Visualizing Networks and Trees

Arrange Networks and Trees
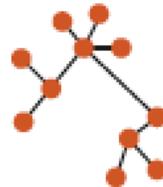
Node-Link Diagrams for Trees

- The most common visual representation
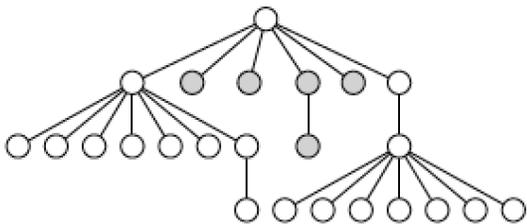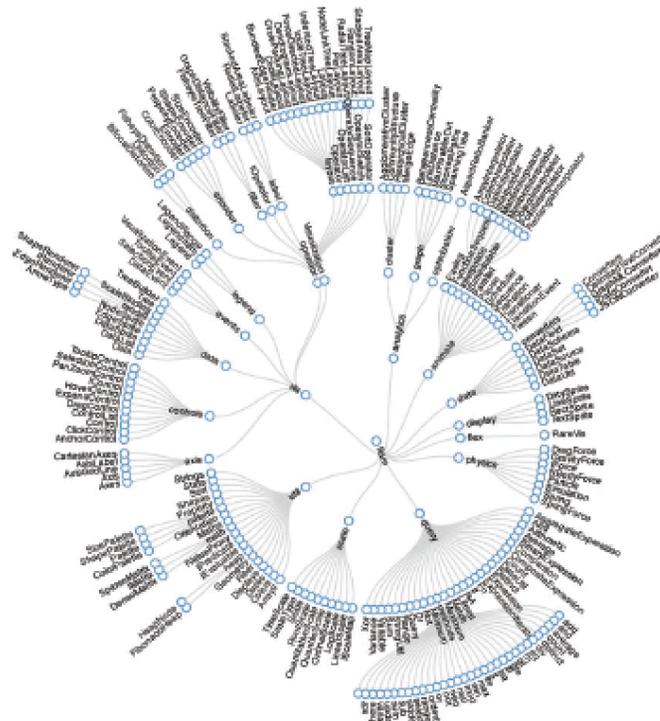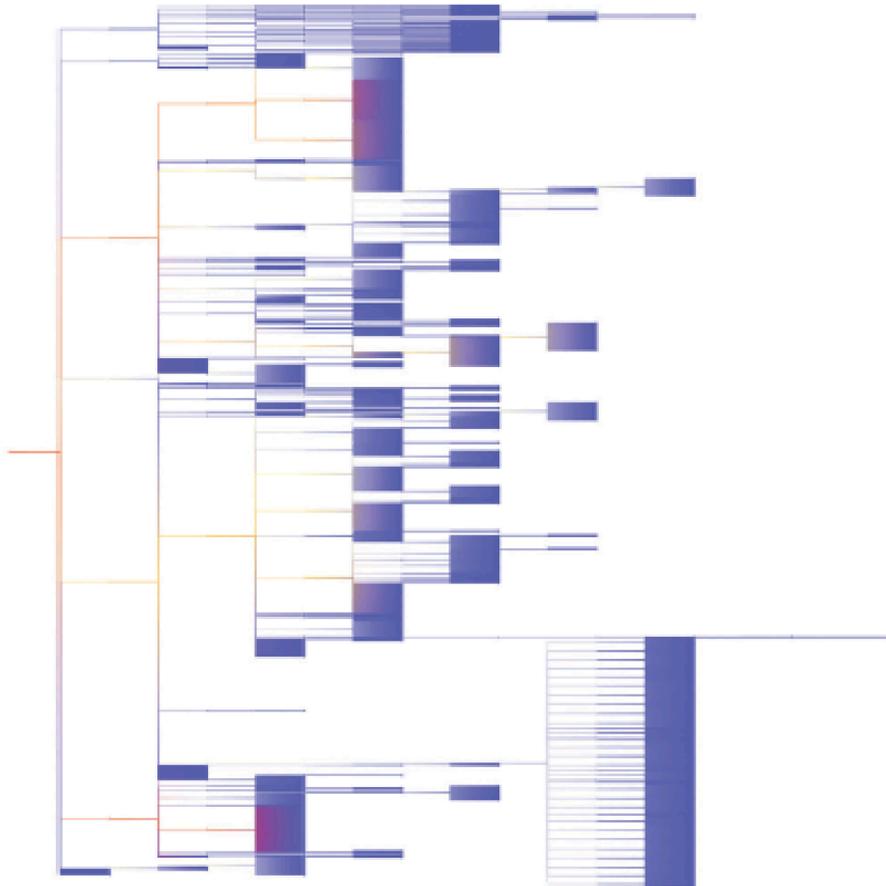 - Nodes: point marks; Links: line marks

Triangular vertical layout -
vertical spatial location to show
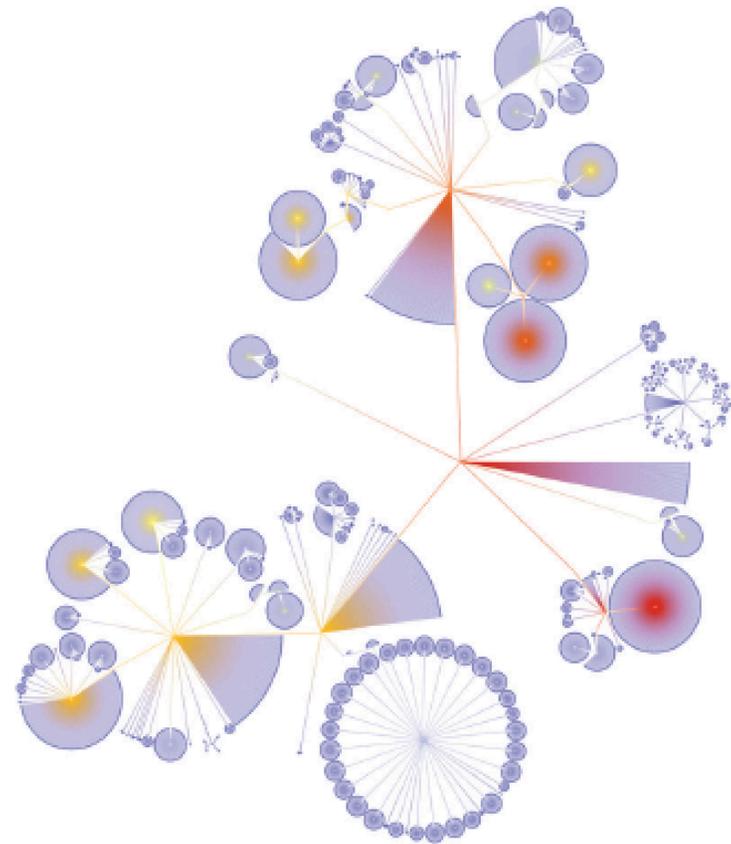the depth

Spline radial layout – depth is shown as distance to
the center

Larger Trees

Rectangular horizontal layout

Radial bubble tree
- each subtree is a circle

# Node-Link Diagrams for Networks

- Also known as graphs
- Distance is measured as hops
- It is often used to analyzed the topology – direct and indirect connections between nodes
  - Finding all possible paths from one node to another
  - Finding the shortest paths between two nodes
  - Finding all the adjacent nodes from a target node
  - Finding nodes that act as bridges between two groups of nodes

Visualizing Networks

- Force-directed layout
- Matrix
- Enclosure
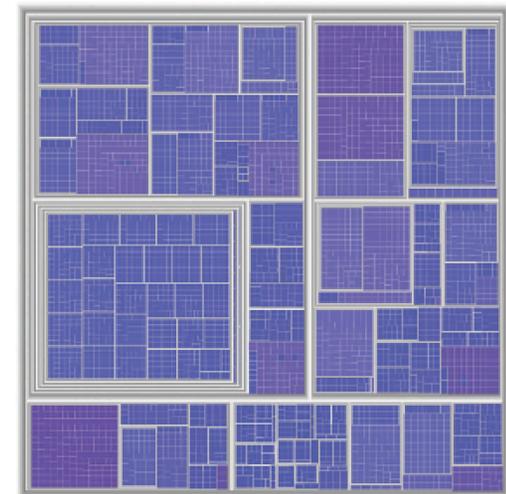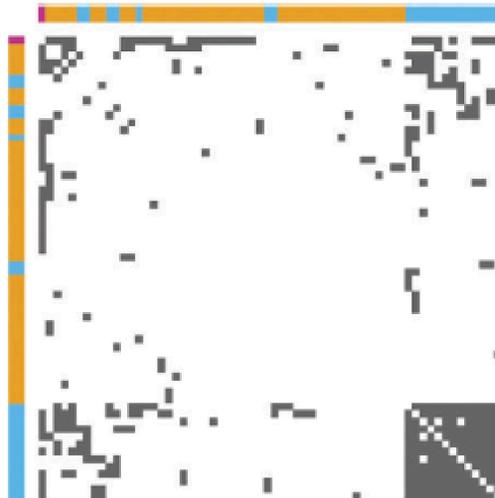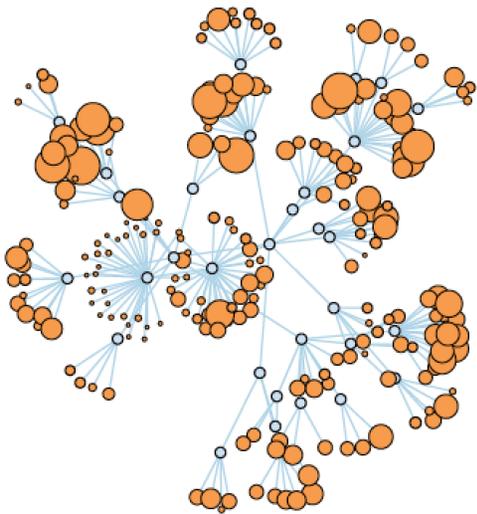
Force-Directed Layout

- Nodes are placed according to a simulation of physical forces
  - Links act like springs
  - Nodes push each other away
- Nodes are placed randomly initially and their positions are updated iteratively
- The goal is to minimize the number of distractions such as edge crossing and node overlap
- Nodes and links can be colored coded with different sizes to represent attributes

(a)

(b)
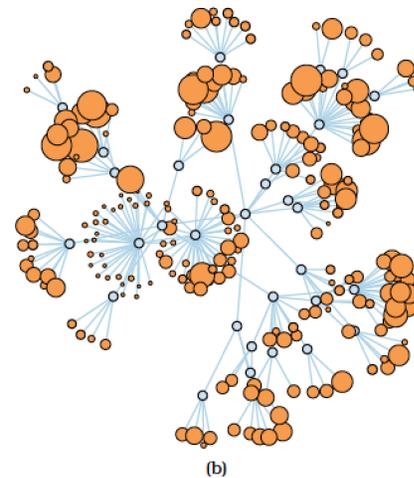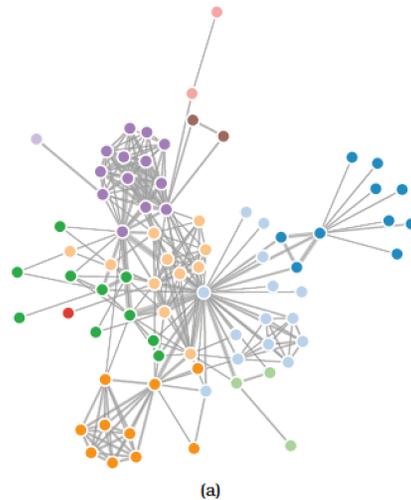
Force-Directed Layout

- Major weakness – scalability
- Sometimes the nodes never settle down
- Multilevel drawing can help

JGD_Homology@cis-n4c6-b4. 26028 nodes, 100290 edges.
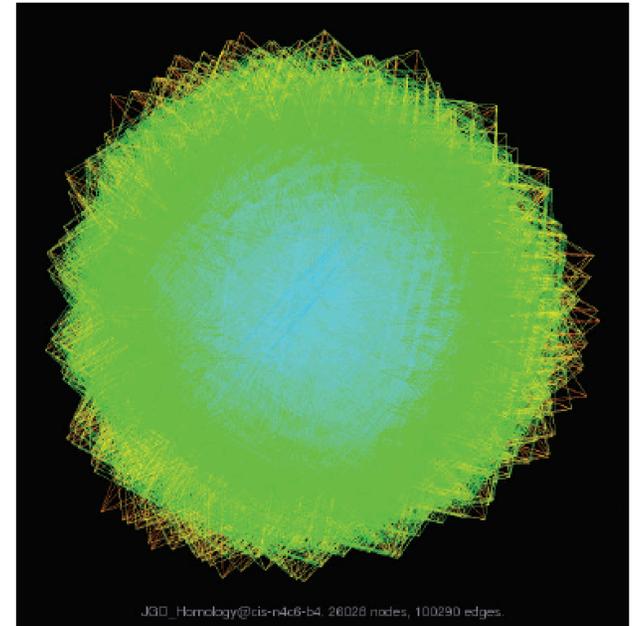
20 K nodes – hairball problem

| Idiom	Force-Directed Placement
What: Data	Network.
How: Encode	Point marks for nodes, connection marks for links.
Why: Tasks	Explore topology, locate paths.
Scale	Nodes: dozens/hundreds. Links: hundreds. Node/link density: $L < 4N$

Matrix Views for Networks

- Nodes are labeled in rows and columns and each cell in the matrix represents whether there is a direct link

- Matrix cells can be encoded with different colors

- For undirected networks, i.e., links have no directions, only half of the matrix is needed
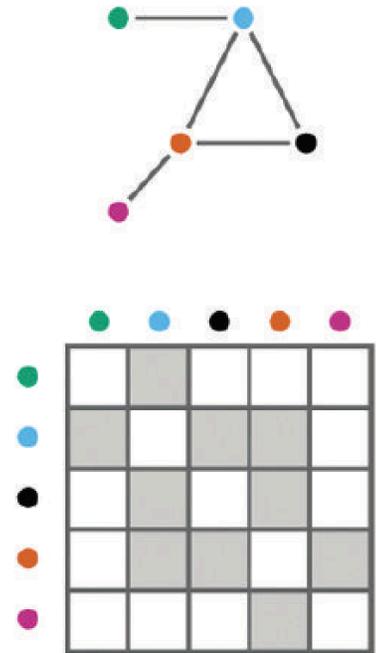
Matrix Views for Networks
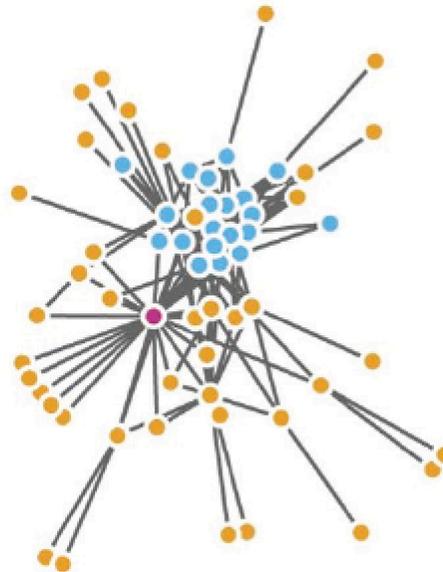
Larger networks

| Idiom	Adjacency Matrix View
What: Data	Network.
What: Derived	Table: network nodes as keys, link status between two nodes as values.
How: Encode	Area marks in 2D matrix alignment.
Scale	Nodes: 1000. Links: one milllion.

Costs and Benefits

- Node-link diagrams
  - Intuitive for small networks
  - Good for understanding network topology and substructures
  - Major weakness in handling large networks
    - Link density: number of links vs. nodes
    - Hard to handle networks with link density > 4
- Matrix views
  - Can handle large networks effectively
  - High predictability in the required screen space
  - More stable views: add a node/link won't change the view much
  - Roles/Columns can be re-ordered to show structures
  - It is easier to find a node in the networks : check the labels
  - Weakness: harder to interpret, often requires training
  - Weakness: lack of support for analyzing network topology
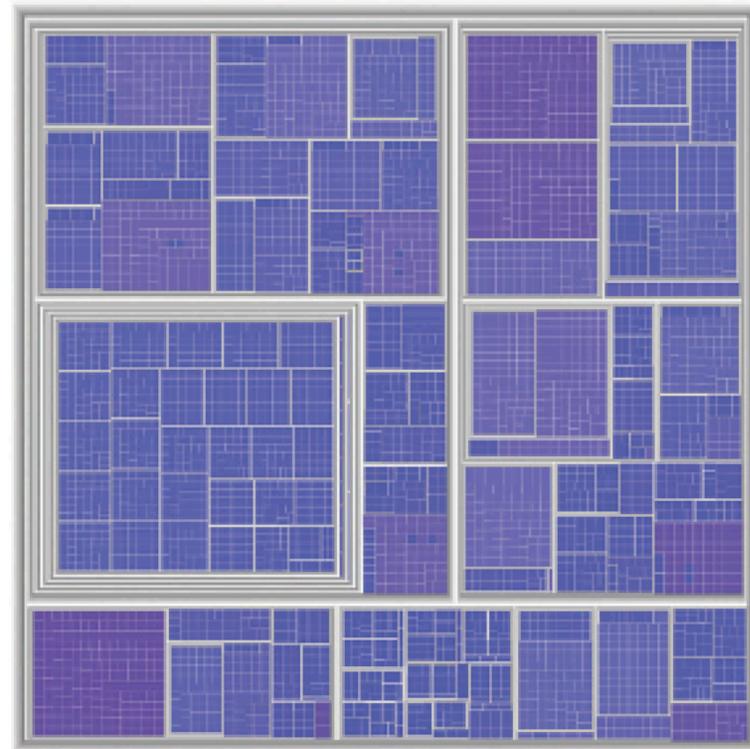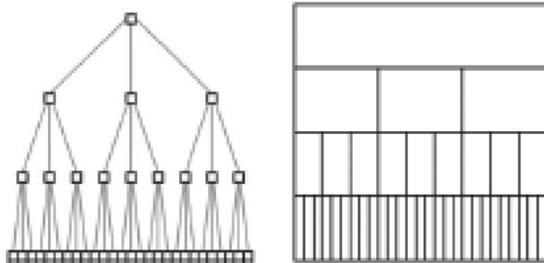
Costs and Benefits

- Node-link diagrams
  - Intuitive for small networks
  - Good for understanding network topology and substructures
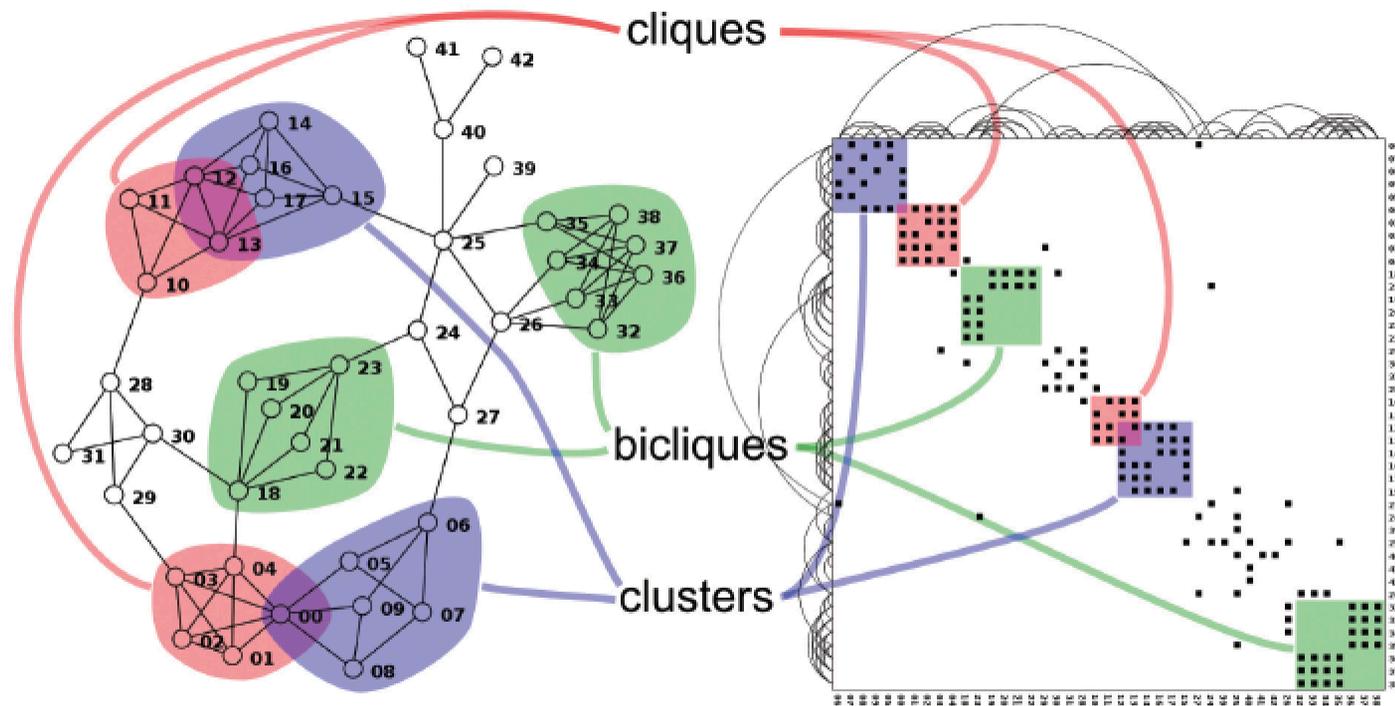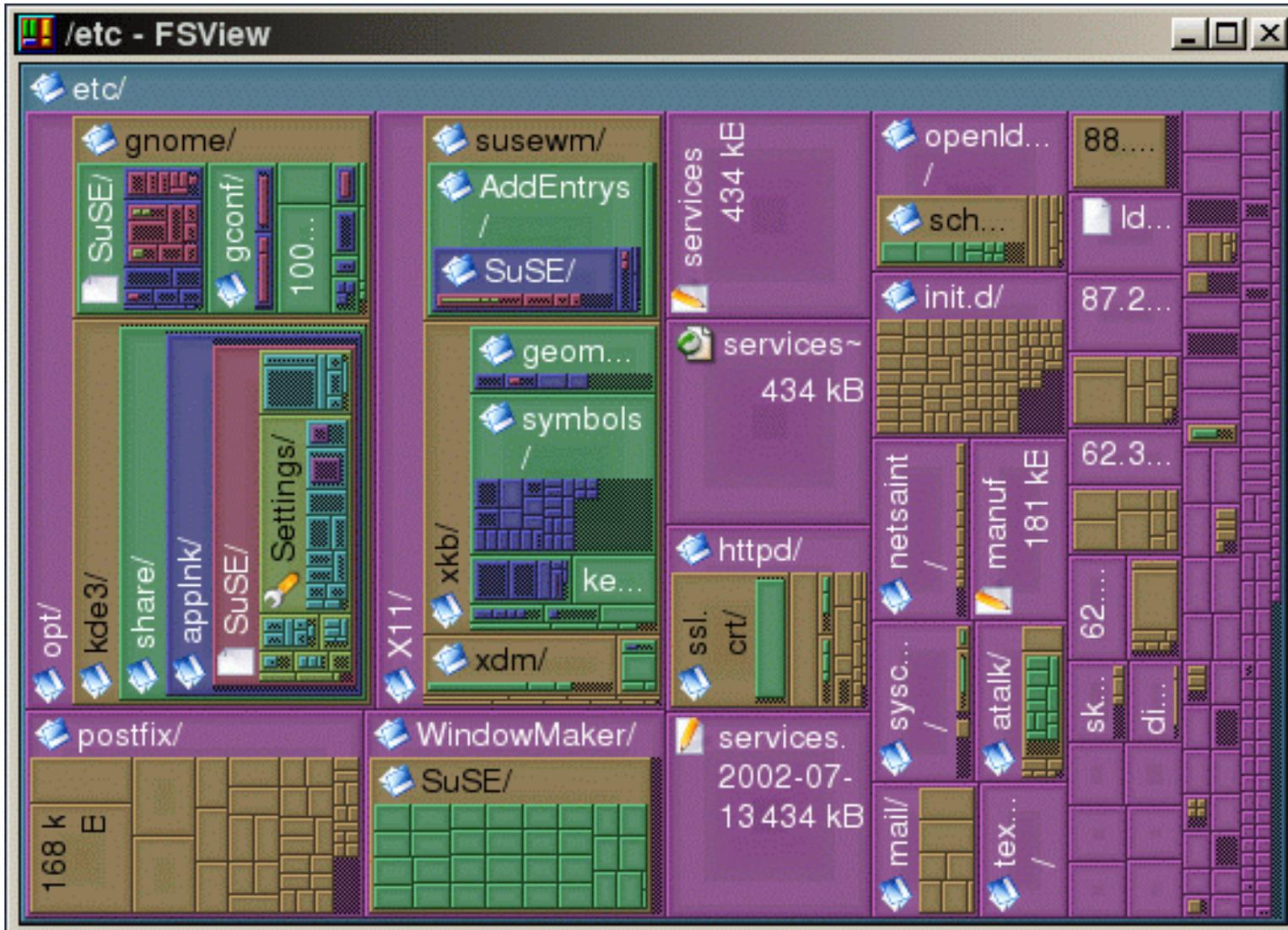

nuch

Containment: Hierarchy Marks

- The focus is to show the hierarchical structure
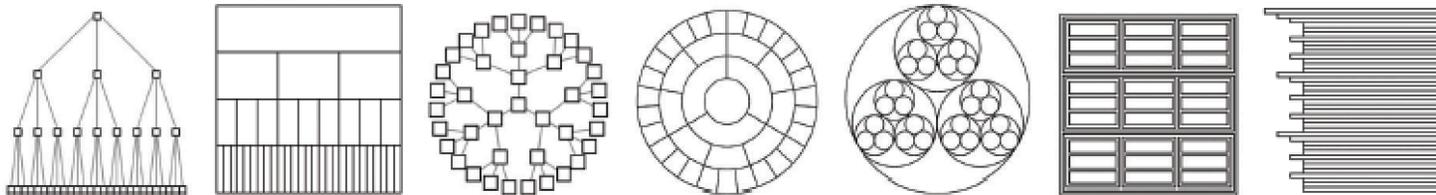 - Node-link diagrams only show pair-wise relationships
- Example: Treemaps

Treemaps

Treemaps

Idiom	Treemaps
What: Data	Tree.
How: Encode	Area marks and containment, with rectilinear layout.
Why: Tasks	Query attributes at leaf nodes.
Scale	Leaf nodes: one million. Links: one million.

More visual encoding of hierarchies