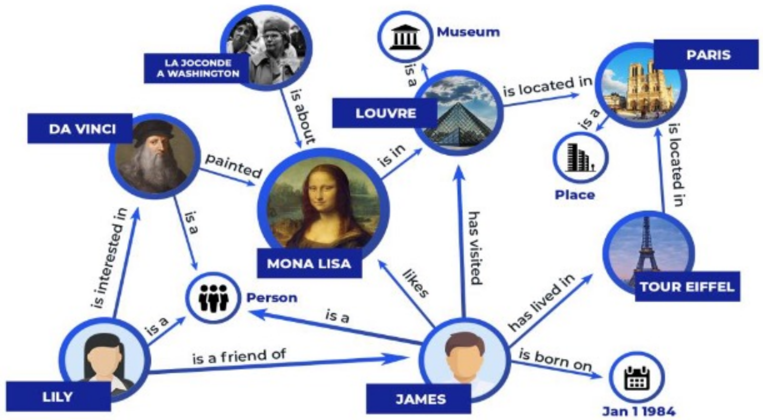


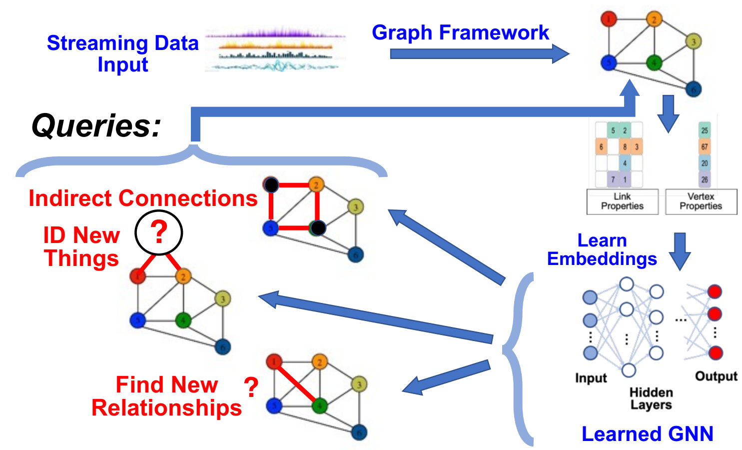
“Forensic Analysis”

Streaming Data Analytics

“Predictive Analysis”

Workflow 1 – Knowledge Graph





What is it:

- A semantic network of persons, places, objects, events, situations, or concepts, and the relationships among them
- Integrates multiple data sources with disparate types of entities (vertices) and relationships (edges)
- Ontologies are used to establish a logical, hierarchy of types creating a formal representation of the entities in the graph

Knowledge graph use cases:

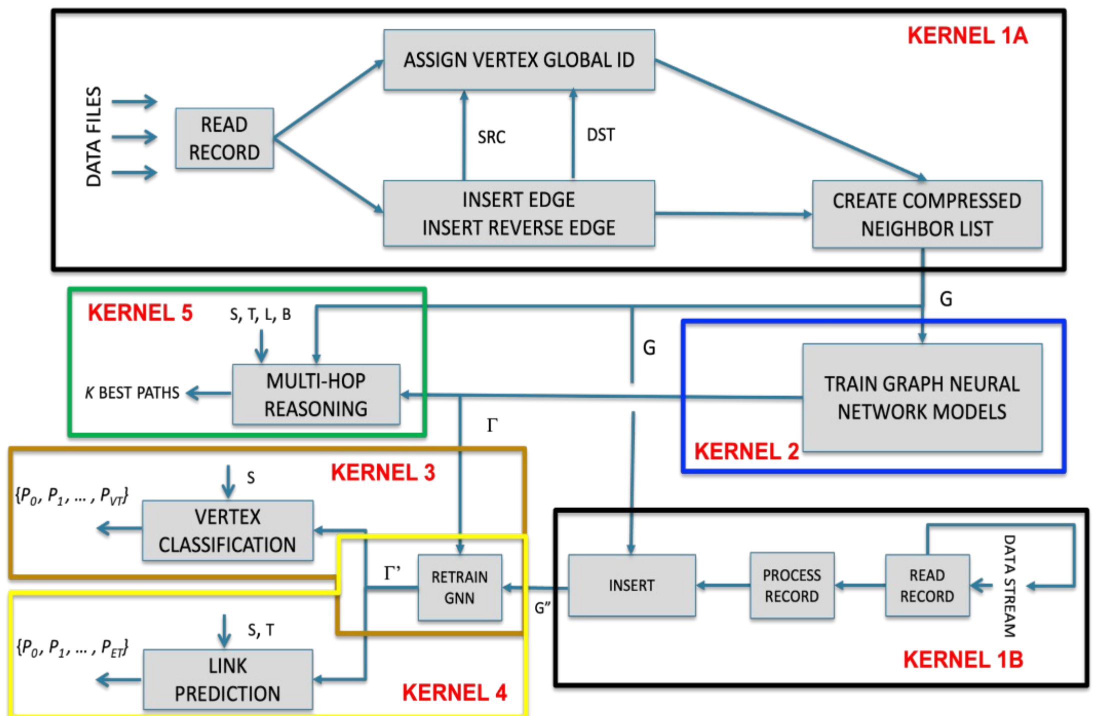
- Discover new entities, relationships & facts
- Explain the contextual reasons for a particular event
- Explain why a human expert should look at emerging event
- Answer complex questions that are beyond database queries

Workflow 1 – Kernels

- **Multi-hop Reasoning – Kernel 5**
  Indirect connections
  given vertices *s* and *t* in G, return the "best" *k* paths from *s* to *t*

- **Vertex Classification – Kernel 3**
  ID new things
  given unlabeled *v* in G with properties (p_1, … p_n) and incident edges {e_1, … e_k}, return the type of *v*

- **Link Prediction – Kernel 4**
  Find new relationships
  given *s, t* in G such that edge {*s, t*} does not exist in G, predict the existence and edge type of {*s, t*}



Workflow 1: Target Metric Table

Kernel	Metric	Today	AGILE Target
1A	Data Ingestion Rate (file): Time to read a data file and build internal data structures	1 G (G=10^9) data-element file per 1 minute	1 G data-element per 1 second (60x faster)
1B	Data Ingestion Rate (streaming): Time to process streaming data and insert data into internal data structures	0.1 G data-elements per second from a single source, single data type	10 G data-elements per second from 3 or more sources and data types (100x faster for each of 3 sources)
2	Learn models: Time to construct embedding and train GNN models	> 1,440 minutes	30 minutes (50x faster)
3	Classify vertices: Time to retrain model and classify unlabeled vertices in data streams	> 1,440 minutes	30 minutes (50x faster)
4	Predict and infer a new relationship: Time to retrain model and infer a new relationship in data streams	> 1,440 minutes	30 minutes (50x faster)
5	Perform reasoning: Time to reason about higher-order relationships using multi-hop reasoning	1 to 2 hops and branching factor not greater than 3 in 30 minutes	3 to 5 hops and score dependent branching in a minute (30x faster)

Technical Approaches

Classical Computer	AGILE Computer
Fragmented system: 1) subsystems improved independently 2) communication via message passing	**Tightly integrated system:** components (communication, memory, compute & runtime) co-designed simultaneously
Pre-programmed sequential processing: of streams of instructions causing load/store of data	**Data-driven processing:** including moving the compute to the data
Local memory management: and zones of trust. Deep hierarchical memories	**Distributed memory management:** and security with fine-grained addressing and protection of objects
Static data movement mechanisms: networks designed to move large messages only	**Intelligent data movement mechanisms:** supports massive numbers of random, time-varying small messages
Local name space: localized on the node; data transfer is driven by pre-programmed instructions	**Global name space:** global adaptive data transfer driven by complex workflow requirements
Static runtime: resources are pre-determined with no hardware support	**Dynamic runtime:** adaptive, with continuous optimization of resource usage and hardware support

AGILE Performers

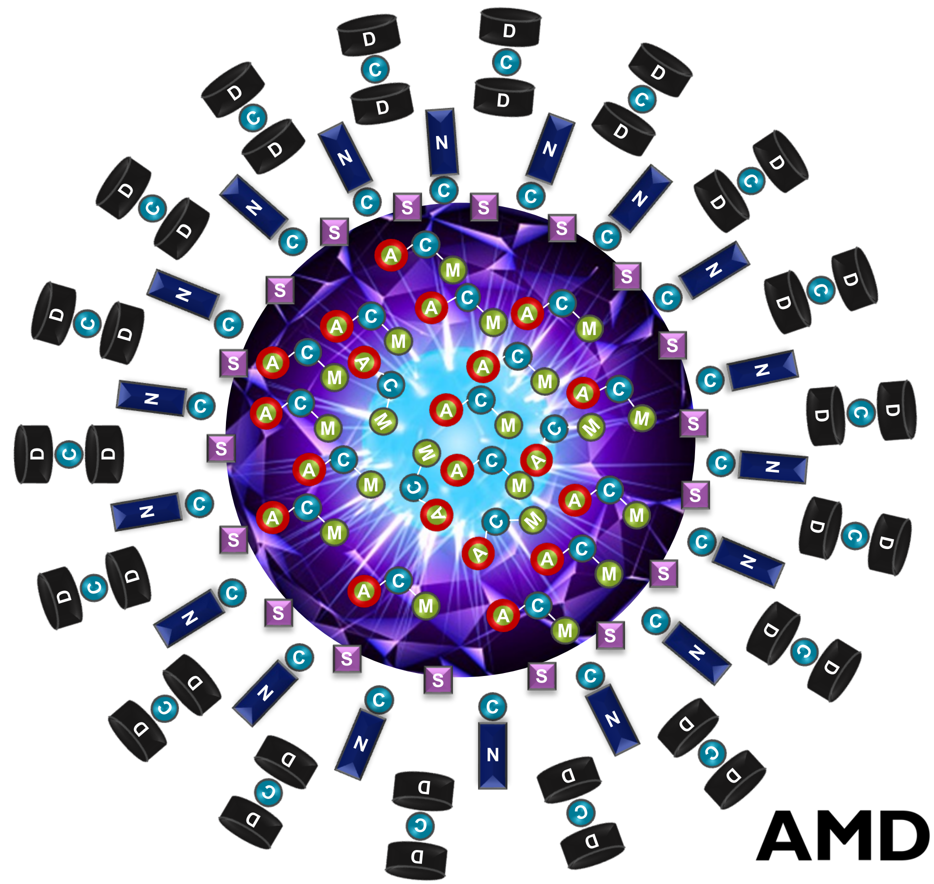
Through a competitive Broad Agency Announcement, released by the Army Research Office (ARO), the following AGILE research contracts were awarded:

- **Advanced Micro Devices, Inc.**
- **Georgia Institute of Technology**
- **Indiana University**
- **Intel Federal LLC**
- **Qualcomm Intelligent Solutions, Inc.**
- **The University of Chicago**

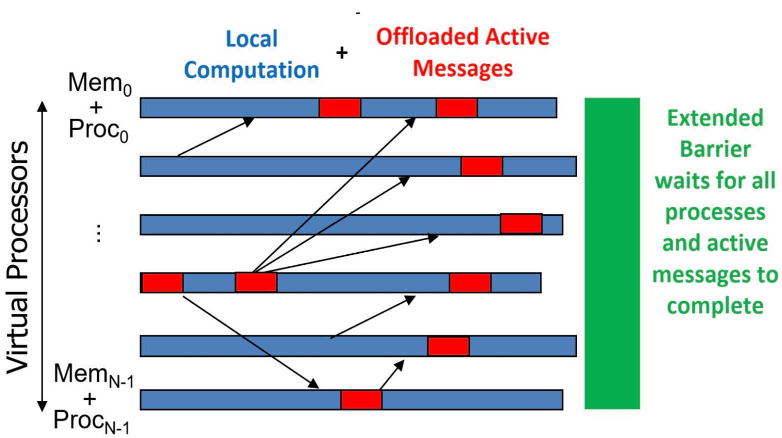
AMD: Compute Everywhere

PANDO: Parallel Architecture for Native Data-Graph Analytics Operations

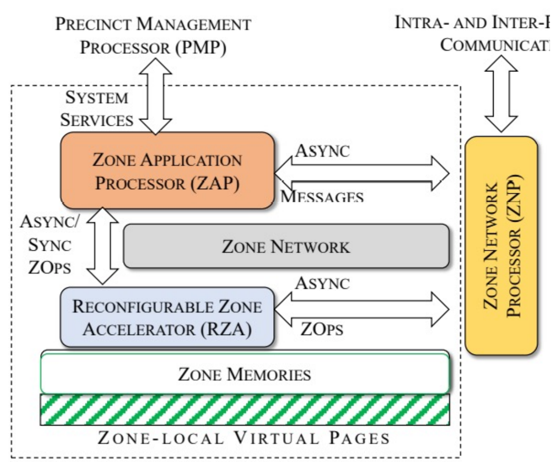
- New computational methods and architectures that:
  - Minimize data movement
  - Bring “smart-ness (compute)” everywhere – memory, storage, network, ingest, security
  - Scale from work-station to cluster



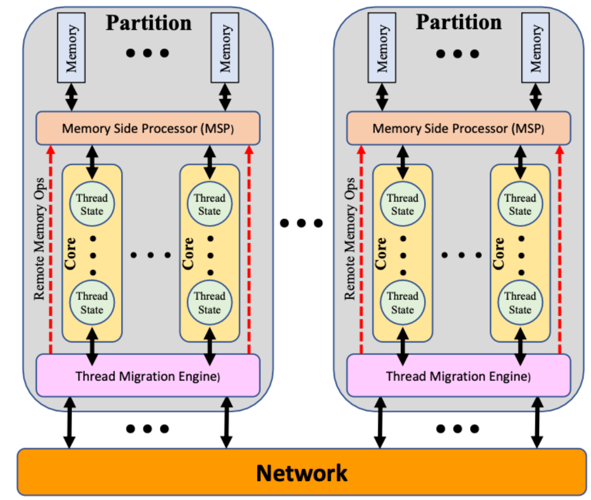
C Compute
S Smart switch
M Memory
A In-memory accel
N SCM
D Disk



Actor Messages

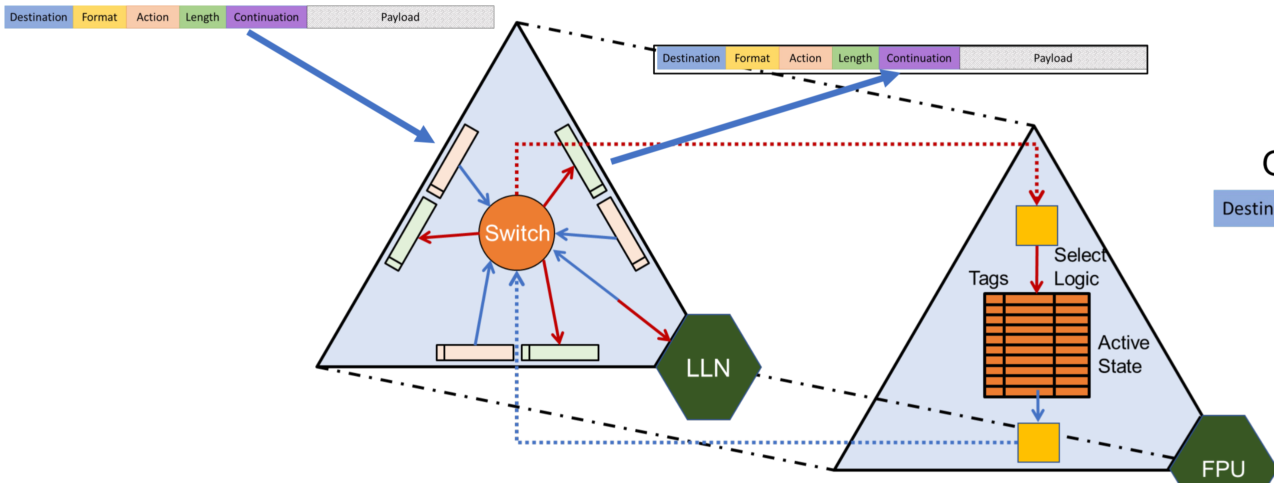


Atomic Operations



Migrating Threads

Indiana University: Parallel Flow Control

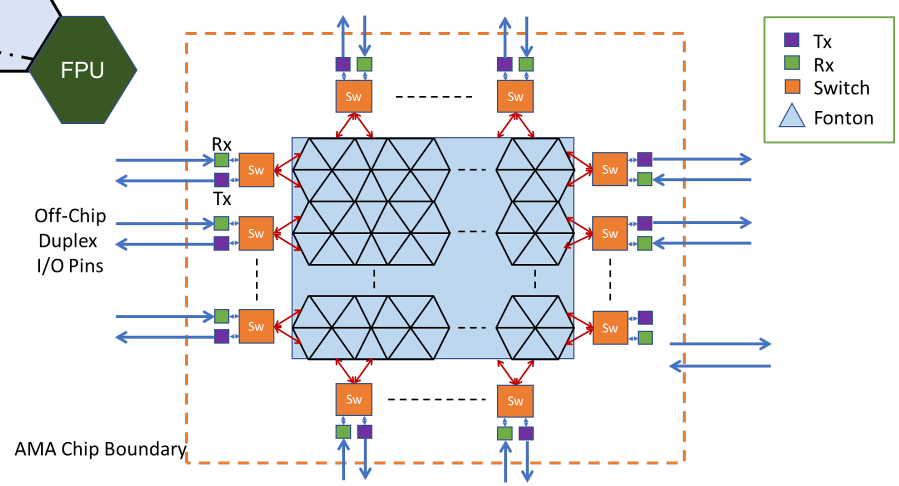


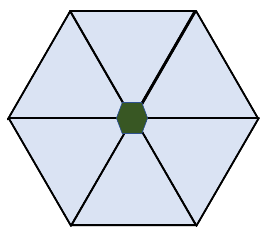
2 Functional Layers in Fonton Design

Operon – Communication/Compute Packet

Destination	Format	Action	Length	Continuation	Payload

AMA – Active Memory Architecture





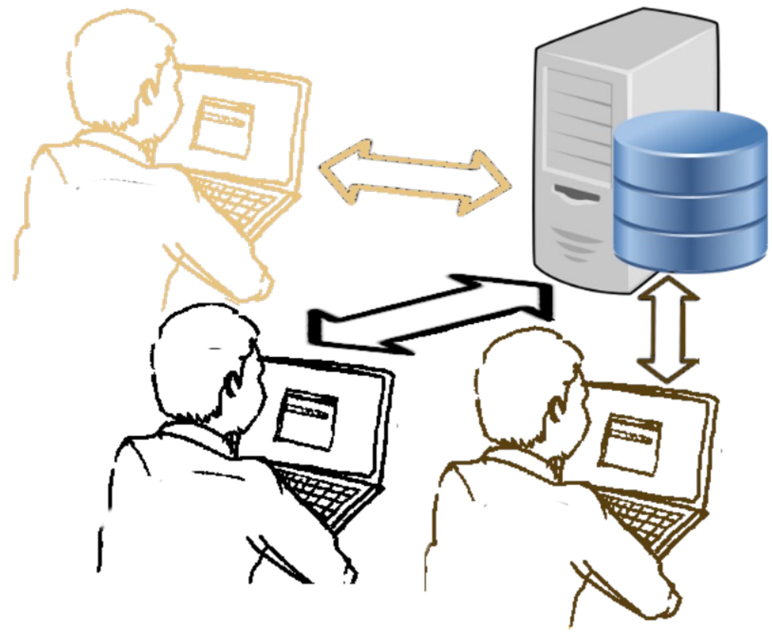
Fonton Cluster

AGILE Security Challenge

Multi-tenancy and multiple applications working cooperatively in the same memory space present security challenges in terms of:

- System security
- Data integrity
- Services compliance

This can be summarized in two main challenges



Challenge 1:

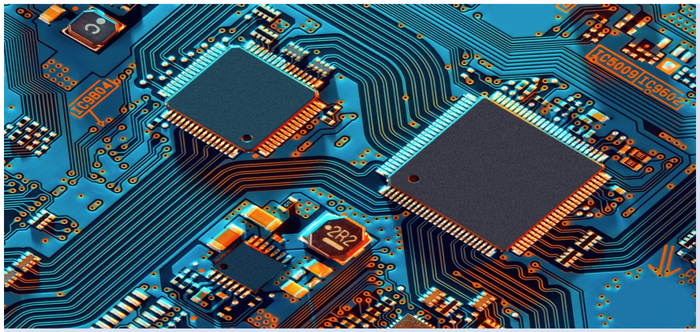
Isolate the edits of different data analytics from one another until those edits are approved and committed

Challenge 2:

Prevent unauthorized analysts from seeing data or traversing paths that they are not authorized to access

T&E Evaluation Efforts

Design V&V



Validate Performers' Hardware & Application Test Plans

Provide FireSim

Evaluate Performers' models/designs for correctness & completeness

Validate the results generated using SST

Lawrence Berkeley National Laboratory (LBNL)

ModSim

Validate Performers' models in the SST (Toolkit)

Using SST, provides performance estimates and correctness of the Performers' models/designs

SST = Structural Simulation Toolkit (Modeling and Simulation Environment)

Sandia National Laboratory (SNL)

Application Codes

Develop AGILE Workflows and kernels

Baseline performance

Validate changes to the Performers' versions of the AGILE Workflows, kernels and benchmarks (optimized for their systems)

Pacific Northwest National Laboratory (PNNL)

SST & FireSim Evaluation

- Demonstrating that the designs can achieve or exceed target metrics
- Modeling a system level design when executing AGILE Applications, with realistic data sets
- Runtime system –
  - Required by the design – evaluated using SST & FireSim
  - Developed on conventional platform – evaluated on baseline platform
- Must complete modeling and simulation in a reasonable amount of time
- Verifying the design
- Evaluating security

SST – Scaling Challenges

- Scaling simulation environments is an understood problem
  - SST provides a parallel simulation environment providing opportunity for scaling to large models when compared to traditional, sequential models
- SST – Like all parallel discrete event simulator (**PDES**) , is challenging to scale
  - Frequent asynchronous communication
  - Partitioning and load balance issues
  - Small message sizes
- SST modeling effort for AGILE will result in extreme scale simulation runs
- AGILE Simulation effort has two parts to its complexity:
  - Model Scale – need to predict entire system performance
  - Workload Complexity – long running applications, large datasets

Scaling Challenges – System Complexity

- SST models of AGILE systems will require potentially millions of components
  - AGILE Systems are heterogenous
  - Simulation of the *entire* system requires simulation of *all* component types – adding to complexity
- Impossible to model complete system except smaller scale problems and expect reasonable runtimes
- Multi-scale modeling is one solution
  - Strategically replace individual, complex elements with statistical models
  - Straightforward at node level, much more difficult at system level modeling scale

Scaling Challenges – Workload Complexity

- AGILE Workflows are large, long running, multi-modal applications
  - Complex applications with multiple phases, intricate communication
- Architectural simulation environments typically support running “bare metal”
  - No operating system services!
  - Need to create printf from scratch
  - Contrast with software development environments, such as QEMU
- Results in a dual porting effort!
  - Port once for AGILE architecture, Port a second time to run in a simulation environment
- Debugging simulation workflows its own effort
  - Functional bugs – is the answer correct?
  - Performance bugs – do I get the correct performance projections?
  - Requires extensive list of tests

Summary

- **HW / SW development process that utilizes co-design**
- **SST / FireSim are the best tools for the process, but are challenging**
- **The chasm between the *DEMANDS* of today's escalating data-intensive problems and the *CAPABILITIES* of yesterday's computing systems is unbridgeable**
- **AGILE is the first program in decades to offer a clean slate for completely re-thinking system-level computing architectures**
- **AGILE systems will enable new areas of data analytic applications that turn chaos into order**

Backup

William Harrod | Program Manager | May 15, 2023

Intelligence Advanced Research Projects Activity

IARPA

Creating Advantage through Research and Technology

Data Analytics Problem

- Data analytic problems are represented by graphs, For example, FaceBook social media graphs.
- Graphs have entities represented by vertices (V) with types and properties, and relationships are represented by edges (E) with types and properties.
- The graphs are typically sparse, that is $|E| <<< |V|^2$

Graphs	Vertices	Edges
Social network	1 Billion	100 Billion
Internet	50 Billion	1 Trillion
Brain	100 Billion	100 Trillion

Technical Report NSA-RD-2013-056002v1,
U.S. National Security Agency

Extracting Actionable Knowledge Methods				
Graph Analytics	Machine Learning	Statistics Methods	Linear Algebra	Data Filtering

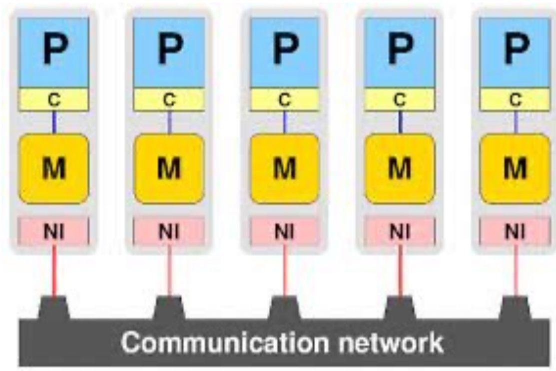
"The variety and volume of data collected (today) … far outpace the abilities of current systems to execute complex analytics … and extract meaningful insights."

– Buono, D., Computer, August 2015

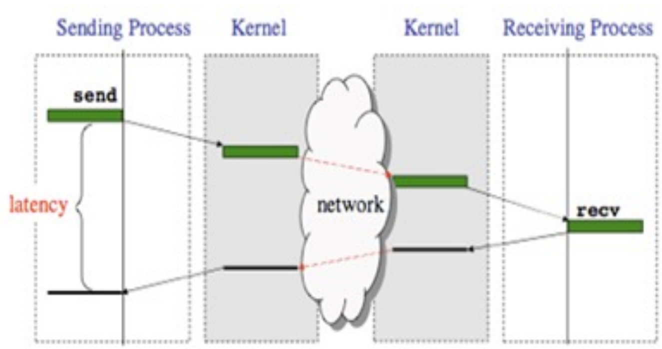
Old Approach: Moving Data to Compute

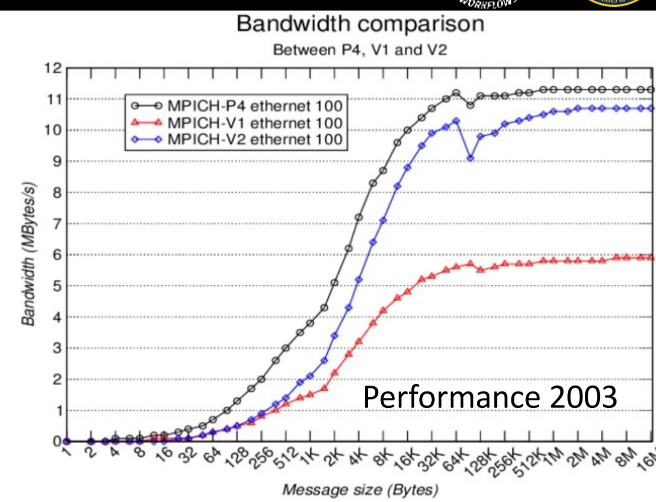
Classic HPC Systems – Bulk Synchronous Processing (BSP)

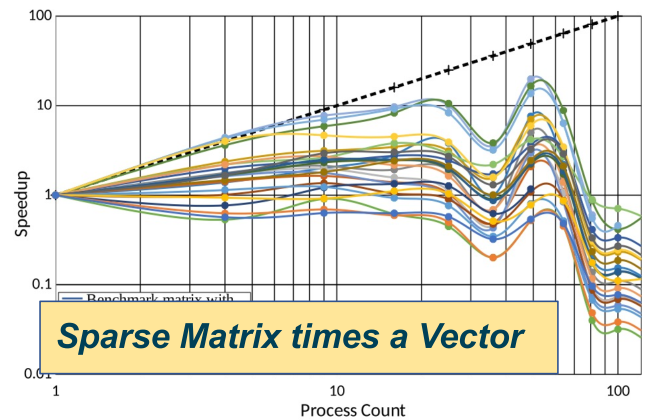
- Today's high-end computers have physically distributed processors and memory spaces.
- This requires functions to move data from one address space to another address space.
- Systems are optimized for moving large blocks of data.
- Highly inefficient for the fine-grained asynchrony and data distribution required by large-scale data analytics applications
- The problem is getting worse with time.



Classic HPC Architecture







Runtime System

- Runtime is an abstraction of computing system software structure and operation for a specific system model
- Provides a conceptual framework for the co-design of technology: architecture, programming interfaces, and system software
- Attributes:
  - Extreme parallelism
  - Asynchrony
  - Self-discovered parallelism
  - Adaptive management
  - Global name space