

Fast graph processing

→ vertex-centric approach

Guess what: edge, subgraph-centric

Python → C/C++ and parallel

Faster: GPUs

Fasterer: distributed

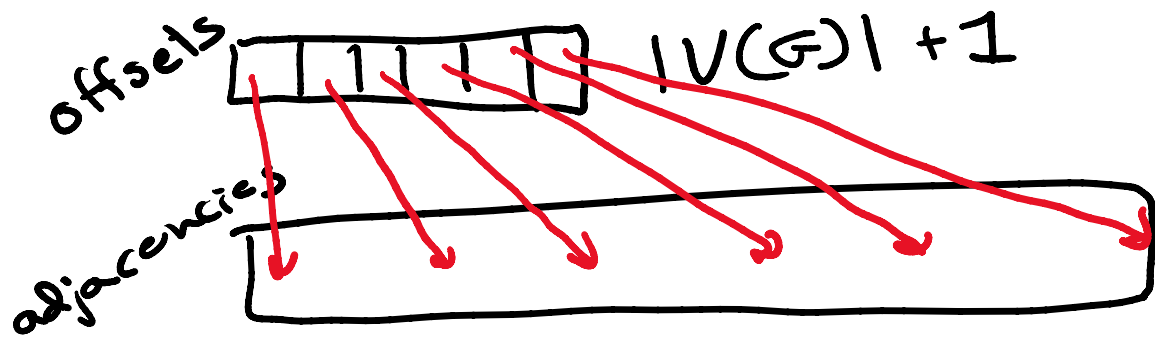
However: implementation complexity
blows up

Issues: race condition, workload
distribution, communication overhead,

Solutions: multi-level queues,
asynchronous, etc.

Storage of graphs

CSR - compressed sparse row



used for graphs/matrices
(CSC)

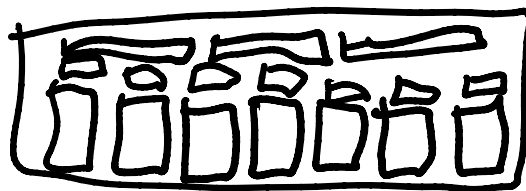
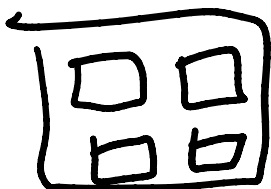
CPU parallelism with OpenMP

→ pragmas

We can modify our serial BFS
with OpenMP

→ queues per thread

GPU parallelism



For v in Q in parallel

For v in U in parallel

For w in $N(v)$

→ loop collapse

For e in Q

Distributed-memory

