Distributed Graph Processing

Immanuel Trummer
itrummer@cornell.edu
www.itrummer.org

Outlook: Beyond Relational Data

- **Graph data**
- Data streams
- Spatial data

Reading List

- "*Pregel: a system for large-scale graph processing*", SIGMOD 2010, G. Malewicz et al. [Google]
- "*One trillion edges: graph processing at Facebook-scale*", VLDB 2015, A. Ching et al. [Facebook]

Motivation: Large Graphs

- Graphs may **exceed** resource limits of single machines
  - Graphs representing the entire **Web** (Google)
  - Graphs representing large **social networks** (FB)
  - ...
- This motivates graph processing in **clusters**

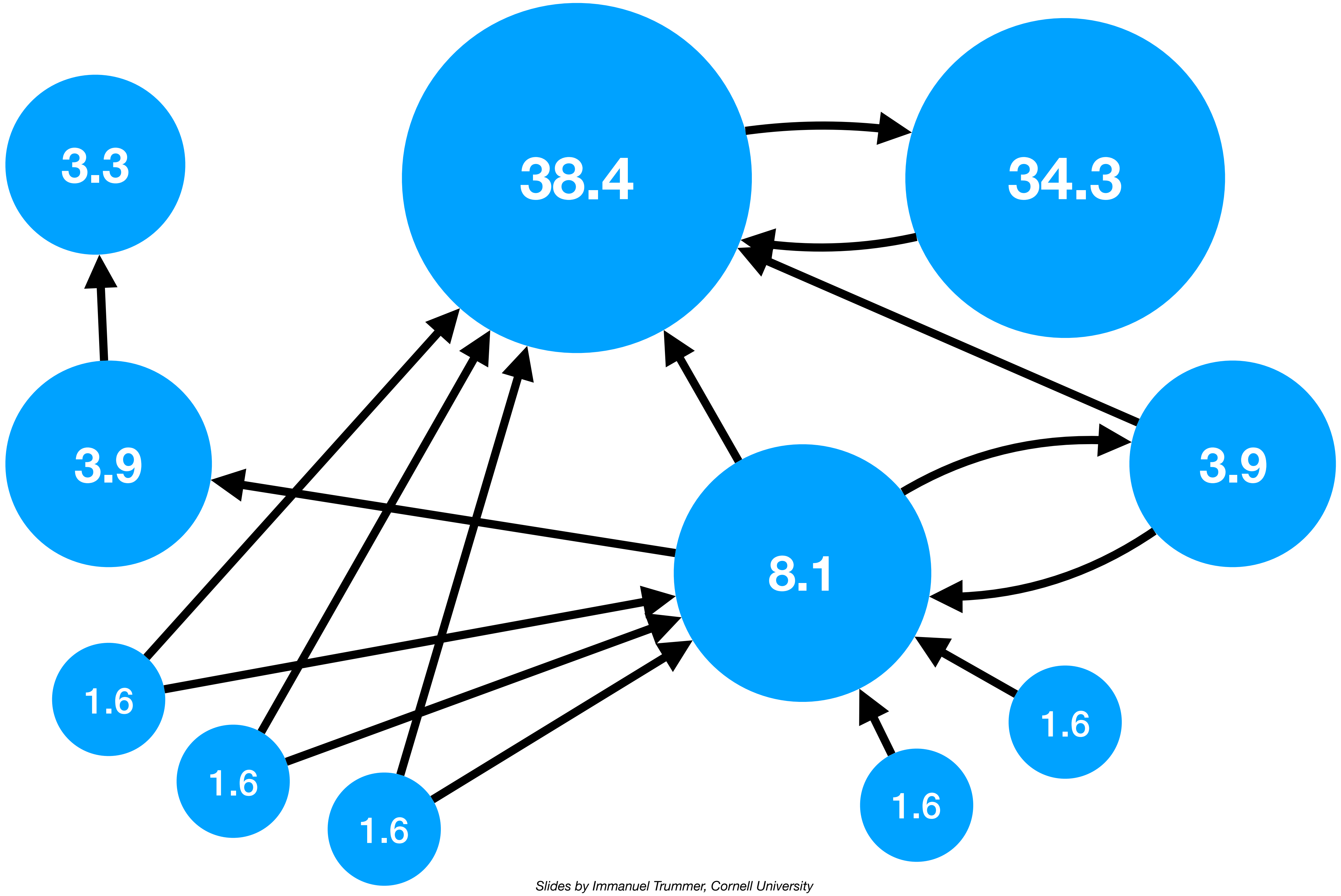
Example: PageRank

- Google ranks search results via the **PageRank** algorithm
- Operates on a graph representation of the **Web**
  - **Nodes** represent Web sites
  - **Edges** represent links
- Pages with higher PageRank are **preferable**

Random Surfer

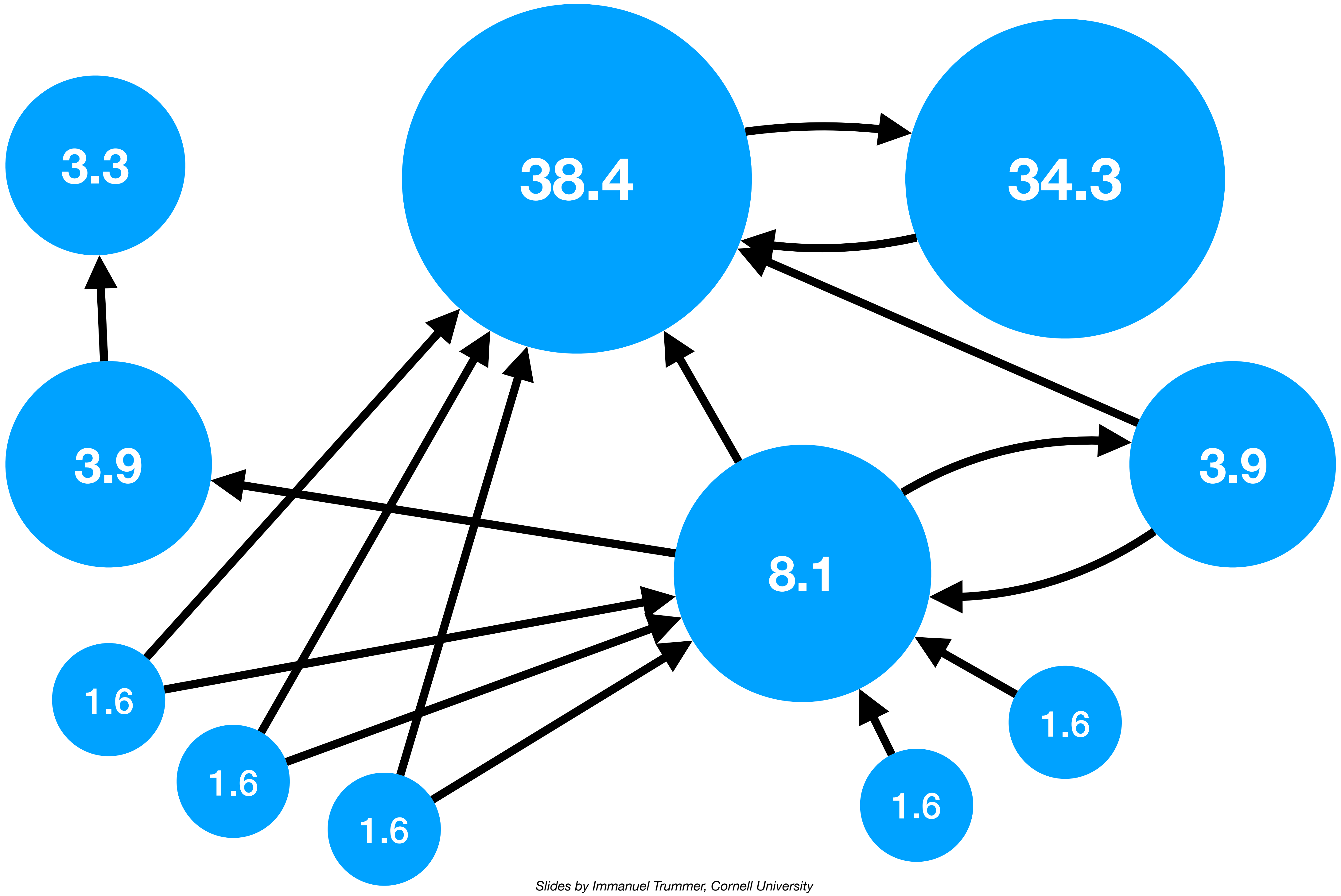
- PageRank is based on the **random surfer** model
- Random surfer **starts** from random Web site
- Randomly selects outgoing **links** to follow
  - May select **random** page with probability α
  - Selects random page if **no outgoing** links
- PageRank: **probability** to visit site at specific instant

PageRank Example



PageRank Example

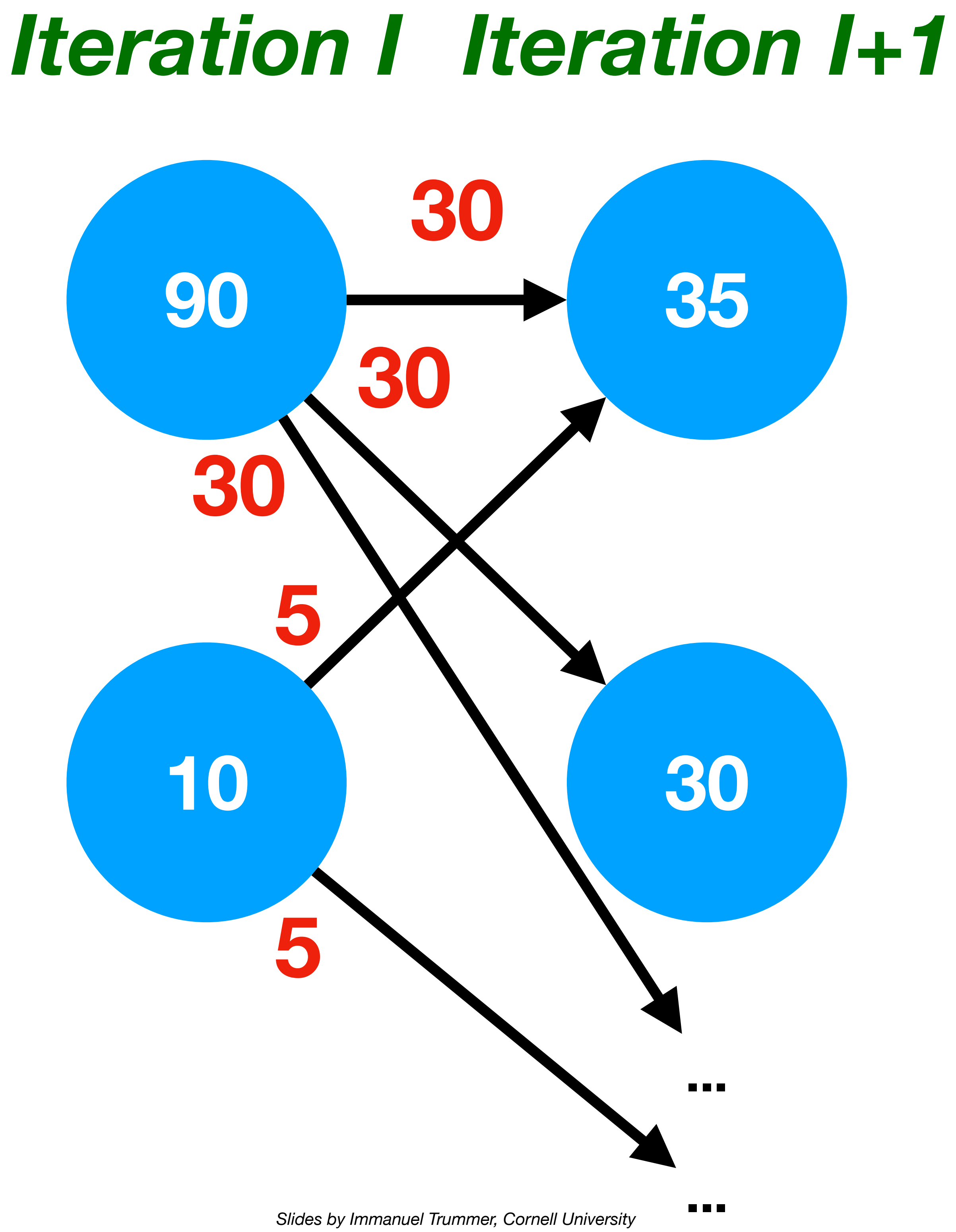
Do We Sometimes Select Random Pages?



Calculating PageRank

- We can calculate PageRank via an **iterative** algorithm
- We initialize each node's PageRank to **1/NrNodes**
- In each iteration, we **redistribute** PageRank over links
  - Each node partitions PageRank among outgoing links
  - PageRank in next iteration is sum over incoming links

PageRank Iterative Updates



Pregel Overview

- **Pregel** is a system for distributed graph processing
- Proposed in 2010 (Google), **PageRank** is use case
- Pregel **distributes** graph partitions over cluster nodes
- Worker nodes process their partition in **parallel**

Pregel Computation Model

- Computation is divided into **iterations** ("supersteps")
- In each iteration, we invoke **Compute** for each node
  - Compute function can be **customized** by user
  - **Input**: messages sent to this vertex in prior iteration
  - Can **message** other nodes, delivered in next iteration
- Computation ends once all nodes vote to **halt**

Illustration of Computation

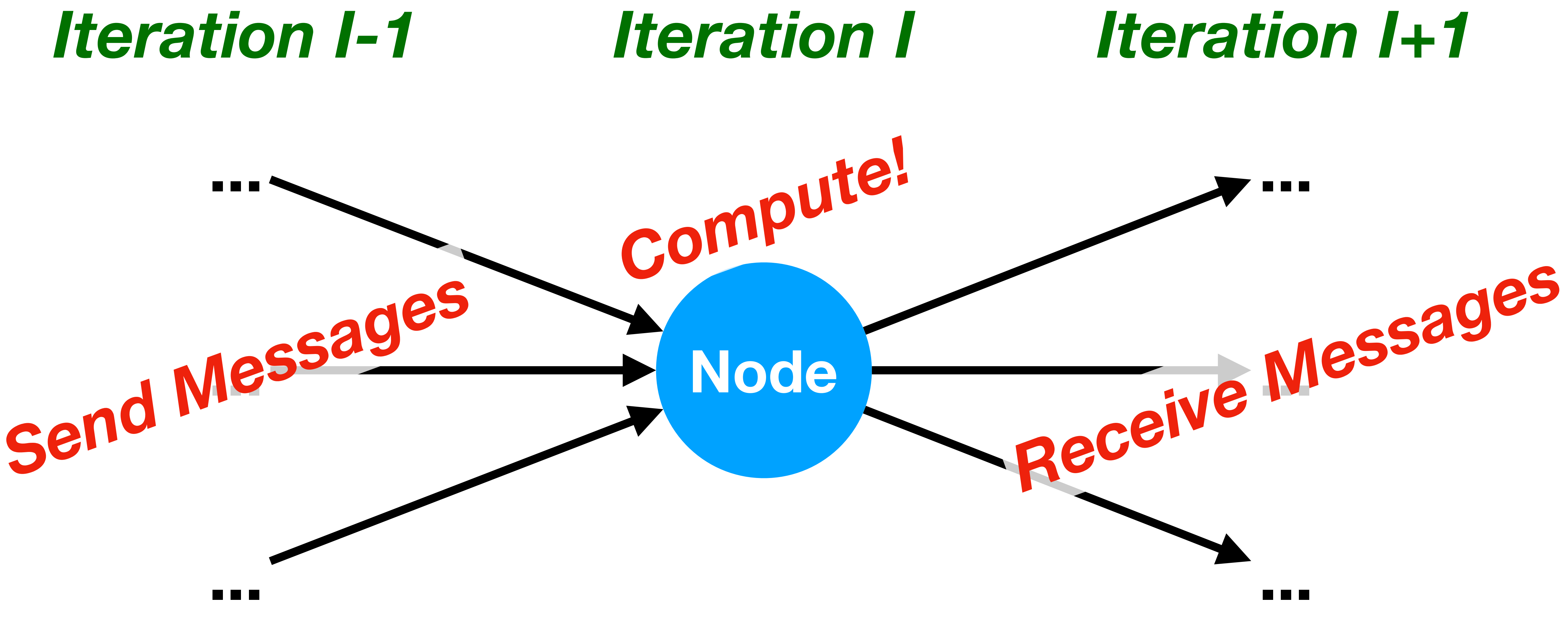
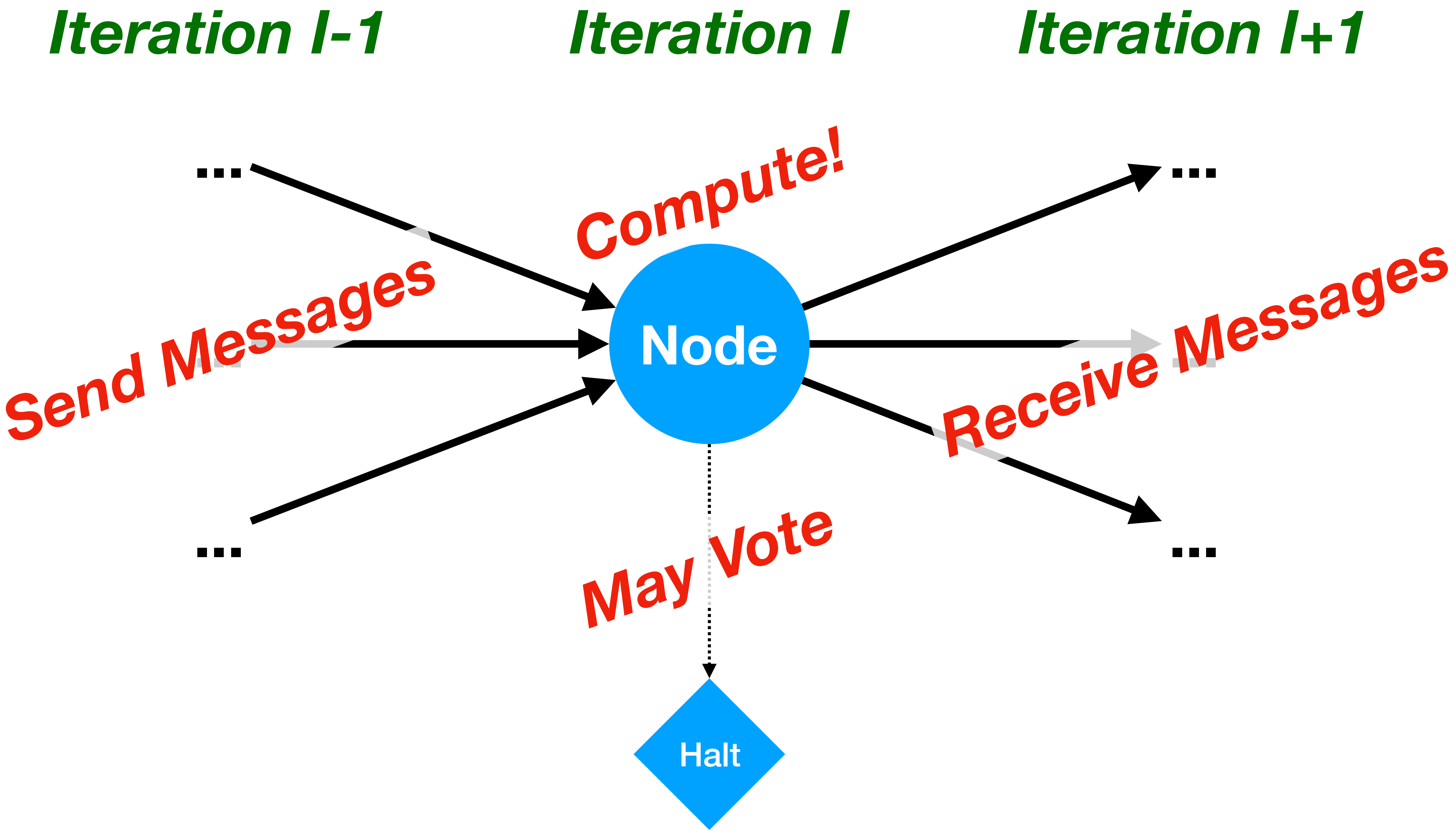
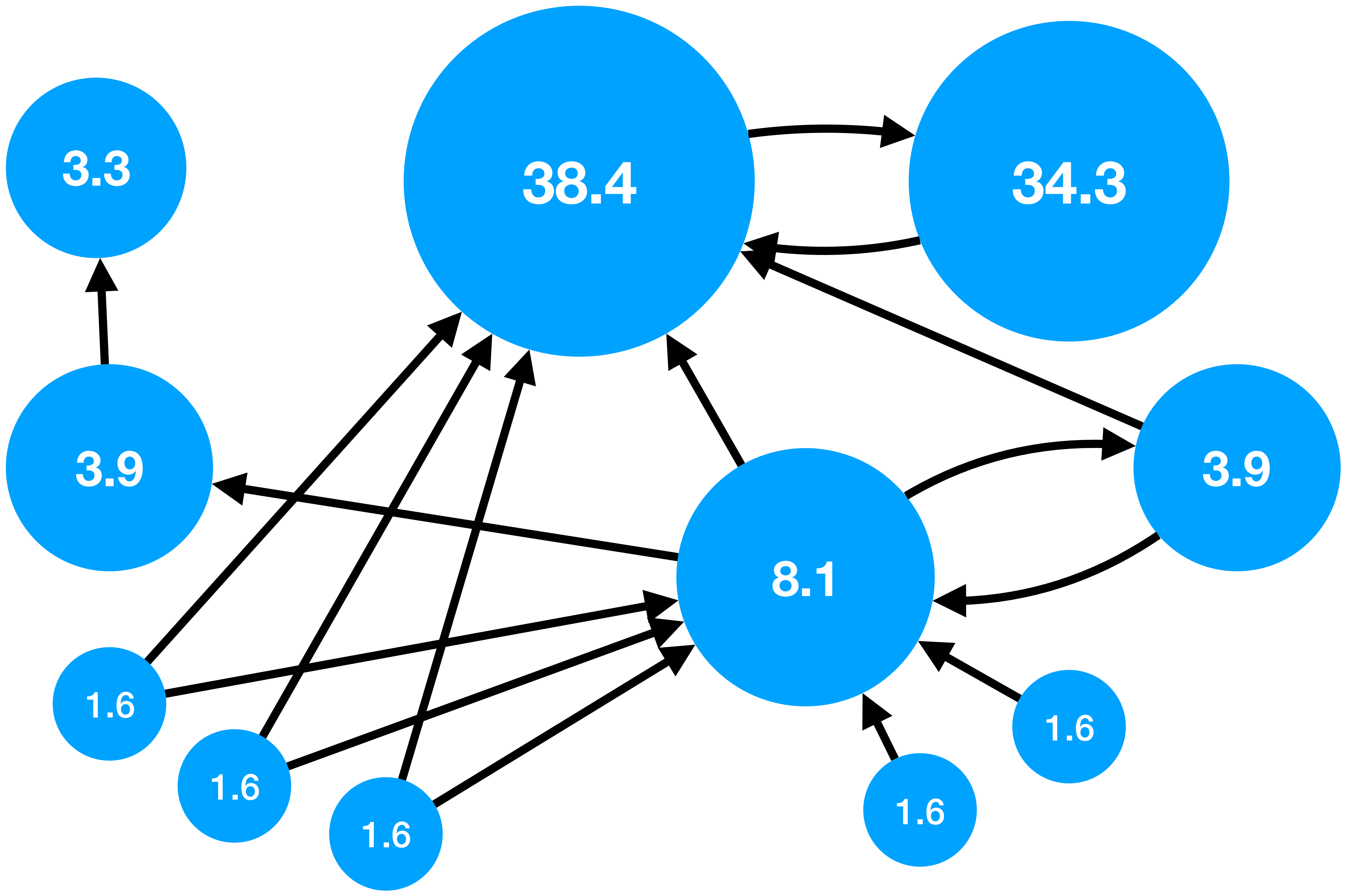


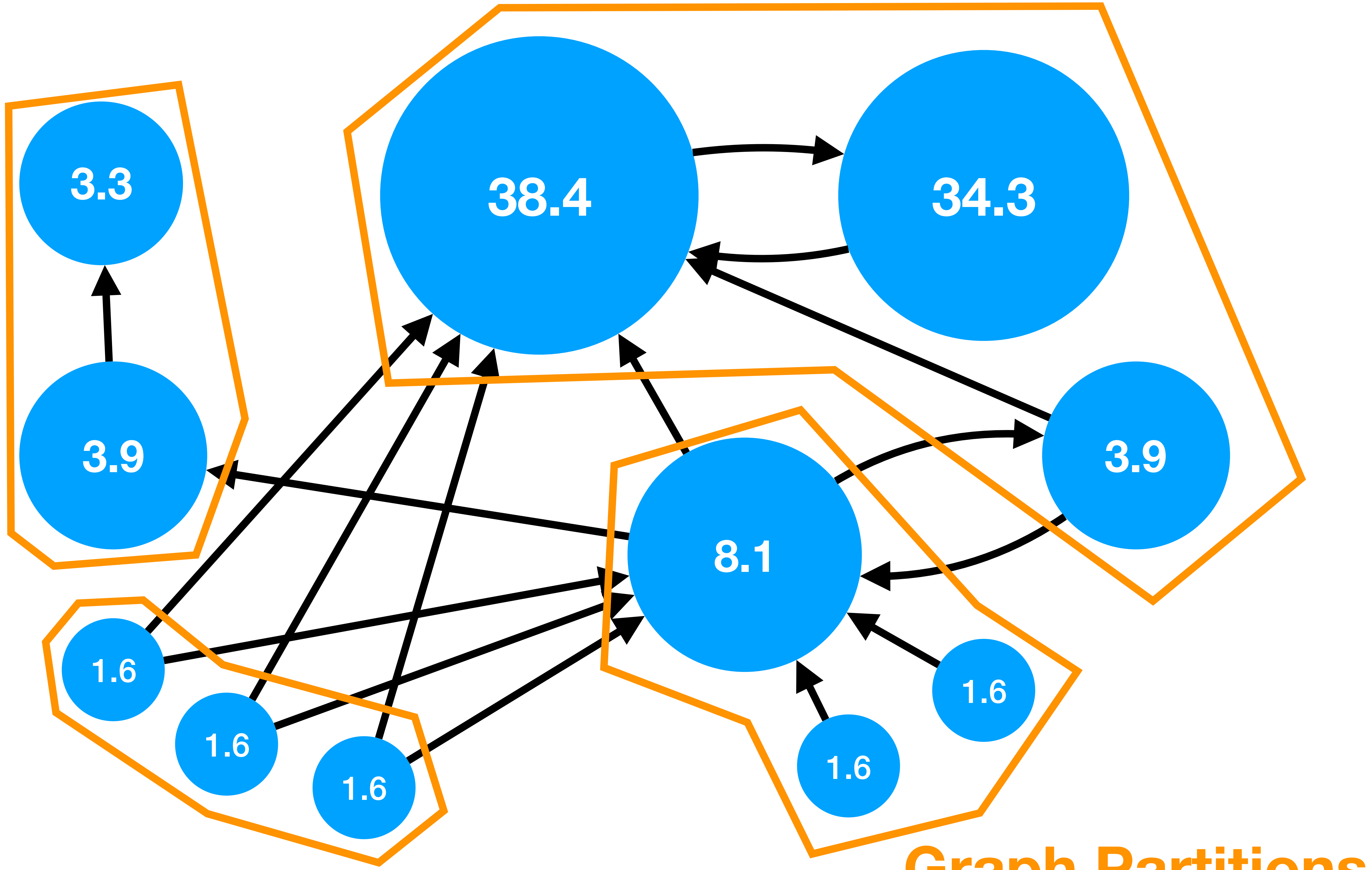
Illustration of Computation



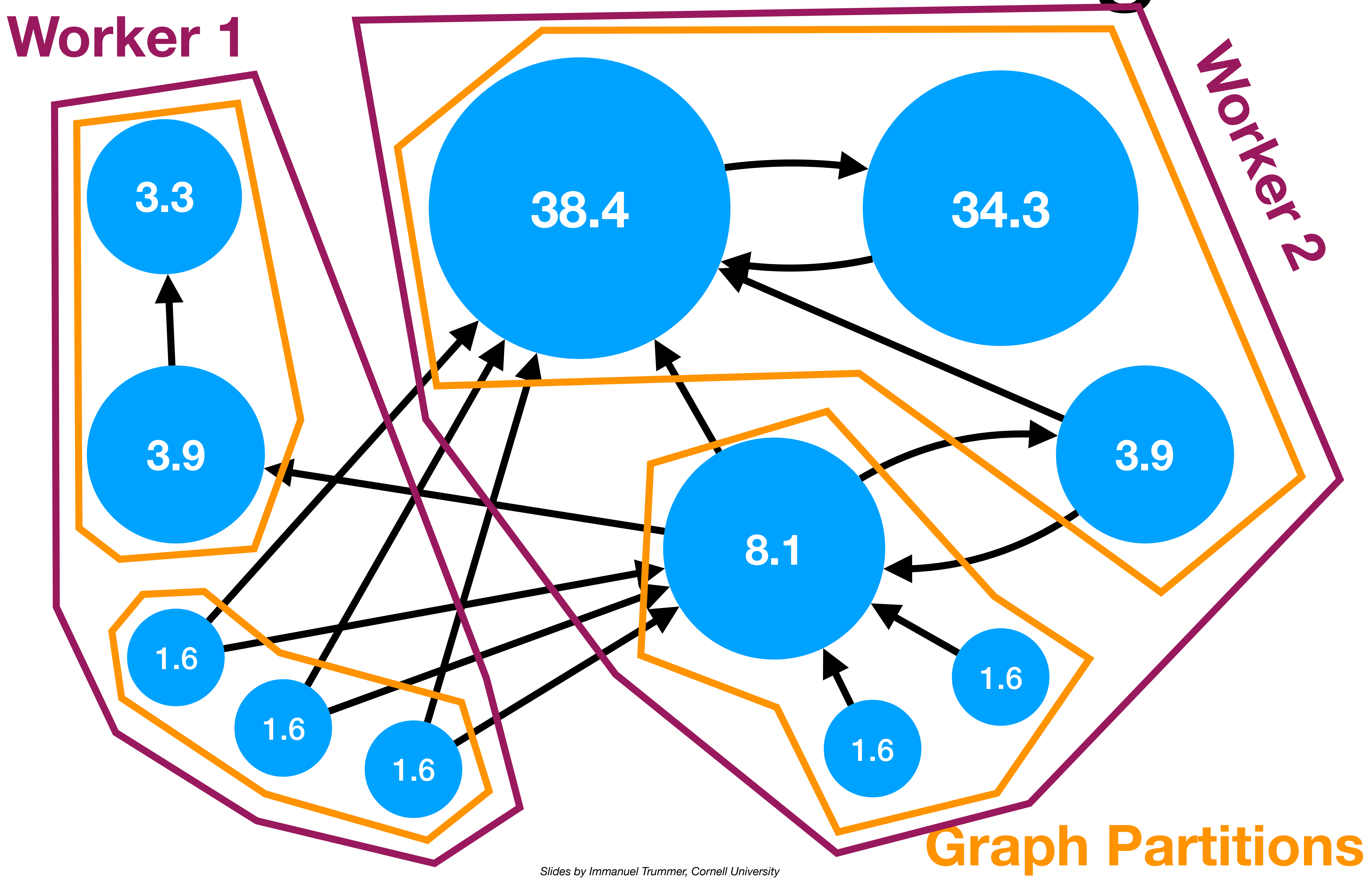
Parallel Processing



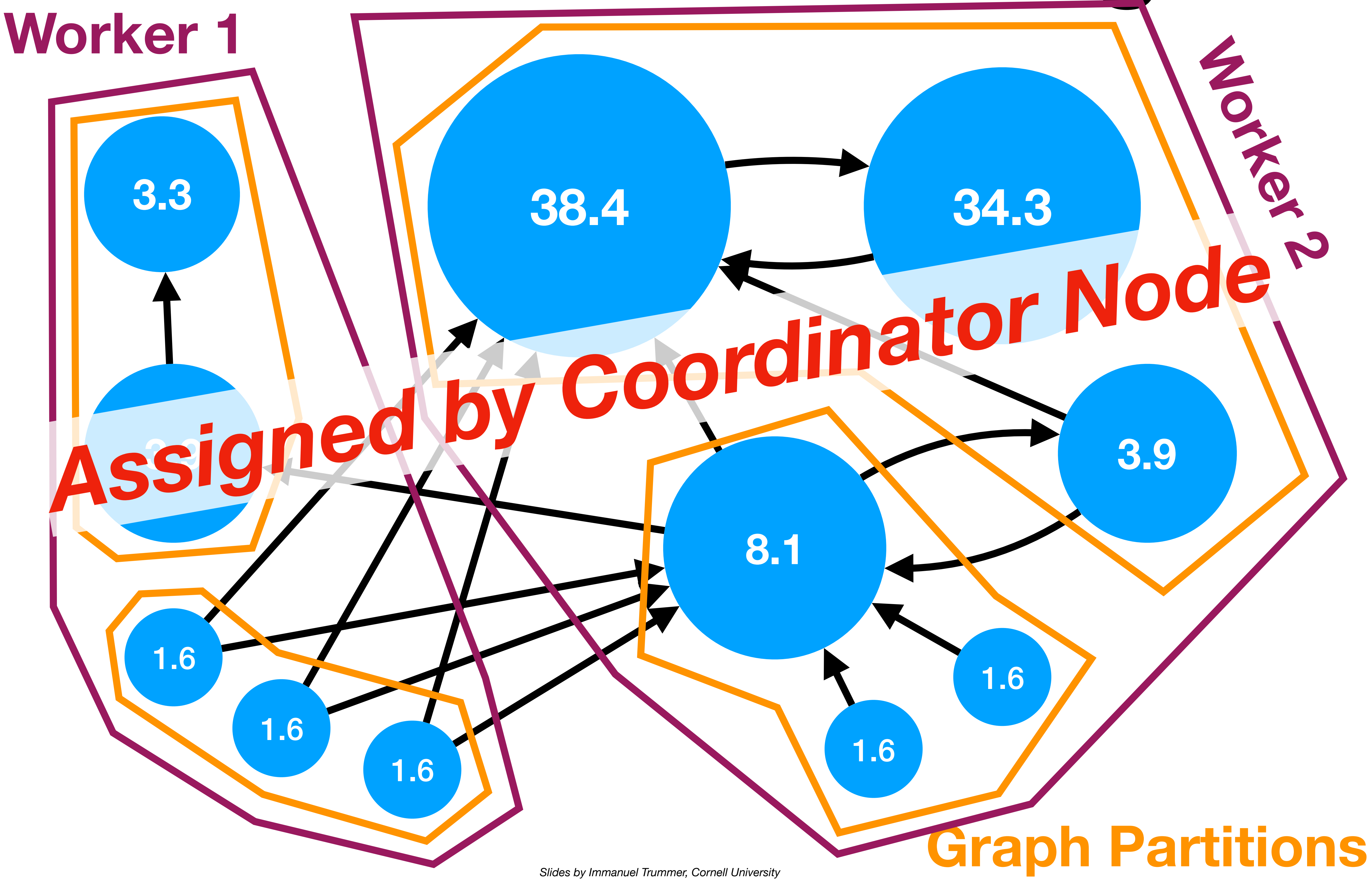
Parallel Processing



Parallel Processing



Parallel Processing



Fault Tolerance

- Workers **persist** input and state at iteration start
- Coordinator **detects** worker failures via pings
  - **Recovery** may start several supersteps earlier
  - **Re-partition** graph to replace failed workers
- "**Confined recovery**" restricted to failed partitions
  - Requires persisting outgoing messages as well

PageRank in Pregel

```
Compute(ReceivedPR : int[]):
    NewPR = sum(ReceivedPR)
    For o in OutgoingLinks:
        Send(o.target, NewPR/|OutgoingLinks|)
```

(Extensions required for random jumps and handling "dead ends")

Better Performance with Combiners

- Basic version **sends** lots of page rank values
- Can **aggregate** messages via custom "Combiners"
- Here: can combine page rank for same target as **sum**