Brief Announcement: Improved Massively Parallel Triangle Counting in $O(1)$ Rounds

Quanquan C. Liu

Yale University

quanquan.liu@yale.edu

C. Seshadri

UC SANTA CRUZ

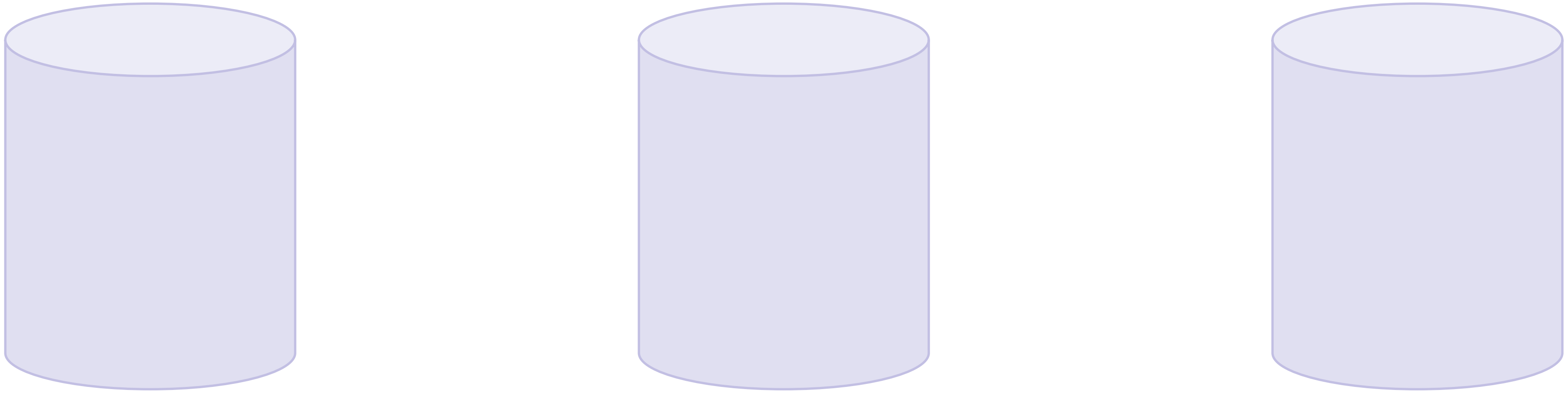
sesh@ucsc.edu

MPC Model Definition

- M machines
- Synchronous rounds

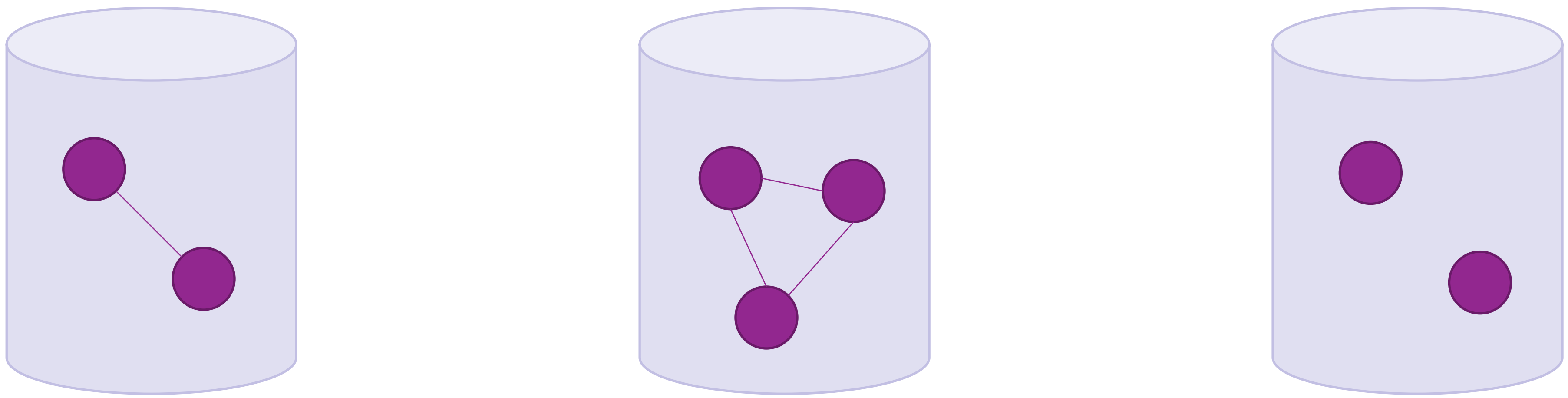
MPC Model Definition

- *M* machines
- Synchronous rounds



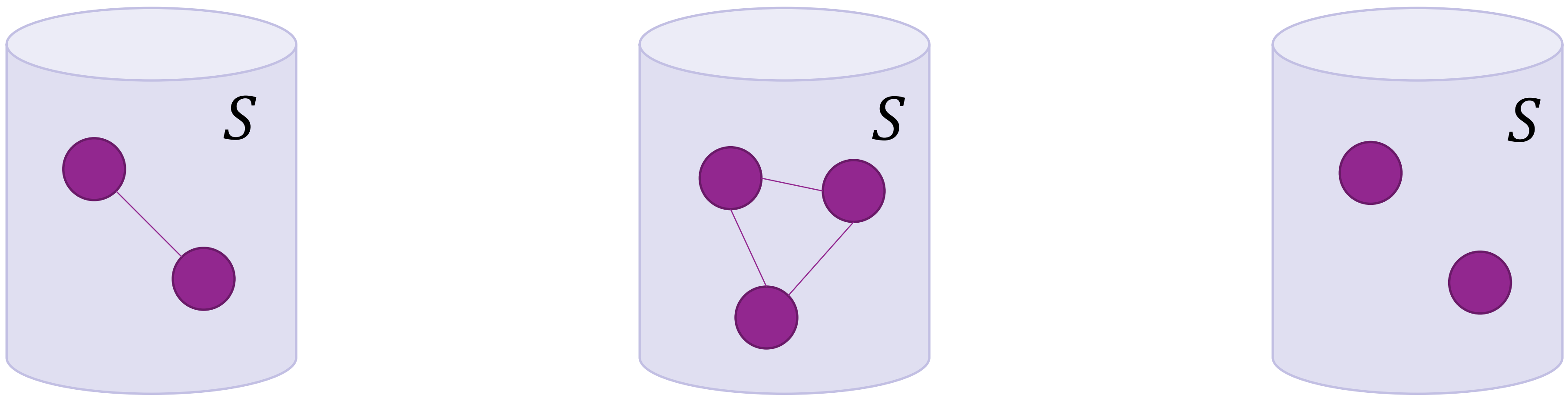
MPC Model Definition

- *M* machines
- Synchronous rounds



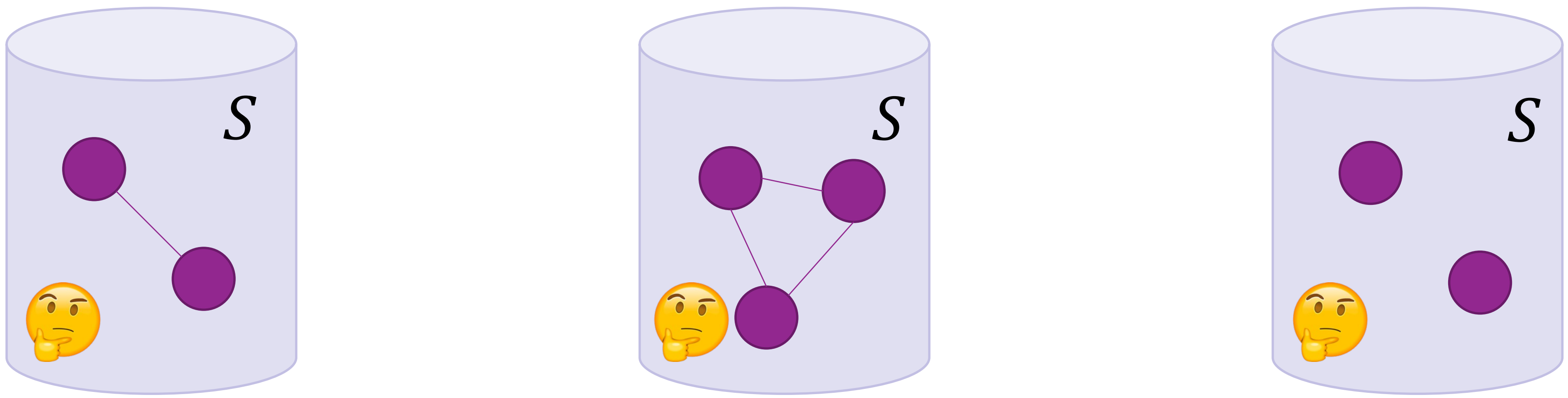
MPC Model Definition

- M machines
- Synchronous rounds



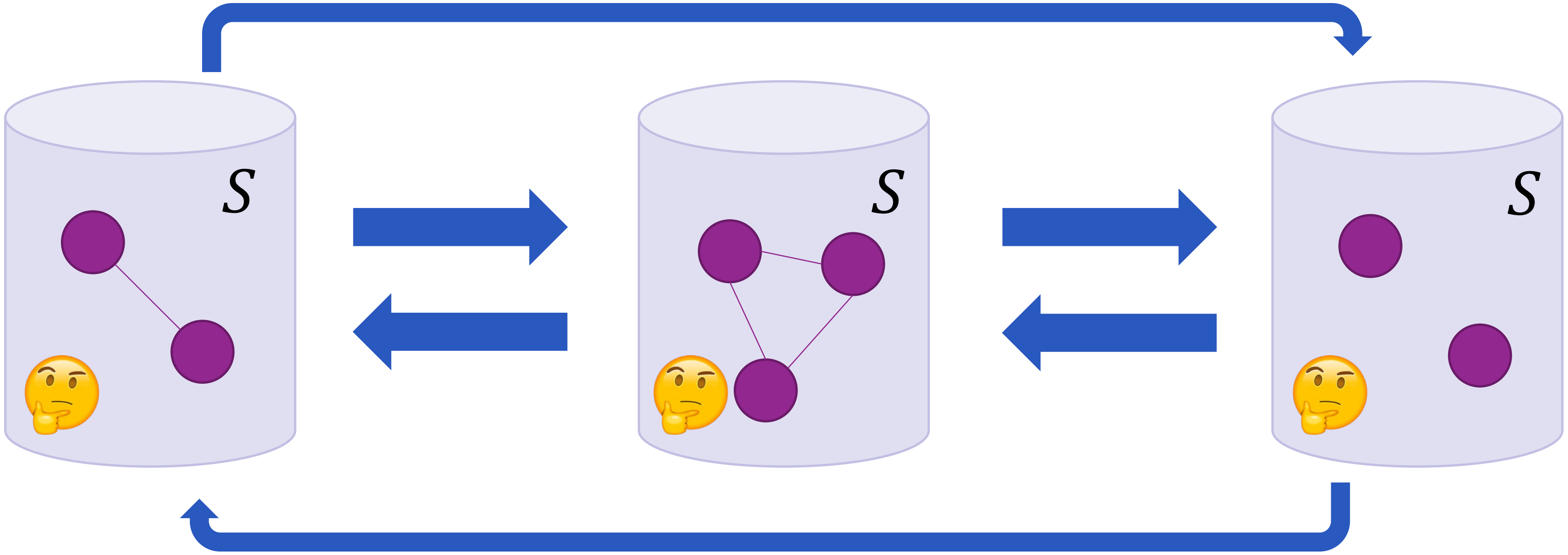
MPC Model Definition

- *M* machines
- Synchronous rounds



MPC Model Definition

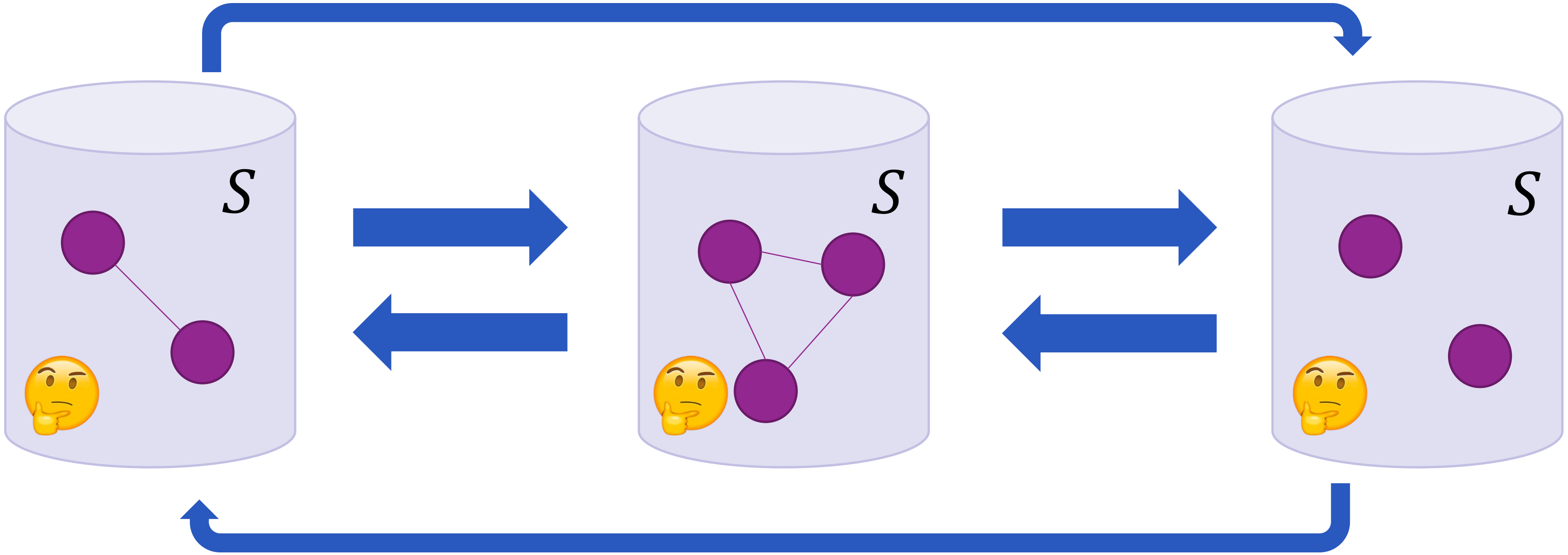
- *M* machines
- Synchronous rounds



MPC Model Definition

- M machines
- Synchronous rounds

Total Space: $\boldsymbol{M \cdot S}$



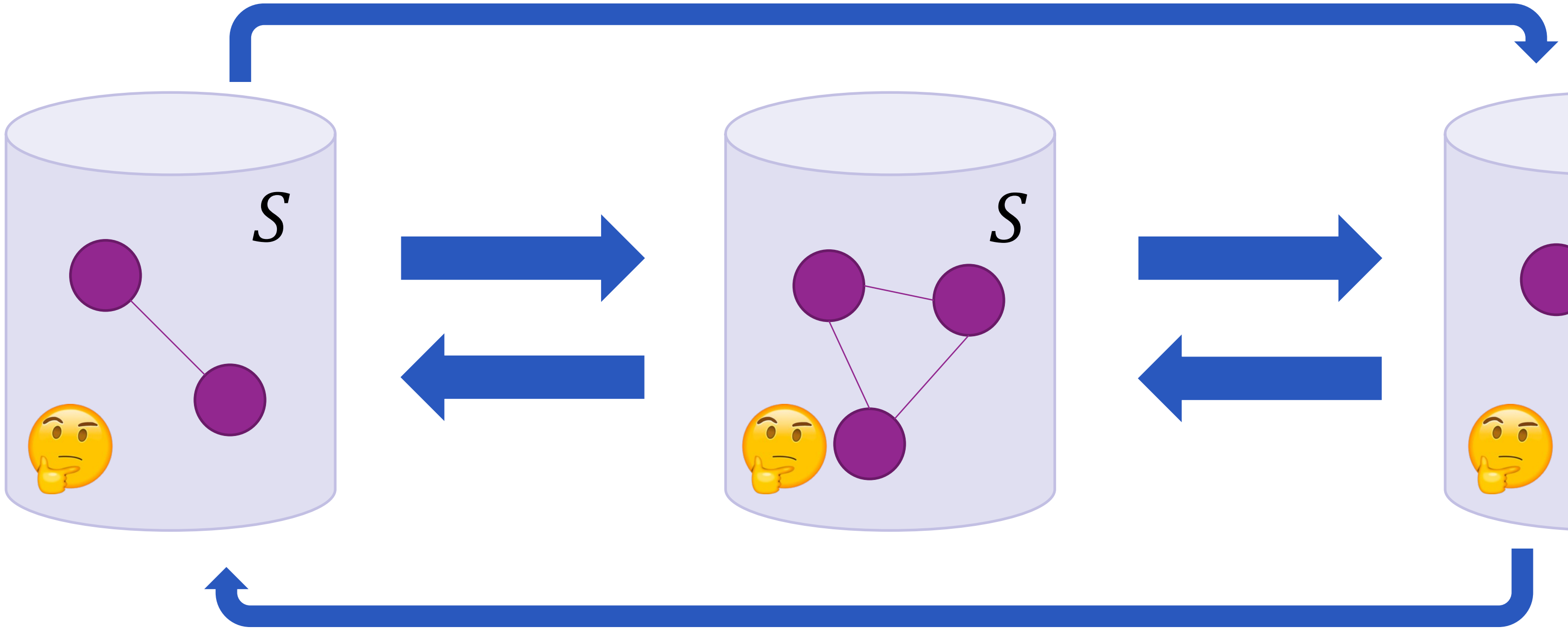
MPC Model Definition

- M machines
- Synchronous rounds

Complexity measures:

- Total Space
- Space Per Machine
- Rounds of communication

Total Space: $\boldsymbol{M \cdot S}$



Space per Machine in MPC

- **Strongly sublinear memory**:
  - $S = n^{\delta}$ for some constant $\delta \in (0, 1)$
- **Near-linear memory**:
  - $S = \tilde{\Theta}(n)$ (ignoring $\mathrm{poly}(\log(n))$ factors)
- **Strongly superlinear memory**:
  - $S = n^{1+\delta}$ for some constant $\delta > 0$

Also want: $\boldsymbol{O(1)}$ rounds

Space per Machine in MPC

- **Strongly sublinear memory**:
  - $S = n^{\delta}$ for some constant $\delta \in (0, 1)$
- **Near-linear memory**:
  - $S = \widetilde{\Theta}(n)$ (ignoring $\mathrm{poly}(\log(n))$ factors)
- **Strongly superlinear memory**:
  - $S = n^{1+\delta}$ for some constant $\delta > 0$

Also want: $\boldsymbol{O(1)}$ rounds

Also want: $\boldsymbol{\widetilde{O}(n + m)}$ total space

Space per Machine in MPC

- **Strongly sublinear memory**:
  - $S = n^\delta$ for some constant $\delta \in (0, 1)$
- **Near-linear memory**:
  - $S = \widetilde{\Theta}(n)$ (ignoring $\mathrm{poly}(\log(n))$ factors)
- **Strongly superlinear memory**:
  - $S = n^{1+\delta}$ for some constant $\delta > 0$

Also want: $\boldsymbol{O(1)}$ rounds

Also want: $\boldsymbol{\widetilde{O}(n+m)}$ total space

All space per machine are **sublinear in number of edges *m*** in graph

Space per Machine in MPC

- **Strongly sublinear memory**:
  - $S = n^{\delta}$ for some constant $\boxed{\delta \in (0, 1)}$
- **Near-linear memory**:
  - $S = \widetilde{\Theta}(n)$ (ignoring $\mathrm{poly}(\log(n))$ factors)
- **Strongly superlinear memory**:
  - $S = n^{1+\delta}$ for some

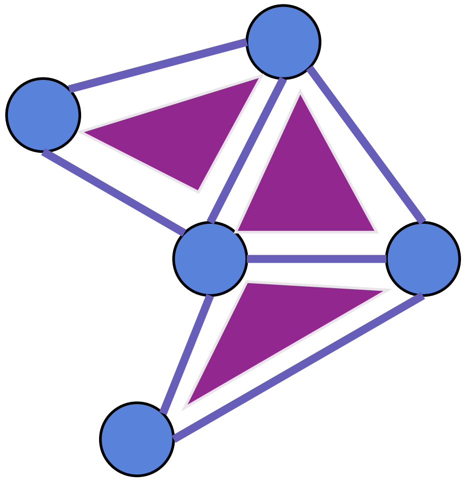
Also want: $O(1)$ rounds

Also want: $\widetilde{O}(n + m)$ total space

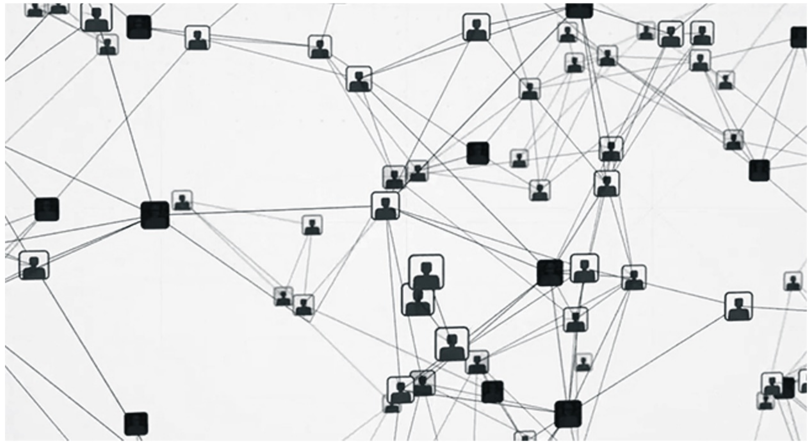
Often $\delta = \frac{1}{2}$

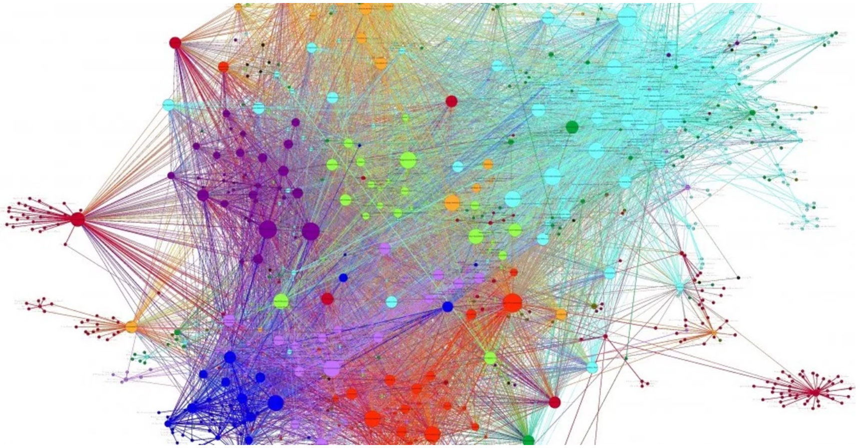
All space per machine are **sublinear in number of edges** ***m*** in graph

Triangle Counting



3 triangles





- Clustering Algorithms
- Identifying thematic structures of networks
- Spam and fraud detection
- Link classification and recommendation
- Joining three relations in a database
- Database query optimization

Triangle Counting in MPC Model

Exact Setting			
Previous Work	MPC Rounds	Space Per Machine	Total Space
[SV11]	1	$O(m/\rho^2)$	$O(\rho m)$
[CC11]	$O(n)$	$O(n)$	$O(m)$
Folklore [CN85]	$O(\log n)$	$O(\alpha^2)$	$O(m\alpha)$
[BELMR22]	$O(\log \log n)$	$O(n^\delta)$	$O(m\alpha)$

$\boldsymbol{\delta > 0}$ is **any constant**

[SV11]: Suri and Vassilvitski, WWW '11
[CC11]: Chu and Cheng KDD '11
[CN85]: Chiba and Nishizeki SICOMP '85
[BELMR22]: Biswas, Eden, **L**, Mitrović, Rubinfeld APPROX '22

Triangle Counting in MPC Model

Exact Setting			
Previous Work	MPC Rounds	Space Per Machine	Total Space
[SV11]	1	$O(m/\rho^2)$	$O(\rho m)$
[CC11]	$O(n)$	$O(n)$	$O(m)$
Folklore [CN85]	$O(\log n)$	$O(\alpha^2)$	$O(m\alpha)$
[BELMR22]	$\boldsymbol{O(\log\log n)}$	$\boldsymbol{O(n^\delta)}$	$\boldsymbol{O(m\alpha)}$

$\boldsymbol{\delta > 0}$ is **any constant**

[SV11]: Suri and Vassilvitski, WWW '11
[CC11]: Chu and Cheng KDD '11
[CN85]: Chiba and Nishizeki SICOMP '85
[BELMR22]: Biswas, Eden, **L**, Mitrović, Rubinfeld APPROX '22

Triangle Counting in MPC Model

Exact Setting			
Previous Work	MPC Rounds	Space Pe	
[SV11]	1	$O(m$	
[CC11]	$O(n)$	$O($	
Folklore [CN85]	$O(\log n)$	$O($	
[BELMR22]	$\boldsymbol{O(\log\log n)}$	$\boldsymbol{O(}$	

Arboricity α: number of forests that edges can be partitioned into

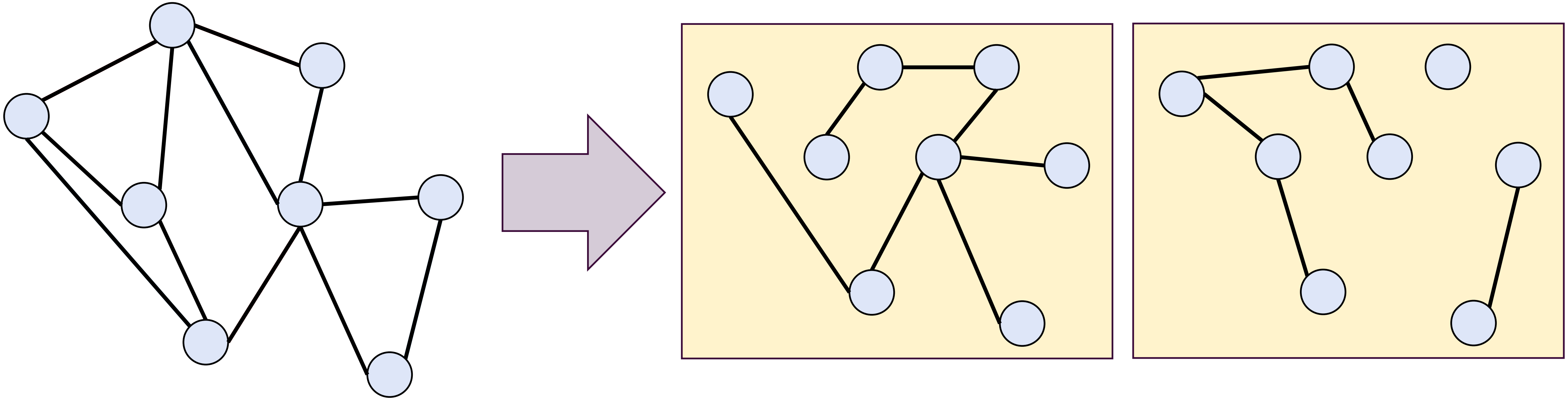
Real-world graphs: **arboricity generally** $\mathrm{poly}(\log \boldsymbol{n})$

$\boldsymbol{\delta > 0}$ is **any constant**

[SV11]: Suri and Vassilvitski, WWW '11
[CC11]: Chu and Cheng KDD '11
[CN85]: Chiba and Nishizeki SICOMP '85
[BELMR22]: Biswas, Eden, **L**, Mitrović, Rubinfeld APPROX '22

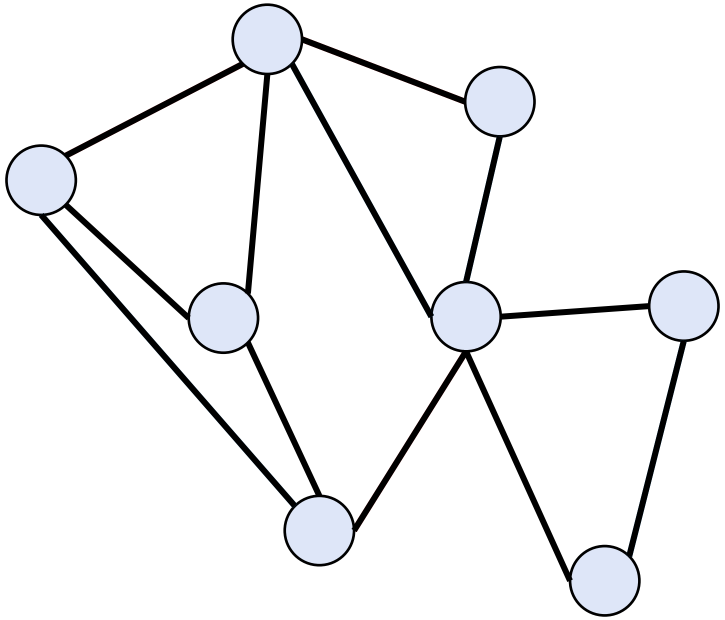
Arboricity of the Graph

- **Arboricity** of the graph
    - Minimum number of forests to decompose the graph



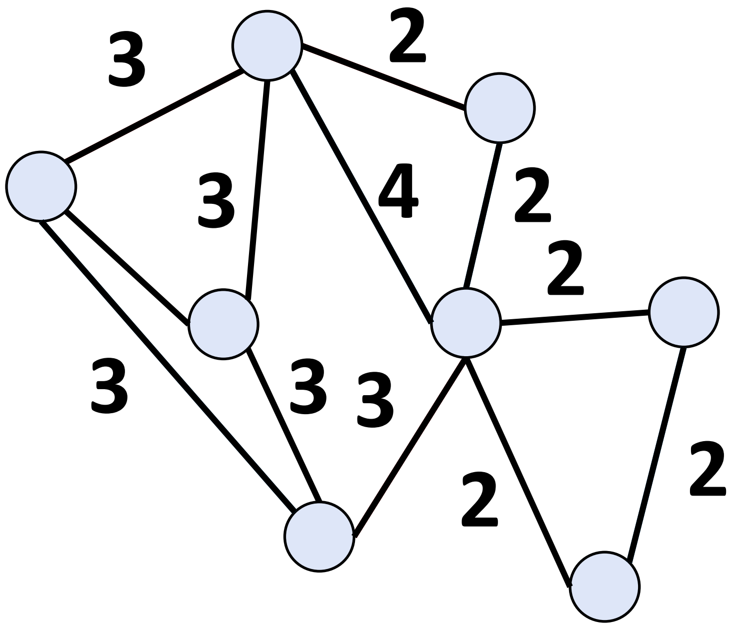
Chiba-Nishizeki Sum of Minimum Degree Endpoints [CN85]

- Given an input graph $G = (V, E)$ with arboricity $\boldsymbol{\alpha}$, it holds sum of minimum degrees endpoints of every edge is at most $\mathbf{2}\boldsymbol{m\alpha}$



Chiba-Nishizeki Sum of Minimum Degree Endpoints [CN85]

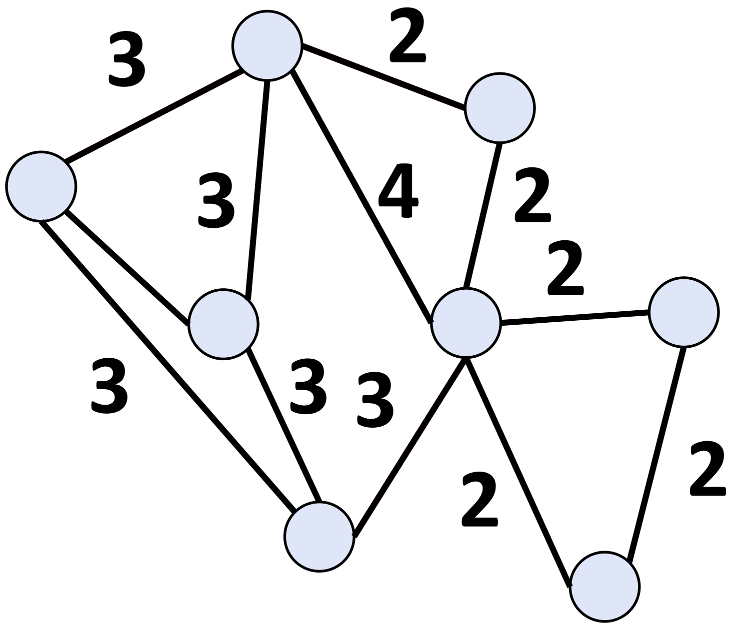
- Given an input graph $G = (V, E)$ with arboricity $\boldsymbol{\alpha}$, it holds sum of minimum degrees endpoints of every edge is at most $\boldsymbol{2m\alpha}$



Chiba-Nishizeki Sum of Minimum Degree Endpoints [CN85]

- Given an input graph $G = (V, E)$ with arboricity $\boldsymbol{\alpha}$, it holds sum of minimum degrees endpoints of every edge is at most $\boldsymbol{2m\alpha}$

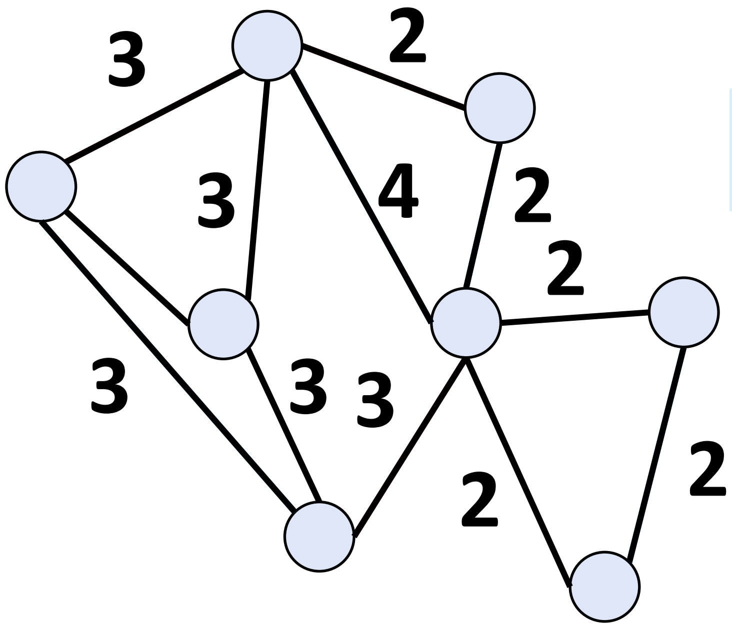
Sum of minimum degrees of endpoints = **29**



Chiba-Nishizeki Sum of Minimum Degree Endpoints [CN85]

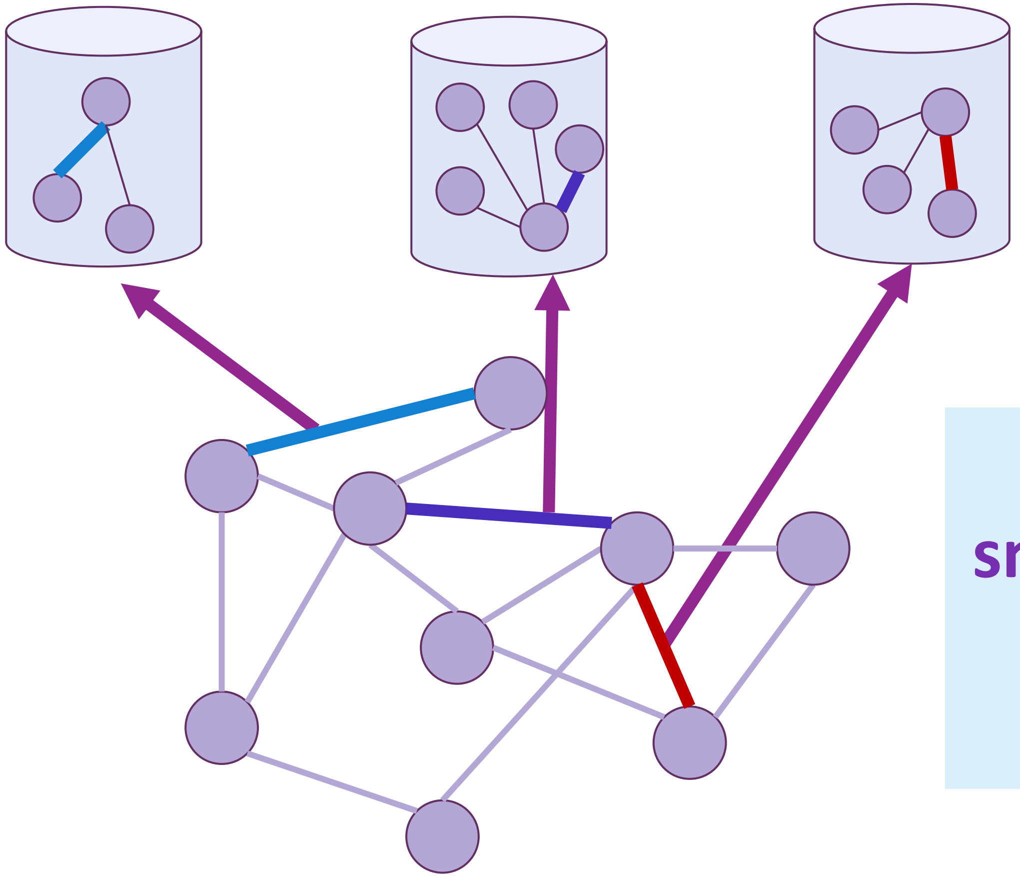
- Given an input graph $G = (V, E)$ with arboricity $\boldsymbol{\alpha}$, it holds sum of minimum degrees endpoints of every edge is at most $\boldsymbol{2m\alpha}$

Sum of minimum degrees of endpoints = **29**



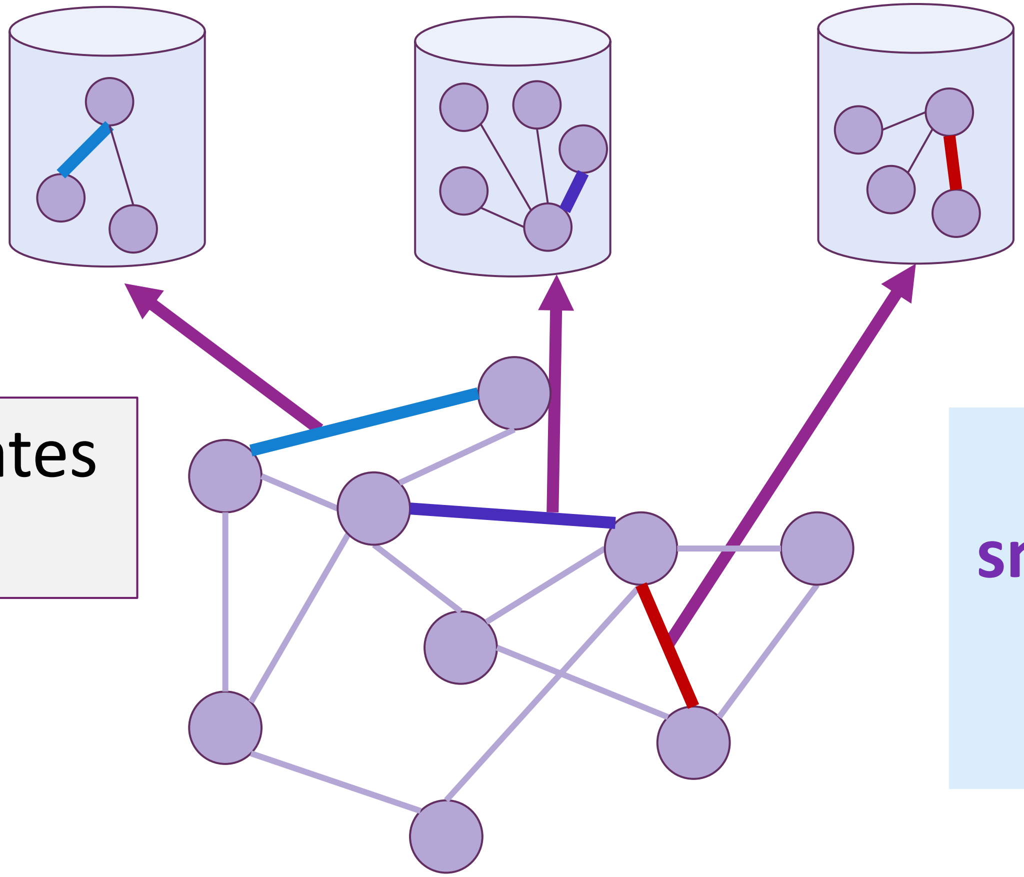
$$\mathbf{29 \le 2 \cdot 12 \cdot 2 = 48}$$

MPC Triangle Counting Algorithm



Send adjacency list of **smaller degree endpoint of each edge** to machines

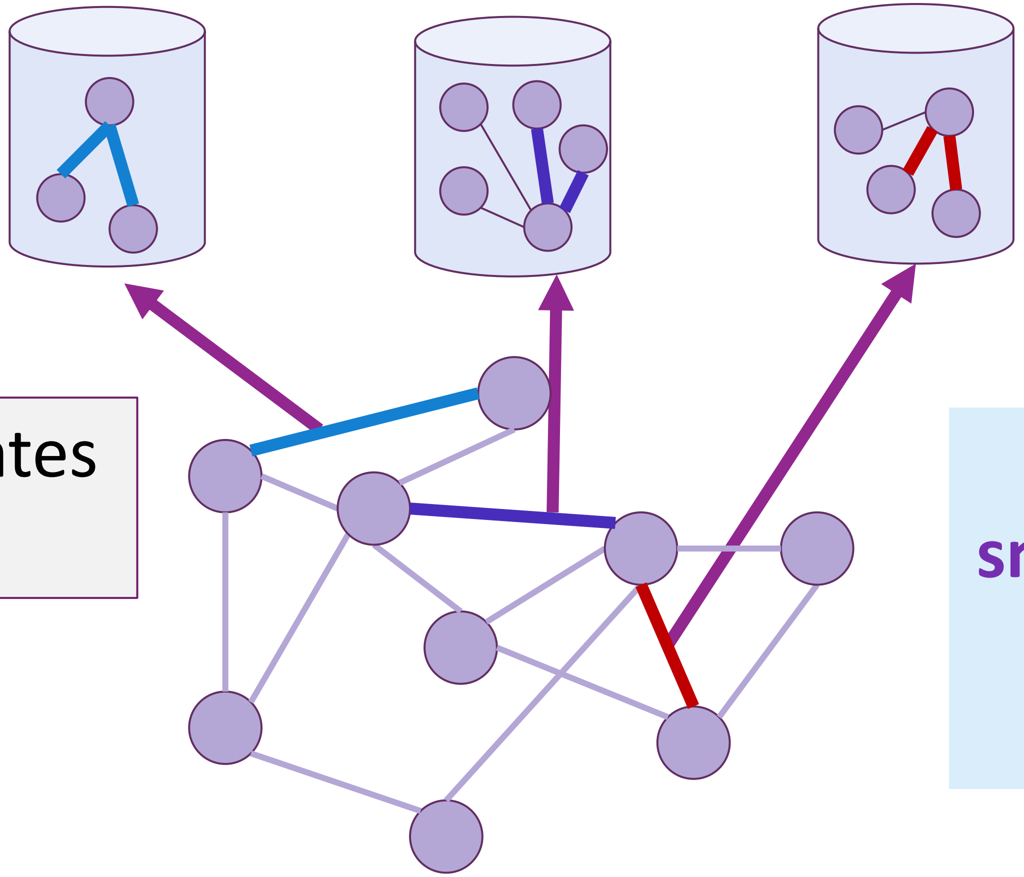
MPC Triangle Counting Algorithm



Each machine generates **wedge queries**

Send adjacency list of **smaller degree endpoint of each edge** to machines

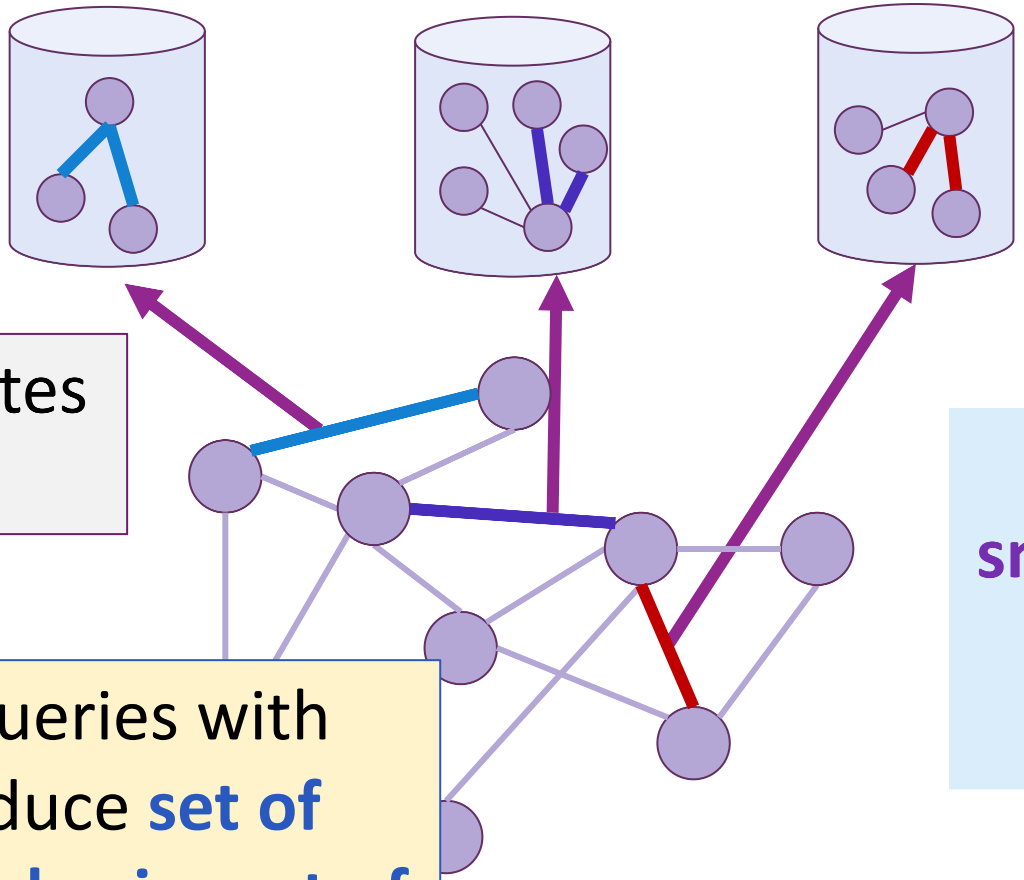
MPC Triangle Counting Algorithm



Each machine generates **wedge queries**

Send adjacency list of **smaller degree endpoint of each edge** to machines

MPC Triangle Counting Algorithm

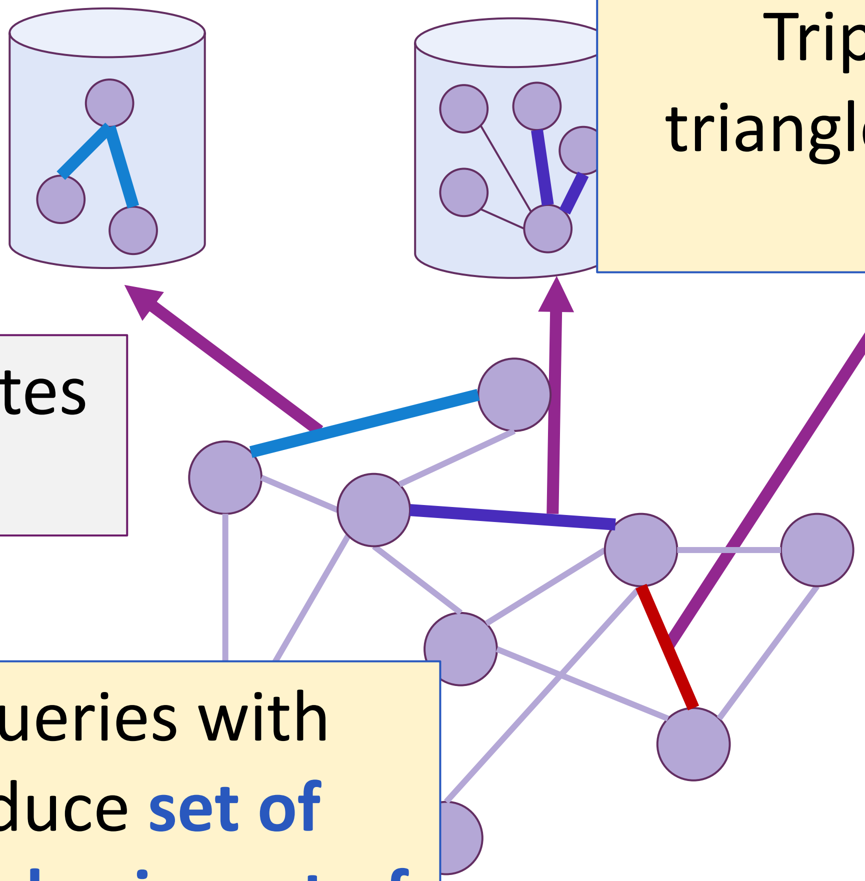


Each machine generates **wedge queries**

Sort wedge queries with edges to produce **set of triangles each edge is part of**

Send adjacency list of **smaller degree endpoint of each edge** to machines

MPC Triangle Counting Algorithm



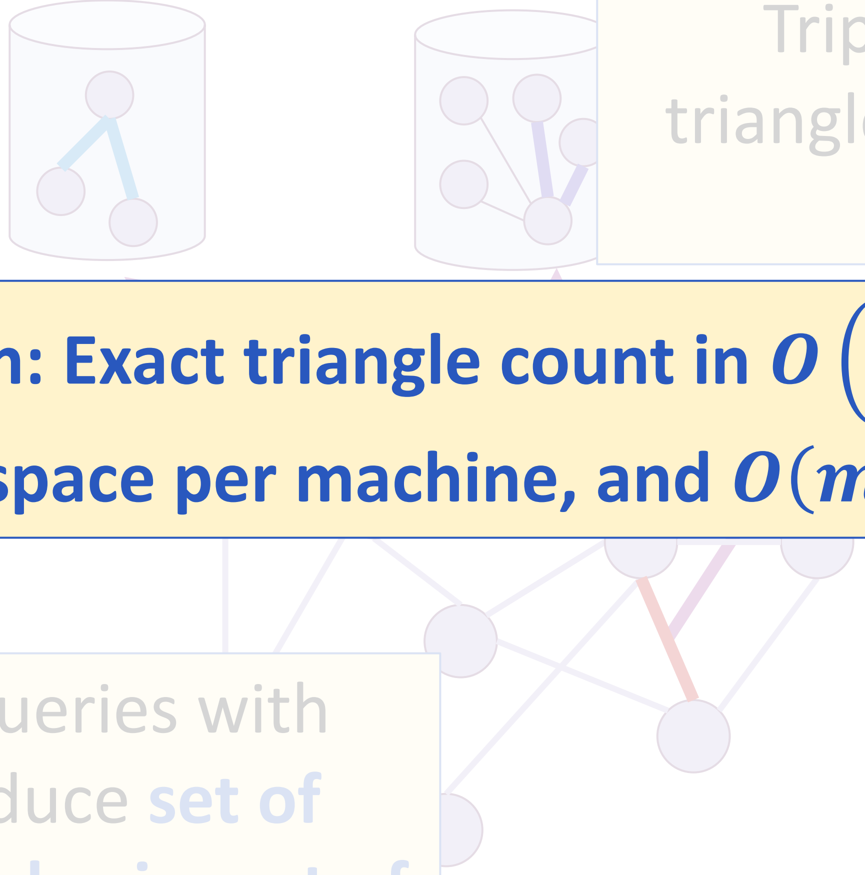
Triple the number of triangles: divide by 3 to get count

Each machine generates **wedge queries**

Send adjacency list of **smaller degree endpoint of each edge** to machines

Sort wedge queries with edges to produce **set of triangles each edge is part of**

MPC Triangle Counting Algorithm



Triple the number of triangles: divide by 3 to get count

Each machi
wedge

Theorem: Exact triangle count in $O\left(\frac{1}{\delta}\right)$ MPC rounds, $O(n^{\delta})$ space per machine, and $O(m\alpha)$ total space.

ncy list of **smaller degree endpoint of each edge** to machines

Sort wedge queries with edges to produce **set of triangles each edge is part of**

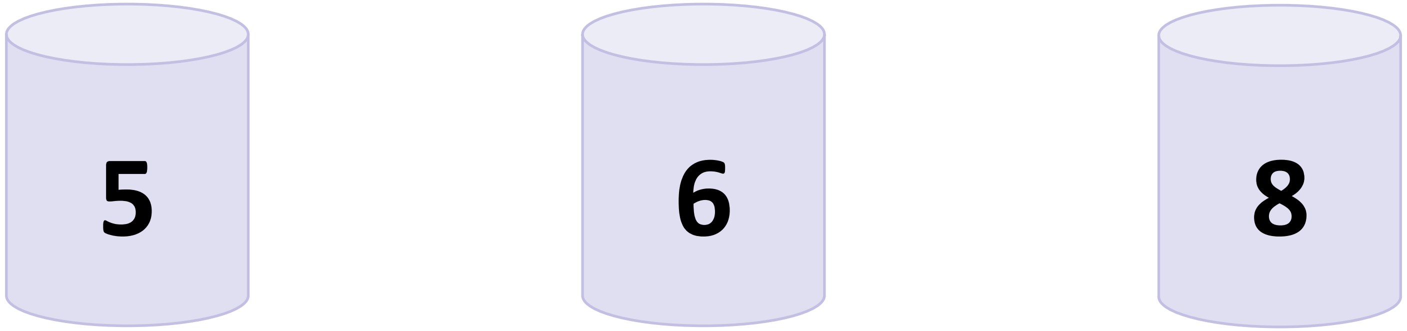
Matching Lower Bound

Theorem: Exact triangle count requires $\Omega\left(\frac{1}{\delta}\right)$ MPC rounds, when given $O\left(n^{\delta}\right)$ space per machine, and $O(m\alpha)$ total space.

Matching Lower Bound

Theorem: Exact triangle count requires $\Omega\left(\frac{1}{\delta}\right)$ MPC rounds, when given $O(n^{\delta})$ space per machine, and $O(m\alpha)$ total space.

Each machine contains disjoint count

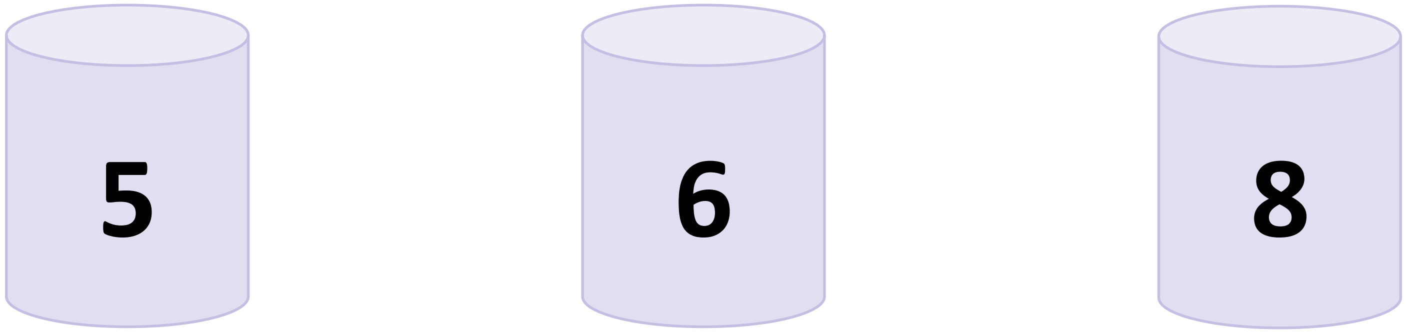


m/n^{δ} machines

Matching Lower Bound

Theorem: Exact triangle count requires $\Omega\left(\frac{1}{\delta}\right)$ MPC rounds, when given $O(n^{\delta})$ space per machine, and $O(m\alpha)$ total space.

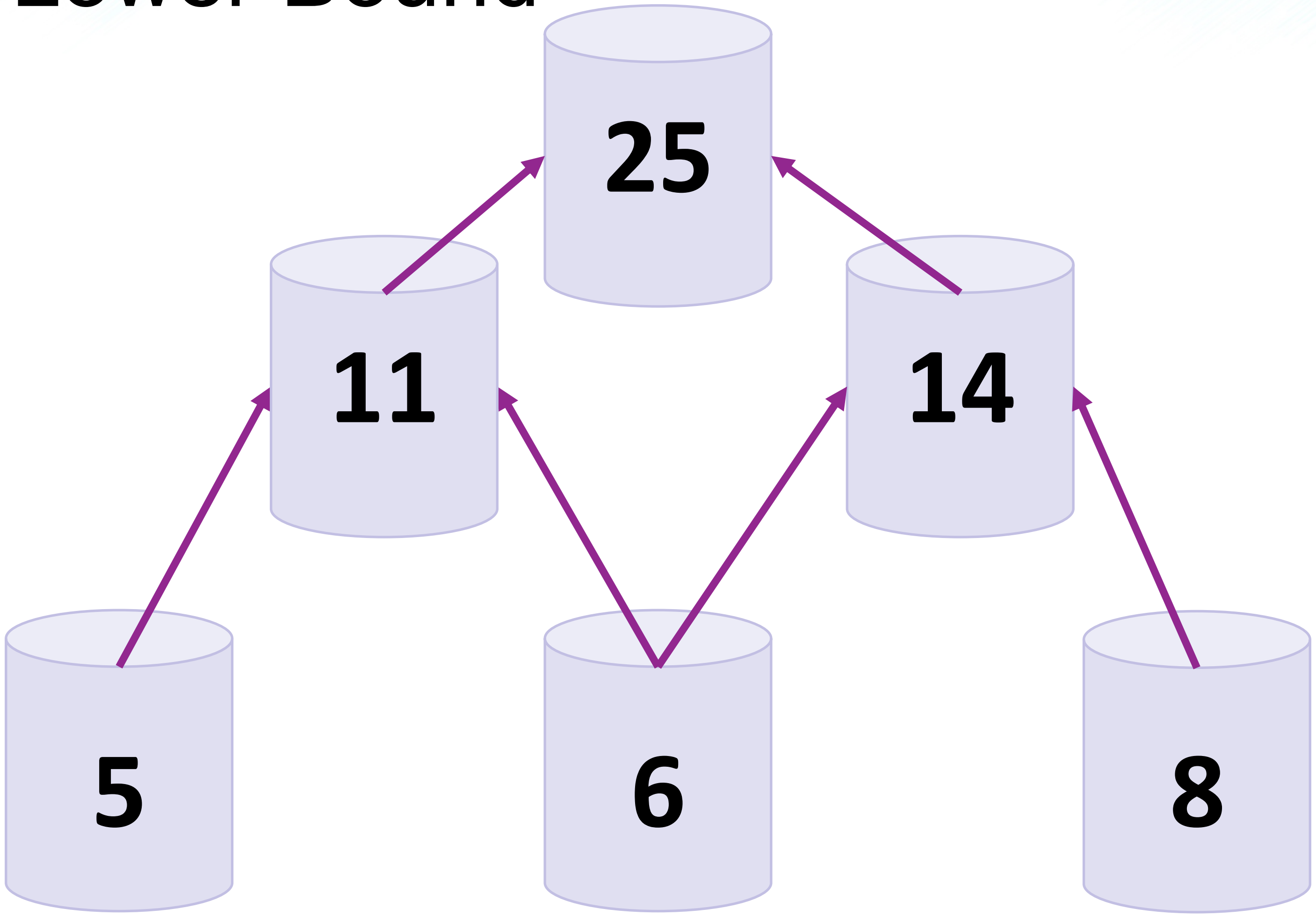
Aggregate count into one machine—tree with m/n^{δ} leaves



m/n^{δ} machines

Matching Lower Bound

Aggregate count into one machine—tree with m/n^δ leaves

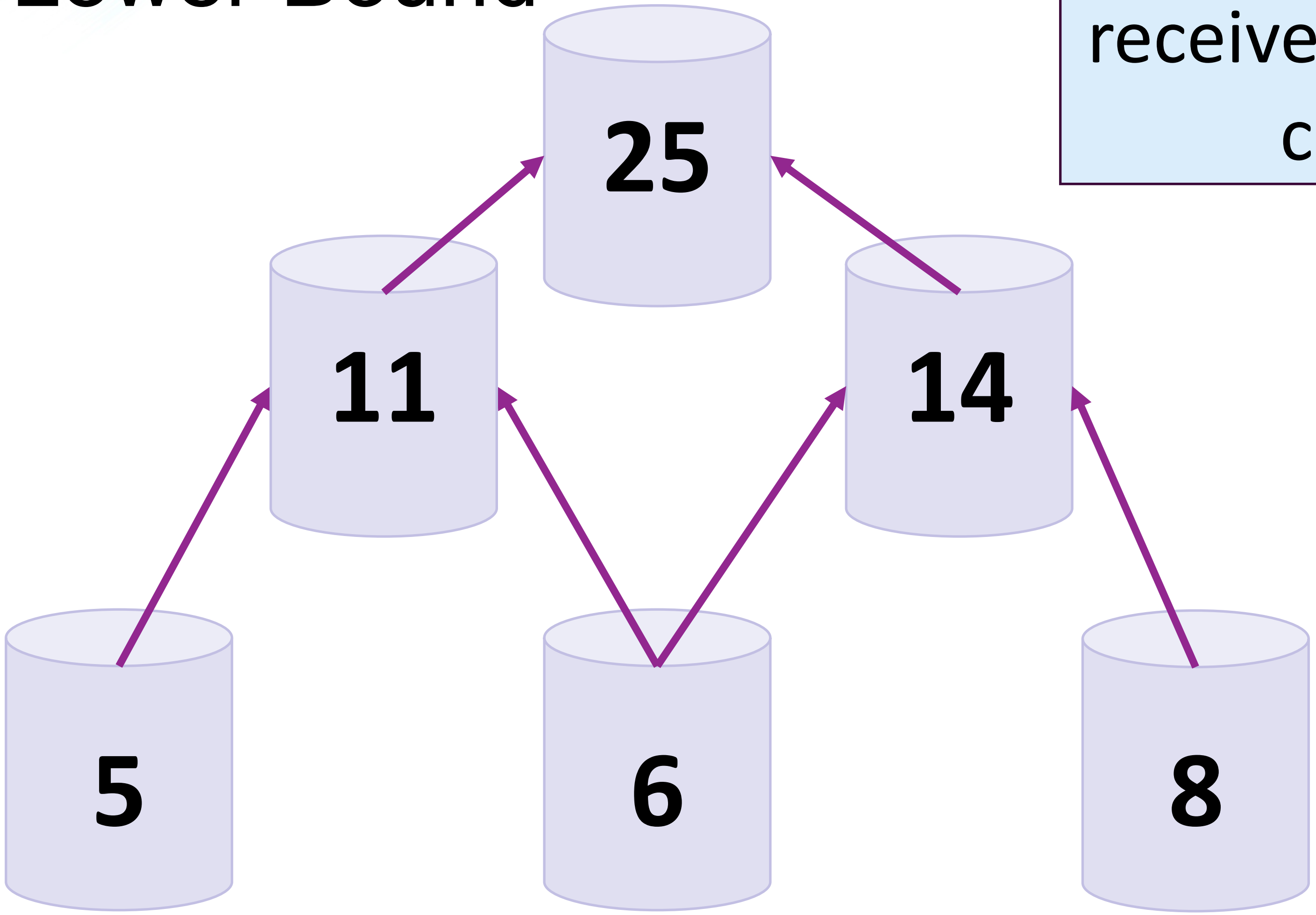


m/n^δ machines

Matching Lower Bound

Each machine can receive at most n^{δ} counts

Aggregate count into one machine—tree with m/n^{δ} leaves



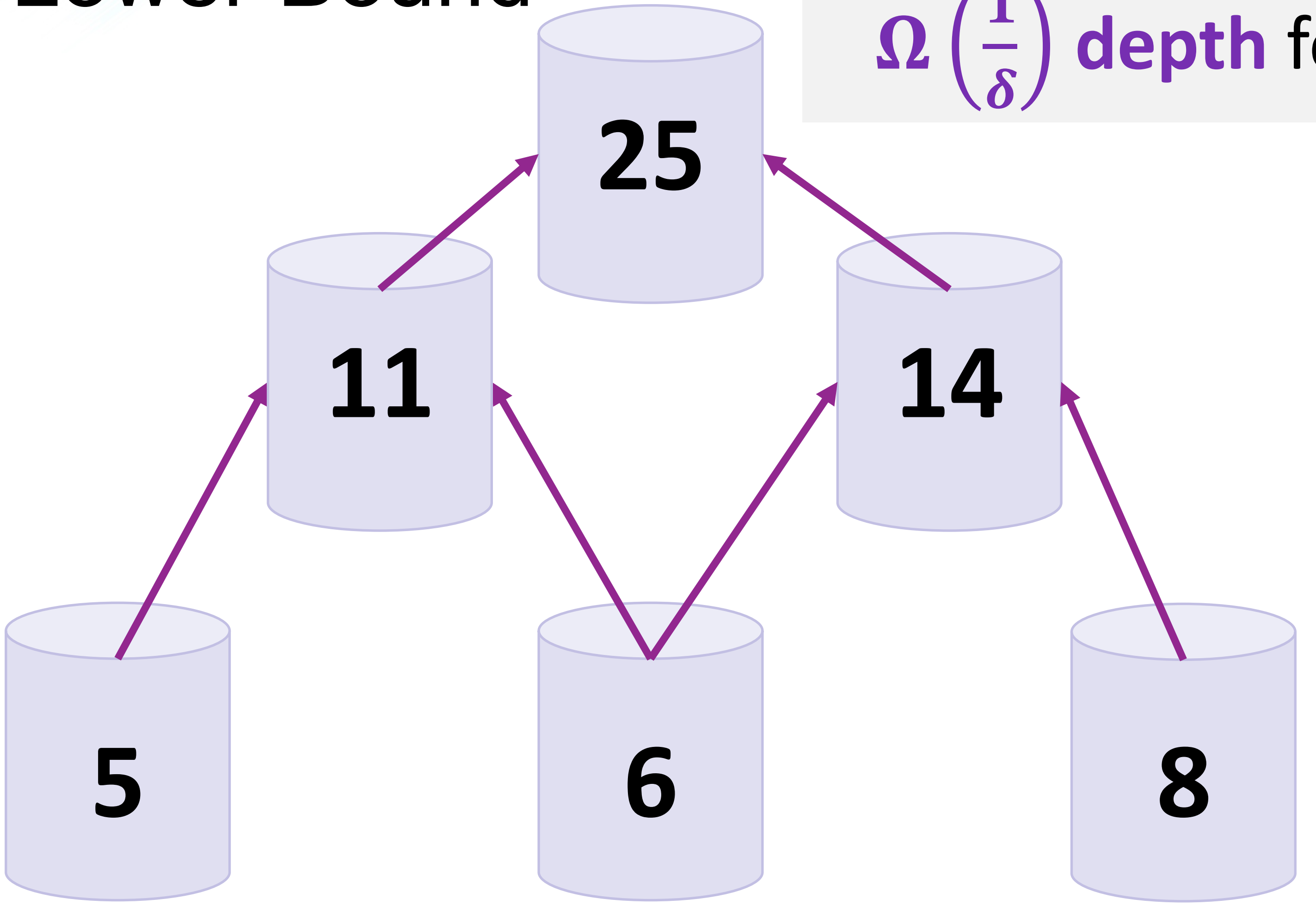
m/n^{δ} machines

Matching Lower Bound

$$\log_{n^\delta}\left(\frac{m}{n^\delta}\right) \le \log_{n^\delta}(n^2) =$$

$\Omega\left(\frac{1}{\delta}\right)$ **depth** for $\delta \in (0,1)$

Aggregate count into one machine—tree with m/n^δ leaves



m/n^δ machines

Open Questions and Future Directions

- Small subgraph counting for a **broader class of small subgraphs**

Open Questions and Future Directions

- Small subgraph counting for a **broader class of small subgraphs**
- **Decrease the total space usage** for exact triangle counting in $O(1)$ rounds to $O(m+n)$ (even $\sqrt{n}$ number of rounds, linear space per machine not known)

Open Questions and Future Directions

- Small subgraph counting for a **broader class of small subgraphs**
- **Decrease the total space usage** for exact triangle counting in $O(1)$ rounds to $O(m+n)$ (even $\sqrt{n}$ number of rounds, linear space per machine not known)
- Approximate triangle counting in $O(1)$ rounds and strictly sublinear space in **sparse graphs** where $m = \tilde{O}(n)$