



2022 OFA Virtual Workshop

# GRAPHCORE IPU OVER FABRIC (IPUOF)

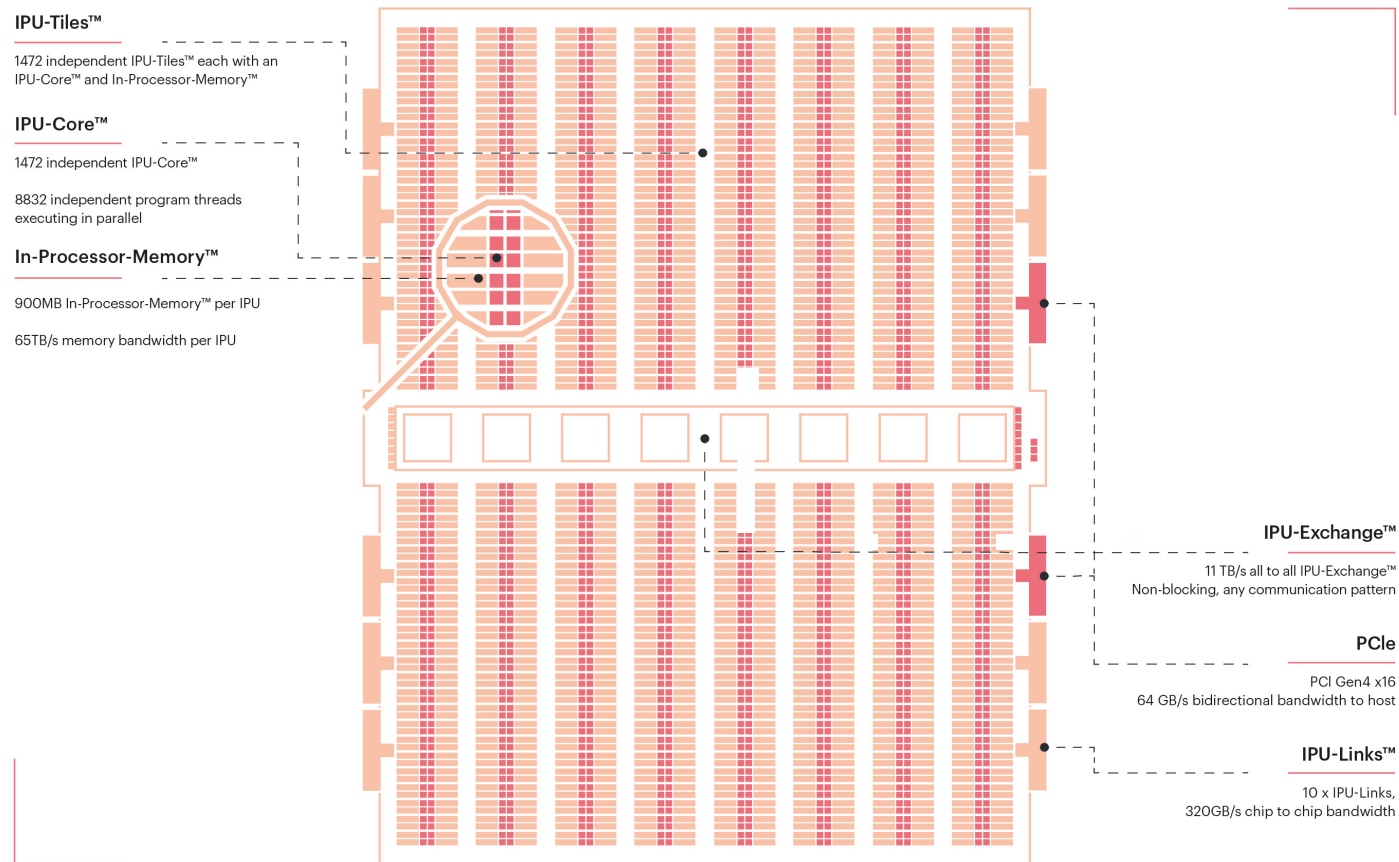
Wei Lin Guay, Dag Moxnes, Ville Silventoinen, Lars Paul Huse and Ola Tørudbakken



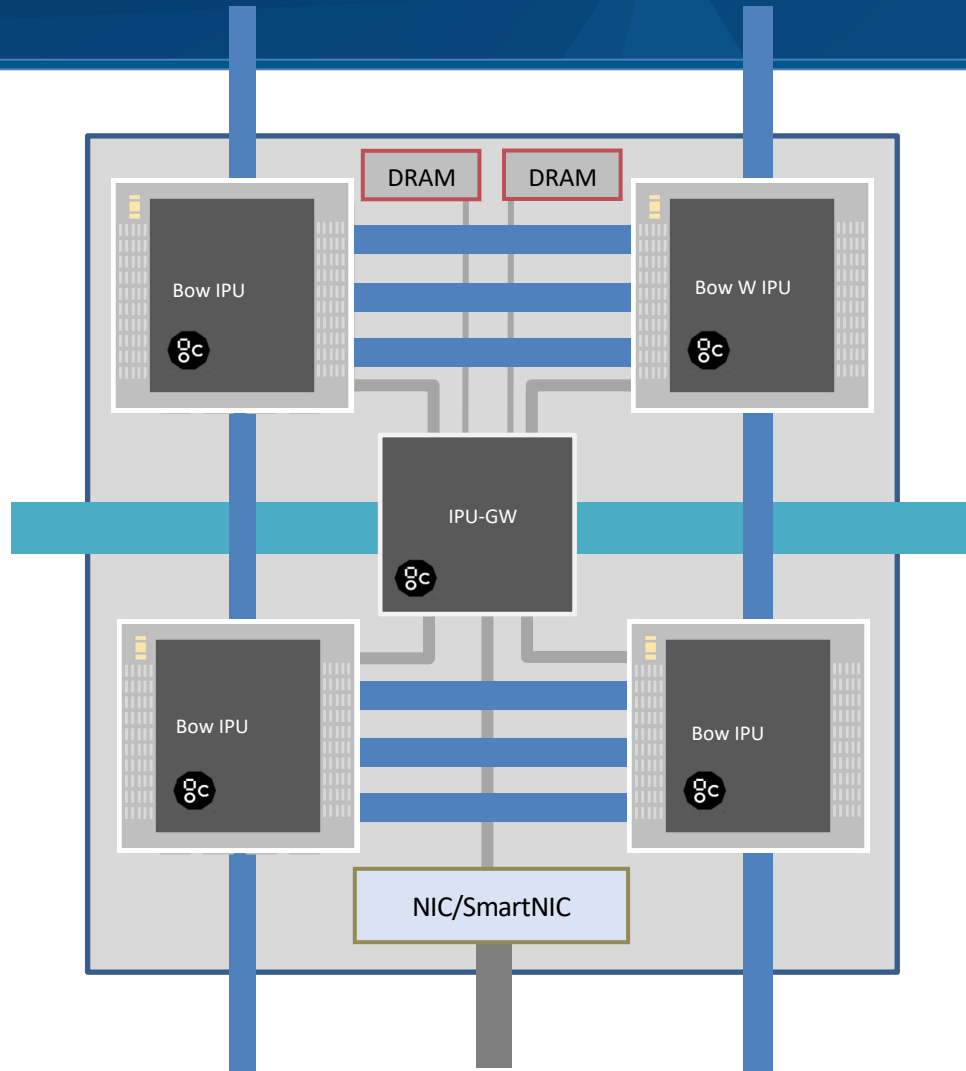
# AGENDA

- **Introduction to Graphcore IPU**
- **Graphcore BOW-M2000**
- **IPU over Fabric (IPUoF)**
  - What is IPUoF?
  - Why IPUoF?
- **Heterogeneous Memory**
  - RDMA memory registration (Linux reserved memory)
  - 3<sup>rd</sup> party device memory (Peer-to-Peer)
- **Data flow**
- **Conclusion**

# INTRODUCTION TO GRAPHCORE IPU



# GRAPHCORE BOW M2000



COMPUTE

## 4x Bow IPUs

- 1.4 PFLOP<sub>16</sub> compute
- 5,888 processor cores
- > 35,000 independent parallel threads



DATA

## Exchange Memory





- 3.6GB In-Processor Memory @ 260 TB/s
- 128GB Streaming Memory DRAM (up to 256GB)



COMMUNICATIONS

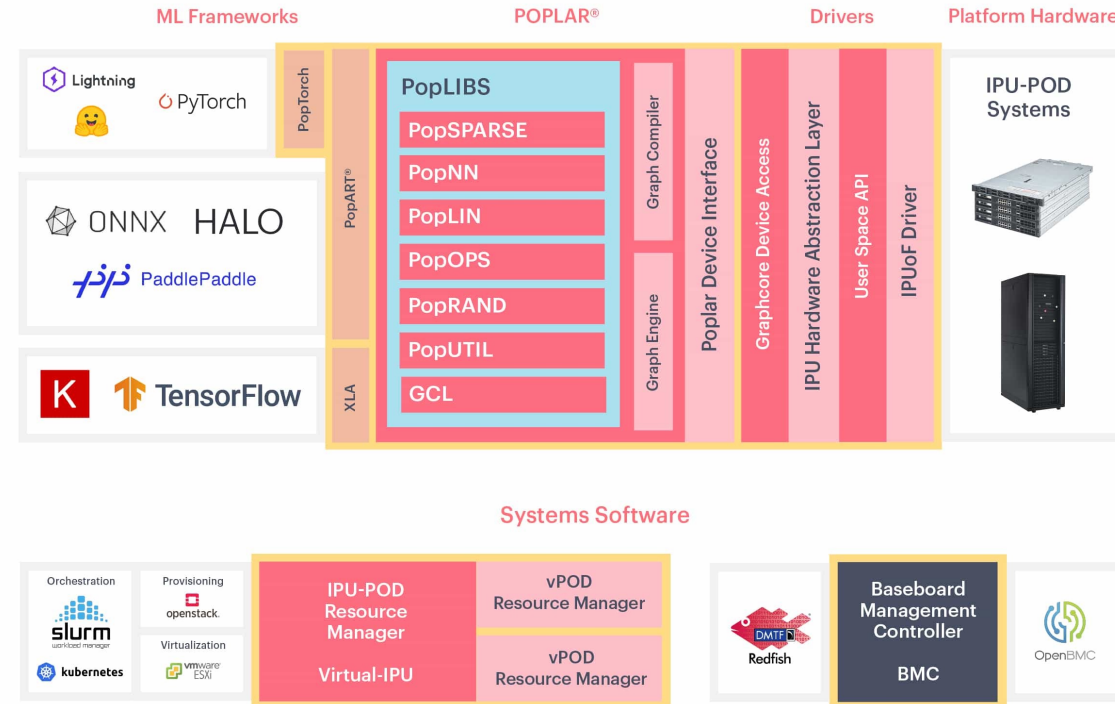
## IPU-Fabric managed by IPU-GW

- Host-Link – 100GE to Poplar Server for standard data center networking
- IPU-Link – 2D Torus for intra-POD64 communication
- GW-Link - 2x 100Gbps Gateway-Links for inter-POD64–flexible topology

-  **x16 IPU-Link [64GB/s]**
-  **Host-Link Network I/F [100Gbps]**
-  **IPU-GW Link [100Gbps]**
-  **x8 PCI<sub>4</sub> G4 [32GB/s]**

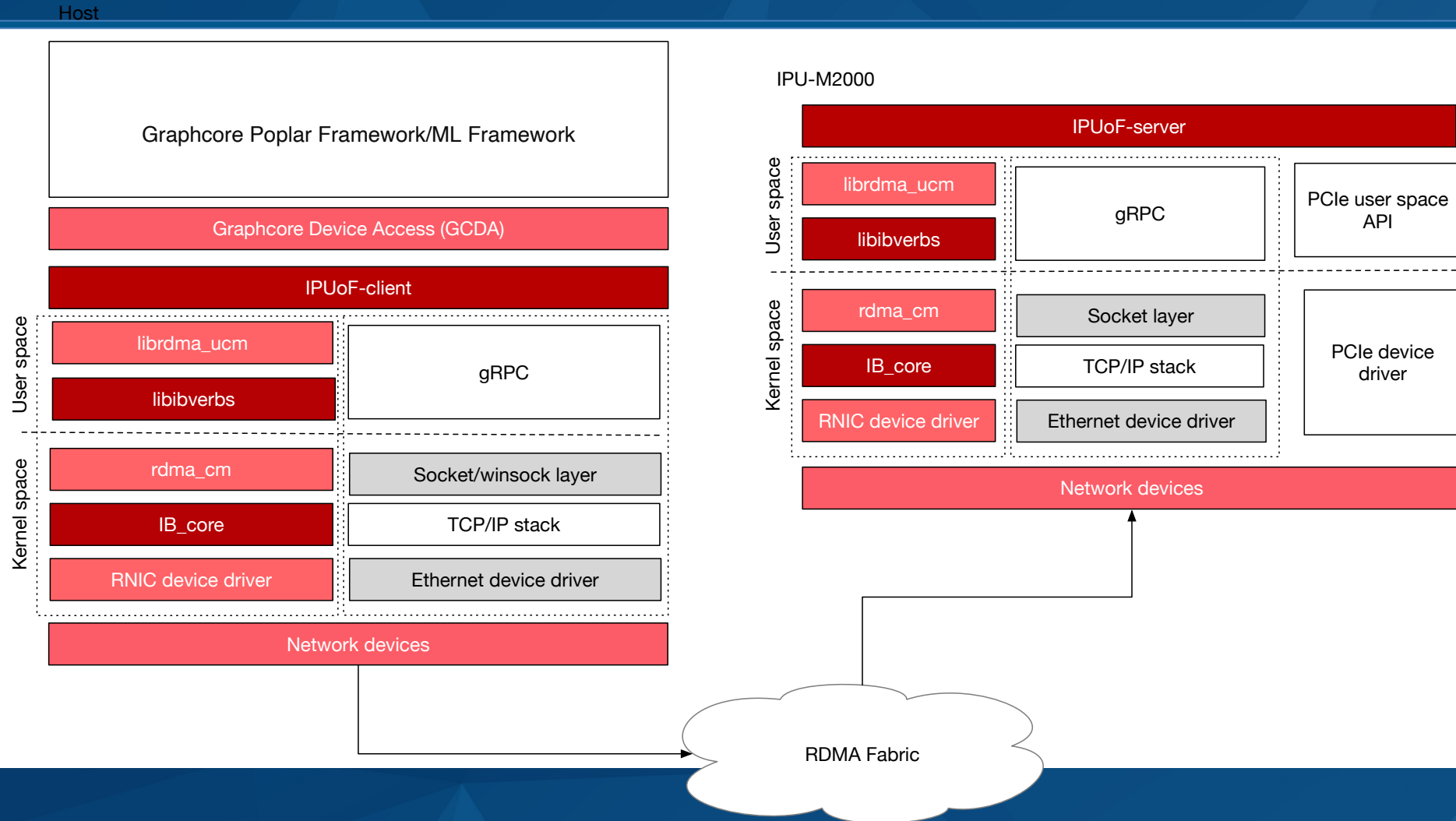
# IPU OVER FABRIC

## Software Architecture



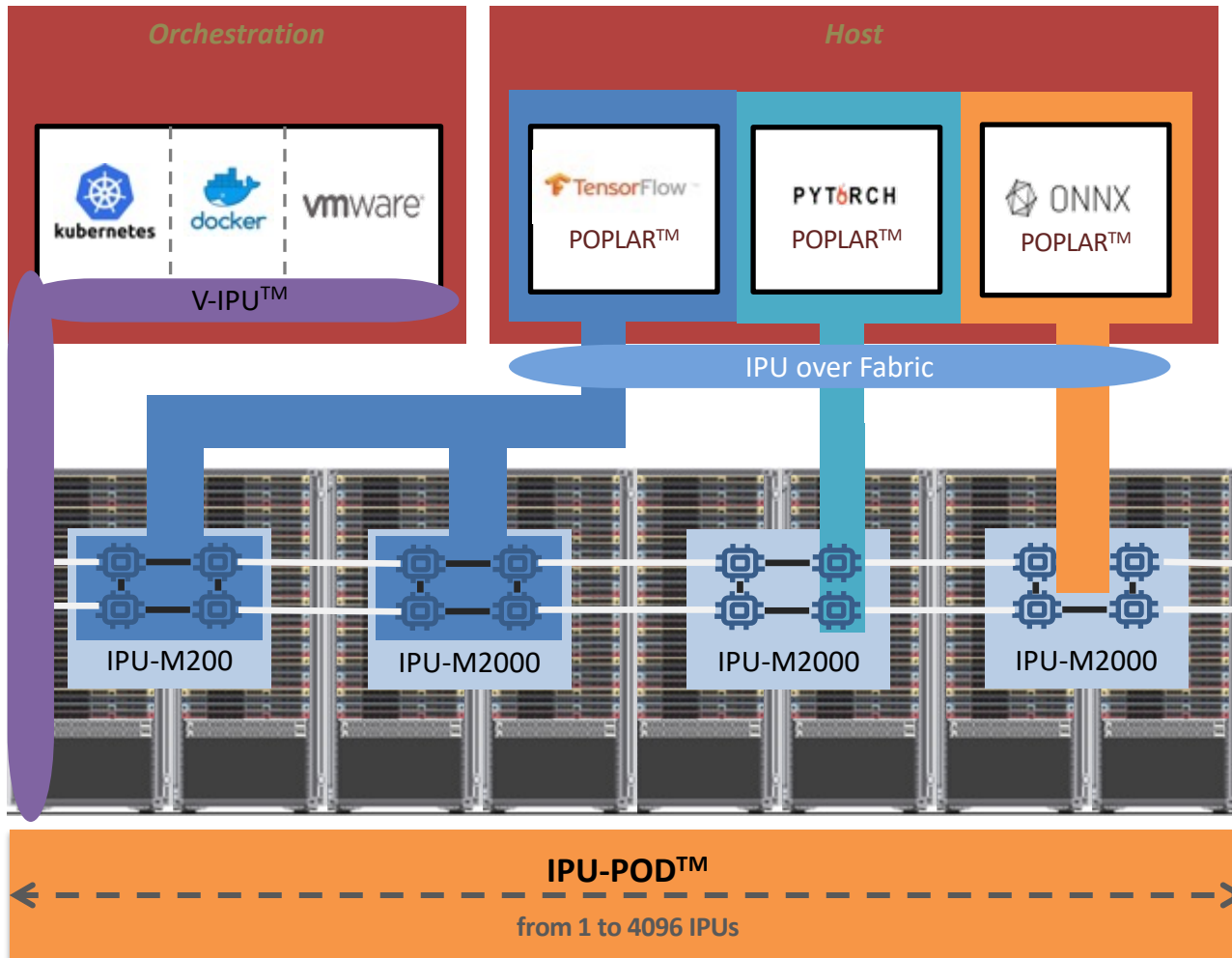
# IPU OVER FABRIC (IPUOF)

## What is IPuof?



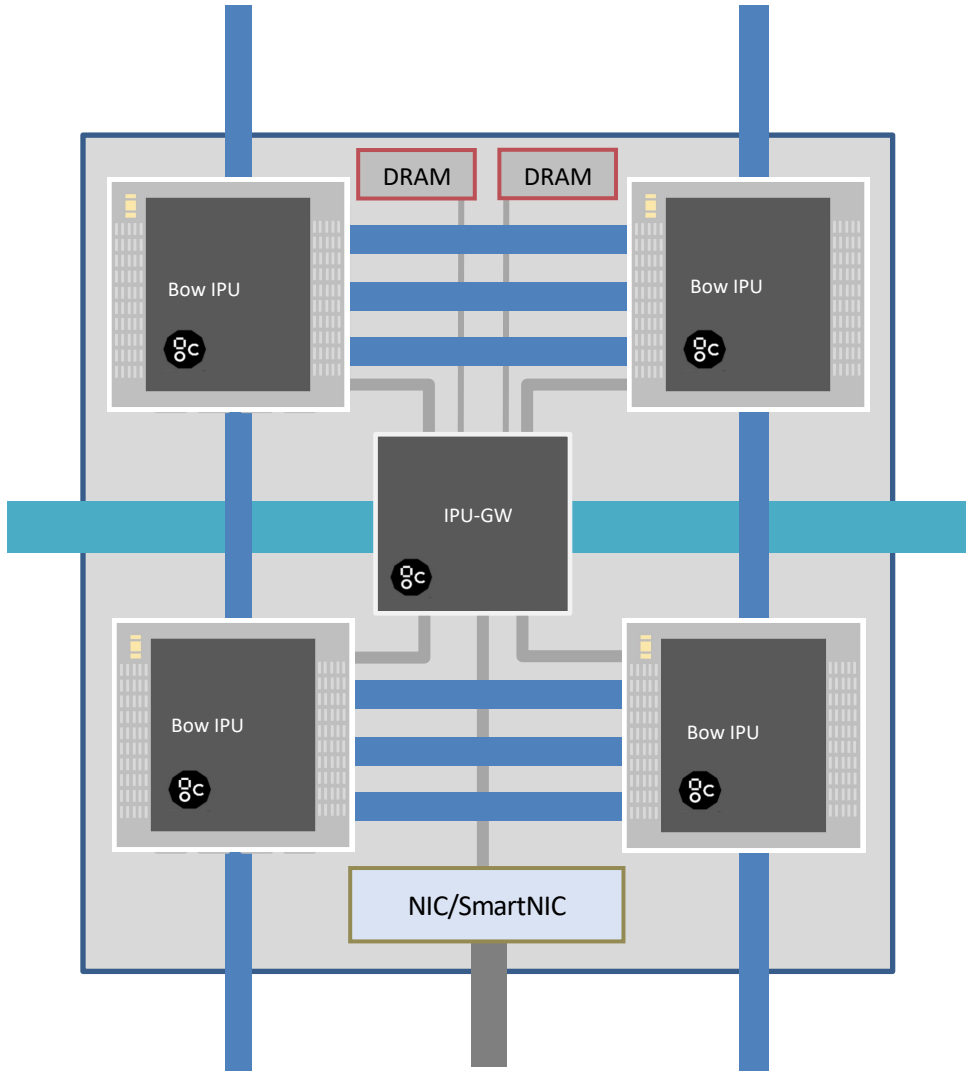
# IPU OVER FABRIC

## Why IPUoF?



- IPUoF is the data path path that provides disaggregation of IPU processing pool from the host processing pool.

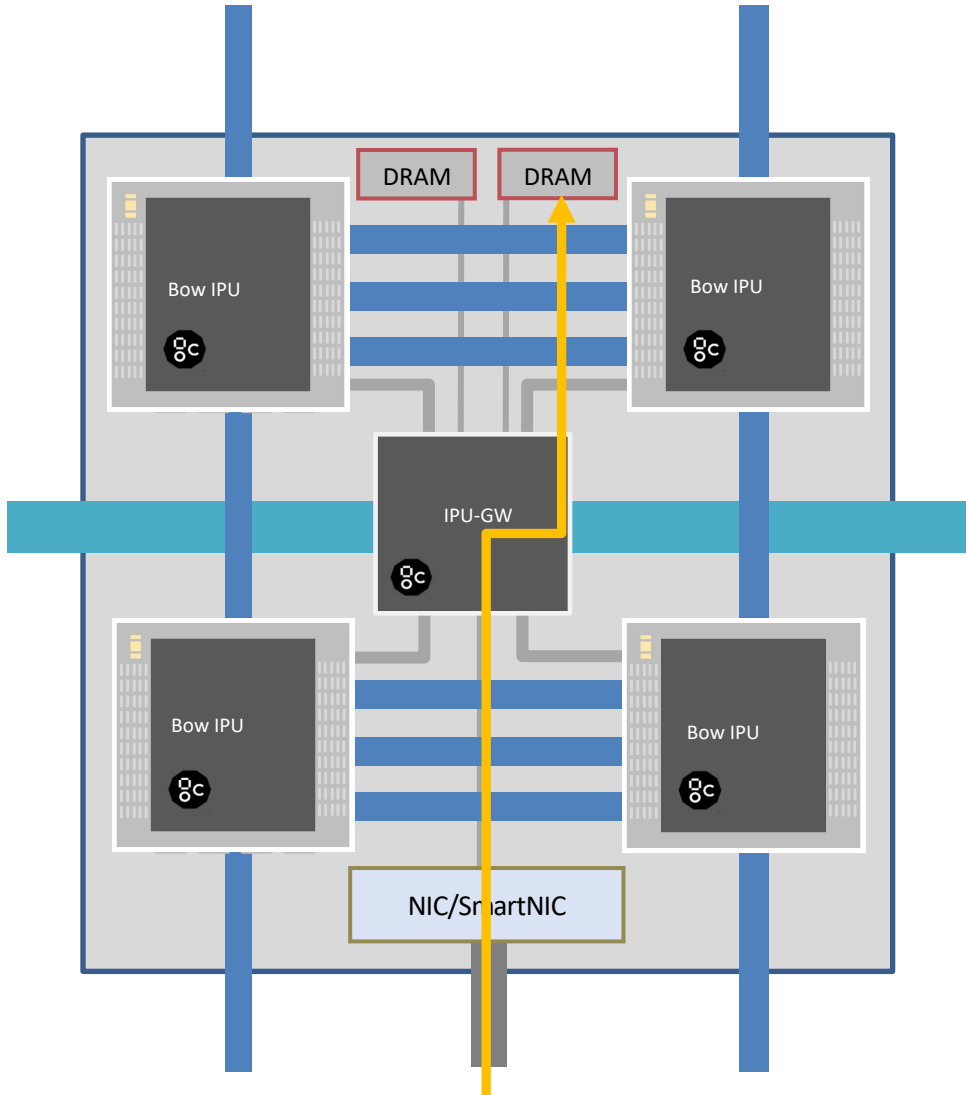
# HETEROGENOUS MEMORY



- Graphcore IPU-M2000 platform consists of heterogenous memory.

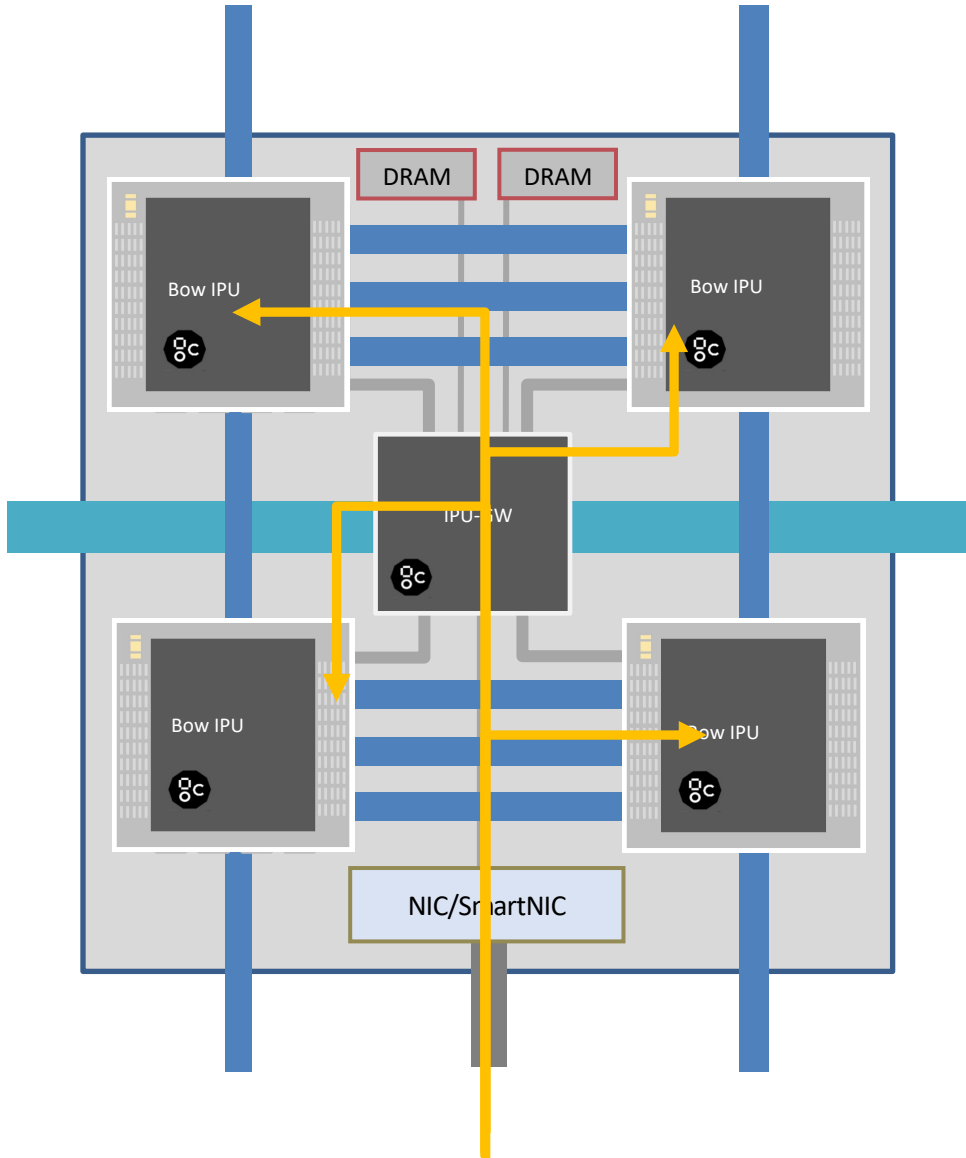


# HETEROGENOUS MEMORY



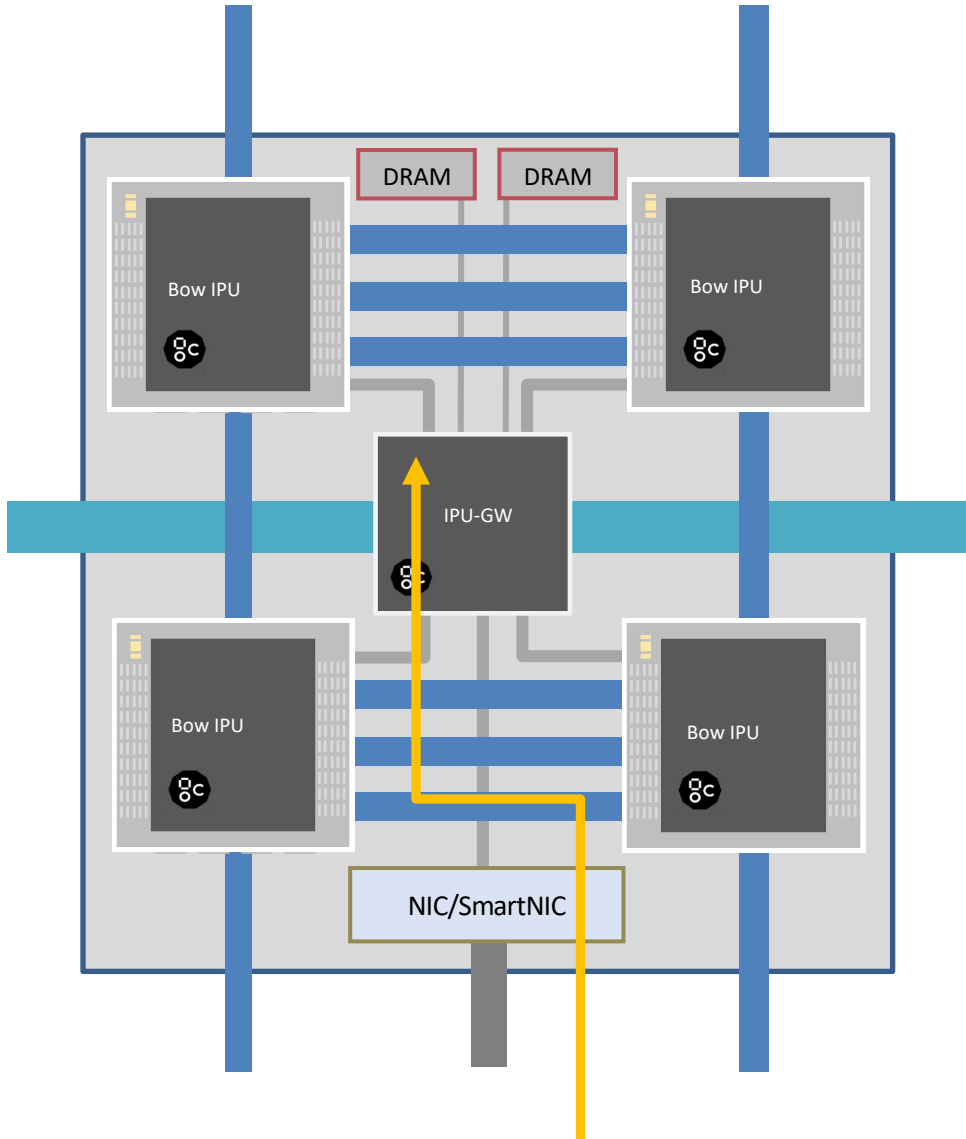
- Graphcore IPU-M2000 platform consists of heterogenous memory.
  - Graphcore Streaming memory DRAM

# HETEROGENOUS MEMORY



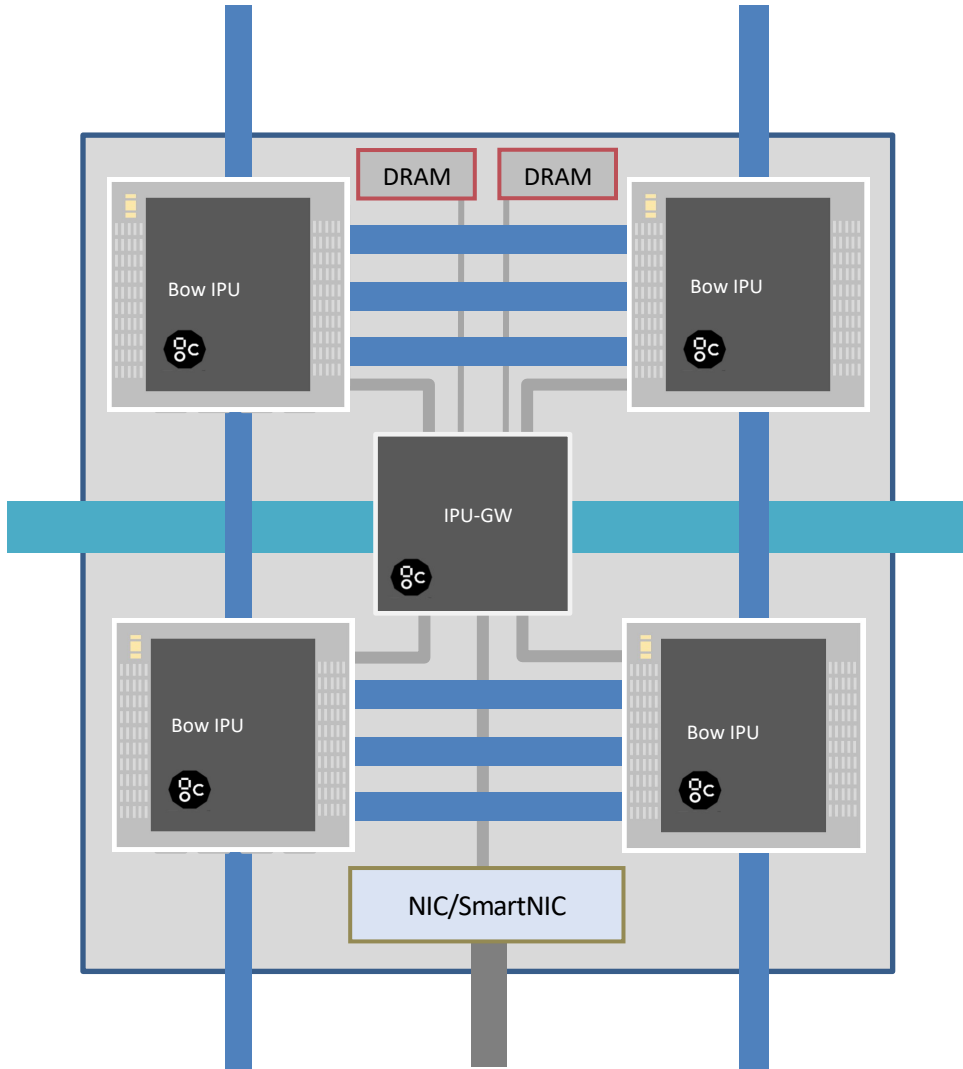
- Graphcore IPU-M2000 platform consists of heterogenous memory.
  - Graphcore Streaming memory DRAM
  - Graphcore tile memory (SRAM)

# HETEROGENOUS MEMORY



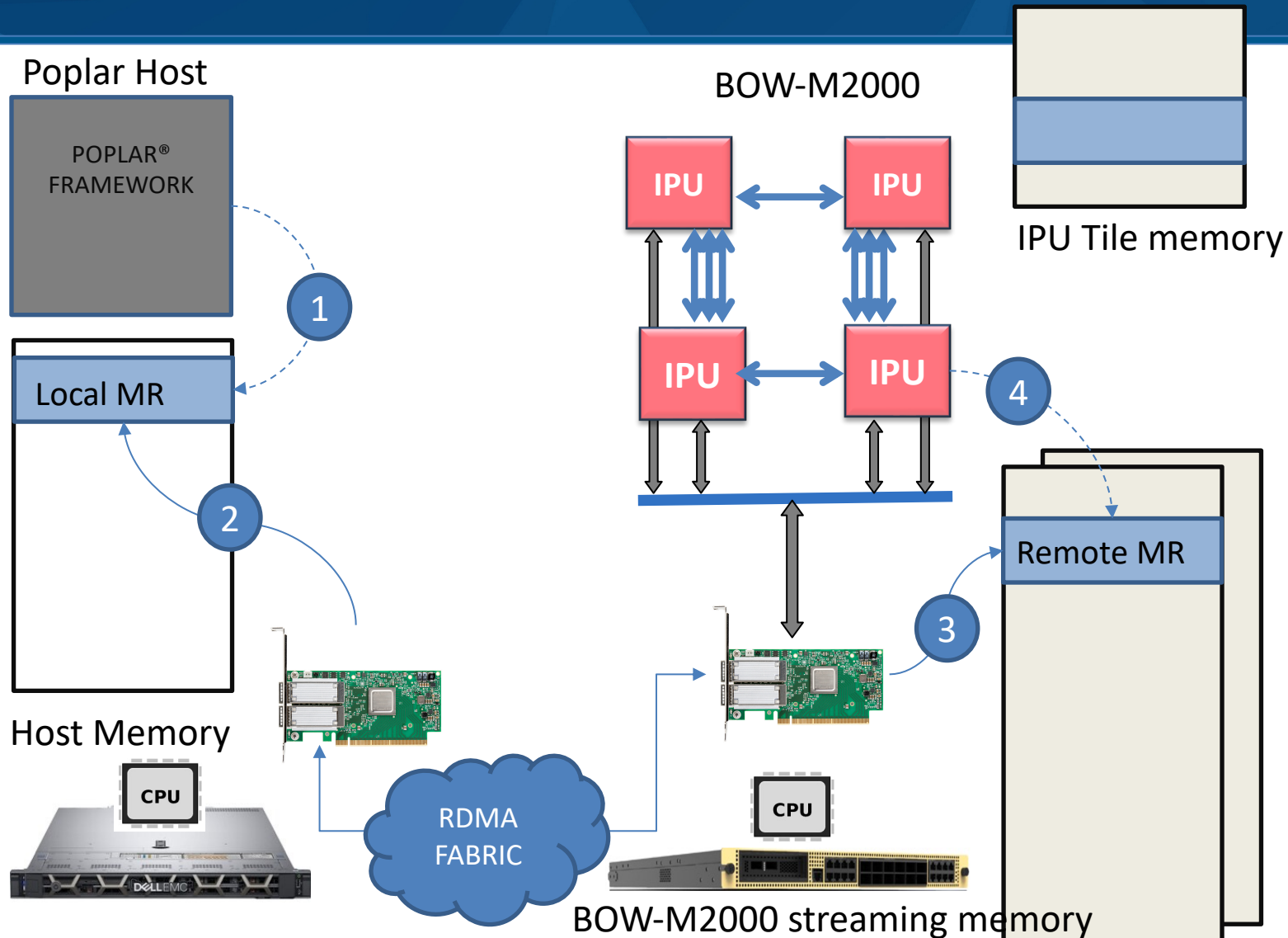
- Graphcore IPU-M2000 platform consists of heterogenous memory.
  - Graphcore Streaming memory DRAM
  - Graphcore tile memory (SRAM)
  - IPU-M2000 Host (SoC) memory

# HETEROGENOUS MEMORY



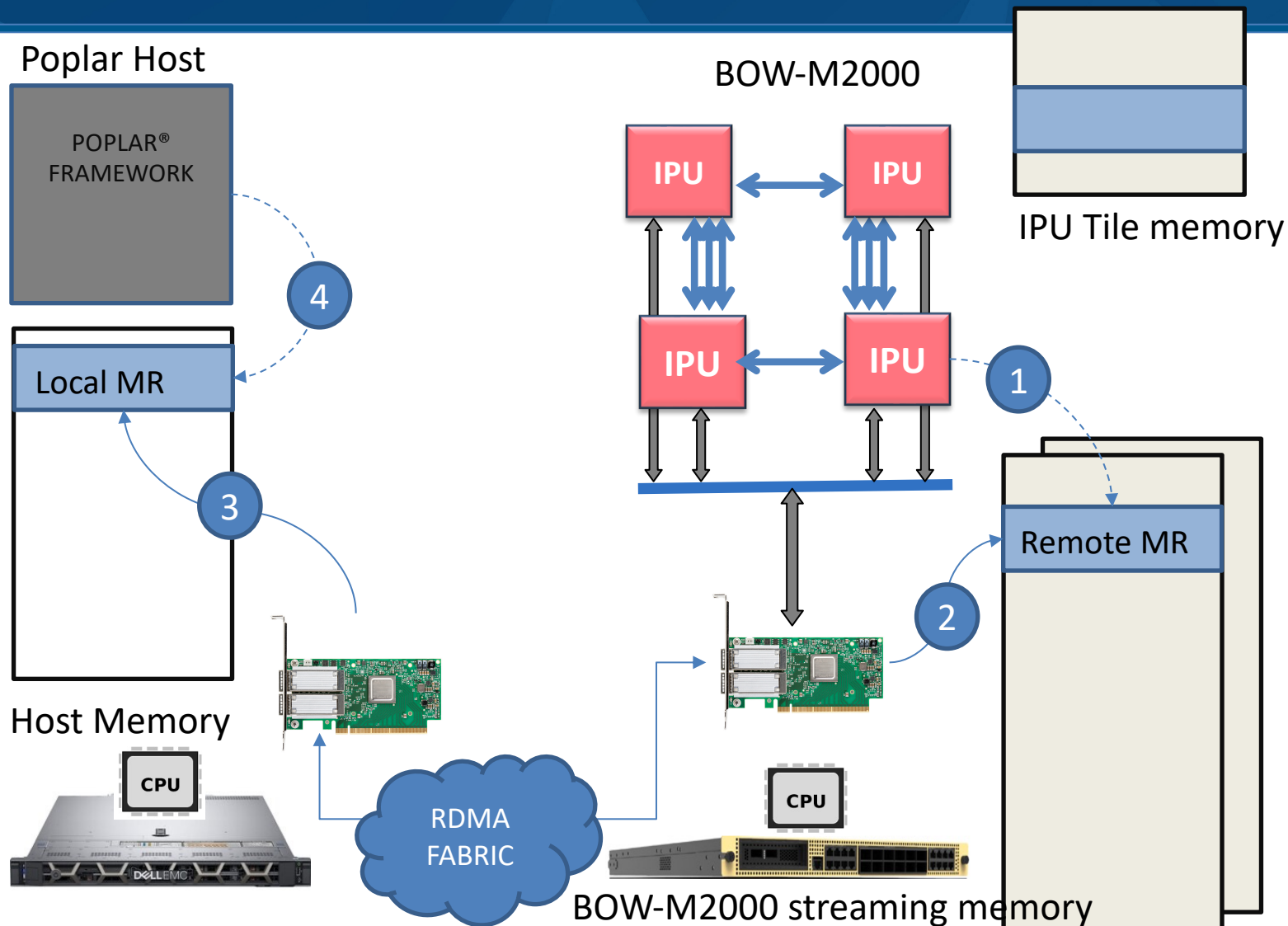
- Graphcore IPU-M2000 platform consists of heterogenous memory.
  - Graphcore Streaming memory DRAM
  - Graphcore tile memory (SRAM)
  - IPU-M2000 Host (SoC) memory
- Challenges
  - Memory is not “just” memory, from software perspective.
  - Streaming memory is used by RNIC and IPU.
  - mmap (remap\_pfn\_range) is incompatible with ibv\_reg\_mr
    - <https://www.spinics.net/lists/linux-rdma/msg70401.html>

# STREAMING MEMORY DATA PATH (HOST-TO-DEVICE)



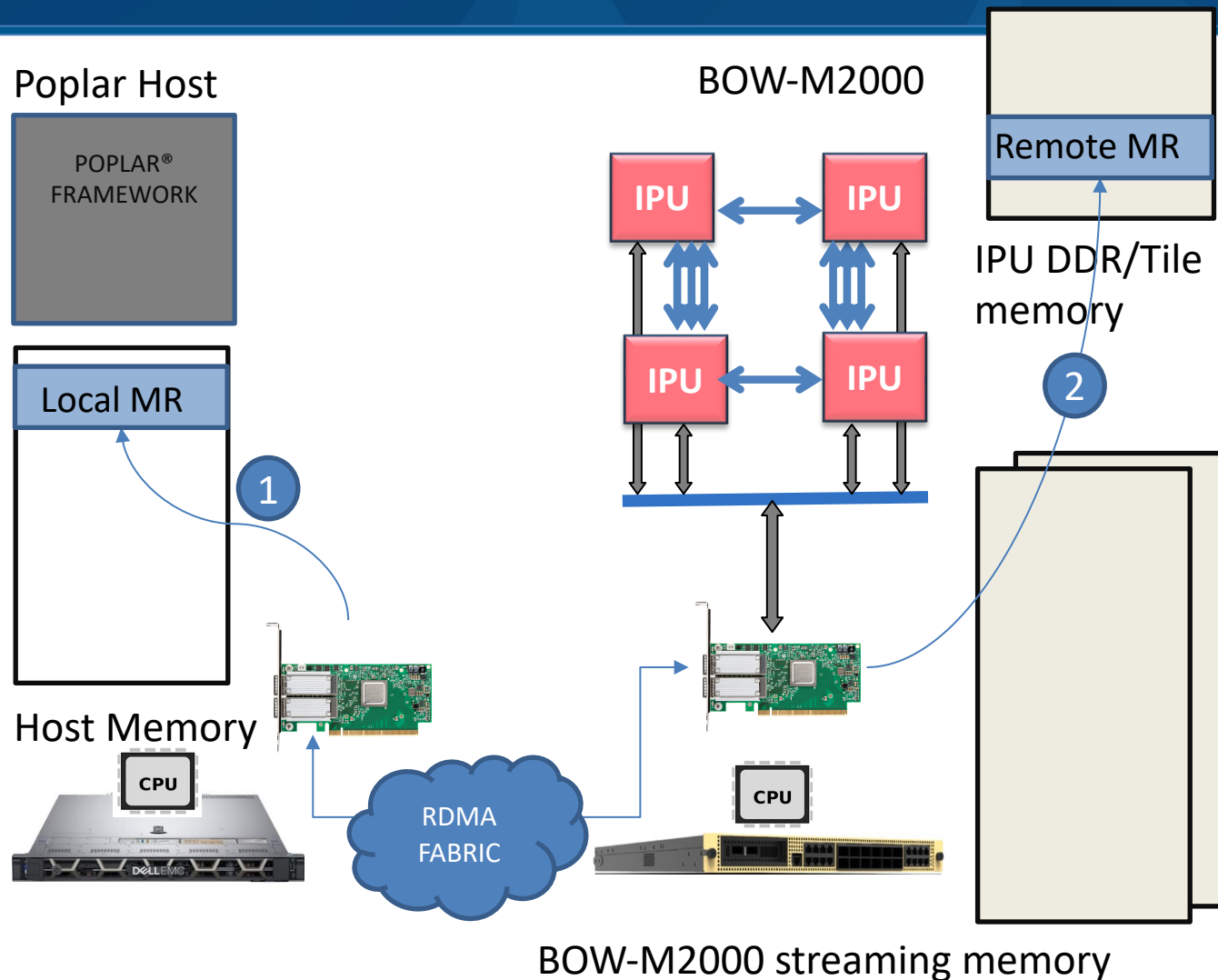
1. Application memcpy to Local MR.
2. Poplar host (IPUoF-client) initiates RNIC memRd(hostMem) from Local MR.
3. (remote) RNIC memWr(PL DDR) to Remote MR.
4. After Host-Sync, IPU (master) memRd(PL DDR) from Remote MR.

# STREAMING MEMORY DATA PATH (DEVICE-TO-HOST)



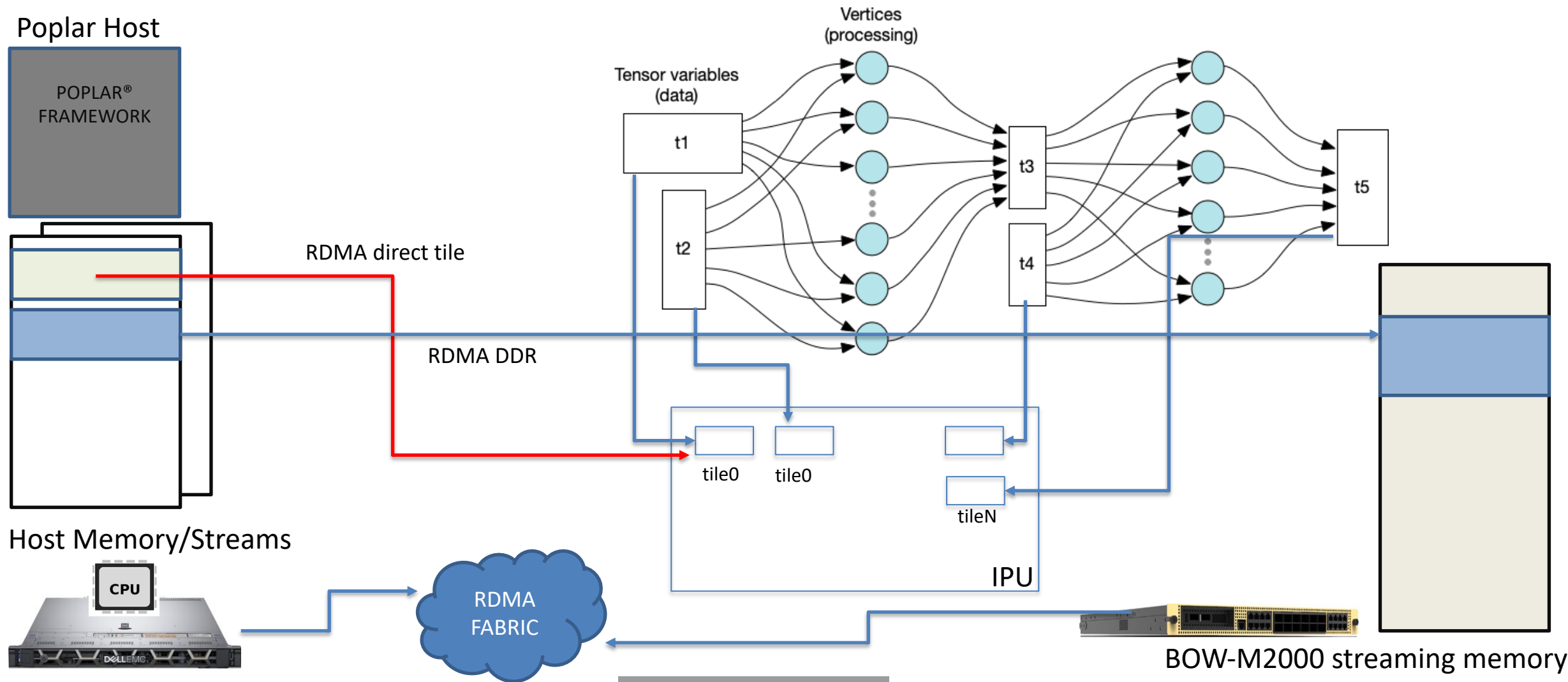
1. IPU memWr (PL DDR)
2. After Host-Sync, Poplar host initiates remote RNIC to MemRd(PL DDR).
3. RNIC memWr() to Local MR.
4. Framework/Poplar reads from Local MR (callback).

# TILE MEMORY DATA PATH (PCIE P2P)



1. PCIe peer-to-peer (P2P) supports DMA transfer between two devices.
2. Optimized data path for IPUoF host-to-device and IPUoF config/exchange access.
  - Avoid the use of bounce buffer on the IPU-M.

# DATA FLOW





# CONCLUSION

- **What is IPUoF?**
  - **Software layer that operates across different Fabric Technologies.**
  - **Implemented with RDMA/RoCE today.**
- **Why IPUoF?**
  - **Data mover to feed and fetch data from IPU.**
  - **Provide disaggregation of IPU Processing resources from the Host.**
- **Kudos to the RDMA community that support aarch64 out-of-box and support of heterogeneous memory with no changes on the RDMA stack.**



2022 OFA Virtual Workshop

**THANK YOU**

Wei Lin, Guay

