ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

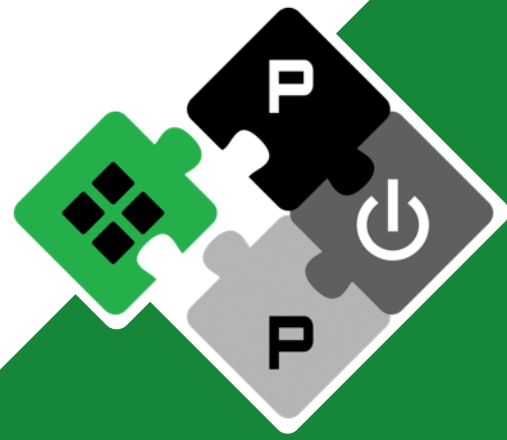
ICCD 2025
43rd IEEE International Conference on Computer Design

TeraNoC: A Multi-Channel 32-bit Fine-Grained, Hybrid **Mesh-Crossbar NoC** for Efficient **Scale-up of 1000+ Core** Shared-L1-Memory Clusters

Integrated Systems Laboratory (ETH Zürich)

Yichao Zhang	yiczhang@iis.ee.ethz.ch
Zexin Fu	zexifu@iis.ee.ethz.ch
Tim Fischer	fischeti@iis.ee.ethz.ch
Yinrong Li	yinrli@student.ethz.ch
Marco Bertuletti	mbertuletti@iis.ee.ethz.ch
Luca Benini	lbenini@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!



@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

Outline

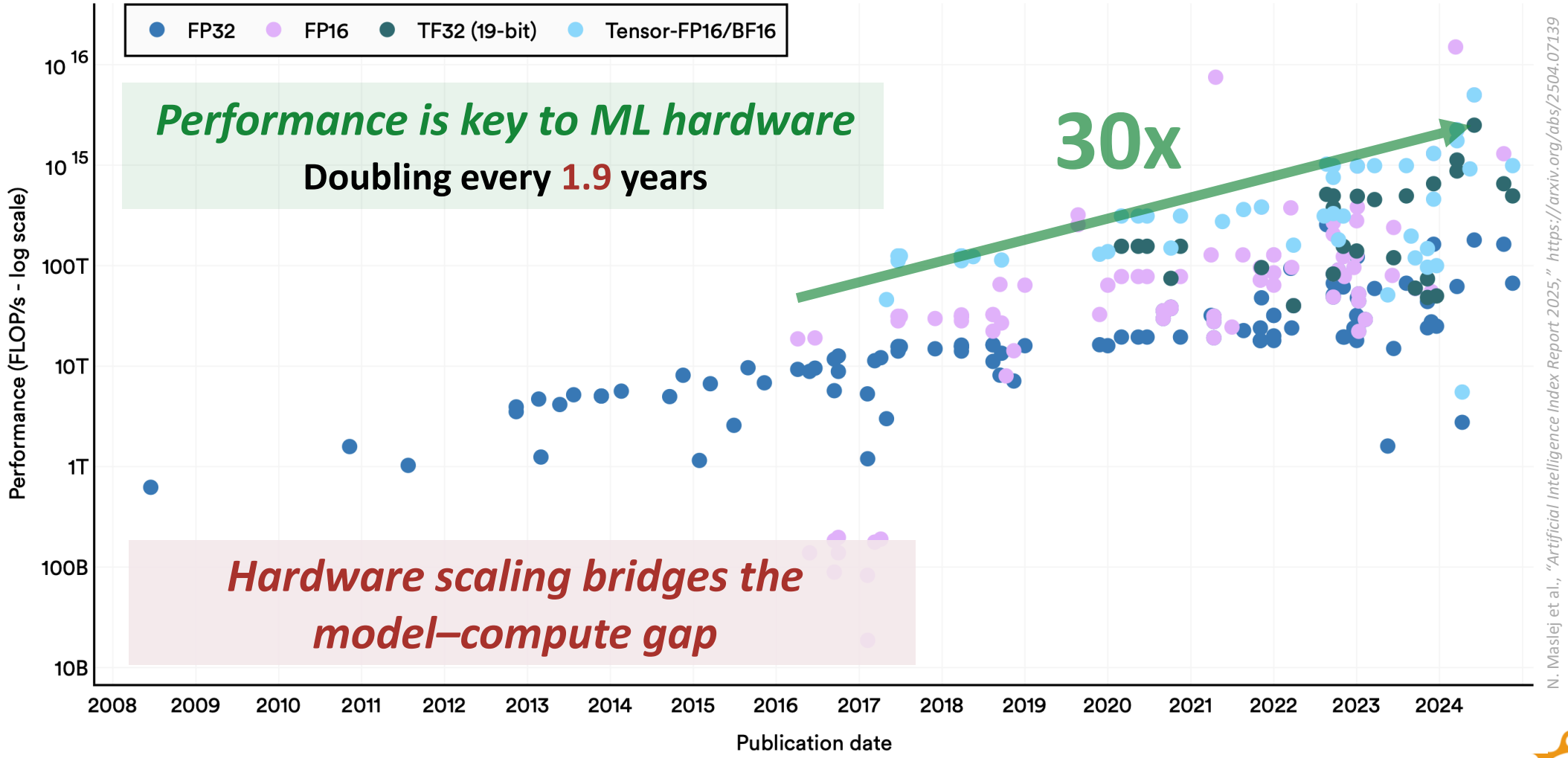
- ☐ Why do we need a core-to-L1 NoC?
- ☐ TeraNoC: Hybrid Xbar-Mesh Scaling-Up
- ☐ Scaling-Up 1000+ Cores Shared-L1 Memory
- ☐ Performance and Area Efficiency
- ☐ Conclusion

Outline

- **Why do we need a core-to-L1 NoC?**
- TeraNoC: Hybrid Xbar-Mesh Scaling-Up
- Scaling-Up 1000+ Cores Shared-L1 Memory
- Performance and Area Efficiency
- Conclusion

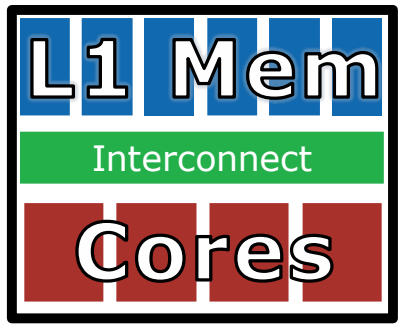
Hardware Scaling is key to AI Progress (cloud & edge)

Peak computational throughput of notable ML hardware (energy efficiency must track!)

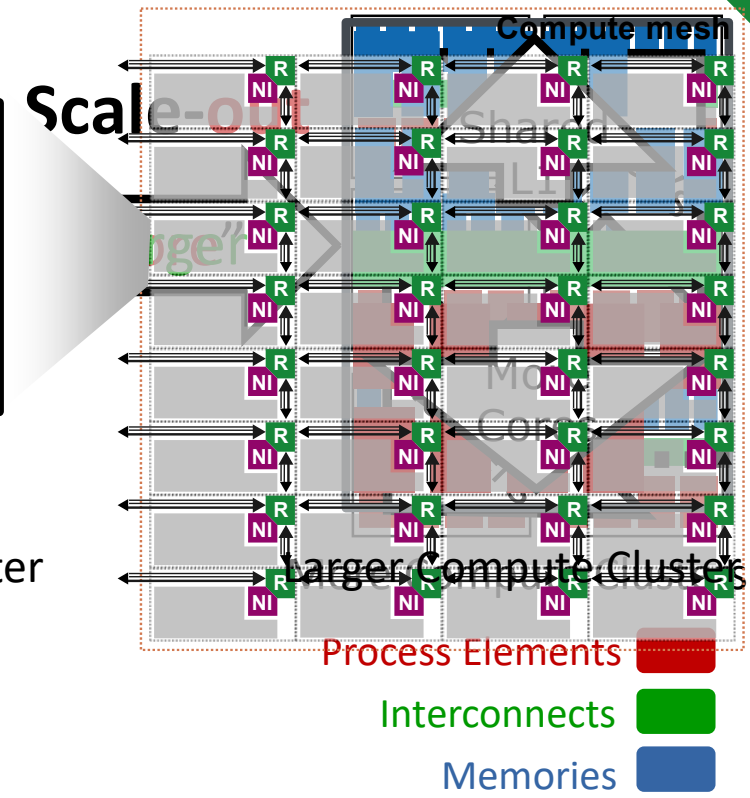


Hardware Scaling: OUT, or UP?

- **Scaling Out => More computing clusters**
- **Scaling Out by 2D-mesh NoC:**
  - ✓ Easy to scale (many small, identical processor clusters)
  - ✗ HW&SW overheads:
    - inter-cluster **communication** and **synchronization**
    - Data **allocation-splitting**, workload **distribution**
    - **High-latency** global interconnect to main memory, latency tolerant due to the limited size of L1



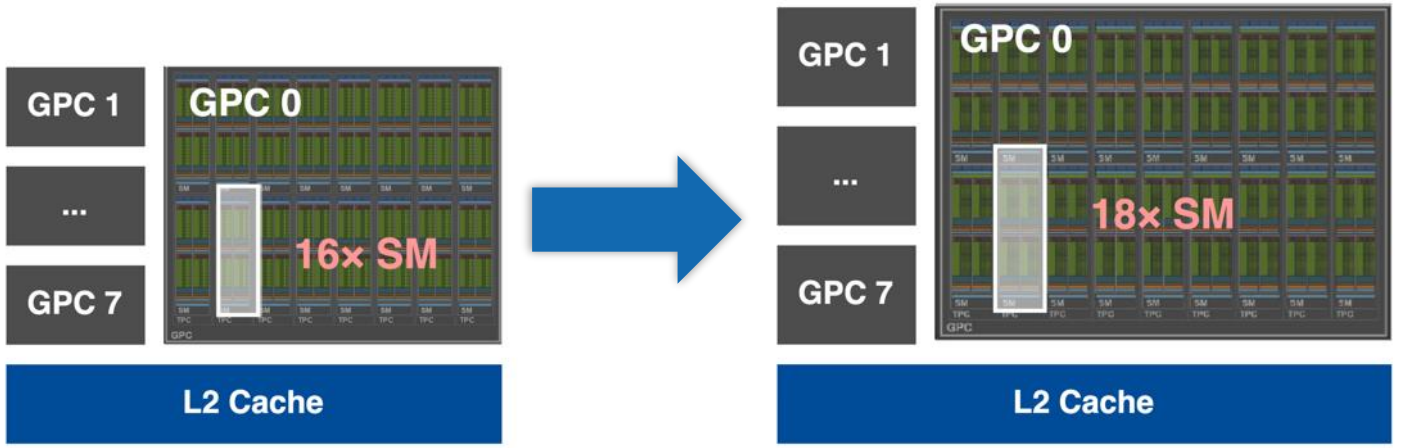
Small Compute Cluster



- **Scaling Up => Larger computing cluster**
  - Better arithmetic intensity (AI), more data reuse
  - Better latency tolerance. Little's Law: latency+(problem/bw) < (problem*AI)/compute
  - Less synchronization overhead; reduce data split-merge; easy to program … …

Scaling Example: From NVIDIA Ampere to Blackwell

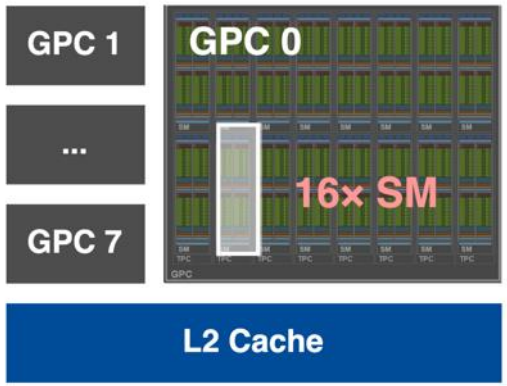
GPU *Scale-Out:*

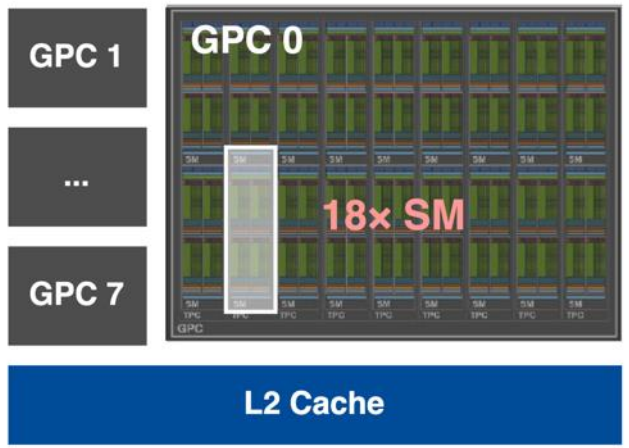


Ampere (NV A100)	**Hopper** (NV GH100)	**Blackwell** (NV GB200)
108 SMs per GPU	**132 SMs** per GPU	**144 SMs** per GPU

Scaling Example: From NVIDIA Ampere to Blackwell

GPU *Scale-up:*



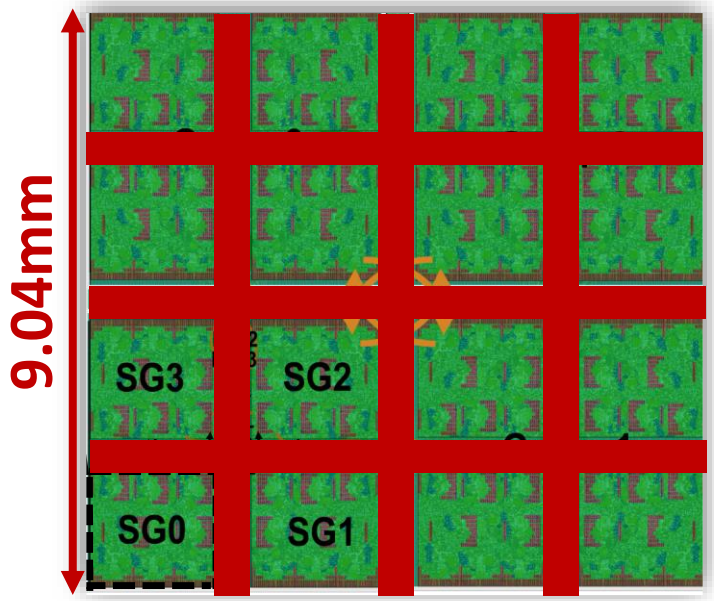


Ampere (NV A100)	**Hopper** (NV GH100)	**Blackwell** (NV GB200)
108 SMs per GPU	**132 SMs** per GPU	**144 SMs** per GPU
64 shading units, **4x 3th gen tensor cores (312 TOPS*),** **192 KiB L1 cache** per SM	**128 shading units,** **4x 4th gen tensor cores (624 TOPS*),** **256 KiB L1 cache** per SM	**128 shading units,** **4x 5th gen tensor cores (1248 TOPS*)** per SM

**: with FP16 precision*

Scaling-UP Makes L1 Interco More Critical

- **TeraPool: The largest scaling-up cluster design in PULP**
  - 1024 programmable FP RV-cores
  - Fully shared 4096 1KiB L1 SPM banks
  - Core-to-L1 latency < 10 cycles
- **2D-Mesh NoC:**
  - Regular wiring, more area-efficient, better scaling
  - Can we use it for **Core-to-L1-Bank interco**?



TeraPool (GF12nm): 1024 Cores shared 4MiB L1

Hier-Xbar Interconnect: **40.7%** of cluster area due to routing channels

Area-Efficient, Easy Routing

Internal L1 Latency Increase

Bandwidth Bottlenecks

Let's solve them by TeraNoC

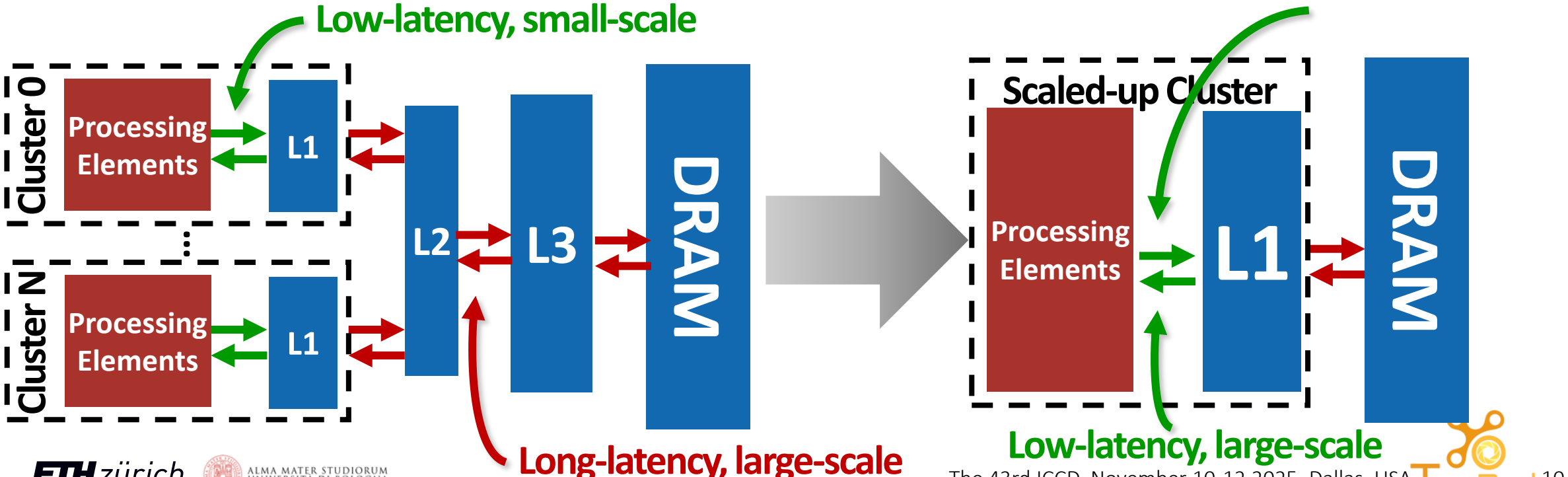
Outline

- ☐ Why do we need a core-to-L1 NoC?
- ☐ **TeraNoC: Hybrid Xbar-Mesh Scaling-Up**
- ☐ Scaling-Up 1000+ Cores Shared-L1 Memory
- ☐ Performance and Area Efficiency
- ☐ Conclusion

TeraNoC Design Idea

- **Low-Latency Core-to-L1-Bank** => **Fully Connected Crossbar**
- **Area-Efficient Scaling** => **Word-Width (32b) 2D-Mesh NoC**
- **Large scale (1000+ cores)** => **Hierarchical Design**
- **Boost NoC Bandwidth** => **Multiple Parallel Channels**

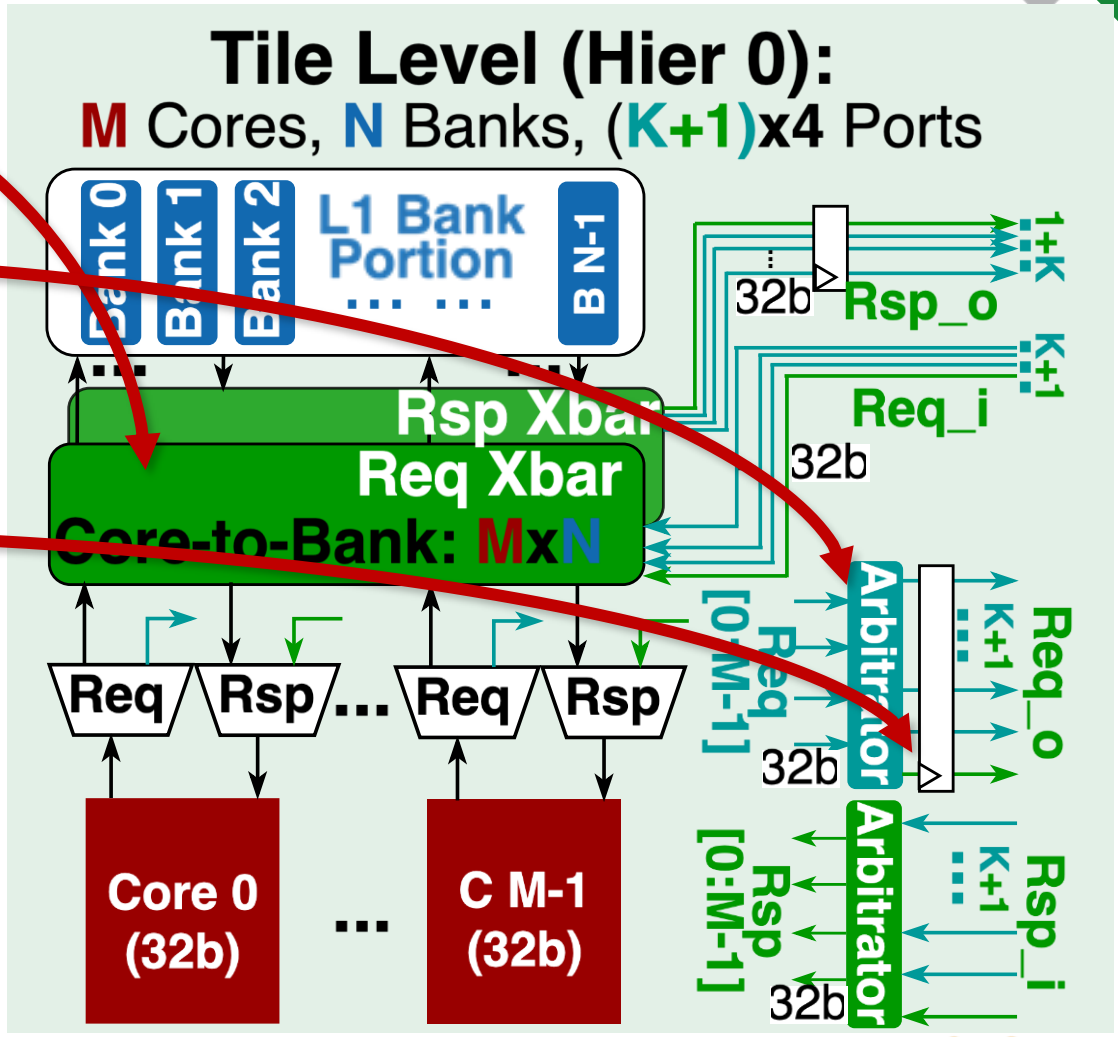
TeraNoC



Hierarchy 0 (Tile): Close to Core, Low-Latency Crossbar

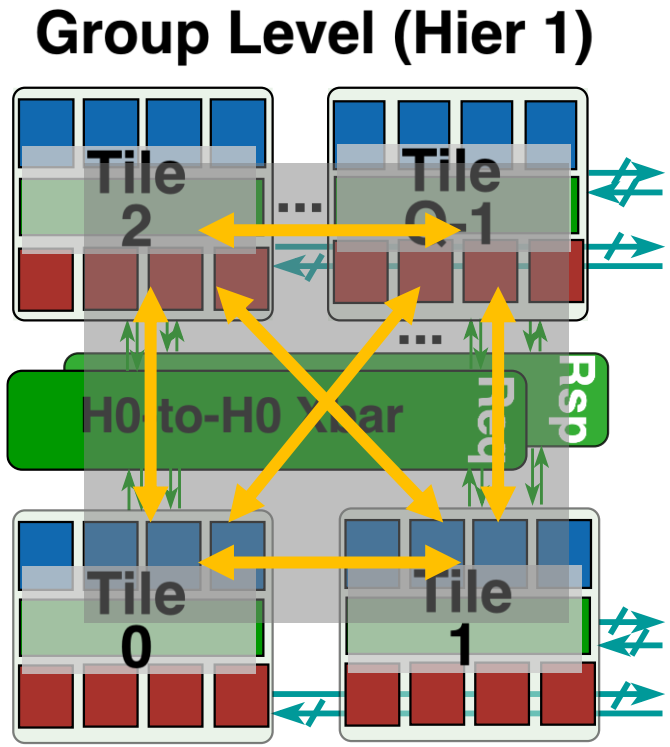
- **Cores to the local bank portion**
  - Fully connected crossbar, 1 cycle latency
- **Interface for remote hierarchies**
  - K routers for far connection
  - 1 to next-level crossbar
- **Spill Reg only on the interface**
  - Keep low-latency
  - Cut long physical connections

Close to core:
Crossbar to keep low latency



Hierarchy 1 (Group): Crossbar + Multi-Parallel Routers

- **Tile-to-Tile: Crossbar**
  - Physically close to each other
  - Keep low latency (3 cycles)
- **To Other Groups**
  - To routers
  - = Q Tiles x **K Routers per Tile**
  - Router Remapper balances network utilization
- **From Other Groups**
  - Xbar connects Routers to Tiles
  - QxK routers to Q Tiles



Crossbars + Multi-Parallel Routers
=
Low-Latency + High BW Scaling

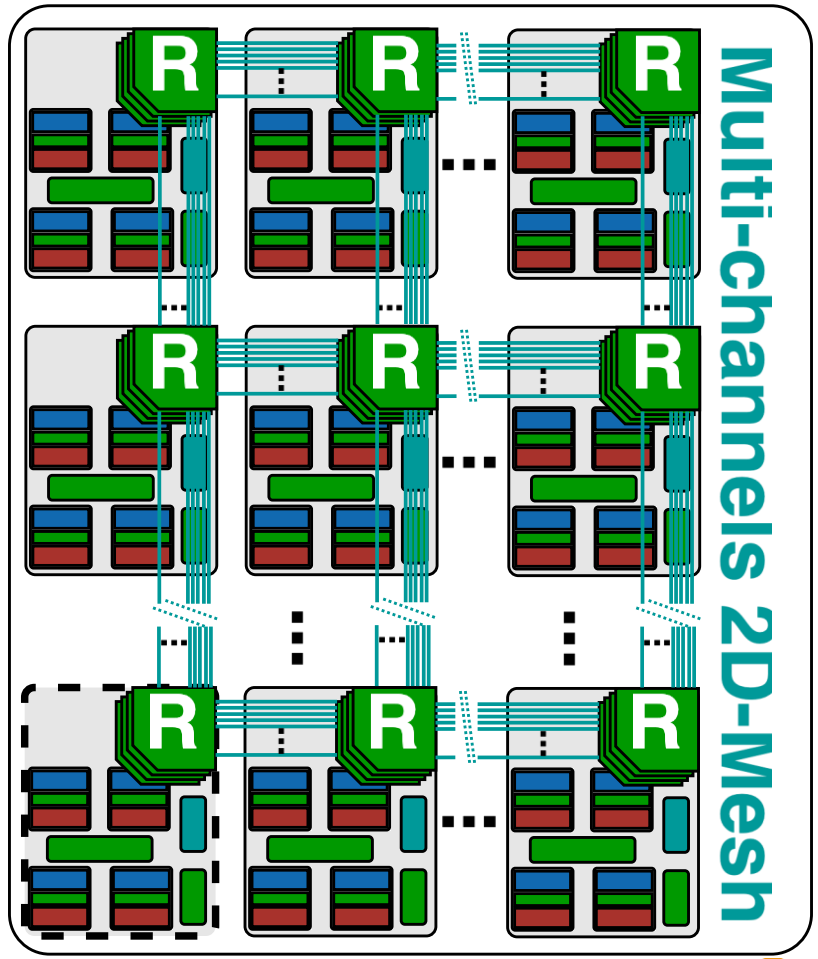
Higher Hierarchy (Cluster): 2D-Mesh NoC

- **2D-mesh NoC scaling up**
  - Regular wiring, low routing complexity
- **Asymmetric Router, multi-channel, fine-grained**
  - [Tile -> **Router** -> Target Group] -> Xbar -> Tile
  - Hardware choice of “No. Routers/Tile”, depends on silicon resources
  - 32-bit multi-channel, router-remapper balance utilization.

At Top Hierarchy:

2D-Mesh → Area-Efficient Scaling

Shared-L1-Mem Cluster (Scaling Up)



Outline

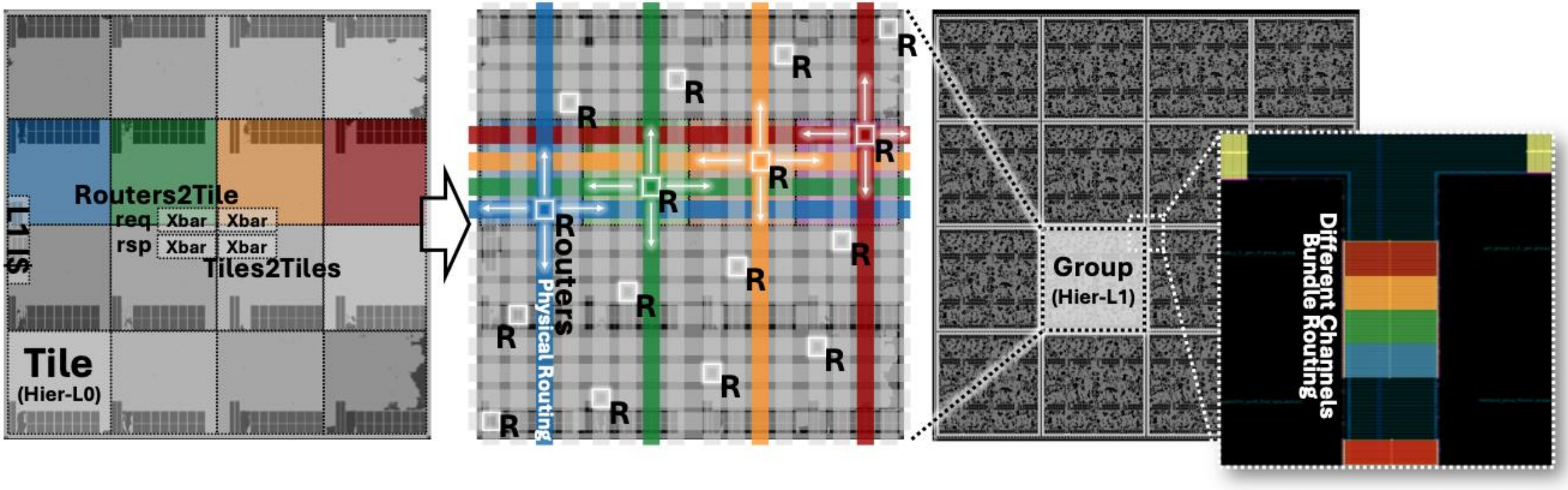
- ☐ Why do we need a core-to-L1 NoC?
- ☐ TeraNoC: Hybrid Xbar-Mesh Scaling-Up
- ☐ **Scaling-Up 1000+ Cores Shared-L1 Memory**
- ☐ **Performance and Area Efficiency**
- ☐ Conclusion

By TeraNoC: 1024 RV Cores Fully Shared 4MiB L1-Memory

- **1024 Cores share 4096 L1 Banks in ONE cluster built by TeraNoC:**
  - (4 cores+16 L1 banks)/Tile (Hier 0), crossbar-connected, 1-cycle latency
  - 2 routers/Tile (32 routers/Group), crossbar-connected, 3-cycle latency
  - 16 Groups/Cluster, 2D-Mesh NoC, 2-cycles latency/hop

Multi-Parallel Routers → Logic-Physical Co-Design

- **Stagger placed routers for straight connections**
- **Interleaved placed router ports at Group boundaries**



L1 Interco: TeraNoC vs. Hierarchical Crossbar

Fareast Group, 7 Hops + Xbars

Latency tolerable by PEs

Core-to-L1 Interco.	TeraNoC	3 Stages Hier-Xbar	16x16 2D Tile Mesh
Cluster Scale	1024 Cores, 4096 L1 Banks		4 Cores, 4096 L1 Banks
Longest Latency	31 Cycles	11 cycles	127 Cycles
Average Latency	**13.7 Cycles**	9.3 Cycles	45.7 Cycles
Peak Bandwidth	4 KiB/Cycle	4KiB /Cycle	4 KiB/Cycle
Bi-Section Bandwidth	**2 KiB/Cycle (BW Group)** **4 KiB/Cycle (In-Tile)**	1.875 KiB/Cycle	1 KiB/Cycle *(Non-Parallel Channels)*
Physical Frequency (GF12, TT@0.8V)	**936MHz**		N.A.

1536 unidirectional data response channels

+10% than Hier-Xbar; not the critical path

Around 10+ cycles latency can be **tolerated by the core**:

- The **transaction table** retires loads out of order (Up to **16 entries**)
- The scoreboard ensures in-order delivery to the execution units

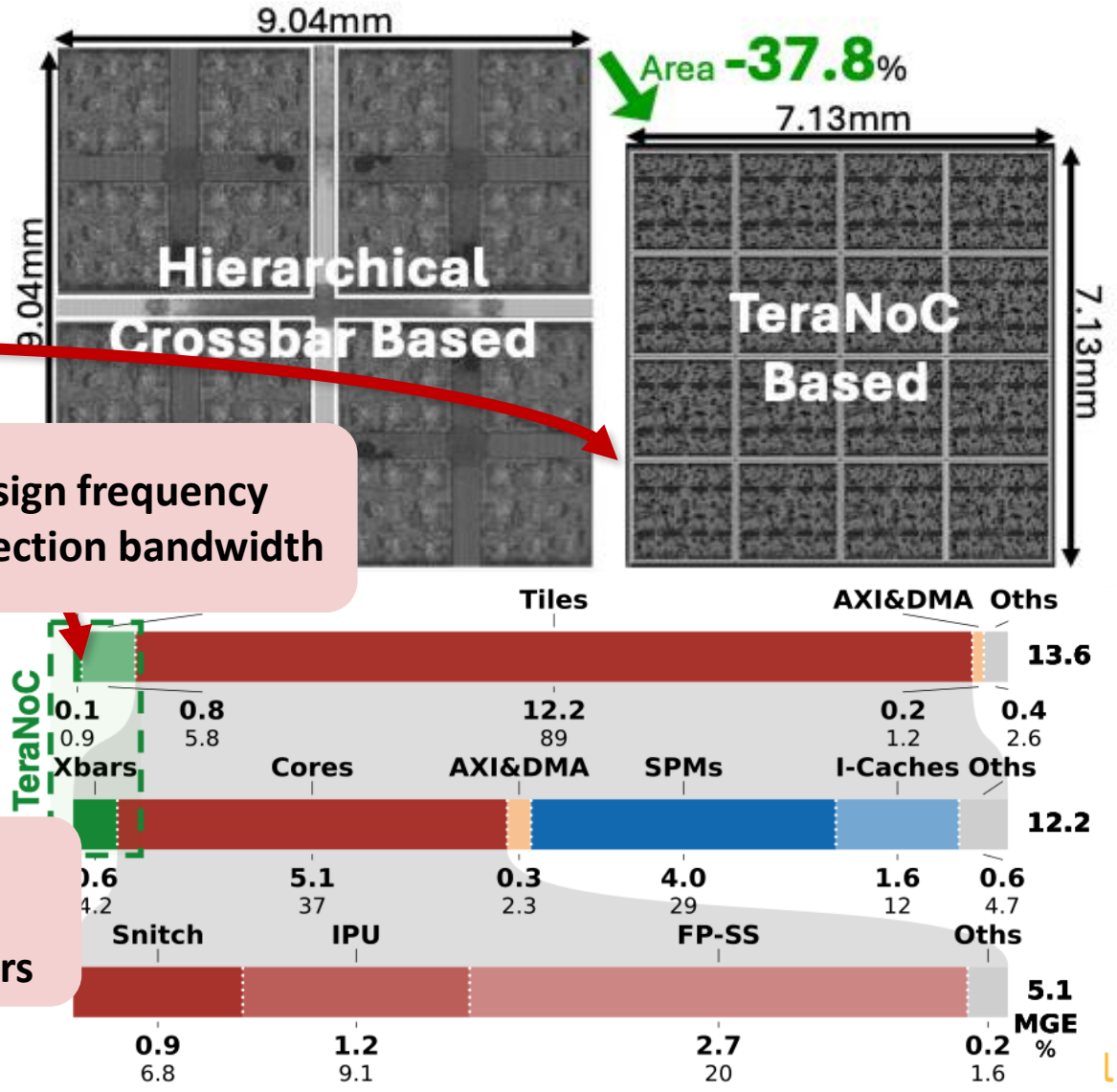
Area Efficient Design

- **Compare with TeraPool (Hier-Xbar-based) cluster:**
  - Same architectural scale, implemented in GF12nm
  - Reduce physical area by **37.8%**
  - Only **10.9%** of the total logic a
  - Computer throughput +**19%**;

    Area efficiency +**98.7%**

    *(GFLOP/s/mm², benchmarked by **MatMul_f16**)*

Improved design frequency & better bi-section bandwidth



512x512x512 MatMul:
- Row/column data interleaved through all banks
- Extremely global access-dominated kernel through routers

Related Works and SoA

Related Work	Low-latency for in-cluster	open-source	#Links & width	Target Design BW	Target Design Latency (avg)
Tenstorrent [1]	✗	✗	2048	20 B/Cycle	N.A.
HammerBlade[2]	✗	✗		0.5 KiB/Cycle	4 + 2 x hops
Celerity [3]	✗	✓			
Ou et al. [4]	✗	✓	256	32 B/Cycle	N.A.
TeraPool[5]	✓	✓			3 Cycles
This work	✓	✓			

2D-Mesh/Ruche, good scaling but long latency

Open-sourced 2D-mesh for scaling out

Low-Latency Xbar for scaling-up but area cost

The first open-sourced, area-efficient, word-width mesh-NoC for core-to-L1 low-latency interconnect

SoA References

[1] J. Vasiljevic and Davor Capalija, “**The Standalone AI Computer and its Programming Model**,” in 2024 IEEE Hot Chips 36 Symposium (HCS), 2024.

[2] D. C. Jung, M. Ruttenberg, P. Gao, S. Davidson, D. Petrisko, and M. B. et al., Taylor, “Scalable, Programmable and Dense: The HammerBlade Open-Source RISC-V Manycore,” in ISCA. Buenos Aires, Argentina: ACM/IEEE, Aug. 2024, pp. 770–784

[3] A. Rovinski, C. Zhao et al., “**Evaluating celerity: A 16-nm 695 giga-risc-v instructions/s manycore processor with synthesizable pll**,” *IEEE Solid-State Circuits Letters*, vol. 2, no. 12, pp. 289–292, 2019.

[4] Y. Ou, S. Agwa, and C. Batten, “**Implementing low-diameter on-chip networks for manycore processors using a tiled physical design methodology**,” in 2020 *14th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, 2020, pp. 1–8.

[5] Y. Zhang et al., "**TeraPool: A Physical Design Aware, 1024 RISC-V Cores Shared-L1-Memory Scaled-Up Cluster Design With High Bandwidth Main Memory Link**," in IEEE Transactions on Computers, vol. 74, no. 11, pp. 3667-3681, Nov. 2025.

Outline

- ☐ What is “TeraNoC”?
- ☐ Why “TeraNoC”?
- ☐ Hybrid Xbar-Mesh Design
- ☐ Implement 1000+ Cores Shared-L1 Memory
- ☐ Performance and Area Efficiency
- ☐ **Conclusion**

Conclusion

- **TeraNoC is an area-efficient solution for core-to-L1 interconnect scaling up**
- **Hybrid mesh-crossbar solution**
  - 32b narrow fine-grained, parallel router-based 2D-mesh NoC + Crossbar
  - Router-remapper balance traffic load
- **Build 1024 cores sharing 4096 L1 SPM banks:**
  - 3.74 TiB/s peak bandwidth, 0.47TiB/s bisection bandwidth
  - Only 10.9% of the total logic area
  - Compare with the Hier-Xbar-based solution:
    - The total cluster die area **-37.8%**
    - Computing throughput **+19%** **(by MatMul, an extreme global L1 banks accessing kernel)**
    - Area efficiency (GFLOP/s/mm^2) **+ 98.7%**

PULP Platform

Open Source Hardware, the way it should be!

Yichao Zhang **yiczhang**@iis.ee.ethz.ch

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Universitá di Bologna
Viale del Risorgimento 2
Bologna, Italy

@pulp_platform

pulp-platform.org

youtube.com/pulp_platform