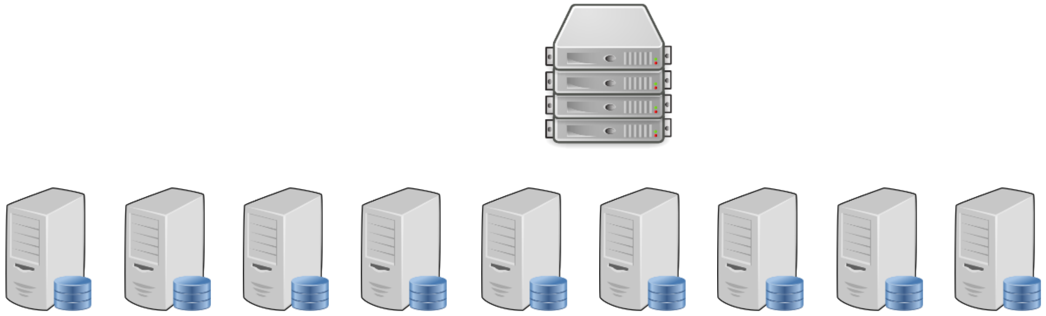
Massively Parallel Computing in a Heterogeneous Regime

ORR FISCHER

WEIZMANN INSTITUTE OF SCIENCE

BASED ON A JOINT WORK WITH ADI HOROWITZ AND ROTEM OSHMAN (TEL AVIV UNIVERSITY), AND SEVERAL PRIOR WORKS



Plan

Model & Motivation

Techniques in HMPC

Prior and New Results

Open Problems

Massively-Parallel Computing (MPC)

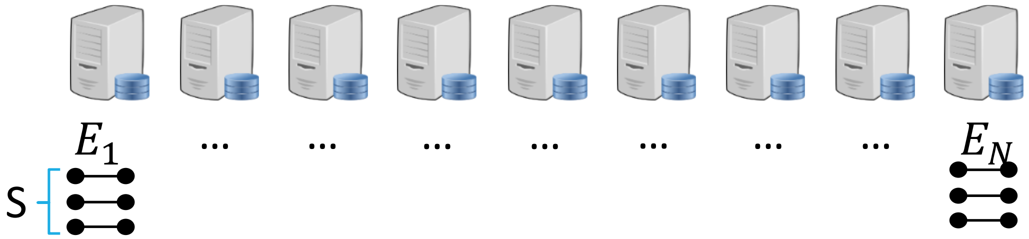
[Karloff, Suri, Vassilvitski'10…]

Input partitioned across N machines
- This talk: graph problems
- m – #edges, n – #vertices, d = 2m/n

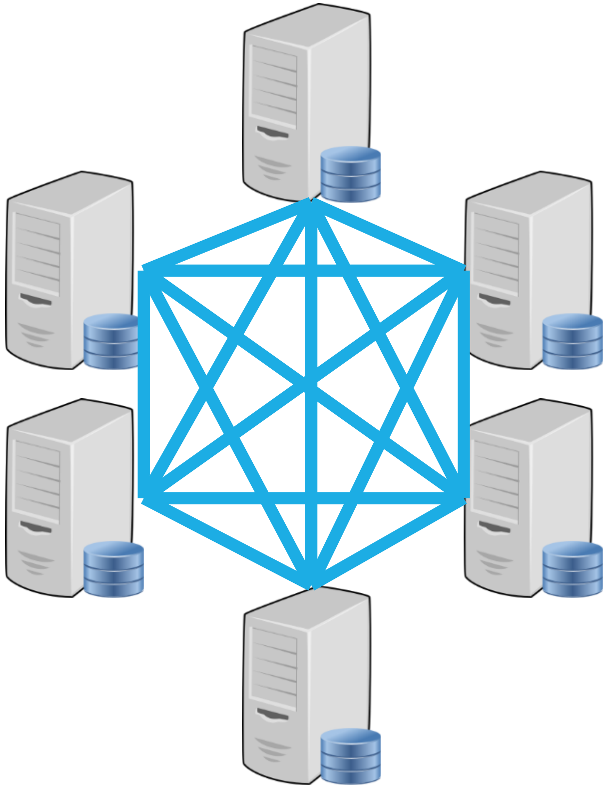
Space: S per machine, $\tilde{O}(m)$ total

Communication round:
- Send and receive S bits
- Local computation (unbounded)

Complexity measure: Rounds. Ideally O(1)…



$$E = E_1 \cup \cdots \cup E_N$$
$$|E| = m$$



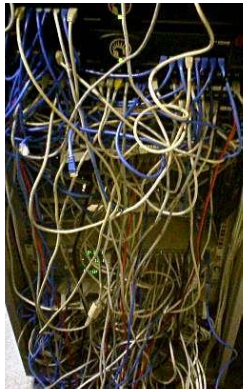
Space Regimes

Sublinear: $S = \tilde{\Theta}(n^{\gamma})$ for $\gamma \in (0,1)$ 

Near-linear: $S = \tilde{\Theta}(n)$ 

Superlinear: $S = \tilde{\Theta}\left(n^{1+\gamma}\right)$ for $\gamma \in (0,1)$ 

Lower Bounds & MPC

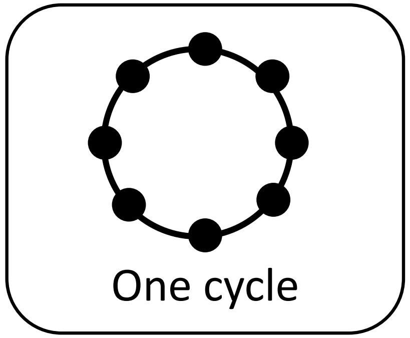


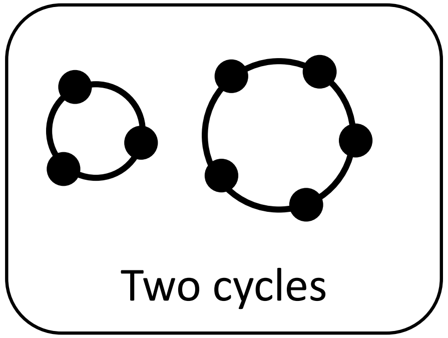
Lower bounds in MPC -> Strong circuit complexity lower bounds

Conditional lower bounds?

2-vs.-1 Cycle Problem

Distinguish between:





Conjectured to require $\Omega(\log n)$ rounds in sublinear MPC

Implies immediate hardness for many fundamental problems

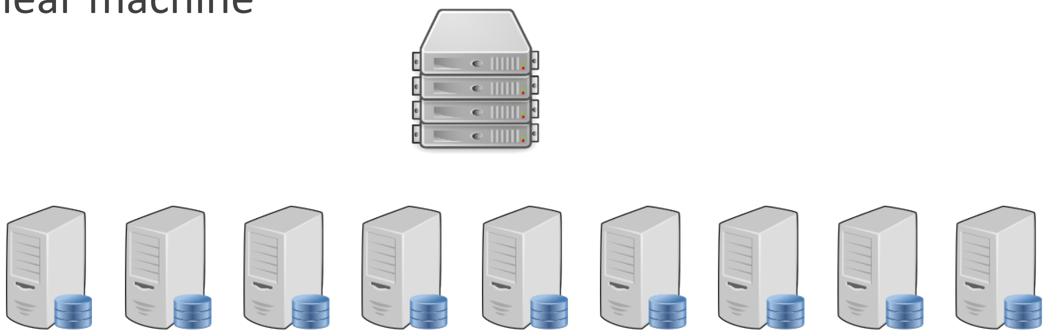
[Ghaffari, Kuhn, Uitto’19], [Czumaj, Davies, Parter’21]: conditional hardness results (approximate max matching / vertex cover, coloring, spanners,…)

Our Question

2-vs.-1 Cycle is easy if we have one near-linear machine

New model: Heterogeneous MPC model (special case):

- N sublinear machines
- 1 near-linear machine

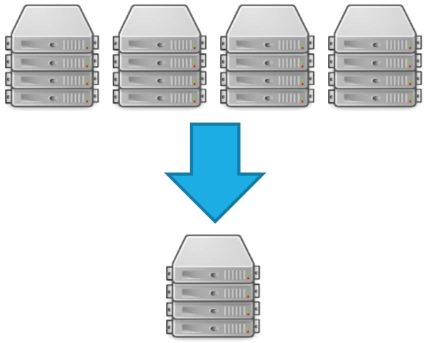


Lower bounds still hold?

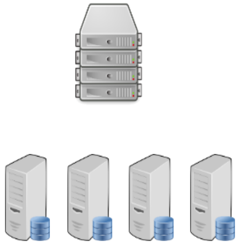
Significantly faster algorithms?

Motivation

Efficiency - only one large machine needed!



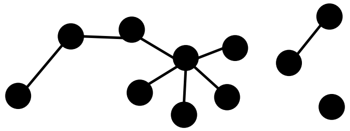
“Minimal” strengthening of sublinear MPC bypasses all lower bounds!



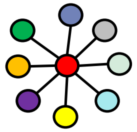
(Implicit) Results from Previous Works

Some State-of-the-Art results for the near-linear regime can be translated directly to HMPC

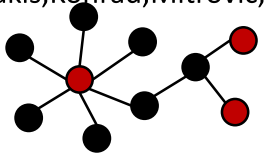
Connectivity – O(1)
[Ahn, Guha, McGregor’12]



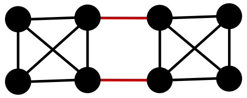
$(\Delta + 1)$-coloring – O(1)
[Assadi,Chen,Khanna’19]

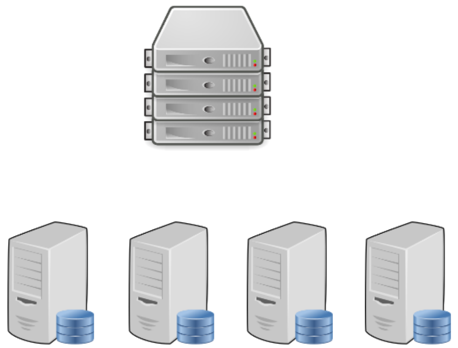


Maximal Independent Set – $O(\log \log \Delta)$
[Ghaffari,Gouleakis,Konrad,Mitrovic,Rubinfeld'18]



Exact Minimum-Cut – O(1)
[Ghaffari,Nowicki,Thorup'20]





Results from [**F.**, Horowitz, Oshman'22]

	Sublinear	**HMPC**	**Near-Linear**
Minimum-weight spanning tree	$O(\log n)$ [ASSWZ'19]	$\boldsymbol{O(\log\log(\frac{m}{n}))}$	$O(1)$ [AGM'12]
$\boldsymbol{O(k)}$**-spanner of size** $\boldsymbol{O(n^{1+1/k})}$	$O(\log k)$ [BDGMN'21] * Stretch $k^{\log 3}$	$\boldsymbol{O(1)}$	$O(1)$ [D**F**KL'21]
Maximal matching	$O(\sqrt{\log\Delta}\log\log\Delta$ $+\sqrt{\log\log n})$ [GU'19]	$\boldsymbol{O\left(\sqrt{\log\frac{m}{n}\log\log\frac{m}{n}}\right)}$	$O(\log\log\Delta)$ [BHH'19]

Plan

Model & Motivation

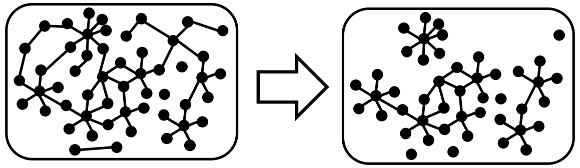
Techniques in HMPC

Prior and New Results

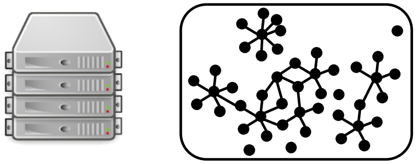
Open Problems

The Simplest Framework

Sparsify to size O(n)

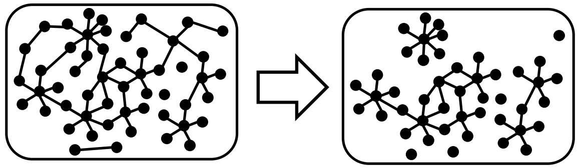


Solve on large machine

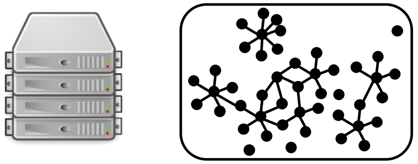


A More General Approach

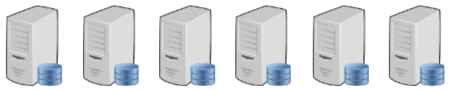
Randomly sample graph of size $O(n)$



Solve on large machine

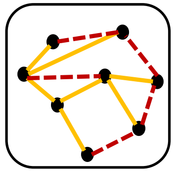


Expand solution to entire graph



A More General Approach

Sampling Lemma



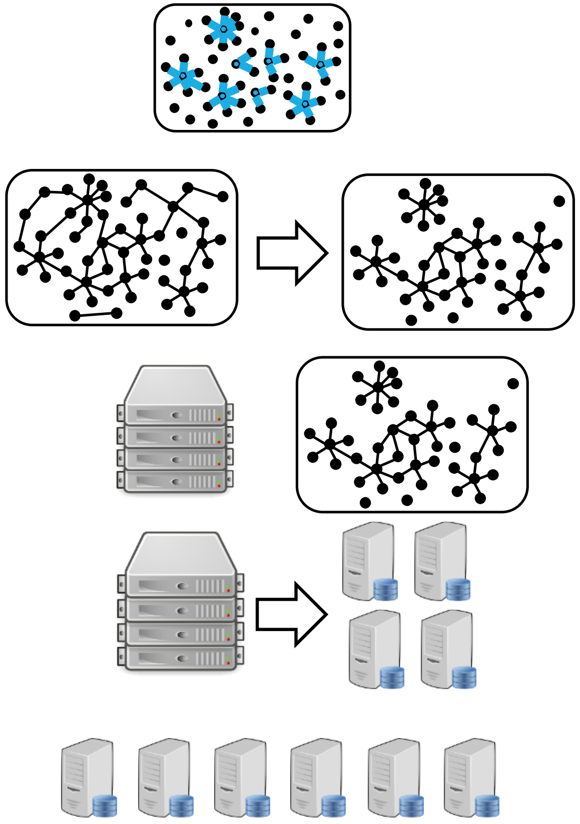
Reduce Size

Sample Random Subgraph
and send to large machine

Local computation on Random Subgraph
Solve related problem on sparsified subgraph

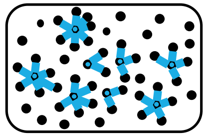
Augment Data
Edges stored on are augmented with global data

Expand
Expand solution to entire graph using augmented data

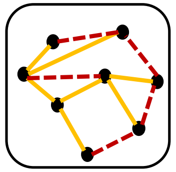


A More General Approach

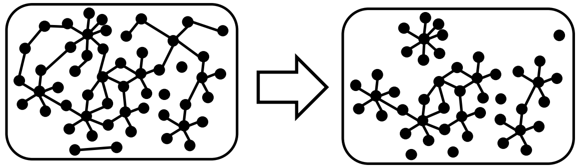
Reduce Size



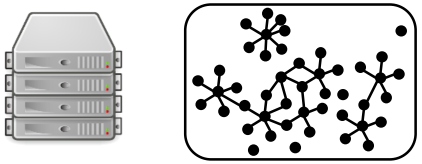
Sampling Lemma



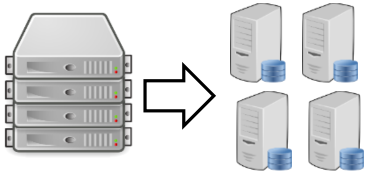
Sample Random Subgraph
and send to large machine



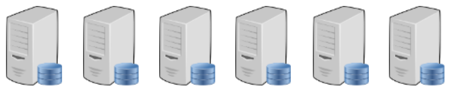
Local computation on Random Subgraph
Solve related problem on sparsified subgraph



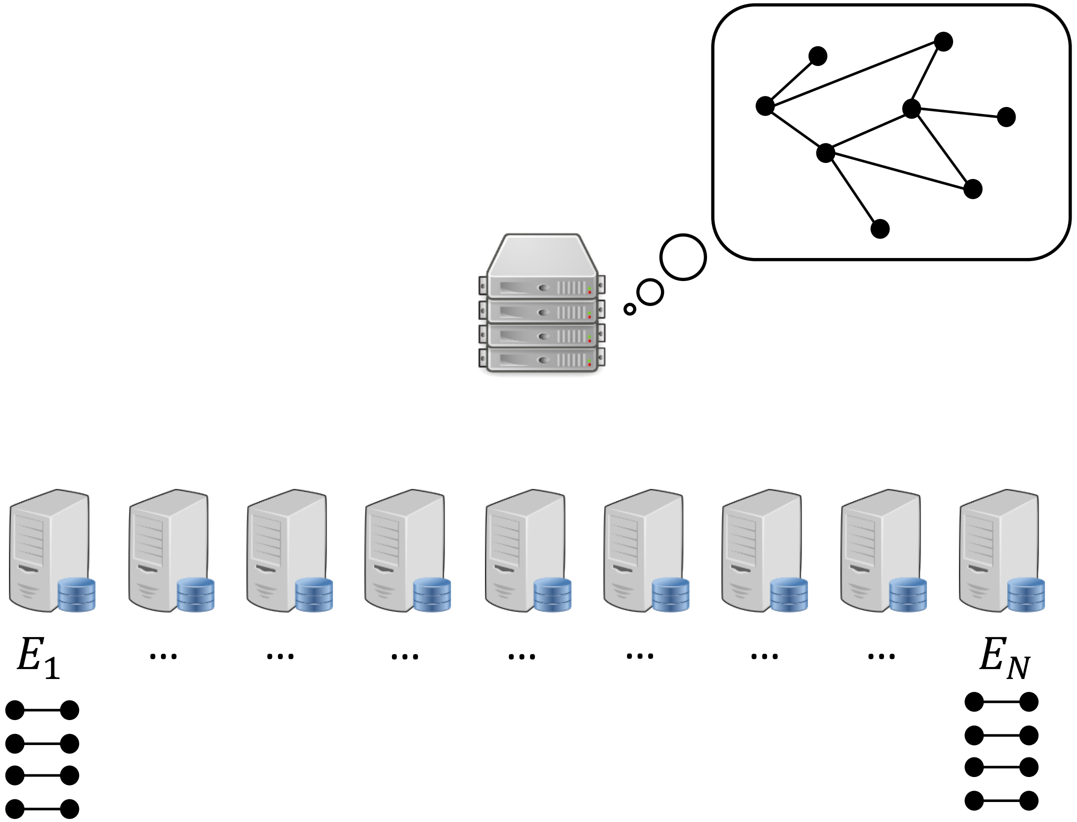
Augment Data
Edges stored on are augmented with global data



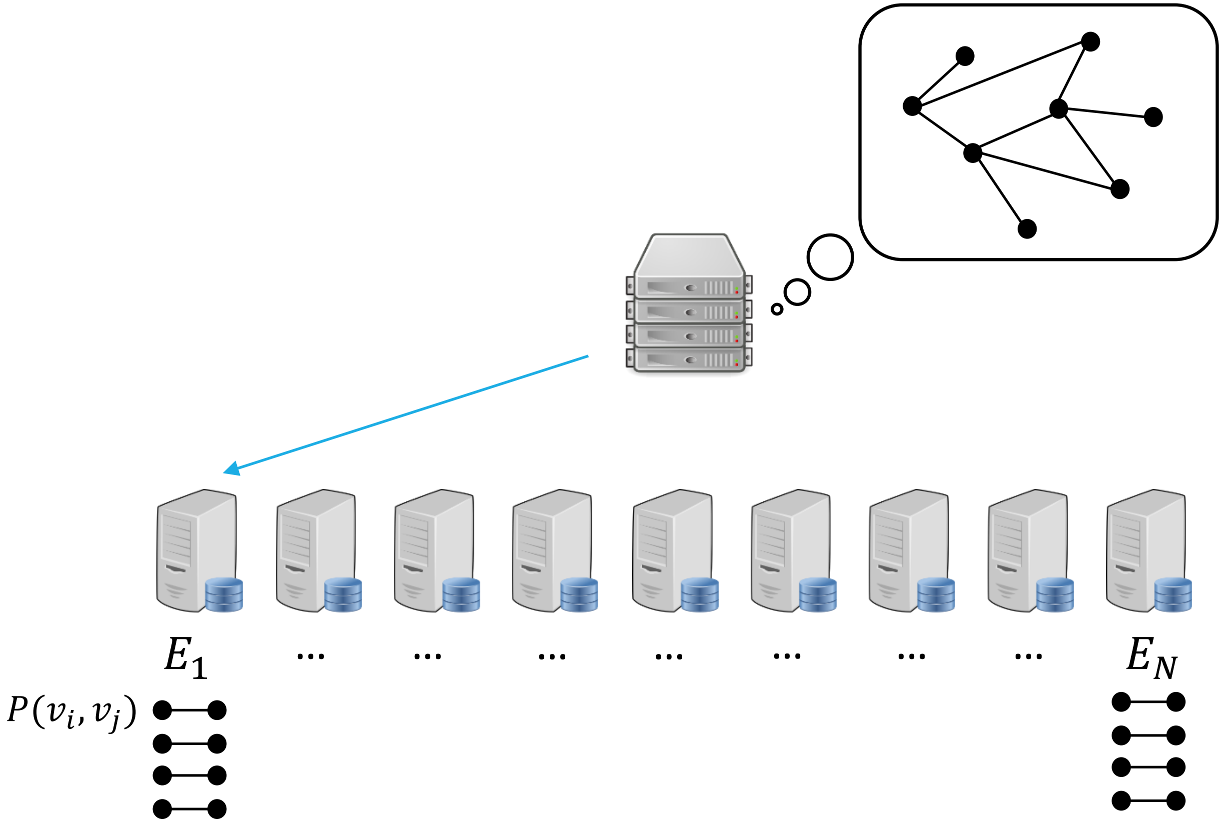
Expand
Expand solution to entire graph using augmented data



Augmenting data & Labeling Schemes



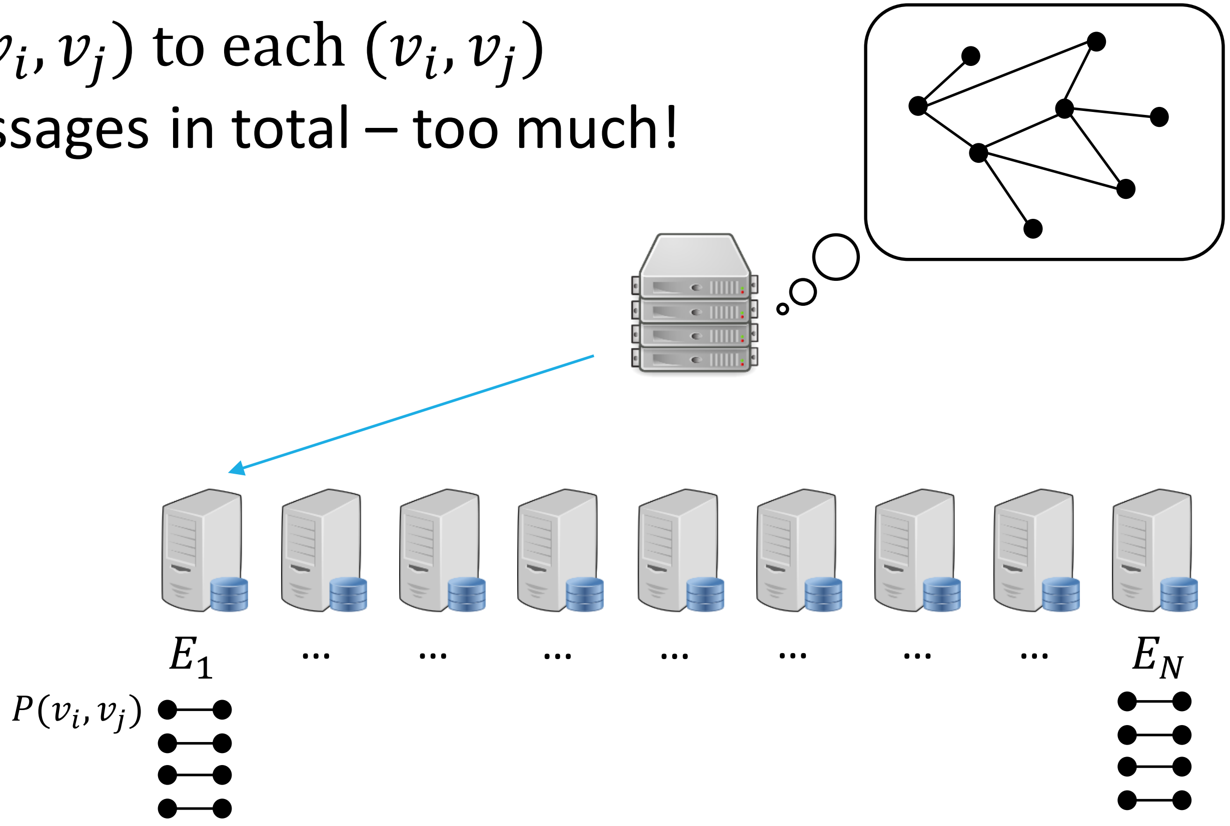
Augmenting data & Labeling Schemes



Augmenting data & Labeling Schemes

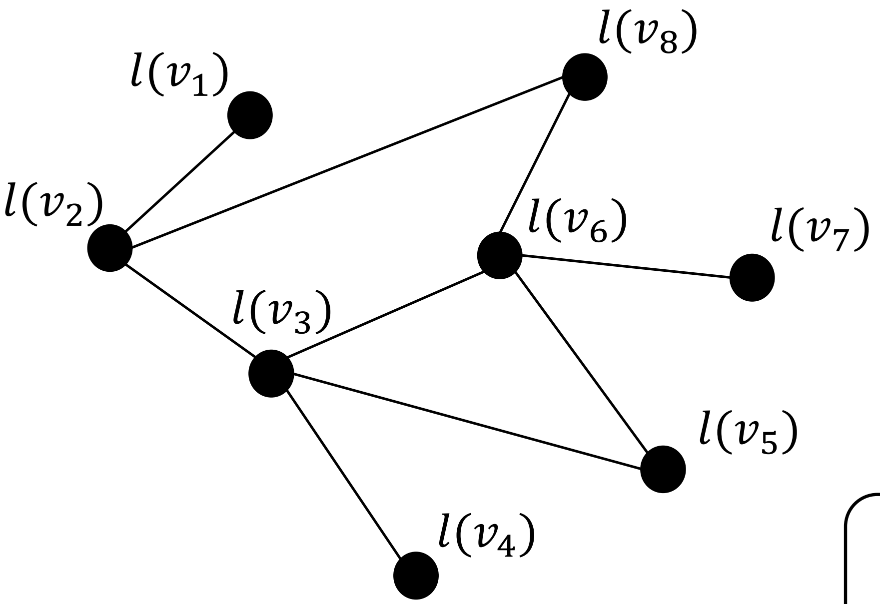
Want: send $P(v_i, v_j)$ to each (v_i, v_j)

Cost: $\Omega(m)$ messages in total – too much!



Labeling Schemes

$P: VxV \to \mathbb{R}$ (e.g. adjacency, distance)



Given $l(v_i), l(v_j)$,
what is $\mathrm{P}(v_i, v_j)$?

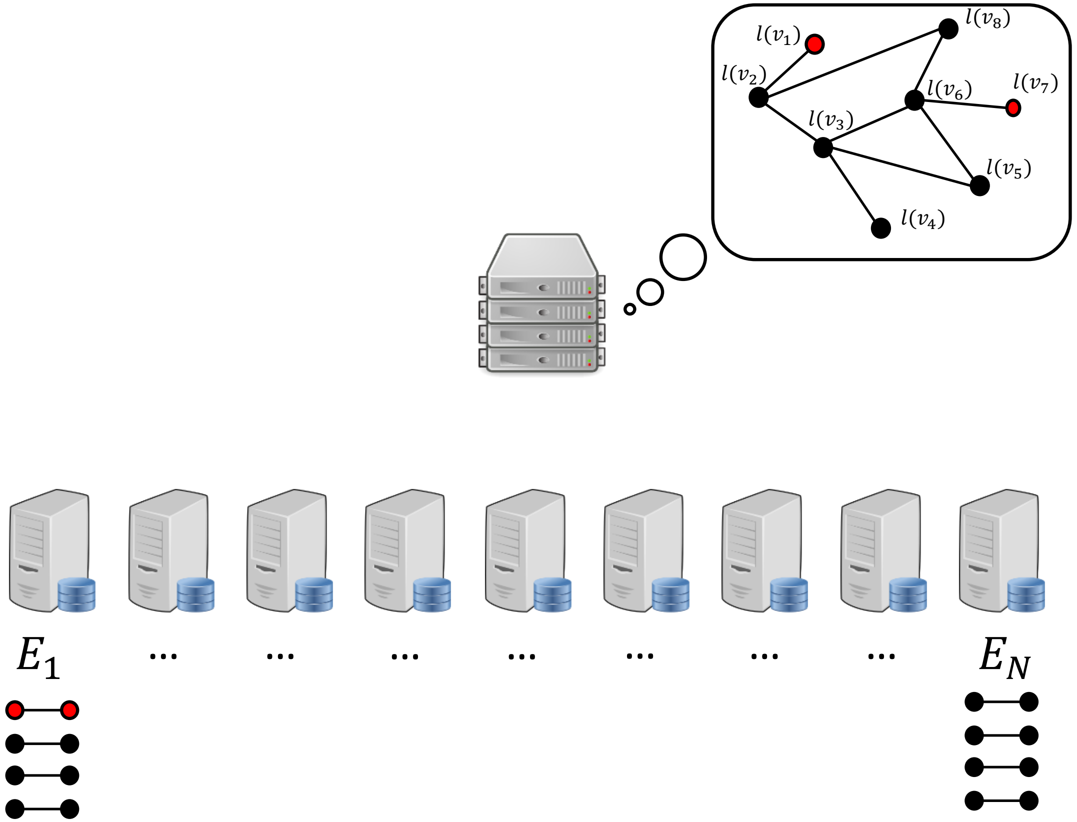
Given $l(v_i), l(v_j)$,
are v_i, v_j adjecent?

Given $l(v_i), l(v_j)$,
What is $\mathrm{dist}(v_i, v_j)$?

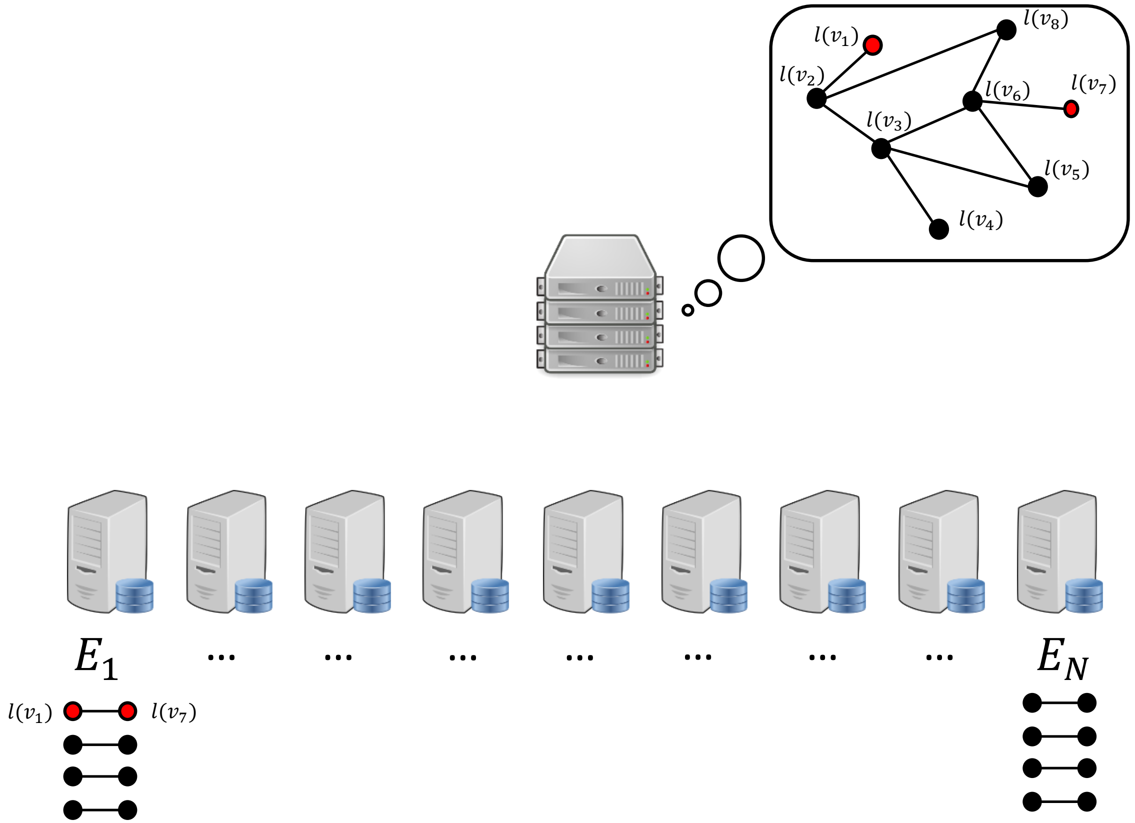
Complexity = label sizes

Efficient = $\mathrm{O}(\mathrm{poly}\log n)$

Augmenting data & Labeling Schemes



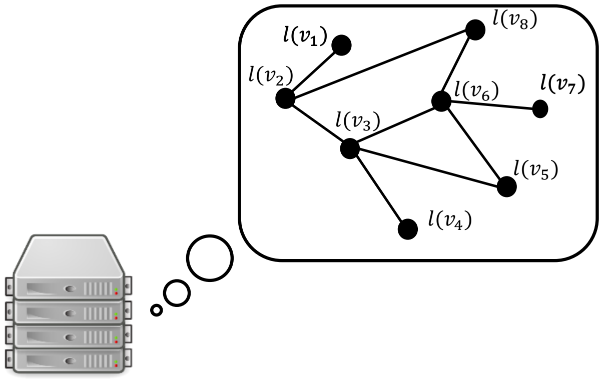
Augmenting data & Labeling Schemes

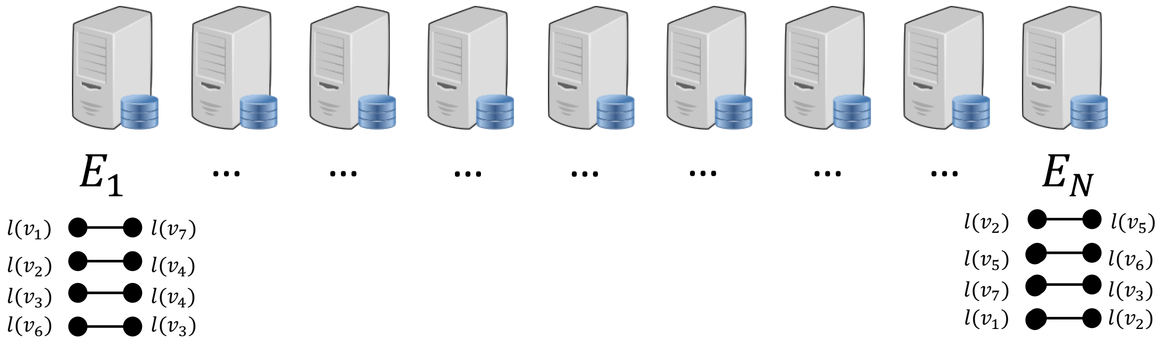


Augmenting data & Labeling Schemes

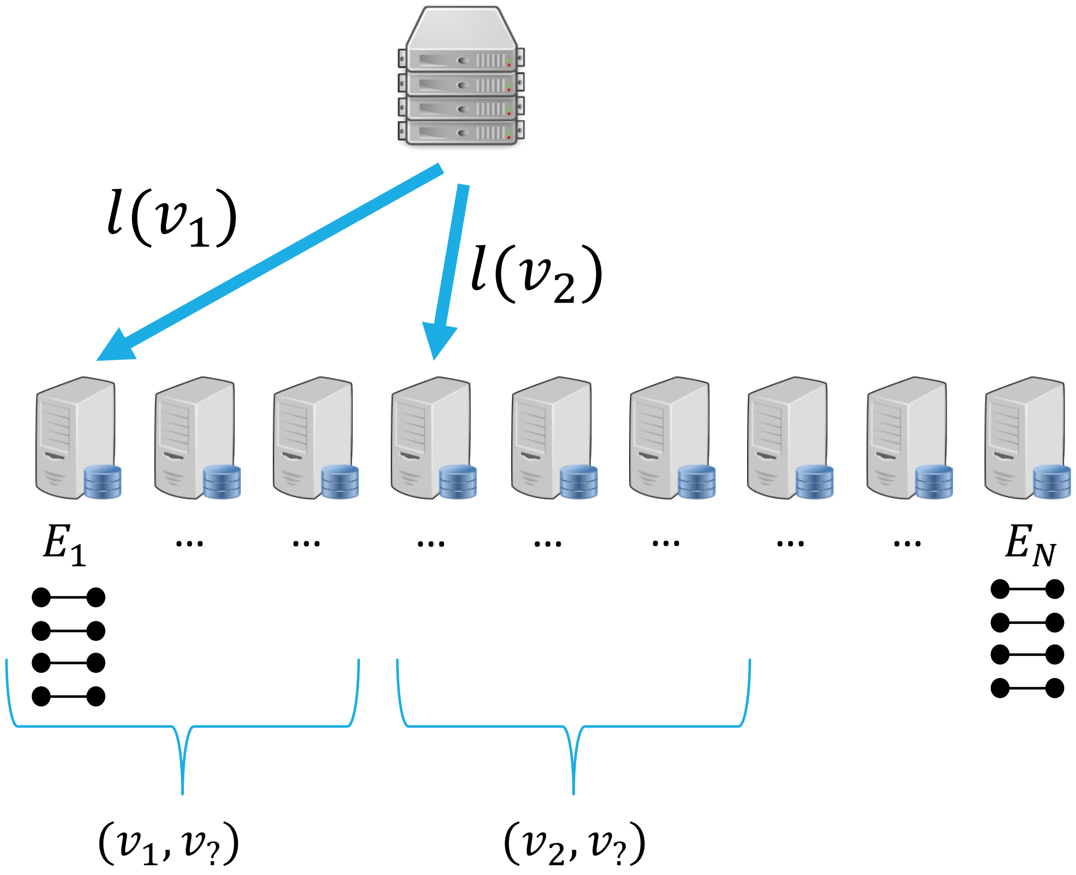
O(1) rounds!

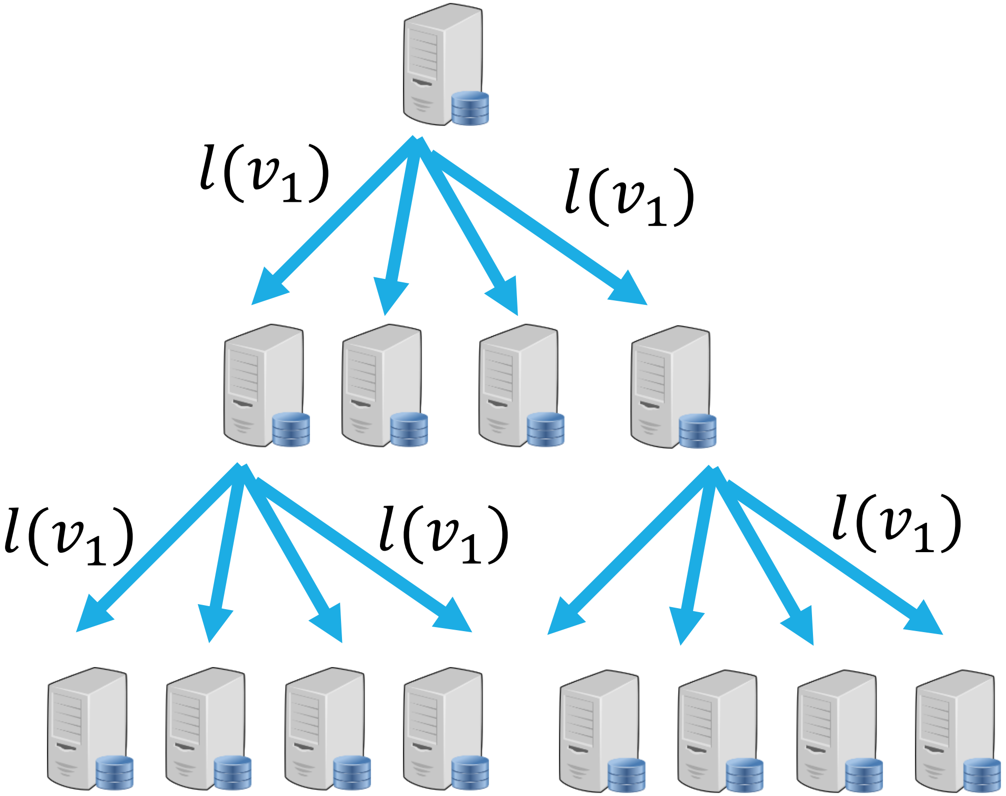
Implementation: sublinear-space sorting alg’ of [Goodrich, Sitchinava and Zhang'11]





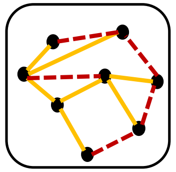
Augmenting data & Labeling Schemes



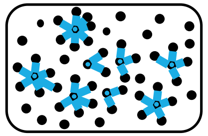


A More General Approach

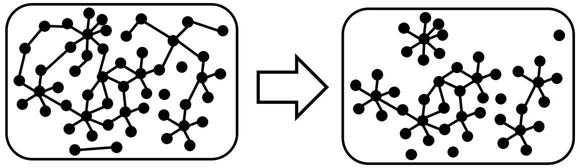
Sampling Lemma



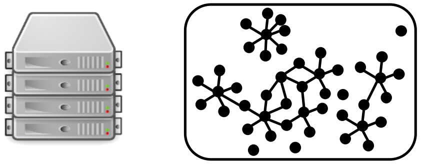
Reduce Size



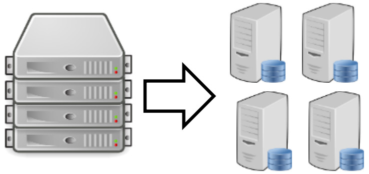
Sample Random Subgraph
and send to large machine



Local computation on Random Subgraph
Solve related problem on sparsified subgraph



Augment Data
Edges stored on are augmented with global data



Expand
Expand solution to entire graph using augmented data

Plan

Model & Motivation

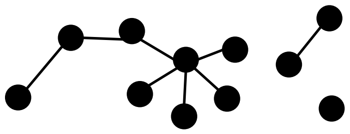
Techniques in HMPC

Prior and New Results

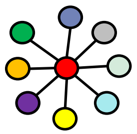
Open Problems

Examples

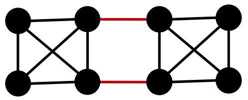
Connectivity – O(1)
[Ahn, Guha, McGregor'12]



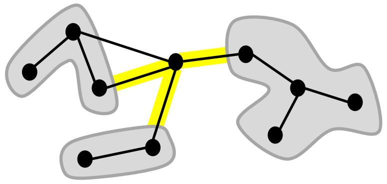
$(\Delta + 1)$-coloring – O(1)
[Assadi,Chen,Khanna'19]



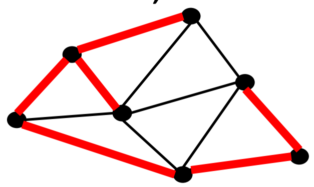
Exact Minimum-Cut – O(1)
[Ghaffari,Nowicki,Thorup'20]



Spanners – O(1)
[F., Horowitz, Oshman'22]

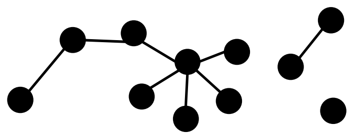


MST – O(loglog(m/n))
[F., Horowitz, Oshman'22]



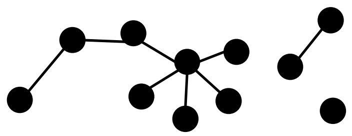
Connectivity

Connectivity – O(1)
[Ahn, Guha, McGregor'12]



Sketching Based

Connectivity – O(1)
[Holm, King, Thorup, Zamir, Zwick'19]

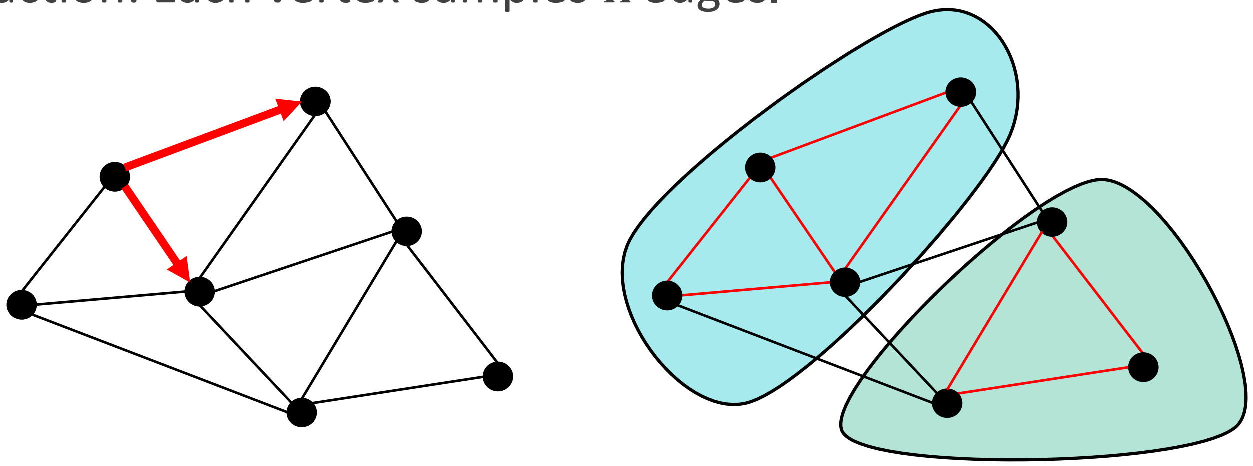


Sampling based

Connectivity Algorithm [Holm, King, Thorup, Zamir, Zwick'19]

k-out-contraction: Each vertex samples k edges.

k=2



Sampling lemma:
For any $k = \Omega(\log n)$, the expected number inter-component edges of a random k-out-contraction is $O(n/k)$.

Connectivity Algorithm [Holm, King, Thorup, Zamir, Zwick’19]

Sampling lemma:

For any $k = \Omega(\log n)$, the expected number inter-component edges of a random k-out-contraction is $O(n/k)$.

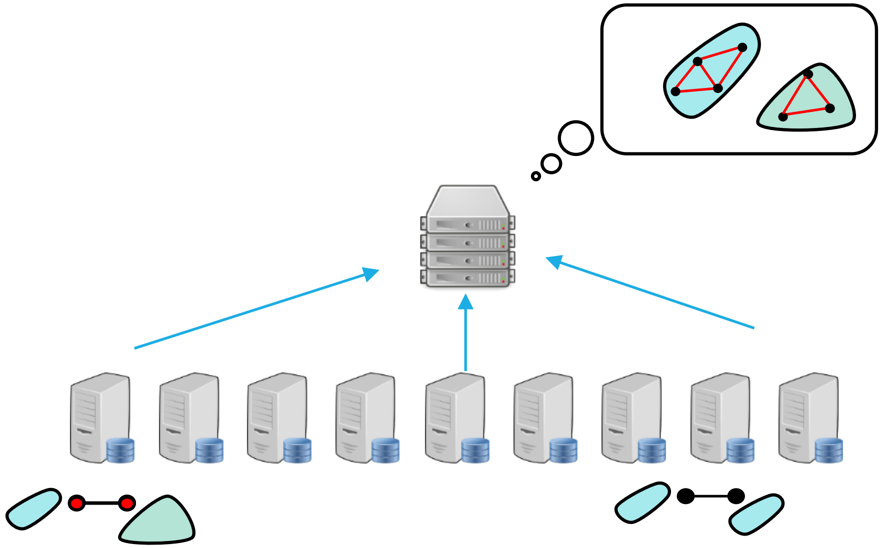
Algorithm:

(a) Send to  $\Theta(\log n)$ random edges from each $v \in V$

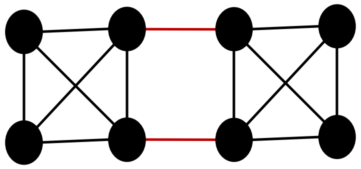
(b)  computes the C.C of G’. Augments information of 

(c)  locally marks inter-component edges and sends to 

(d)  outputs connected components



Exact Minimum-Cut – O(1) [Ghaffari,Nowicki,Thorup'20]

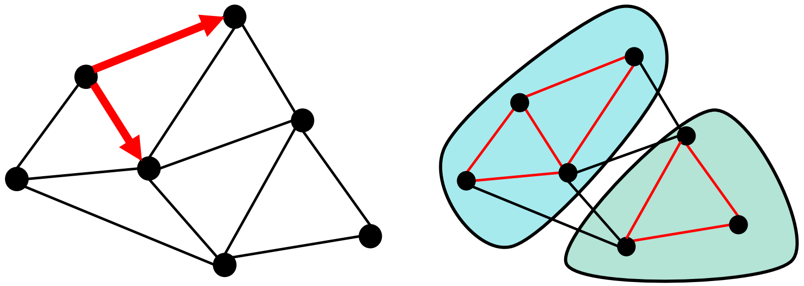


Sparsification lemma:

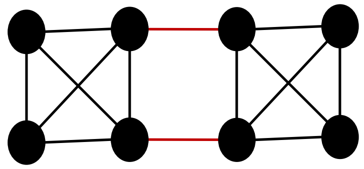
(1) 2-out-contractions reduce number of **vertices** to $O(n/\delta)$

(2) Contracting the random graph $E_{1/2\delta}$ reduces number of **edges** to $O(n\delta)$

Both do not affect the min-cut with good probability *



Exact Minimum-Cut – O(1) [Ghaffari,Nowicki,Thorup'20]

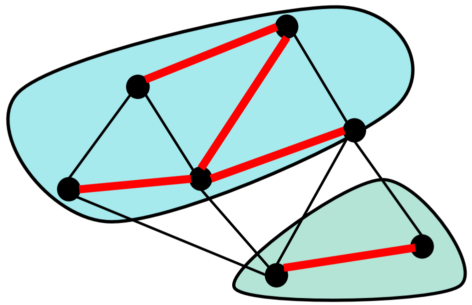


Sparsification lemma:

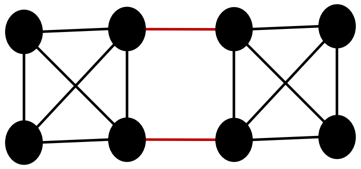
(1) 2-out-contractions reduce number of **vertices** to $O(n/\delta)$

(2) Contracting the random graph $E_{1/2\delta}$ reduces number of **edges** to $O(n\delta)$

Both do not affect the min-cut with good probability *



Exact Minimum-Cut – O(1) [Ghaffari,Nowicki,Thorup'20]



Sparsification lemma:

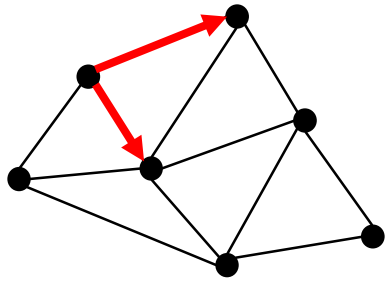
(1) 2-out-contractions reduce number of **vertices** to $O(n/\delta)$

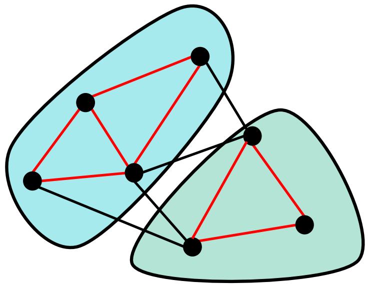
(2) Contracting the random graph $E_{1/2\delta}$ reduces number of **edges** to $O(n\delta)$

Both do not affect the min-cut with good probability *

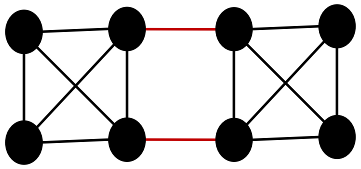
Algorithm:

(a) Apply (1) to reduce number of vertices to $O(n/\delta)$





Exact Minimum-Cut – O(1) [Ghaffari,Nowicki,Thorup'20]



Sparsification lemma:

(1) 2-out-contractions reduce number of **vertices** to $O(n/\delta)$

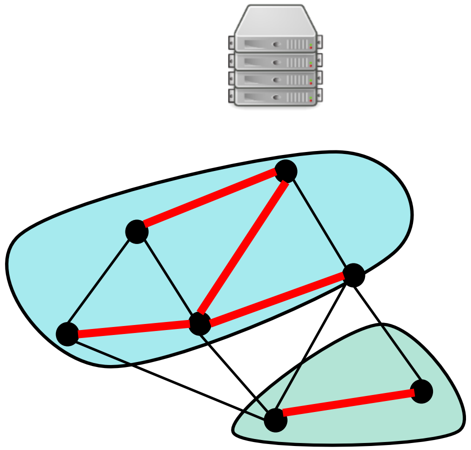
(2) Contracting the random graph $E_{1/2\delta}$ reduces number of **edges** to $O(n\delta)$

Both do not affect the min-cut with good probability *

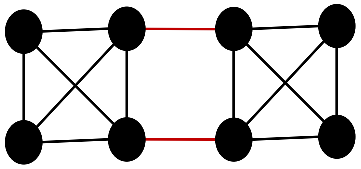
Algorithm:

(a) Apply (1) to reduce number of vertices to $O(n/\delta)$

(b) Apply (2) to reduce number of edges to $O\left(\frac{n}{\delta} \cdot \delta\right) = O(n)$



Exact Minimum-Cut – O(1) [Ghaffari,Nowicki,Thorup'20]



Sparsification lemma:
(1) 2-out-contractions reduce number of **vertices** to $O(n/\delta)$
(2) Contracting the random graph $E_{1/2\delta}$ reduces number of **edges** to $O(n\delta)$
Both do not affect the min-cut with good probability *

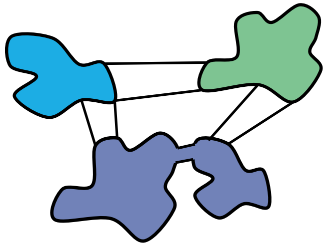
Algorithm:

(a) Apply (1) to reduce number of vertices to $O(n/\delta)$

(b) Apply (2) to reduce number of edges to $O\left(\frac{n}{\delta} \cdot \delta\right) = O(n)$

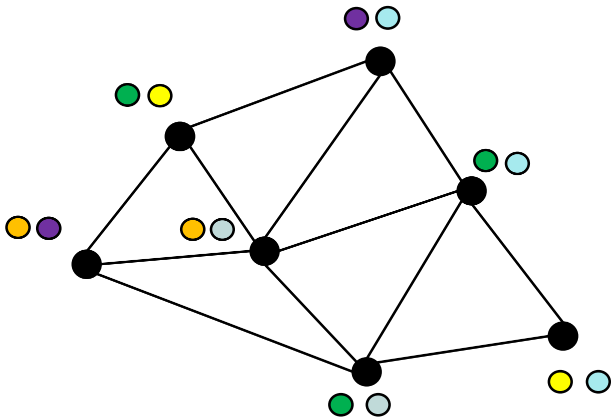
(c) Send all edges to 

(d)  outputs the min-cut of the graph

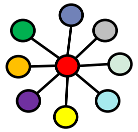


$(\Delta + 1)$-coloring [Assadi,Chen,Khanna'19]

Sampling lemma:
If each $v \in V$ chooses a random set $\mathrm{C(v)} \subseteq \{1,2,\ldots,\Delta+1\}$ of size $\Theta(\mathrm{polylog}\, n)$,
then w.h.p. there is a proper coloring such that each vertex is colored from $\mathrm{C(v)}$



Each edge remains w.p. $p \leq \mathrm{polylog}\, n\,/\Delta$
=> Remaining graph is of size $O(n\, \mathrm{polylog}\, n)$

$(\Delta + 1)$-coloring [Assadi,Chen,Khanna'19] 

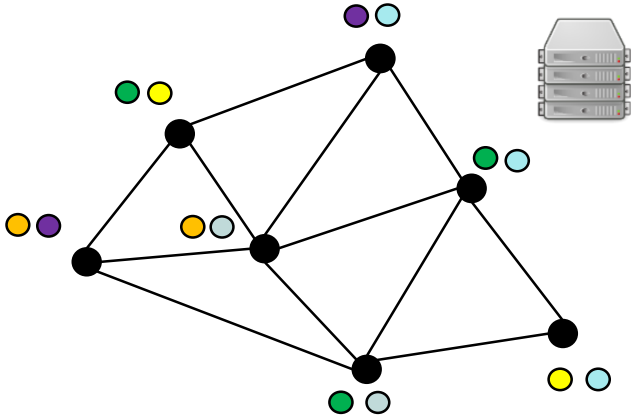
Sampling lemma:
If each $v \in V$ chooses a random set $\mathrm{C(v)} \subseteq \{1,2,\ldots,\Delta+1\}$ of size $\Theta(\mathrm{polylog}\, n)$,
then w.h.p. there is a proper coloring such that each vertex is colored from $\mathrm{C(v)}$

Algorithm:

(a) $\forall v \in V$,  samples $\mathrm{C(v)} \subseteq \{1,2,\ldots,\Delta+1\}$ of size $\Theta(\mathrm{polylog}\, n)$

(b) Let $E' = \{\{u,v\} \mid C(u) \cap C(v) \neq \emptyset\}$. Send E' to 

(c) Have  output a proper coloring

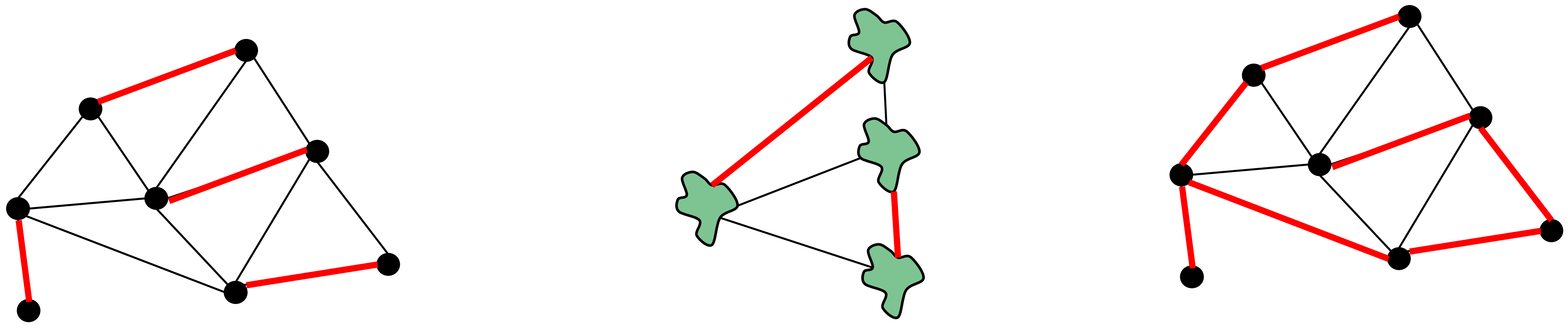


Next: Algorithms from [**F.**, Horowitz, Oshman'22]

	Sublinear	HMPC	Near-Linear
Minimum-weight spanning tree	$O(\log n)$ [ASSWZ'19]	$\boldsymbol{O(\log\log(\frac{m}{n}))}$	$O(1)$ [AGM'12]
$\boldsymbol{O(k)}$**-spanner of size** $\boldsymbol{O\left(n^{1+1/k}\right)}$	$O(\log k)$ [BDGMN'21] * Stretch $k^{\log 3}$	$\boldsymbol{O(1)}$	$O(1)$ [D**F**KL'21]
Maximal matching	$O(\sqrt{\log\Delta}\log\log\Delta + \sqrt{\log\log n})$ [GU'19]	$\boldsymbol{O\left(\sqrt{\log\frac{m}{n}}\log\log\frac{m}{n}\right)}$	$O(\log\log\Delta)$ [BHH'19]

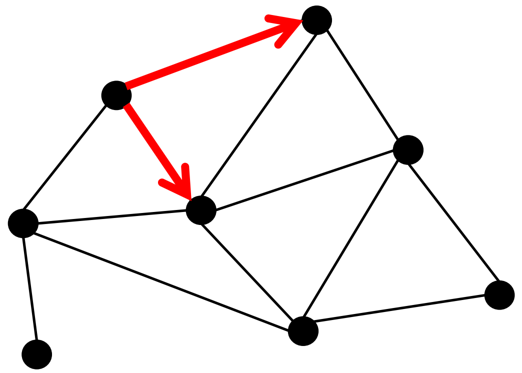
MST Algorithm Overview

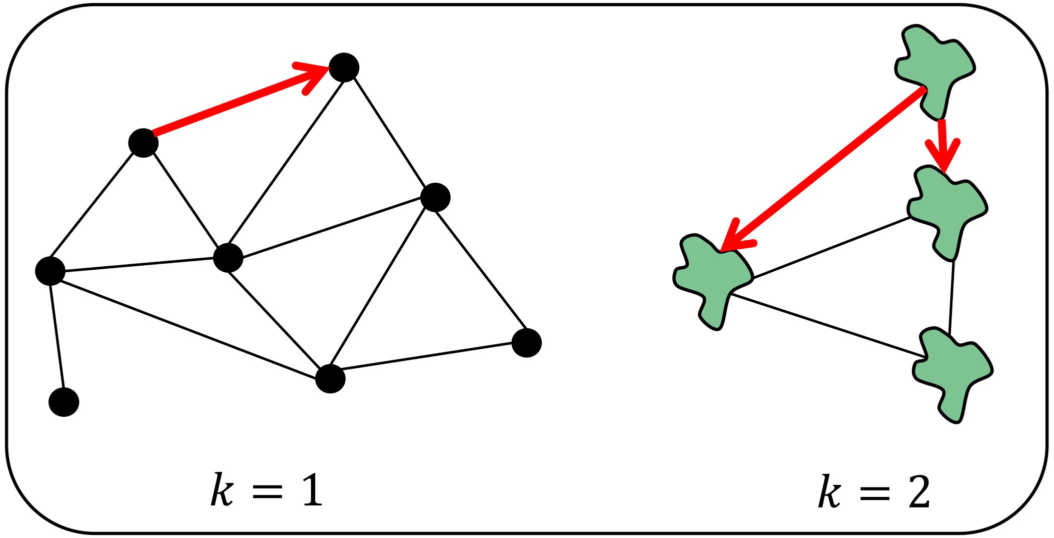
Borůvka:



MST Algorithm Overview

Doubly exponential Borůvka [Lotker, Pavlov, Patt-Shamir, Peleg'03]:

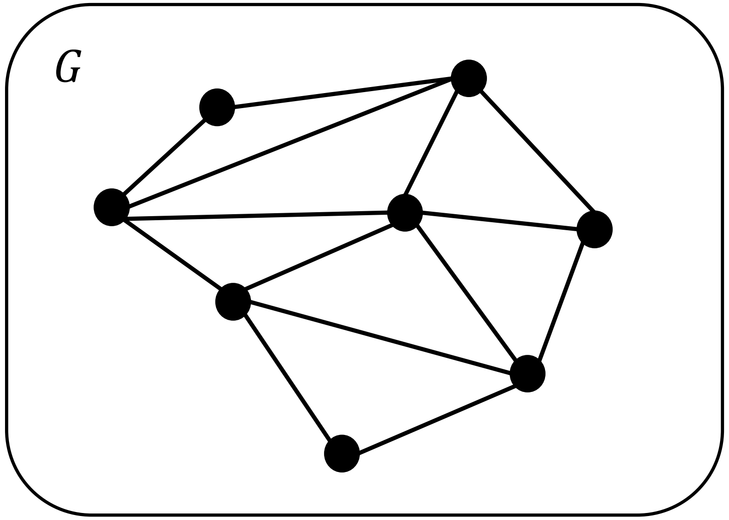


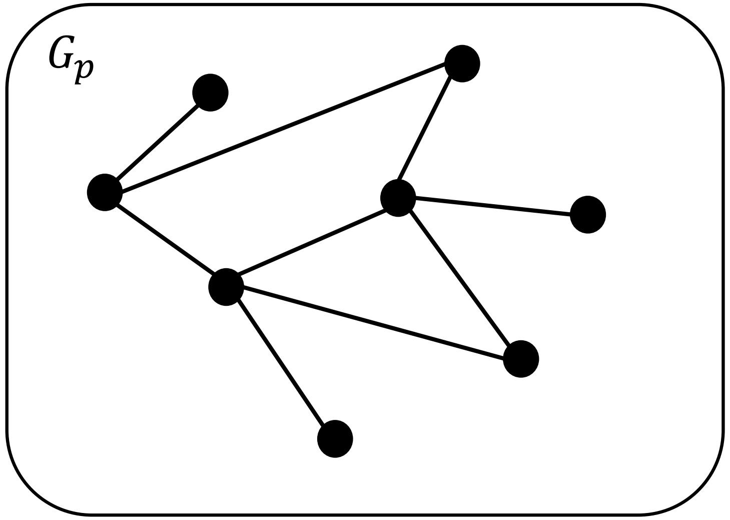


$|F| \rightarrow |F|/(k+1)$

After x iterations: reduce to $\frac{\mathrm{n}}{2^{2^x}}$ components

MST Algorithm Sampling Lemma (KKT'95)





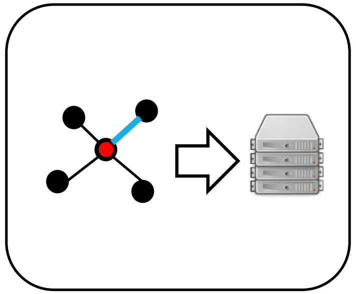
Edge is heavy if weight( ) > weight( ) for all in the path of the tree between its two endpoints

Sampling Lemma [Karger, Klein, Tarjan'95]: There are at $\leq n/p$ edges which are light in *G*
(In expectation)

MST Algorithm Overview

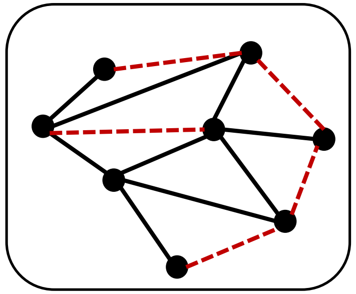
Sampling Lemma [Karger, Klein, Tarjan'95]: There are at $\leq n/p$ edges which are light in G

Reduce Size



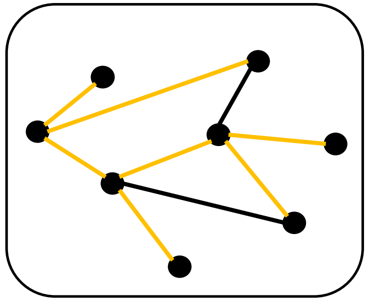
Doubly exponential Borůvka
Reduce to n/d vertices in
$O(\log \log d)$ rounds

Sample



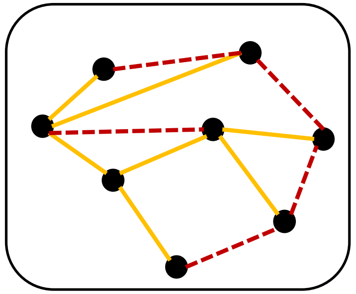
Take every edge
w.p. $1/d$.
Send to 

Compute & Augment



Solve MST on G_p
$P(v_i, v_j)$ = heaviest
edge between v_i, v_j
In MST(G_p)

Expand



Find light edges in
G using labels

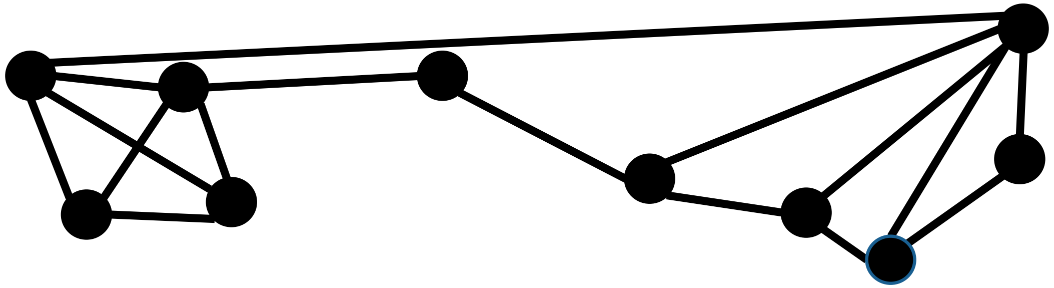
$O(n)$ edges remaining,
Solve in large machine

Next: Algorithms from [**F.**, Horowitz, Oshman'22]

	Sublinear	HMPC	Near-Linear
Minimum-weight spanning tree	$O(\log n)$ [ASSWZ'19]	$\boldsymbol{O(\log\log(\frac{m}{n}))}$	$O(1)$ [AGM'12]
$\boldsymbol{O(k)}$**-spanner of size** $\boldsymbol{O\left(n^{1+1/k}\right)}$	$O(\log k)$ [BDGMN'21] * Stretch $k^{\log 3}$	$\boldsymbol{O(1)}$	$O(1)$ [D**F**KL'21]
Maximal matching	$O(\sqrt{\log\Delta}\log\log\Delta + \sqrt{\log\log n})$ [GU'19]	$\boldsymbol{O\left(\sqrt{\log\frac{m}{n}}\log\log\frac{m}{n}\right)}$	$O(\log\log\Delta)$ [BHH'19]

Spanners

A spanner is a set of edges $H \subseteq E$ of small size that approximately maintains distances of the original graph.



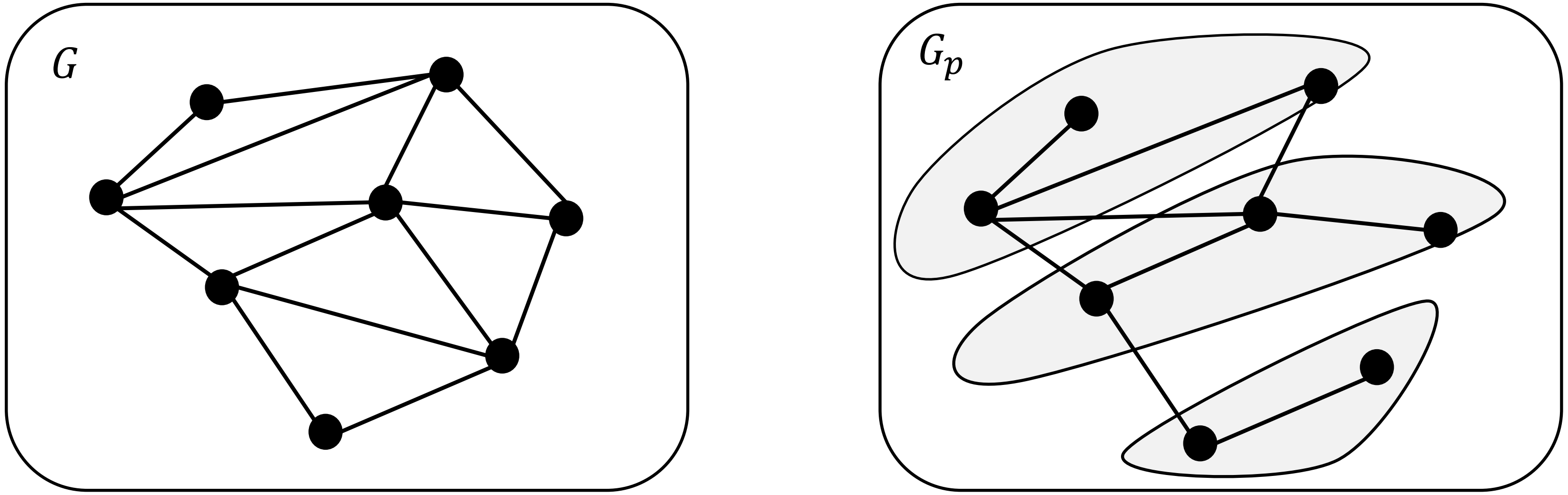
-edges form a 2-spanner

Good Parameters:

Size: $|H| = O(n^{1+1/k})$

Approximation: $\forall_{u,v} \mathrm{dist}_G(u,v) \leq \mathrm{dist}_H(u,v) \leq (2k-1) \cdot \mathrm{dist}_G(u,v)$

Towards a Sampling Lemma for Spanners



Size: $O(n^{1+\frac{1}{k}}/p)$

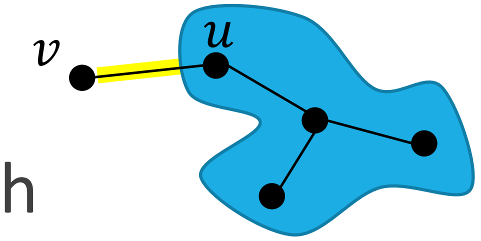
The Baswana-Sen Spanner

For $\ell = 1, \dots, k$:

- Each cluster survives w.p. $1/n^{1/k}$

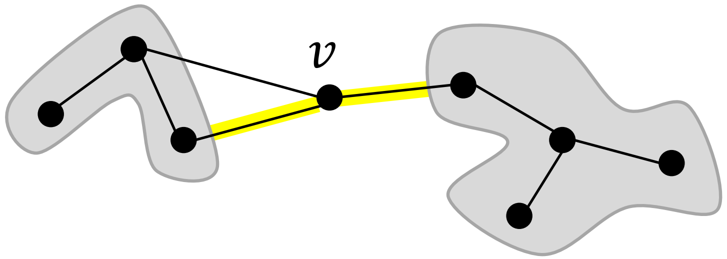
$\boxed{C_k = \emptyset}$

- If v's cluster is destroyed:
  - If ∃neighbor u in surviving cluster: assign v to u's cluster



+ add $\{u, v\}$ to spanner

  - Else: remove v from the graph



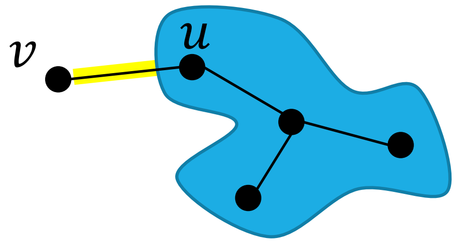
+ add one edge to each previous adjacent cluster

Analysis: Size of the Spanner

Edges are added when:

Node v is re-clustered:

- At most once per level $\Rightarrow O(k)$ edges total

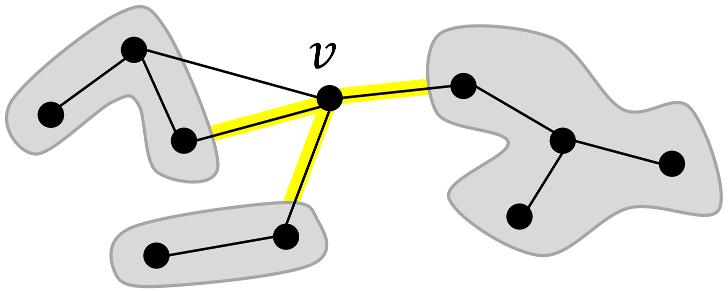


Node v is removed:

- No adjacent cluster survived

$\Rightarrow v$ was adjacent to $\mathrm{O}\left(n^{1/k}\right)$ clusters (w.h.p.)

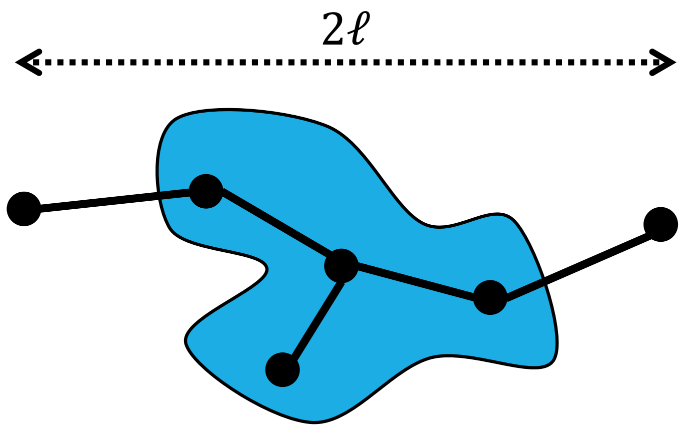
$\Rightarrow O\left(n^{1/k}\right)$ edges added (one per adjacent cluster)



Total: $O\left(n^{1+1/k}\right)$

Analysis: Stretch

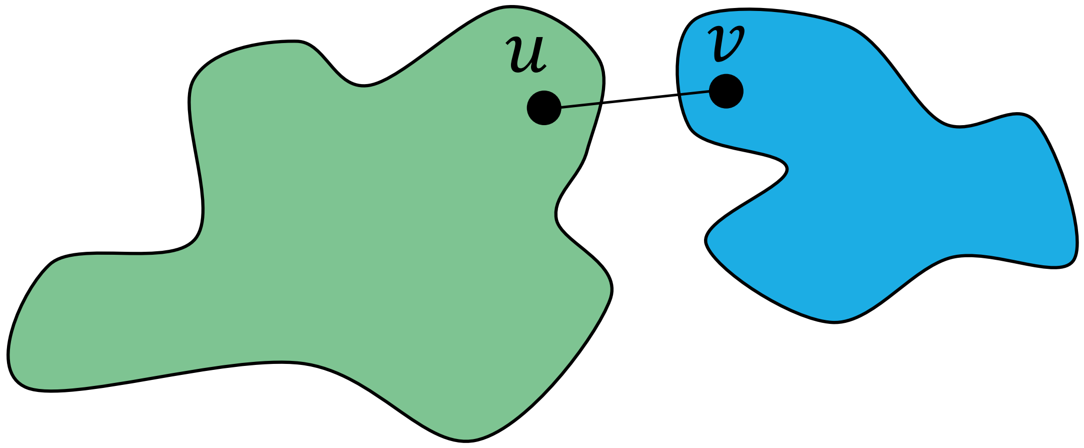
At level ℓ: cluster diameter $\leq 2\ell$



Let $\{u, v\} \in E$

Suppose u removed no later than v :

Level $\ell \leq k - 1$:

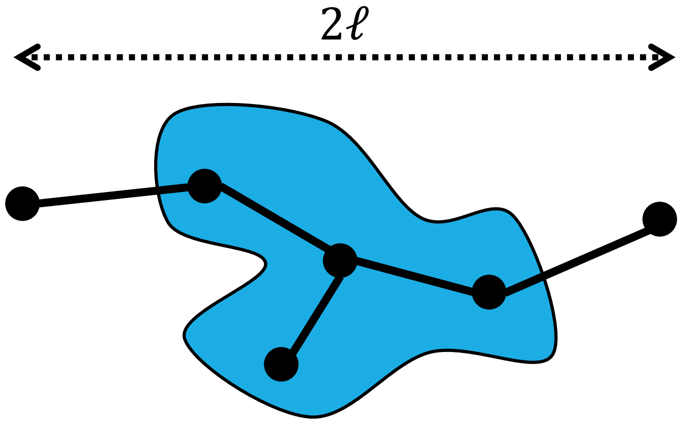


Analysis: Stretch

At level ℓ: cluster diameter $\leq 2\ell$

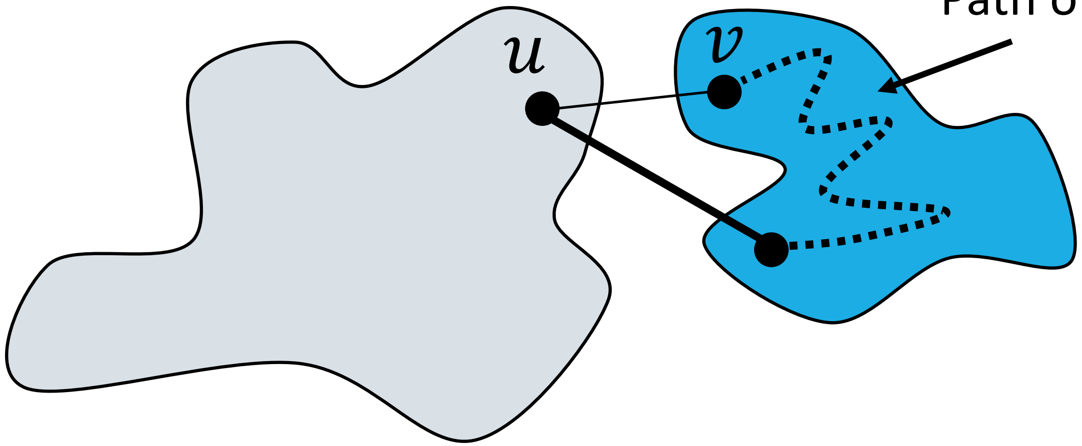
Let $\{u, v\} \in E$

Suppose u removed no later than v :



Level $\ell \leq k-1$:

Path of length $\leq 2\ell \leq 2(k-1)$



$\Rightarrow$ Stretch $\leq 2k-1$

Towards an HMPC Implementation

The large machine can't hold G

Sub-sample edges of $G \Rightarrow G_p$

For $\ell = 1, \dots, k$:

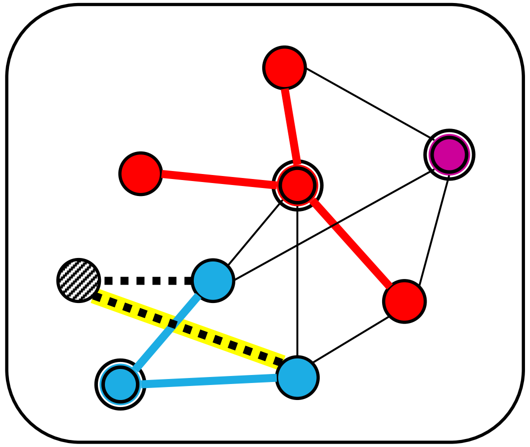
- Each center survives w.p. $1/n^{1/k}$ *
- If v's cluster died:
  - If ∃neighbor u in surviving cluster: assign v to u's cluster
    - Add $\{u, v\}$ to spanner
  - Else: remove v from the graph
    - Add one edge to each previous adjacent cluster

Large machine, on G_p

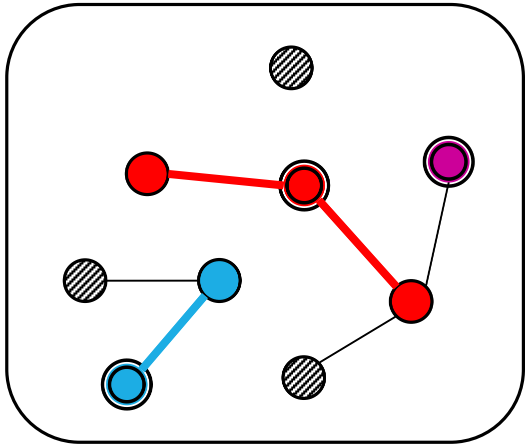
Small machines, on G

True vs. Sub-Sampled Baswana-Sen

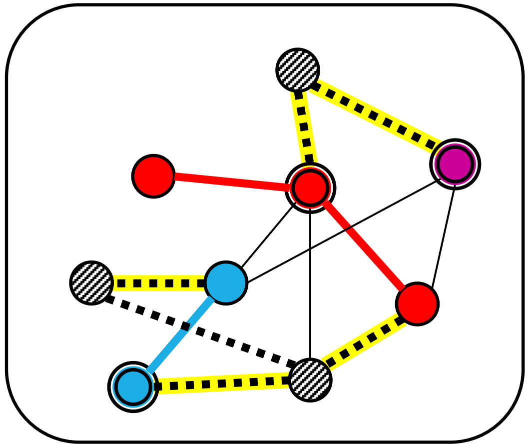
Level 1:



True Baswana-Sen



Perspective of the large machine



Perspective of the small machines

Sub-sampled Baswana-Sen

Sub-Sampled Baswana-Sen

Stretch: unchanged – depends on

- Cluster diameter $\leq 2k$
- Adding edges to all adjacent clusters upon removal

Size?

Analysis: Size of the Spanner

Edges are added when:

Node v is re-clustered:

- At most once per level $\Rightarrow O(k)$ edges total

✓

Node v is removed:

in G_p

- No adjacent cluster survived

$\Rightarrow v$ was adjacent to $O(n^{1/k})$ clusters (w.h.p.)

$\Rightarrow \cancel{O(n^{1/k})}$ edges added (one per adjacent cluster)

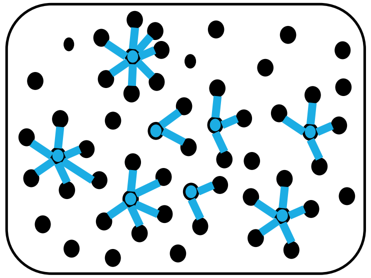
$O(n^{1/k}/p)$ clusters in G

in G

Total: $O(n^{1+1/k}/p)$

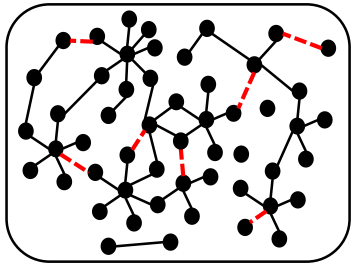
Spanner Algorithm Overview

Reduce Size



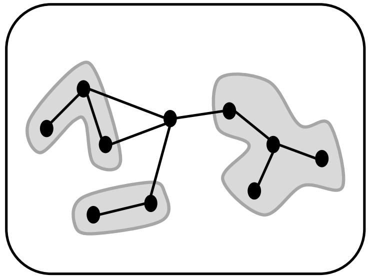
Reduce to n/d vertices using star contraction (Like [DFKL'21])

Sample



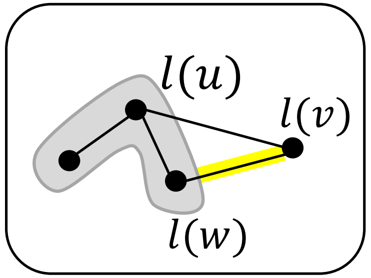
Take every edge w.p. $1/d$.
Send to large Machine 

Compute & Augment



Baswana-Sen on G_p. Send $l(v_i), l(v_j)$ inter-cluster labels

Expand



Using labels, find added edges between clusters

Plan

Model & Motivation

Techniques in HMPC

Prior and New Results

Open Problems

Open problems

O(1)-round MST algorithm?

In near-linear MPC: MST $\rightarrow m/n$ instances of Connectivity

Is MST really as hard as many independent instances of Connectivity?

Open problems

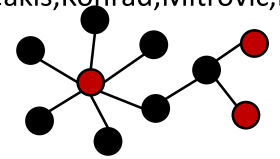
O($\log\log\Delta$)-round Maximal Matching algorithm?

Intuition: MIS ≥ MM in most distributed models

In near-linear MPC: both MIS and MM in O($\log\log\Delta$) rounds

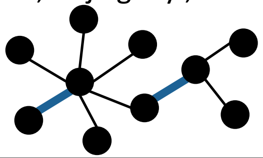
Maximal Independent Set – $O(\log\log\Delta)$

[Ghaffari,Gouleakis,Konrad,Mitrovic,Rubinfeld'18]



Maximal Matching in $O(\log\log\Delta)$

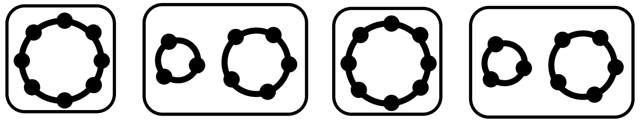
[Behnezhad, Hajiaghayi, Harris'19]

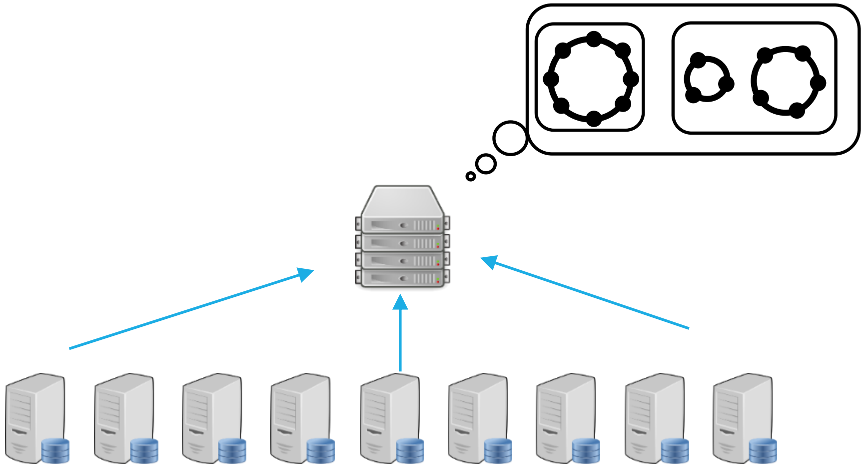


Open problems – Conditional Lower Bounds

Conditional Lower Bounds?

Possible candidate: Many 2-vs-1 cycles?

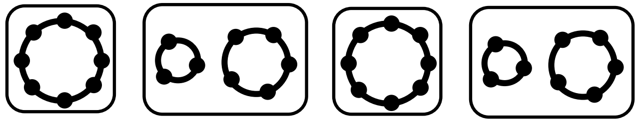


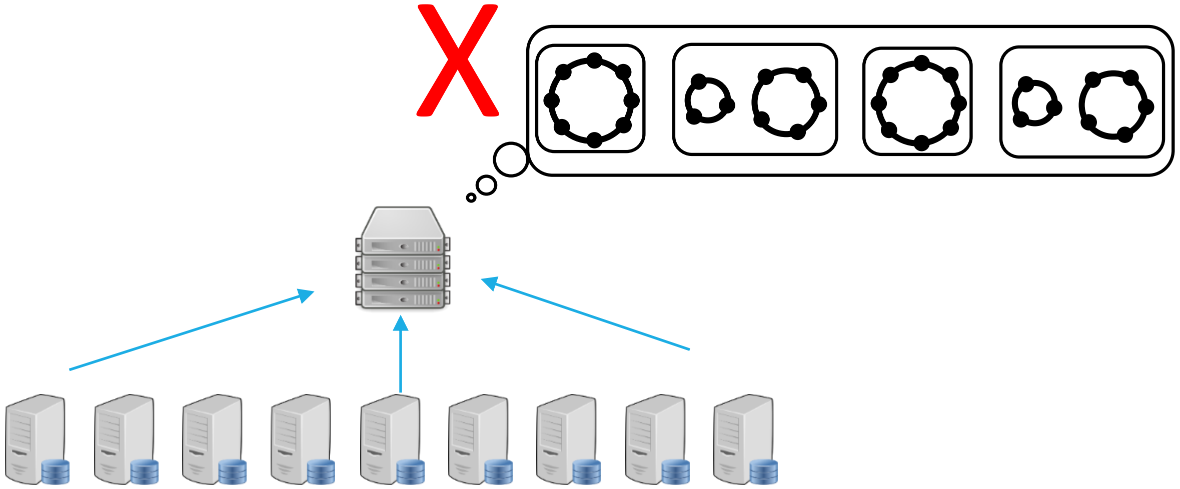


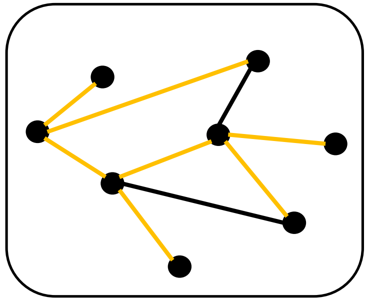
Open problems – Conditional Lower Bounds

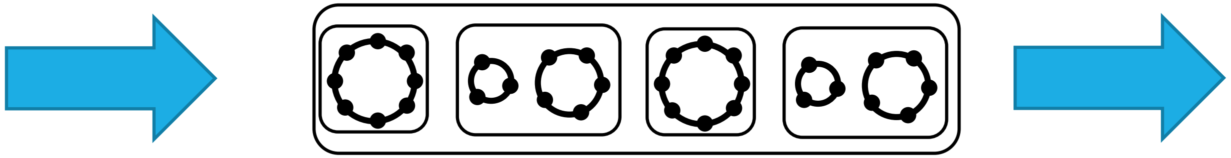
Conditional Lower Bounds?

Possible candidate: Many 2-vs-1 cycles?



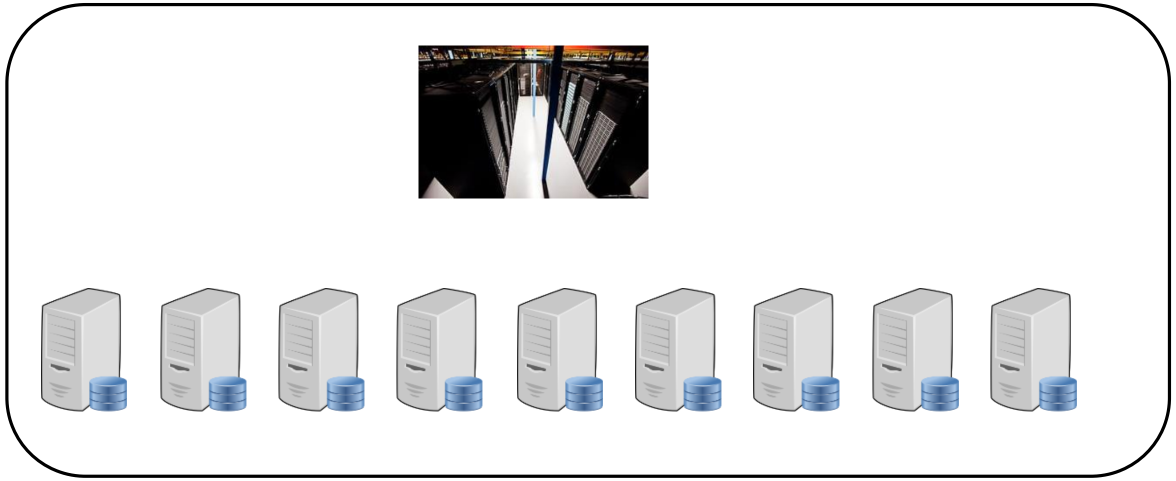


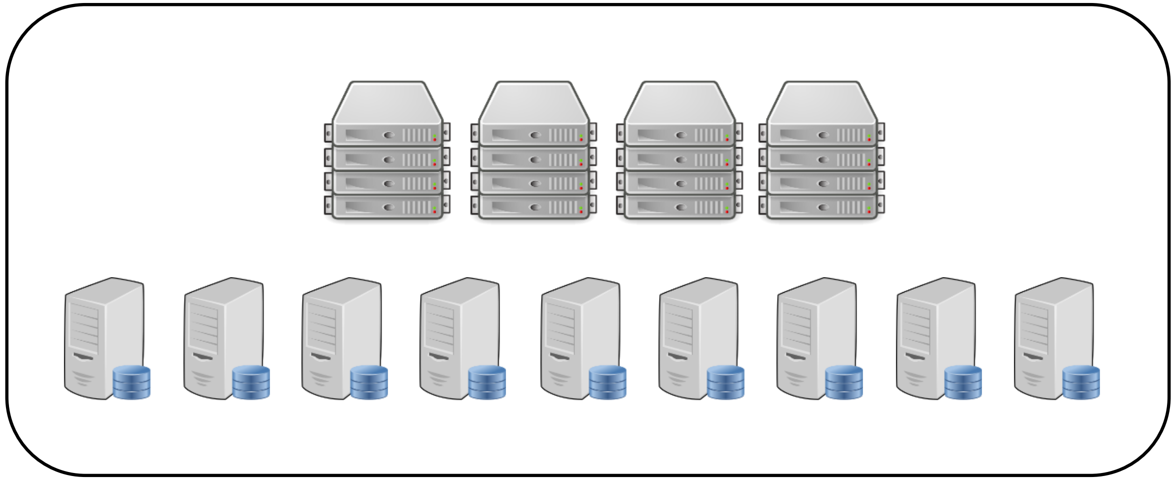




Hardness

Open problems – Results in Generalized HMPC





Super-linear total memory?

Open problems – Deterministic Algorithms

Can we get any speedup compared to sublinear-MPC?

O(1)-round connectivity algorithm

Can other algorithms be derandomized as well?

Inherent deterministic technique?

Conclusions, Extensions & Open Problems

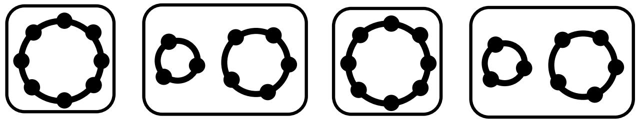
Conclusion: Heterogeneous MPC circumvents hardness of sublinear MPC, and allows very fast algorithms

Open Problems:

O(1)-round MST algorithm?

Deterministic algorithms?

Conditional Lower Bounds?
Possible candidate: Many 2-to-1 cycles?



Extensions:

$(M_{sub}, M_{lin}, M_{sup})$ −Heterogeneous Model

$M_{sub}(m,n)$ total memory of 

$M_{lin}(m,n)$ total memory of 

$M_{sup}(m,n)$ total memory of