BFS over GPU cluster Using MPI and CUDA

By Balaji Shashipreeth, Racherla

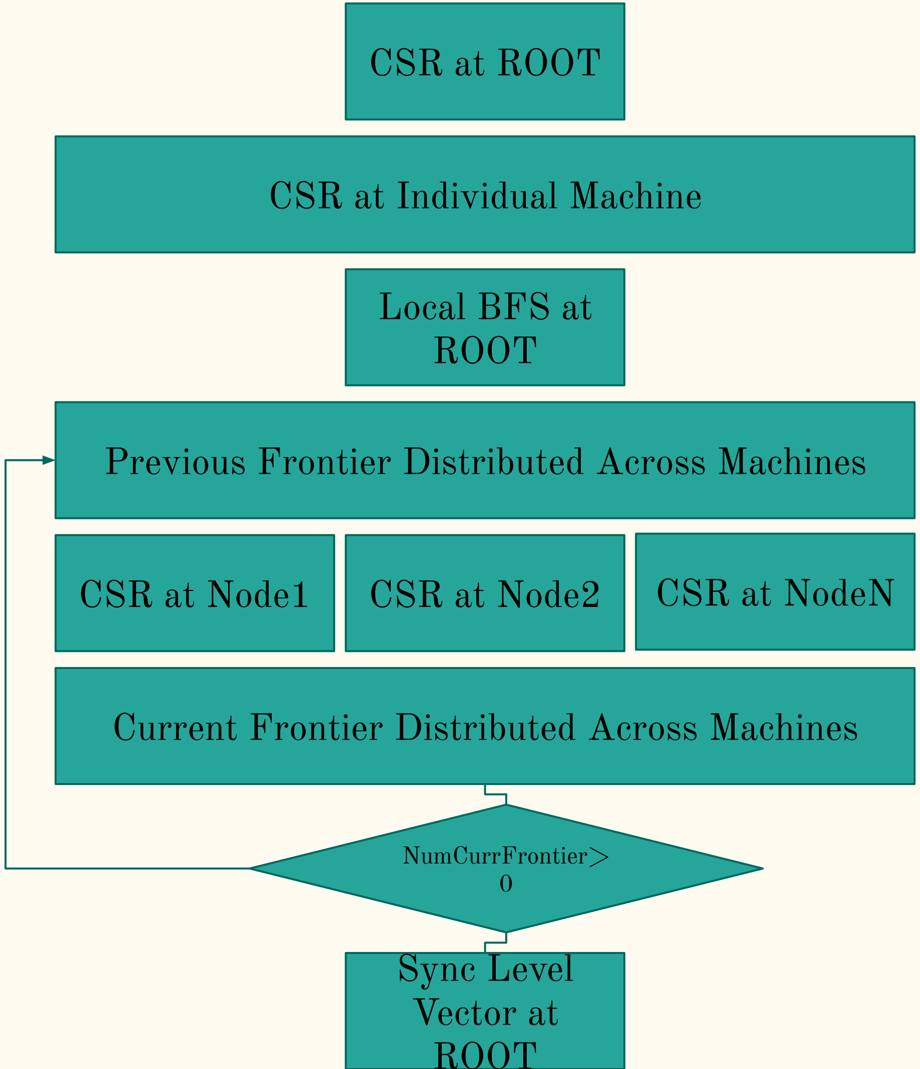
Main Idea

BFS Over a large Graph

- Data from SNAP
- Convert Edge Information to CSR
- CSR to be transferred to all machines
- Work Split Across machines
- Output is a level vector containing distance of shortest path from vertex given

Pseudocode

1. Load CSR into Memory
2. Distribute CSR to all Nodes
3. Create level, previous Frontier, current Frontier Vectors
4. Perform a Few Levels of BFS at ROOT till we reach a good number of current Frontiers
5. Copy current Frontier, level Vectors to all nodes
6. Distribute work to all nodes and each node works and generate current Frontier
7. Sync the previous Frontier across all nodes using MPI_ALLGatherv, use the gathered information to update level information locally
8. If number of previous frontier is greater than zero go Step 6
9. Sync the level vector at ROOT and output the vector


CSR at ROOT
CSR at Individual Machine
Local BFS at ROOT
Previous Frontier Distributed Across Machines
CSR at Node1
CSR at Node2
CSR at NodeN
Current Frontier Distributed Across Machines
NumCurrFrontier> 0
Sync Level Vector at ROOT

Specifics Implementations

- Level persisted in global memory (in GPU) to avoid copying
- We sync this level using the nodes gathered from other machines using a kernel inside GPU
- CSR is read as a CSV where 0th row with row consisting row pointers and 1st row consisting of column Indices
- The program can be guided to work on other graphs, as well as vertex specified in input given that they are in the format expected

Metrics Collected

Graph	No of Graph Nodes	No of Graph Edges	Serial Execution Time (ms)	CUDA Execution Time (ms)	MPI+CUDA Execution Time (ms) (2 procs locally)	Max Level Discovered
LiveJournal (A social Network)	4847571	68993773	917	257	647 (>5000 on network)	14
roadNet_CA (a network of roads in state of California)	1971281	5533214	107	616	1715	545

Understanding the metrics collected

- The MPI+CUDA program seems to improve in speed as the number of nodes increase
- The graph structure seems the have a effect such as roadNet where most of the nodes are interconnected, but have a fewer neighbours and result in a deeper level while social networks have shallow levels

Future Work

- Overlap Computation and Communication using asynchronous communication such as MPI_ISend, MPI_IRecv
- Use CUDA aware MPI to reduce host to device transfers
- May be one-sided communication to steal work asynchronously to balance the load