Reporting
Jim Belak's Homework Assignment

Traleika Glacier (X-Stack)

https://sites.google.com/site/traleikaglacierxstack/

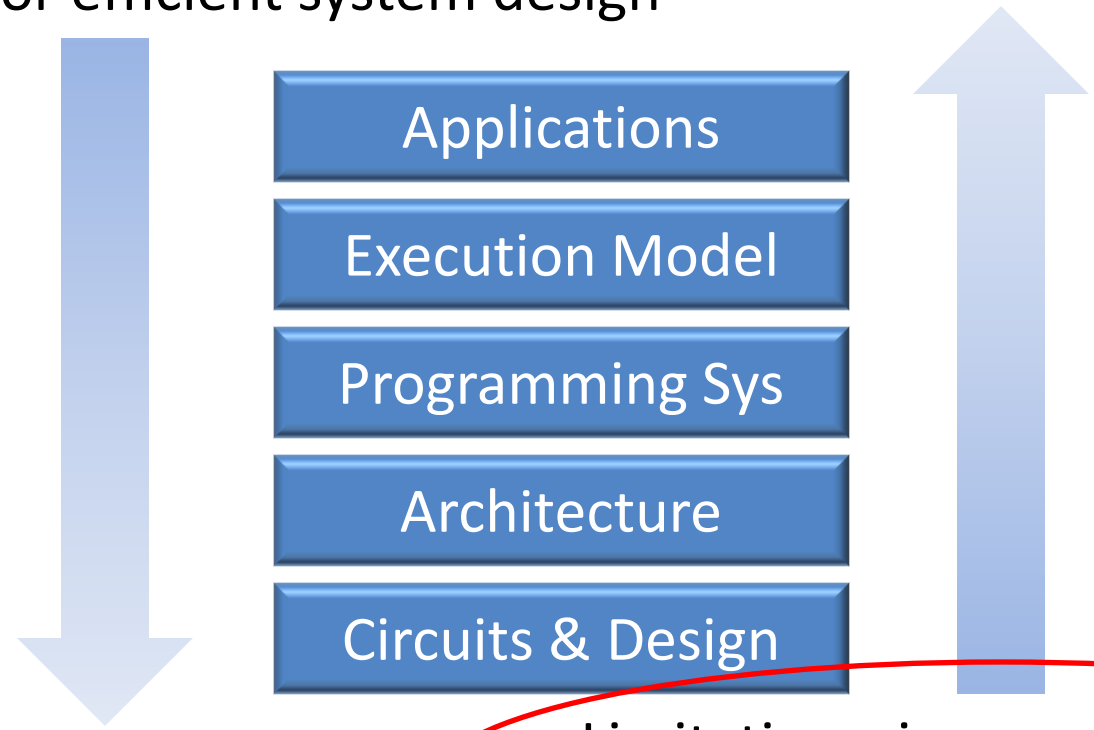
TG Team

March 20, 2013

Assignment: What does an application developer need to know about your HW/SW approach…

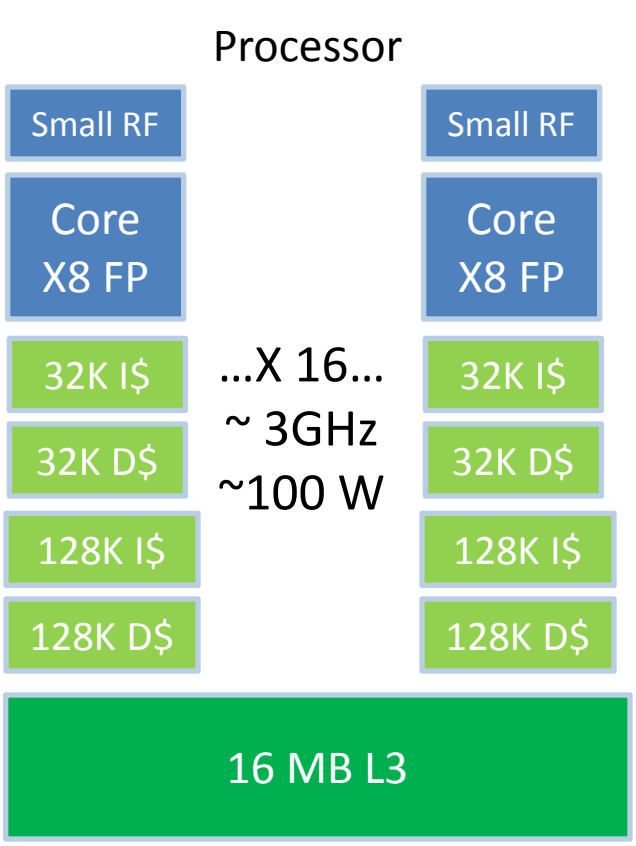
HW-SW Co-design

Applications and SW stack provide guidance for efficient system design

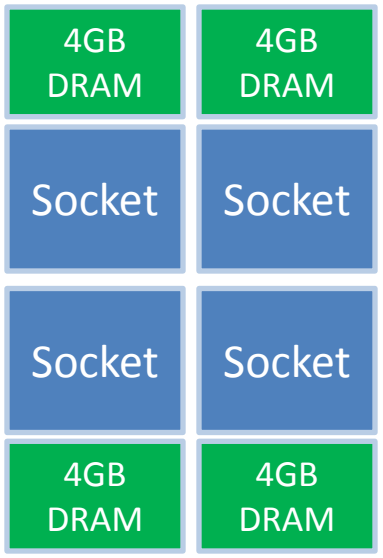


Limitations, issues and opportunities to exploit

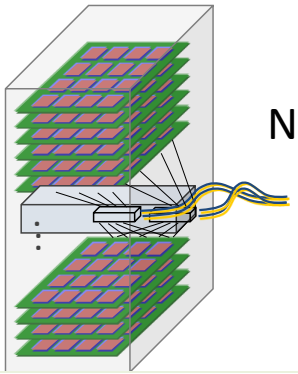
Today's HW System Architecture



384 GF/s Peak
260 pJ/F ➲ 260 MW/Exa
55 µB of local memory/F



Coherent domain
1.5 TF Peak
660 pJ/F ➲ 660 MW/Exa
10 mB of DRAM/F

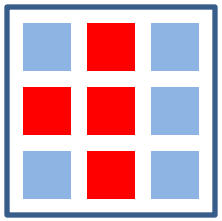


Non-coherent domain

Today's programming model comprehends this system architecture

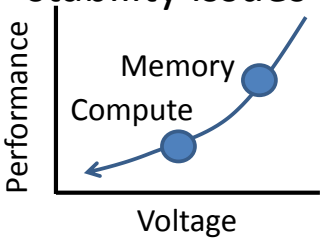
Technology Challenges

1. NTV reduces energy but exacerbates variations



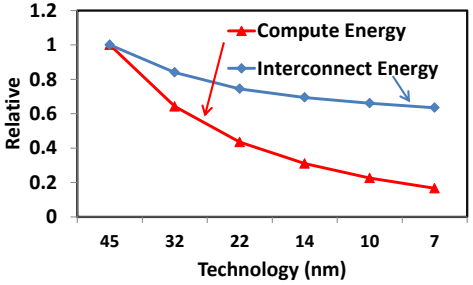
Small & Fast cores
Random distribution
Temp dependent

2. Limited NTV for arrays (memory) due to stability issues



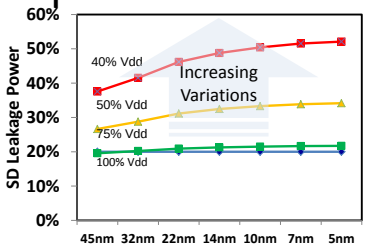
Disproportionate
Memory arrays can be made larger

3. On-die Interconnect energy (per mm) does not reduce as much as compute



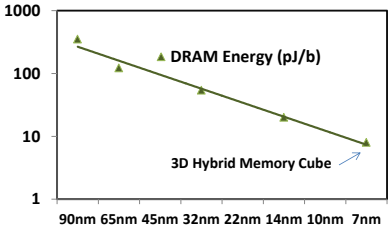
6X compute
1.6X interconnect

4. At NTV, leakage power is substantial portion of the total power



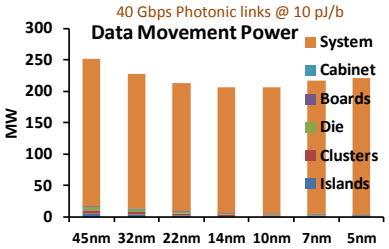
Expect 50% leakage
Idle hardware consumes energy

5. DRAM energy scales, but not enough



50 pJ/b today
8 pJ/b demonstrated
Need < 2pJ/b

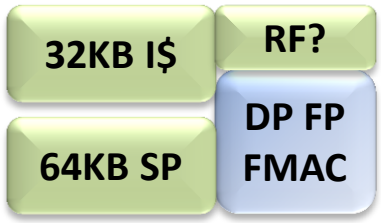
6. System interconnect limited by laser energy and cost



BW tapering and locality awareness necessary

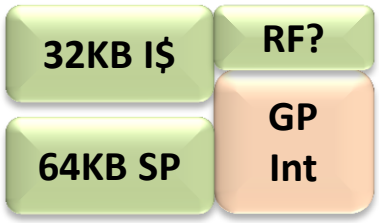
Straw-man Architecture

Execution Engine (XE)



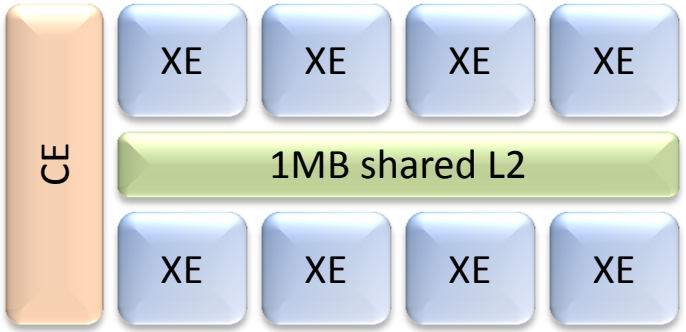
Application specific

Control Engine (CE)

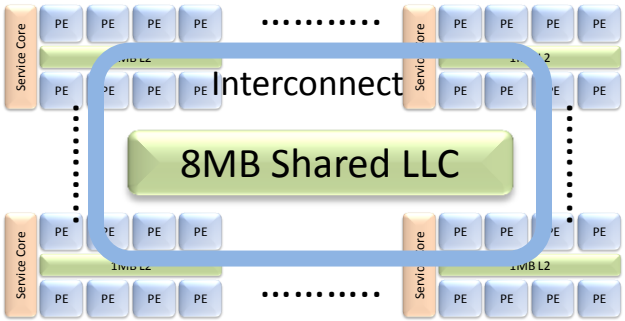


System SW

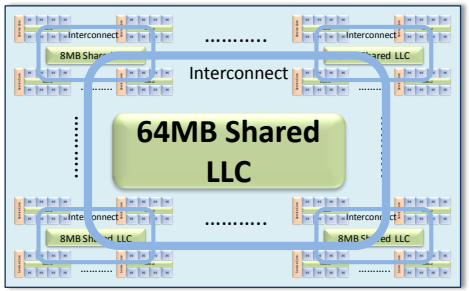
Block (8 XE + CE)



Cluster (16 Blocks)



Processor Chip (16 Clusters)



Technology	7nm, 2018
Die area	500 mm2
XE/die	2048
Frequency	4.2 GHz@Vdd, 600 MHz@50% Vdd
TFLOPs	17.2@Vdd, 2.5@50% Vdd
Power*	600 W@Vdd, 37 W@50% Vdd
E Efficiency*	34 pJ/F@Vdd, 15 pJ/F@50% Vdd
Memory B/F	39 µB/F@Vdd, 268 µB/F@50%Vdd

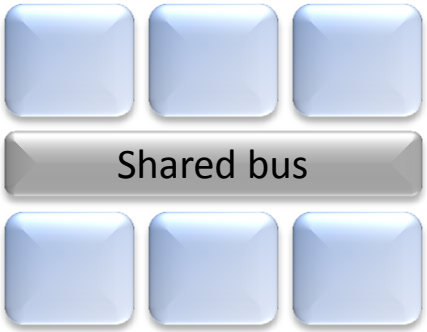
* Without interconnect (data movement)

Wide dynamic range in HW

Global, shared, non-coherent address space

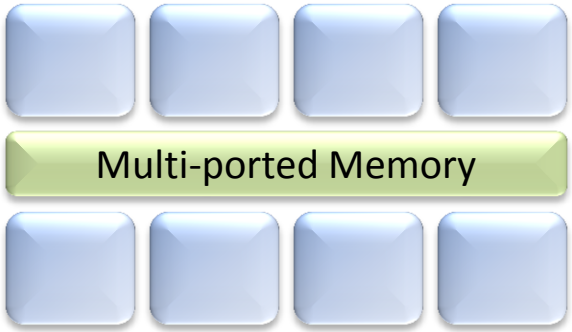
Interconnect Structures and Topologies

Buses over short distance



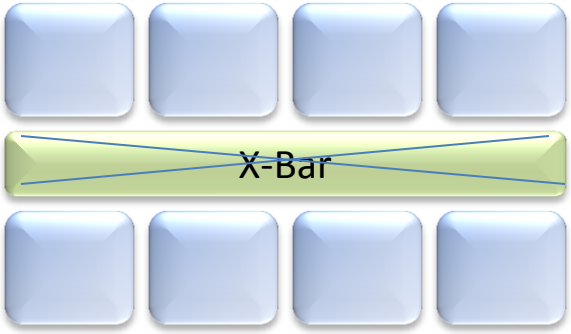
1 to 10 fJ/bit
0 to 5mm
Limited scalability

Shared memory



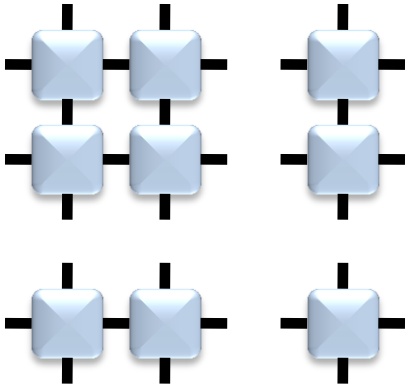
10 to 100 fJ/bit
1 to 5mm
Limited scalability

Cross Bar Switch



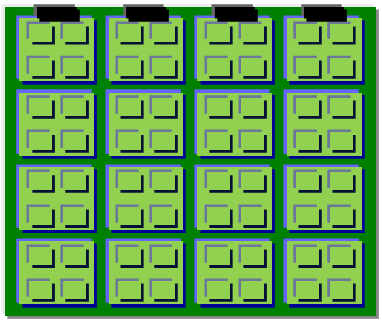
0.1 to 1pJ/bit
2 to 10mm
Moderate scalability

Packet switched network



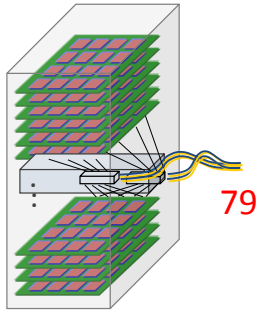
1 to 3pJ/bit
>5 mm, scalable

Board



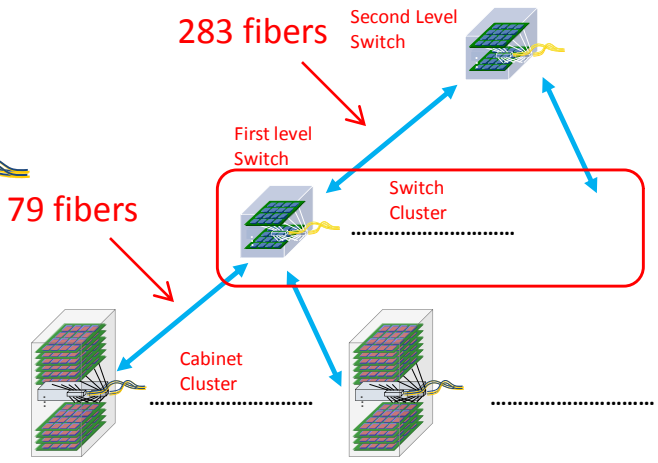
1.39 TB/sec
2,600 wires

Cabinet

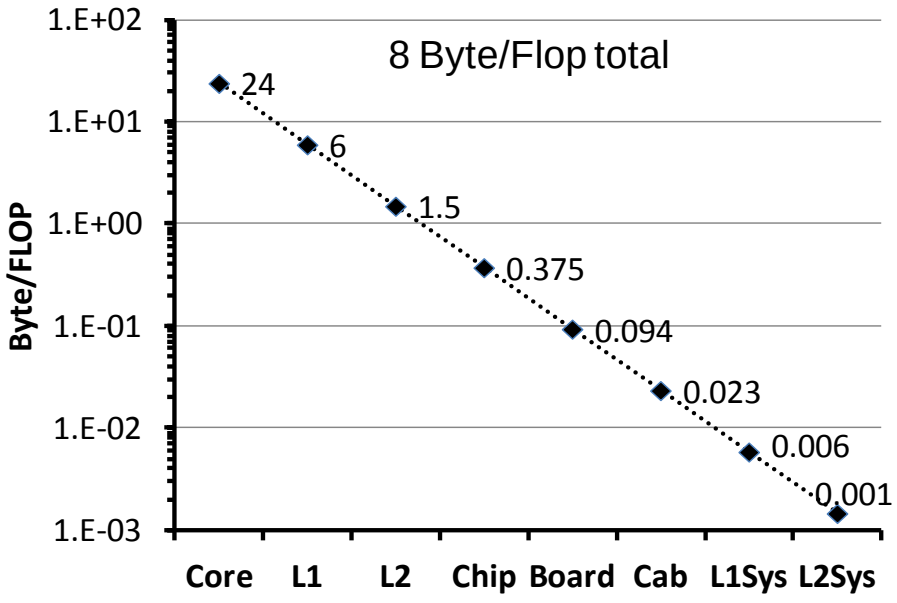


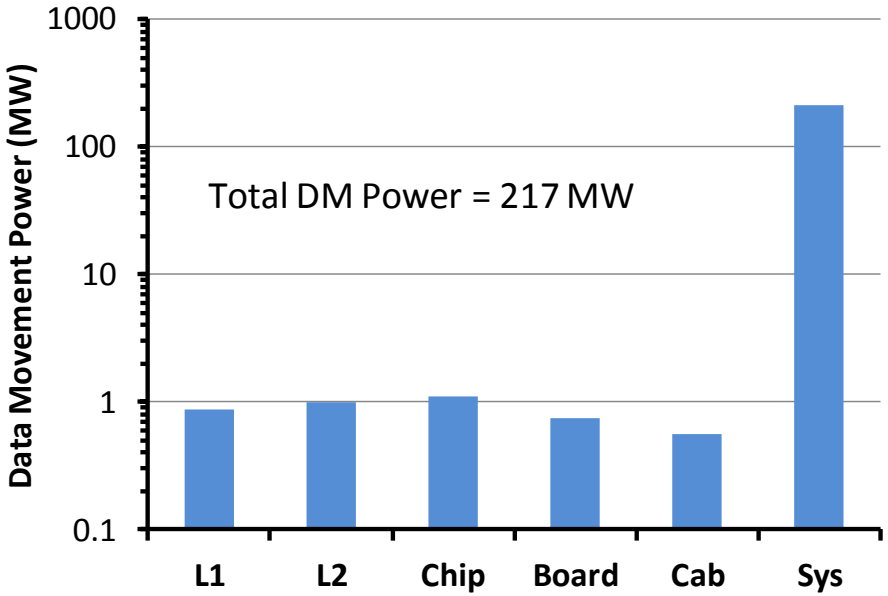
3.76 TB/sec
7,200 wires

System



Data Movement Energy





8 B/Flop at the system level

Naïve BW tapering

Geometric—4X each level

Almost constant/hierarchy

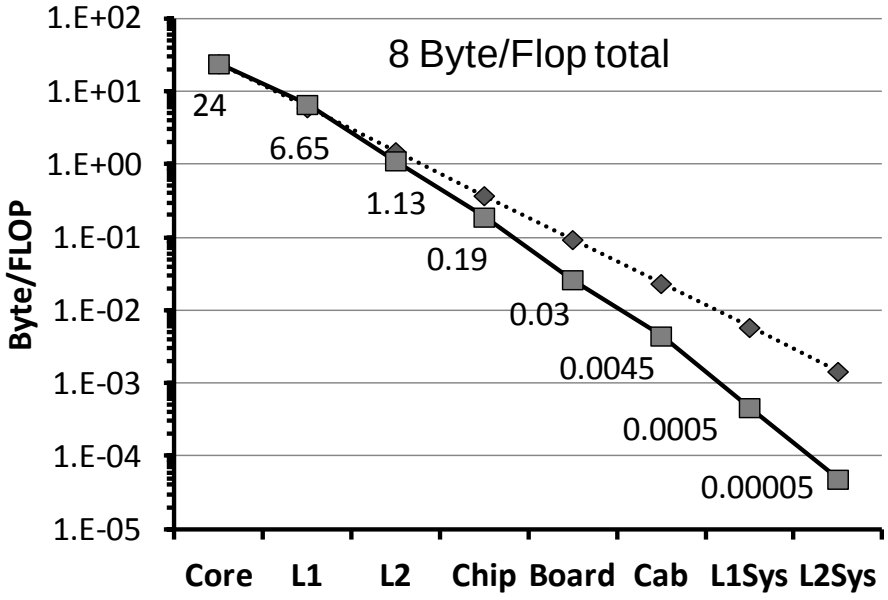
System level dominates

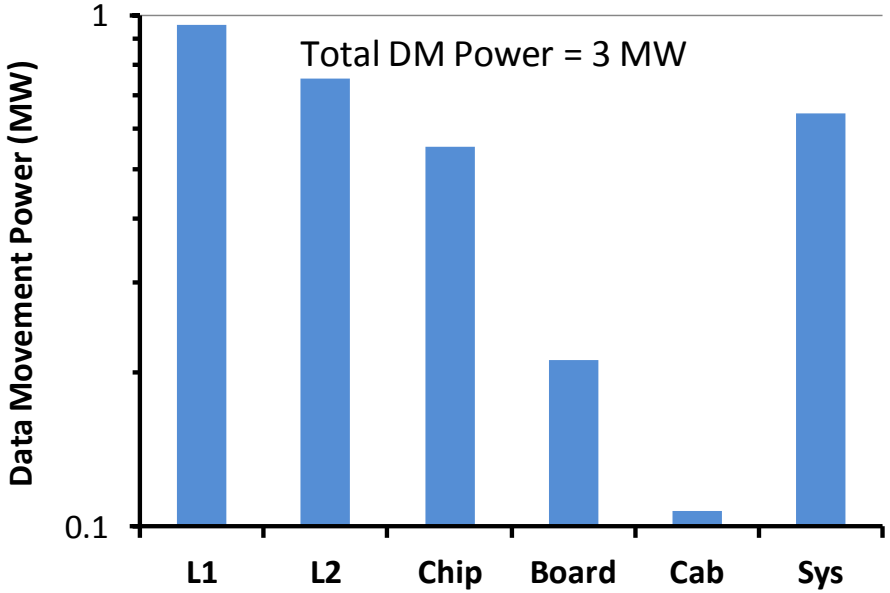
Disproportionately large

Intelligent BW tapering is necessary

Intelligent BW Tapering

BW tapering inversely proportional to the performance





8 B/Flop at the system level

Tapering increases/hierarchy

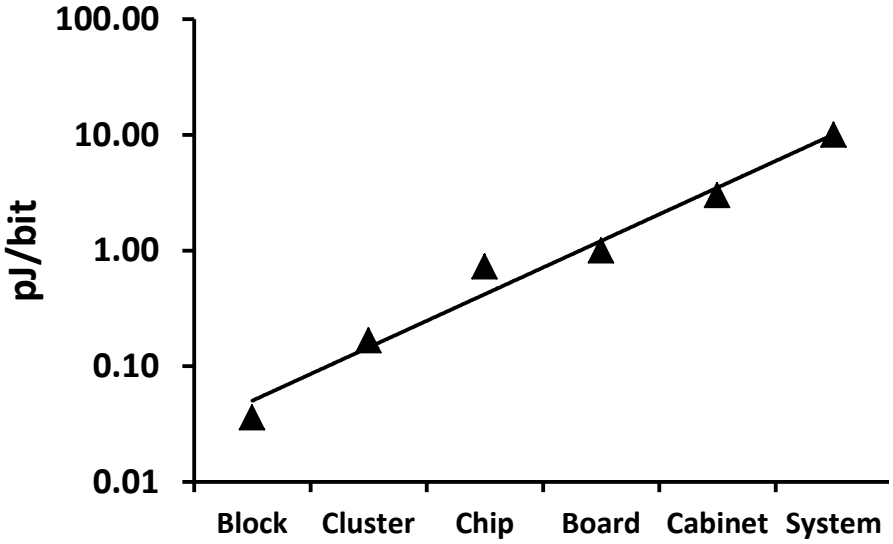
This is counterintuitive…!

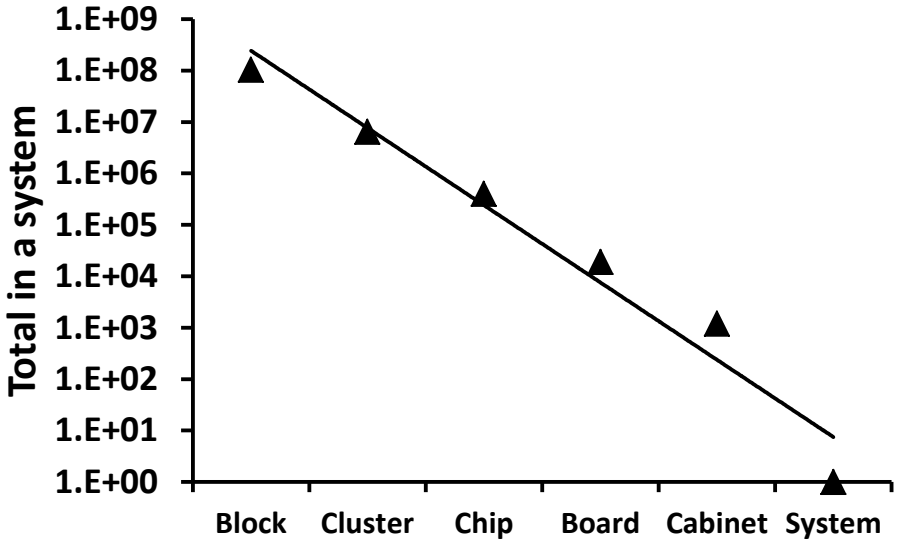
Decreases with hierarchy

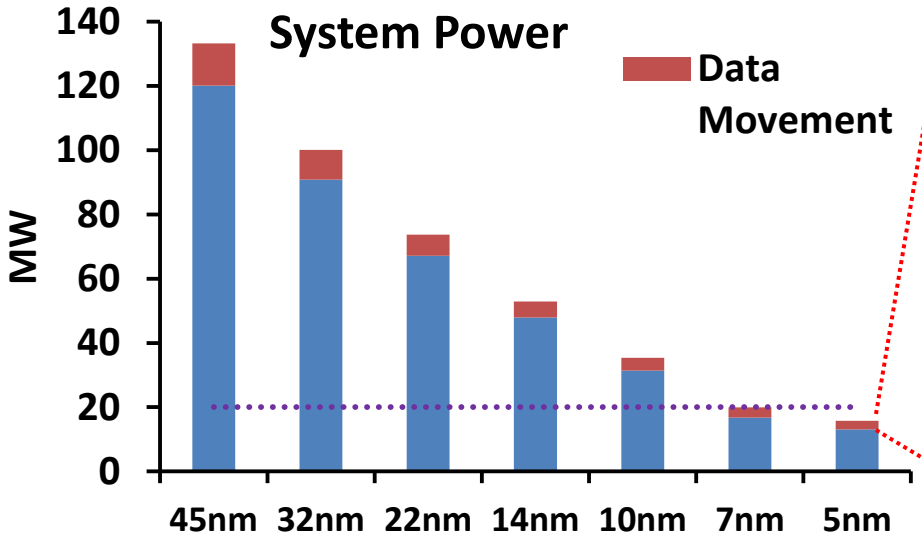
Meets system power goal

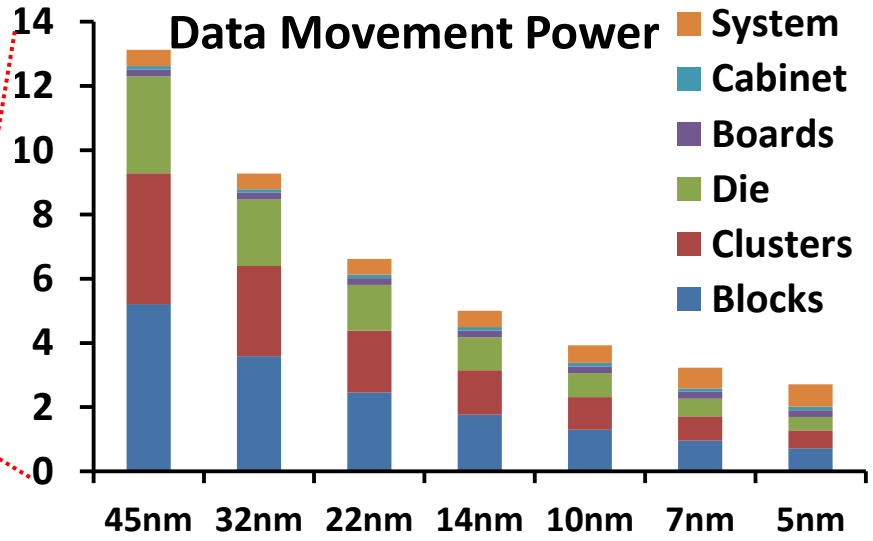
Data locality—key challenge for software

Exascale Data Movement Power (with Tapering)









Simulators Capture Straw-man Architecture

Simulator	Pros	Cons
OCR based behavioral simulator	Near-native application code execution on host processor Rapid application development Epoch statistics as well as total statistics	Does not model real architecture Does not model advanced ISA features Does not reflect expected timing of simulated system
Fsim Functional Simulator	Models HW units, simple timing model Complete statistics and trace file 1-10 MIPS per core speed Massively parallel and distributed	Lower speed, highly detailed

Tool	Purpose	Advantage	Weakness
Power Estimator	• Uses statistics/counters to make energy and area estimates for application behavior	• Scales from 45nm to 7nm projections • Automatic analysis of outputs from FSim runs	• Only models dynamic power, uses circuit models for leakage • Calibrated to existing commercial devices
Memory Analyzer	• Detailed models for cache and/or scratchpad hierarchies, various levels & types of coherence • Compares configurations	• Enables limited behavioral trace power estimation	• Does not model Instruction fetch/execution • Limited to behavioral memory traces at this time

Tools almost ready: LLVM, Utilities, etc.

Available to the community

SW Challenges

Programming model

Execution model

1. Extreme parallelism (1000X due to Exa, additional 4X due to NTV)
2. Data locality—reduce data movement
3. Intelligent scheduling—move thread to data if necessary
4. Fine grain resource management (objective function)
5. Applications and algorithms incorporate paradigm change

Programming & Execution Model

Event driven tasks (EDT)

Dataflow inspired, tiny codelets (self contained)

Non blocking, no preemption

Programming model:

Separation of concerns: Domain specification & HW mapping

Express data locality with hierarchical tiling

Global, shared, non-coherent address space

Optimization and auto generation of EDTs (HW specific)

Execution model:

Dynamic, event-driven scheduling, non-blocking

Dynamic decision to move computation to data

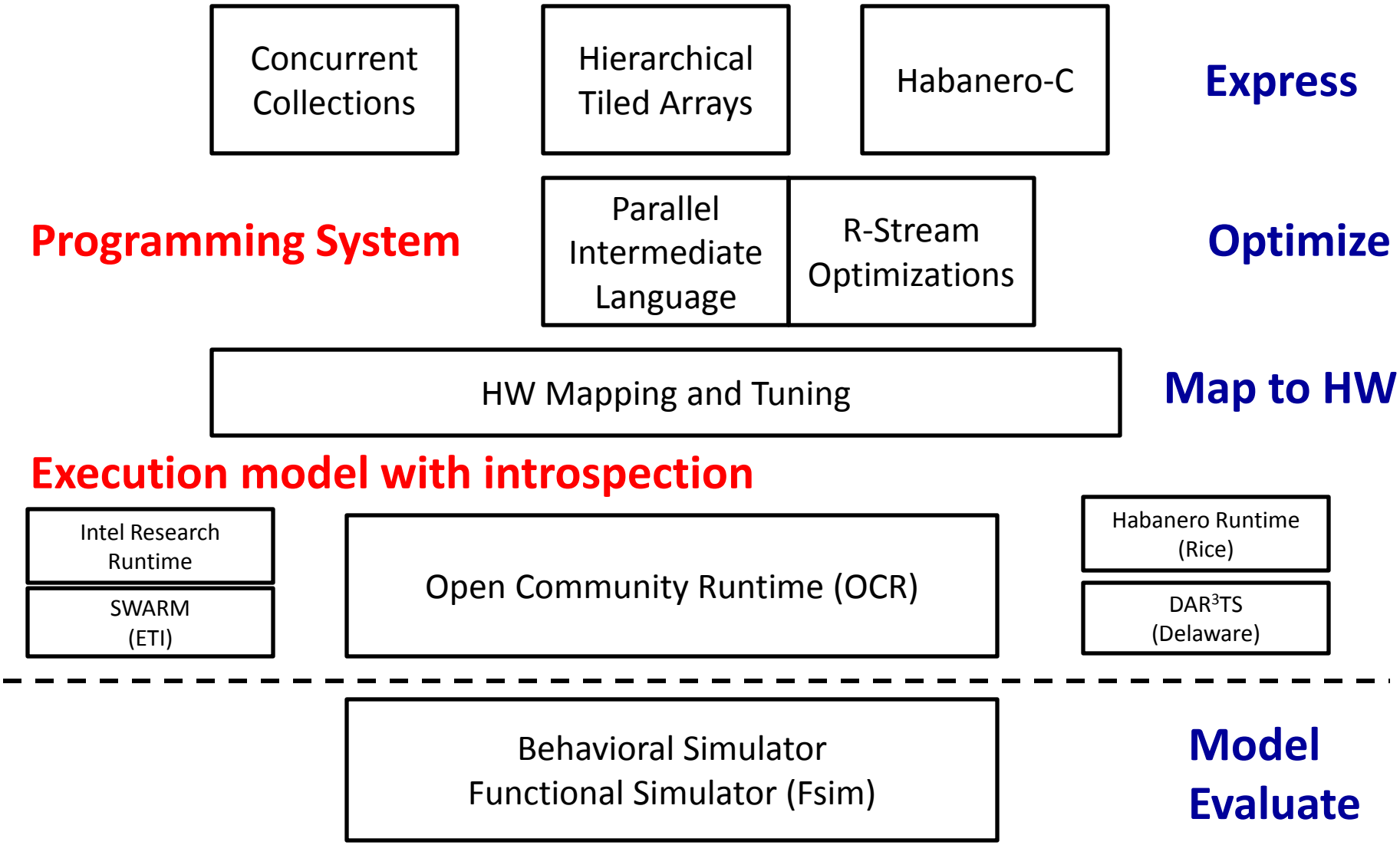
Observation based adaption (self-awareness)

Implemented in the runtime environment

Separation of concerns:

User application, control, and resource management

Traleika Glacier SW Stack



Over-provisioning, Introspection, Self-awareness

Addressing variations

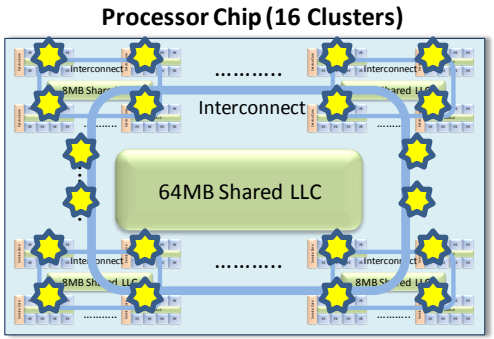
1. Provide more compute HW
2. Law of large numbers
3. Static profile

Fine grain resource mgmt

Dynamic reconfiguration:

1. Energy efficiency
2. Latency
3. Dynamic resource management

Sensors for introspection



1. Energy consumption
2. Instantaneous power
3. Computations
4. Data movement

1. Schedule threads based on objectives and resources

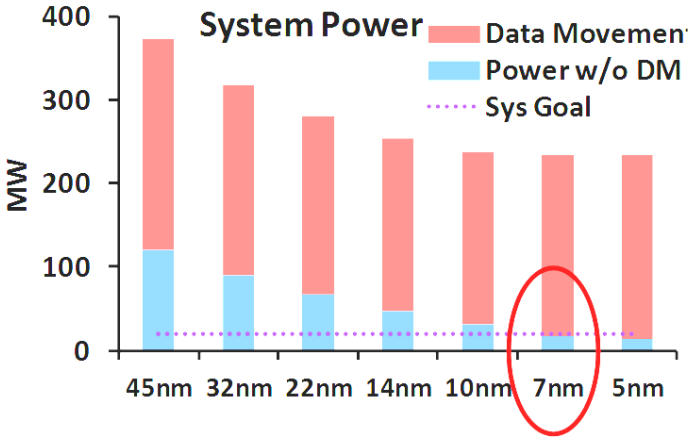
2. Dynamically control and manage resources

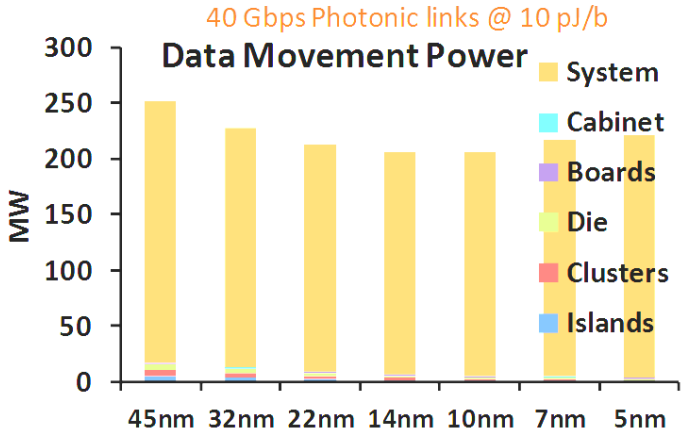
3. Identify sensors, functions in HW for implementation

System SW implements introspective execution model

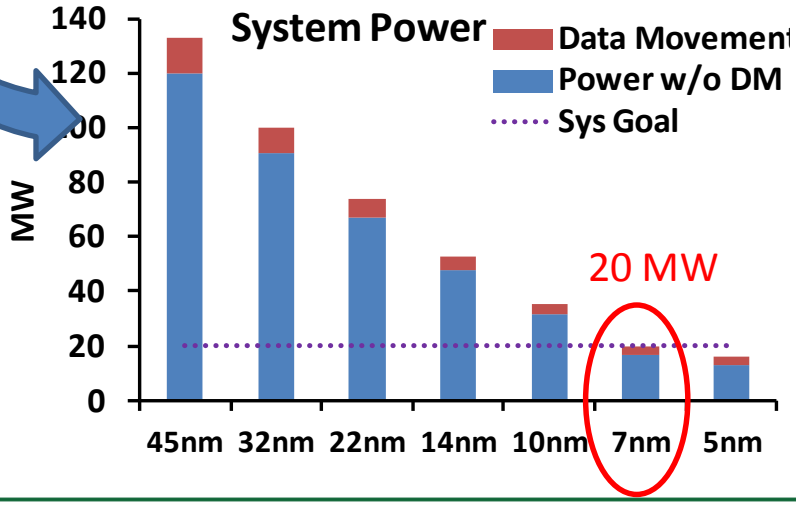
Over-provisioned Introspectively Resource Managed System

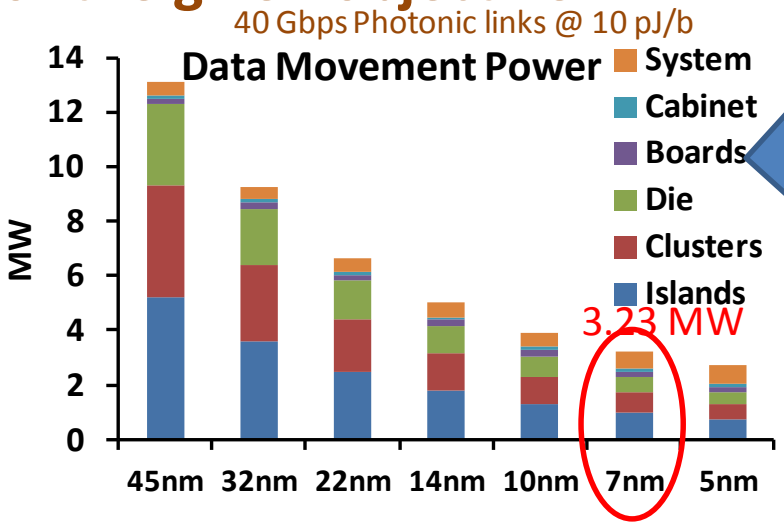
Over-provisioned in design



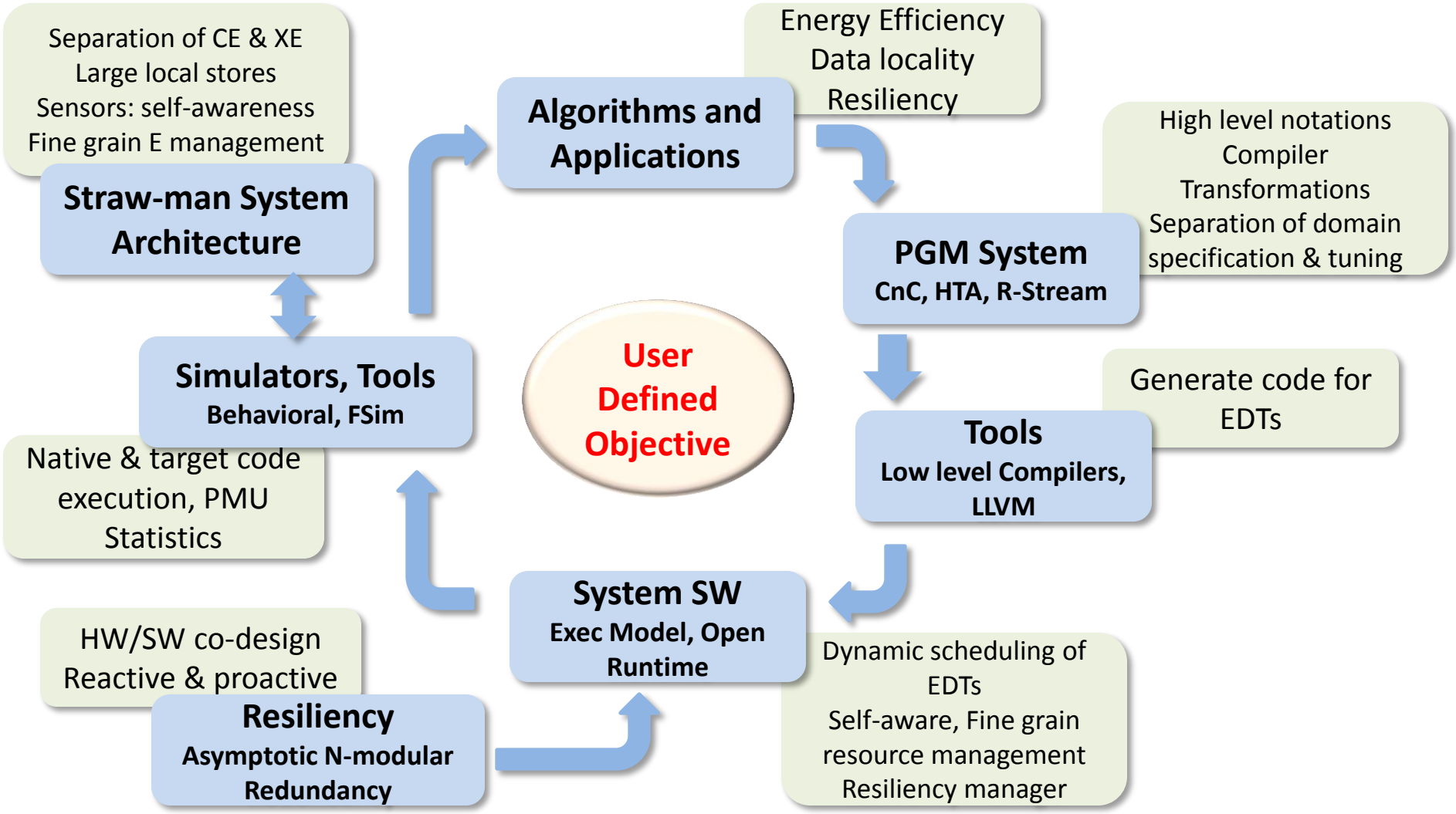


Dynamically tuned for the given objective



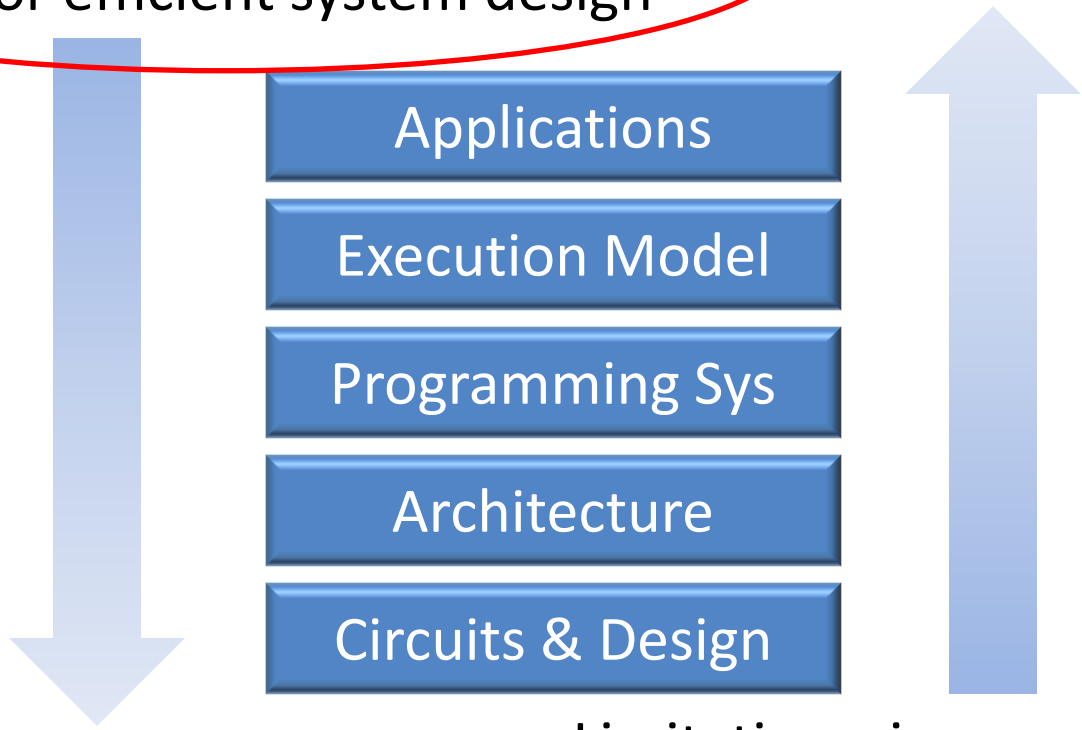


X-Stack Components Put Together



HW-SW Co-design

Applications and SW stack provide guidance for efficient system design



Limitations, issues and opportunities to exploit

Summary

- **Straw-man architecture comprehends technology challenges**
- **Simulators capture the architecture, ready for evaluation**
- **Tools and infrastructure are getting ready**
- **Software stack is making good progress**
- **Getting ready for thorough evaluation**