Data Analytics over Decentralized Architectures

From Clusters to the Edge

Davide Frey
davide.frey@inria.fr
Inria Rennes

ASAP Team @ Inria Rennes

DISC'15
SODA'15

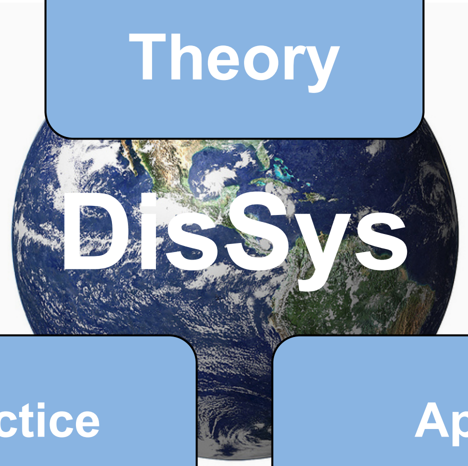
Information dissemination over social networks

Principles of Distributed Algorithms

Computability and efficiency of distributed Systems

PODC'14
STOC'15

Theory



mediego

erc

eit Digital

Tech Transfer

Eurosys'14
Middleware'14
ATC'15

Cloud computing meets P2P

Scalability & Privacy through Decentralization

Practice

App

Privacy-aware Affordable Personalization

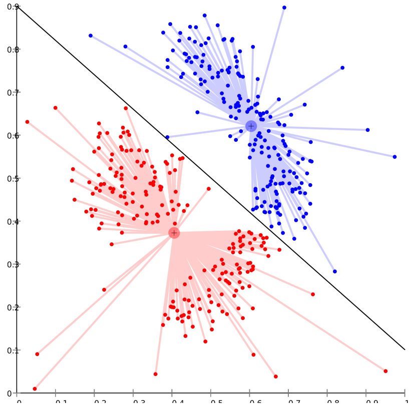
Privacy-aware decentralized computation

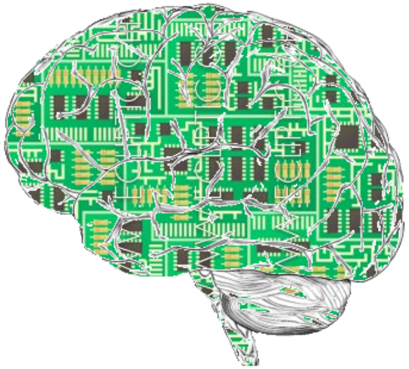
TCS'13
DSN'15

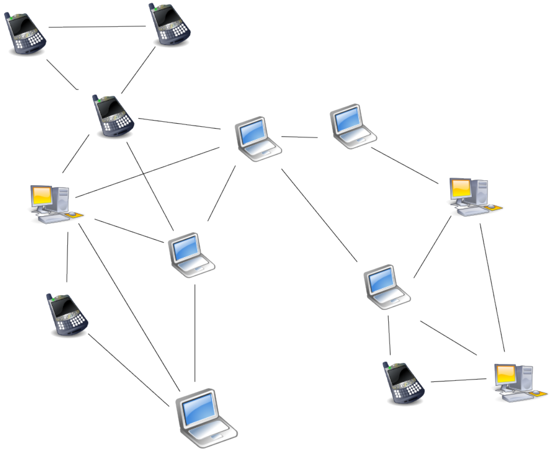
ICDE'16
VLDB'16

Scalable KNNs graphs & queries

Decentralized Data Analytics







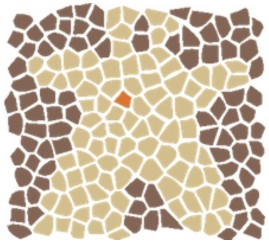
Why Distribute Computation

- Speed/Parallelization
- Scale
- Privacy / Cost /Energy

- Parallelize for Performance
- Decentralize for Simplicity

Are we Already Done?

Spark



GraphLab

Outline

- Brief SOTA
  - Map Reduce / Hadoop
  - Data Parallelism
  - Model Parallelism
- ASAP's Focus
  - Massively Decentralized Data
  - Privacy

MapReduce Example: G-Means

G-Means as a collection of map-reduce jobs

Algorithm 2 KMeansAndFindNewCenters Mapper

Input: point (text)
Output:
centerid (long) $\Rightarrow$ coordinates (float[]), 1 (int)
centerid + *OFFSET* (long) $\Rightarrow$ coordinates (float[]), 1 (int)

procedure MAP(*key*, *point*)
  Find nearest *center*
  Emit(*centerid*, *point*)
  Emit(*centerid* + *OFFSET*, *point*)
end procedure

Algorithm 4 TestClusters Reducer

Input: *vectorid* (int) $\Rightarrow <$ *projection* (double) $>$

procedure REDUCE(*vectorid*, *projections*)
  Read *projections* to build a *vector*
  Normalize *vector* (mean 0, stddev 1)
  ADTEST(*vector*)
  if normal **then**
    Mark cluster as found
  end if
end procedure

Algorithm 3 TestClusters Mapper

Input: *point* (text)
Output: *vectorid* (int) $\Rightarrow$ *projection* (double)

procedure SETUP
  Build vectors from center pairs
  Read centers from previous iteration
end procedure

procedure MAP(*key*, *point*)
  Find nearest *center*
  Find corresponding *vector*
  Compute *projection* of *point* on *vector*
  Emit(*vectorid*, *projection*)
end procedure

Algorithm 5 TestFewClusters Mapper

Input: *point* (text)
Output: *vectorid* (int) $\Rightarrow A^{*2}$ (double)

procedure SETUP
  Build vectors from center pairs
  Read centers from previous iteration
end procedure

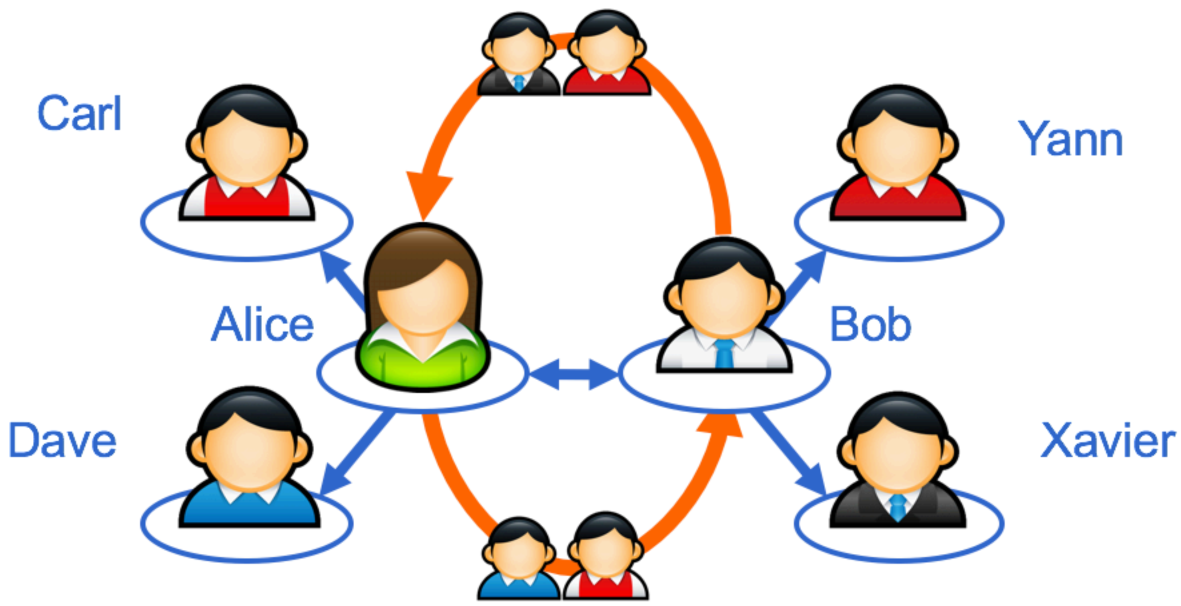
procedure MAP(*key*, *point*)
  Find nearest *center*
  Find corresponding *vector*
  Compute *projection* of *point* on *vector*
  Add *projection* to *list vectorid*
end procedure

procedure CLOSE
  for Each *list* **do**
    Read projections to build a *vector*
    Normalize *vector* (mean 0 , stddev 1)
    Compute $A^{*2} = adtest(vector)$
    Emit($vectorid \Rightarrow A^{*2}$)
  end for
end procedure

[Deb14a] Thibault Debatty, Pietro Michiardi, Olivier Thonnard, Wim Mees. Determining the k in k-means with MapReduce. In Proc. of BeyondMR 2014.

Scalable KNN computation

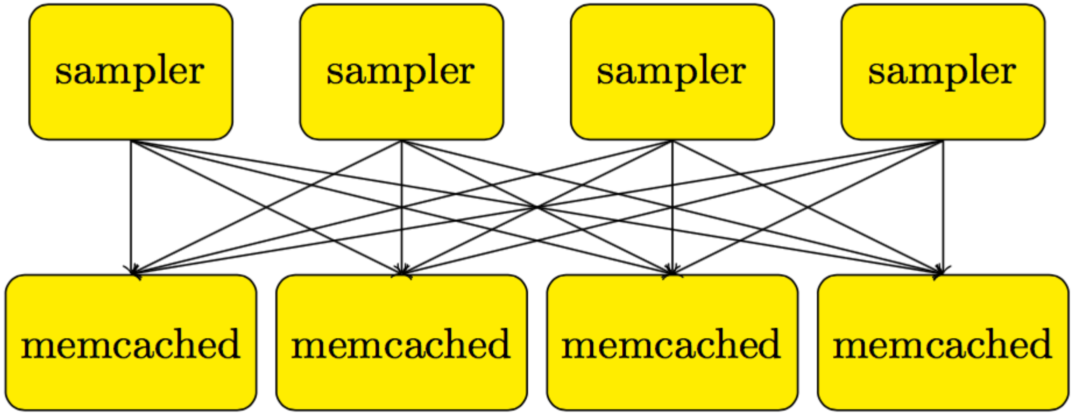
Exploit greedy solutions



[Don11] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th international conference on World wide web (WWW '11). ACM, New York, NY, USA, 577-586.

Data Parallelism: Parameter Servers

- Workers share common model
- Treat different portions of the data
- (Independently) update parameters

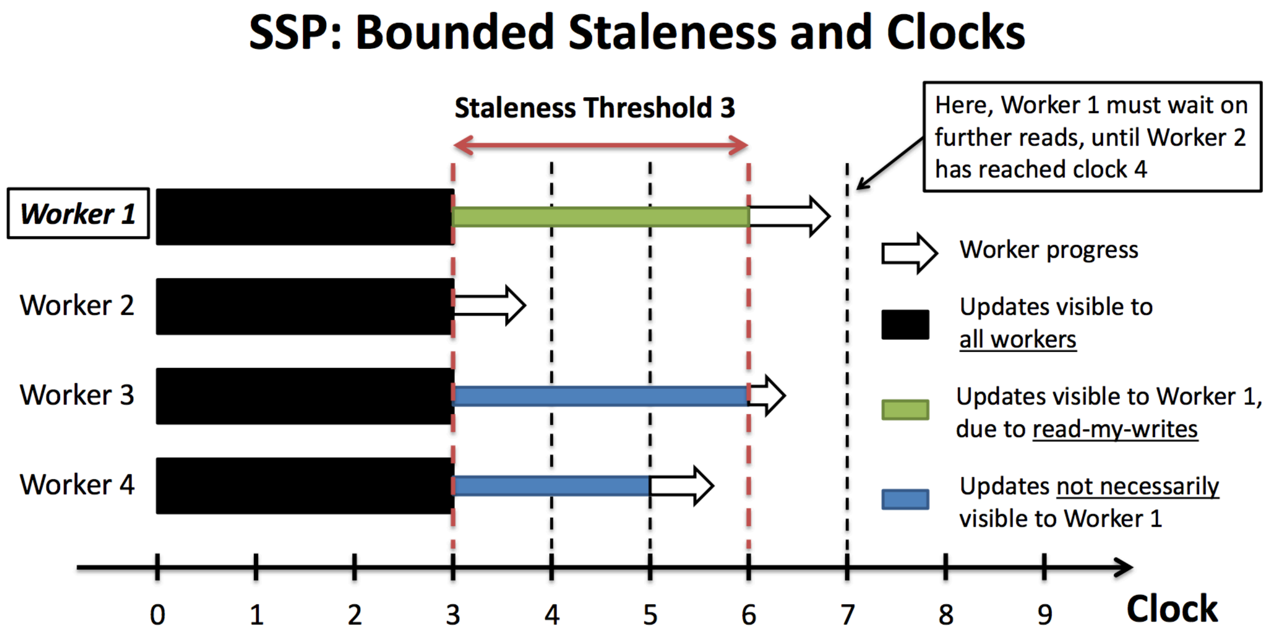


[Smo10] Alexander Smola and Shravan Narayanamurthy. 2010. An architecture for parallel topic models. Proc. VLDB Endow. 3, 1-2 (September 2010), 703-710.

Data Parallelism: Parameter Servers

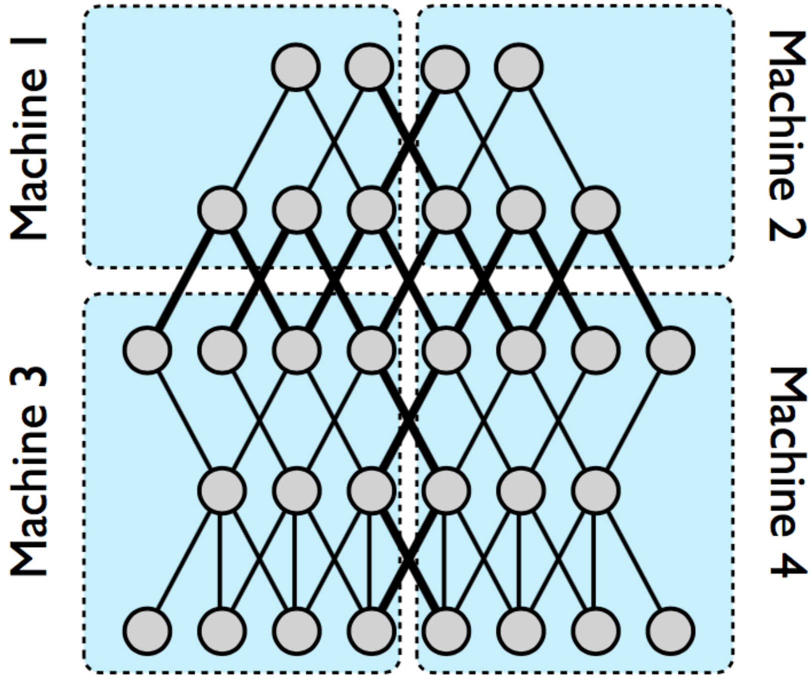
Stale Synchronous Parallel model

- Commutative associative parameter updates: $\theta \leftarrow \theta + \delta$



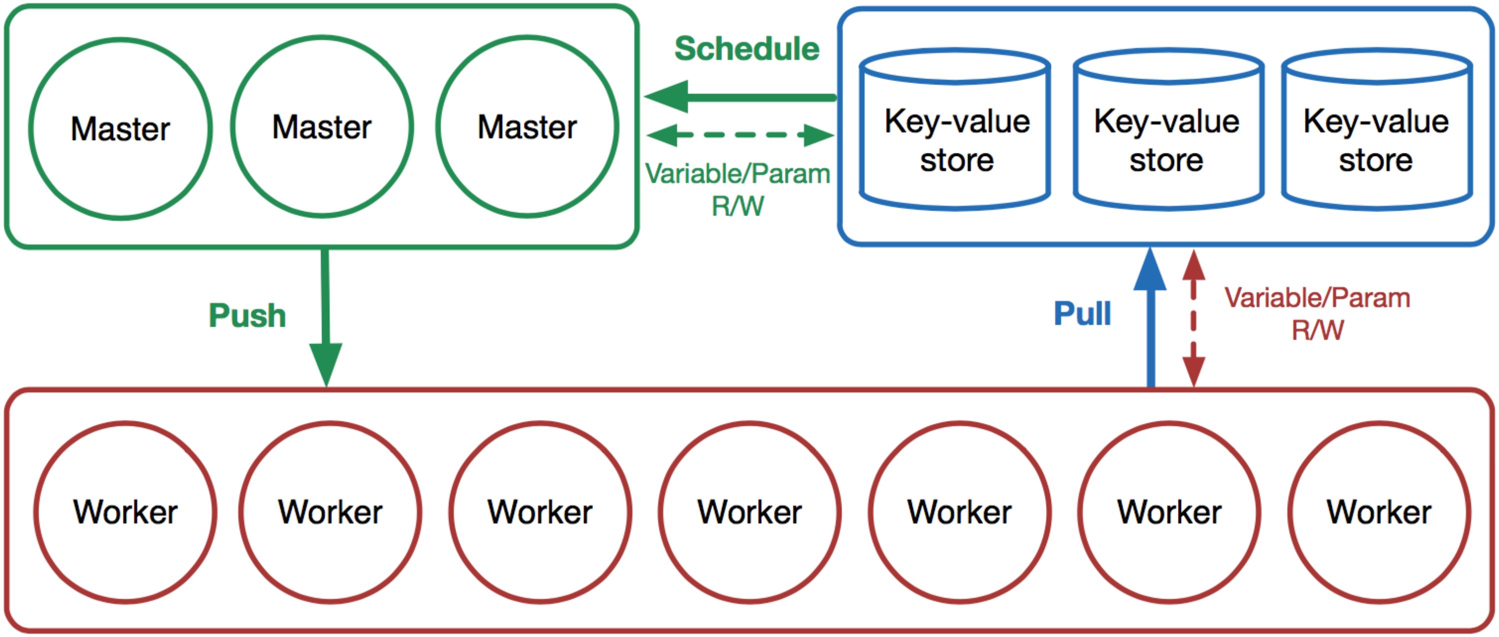
[ho14] Q. Ho, J. Cipar, H. Cui, J. Kim, S. Lee, P. B. Gibbons, G. Gibson, G. R. Ganger, and E. P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In NIPS, 2013.

Model Parallelism: Google DistBelief



Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le and Andrew Y. Ng "Large Scale Distributed Deep Networks". Advances in Neural Information Processing Systems 2012.

Model Parallelism: STRADS



[Lee14] Seunghak Lee, Jin Kyu Kim, Xun Zheng, Qirong Ho, Garth A. Gibson, and Eric P. Xing. On Model Parallelization and Scheduling Strategies for Distributed Machine Learning. Neural Information Processing Systems, 2014 (NIPS 2014)

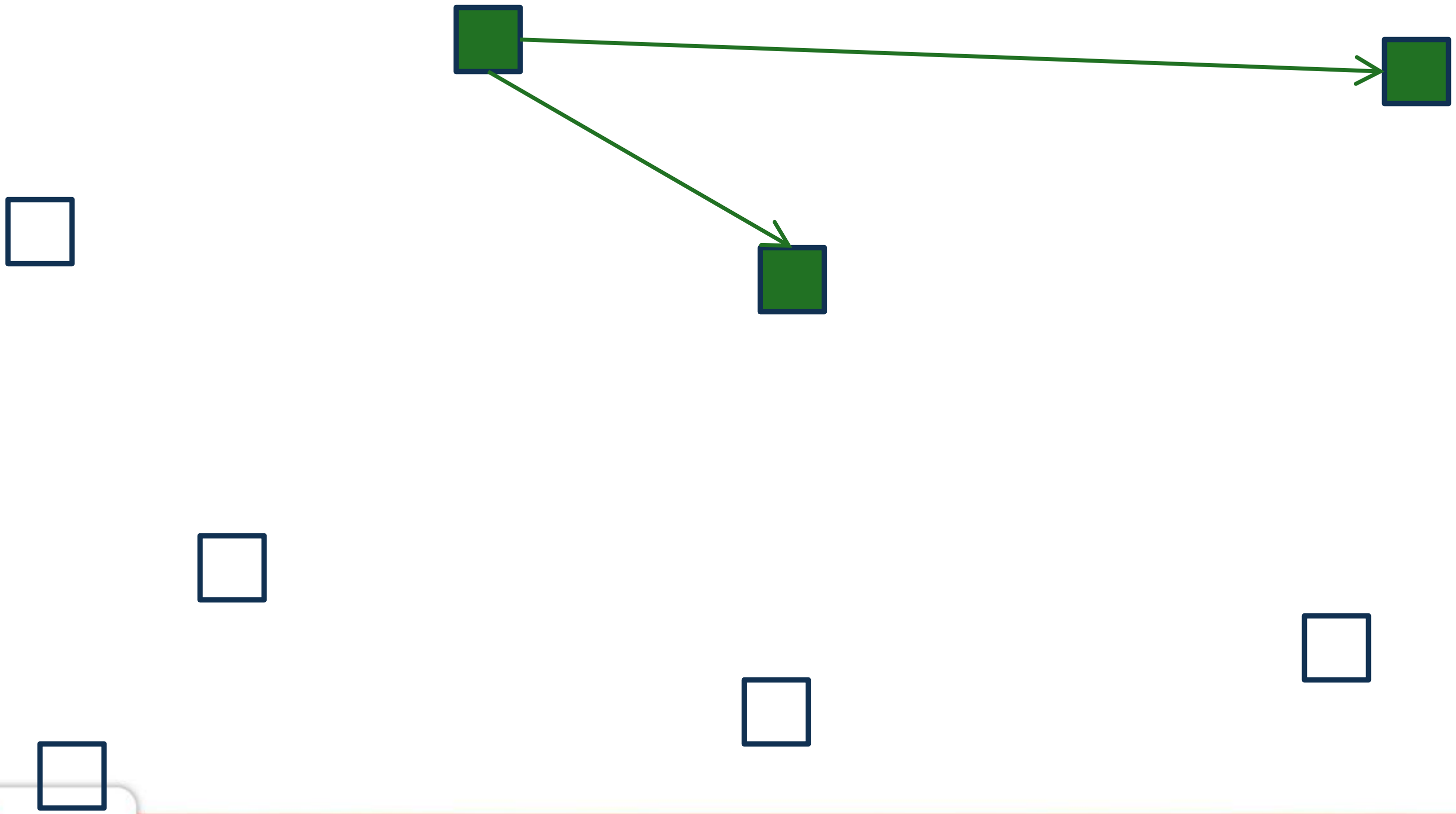
Going Beyond

Massive Decentralization

- Iot
- Set-top boxes
- Edge

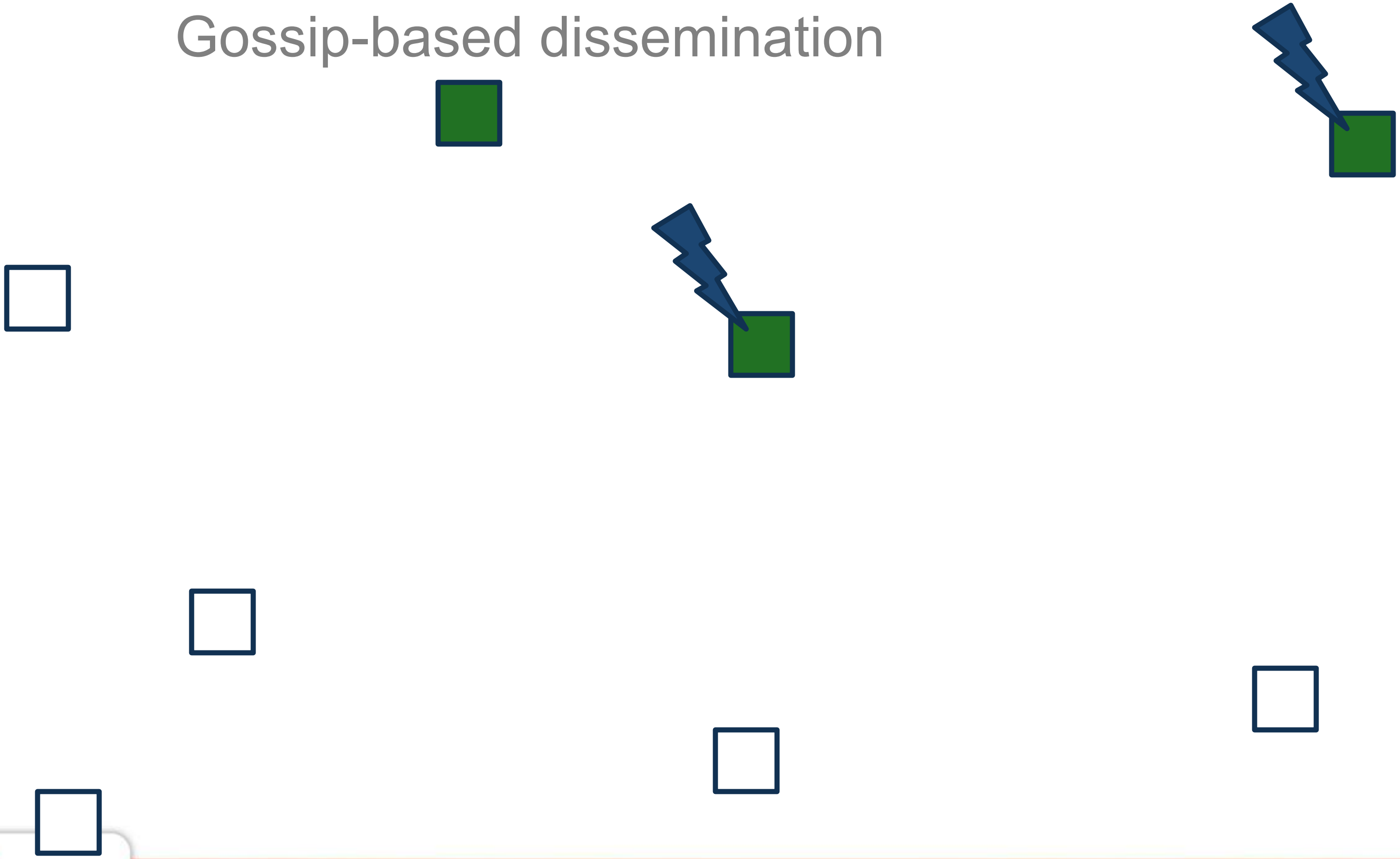
Epidemics as a Tool

Gossip-based dissemination



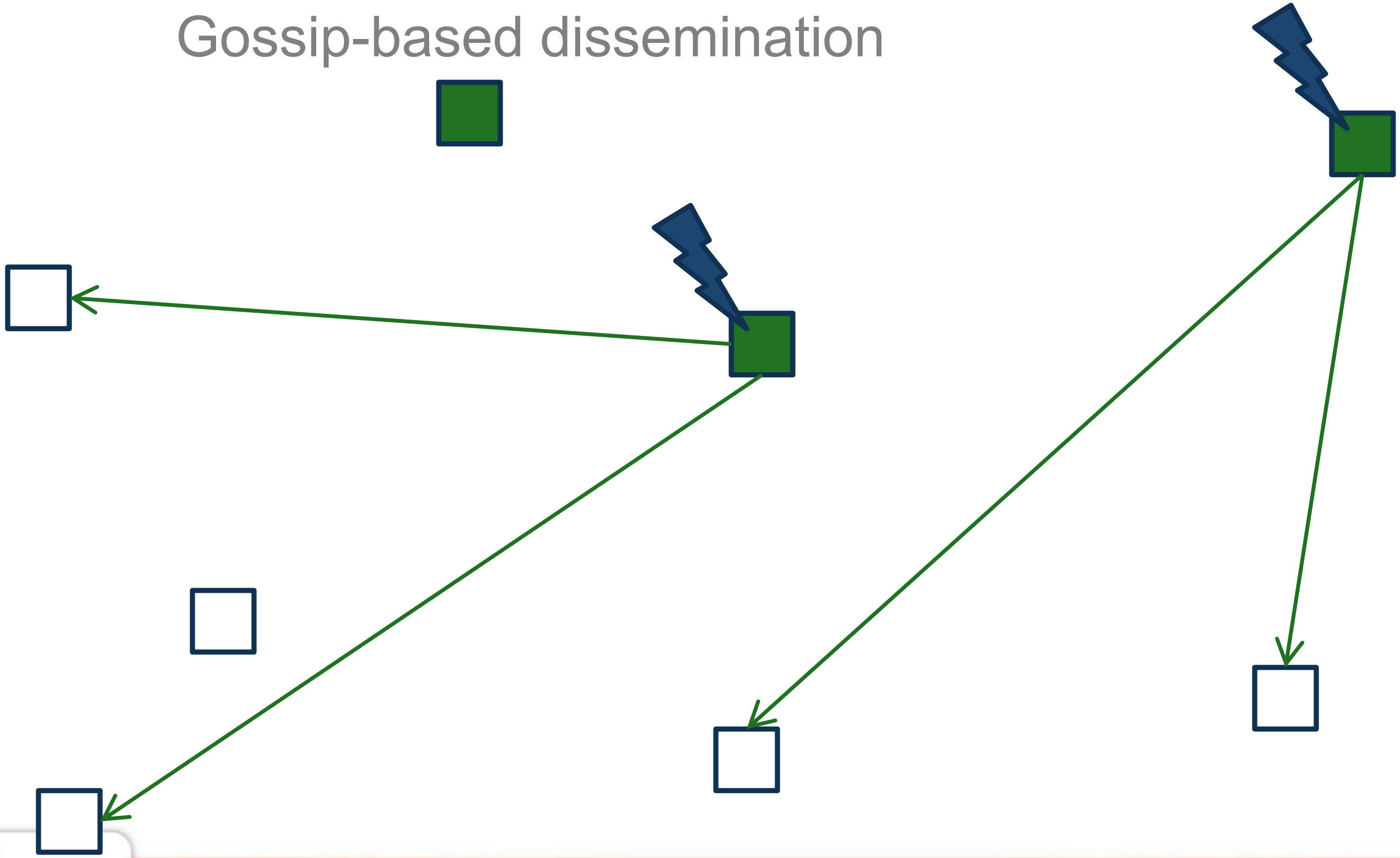
Epidemics as a Tool

Gossip-based dissemination



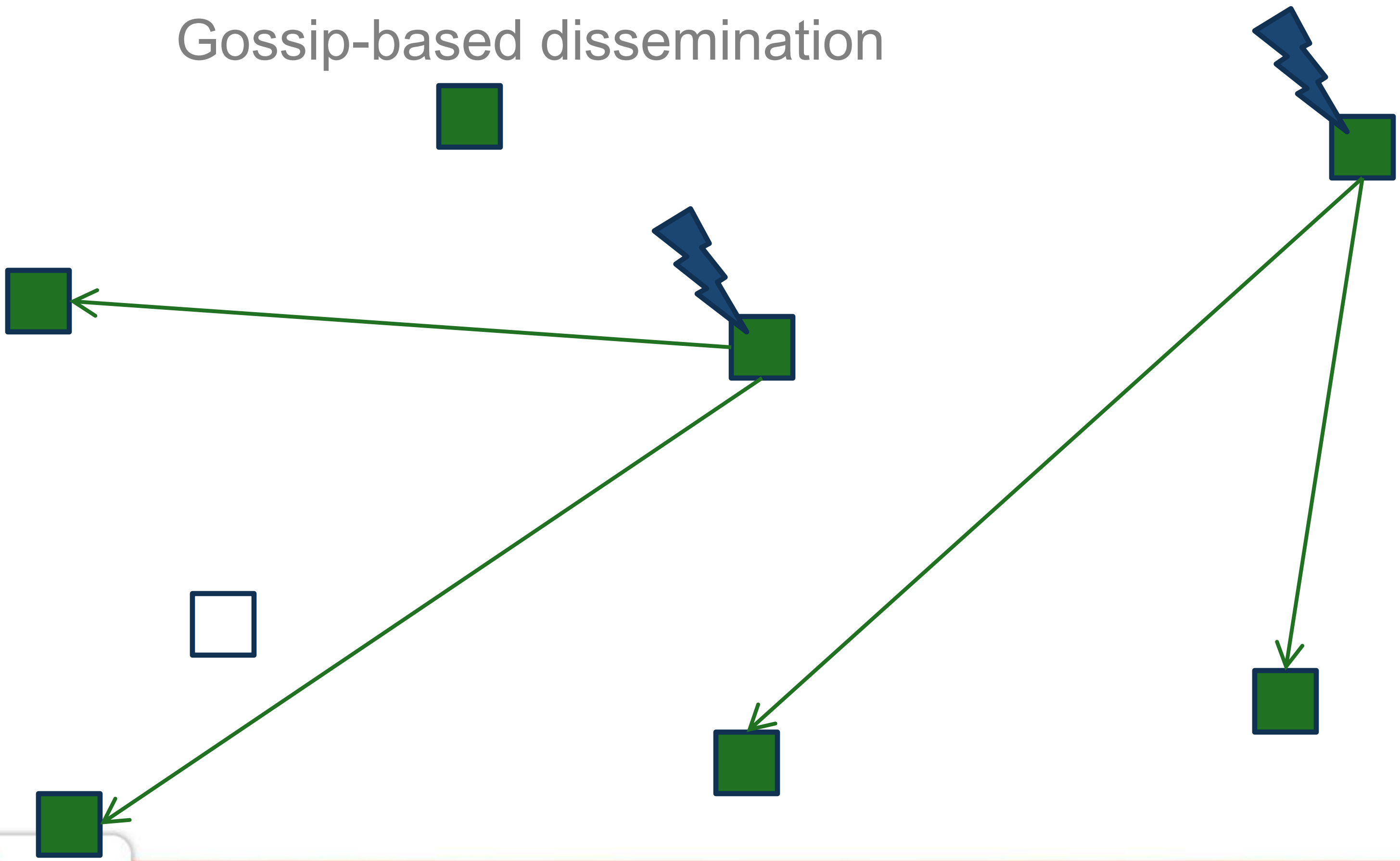
Epidemics as a Tool

Gossip-based dissemination



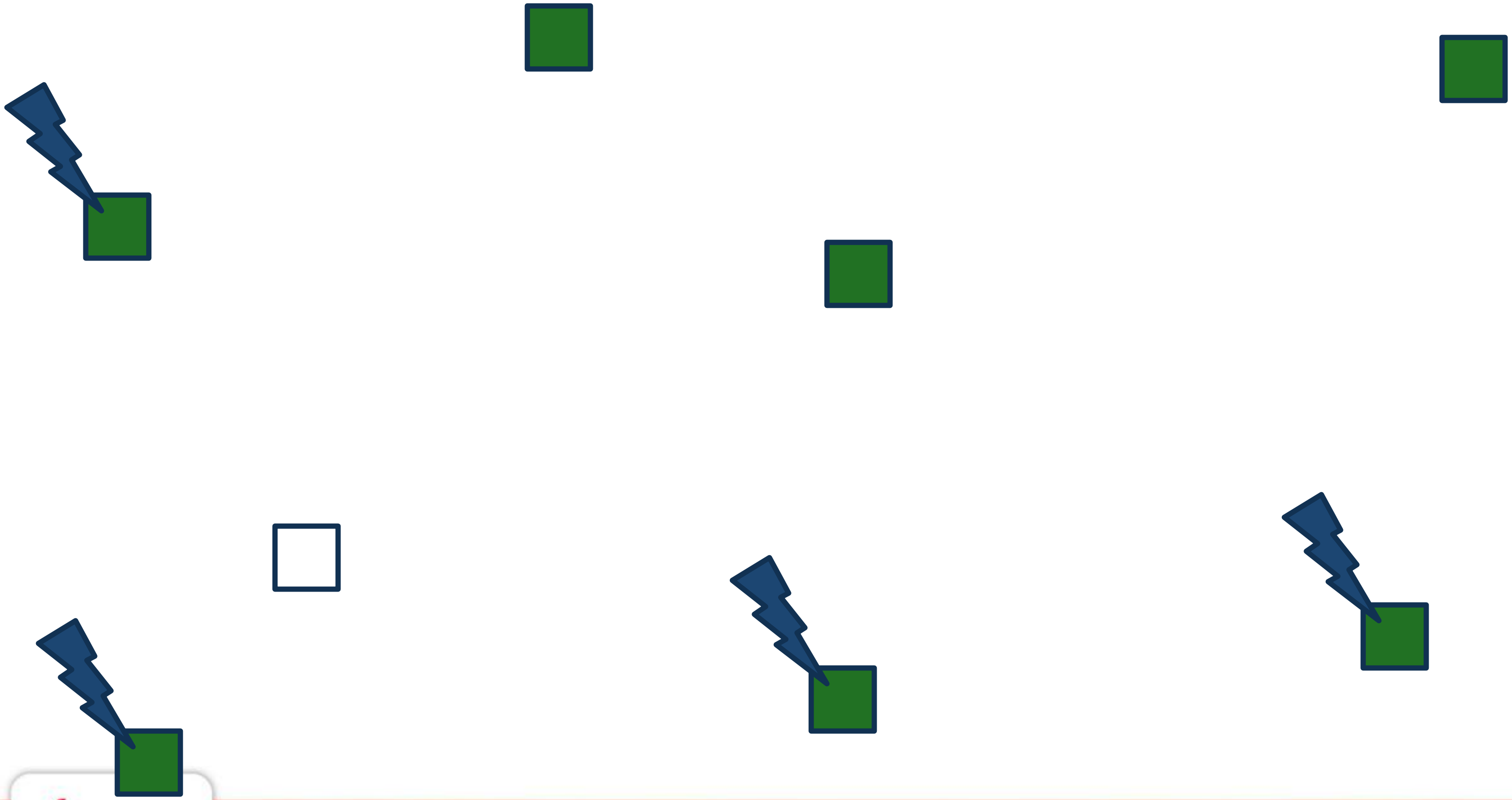
Epidemics as a Tool

Gossip-based dissemination



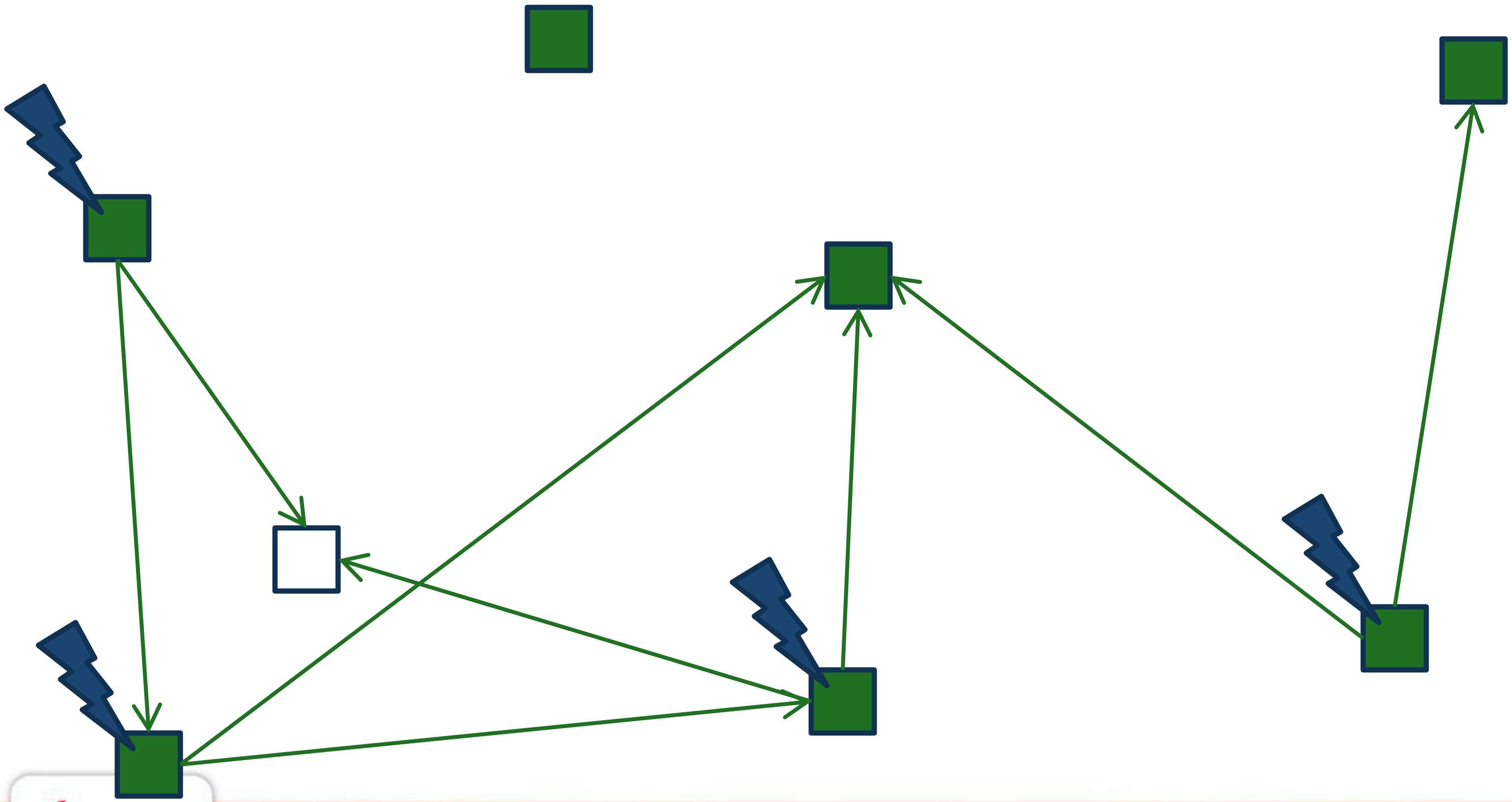
Epidemics as a Tool

Gossip-based dissemination



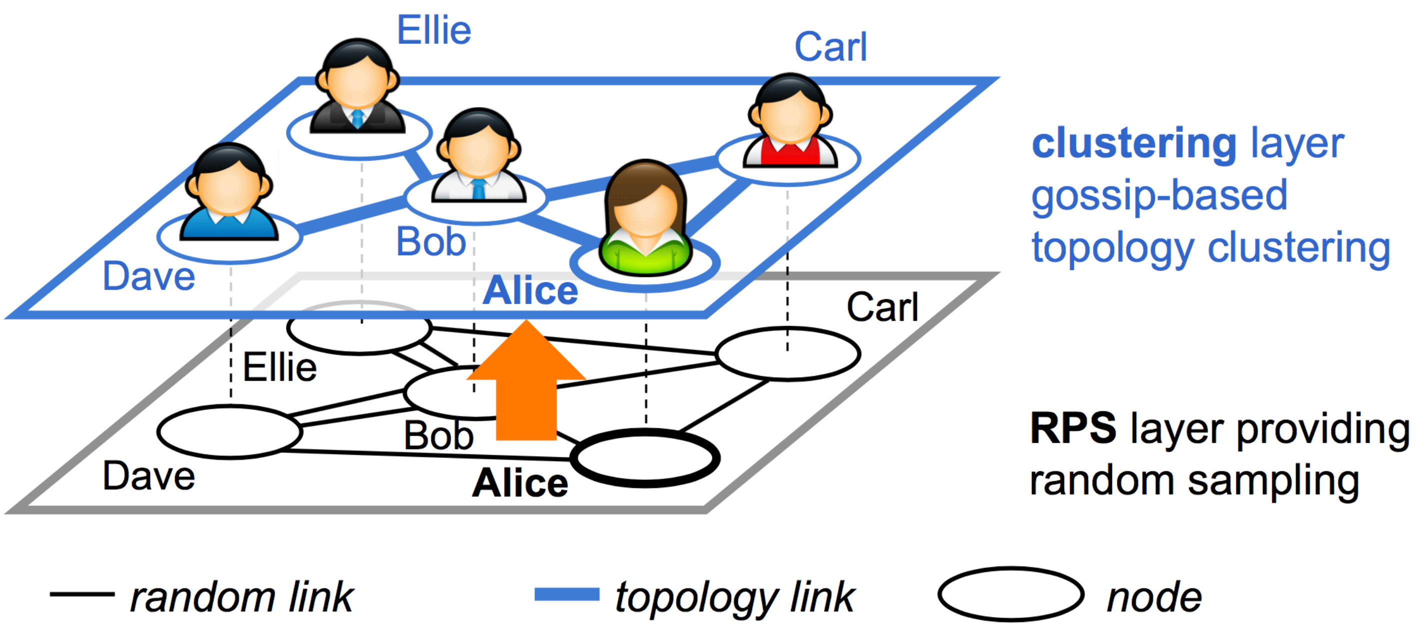
Epidemics as a Tool

Gossip-based dissemination



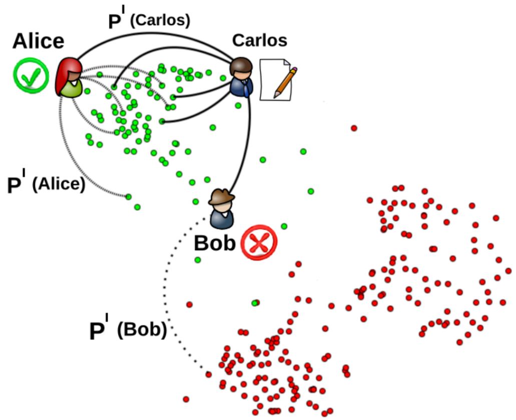
Epidemic Recommendation

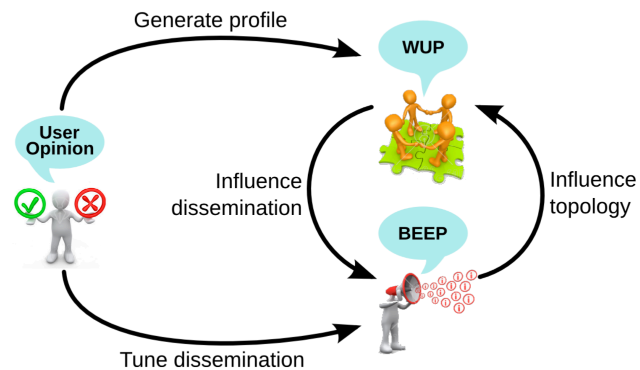
Exploit epidemic clustering to build KNN



[Fre10] Marin Bertier, Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, and Vincent Leroy. 2010. The GOSSPLE anonymous social network. In Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware (Middleware '10). Springer-Verlag, Berlin, Heidelberg, 191-211.

The Case of News Items (and beyond)



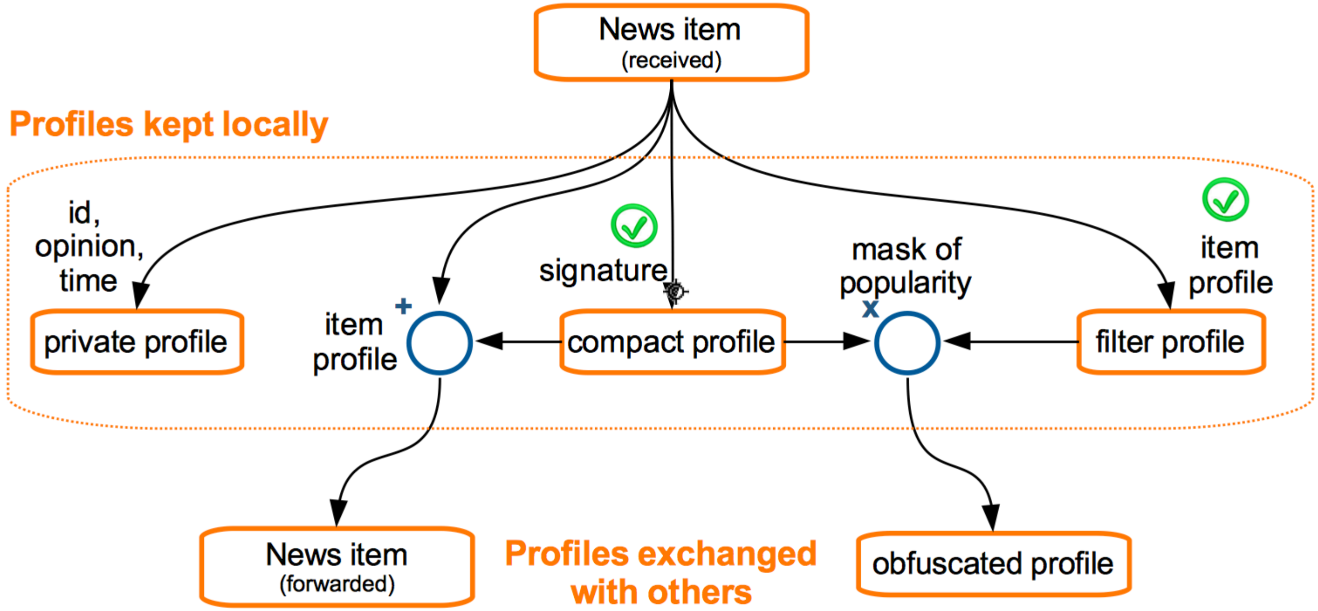


mediego

http://www.mediego.com/

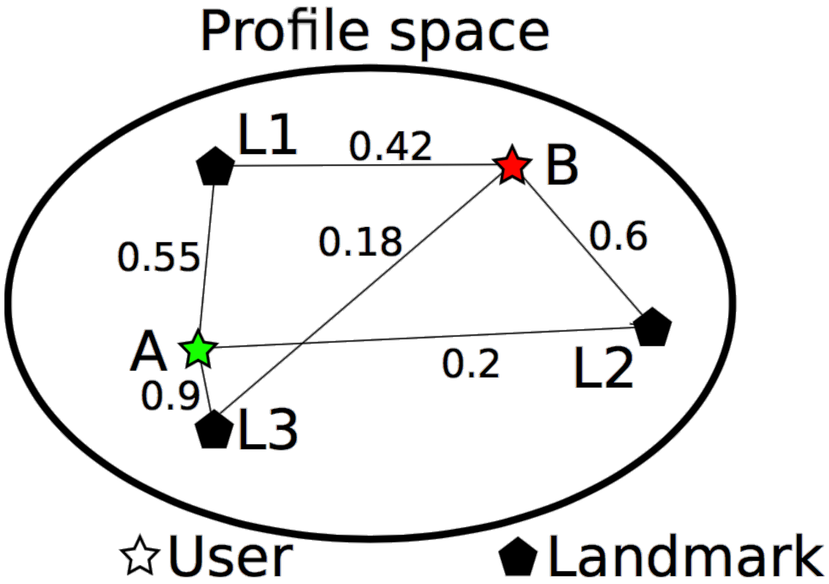
[Fre13] Antoine Boutet, Davide Frey, Rachid Guerraoui, Arnaud Jégou, Anne-Marie Kermarrec: WHATSUP: A Decentralized Instant News Recommender. IPDPS 2013: 741-752

Making Recommendation Private: Obfuscation



[Fre16] Antoine Boutet, Davide Frey, Rachid Guerraoui, Arnaud Jégou, Anne-Marie Kermarrec: Privacy-preserving distributed collaborative filtering. Computing 98(8): 827-846 (2016)

Making Recommendation Private: Landmarks



	L1	L2	L3
A	0.55	0.2	0.9
B	0.42	0.6	0.18

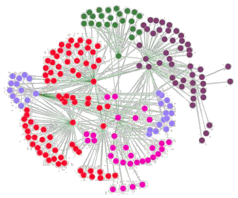
[Fre15] Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, Antoine Rault, François Taïani, Jingjing Wang: Hide & Share: Landmark-Based Similarity for Private KNN Computation. DSN 2015: 263-274

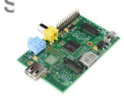
A Slide to Bring Home

Decentralize computation on private data

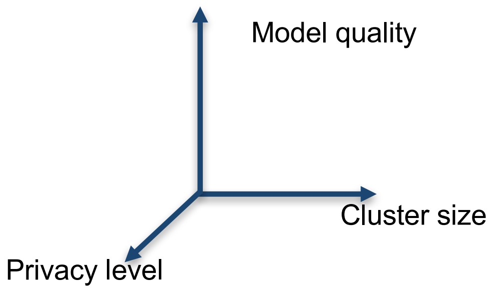
- Cluster users by interest, privacy level, …
- Build local models
- Aggregate models at higher levels

Create networks of



- Browsers 
- Set-top boxes 
- Small/Tiny devices (plug-computers, smartphones, sensors)  

Explore tradeoff spaces



Build Private Applications

- Recommendation
- Aggregation
- Personalized Services
- Inter-Silo Analytics