Version 1 of Exascale Abstract Machine Models and Associated Proxy Architectures

Sue Kelly (smkelly@sandia.gov)

for LBNL/SNL CAL team

ModSim Workshop August 13-14, 2014

U.S. DEPARTMENT OF ENERGY | Office of Science

Key Concepts

- **An Abstract Machine Model (AMM) is the schematic for a future computer hardware architecture**
  - Used for communication between hardware vendors and application developers
- **A Proxy Architecture fills that in with speeds and feeds**
  - Used for communication between hardware vendors and ModSim developers
- **Success**
  - A model is developed
  - An input deck instantiates one or more proxy architecture s
  - An application developer is able to do performance analysis on a benchmark, proxy app, or an application(!)
- **Nirvana**
  - Two different models use the same parameters and cross validate their results

- **The Computer Architecture Laboratory (CAL) project will**
  - advance *Exascale Design Space Exploration*
  - for energy efficient and effective processor and memory architecture R&D
  - for DOE's Exascale program
- **A set of Proxy Architectures is a key (evolving) deliverable**
- **Current document concentrates on node models**
- **Distills FastForward confidential project information into sharable abstractions**

Abstract Machine Models and Proxy Architectures for Exascale Computing

Rev 1.1

J.A. Ang[1], R.F. Barrett[1], R.E. Benner[1], D. Burke[2], C. Chan[2], D. Donofrio[2], S.D. Hammond[1], K.S. Hemmert[1], S.M. Kelly[1], H. Le[1], V.J. Leung[1], D.R. Resnick[1], A.F. Rodrigues[1], J. Shalf[2], D. Stark[1], D. Unat[2], N.J. Wright[2]

Sandia National Laboratories, NM[1]
Lawrence Berkeley National Laboratory, CA[2]

May, 16 2014
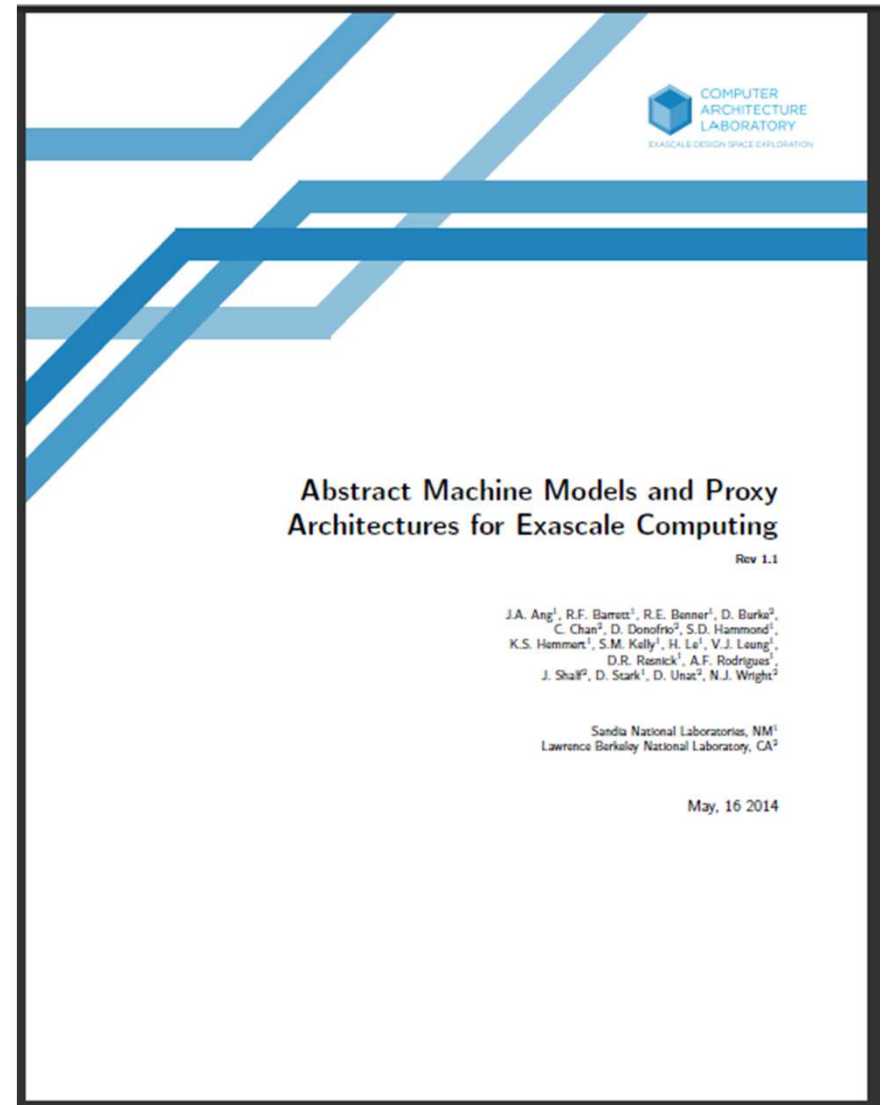
- **An important aspect of co-design is the feedback loop between hardware architectures and application** ~~software~~
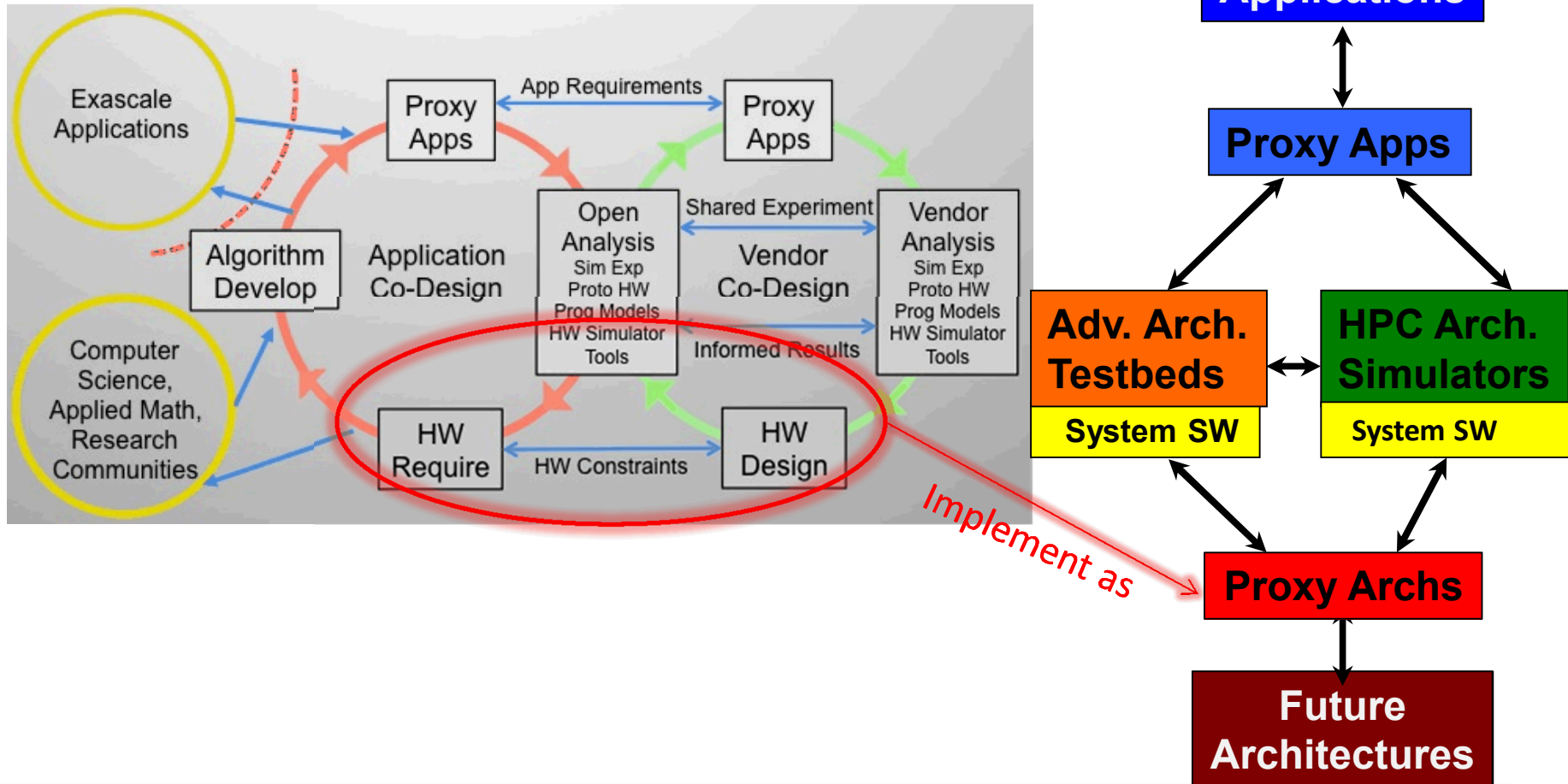
- **Cores - *medium* and *fat* cores**
  - Standard serial processors

- **Accelerators - *thin* cores**
  - Think highly threaded and/or wide (>32) vector

- **Network on chip**
  - Something more sophisticated than a ring

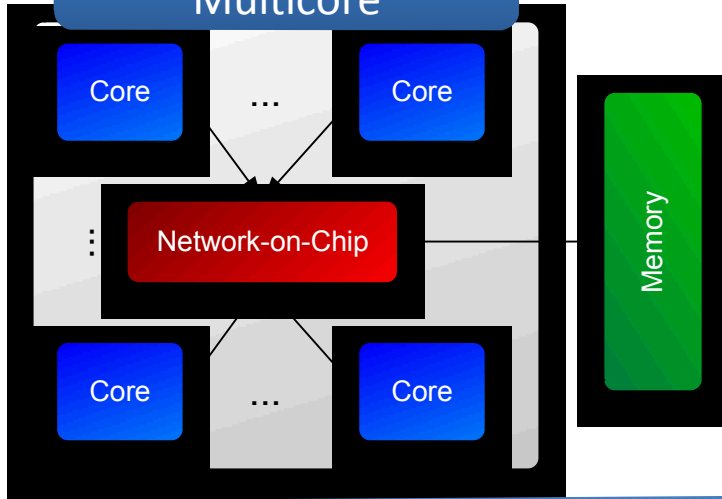- **Memory**
  - Multiple levels - HMC/HBM/WideIO, DRAM and NVRAM
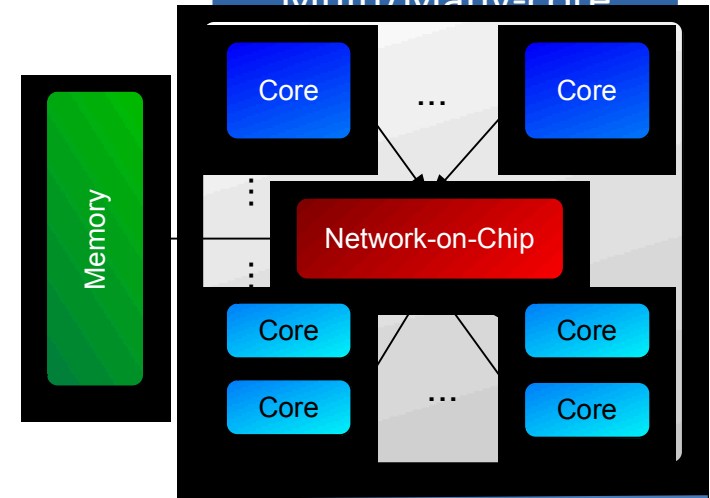
Hardware Reality
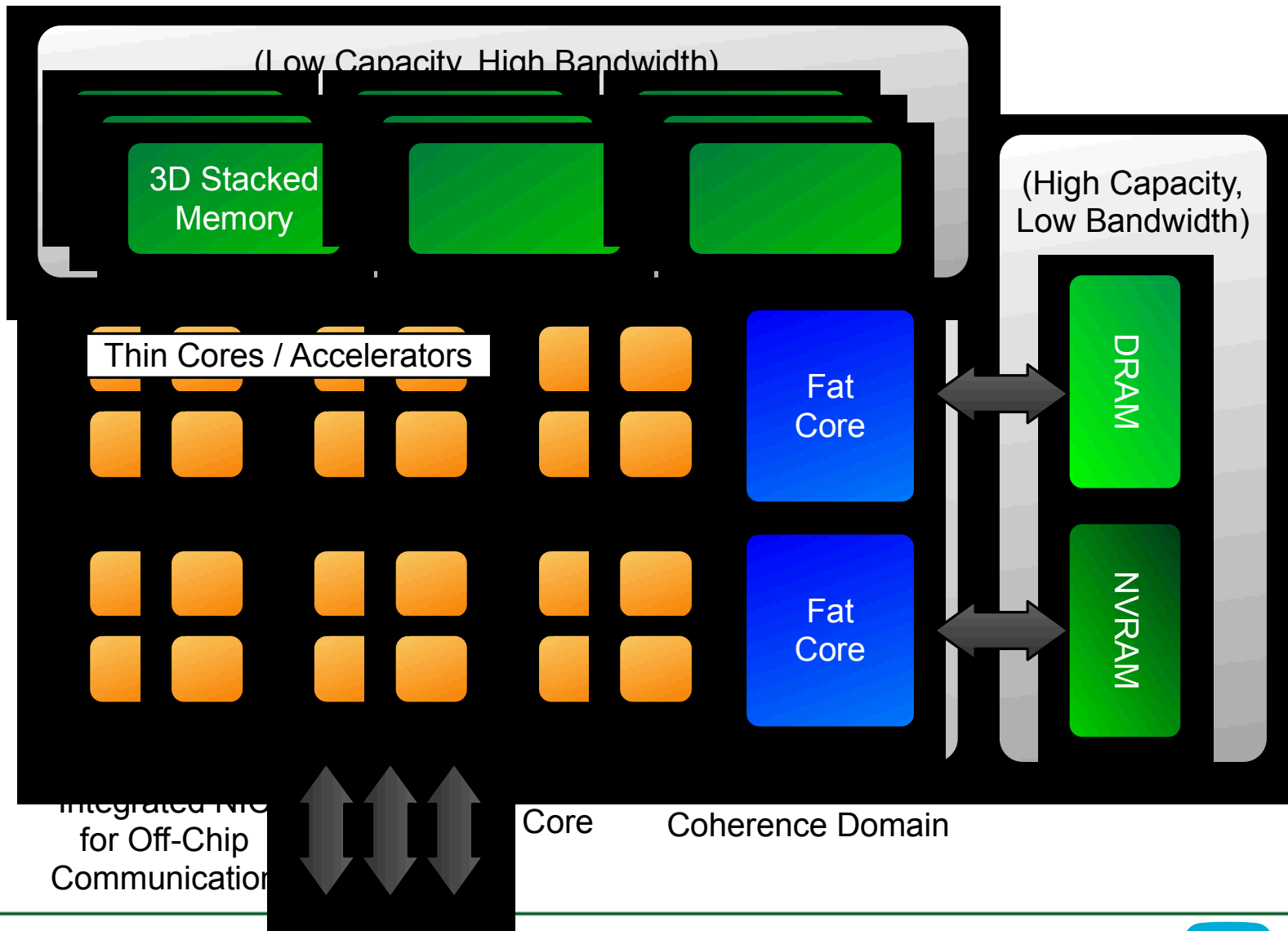
- **Many vendors want to pursue processors for *existing* markets**
  - Bigger than HPC?

- **Probably won't see *all* of these features in a single die**

- **Leads us to think about *plausible* models for the future**
  - Economically possible, performance possible models which are more likely to be delivered

- **Subject to *perfecting* by codesign**
  - We *can* change the future but we may not be able to radically redesign every aspect of the processor
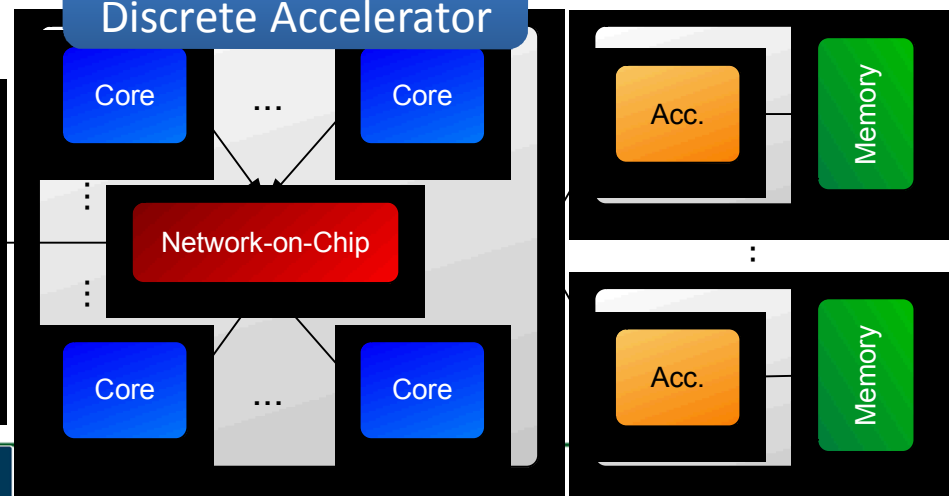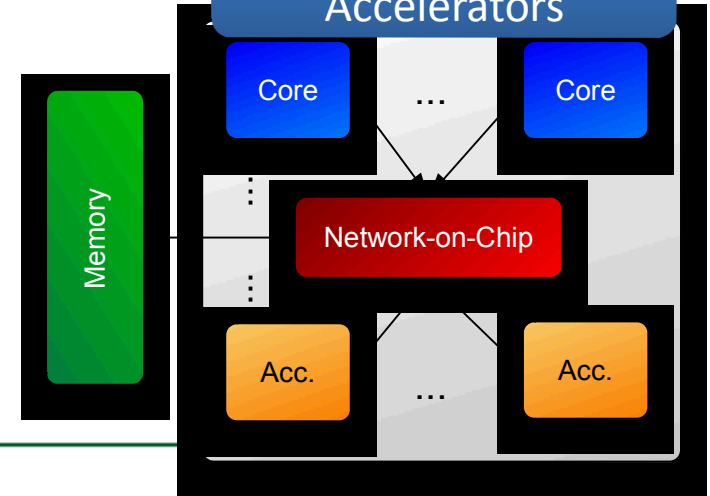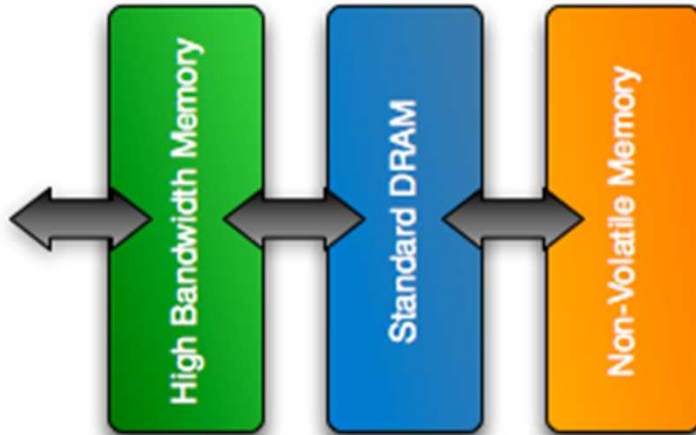
Families of AMMs

Homogeneous Multicore

Core ... Core

Network-on-Chip

Core ... Core

Memory

Heterogeneous Multi/Many-core

Core ... Core

Network-on-Chip

Core ... Core

Core ... Core

Memory

Homogeneous with Discrete Accelerator

Core ... Core

Network-on-Chip

Core ... Core

Memory

Acc. — Memory

Acc. — Memory

Integrated CPU and Accelerators

Core ... Core

Network-on-Chip

Acc. ... Acc.

Memory
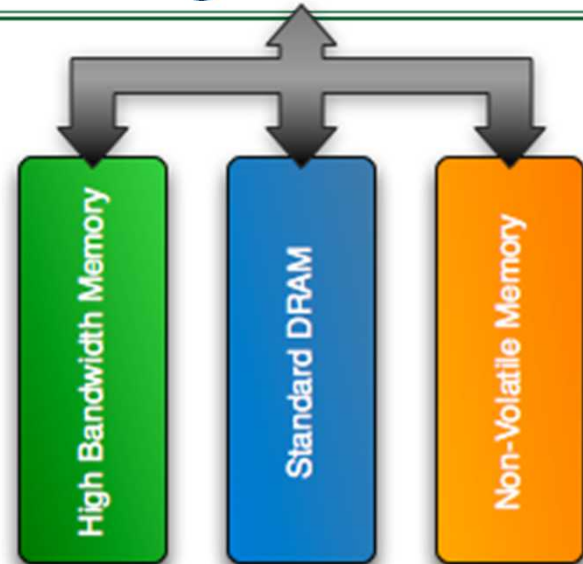
Cache Model — Explicit Allocation Model

- **Levels may be used as a cache**
    - Bad option for graphs, analytic applications, good if you *can* stream
- **Explicit allocation ("partitioned hardware address")**
    - Headache of the programmer but performance will be high
- **Combination of the above?**

NO: this is just a reflection of what is seen developing in industry. Specialization & other architectures possible. See Sandia XGC Project
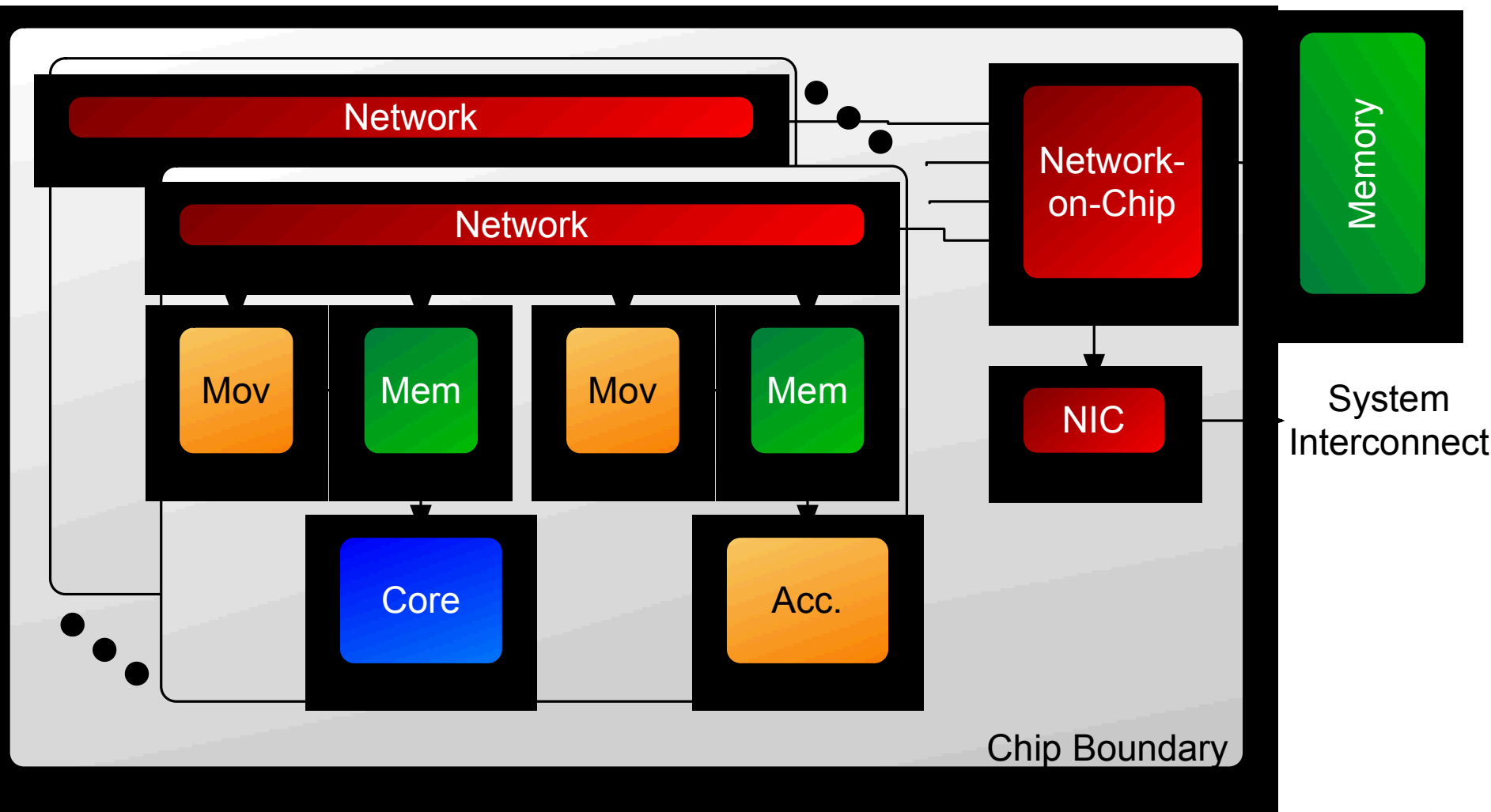
AMM is the topology and schematic for future machines

The Proxy Machine Model fills that in with speeds and feeds

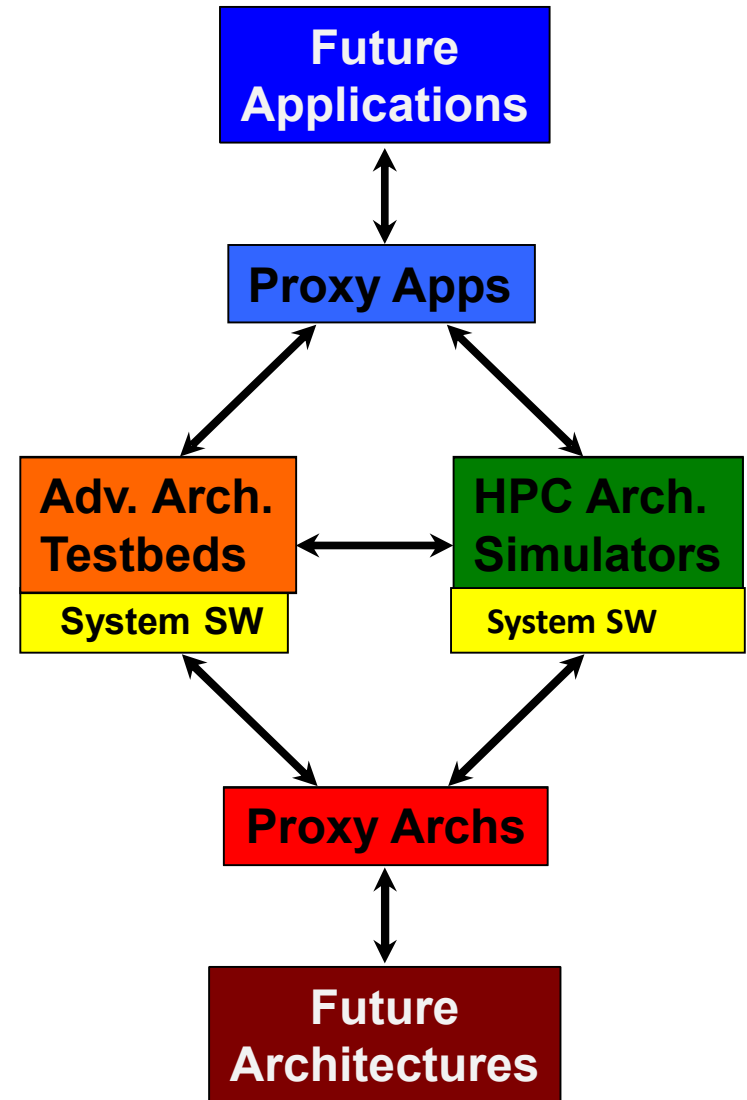|	Processor Cores	Gflop/s per Proc Core	NoC BW per Proc Core (GB/s)	Processor SIMD Vectors (Units x Width)	Accelerator Cores	Acc Memory BW (GB/s)	Acc Count per Node	TFLOP/s per Node[1]	Node Count
Homogeneous M.C. Opt1	256	64	8	8x16	None	None	None	16	62,500
Homogeneous M.C. Opt2	64	250	64	2x16	None	None	None	16	62,500
Discrete Acc. Opt1	32	250	64	2x16	O(1000)	O(1000)	4	16C + 2A	55,000
Discrete Acc. Opt2	128	64	8	8x16	O(1000)	O(1000)	16	8C + 16A	41,000
Integrated Acc. Opt1	32	64	64	2x16	O(1000)	O(1000)	Integrated	30	33,000
Integrated Acc. Opt2	128	16	8	8x16	O(1000)	O(1000)	Integrated	30	33,000
Heterogeneous M.C. Opt1	16 / 192	250	64 / 8	8x16 / 2x8	None	None	None	16	62,500
Heterogeneous M.C. Opt2	32 / 128	64	64 / 8	8x16 / 2x8	None	None	None	16	62,500
Concept Opt1	128	50	8	12x1	128	O(1000)	Integrated	6	125,000
Concept Opt2	128	64	8	12x1	128	O(1000)	Integrated	8	125,000

Table 5.1: $Opt1$ and $Opt1$ represent possible proxy options for the abstract machine model. $M.C$: multi-core, Acc: Accelerator, BW: bandwidth, $Proc$: processor, For models with accelerators and cores, C denotes to FLOP/s from the CPU cores and A denotes to FLOP/s from Accelerators.

The Value to ModSim

- If a benchmark or application simulation can be run using the various models, all of which use the same parameter space and ranges, there is an opportunity for model evaluation and cross analysis

- Rather than focus on interfaces or sharing of simulation components, another approach is to share a set of common targets, providing gap coverage for each of the simulators.

- A high profile example of this type of process is evident in the climate community where a single dataset is processed by multiple climate applications

In closing …

- **ModSim is part of a viable ecosystem for exascale R&D**

- **Start with AMMs that have a basis in the latest technologies**

- **Provide parameterized building blocks for future trends**