Distributed algorithms

Pierre Pradic Oguz Kaya

07/12/2016

We work only with asynchronous message passing. Message complexity is the number of messages exchanged in the network and time complexity will be defined as the maximal length maximal causal chains in computations.

Unless specified otherwise, processes do not have a unique ID and do not have access to any topological information save for the list of their neighbours and whatever global constraint we may impose on the network.

1 Wave algorithms

Let us recall that a *wave* algorithm is a distributed algorithm, together with a distinguished set of internal events called *decide* events, satisfying the following desiderata:

Termination Each execution is finite.

$$\forall C. |C| < \infty$$

Decision Each computation contains at least one decide event.

$$\forall C. \exists d \in C. d \text{ is a decide event}$$

Dependence For every decide event $d \in C$, d is causally preceded by an event in C_p for *every* p.

$$\forall C.\ \forall d \in C.\ d \text{ is a decide event} \Rightarrow \forall q.\ \text{processor}. \exists f \in C_q.\ f \preceq d$$

Question 1

a) Suppose that each process holds a value v and that we have an algorithm computing the minimum of those values (for a nontrivial order) in at least one process. What set of hypothesis can you make to consider such an algorithm a wave algorithm?

b) Conversely, show how to reduce the computation of an infimum to an arbitrary wave algorithm.

c) Can you instrument this to compute the reduction of an arbitrary associative, commutative and idempotent operator $(a, b) \mapsto a \cdot b$?

Let us recall that the *phase algorithm* is a decentralized wave algorithm working over directed graphs. All processes are supposed to know the diameter D of the network before proceeding as described in Algorithm **??**.

Question 2

a) Show that the phase algorithm is indeed a wave algorithm. What is its message complexity?

b) Does D need to be exactly the diameter of the graph?

c) *(Hard)* Assume that processes are given unique identifiers. Modify the phase algorithm so that it computes an interesting over-approximation of D.

2 Traversal algorithms

Recall that a *traversal* algorithm is a wave algorithm satisfying the following additional desiderata:

- There should be a unique initiator i and a unique decision event d with $d \in C_i$ for every computation C.
- Upon receiving a single message, a process should either send a unique message or decide.

Algorithm 1: Phase algorithm

```
const int: D (* network diameter *)
var set  : Out (* set of successor vertices *)
var set  : In (* set of predecessor vertices *)
var int  : Rec[] ⟵ 0 (* number of messages recieved from q ∈ In_p *)
var int  : Sent ⟵ 0 (* number of messages sent to each neighbour *)
if I am an initiator then
    for r ∈ Out do
        send to r
    increment Sent
while min_{q∈In} Rec[q] < D do
    receive from q_0
    increment Rec[q_0]
    if min_q Rec[q] ≥ Sent and Sent < D then
        for r ∈ Out do
            send to r
        increment Sent
decide
```

Question 3

a) Notice that we can conflate time complexity with message complexity for traversal algorithms.

We call an algorithm an f-traversal algorithm if it is a traversal algorithm such that after $f(x)$ rounds of sending/receiving, the message has gone through at least $\min(x+1, N)$ processors.

Question 4

a) Recall for which f the obvious traversal algorithm for the ring topology is an f-traversal algorithm.

b) Devise a $2x$-traversal algorithm for complete bipartite graphs (Hint: here, you are allowed to make messages carry an additional bit of information)

Recall that, in an execution of Tarry's algorithm, satisfy the following constraints:

- p does not forward the token twice to the same processor if others are available
- otherwise, p forwards the token to its first corresponding process if it is not the initiator; otherwise, it decides

Question 5

a) Write down Tarry's algorithm.

b) Carefully prove that Tarry's algorithm is indeed a traversal algorithm. What is its complexity?

c) Show how to instrumentalize this algorithm to compute a spanning tree.

3 Computing sums

Notice that our instrumentalization of wave algorithms in the first section enabled us to do reduction only when the operator we used was idempotent.

Question 6

a) Suppose that A is a wave algorithm. Additionally, assume that for every computation, active channels give rise to a spanning tree over the network of processors. Assuming that each processor holds an integer v_p, explain how to modify A to compute $\sum_p v_p$[1].

b) Can you use Tarry's algorithm to compute a sum using this reduction?

c) Do you know any other suitable wave algorithm for this purpose?

[1] Notice that this works for any commutative semigroup operation.

Question 7

a) Suppose that you are given a wave algorithm and that each processor possesses a unique identifier. Argue that you can instrumentalize it to compute a sum as in the previous question.

b) Notice that the messages can get quite bigger.

Question 8

a) Assuming the following statement, contemplate why one cannot apply a similar trick in the most general case.

> There exists no anonymous distributed and decentralized distributed algorithm computing the size of the network, even if it is known that $D \leq 2$.

b) *(I don't know how to do it)* Prove the lemma. (Hint: you can restrict the class of networks whose every vertex has degree exactly 3.)