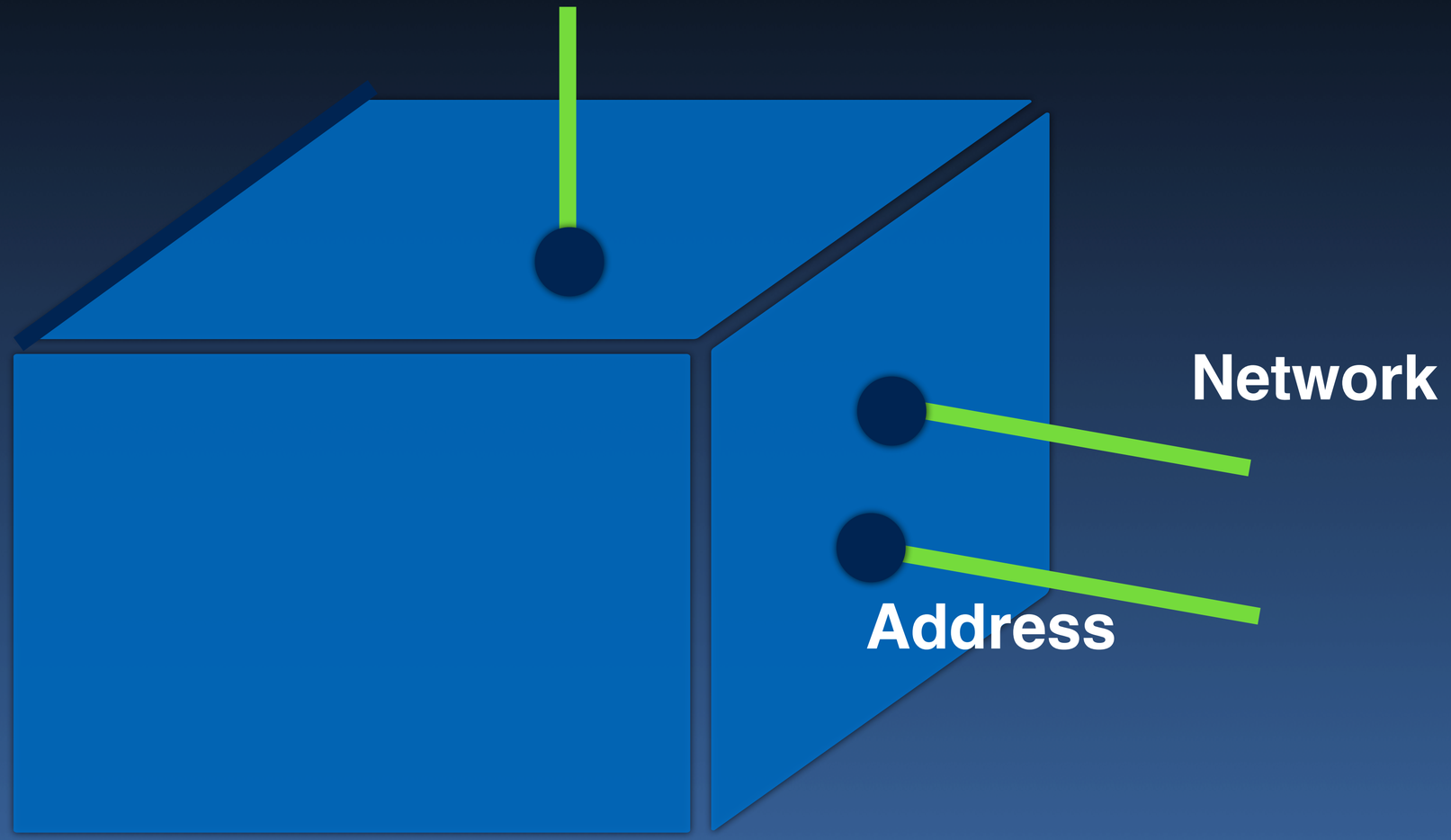
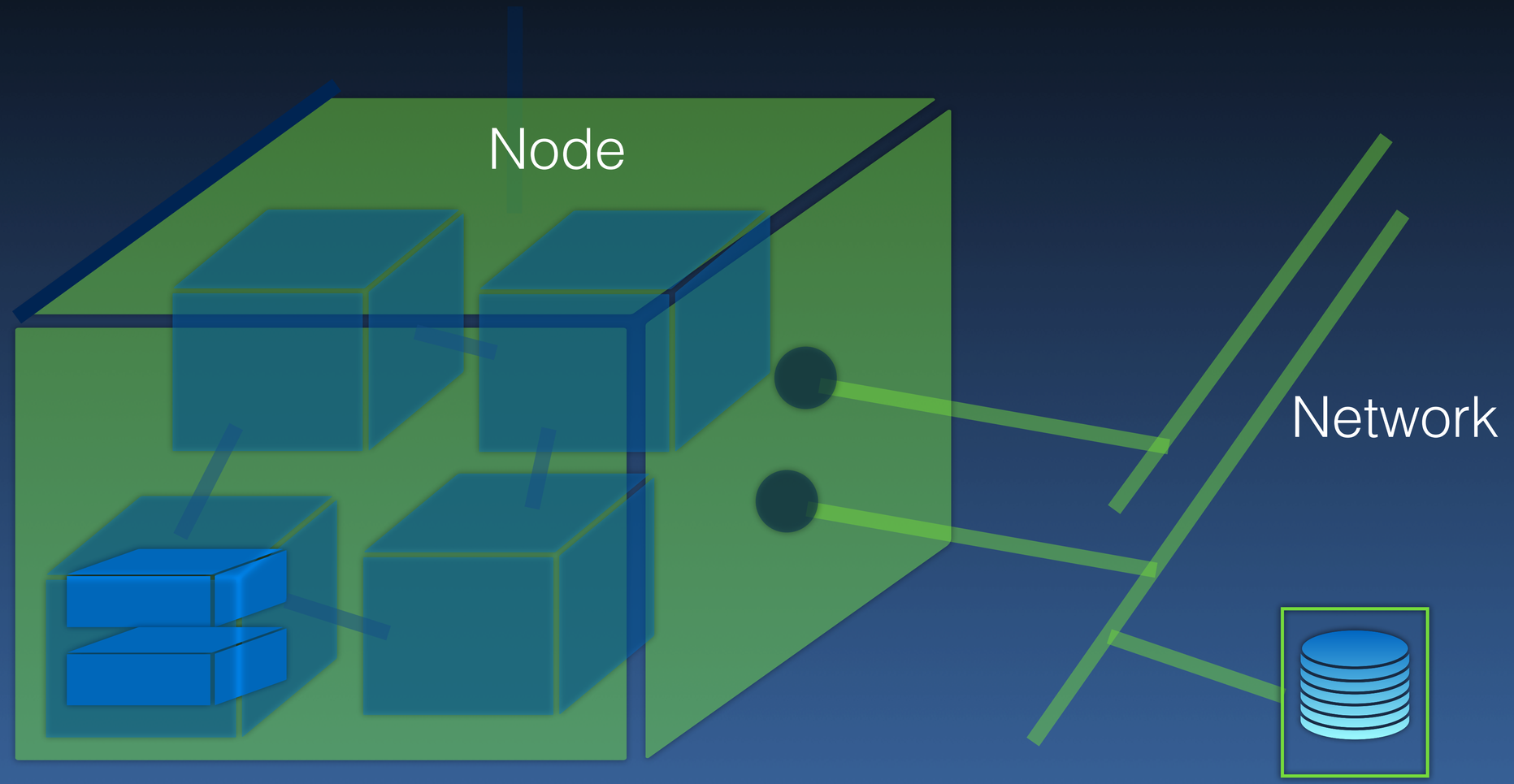
COL380

Introduction to
Parallel & Distributed Programming

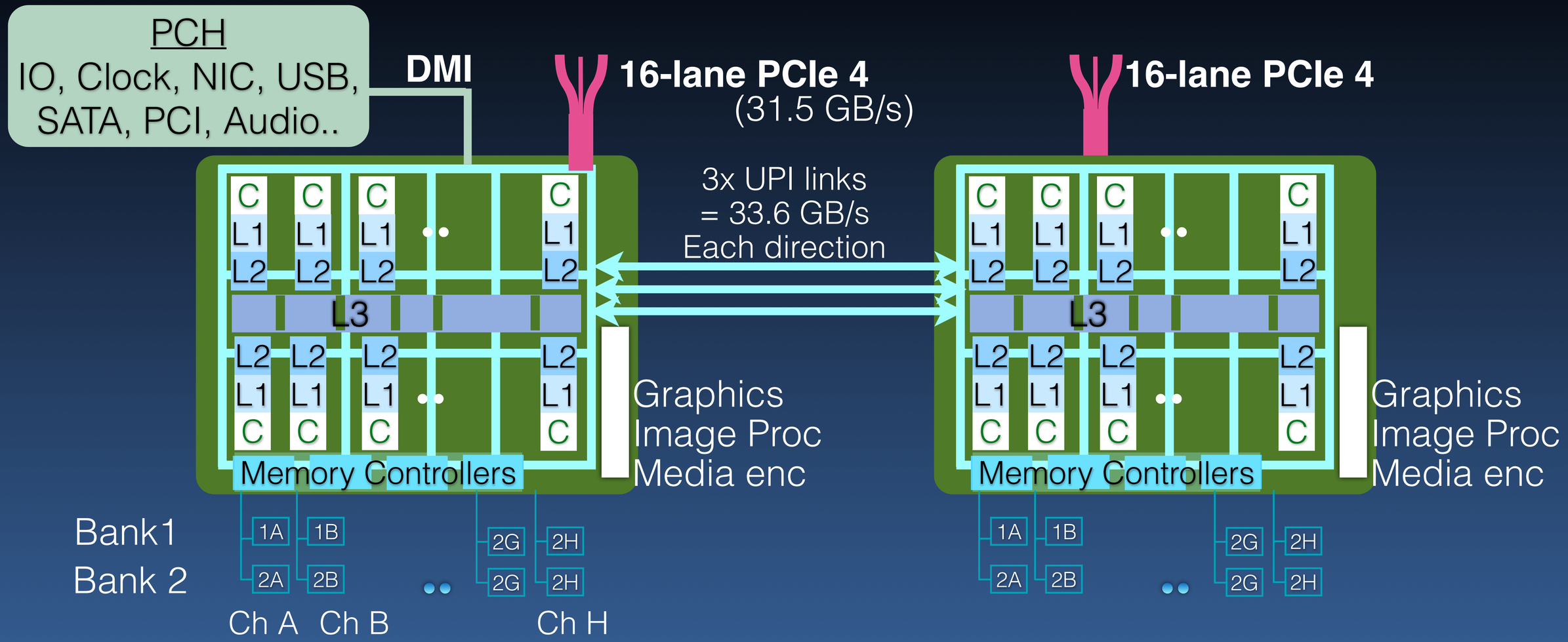
Supercomputer Node



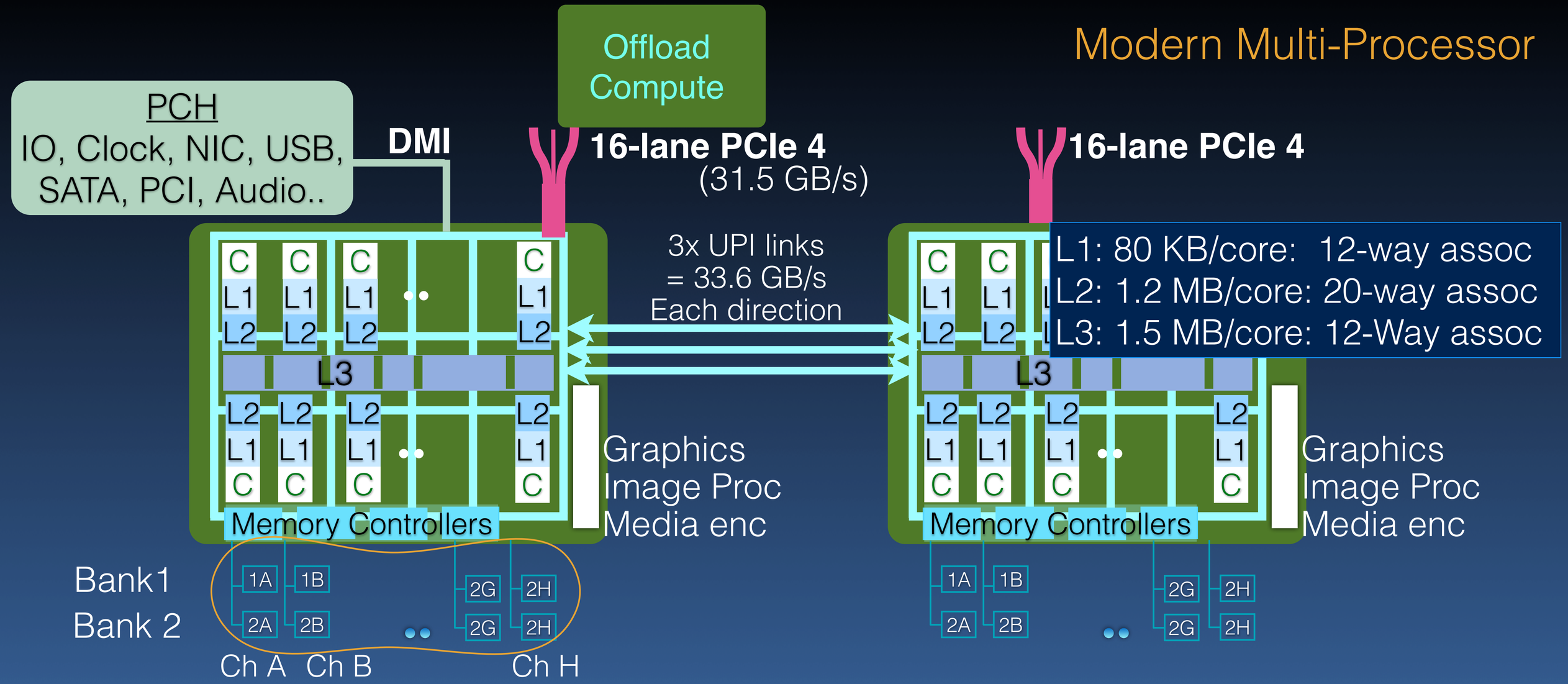
Supercomputer Node



Modern Multi-Processor



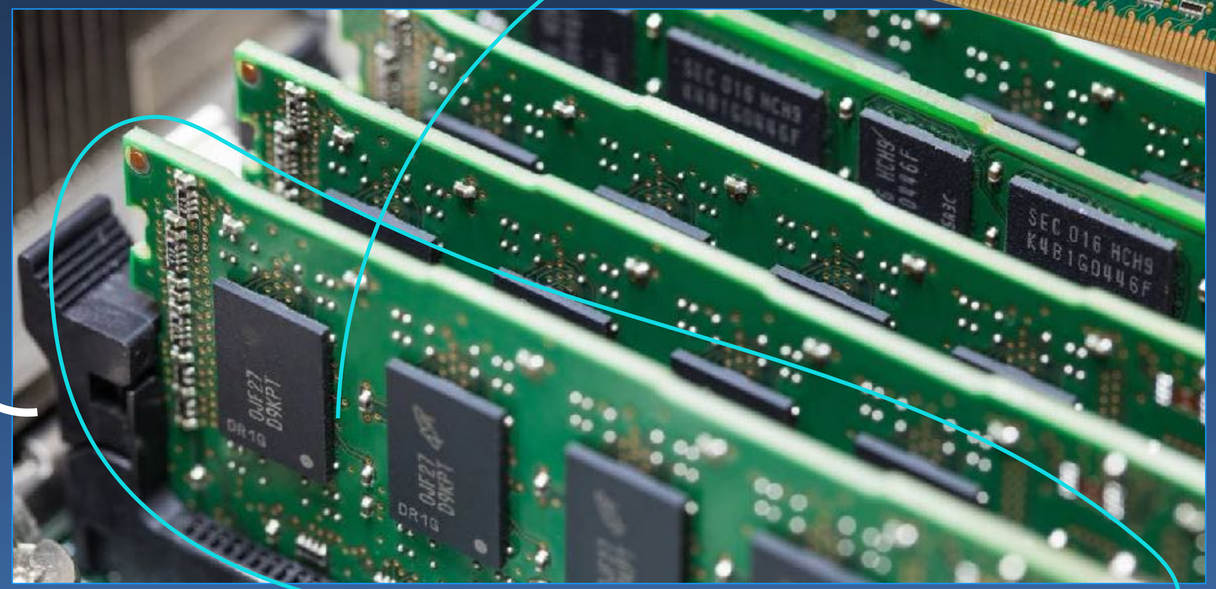
Modern Multi-Processor



Subodh Kumar

Memory

Memory
Controller



Subodh Kumar

Memory Bottleneck

- **Fused Multiply Add**
  - Double Precision A += B * C
  - 2 FLOPS, 3+1 Operands (32 bytes)
- **Example**
  - 2x AVX512 on Intel core = 32 FLOP/cycle
  - 96 GF/core @3GHz
    - Needed 1536 GB/s memory bandwidth/core
  - Compare DDR4: Throughput ~25GB/s

Latency can be hidden

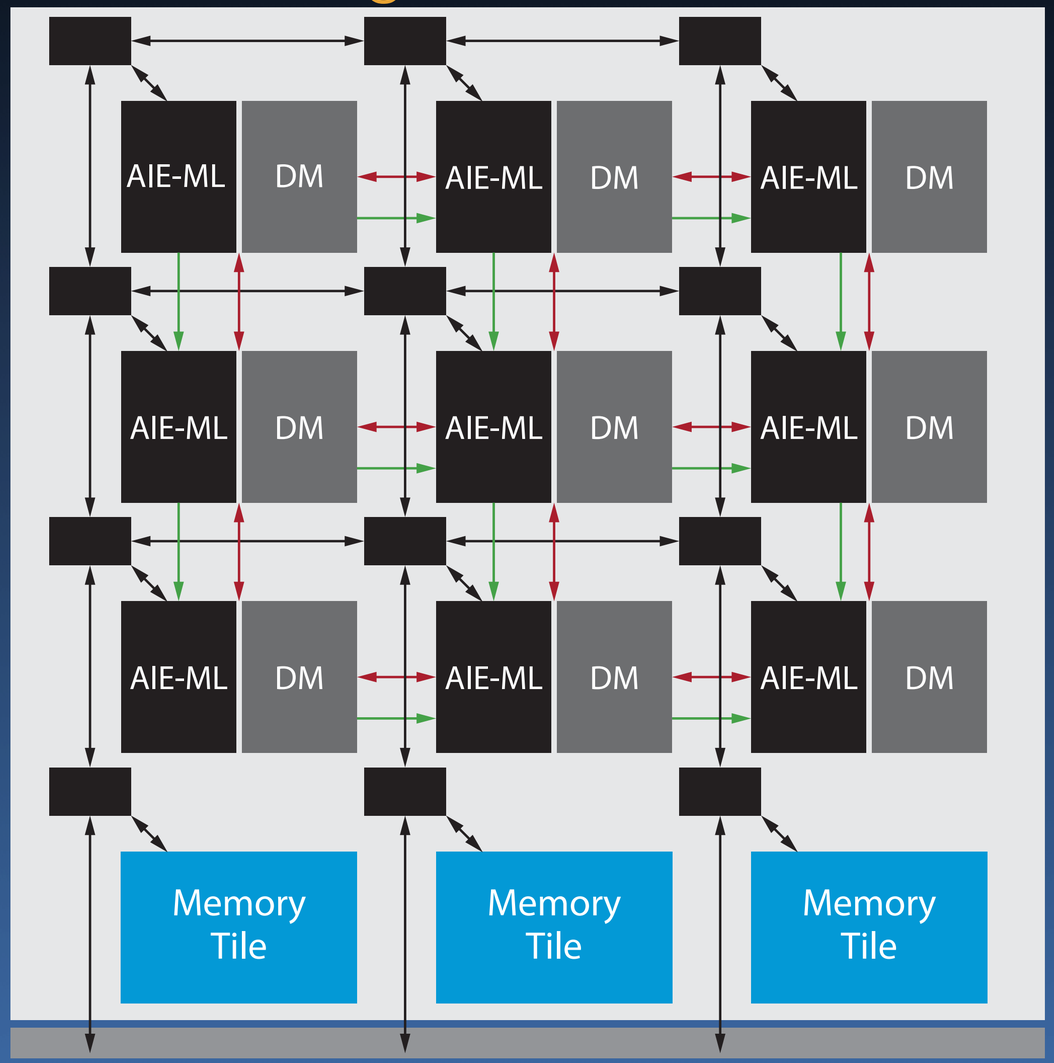
```
A[i] += B[i]*C[i]
A[i+1] += B[i+1]*C[i+1]
A[i+2] += B[i+2]*C[i+2]
A[i+3] += B[i+3]*C[i+3]
```

Caches can help with throughput but working set can be large

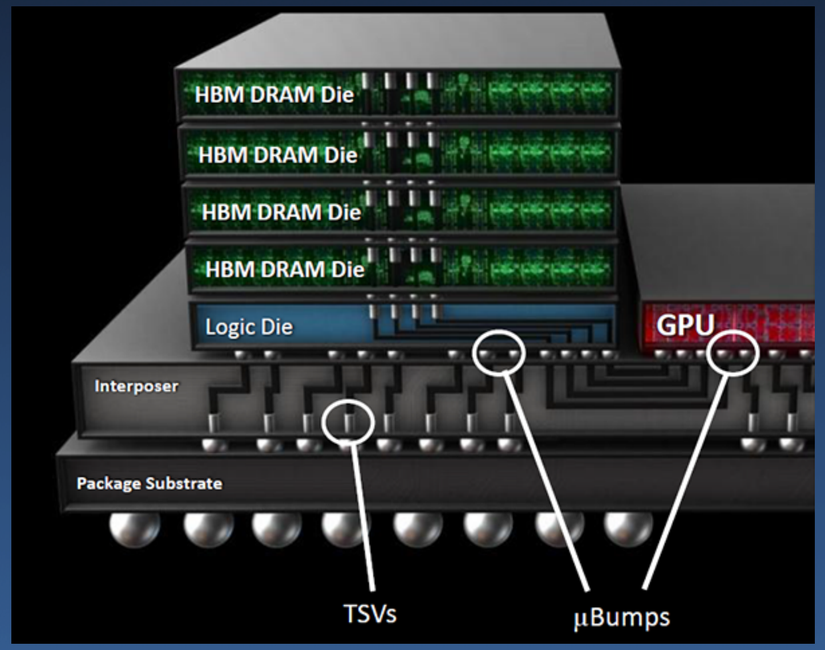
✳ Must be used wisely

Memory Gap Mitigation

AMD AI Engine

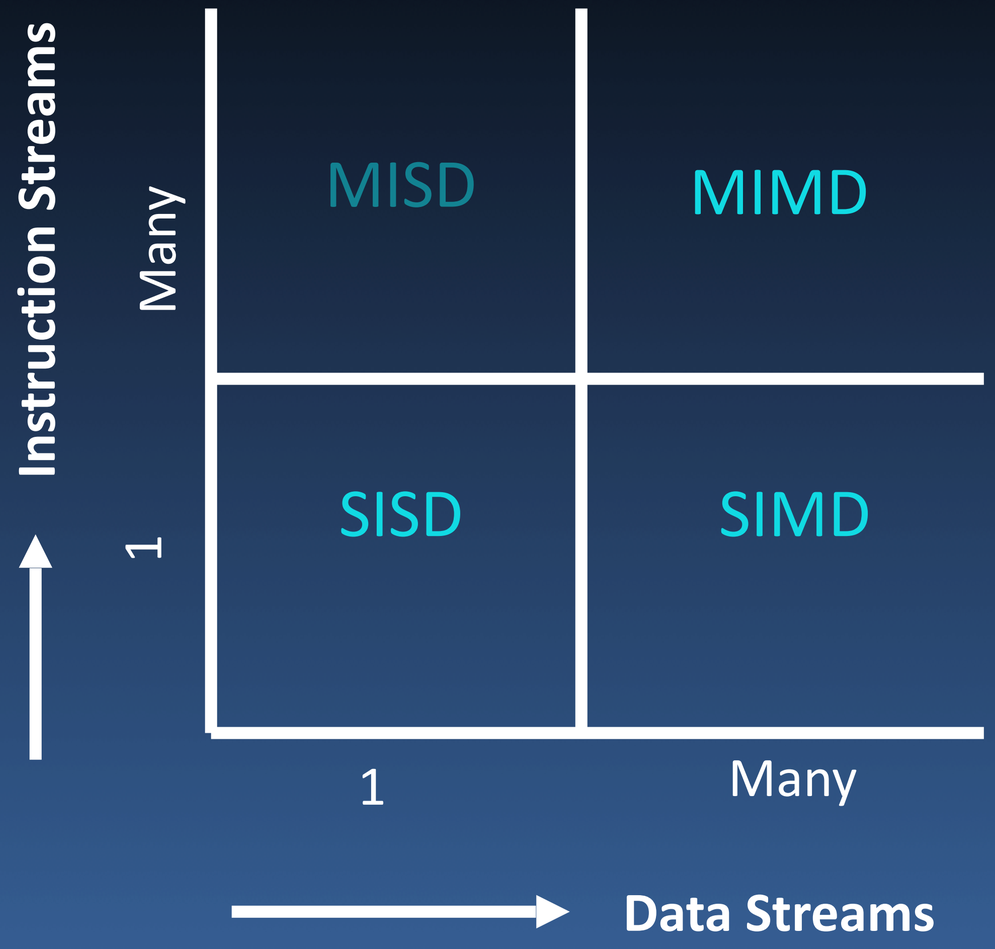


High Bandwidth Memory



~600 GB/site

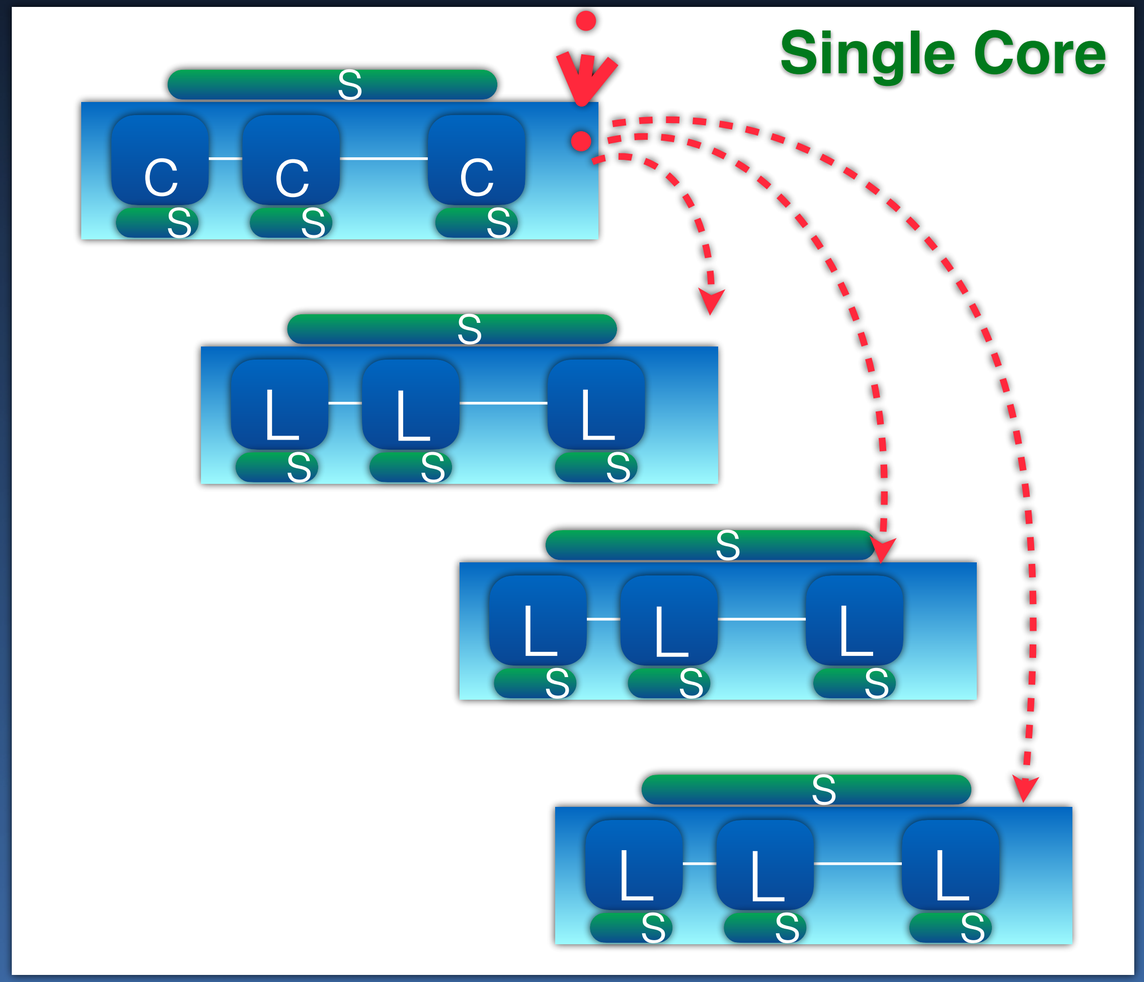
Flynn's Classification



- A number of instruction streams
- A number of data/operand streams

NOT Sequential

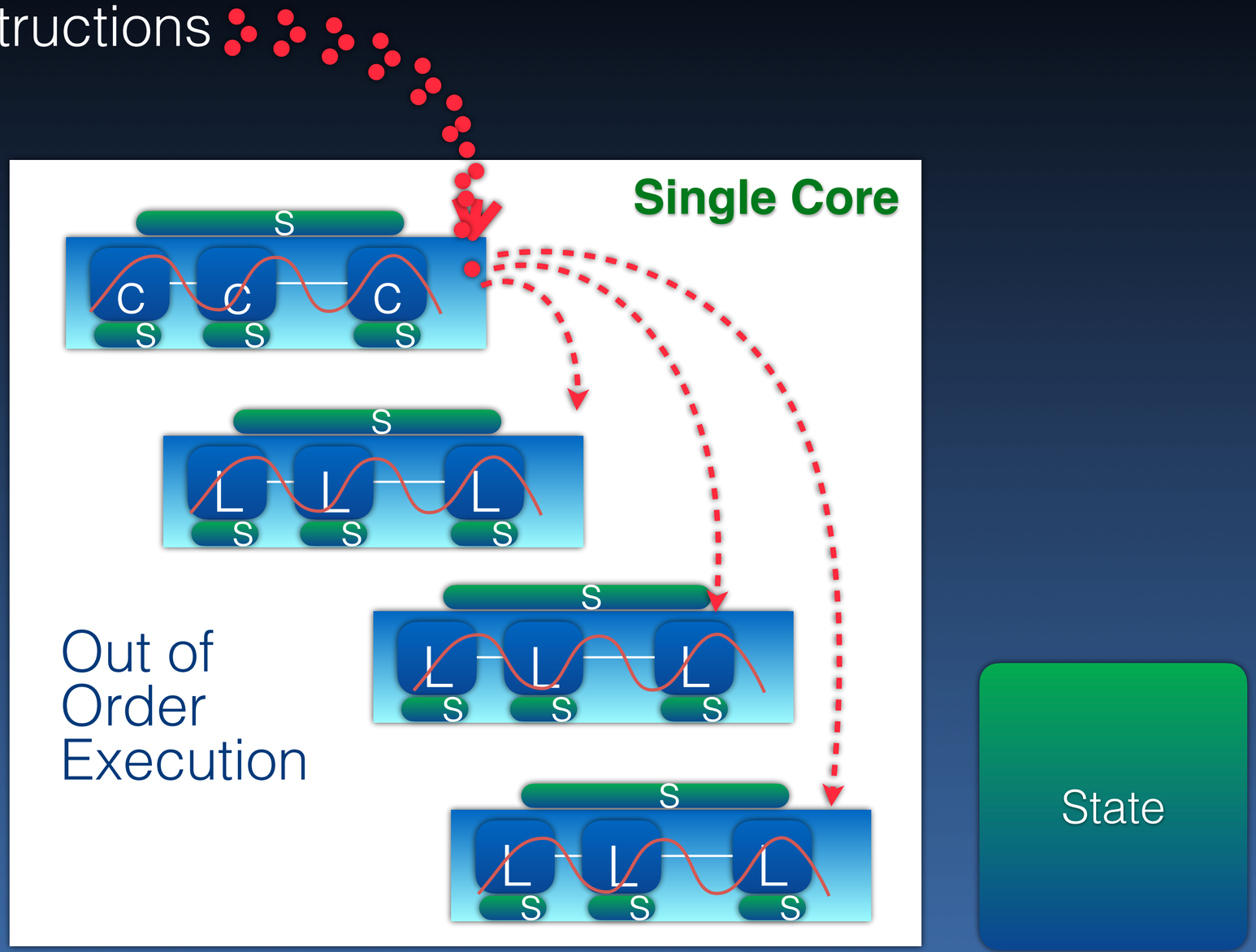
Instructions



State

Subodh Kumar

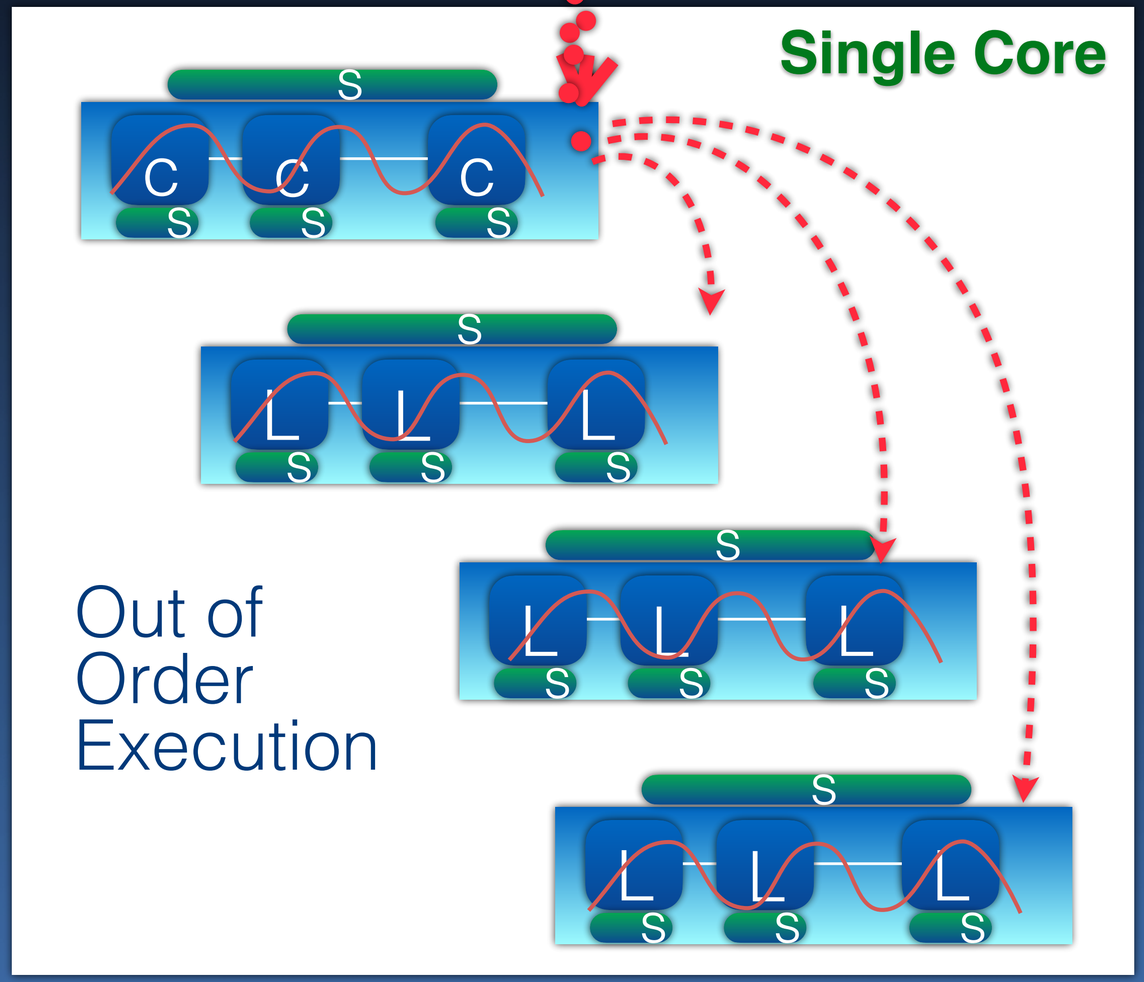
NOT Sequential



Subodh Kumar

NOT Sequential

Instructions



volatile int x;

Access x

.

.

R1 = x

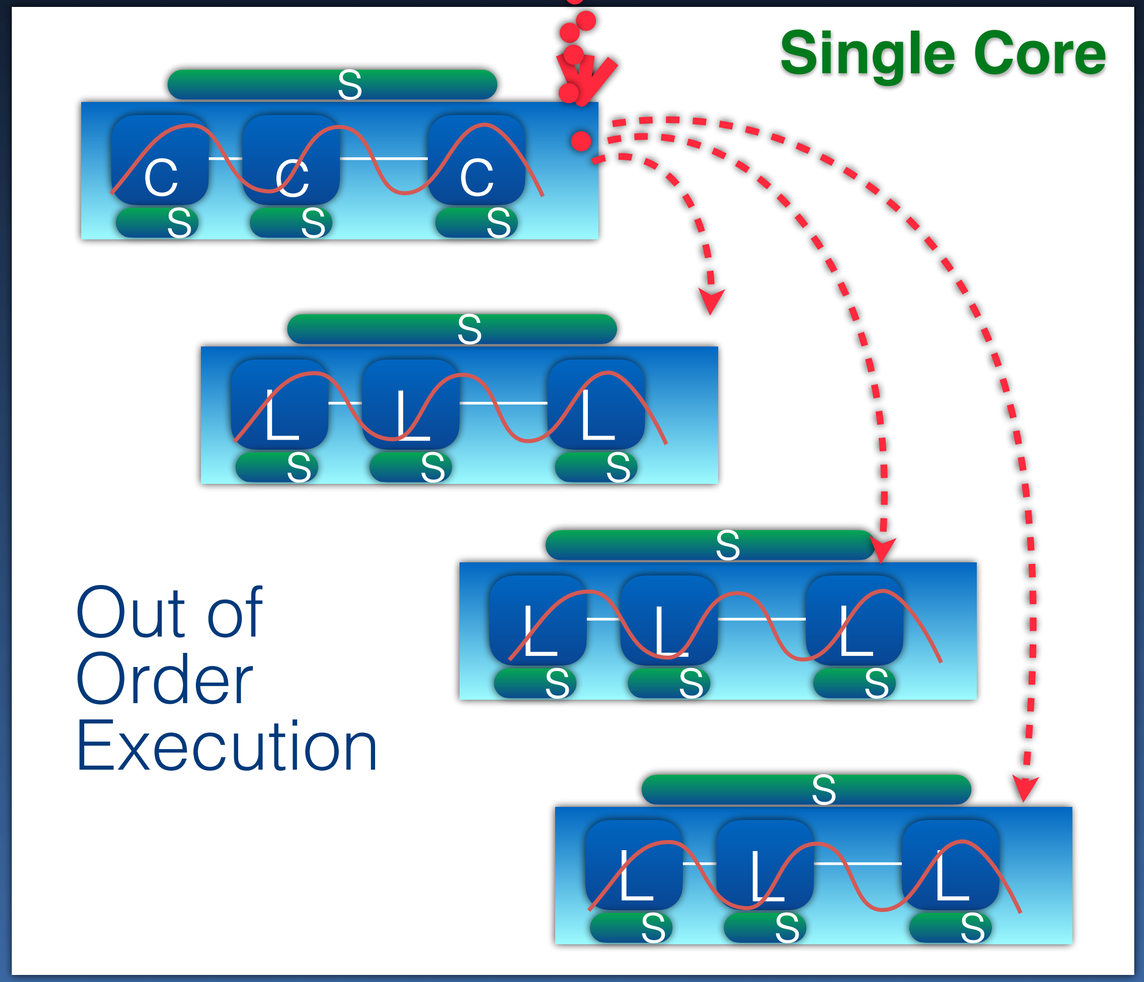
R2 = y

Z = R1+R2

State

NOT Sequential

Instructions



State

Mem Mgmt Unit

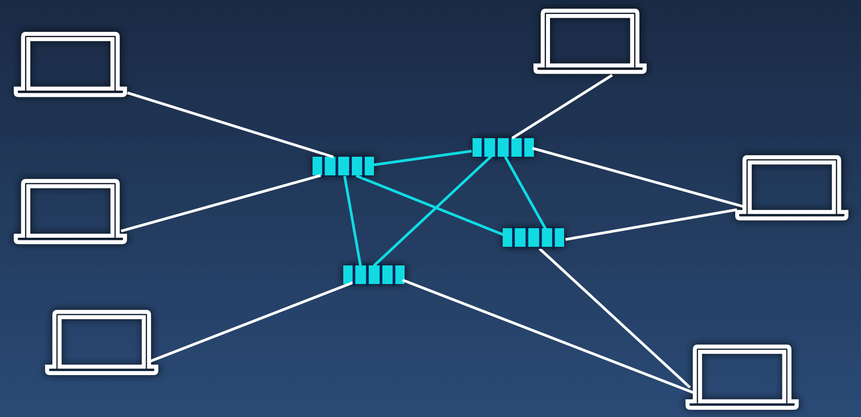
Network Controllers

DMA Engines

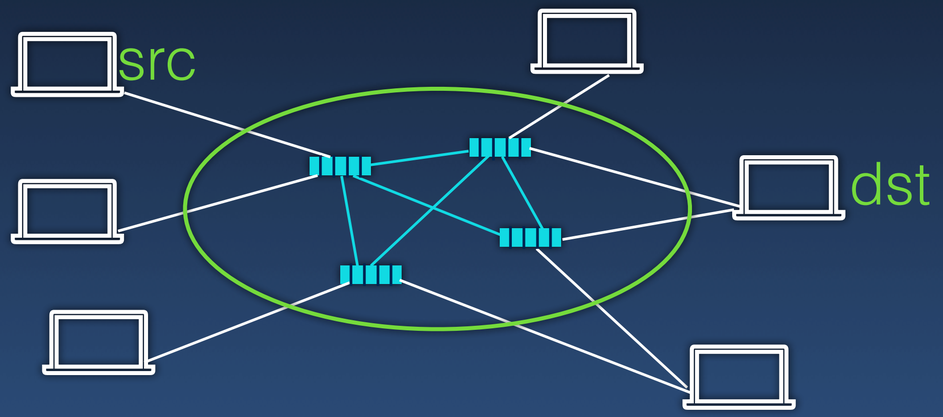
IO controller

Subodh Kumar

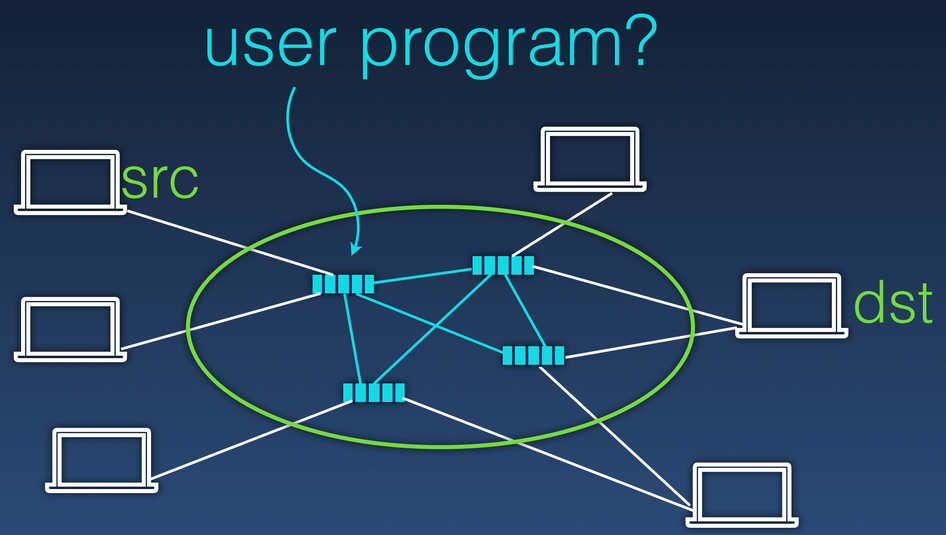
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



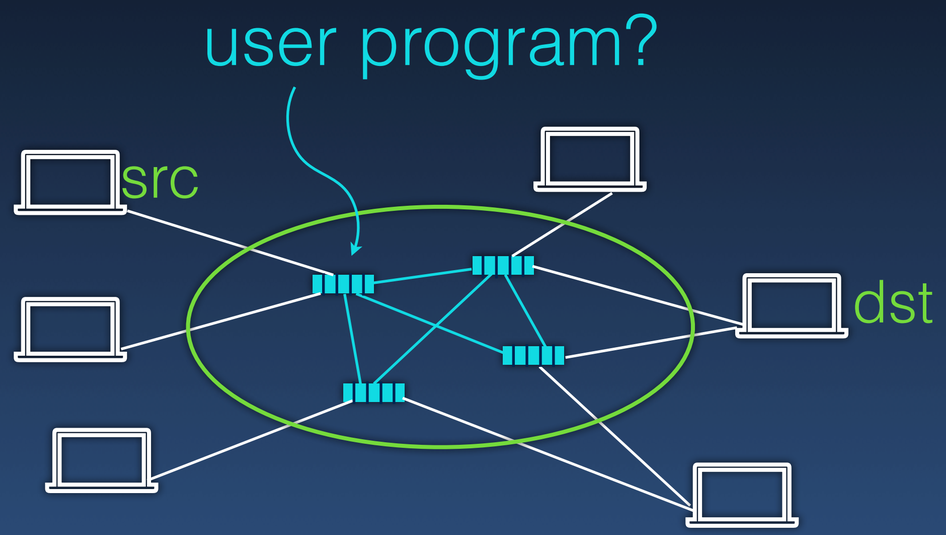
- Routing algorithm
  - ➡ Address, Low latency, High bandwidth



- Routing algorithm
  - ➡ Address, Low latency, High bandwidth
- Metrics
  - ➡ Number of links required
  - ➡ Number of ports on a node
  - ➡ Distance between nodes
  - ➡ Redundancy in routes



- Routing algorithm
  - ➡ Address, Low latency, High bandwidth
- Metrics
  - ➡ Number of links required
  - ➡ Number of ports on a node
  - ➡ Distance between nodes
  - ➡ Redundancy in routes



- Diameter: Longest path
- Bisection width: Min #links failures to bi-partition the nodes
- Blocking: If independent pairs can communicate at each step

Memory Organization

- Unified memory addressing
  - ➡ Non-uniform access (NUMA)
- Address-based mapping
  - ➡ Requires memory agents/controllers
    - ▸ And a network path to them (with routers)
- Distributed memory
  - ➡ Goes through a thread of control (code)