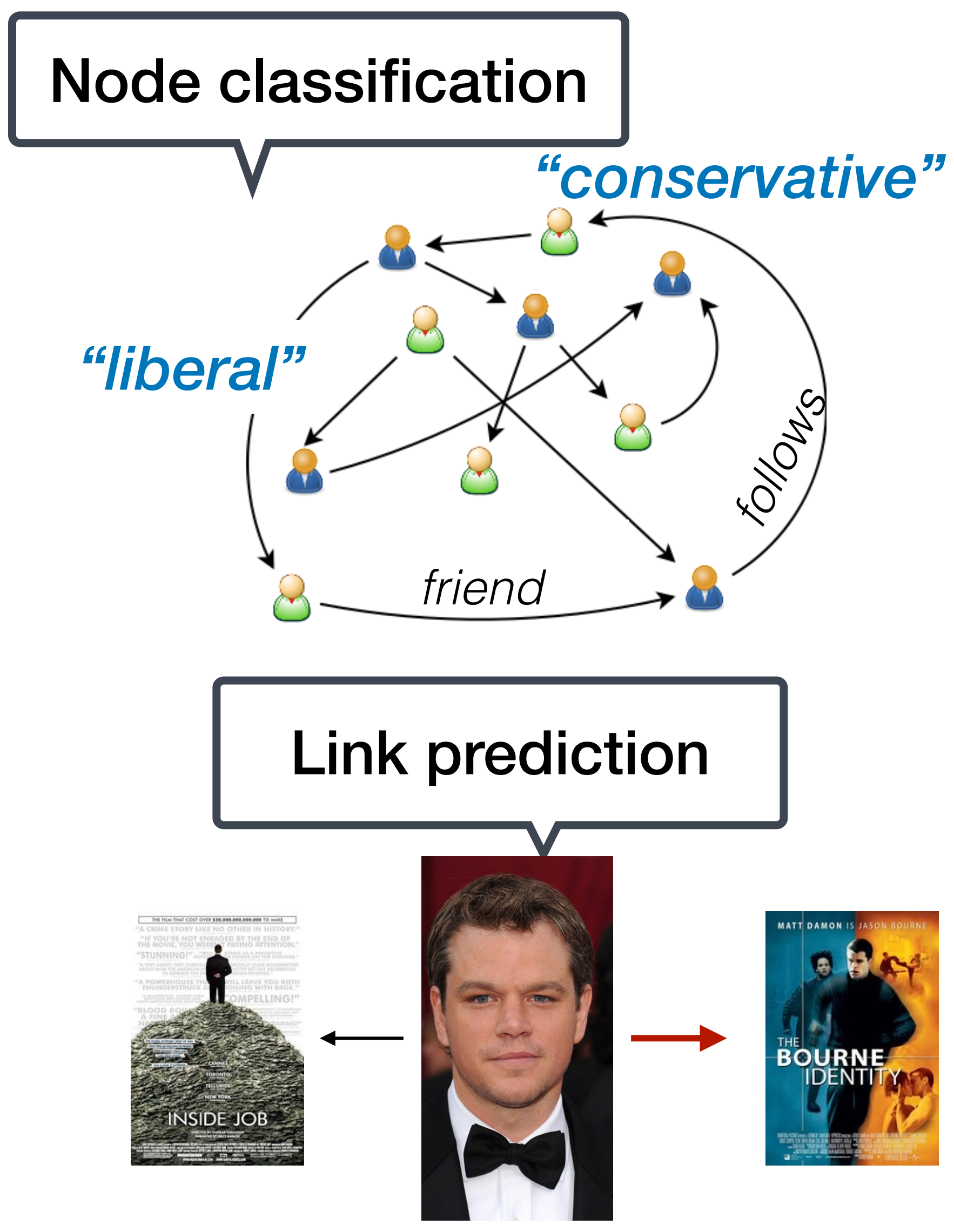
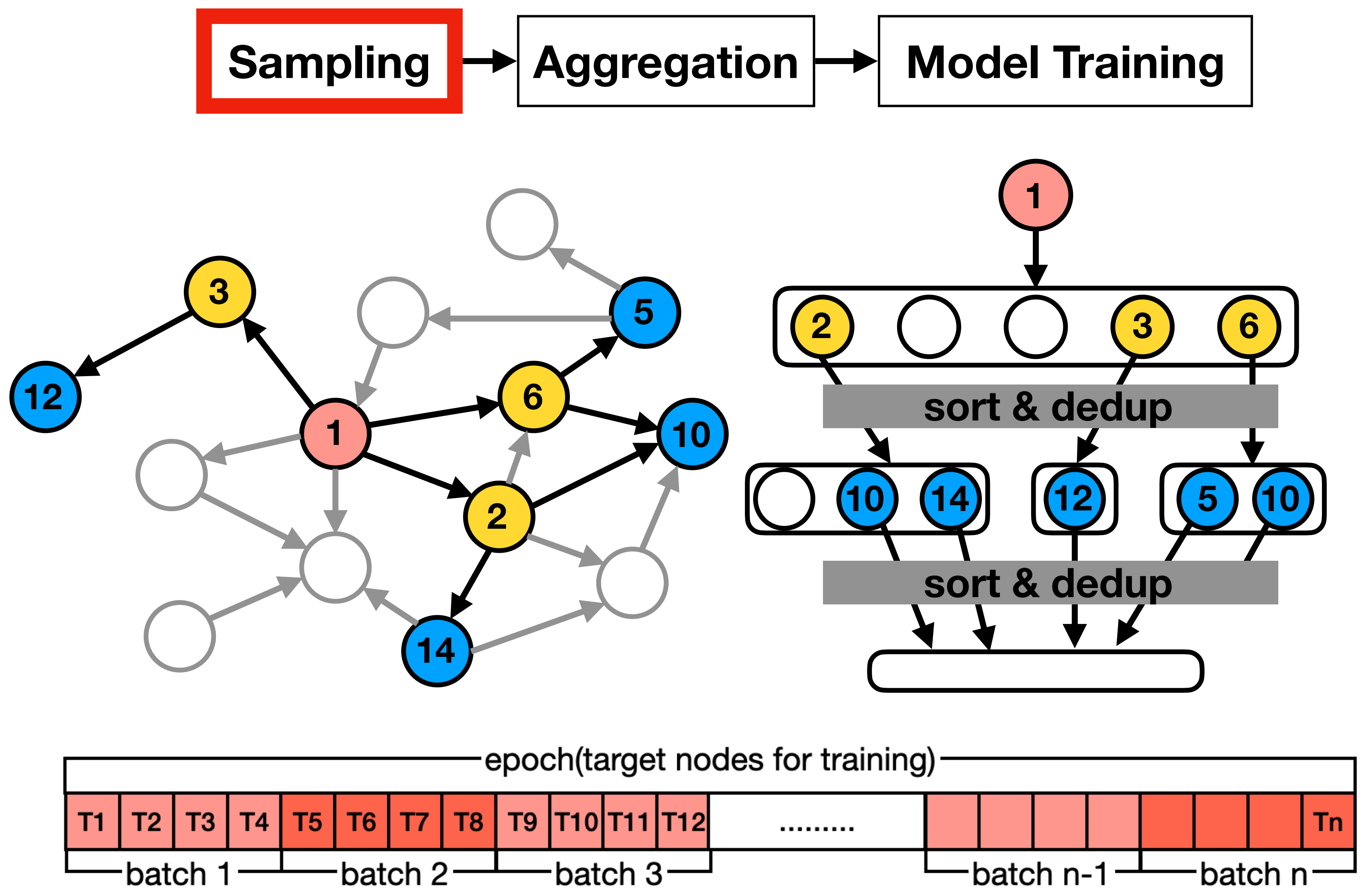
Scaling GNN Sampling on Large-Scale Graphs with io_uring

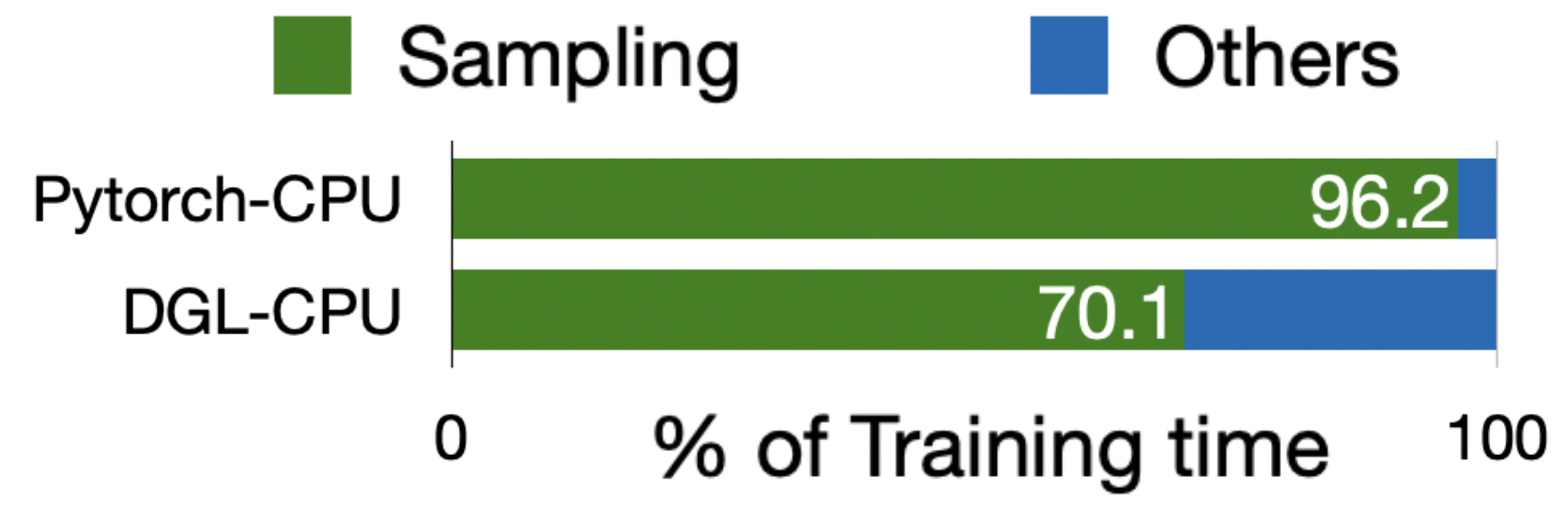
Qixuan Chen taliac@bu.edu · Yuhang Song yuhangs@bu.edu · Melissa Martinez melimtz@bu.edu · Vasiliki Kalavri vkalavri@bu.edu

BOSTON UNIVERSITY · CASP Systems

Graph Representation Learning and GNN Sampling







Datasets	#Nodes	#Edges	Size
Papers100M	111M	1.62B	70GB
Hyperlink	3.5B	128B	3.4TB
Facebook	1.4B	1T	8.5TB

Graph size may exceed main memory

Current Approaches and Limitations

CPU-based

(MariusGNN, Ginex)

- Unnecessary I/O from loading full neighborhoods into memory
- Lower computation power than GPU
- High data movement overhead between memory and SSD

GPU-based

(NextDoor, gSampler)

- Constrained GPU memory
- Sampling competes with other tasks for GPU resources
- High computation cost

SSD-based

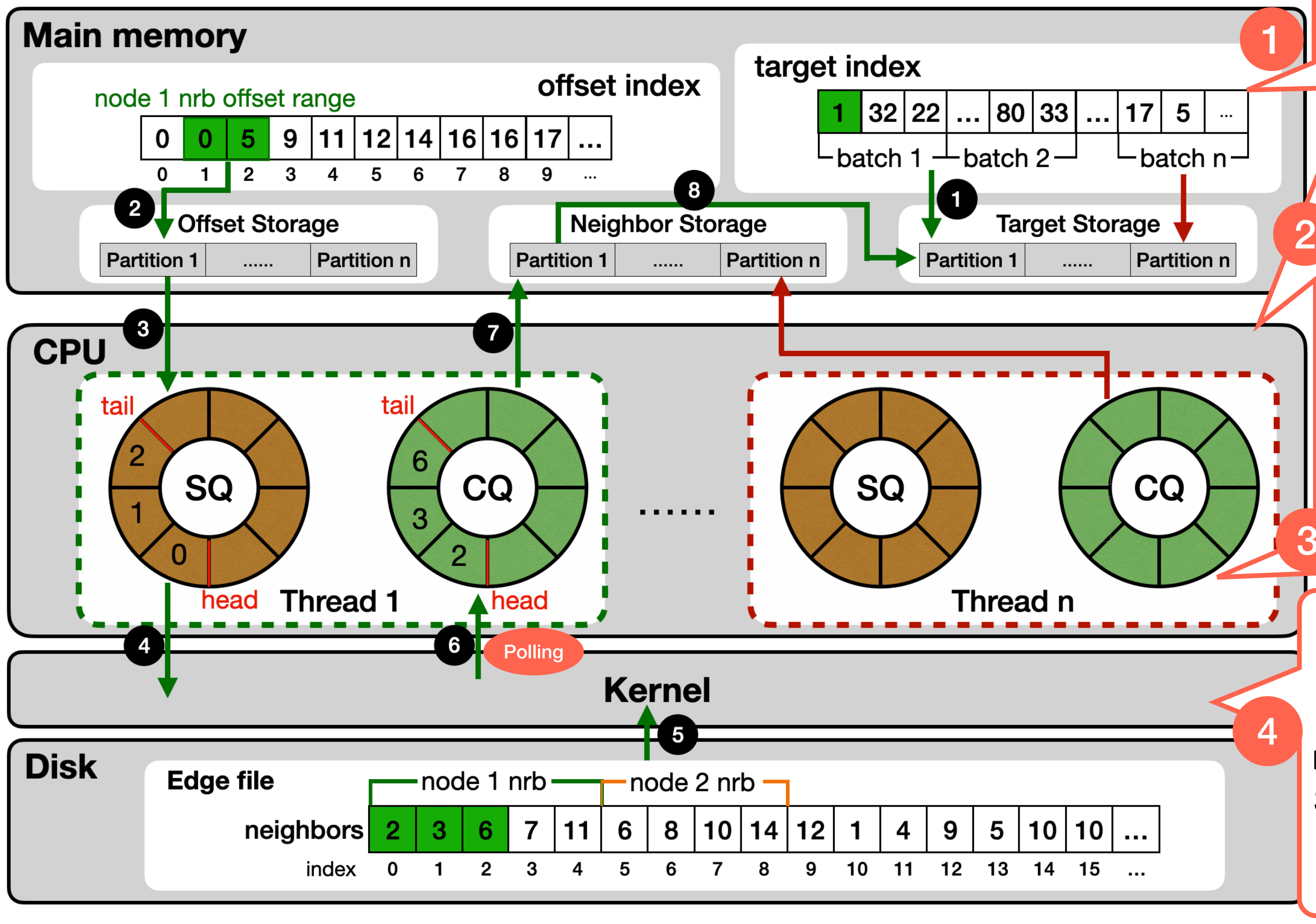
(In-situ SmartSSD, FlashGNN)

- Difficult to adopt widely across different hardware configurations
- Limited bandwidth compared to main memory

My proposal: Leverage modern storage APIs and high-bandwidth SSDs to perform sampling on larger-than-memory graphs

CPU-Based Sampling System Accelerated by io_uring

1. **How to** minimize data movement between SSD and CPU?
2. **How to** integrate io_uring to GNN Sampling?
3. **How to** implement multi-threading to maximize CPU utilization?
4. **How to** take the advantage of asynchronous I/O?



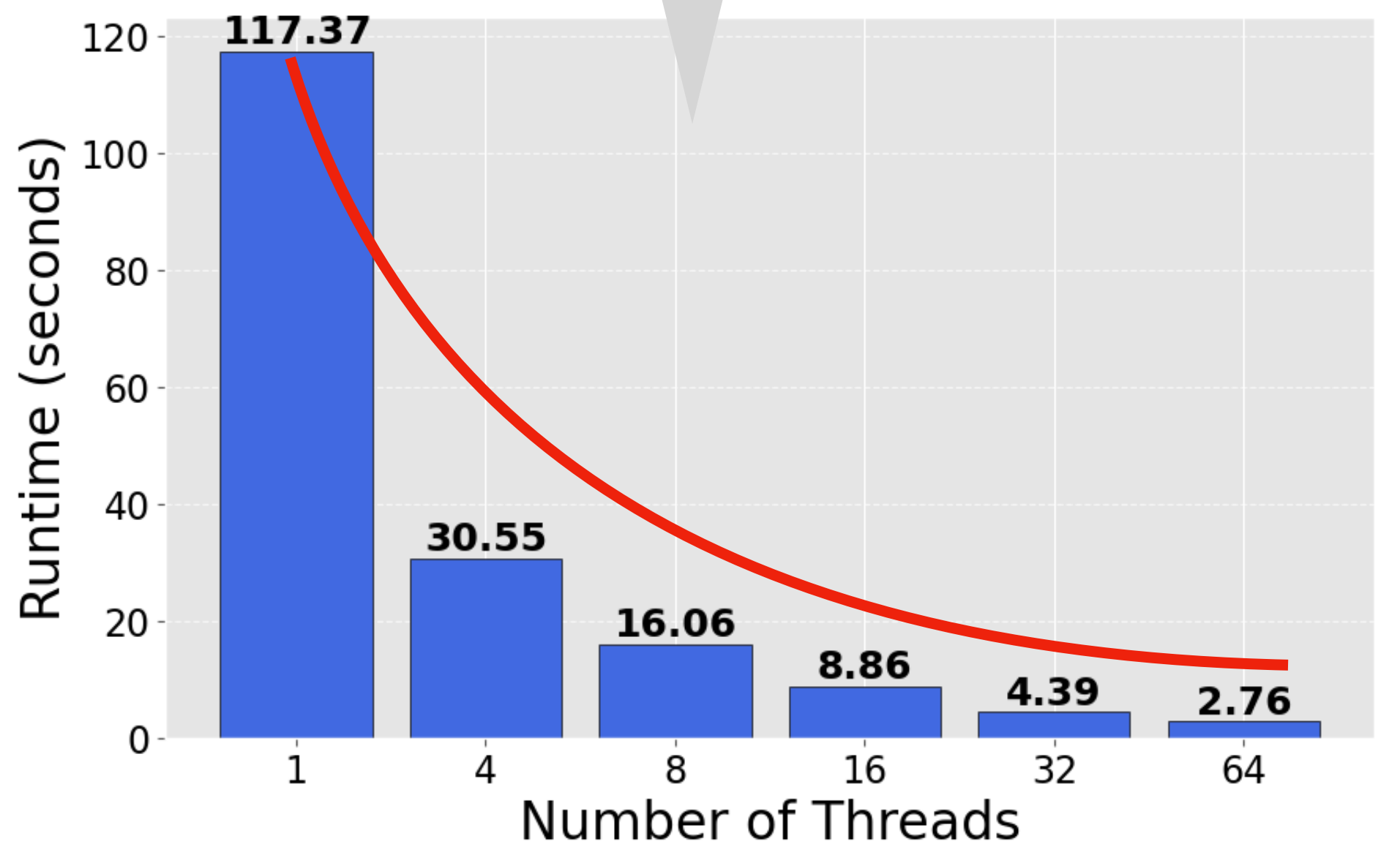
1. Create two indexes to efficiently select neighbor offsets, allowing direct access to sampled neighbors without loading all neighbors from disk

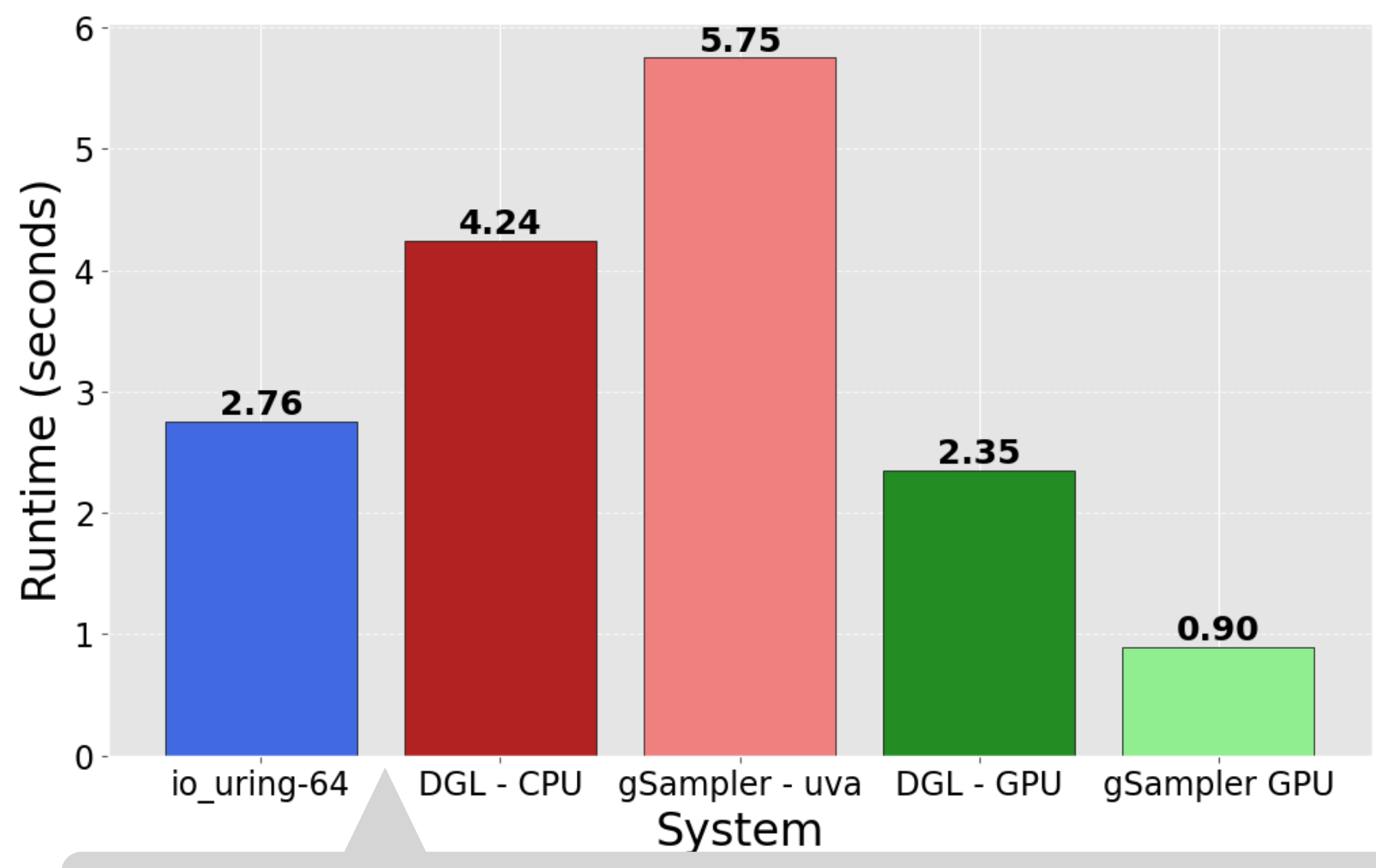
2. Batch I/O requests to read neighbors

3. Equally distribute batches to threads. Each thread works in parallel with their own io_uring buffers without interrupting each other

4. While CQ is polling completions, we prepare and load the next I/O batch into the SQ, submitting it once the previous batch is done

performance scales almost linearly with thread count





Sampling billion-edge graph

- 1.5x faster than baseline
- Comparable with GPU-based
- Can process larger-than-memory graphs