

# DiCeS: Detecting Communities in Network Streams Over the Cloud

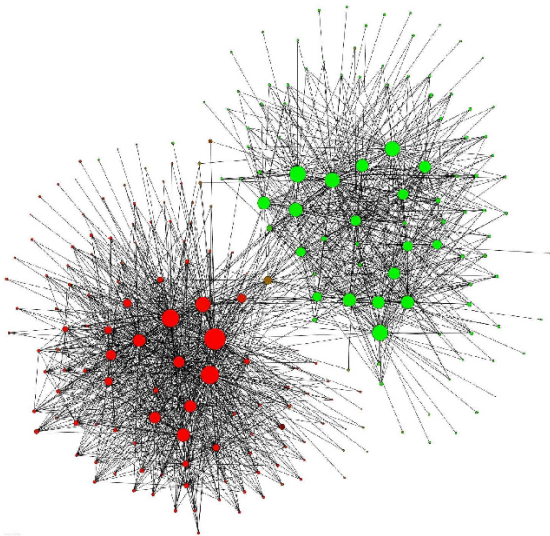
Panagiotis Liakos<sup>†</sup> - Katia Papakonstantinou<sup>‡</sup>  
Alexandros Ntoulas<sup>†</sup> - Alex Delis<sup>†</sup>

<sup>†</sup>University of Athens

<sup>‡</sup>Athens University of Economics and Business

12<sup>th</sup> IEEE International Conference on Cloud Computing, Milan, Italy  
July 8<sup>th</sup>–13<sup>th</sup>, 2019

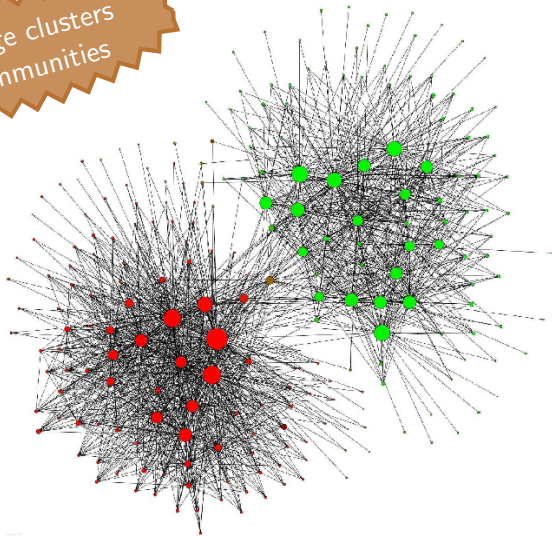
# Belgian Mobile Phone Network



Fast unfolding of community hierarchies in large networks: Blondel et al.

# Belgian Mobile Phone Network

two large clusters  
of communities

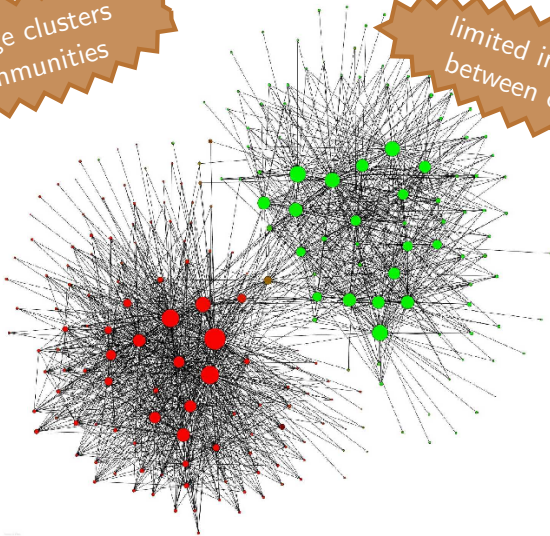


Fast unfolding of community hierarchies in large networks: Blondel et al.

# Belgian Mobile Phone Network

two large clusters  
of communities

limited interaction  
between clusters!

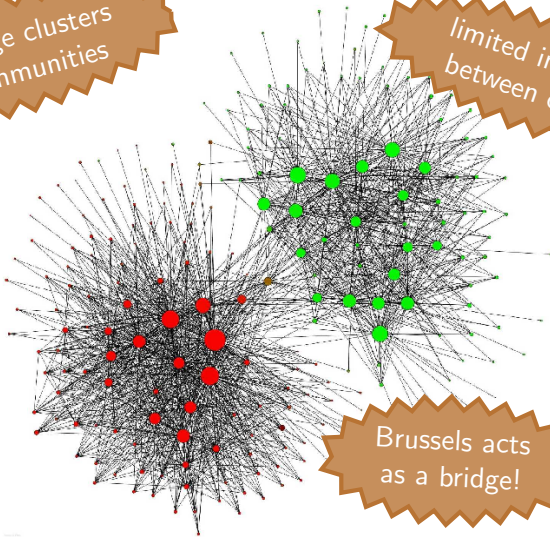


Fast unfolding of community hierarchies in large networks: Blondel et al.

# Belgian Mobile Phone Network

two large clusters  
of communities

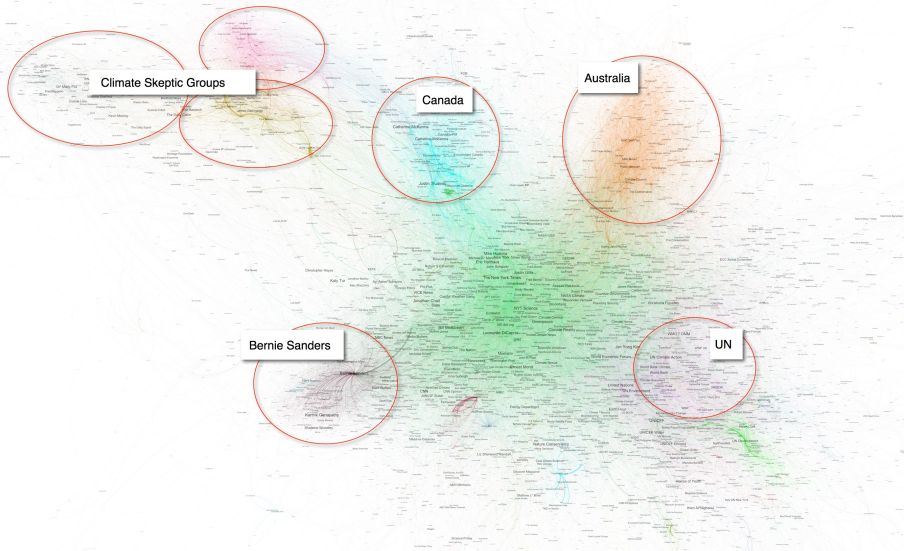
limited interaction  
between clusters!



Brussels acts  
as a bridge!

Fast unfolding of community hierarchies in large networks: Blondel et al.

# Climate change conversation on Twitter



# Climate change conversation on Twitter

Climate Sceptic Groups

Canada

Australia

real-world networks  
are massive!

Bernie Sanders

UN

carbonbrief.org

# Climate change conversation on Twitter

Climate Sceptic Groups

Canada

Australia

real-world networks  
are massive!

change rapidly!

Bernie Sanders

UN

carbonbrief.org

# Climate change conversation on Twitter

Climate Sceptic Groups

Canada

Australia

real-world networks  
are massive!

change rapidly!

Bernie Sanders

UN

exhibit community structure!

carbonbrief.org

# Motivation

We want to extract the community structure of nodes in a network that changes rapidly.

Many useful applications:

- we can launch **accurate & successful** advertising campaigns
- we can provide more **informative & engaging** social network feeds
- we can gain **insights on the evolution** of large real-world networks

Size of graph data appears to be ever-increasing:

- Facebook has more than **2 billion** registered users
- Google indexes more than **1 trillion** unique URLs

## CoEuS [LND17] IEEE Big Data 2017

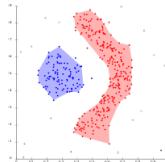
A novel community detection algorithm that operates on a graph stream, using space sublinear to the number of edges.

### Additionally:

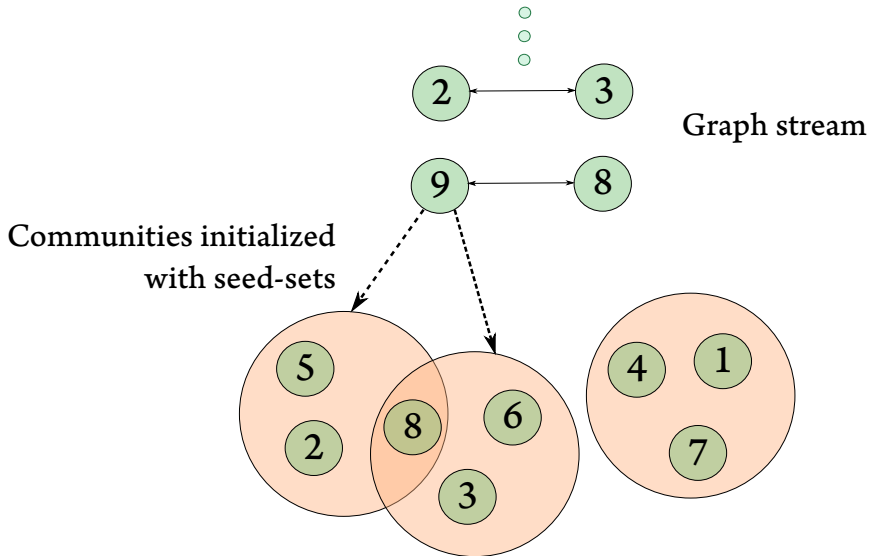
A PageRank-like  
Edge Quality Variation



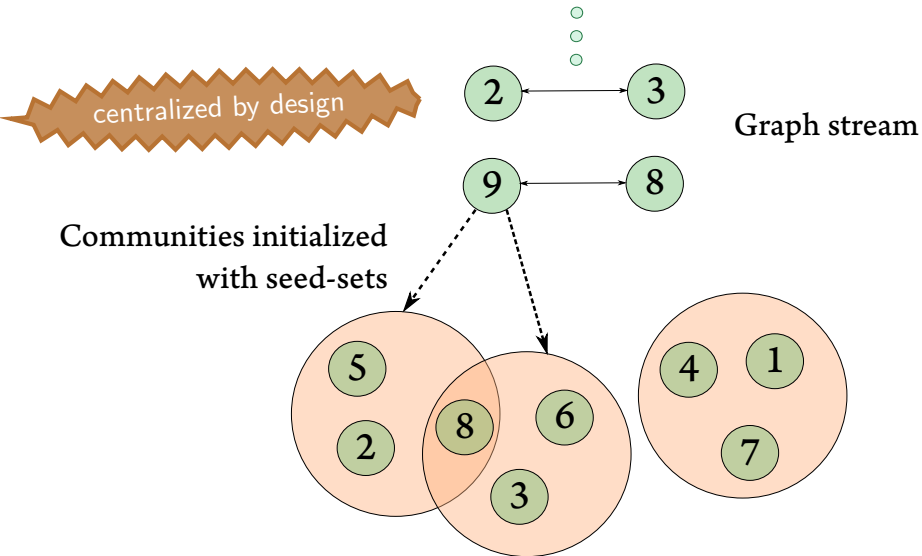
A Novel Clustering Technique  
for Community Size Determination



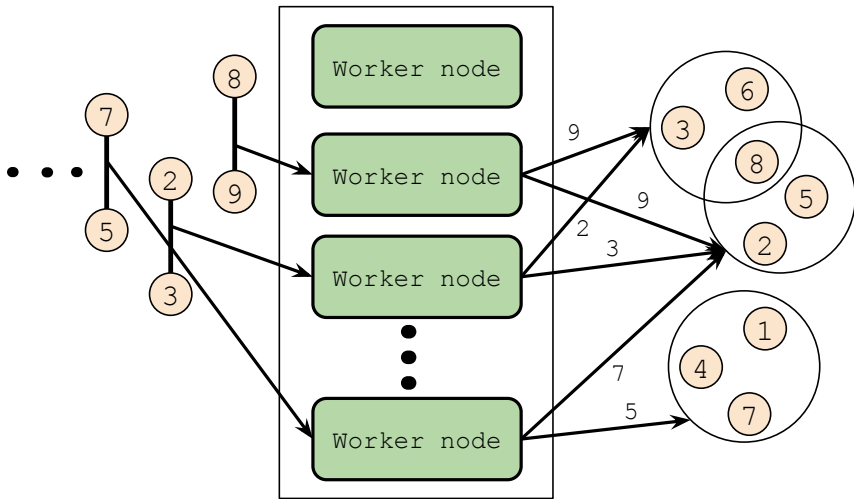
# CoEUS' context



# CoEUS' context



# DiCES' context



# Our Contribution

- We propose **DICES**, a novel **distributed** community detection algorithm for **network streams**.
  - We implement **DICES** as a **cloud application** that handles streams of real-world networks at **impressive rates**.
- 
- Using just 8 workers we can handle **50 million edges per hour**.
  - We achieve **horizontal scalability** that is close to **linear**.
  - We offer **significant improvements** with regard to **accuracy**.

## Apache Storm:

Stream processing framework with broad use in production environments.

- **Tuple:** fundamental data unit
- **Spout:** source of tuples
- **Bolt:** responsible for transforming streams into the desired result
- **Grouping:** determines how the tuples are exchanged

## Redis:

In-memory key-value data store.

- Ultra-fast read/write operations
- Complex data types:
  - **Strings**
  - **Sets**
  - **Sorted Sets**
- Redis-cluster

- Scalability
  - Isolate the processing for every edge
  - Distributed key-value store
- Fault Tolerance
  - All edges must be processed
  - Failing nodes must be restored
- Interactivity
  - Updating the target communities
  - Obtaining results on demand

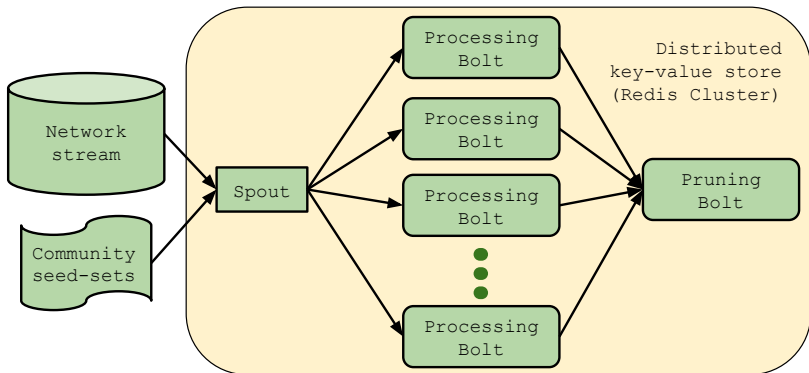
- Community initialization
- Stream ingestion



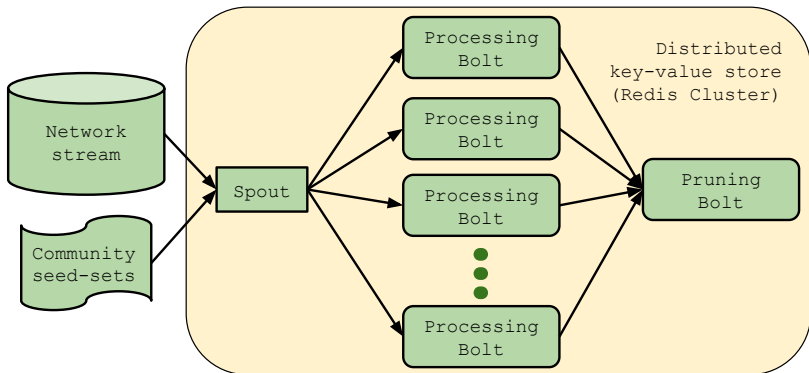
# DiCES' BOLTS

- Stream processing
- Community expansion
- Community pruning

# Our topology



# Our topology



```
$ storm rebalance topology-name [-n new-num-workers] [-e component=parallelism]*
```

## Algorithm 1: DiCES

input  
begin

: A tuple emitted from the spout.

```

if tuple.length == 1 then
  // renewed set of communities
  communities ← tuple[0];
else
  // handling of an edge
  u ← tuple[0];
  v ← tuple[1];
  degrees[u] + = 1;
  degrees[v] + = 1;
  foreach C ∈ {nc[u] ∪ nc[v]} do
    if u ∈ C then
      cDegrees[C][v] + =  $\frac{cDegrees[C][u]}{degrees[u]}$ ;
    if v ∈ C then
      cDegrees[C][u] + =  $\frac{cDegrees[C][v]}{degrees[v]}$ ;
    if u ∈ C then
      communities[C].put(v,  $\frac{cDegrees[C][v]}{degrees[v]}$ );
      nc[v].add(C);
    if v ∈ C then
      communities[C].put(u,  $\frac{cDegrees[C][u]}{degrees[u]}$ );
      nc[u].add(C);
  emit(1);

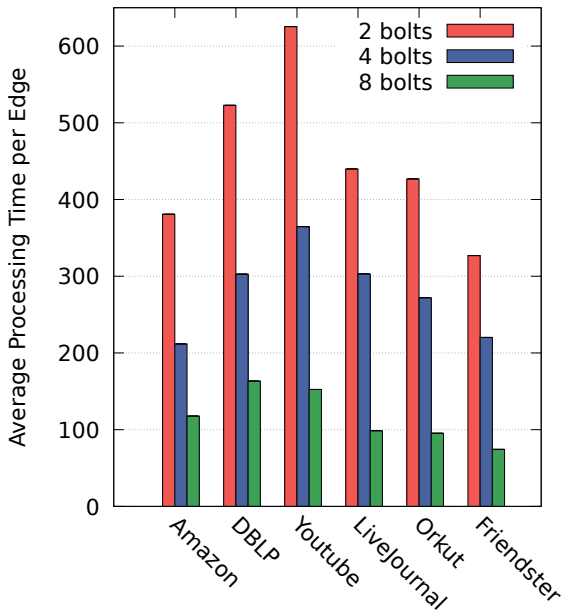
```

# Dataset

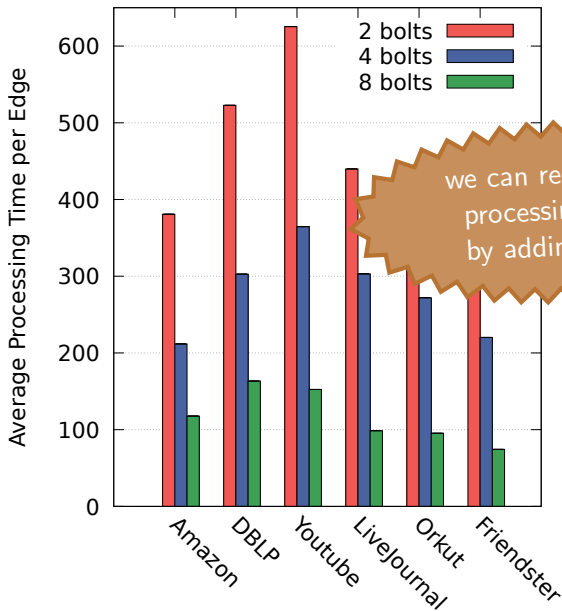
| Graphs             | Type          | Nodes      | Edges         | Av. Degree | Av. Community Size |
|--------------------|---------------|------------|---------------|------------|--------------------|
| <b>DBLP</b>        | Co-authorship | 317,080    | 1,049,866     | 3.31       | 22.45              |
| <b>Amazon</b>      | Co-purchasing | 334,863    | 925,872       | 2.76       | 13.49              |
| <b>Youtube</b>     | Social        | 1,134,890  | 2,987,624     | 2.63       | 14.59              |
| <b>LiveJournal</b> | Social        | 3,997,962  | 34,681,189    | 8.67       | 27.80              |
| <b>Orkut</b>       | Social        | 3,072,441  | 117,185,083   | 38.14      | 215.72             |
| <b>Friendster</b>  | Social        | 65,608,366 | 1,806,067,135 | 27.53      | 46.81              |

- Networks exceeding 1.8 billion links
- Accompanying ground-truth communities allow for the evaluation of accuracy

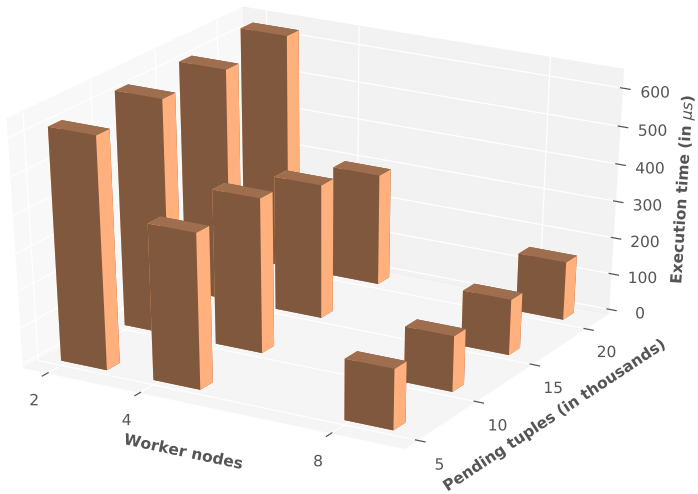
# Performance



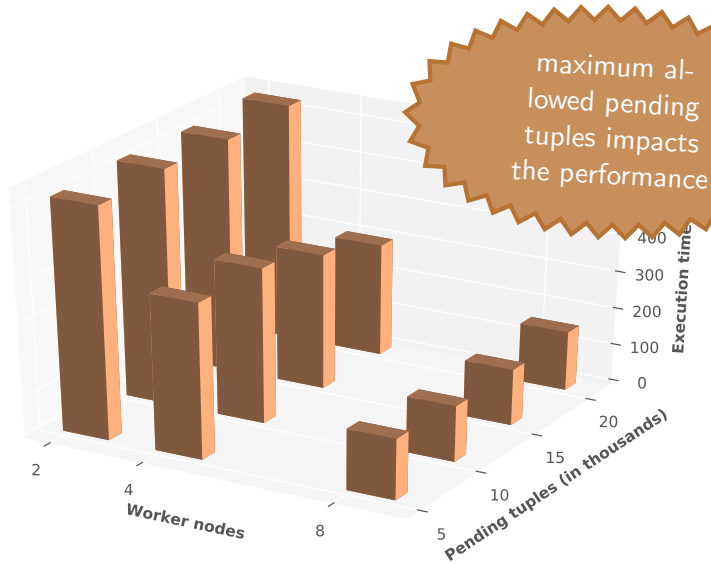
# Performance



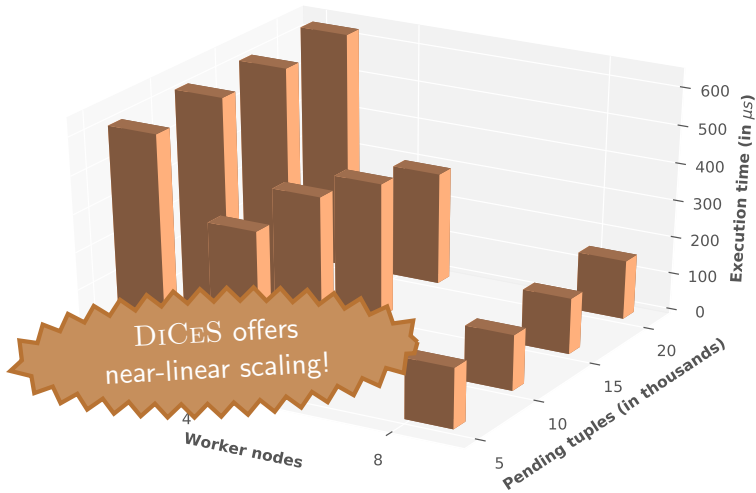
# Scalability



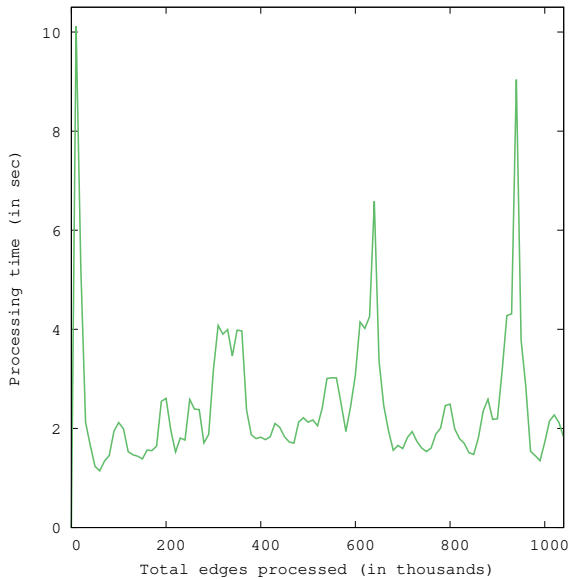
# Scalability



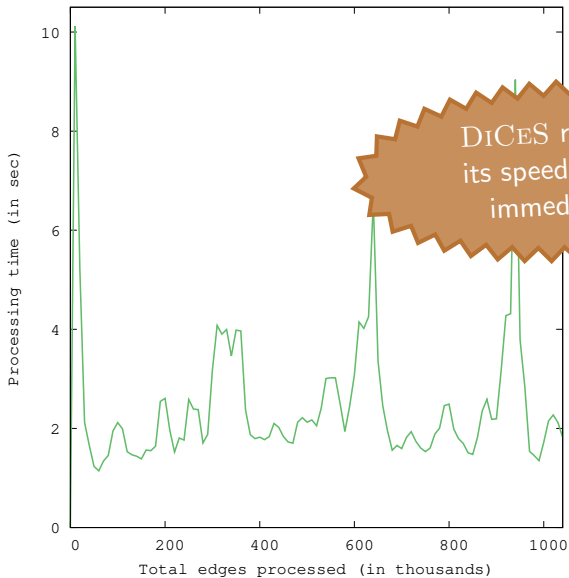
# Scalability



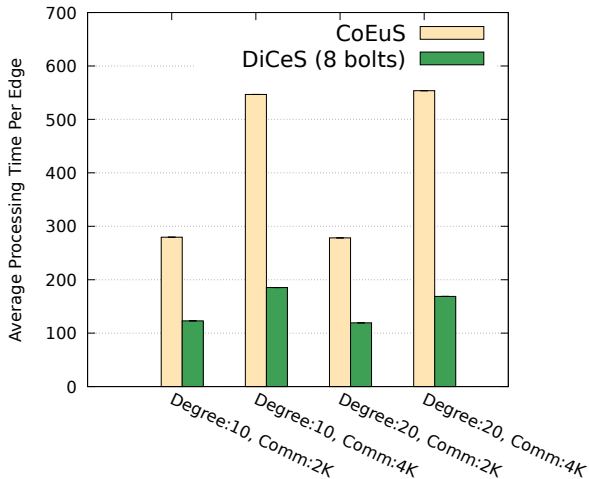
# Fault Tolerance



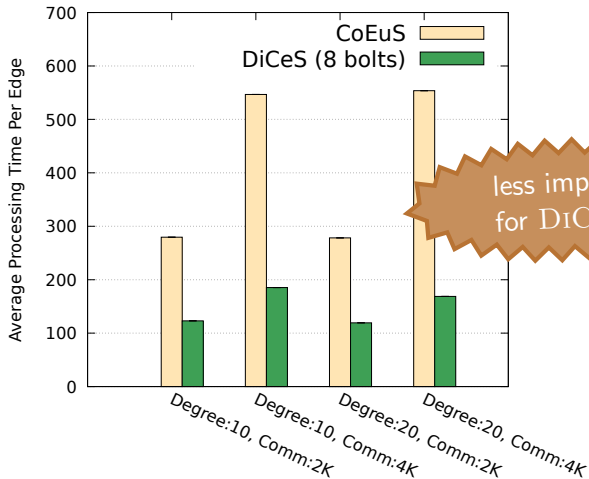
# Fault Tolerance



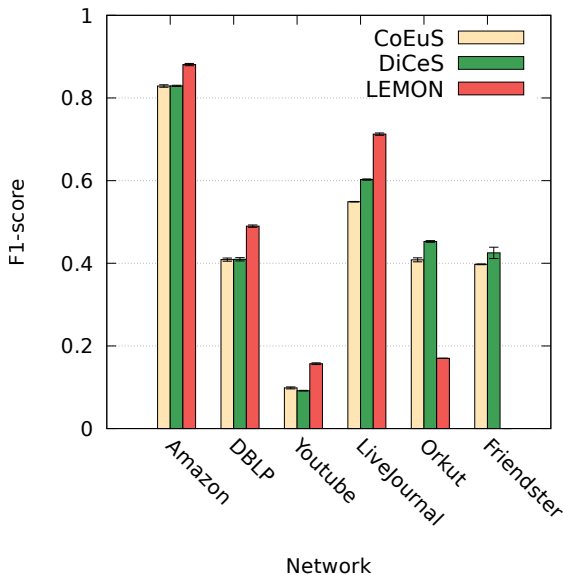
# Average Degree & Number of Communities



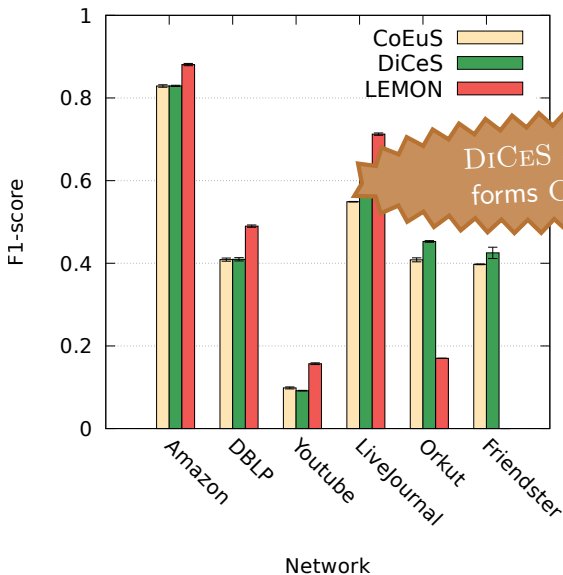
# Average Degree & Number of Communities



# F1-score comparison



# F1-score comparison



DiCeS outperforms CoEuS

# Conclusion

- DiCES is a **streaming** community detection **virtual infrastructure** for large-scale networks that evolve rapidly.
- DiCES **distributes** load to worker nodes in the cloud.
- We can process almost **50 million edges per hour** using only 8 worker nodes.
- DiCES is shown to scale **almost linearly**.

- [LND17] Panagiotis Liakos, Alexandros Ntoulas, and Alex Delis.  
COEUS: community detection via seed-set expansion on graph streams.  
In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 676–685, 2017.

# thank you!

`https://github.com/panagiotisl/DiCeS`

for further details email me at:

`p.liakos@di.uoa.gr`

`hive.di.uoa.gr/network-analysis/`   `www.madgik.di.uoa.gr/`