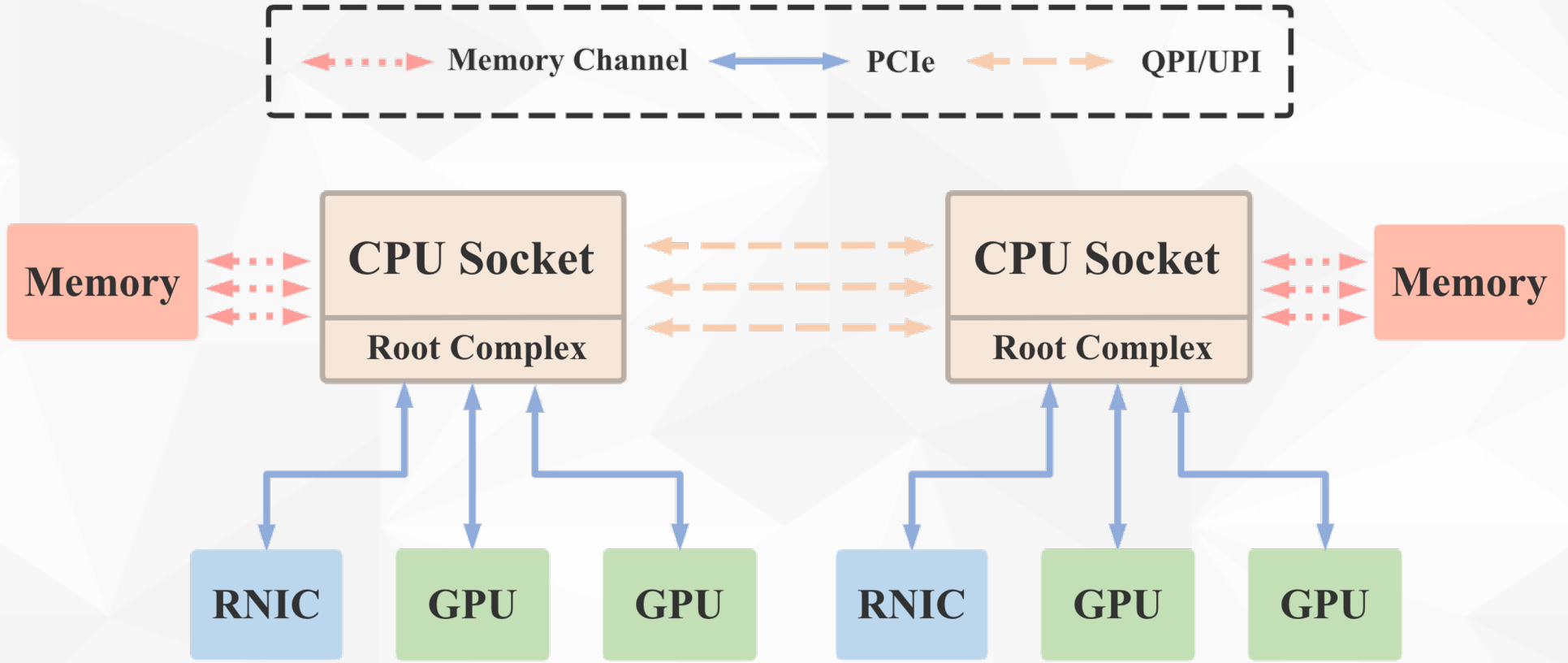
Hostping: Diagnosing Intra-host Network Bottlenecks in RDMA Servers

Kefei Liu, Zhuo Jiang, Jiao Zhang, Haoran Wei, Xiaolong Zhong
Lizhuang Tan, Tian Pan and Tao Huang

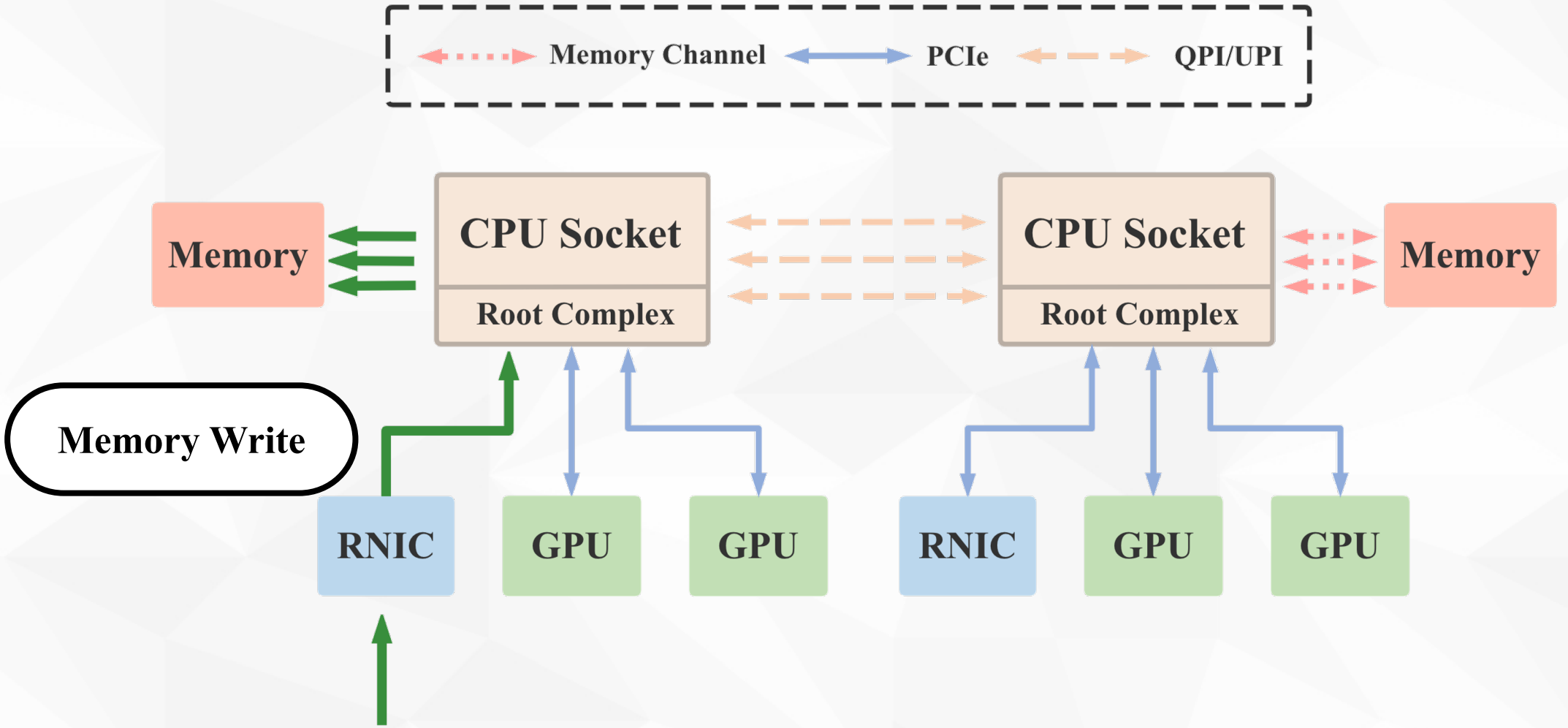
Background & Motivation

■ Intra-host Network Bottlenecks



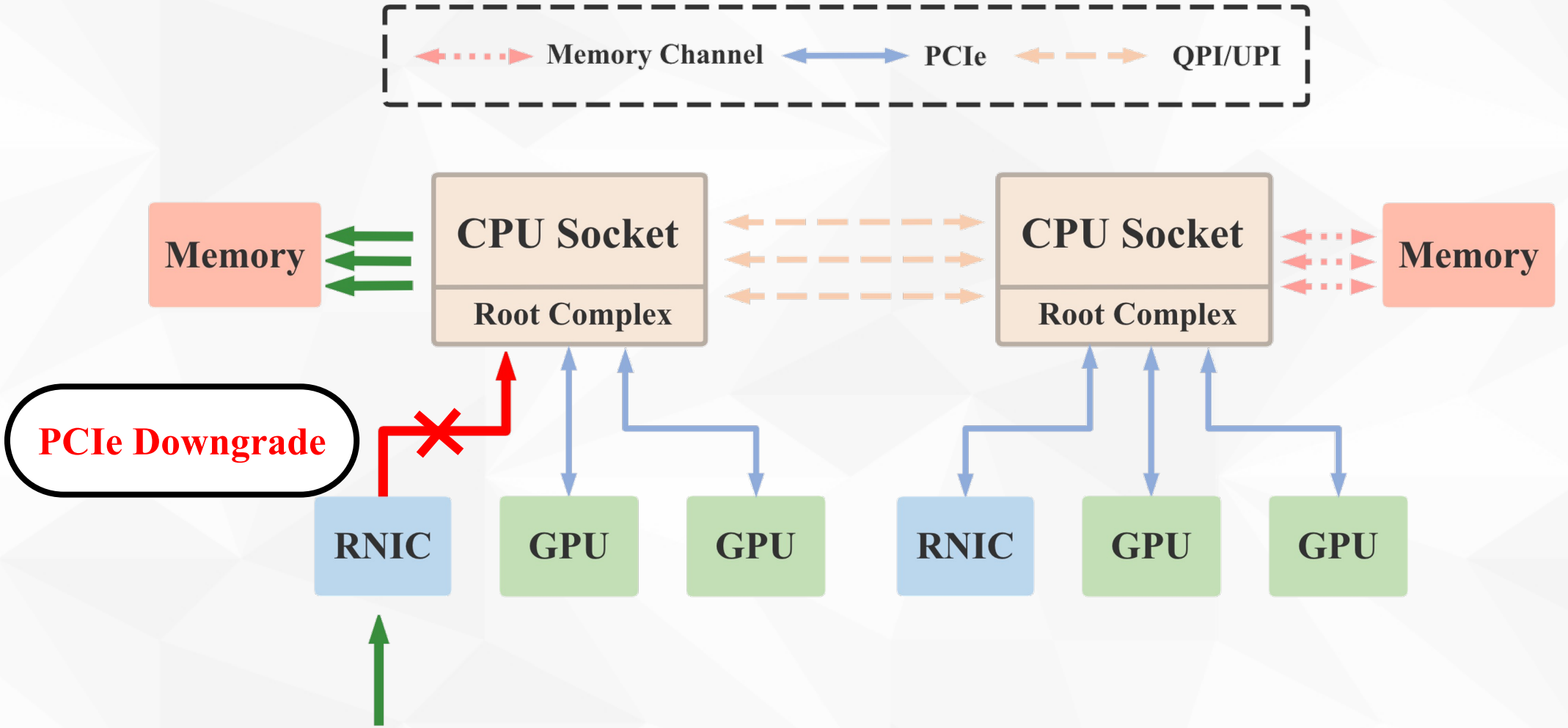
Background & Motivation

■ Intra-host Network Bottlenecks



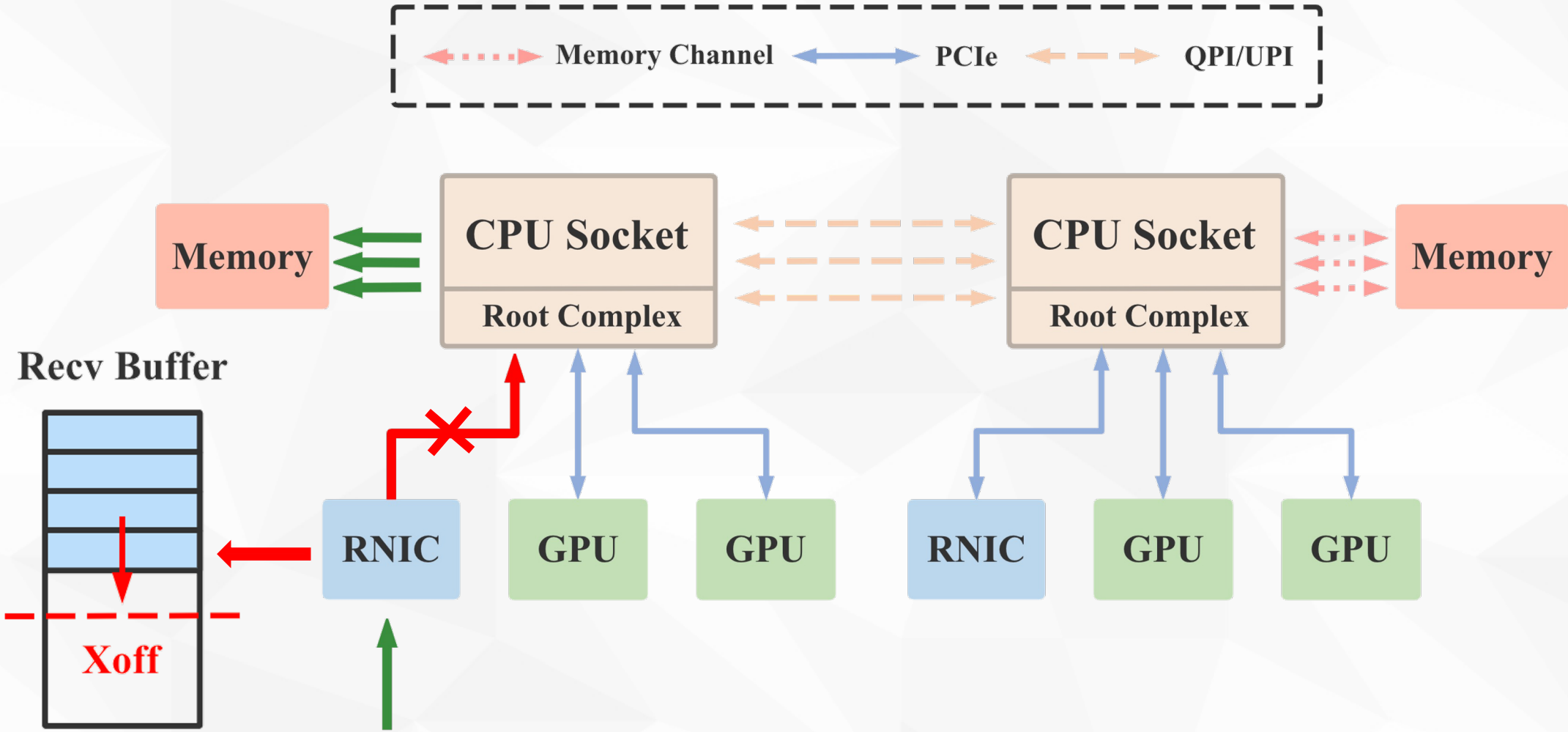
Background & Motivation

■ Intra-host Network Bottlenecks



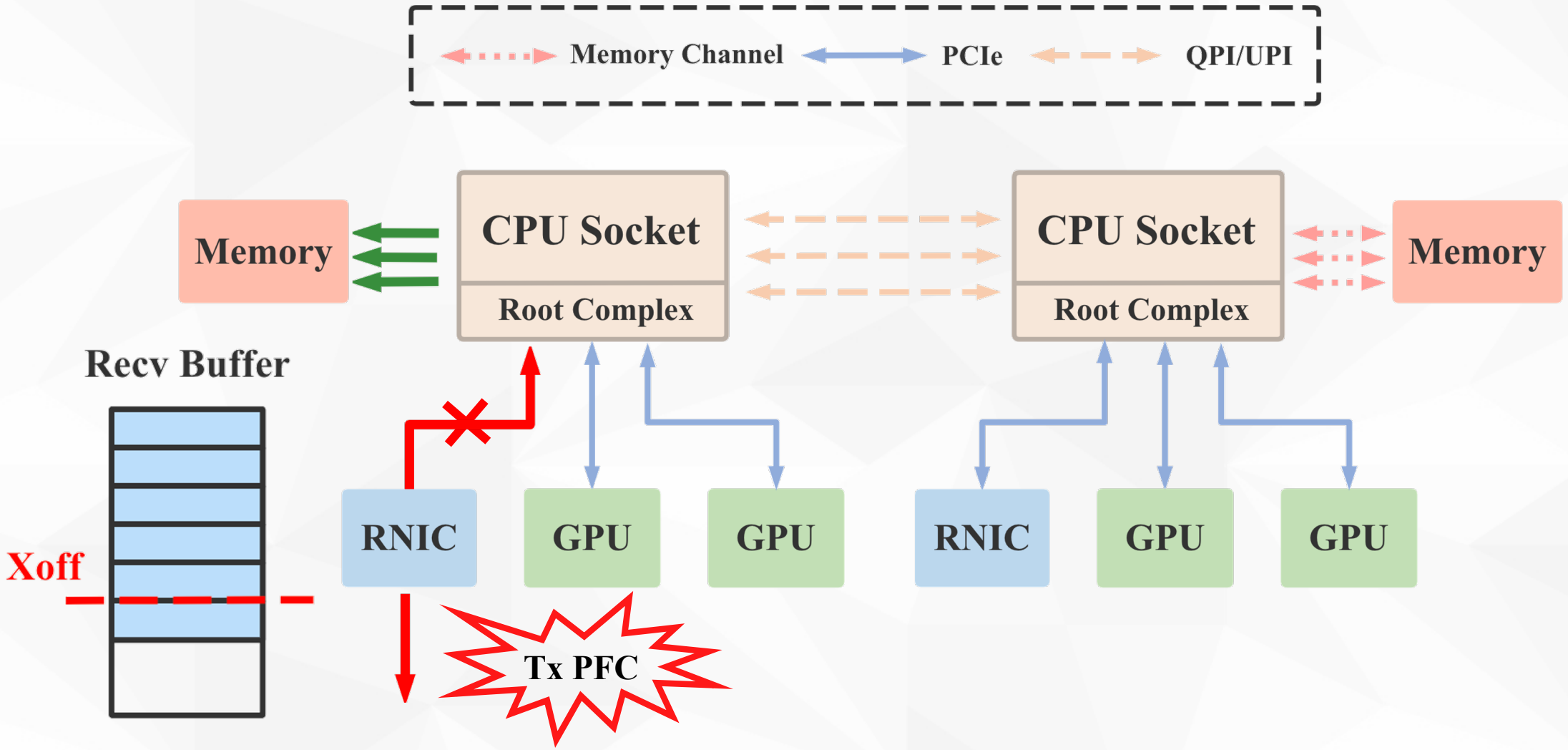
Background & Motivation

■ Intra-host Network Bottlenecks



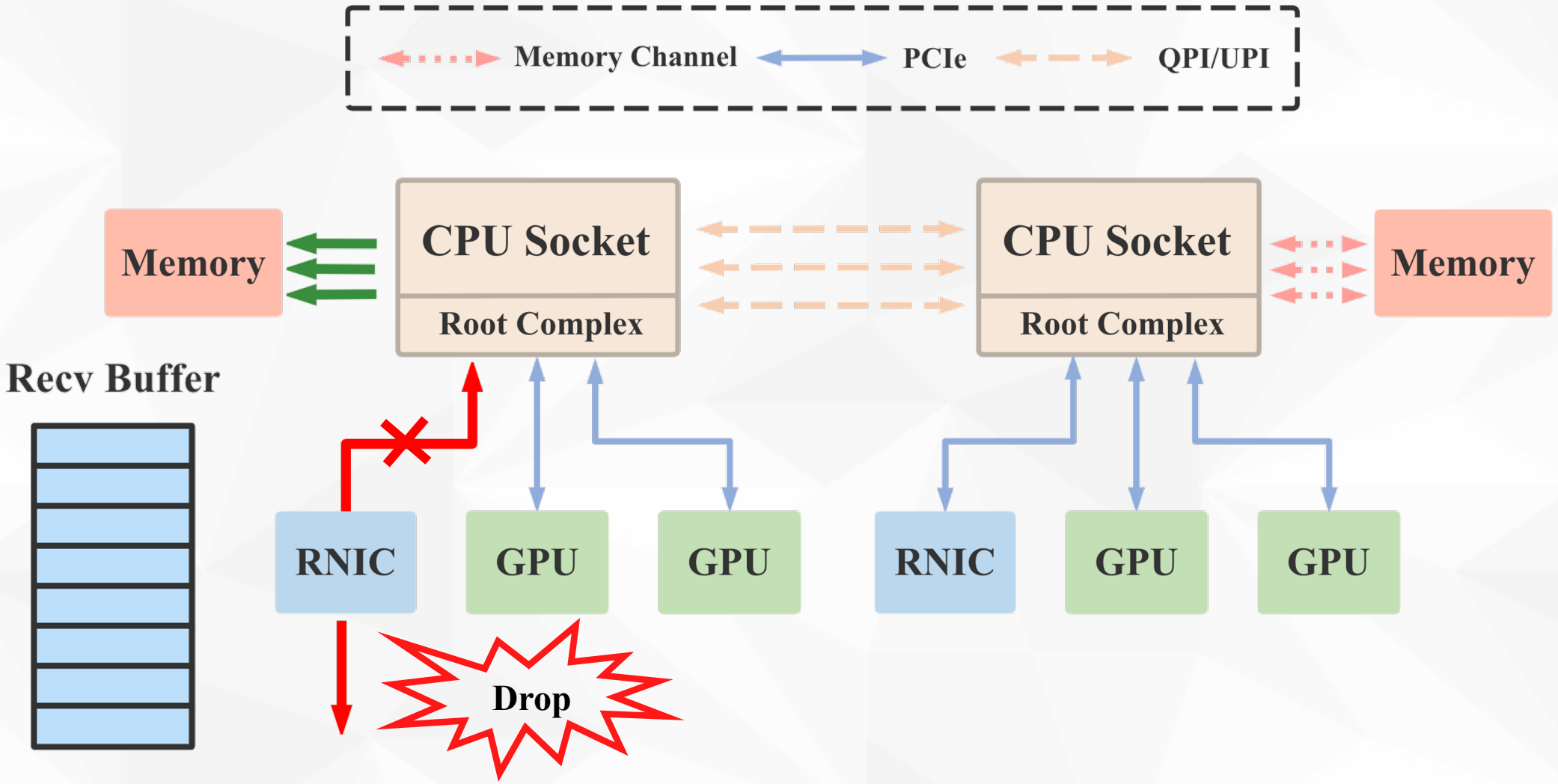
Background & Motivation

■ Intra-host Network Bottlenecks



Background & Motivation

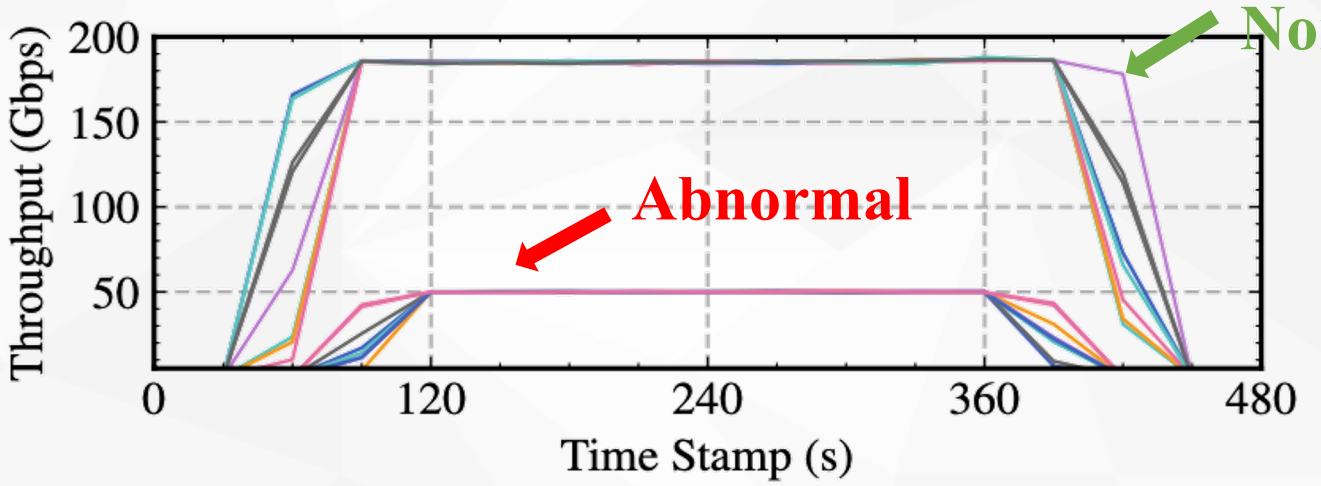
■ Intra-host Network Bottlenecks



Background & Motivation

■ The Impact of Intra-host Network Bottlenecks

- Bandwidth degradation, worse in a lossy environment
- PFC storm, PFC deadlock, which may bring down the whole network
- One single intra-host bottleneck may significantly degrade the whole system



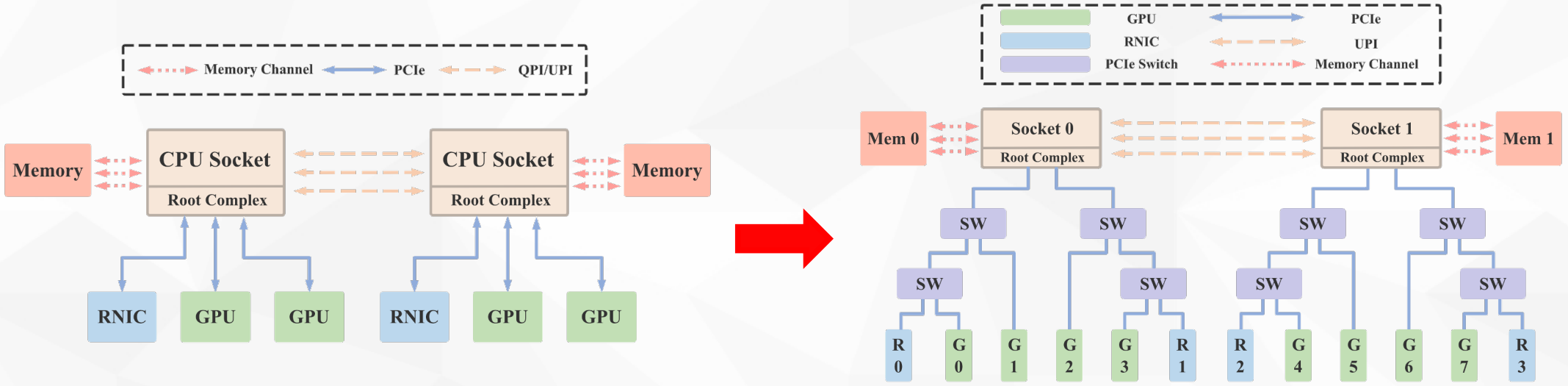
8x Servers
Each has one 200Gb/s RNIC

Intra-host bottlenecks should be discovered, diagnosed, and resolved as soon as possible

Background & Motivation

■ Intra-host Network Bottlenecks are on the RISE

- PCIe provides less bandwidth redundancy
  - 62.96Gb/s PCIe Gen3x8 vs. 25Gbps RNICs ➔ 252.06Gb/s PCIe Gen4x16 vs. 200Gbps RNICs
- Intra-host topology becomes more complex, making link failures more frequent



- Misconfigurations become more frequent. Some may lead to bottlenecks

Background & Motivation

■ Our Targets

- Effectively discover intra-host bottlenecks whenever they appear
- Quickly diagnose the root causes of intra-host bottlenecks

■ Limitations of Existing Bottleneck Diagnosis Mechanisms

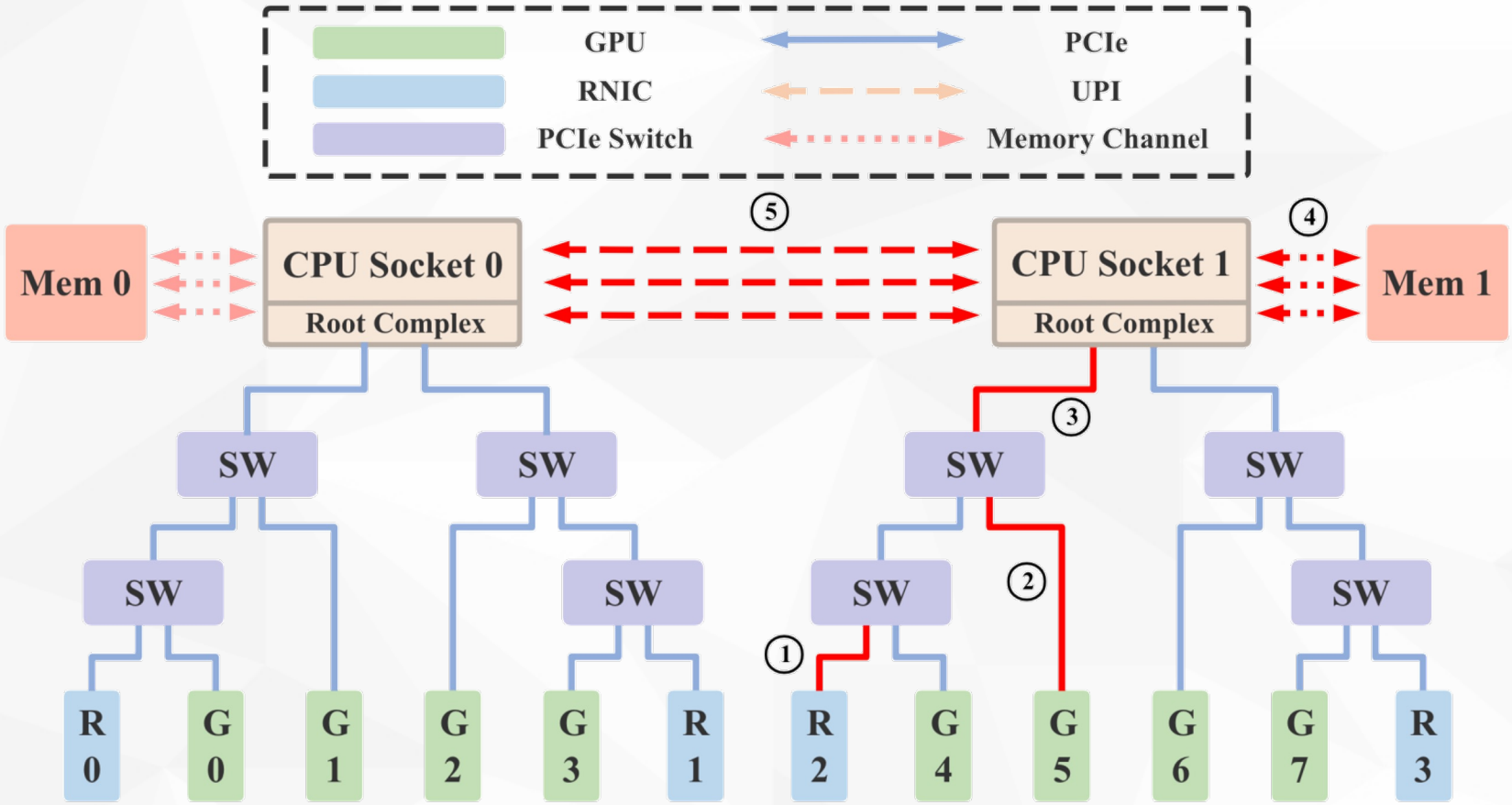
- Lack of an efficient intra-host network bottleneck monitoring system
- Could not quickly judge whether the bottleneck lies in the host or the network
- Intra-host bottleneck diagnosis is challenging due to complicated topologies

Need an effective bottleneck monitoring and diagnosis system dedicated to intra-host networks

Understanding: Symptoms of Intra-host Bottlenecks

■ Intra-host Bandwidth Degradation

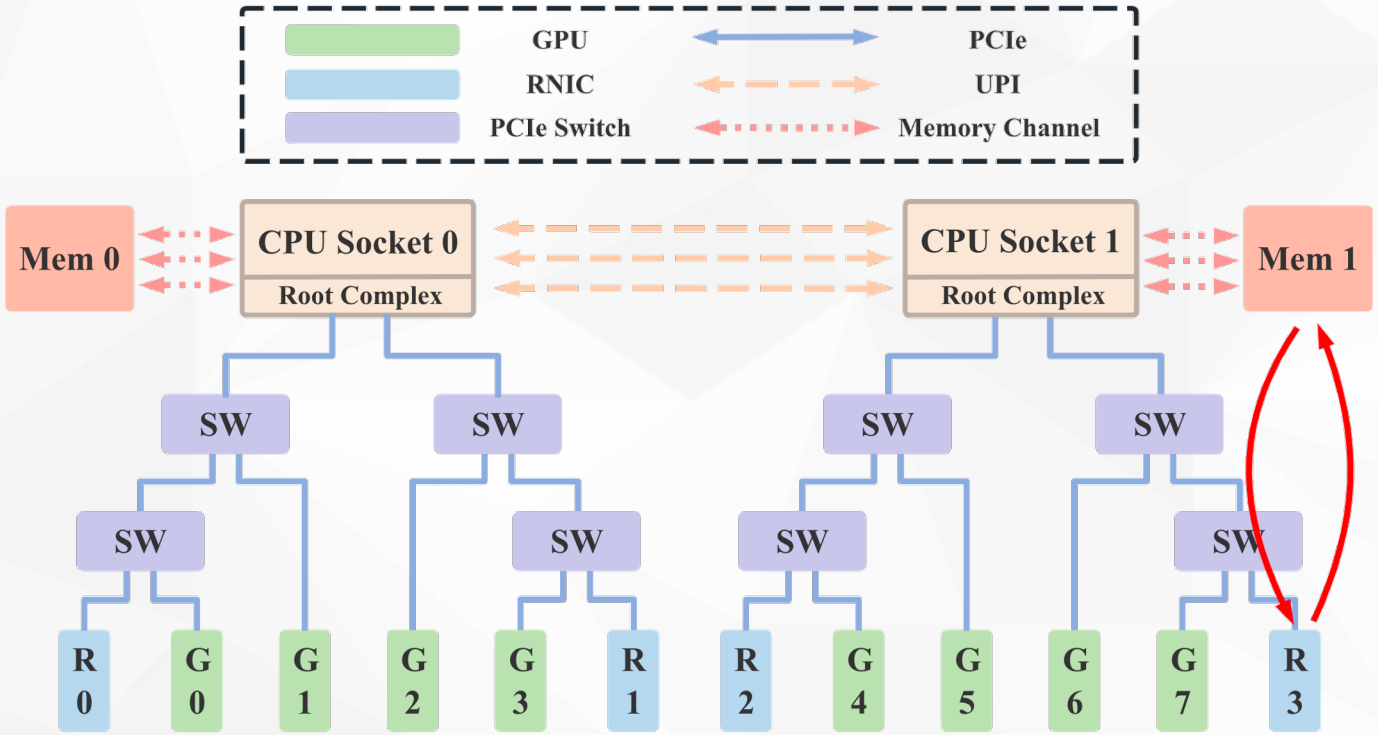
- **All intra-host links may suffer from degraded bandwidth**



Understanding: Symptoms of Intra-host Bottlenecks

■ Intra-host Bandwidth Degradation

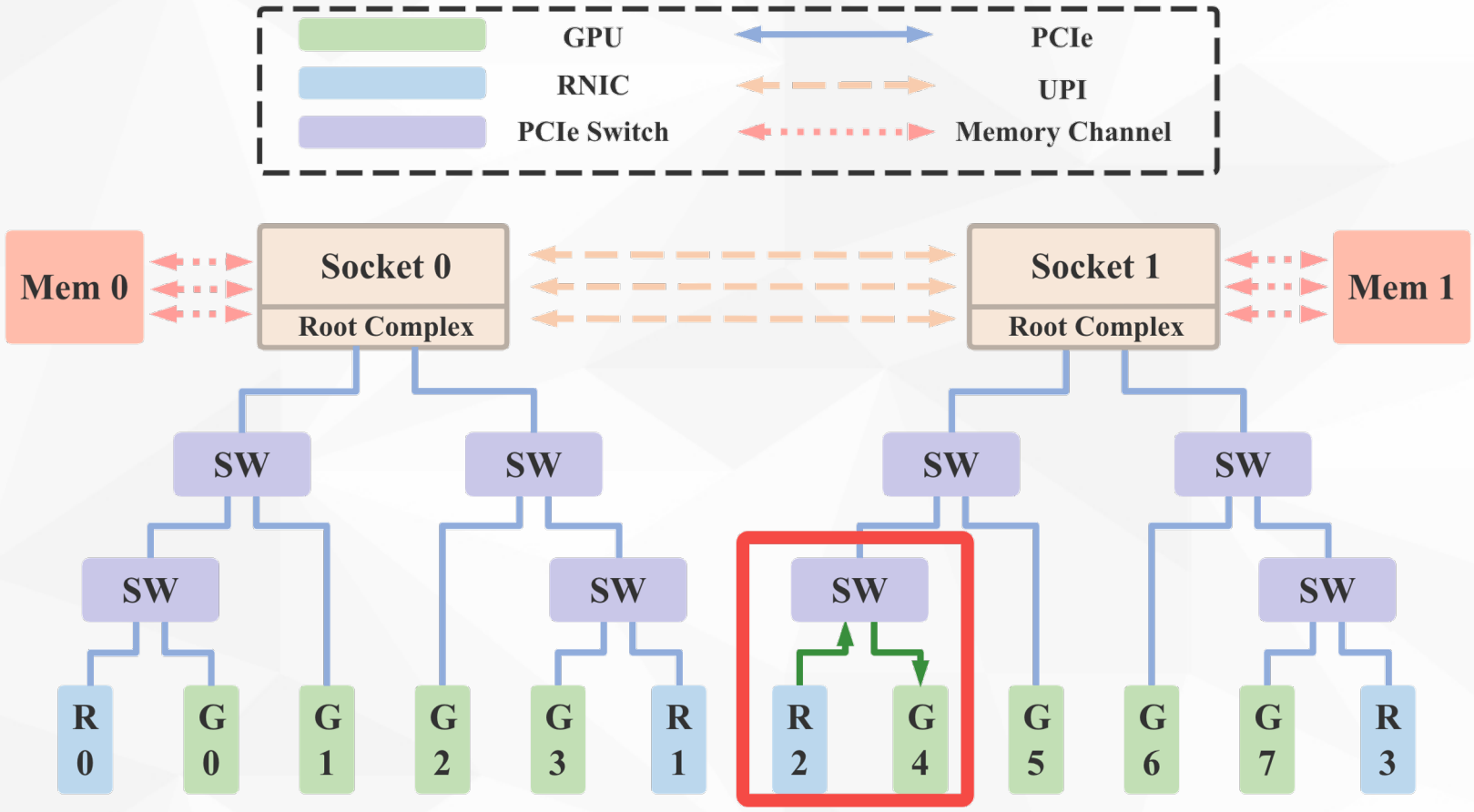
- Other traffic (e.g., loopback traffic, TCP) in the host may consume PCIe or memory bandwidth, leading to degraded intra-host bandwidth for receiving traffic



Understanding: Symptoms of Intra-host Bottlenecks

■ Intra-host Latency Increase

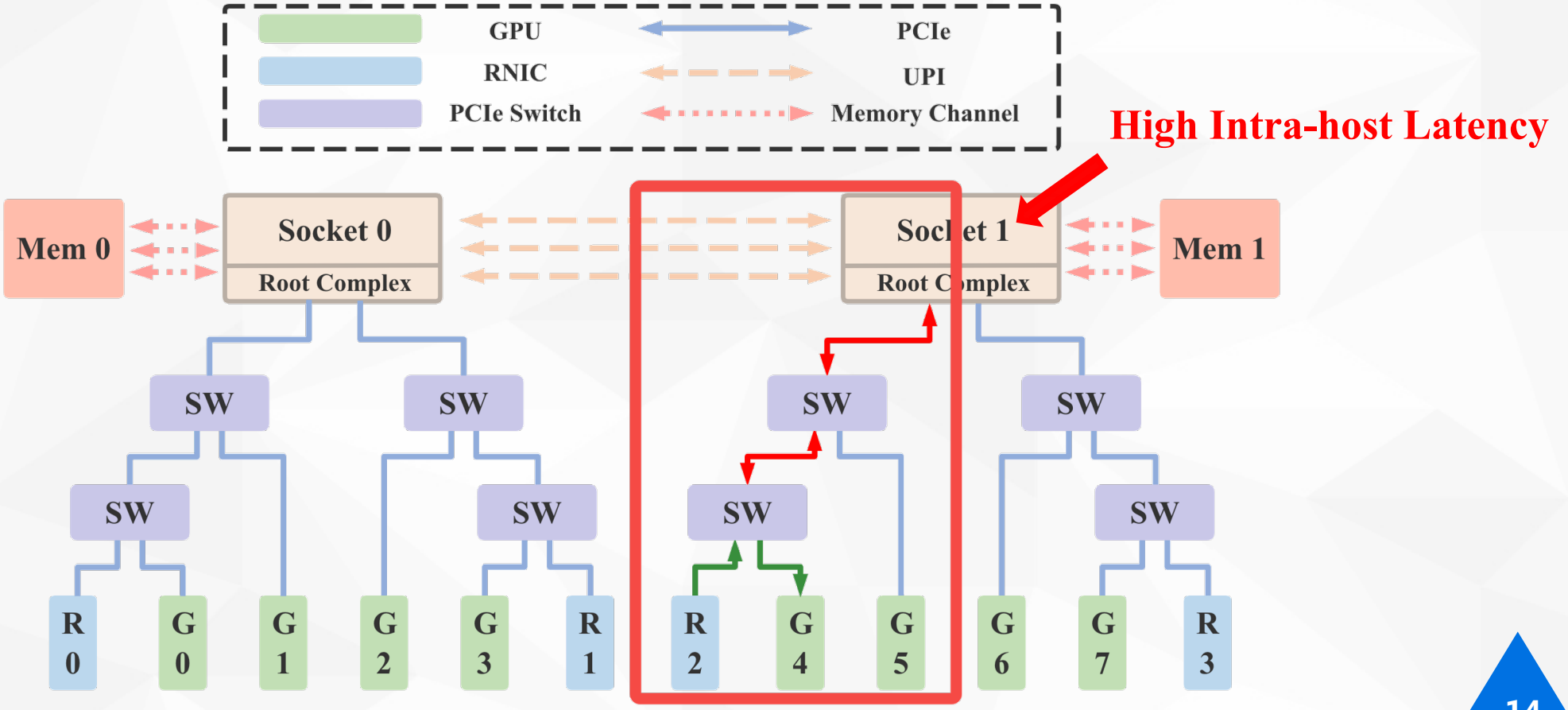
- **Normal GDR (GPU Direct RDMA) Read/Write Process**



Understanding: Symptoms of Intra-host Bottlenecks

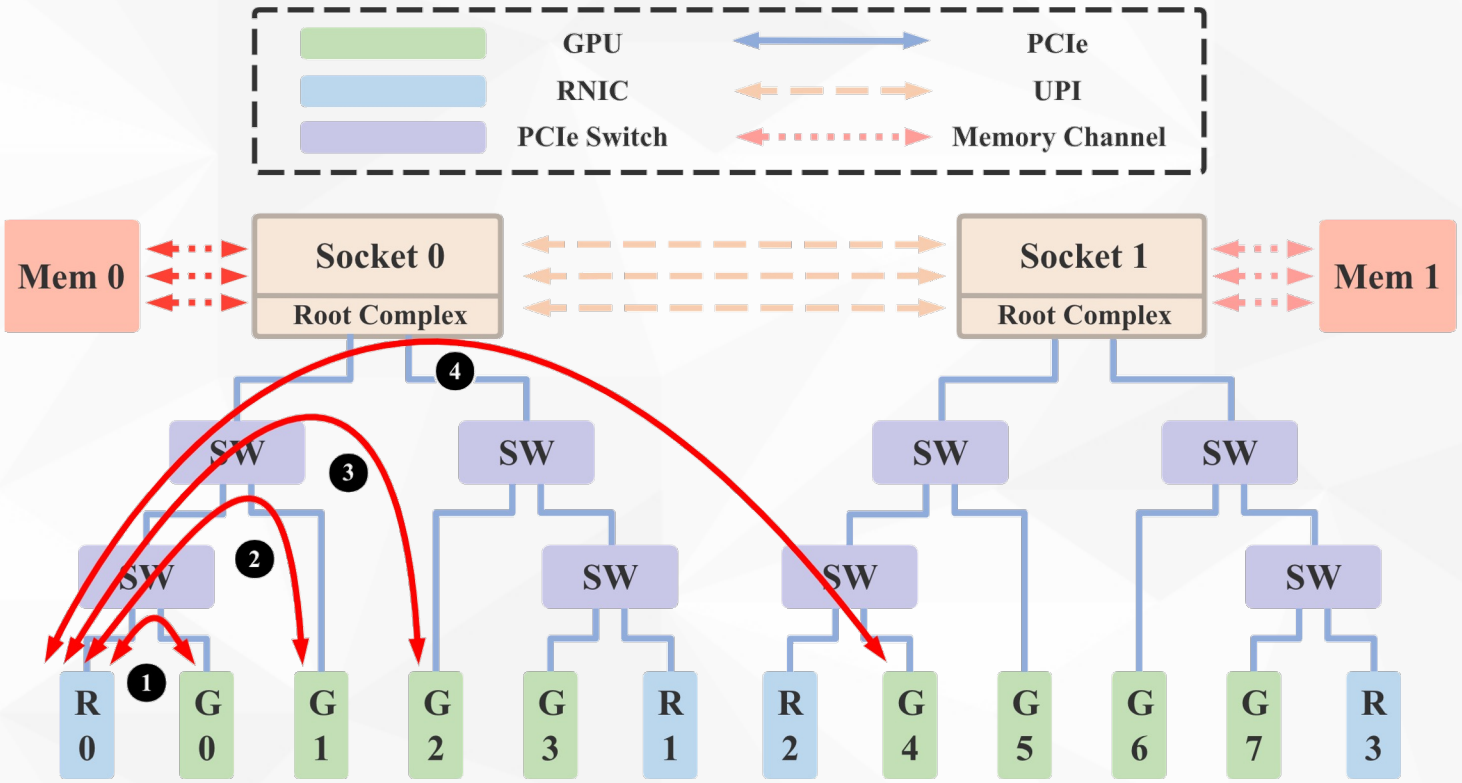
■ Intra-host Latency Increase

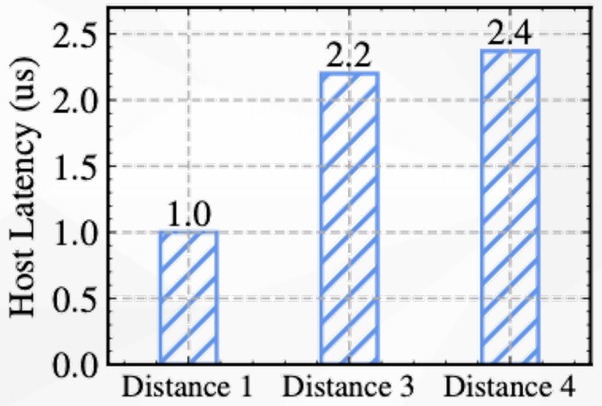
- Abnormal GDR Read/Write under misconfigurations

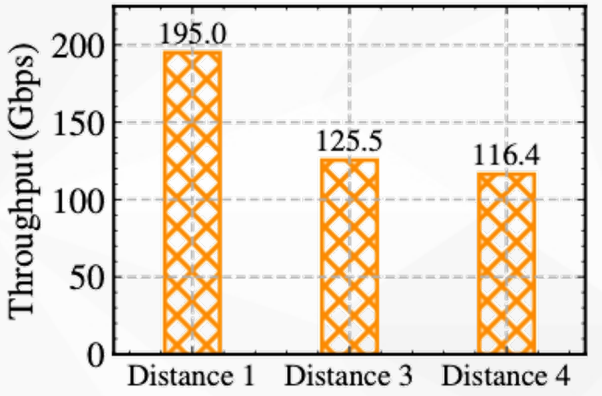


Understanding: Symptoms of Intra-host Bottlenecks

■ High Intra-host Latency Hurts Intra-host Bandwidth



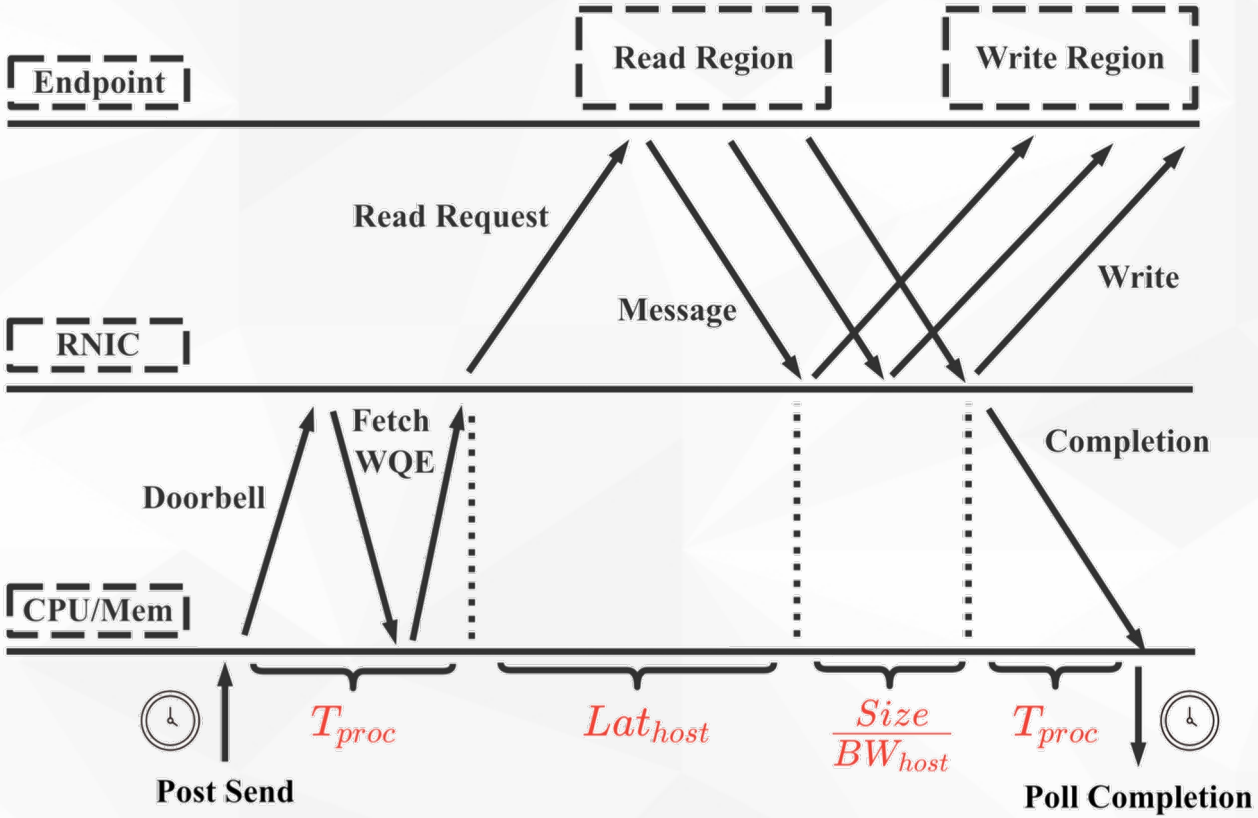




Intra-host latency and **bandwidth** can effectively reflect intra-host bottlenecks

Hostping Design: Core Idea

■ Measure Intra-host Latency and Bandwidth with Loopback Tests



Measured Loopback Latency:

$$Lat = T_{proc} + Lat_{host} + \frac{Size}{BW_{host}}$$

Use small messages to measure latency:

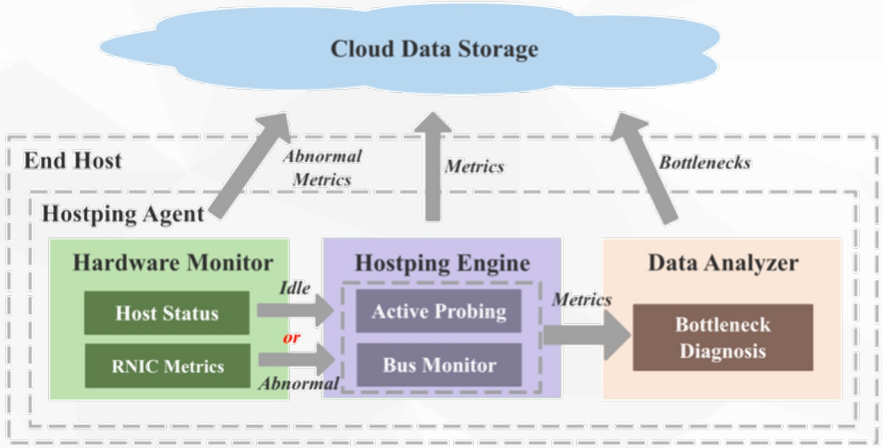
$$\lim_{size \to 0} Lat = T_{proc} + Lat_{host}$$

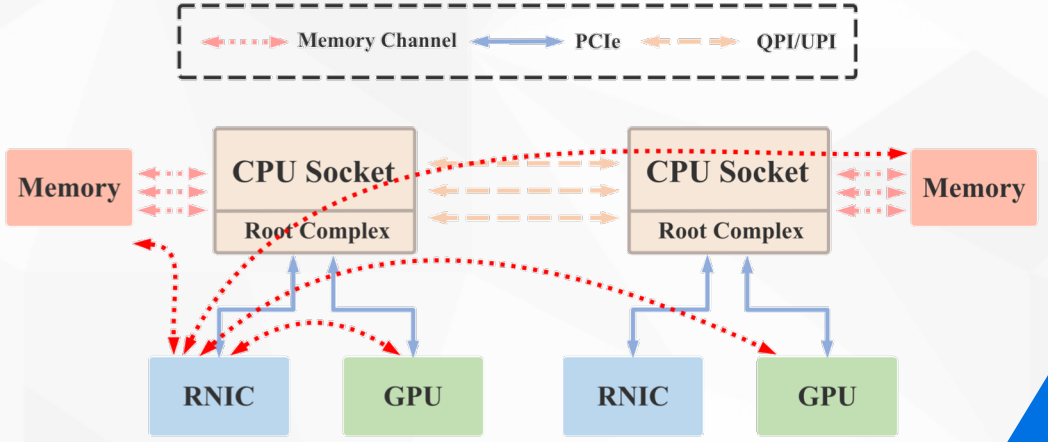
Use large messages to measure bandwidth:

$$\lim_{size \to \infty} Lat = \frac{Size}{BW_{host}}$$

Hostping Design: Framework

- An always-on service on RDMA servers
- When the host is idle (has no running task), Hostping periodically measures intra-host latency and bandwidth with loopback tests, and uses these metrics to (1) monitor intra-host path status and (2) diagnose the root causes of bottlenecks
- When the host is busy with services, Hostping runs loopback tests to diagnose intra-host bottlenecks when RNICs show abnormal counters (e.g., Tx pause frames, packet drops)

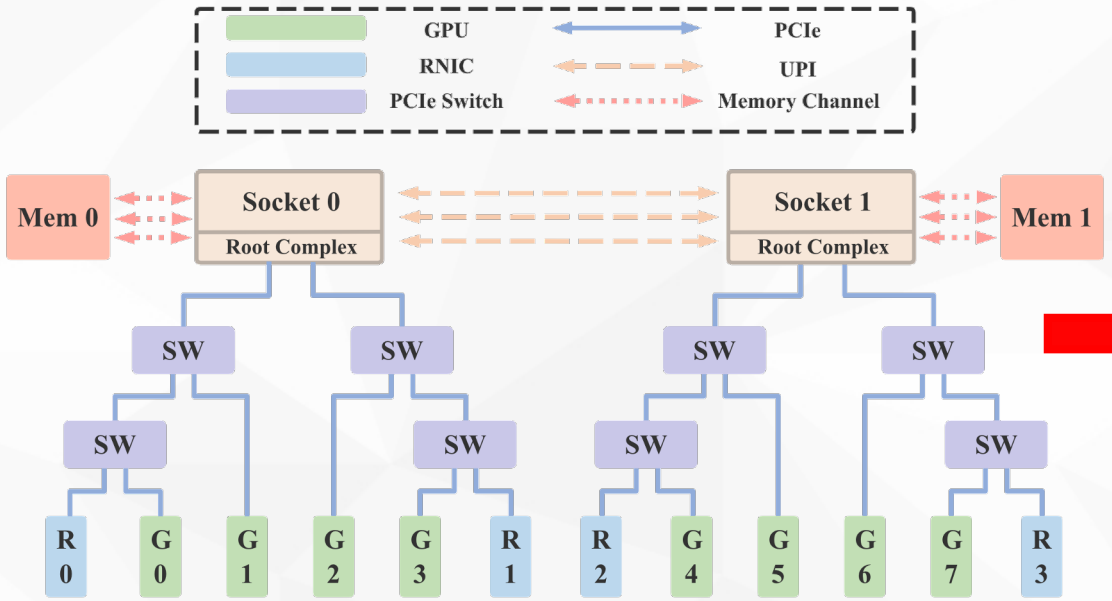




Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Get Path Status: Get Bandwidth and Latency Matrices

- Periodically conduct full-mesh loopback tests between all RNICs and all endpoints (memory nodes and GPUs) in the host to get bandwidth and latency matrices



Bandwidth matrices (Gbps)

```
**************** loopback ****************
        mem0    mem1    GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7
mlx5_0  197.928 195.544 194.636 194.499 90.113  90.024  72.109  71.847  72.674  72.746
mlx5_1  196.953 193.309 82.282  81.827  191.463 190.159 72.018  71.912  71.697  71.731
mlx5_2  197.702 198.136 71.526  71.397  72.557  72.333  192.03  191.36  94.824  94.67
mlx5_3  197.132 198.141 72.81   72.406  72.238  72.349  95.367  95.185  191.949 192.3
```

Latency matrices (us)

```
**************** loopback ****************
        mem0    mem1    GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7
mlx5_0  2.544   2.52    2.807   2.826   3.486   3.466   3.534   3.562   3.548   3.559
mlx5_1  2.511   2.584   3.47    3.483   2.807   2.815   3.563   3.541   3.544   3.554
mlx5_2  2.719   2.719   3.7     3.717   3.711   3.699   2.964   2.969   3.643   3.64
mlx5_3  2.567   2.514   3.545   3.555   3.548   3.553   3.468   3.489   2.824   2.817
```

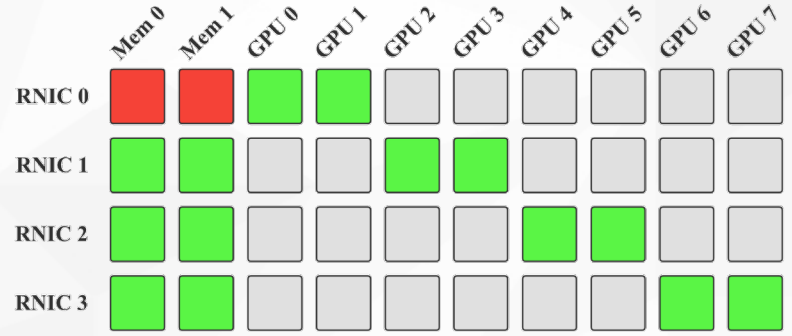
Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Get Path Status: Compare Measured Bandwidth with Baseline

- For endpoints affinitive to RNICs, judge if the paths are normal or abnormal

********** loopback **********

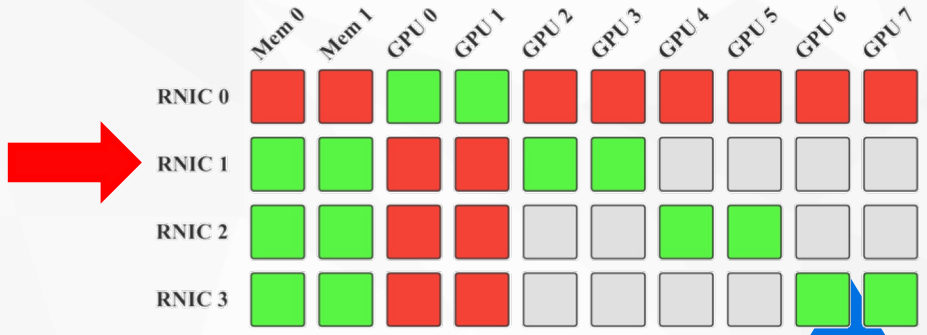
	mem0	mem1	GPU0	GPU1	GPU2	GPU3	GPU4	GPU5	GPU6	GPU7
mlx5_0	12.928	12.544	194.636	194.499	12.113	12.024	12.109	11.847	12.674	12.746
mlx5_1	196.953	193.309	12.282	11.827	191.463	190.159	72.018	71.912	71.697	71.731
mlx5_2	197.702	198.136	11.526	11.397	72.557	72.333	192.03	191.36	94.824	94.67
mlx5_3	197.132	198.141	12.81	12.406	72.238	72.349	95.367	95.185	191.949	192.3



- For endpoints not affinitive to RNICs, only judge if the paths are abnormal

********** loopback **********

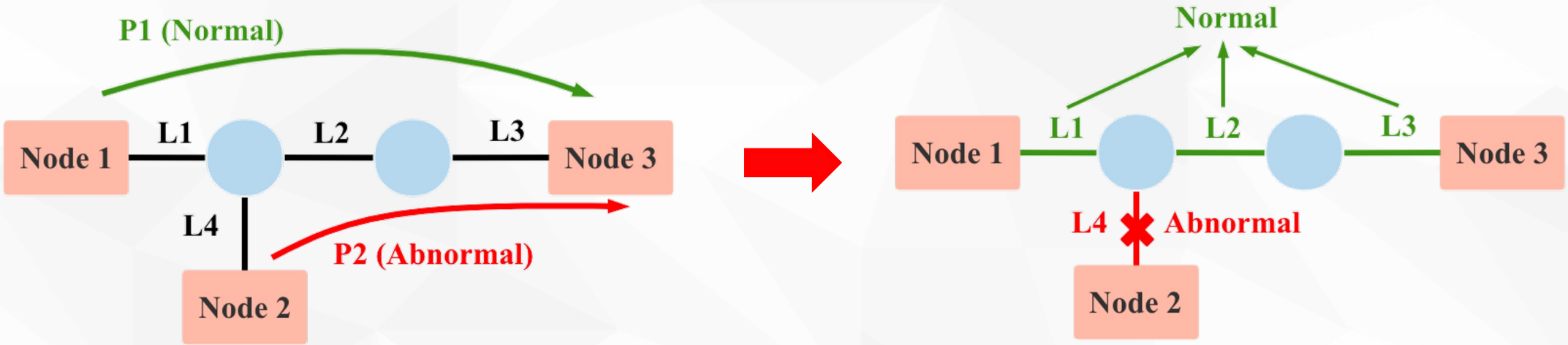
	mem0	mem1	GPU0	GPU1	GPU2	GPU3	GPU4	GPU5	GPU6	GPU7
mlx5_0	12.928	12.544	194.636	194.499	12.113	12.024	12.109	11.847	12.674	12.746
mlx5_1	196.953	193.309	12.282	11.827	191.463	190.159	72.018	71.912	71.697	71.731
mlx5_2	197.702	198.136	11.526	11.397	72.557	72.333	192.03	191.36	94.824	94.67
mlx5_3	197.132	198.141	12.81	12.406	72.238	72.349	95.367	95.185	191.949	192.3



Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Basic Idea

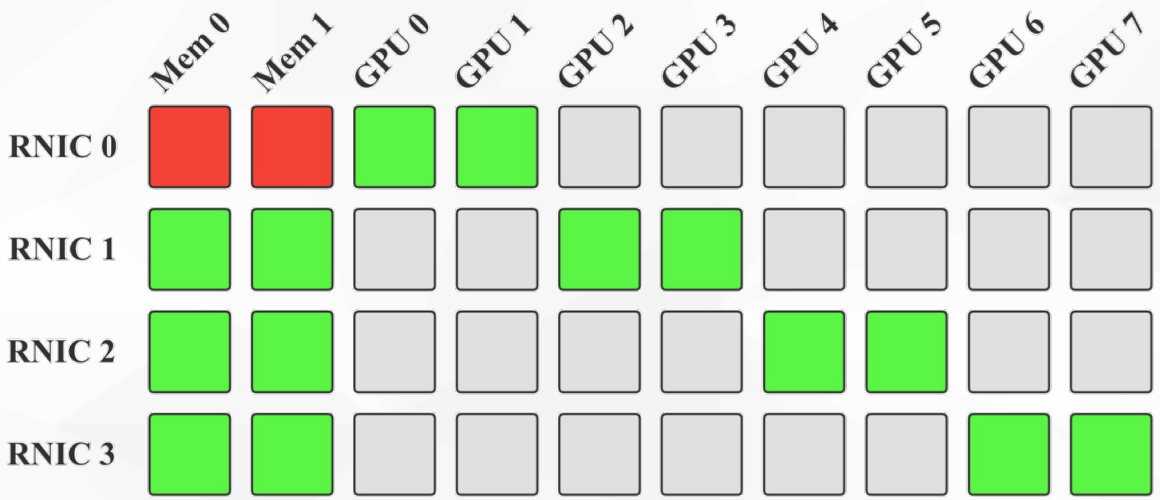
- The basic idea comes from binary network tomography: leverage binary path status (abnormal or normal) to infer the most likely abnormal links or nodes
- If a path's status is normal, all links on this path are normal
- If a path's status is abnormal, one or more links on this path are abnormal

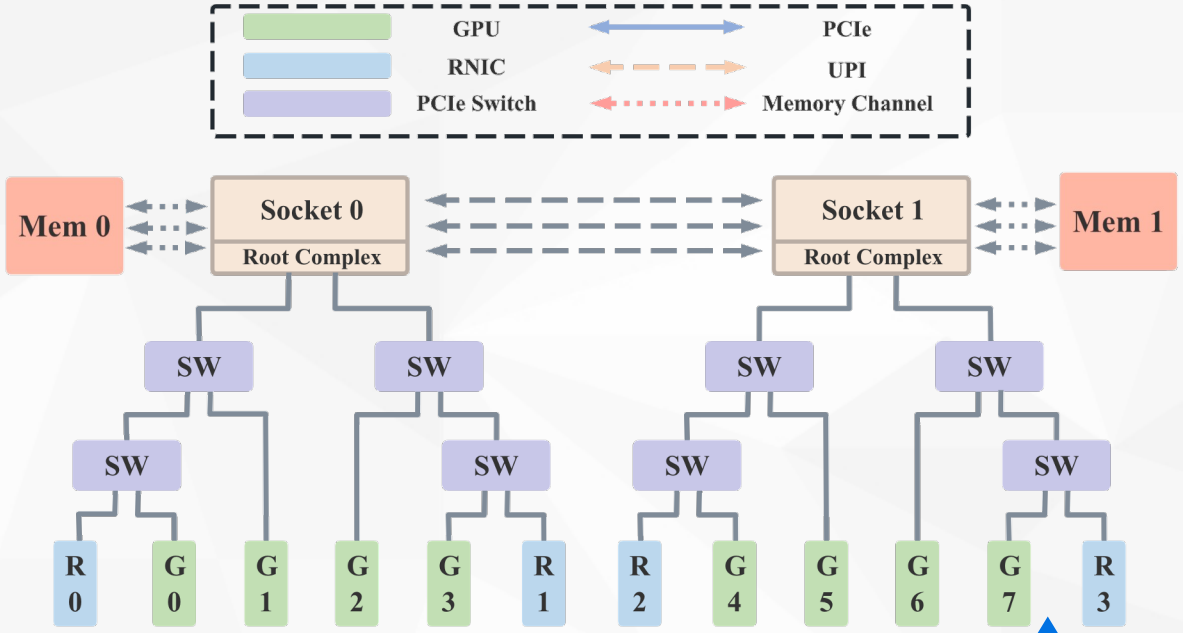


Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Infer Abnormal Links

- Infer abnormal links based on the path status matrix
- First, we mark all links' status in the host as uncertain (gray)

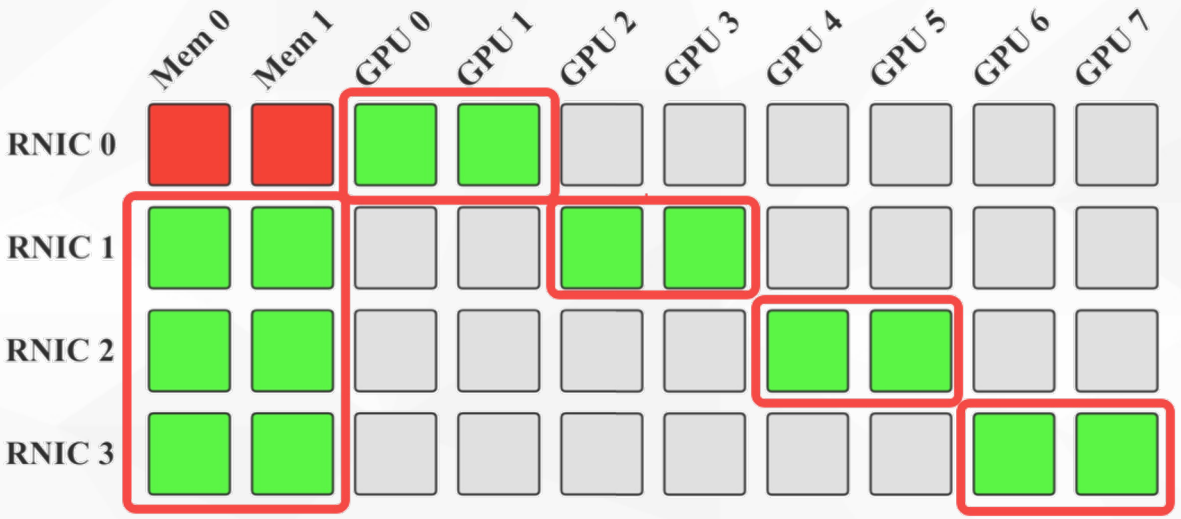


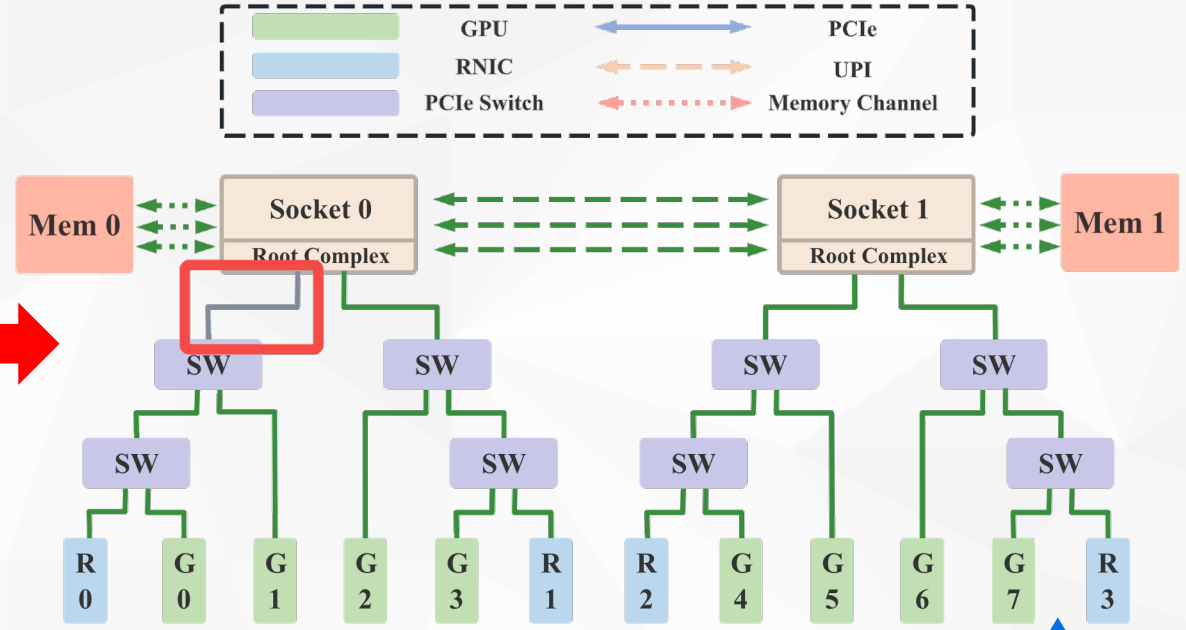


Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Infer Abnormal Links

- Green shows the path between the RNIC and the endpoint is normal
- Next, Traverse all normal paths and mark all links on them as normal

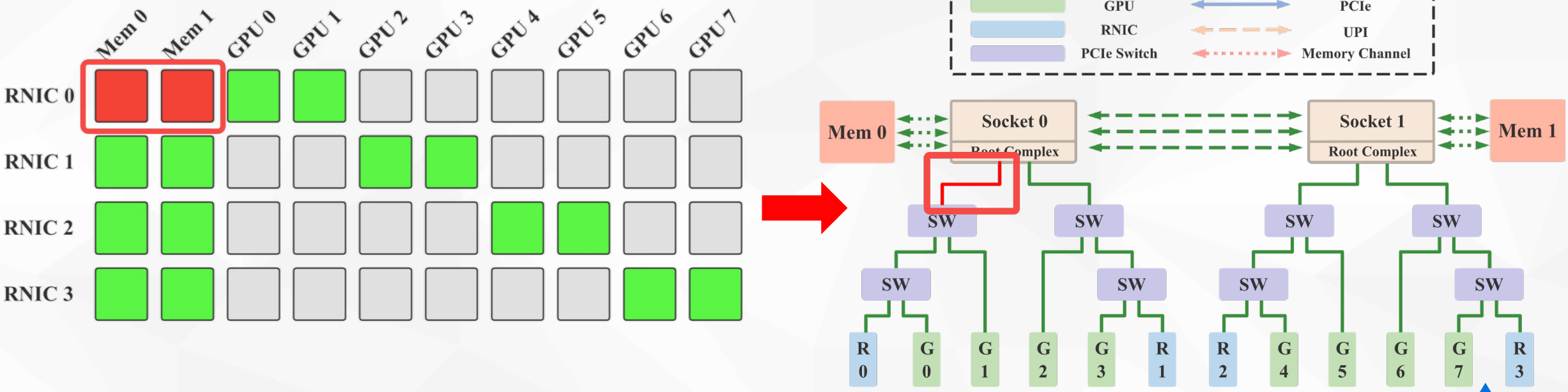




Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Infer Abnormal Links

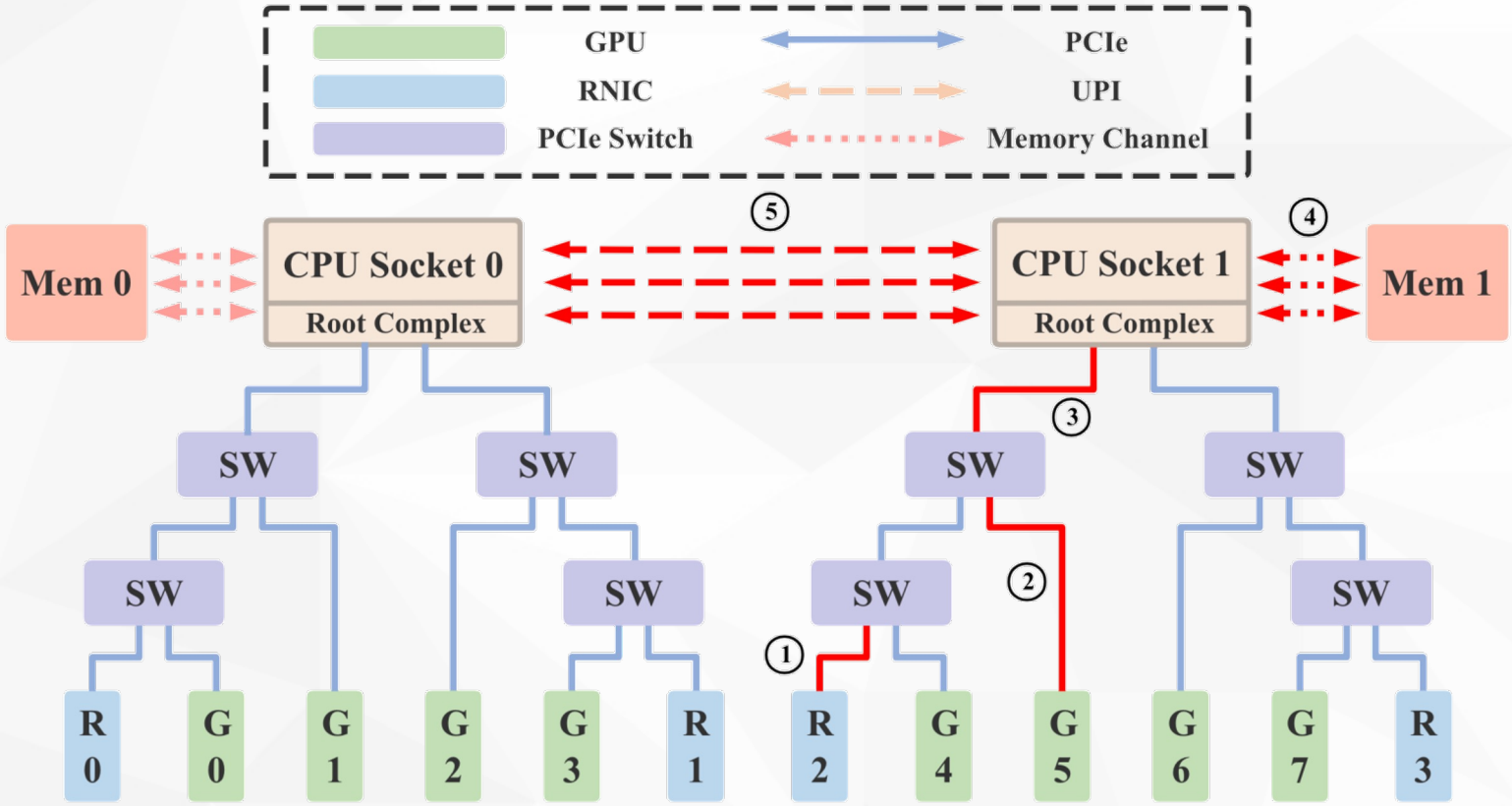
- **Red** shows the path between the RNIC and the endpoint is **abnormal**
- Traverse all **abnormal** paths, and mark all uncertain links as **abnormal**



Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Diagnose Root Causes

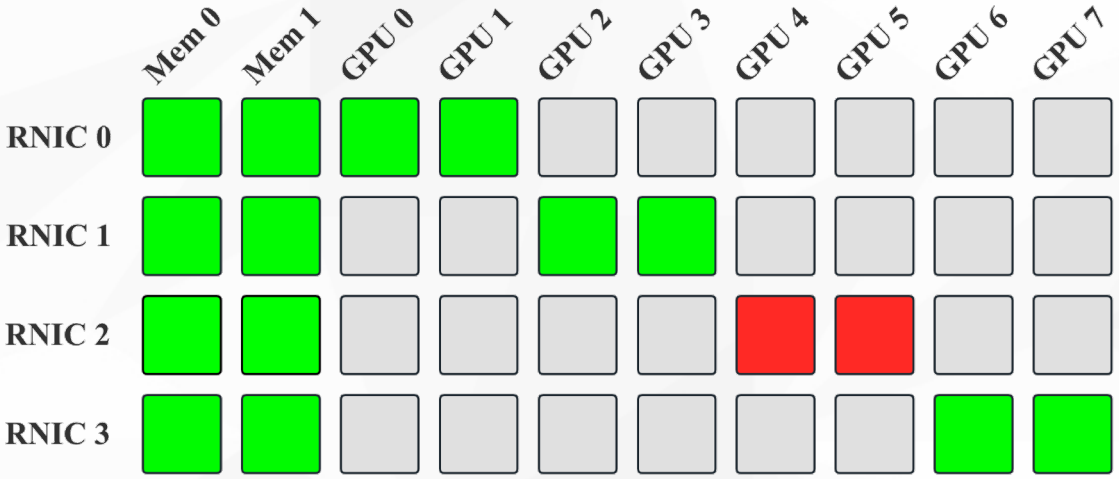
- Root causes may be **link failures** or **misconfigurations**

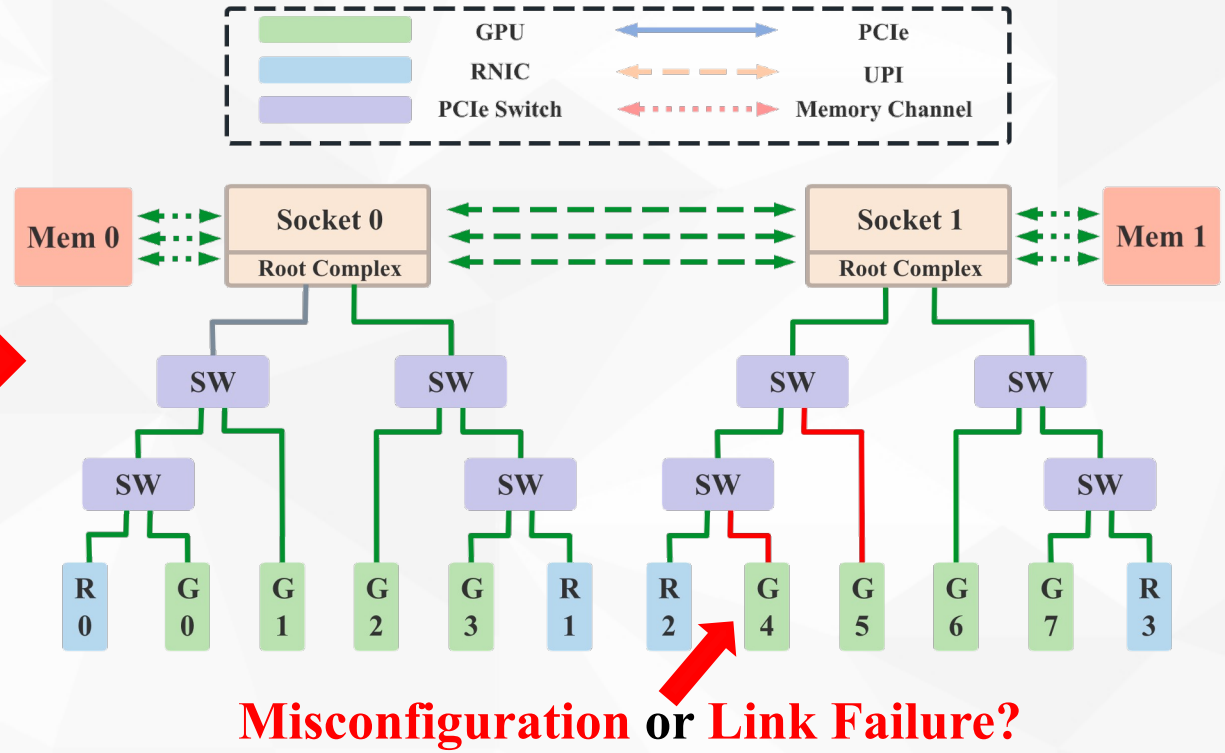


Hostping Design: Bottleneck Diagnosis (Idle Status)

■ Diagnose Root Causes

- In some cases, root causes may either be link failures or misconfigurations



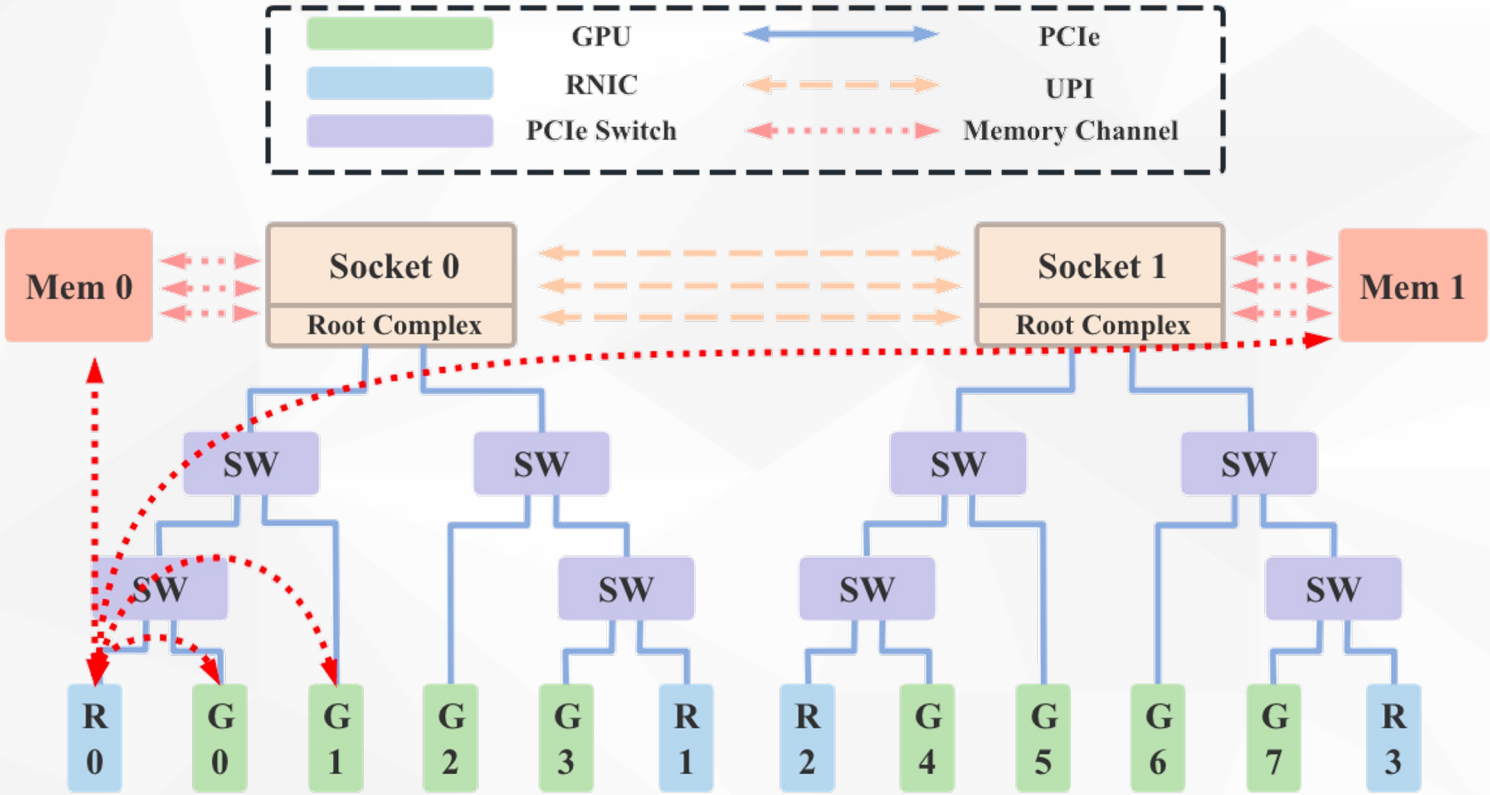


Use intra-host latency to assist in root cause diagnosis

Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Judge Path Status

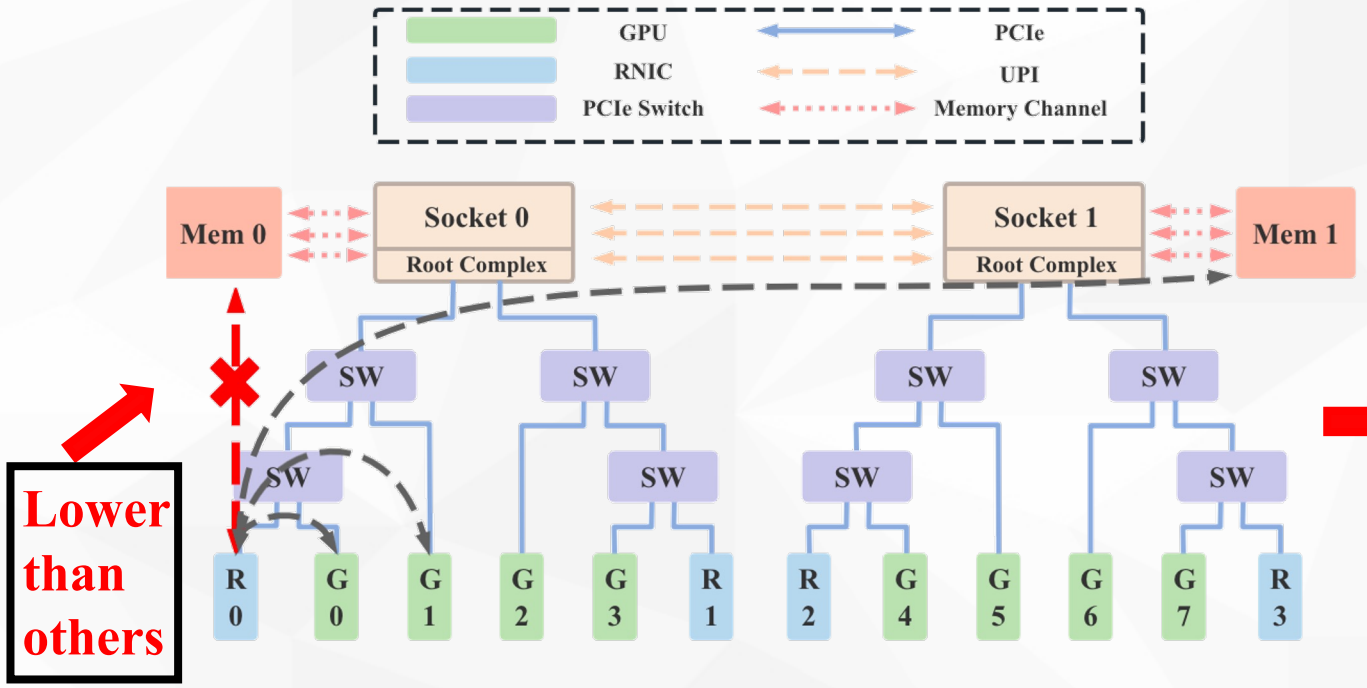
- Conduct loopback tests between all RNICs and their affinitive endpoints

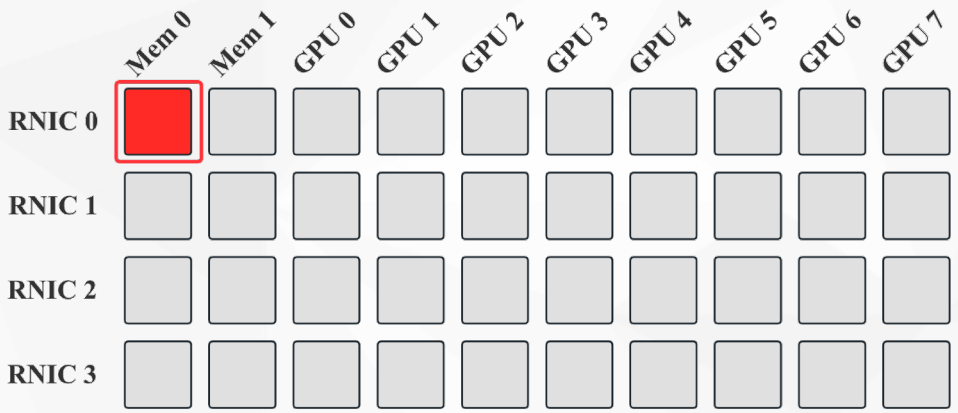


Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Judge Path Status

- For RNICs with traffic, get abnormal paths by bandwidth intercomparison

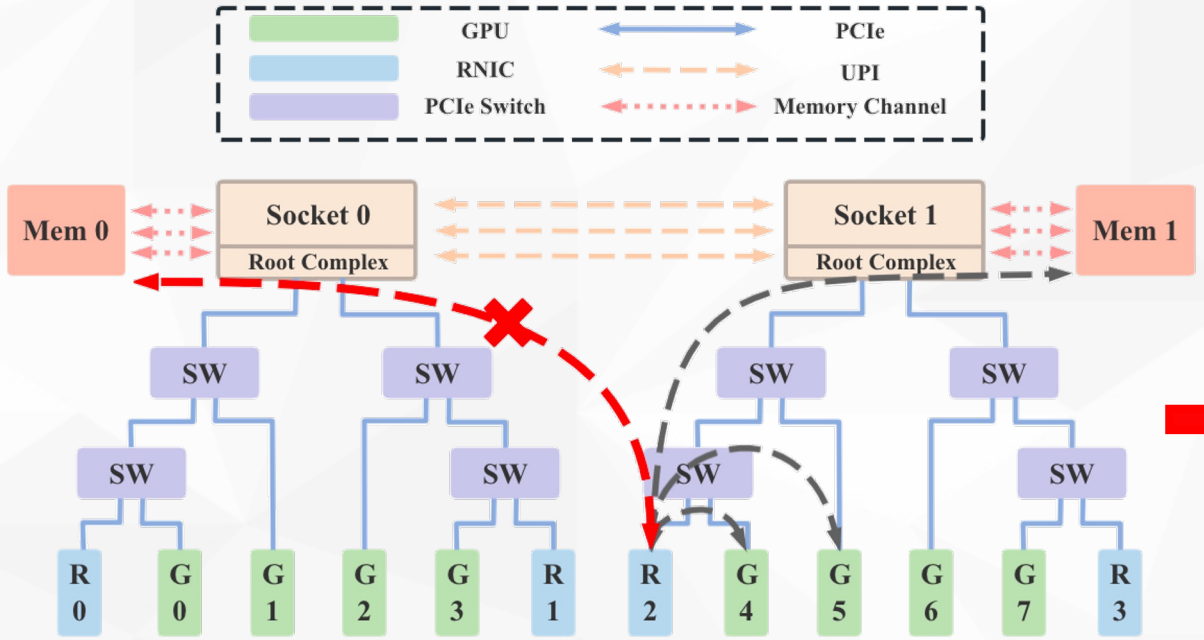


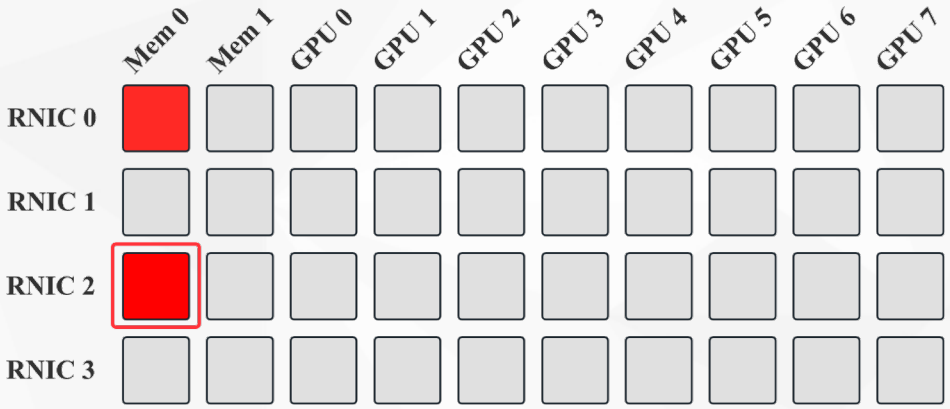


Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Judge Path Status

- For RNICs with traffic, get abnormal paths by bandwidth intercomparison

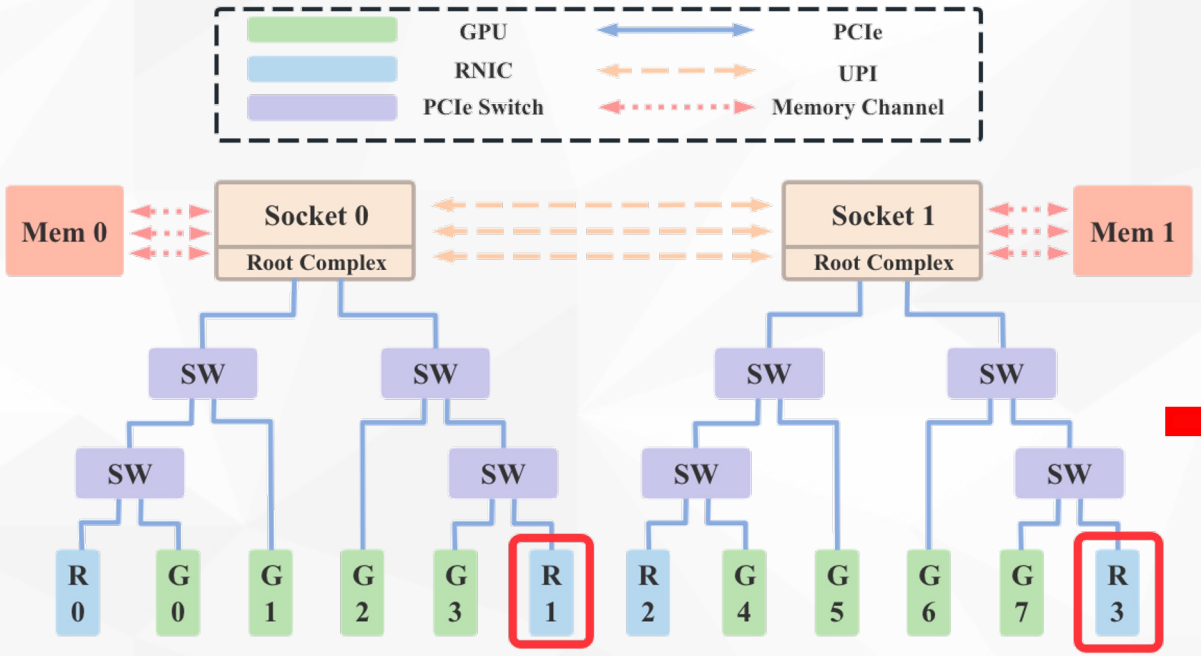


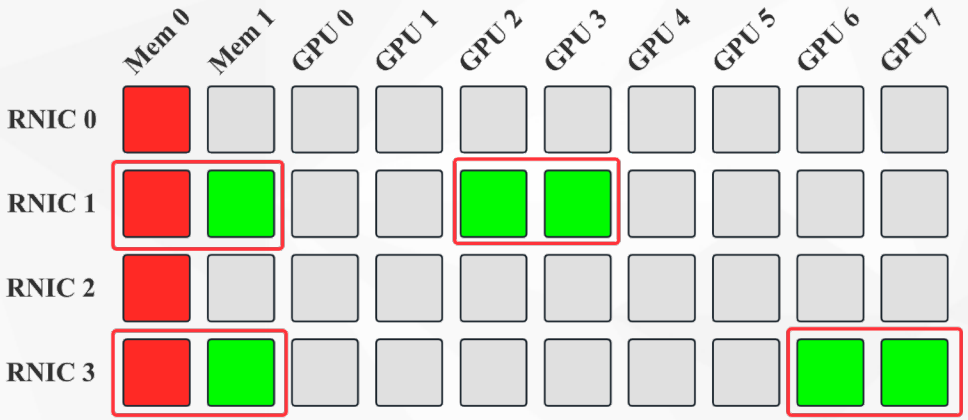


Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Judge Path Status

- For idle RNICs, compare measured bandwidth with the baseline to get path status

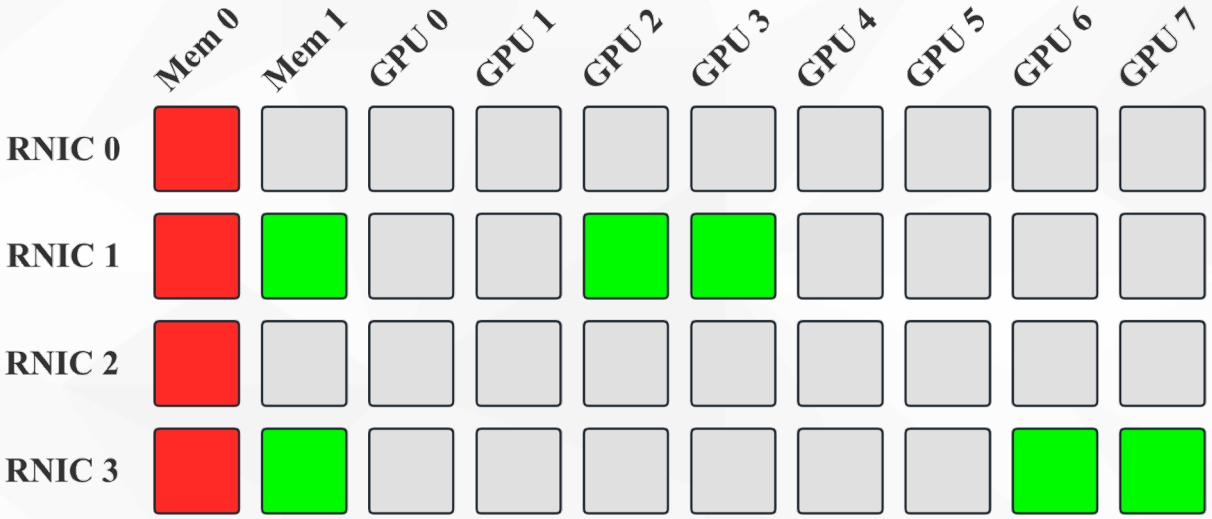


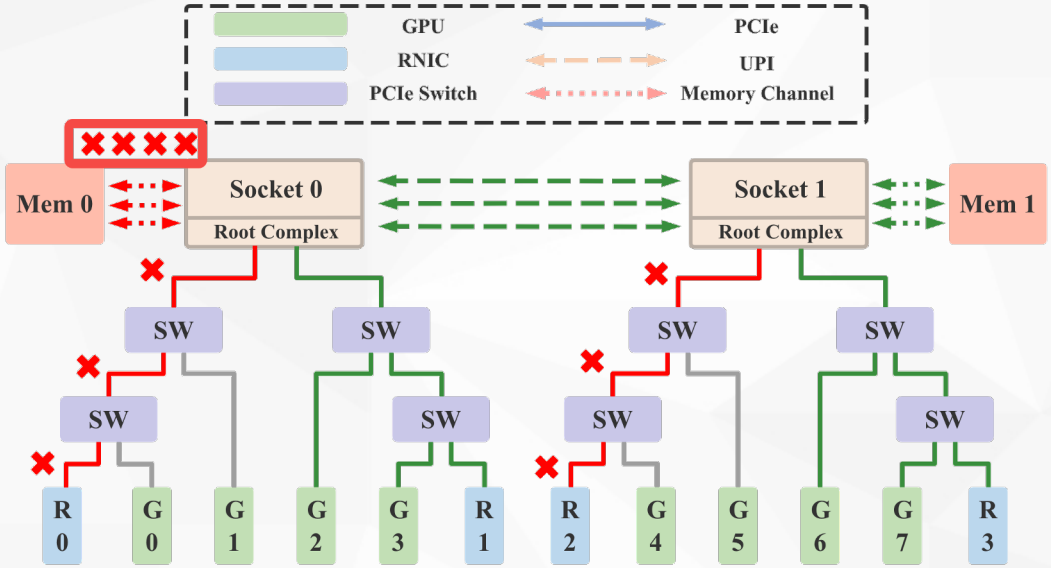


Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Infer Abnormal Links

- As RNICs with traffic cannot judge whether a path is normal, some normal links may be marked as abnormal
- Links marked abnormal by more RNICs are most likely abnormal

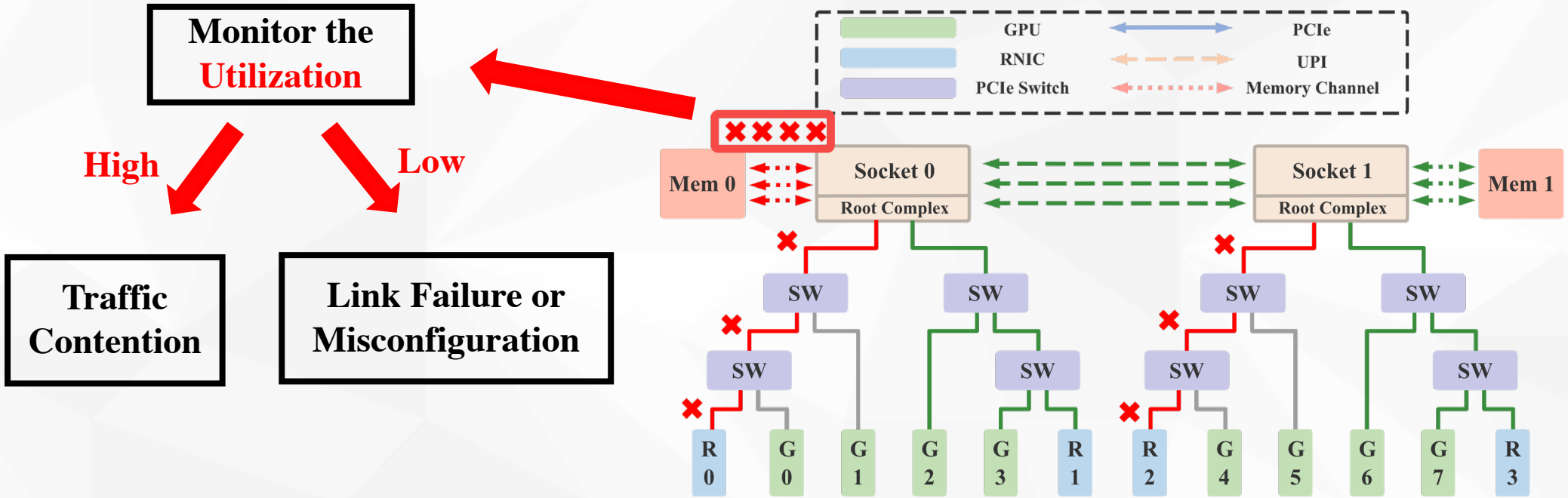




Hostping Design: Bottleneck Diagnosis (Busy Status)

■ Diagnose Root Causes

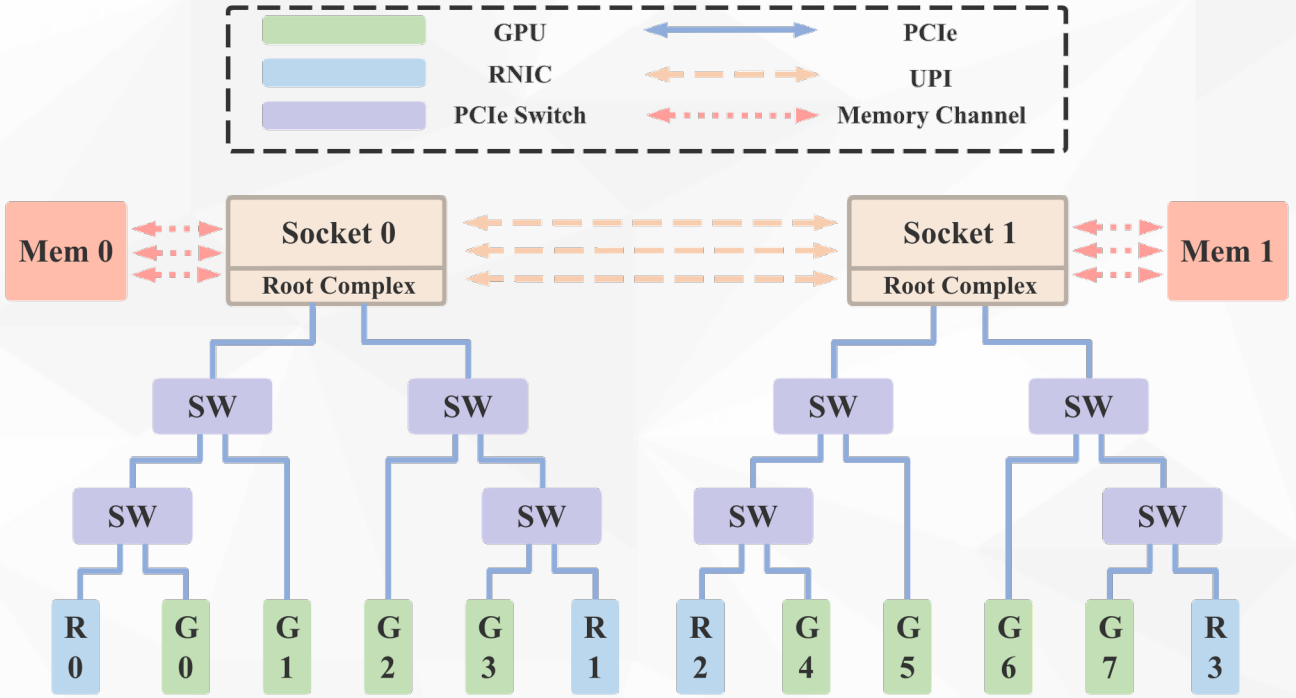
- When triggered by abnormal metrics, abnormal links are usually overloaded
- We could monitor their utilization to diagnose the root cause



Evaluation

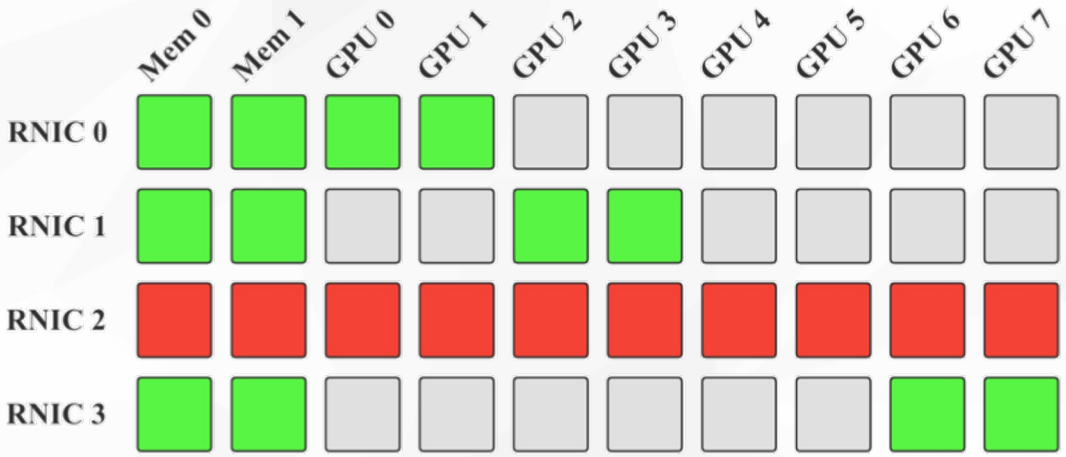
■ Deploy Hostping on over 300 Nvidia DGX A100 servers

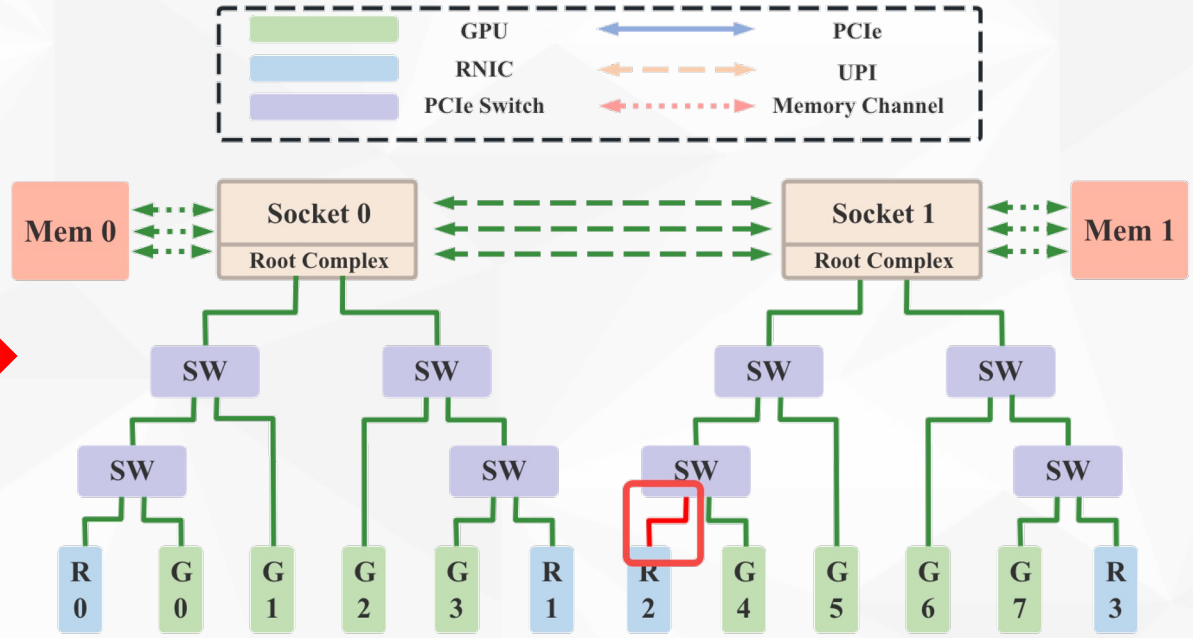
- 8X A100 GPUs + 4X 200Gb/s CX6-DX RNICs



Evaluation

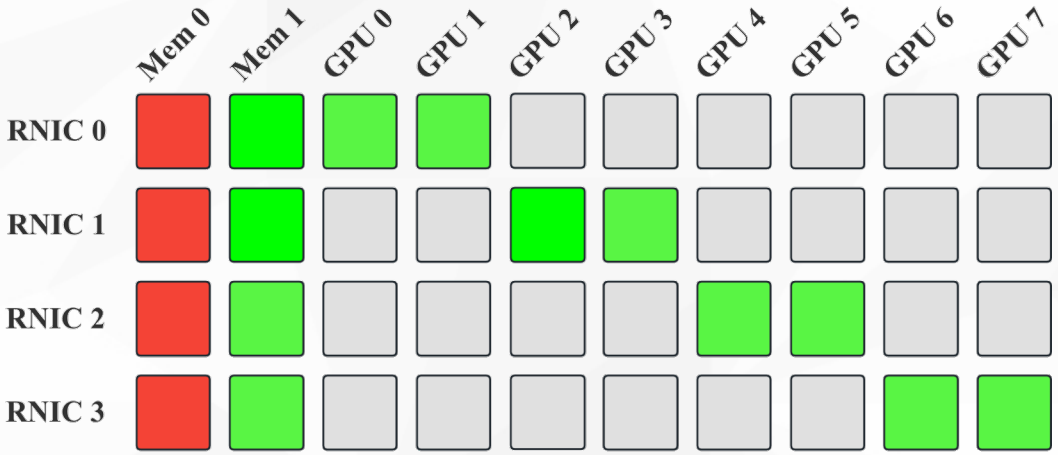
■ **Bottlenecks Found: RNIC PCIe Link Failure**

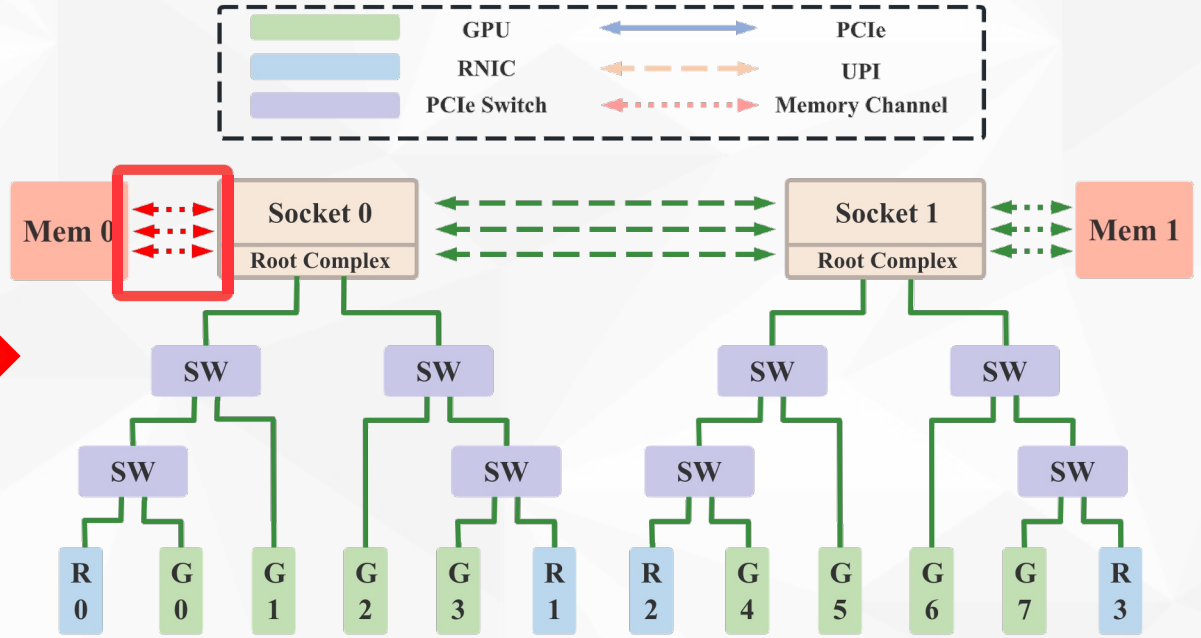




Evaluation

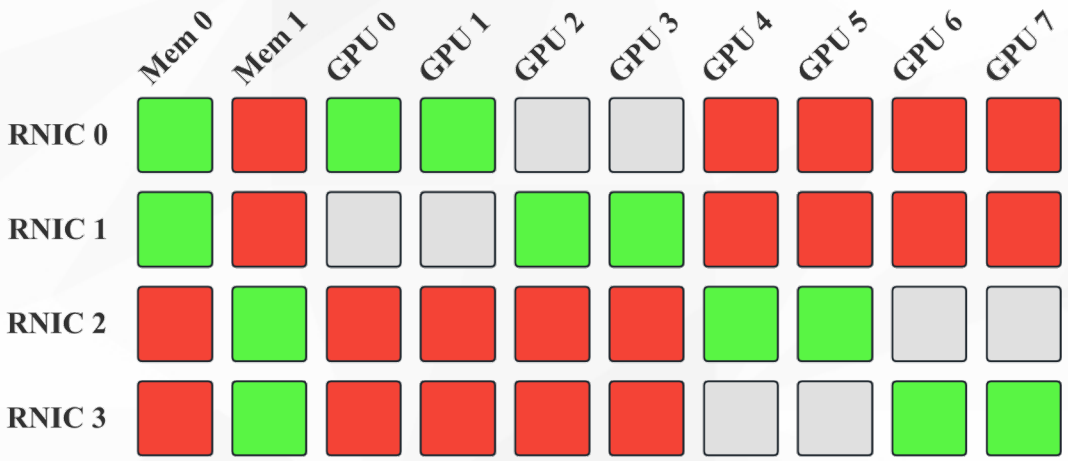
■ Bottlenecks Found: Memory Channel Failure

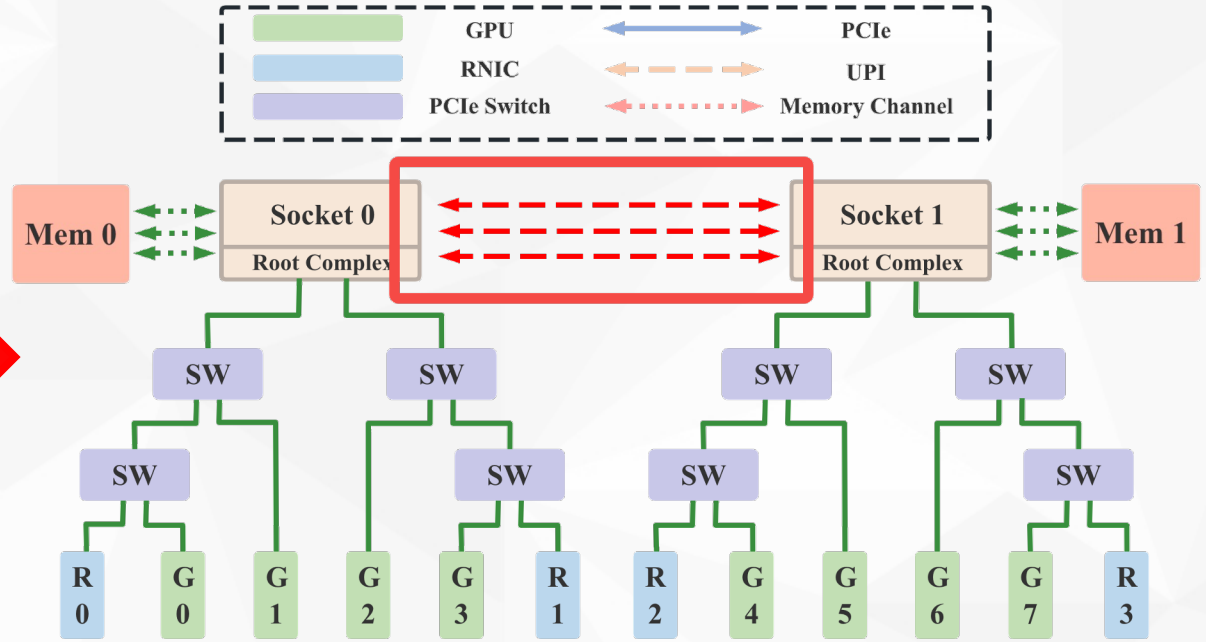




Evaluation

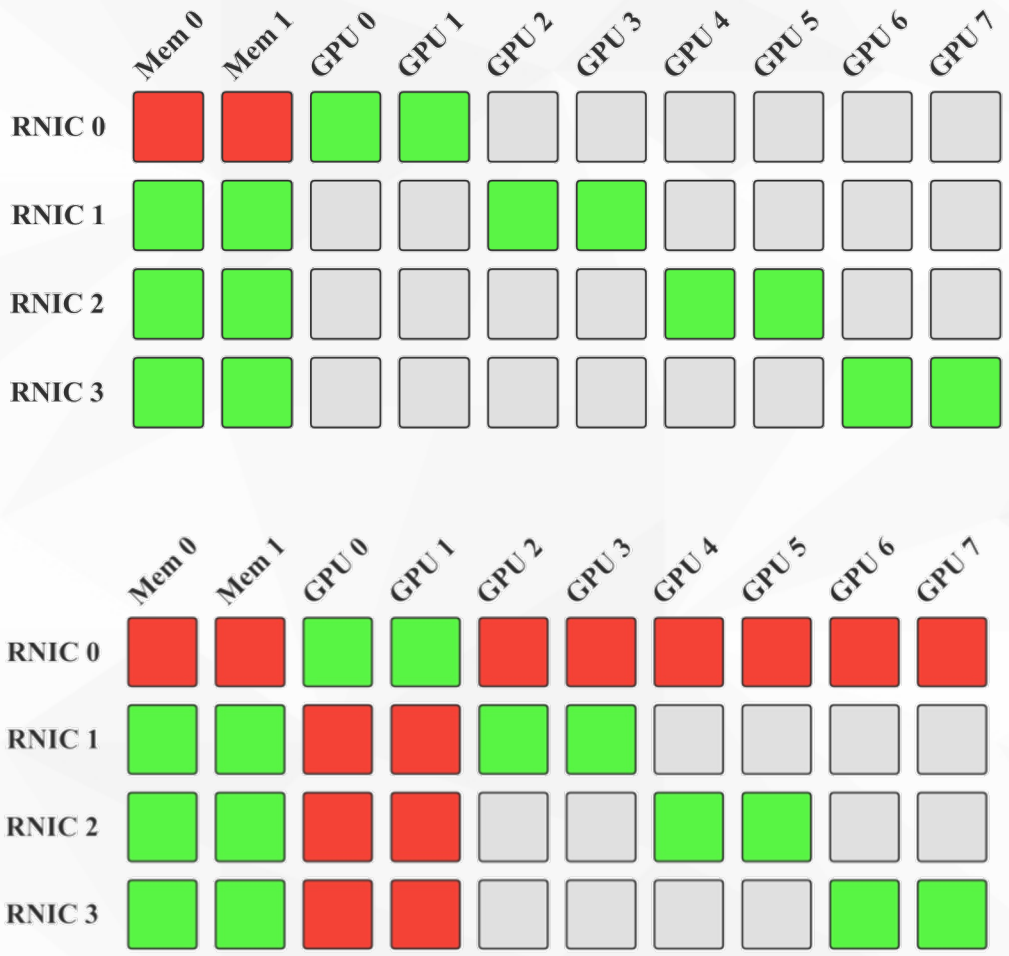
■ Bottlenecks Found: UPI Failure

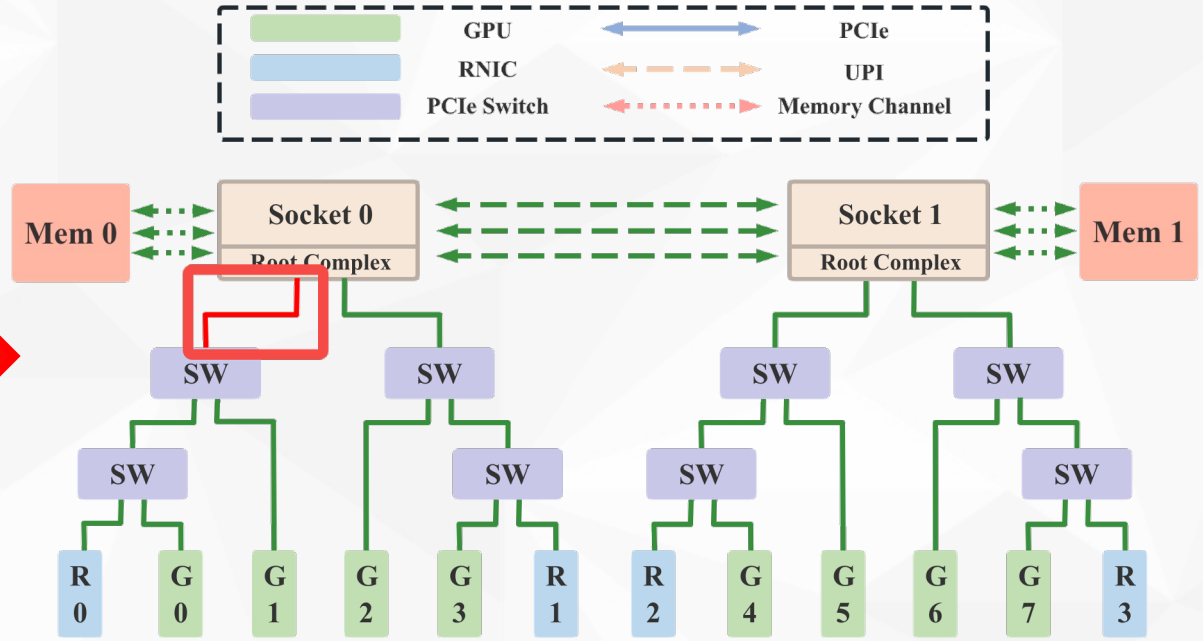




Evaluation

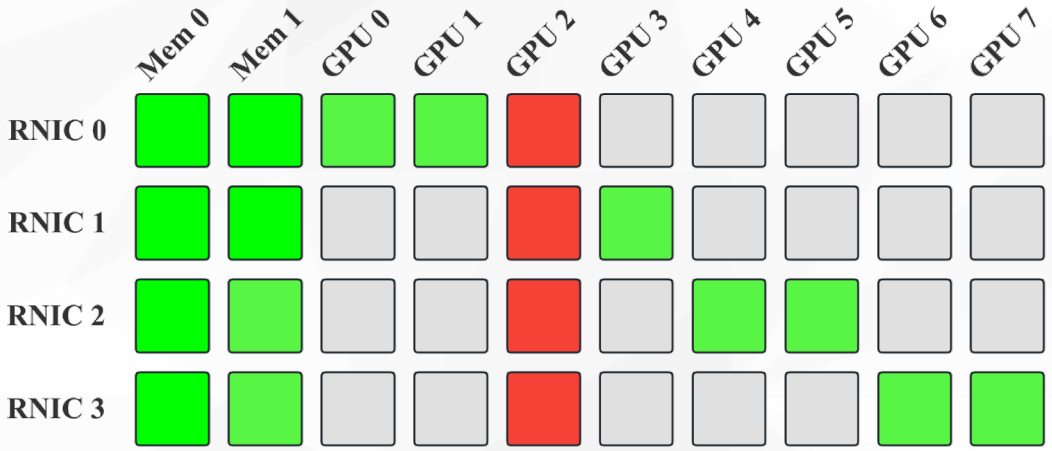
■ Bottlenecks Found: CPU Root Port Failure

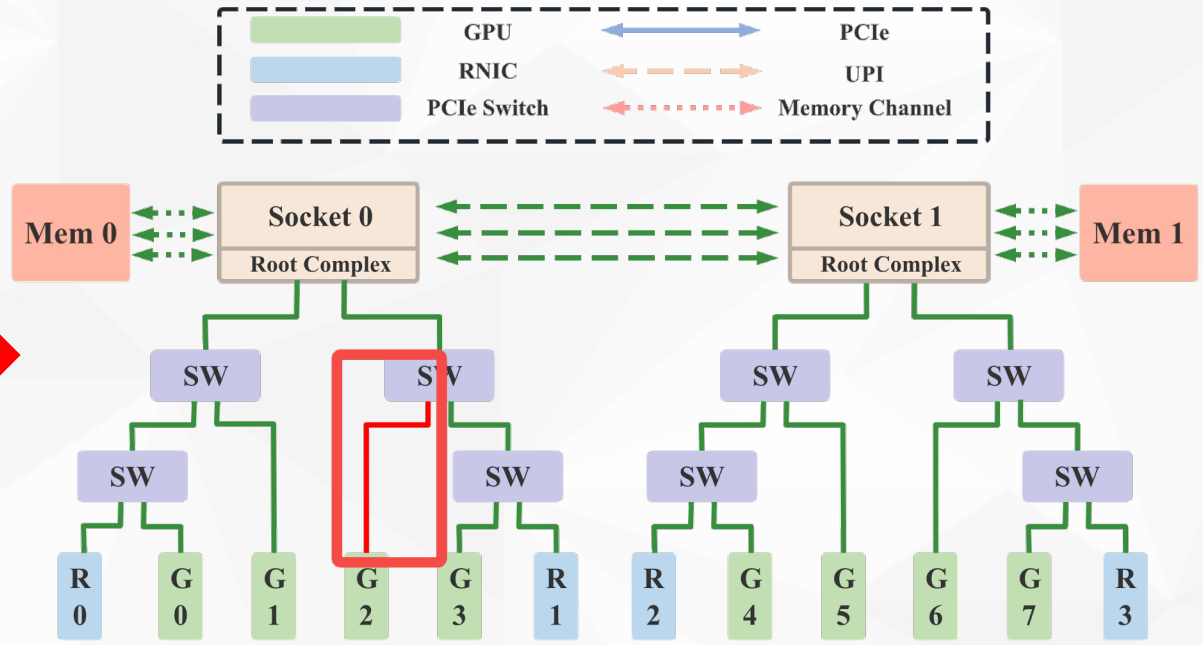




Evaluation

■ **Bottlenecks Found: GPU PCIe Link Failure**

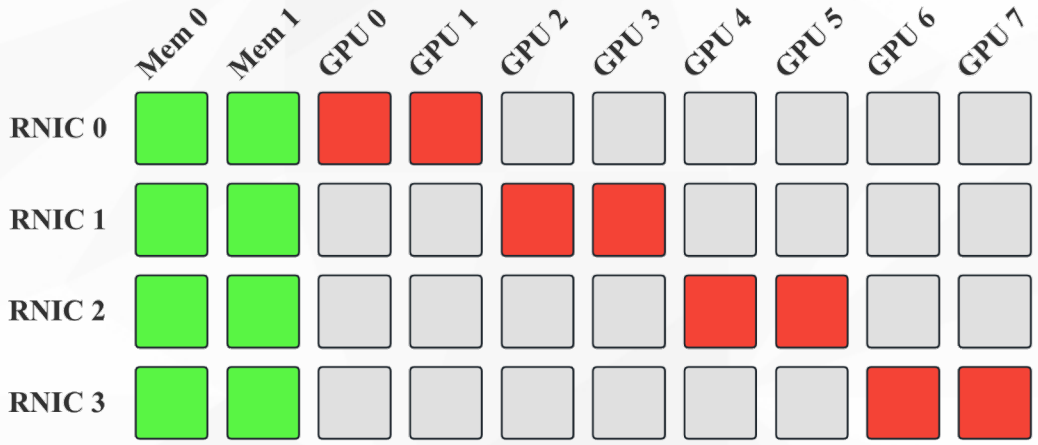




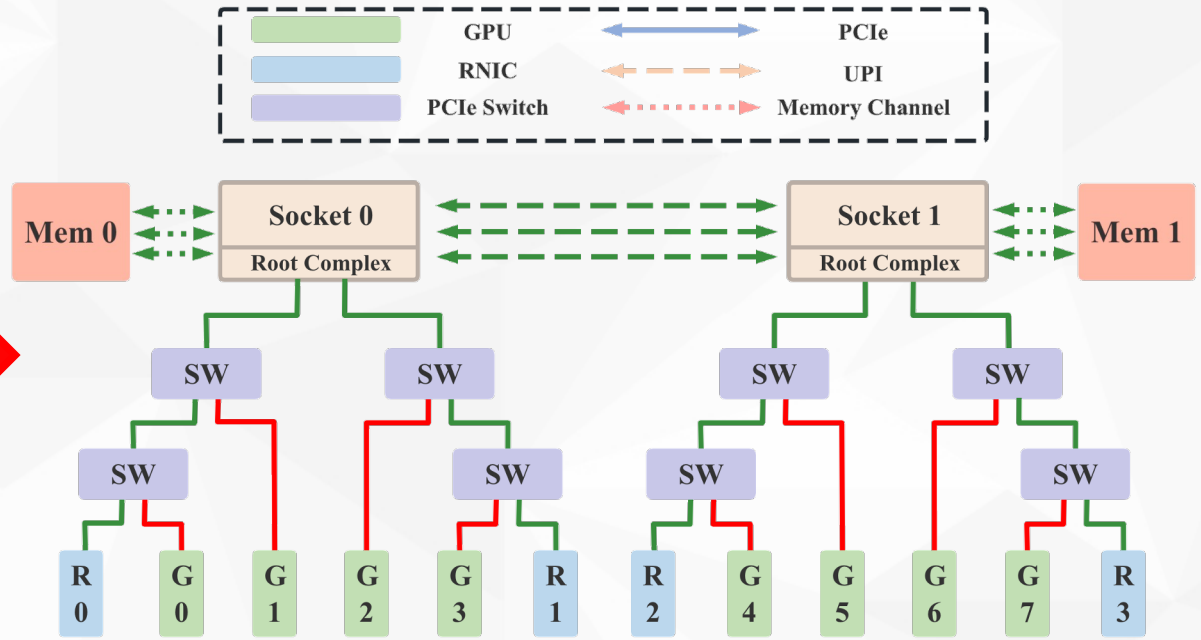
Evaluation

■ Bottlenecks Found: Misconfiguration

- Disable ATS in virtualized env or Enable ACS in non-virtualized env



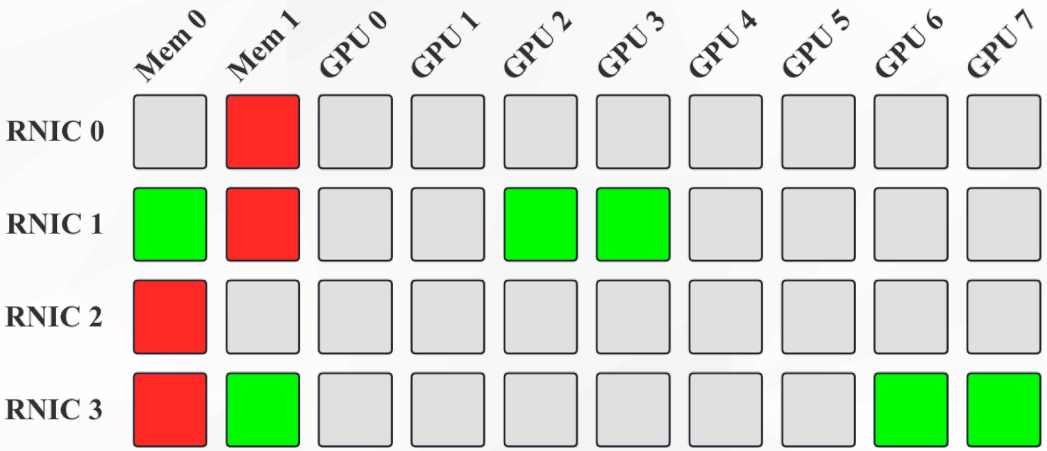
*Both bandwidth and latency are abnormal

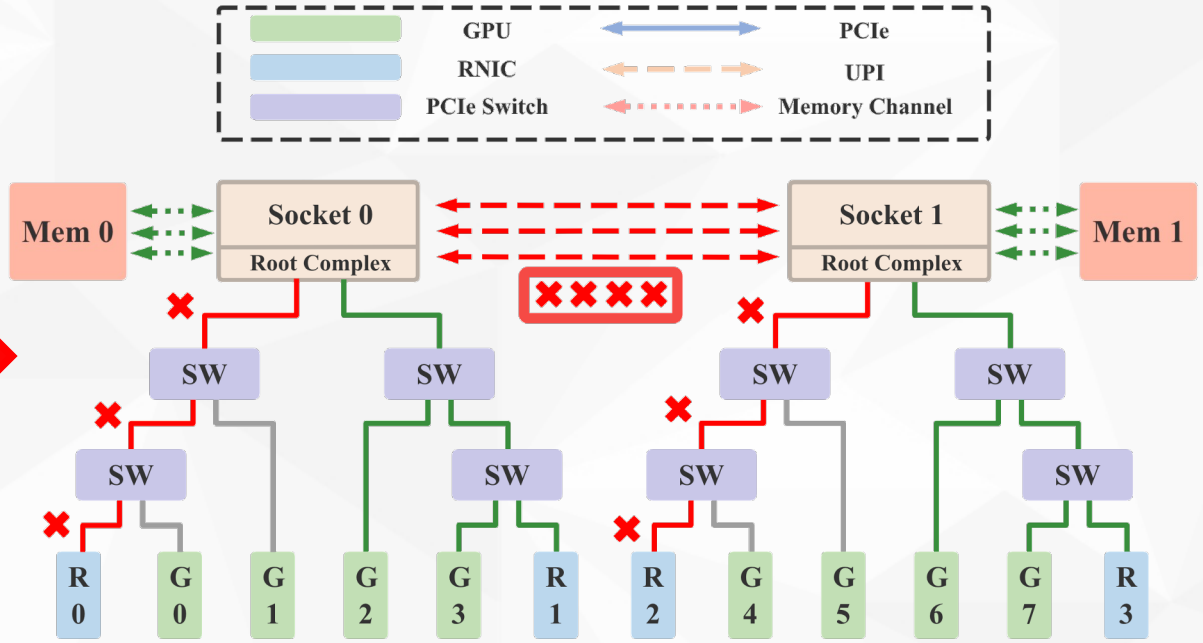


Evaluation

■ Bottlenecks Found: UPI Overloaded

- UPI is overloaded by malfunctioning applications





Conclusion

- Deploy Hostping on over 300 Nvidia DGX A100 servers.
- Hostping could effectively find intra-host bottlenecks.
- For known bottlenecks, Hostping could quickly diagnose their root causes.
- Hostping also reveals six bottlenecks we did not notice before.

THANK YOU