Detection of Monomorphic Nodes in Large Graphs to Improve Privacy of Users in Online Social Networks

Hossein Shafiei

What, Why and How

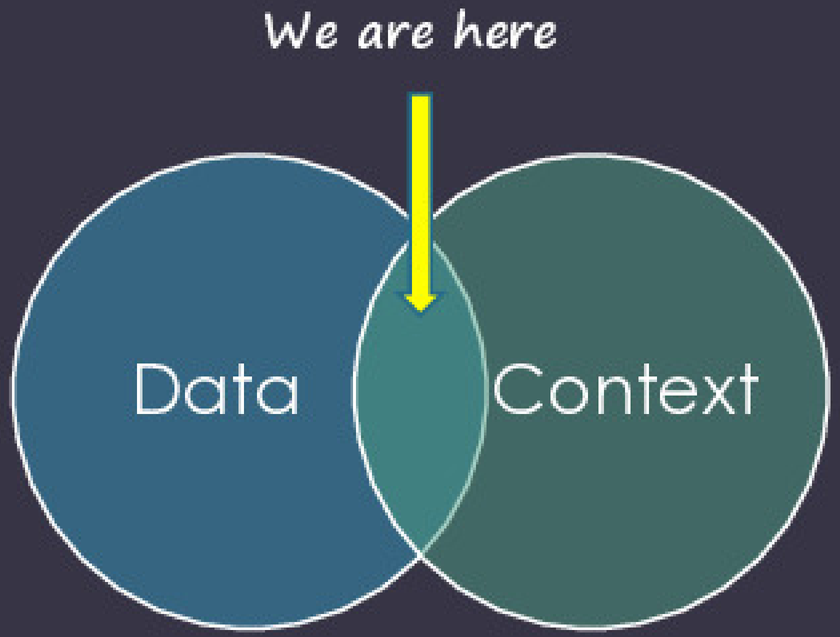
- What?
  - What is (implicit/explicit)-data privacy in online social networks?
- Why?
  - Why we should protect implicit-data privacy of users?
- How?
  - How can we detect and protect vulnerable users?

Definition & Regulation on Privacy

- European Union GDPR:
  - Data privacy means empowering users to make their own decisions about who can process their data and for what purpose.
- California State CCPA:
  - AB 375 allows any California consumer to demand to see all the information a company has saved on them, as well as a full list of all the third parties that data is shared with

Types of Privacy

- Data Privacy
  - Personal
  - Social
- Context Privacy
  - Location Privacy
  - Temporal Privacy
  - Rate Privacy



Data Privacy

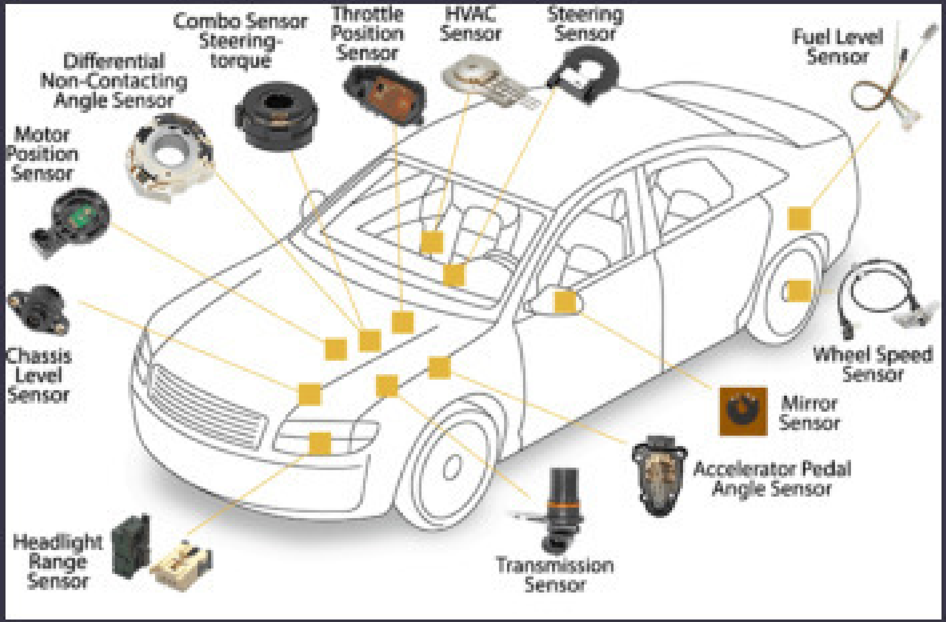
- Scope:
  - Personal: each individual's data (what is your name, what color is your car, ...)
  - Social: data about ones social interactions (who are your friends, what are their jobs ,...)
- Types of User Data:
  - Explicit, such as names, ids, etc.
  - Implicit, indirect data about user that collectively can divulge user's identity

Context Privacy

- Temporal Privacy:
  - When an specific event related to user happens? (e.g., when do they usually tweet)
- Location Privacy:
  - Where an specific event related to user happens? (e.g., From which location a request is initiated)
- Rate Privacy:
  - At which rate user events occur
- …

It's a scary new world (1)

- Driver identification
- With only few car sensors:
  - Steering wheel
  - Gyroscope
- Using convolutional neural networks (CNNs)
  - Drivers identified with up to 85% precision



It's a scary new world (2)

- Identification of Individuals based on their hourly cell phone traces
- Using Cellular antennas:
  - Only few spatio-temporal points are enough to uniquely identify 95% of the individuals

It's a scary new world (3)

- Identification of masked users in online social networks
  - Political reasons
  - Commercial incentives
  - Ransomware attacks

Privacy of Users in OSNs

- Exposed by your friends
  - "Tell me who your friends are and I'll tell you who are"
  - Typical approach would be to hide your sensitive friends
  - Even when users hide some of their friends, "links reconstruction attack" could be formed to predict user's hidden friends with high accuracy.

Privacy of Users in OSNs

- Exposed by your interests
  - Users may want to hide their interests, i.e., participated groups to improve their privacy
  - It has been shown that even with hiding 50% of users interests, attacker could predict their other half of interest with accuracy up to 90%.

Privacy of Users in OSNs

- Identification by social trolls
  - A **social troll** is someone who purposely says something controversial in order to get a rise out of other users
  - Piecemeal gathering of implicit data (Piecemeal Attack)
  - Fusion of those implicit data to identify users or their friends

Piecemeal Attack

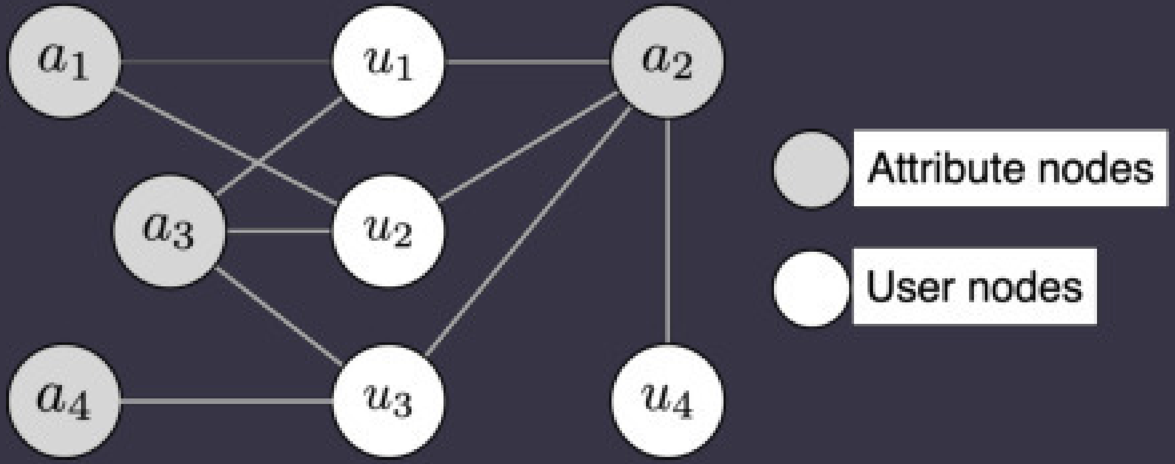
- Examples from Farsi twitter

Piecemeal Attack

- Examples from Farsi twitter

Modeling using Graphs

- Attribute graph
  - Two kind of vertices: (1) users (2) attributes
  - There is an edge between two vertices a_1 and u_1 if u_1 has the attribute a_1
  - $|V|$ vertices and $|E|$ edges
  - Maximum degree of attribute vertices (D_u)
  - Maximum degree of user vertices (D_a)
  - neighboring set of the node u (A_u)

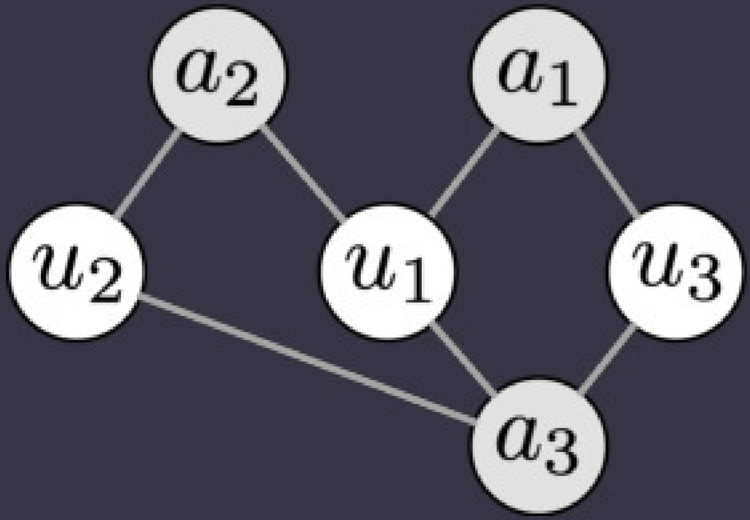


Properties of Attribute graphs

- Clustering coefficient is zero:
  - Lemma: Every cycle in an attribute graph has an even number of nodes. (Thus no triangles)
  - Clustering coefficient is the number of closed triplets (or 3 x triangles) over the total number of triplets
- $D_a << D_u$

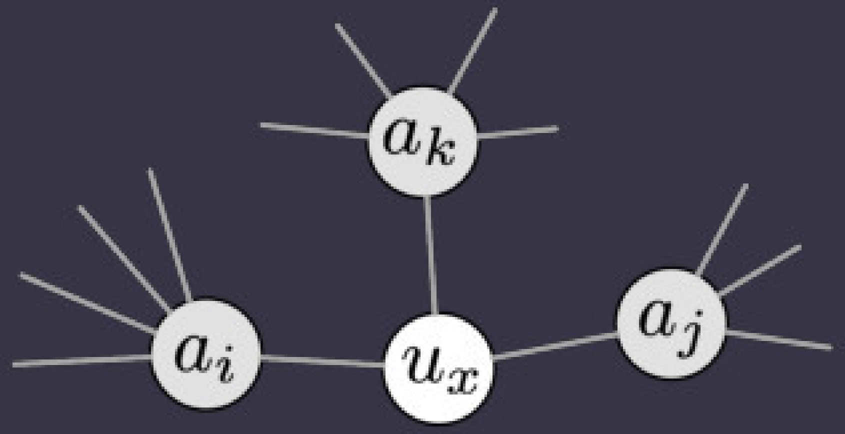
Attribute graphs

- Neighboring subsets:
  - Lemma: If every 4-cycle that starts from U_x either passes through U_y or passes through attributes connected to U_x, then $A_{Ux} \subseteq A_{Uy}$
  - For example, $A_{U3} \subseteq A_{U1}$ and not the other way



Monomorphism in Attribute Graphs

- Ux is monomorphic if there is no U_y such that $A_{U_y} \subseteq A_{U_x}$
- There is a $O(|V|^3)$algorithm to detect monomorphic vertices with $O(|V|+|E|)$ storage requirements
- Such graph for Facebook has **10^{12}** vertices

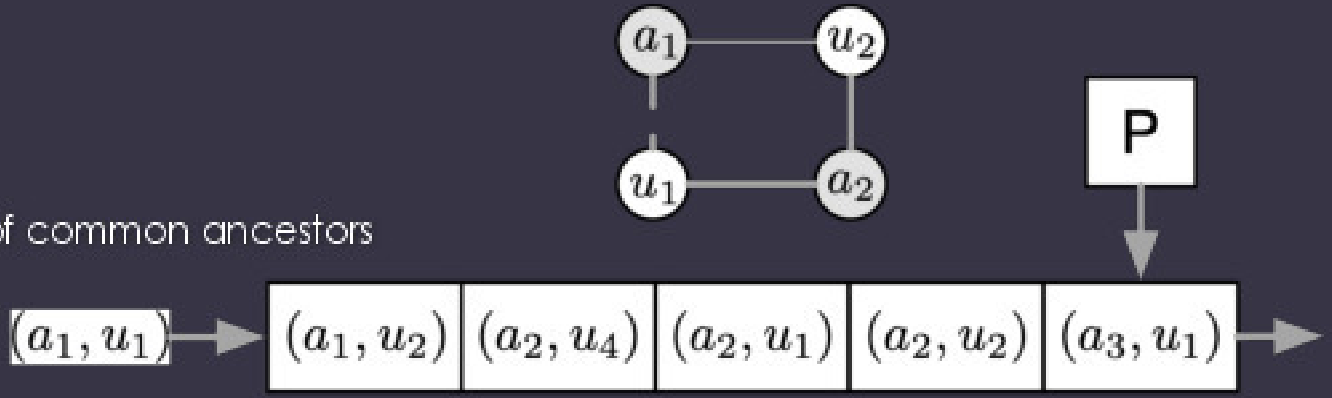


Detection Approaches

- Centralized
  - Not feasible for large graphs
- Streaming
  - Feeding the vertices and edges gradually to a computation unit
- Massively Parallel
  - Existing approaches are not suitable due to zero clustering

Streaming Approach

- Vertices are fed to a computing machine gradually
- The machine processes the input in a multi-pass manner
- The number of times that the machine linearly scans the memory is an important measure for the performance
- Vertex feed:
  - Randomized (using random walk)
  - Deterministic algorithm (BFS)
  - Approximation algorithm:
    - Weight probability based on number of common ancestors



Streaming Approach

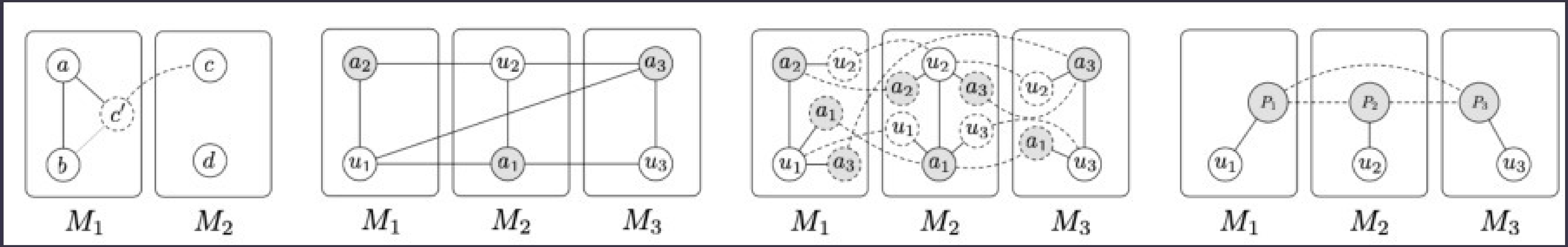
- Randomized (using random walk)
  - $O(|V|)$ space in worst case with $O(\log^2|v|)$ passes
- Deterministic algorithm
  - $O(D_u \log |V|)$ passes with $O(D_u{}^2)$ space
- Approximation algorithm
  - $O(D_u \log |V|)$ passes with $O(D_u \log |V|)$ space with D_a ratio

Massively Parallel Approach

- The graph is distributed over trusted computation machines
- Machines communicate with each other using message passing or via memory sharing
- Two Types:
  - Vertex-centric: Iteratively execute an algorithm over vertices of a graph for a predefined number of times or until they converge to the desired properties.
  - Edge-centric
- Existing approaches:
  - Google's Pregel
  - Facebook's GraphLab

Massively Parallel Approach

- None of the existing approaches perform well for attribute graph due to its clustering coefficient



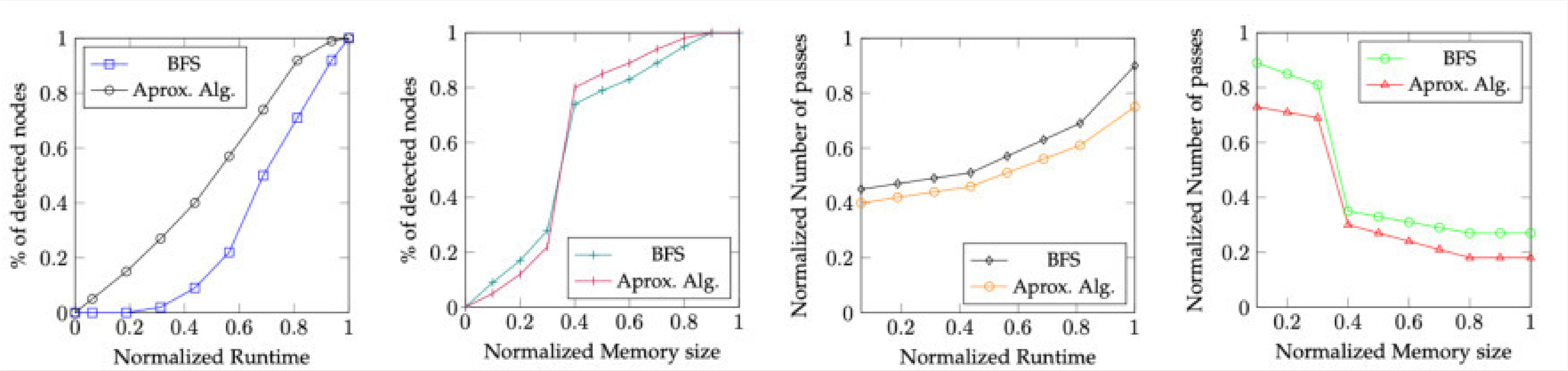
Our MP Approach

- User nodes are distributed into machines
- Each machine contains a node called the proxy node
- Each node performs a two-hop neighbor discovery using proxy node to communicate with each other
- After $O(D_u)$ iteration the algorithm converges
- T = inbound messages/all messages is an important performance metric

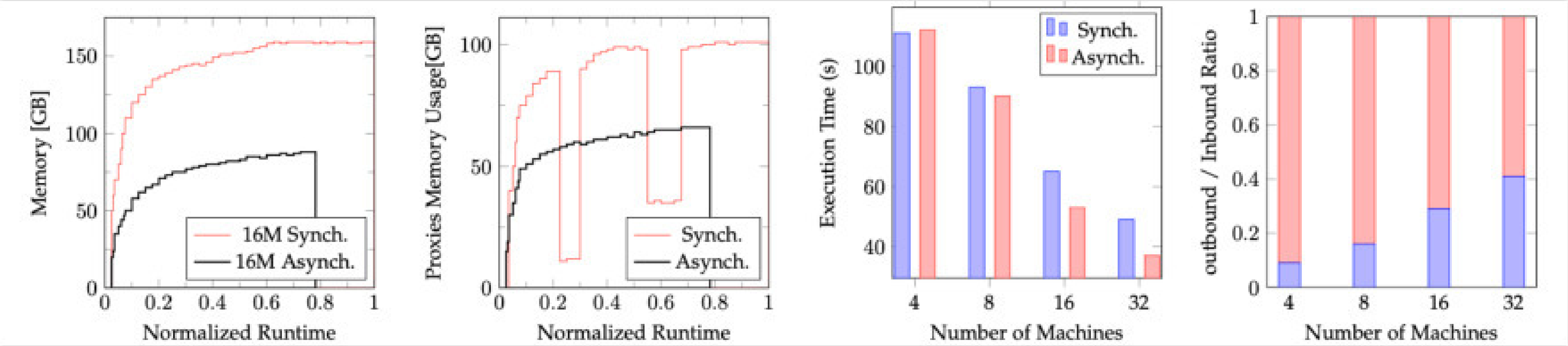
Our MP Approach

- Node distribution method highly impact T:
  - Randomized
    - Lowest overhead
  - Balanced Hash function (Synch)
    - Optimal with high overhead
  - "Secretary Problem" online algorithm (Asynchronous)
    - Sub-optimal with low overhead (1/e probability)

Evaluation of Streaming Approach



Evaluation of MP Approach



Evaluation of MP Approach

		$M = 4$	$M = 8$	$M = 16$
GraphLab	Exec. Time (s)	749	534	313
	Max. Mem. (GB)	743	612	509
Ours	Exec. Time (s)	107	88	59
	Max. Mem. (GB)	340	229	159

Thanks

Any Question?