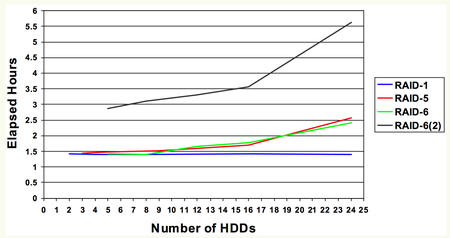
Grouping Data for Faster Rebuilds: The Art of Failing Silently

Avani Wildani, Ethan Miller
University of California, Santa Cruz, CA

Overview

- Data is accessed in groups
  - Groups: any set of data accessed together
- It is possible to predict grouped accesses
- Reliability schemes trade availability for lowered costs
- Rebuild can cause long periods of unavailability

Rebuild Time for the IBM DS5000



Graph from the IBM RedPaper "Considerations for RAID-6 Availability and Format/Rebuild Performance on the DS5000". Disks are 300GB.

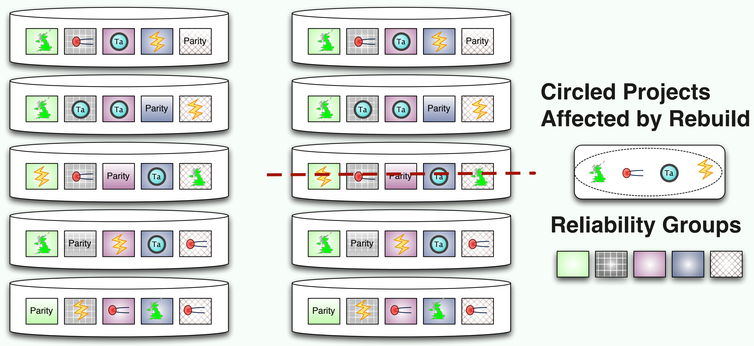
- Data is typically placed evenly across reliability groups
  - Many access groups affected per rebuild
- Correlated failures could cause many groups to have to go to secondary/tertiary storage
- **Goal**: **Lay out grouped data on disk so that as few distinct groups of data as possible are affected by a rebuild or correlated failure on an erasure-coded system**

Data Processing

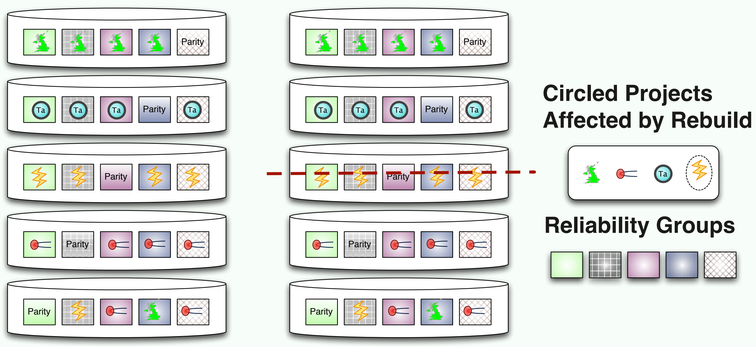
- Set sliding window size based on available memory and workload domain knowledge
- If metadata exists, group first by metadata
- Otherwise, calculate $m \times m$ distance matrix
  - m : number of unique block offsets
  - Distance metric: Use spatio-temporal locality
  - Done in real-time per window
- If a group of elements is close to most other elements, remove them (on-disk cache)
- Group per window in the background
- Remove all groups of size 1

Rebuilding

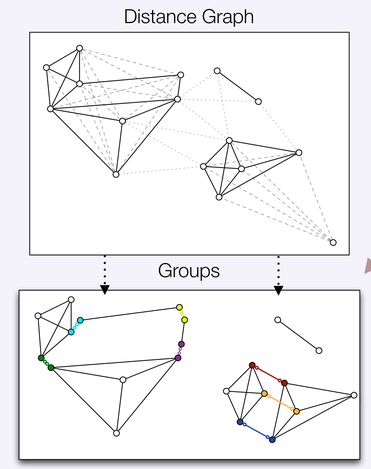
Random Data



Isolated Data



Grouping



- Nodes: block offsets
- Edges: pre-processed distances
  - Above a similarity threshold, edges are removed
  - Set threshold value proportional to desired group size
- Clique cover on remaining graphs determines groups
  - Colored nodes indicate nodes that are members of multiple groups
- If previous grouping exists, combine groupings by fuzzy set union + symmetric difference
- Good way to get groupings of a desired size without overfitting
- Update group likelihoods based on most recent grouping

Runtime

- Recording and splitting accesses: $O(1)$
- Calculate distance matrix: $O(m^2)$
- Grouping: $O(m^G)$, where G is your maximum group size
- Most groups are small, so runtime ~ $O(m^2 + j^3 + k^4 + ... + z^G)$ where $m >> j >> k >> z$
- Combining groupings: $O(n)$, where n is the number of groups

Experimental Status

- Workloads collected: large enterprise, water quality, state records, academic
- Fault injecting simulator written in Python
  - Faults are added uniformly biased by an increased chance of failure per disk access
  - Rapid scrubbing to detect "silent" failures

Next Steps

- Add correlated failure to fault injector
- Estimate productive time lost
- Examine effect of different scrubbing frequencies
- Examine different underlying reliability schemes
- Power impact

Acknowledgements

Supported in part by the National Science Foundation under awards CNS-0917396 and IIP-0934401. We also thank the industrial sponsors of the Center for Research in Intelligent Storage and the Storage Systems Research Center for their generous support.