# SIMULATION

● HYPERPARAMETERS

- Num_experts = 4
- Seqlen = 2
- Top_K = 2

generate_routing ( ) → routing = $\begin{bmatrix} 2, & 1 \\ 0, & 1 \end{bmatrix}$

generate_routing ( ) → weights = $\begin{bmatrix} 0.63, & 0.12 \\ 0.42, & 0.30 \end{bmatrix}$

token 0

- Hot_ratio = 0.5
- Hot_weight = 0.8

=) 50% of experts get 80% of routings

weights → npu_identify ( ) → labelled weights

$$\begin{bmatrix} 0.63, & 0.12 \\ 0.42, & 0.30 \end{bmatrix}$$

$$\left[\begin{array}{cccc} 0.63, & 0.12, & 0 & 0 \\ 0.42, & 0.30, & 1 & 1 \end{array}\right]$$

src npu id

token id

# PERFORMANCE MODEL

labelled weights
+
routing → convert_to_bytes() → load dictionary

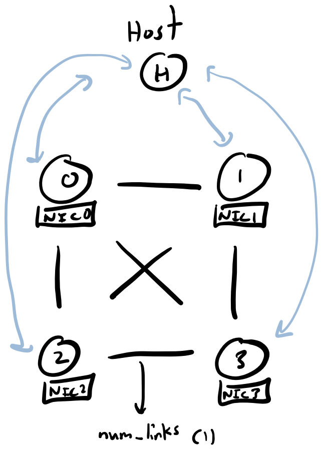
{0: {0:0, 1:2}, 1: {0:3, 1:0}}

src dest load

* Num_links = 1

* Num_nodes = 4

## Mock 1st round



* initial cpu delay

loop through load and send packets until one or multiple of these:

1) No more load to send (DONE)

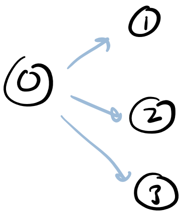
2) No more available links to send remaining load

3) No more bandwidth in desired links (leads to 2))

round_time = (most packets in one link) * *packet prep delay

+ (largest bandwidth usage) / *intra_bw

+ *Base_delay GPU → CPU confirmation



## Load

{0:{1:3, 2:4, 3:1}}

## Order of packets sent

#1) 0→1 2B

#2) 0→2 2B

#3) 0→3 1B

#4) 0→1 1B

#5) 0→2 2B

## Assuming a balanced routing,

$$\text{Round\_time} = \left(\ast\text{initial\_cpu\_delay} + \ast\text{base\_delay} + \left\lfloor\left(\frac{\ast\text{intra\_bw}}{\ast\text{packet\_size}}\right)\right\rfloor\left(\ast\text{packet\_prep\_delay} + \frac{\ast\text{packet\_size}}{\ast\text{intra\_bw}}\right)\right.$$

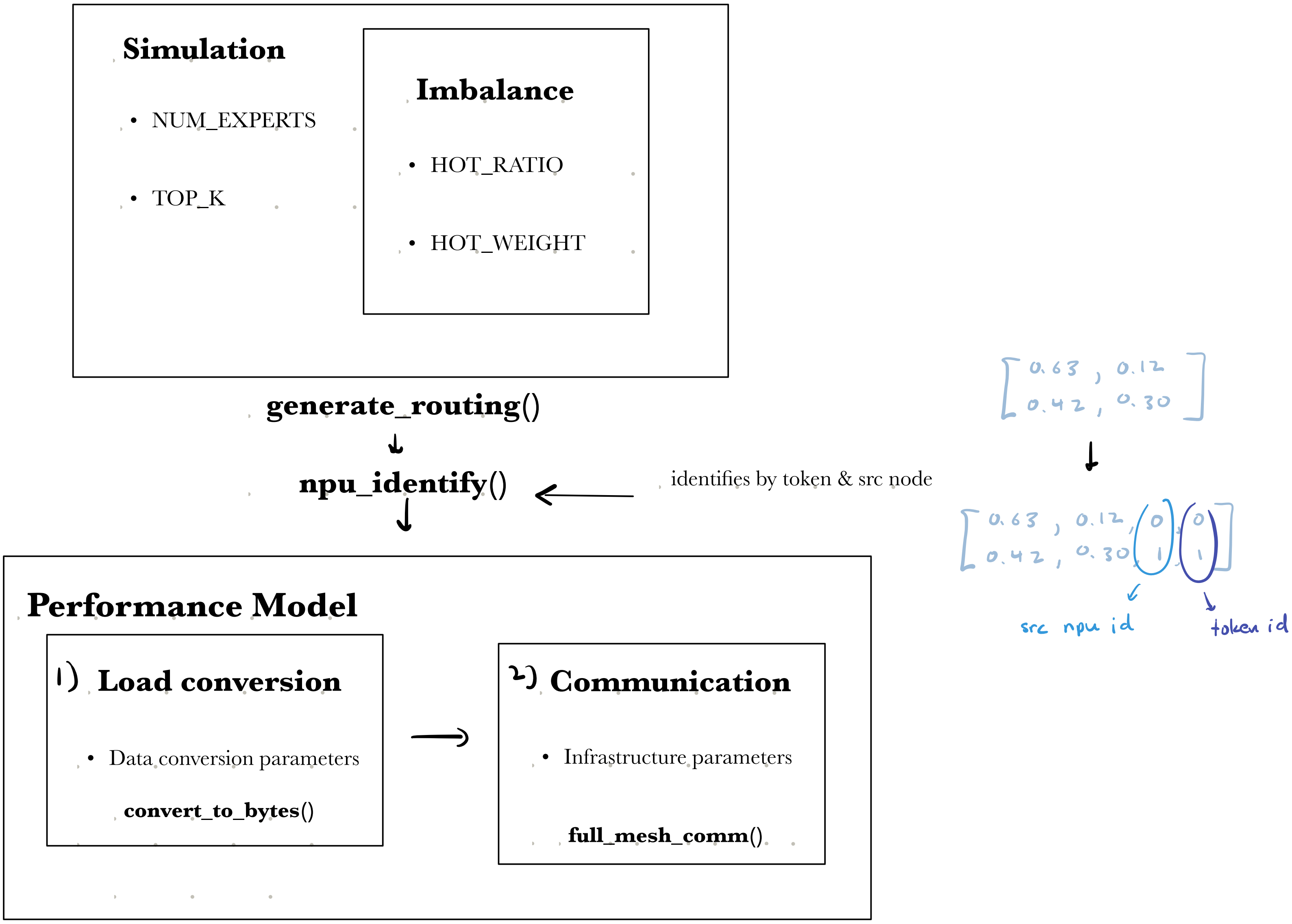
$$\text{Num\_round} = \frac{\ast\ \text{load\_per\_node} \cdot \ast\text{num\_nodes}}{\left(\frac{\ast\text{num\_links} \cdot \ast\text{num\_nodes} \cdot (\ast\text{num\_nodes} - 1)}{2}\right) \cdot \ast\text{intra\_bw}}$$

$$\text{Total\_time} = \text{Round\_time} \cdot \text{Num\_rounds}$$

$\ast$ load = {0: {0:0, 1:10, 2:10},
1 = {0:10, 1:0, 2:10},
2 : {0:10, 1:10, 2:0}}

⇓

load_per_node = 20



**Simulation**

- NUM_EXPERTS
- TOP_K

**Imbalance**

- HOT_RATIO
- HOT_WEIGHT

**generate_routing()**

**npu_identify()**

identifies by token & src node

**Performance Model**

1) **Load conversion**

- Data conversion parameters

**convert_to_bytes()**

2) **Communication**

- Infrastructure parameters

**full_mesh_comm()**

$$\begin{bmatrix} 0.63 , & 0.12 \\ 0.42 , & 0.30 \end{bmatrix}$$

$$\begin{bmatrix} 0.63 , & 0.12 , & \begin{pmatrix} 0 \\ 1 \end{pmatrix} , & \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0.42 , & 0.30 & & \end{bmatrix}$$

src npu id

token id