Computing Graphs at Scale

Felix Cuadrado

Centre for Intelligent Sensing
Queen Mary University of London

CIS centre for intelligent sensing

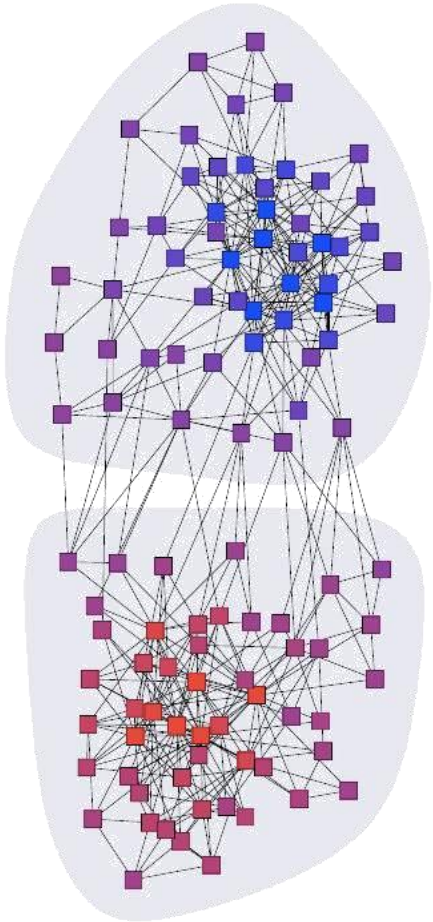
Queen Mary University of London

Outline

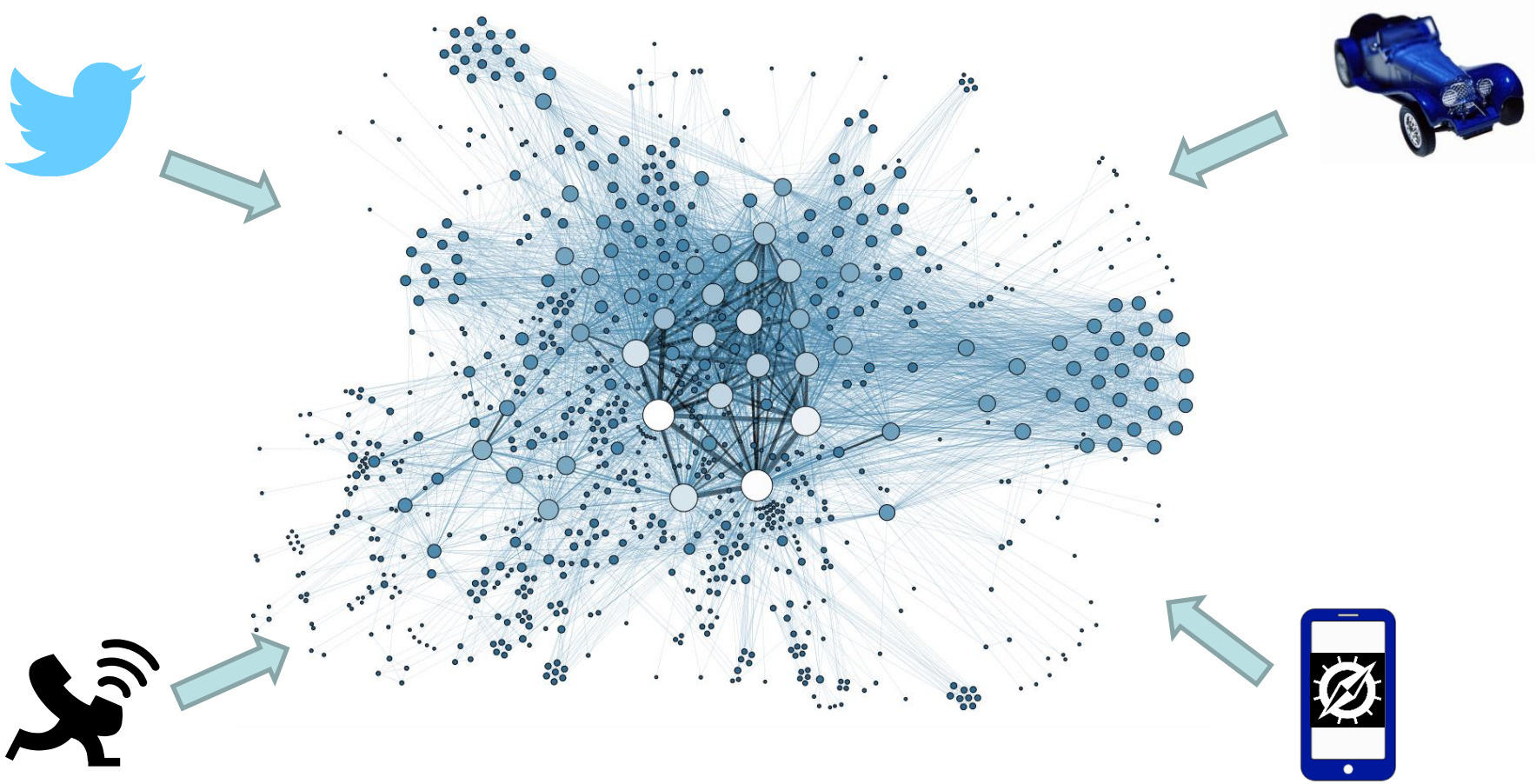
- Introduction to graphs
- Distributed graph processing
- Managing large dynamic graphs

Graph Definition

- A graph G = (V,E), where
  - V represents the set of vertices (nodes)
  - E represents the set of edges (links)
  - Both vertices and edges may contain additional information
- Different types of graphs:
  - Directed vs. undirected edges
  - Presence or absence of cycles

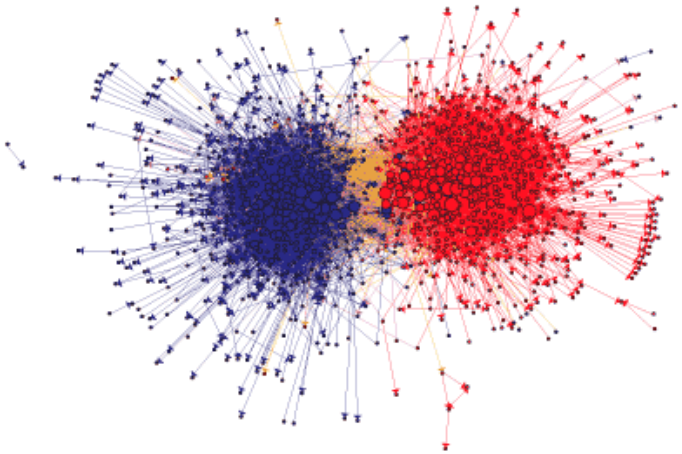


Graphs model real world sources



Real-World Graph Applications

- Finding shortest paths
  - Routing Internet traffic and UPS trucks
- Finding Max Flow
  - Airline scheduling
- Bipartite matching
  - Monster.com, Match.com
- Rank search results
  - PageRank
- Identify leaders in a community
  - Measure influence
- Identify “special” nodes and communities
  - Breaking up terrorist cells
  - Track spread of ebola propagation



Graph-Processing Algorithms

- Literature survey of 10 top research conferences in databases and systems: SIGMOD, VLDB, WWW … [1]

Class	Examples	%
Graph Statistics	Diameter, PageRank	16.1
Graph Traversal	BFS, SSSP, DFS	46.3
Connected Component	Reachability, BiCC	13.4
Community Detection	Clustering, Nearest Neighbor	5.4
Graph Evolution	Forest Fire Model, PAM	4.0
Other	Sampling, Partitioning	14.8

Outline

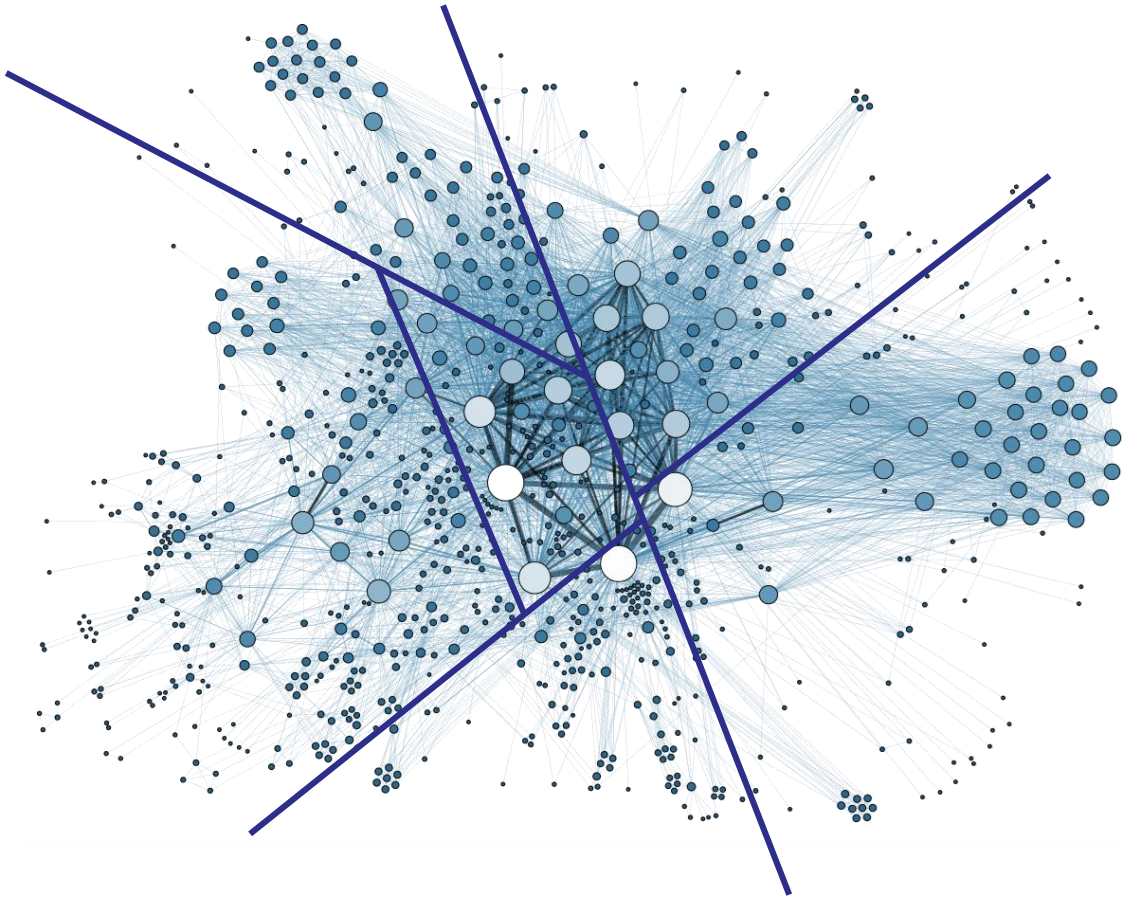
- Introduction to graphs
- Distributed graph processing
- Managing large dynamic graphs

Size of real world graphs

- World Wide Web:
  - Several billions of pages and links
- Twitter's social graph has billions of follower relationships
- Facebook has one billion users

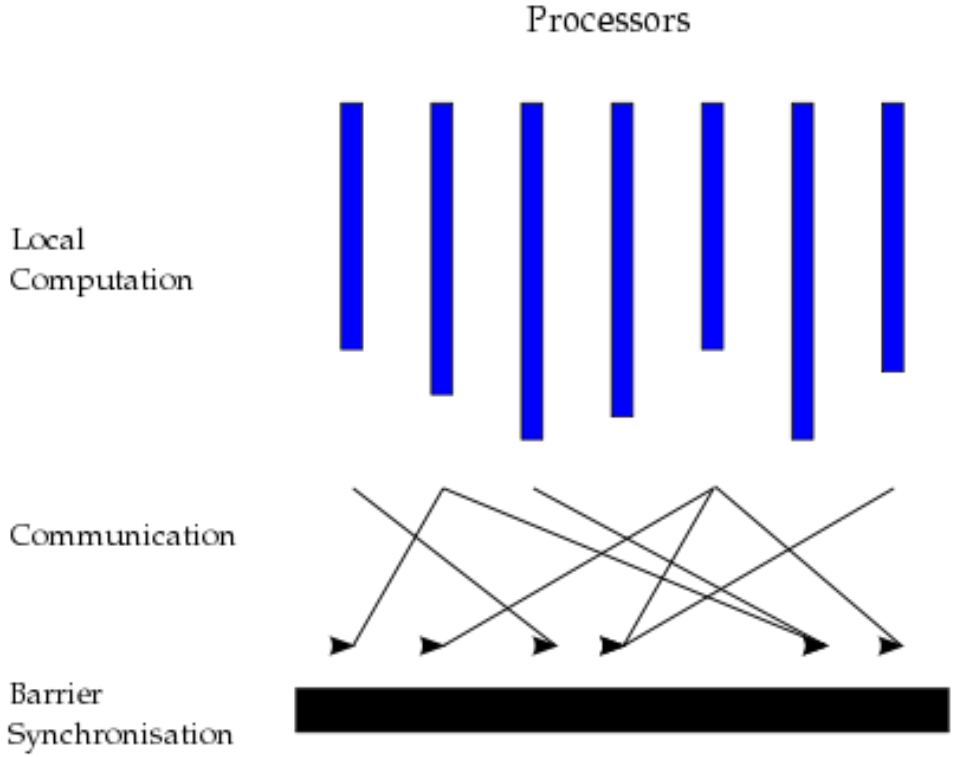
- … in memory footprint: TBs to PBs

Distributed graph processing



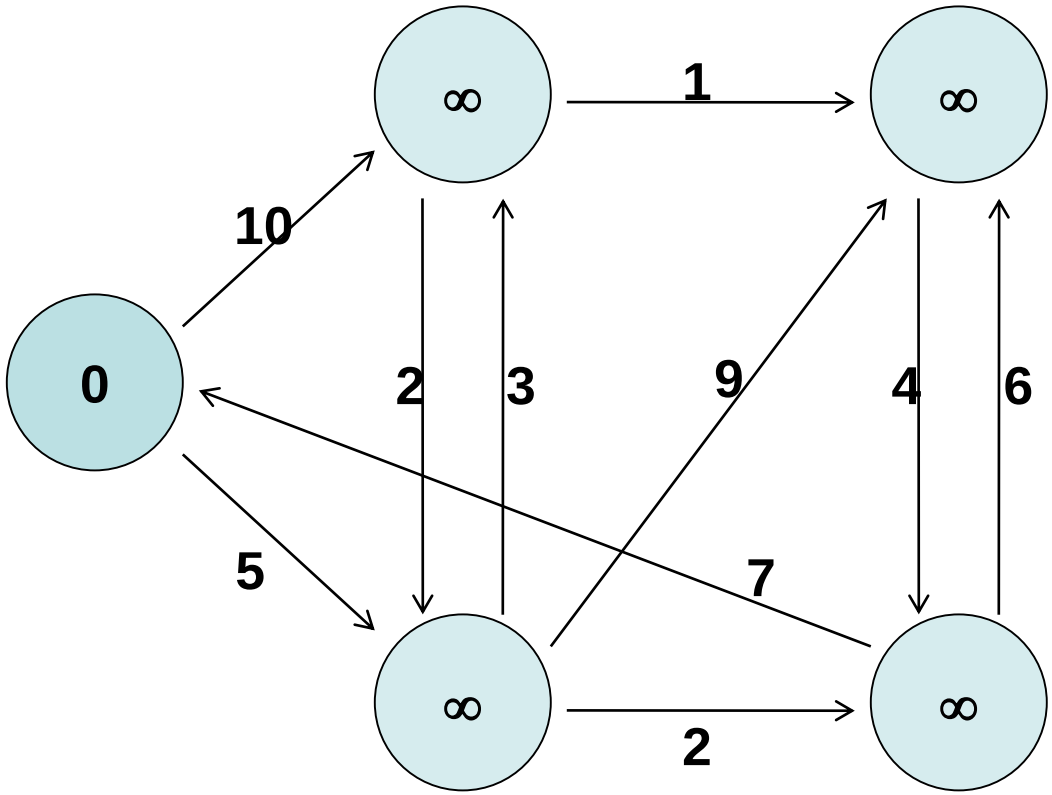
Pregel: BSP Vertex-Centric Computation

- "Think like a vertex" [2]
- Inspired by Valiant's Bulk Synchronous Parallel model [3]



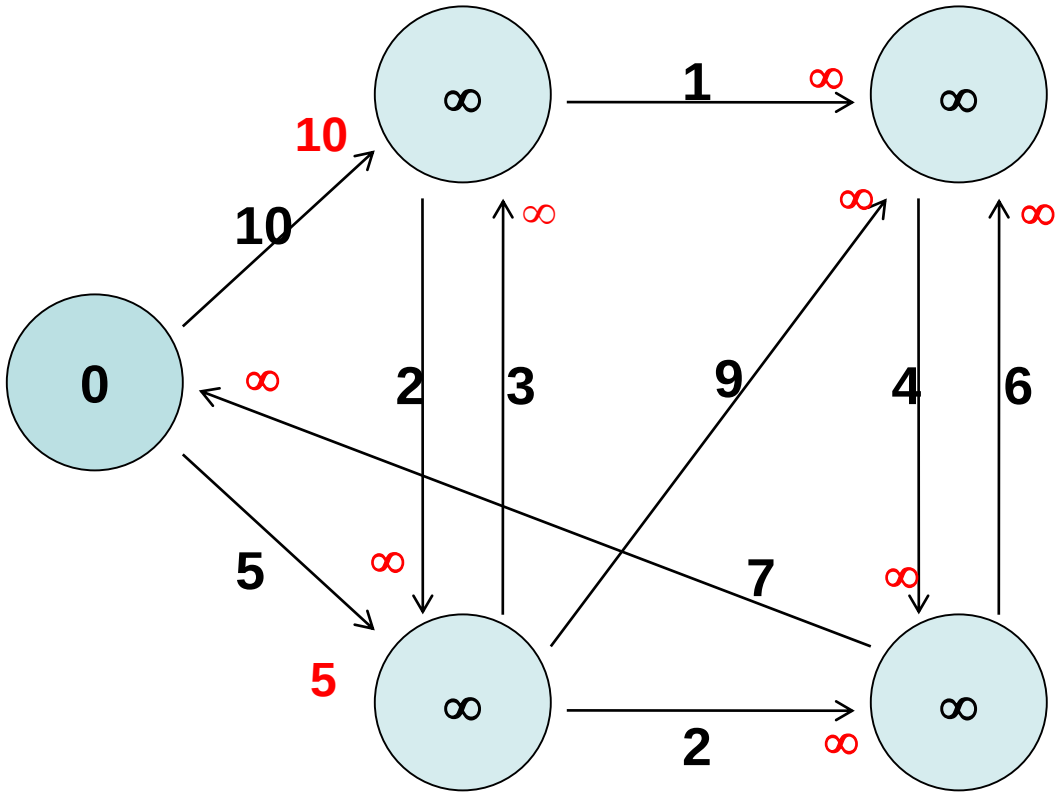
Example: SSSP – Parallel BFS in Pregel

- Superstep 1



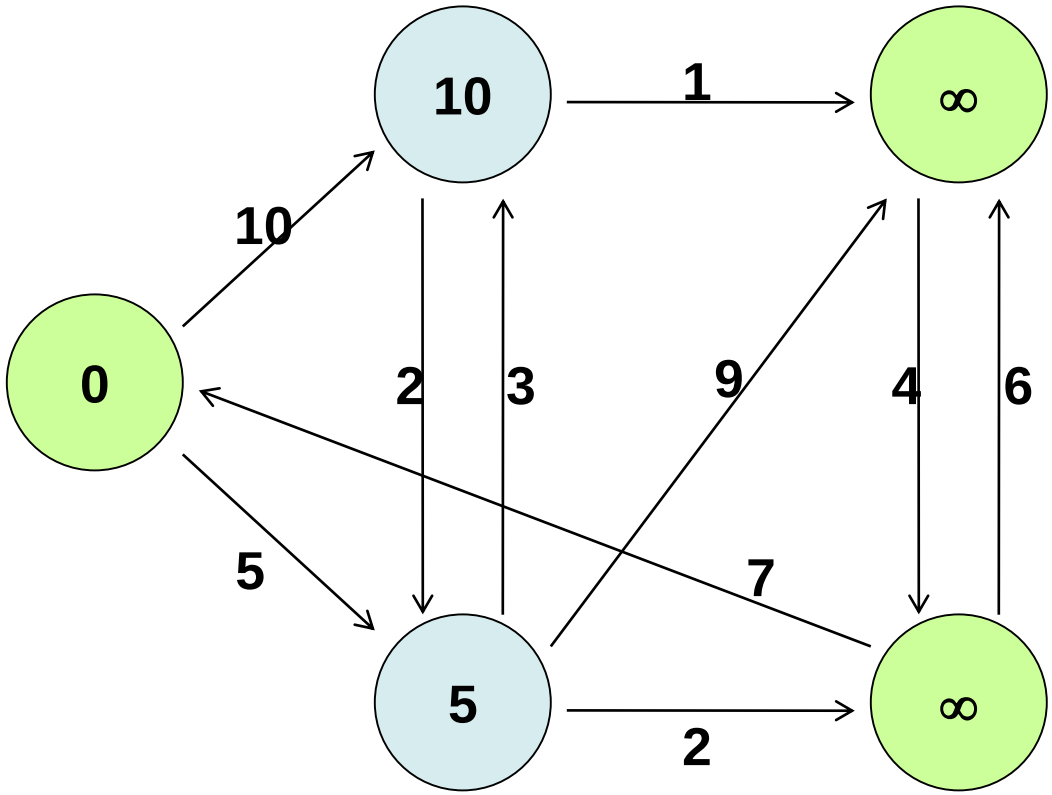
Example: SSSP – Parallel BFS in Pregel

- Superstep 1 - **Messaging**



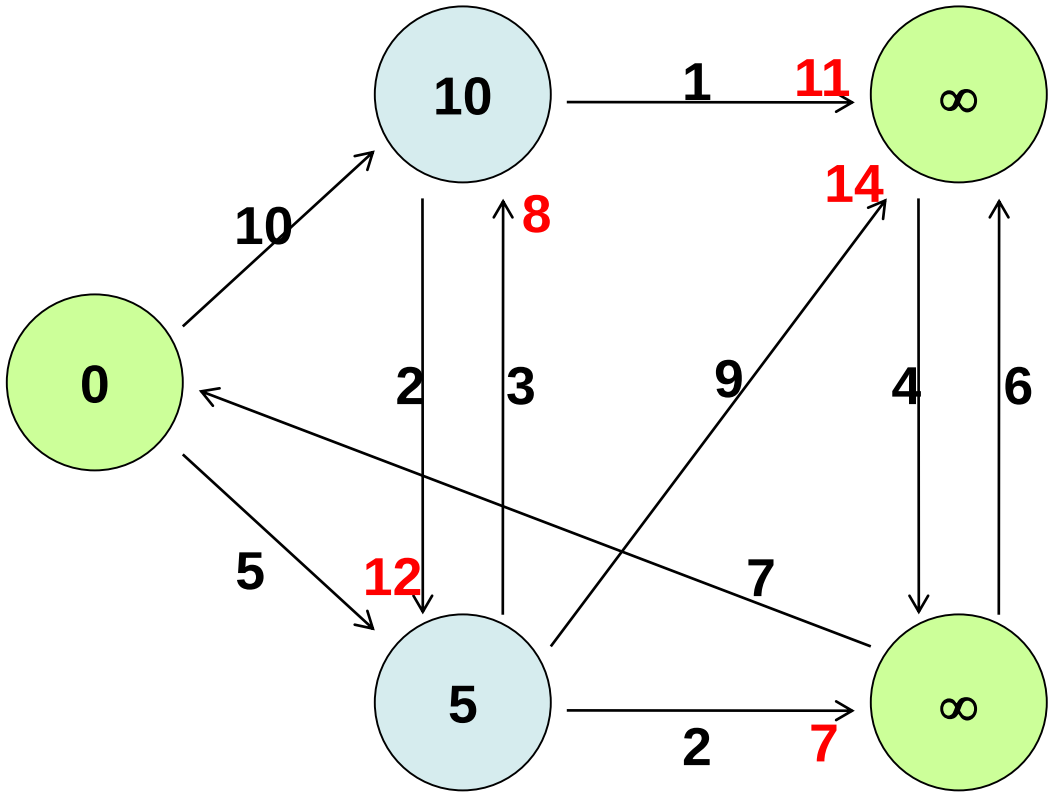
Example: SSSP – Parallel BFS in Pregel

- Superstep 2



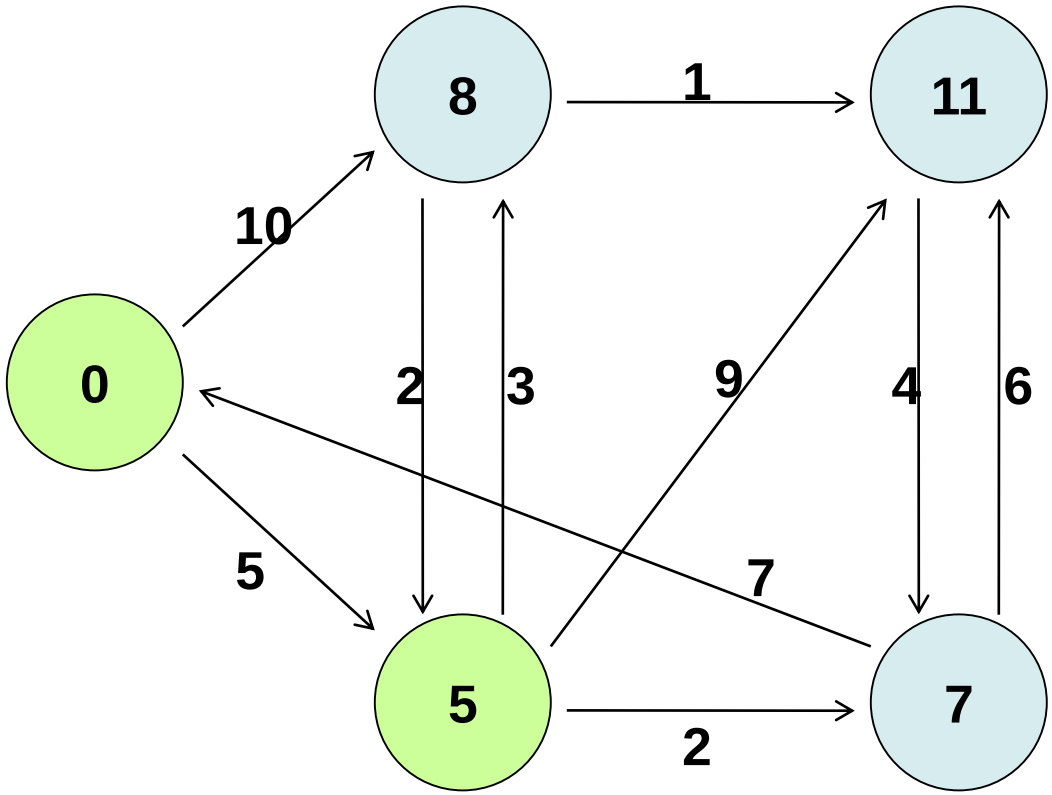
Example: SSSP – Parallel BFS in Pregel

- Superstep 2 - **Messaging**



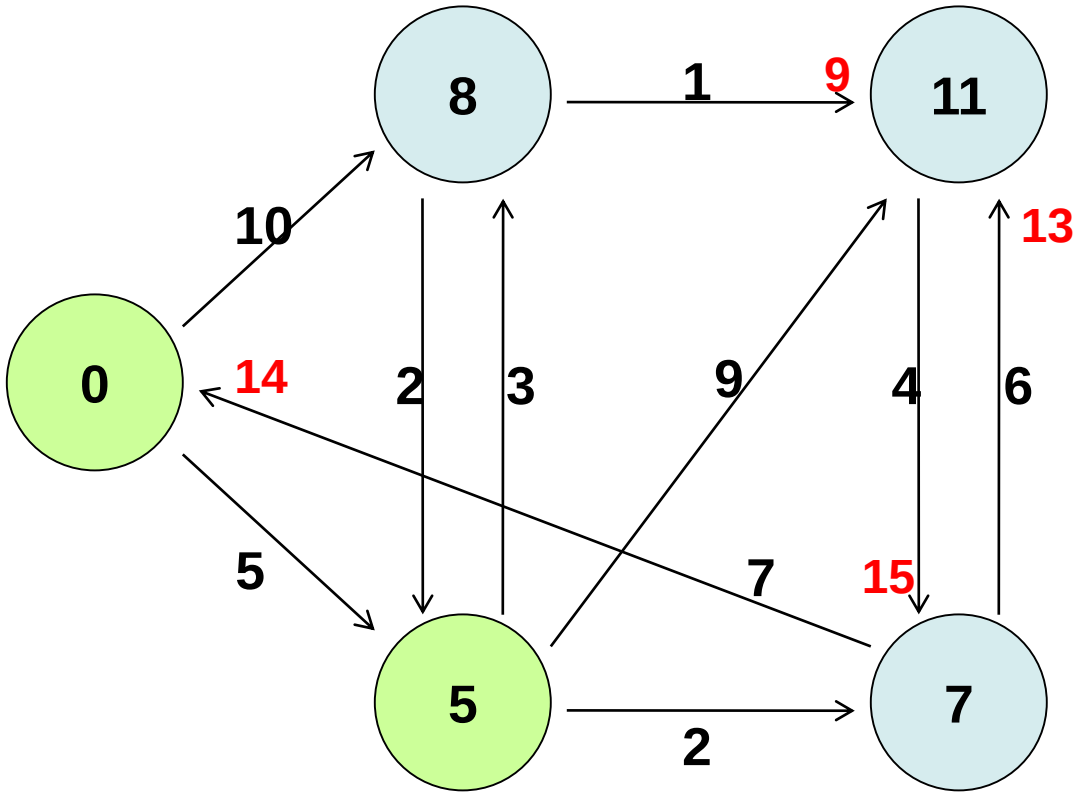
Example: SSSP – Parallel BFS in Pregel

- Superstep 3



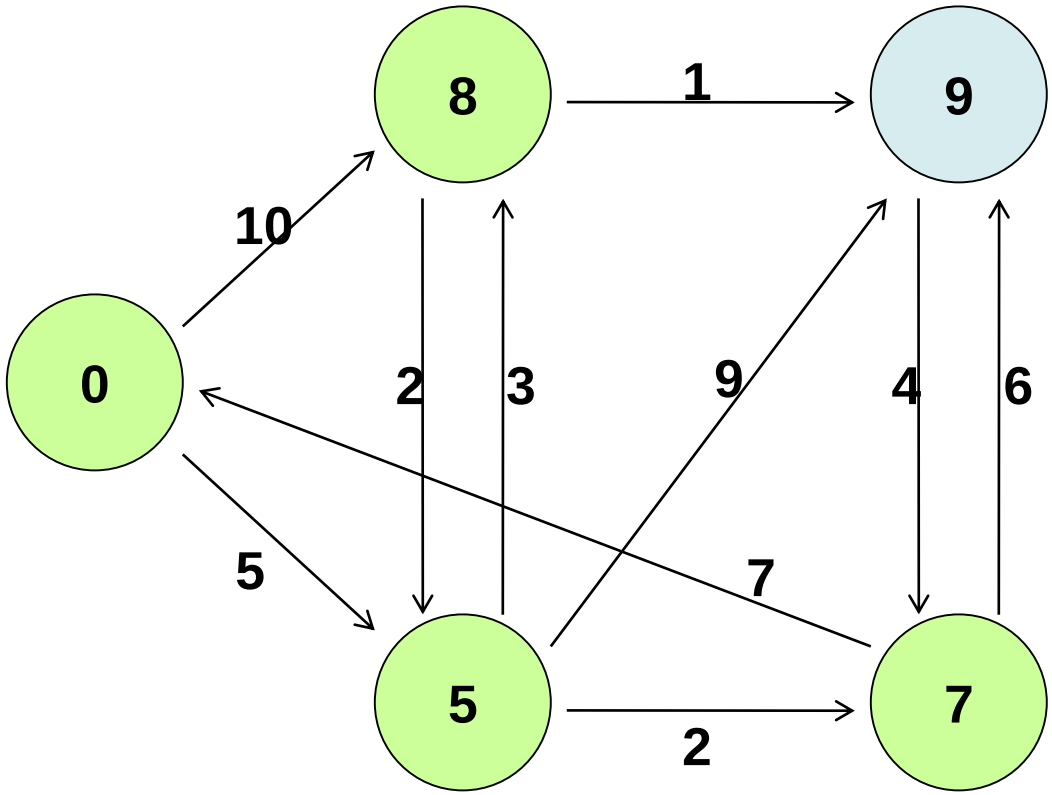
Example: SSSP – Parallel BFS in Pregel

- Superstep 3 - **Messaging**



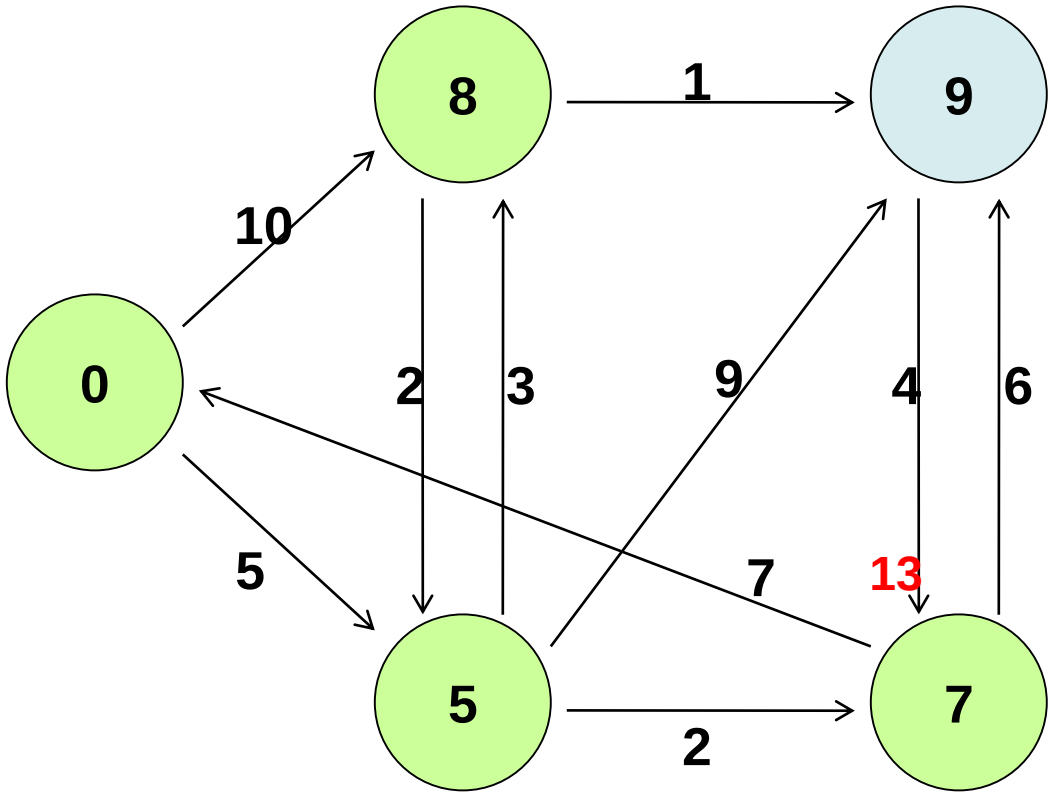
Example: SSSP – Parallel BFS in Pregel

- Superstep 4



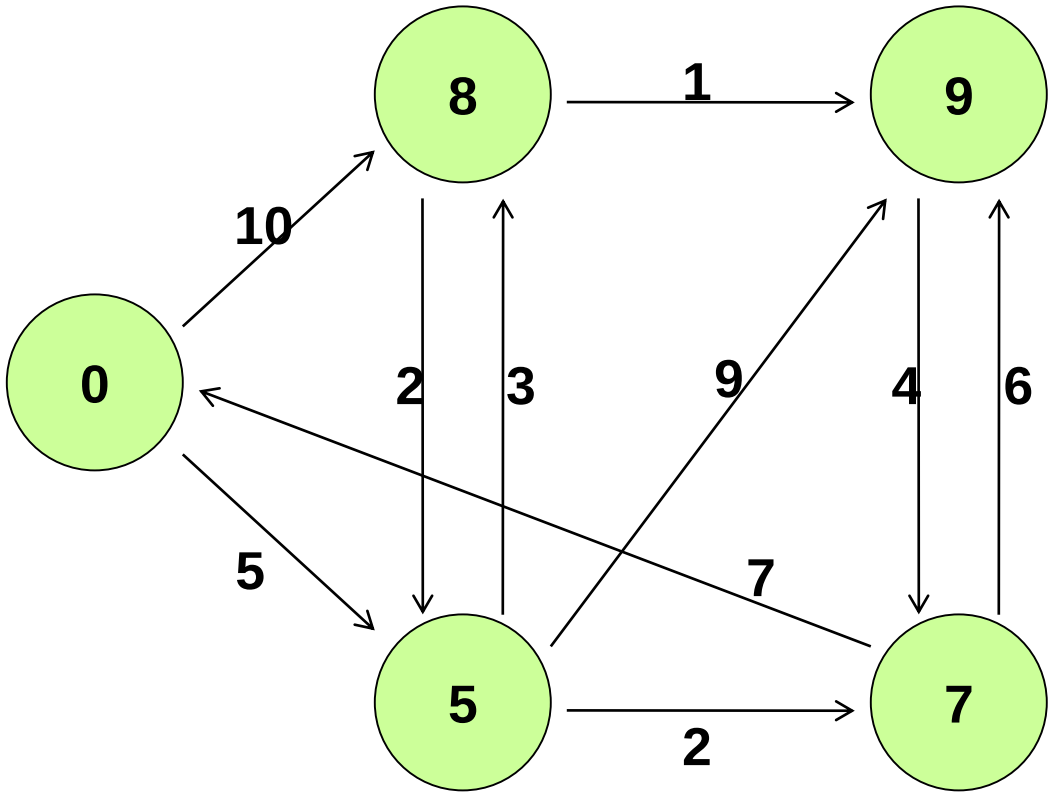
Example: SSSP – Parallel BFS in Pregel

- Superstep 4 - **Messaging**



Example: SSSP – Parallel BFS in Pregel

- Superstep 5

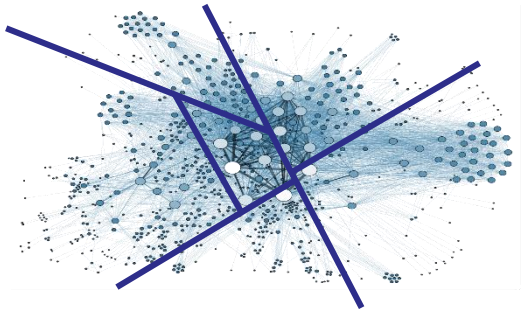


How Pregel parallelises graph computations

- Goal: compute the graph algorithm when the graph is large (**scale**), in a reasonable time (**performance**)
- Approach:
  - **Partition** the graph among the computation nodes
  - Assign each worker to compute the function on its nodes
- Performance pitfalls
  - Uneven **load balancing**: some nodes wait for another
  - **Overheads** from the parallel execution: time wasted not doing the computation work

Balanced K-way graph Partitioning

- Given a graph G = (N, E)
  - N = nodes (or vertices),
  - E = edges



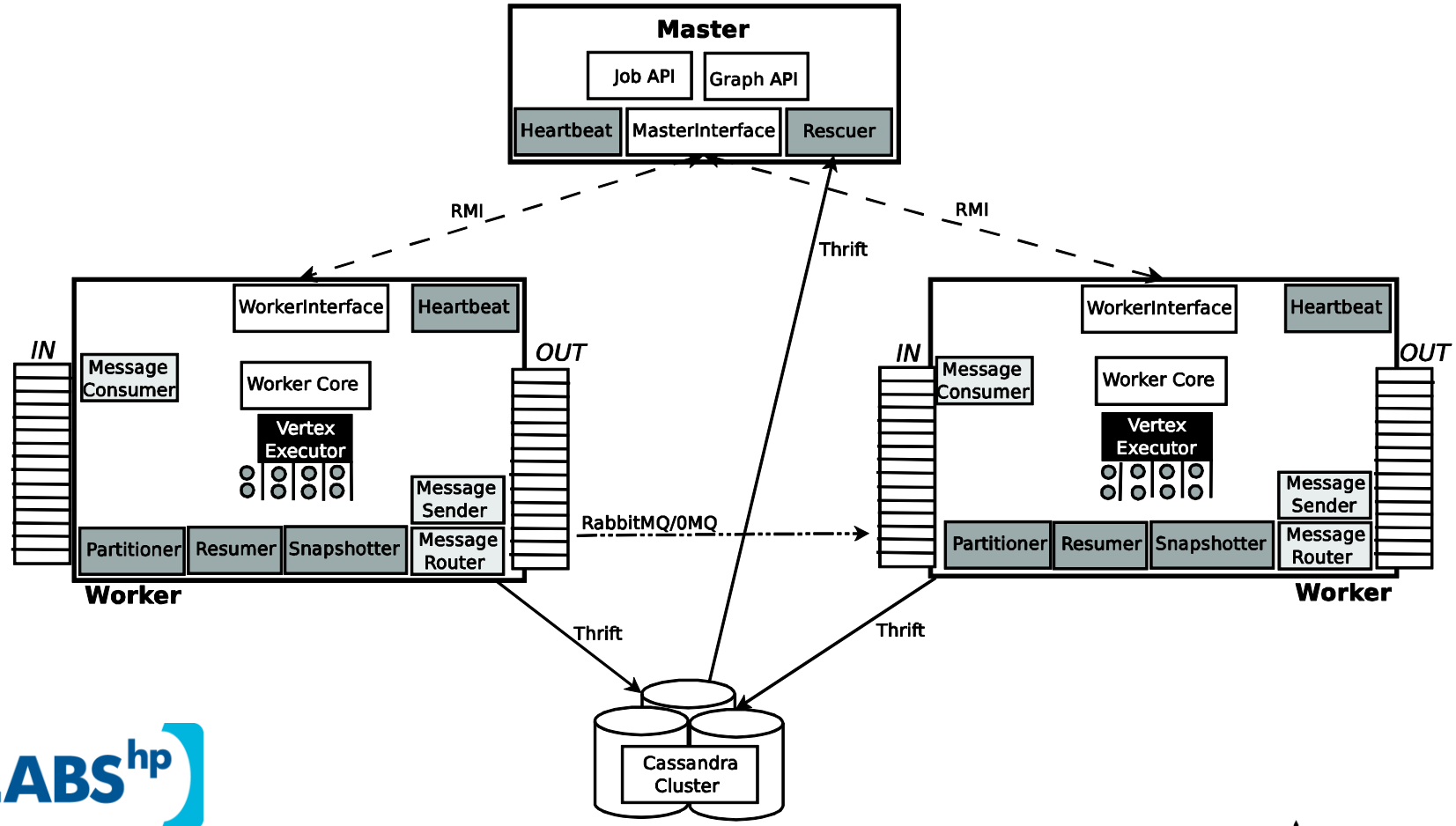
- Choose a partitioning $N = N_1 \cup N_2 \cup \ldots \cup N_k$ such that
  - The cardinality of each N_i is "about the same"
  - The sum of all cut edges (edges connecting nodes from different partitions N_i and N_j) is minimized
- Problem is NP-Complete

Graph partitioning effect on performance

- In vertex-centric distributed computations partition is key
  - Balanced partitioning -> Load balancing
  - Not all edges are equal (some are cut, and take orders of magnitude more time to send messages through)
- Communication is the bottleneck
  - time required to send messages across cut edges is by far the dominant factor in computation time (>80%)

- …but scalability reasons make the default solution is random partitioning (uniform hashing).

System Architecture



LABS hp

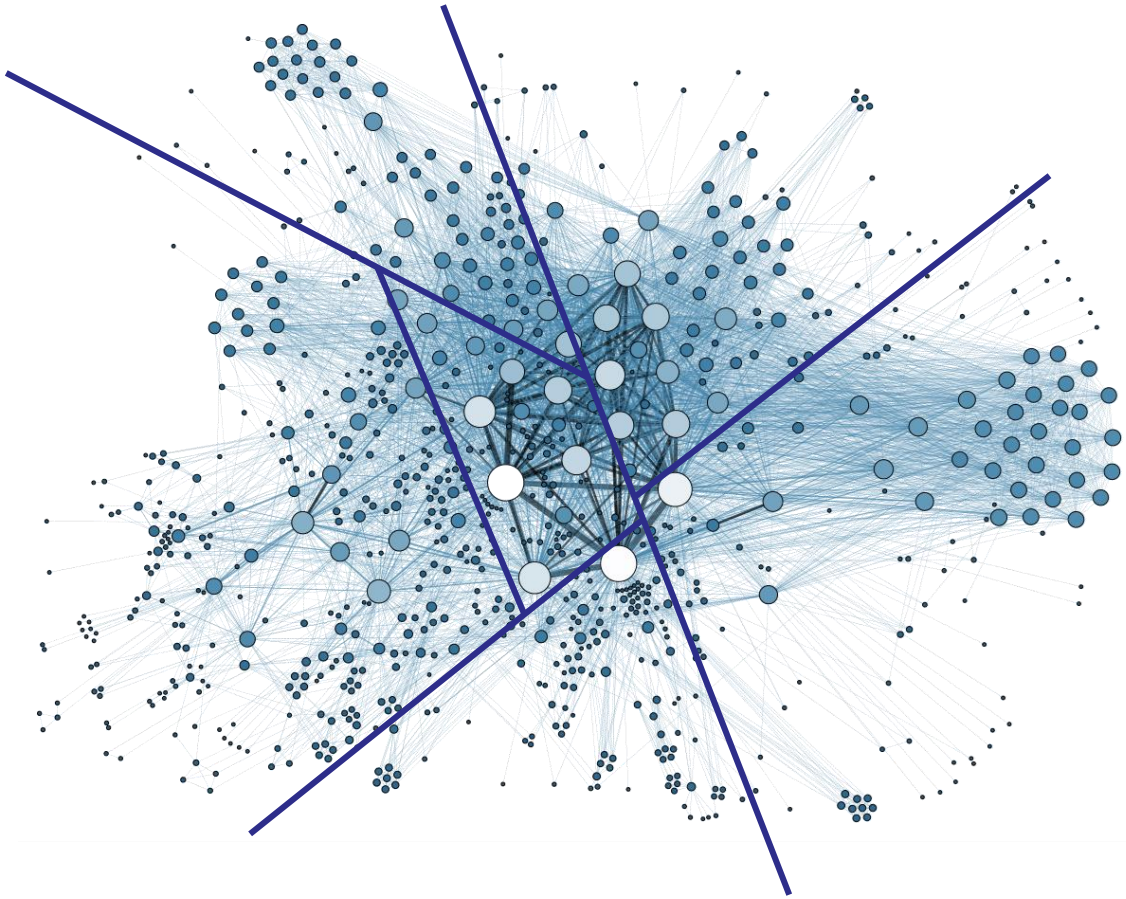
CIS centre for intelligent sensing

Queen Mary University of London

Outline

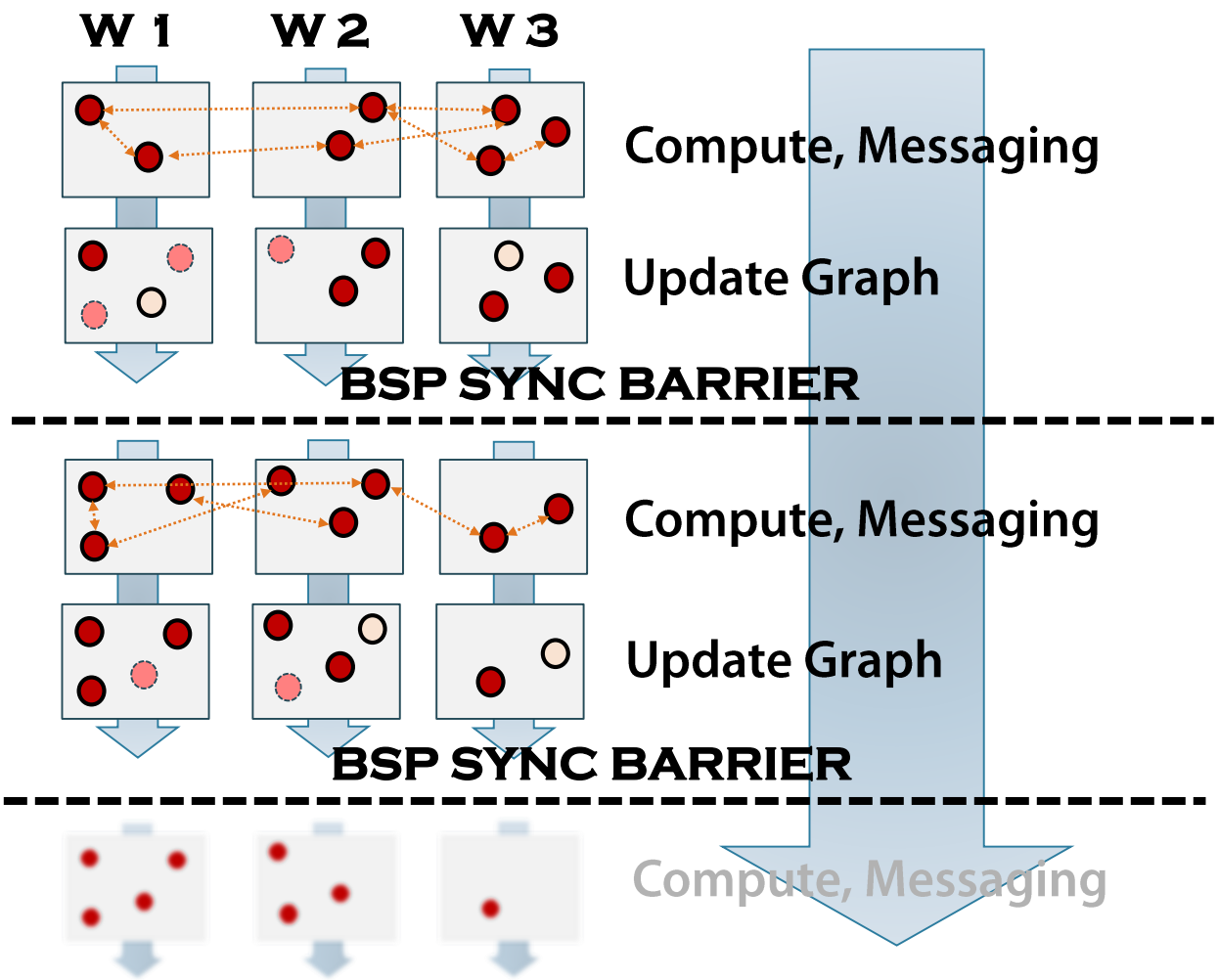
- Introduction to graphs
- Distributed graph processing
- Managing large dynamic graphs

Real graphs are dynamic

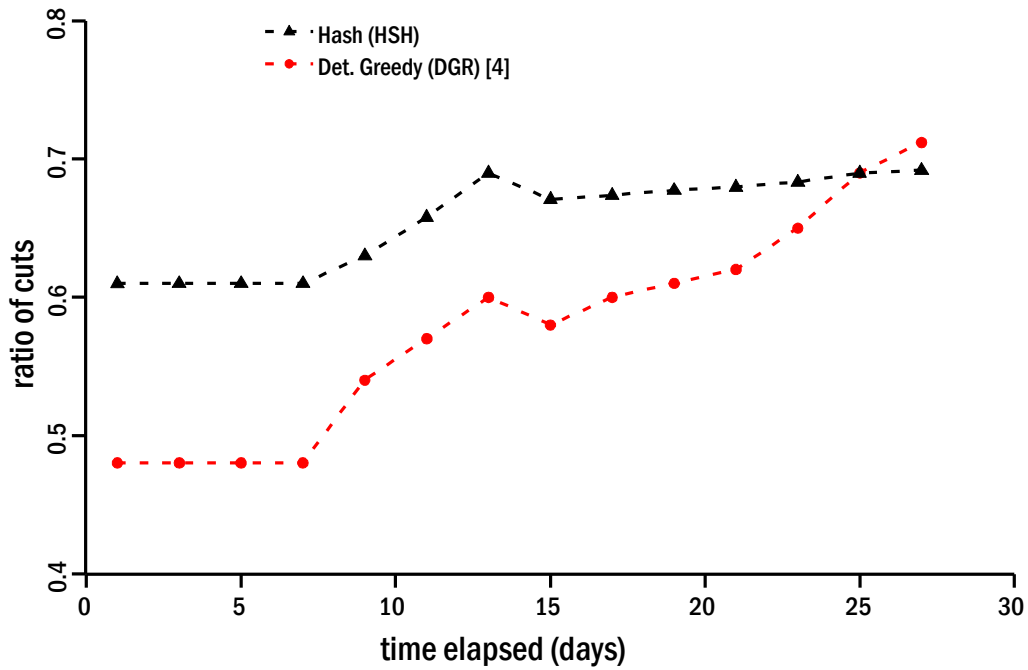


Graph changes with time

Dynamic Graph Computation Cycle



Partitioning dynamic graphs



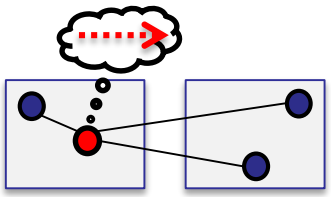
Initial partitioning is not suitable for dynamic graphs

Repartitioning carries a high cost

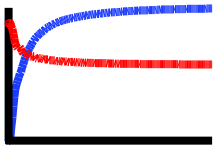
Approach: iterative adaptive vertex migrations

Algorithm Key principles [5]

Migrate where your neighbors are

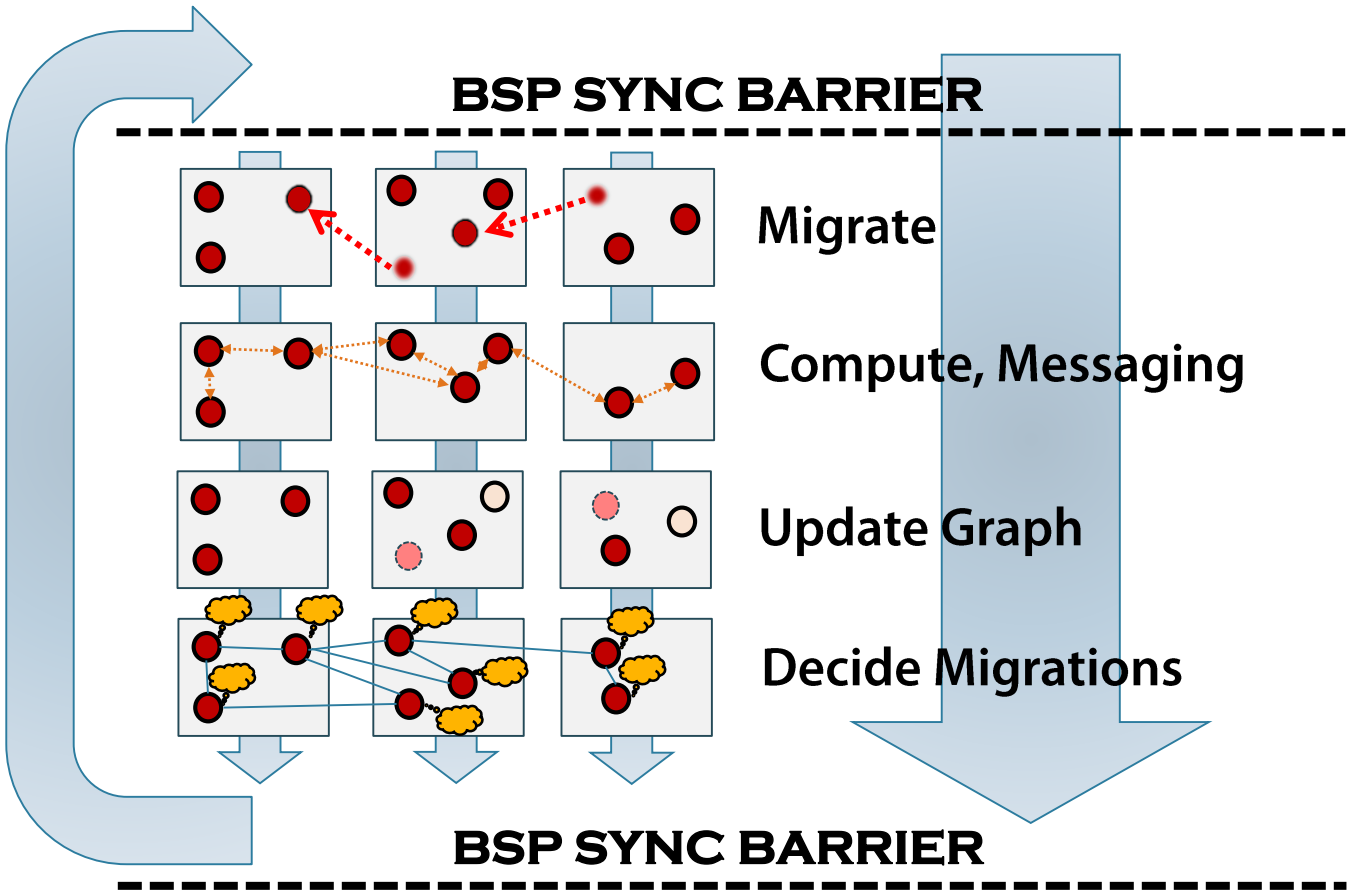


Iterative algorithm



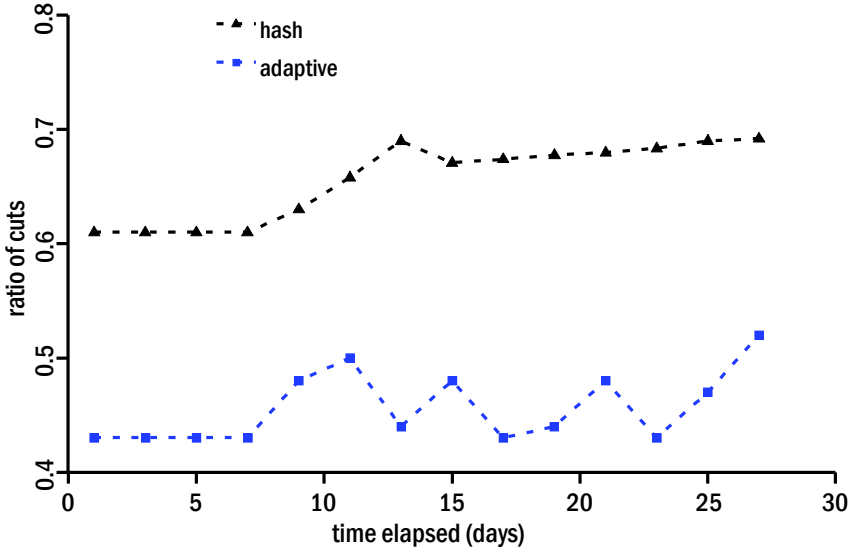
Migration **Quotas**

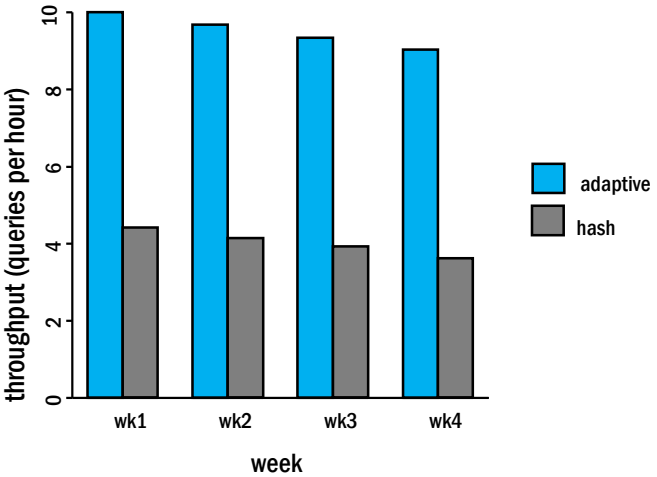
Adaptive Migration Integration



CDR Analysis

One month of Call Detail Records
21 million nodes from phone calls
Nodes and edges expire with a sliding window

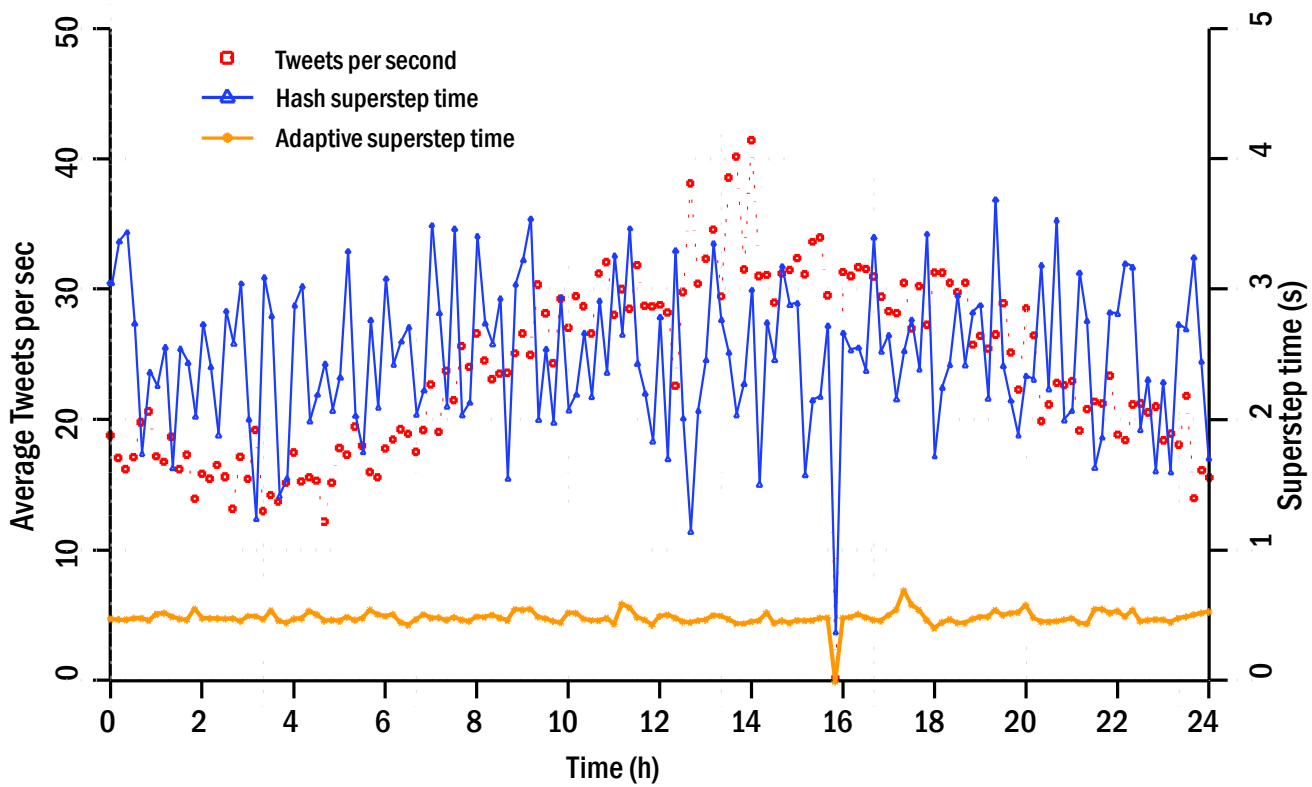




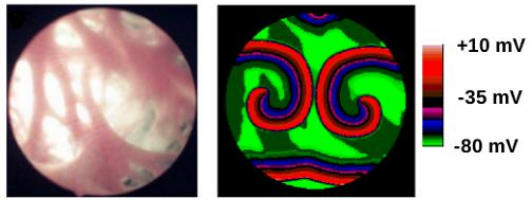
Quality of Partitioning → Maximum clique performance

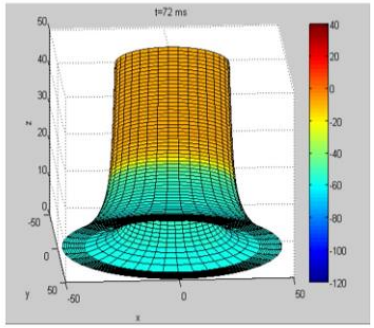
Adapting Real Time Social Graphs

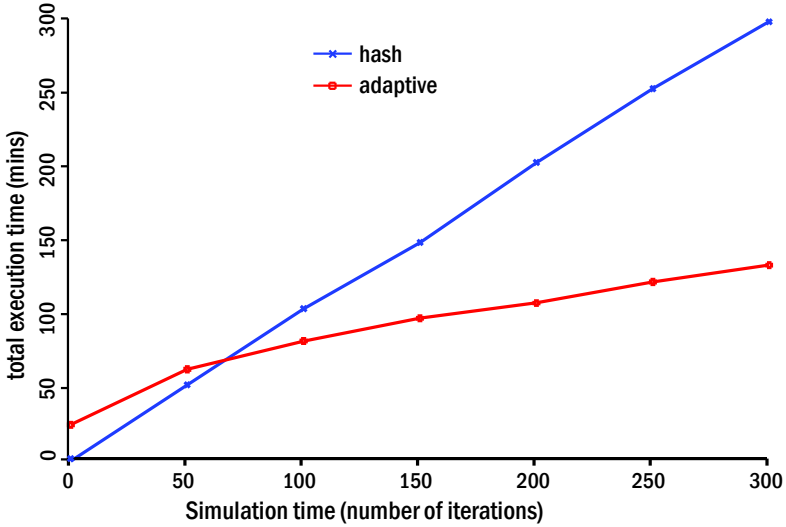
TunkRank (User influence metric) computation
Dataset: one week of tweets published from London



Biomedical Simulation







Heartbeat Stem Cells Simulation
110M Nodes
64 nodes cluster
3TB memory footprint

Summary

- Large graphs require parallel computation
  - Distributed systems research challenges include graph partitioning, load balancing, minimising communications overhead
- Partitioning is a key aspect of the performance of large-scale graph processing
  - Ratio of cuts ←→ Performance
  - Scalability
  - Performance overhead
- Dynamic graphs make the problem more challenging
  - Adaptive migration algorithm for dynamic graph partitioning

References

[1] Y. Guo, M. Biczak, A. L. Varbanescu, A. Iosup, C. Martella, and T. L. Willke. “How Well do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis”, In Proc. of IPDPS 14

[2] G. Malewicz, M.H. Austern, A.J.C. Bik, J.C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing." In Proc. of ACM SIGMOD, 2010.

[3] L. Valiant, "A bridging model for parallel computation." Communications of the ACM 33(8), 1990.

[4] I. Stanton, and G. Kliot. "Streaming graph partitioning for large distributed graphs." In Proc. of ACM SIGKDD, 2012.

[5] L.M. Vaquero, F. Cuadrado, D. Logothetis and C. Martella, “Adaptive Partitioning for Large-Scale Dynamic Graphs”, In Proc. of IEEE ICDCS, 2014.

THANKS!!

Félix Cuadrado

http://eecs.qmul.ac.uk/~fcuadrado

felix.cuadrado@qmul.ac.uk