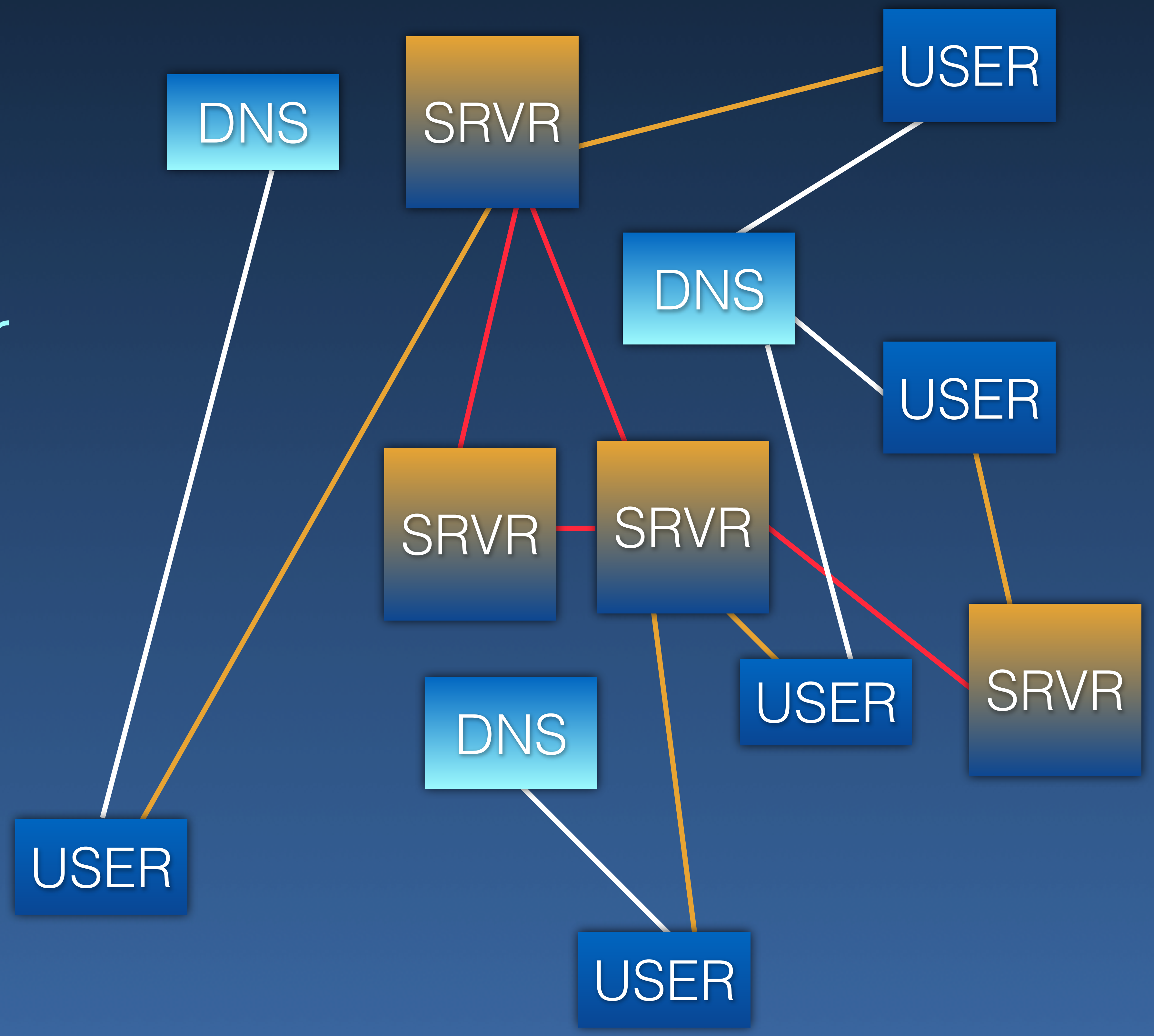
COL380

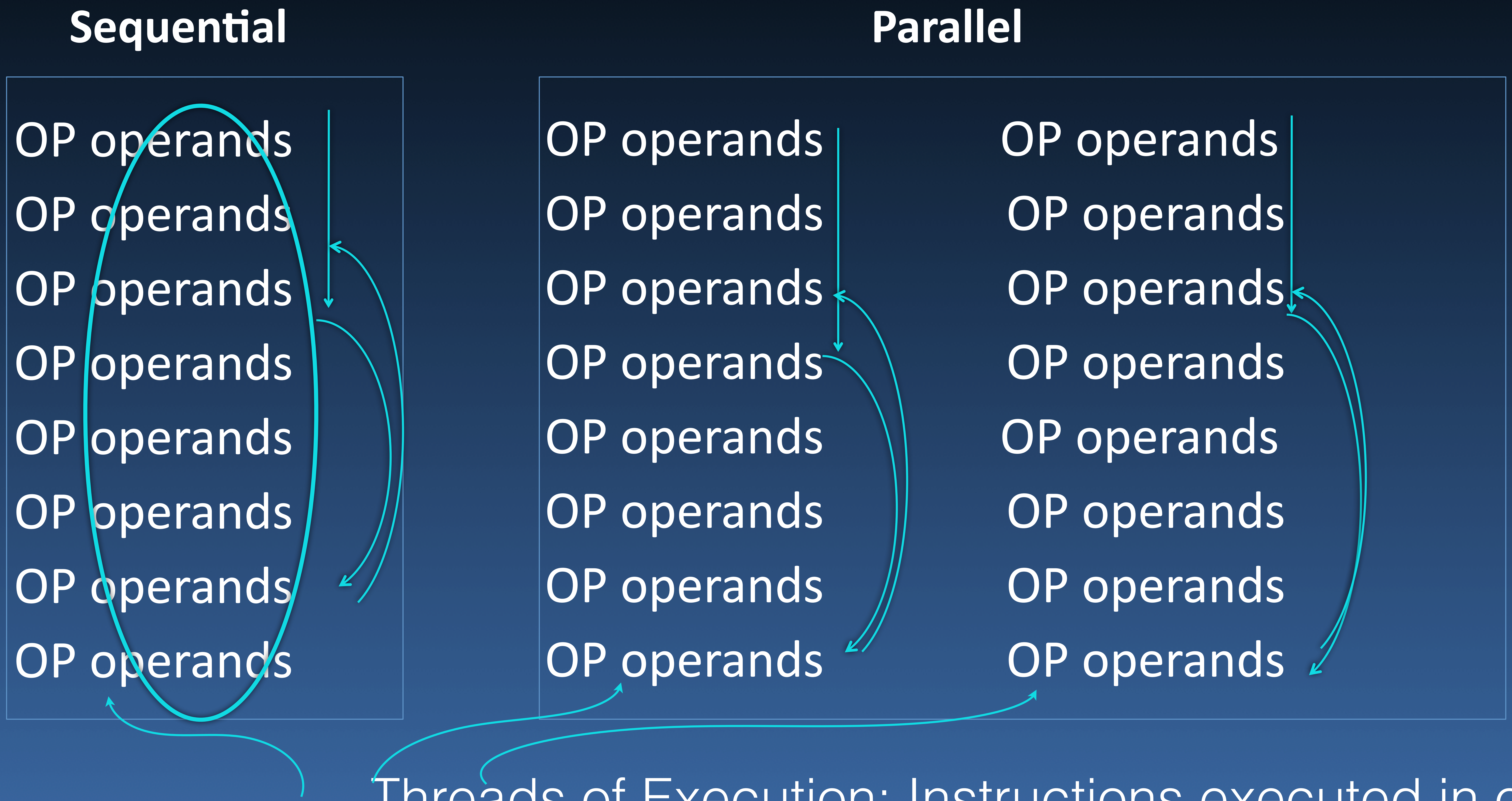
Introduction to
Parallel & Distributed Programming

Parallel & Distributed

- Parallel:
  - ➡ Focus on doing many things at the same time
- Distributed:
  - ➡ How the multiple things interact with each other
- Concurrency
  - ➡ Unordered



Threads of Execution



Threads of Execution: Instructions executed in order

Subodh Kumar

Why Parallel

- Can't clock faster ⇒ Do more per clock
  - ➡ Execute many simple instructions on many cores
- Can't continue to miniaturize
  - ➡ Wires and dimensions are too small, Cannot integrate at instruction level
  - ➡ External network, delay, bandwidth limitation

Software orchestration

- Not just compute more
  - ➡ Sometimes, the focus is on parallelizing data access (Memory, IO)
  - ➡ Multiple processors can access memory in parallel, disrupt caches

Complex Problems..

- Weather/Climate simulation
  - ➡ 3D-grid, Long duration simulation
- Data science
  - ➡ Filter, Join, Cross, Sort
- Financial processing
  - ➡ market prediction, investing, Blockchain
- Computational biology
  - ➡ drug design, gene sequencing, molecular simulation

Supercomputer

2021: “Fastest supercomputer in the world”

[HPL Rmax: 415000000000000000]

Nodes: 158,976

299,072

4.85 PB of total memory with

63 PB/s aggregate bandwidth

nect: 6D Torus

: 432

e: 21,000 SqFt

FUGAKU

Arm CPU

A64FX: 48 core CPU with 512-bit SIMD

Peak Flops: 3379G (DP) [70.4G/core]

Memory BW: 1 TB/s

N

Intel CPU

2.3 GHz x40 cores (~3000 GFlop)

+ 2x AVX-512 FMA units

Maximum Memory Speed: 3200 MHz

Memory Channels: 8

Memory bandwidth: ~200 GB/s

Nvidia GPU

1000+ Cores: 9.7 TF (DP)

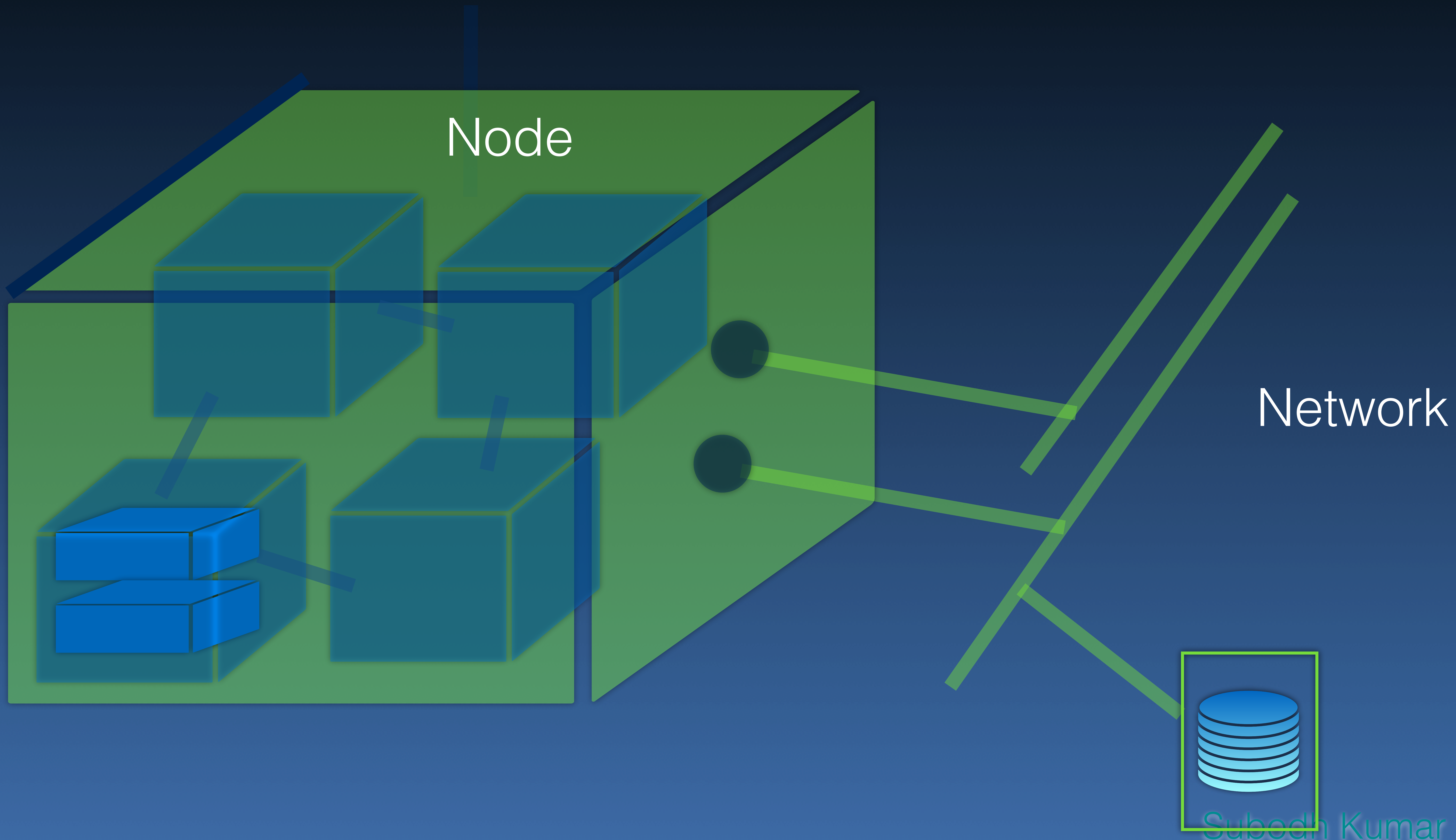
19.5 TF with Tensor Core

GPU Memory Bandwidth: 1.6 TB/s

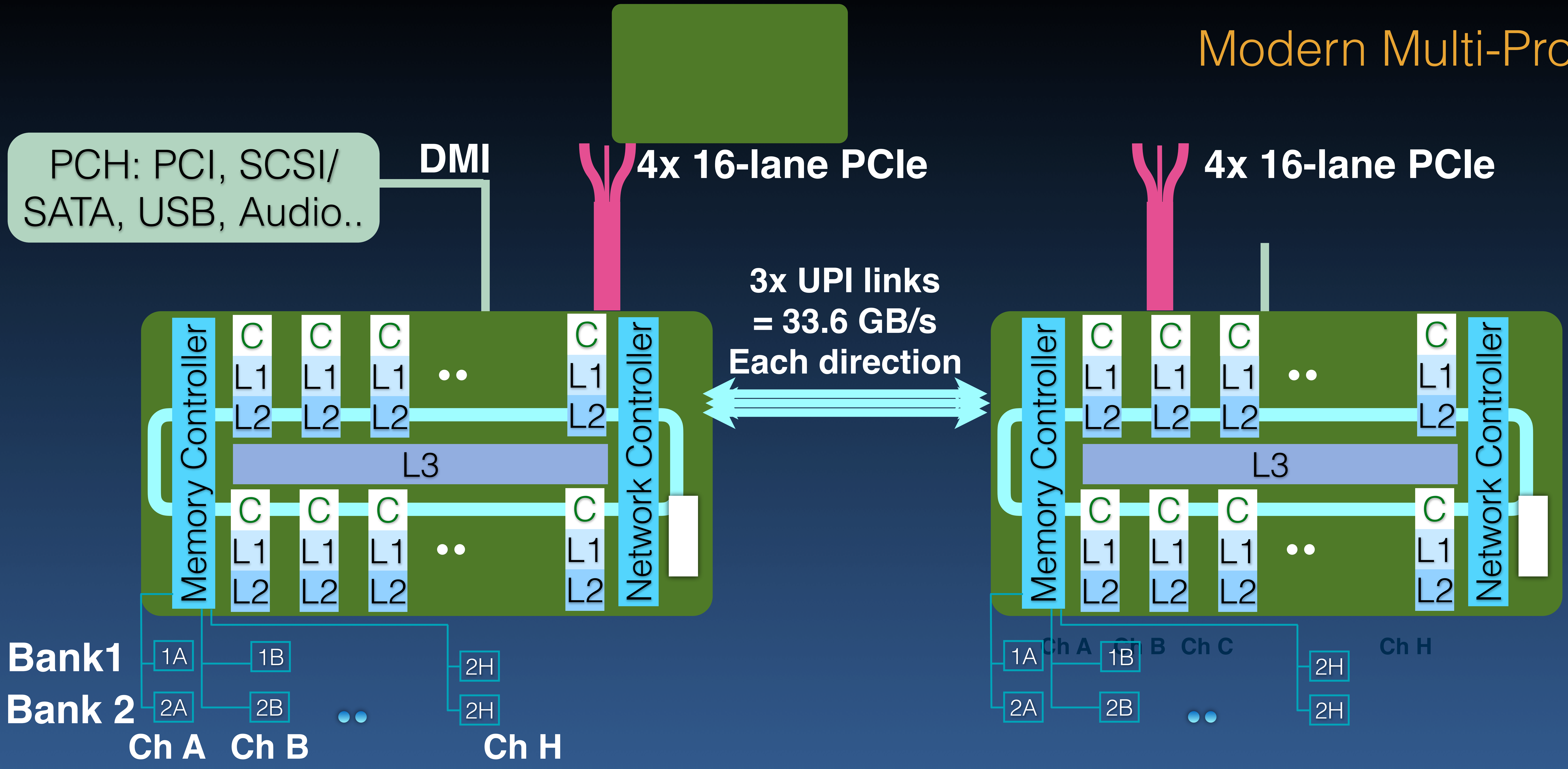
Network: NVLink 600 GB/s

Kumar

Parallel Computer



Modern Multi-Processor

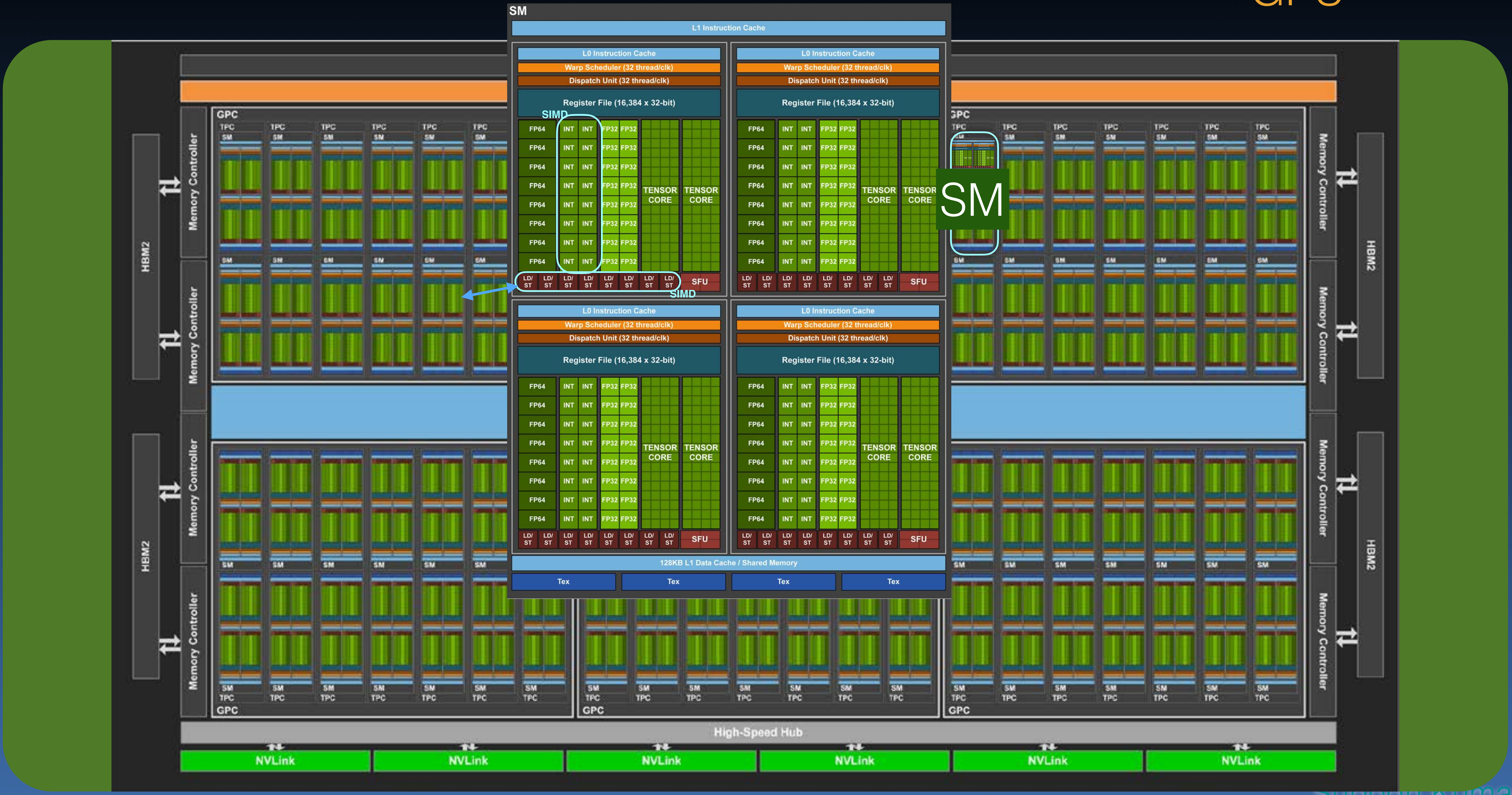


Graphics
Image Pr
Media en

8 channels of DDR4-3200 RAM/socket

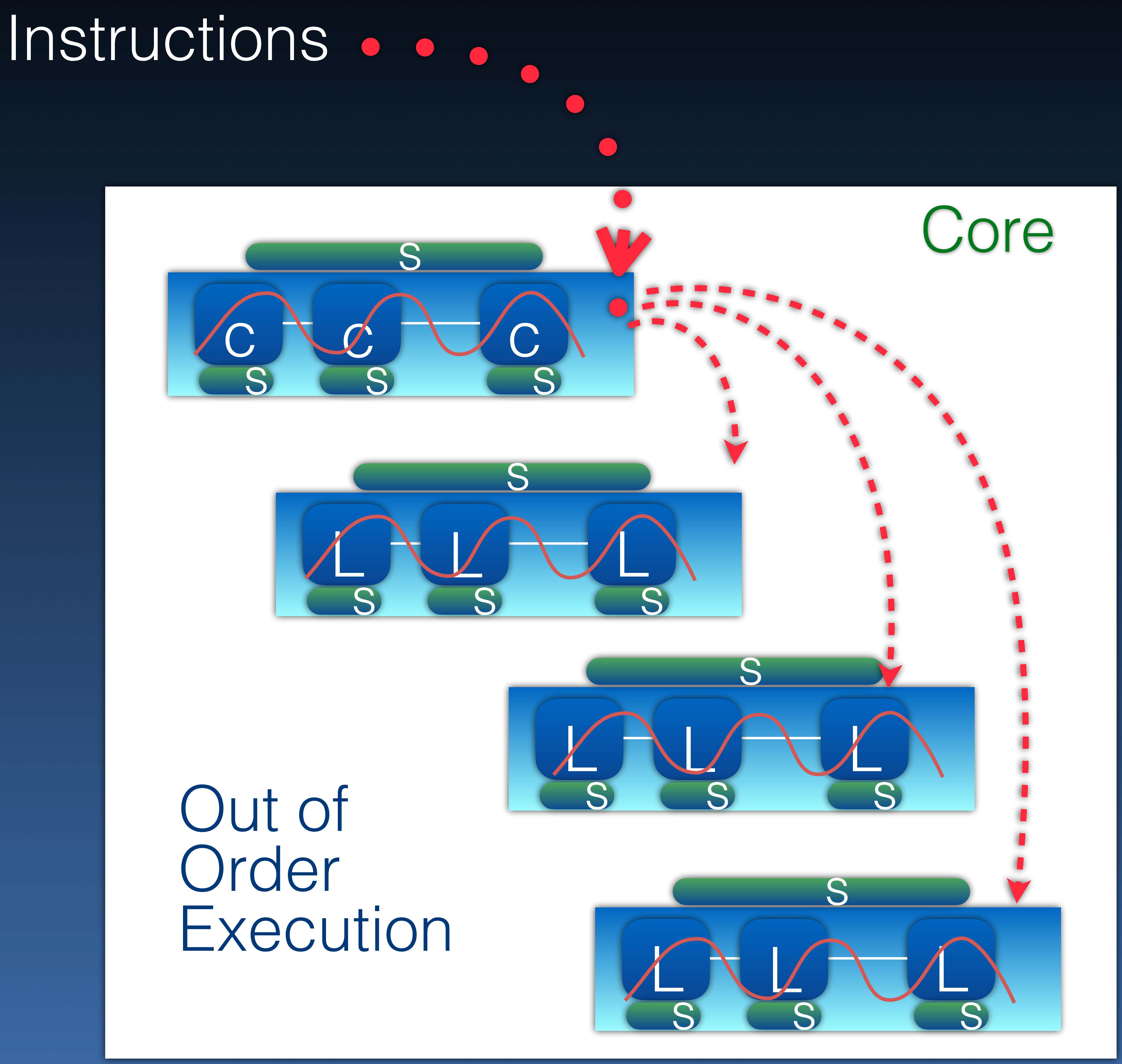
L1: 48 KB/core: 12-way assoc
L2: 512 KB/core: 8-way assoc
L3: 1.5 MB/core: 16-Way assoc (Sliced)

GPU



Subodh Kumar

NOT Sequential



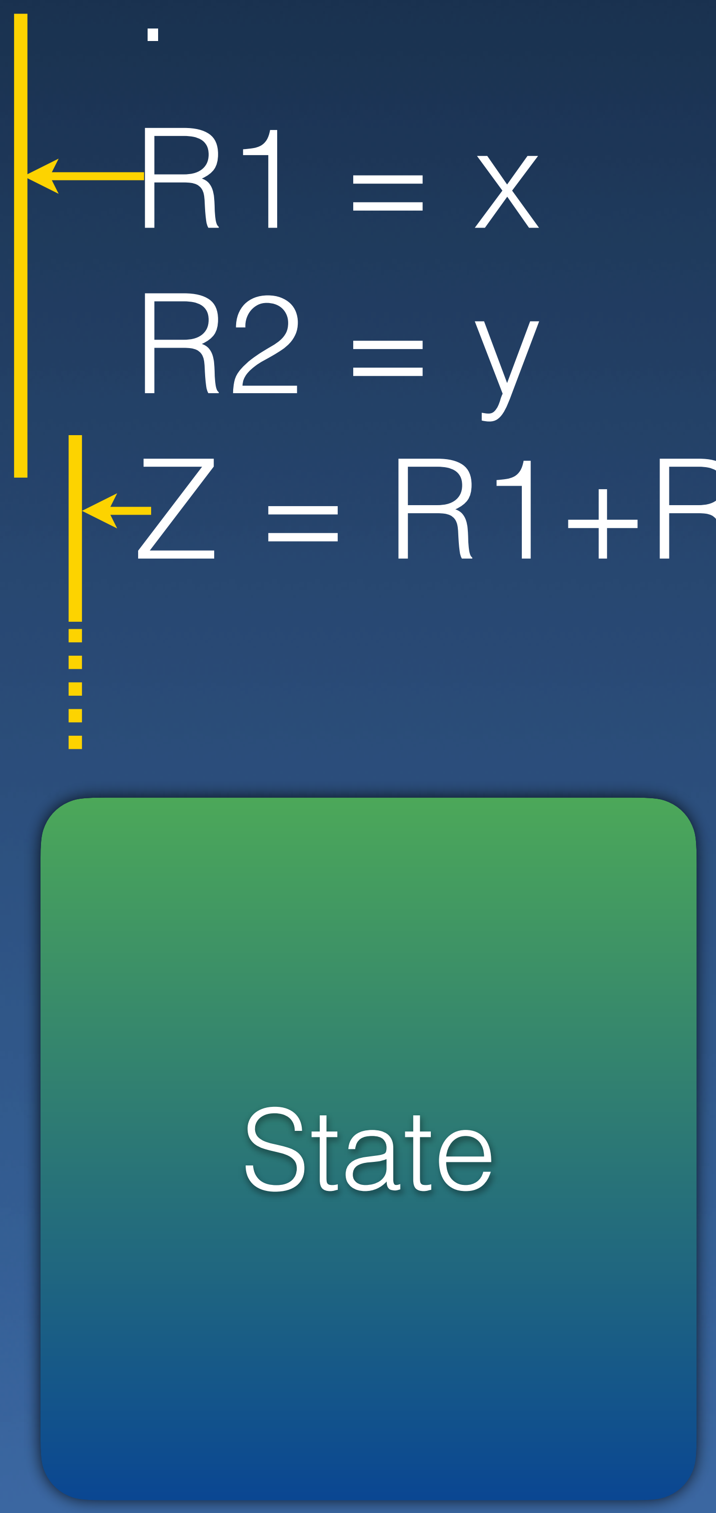
volatile int x;

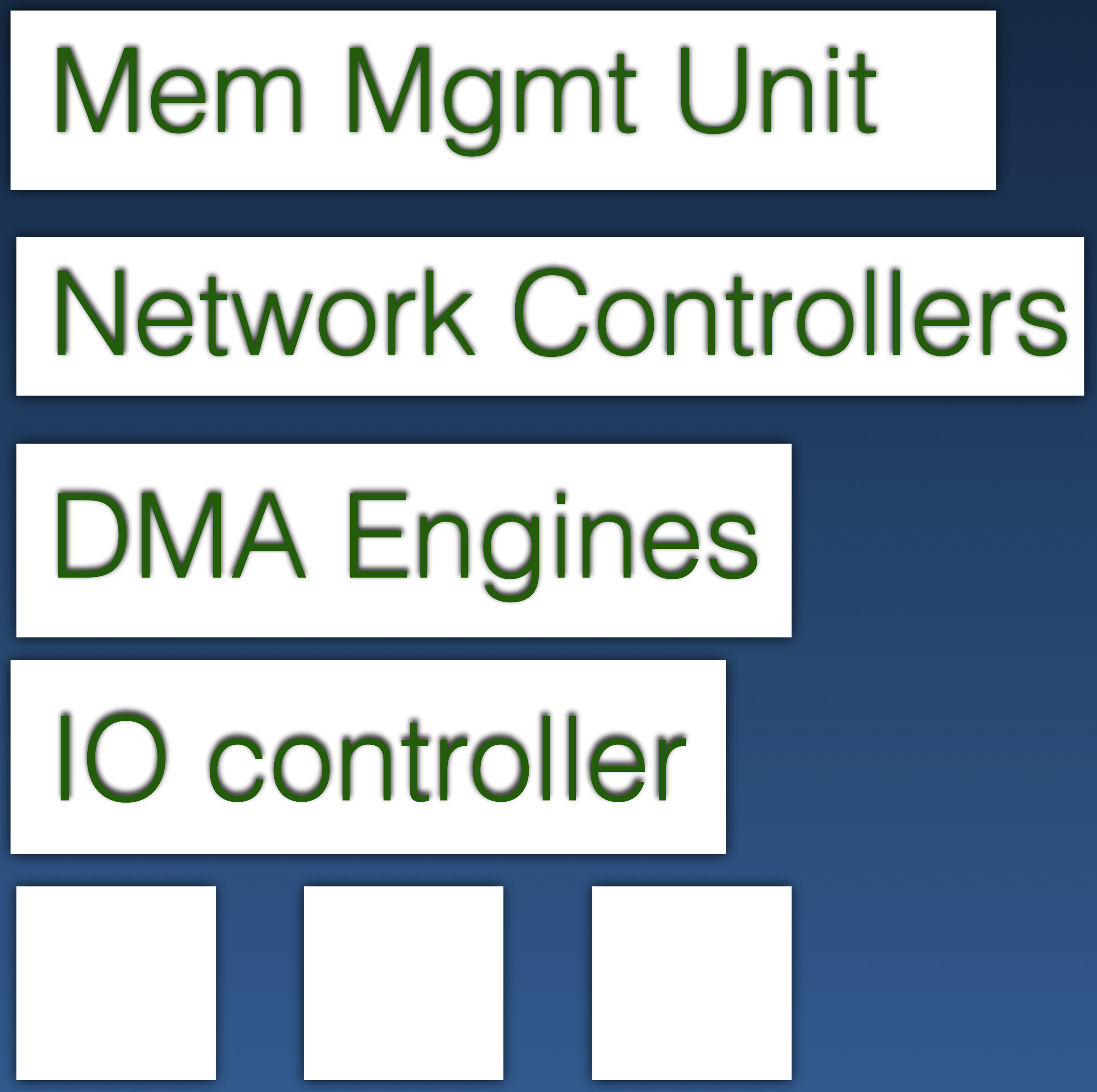
Access x

.

.

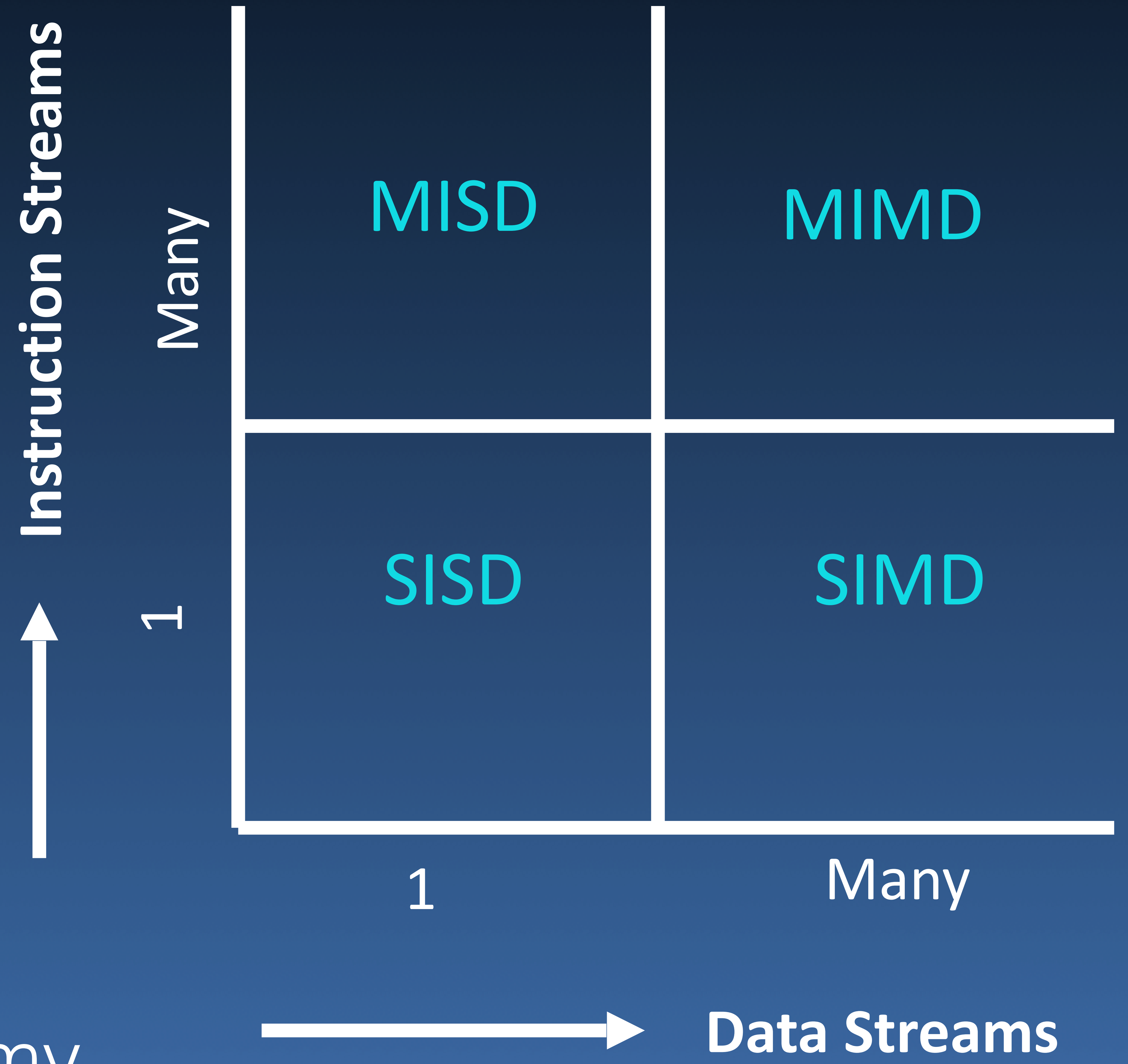
R1 = x
R2 = y
Z = R1+R2





Parallel Execution

- A number of instruction threads
- A number of data of data threads



Flynn's Taxonomy

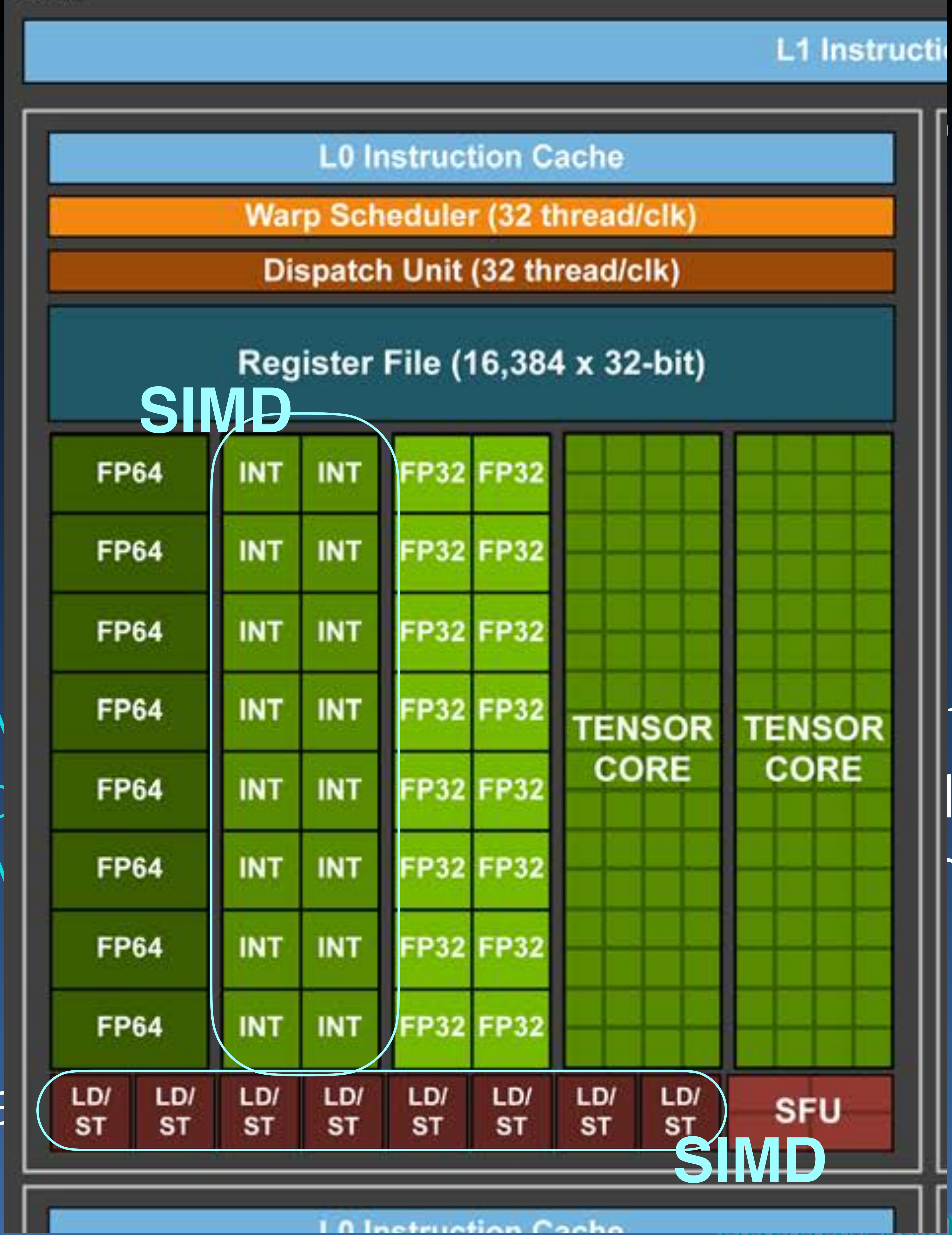
```
float *d1, *d2;
Loop: d1[I] += d2[i];
```

```
movss  xmm0,DWORD PTR [rdi+rax*1]
addss  xmm0,DWORD PTR [rsi+rax*1]
movss  DWORD PTR [rdi+rax*1],xmm0
```

mov
add
mov

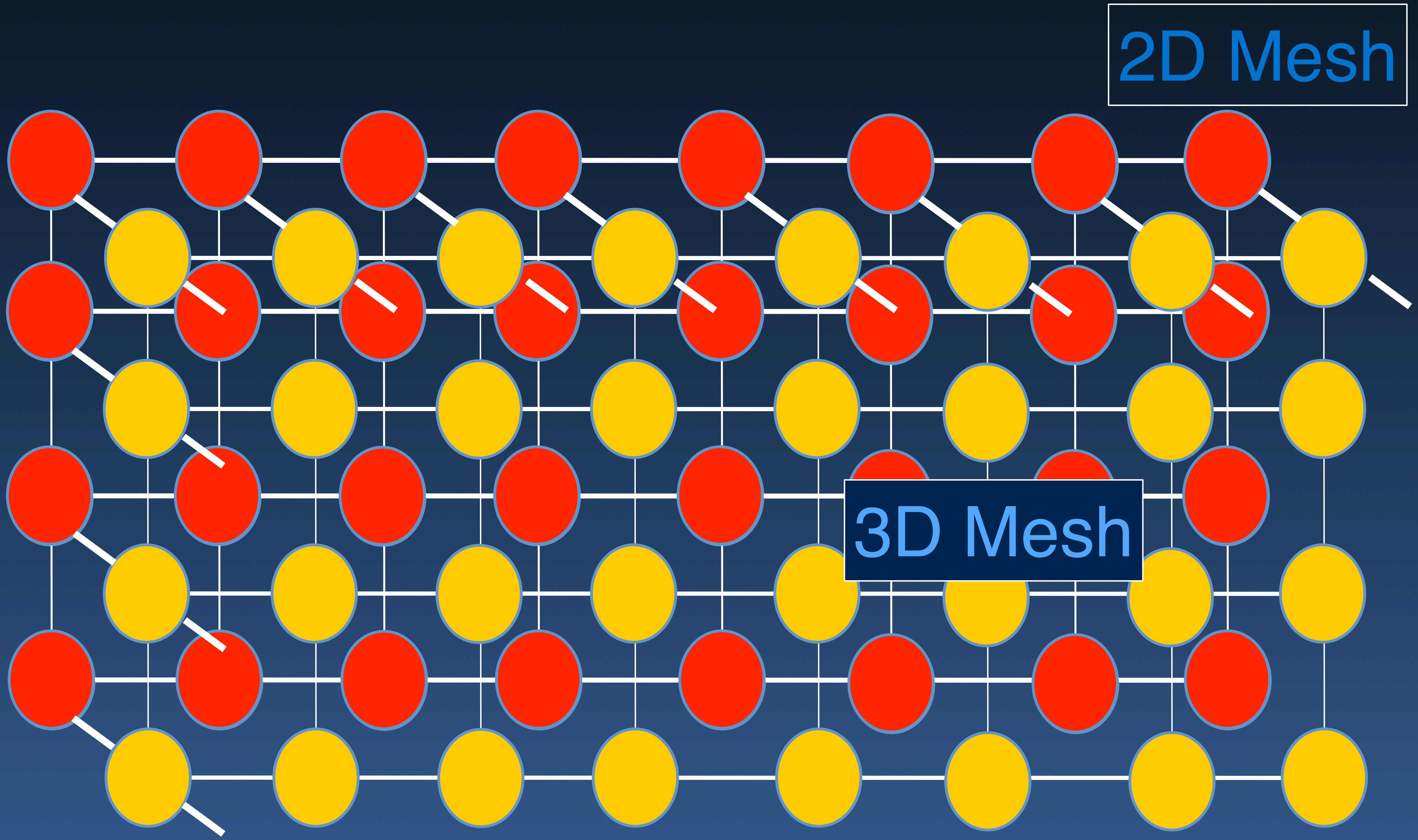
```
float *d1, *d2;
```

```
vmovss xmm0,DWORD PTR [rdi+rax*1]
vaddss xmm0,xmm0,DWORD PTR [rsi+ra
vmovss DWORD PTR [rdi+rax*1],xmm0
```



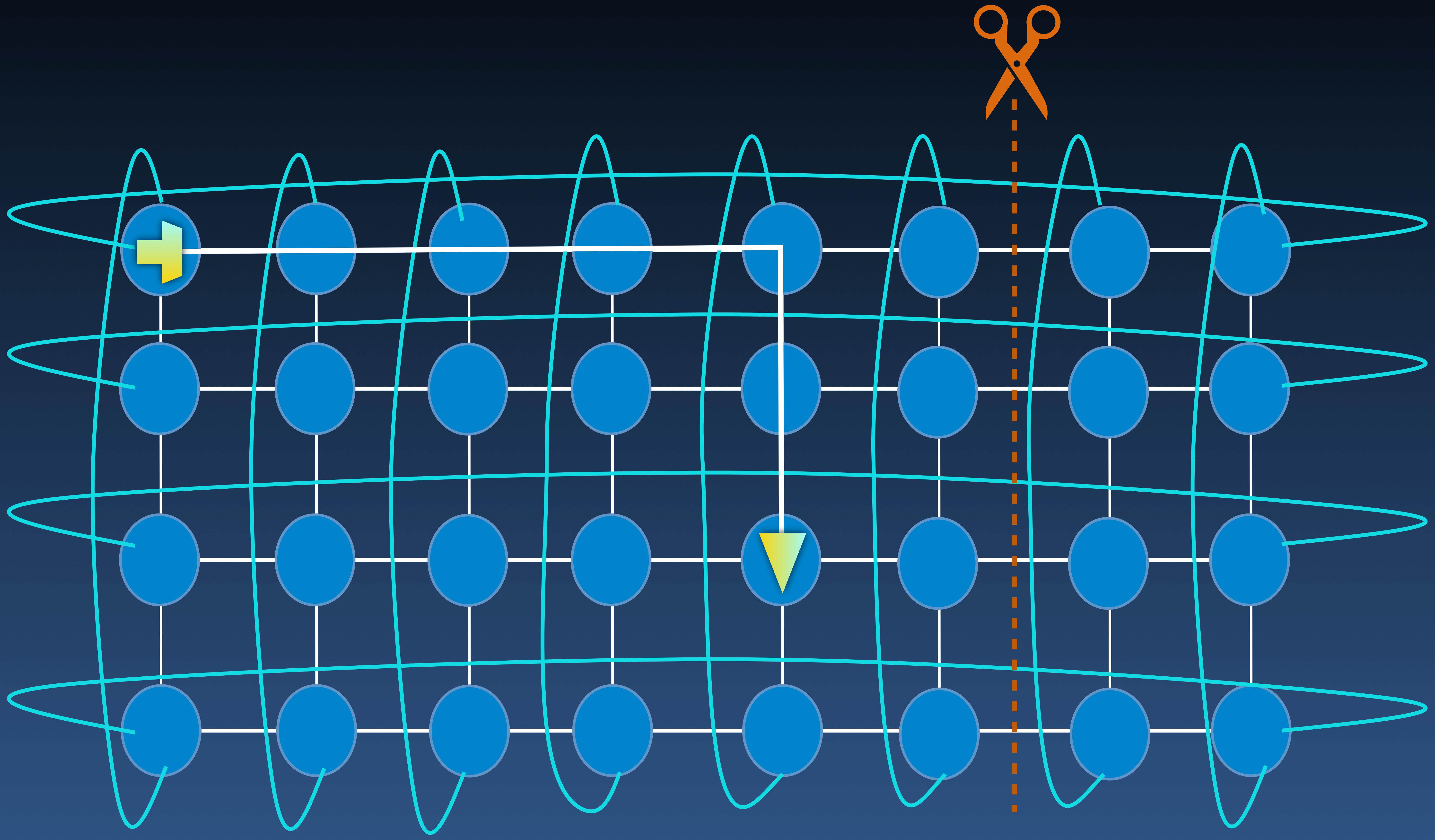
n
1]
1]
n0

Mesh Network



Subodh Kumar

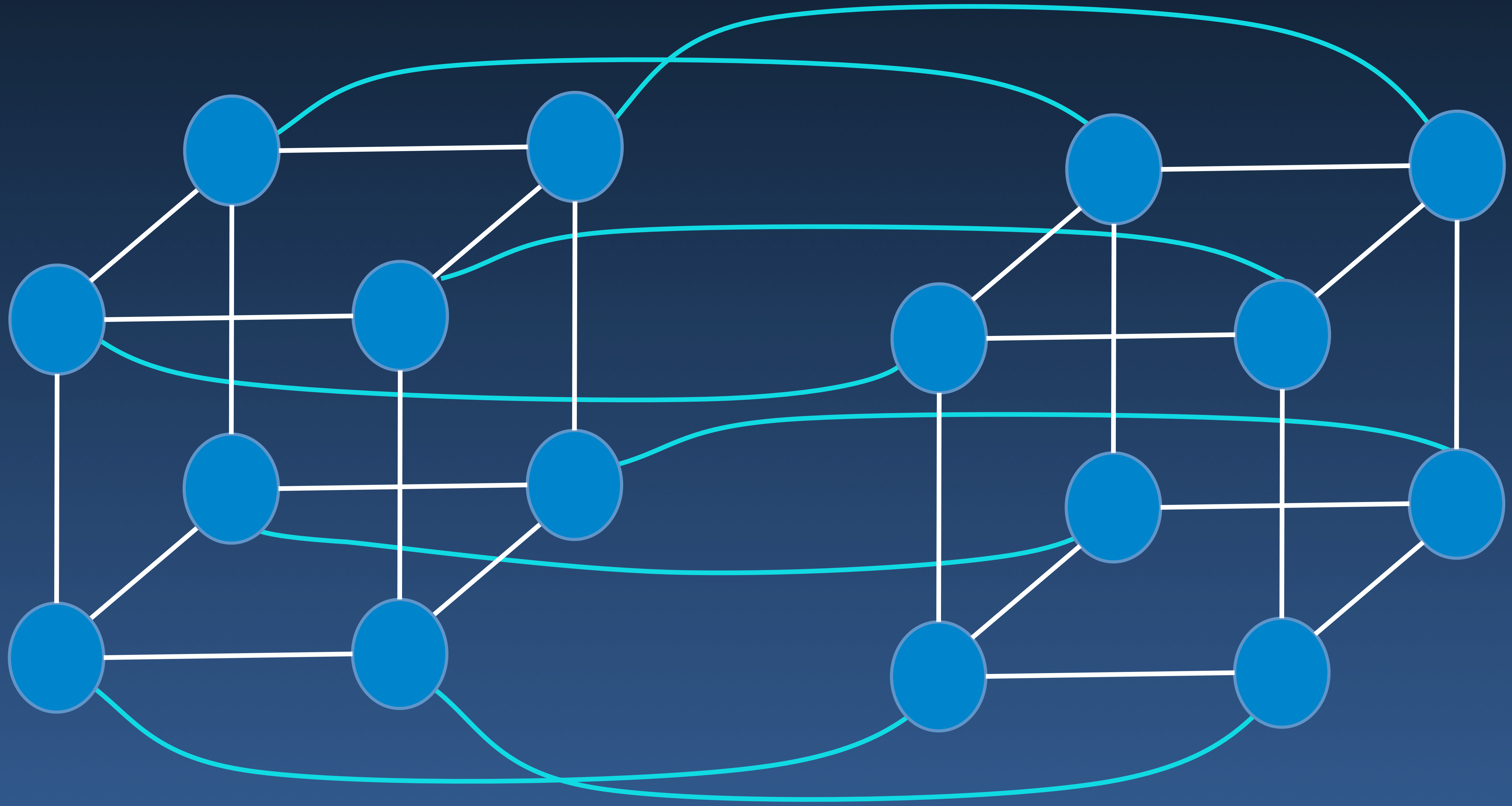
Torus Network



Circuit switching vs Packet switching

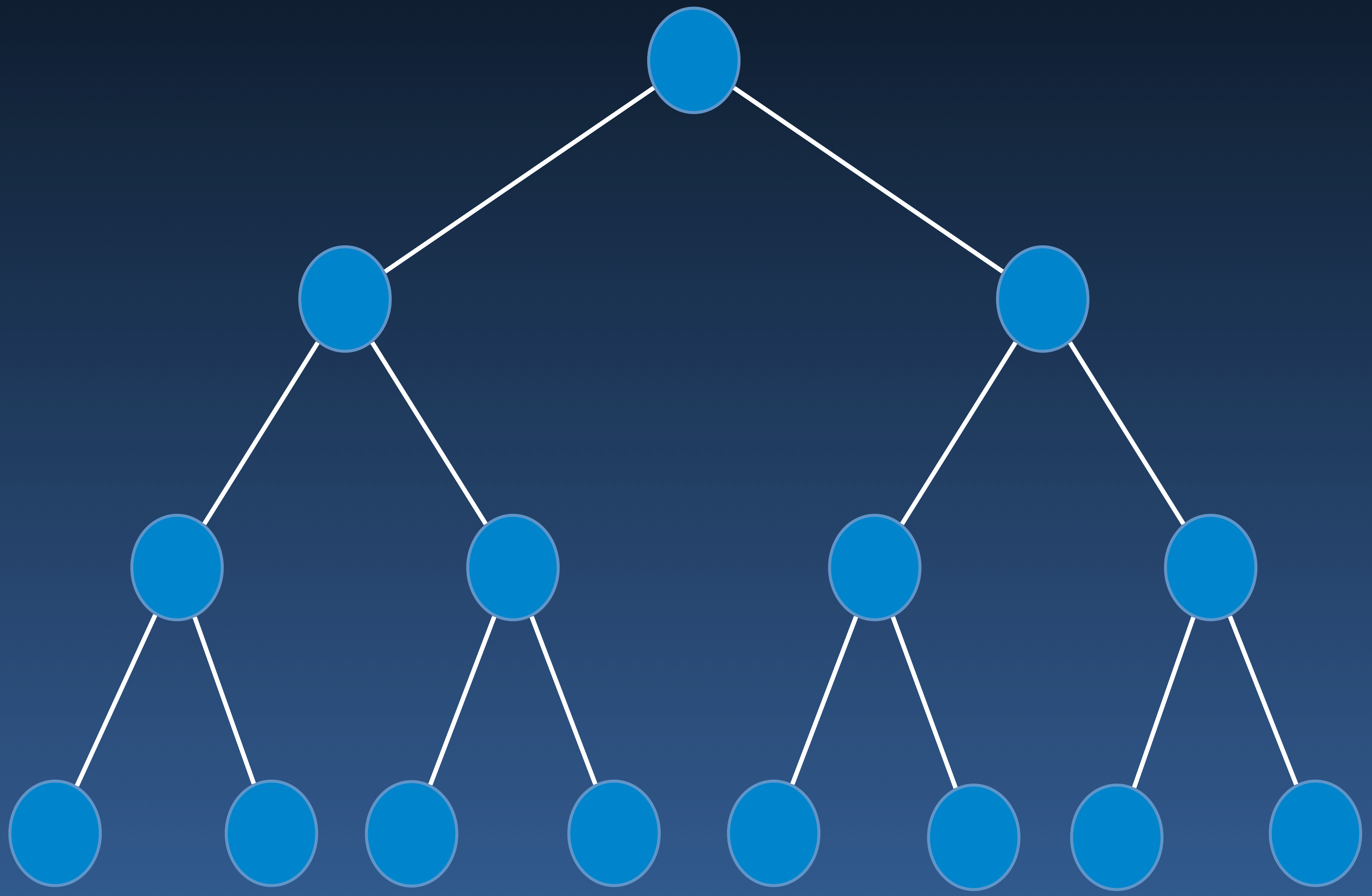
Diameter =?
Bisection width = ?
Blocking?

Hypercube



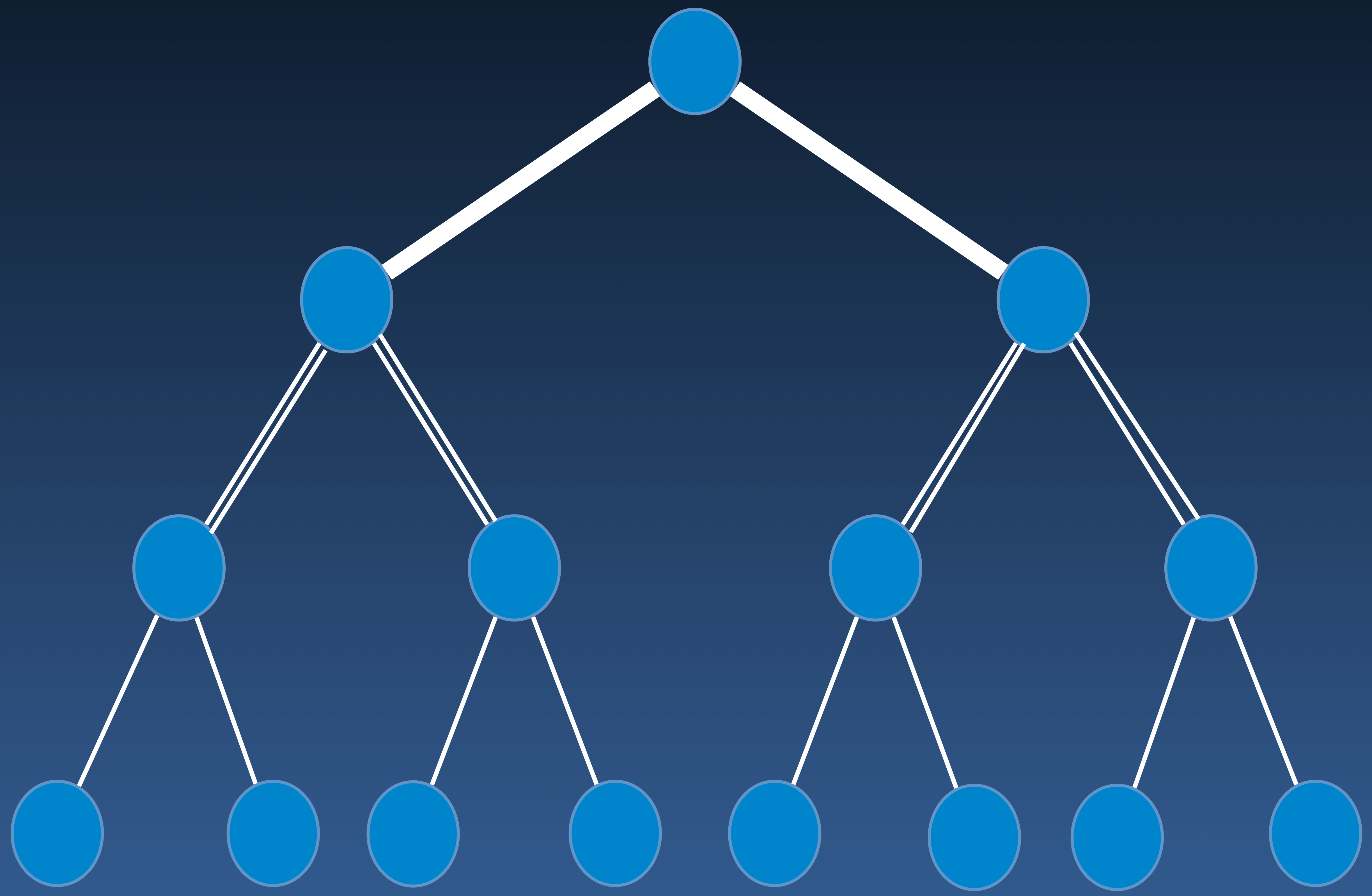
No. of links = ?
Diameter =?
Bisection width = ?

Tree Network



No. of links = ?
Diameter =?
Bisection width = ?

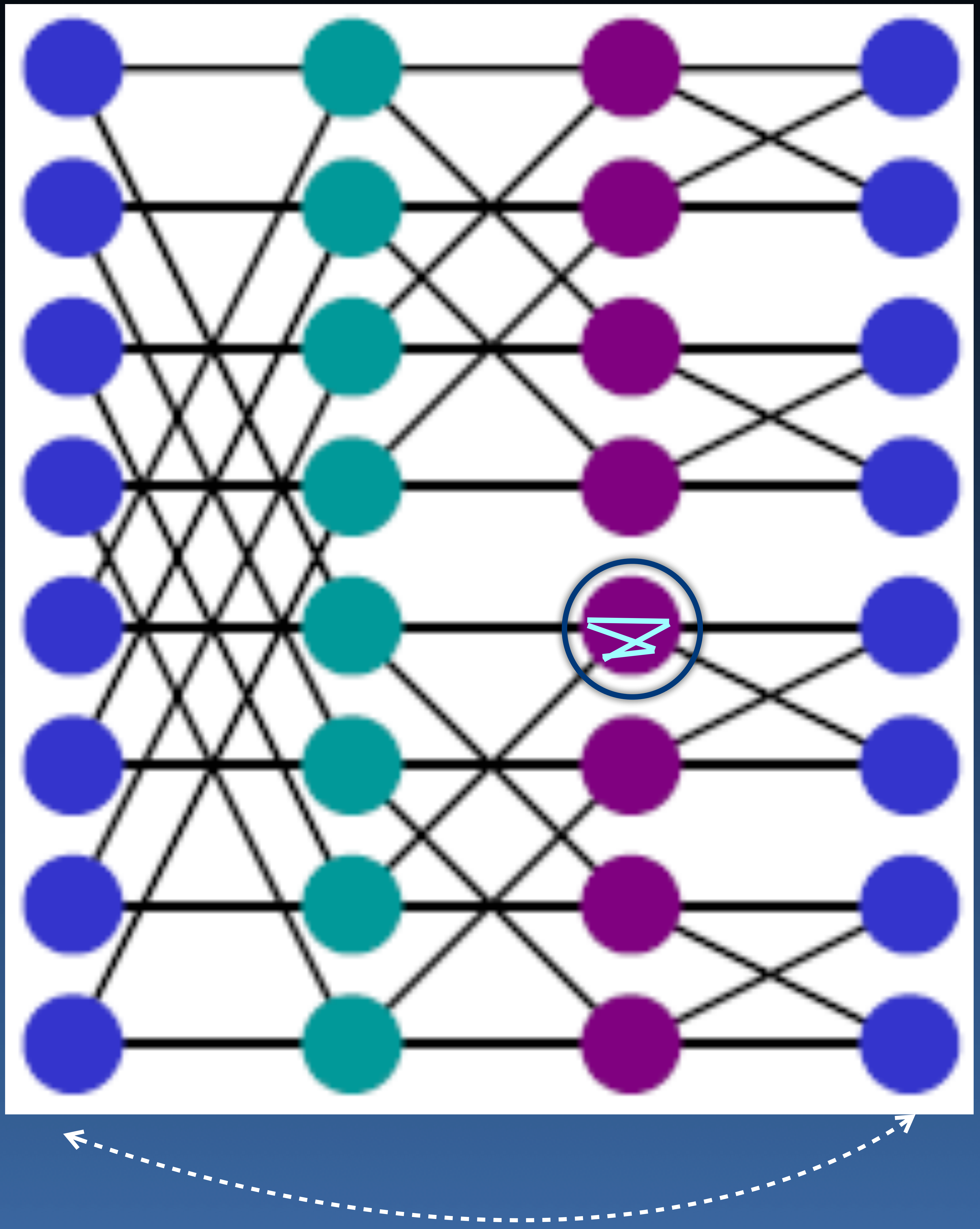
Fat Tree Network



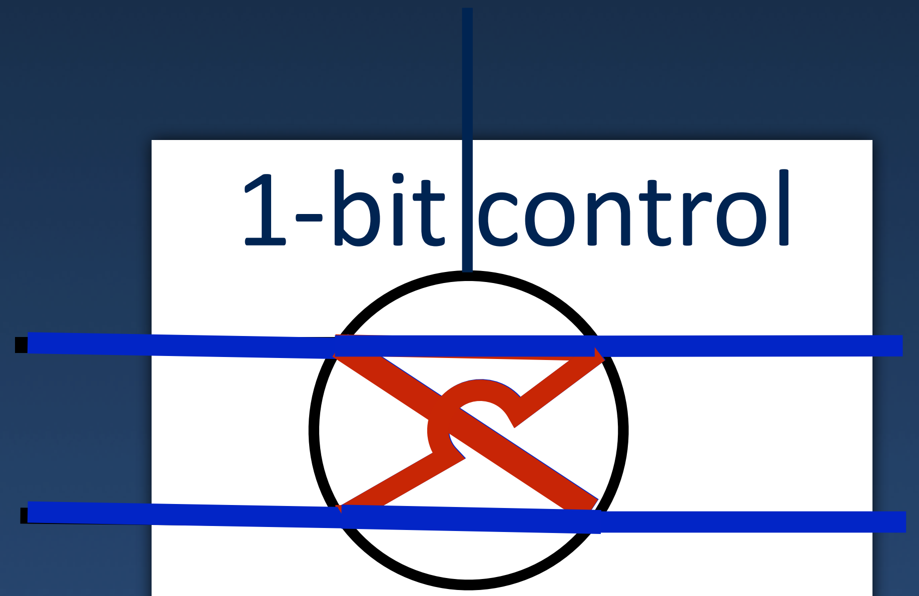
Subodh Kumar

Butterfly

Shuffle Exchange

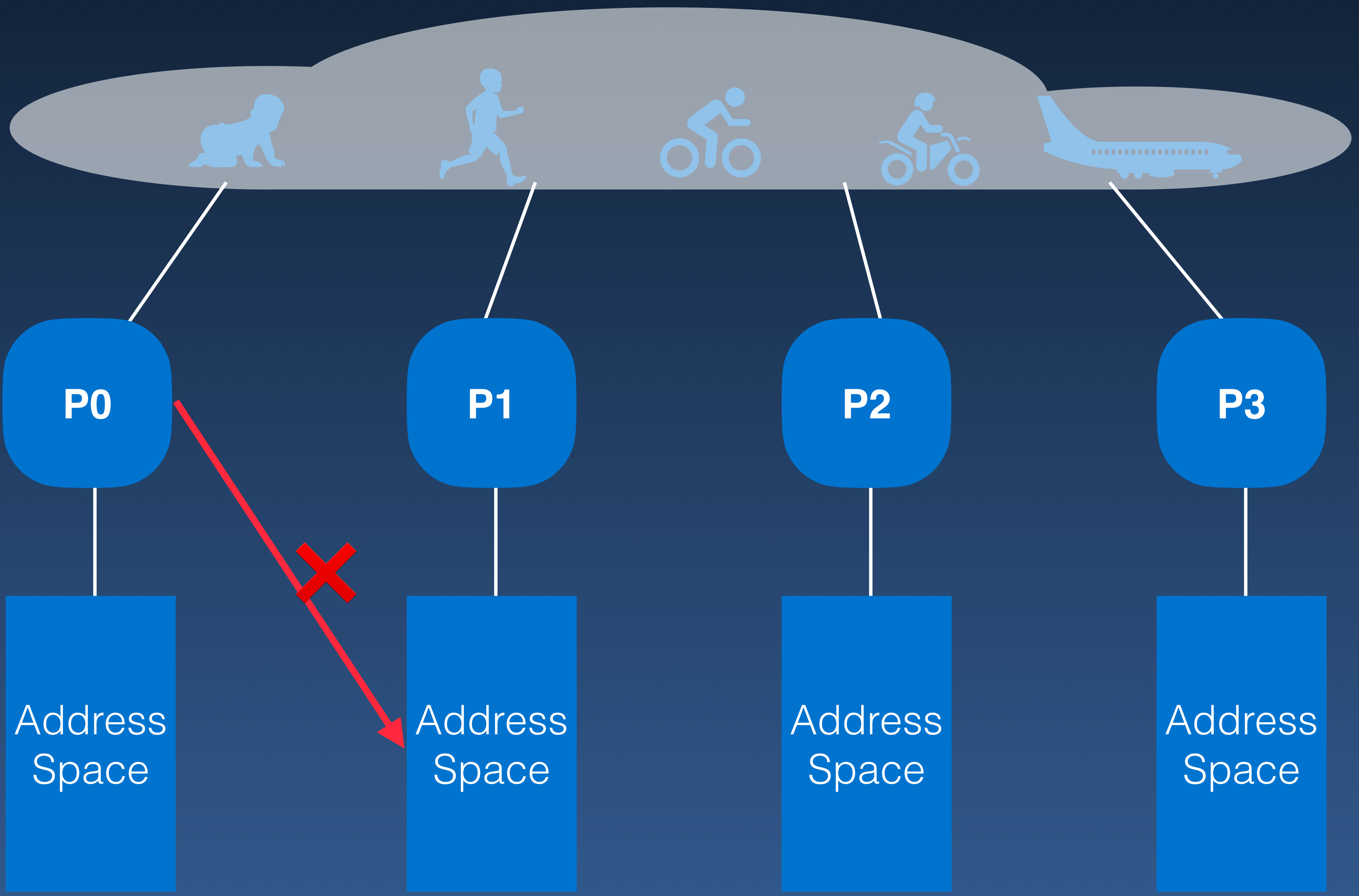


Multi-stage Network



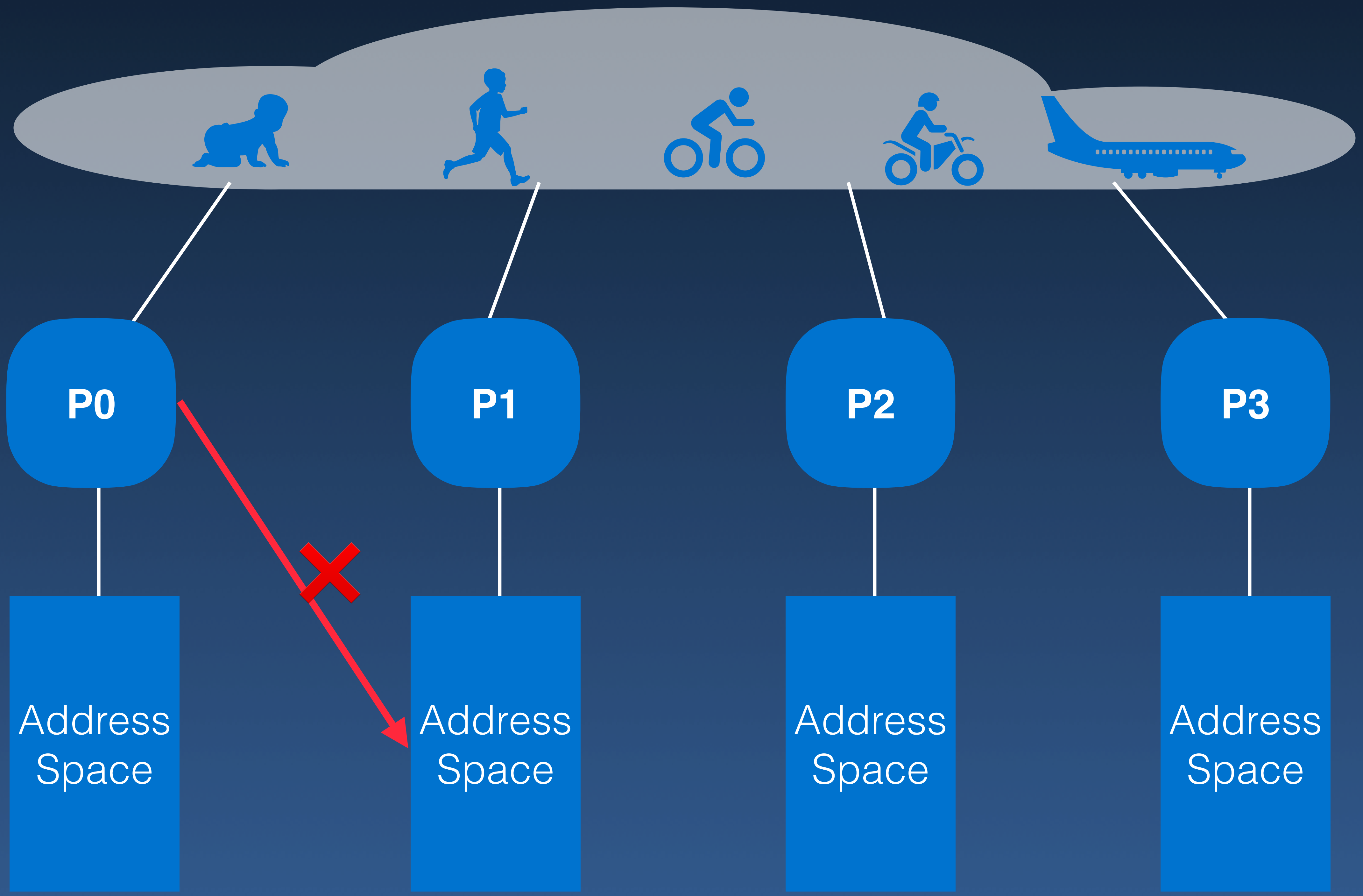
Pass-through or Crossover

Distributed Memory

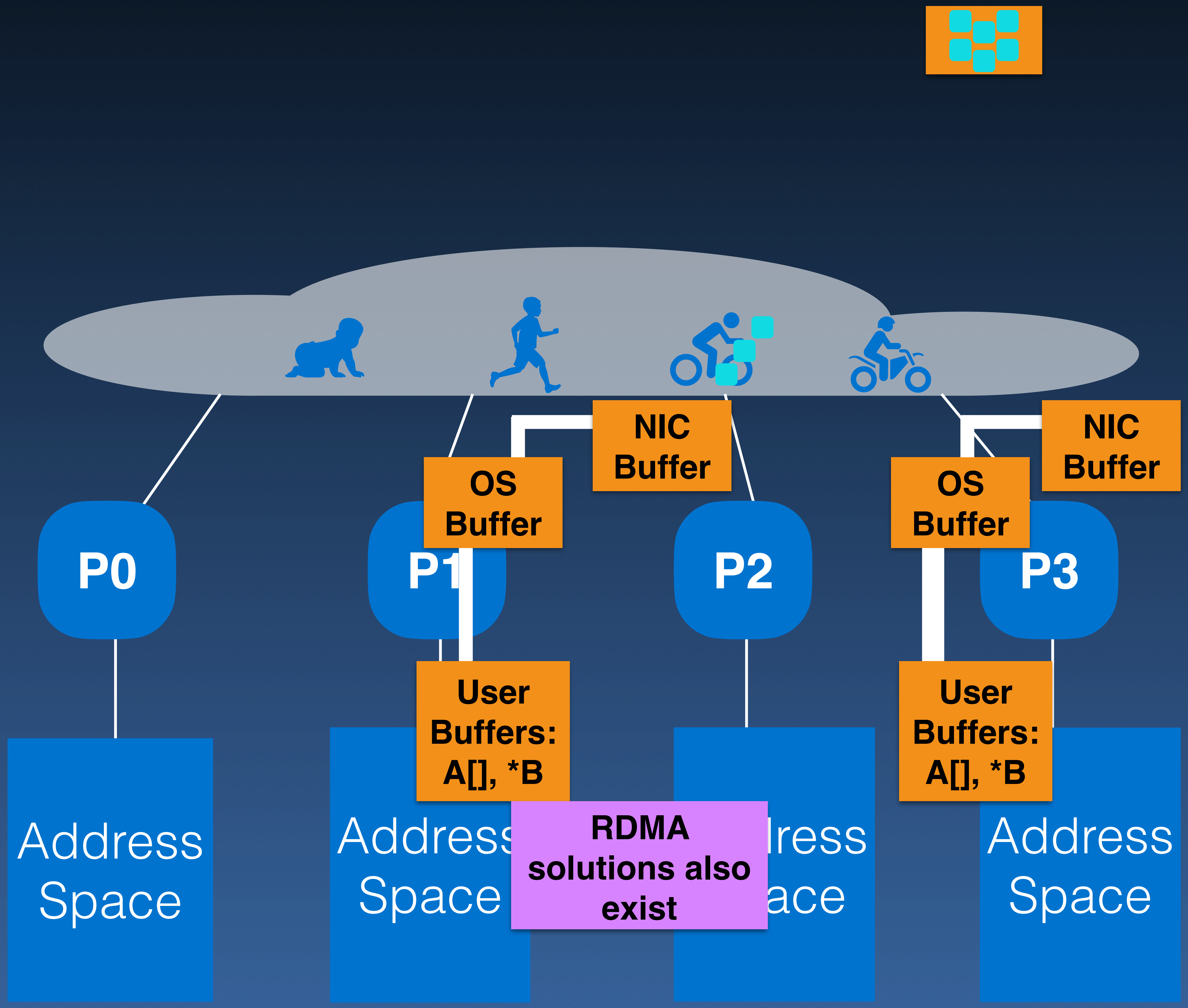


Subodh Kumar

Distributed Memory



Distributed


P0
P1
P2
P3
Address Space
Address Space
Address Space
Address Space
OS Buffer
NIC Buffer
OS Buffer
NIC Buffer
User Buffers: A[], *B
User Buffers: A[], *B
RDMA solutions also exist