Storing Trees on Disk Drives

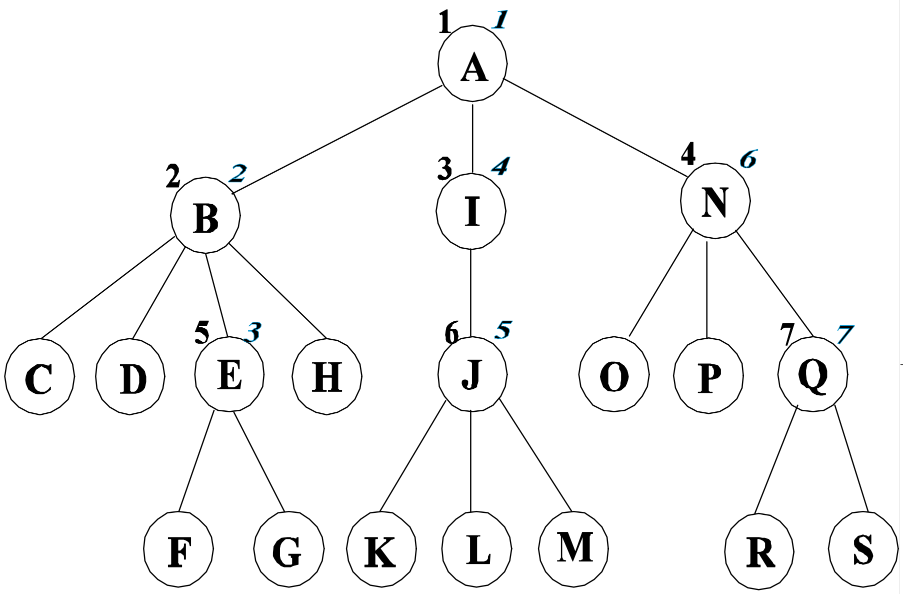
Medha Bhadkamkar, Fernando Farfan, Vagelis Hristidis, Raju Rangaswami

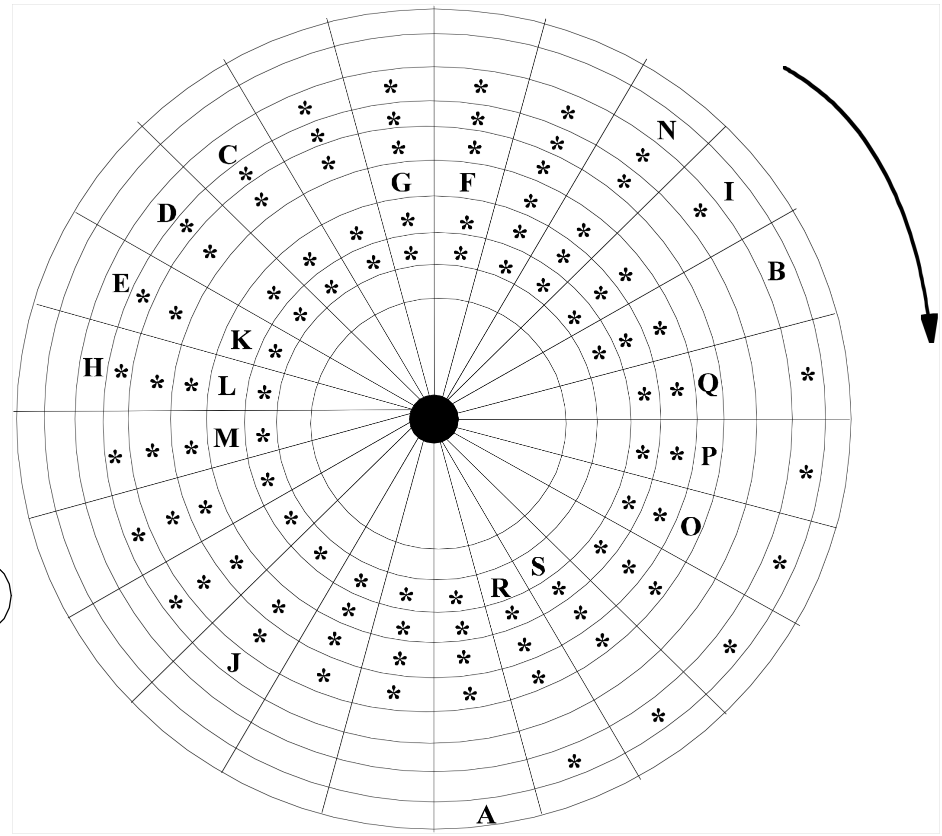
FLORIDA INTERNATIONAL UNIVERSITY

Introduction

- Tree data are becoming commonplace:
  - Offer an intuitive, natural way for organizing information.
  - Examples: XML, multi-res video, natural sciences data (e.g. Bioinformatics), even traditional directory-file hierarchies.
- Disk drives are ubiquitous and seem irreplaceable
- Current approaches:
  - Use relational databases
  - Use flat files
- Our contributions
  - Examine the tree storage problem
  - Propose native data layout strategies for tree data

Tree Structured Placement





Idea: Optimize common accesses

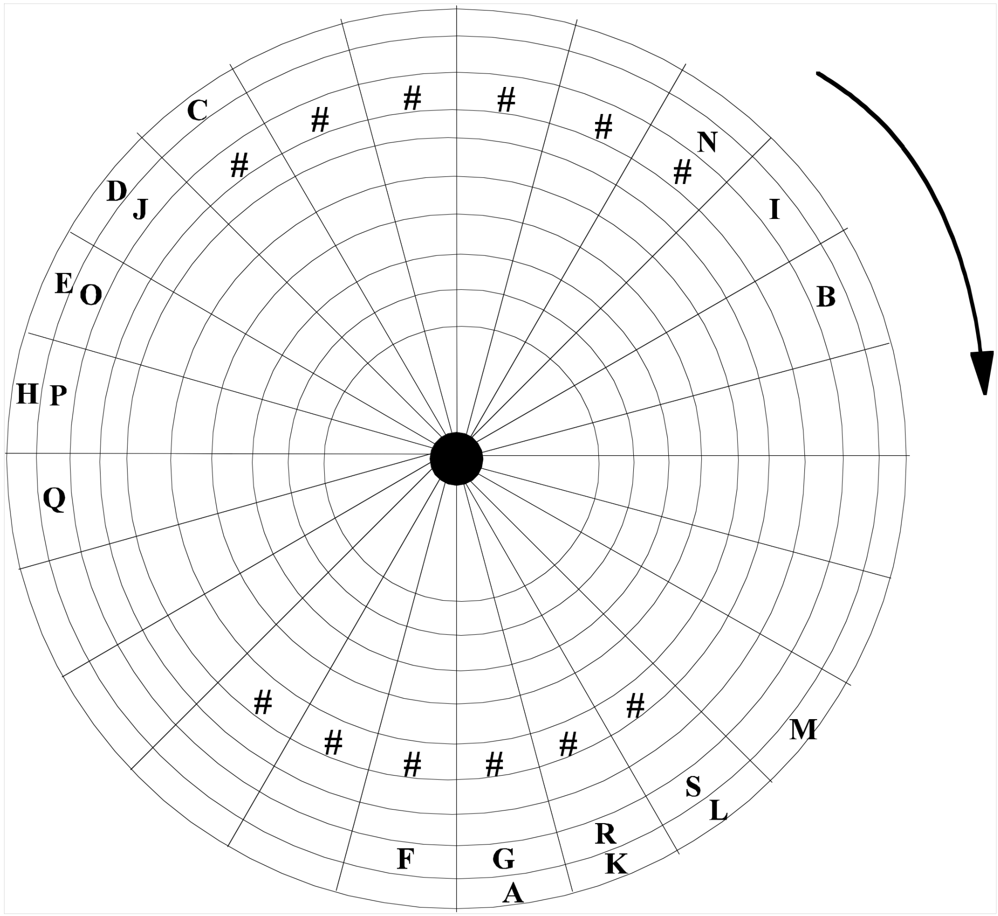
- Parent to child
- Node to sibling

Assumptions:

- Each node occupies an entire disk block
- Semi-sequential access information available

Optimized Tree-Structured Placement

- Problems with basic tree placement:
  - Significant fragmentation.
  - Large random seeks
- Solution:
  - Use non-free tracks
  - Use rotationally-optimal track-regions

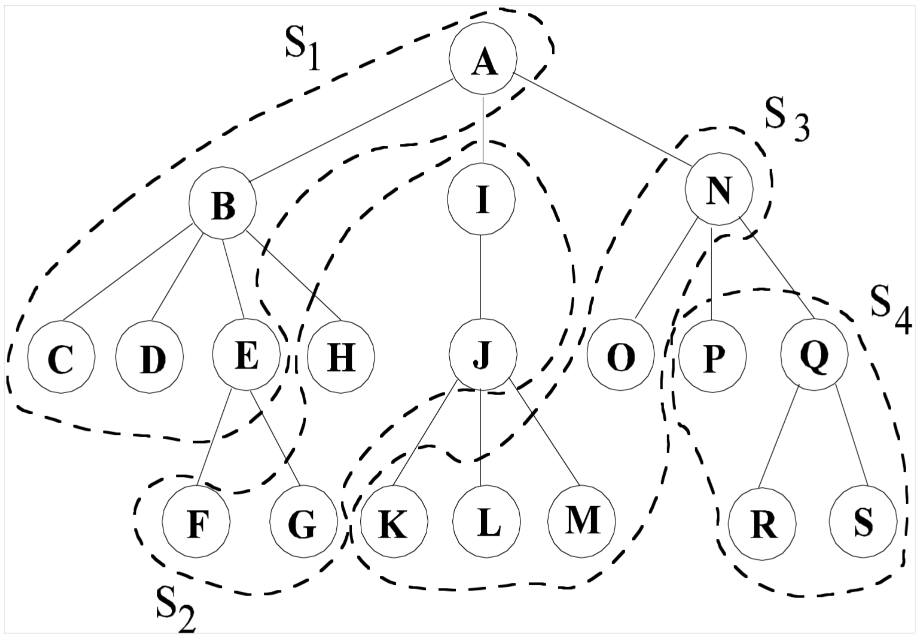


Grouping

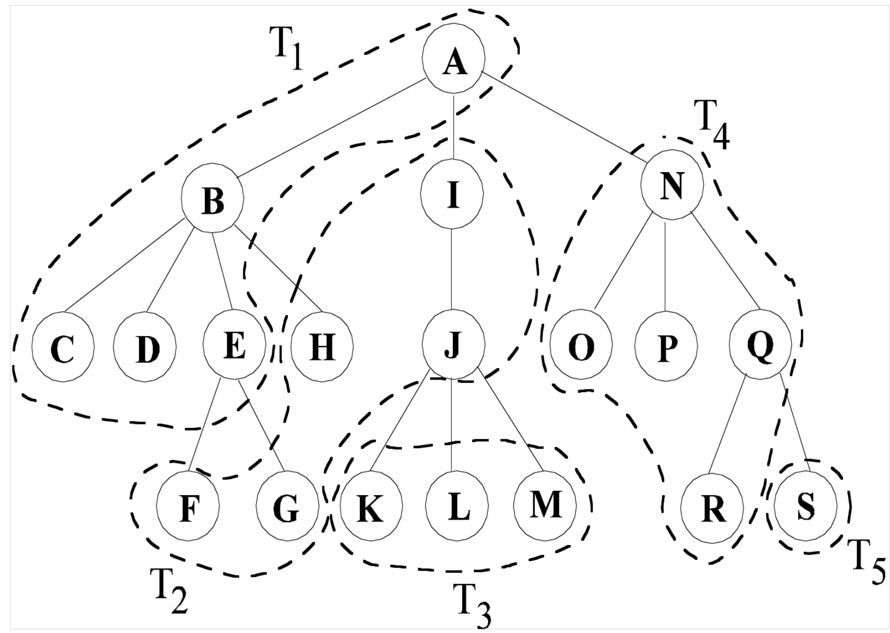
- Sequential
  - Add nodes to ‘supernode’ until its capacity allows.
  - Use depth-first traversal to get next node
  - Low fragmentation
- Tree-preserving
  - Groups adjacent nodes
  - Avoids cycles in original tree
  - Preserves original tree structure in grouping
  - Greater fragmentation

Grouping Examples

Sequential



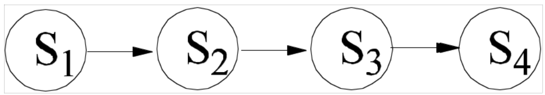
Tree-preserving



Assumption: Supernode can fit 5 nodes

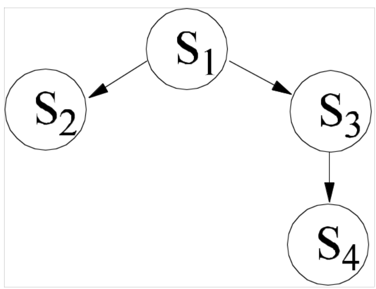
Building Supernode Trees

Sequential Supernode List



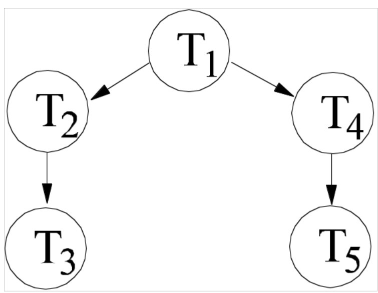
- Uses sequential grouping
- Nodes linked in the order they are created

Tree-Preserving Supernode Tree



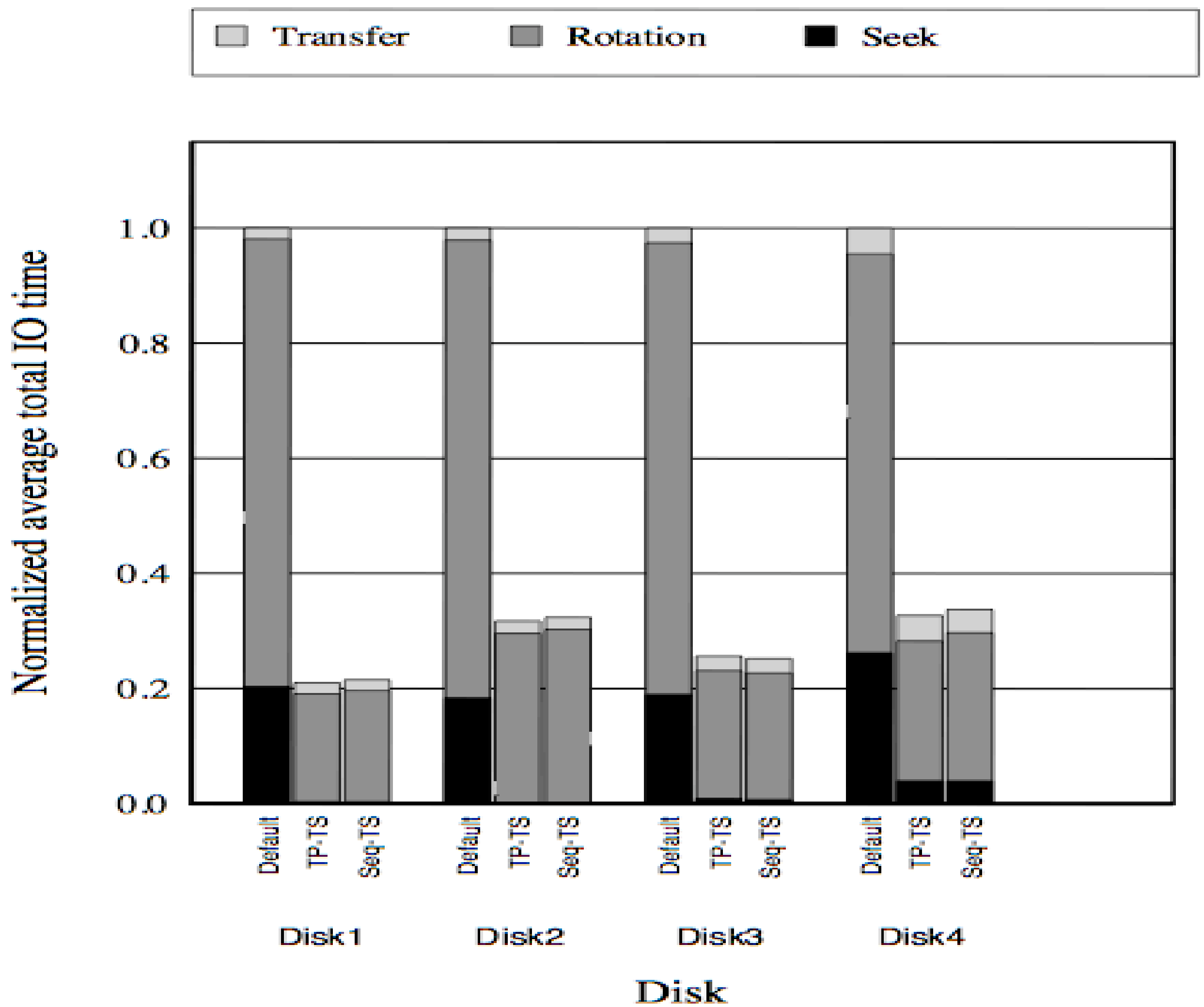
- Uses tree-preserving grouping
- Edges according to original tree

Sequential Supernode Tree



- Uses sequential grouping
- Several possibilities for edge creation
- Avoid cycles

Performance Evaluation



Future Work

- Multiple drives

- Modeling more complex data and access patterns
  - Allows *data* and *application* directed layout
  - Requires detailed model for the disk-drive

- Storing **graphs** on disk drives…
  - More generic than trees!
  - Can use directed and weighted
  - Can model several data-types and access patterns
  - Can model relational data as well!