June 27 - July 2, 2014
Anchorage, Alaska

MapReduce-based Distributed *K*-shell Decomposition for Online Social Networks

Katerina Pechlivanidou
Dimitrios Katsaros
Leandros Tassiulas

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Presentation Structure

June 27 – July 2, 2014
Anchorage, Alaska

1. *K*-shell Decomposition
2. Why k-shell decomposition?
3. Filling the gap
4. Why Hadoop's MapReduce?
5. Contributions
6. The MR-SD Algorithm
7. Results and evaluation
8. Future work

University of Thessaly

Contact Information

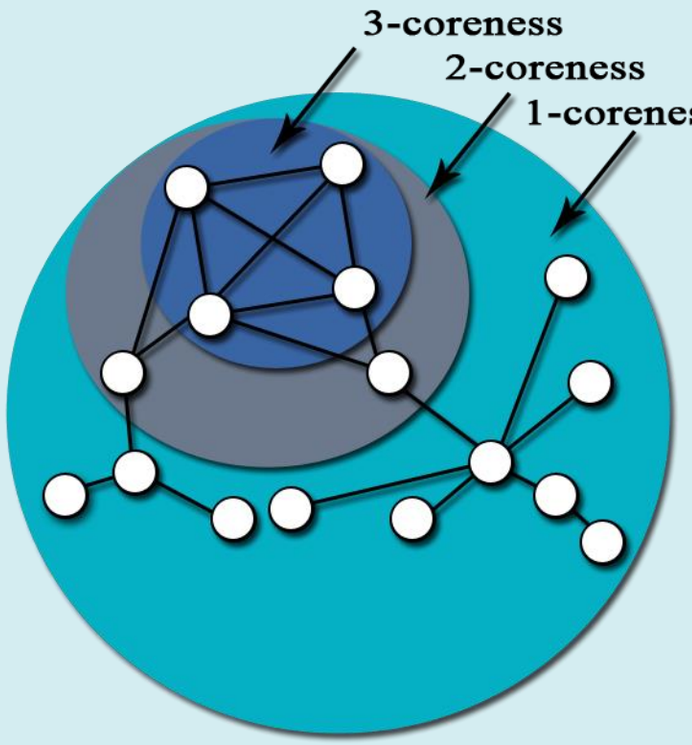
Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

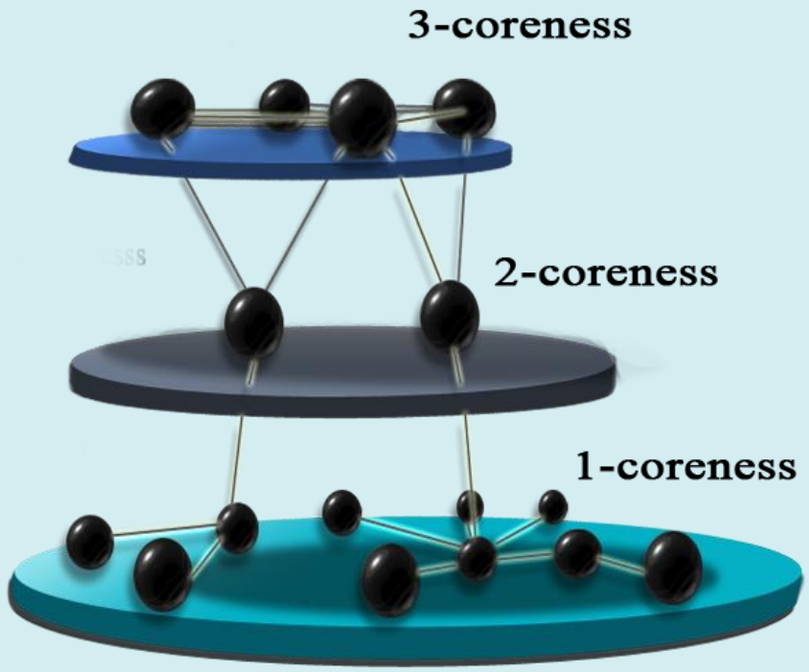
Services Congress

K-shell Decomposition

June 27 - July 2, 2014
Anchorage, Alaska

If from a given graph we recursively delete all vertices, and lines incident with them, of degree less than *k*, the remaining graph is the *k*-core





The terms *k*-core and *k*-shell are used interchangeably

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Why *k*-shell decomposition?

June 27 - July 2, 2014
Anchorage, Alaska

K-shell decomposition is appealing in Social Network Analysis because it

- ✓ reveals the hidden hierarchies in large network graphs
- ✓ highlights the core structure of network
- ✓ characterizes the network nodes beyond classic centrality measures like the degree distribution
- ✓ covers a wide range of diverse application scenarios

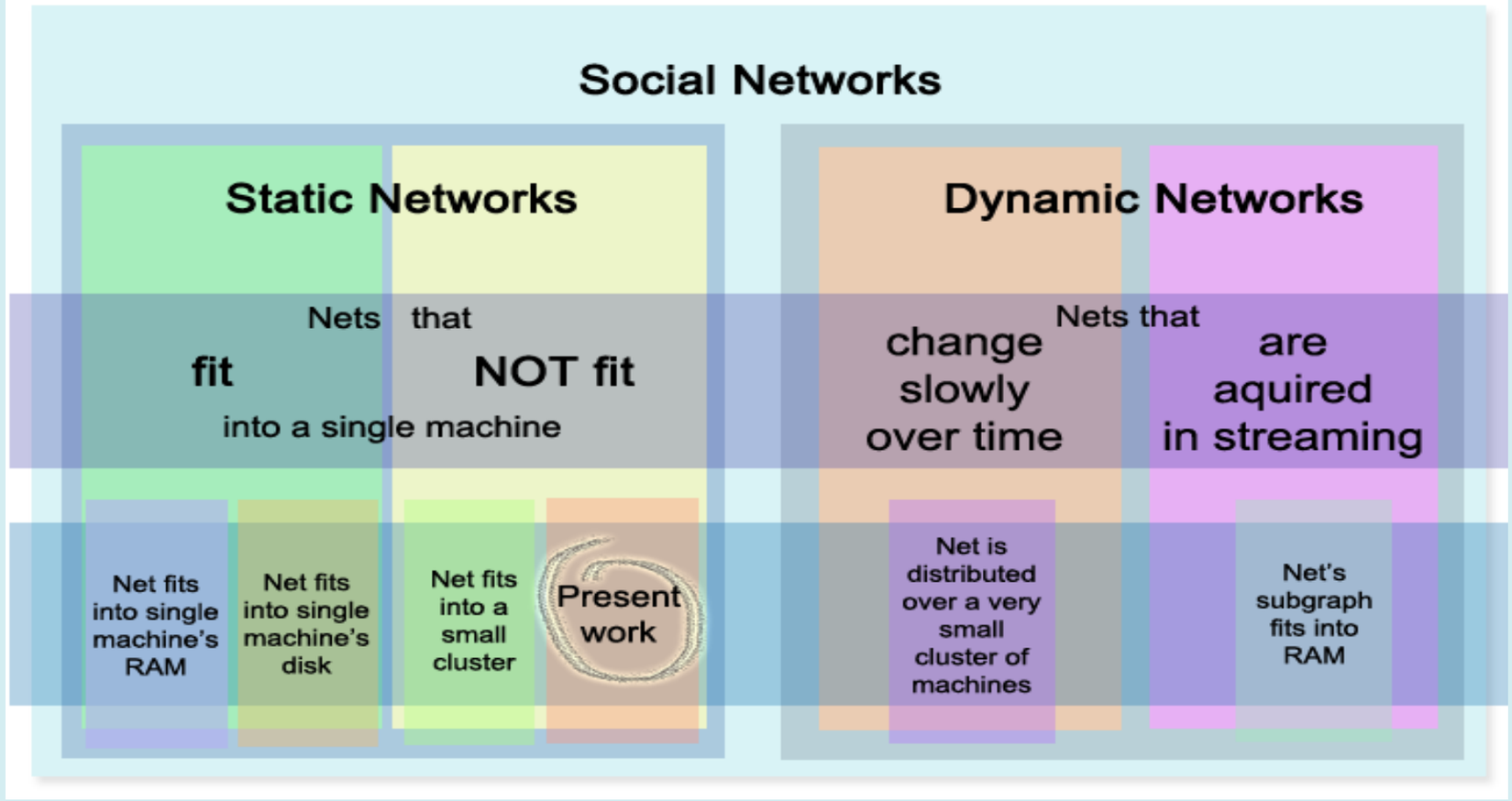
University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Filling the gap

June 27 - July 2, 2014
Anchorage, Alaska



No suitable algorithm exist for modern datacenter environments

All existing algorithms are doomed to fail eventually due to lack of computational resources

University of Thessaly

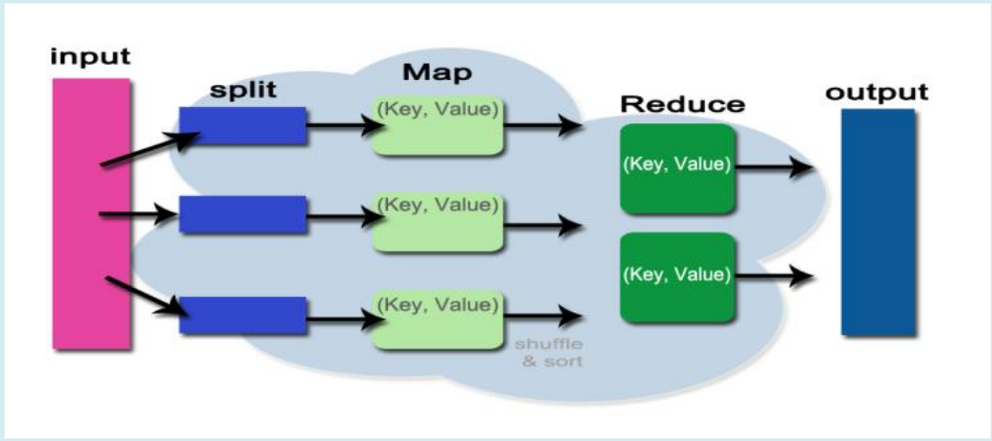
Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Why Hadoop's MapReduce?

June 27 - July 2, 2014
Anchorage, Alaska

- ✓ Process and analysis of very large datasets
- ✓ Time performance
- ✓ Suitable for distributed and parallel algorithms
- ✓ Suitable for huge high-programmed datacenters owned by Internet giants (Google, Yahoo!, Facebook, etc.)



University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Contributions

June 27 – July 2, 2014
Anchorage, Alaska

- ✓ We developed a distributed algorithm for the computation of the *k*-shells of a given network graph
- ✓ For the first time in the literature we present a parallel and distributed algorithm for the *k*-shell decomposition based on Hadoop's MapReduce suitable for huge datacenter environments

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Contributions

June 27 - July 2, 2014
Anchorage, Alaska

- ✓ Our work assesses the performance of the proposed algorithm proven with experimental results based on real datasets
- ✓ Our algorithm is suitable for static and unweighted networks, which are the most encountered and appealing in the majority of today's online social network algorithms
- ✓ Our work analyzes various tradeoffs in our algorithm's operation

University of Thessaly

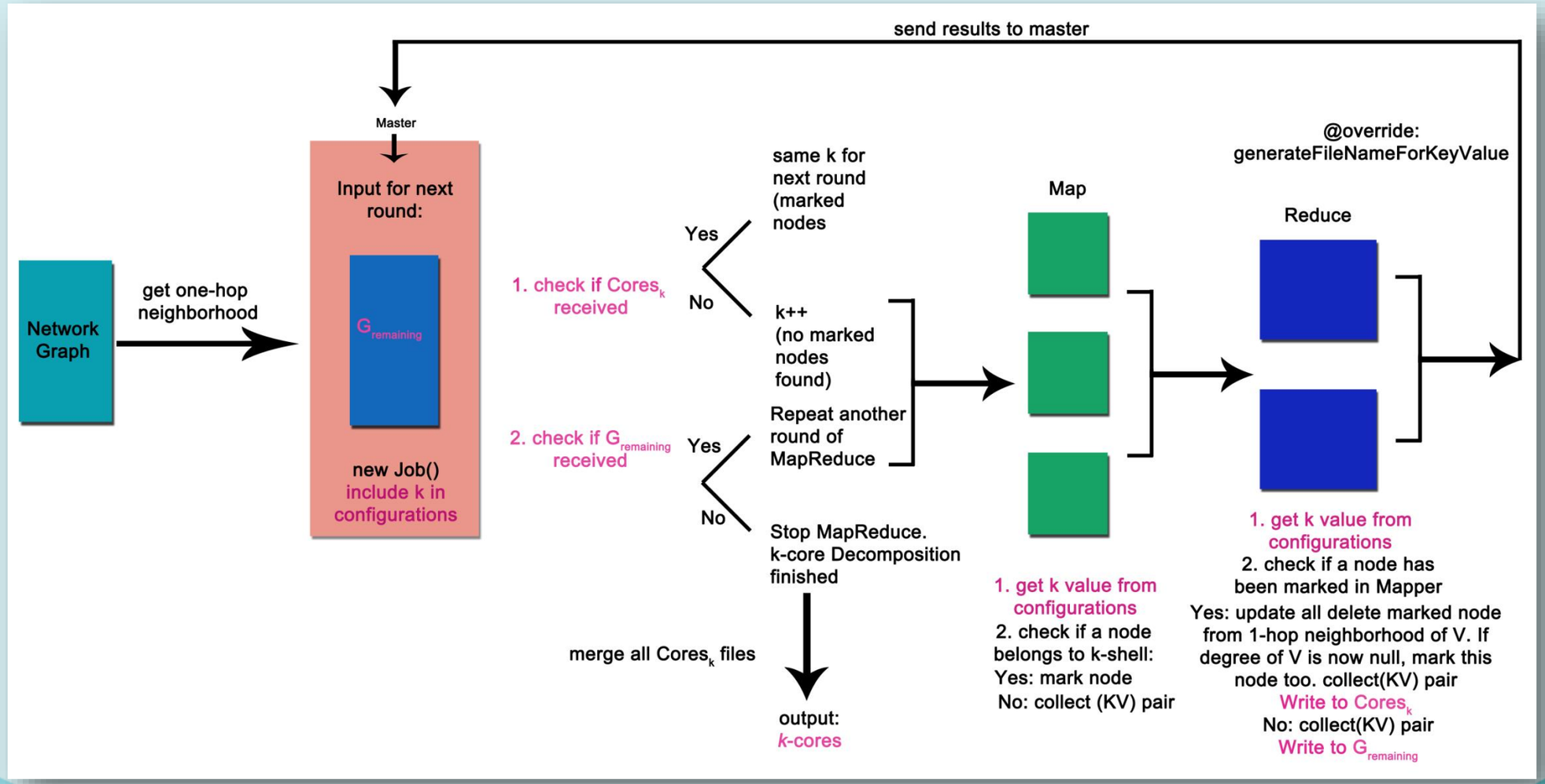
Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The MR-SD Algorithm

June 27 - July 2, 2014
Anchorage, Alaska

MapReduce-based Shell Decomposition



University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Occurring Scenarios

June 27 - July 2, 2014
Anchorage, Alaska

During each MapReduce Job the above scenarios are possible to occur

- *Scenario 1*
  node *K* was marked in preceding Map task

- *Scenario 2*
  node *K* was not marked by previous Mapper

- *Scenario 3*
  node *K* comes coupled with additional information meaning that one or more neighbors are going to be deleted in this pruning round

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Progress and Termination

June 27 – July 2, 2014
Anchorage, Alaska

Progress

Cores$_k$ & G$_{remaining}$

Some nodes were deleted in previous pruning phase

Force another round with the same core value

Only G$_{remaining}$

No nodes were deleted previously

Increase *k* value and force another round

Termination

Only Cores$_k$

No nodes left to prune

k-cores retrieved

University of Thessaly

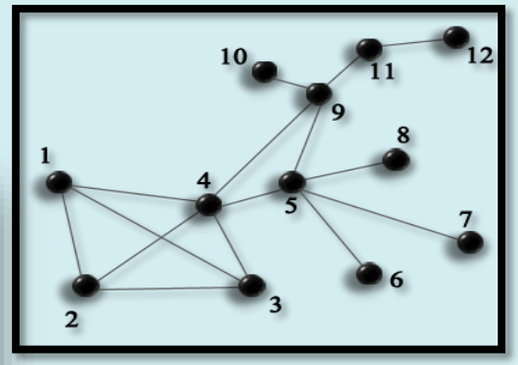
Contact Information

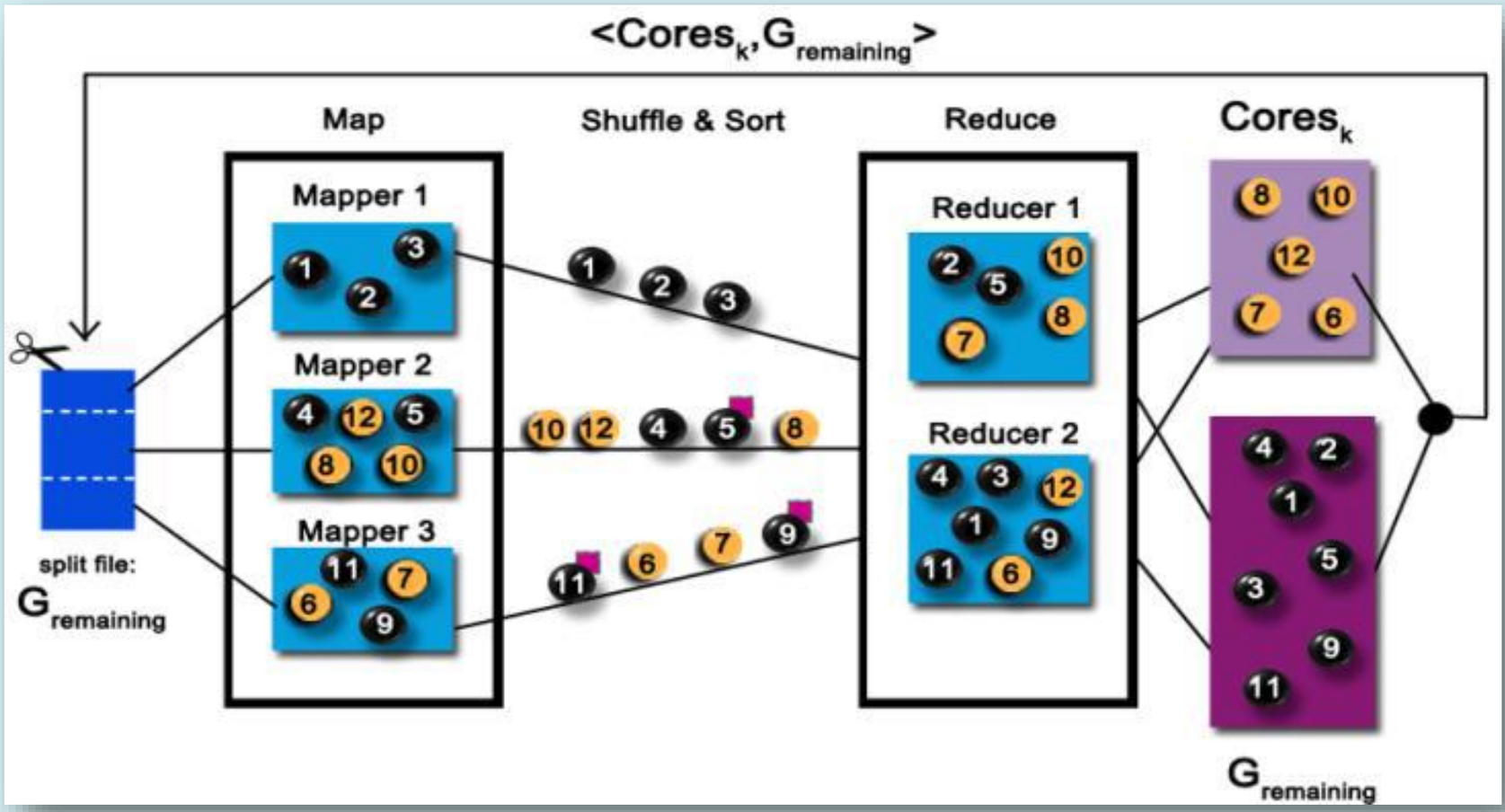
Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

An example running of MR-SD

June 27 - July 2, 2014
Anchorage, Alaska

One (the first) pruning round of MR-SD





University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The evaluation platform

June 27 - July 2, 2014
Anchorage, Alaska

Our cluster consists of five nodes: One master and four slaves

Each node:

- 42GB disk space
- 12GB RAM
- 8-core Intel CPU-based blade
- CentOS
- 10-gigabit Ethernet connection

There was no significant interference from other workloads during any experiment

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The Datasets

June 27 - July 2, 2014
Anchorage, Alaska

Data sets

Experiment	Social Network name	Number of nodes	Number of edges	Number of Jobs
1	**Autonomous systems AS-733**	6474	13895	61
2	**DBLP collaboration network**	317080	1049866	360
3	**Autonomous systems by Skitter**	1696415	11095298	1305
4	**LiveJournal online Social Network**	3997962	34681189	3363
5	**Orkut online social network**	3072441	117185083	5918
6	**Amazon product co-purchasing network**	334863	925872	87
7	**Deaseasome**	7533	22052	118
8	**Protein Interaction Network in budding Yeast**	2361	7182	74

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The Performance Measures

June 27 - July 2, 2014
Anchorage, Alaska

✓ ***Average CPU time spend (msec)***

stands for the time spend solely by the CPU to perform the computations

✓***Average Total Execution Time (sec)***

stands for the total execution time for the *k*-shell decomposition of the input network

✓***Average Total committed heap usage (Bytes)***

it shows the amount of heap memory that is required during the experimentation phase

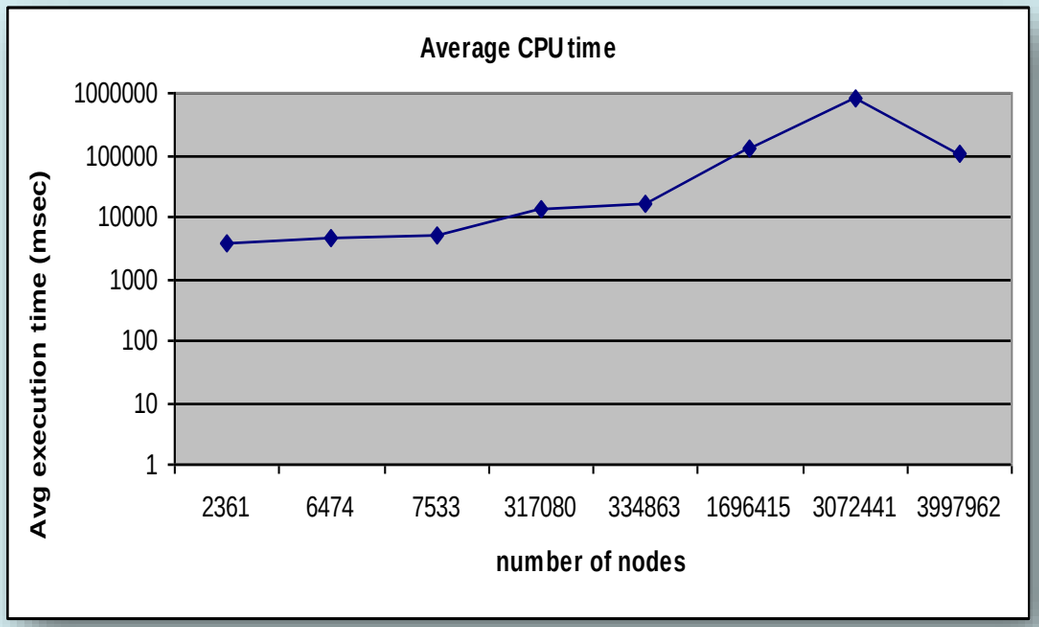
University of Thessaly

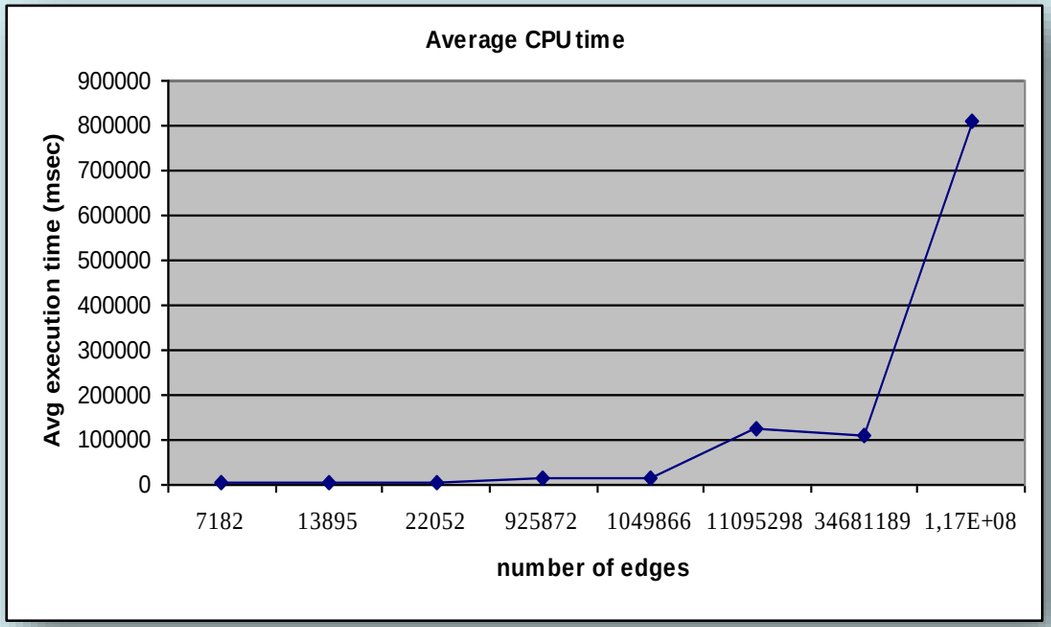
Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Average execution time & Network's Density

June 27 – July 2, 2014
Anchorage, Alaska





Experimental results showed that:

The density of the network is directly correlated to number of MR-SD rounds that are required to decompose the graph

denser network ⟹ **more rounds**

University of Thessaly

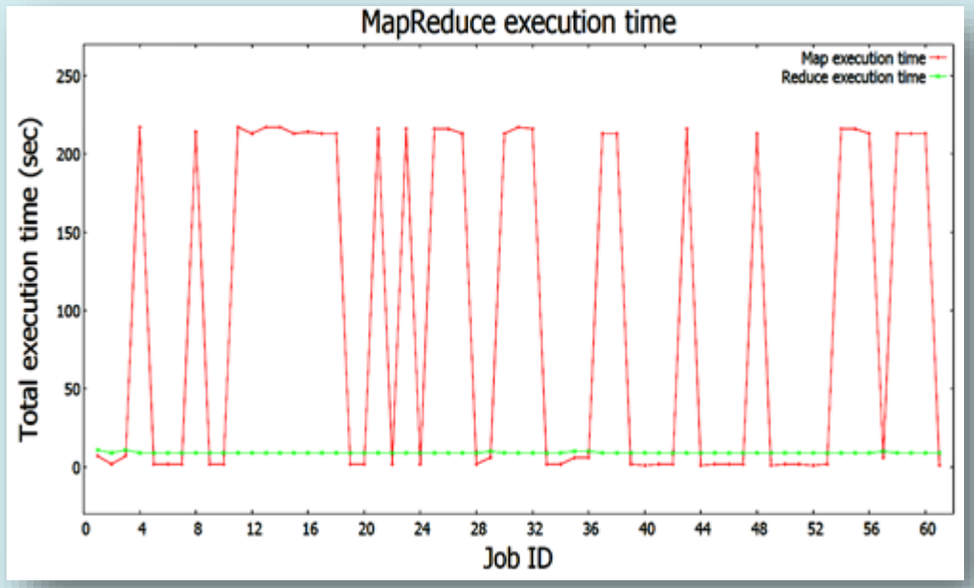
Contact Information

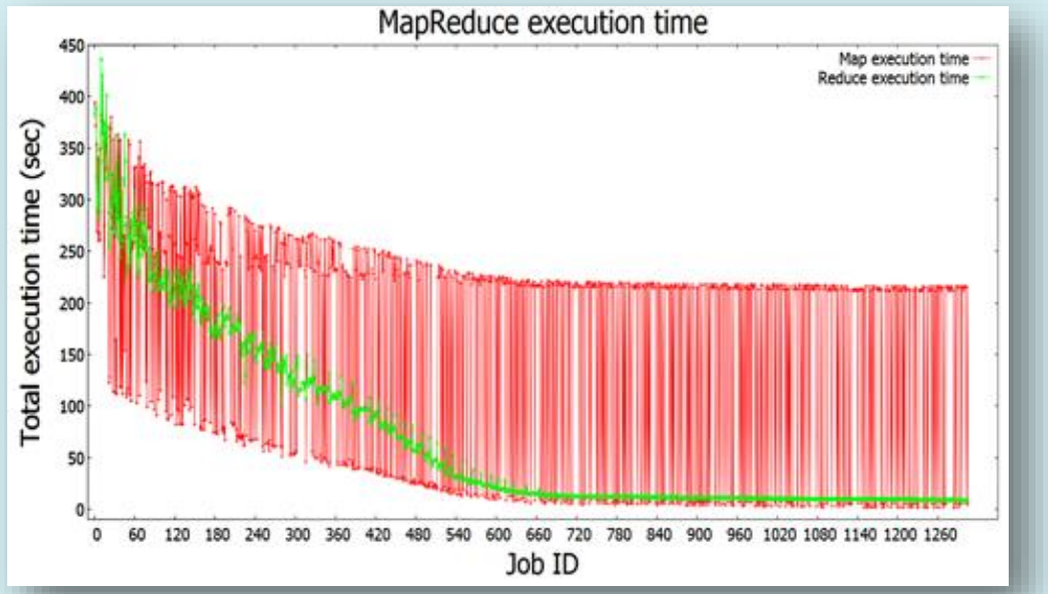
Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Services Congress

Impact of the number of VMs

June 27 - July 2, 2014
Anchorage, Alaska





Experimental results showed that:

When operating **a small cluster** with only few VMs then decomposing **very large networks** means that **each node runs more than one map task** resulting a **greater time**

University of Thessaly

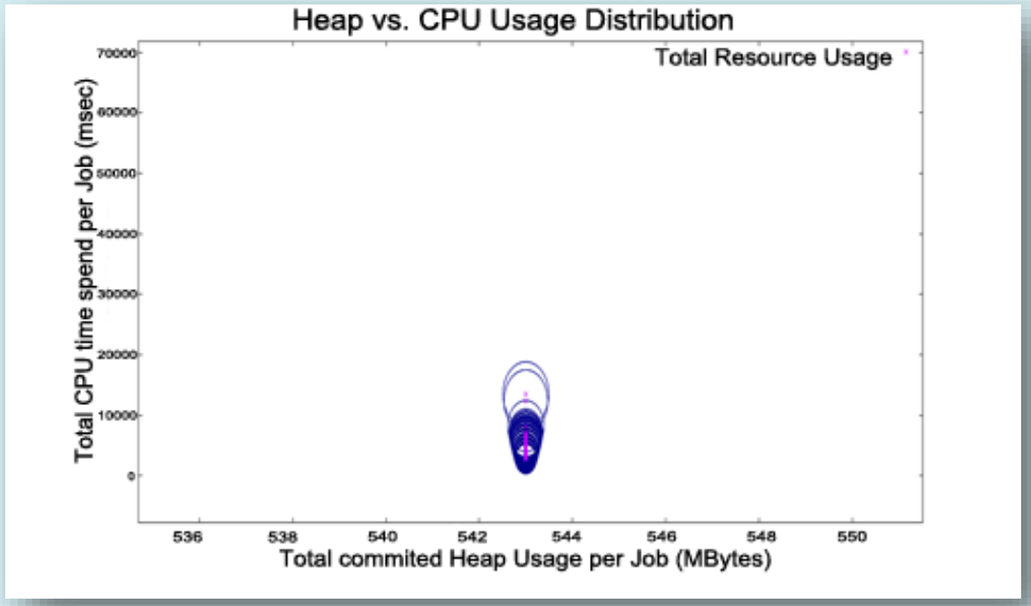
Contact Information

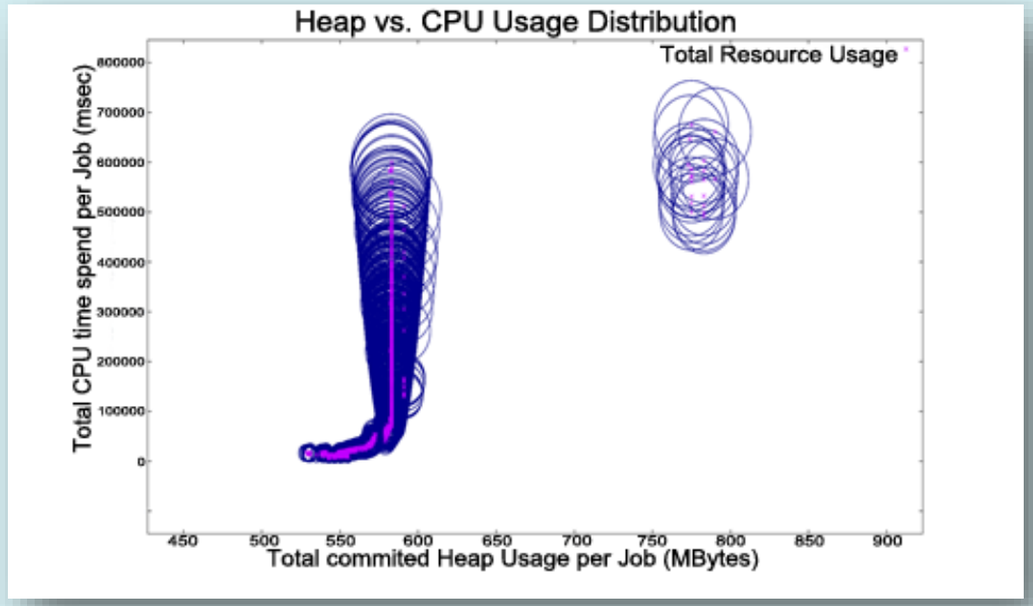
Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Services Congress

Impact of the machine load

June 27 - July 2, 2014
Anchorage, Alaska





Experimental results showed that:

The workload demands for the decomposition of the network graphs require an **increasing processing power** while the **heap size remains actually almost stable** during this interval

Reduce tasks tend to be **more time consuming in the first 20% to 45% of the pruning rounds** than in the last ones

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

What's next?

June 27 – July 2, 2014
Anchorage, Alaska

We plan to

- ✓ find ways to gain some additional speedup
- ✓ improve the way the tasks are distributed over the cluster
- ✓ perform experimentation on a MapReduce environment of a major cloud service provider

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Services Congress

June 27 - July 2, 2014
Anchorage, Alaska

Thank you

University of Thessaly

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

CROWN

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

MR-SD's Complexity Analysis

June 27 – July 2, 2014
Anchorage, Alaska

Algorithm 1: computeOneHopNeighborHood(NetworkGraph G) A Map-Reduce pair to compute the one-hop neighborhood for each node in G

```
Mapper:
  on map do
    for each KV pair do
      K ← nodeId
      V ← neighbor;
      collect(K, V);
  end map

Reducer:
  on reduce do
    for each KV pair do
      collect(K,V);
  end reduce
```

Each node has to be examined one time only and each slave reads only a chunk of the actual network graph

$$O\left(\frac{n}{\#slaves}\right) * O\left(\max(degree)\right)$$

Shuffle & Sort: $k * O(n \log(n))$

Algorithm 2: Driver Routine executed to coordinate the job and detect termination of the k-core decomposition process

```
on initialization do
  configure(Job);
  G_remaining ← computeOneHopNeighborhood(G_in);
  k ← 1;
end initialization

repeat until each node has k-coreness
  configure(Job);
  <Cores_k, G_remaining> ← MR-SD(G_remaining);

  if Cores_k not received then
    k ← k+1;
  if G_remaining not received then
    k-cores ← mergeIntermediate(Cores_k);
  return k-cores;
```

The MR-SD algorithm : A Map-Reduce pair that implements the pruning phase of the *k*-shell decomposition process

```
Mapper:
  on map do
    k ← get(k);
    for each KV pair do
      degree ← ||V||;
      if degree ≤ k then
        node ← mark(node);
        for each vϵV do
          collect(v, attachedInfo);
          collect(node, k);
      else
        for each vϵV do
          V ← V + v;
          collect(K, V);
  end map
Reducer:
  on reduce do
    k ← get(k);
    for each KV pair do
      if attachedInfo received from Mapper then
        for each attachedInfo received do
          oneHopNeiborhood ←{V} – attachedInfo;
      degree ← ||V||;
      if degree == 0 then
        mark(node);
        Cores_k ← collect(K,k);
      else
        V ← oneHopNeiborhood;
        G_remaining ← collect(K,V);
  end reduce
```

$$O\left(\frac{n}{\#slaves} * \max(degree)\right)$$

$$O\left(\frac{n}{\#slaves} * \max(degree)\right)$$

The pruning procedure has to examine all core values

$$k * O\left(\frac{n}{\#slaves} * \max(degree)\right)$$

Each slave has to parse a chunk of the network graph and read only the one-hop neighborhood of each node

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Some application scenarios

June 27 - July 2, 2014
Anchorage, Alaska

K-shell has been used in Social Network Analysis for

- detecting influential spreaders
- discovering communities
- the Internet structure analysis at the autonomous level
- visualization purposes

University of Thessaly

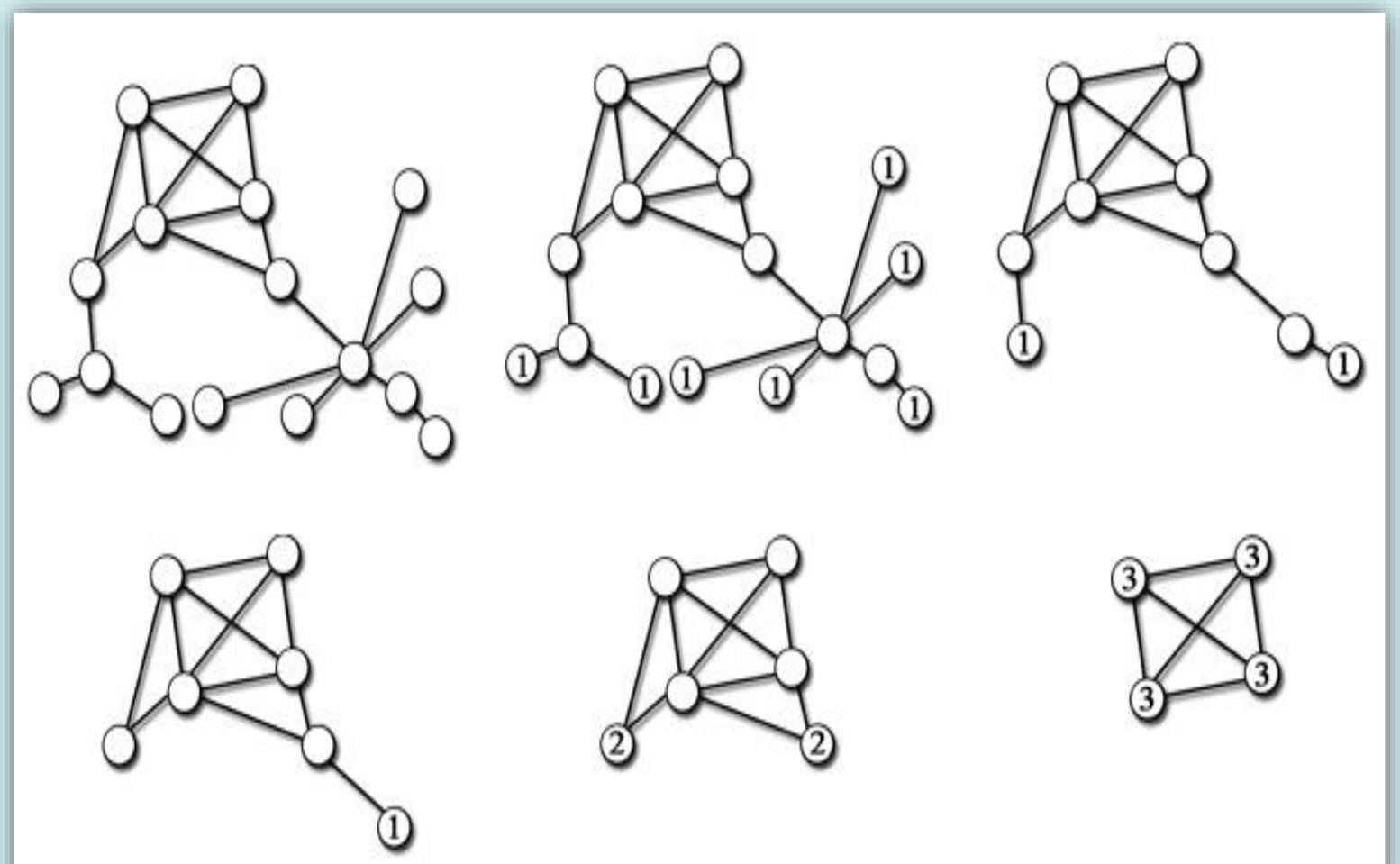
Contact Information

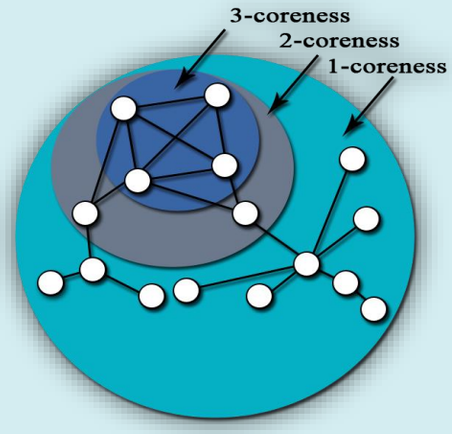
Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Services Congress

An example of *k*-shell Decomposition

June 27 - July 2, 2014
Anchorage, Alaska





University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Algorithm 1

Algorithm 1: computeOneHopNeighborHood(NetworkGraph G) A Map-Reduce pair to compute the one-hop neighborhood for each node in G

Mapper:

```
on map do
    for each KV pair do
        K ← nodeId
        V ← neighbor;
        collect(K, V);
end map
```

Reducer:

```
on reduce do
    for each KV pair do
        collect(K,V);
end reduce
```

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

Algorithm 2

Algorithm 2: Driver Routine executed to coordinate the job and detect termination of the k-core decomposition process

```
on initialization do
    configure(Job);
    G_remaining ← computeOneHopNeighborhood(G_in);
    k ← 1;
end initialization

repeat until each node has k-coreness
    configure(Job);
    <Cores_k, G_remaining> ← MR-SD(G_remaining);

    if Cores_k not received then
        k ← k+1;
    if G_remaining not received then
        k-cores ← mergeIntermediate(Cores_k);
return k-cores;
```

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The MR-SD Algorithm

June 27 - July 2, 2014
Anchorage, Alaska

The MR-SD algorithm : A Map-Reduce pair that implements the pruning phase of the *k*-shell decomposition process

```
Mapper:
  on map do
      k ← get(k);
      for each KV pair do
          degree ← ||V||;
          if degree ≤ k then
              node ← mark(node);
              for each v∈V do
                  collect(v, attachedInfo);
                  collect(node, k);
          else
              for each v∈V do
                  V ← V + v;
                  collect(K, V);
  end map
Reducer:
  on reduce do
      k ← get(k);
      for each KV pair do
          if attachedInfo received from Mapper then
              for each attachedInfo received do
                  oneHopNeiborhood ←{V} – attachedInfo;
          degree ← ||V||;
          if degree == 0 then
              mark(node);
              Cores_k ← collect(K,k);
          else
              V ← oneHopNeiborhood;
              G_remaining ← collect(K,V);
  end reduce
```

University of Thessaly

Contact Information

Katerina Pechlivanidou: kapehliv@uth.gr
Dimitrios Katsaros: dkatsar@uth.gr
Leandros Tassiulas: leandros@uth.gr

The Results in numbers

June 27 - July 2, 2014
Anchorage, Alaska

Average CPU time spent (msec)			
	Map	**Reduce**	**Total (Job)**
Autonomous systems AS-733	2061.31	2715.74	4777.05
DBLP collaboration network	7696.08	5768.33	13464.42
Autonomous systems by Skitter	61377.72	61982.36	123360.08
LiveJournal online Social Network	79122.92	28758.76	107881.68
Orkut online social network	505243.72	304598.03	809841.74
Amazon product co-purchasing network	9803.56	6339.20	16142.76
Deaseasome	2223.05	2718.64	4941.69
Protein Interaction Network in budding Yeast	1559.32	2256.49	3815.81

The Results in numbers

June 27 – July 2, 2014
Anchorage, Alaska

Average Total Execution Time (sec)			
	Map	**Reduce**	**Total (Job)**
Autonomous systems AS-733	99.95	9.13	16.18
DBLP collaboration network	104.79	10.30	19.33
Autonomous systems by Skitter	158.55	71.48	90.83
LiveJournal online Social Network	159.53	33.63	56.77
Orkut online social network	311.50	299.78	339.79
Amazon product co-purchasing network	200.13	10.75	20.06
Deaseasome	175.80	8.87	16.26
Protein Interaction Network in budding Yeast	156.41	8.82	15.00

The Results in numbers

June 27 - July 2, 2014
Anchorage, Alaska

Average Total committed heap usage (Bytes)			
	Map	**Reduce**	**Total (Job)**
Autonomous systems AS-733	379584512	189792256	569376768
DBLP collaboration network	381046693	192004460	573051153
Autonomous systems by Skitter	397844599	199193282	597037881
LiveJournal online Social Network	486663048	201195325	687858373
Orkut online social network	1695512575	157003260	1852515835
Amazon product co-purchasing network	376372495	191817092	568189587
Deaseasome	379584512	189792256	569376768
Protein Interaction Network in budding Yeast	379584512	189792256	569376768