Data and Model Convergence: a case for Software Defined Architectures

ARM Research Summit
Austin, TX, USA

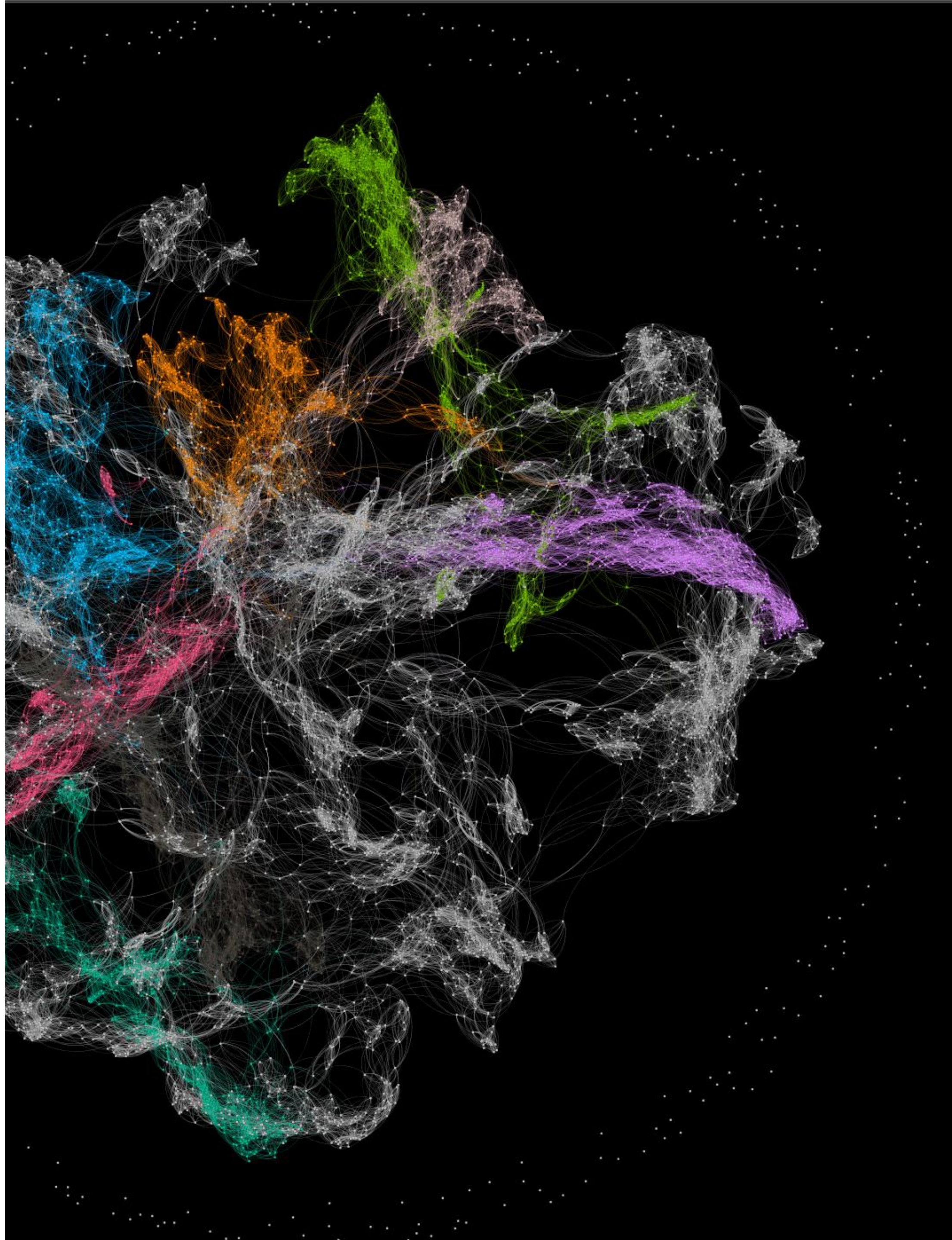
Antonino Tumeo, Marco Minutoli,

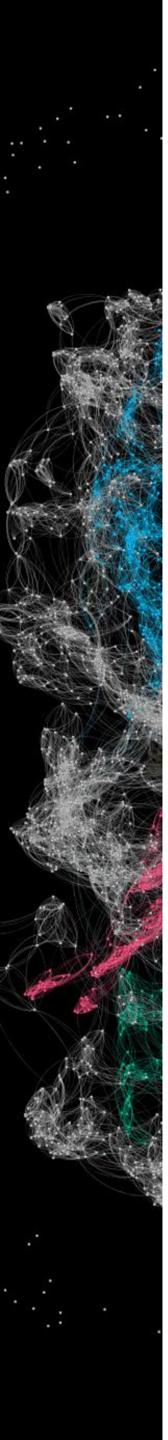
Vito Giovanni Castellana

(DMC Initiative Overview slides provided by Jim Ang,

PCSD Chief Scientist for Computing)

February 27, 2019



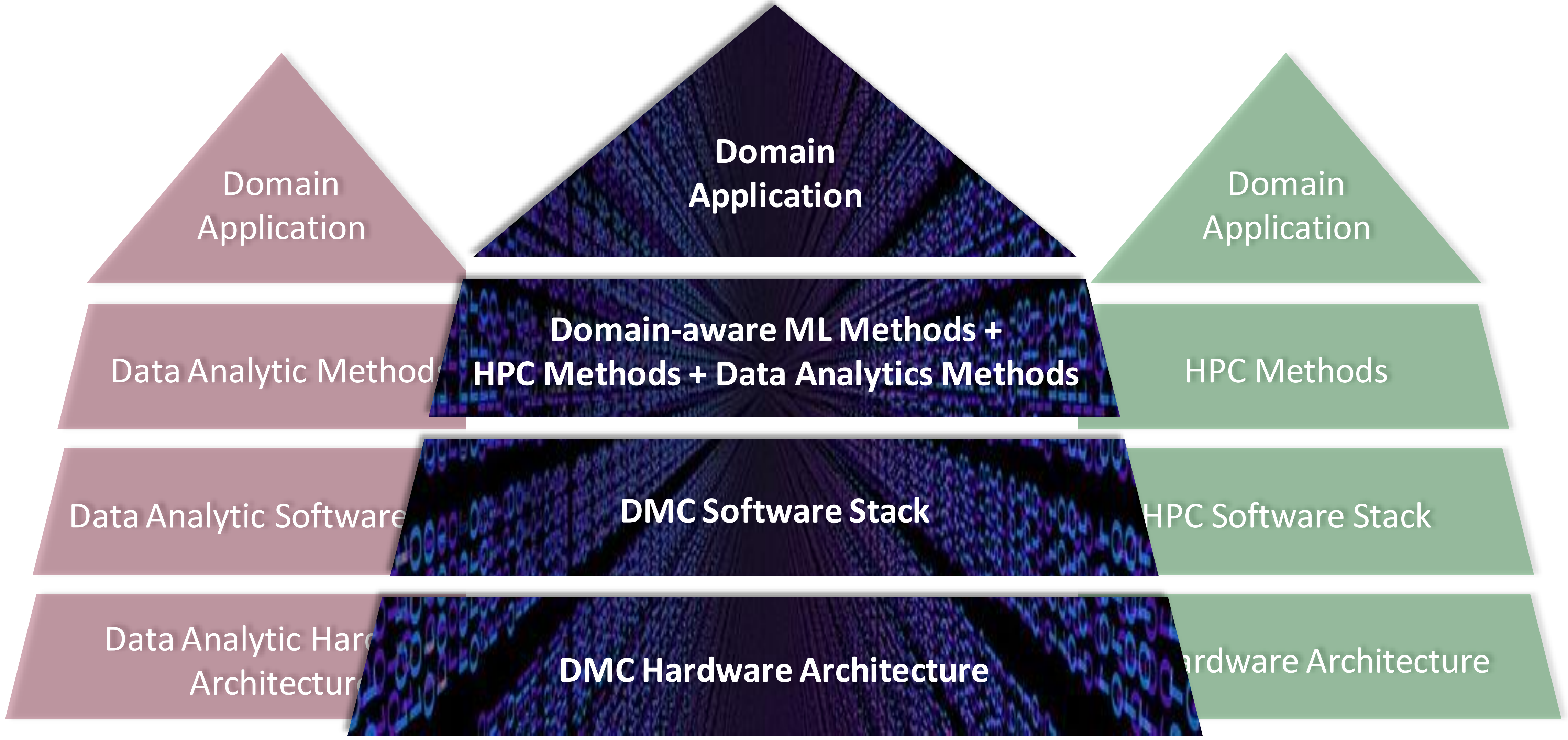


Definition of DMC

- Many contemporary science and engineering problems facing PNNL and DOE—such as grid optimization or materials discovery—are best solved by integration of:
  - High performance computing
  - Large-scale data analytics
  - Machine learning methods
- We call this integration "Data-Model Convergence (DMC)"
- Supports directly PNNL Lab Objective:

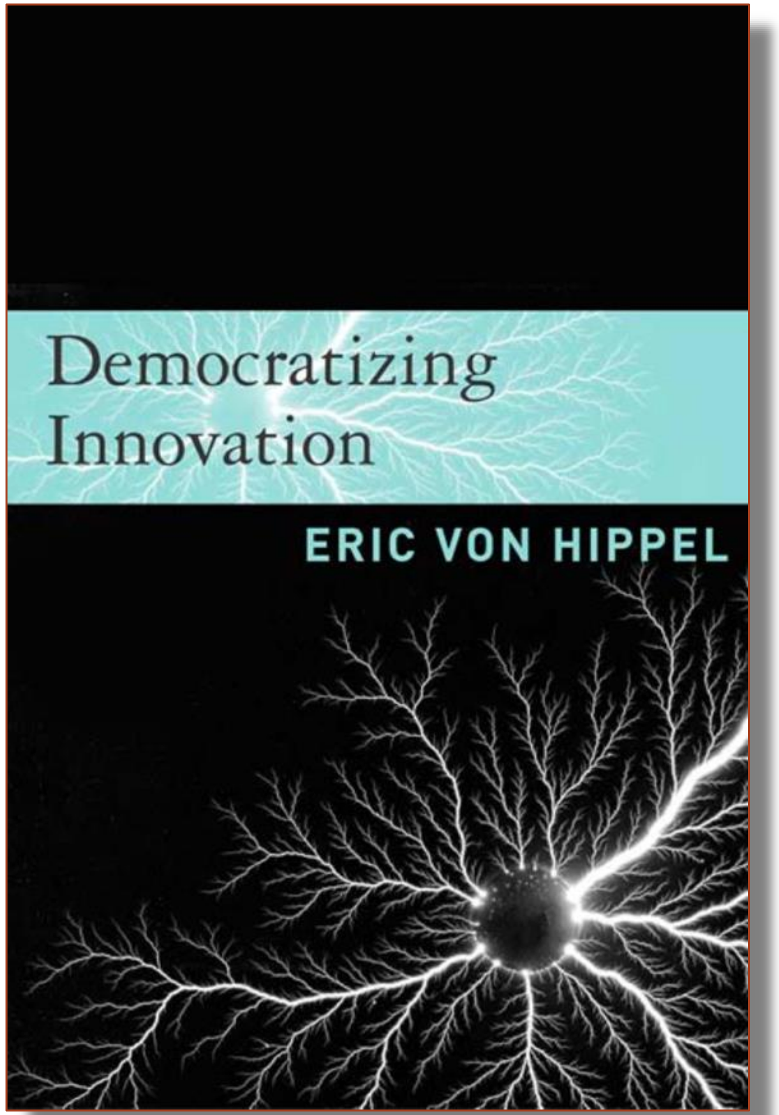
Accelerating Scientific Discovery through
Extreme-Scale Data Analytics and Simulation

DMC Approach for Converged Computing Paradigms



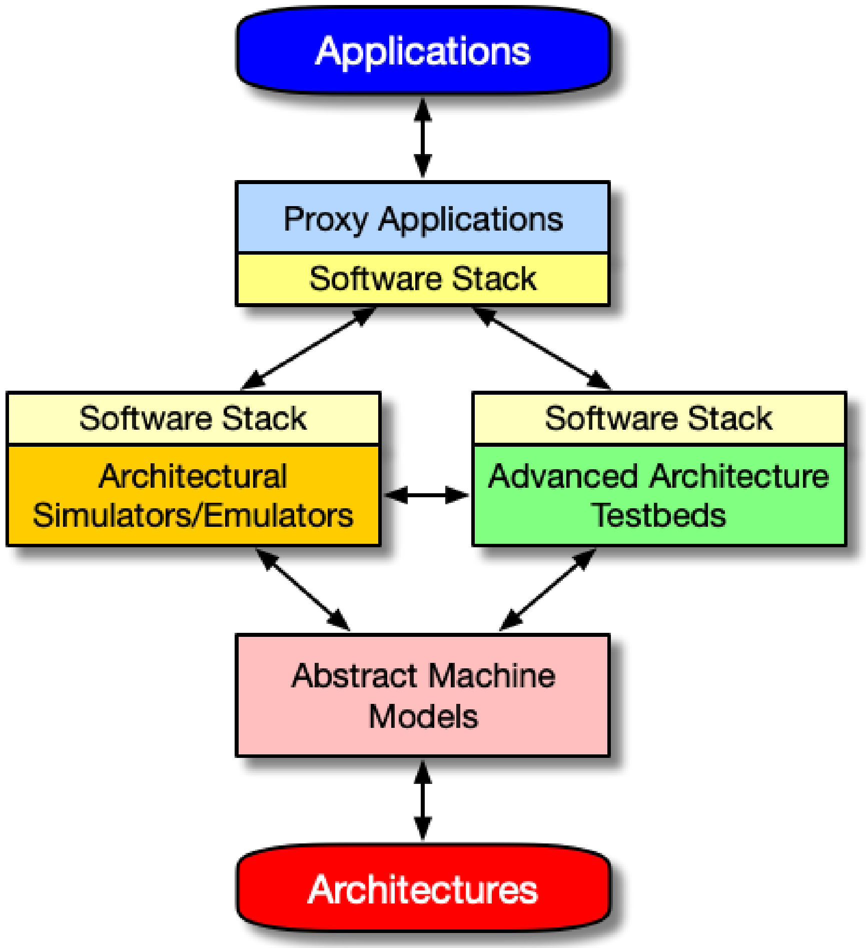
Approach: The Lead User

- Lead User is a key concept from *Democratizing Innovation* by Eric Von Hippel
- Lead Users are "a source of novel product concepts"
- With Open Source, innovation does not only come from manufacturers
- "Users are firms or individual consumers that expect to benefit from using a product or service. In contrast, manufacturers expect to benefit from selling a product or service."



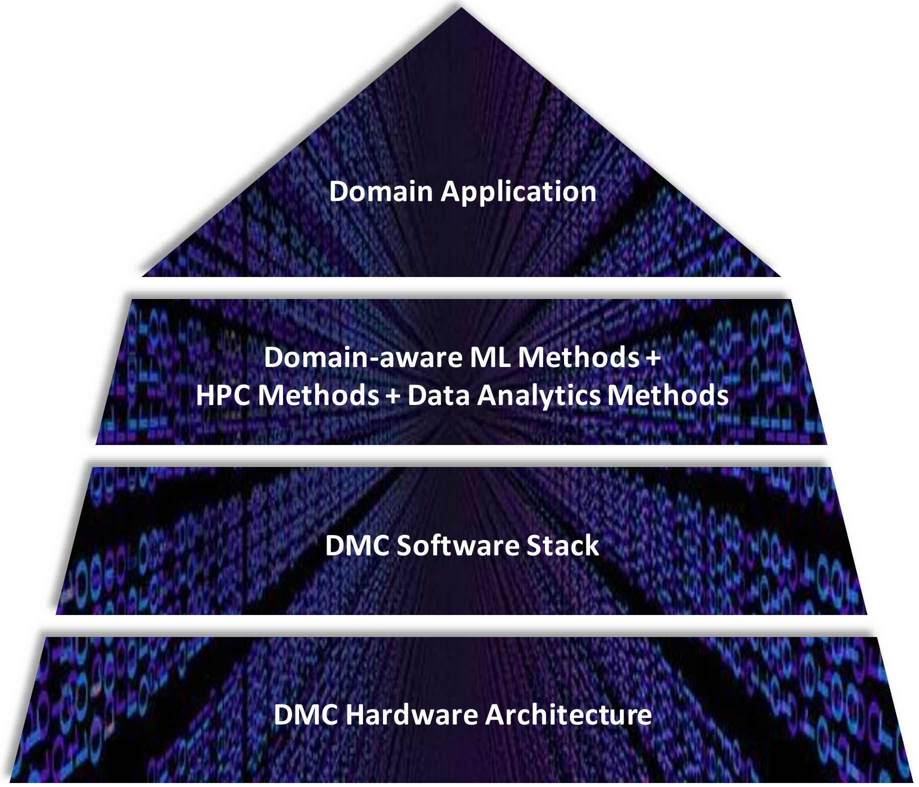
Approach: Holistic Co-design

- We can be Lead Users
  - Our goal is to co-design and develop DMC computer architectures and software stack for DMC workloads
- Purpose-designed Hardware Specialization
  - Processor and Memory manufacturers don't have sufficient insight into our applications to know the most effective architectural innovations
  - As Lead Users, the DMC Initiative can establish multi-disciplinary collaborations to develop advanced design concepts



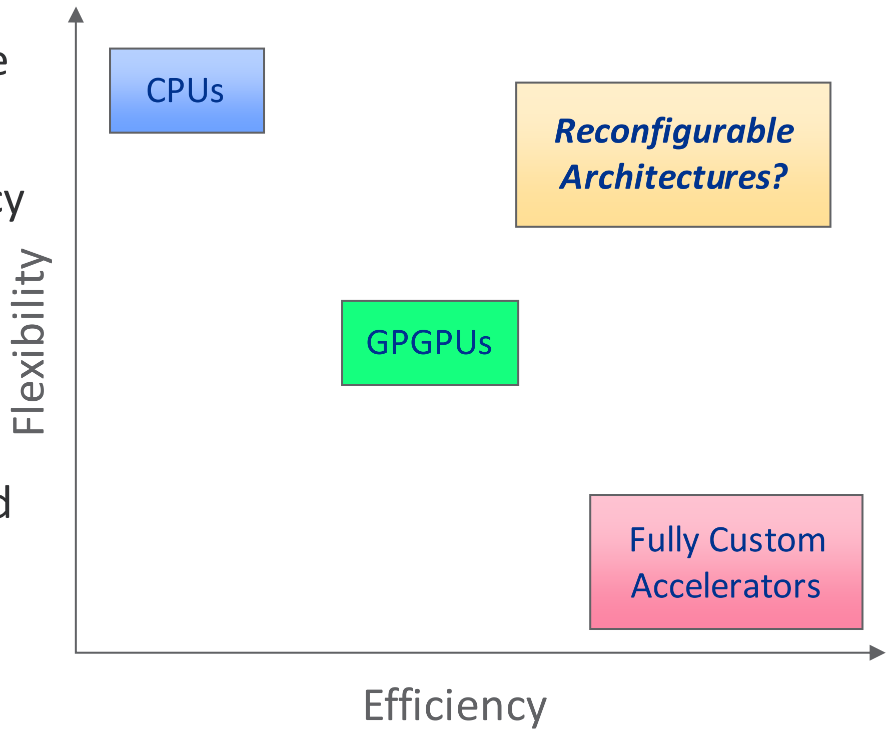
Approach: DMC Initiative Thrusts

- Domain Applications
  - Define DMC challenge problems
  - Deliver DMC capabilities to domain scientists
- Domain-aware Machine Learning
  - Address gap between domain models and scalable approaches to scientific ML
  - Deliver new theory and tools
- Software Stack
  - Address need for single programming framework for developing combined data, HPC, and ML applications
  - Deliver scalable software framework to support next-generation applications on heterogeneous hardware
- Hardware Architecture
  - Address need for novel architectures to improve system efficiency and performance in DMC applications
  - Deliver tools, methods, and prototype designs for next-generation architectures


Domain Application
Domain-aware ML Methods + HPC Methods + Data Analytics Methods
DMC Software Stack
DMC Hardware Architecture

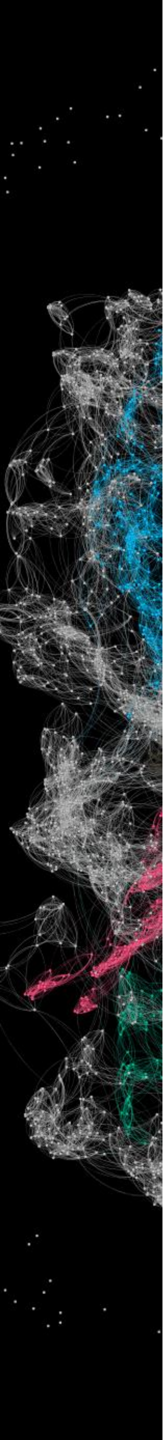
Reconfigurable Architectures For DMC?

- DMC workflows are a complex mix of methods, with different behaviors across phases
  - Scientific simulation, graph algorithms, and ML methods: sparse vs dense data structures
  - ***Memory-*** vs. ***computation-bound***, high vs. low precision, latency vs throughput, ”regular” vs. “irregular”
- To reach higher efficiency, architectures will either:
  - Integrate a sea of ***application specific accelerators***
  - Integrate a set of functional units and memories interconnected with ***reconfigurable*** on-chip networks
- Reconfigurable architectures promise ***efficiency*** through *adaptation* without trading off ***flexibility***
  - Recent uptake of Field Programmable Gate Arrays (data centers, machine learning, even HPC)



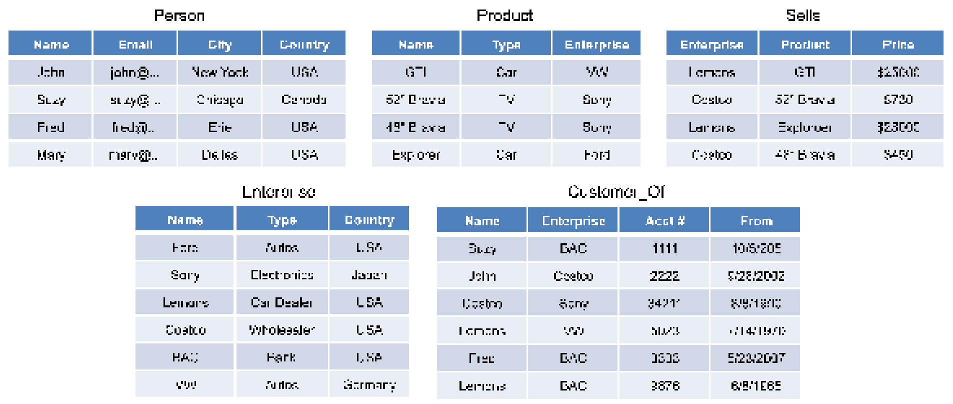
Gaps in the software/hardware stack

- Abstractions for domain experts and applications developers
  - Reconfigurable hardware is spatially programmable
  - Do not want to deal with the architectural details
- Hardware is not fixed anymore
  - The software stack needs to jointly explore software and hardware
- Dynamic reconfigurable architecture
  - Joint exploration of hardware (components, parameters) and software
  - Reconfigurable components, interfaces for reconfiguration, reconfiguration granularity
  - Scheduling and mapping of hardware and software work units



SO(DA)^2: Software Defined Architectures for Data Analytics

- Design a toolchain to make ***reconfigurable architectures practical*** for ***High Performance Data Analytics applications***
- Investigate methodologies for:
  - ***Data-aware High-level Abstractions and Tools***. Linear algebra, deep learning, relational data, and graph-based data.
  - ***Parallel Data Driven Optimizations.*** Data statistics and data analysis to drive optimizations.
  - ***Multi-objective offline and online Design Space Exploration.*** Exploration, identification, and synthesis of pareto-optimal configurations (and code) though multi-objective algorithms.
  - ***Runtime Partial Dynamic Reconfiguration and Introspection.***
  - ***Architectural Exploration.*** Prototyping and modeling on fine-grained reconfigurable devices (FPGAs). Evaluate impacts on coarse-grained, non-von Neumann designs.



Person

Name	Email	City	Country
John	john@...	New York	USA
Suzy	suzy@...	Chicago	Canada
Fred	fred@...	Erie	USA
Mary	mary@...	Dallas	USA

Product

Name	Type	Enterprise
GTI	Car	VW
52" Bravia	TV	Sony
46" Bravia	TV	Sony
Explorer	Car	Ford

Sells

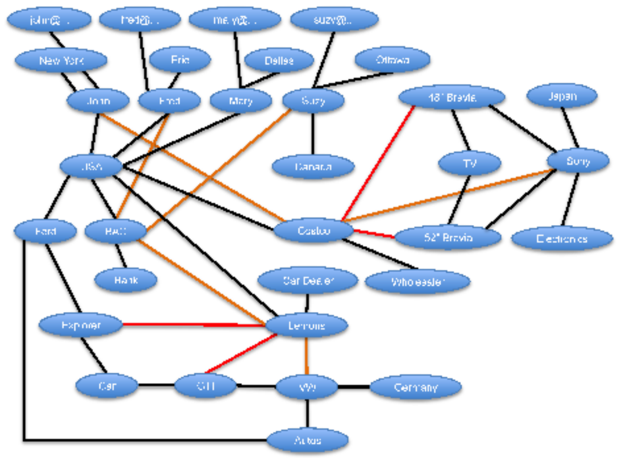
Enterprise	Product	Price
Lemons	GTI	$25000
Costco	52" Bravia	$730
Lemons	Explorer	$23000
Costco	46" Bravia	$450

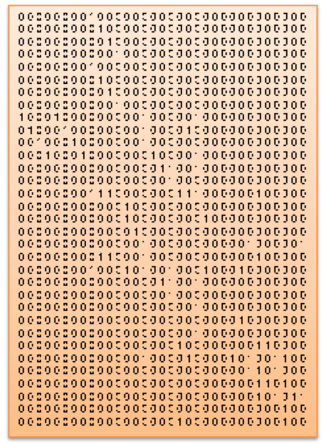
Enterprise

Name	Type	Country
Ford	Autos	USA
Sony	Electronics	Japan
Lemons	Car Dealer	USA
Costco	Wholesaler	USA
BAC	Bank	USA
VW	Autos	Germany

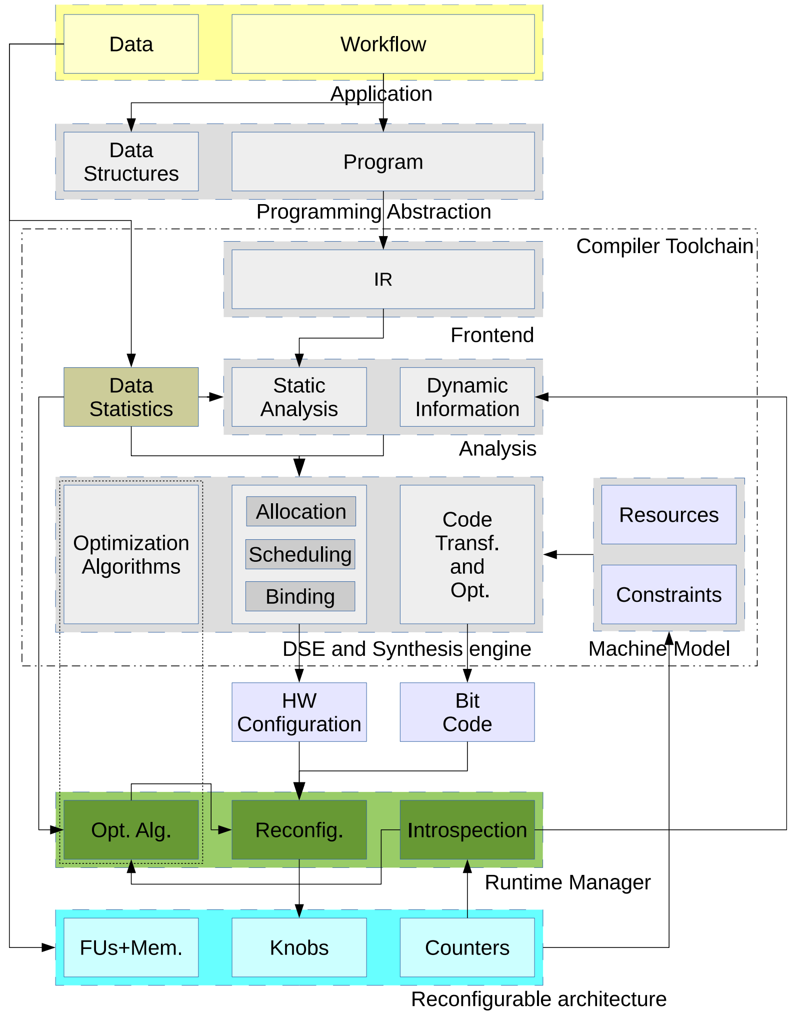
Customer_Of

Name	Enterprise	Acct #	From
Suzy	BAC	1111	10/5/205
John	Costco	2222	5/28/2002
Costco	Sony	34211	8/8/1980
Lemons	VW	5023	7/14/1970
Fred	BAC	3333	5/23/2007
Lemons	BAC	3876	6/8/1985



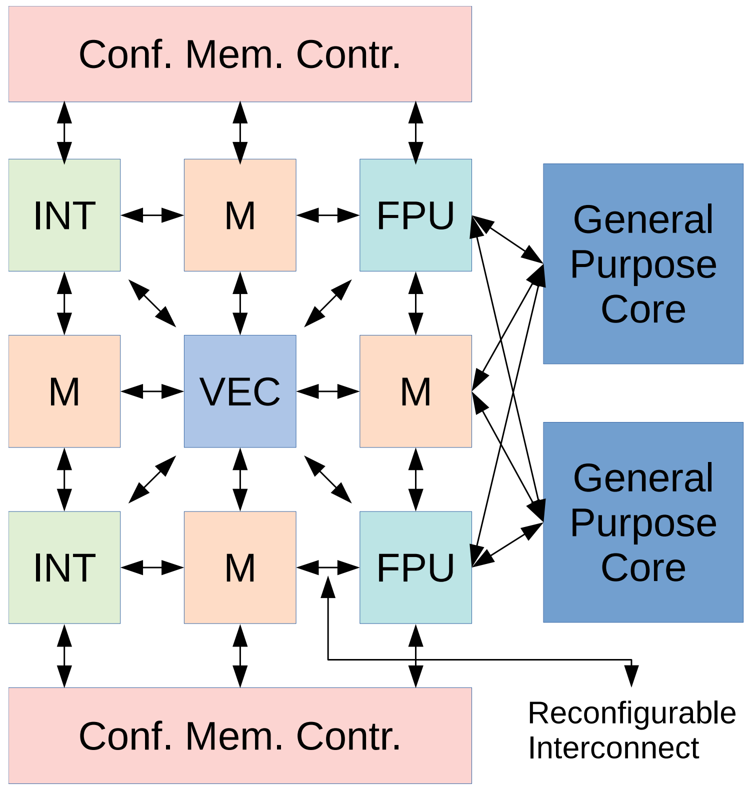


SO(DA)^2 Toolchain and Features



- ***High-Level abstraction and data-aware analysis***
  - Appropriate Domain Specific Language for DMC and reconfigurable architectures
  - Static and dynamic analysis
  - Generates a hierarchical task-based representation
- ***Design Space Exploration and Synthesis (DSES) engine***
  - Pluggable libraries of optimization algorithms and machine models
  - Joint hardware-software optimizations
  - Multi-objective (time, power, area, reuse) optimization algorithms, including bio-inspired heuristics (Ant Colony Optimization, Genetic Algorithms)
  - Map tasks to resources, identifies configurations
- ***Runtime Manager***
  - Schedules and maps configurations and compiled codes, executes reconfiguration and fine-grained adjustments
  - Monitors execution, provides information back to DSES

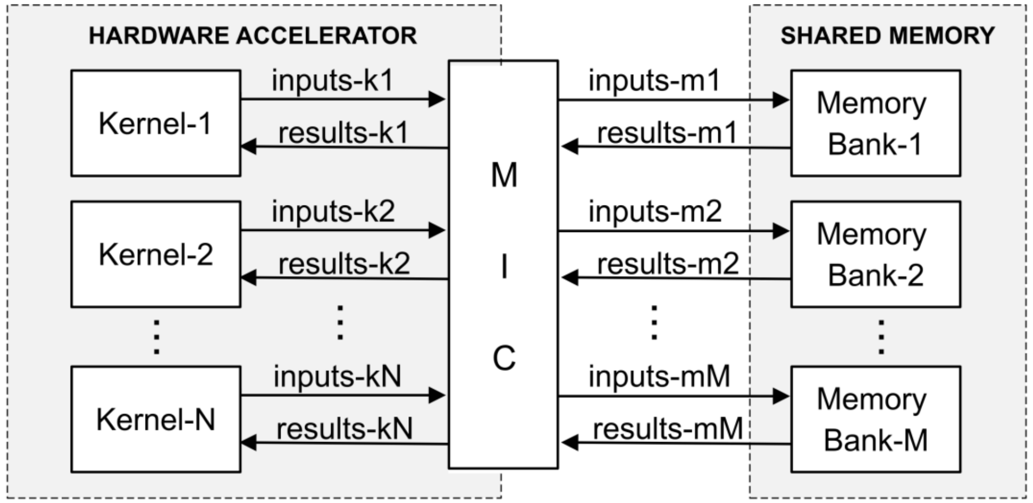
SO(DA)^2 Toolchain and Features (2)

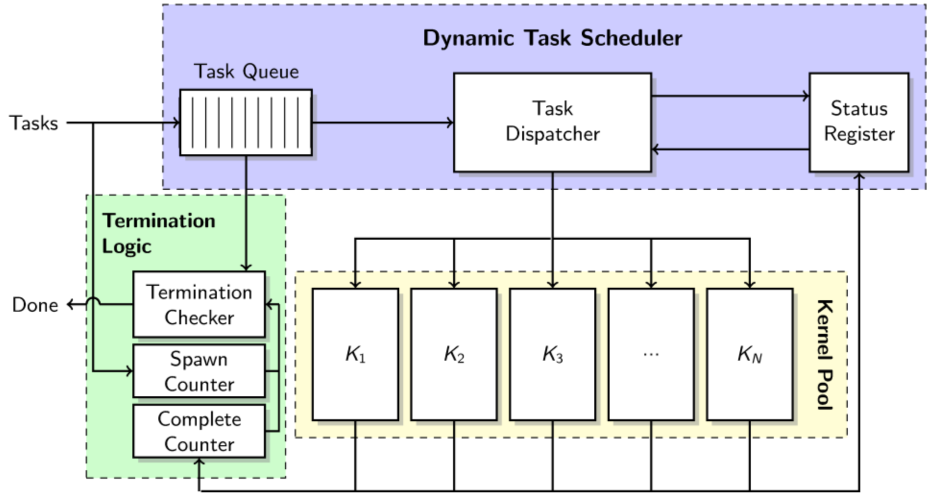


- ***Reconfigurable Architectures.***
  - Forward looking target: Coarse-Grained (quickly) Reconfigurable Arrays (CGRAs)
  - Application analysis and modeling to co-design the target architecture components and parameters
  - Validation of the approaches on modern Field Programmable Gate Arrays (FPGAs)
  - Leveraging approaches and modules previously implemented in an open-source high-level synthesis toolchain

Foundations and Preliminary Results

- Leveraging state-of-the art approaches and modules implemented in an open-source High-Level Synthesis Toolchain
- Extending methods to support memory-intensive, sparse kernels
  - Graph walks, tensor operations

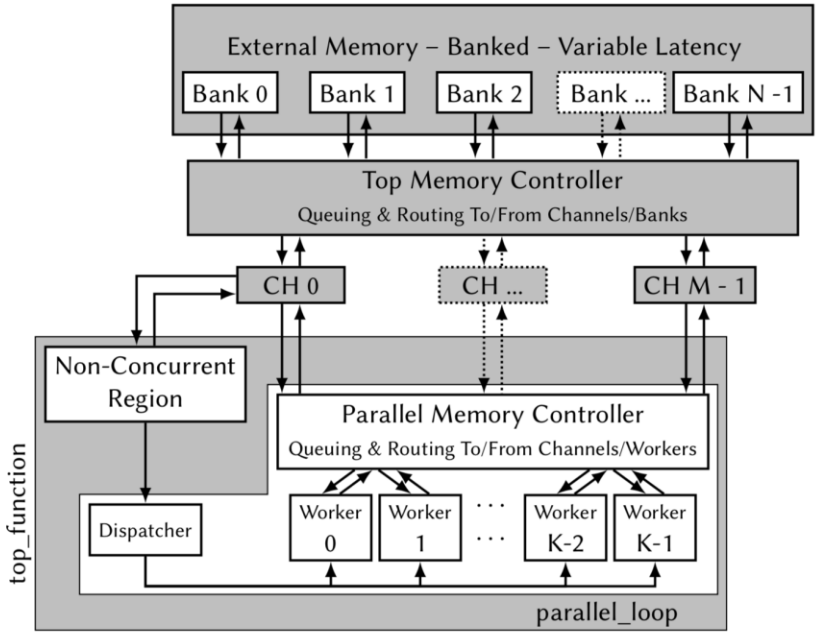




Bambu HLS

https://panda.dei.polimi.it

Synthesis Example: extensions to Multithreading

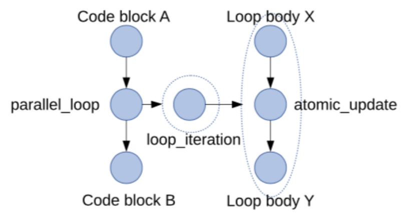


```
void top_function(...) {
  {...} // code block A
  #pragma omp parallel for
  for (size_t i = 0; i < N; ++i) {
    {...} // loop body X
    #pragma omp atomic
    update_results(...);
    {...} // loop body Y
  }
  {...} // code block B
}
```

(a) Example of OpenMP application to be synthesized.

```
void atomic_update(...) {
  update_results(...);
}
void loop_iteration(size_t i, ...) {
  {...} // loop body X
  atomic_update(...);
  {...} // loop body Y
}
void parallel_loop(...) {
  for (size_t i = 0; i < N; ++i)
    loop_iteration(i, ...);
}
void top_function(...) {
   {...} // code block A
  parallel_loop();
   {...} // code block B
}
```

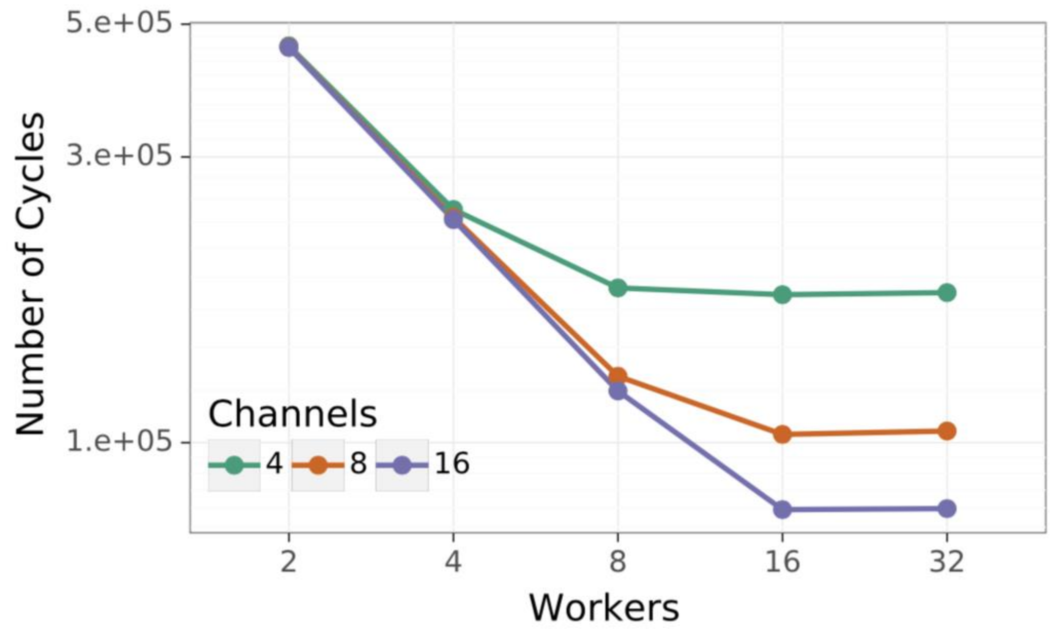
(b) Example of OpenMP application to be synthesized after HLS transformations.

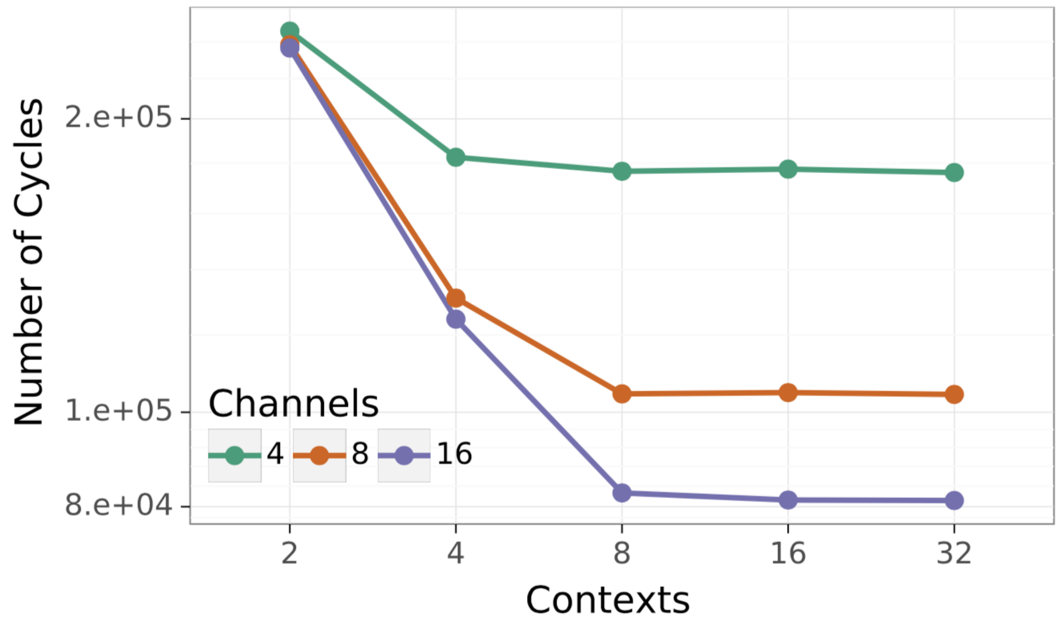


(c) HTG representation of the transformed OpenMP.

- Architectural templates: SO(DA)^2 resources, expose set of parameters (number of accelerators, memory channels, contexts) to explore
- Synthesizes effectively parallel loop iterations with atomic memory operations
- Currently starting from C code annotated with OpenMP
  - Parallel C description of a graph algorithm (for each vertex, for each edge)

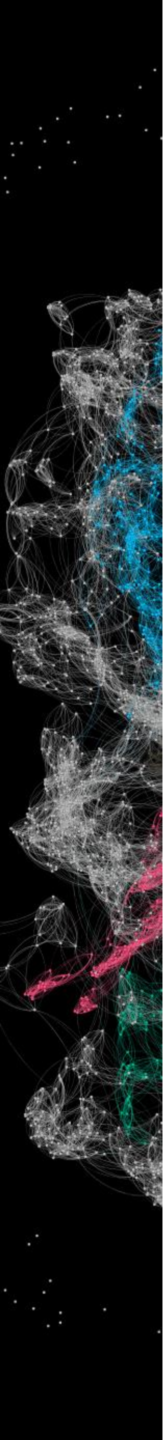
Design Space Exploration (brute force)





- Single context

- 2 workers, multiple contexts



Design Space Exploration Example: ACO

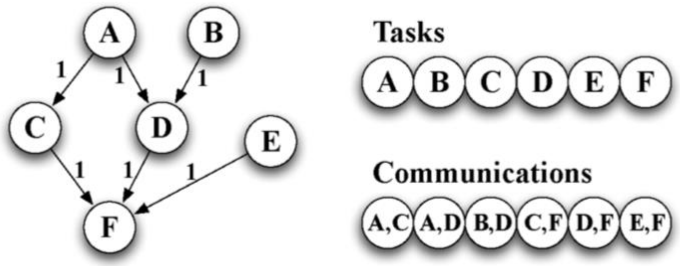
- Ant Colony Optimization: multi-agent optimization heuristic
  - Ants randomly explore different paths to the food.
- At each decision point:

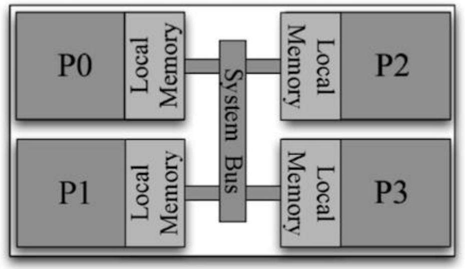
$$p_{x,y} = \frac{[\tau_{x,y}]^{\alpha} * [\eta_{x,y}]^{\beta}}{\sum_{l \in \Omega_x} [\tau_{x,l}]^{\alpha} * [\eta_{x,l}]^{\beta}}$$

- They deposit pheromone proportionally to the length of the path, which suggests other ants to follow the same trail. Pheromone also evaporates with time.

$$\tau_{x,y} = (1 - \rho) * \tau_{x,y} + \epsilon$$

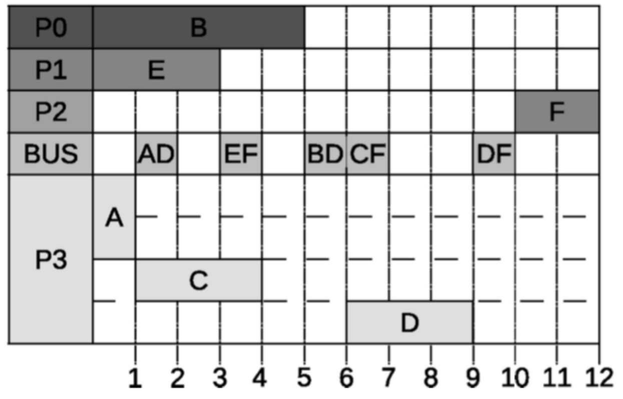
Design Space Exploration Example





Task	$P0$		$P1$		$P2$		$P3$			
	$i_{t,0}$		$i_{t,1}$		$i_{t,2}$		$i_{t,3}$		$i_{t,4}$	
	Time	q_0	Time	q_1	Time	q_2	Time	q_3	Time	q_3
A	8	2	2	3	6	4	1	2	3	1
B	5	3	10	4	6	3	2	2	4	1
C	4	4	6	2	4	1	1	2	3	1
D	8	1	2	2	7	3	3	1	1	3
E	10	3	3	1	8	3	1	3	2	1
F	3	4	7	3	2	2	1	2	2	1

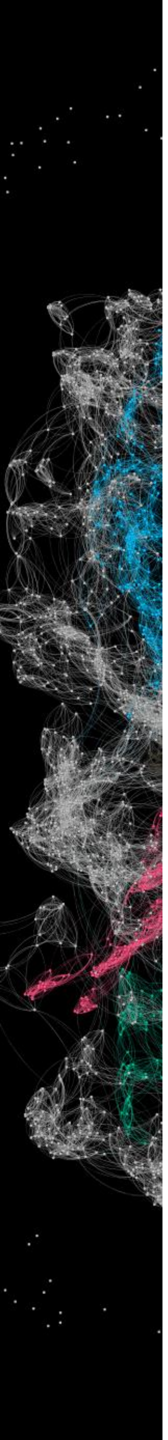
Candidates		Trace	Candidates		Trace
A B E	1)	i(A,3)	F B,D	7)	i(F,2)
C B E	2)	i(C,4)	B,D C,F D,F E,F	8)	i([E,F], 5)
B E	3)	i(B,0)	B,D C,F D,F	9)	i([B,D], 5)
D E	4)	i(E,1)	C,F D,F	10)	i([C,F], 5)
D	5)	i(D,3)	D,F	11)	i([D,F], 5)
F A,D B,D	6)	i([A,D], 5)			



SO(DA)^2 Expected Outcomes

- SO(DA)^2 will:
  - Provide an ***integrated toolchain*** to make novel reconfigurable architectures viable for complex workflows
  - Investigate ***high-level abstractions*** to enable domain scientists to describe applications in a declarative way and enable the toolchain to exploit reconfigurable architectures
  - Provide novel approaches to ***co-optimize hardware and software*** to reach higher-levels of efficiency without trading off programmer productivity
  - Explore ***runtime*** and ***architecture*** designs to make dynamic reconfiguration effective and to allow ***hardware-in-the-loop*** optimization

- Government, industry and academia are interested in reconfigurable architectures as an alternative to highly heterogeneous custom architectures
- SO(DA)^2 will ***address the gaps*** that did not allow previous approaches to succeed



Pacific Northwest
NATIONAL LABORATORY

Thank you