

Topology-aware Process Mapping on Supercomputer Fugaku

- **Background & Motivation**
- **Methods**
- **Experiment Results**
- **Summary**

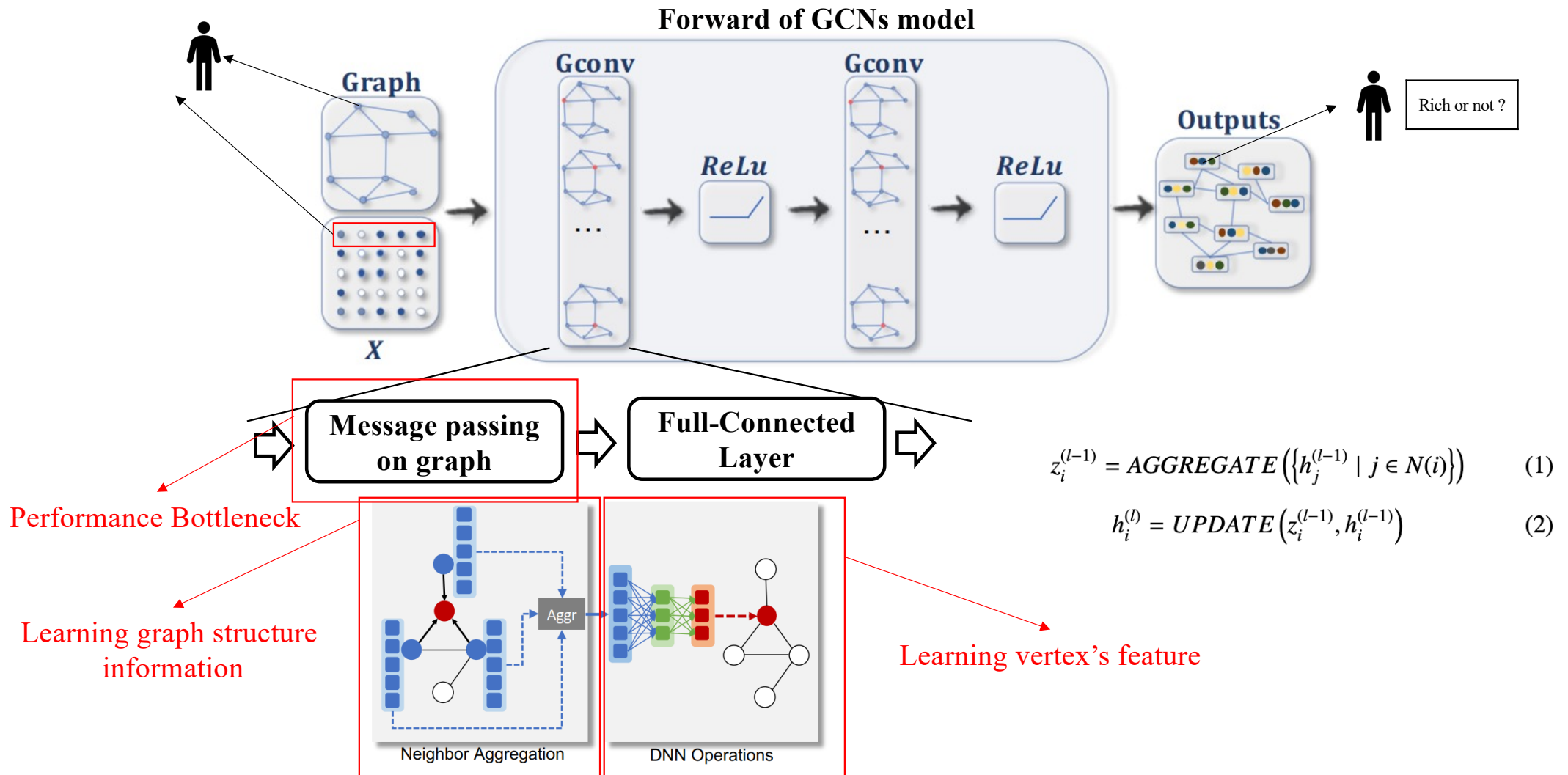


- Background & Motivation



Background & Motivation - Distributed Full-batch GNNs training

- Performance analysis of distributed GNN



Methods – Overview (Proposed workflow)

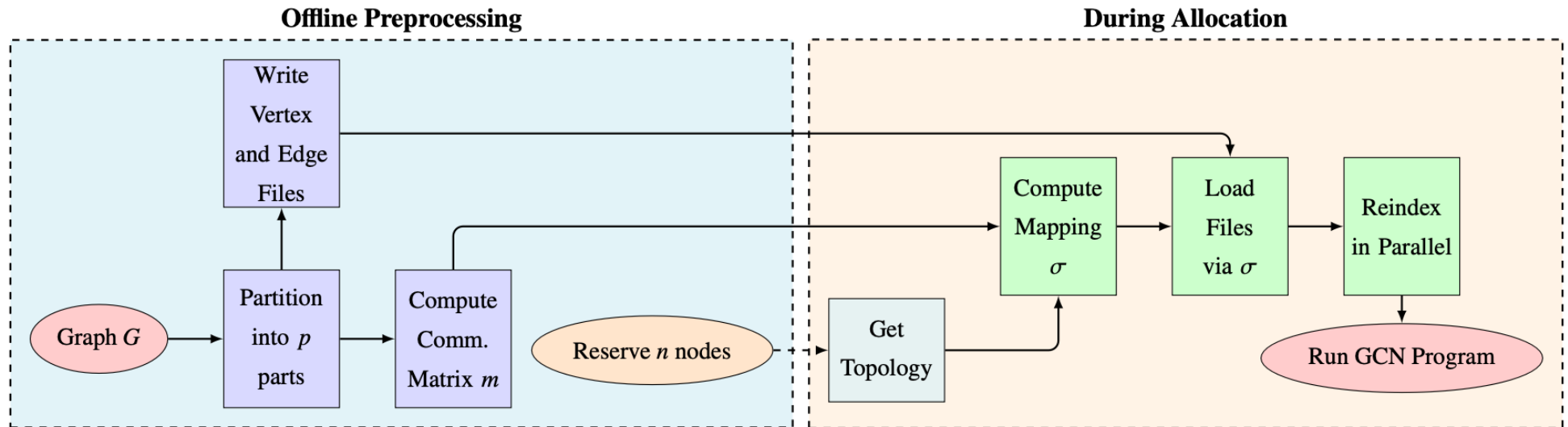


Fig. 3: Workflow for Topology-Aware Graph Distribution and Execution

Before job allocation (offline preprocessing)

1. **Graph partitioning and save results (separate vertices (feature vector, label) and edges files)**
2. Get communication pattern matrix

During Allocation (Online)

1. **Get physical interconnection topology**
2. **Compute process mapping based on communication matrix and topology**
3. Load vertices and edges files and reindex id of vertices and edges based on process mapping result
4. Go to GNN training

Methods - Overview

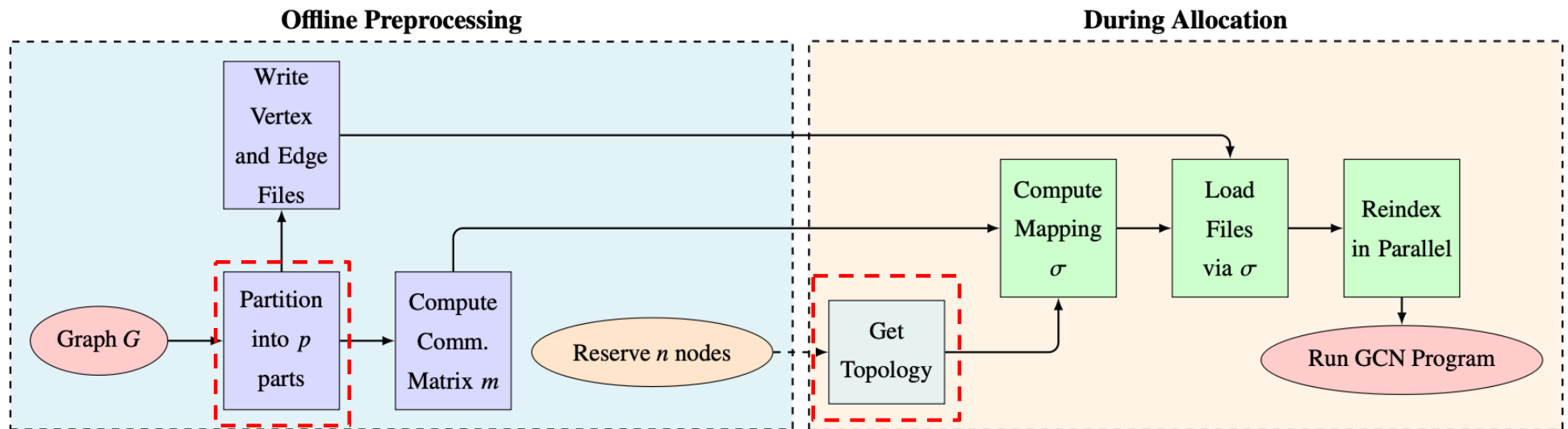


Fig. 3: Workflow for Topology-Aware Graph Distribution and Execution

Graph partitioning algorithm

- Default algorithm: *Metis* (min-cut)
- Topology-aware partitioning algorithm: *Scotch*

Futjisu API to get physical interconnection and coordinates

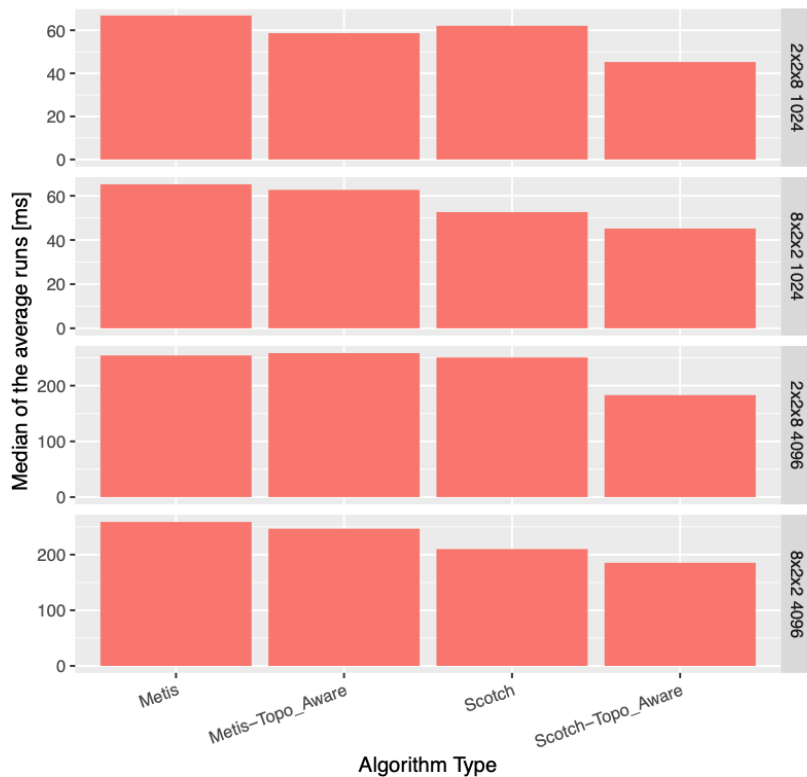
- `FJMPI_Topology_get_coords`

Experiment results – Communication performance on dataset Ogbn-products for GNN

4 different setting:

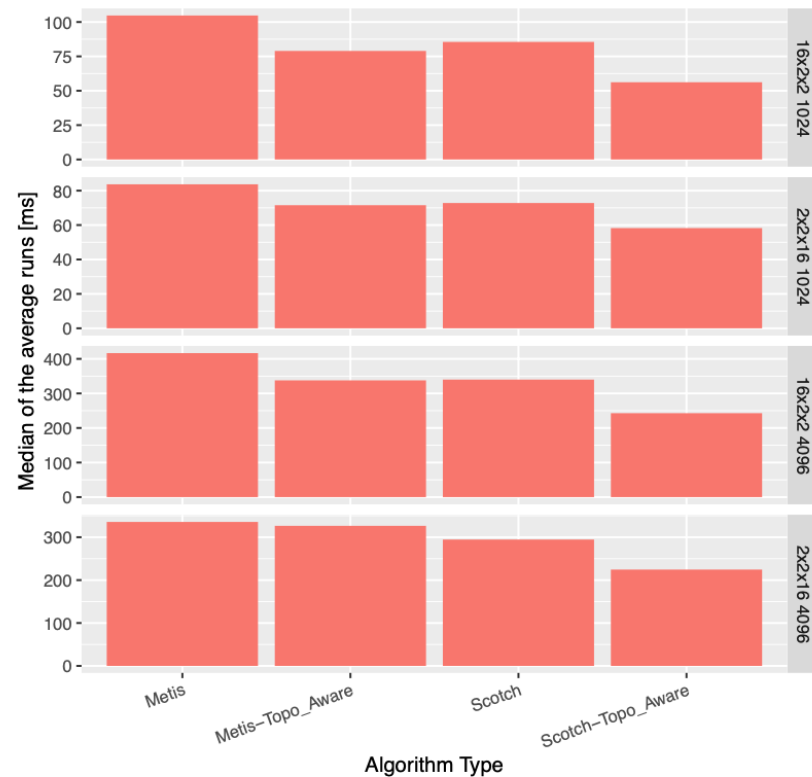
1. Default graph partitioning **Metis**
2. **Metis** + Topology-aware process mapping
3. Topology-aware graph partitioning **Scotch**
4. **Scotch** + Topology-aware process mapping

All-to-All-V timings (128 ranks, 4 procs per nodes – product)



(a) Alltoallv runtime for Ogbn-product with 128 processes/parts

All-to-All-V timings (256 ranks, 4 procs per nodes – product)



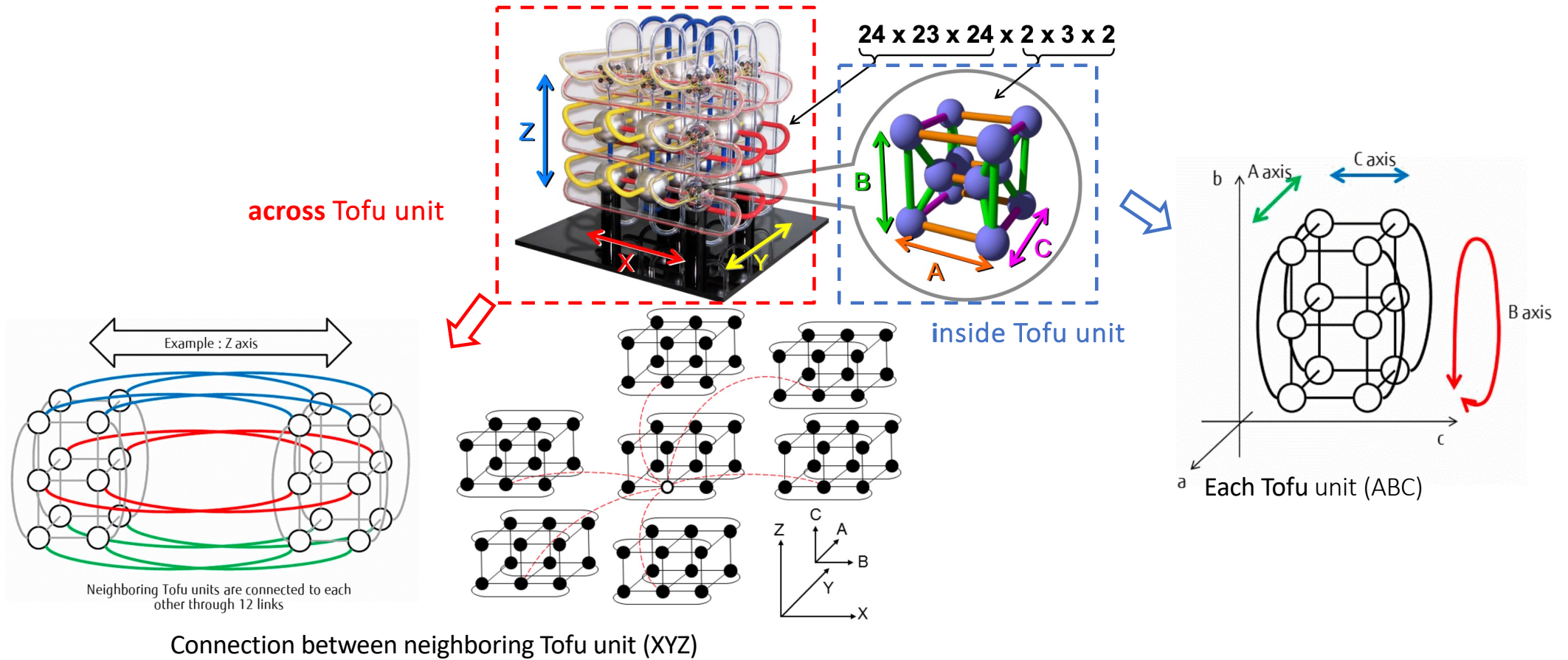
(b) Alltoallv runtime for Ogbn-product with 256 processes/parts

**Scotch + Topology-aware
process mapping achieve the
best performance**

Want to do the same thing for tomography
reconstruction

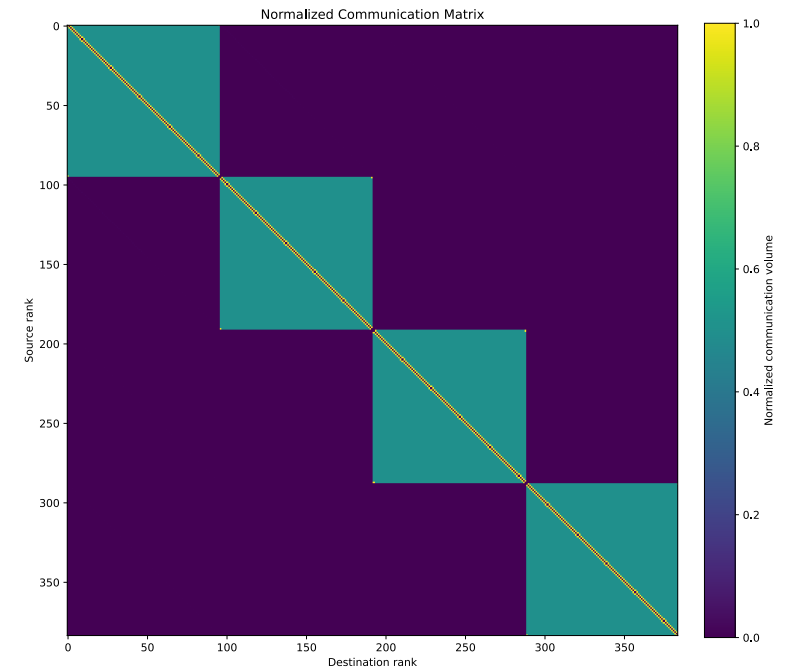
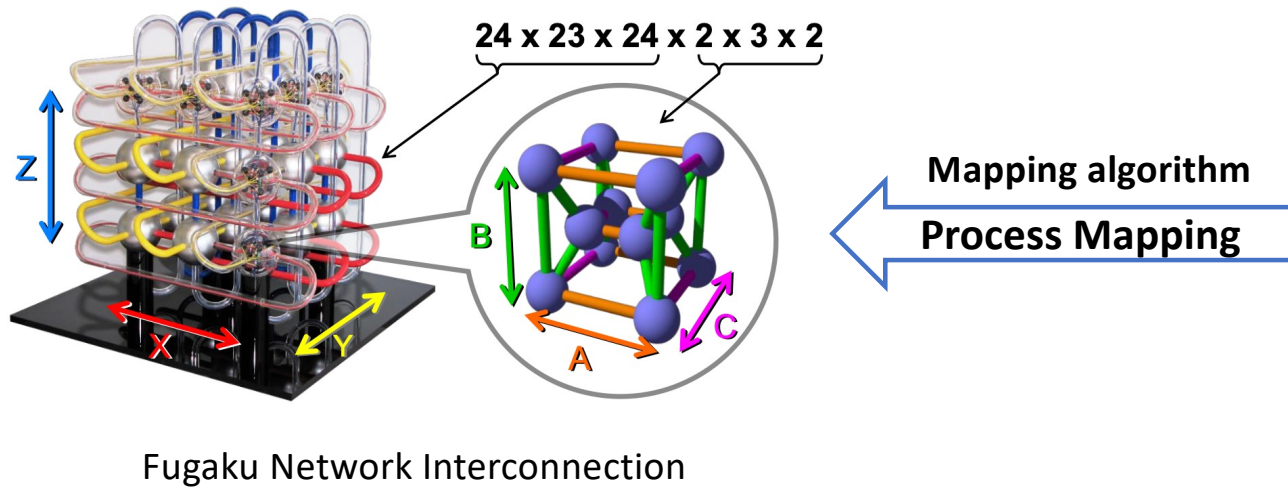
Background & Motivation – Fugaku interconnection network

- TofuD – 6D mesh/torus Interconnect (XYZABC)
- Each Tofu unit is 2x3x2, All Tofu units are 24x23x24



Background & Motivation – Process mapping problem

- Imbalanced communication pattern in GNN -> Uniform interconnection



Communication pattern in
Tomography reconstruction

**How to do the process mapping to reduce communication overhead
by better interconnection utilization**

Fugaku topology file

158979 lines

158976 nodes

Each has 10 neighbours

0

158976 1589760

0 101

0 1 10 1 2 4 6 12 288 6624 152352 6336 276

node_id function (x,y,z,a,b,c) -> $x*12+y*288+z*6624+a+b*2+c*6$

Booking a set of (384 nodes)

My Dimension=3

My Shape: X=48, Y= 2, Z= 4

rank to xyz: rank=0, hostname=d16-7200c, cpu_id=12, (X,Y,Z)=(0,0,0), (x,y,z,a,b,c)=(20,10,0,0,0,0)

rank to xyz: rank=1, hostname=d16-3208c, cpu_id=12, (X,Y,Z)=(1,0,0), (x,y,z,a,b,c)=(21,10,0,0,0,0)

rank to xyz: rank=2, hostname=b16-4000c, cpu_id=12, (X,Y,Z)=(2,0,0), (x,y,z,a,b,c)=(22,10,0,0,0,0)

rank to xyz: rank=3, hostname=b16-0008c, cpu_id=12, (X,Y,Z)=(3,0,0), (x,y,z,a,b,c)=(23,10,0,0,0,0)

(...)

rank to xyz: rank=381, hostname=b16-4109c, cpu_id=12, (X,Y,Z)=(45,1,3), (x,y,z,a,b,c)=(22,10,0,1,1,1)

rank to xyz: rank=382, hostname=d16-3101c, cpu_id=12, (X,Y,Z)=(46,1,3), (x,y,z,a,b,c)=(21,10,0,1,1,1)

rank to xyz: rank=383, hostname=d16-7109c, cpu_id=12, (X,Y,Z)=(47,1,3), (x,y,z,a,b,c)=(20,10,0,1,1,1)

- Methods



Using Scotch

- Extract node id from node file => phys_sub_nodes_file.txt
- Build sub topology : amk_grf tofu.grf -lphys_sub_nodes_file.txt > sub_tofu.tgt
- Build com_mat.grf
- Create strong strat (but can use default): strategy = 'm{asc=b{width=3,bnd=d{pass=40,diff=1,rem=0}{move=2000,pass=-1,bal=0},org=f{move=2000,pass=-1,bal=0}},low=r{job=t,bal=0,map=t,poli=S,sep=((m{asc=b{bnd=(d{pass=40,type=b})}{move=120,pass=-1,bal=0,type=b},org=f{move=120,pass=-1,bal=0,type=b},width=3},low=h{pass=10}{move=120,pass=-1,bal=0,type=b},vert=120,rat=0.8})|m{asc=b{bnd=(d{pass=40,type=b})}{move=120,pass=-1,bal=0,type=b},org=f{move=120,pass=-1,bal=0,type=b},width=3},low=h{pass=10}{move=120,pass=-1,bal=0,type=b},vert=120,rat=0.8})|m{asc=b{bnd=(d{pass=40,type=b})}{move=120,pass=-1,bal=0,type=b},org=f{move=120,pass=-1,bal=0,type=b},width=3},low=h{pass=10}{move=120,pass=-1,bal=0,type=b},vert=120,rat=0.8})},vert=10000,rat=0.8,type=h}';
- Call gmap: gmap -m"\$strategy" com_mat.grf sub_tofu.tgt > scotch_strat_res
- Enforce mapping during execution (using rank reordering)
- In general no gain from standard mapping

Mapping example

[Here](#)

- Summary



Summary

Full chain to map application to Fugaku

Worked well for GNN training app

Does not work for Tomography reconstruction with all reduce. But Allreduce notably good on Fugaku..



3.1 Allreduce

Communication time [sec]

Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.20E-07	1.30E-07	1.40E-07
4	4.55E-06	6.11E-06	1.24E-05
8	4.56E-06	6.29E-06	1.21E-05
16	3.11E-05	4.23E-05	6.62E-05
32	3.58E-05	5.10E-05	7.81E-05
64	3.56E-05	5.25E-05	7.80E-05
128	3.63E-05	5.42E-05	7.89E-05
256	3.91E-05	5.50E-05	8.54E-05
512	4.74E-05	6.46E-05	9.89E-05
1024	5.18E-05	7.26E-05	1.27E-04
2048	6.74E-05	9.42E-05	1.55E-04
4096	8.95E-05	1.36E-04	1.77E-04
8192	1.37E-04	2.20E-04	2.01E-04
16384	2.30E-04	3.94E-04	2.67E-04
32768	4.24E-04	7.40E-04	3.57E-04
65536	8.84E-04	4.22E-04	6.39E-04
131072	3.37E-04	4.81E-04	9.07E-04
262144	4.41E-04	6.04E-04	1.02E-03
524288	5.83E-04	7.37E-04	1.48E-03
1048576	8.27E-04	9.26E-04	1.67E-03
2097152	1.23E-03	1.31E-03	2.03E-03
4194304	2.02E-03	1.89E-03	2.74E-03

