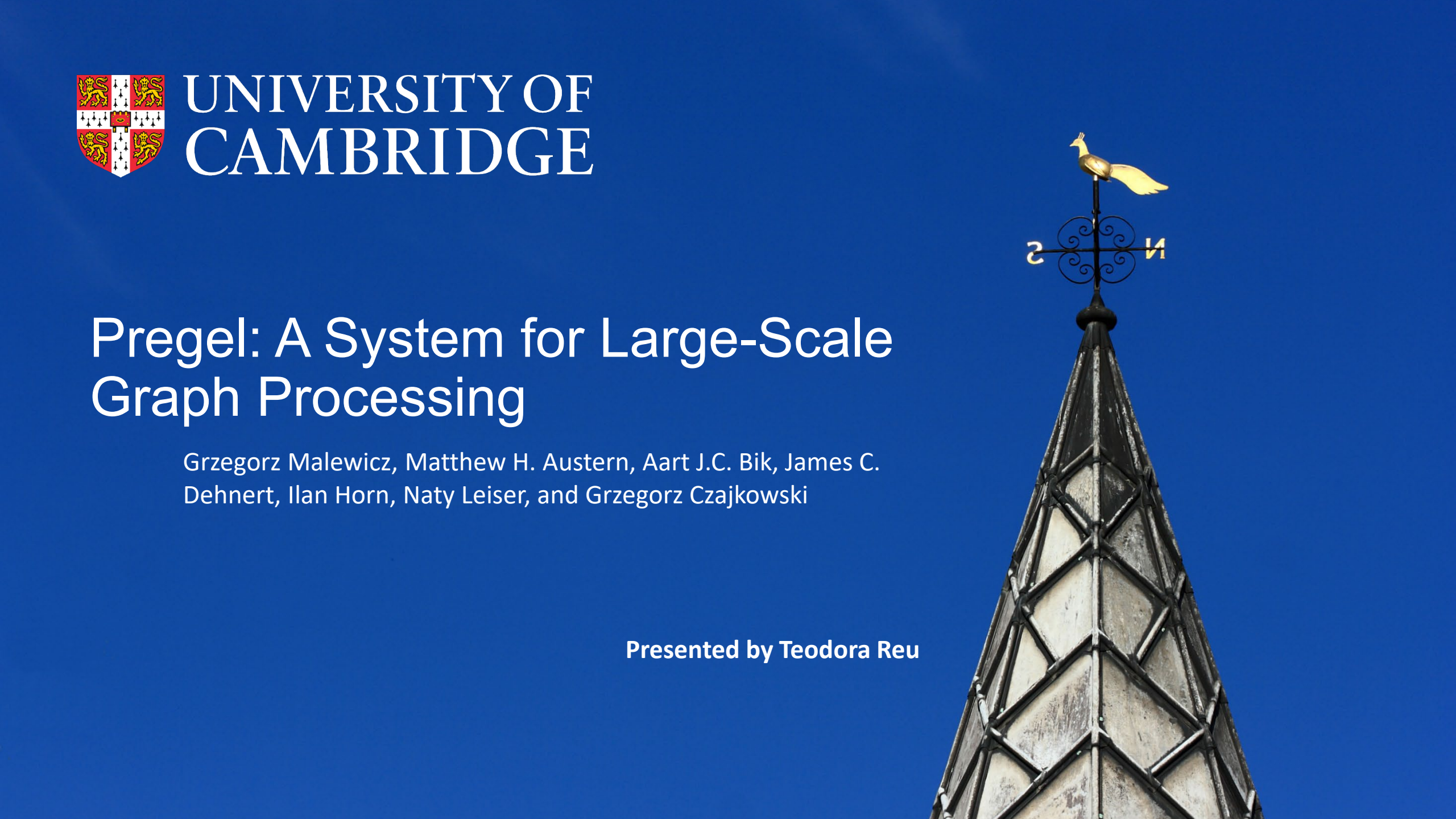


Pregel: A System for Large-Scale Graph Processing

Grzegorz Malewicz, Matthew H. Austern, Aart J.C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski

Presented by Teodora Reu

State of the art and its limitations

BGL, LEDA, NetworkX, JDSL, Stanford GraphBase, FGL

- **Single-computer graphs => not distributed**

Parallel BGL, CGMgraph

- **Distributed but do not address fault tolerance**

MapReduce

- **Can be used to mine large Graphs but leads to suboptimal performance**

SQL

- **Not ideal for graph algorithms**

Design Decisions

An **ideal implementation** of algorithm to process large graph would be:

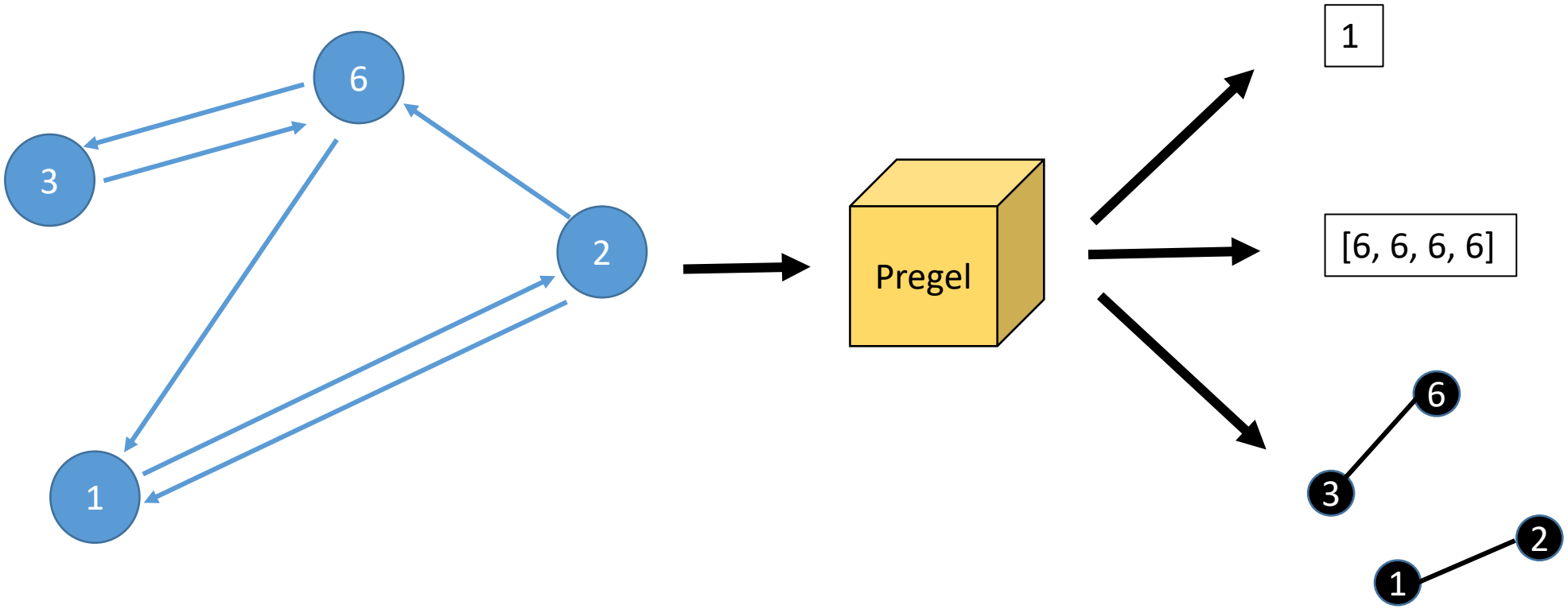
- **Distributed**
- **Fault-Tolerant**
- **Algorithm-Flexible**
- **Scalable**

Pregel:

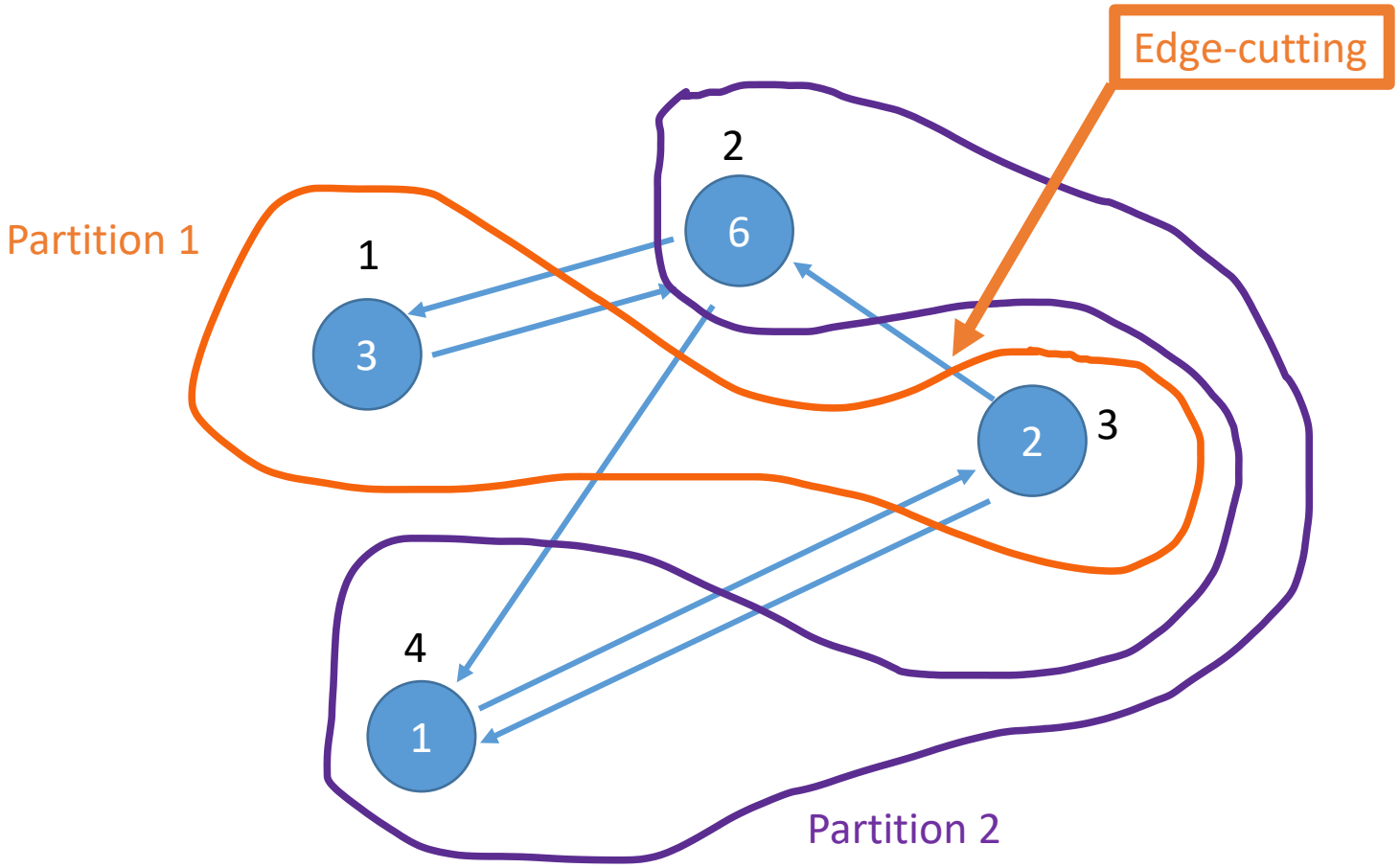
- **Inspired by Valiant's Bulk Synchronous Parallel Model [1]**
- **Sequence of iterations called super steps**
- **Voting system for nodes to halt**
- **C++ implementation**

UNIVERSITY OF CAMBRIDGE

How does it work?

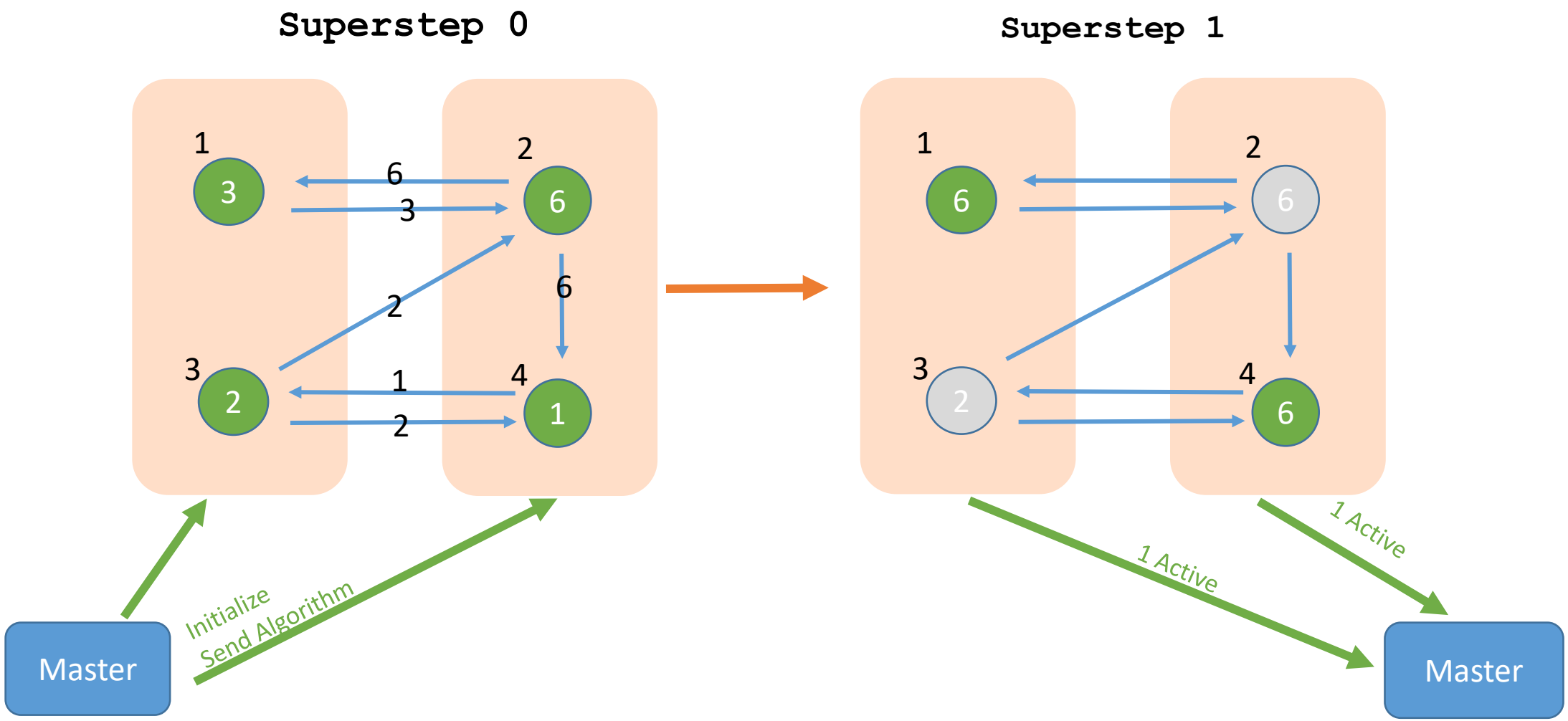


UNIVERSITY OF CAMBRIDGE

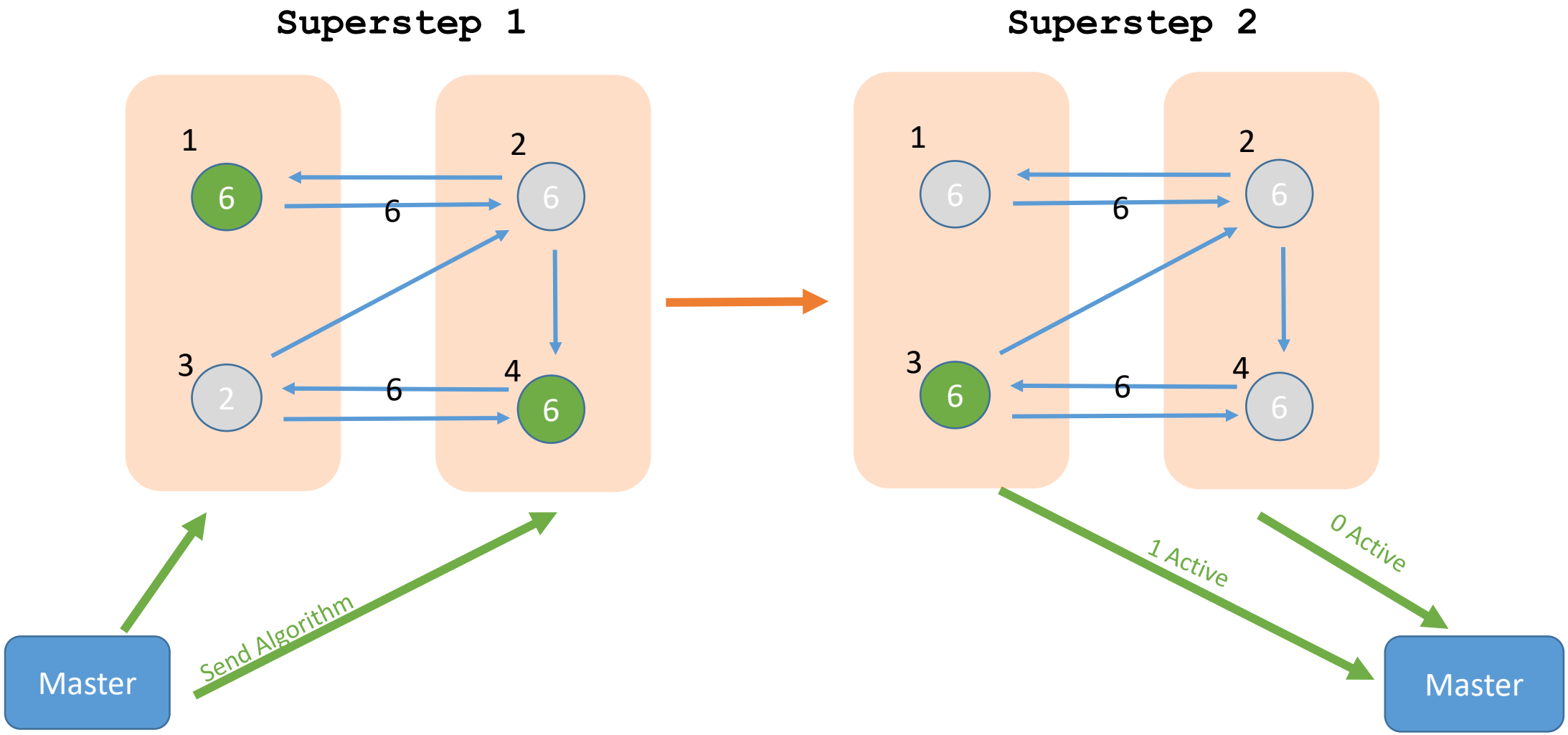


Master

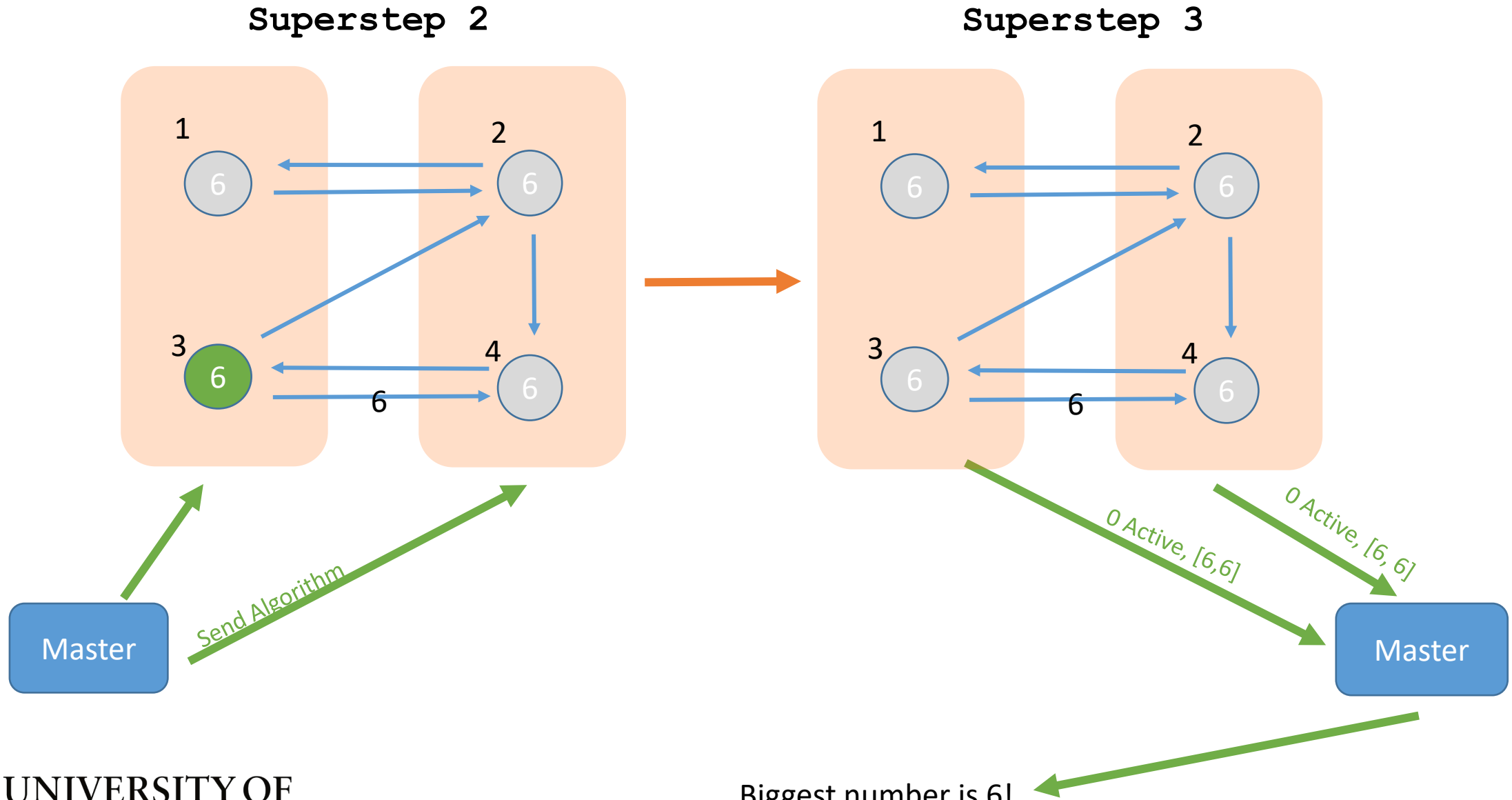
- Assigns unique IDs
- Partitions based on the modulo N value of the ID
- Maintains a list of alive workers and their value



UNIVERSITY OF CAMBRIDGE



UNIVERSITY OF CAMBRIDGE



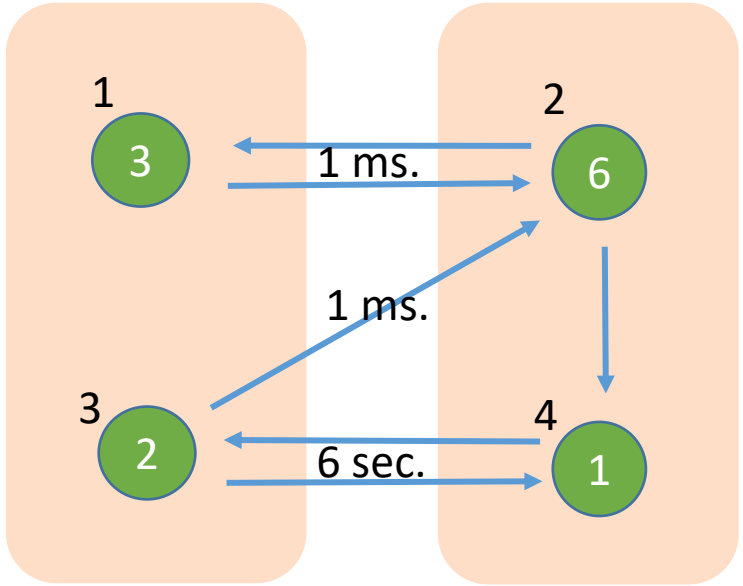
UNIVERSITY OF CAMBRIDGE

How strong is the problem solving?

- PageRank
- Shortest Paths
- Bipartite Matching (Mariage Problem)
- Semi-Clustering

Limitations & Discussion & Criticism

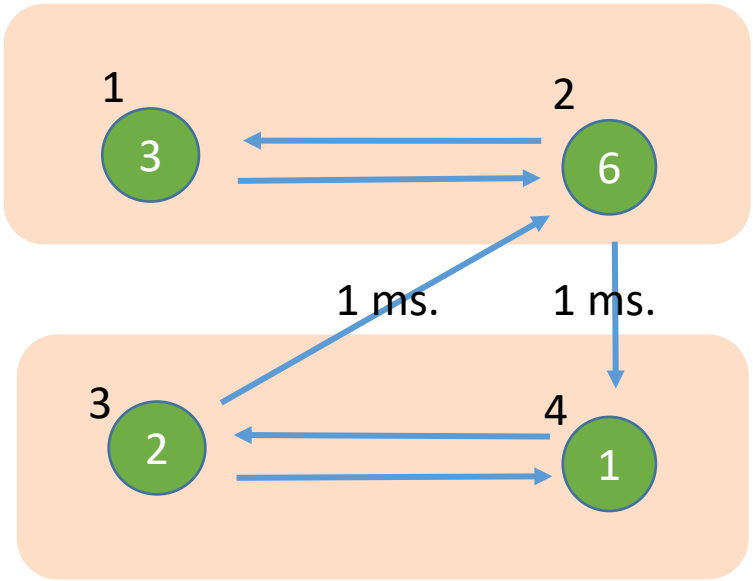
- Partitioning graphs is a **hard problem**




```
Step runtime: 6 sec.
```

Limitations & Discussion & Criticism

- Partitioning graphs is a **hard problem**
  - And maybe modulo N of ID is not the best solution
  - Maybe we should take in consideration the **structure of the graph**

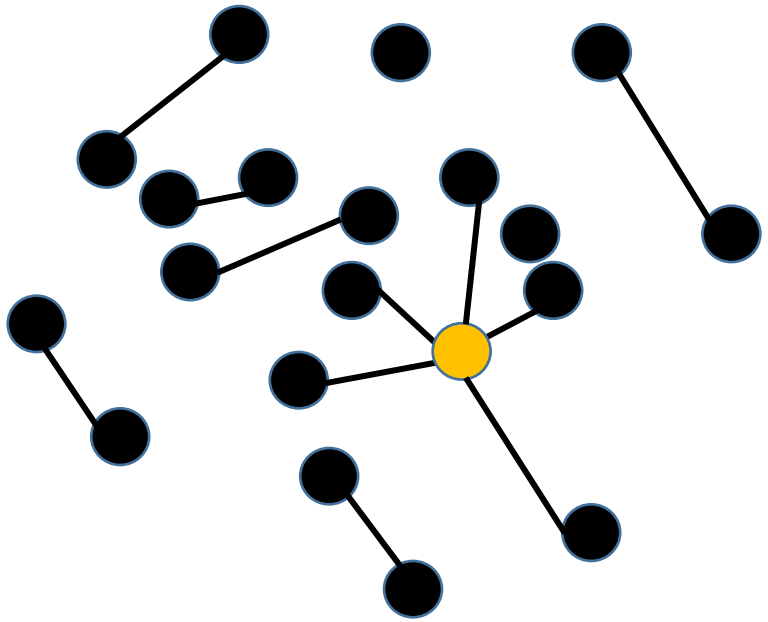


Step runtime: 1 ms.

UNIVERSITY OF CAMBRIDGE

Limitations & Discussion & Criticism

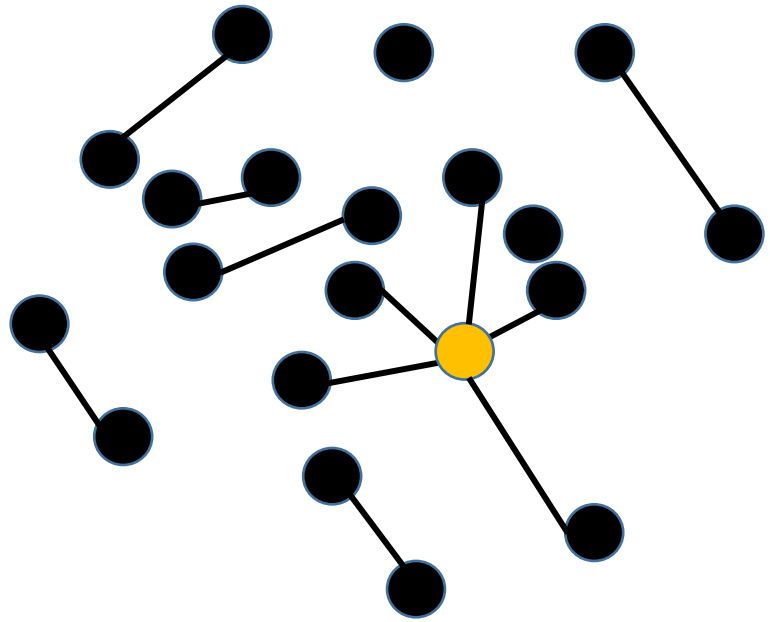
- Is bulk-synchronous computation a **good solution**?
  - "Natural Graphs" have generally weird structure with some nodes being more popular than others.
  - Shouldn't "celebrity" nodes have more priority? Is there a heuristic for that?



References

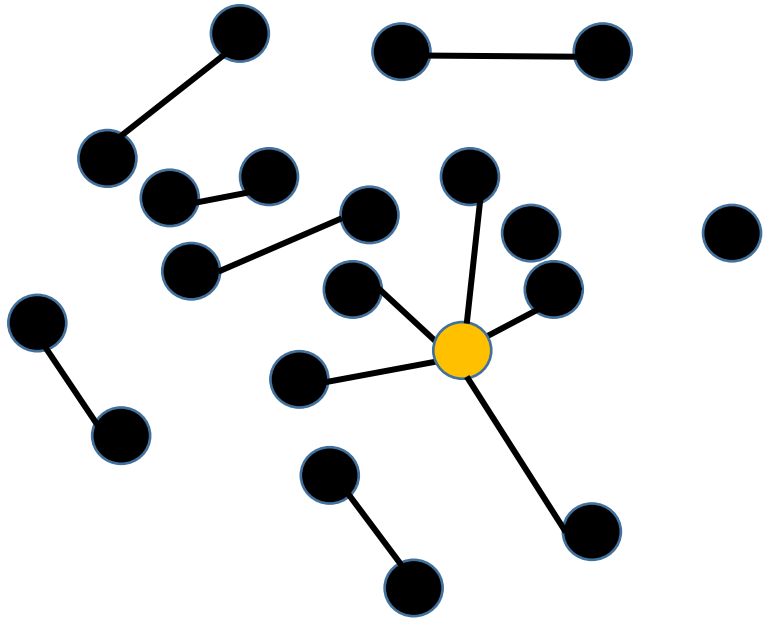
[0] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 2010.

[1] Leslie G. Valiant, A Bridging Model for Parallel Computation. Comm. ACM 33(8), 1990, 103–111.



UNIVERSITY OF CAMBRIDGE

Questions?



UNIVERSITY OF CAMBRIDGE

Slides for "Questions?" - Evaluation

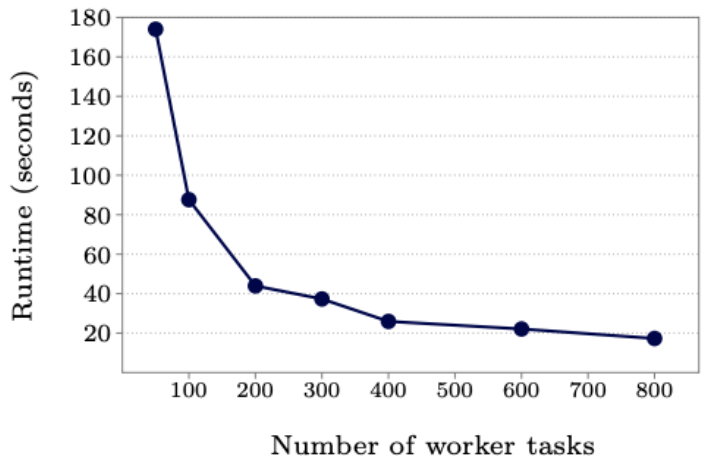


Figure 7: SSSP—1 billion vertex binary tree: varying number of worker tasks scheduled on 300 multicore machines

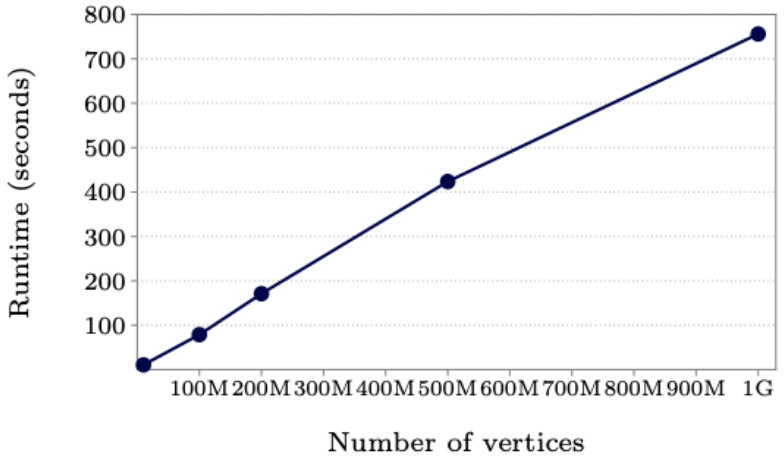
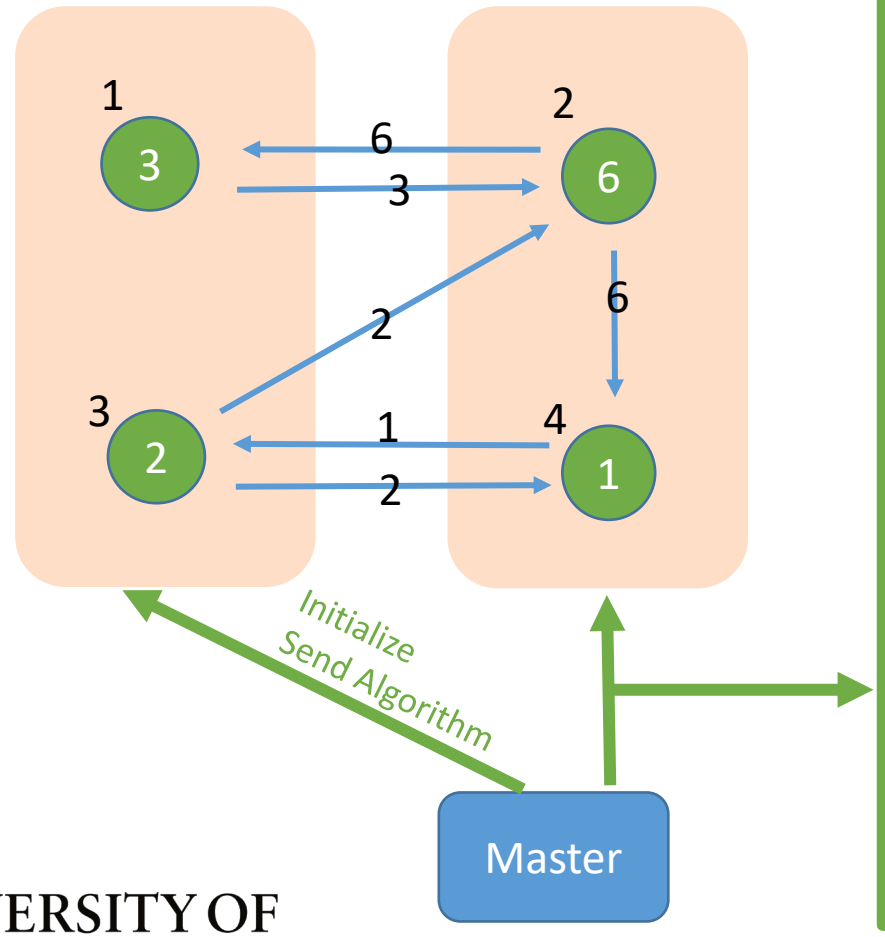


Figure 9: SSSP—log-normal random graphs, mean out-degree 127.1 (thus over 127 billion edges in the largest case): varying graph sizes on 800 worker tasks scheduled on 300 multicore machines

UNIVERSITY OF CAMBRIDGE

Slides for "Questions?" - Code Example




```
class PageRankVertex
    : public Vertex<double, void, double> {
 public:
  virtual void Compute(MessageIterator* msgs) {
    if (superstep() >= 1) {
      double sum = 0;
      for (; !msgs->Done(); msgs->Next())
        sum += msgs->Value();
      *MutableValue() =
          0.15 / NumVertices() + 0.85 * sum;
    }

    if (superstep() < 30) {
      const int64 n = GetOutEdgeIterator().size();
      SendMessageToAllNeighbors(GetValue() / n);
    } else {
      VoteToHalt();
    }
  }
};
```

Figure 4: PageRank implemented in Pregel.

UNIVERSITY OF CAMBRIDGE