Wave and Traversal Algorithms

- Introduction
- Results about Wave algorithms
- Algorithms
- Traversal Algorithms
- Complexity results
- Overview

Introduction to Wave algorithms

- Many things in distributed systems can be achieved using Message Passing. Message Passing systems are called Wave algorithms.
- Some examples are broadcasting, synchronization, triggers, distributed computing of a function.
- Used in many distributed tasks as subtasks.
- A fixed, undirected and connected network topology is assumed.
- Define a causal precedence relation $\preccurlyeq$

Definition of Wave Algorithms

- Wave algorithm is a distributed algorithm that satisfies the following three conditions
- (1) *Termination*. Each computation is finite, i.e. $\forall C : |C| < \infty$.
- (2) *Decision*. Each computation contains at least one decide event, i.e. $\forall C : \exists e \in C : e$ is a decide event.
- (3) *Dependence*. In each computation each decide event is causally preceded by an event in each process, i.e. $\forall C : \forall e \in C : ($ e is a decide event $\Rightarrow \forall q \in \mathbb{P} \exists f \in C_q : f \preccurlyeq e)$.
- A computation of a wave algorithm is called a wave.
- Initiators and non-initiators.

Differences between Wave algorithms

- Centralization
- Topology
- Initial knowledge
  - Process identity
  - Neighbors' identities
  - Sense of Direction
  - Number of Decisions
  - Complexity

Elementary Results about Wave Algorithms

- Each event in a computation is preceded by an event in an initiator.
- A wave with one initiator defines a spanning tree of the network when for each non-initiator the channel is selected through which the first message is received.
- Each decide event is causally preceded by a send event in all other processes except for the one it takes place in.
- Let C be a wave with one initiator p, such that a decide event occurs in p. Then at least N messages are exchanged in C.
- Let A be a wave algorithm for arbitrary networks without initial knowledge of the neighbors identities. Then A exchanges at least $|E|$ messages in each computation.

Propagation of Information with Feedback

- Some processes hold information that needs to be propagated to all processes. Also some processes must receive a notification when the broadcast is complete.
- Theorem: Every PIF algorithm is a wave algorithm
- Theorem: Every wave algorithm can be employed as a PIF algorithm

Propagation of Information with Feedback (continued)

- Let P be a PIF algorithm.
- All computations in P must be finite.
- In each computation a decide event must occur.
- If a decide event happens in process p that is not preceded by any event in process q, then there is an execution where q has not received any messages which contradicts the requirements.

Synchronization

- Synchronization problem is such that there are a number of processes. Each process p has to execute an event a_p and some processes must execute event b_p. All events a_p must be executed before any of the events b_p is executed.
- Theorem: Every synchronization algorithm is a wave algorithm.
- Theorem: Every wave algorithm can be employed as a synchronization algorithm.

Computation of Infimum functions

- There is a class of algorithms that depend on the input of every process. An example of such algorithms is a computation of infimum over all inputs, which must be drawn from partially ordered set.
- Let $(X, \leq)$ be a partial order. c is called infimum of a and b if $c \leq a, c \leq b$ and $\forall d: \ (d \leq a \wedge d \leq b \Rightarrow d \leq c)$. This infimum is denoted by $a \curlywedge b$.
- $\curlywedge$ is commutative ($a \curlywedge b = b \curlywedge a$), and associative ($a \curlywedge (b \curlywedge c) = (a \curlywedge b) \curlywedge c$).
- Theorem: Every INF algorithm is a wave algorithm.
- Theorem: Every wave algorithm can be used to compute an infimum.

Computation of Infimum functions (continued)

- Infimum Theorem: If $\star$ is a binary operator on a set X such that
  - $\star$ is commutative, i.e. $a \star b = b \star a$.
  - $\star$ is associative, i.e. $(a \star b) \star c = a \star (b \star c)$
  - $\star$ is idempotent, i.e. $a \star a = a$.

  then there is a partial order $\leq$ on X such that $\star$ is the infimum function.
- Some examples of infimum functions are $\wedge$, $\vee$, min, max, gcd, lcm, $\cap$, and $\cup$.

Algorithms

- The Ring Algorithm
- The Tree Algorithm
- The Echo Algorithm
- The Polling Algorithm
- The Phase Algorithm
- Finn's Algorithm

Ring Algorithm

- Centralized algorithm. Initiator sends a message $< tok >$ called token.
- Each process waits until it receives the token and then passes it on. When initiator receives the token, it decides.
- The Ring Algorithm is a wave algorithm

The Tree Algorithm

- A wave algorithm in which leaves are initiators.
- Works in a tree network.
- If a process had received messages through all their incident channels except one, it sends a message through the last one.
- If a process has received a message through all its incident channels, it decides.
- The Tree algorithm is a wave algorithm

The Echo Algorithm

- The echo algorithm floods messages from the initiator to all the nodes and back.
- Works in arbitrary networks and forms a spanning tree starting from the initiator.
- The initiator sends a message $< tok >$ to all its neighbors. When a non-initiator receives $< tok >$ for the first time it sends a ¡tok¿ to all its neighbors.
- When a non-initiator has received $< tok >$ from all its neighbors, it sends a $< tok >$ to its father, i.e. the node that first sent $< tok >$ to it. When initiator has received a $< tok >$ from all its neighbors, it decides.
- Theorem: The Echo Algorithm is a wave algorithm.

The Polling Algorithm

- The polling algorithm works in clique networks.
- The initiator sends a query to all nodes and decides after it has received receipt from all nodes.
- Theorem: The Polling Algorithm is a wave algorithm.

Traversal Algorithms

- Traversal Algorithms are an important sub class of wave algorithms. They are defined by the following properties:
  - In each computation there is one initiator, which starts the algorithm by sending out exactly one message.
  - A process which receives a message, either sends out a message or decides.
  - The algorithm terminates in the initiator and when this happens each process has sent a message at least once.
- This means that there is always exactly one message in transit, or exactly one process has just received a message and has not yet sent a message.

Traversal Algorithms (continued)

- An algorithm is an f-traversal algorithm (for a class of networks), if
  - it is a traversal algorithm
  - in each computation at least $min(N, x + 1)$ processes have been visited after $f(x)$ token passes.
- For example ring algorithm is an x-traversal algorithm, because x+1 processes have been visited after x passes.

Traversal Algorithms

- Cliques
- Tori
- Hypercubes
- Connected Networks (Tarry's algorithm)

Cliques

- A clique can be traversed by sequential polling. One neighbor at a time is polled and when a reply is received, another is polled.
- Theorem: Clique is 2x-traversal algorithm.

Tori

- $n \times n$ torus graph is the graph $G = (V, E)$ where
  $V = \mathbb{Z}_n \times \mathbb{Z}_n = \{(i, j) : \ 0 \le i.\ j < n\}$
  and
  $E = \{(i, j)(i', j') : (i = i' \wedge j = j' \pm 1) \ \vee \ (i = i' \pm \wedge j = j')\}$
  with addition and substraction modulo n.
- The Torus is a Hamiltonian graph and the token is sent along a Hamiltonian cycle. That is achieved by sending the token up every kth step, i.e. when $n|k$. Otherwise the token is sent to the right.
- Theorem: The torus algorithm is an x-traversal algorithm for torus.

Traversing Connected Networks

- Tarry's algorithm
  - R1. A Process never forwards the token twice through the same channel.
  - R2. A non-initiator forwards the token to its *father* only if there is no other channel possible according to previous rule.
- Theorem: Tarry's algorithm is a traversal algorithm
- Each computation of Tarry's algorithm defines a spanning tree of the network.

Time Complexity of wave algorithms

- In asynchronous systems the time between sending of a message and its receipt may vary a lot. Thus a instruction count will be used for measuring Time Complexity.
- Definition: The time complexity of a distributed algorithm is the maximum time taken by a computation of the algorithm under the following assumptions.
  - A process can execute any finite number of events in zero time.
  - The time between sending and receipt of a message is at most one time unit.
- Lemma: For Traversal algorithms the time complexity equals the message complexity.

Distributed Depth-first search

- Restrict Tarry's algorithm to gain classical depth-first search.
- When a process receives the token it sends it back through the same channel, if this is allowed by rules R1 and R2.
- Theorem: Classical depth-first search algorithm computes a depth-first search spanning tree using $2|E|$ messages and $2|E|$ time units.

Distributed Depth-first Search using Linear Time

- The time complexity of depth-first search can be reduced by traversing edges in parallel, rather than serially.
- Awerbuch's solution
- When process p is first visited by a token $< tok >$, it sends a $< vis >$ message to all it's neighbors except its father. These respond to it by sending $< ack >$. Only when it has received $< ack >$ from all its neighbors will p forward the $< tok >$. When the $< tok >$ later arrives at r, it will not forward $< tok >$ to p unless p is its father.
- Theorem: Awerbuch's algorithm computes a depth-first search tree in $4N - 2$ time units and $4|E|$ messages.

Cidon Algorithm

- Cidon's algorithm improves from Awerbuch's algorithm by not waiting for $< ack >$ messages.
- When process p passes the token, it marks down the process it passed the token to. If p receives the token from some other neighbor process, it ignores the token and marks the edge used. If a process r receives an $< ack >$ from a process it has sent token to, it will resend the token to another node.
- Theorem: Cidon's algorithm computes a depth-first search tree in $2N - 2$ time units using $4|E|$ messages.

Overview

- Wave and Traversal algorithms can solve a wide range of fundamental problems in distributed algorithms. These can often be found as subproblems of larger problems.
- These problems include synchronization between processes, broadcasting information, forming a spanning tree of the network and many others.
- Traversal algorithms are totally ordered by causality.

The Phase Algorithm

- Decentralised algorithm for arbitrary directed networks.
  In-neighbors of process p are those that can send to process p and out-neighbors of process p are those that it can send to.
- Some upper limit to networks diameter D must be known.
- Each process sends exactly D messages to each out-neighbor.
  Only after receiving at least i messages from each in-neighbor, will the *(i+1)*th message be sent.
- Theorem: The Phase Algorithm is a wave algorithm

Finn's Algorithm

- Finn's Algorithm is a phase algorithm that can be used for arbitrary directed networks. It doesn't require knowledge of upper bound on diameter D, but does require availability of unique identities for the processes.