Self-Learning Disk Scheduling

Yu Zhang and Bharat Bhargava
Fellow, IEEE

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING
VOL. 21, NO. 1, JANUARY 2009

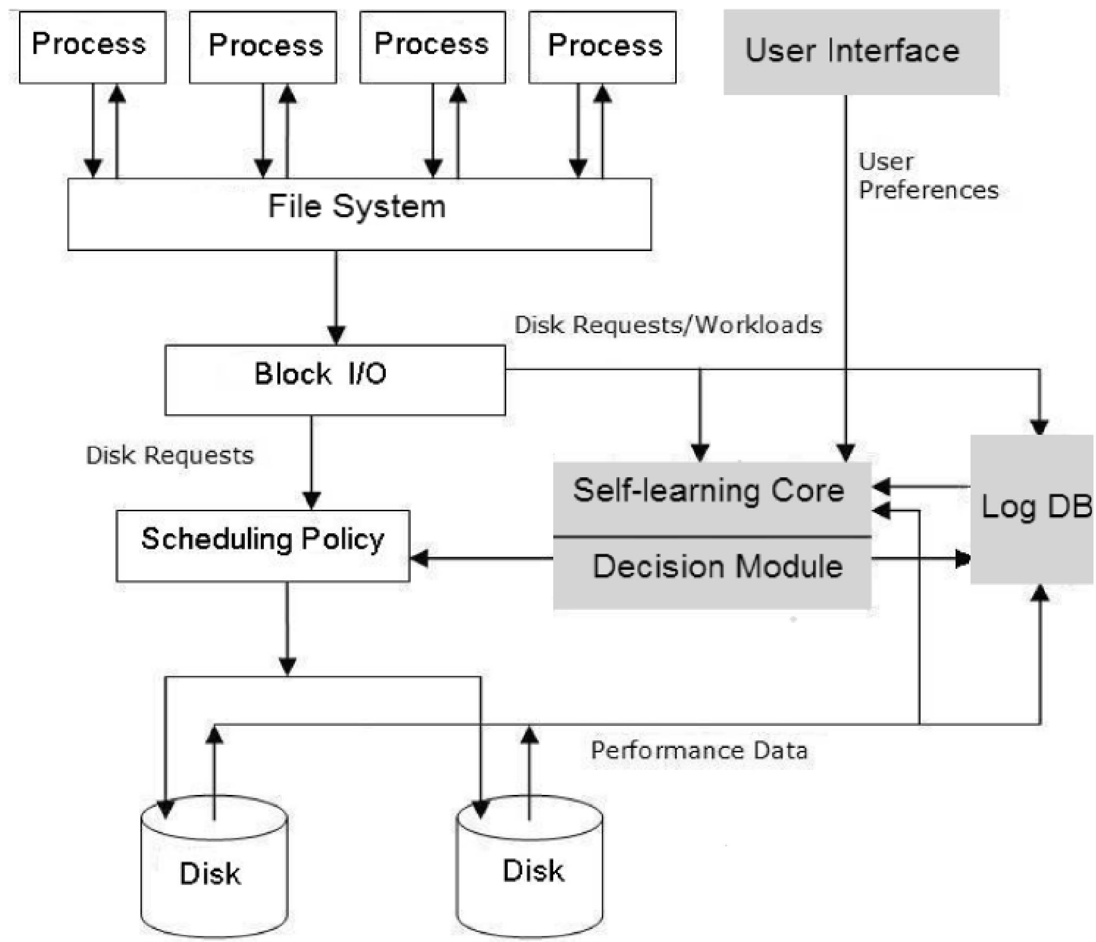
Outline

- Introduction
- Architecture
- Candidate Self-Learning Core Algorithms
- Potential Machine Learning Algorithms
- Experiments for Identifying Self-Learning Parameters
- Experiments on Simulated Scenarios
- Conclusions

Introduction

- There is no single disk scheduler that could provide good performance consistently under varying environment
  - Be affected by workload, file system, disk system…
- Therefore, a self-learning schedulers are proposed which can adapt to various types of workloads, and make optimal scheduling decisions

Architecture



Candidate Self-Learning Core Algorithms

- Four algorithms
  - 1. Change-Sensing Round-Robin Selection
  - 2. Feedback Learning
  - 3. Per-Request Disk I/O Scheduler
  - 4. Two-Layer Combined Learning Scheduler

1. Change-Sensing Round-Robin Selection (1/2)

- The self-learning core in the operating system invokes all schedulers in a round-robin fashion
- Phases
  – Phase 1: Selection phase
  – Phase 2: Execution phase

1. Change-Sensing Round-Robin Selection (2/2)

```
For(;;) { // repeat infinitely
   For(each i(S) out of m(S) disk I/O schedulers) {
      Execute(i(S));
      Log(ResponseTime, Throughput);
   }
   NS = Max(i(S) in m(S) schedulers, Pref);
   If(NS != CS) { // Phase 1: Selection phase
      CS = NS;
      Load(CS);
   }
   While(!(WorkloadChange || BadPerformance)) {
      Wait(T_select); // Phase 2: Execution phase
   }
}
```

- **i(S)**
  - Individual scheduler
- **m(S)**
  - The number of available schedulers
- **NS**
  - The selected scheduler for next round
- **CS**
  - The current scheduler
- **Pref**
  - The preference and can be set by users via User interface

Minimize the cost by switching from phase 2 to phase 1 only under the following conditions

- When a significant change of the workload is detected
- When a significantly deteriorated system performance is observed

2. Feedback Learning (1/4)

- The round-robin execution and switching are moved offline
- Phase
  - Phase 1: Training phase
  - Phase 2: Decision phase
  - Phase 3: Feedback phase

2. Feedback Learning (2/4)

- Logged Features for Workloads
  - Number for reads, number of writes, and read/write ratio
  - Average request size
  - Sequential/random ratio
  - Average request arrival rate
  - Average number of processes
  - Average think time

2. Feedback Learning (3/4)

(Training phase)

```
For(each i(S) out of m(S) disk I/O schedulers) {
    Training(i(S), DiskIOIntensiveApp);
    Training(i(S), SyntheticWorkload)
    Log(ResponseTime, Throughput);
}
Model = Run_LearningAlgorithm();
```

- i(S)
  - Individual scheduler
- m(S)
  - The number of available schedulers
- Model
  - The learning model generated by the learning algorithm

2. Feedback Learning (4/4)

(Decision/Feedback Phase)

```
Initialize(TotalRequest, NULL);
For(;;) { //repeat infinitely
   While(Size(CollectedRequest) <= X) {
      Collect(incoming request);
   }
   NS = Model(Workload);
   If(NS != CS) {
      CS = NS;
      Load(CS);
   }
   Log(ResponseTime, Throughput);
   Append(TotalRequest, CollectedRequest);
   If(Size(TotalRequest) mod Y == 0){
      Model = Run_LearningAlgorithm();
   }
   Clear(CollectedRequest);
}
```

- CollectedRequest
  - The incoming requests collected by the algorithm
- TotalRequest
  - The number of all processed requests, which is used to invoke the periodic update of the learning model
- X
  - The predetermined value used to perform request-sensing decision(default value 3,000)
- Y
  - How frequently we update the learning model (default value 1,000,000)
- NS
  - The selected scheduler for the next round
- CS
  - The current scheduler
- Model
  - The learning and decision model that is generated in the Training Phase

3. Per-Request Disk I/O Scheduler (1/4)

- The self-learning scheduler makes scheduling decisions at the request level instead of the workload level
- No longer log or compare the performance of the existing scheduling policies
- Phase
  - Phase 1: Training phase
  - Phase 2: Decision phase
  - Phase 3: Feedback phase

3. Per-Request Disk I/O Scheduler (2/4)

- Logged Features for Requests
  - Types of current and x previous requests
  - Individual request size
  - Sequential or random
  - Arrival times of current and x previous requests
  - Number of processes (issuing requests)
  - Think time for each request
  - Inter-request block number distances between current request and x previous requests
  - Logical block number of each request

3. Per-Request Disk I/O Scheduler (3/4)

- **Phase 1: Training phase**
  - There are two methods to jump-start the self-learning scheduler
    - Pick a disk scheduler, such as Anticipatory, and feed the system with different types of requests to collect response time data
    - Train the system with sophisticated workloads and build the response time estimation model
      - Issue requests with different combinations of features and gather response time data to train the system
  - The two methods can be used together

3. Per-Request Disk I/O Scheduler (4/4)

(Decision/Feedback Phase)

```
Initialize(TotalRequest, NULL);
For(;;) { // repeat infinitely
  For(each i(R)) {
    EstimateResponseTime = ResponseTimeModel(i(R));
    Insert(SchedulerQueue, i(R), ResponseTimeEstimate);
    NR = Head(SchedulerQueue);
    Schedule(NR);
    Log(ResponseTime, Throughput);
    Append(TotalRequest, NR);
    If(Size(TotalRequest) mod Y == 0){
      ReseponeTimeModel = Run_LearningAlgorithm();
    }
  }
}
```

- i(R)
  - Individual request
- EstimateResponseTime
  - The estimated response time for each request based on the classification model
- SchedulerQueue
  - The queue the per-request scheduler uses to rank the requests
- NR
  - The next request to be scheduled
- TotalRequest
  - The number of all requests processed, which is used to invoke the periodic update of the learning model
- Y
  - How frequently we update the learning model (default value 1,000,000).
- SchedulerQueue
  - is sorted and the head request in queue has the shortest estimated response time

4. Tow-Layer Combined Learning Scheduler (1/2)

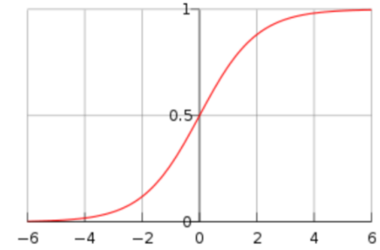
- Incorporates algorithm 2 and 3 into a two-layer self-learning scheduling scheme
- Phase
  - Phase 1: Training phase
  - Phase 2: Decision phase
  - Phase 3: Feedback phase

4. Tow-Layer Combined Learning Scheduler (2/2)

- **Phase 1: Training phase**
  - Step
    - 1. Train the per-request decision scheduler by the methods in Algorithm 3
    - 2. Train the scheduling scheme that consists of traditional schedulers plus per-request decision scheduler by the training procedures for Algorithm 2
- **Phase 2: Decision phase / Phase 3: Feedback phase**
  - Remain mostly unchanged, except that the per-request decision scheduler becomes one of the possible schedulers

Potential Machine Learning Algorithms (1/2)

- C4.5 decision tree algorithm
  - C4.5 generates a decision tree, which is a classifier in the form of a tree structure, based on the ID3 algorithm
- Logistic regression
  - Logistic regression is a regression method for Bernoulli-distributed dependent variables that utilizes a logistic function as the link function
  - $f(z) = \frac{1}{1+e^{-z}}$, $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k$



Potential Machine Learning Algorithms (2/2)

- Naïve Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

  - The Naïve Bayes classifier applies Bayes' theorem with Naïve independence assumptions
- Neural networks
  - The neural network (NN) is an adaptive system that adapts itself based on external or internal information that travels through the network
- SVM(Support Vector Machine)
  - The SVM algorithm maps input vectors to a higher dimensional space

Experiments for Identifying Self-Learning Parameters

- Experiment Setup
- Comparison for Training Schemes
- Comparison for Learning Level
- Comparison for Learning Algorithms
- Comparison for Window Sizes

Experiment Setup

Real-World Training Workloads

Workload	Description
Sequential Reading	Reading sequentially 30 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB
Concurrent Reading	Reading concurrently 30 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB
Sequential Writing	Writing sequentially 30 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB in sequence
Concurrent Writing	Writing concurrently 30 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB
Linux Compilation	Compiling Linux kernel
HTTP Server	Running Apache HTTP server benchmark tool [52]
Multimedia Server	Concurrent streaming of 50 video files of 2GB each
Concurrent Access	Playback of a 2GB video file in the fast forward mode and concurrent copying of 10 files of 1GB each

Real-World Test Workloads

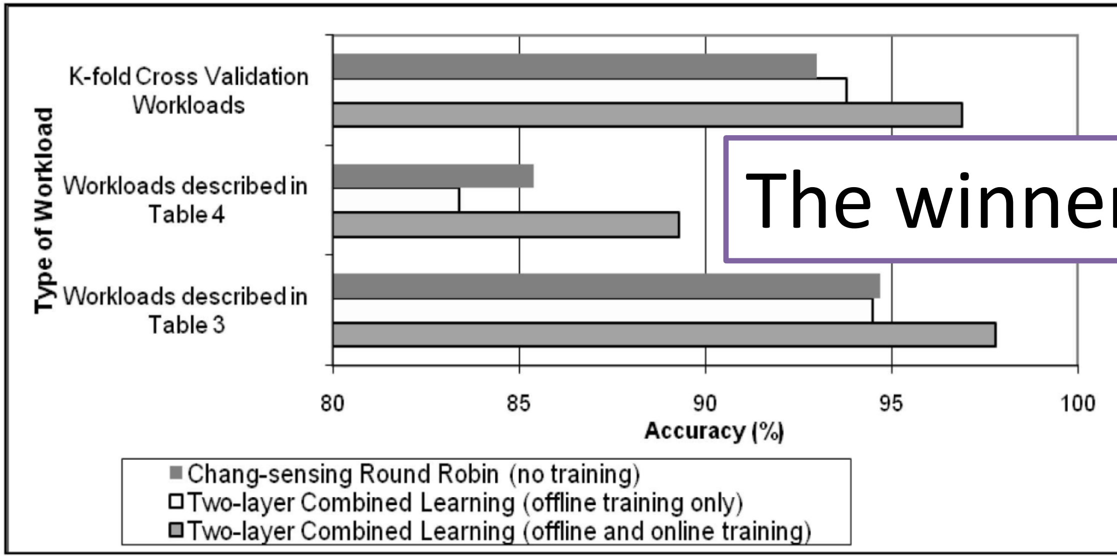
Workload	Description
File Reading	Reading 20 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB in sequence and other 20 files concurrently
File Writing	Writing 20 files of size 1KB, 2KB, 4KB, 64KB, 128KB, 512KB, 1MB, 32MB, 128MB, 512MB in sequence and other 20 files concurrently
Random SQL database queries	MySQL benchmark tool [51]
Mixed Access Pattern 1	Playback of a 2GB video file in the fast forward mode, copying 15 files of size 2GB, and running the MySQL benchmark
Mixed Access Pattern 2	Reading 30 files of 2GB each, followed by a playback of 2GB video file in the fast forward mode, and running the MySQL benchmark

Comparison for Training Schemes (1/2)

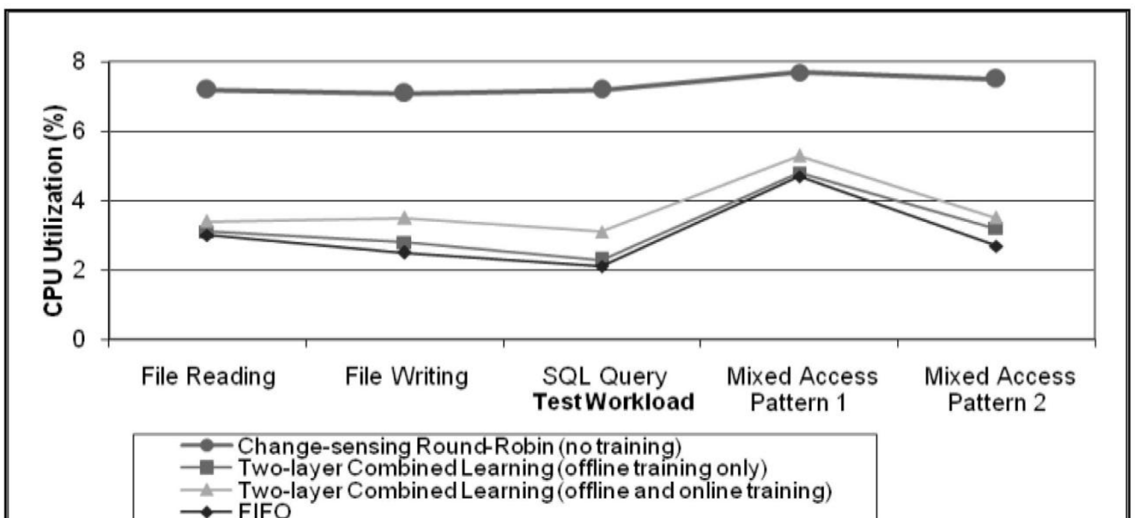
- **Training Schemes: ALL**
- Learning Level: ALL
- Learning Algorithms: SVM
- Window Sizes: 100s

- $\text{Accuracy} = \dfrac{\text{number of correct decisions}}{\text{number of all decisions}}$

Comparison for Training Schemes (2/2)



The winner is TCLOO

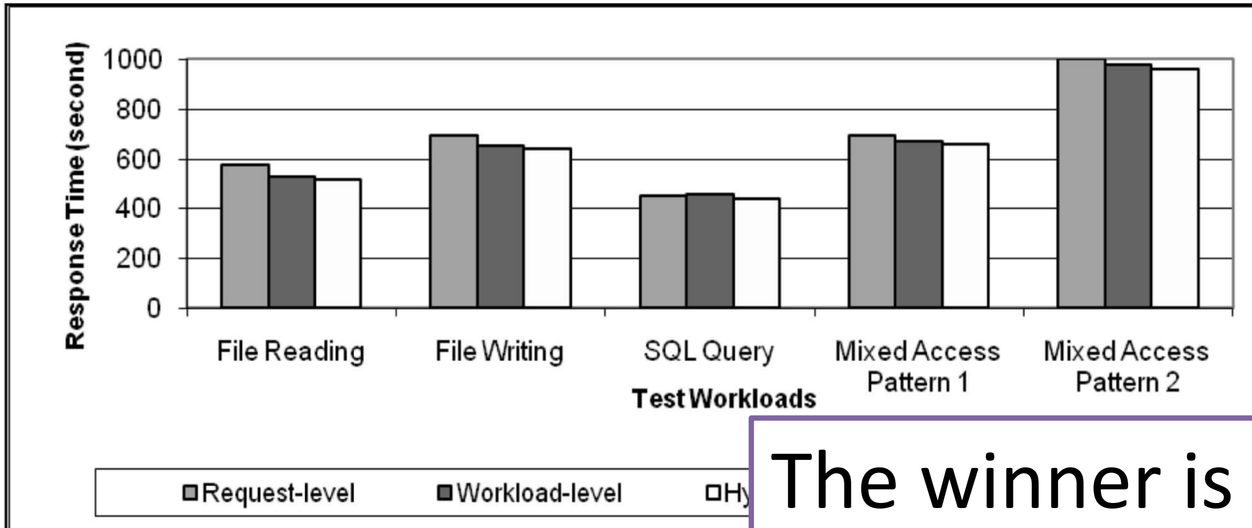


Comparison for Learning Level (1/2)

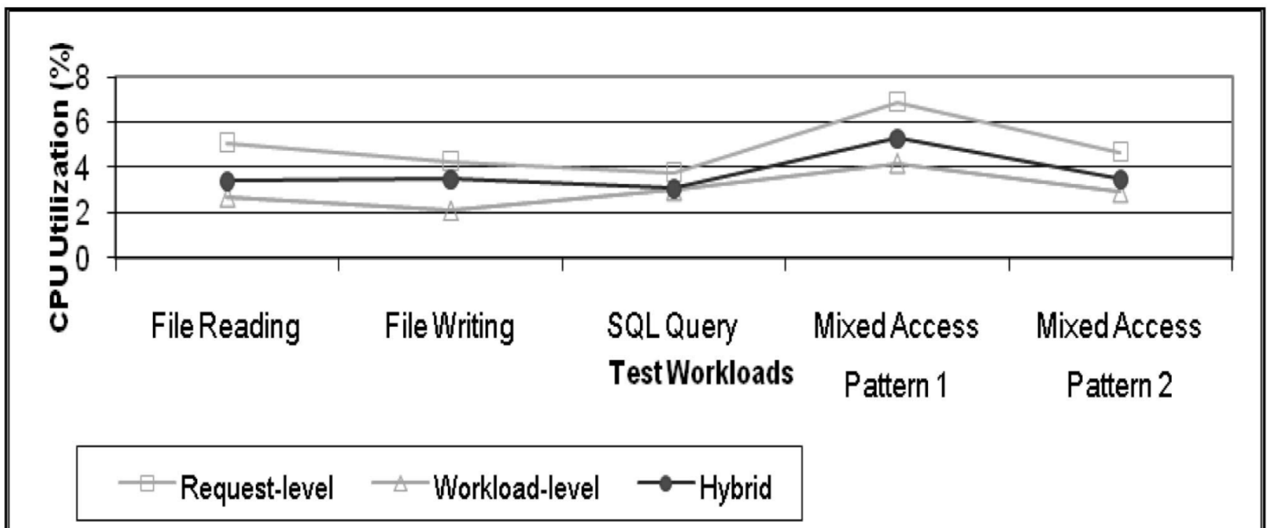
- Training Schemes: TCLOO
- **Learning Level: ALL**
- Learning Algorithms: Logistic regression
- Window Sizes: 100s

- The request-level I/O scheduler gets selected is approximately 15.6 percent in the hybrid learning algorithm

Comparison for Learning Level (2/2)



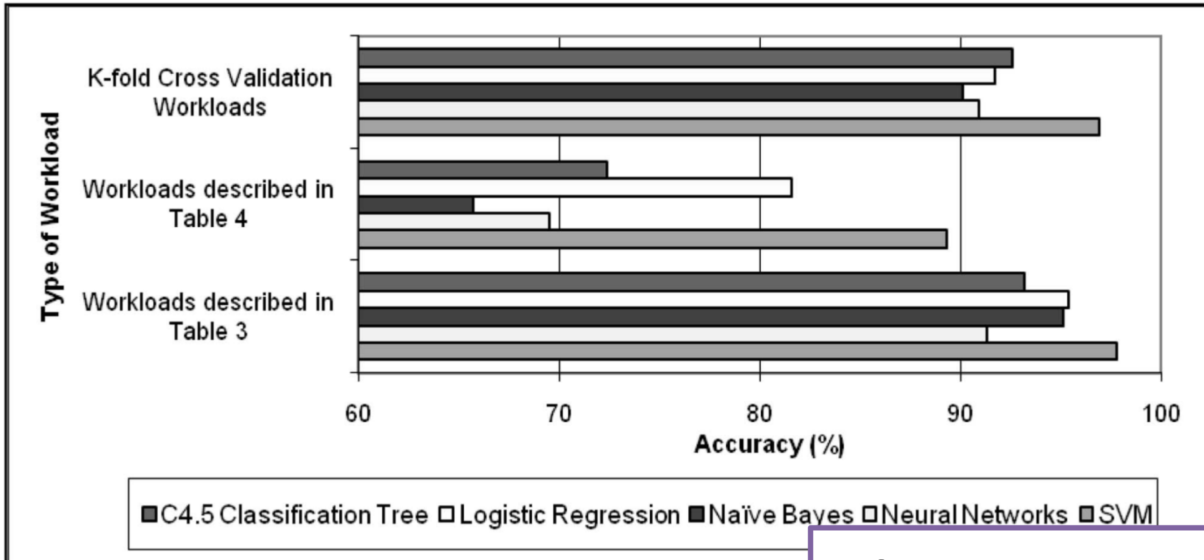
The winner is Hybrid

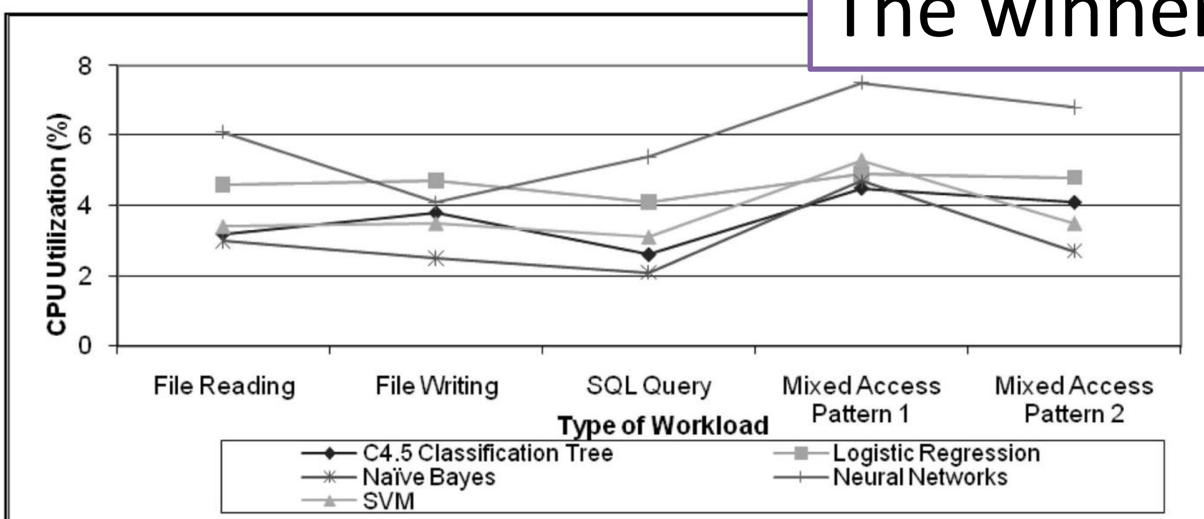


Comparison for Learning Algorithms (1/2)

- Training Schemes: TCLOO
- Learning Level: Hybrid learning scheme
- **Learning Algorithms: ALL**
- Window Sizes: 100s

Comparison for Learning Algorithms (2/2)





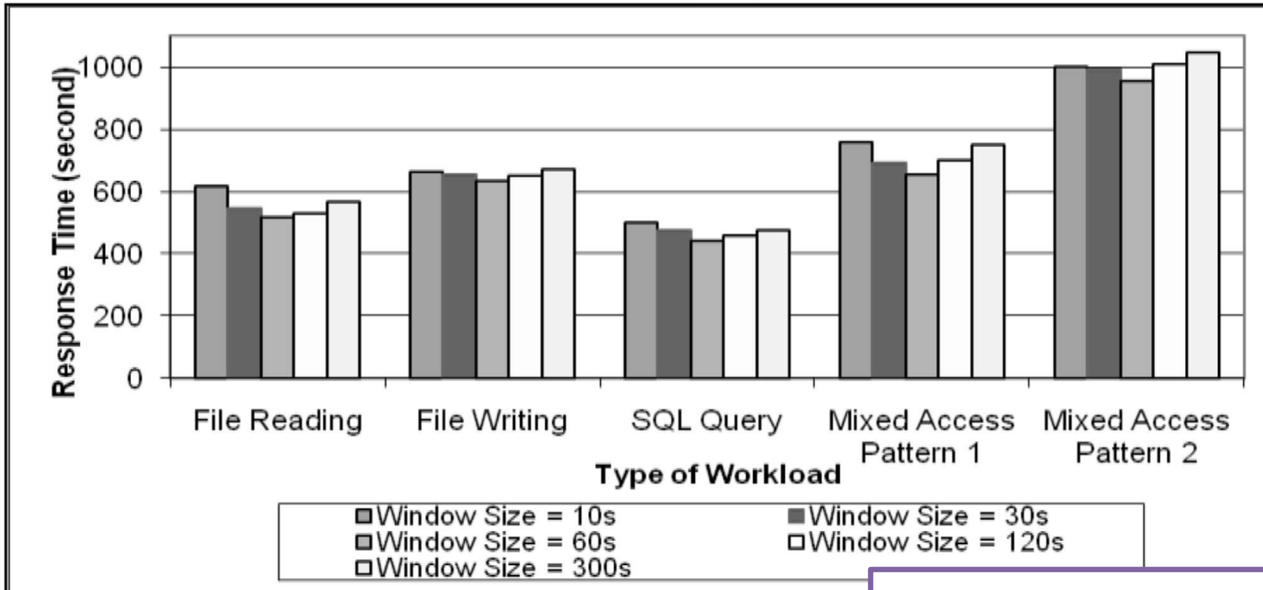
The winner is SVM

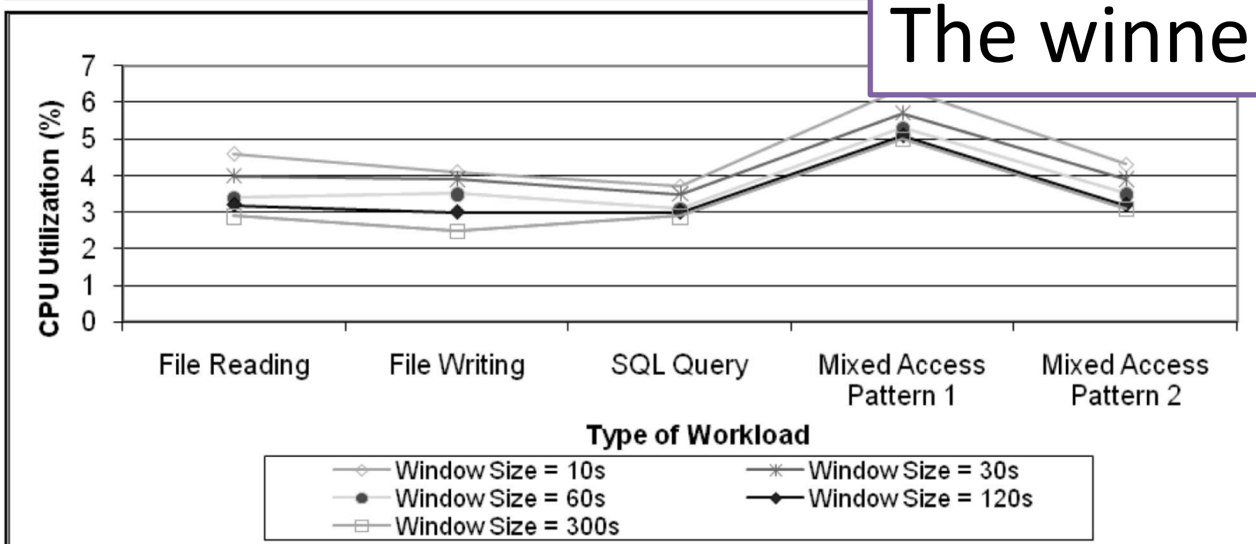
Comparison for Window Sizes (1/2)

- Training Schemes: TCLOO
- Learning Level: Hybrid learning scheme
- Learning Algorithms: SVM
- **Window Sizes: ALL**

- SW(Scheduling window)
  - An SW is a window that contains a subset of disk I/O requests
  - The range of the SW is determined by the left window boundary (LWB) time and the right window boundary (RWB) time
- SWZ(SW Size)
  - SWZ=RWB-LWB
- Need only to identify a suitable value for the “window size” at workload level

Comparison for Window Sizes (2/2)





The winner is 60s

Experiments on Simulated Scenarios

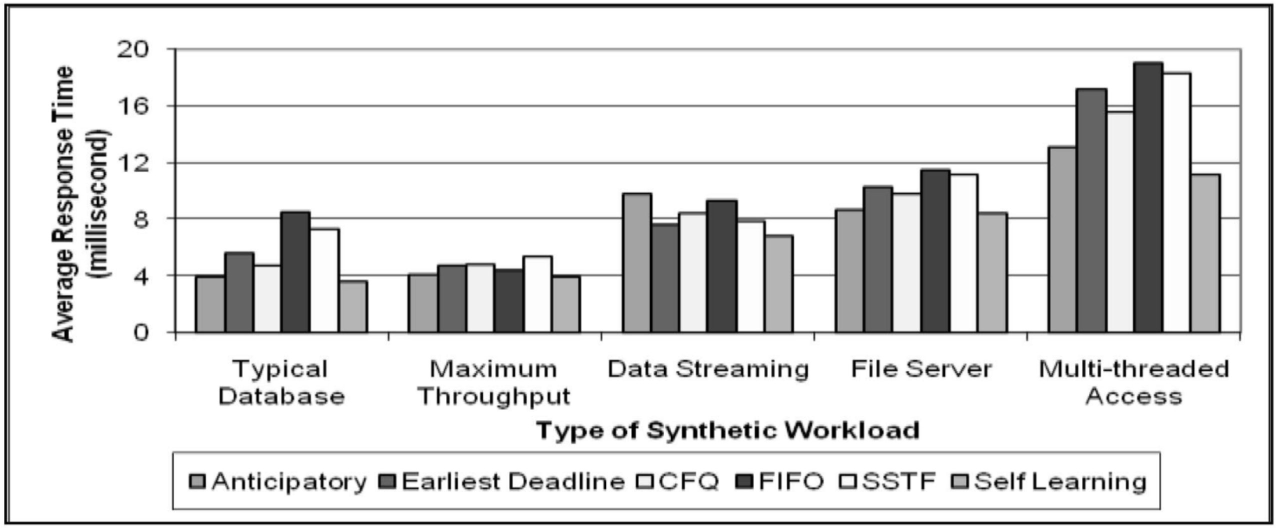
- Implementation Details
- Results

Implementation Details

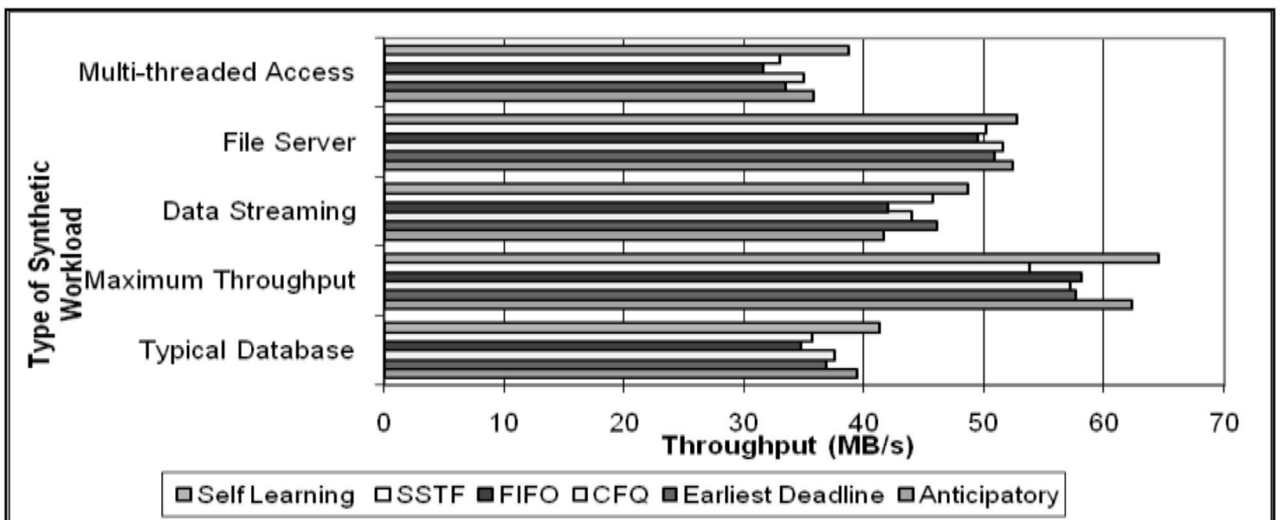
– Use the SVM learning algorithm
– Use the TCLOO scheduling scheme and the hybrid learning algorithm
– Set the window size to 60 seconds

Synthetic Workload	Description
Typical Database	2KB random I/Os with a mix of 67% reads and 33% writes. 8 outstanding I/Os per target.
Maximum Throughput	512KB Transfer Request Size, 100% Read and 100% sequential. 16 outstanding I/Os per target.
Data Streaming	512KB Transfer Request Size, 30% sequential, and 50% read/write distribution. 32 outstanding I/Os per target.
File Server	100% random, 80% read, 60% 4KB blocks with the remainder spread from 512KB to 64KB. 32 outstanding I/Os per target.
Multi-threaded Access	100% random, 50% read/write distribution, transfer request size 4KB. 256 outstanding I/Os per target.

Results



Average Response Time (millisecond)
20
16
12
8
4
0
Typical Database
Maximum Throughput
Data Streaming
File Server
Multi-threaded Access
Type of Synthetic Workload
Anticipatory
Earliest Deadline
CFQ
FIFO
SSTF
Self Learning



Type of Synthetic Workload
Multi-threaded Access
File Server
Data Streaming
Maximum Throughput
Typical Database
0
10
20
30
40
50
60
70
Throughput (MB/s)
Self Learning
SSTF
FIFO
CFQ
Earliest Deadline
Anticipatory

Conclusions

- The self-learning disk scheduling schemes can learn about the storage system, train themselves automatically, adapt to various types of workloads, and make optimal scheduling decisions
- Experiments show that self-learning disk schedulers outperform existing disk schedulers and achieve the best system performance without human intervention