Using Swarm Intelligence in Linda Systems

Robert Tolksdorf
Freie Universität Berlin

Ronaldo Menezes
Florida Institute of Technology
(presenter)

Linda

- The success story
  - Simple
  - Used as basis for commercial implementations
    - T Spaces, JavaSpaces and GigaSpaces
- The problems
  - Lacks scalability
  - Not able to adapt to changes at the application level
    - Lack of adaptiveness

We've tried!!!

Scalability Approaches

- Centralization
  - Also called distributed central servers approach
  - Easy to implement
  - Carries all the disadvantages of centralized services
- Partitioning
  - Similar tuples are placed in a set of tuple spaces servers
  - May lead to unbalanced distribution of load unless a good hashing function is chosen
  - Includes a level of centralization for groups of tuples
  - Reconfigurations are difficult to handle

We've tried!!!

Scalability Approaches

- Full replication
  - Places copies of tuples spaces in several nodes at different locations
  - Addition and removal of tuples need to be replicated across all node
    - This is done so that searches can be performed locally
  - Very high cost of keeping replicas requiring locking protocols
- Intermediate replication
  - Grid of nodes formed by logical intersecting buses
    - Each node has one inbus and one outbus
  - Data is replicated on all nodes of the outbus
  - Searches are performed in the inbus
  - Number of concurrent access is proportional to the number of inbuses and outbuses
  - Still caries some of the cost of full replication

“Ultimate” Scalable Systems

- Natural forming multi-agent systems
  - Ant colonies, Termites mounds, etc.
  - Enormous in size
- Artificial Ants
  - Proved effective in practice
    - Optimization tool (meta-heuristic)
    - Network routing
- Why use it in Linda?
  - Small objects —» Tuples
  - Stigmergy —» Associative Communication
  - Mobility —» Tuples and Processes

A Bit About Swarms

- Stigmergy and Direct Communication are used
  - But always local
- Random choices of path

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

- Probability is then used to decide which path to choose
- An example of the pheromone update rule can be given by

$$\tau_{il}(t) \Leftarrow (1-\rho) \cdot \tau_{il}(t) + \Delta\tau_{il}(t)$$

Negative Feedback

Reinforcement

Concepts of SwarmLinda

- Ants
  - Templates or tuples
- Environment (terrain)
  - Linda system (or the set of tuple space nodes)
- State
  - Information about the current condition of the system
- We wanted to be true to swarm concepts in SwarmLinda
  - Simplicity
  - Dynamism
  - Locality

Searching for Tuples

- Tuples are food
- Templates are ants
- State is given by a tuple scent
- Tuple-space nodes are the terrain
  - maybe different for each tuple format
- The steps are as follows:
  - Spread the scent of the process
  - Check for a matching tuple at the ant's current location
    - If found return to origin dropping the scent for that template
    - If not found check neighborhood for traces of desired scent
      - If there is no scent, choose the next step randomly.
      - If there is a scent, move towards the scent location (i) with probability P_{ci}.
  - So what happens after several unsuccessful steps?

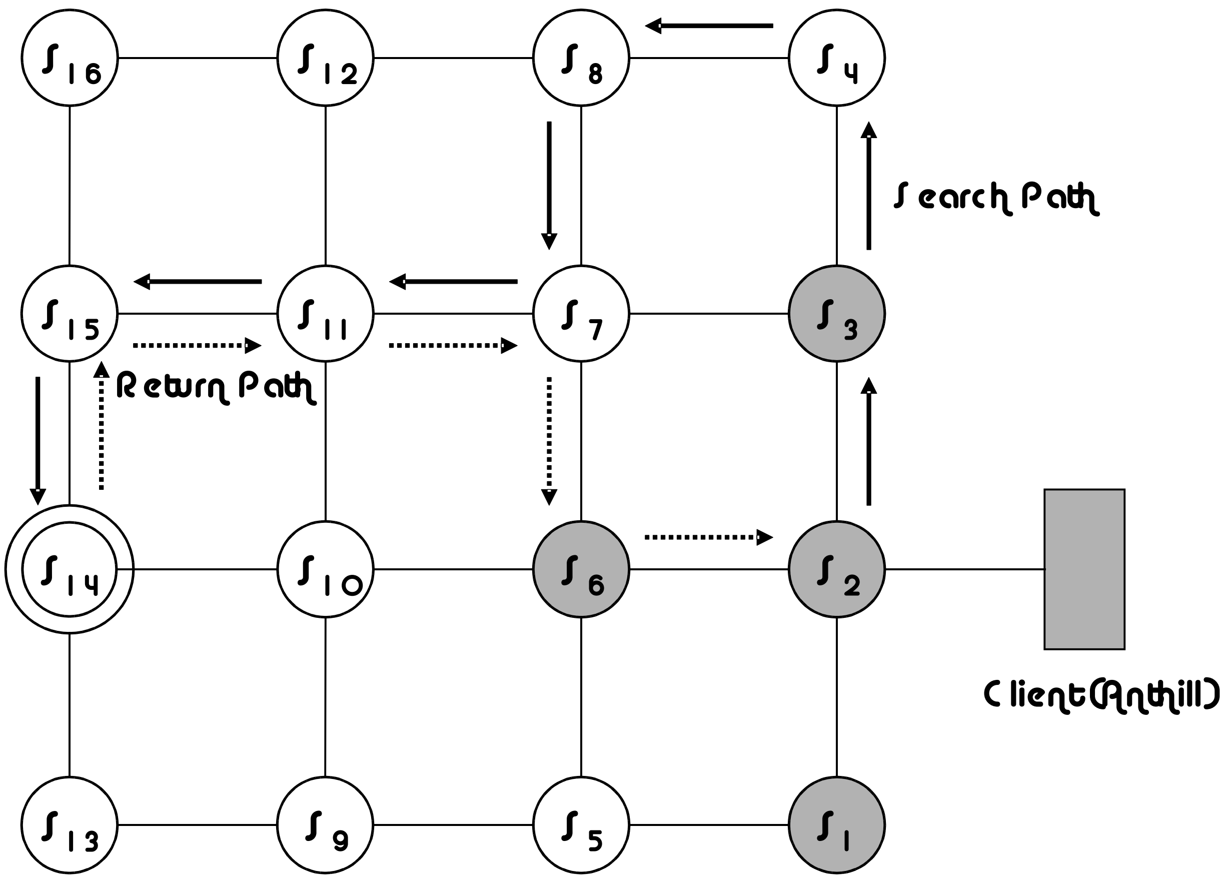
Is it Better Searching Mechanism?

- SwarmLinda describe an emergent behavior
  - Application specific paths are formed between tuple producers and consumers
- The alternative is hashing-based schemes
  - Certainly fast!
  - Not adaptive
    - Tuples always placed in the same location
  - Size of the system has no influence on the choice of location
  - Dynamic configurations are hard to handle
    - And on-the-fly factors may need to be considered when deciding where to place a tuple
      - Load balancing
      - Bottlenecks
      - Scheduled (and unscheduled) down time on the servers

Pictorial View of Searching

- SwarmLinda describe an emergent behavior
  - Application specific paths are formed between tuple producers and consumers
- The alternative is hashing-based schemes
  - Certainly fast!
  - Not adaptive
    - Tuples always placed in the same location
  - Size of the system has no influence on the choice of location
  - Dynamic configurations are hard to handle
    - And on-the-fly factors may need to be considered when deciding where to place a tuple
      - Load balancing
      - Bottlenecks
      - Scheduled (and unscheduled) down time on the servers

Pictorial View of Searching



Distribution Mechanism

- Dynamic mechanism based on brood sorting
- Tuples are ants
- Tuple-space nodes are the terrain
- State is the set of tuples stored thus far
- The steps are as follows:
  - Starting visiting the nodes once an *out* is executed.
  - Observe the kind of tuples the nodes are storing.
    - Each out-ant has a limited memory to guarantee the decision is local
    - Store tuples if nearby nodes contain similar tuples.
      - The likelihood of storing a tuple is calculated stochastically based on the kinds of objects in memory
    - If nearby nodes don't contain similar tuples, randomly choose to drop or carry the tuple to the next node
      - An aging mechanism needs to be in place in this step

Is it a Better Distribution Mechanism?

- Just compare with the standard distribution mechanisms
  - Swarmlinda should be able to improve the availability of the system
- Here there are no
  - Assumptions about the behavior of the applications
  - Pre-defined distribution schema
  - Special scenarios to deal with failures in the nodes

Dealing With Openness

- The openness of a Linda system is dictated by how well it can deal with reconfigurations
- SwarmLinda may show adaptive behavior in reconfigurations we need
  - Tuples as ants
  - The tuple-space nodes as the environment
  - The scents as the state
- With the goal of maintaining similar tuples closed to each other one need to
  - Make tuples (already stored) sense their environment for tuple scents and template scents
  - Based on the strength of the scents the tuple may decide to move
- Note that we're now assuming that *out* also leave scent trails

Balancing the Movement of Tuples and Templates

- The issue here is related to “who” should be moving: tuples or templates
- At may make no sense to have template-ants moving around if the the producers of tuples know how to get to the consumer
  - In this case one might choose to leave the template-ants stationary while tuple-ants move towards it
- This may be achieved with the introduction of a producer-consumer scent randing from -p to +p.
  - Positive values indicate that template ants were successfully matched with a tuple – the location is a producer of tuples
  - Negative values indicate that tuple ants were successfully matched with a template – the location is a consumer of tuples

Implementing SwarmLinda

- Several practical issues need to be addressed when implementing SwarmLinda
  - Here we describe our current approach and where we want to get
- The issues are as follow
  - Network topology
  - Separation of concerns
  - Movement of ants
  - Ant activity and the environment
  - Ants' lifecycle

Network Topology

- All the algorithms assume an environment
- How is this environment implemented?
- Currently
  - Nodes are organized in a (pseudo-)toroidal grid
  - The topology is fixed.
  - Updates are sent in a similar fashion as updates for routing tables in computer networks
- Full implementation
  - Nodes keep dynamic lists of neighbors that can be updated according to their “fitness” as a neighbor
    - How often does a node simply routed the ant to another node
      - This can help the creation of shortcuts
  - Neighbor relation is not symmetric and dynamic

Separation of Concerns

- In Linda terms this normally refers to the ability to create multiple spaces
- However at no point in the algorithms there is a reference to multiple tuple spaces
- The current implementation does not include the MTS concept at the Linda level
  - However Linda primitives still define a tuple space handle
  - The handle is just added as yet another element of the tuple
- In a full implementation
  - The tuple space name may be used as a differential scent
  - This will take advantage of the fact that tuples belonging to the same space belong together for some application specific purpose

Movement of Ants

- The movement of ants (tuple and template alike) are of prime importance to SwarmLinda
- In the current implementation we utilize two concepts
  - Scent strength of the neighbors
    - Simply a counter for the scents
  - Direction System
    - Pre-defined values to directions

30	40	27
10	↑	13
5	5	5

Direction System
sum = 135

1	1	1
6	↑	1
5	3	2

Scent Levels

30	40	27
60	↑	13
25	15	2

Resulting Values
sum = 212

27	13	5
40	←	5
30	10	5

Rotated System
sum = 135

- In a full implementation the above can be seen as list of pairs representing one direction *d = (ds,sl)*

Ant Activity and the Environment

- Ants work in a cycle
  - Observe environment
    - Observe and generate a vector of scents for a given location
  - Plan change of the environment
    - Decide the delta factor to be added to the current vector of values
    - The current approach also add one unit of scent this may be modified to add more according to some different criteria
  - Change environment
    - All delta vectors added give a delta matrix that is atomically added to the environment matrix
- Evaporation of scents can use a similar approach where the values of the delta vector are negative

Ants' Lifecycle

- How do ants die?
- Ants keep a value that is incremented at every step they take
  - This value is the ant life
- Not only this is used in the probability of an ant to drop the food (in case of out), it is also used in consumer primitives
- For consumer primitives, an ant dying consists of a failure mode
  - This is fine with *inp* and *rdp* but not for blocking primitives
- For the blocking primitives
  - Failure causes the ants to be placed in a sleep list – sleep may allow the configuration of the system to change
  - Too many sleeps can yield a recreation of the ant at the origin.
  - Too many recreations may cause a “warp”.

Conclusion

- What have been done
  - We have described schemes for adapting swarm techniques to Linda
  - These techniques provide a good alternative for standard approaches in Linda
  - We have demonstrated that the implementation is very simple and requires very few ideas to be incorporated in the Linda system
  - We have also discussed general approaches for future SwarmLinda implementations
- There is a lot to be learned on this arena
  - Blocking as understood today may be hard to implement
  - Experimental results are now need to validate our theoretical description and analysis