GraphLens: Mining Enterprise Storage Workloads Using Graph Analytics

Yang Zhou, Sangeetha Seshadri, Lawrence Chiu, Ling Liu
Georgia Institute of Technology, IBM Almaden Research Center

Motivation

Background

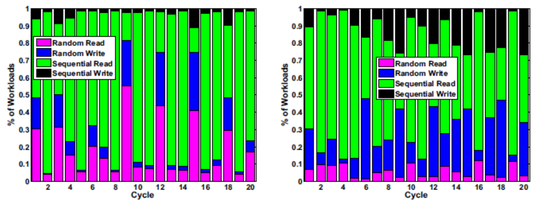
➢ Multi-tier enterprise storage systems optimize system performance and maximize utilization by migrating hot data to faster devices and cold data to appropriate devices

➢The commonly used approaches collect statistics on data access patterns and evaluate data at regular intervals – i.e. “decision windows” in order to identify which data is hot and which data is cold

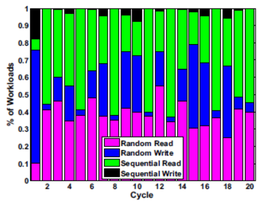
Challenges

➢**Decision window selection:** Static decision window selection results in inefficiencies, sub-optimal system performance or sub-optimal utilization
➢**Choosing the right optimization metric:** Sequential and random workloads require optimization based on different metrics
➢**Scheduling:** Identify workload changes and periodicity to perform better scheduling thereby improving responsiveness
➢**Identifying Dependencies:** Optimal placement strategies

Example: I/O Workloads by Four Access Patterns



(a) Bank Trace (b) Email Trace



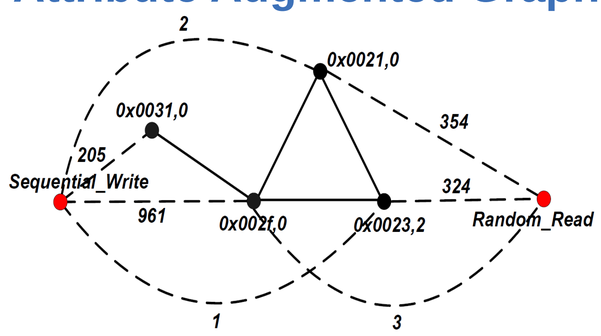
(c) Store Trace

Graph-based Approach

Graph Modeling

➢Structure vertex: the combination of Lun and Extent
➢Attribute vertex: random-read, random-write, sequential-read, sequential-write
➢Structure edge: the edge between structure vertices
➢Attribute edge: the edge between structure vertices and attribute vertices

Attribute Augmented Graph



Methodology Overview

➢Collect access statistics for each extent with six attributes
➢Generate the attribute augmented graph
➢Based on the attribute graph, partition extents into clusters (Hot, Warm, Cold) on each time cycle based on multiple attributes

Experimental Setup

Statistics Collection

➢Collect the statistics scalars of IO Size, IO Count and Latency for four access activities: random read, random write, sequential read, and sequential write
➢Collect the scalars every x minutes
➢Collected the scalars at two levels of granularity: Extent and LUN

Core Feature

➢Identify hotspots for each time unit by clustering addresses based on weighted attributes
➢Identify average lifetime of hotspots based on identifying similarity between time units
➢Use the average lifetime and other parameters to select decision window for workloads

Existing Approach

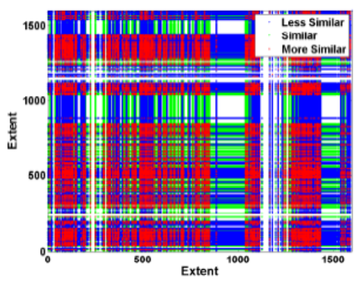
➢Static decision window selection
➢Non data mining method
➢Straightforward and easy to implement
➢Sub-optimal system performance or sub-optimal utilization

Results

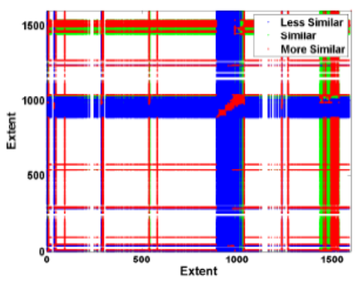
➢Observation 1: similar behavior is exhibited both within and across volumes
➢Observation 2: spatial similarity varies by the dimension under consideration

➢Observation 3: data expresses stronger similarity under the random read access pattern
➢Observation 4: the strongest access pattern dominates the similarity metric

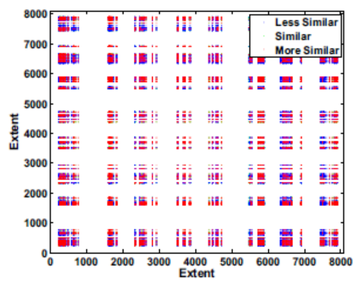
➢Observation 3: data expresses stronger similarity under the random read access pattern
➢Observation 4: the strongest access pattern dominates the similarity metric



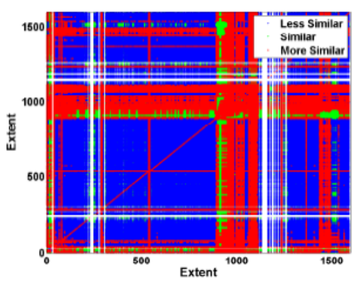
(a) Random Read



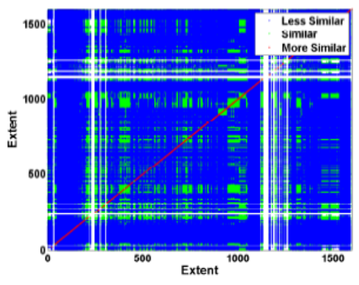
(b) Random Write



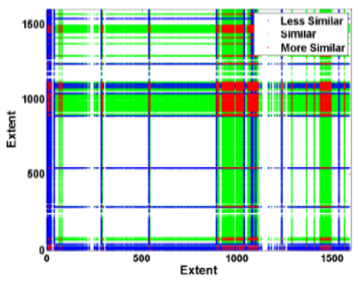
(a) Bank Trace



(b) Email Trace

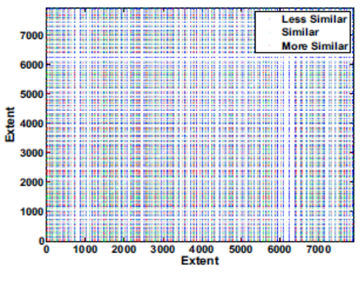


(c) Sequential Read



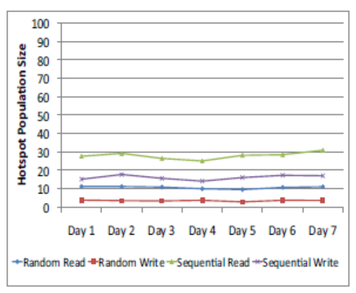
(d) Sequential Write

Extent Similarity on Email Trace

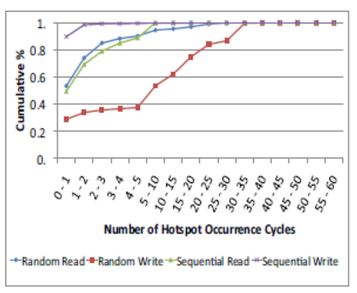


(c) Store Trace

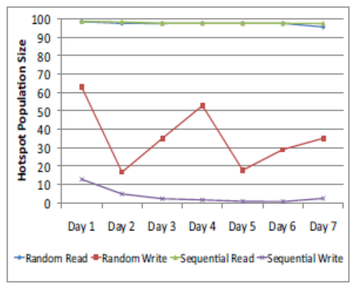
Unified Extent Similarity on Different Traces



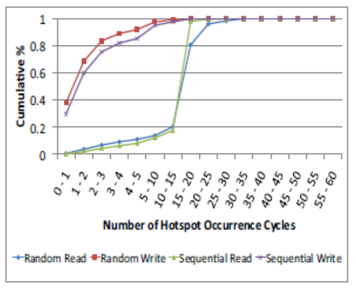
(a) Bank Trace: Population Size



(b) Bank Trace: Burstiness



(a) Store Trace: Population Size



(b) Store Trace: Burstiness

This work is partially sponsored by NSF under Grants IIS-0905493, CNS-1115375, IIP-1230740 and a grant from Intel ISTC on Cloud Computing

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley · UNIVERSITY of WASHINGTON