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)
  - 3rd party device memory (Peer-to-Peer)
- Data flow
- Conclusion
INTRODUCTION TO GRAPHCORE IPU

IPU-Tiles™
1472 independent IPU Tiles™ each with an IPU-Core™ and In-Processor-Memory™

IPU-Core™
1472 independent IPU Core™
8832 independent program threads executing in parallel

In-Processor-Memory™
900MB In-Processor-Memory™ per IPU
65TB/s memory bandwidth per IPU

IPU-Exchange™
11 TB/s all to all IPU-Exchange™
Non-blocking, any communication pattern

PCIe
PCI-Gen4 x16
64 GB/s bidirectional bandwidth to host

IPU-Links™
10 x IPU-Links,
3200GB chip to chip bandwidth
**GRAPHCORE BOW M2000**

- **4x Bow IPUs**
  - 1.4 PFLOPS compute
  - 5,888 processor cores
  - > 35,000 independent parallel threads

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

- **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

**Diagram Components**:
- Bow IPU
- Bow W IPU
- IPU-GW
- NIC/SmartNIC
- DRAM

**Connections**:
- x16 IPU-Link [64GB/s]
- Host-Link Network I/F [100Gbps]
- IPU-GW Link [100Gbps]
- x8 PCIe G4 [32GB/s]
IPU OVER FABRIC (IPUoF)

What is IPUoF?

Graphcore Poplar Framework/ML Framework

Graphcore Device Access (GCDA)

IPUoF-client

librdma_ucm
libibverbs
rdma_cm
IB_core
RNIC device driver

User space

Kernel space

Network devices

RDMA Fabric

IPU-M2000

IPUoF-server

gRPC

User space

Socket layer
TCP/IP stack
Ethernet device driver
PCIe device driver
PCIe user space API

Network devices

Host
IPU OVER FABRIC

Why IPUoF?

- IPUoF is the data path path that provides disaggregation of IPU processing pool from the host processing pool.
• Graphcore IPU-M2000 platform consists of heterogenous memory.
• Graphcore IPU-M2000 platform consists of heterogenous memory.
  - Graphcore Streaming memory DRAM
• Graphcore IPU-M2000 platform consists of heterogenous memory.
  – Graphcore Streaming memory DRAM
  – Graphcore tile memory (SRAM)
Graphcore IPU-M2000 platform consists of heterogeneous memory.
- Graphcore Streaming memory DRAM
- Graphcore tile memory (SRAM)
- IPU-M2000 Host (SoC) 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
    o https://www.spinics.net/lists/linux-rdma/msg70401.html
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.
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).
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

Poplar Host

POPLAR® FRAMEWORK

RDMA direct tile

RDMA DDR

Host Memory/Streams

CPU

RDMA FABRIC

Vertices (processing)

Tensor variables (data)

t1

t2

t3

t4

t5

BOW-M2000 streaming memory

© OpenFabrics Alliance
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.
THANK YOU
Wei Lin, Guay

Graphcore