THE UNIVERSITY of TENNESSEE KNOXVILLE

SSD-Optimized Workload Placement with Adaptive Learning and Classification in HPC Environments

Lipeng Wan, Zheng Lu, Qing Cao

lwan1@utk.edu, zlu12@utk.edu, cao@utk.edu

Feiyi Wang, Sarp Oral, Bradley Settlemyer

fwang2@ornl.gov, oralhs@ornl.gov, settlemyerbw@ornl.gov

Outline

- Introduction
- System Design
- Data Popularity Prediction
- Data Placement Model
- Evaluation
- Conclusions
- Future Work

Introduction: Challenges

- Challenges of developing resilient and efficient storage system for HPC applications
  - Massive data is being generated
  - I/O workload evolves overtime
  - Flash-based storage devices might be used in HPC environments

Introduction: Scenario

- Scenario we focus on
  - All-flash storage solution is expensive for HPC environments, not feasible in practice
  - One possible way is designing storage systems consist of both SSDs & HDDs, in which only small portion of storage devices are SSDs

Introduction: Existing Work

- Some existing work on data placement algorithms and heterogeneous storage system design in HPC environments

Data placement algorithms	Heterogeneous storage system design
Distributed Hash Tables (DHTs), chain placement, RUSH, CRUSH, etc.	Flashcache, iTransformer, SieveStore, Hystor, I_CASH, ComboDrive, etc.

- Effective data placement algorithms designed for heterogeneous storage system are still needed

Introduction: Our Contributions

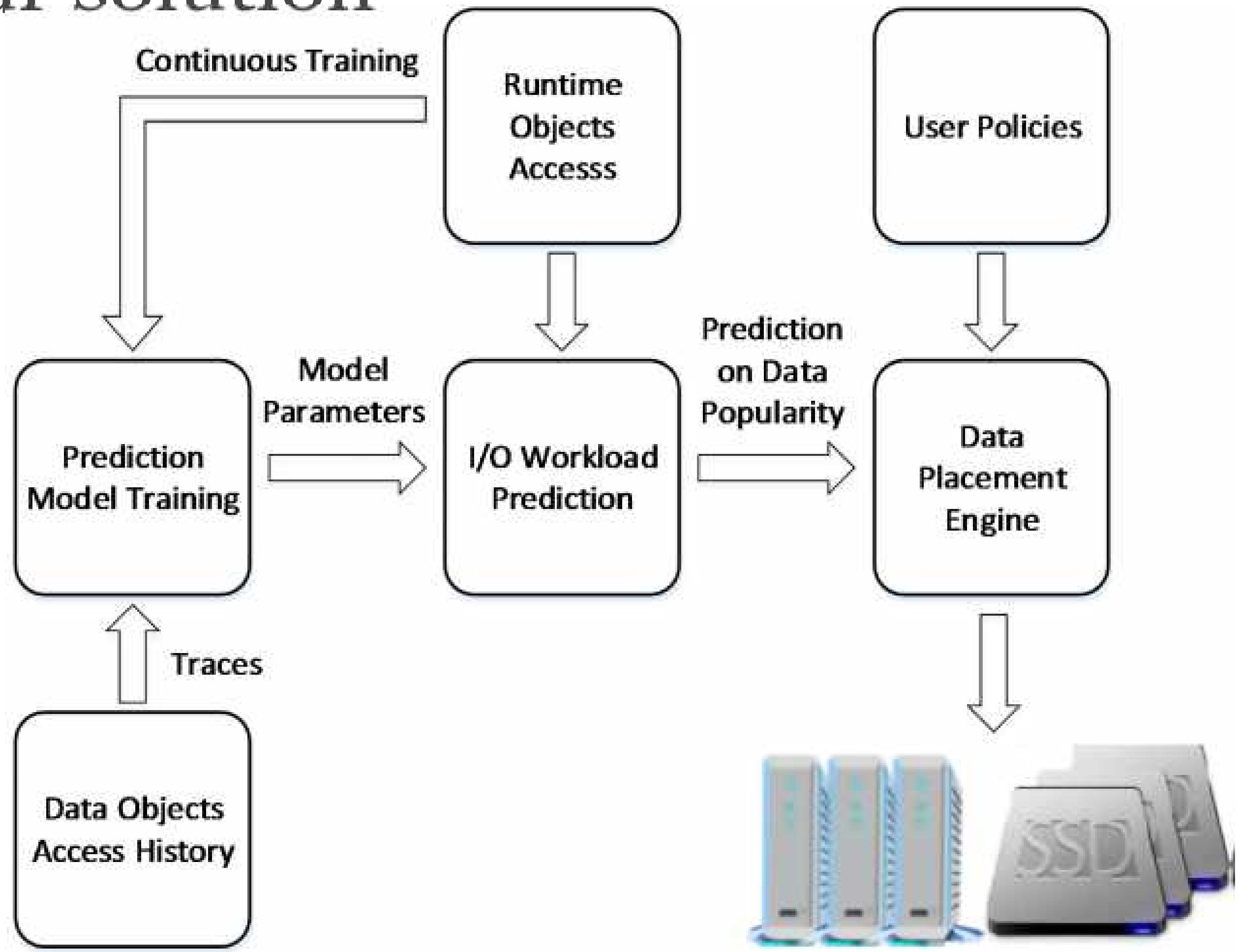
- Two major contributions of our work:
  - Designed an adaptive learning algorithm to capture the dynamics of data access in HPC environments which can be exploited by data placement algorithms
  - Proposed an optimization model for data placement among heterogeneous storage devices without violating the user policies, such as replication schemes

System Design: Intuition

- Intuition of improving heterogeneous storage systems' performance:
  - Put data objects that will be most frequently accessed in the future on SSDs
- But how?
  - How to predict future access frequency of data objects?
  - How to place data objects among heterogeneous storage devices with putting the user policies into consideration?

System Design: Solution

- Our solution



Data Popularity Prediction: Requirement

- Most of existing work tried to predict the data popularity by simply counting the access times of each data block
- We think more complex model should be built so that the dynamics of data access can be captured more accurately

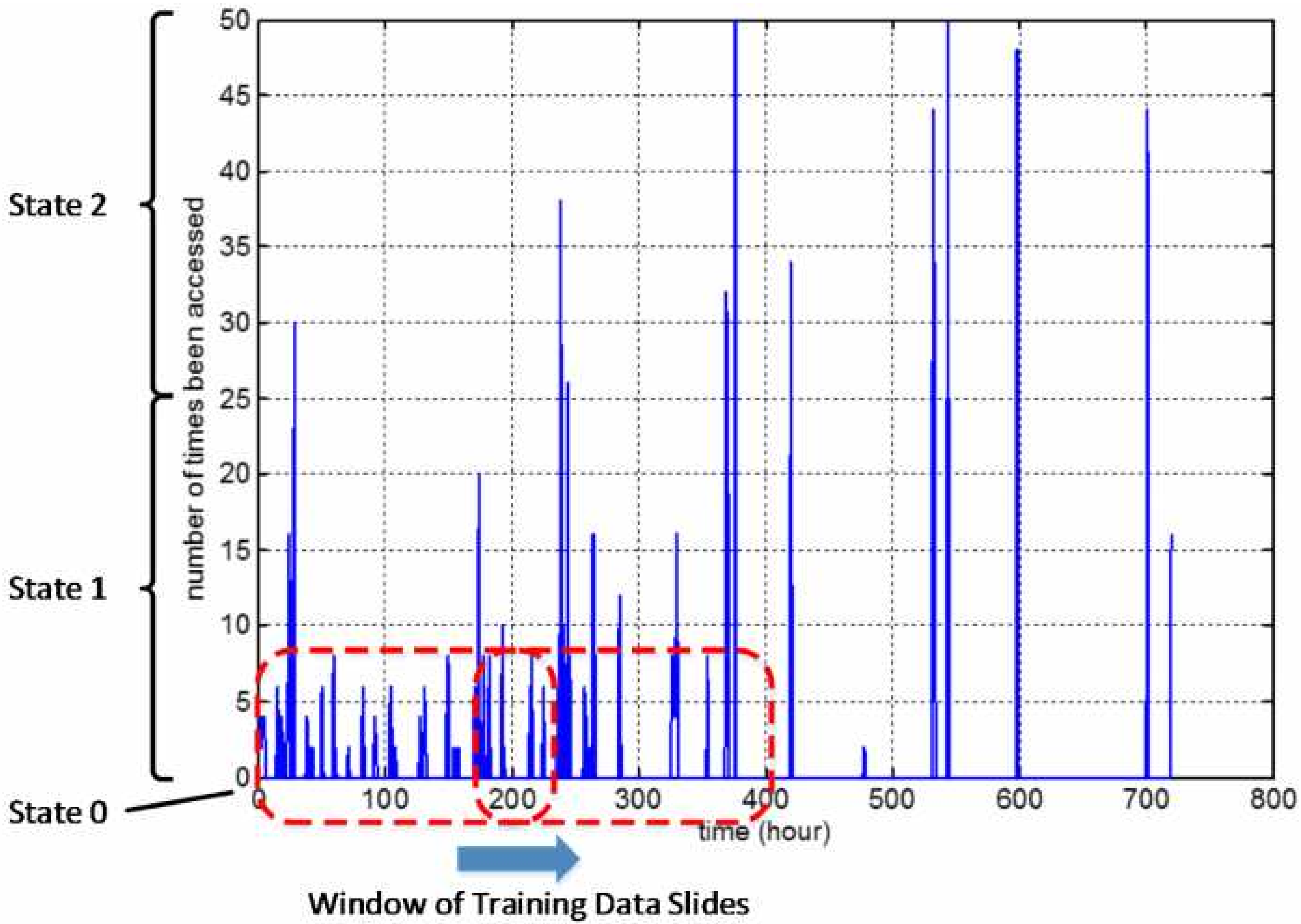
Data Popularity Prediction: Solution

- Our solution: Markov chain based prediction
  - Inspired by existing work on network traffic prediction
  - Can capture dynamics of streaming data

Data Popularity Prediction: Model Training

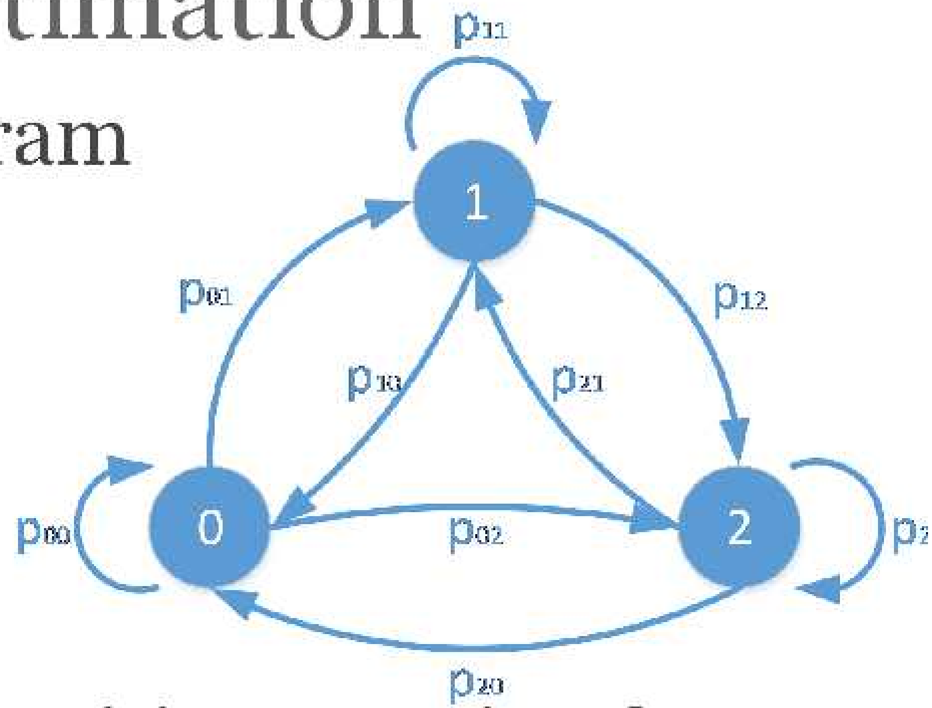
- Training data
  - Only recent access history of each data object is maintained and updated periodically
  - A tradeoff between data collection overhead and prediction accuracy

Data Popularity Prediction: Training Data Collection



Data Popularity Prediction: Parameter Estimation

- Transition diagram



- Estimate the transition matrix of Markov chain using the access history data

Data Popularity Prediction: Prediction

- Calculate the stationary distribution of Markov chain
- Rank the data objects
  - Use a weighted sum of stationary distribution to rank future popularity of data objects

Data Placement Model: Framework

- Linear programming model
  - Optimization objective function and constrains

$$maximize \sum_{i \in M} f_i \times \max[\forall j \in N, at_j \times e_{ij}]$$

$$subject\ to \sum_{j \in N} e_{ij} = cp_i, \forall i \in M$$

$$\sum_{\forall i\ s.t.\ e_{ij}=1} ds_i \leq cs_j, \forall j \in N$$

……

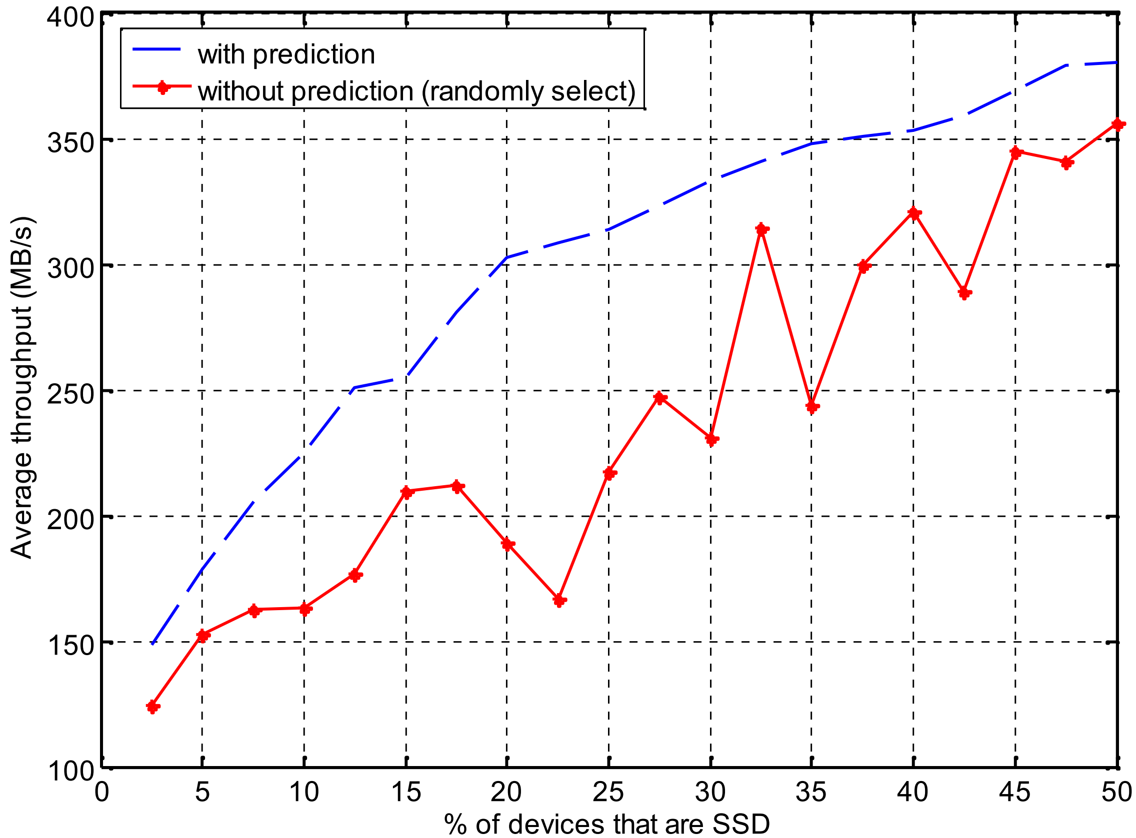
more constrains from user policies

Evaluation: Realistic I/O Traces

- Evaluation is based on LASR traces
  - Long-term I/O traces collected at system-call level

Evaluation: Average Read Throughput Results

- Average read throughput comparison between our approach and baseline



Conclusions

- Designed an adaptive learning algorithm to predict future access frequency of data objects
- Proposed an optimization model for data placement among heterogeneous storage devices with putting the user policies into consideration

Future Work

- Better model to capture the dynamics of I/O workload pattern
  – Predict not only access frequency, but also access pattern, such as random/sequential read/write, etc.
- Better model to optimize data placement
  – Optimize not only the average throughput, but also the reliability

Questions?

Thank you!