# Rebooting the Data Access Hierarchy of Computing Systems

Wen-mei Hwu

Professor and Sanders-AMD Chair, ECE, NCSA, CS

University of Illinois at Urbana-Champaign

with

Izzat El Hajj, Simon Garcia, Carl Pearson, Nam Sung Kim, Deming Chen,

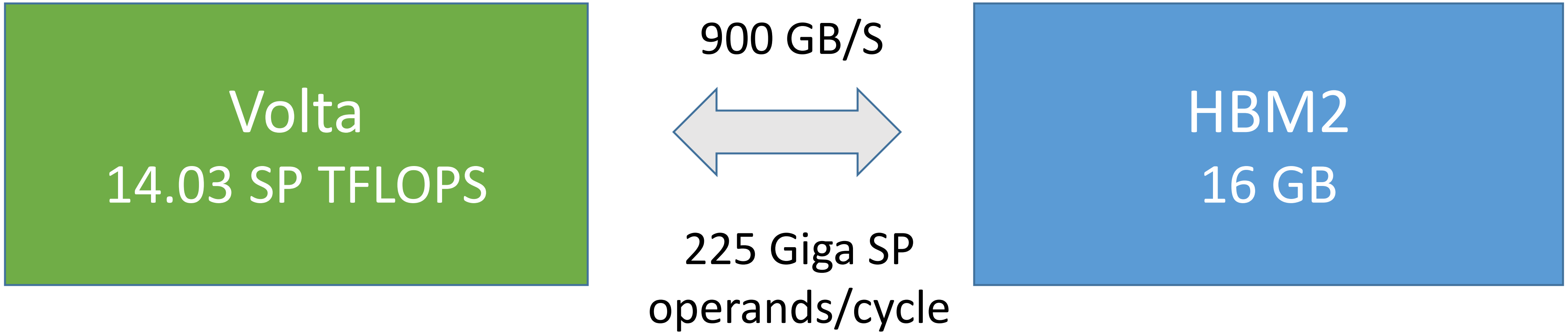
Jinjun Xiong (IBM), Zehra Sura (IBM)

ECE ILLINOIS C3SR center for cognitive computing systems research NCSA The IMPACT Research Group NVIDIA ILLINOIS

# Agenda

- Data access challenges
- The IBM-Illinois Erudite project

# Data Access Challenge (HBM)

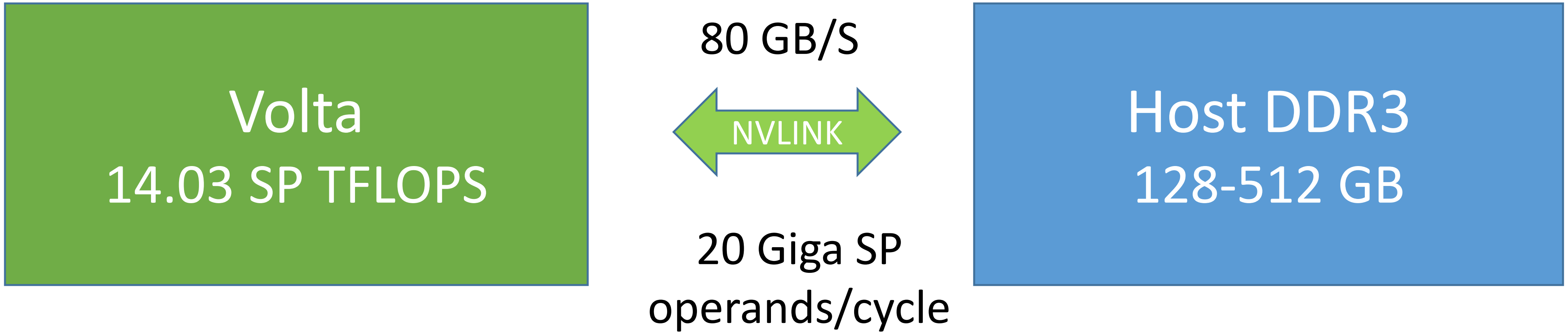



Each operands must be used **62.3 times** once fetched to achieve peak FLOPS rate.

or

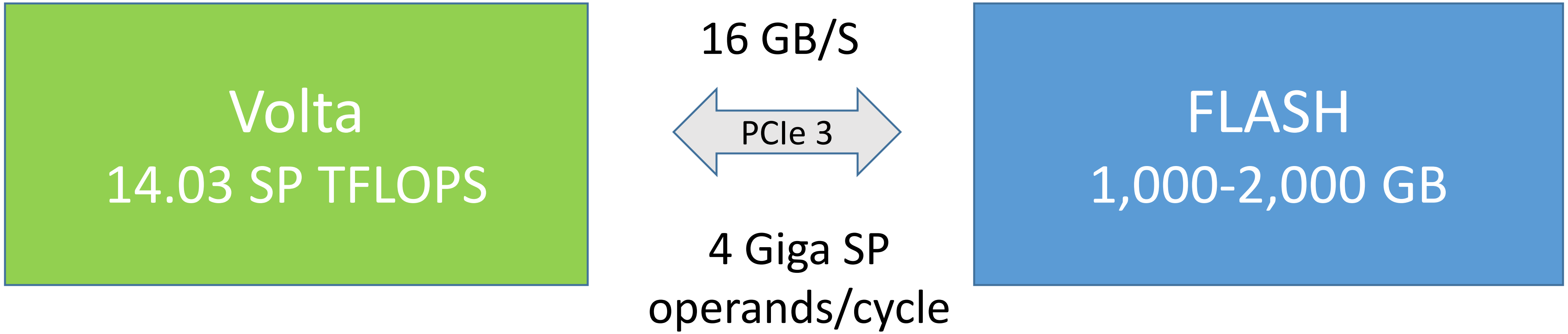
Sustain < **1.6%** of peak without data reuse

# Data Access Challenge (DDR DRAM)




Each operands must be used **700 times** once fetched
to achieve peak FLOPS rate.
or
Sustain < **0.14% peak** without data reuse

# Data Access Challenge (FLASH)




Each operands must be used **3,507 times** once fetched
to achieve peak FLOPS rate.
or
Sustain **< 0.03%** of peak without data reuse

# Large Problem Challenge

- Solving larger problems motivates continued growth of computing capability
  - Inverse solvers for science and engineering applications
  - Matrix factorization and graph traversal for analytics

- As problem size grows,
  - Fast, low complexity (O(n) or O(nlog(n)) algorithms win
  - Sparsity increases, iterative methods win
  - → data reuse diminishes!

# Graph Analytics Example on IBM Minsky with Pascal GPUs

HPCC 2017

TABLE II
TRUSS DECOMPOSITION BENCHMARKS ON REAL-WORLD GRAPHS

| Graph | $n$ | $m$ | $k_{max}$ | Python Baseline | | Zero-copy | | | Unified | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Time (s) | Rate (edges/s) | Time (s) | Rate (edges/s) | Speedup | Time (s) | Rate (edges/s) | Speedup |
| cit-Patents | 3,774,768 | 33,037,894 | 36 | > 4 hrs | - | 28.76 | 1,148,843.8 | - | 5,459.60 | 6,051.3 | - |
| roadNet-CA | 1,965,206 | 5,533,214 | 4 | 526.18 | 10,515.8 | 3.74 | 1,477,924.7 | 140.54 | 4.63 | 1,193,838.4 | 113.53 |
| amazon0601 | 403,394 | 4,886,816 | 11 | 1,443.57 | 3,385.2 | 9.06 | 539,148.4 | 159.27 | 656.23 | 7,446.8 | 2.20 |
| amazon0505 | 410,236 | 4,878,874 | 11 | 2,666.41 | 1,829.8 | 9.43 | 517,160.8 | 282.64 | 684.17 | 7,131.1 | 3.90 |
| amazon0312 | 400,727 | 4,699,738 | 11 | 2,213.74 | 2,123.0 | 10.24 | 459,045.0 | 216.23 | 626.42 | 7,502.6 | 3.53 |
| flickrEdges | 105,938 | 4,633,896 | 574 | > 4 hrs | - | 195.47 | 23,706.8 | - | > 4 hrs | - | - |
| roadNet-TX | 1,379,917 | 3,843,320 | 4 | 368.98 | 10,416.2 | 3.79 | 1,013,800.0 | 97.33 | 4.53 | 848,926.8 | 81.50 |
| roadNet-PA | 1,088,092 | 3,083,796 | 4 | 295.11 | 10,449.7 | 3.57 | 864,595.0 | 82.74 | 4.56 | 676,667.7 | 64.75 |
| amazon0302 | 262,111 | 1,799,584 | 7 | 306.63 | 5,868.9 | 4.96 | 362,934.8 | 61.84 | 61.83 | 29,103.6 | 4.96 |
| soc-Slashdot0811 | 77,360 | 938,360 | 35 | 2,863.68 | 327.7 | 10.64 | 88,185.5 | 269.12 | 183.08 | 5,125.4 | 15.64 |
| cit-HepPh | 34,546 | 841,754 | 25 | 1,888.29 | 445.8 | 15.14 | 55,597.5 | 124.72 | 156.27 | 5,386.4 | 12.08 |
| email-EuAll | 265,214 | 728,962 | 20 | 2,508.10 | 290.6 | 10.96 | 66,521.6 | 228.88 | 144.83 | 5,033.1 | 17.32 |
| cit-HepTh | 27,770 | 704,570 | 30 | 2,387.76 | 295.1 | 18.65 | 37,785.2 | 128.05 | 178.33 | 3,950.9 | 13.39 |
| loc-brightkite_edges | 58,228 | 428,156 | 43 | 1,498.01 | 285.8 | 10.37 | 41,307.1 | 144.52 | 51.61 | 8,296.2 | 29.03 |
| ca-AstroPh | 18,772 | 396,100 | 57 | 854.94 | 463.3 | 14.35 | 27,600.8 | 59.57 | 87.90 | 4,506.2 | 9.73 |
| email-Enron | 36,692 | 367,662 | 22 | 1,053.50 | 349.0 | 14.94 | 24,608.6 | 70.51 | 88.05 | 4,175.8 | 11.97 |
| ca-HepPh | 12,008 | 236,978 | 239 | 1,080.12 | 219.4 | 12.33 | 19,226.5 | 87.63 | 55.78 | 4,248.3 | 19.36 |
| ca-CondMat | 23,133 | 186,878 | 26 | 109.94 | 1,699.8 | 5.00 | 37,342.5 | 21.97 | 11.98 | 15,599.3 | 9.18 |
| facebook_combined | 4,039 | 176,468 | 97 | 1,235.49 | 142.8 | 25.40 | 6,947.1 | 48.64 | 79.89 | 2,208.8 | 15.46 |
| as-caida20071105 | 26,475 | 106,762 | 16 | 143.37 | 744.7 | 5.35 | 19,946.8 | 26.79 | 9.56 | 11,163.7 | 14.99 |
| p2p-Gnutella04 | 10,876 | 79,988 | 4 | 3.82 | 20,939.2 | 3.58 | 22,353.0 | 1.07 | 3.66 | 21,879.4 | 1.04 |
| oregon1_010331 | 10,670 | 44,004 | 16 | 39.43 | 1,115.9 | 5.07 | 8,677.7 | 7.78 | 6.49 | 6,784.1 | 6.08 |
| as20000102 | 6,474 | 25,144 | 10 | 12.13 | 2,073.2 | 4.29 | 5,859.8 | 2.83 | 4.99 | 5,042.1 | 2.43 |

# Agenda

- Data access challenges
- The Erudite project

# IBM-Illinois $C^3SR$ faculties & students (Est. Sep./2016)

## Wen-mei Hwu (Illinois) and Jinjun-Xiong (IBM) Co-directors


Suma Bhat


Minh Do


Deming Chen


Julia Hockenmaier


Wen-mei Hwu


Nam Sung Kim


Dan Roth


Rakesh Nagi

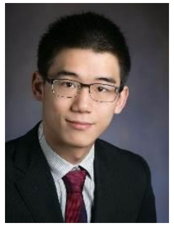
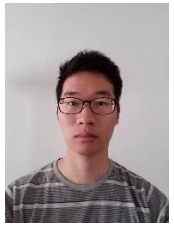

Lav Varshney



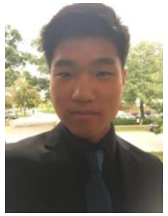






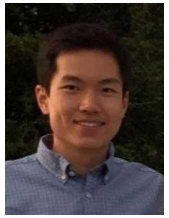








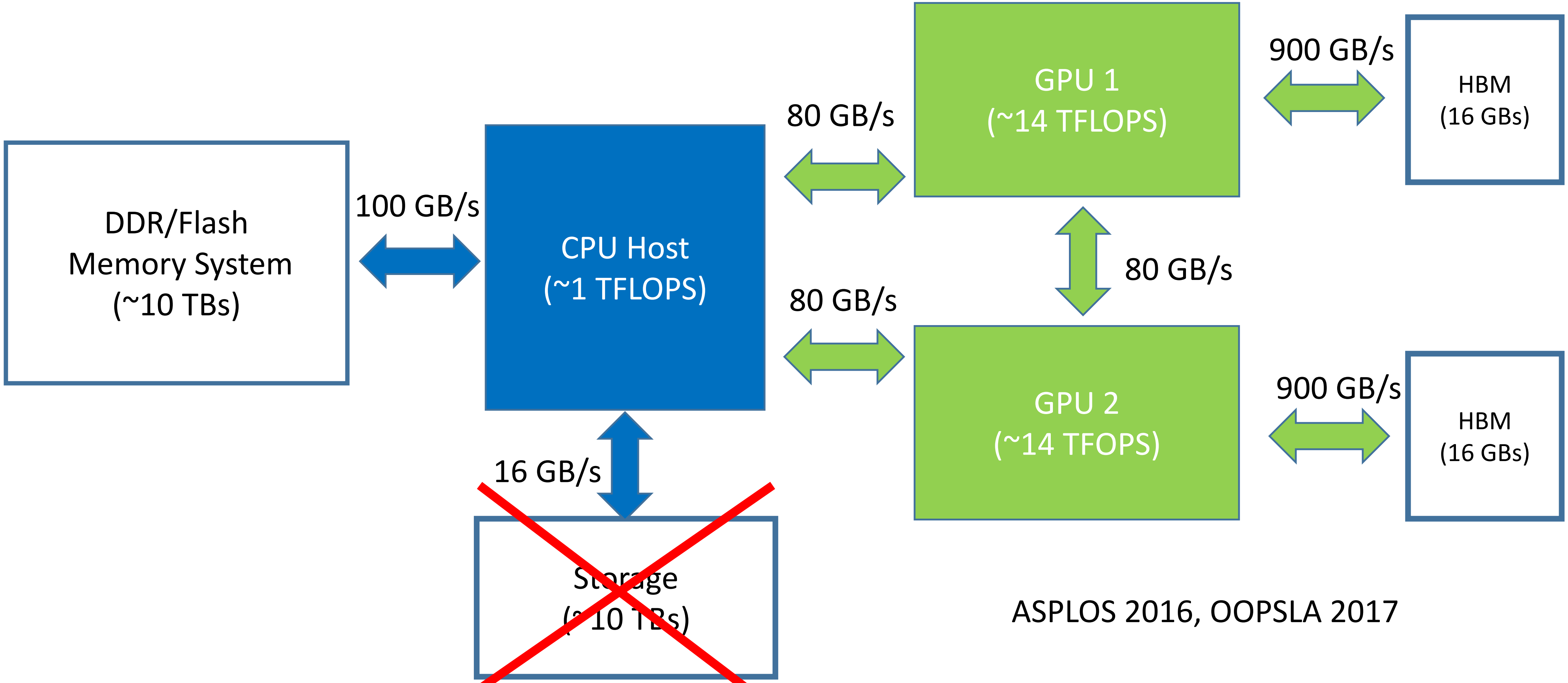
# Erudite Target Computation Types

- Low-complexity iterative solver algorithms
  - Multi-level Fast Multipole Methods, etc.
- Graph analytics
  - Inference, search, counting, etc.
- Large cognitive applications
  - Large multi-model classifiers, etc.

# Erudite Project Approach

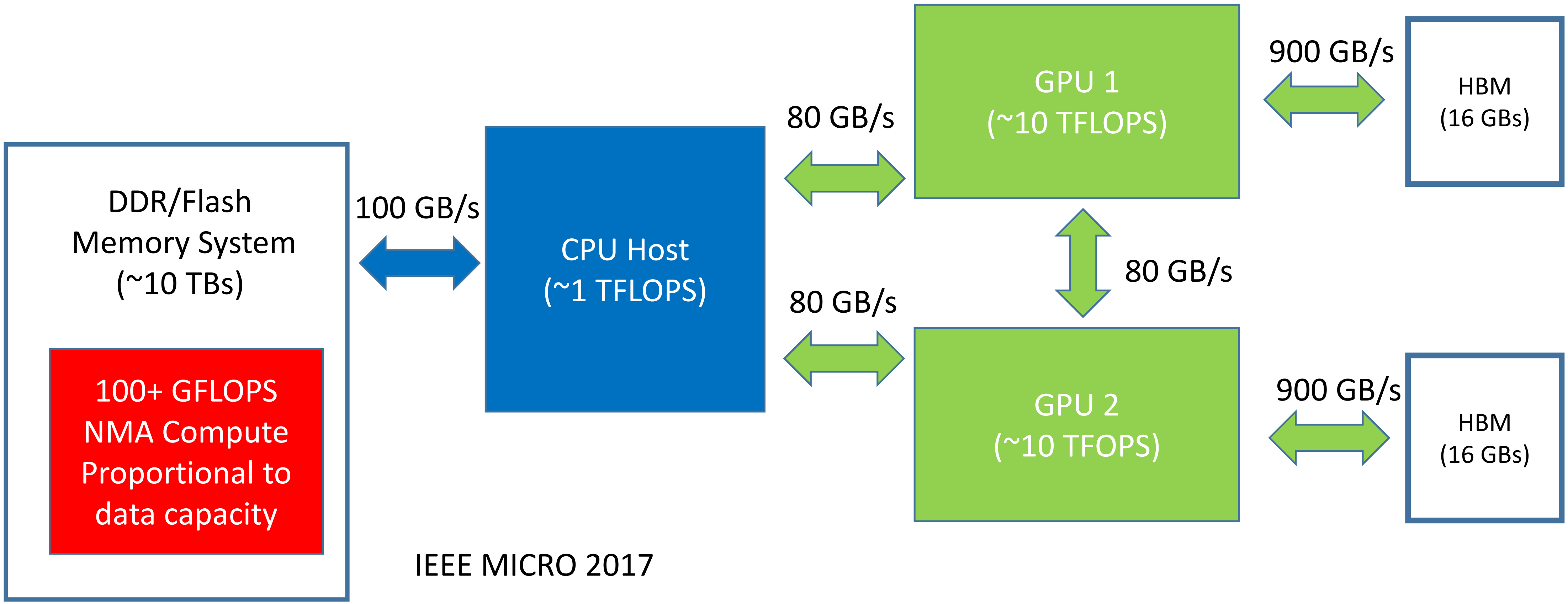
- To achieve > 100x performance/Watt for data-intensive cognitive computing applications
  - Elimination of file-system software overhead for engaging large data sets
  - Placement of computation appropriately in the memory and storage hierarchy
  - Highly optimized kernel synthesis for NMA
  - Collaborative heterogeneous execution of CPU, GPU, and NMA

# Erudite Step 1: remove file system from data access path



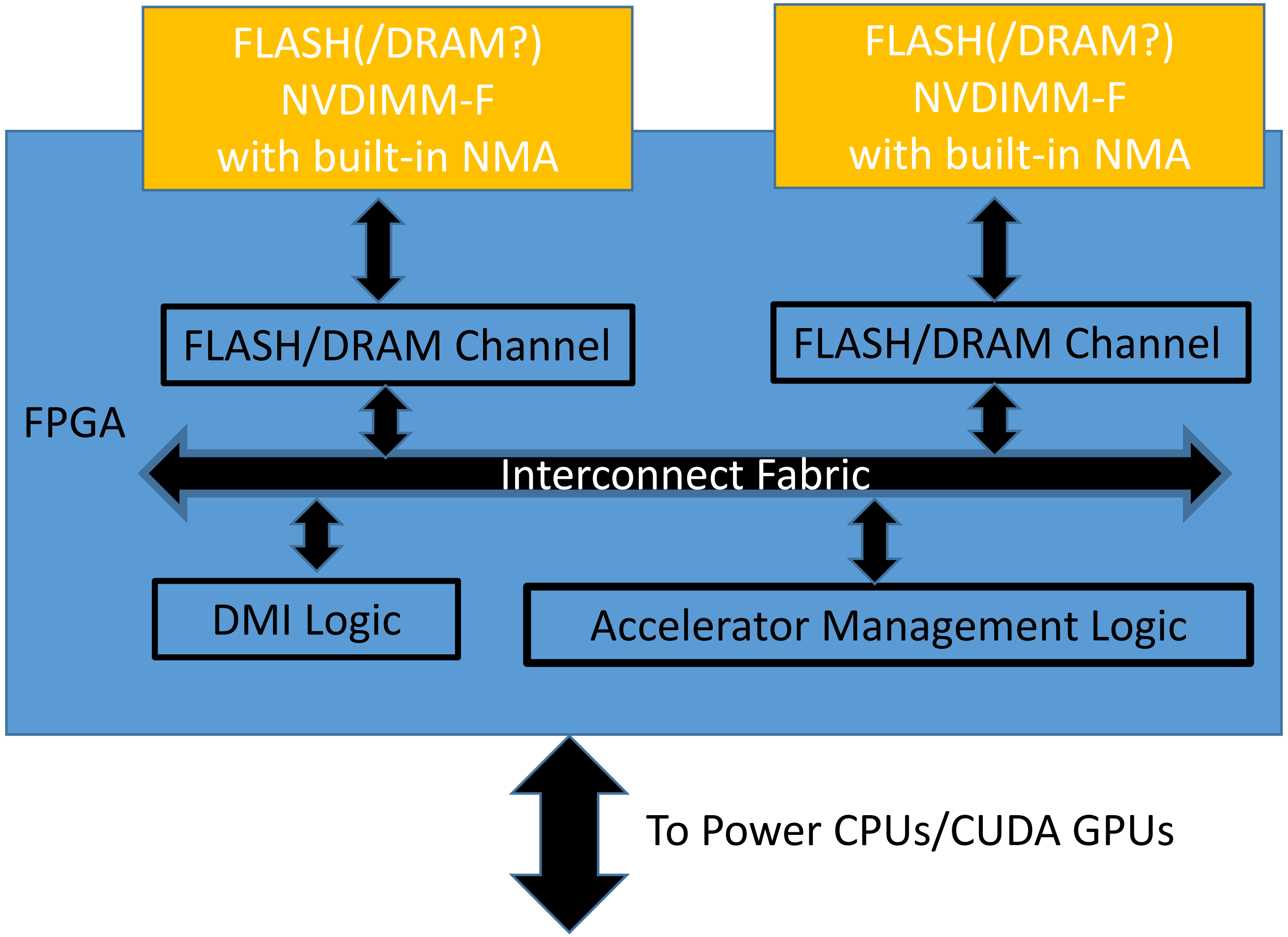

ASPLOS 2016, OOPSLA 2017

# Erudite Step 2: place NMA compute inside memory system



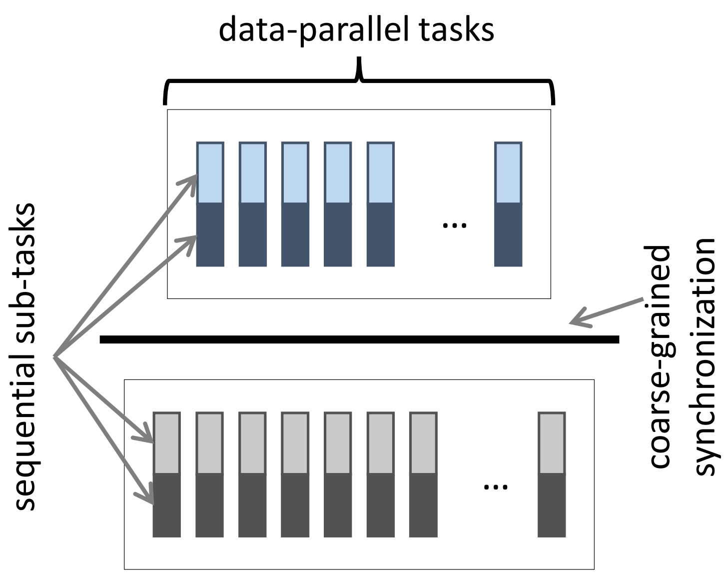

IEEE MICRO 2017

# Erudite NMA Board 1.0

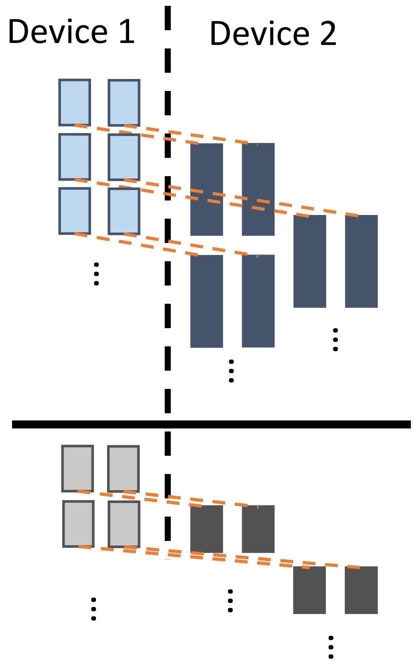



- Develop a principled methodology for acceleration
  - HLS (high-level synthesis) for FPGA based on TANGRAM
  - Hardware/Software partitioning for heterogeneous systems
  - Optimized for cognitive workload
- Throughput proportional to capacity
  - 1 GFLOPS / 10 GB sustained
  - 100 GFLOS sustained for 1TB

# Erudite Step 3
# collaborative heterogeneous computing (Chai)




Program Structure




Fine-grained Task Partitioning

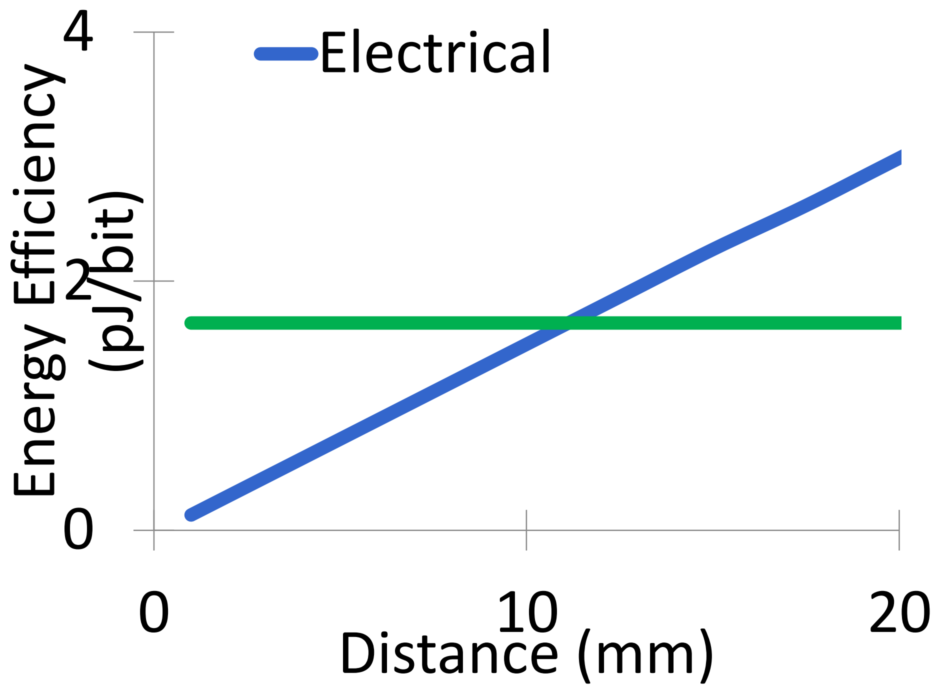
ICPE 2017

# Erudite Research Agenda

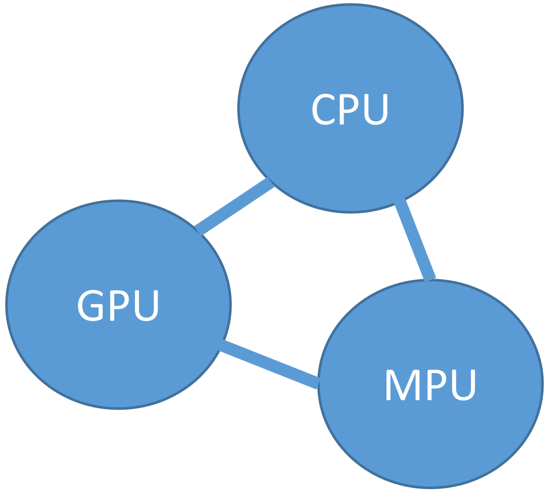
- Package-level integration
  - Post-Moore scaling
  - Optical interconnects in package?
  - Collaboration support for heterogeneous devices
  - Virtual address translation

- System software (r)evolution
  - Persistent objects for multi-language environments
  - Directory and mapping of very large persistent objects

- Power consumption in memory
  - Much higher memory-level parallelism needed for FLASH-based memories
  - Latency vs. throughput oriented memories



# Conclusion and Outlook



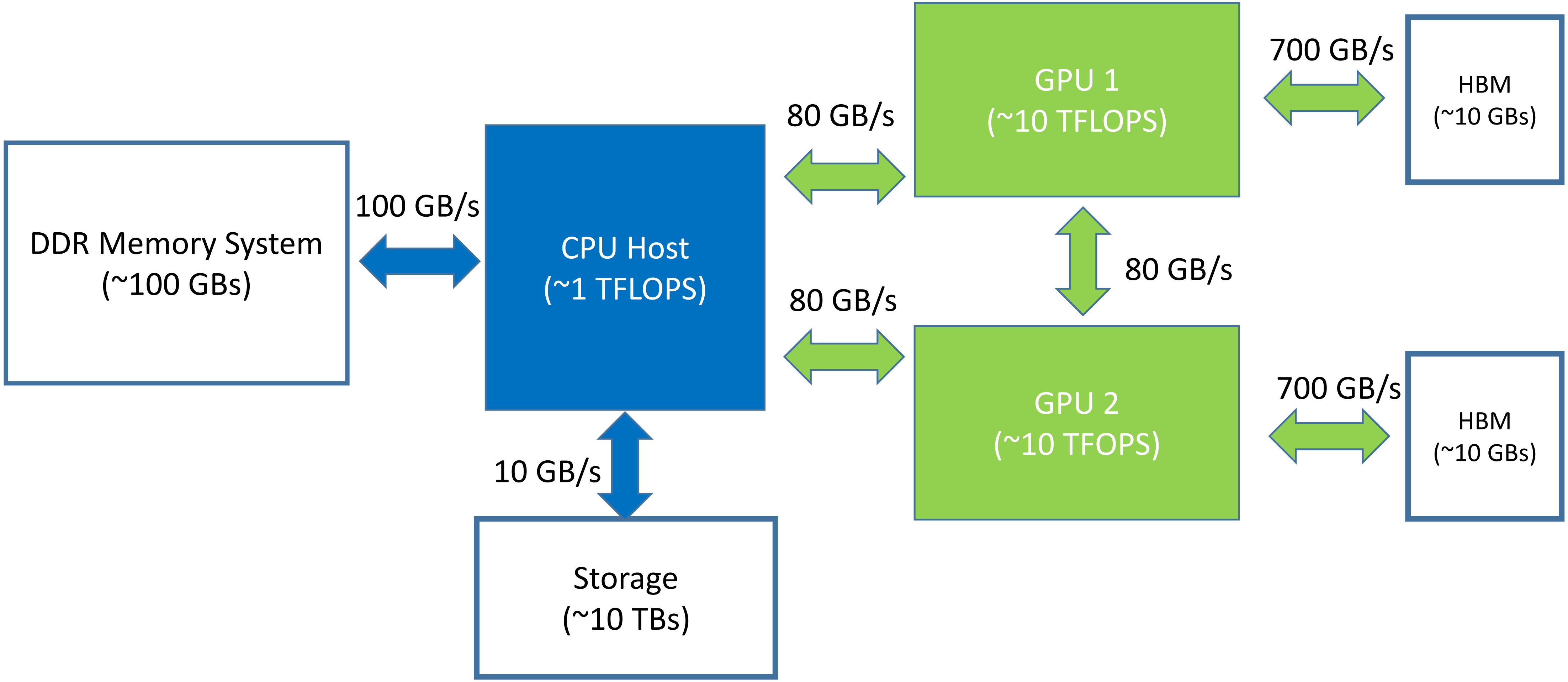

- Drivers for computing capabilities
  - Large-scale inverse problems with natural data inputs
  - Machine-learning-based applications
- Erudite cognitive computing systems project
  - Removing file-system bottleneck from access paths to large data sets
  - Placing compute into the appropriate levels of the memory system hierarchy
  - Memory parallelism (data bandwidth) proportional to the data capacity
  - Collaborative NMA execution with CPUs and GPUs
  - > 100x improvement in power-efficiency and performance

Thank you!

# What is driving new computing innovations?

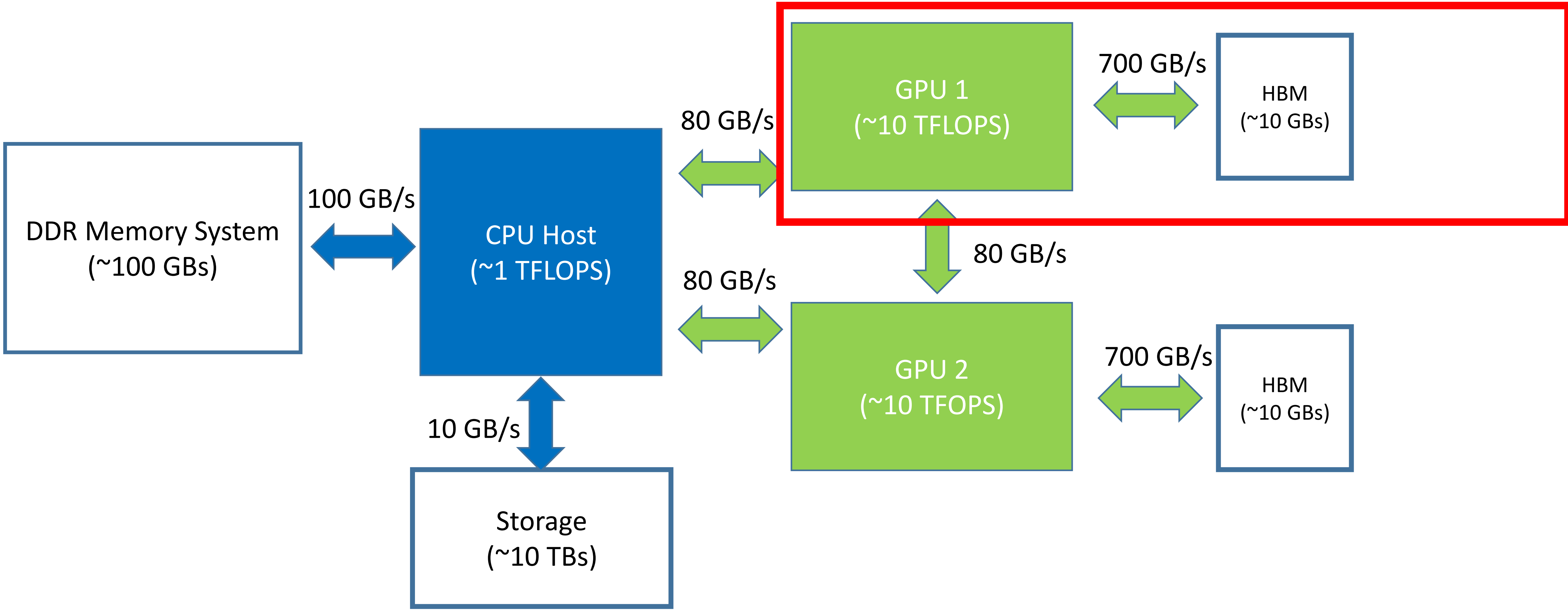
- Applications with large, accurate models
  - Problems that we know how to solve accurately but choose not to because it would be “too expensive”
  - High-valued applications with approximations that cause inaccuracies and lost opportunities
  - Medicate imaging, remote sensing, earthquake modeling, weather modeling, precision digital manufacturing, combustion modeling, ….
- Applications that we have failed to program
  - Problems that we just don’t know how to solve
  - High-valued applications with no effective computational methods
  - Computer vision, natural language dialogs, document comprehension, individualized education, fraud detection, self-driving cars, …

# A Simplified Heterogeneous System
## (IBM Minsky with NVIDIA Pascal GPUs)

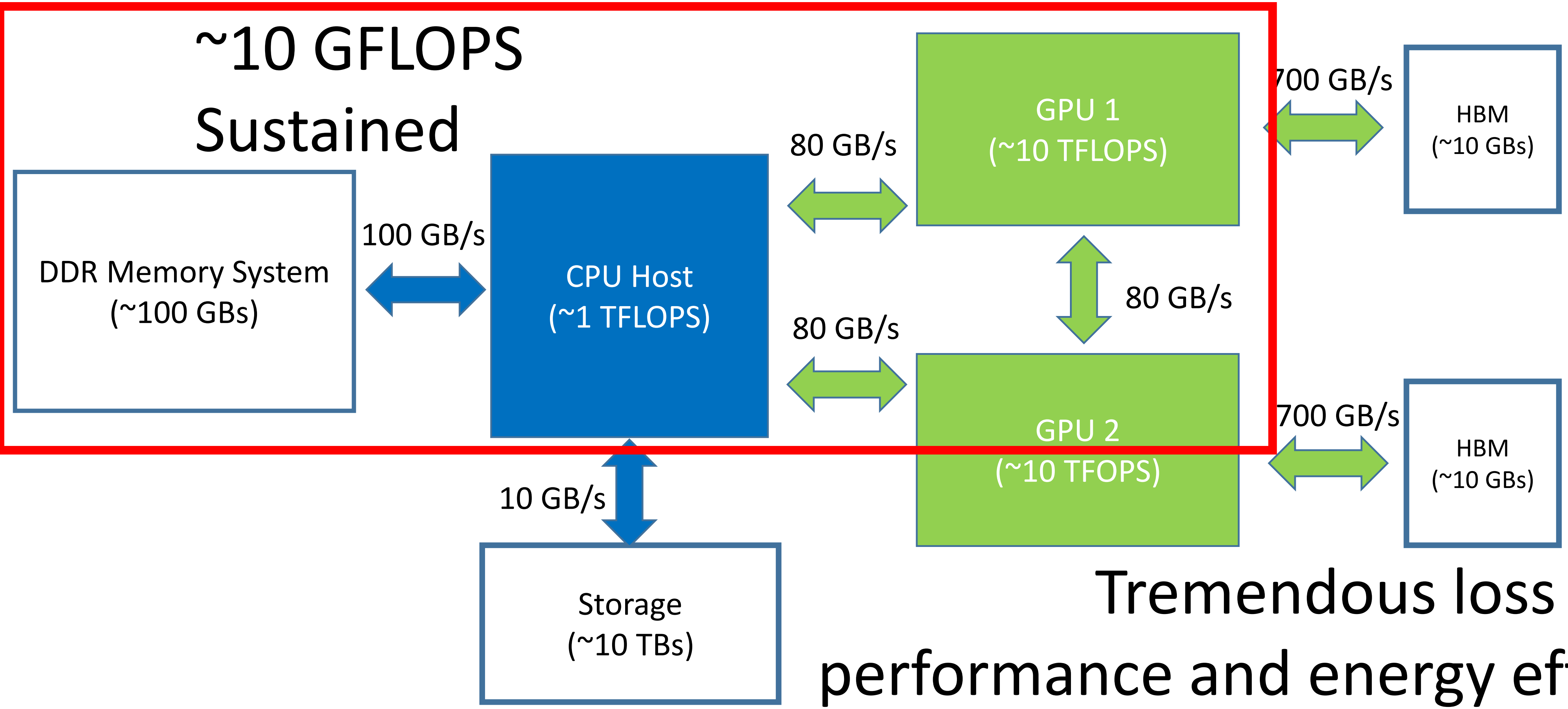


# Iterative Solver Example – If matrix fits into GPU Memory

~100 GFLOPS Sustained

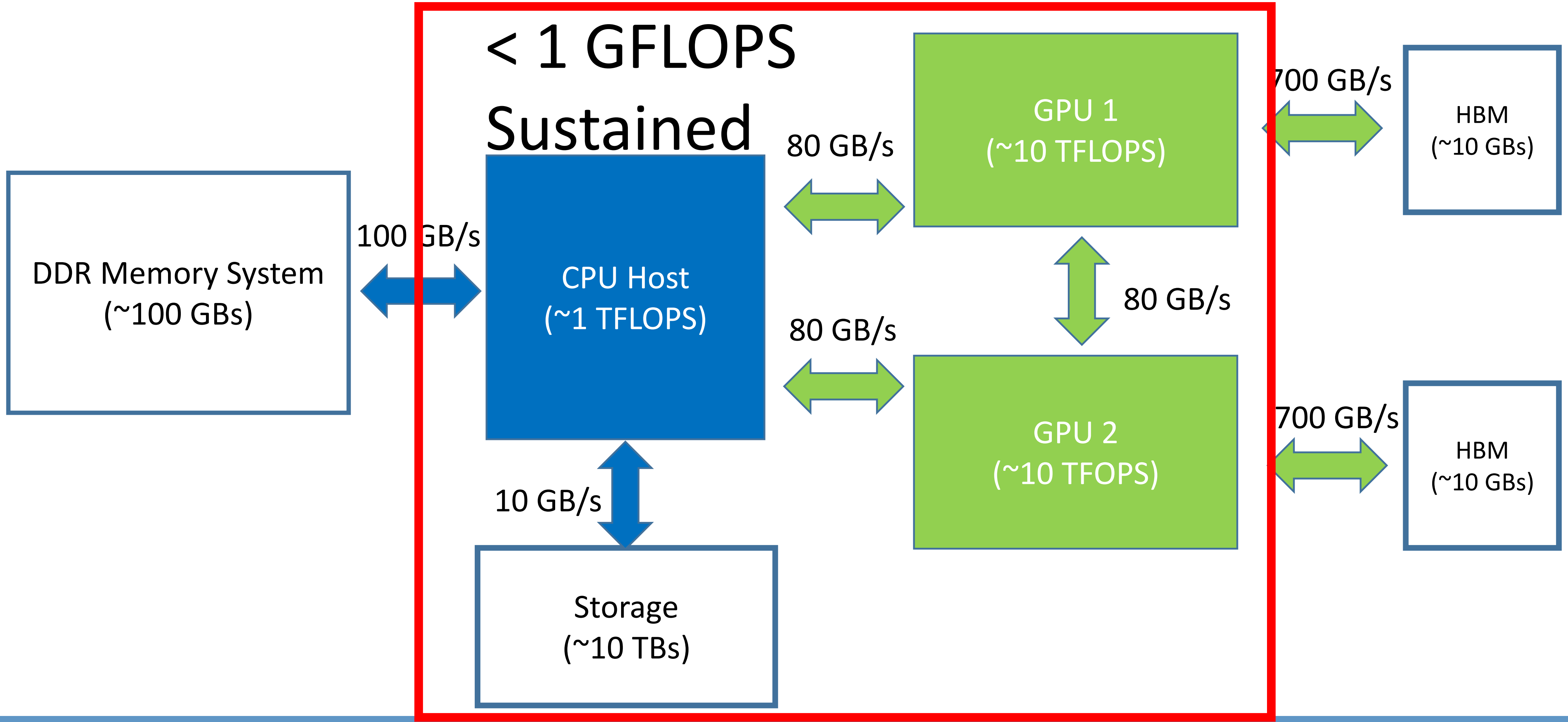


# Iterative Solver Example –
# If matrix fits into Host Memory




Tremendous loss of both performance and energy efficiency

# Iterative Solver Example –
# If matrix has to be accessed from storage



# Problem Statement for the Erudite Project

- Latency and bandwidth limitations on accessing massive data sets
  - Sweeping through large data sets brings systems to their knees
  - Low data reuse creates unnecessary traffic through the memory hierarchy
  - Sustained performance < 1% of peak for memory/storage bound applications

- Large software overhead for data access
  - File system overhead and bottleneck
  - Message passing serialization/deserialization and layers of constructors
  - Excessive data copying between memory address spaces and subspaces

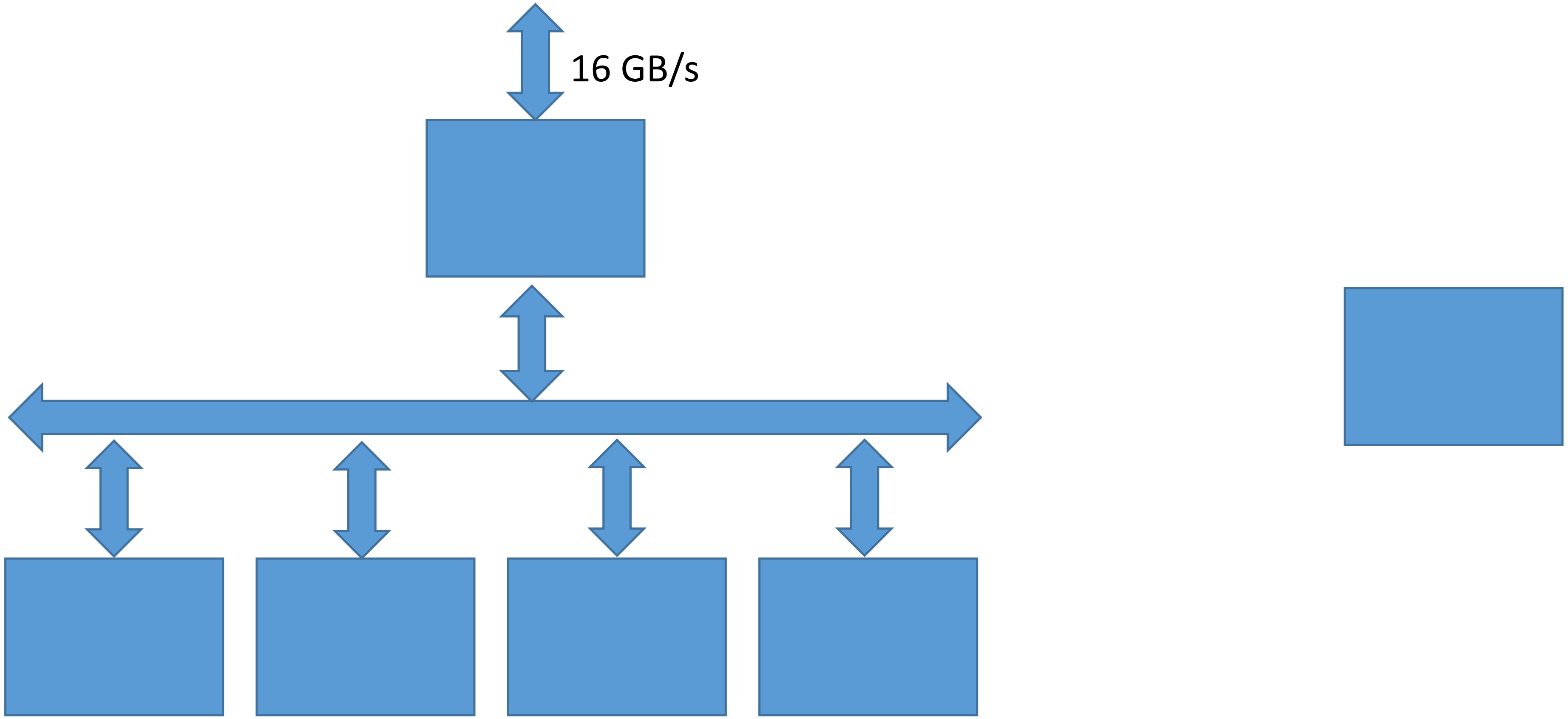
# Example: Direct vs. Iterative Solvers

## Direct Solvers

- Good Locality
  - Data reuse through tiling
- Sparsity
  - Too many fill-ins, data explosion
- Stability
  - Pivoting restricts parallelism

## Iterative Solvers

- Poor Locality
  - Multiple sweeps through matrix
- Good with Sparsity
  - No fill-ins during solution time
- Stability
  - Convergence varies
  - Preconditioning may enlarge matrix


16 GB/s