Lecture 25: Miscellaneous Topics in Multicore Parallel Programming

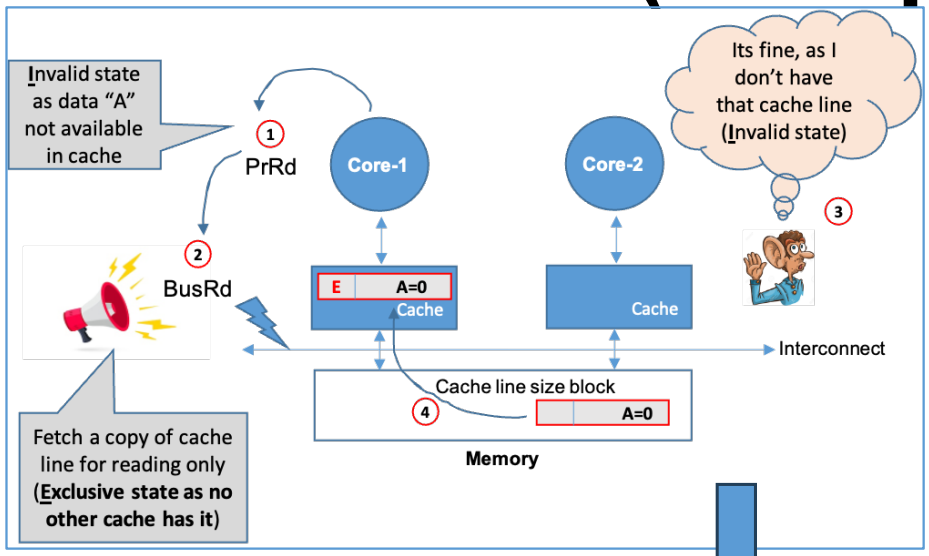
Vivek Kumar

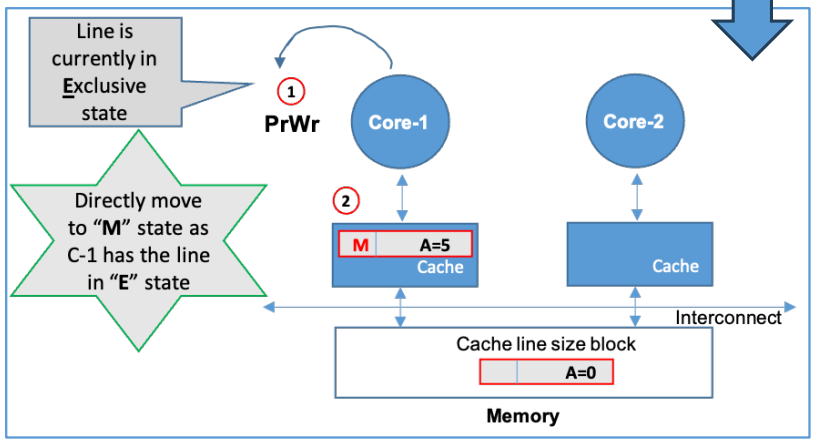
Computer Science and Engineering

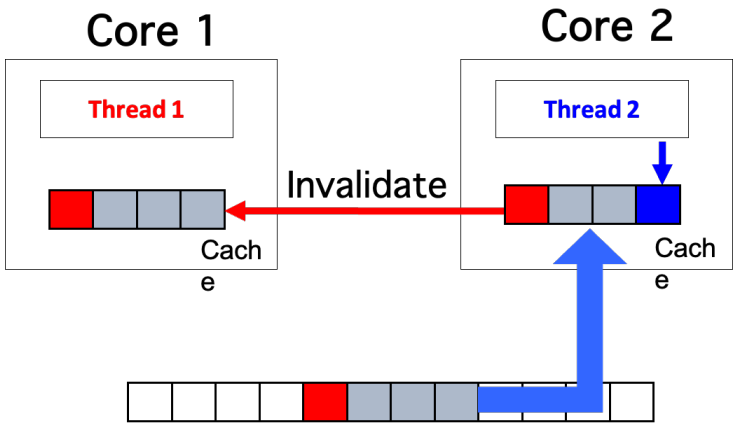
IIIT Delhi

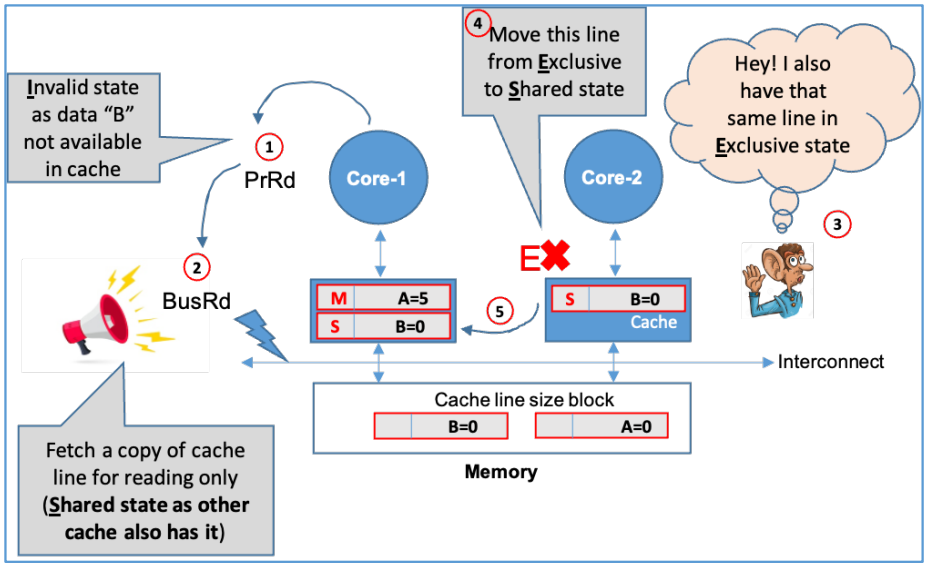
vivekk@iiitd.ac.in

Last Lecture (Recap)









- MESI protocol
- False sharing

Today's Class

- Achieving reliability
- Co-running applications v/s batch execution
- Power capping
- Quiz-5

Exascale Computing Era

Increasing number of sockets and cores per server

Rank of Top500 (November 2025)	Sockets Per Node	Cores Per Node
1	**4**	96
2	1	64
3	2	104
4	**4**	**288**
5	2	96

1. https://top500.org/lists/top500/

- 1 exaflop is 10^{18} floating point operations per second
- Assuming total human population on earth is about 8 billion
  - If each person does 125 million calculations per seconds then together they would roughly equal to 10^{18} operations/sec

Key Challenges for Exascale

- Parallelism
  - Covered in depth in first half of this course
  - **Goal**: Support applications solving science problems 50× faster or more complex than today's 20 PF systems
- Memory and Storage
  - We covered NUMA and locality in context of the memory, but we are not covering storage in this course
  - **Goal**: Reduce memory access latency and support locality over deep memory hierarchies
- Energy Consumption
  - Covered in lecture 21
  - **Goal**: Operate in a power envelope of 20–30 MW
- Reliability
  - **Goal**: Be sufficiently **resilient**

Resilience

- It is the technique for keeping applications running to a correct solution in a timely and efficient manner despite underlying system faults
  - Exascale systems are 1,000 times more powerful will have at least 1,000 times more components and will fail 1,000 times more frequently

Approaches for Resilience (1/3)

- Checkpointing
  - Rollback recovery approach using checkpoint/restart
  - The programmer adds some specific functions in the application to save essential state and restore from this state in case of failure
  - **Drawback:**
    - Amount of data/space needed for saving intermediate state

Approaches for Resilience (2/3)

- Forward recovery
  - In some cases, the application can handle the error and execute some actions to terminate cleanly or follow some specific recovery procedure without relying on classic rollback recovery
    - Any example?
      - ```
        do {
            try {

            } catch (IOException e) {
                  /* Take action to overcome the error and continue */
            }
        } while(condition not met);
        ```
  - **Drawback**: A prerequisite for rollforward recovery is that some application processes and the runtime environment stay alive
    - In the above example, the JVM could stay alive to complete the checkpointing upon faliure
        ```

Silent Data Corruption in Multicore Servers

- 2 x 3 = 7?

- Silent data corruption (SDC) occurs due to erroneous bit flip
  - Often caused by high-energy particles that physically interact with the silicon. These are generally split into two categories
    - External (cosmic rays creating neutrons and bombarding the nucleus of the silicon atom inside the CPU transistor)
    - Internal (trace radioactive elements in some processor elements)
  - Ageing or overuse of the hardware

Approaches for Resilience (3/3)

- Replication (ideal for mitigating SDC)
  - Each process is replicated such that the probability that all replicas would fail is acceptably small
  - Replicas of a process are assigned to different computers
  - They proceed asynchronously with the same code and data such that they can be viewed as an integrated logical entity by others
  - **Drawback**
    - Amount of computational resources is a major challenge
      - Usually double the number of resources actually required by the program

Today's Class

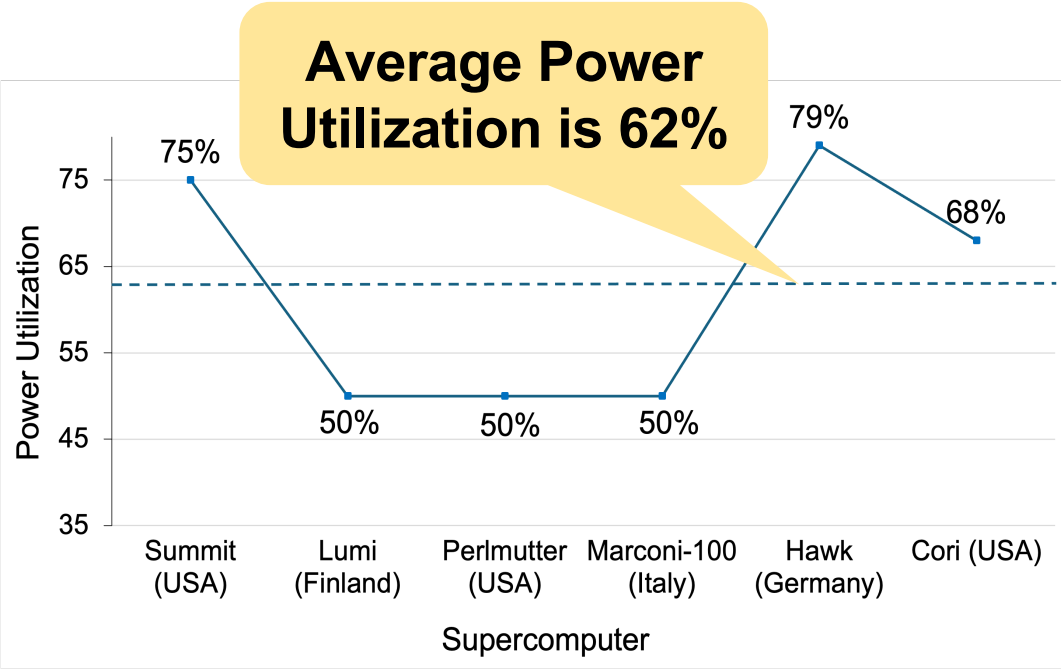
- Achieving reliability
- Co-running applications v/s batch execution
- Power capping
- Quiz-5

Exascale Computing Era

Increasing number of sockets and cores per node

Rank of Top500 (November 2025)	Sockets Per Node	Cores Per Node
1	**4**	96
2	1	64
3	2	104
4	**4**	**288**
5	2	96

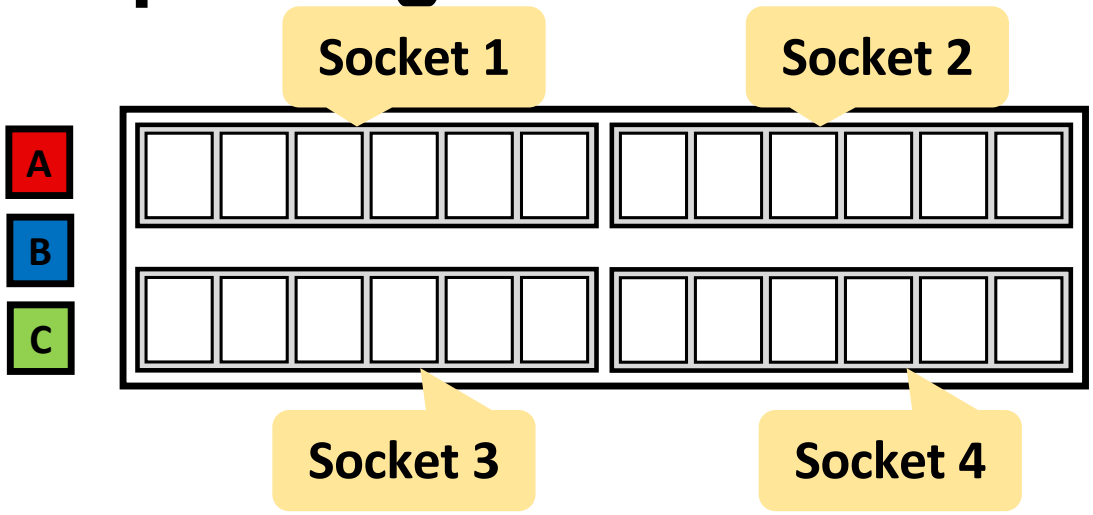
Power usage at supercomputers



It is critical to improve resource utilization for achieving energy efficiency

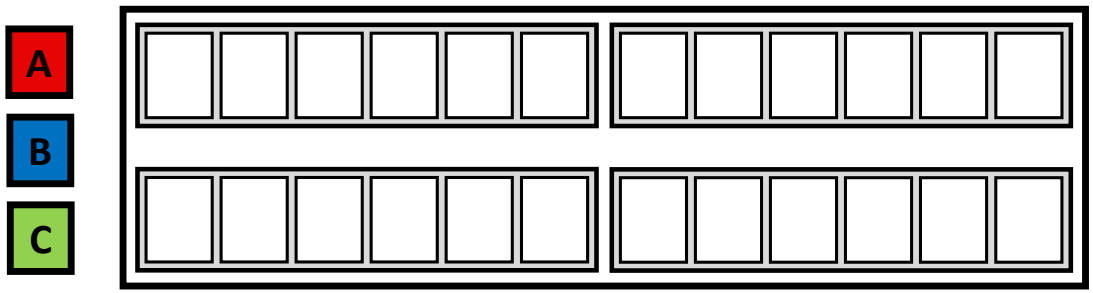
1. https://top500.org/lists/top500/ 2. Patki et.al. [ICS2025]

Improving Resource Utilization via Co-location



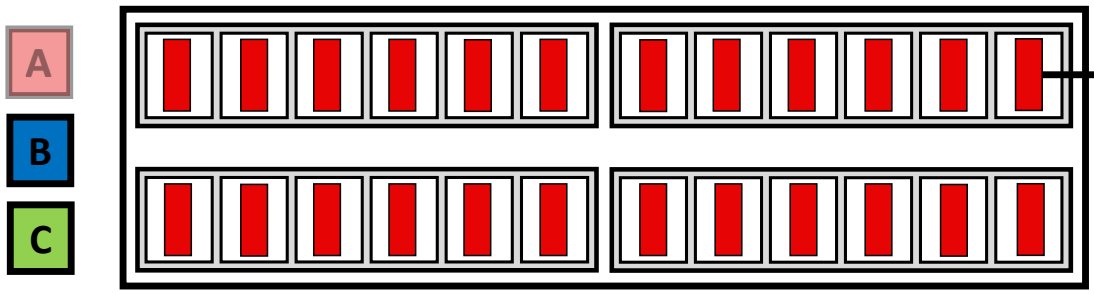
Applications A, B & C to be executed on a quad-socket system

Co-location: For Resource Efficiency



Applications A, B & C to be executed on a quad-socket system

Improving Resource Utilization via Co-location

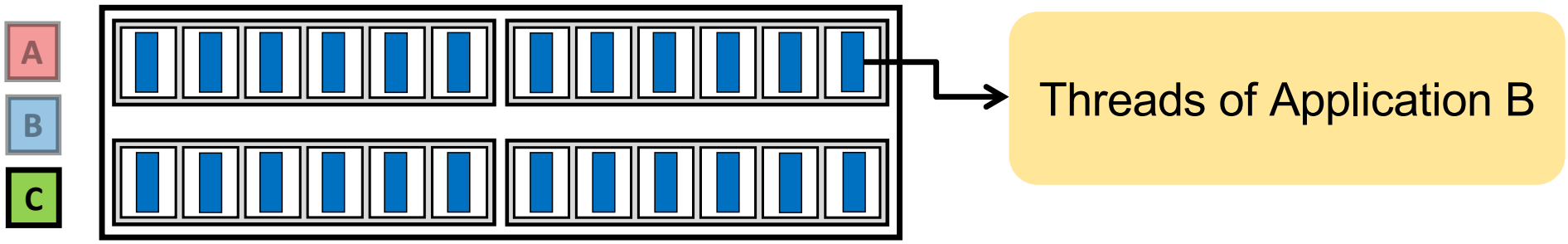


Threads of Application A

Batch Execution

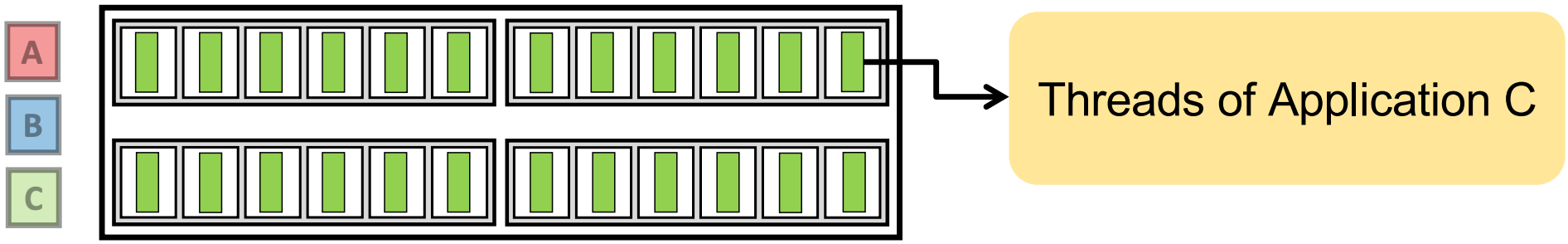
Applications B & C waiting for the CPUs

Improving Resource Utilization via Co-location



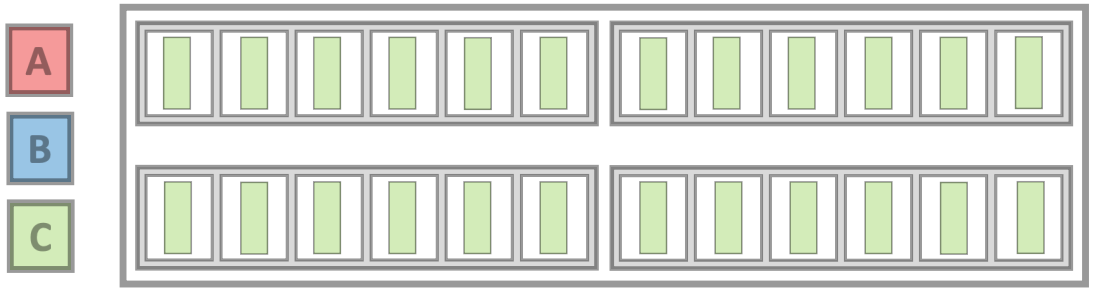
Application C
waiting for the CPUs

Improving Resource Utilization via Co-location

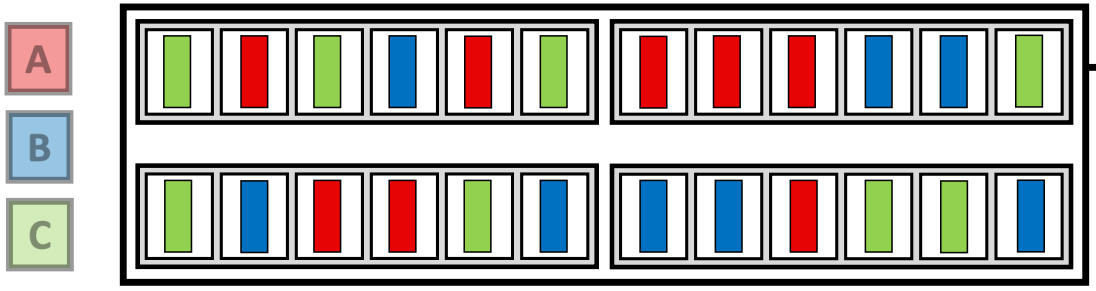


Each application completed their execution one by one

Improving Resource Utilization via Co-location



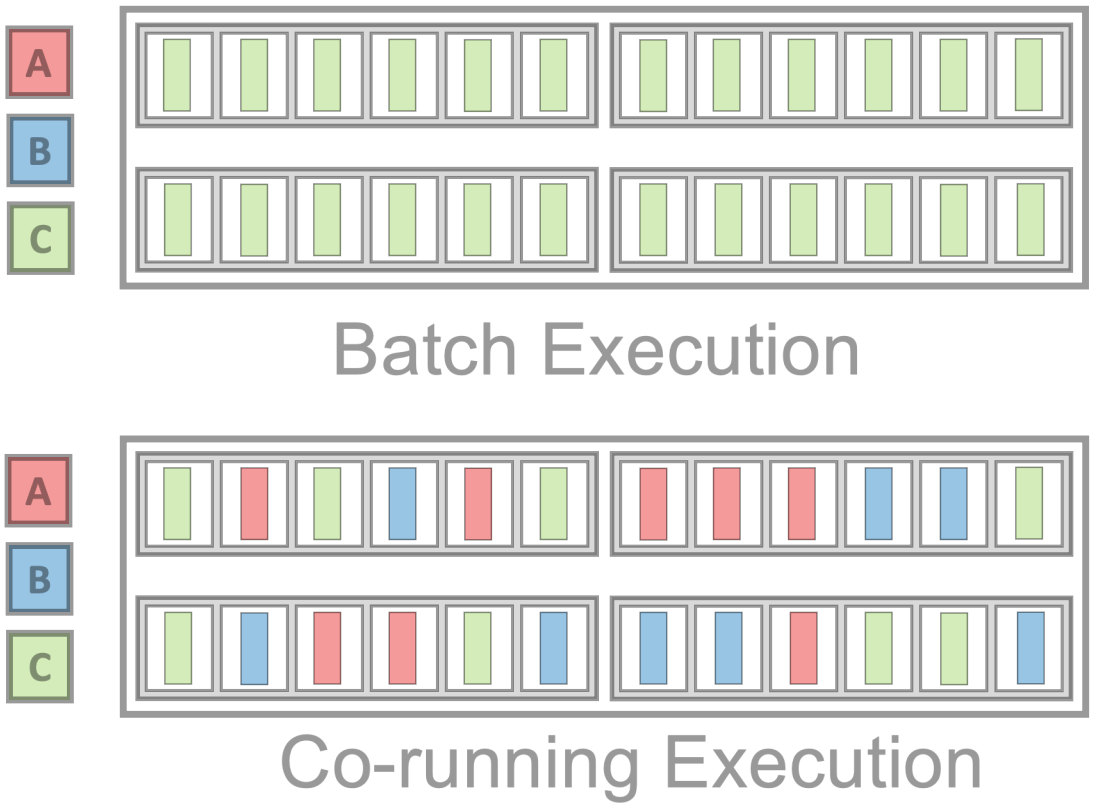
Batch Execution

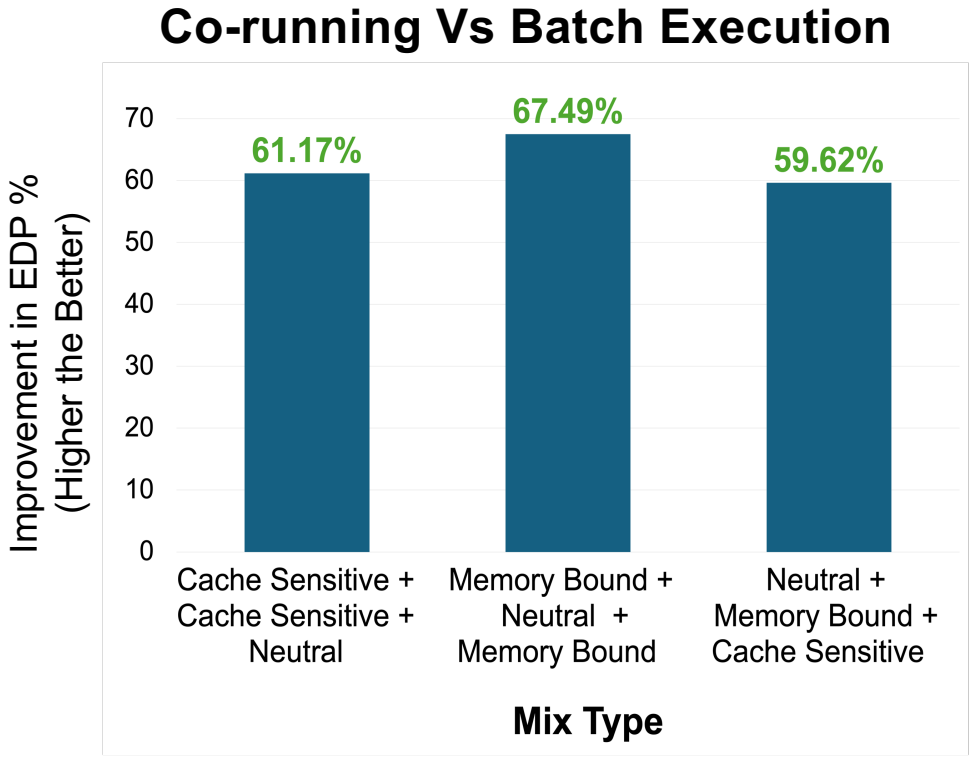


Co-running Execution

Threads of Application
A, B & C
running in parallel
(**Default thread placement**)

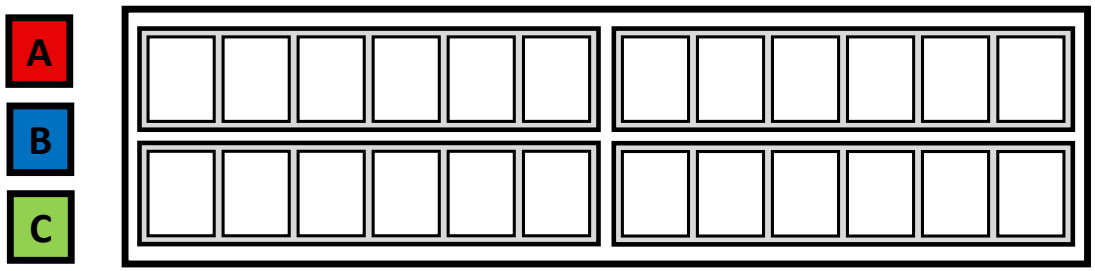
Co-location: For Resource Efficiency



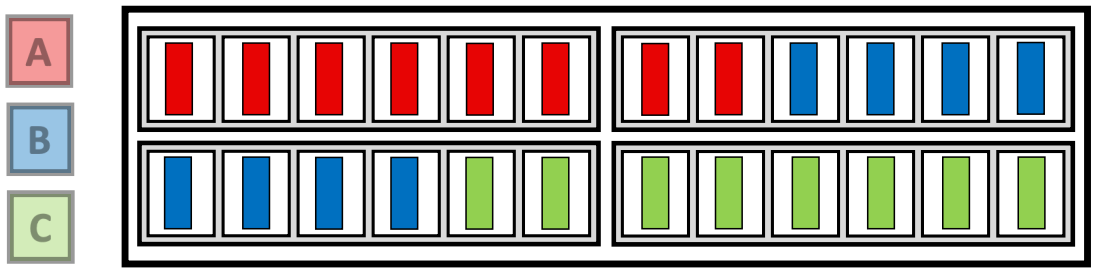


EDP improves by up to 67.5% with co-running applications

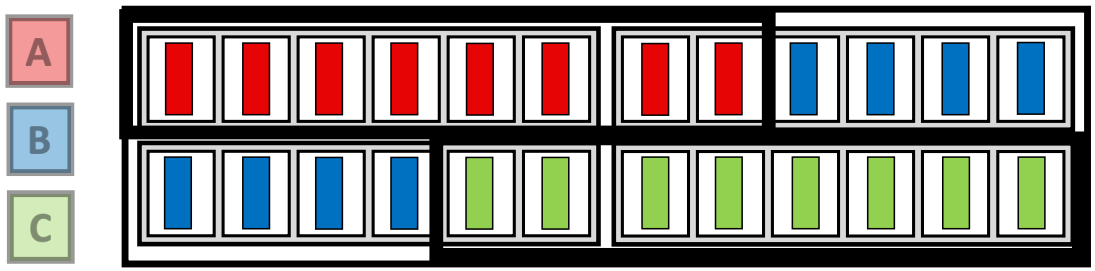
Thread Placement: For Reducing Contention



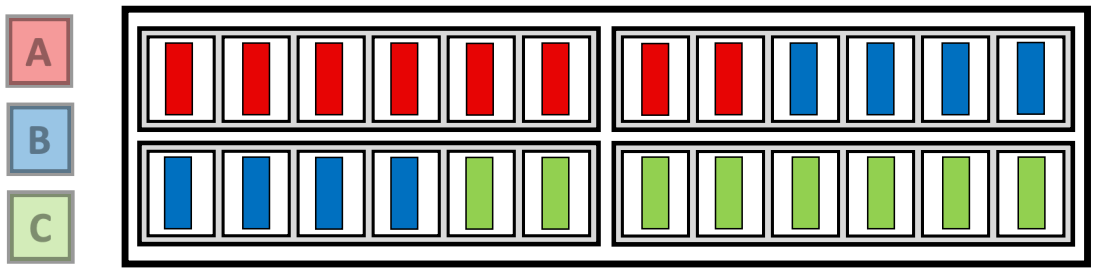
Thread Placement: For Reducing Contention



Thread Placement: For Reducing Contention

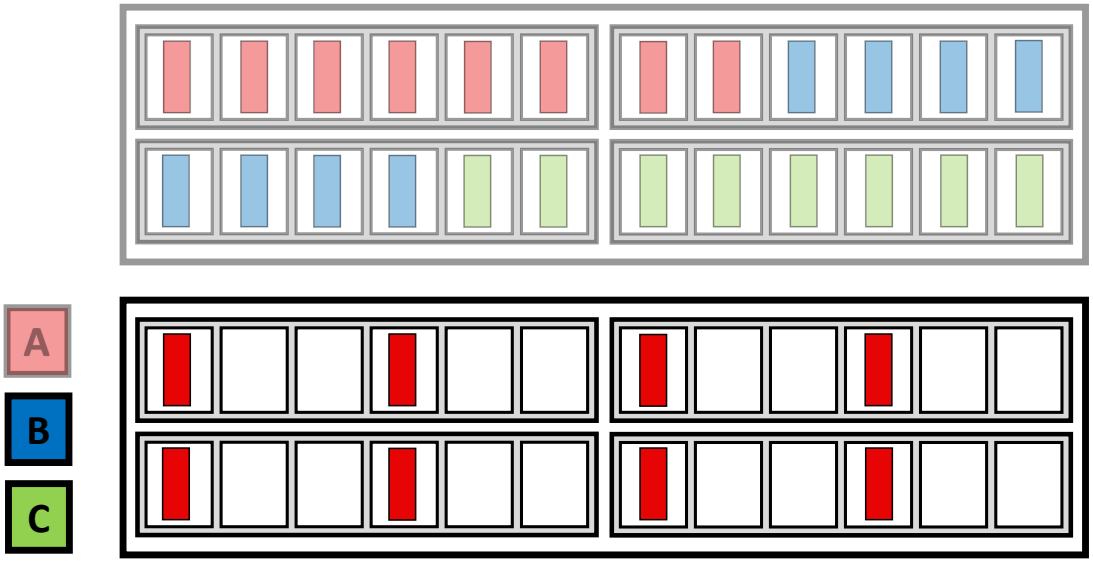


Thread Placement: For Reducing Contention

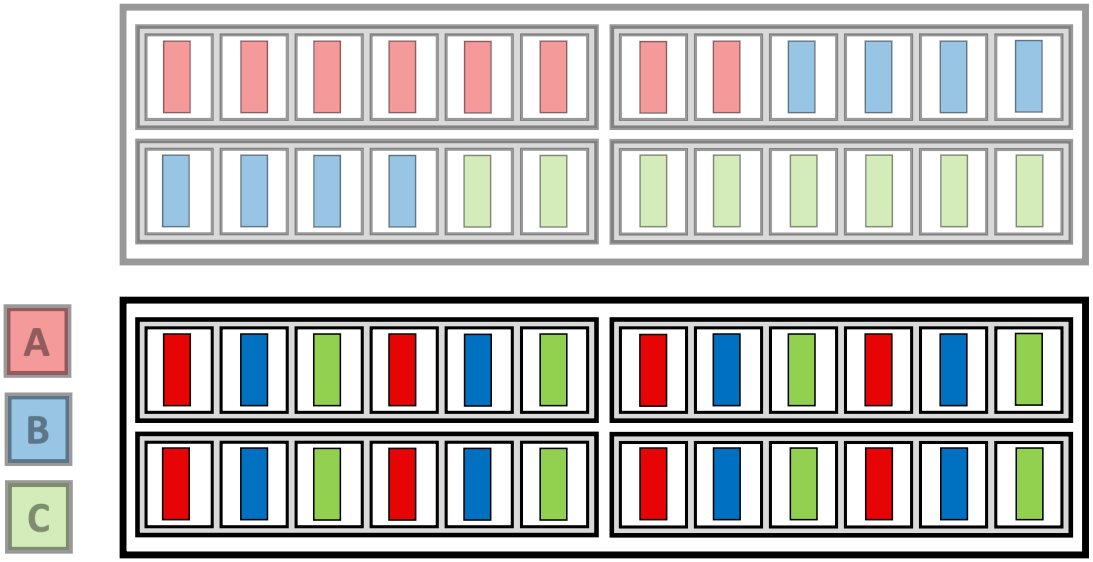


Type: Block-Cyclic

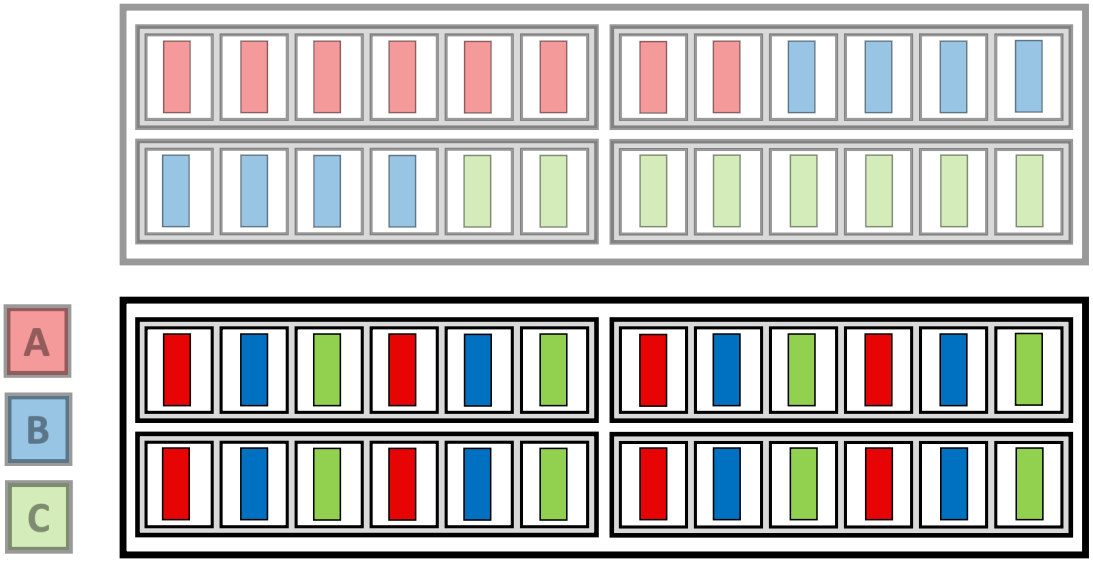
Thread Placement: For Reducing Contention



Thread Placement: For Reducing Contention

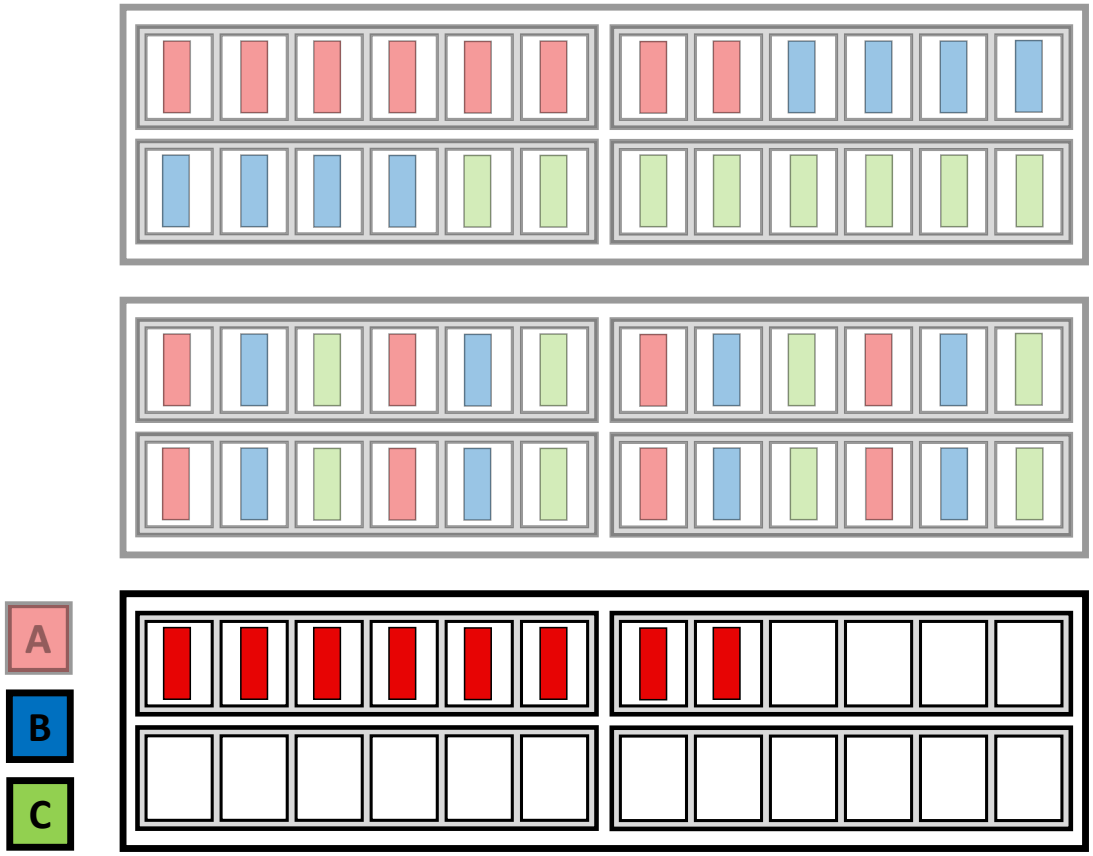


Thread Placement: For Reducing Contention

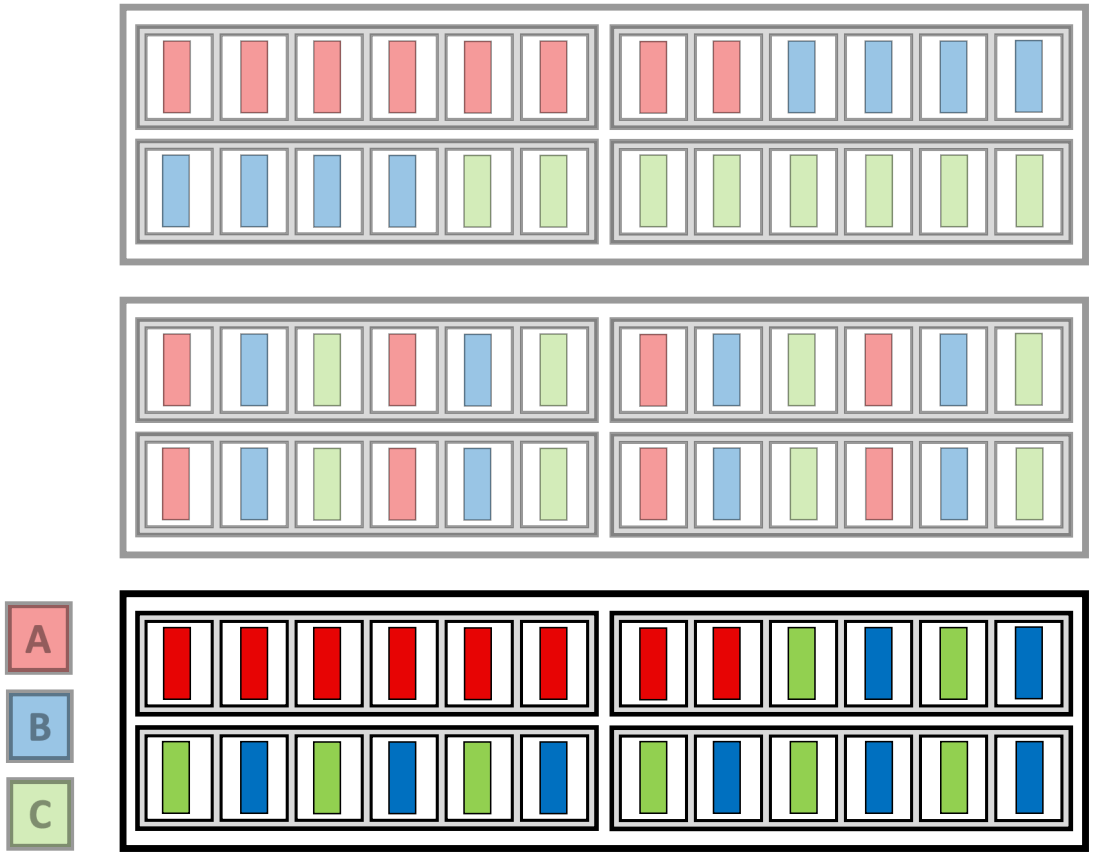


Type: Interleaved

Thread Placement: For Reducing Contention

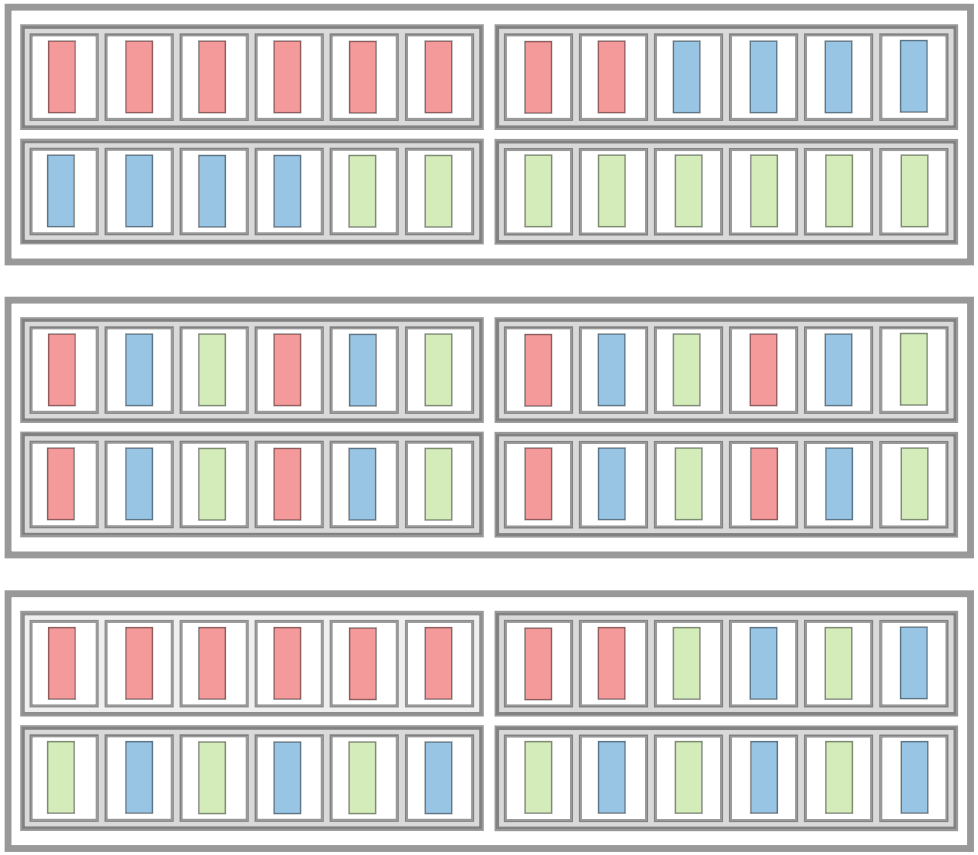


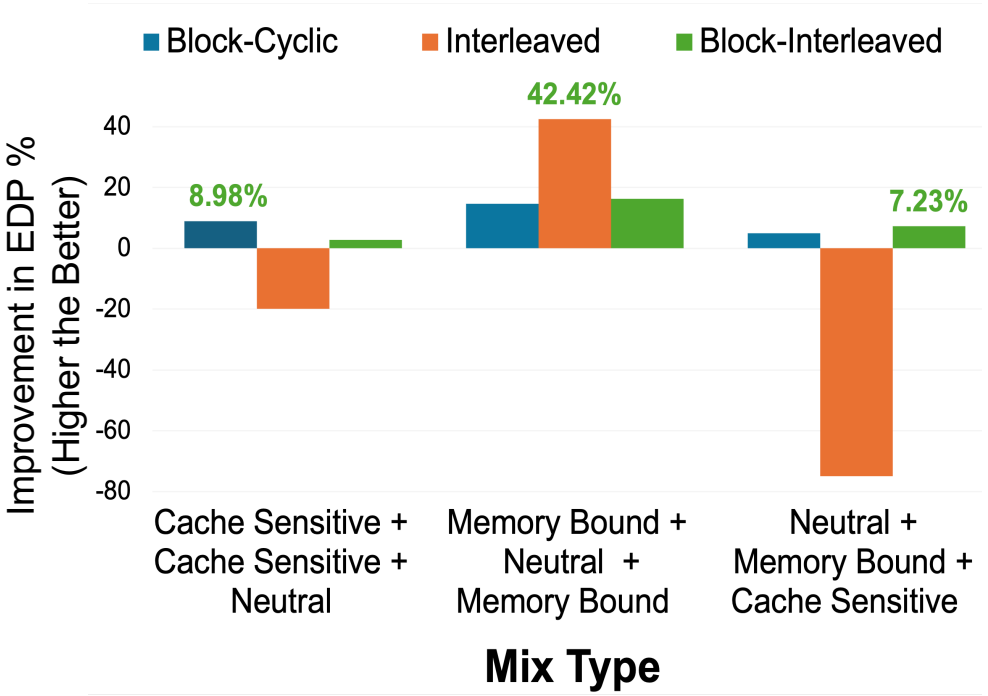
Thread Placement: For Reducing Contention



Type: Block-Interleaved

Thread Placement: For Reducing Contention





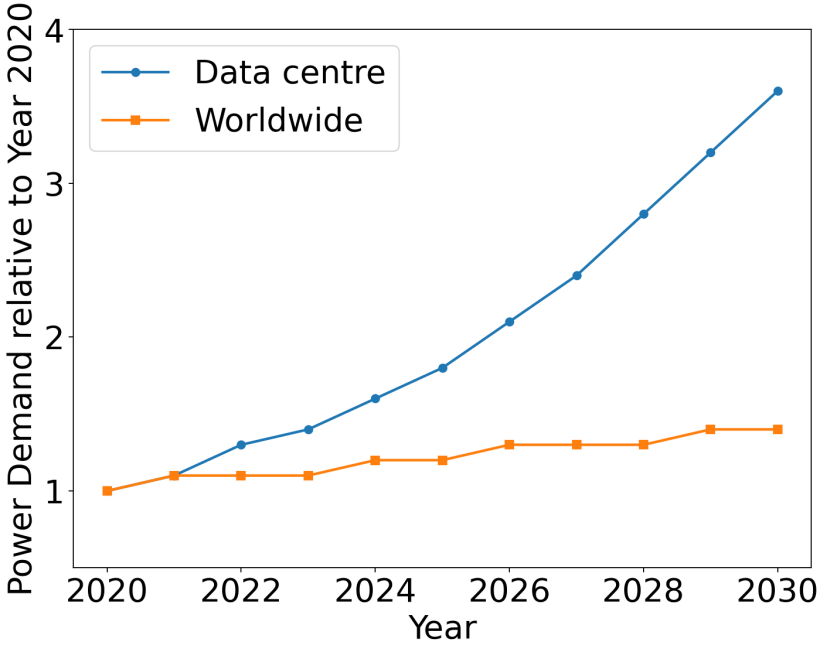
Choosing optimal thread placement over Default improves EDP by up to 42.4%

Today's Class

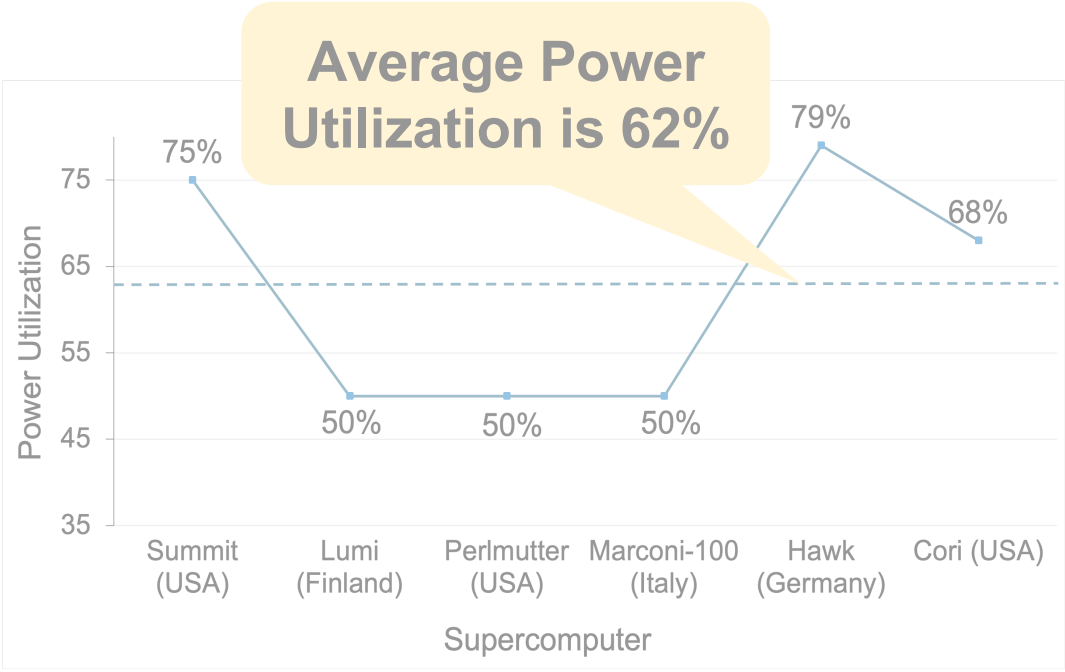
- Achieving reliability
- Co-running applications v/s batch execution
- Power capping
- Quiz-5

Exascale Computing Era

Power demand in the data centres



Power usage at supercomputers



It is extremely essential to improve power efficiency

1. https://www.iea.org/reports/energy-and-ai
2. Patki et.al. [ICS2025]

Hardware Overprovisioning using Power cap

400W

200W 200W

System running under **TDP**

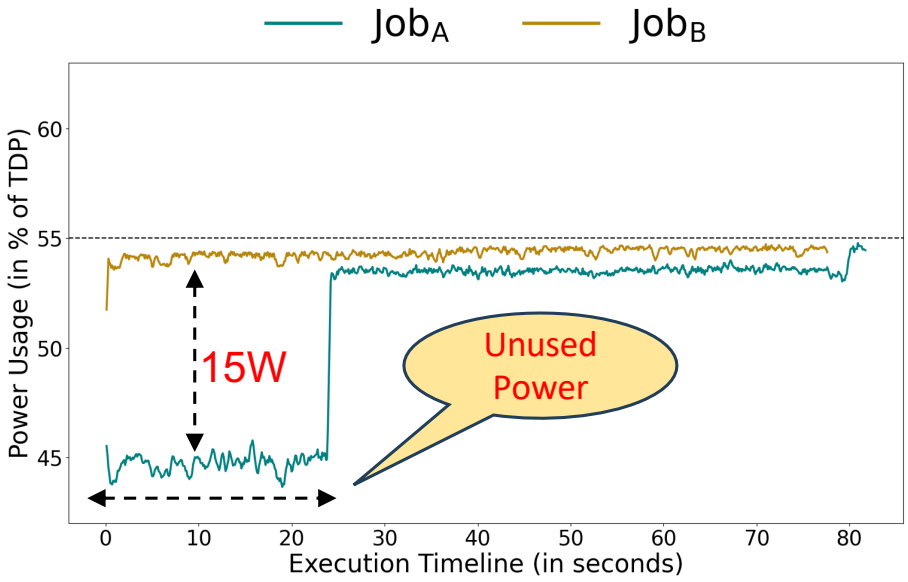
400W

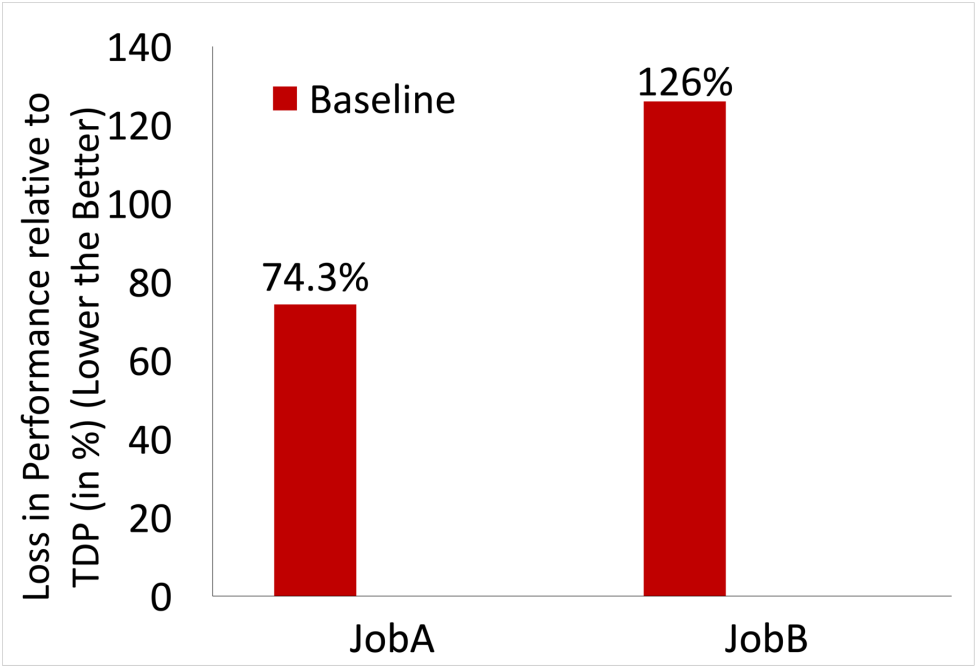
100W 100W 100W 100W

System running under **50% of TDP**

- Servers are designed to operate within the Thermal Design Power (TDP) limit
  - TDP is the maximum power limit
- Power capping (PCAP) restricts power usage below TDP
  - Allows using more servers within the same power budget

Issues with Power Capping (PCAP)

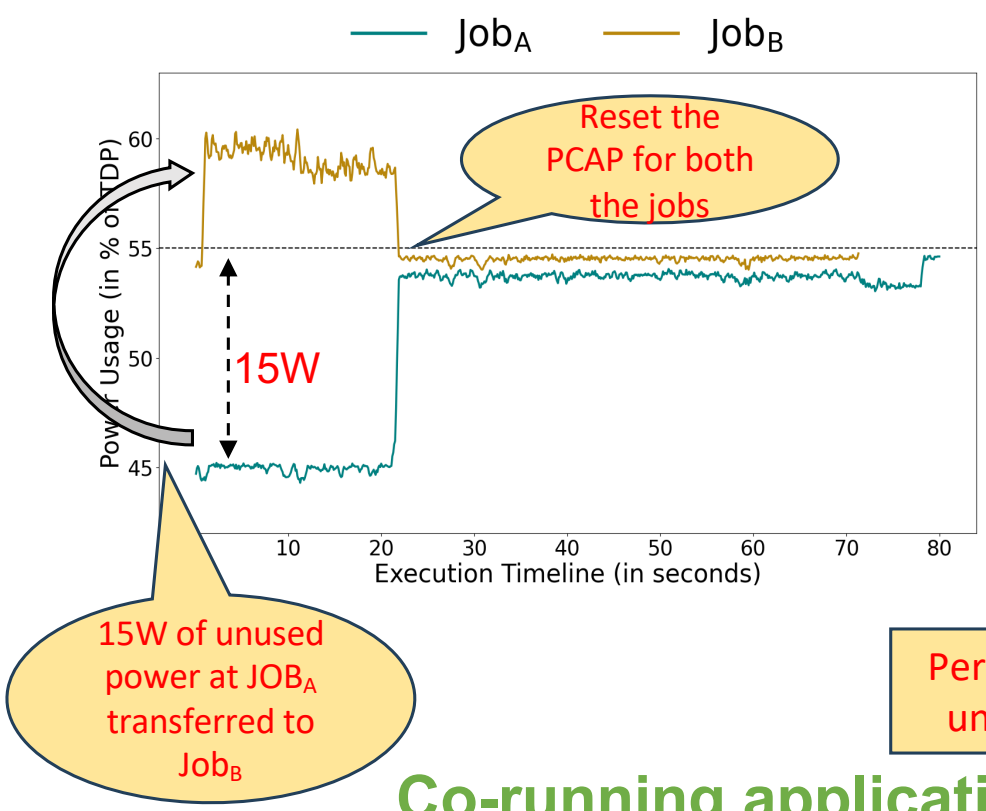


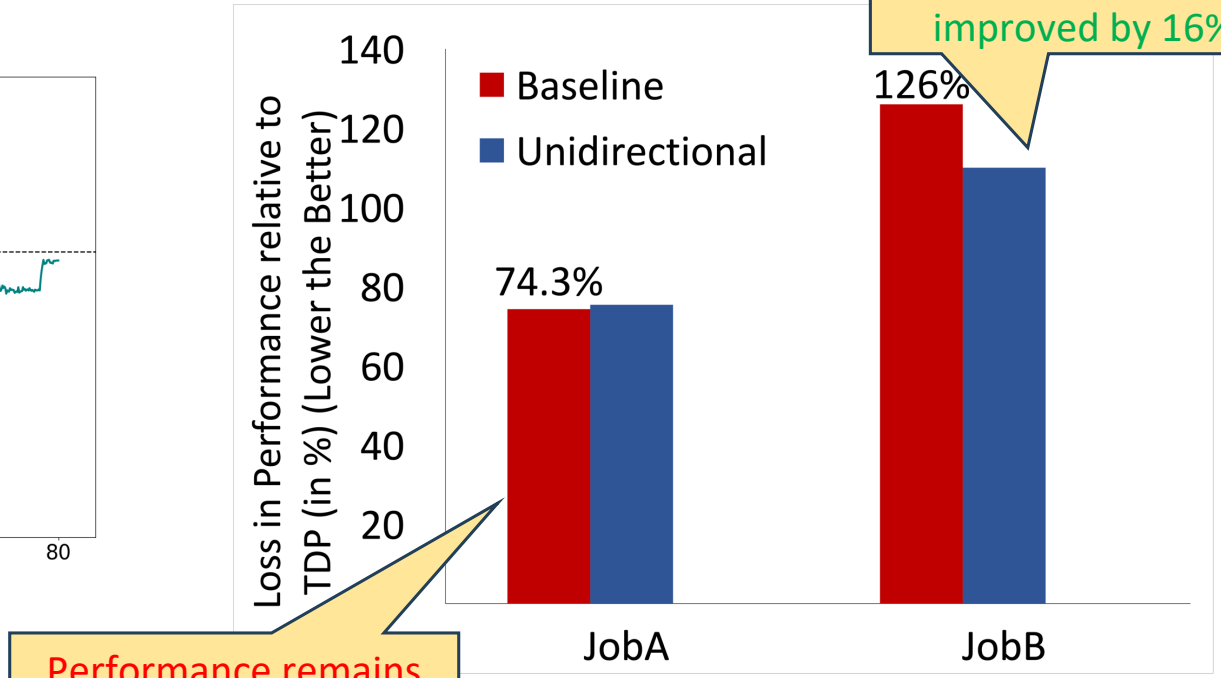


Power usage changes throughout the application execution

Job$_B$ fully utilizes the available power, whereas Job$_A$ only partially utilizes it

Improving Performance under PCAP





Co-running applications on multi-socket servers provides an opportunity to reduce power wastage

Next Lecture

- End semester review