



# A Self-Organized Grouping (SOG) Method for Efficient Grid Resource Discovery

Anand Padmanabhan<sup>1,2</sup>, Shaowen Wang<sup>1</sup>, Sukumar Ghosh<sup>2</sup> and Ransom Briggs<sup>1,2</sup>

<sup>1</sup>Grid Research and education group @ Iowa (GROW), ITS-Academic Technologies

<sup>2</sup>Department of Computer Science, The University of Iowa, Iowa City, IA 52242

{anand-padmanabhan-1, shaowen-wang, sukumar-ghosh, ransom-briggs}@uiowa.edu

**Purpose:** To demonstrate a novel approach to efficient Grid resource discovery by forming and maintaining autonomous resource groups. Each group dynamically aggregates a set of resources that are similar to each other in some pre-specified resource characteristic.

## SOG – A Hybrid Approach

Why?

- Achieves efficient query performance
  - Better performance than decentralized
- Better scalability than centralized
- Controls resource heterogeneity

How?

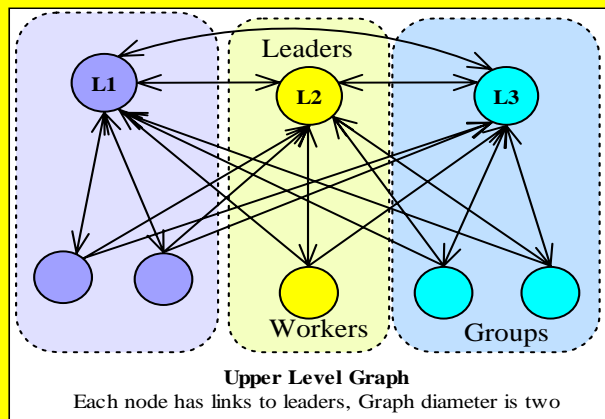
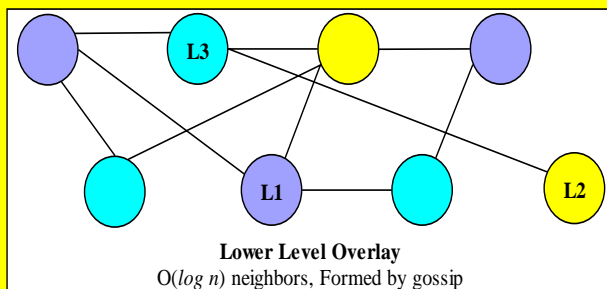
- Collection of resources with similar characteristics form groups
  - Groups represented by a leader
- Controlling resource dynamics by publishing worker resource information to leaders

## Group Formation and Maintenance

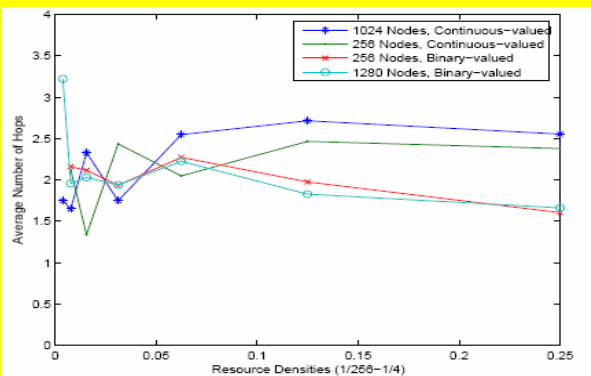
```

Algorithm Group Formation, Maintenance: at node n
event_count ← 0 {Timer count for periodic recalculation of statistic (c)}
1 r ← poll_resource(); {Periodic resource status sampling}
2 Record_entry(R, r); {Add current sample to R, replacing old samples in round-robin fashion if necessary}
3 event_count++;
4 if event_count = x then
5   c ← recalculate_c(R); {Recalculate the temporal characteristics (c) given a set of resource values (R)}
6   event_count ← 0;
7 if l = undefined ∨ (n.isLeader = false ∧ is_not_similar(S(l), c)) then
8   {Either node n belongs to no group or n is a non-leader that has moved away from its group}
9   l' ← find_leader_node(c, L, S); {Find an appropriate leader}
10  if l' = undefined then
11    {No leader was found}
12    gossip_proclaim_leadership(n, c); {Proclaim node as leader through gossip}
13  else
14    {A new leader was found}
15    Inform node l' about n joining its group
16    Inform node l about n leaving its group if l ≠ undefined
17    l ← l' {Assign new leader}
18  end if
19 else if n.isLeader ∧ is_not_similar(c, S(n)) ∧ change_persistent() then
20   {Node n is a leader that has moved away from its group range and the move appears persistent}
21   replace_leader();
22  end if
23  end if
    
```

## Overlay Network



## Single-Attribute Experiments



## Query Handling

- Query handling is metaphorically similar to searching for a word in an English dictionary
- Nodes direct query to an appropriate leader node
  - Based on passive group characteristics
  - From leader election
- Leader node identifies appropriate worker
  - Based on actively published information

## Multiple-Attribute Experiments

|   | Type 1 | Type 2 | Type 3 | Type 4 |
|---|--------|--------|--------|--------|
| <b>Average Hops</b>                       | 2.93   | 1.6    | 1.78   | 0.73   |
| <b>≤ 2 Hops %</b>                         | 54.39  | 83.32  | 73.16  | 95.84  |
| <b>≤ 3 Hops %</b>                         | 72.59  | 95.78  | 85.23  | 99.37  |
| <b>≤ 5 Hops %</b>                         | 89.63  | 99.64  | 95.62  | 99.95  |
| <b>≤ 10 Hops %</b>                        | 98.97  | 100    | 99.65  | 100    |
| <b>Effective Average Hops<sup>†</sup></b> | 3.18   | 2.07   | 2.66   | 1.71   |

## Main Contributions

- A novel approach to Grid resource discovery
  - Self-organizing scheme
    - Resource groups are dynamically formed and maintained
    - Grouping based on statistical characteristics
- Evidence that SOG provides better performance than existing solutions
  - Queries are handled in 2-3 hops
- Resource density, resource type and system size seem to have little impact on SOG performance

For more information contact: Shaowen Wang (319/335-6713) / Email: [shaowen-wang@uiowa.edu](mailto:shaowen-wang@uiowa.edu)

Web Site: <http://www.uiowa.edu/~grow>