Hybrid Threaded Processing for Sparse Data Kernels

Tony Brewer
Chief Architect, Advanced Computing Solutions
May 9, 2018

"This research was, in part, funded by the U.S. Government. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

Micron®

The Challenge

- Sparse data sets that greatly exceed a processor's cache size are a challenge for most systems
  - Processor's are typically optimized for high cache hit rate (>90%)
    - Low cache hit rate results in idle cores
  - Memory accesses are cache line size (64B)
    - Sparse data sets result in memory accesses where the majority of accessed data is not used

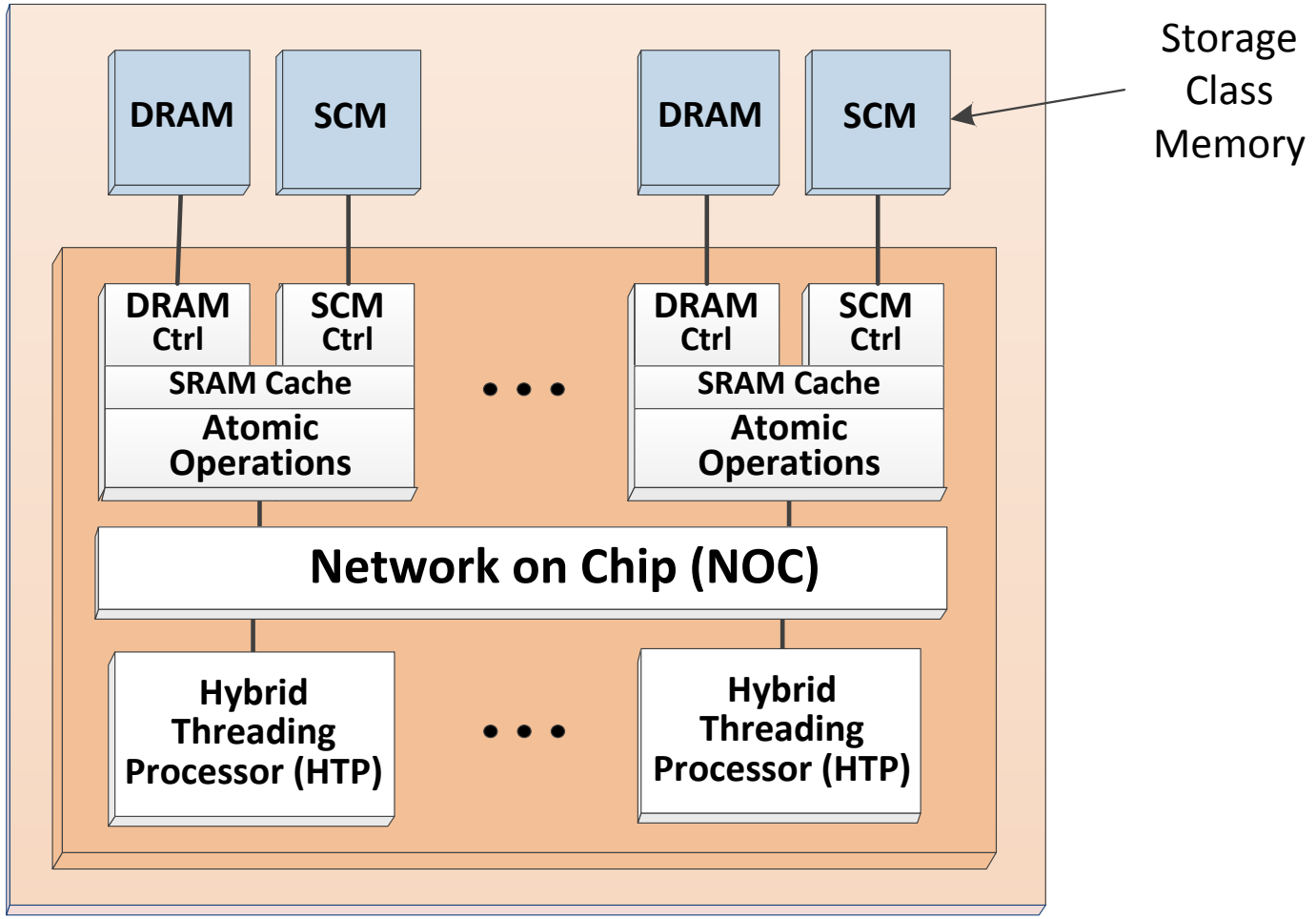
Hybrid Threading Processor (HTP)

- RISC-V ISA (RV64G)
  - Extensions for thread and message management
- High thread count barrel processor
  - Similar to Cray's MTA architecture
  - One instruction per thread per scheduling interval (avoids register hazard checking)
- Event driven processor
  - Pause for memory response
  - Pause for thread join
  - Pause for message reception
- Efficient memory usage
  - Memory access size 8, 16, 32 or 64B
- Software managed coherency
  - Small cache per thread
  - Atomics performed at memory
- User space only
  - Host processor provides system support
- Standard GCC compiler
  - Runtime provides access to new instructions

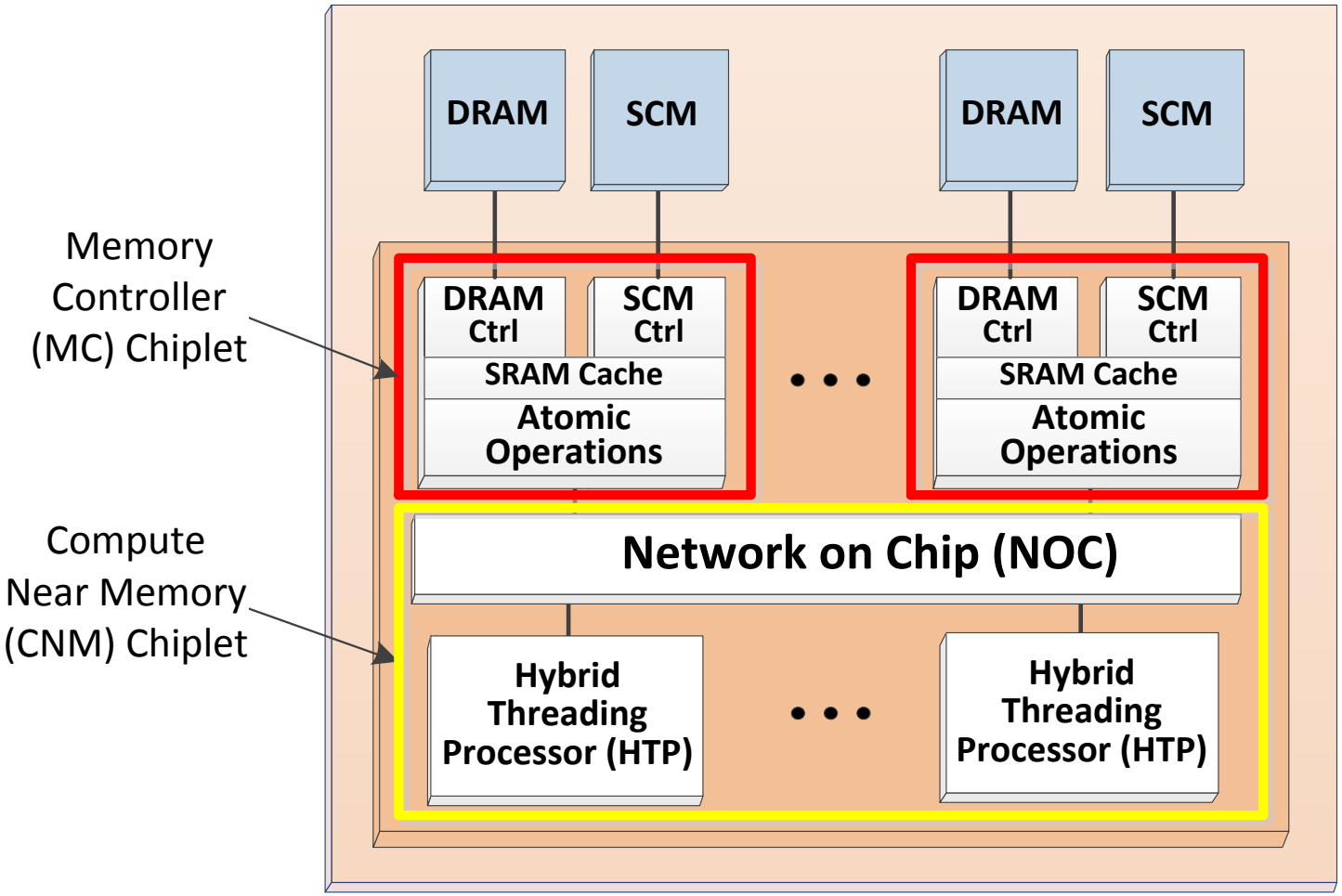
New RISC-V Instructions

- Thread Management
  - Thread Create, Return, Join
- Message Management Instructions
  - Message Send, Broadcast, Receive, Listen
- Non-Cached Loads and Stores
  - Integer and Float

System Architecture



System Architecture

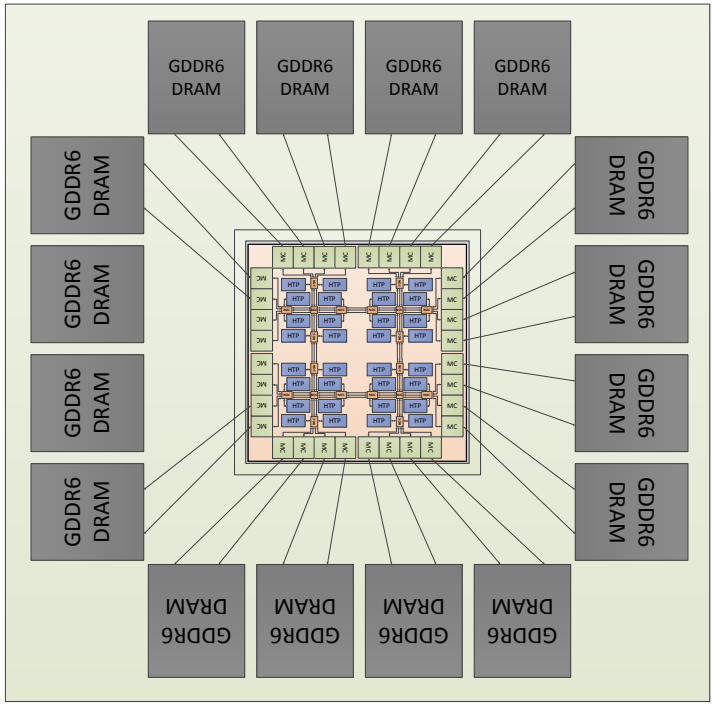


DARPA CHIPS

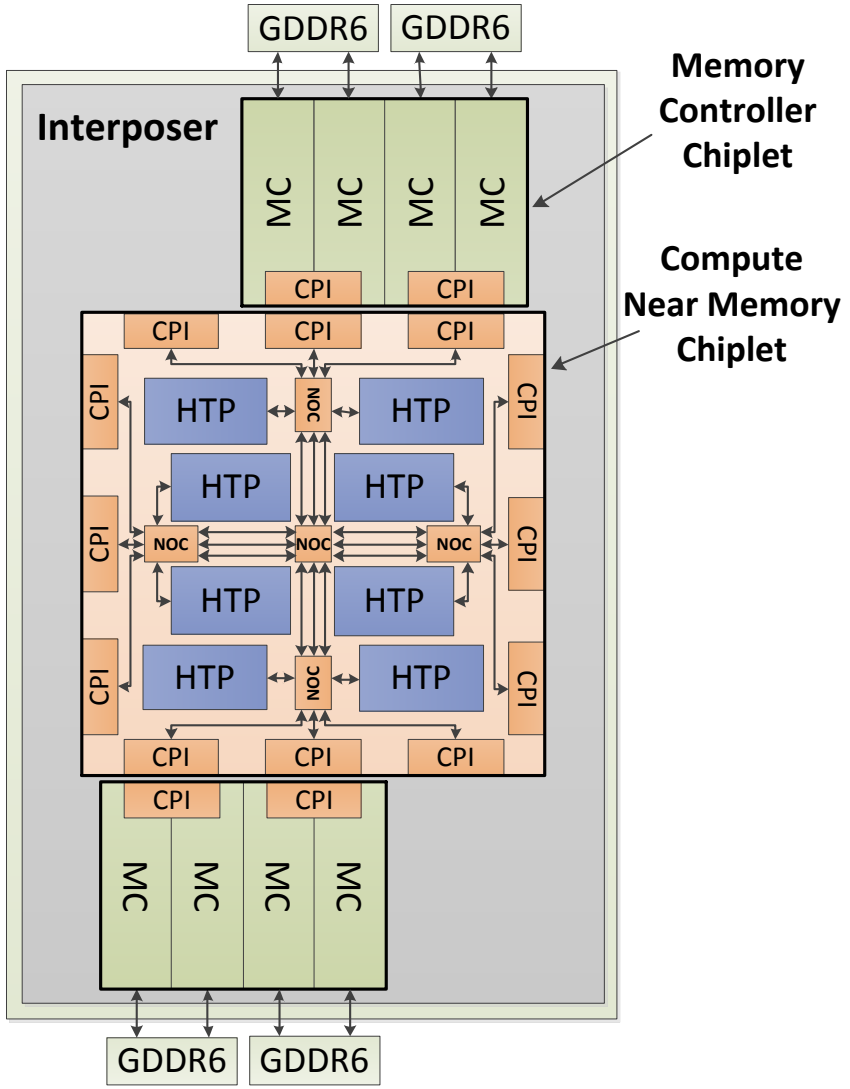
Modeled Configurations

- Focused on two primary configurations:
  - 1x1 CNM Config.
    - 1 CNM Chiplet
    - 2 MC Chiplets
    - 4 GDDR6 Memories
  - 2x2 CNM Config.
    - 4 CNM Chiplet
    - 8 MC Chiplets
    - 16 GDDR6 Memories

2x2 CNM Configuration



1x1 CNM Configuration



Performance and Power

- Simulator models functionality and performance
  - Clocked simulation model
  - Models functionality
  - Models data paths, arbitration and queueing
- Power Estimation Methodology
  - Break design into major components: NOC, HTP, MC, GDDR6
  - Identify all ram structures, ALUs, I/O, long signal runs (NOC), etc.
  - Determine power through foundry power estimation tools
  - Determine application activity factors through simulation
  - Determine power for each chiplet and total solution power

Graph Spectral Clustering

GRAPH COMMUNITY DETECTION USING SPECTRAL METHODS

- Community detection using spectral methods uses linear algebraic to compute eigenvalues for the adjacency matrix associated with a graph. The lowest eigenvalues can be used to partition the graph.
- Sparse data structures store the graph vertices, edges and properties.

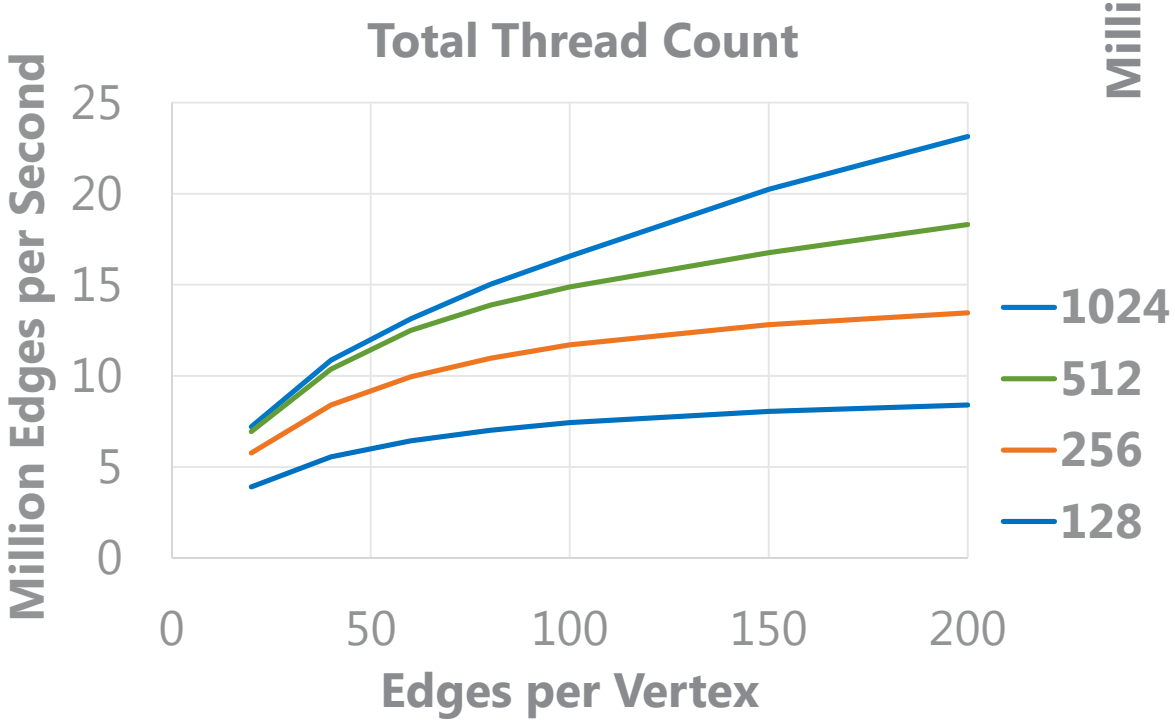
- Profile on an X86 system

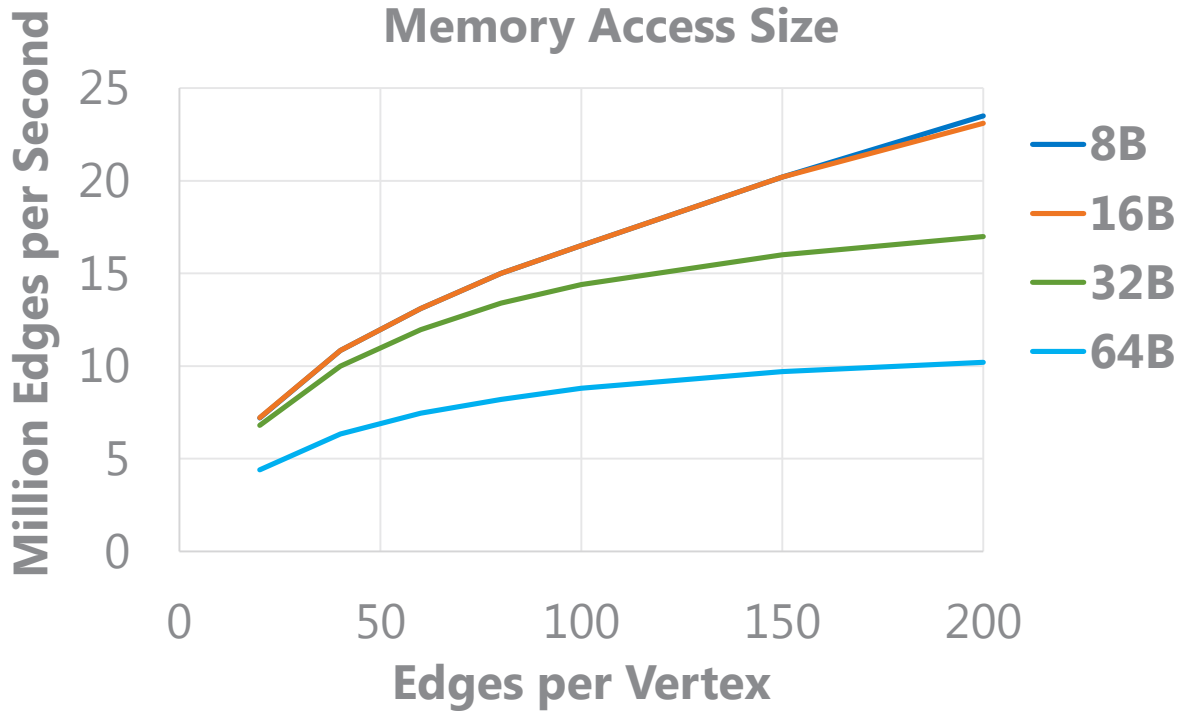
Overhead	Symbol
13.82%	[.] svd_ATxb
13.36%	[.] svd_ATxb2
10.70%	[.] svd_Axb
10.01%	[.] substruct
9.93%	[.] svd_Axb2
6.59%	[.] _IO_vfscanf

Graph Spectral Clustering

SENSITIVITY ANALYSIS

- Sensitivity analysis to determine optimal configuration



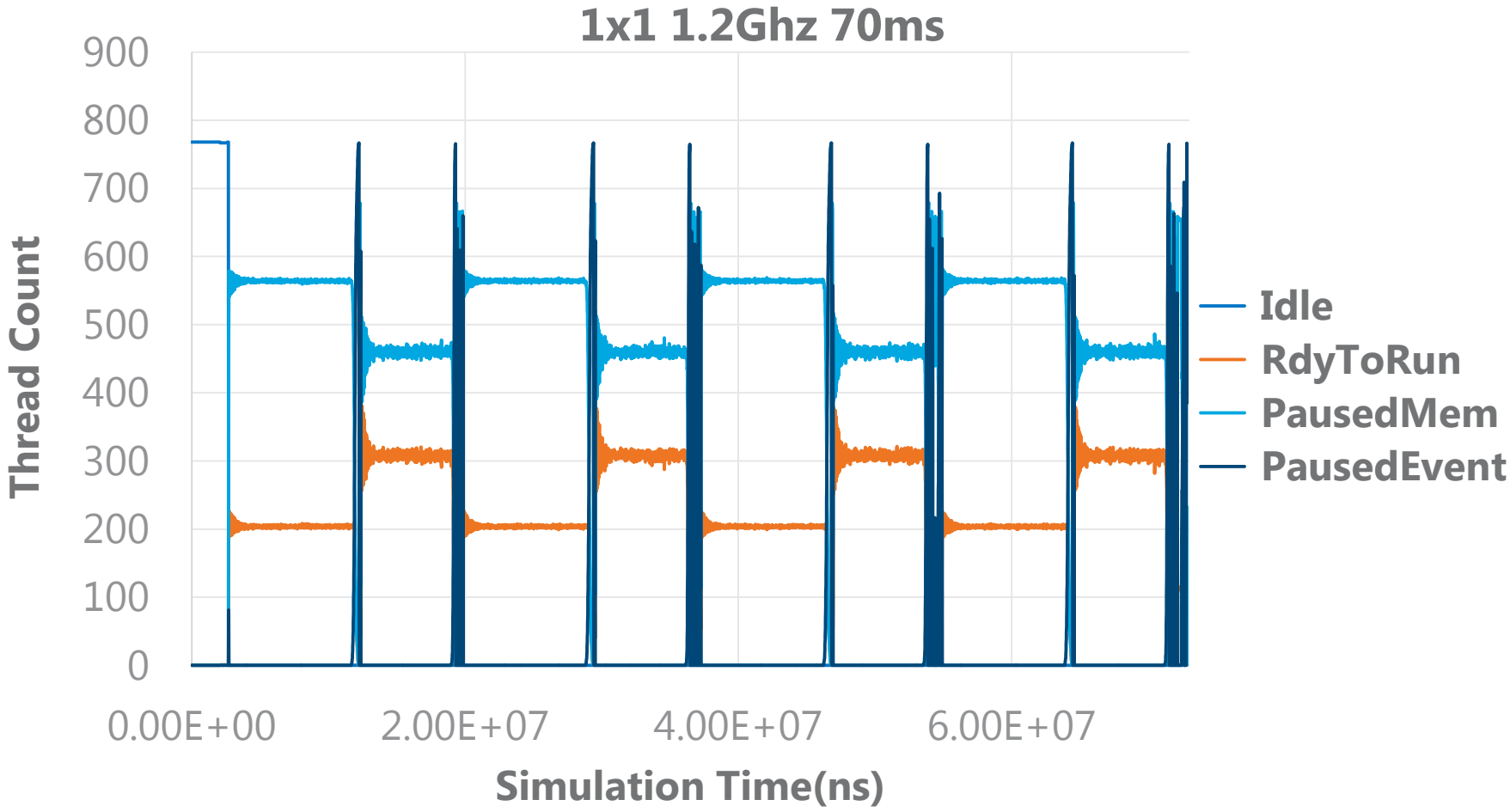


- Other parameters
  - Clock rate
  - Cores per HTP processor

Graph Spectral Clustering

THREAD STATE MONITORING

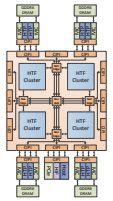
- Provides insights into run time dynamics of application

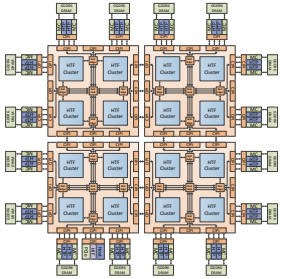


Graph Spectral Clustering

COMPARISON TO REFERENCE PLATFORMS

Haswell 8-threads 1-socket	Nvidia K80 (Host + 1 GPU)	Nvidia DGX-1 (Host + P100) 1 P100 time	NOC 1x1 Config 1.2Ghz Simulated	NOC 2x2 Config 1.2Ghz Simulated
13.5 Sec 140 Watts **1890 Joules** **27.4x**	5.0 Sec[1] 340 Watts **1703 Joules** **24.7x**	3.95 Sec[1] (Note – System was at a cloud provider, no power info)	2.90 Sec 23.8 Watt **69 Joules** **1.0x**	0.814 Sec 90.9 Watt **74 Joules** **1.07x**





Notes:

1. Times reported for GPUs do not include time to copy graph from host to GPU (120 sec).


Micron®