Analyzing Resource Utilization in an HPC System: A Case Study of NERSC's Perlmutter

Lead author: Jie Li, Texas Tech University

Presenter: George Michelogiannakis, Lawrence Berkeley National Laboratory

Co-authors: Brandon Cook, Dulanya Cooray, Yong Chen

Motivating Questions

What do we want to learn?

- How intensely are resources in modern HPC systems used?
  - Focus on GPUs since they are a new resource
- How well are users transitioning to a GPU-accelerated systems?
- Are HPC systems good candidates for resource disaggregation?

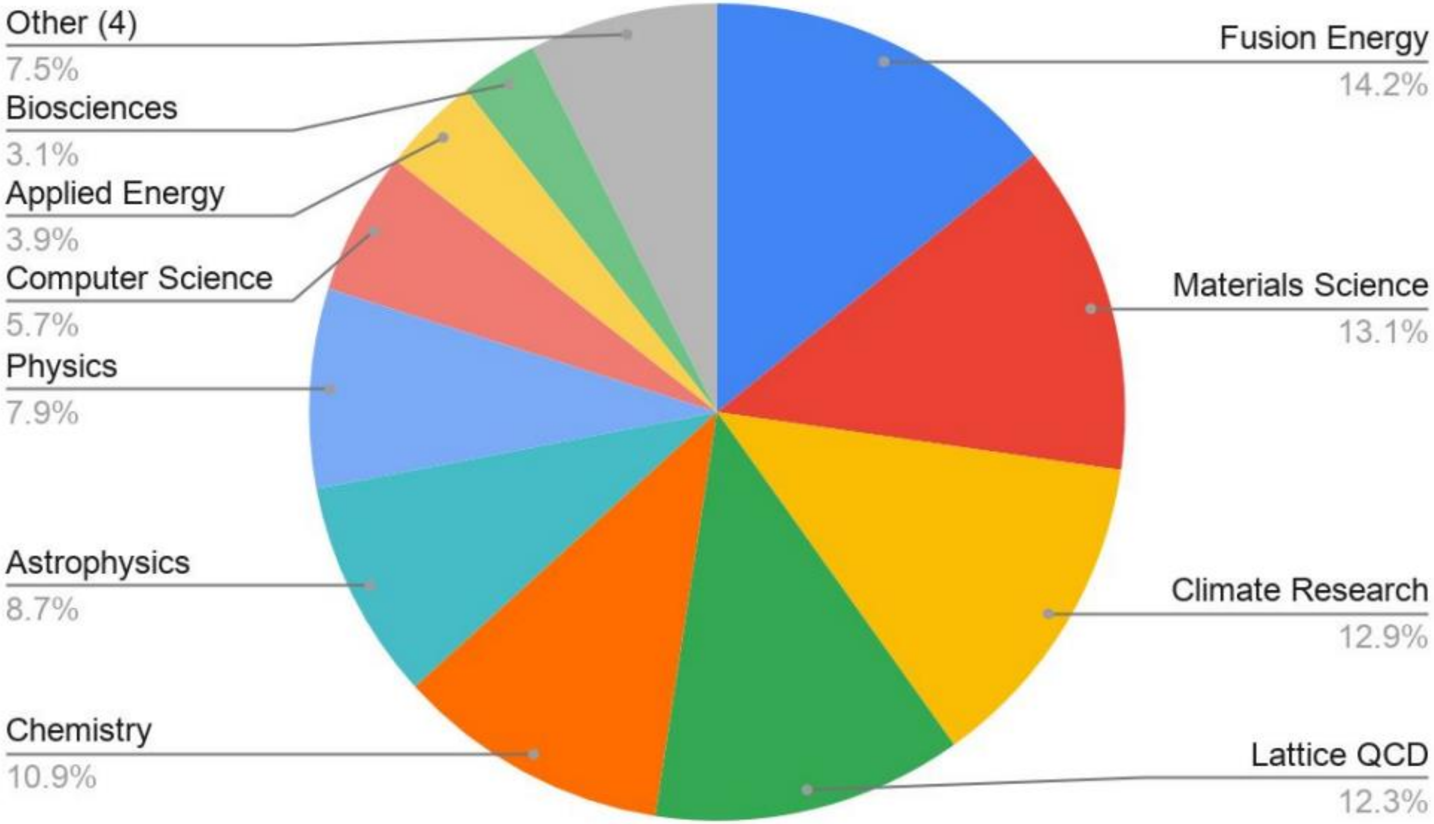
We choose NERSC’s Perlmutter as a representative system

Why Perlmutter?

NERSC's flagship system. Number 8 in top 500 list

- Perlmutter serves an open-science workload
- Perlmutter is the first NERSC system with GPU-accelerated nodes
- Perlmutter offers some key system-wide statistics
- <u>Caveat</u>: Cori was operational in parallel and Perlmutter is not yet fully accepted
  - Therefore, workload may change

High-level view of workload from 2018
Exact CPU and GPU charts in our paper



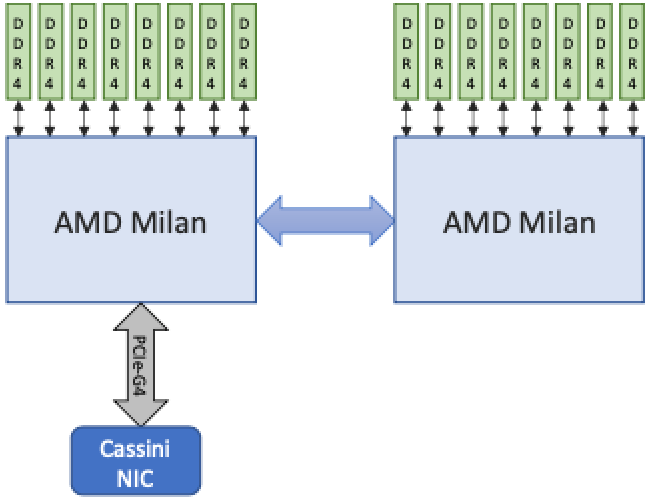
https://portal.nersc.gov/project/m888/nersc10/workload/N10_Workload_Analysis.latest.pdf

Perlmutter's Configuration

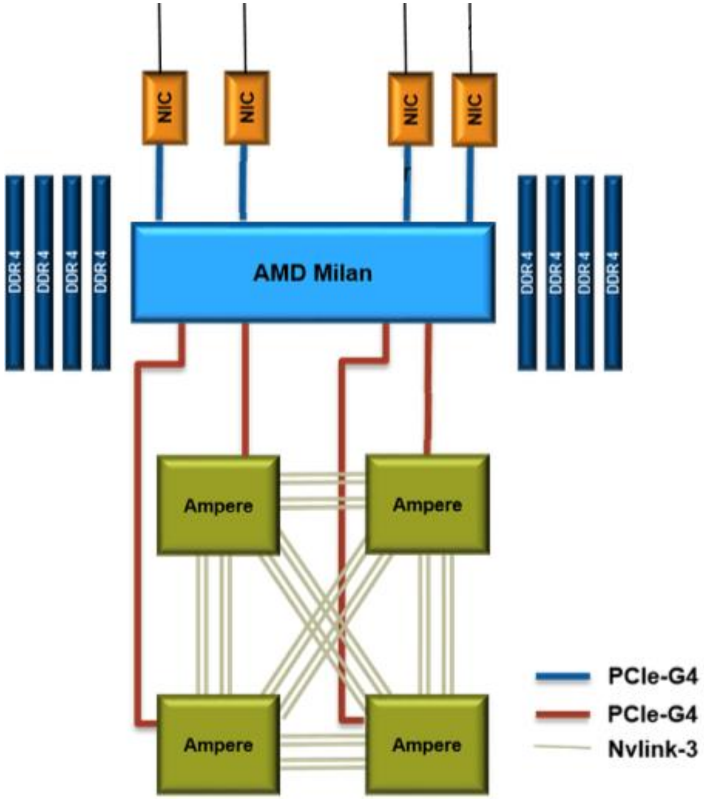
Modern, GPU-accelerated system

- Configuration:
  - 1536 GPU nodes
    - 64 cores per CPU
    - 256 GB DDR4 host DRAM per node
    - 40 GB HBM per GPU
  - 3072 CPU nodes
    - 512 GB DDR4 DRAM per node
  - Slingshot network 11
    - (10 at the time)

CPU Nodes

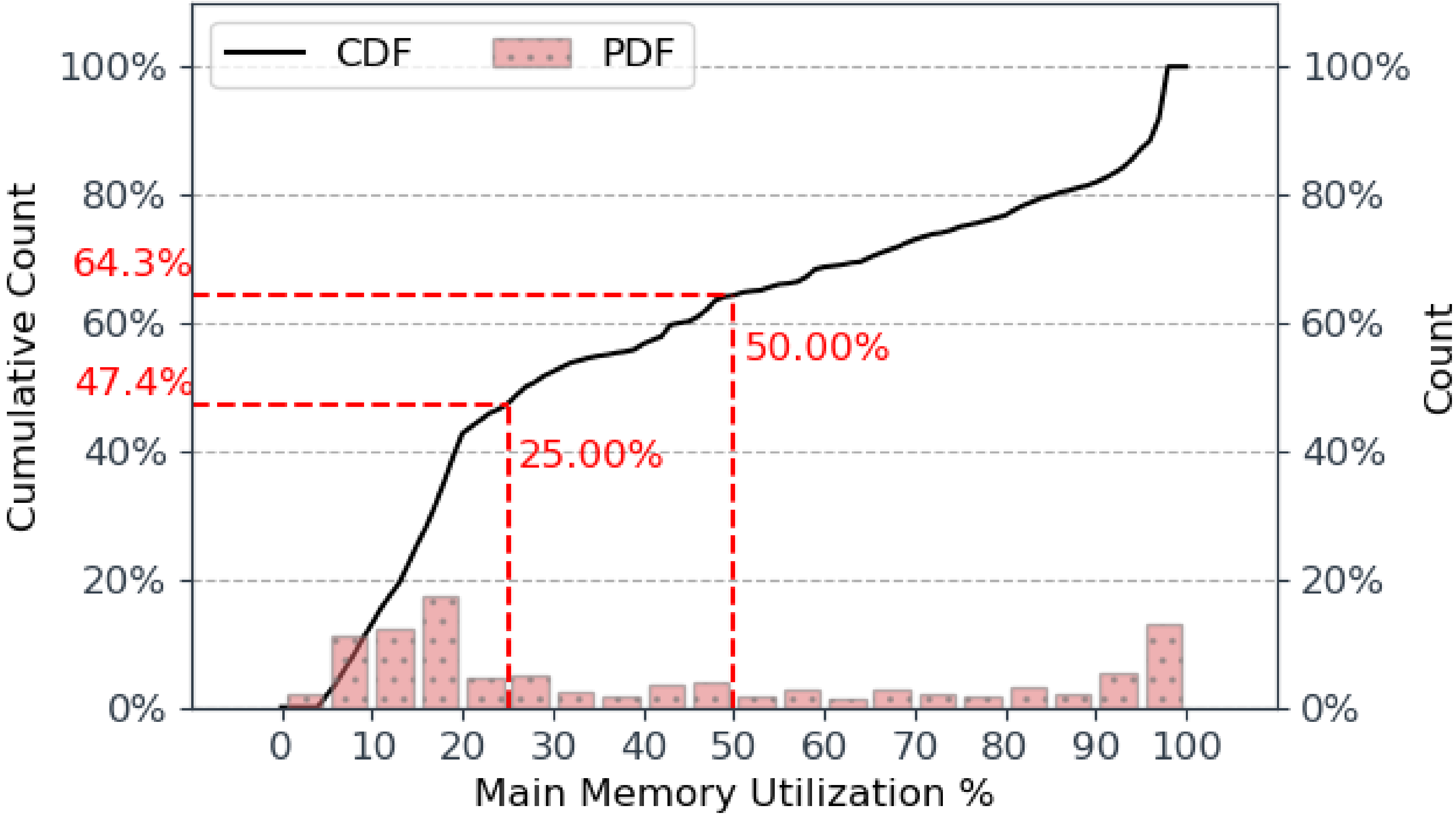


GPU Nodes



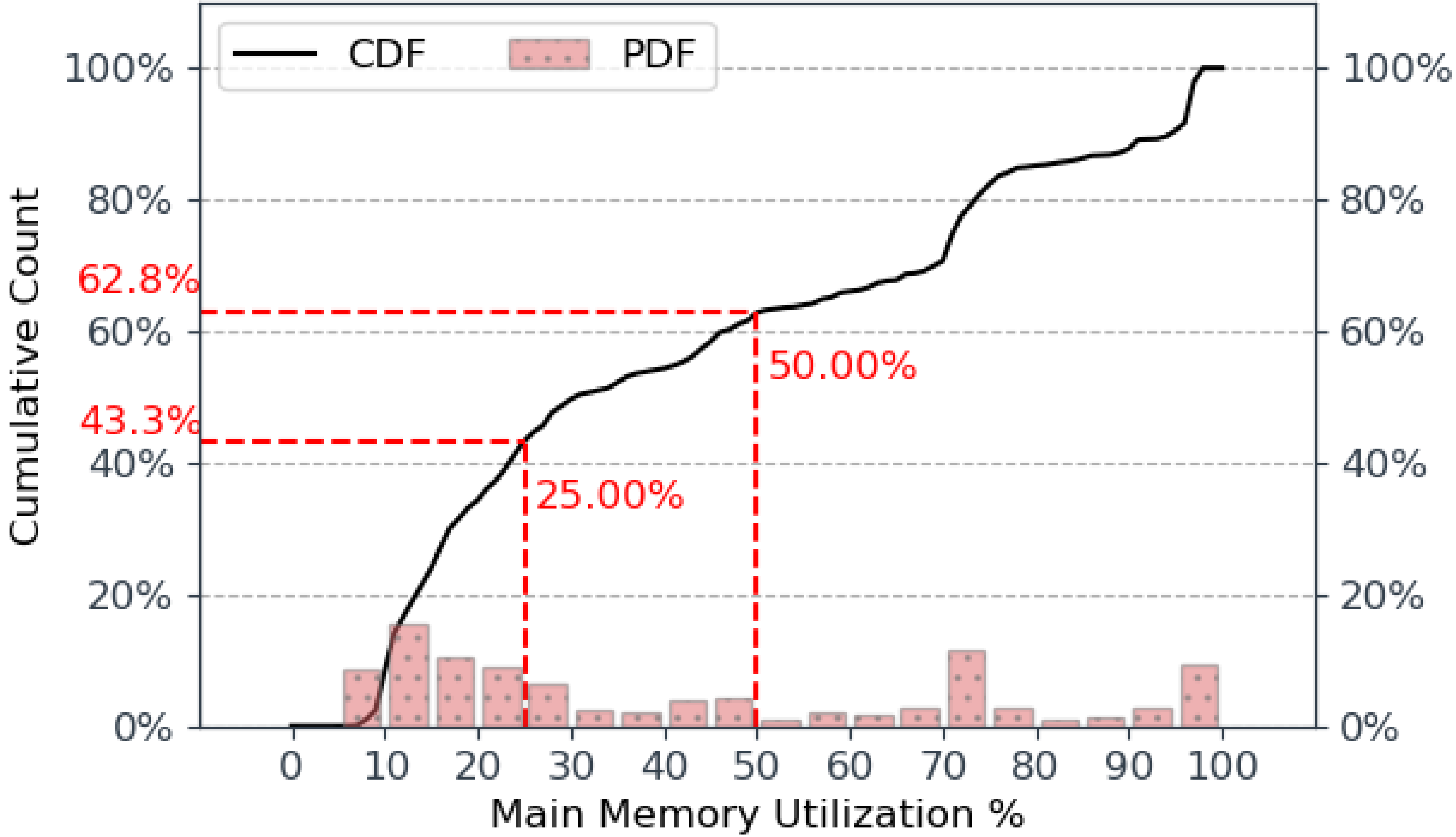
CPU Node Memory Capacity Utilization

Jobs weighed by node-hours
Jobs < 1 hour discarded
Memory capacity is maximum in job's lifetime



GPU Node Memory Capacity Utilization

Jobs weighed by node-hours
Jobs < 1 hour discarded
Memory capacity is maximum in job's lifetime

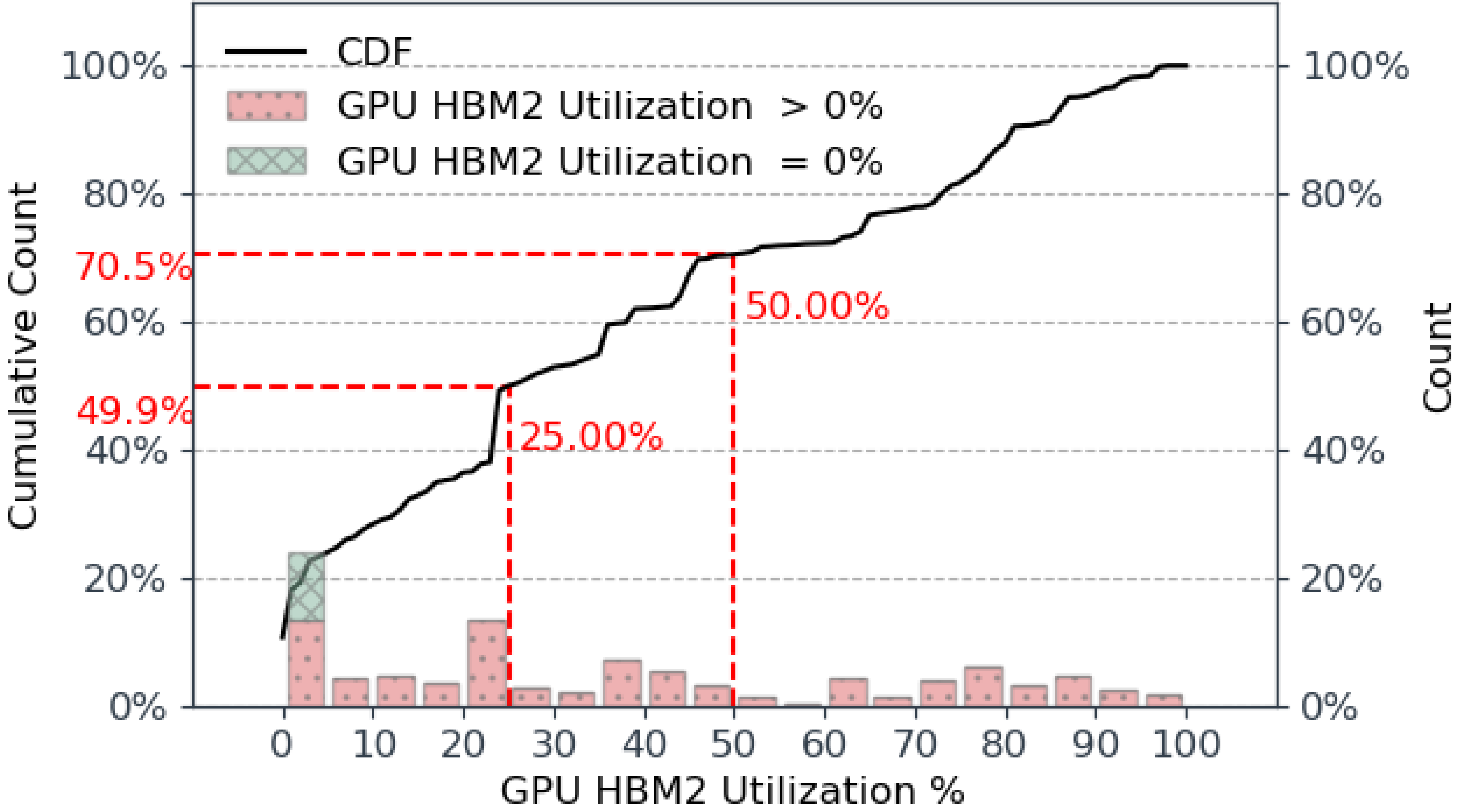


Similar and slightly lower than CPUs

BERKELEY LAB
Bringing Science Solutions to the World

GPU HBM2 Capacity Utilization

Jobs weighed by node-hours
Jobs < 1 hour discarded
Memory capacity is maximum in job’s lifetime



Slightly higher

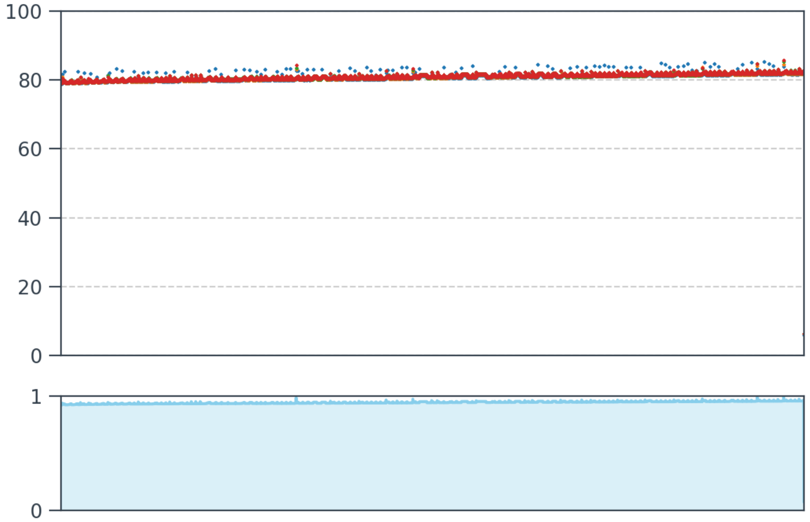
Therefore:

Memory Capacity Underutilized

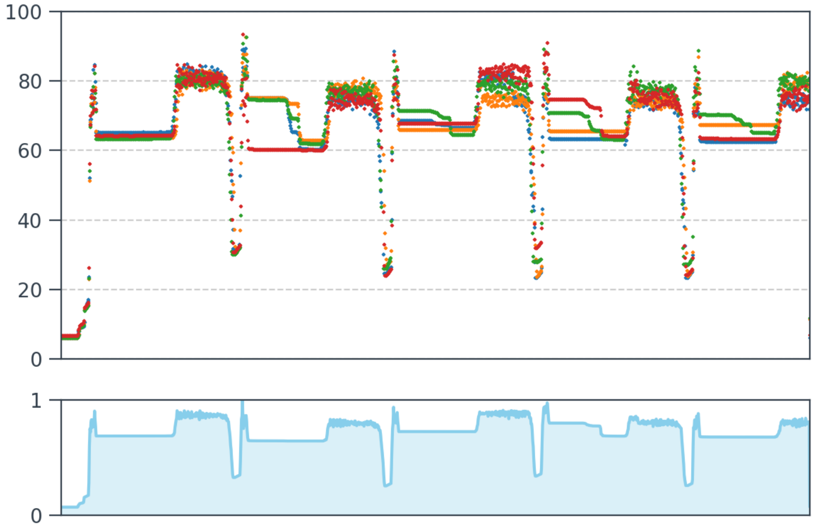
Three Temporal Patterns: Node Memory Capacity (%)

Three example jobs per category. Colors: nodes assigned to a job. Bottom plot is one node

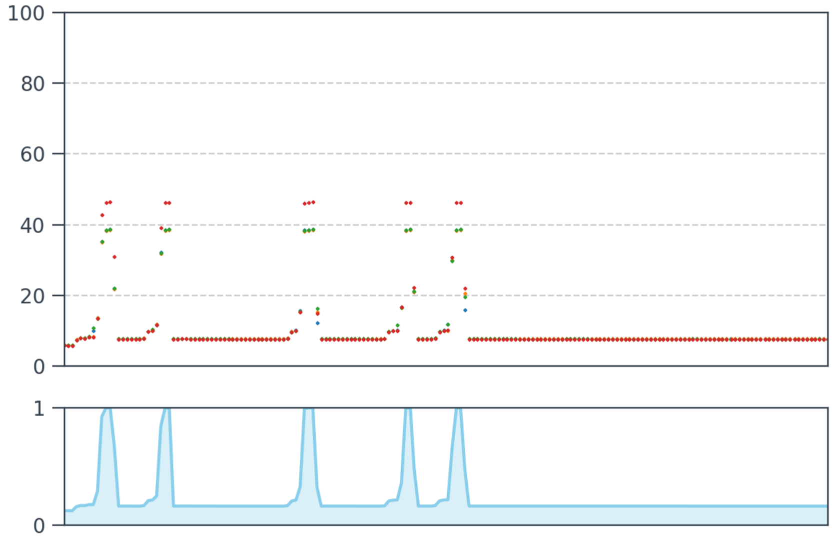
Constant pattern



Dynamic pattern



Sporadic pattern

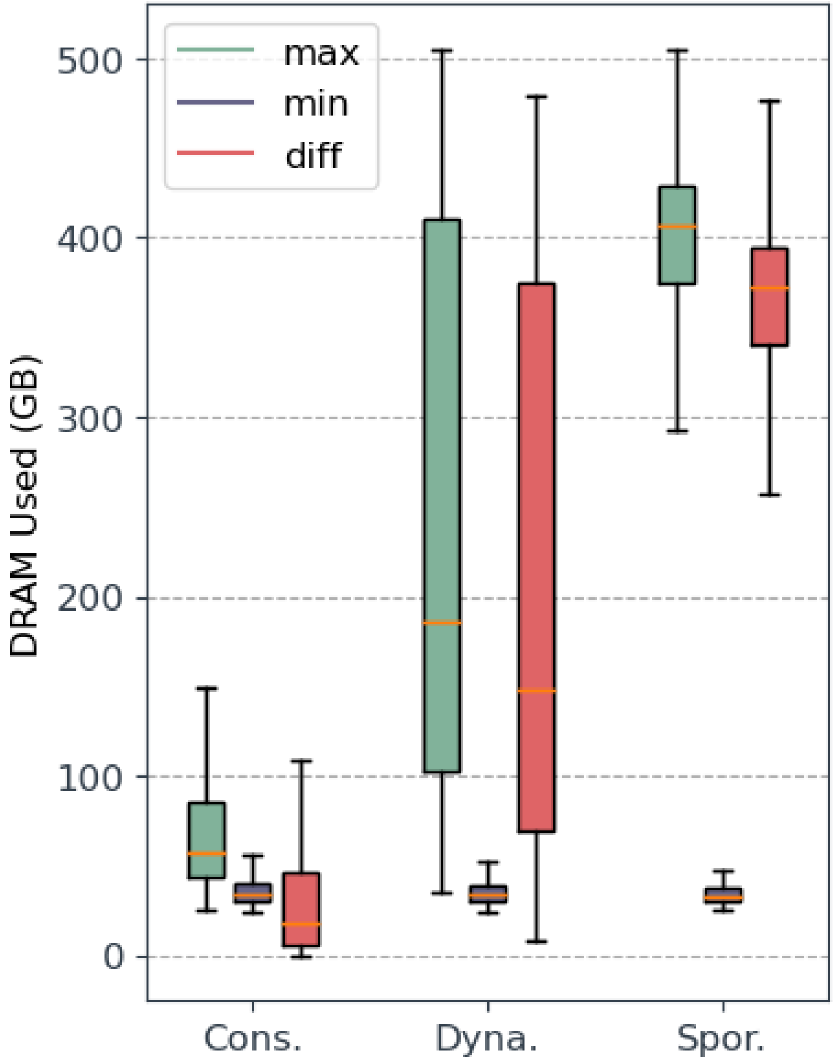


$$RI_{temporal}(r) = \max_{1 \leq n \leq N} \left(1 - \frac{\sum_{t=0}^{T} U_{n,t}}{\sum_{t=0}^{T} \max_{0 \leq t \leq T}(U_{n,t})}\right)$$

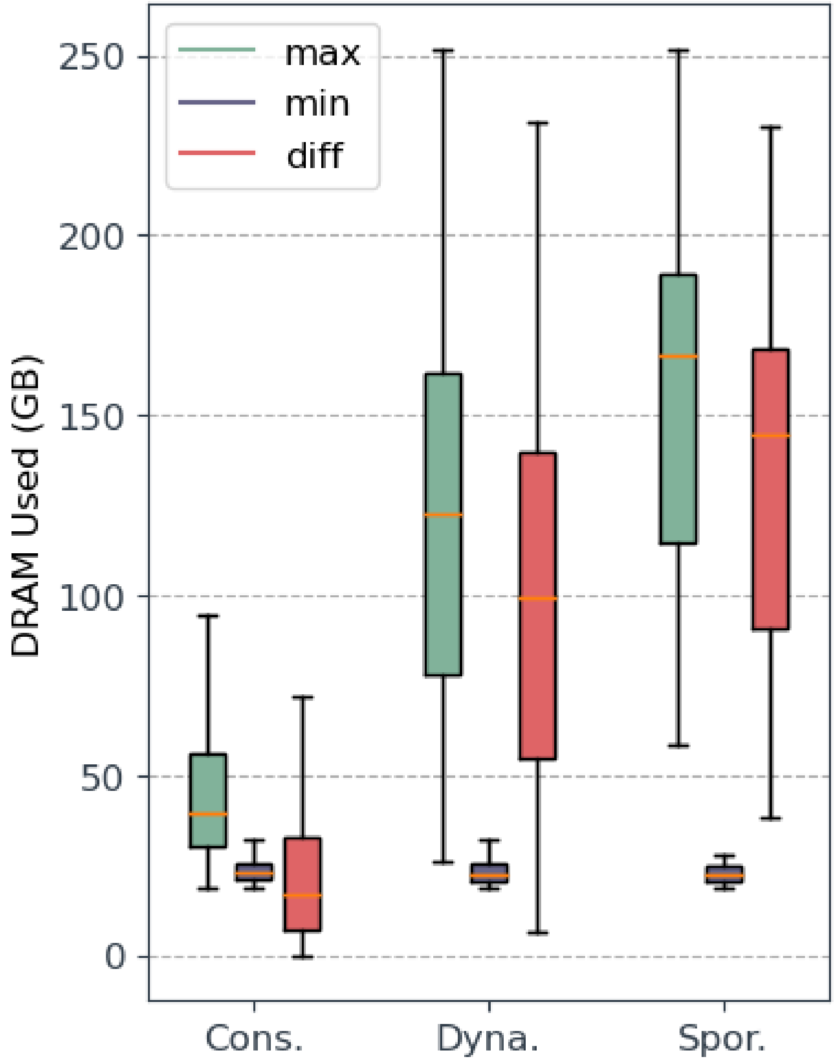
RI < 0.2: Constant
RI between 0.2 and 0.6: Dynamic
RI greater than 0.6: Sporadic

Temporal Distribution By Category

CPU



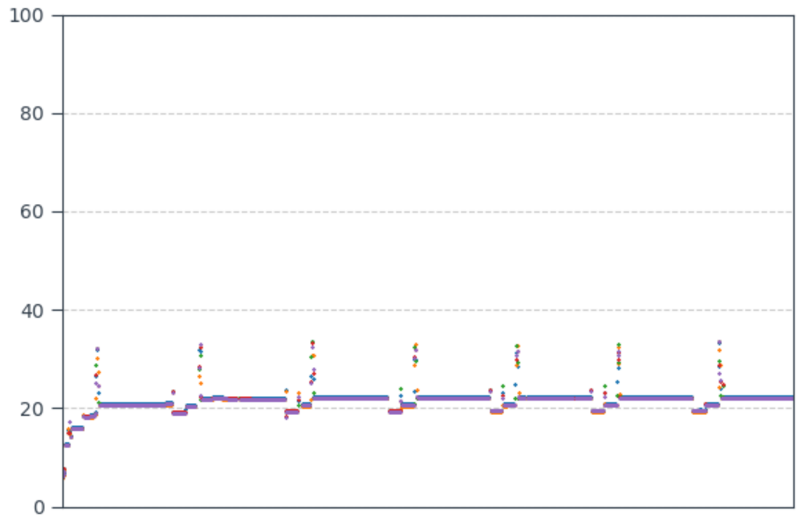
GPU



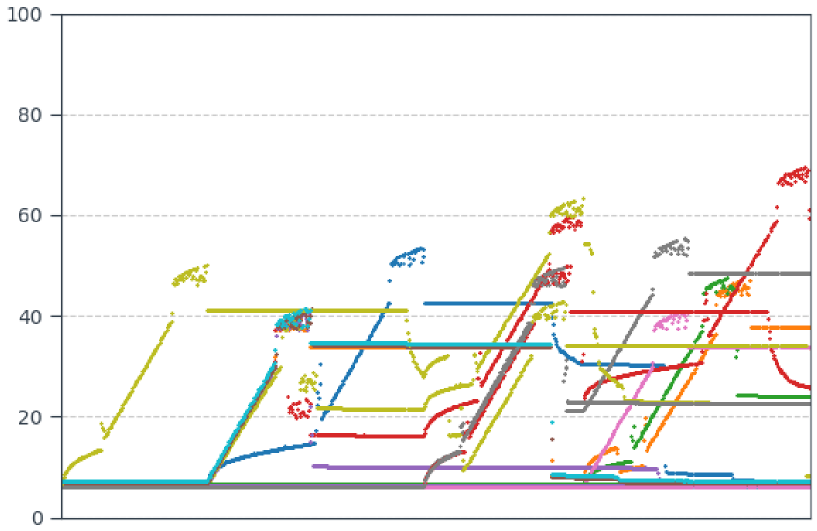
Three Spatial Patterns: Node Memory Capacity (%)

Three example jobs per category. Colors: nodes assigned to a job. Bottom plot is one node

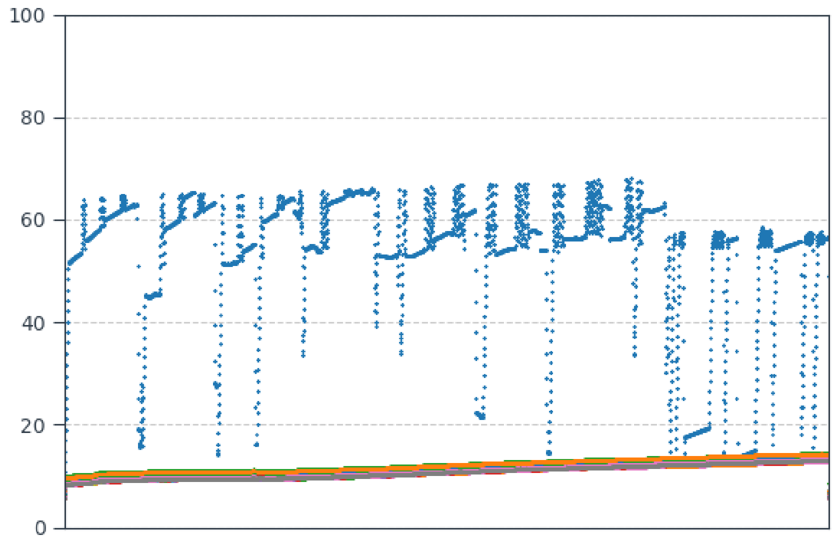
Convergent pattern



Scattered pattern



Deviational pattern

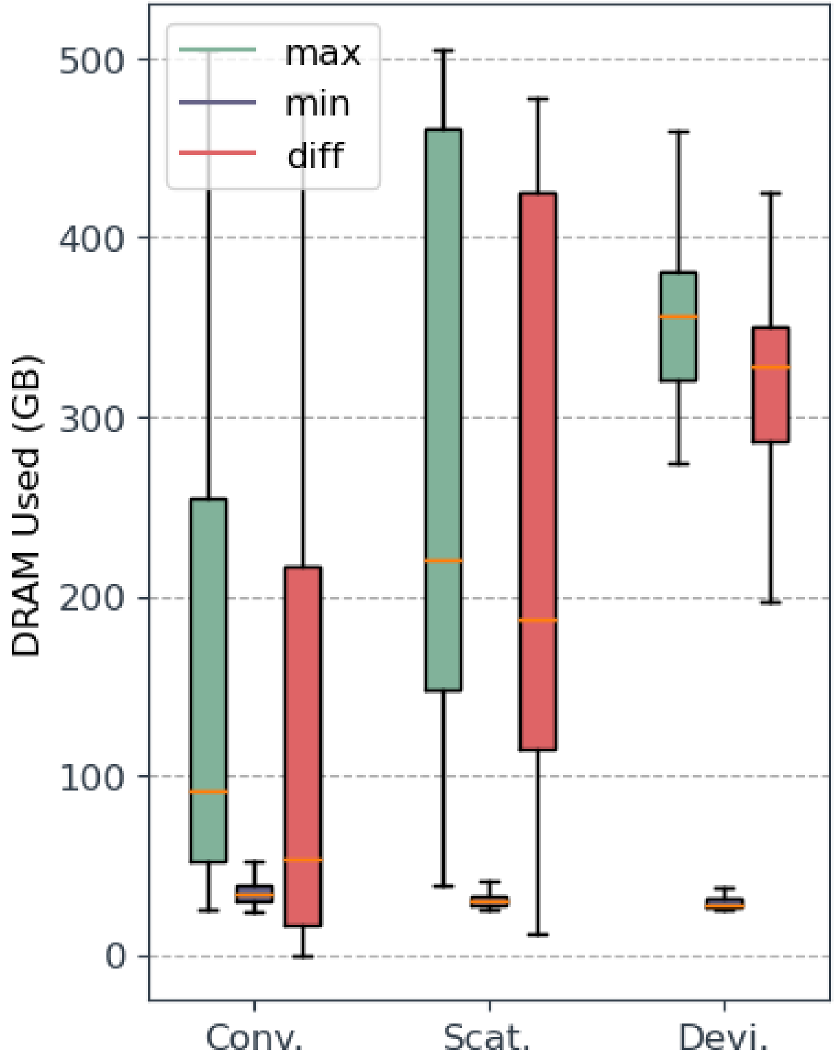


$$RI_{spatial}(r) = 1 - \frac{\sum_{n=1}^{N} \max_{0 \le t \le T}(U_{n,t})}{\sum_{n=1}^{N} \max_{0 \le t \le T, 1 \le n \le N}(U_{n,t})}$$

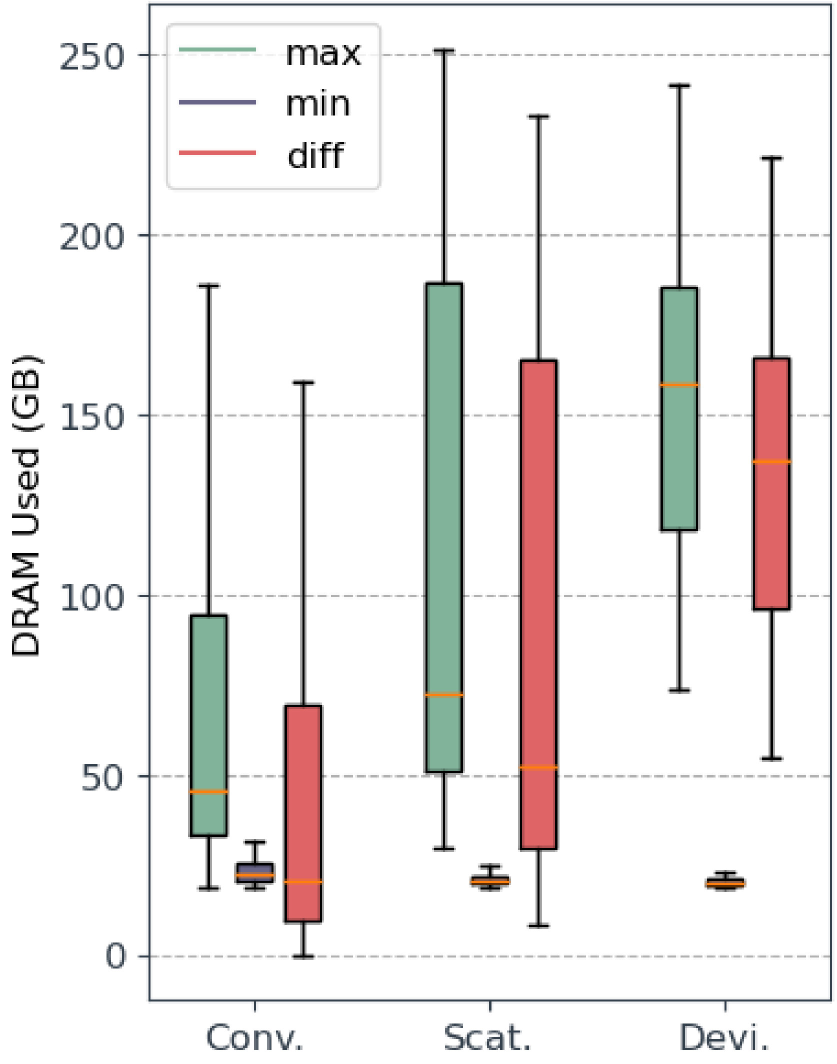
RI < 0.2: Convergent
RI between 0.2 and 0.6: Scattered
RI greater than 0.6: Deviational

Spatial Distribution By Category

CPU



GPU



Takeaways

For details, CPU idle. and metric correlations please see our paper

- Both CPU and GPU jobs have two thirds of jobs that only occupy one node
- GPUs have a higher proportion of short-lived jobs (less than three hours)
- Jobs rarely allocate more than 128 nodes. Majority of jobs fit inside a Perlmutter rack
- GPU jobs use less host memory capacity than CPU jobs
  - 10.6% of GPU hours use no HBM2 capacity
- Jobs with higher temporal imbalance generally have a higher maximum memory capacity
  - Memory capacity not fully utilized for constant pattern jobs
- Jobs have generally good spatial balance

Questions?

Contact:

- mihelog@lbl.gov
- Jie.Li@ttu.edu