CompSci 131

Parallel and Distributed Systems

Prof. A. Veidenbaum

Today’ s topics

- Elections in large-scale systems
- Location Systems

- Reading assignment:
  - Today: 6.4-6.5
  - Next time: 6.7 7.1-7.2

  - Complete the assignment *before* next class
    -

Last Lecture Covered

- Election algorithms

Election in Large-Scale Systems

- So far looked at algorithms to elect a single leader

- Sometimes may need to elect *more than one* leader
  – Example – super-peer selection in P2P systems

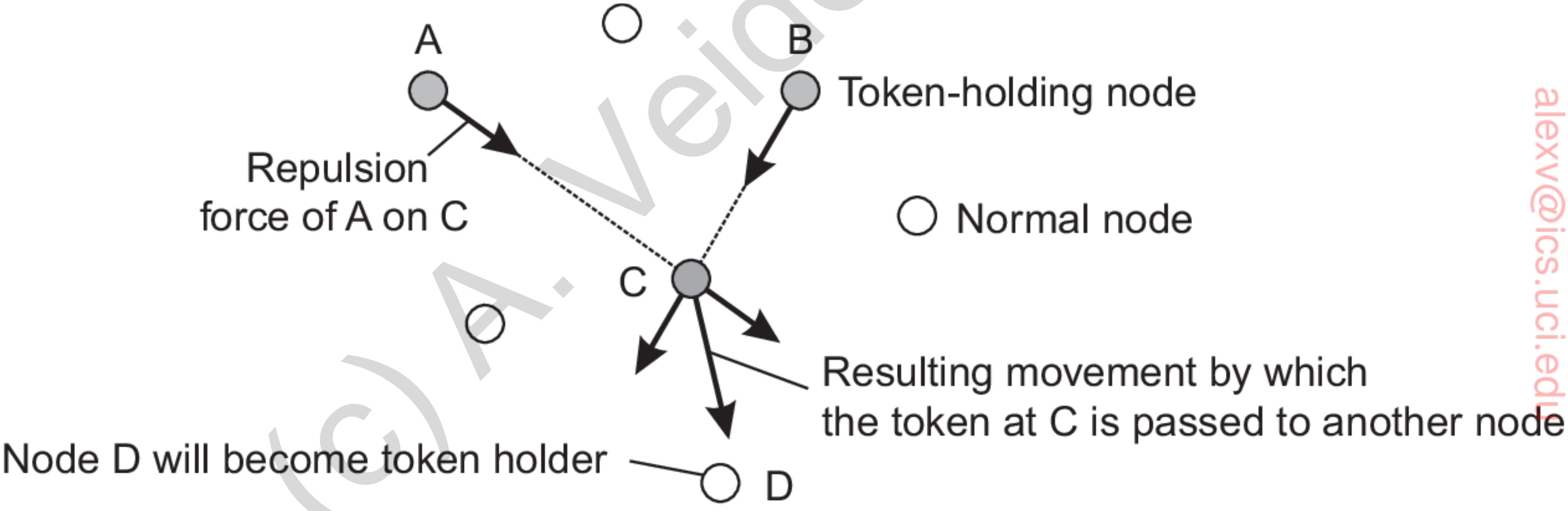
- Requirements for super-peer (S-P) selection
  1. Normal nodes should have low-latency access to S-Ps
  2. S-Ps should be evenly distributed across the network
  3. There should be a predefined portion of S-Ps relative to the total number of nodes in the overlay network.
  4. Each S-P should not need to serve more than a fixed number of normal nodes.

Election in Large-Scale Systems (2)

- **Approach: position N nodes in an m-dimensional space**
  - 2^m nodes in the system, N S-Ps
  - The space is geometric -
  -
- **Spread N tokens to N randomly selected nodes**
  - No more than one token per node
- **Each token represents a repelling force**
  - Other tokens are *likely* to move away from it
    - » S-Ps should be evenly distributed across the network
- **Nodes then need to know where the other tokens are**
  - How to do this?

Election in Large-Scale Systems (3)

- Use a gossip protocol to initially disseminate a token's force throughout the network
  - –
- Then look at forces acting on a node
  - Shaded nodes have a token


A
B
Token-holding node
Repulsion force of A on C
Normal node
C
Resulting movement by which the token at C is passed to another node
Node D will become token holder
D

Epidemic protocols for info dissemination

- Disseminate information as viral infections do
  -
- An epidemic protocol classifies nodes as
  - *Infected* if it has data and is willing to disseminate
  - *Susceptible* if it has not yet seen the data
  - *Removed* if it is no longer willing/able to spread data
- An epidemic protocol works as follows
  - A node picks another node and pulls update from it
    - Or they exchange updates
  - A round = period during which every node initiated an update once
- In O(log(N)) rounds can propagate an update to all

Gossip protocols for info dissemination

- A variant of an epidemic protocol
  - Disseminate information as rumors are spread
- An gossip protocol works as follows
  - A node picks another node and sends it an update
  - A node that received an update will start spreading it too
  - A node stops when it reaches a node that already got the update
- This is a “push” algorithm and it is not a great
  - Pull is much better
- The gossip algorithm is not guaranteed to update all nodes in a large system

Location Systems

- So far did not worry about location of nodes
  - Even super-peers were kinda randomly pushed around
- Sometimes need to know proximity
  - E.g. which update or ftp server to contact
- Use location based techniques to place processes
- One approach – use GPS
  - But as we have discussed, it may not work for every node
- What else can we do?

Location Systems (2)

- Use logical , proximity-based locations
- Most common approach is to use internode latency
  - Can be measured
    - Recall how clock synchronization algorithms compute latency
- Consider a replicated server Q with N replicas
  - Q can redirect the request to the closest replica
    - Or can have delay to all replicas available and go directly
- Another example is the problem of replica placement
  - Compute positions of N servers that minimize average client-replica time

Location Systems (3)

- Centralized positioning works by contacting nodes with known locations

–

- Again, use internode latency to estimate distance to these nodes with location info

- Problems:
  - Latency measurement accuracy and variability
  - A different position is obtained if known nodes change
    - One of them may have gone down

–

- Can use decentralized positioning
  - Assumes nodes are connected by springs and compute what happens

m

Location Systems (4)

- **In decentralized positioning a node P_i repeatedly execute the following steps**
  - **An apostrophe indicates a vector in coordinate space**
  - **Measure the latency d_{ij} to node P_j, and also receive P_j's coordinates x'_j.**
  - **Compute the error e = d1(Pi,Pj) – d2(Pi,Pj)**
    - » **A perturbation of Pi coordinate to current coordinates**
  - **Compute the direction u' = u(x'i – x'j)**
  - **Compute the force F = e * u'**
    - » **Force proportional to distance**
  - **Adjust own position by moving along the force vector**
    - » **$x'_i \leftarrow x'_i + \delta * u'$**
      - **Selecting δ is a key issue, sometimes use an adaptive value**