A Topology Viewer For Distributed Brokering Systems

Amey Dharurkar, Shrideep Pallickara, Geoffrey Fox
Community Grids Lab
Indiana University

http://www.naradabrokering.org

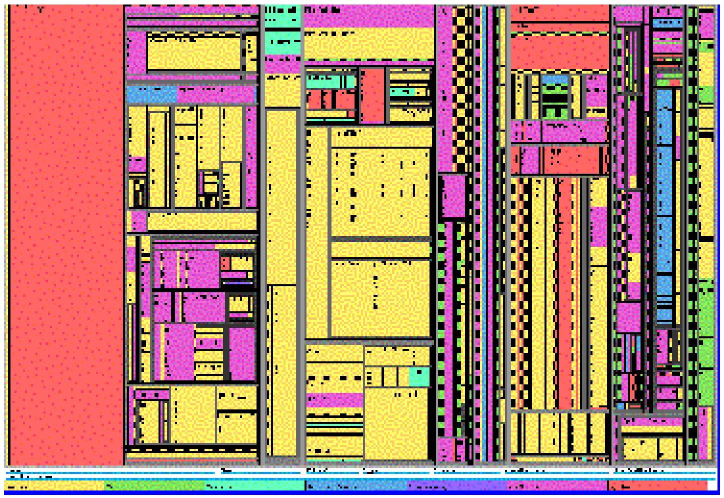
Talk Outline

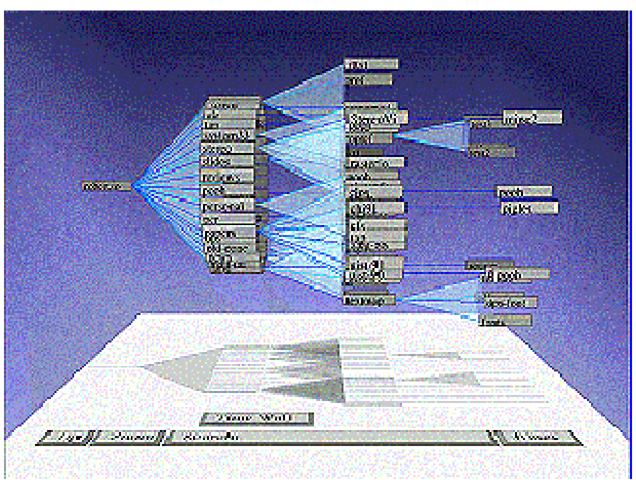
- Introduction
- Related Work
- Our approach
- Results
- Conclusions and Future Work

Introduction

- Network ⇔Graph Visualization
- Challenges
  - Scalability – Displaying large number of nodes and links on screen without losing context.
  - Navigation and Efficiency – Traversing through the networked graph and fetch the information associated with the components.
  - Adaptability – Reconstruction of the graph depending upon dynamic change in the components.
- Typical Graph – Hierarchical in nature
  - Clustering technique helps maintain abstraction

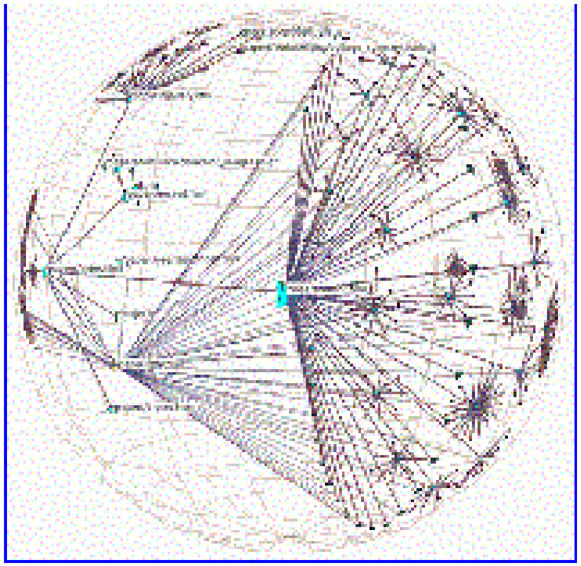
Related Work – I

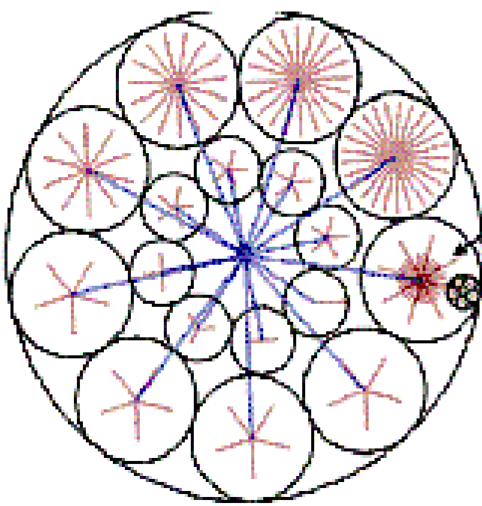




- Treemaps – Space Filling method.
  - Nodes are represented by rectangles. Rectangles are packed in 2D plane.
  - 100% utilization of the space.
  - Not scalable.
- Cone Trees
  - Rooted tree layout
  - Children are placed on the circumference around the cone.
  - Limitation on size of the graph.

Related Work – II





- 3D Layout
  - Use of Hyperbolic geometry
  - Rendering is time consuming
- RINGS
  - Children placed as equal size circles in the concentric rings around center of the parent circle.
  - Minimum edge crossings.
  - Easy navigation – Clicking on the node makes it the focus while overall context is maintained.

NaradaBrokering Overview

- Open source project. http://www.naradabrokering.org
- Provides a variety of services
  - Reliable, ordered and exactly once delivery.
  - Compression and fragmentation of large payloads.
  - Performance Monitoring service
  - Support for multiple subscription types
- Used in the context of A/V applications and to enhanced Grid apps such as Grid-FTP
- Provides support for variety of transports: TCP, UDP, HTTP, SSL, Multicast and parallel TCP streams.
- JMS compliant. Will provide WS-Notification support.
- Includes bridge to GT3. April 2004 release.
- Support for Web Services being incorporated.

NaradaBrokering: Topology Viewer Goals

- NaradaBrokering
  - Runs on large number of co-operating broker nodes.
  - Broker Node is smallest unit and is used to route messages encapsulating any sort of information.
  - Inherent hierarchical nature. Four levels of clusters and so four levels of abstractions.
- Goals of the Topology Viewer
  - Should be quite scalable (Max Nodes = 32 * 32 * 32 * 32)
  - Algorithmic and Navigational efficiency.
  - Incorporate dynamic changes in the network with minimum variation in the layout.

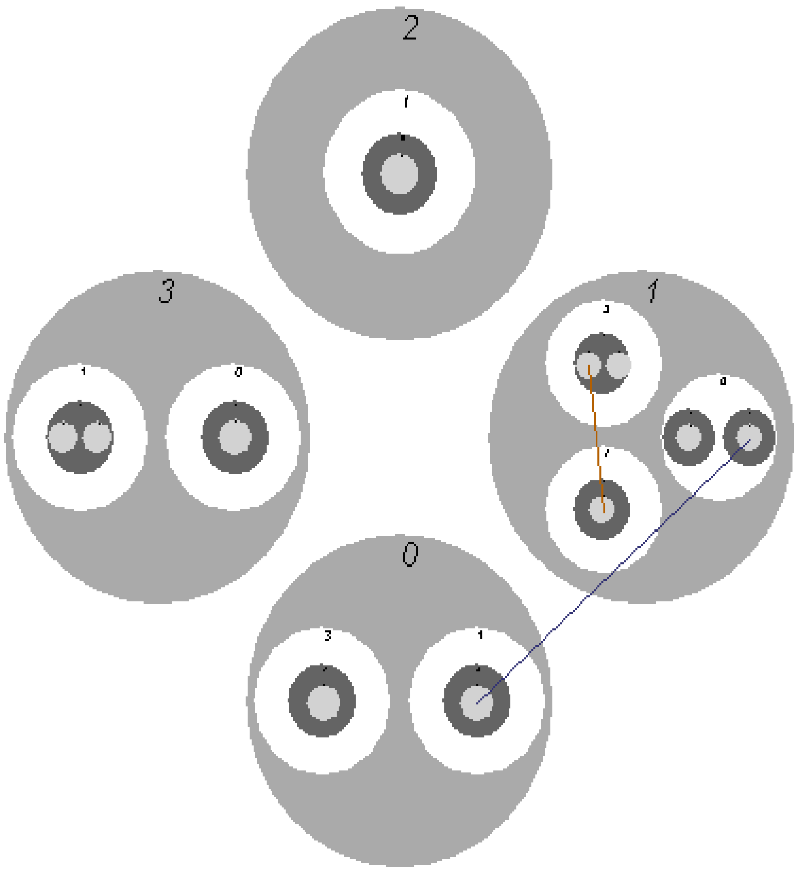
Layout Algorithm

- Node Placement
  - Graph placed inside a BIG circle. Highly Symmetric.
  - Each level (Super Super Cluster, Super Cluster, Cluster and Broker Node) is represented by a circle and components in a same level are placed inside the parent circle.
  - As opposed to RING scheme, children are placed in a single ring instead of multiple concentric rings
    - Improves calculation and navigational efficiency BUT poor space utilization
  - If n children inside parent and then angular separation is $2\pi / n$.

Layout Algorithm – Continued

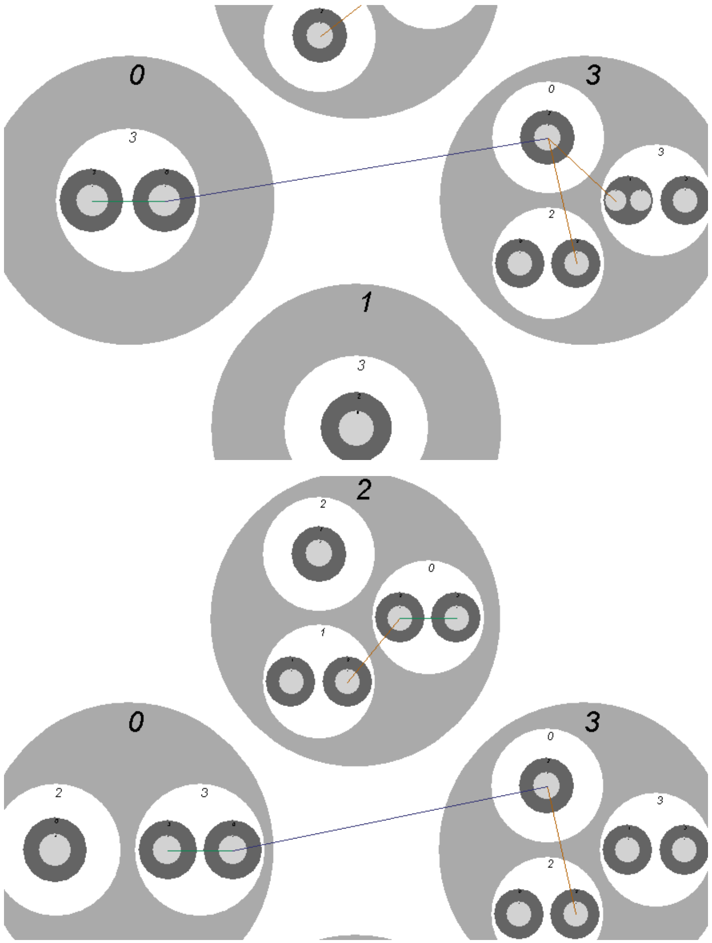
  - Radius of a circle can be easily calculated. If angular separation is Θ then the radius would be $R \sin(\Theta/2)$. R is the radius of parent circle.
  - Decision whether a component lie on the screen is simple. Due to abstraction, the decision can be made with few comparisons.
  - Based on the mouse coordinates, unique node can be located with few comparisons.
- Edge Placement
  - Straight edges reduce drawing and calculation overhead. Minimizing edge crossings.
  - Edges between nodes at diff levels in diff colors.

Sample View – Explanation



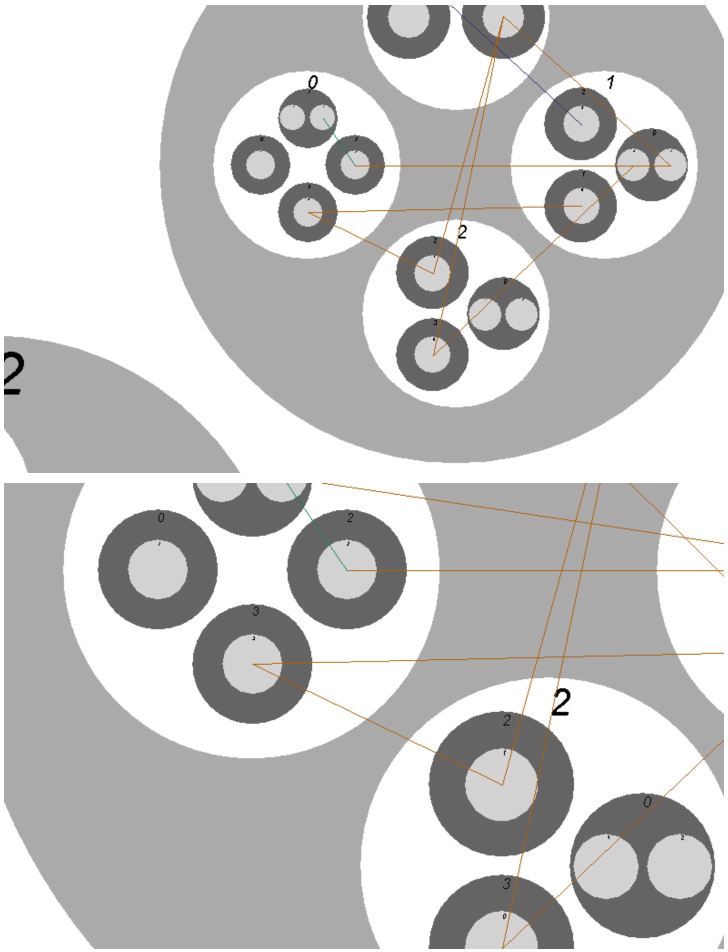
- Entire Diagram lies inside in BIG virtual circle.
- Outermost circles represent super-super clusters, then super-clusters and so on.
- Cluster/ Node Ids are shown at the top.
- Edges are straight and have different colors.

Navigation – Translation



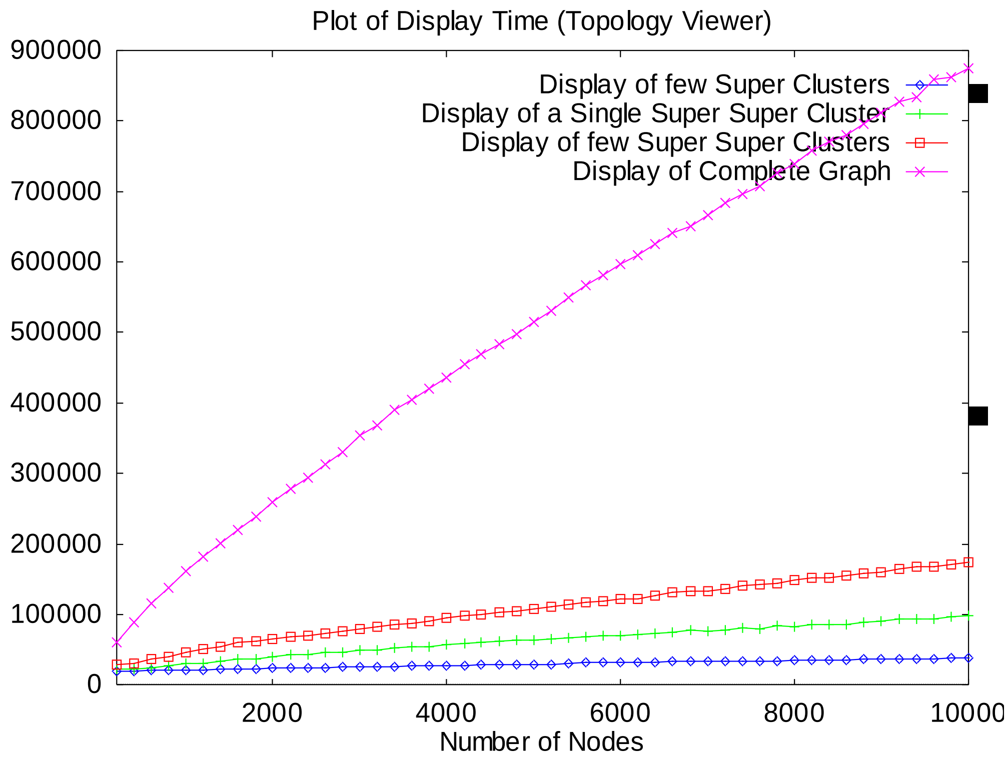
- Difference between the old center of the graph and the new focus is calculated.
- All the centers of components are recalculated.
- Circle - Best choice as it can described by only two parameters (x, y)
- Whole graph is shifted so that new focus becomes center of the screen.

Navigation – Zoom In/ Out



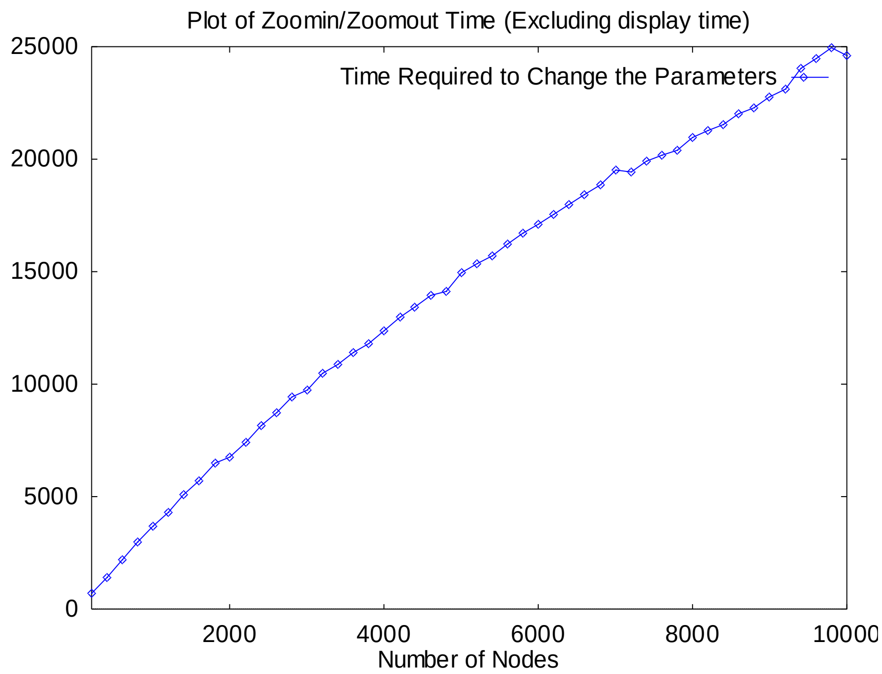
- Two figures correspond to views before and after zooming.
- Radius of each circle is increased by fixed constant.
- Center of the outermost virtual circle is recalculated.
- Whole picture is redrawn on the screen.

Performance – Drawing Response



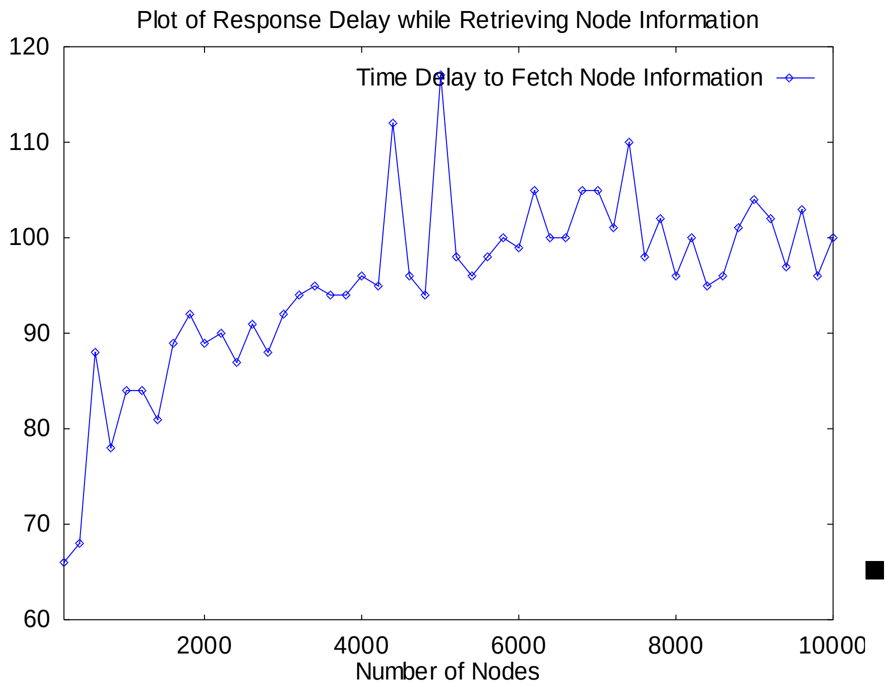
- The system has been tested up to 10000 nodes.
- Restricted view (Few components), display time reduces (due algorithmic and visual simplicity)
- Degree of independence on the number of nodes in system is high. Hence Scalable.

Performance – Zoom Operations



- Adjacent graph shows time required to perform all recalculations and excludes the display time.
- As expected the response time increases linearly with number of nodes.
- Time is in tens of milliseconds is small fraction of actual display time. So does not involve significant overhead compared to display time

Performance – Node Operations



- Complexity for determining a node address from (x, y) coordinates is also O (N). Because of symmetry in the system such decision can be made in at most

  4 * N comparisons.
- Independent on number of nodes in the system. Stabilizes after some node additions.

Conclusion and Future Work

- Very scalable and clear.
- Incorporate dynamic changes in the network.
- Algorithmic and visual simplicity.
- What Next !
  - Provide mechanism so that one can access to monitoring service associated with each node. Viewer acts as information fetching tool.
  - Allow user run their own instances of the viewer and synchronize them with the viewer running at main site

Percentage Space Wasted

- Assume that Radius of the virtual circle is R
- If number of children (Super-super clusters) are N
- Total Available space is ΠR^2
- Space used is $N * \Pi * R^2 \sin^2 (\Pi/N)$
- Percentage space wasted is $(1 - N \sin^2 (\Pi/N))$ %
- As N increases, value of $\sin^2 (\Pi/N)$ decreases drastically and so more space is wasted.