Dynamic Massive Parallel Computation Model for Graph Problems

Isabelle Liu
School of Computer Science
Carleton University, Ottawa, Canada
<isabelleliu@cmail.carleton.ca>

Introduction

- What is Massive Parallel Computing and why MPC
  - Computations on massive amounts of data
  - Traditional models are inadequate
  - MPC model captures the needs of computation at a massive scale
  - A collection of machines
    - communicate through indirect communication channels
  - Computation proceeds in synchronous rounds
- Limitations of MPC
  - Assumes to work on static datasets
  - Use of large volumes of resources

Introduction

- Dynamic algorithm and its benefits
  - Maintain a solution to a given problem throughout a sequence of modifications to the input data
  - Efficiently adjust the maintained solution
  - Detect whether modifications are needed almost instantly
  - Polynomial, and often exponential, speed-up
- Objective
  - Implement a Dynamic MPC (DMPC) algorithm
  - Test on a graph problem, Maximal Independent Set

Related Work

- Classic MPC:
  - $(1+\epsilon)$-approximate matching: $\tilde{O}(\sqrt{\log \Delta})$
  - Connected components: $\tilde{O}(\log D)$
- DMPC:
  - Maximal matching: $O(1)$
  - Connected components: $O(1)$
- Maximal Independent Set
  - Classic MPC: $\tilde{O}(\log \log \Delta)$
  - Dynamic: $O(\log^2 \Delta \cdot \log^2 n)$

Methodology

- MPC Model
  - A set of μ machines $M_1, \ldots, M_\mu$
  - Memory that fits up to S bits per machine
  - Exchange messages in synchronous rounds
  - Send and receive messages of total size up to S per machine per round
  - Input, of size N, stored across the different machines in arbitrary way
  - Output at most S bits per machine

Methodology

- Dynamic graph algorithm
  - Incremental: allow edge insertions only
  - Decremental: deletions only
  - Fully-dynamic: intermixed sequence of both
- Factors that determine the complexity of the dynamic algorithm
  - Number of rounds
  - Number of active machines per round
  - Total amount of communication per round

Methodology

- Terminology of lexicographically first maximal independent set (LFMIS)
  - Graph $G(V, E)$
    - $n = |V|, m = |E|$
    - Ranking $\pi: V \to [0, 1]$
    - Add an alive vertex v to the MIS
    - Kill v and all of its alive neighbors
  - $LFMIS(G, \pi)$: subset of vertices that join the MIS
  - $\deg(v)$: degree of vertex v
  - $count(v)$: number of neighbors of vertex v that are in the MIS
  - $N^+(v)$: set of neighbors of v in the MIS
  - $N^-(v)$: set of neighbors of v not in the MIS
  - Assign ID $(1, \dots, n)$ to each vertex based on its ranking π

Methodology

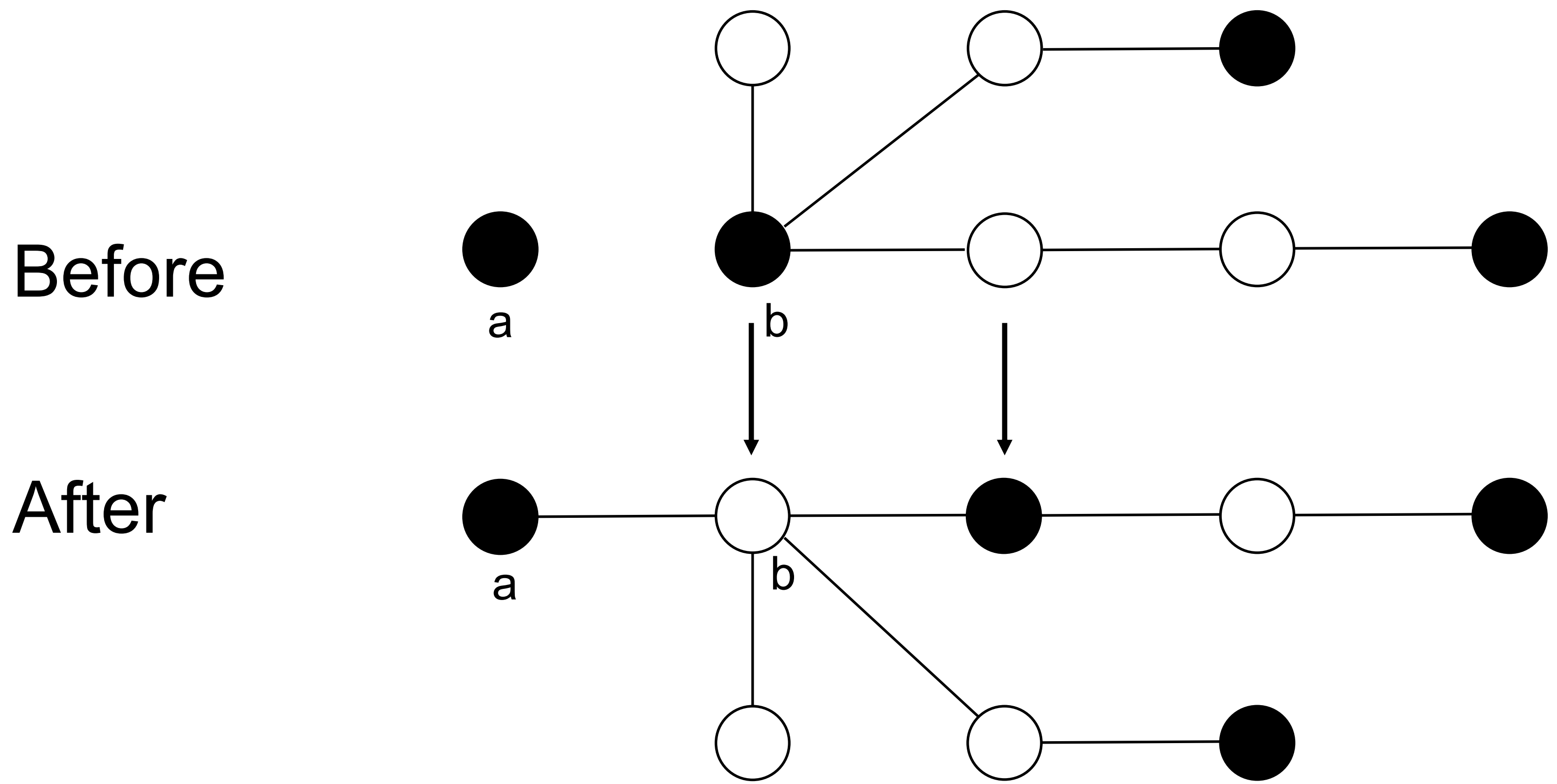
- Preprocessing
  - Input size N; $O(\sqrt{N})$ memory per machine; $O(\sqrt{N})$ machines
    - $N = m + n$
  - $O(n/\sqrt{N})$ machines
    - Stored statistics about each vertex v
      - $\deg(v)$
      - Whether in $LFMIS(G, \pi)$
      - $count(v)$
      - $N^+(v)$ & $N^-(v)$
      - Machine storing its neighbors
    - Vertices with consecutive IDs allocated together

Methodology

- Preprocessing
  - 1 coordinator machine: M_C
    - All updates are sent to it
    - Store additional information
      - Update history $\mathcal{H}$: last $O(\sqrt{N})$ updates
      - Corresponding machine storing statistics for each range of IDs
      - Memory available in each machine
    - Coordinate other machines to perform the update
    - Updated automatically

Methodology

- Insertion

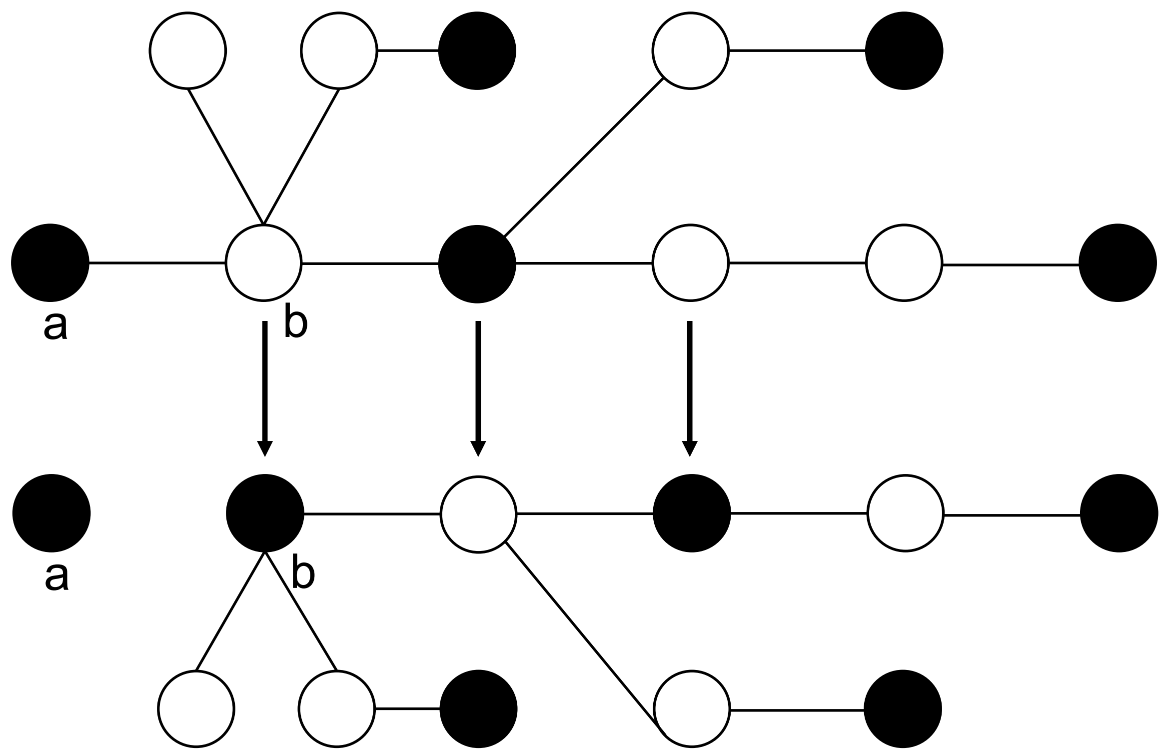


Methodology

- Deletion

Before

After



Methodology

- For each update to the graph
  - Update vertex neighbors based on $\mathcal{H}$
    - Might move vertices to another machine if fit
  - Update each machine in a round-robin fashion
    - Updated after at most $O(\sqrt{N})$ updates

Result

- Complexity of the dynamic algorithm
  - Number of rounds per update
    - Only fixed number of vertices need to be updated
  - Number of active machines & Total communication per round
    - Number of edge updates not yet updated: $O\left(\sqrt{N}\right)$
    - Size of moved vertices: at most the memory of each machine, $O\left(\sqrt{N}\right)$

Conclusion

- The fully-dynamic algorithm in the DMPC model maintains a maximal independent set in $O(1)$ rounds per update, while the number of machines that are active per round is $O(1)$, and the total communication per round is $O(\sqrt{N})$

Q & A

1. What are the benefits of a dynamic algorithm?
2. What are the restrictions of the DMPC algorithm?
3. Is the use of a coordinator in the dynamic algorithm to simulate a centralized algorithm?

Thank you!