Distributed Algorithms 2020

Aalto University

1a Introduction

Distributed problems

- **Initially:** each node only aware of *itself*

Distributed problems

- **Initially:** each node only aware of *itself*

- **Finally:** each node knows *its own part of the solution*

Distributed problems

- **Initially:** each node only aware of *itself*

- **Communication rounds:** each node exchanges messages with *its own neighbors*

Distributed problems

- **Initially:** each node only aware of *itself*

- **Communication rounds:** each node exchanges messages with *its own neighbors*

- **Finally:** each node knows *its own part of the solution*

Example: coloring

- Each node knows **its own color**

- Why enough?

- Coloring used for scheduling:

"I know my color is 5"
=
"I can safely be active during time slot number 5"

Locality

Fast distributed algorithm

=

Small number of communication rounds

=

Nodes only use information
in their own **local neighborhood**

Tell everyone everything

- **Before round 1:** each node only knows about *itself*

- **During round 1:** everyone tells about themselves to their neighbors

- **After round 1:** everyone now knows about their *neighbors* — they see everything up to **distance 1**

Tell everyone everything

- **Before round 2:** everyone knows about their *neighbors* — i.e. up to **distance 1**

- **During round 2:** everyone can send this information to their neighbors

- **After round 2:** everyone also aware of their *neighbors' neighbors* — they see everything up to **distance 2**

Tell everyone everything

- **Before round 3:** everyone knows everything up to **distance 2**

- **During round 3:** everyone can send this information to their neighbors

- **After round 3:** everyone now knows everything up to **distance 3**

Tell everyone everything

- **Before round *T*:** everyone knows everything up to **distance *T* – 1**

- **During round *T*:** everyone can send this information to their neighbors

- **After round *T*:** everyone now knows everything up to **distance *T***

Locality

- Which graph problems are **local**?
  - can be solved so that each node only looks at its own local neighborhood

- Which graph problems are **global**?
  - cannot be solved using only local information
  - we need to see the whole input graph

Two perspectives

- **Mathematics:**
  - graph theory, locality, distances…

- **Communication networks:**
  - computers, network connections, message-passing, algorithms…

Cost of communication

- **Communication:**
  get *one* bit from another computer
  in the same local network ≈ *0.5 milliseconds*

- **Computation:**
  one *billion* arithmetic operations ≈ *0.5 milliseconds*

Understanding nature

- What are the **fundamental limitations** of all kinds of systems that consist of interacting entities?
  - computer networks
  - biological systems
  - social networks
  - job markets
  - animal populations …

Negative results

- How to prove that some problems cannot be solved efficiently in distributed settings?

- How to prove that some problems are inherently global?