Interleaving Widely Different Storage Workloads with Performance Guarantees

Alma Riska
Department of Computer Science
College of William and Mary

Evgenia Smirni
Department of Computer Science
College of William and Mary

Storage clusters

- Increasingly more complex
  - Large number of (heterogeneous) devices
  - Vast set of features to manage/run the cluster
    - Performance
    - Reliability/availability
    - Power
- Workload interleaving is very common
  - Often with different priorities
- Key to an efficiently-run cluster
  - Management of resources
  - Scheduling and shaping of workloads to achieve certain QoS targets

Scheduling workload interleaving

- Two workloads with demands that differ by several orders of magnitude
  - High priority with small demands
  - Low priority with large demands
- Workload scheduling to meet QoS targets
- Low priority is scheduled during idle times of high priority work
- Change high priority workload / idle intervals to achieve certain QoS targets
  - E.g for amount of completed low priority work

Scheduling workload interleaving

- For a given pair of workloads to interleave
  - Learn characteristics of high priority idle intervals
    - In the form of histograms that change over time
  - Use that histogram to determine scheduling of low priority work such that QoS are met
    - Either for high priority workload
    - Or low priority workload
- Change workload characteristics (idle intervals in this case) if the interleaving is unsatisfactory
    - Low priority is almost starved

Aspects of workload interleaving

- Estimation of scheduling parameters for one pair of high/low priority workloads should be
  - computationally efficient
  - accurate
- Such estimations are done multiple times for different pairs of high/low priority workloads resulting from shaping of workloads
  - Determine the most efficient scheduling of workloads based on the system QoS

Accuracy of a schedule estimation

- High priority workload would be affected from low priority workload
  - Penalty of running low priority work
- Because of orders of magnitude difference in demands
  - Delays caused propagate from down multiple high priority busy periods
- Our framework accounts exactly for such propagation
  - Accurate estimation of the impact of low priority workload on high priority workload

Effect of delay propagation

- Probabilistic reasoning based on the histogram of idle time availability
  - Length of consecutive idle intervals
  - Absorption the delay caused
- Result: closed-form recursion on delay values
- Accuracy and computation efficiency dependent on
  - Histogram granularity
  - Granularity of possible delay values

Example – Power Consumption

- Power saving seen as a low priority workload with high demands
- Penalty to bring disk out of power saving mode is penalty used in framework
- Framework determines
  - When to start a power saving mode once the disk becomes idle
  - How long a device should stay in a power saving mode
    - without violating performance constraints or
    - achieving some power saving target

Power Saving Modes in HDDs

Mode	Power Consumption	Penalty
Active Idle	60% of operational	0 ms
Head unloaded Full rotation speed	82% of active idle	500 ms
Head unloaded Reduced rotation speed	70% of active idle	1,000 ms
Head unloaded No rotation Electronics on	50% of active idle	8,000 ms
Disk fully spun down	5-10% of active idle	20,000 ms

- These numbers depend on density, number of platters,etc

Evaluation – Power Savings

- Disk drives are spun up/down to conserve power

Values depend on density, number of platters,etc

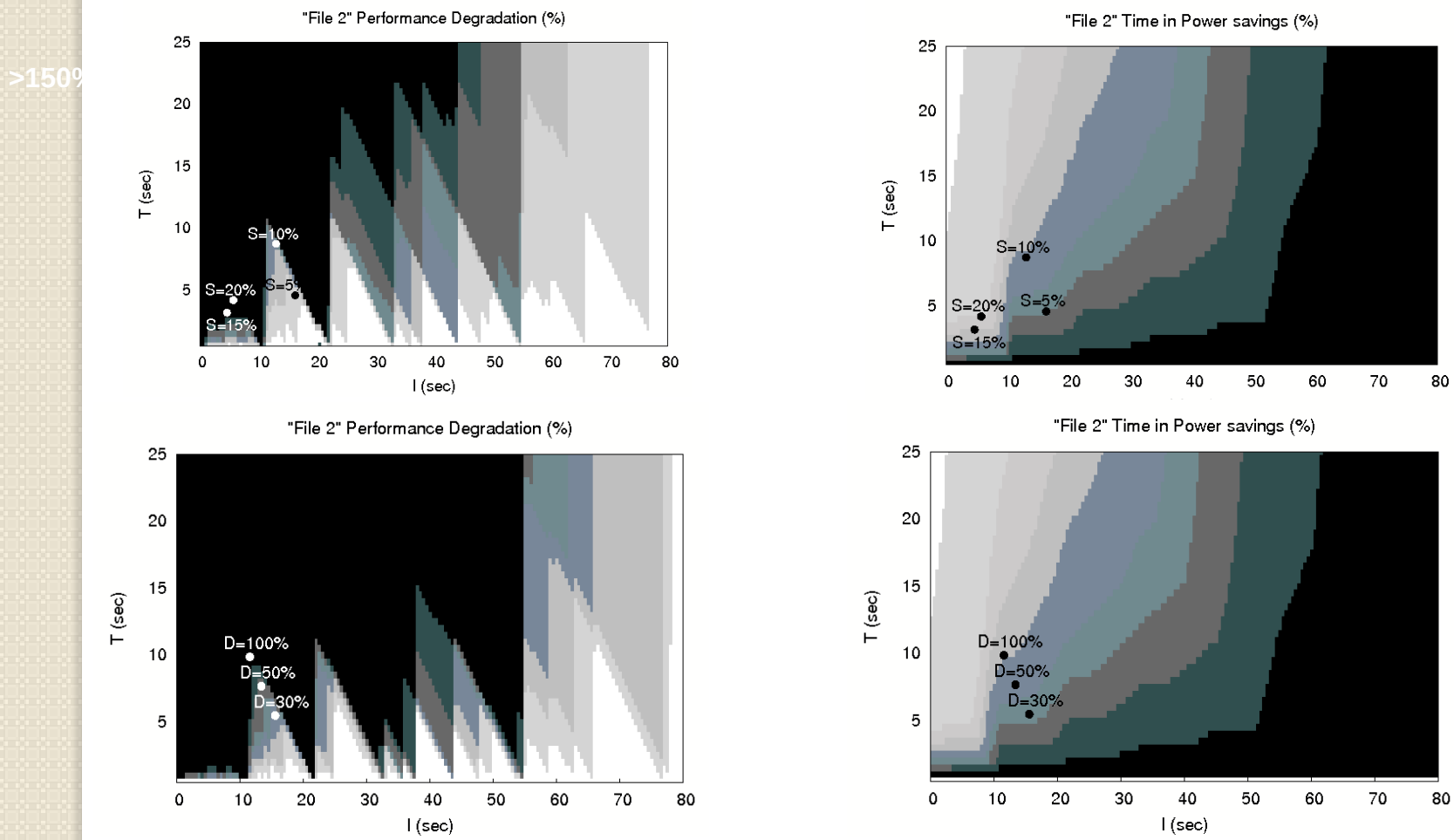
- Which power saving mode to use (multiple choices to estimate)
- Efficiency of workload shaping techniques
- Tradeoffs between performance degradation and power savings

Power savings

- Not straightforward problem
  - An HDD storage device has as many as 5 power saving modes
    - Different penalties and power saving capabilities
  - Workload shaping imposes extra work on the system
    - Which one to activate at a given time

- Our estimation engine allows to
  - Determine efficiency of each power savings modes workload shaping combinations
  - Choose the best one

A high-end HDD workload



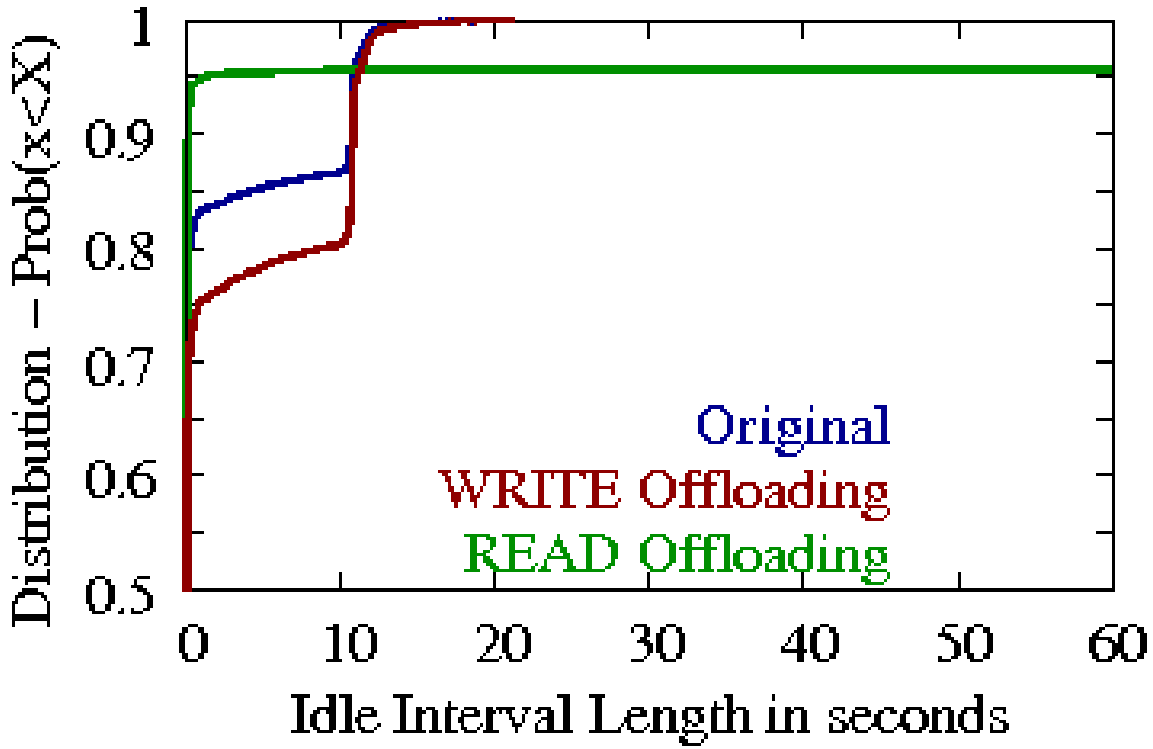
- Estimations for power saving mode – Level 1 and Level 2
- Exhaustive search for the best trade-off in performance/power
- Our solution – estimates only one point; positions accurately

Workload shaping

- WRITE offloading – non-learning
  - Move temporarily write traffic from one HDD to another
- READ offloading - learning
  - Identify a buffer size
  - Fill it with the most accessed READ locations
  - Move that buffer and the subsequent traffic to another HDD in the cluster
- Busy period offloading - learning
  - Frequency of entire busy periods
  - Probabilistically
    - Learn if consecutive idle intervals are correlated

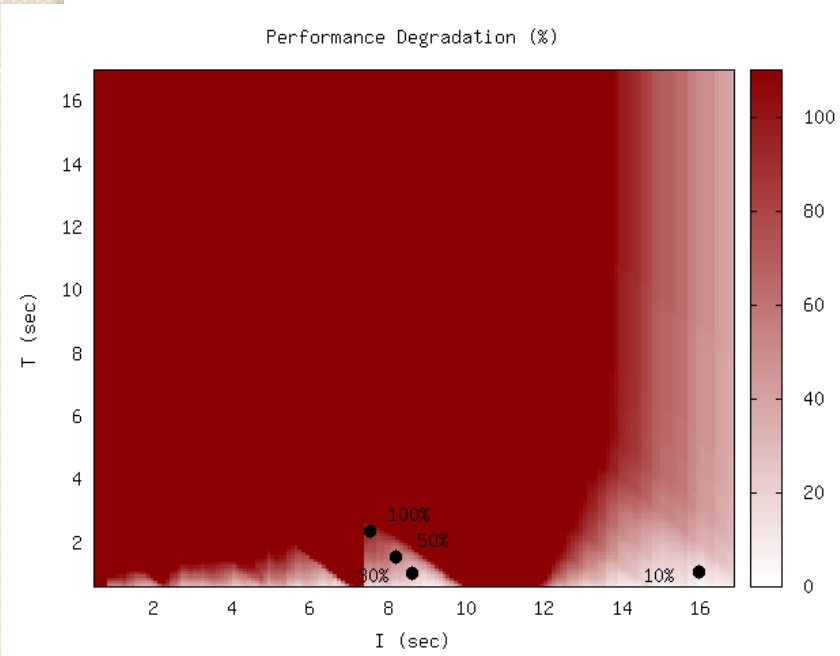
Enterprise HDD workload

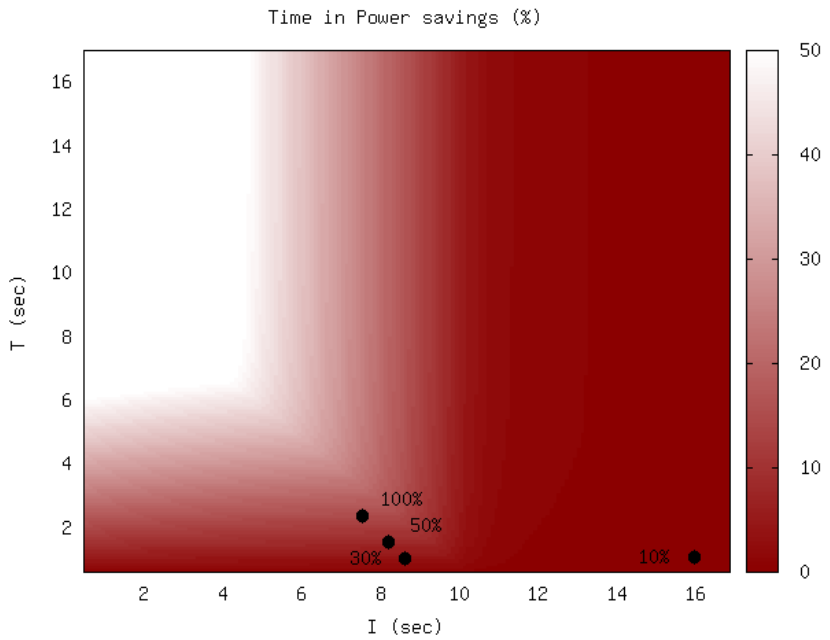
>150%



- From a monitored workload histogram of idleness is constructed for each workload shaping technique
  - Significant differences
  - Challenging to capture with simple metrics
- Our priority scheduling model is applied on each case

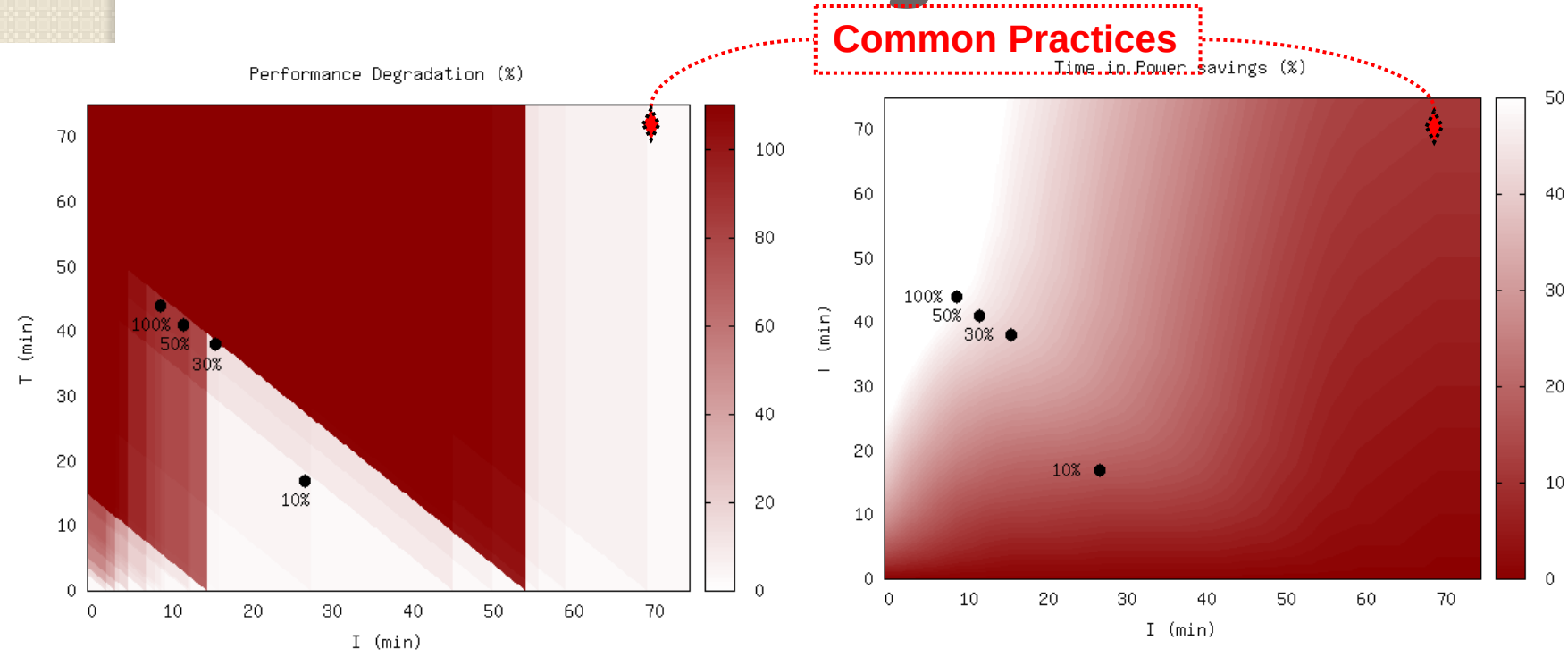
Original HDD workload





- Estimations for power saving mode – Level 1
- Exhaustive search for the best trade-off in performance/power
  - Infeasible to be conducted in practice
- Very limited power saving opportunities
- Our estimations are marked for several QoS targets
  - Identified regions with best potential

READ Offloading



- Estimations for power saving mode – Level 1
- Significantly more opportunities than original workload
  - Very sharp regions – incorrect decisions may be costly
- Our estimations are marked for several QoS targets

Future work

- Deploy workload shaping at the cluster level
  - Currently estimation deal with one node
  - Determine which node to offload work and which node to receive it
  - Challenge: the set of possibilities to be evaluated increases very fast
    - Ways to reduce calculations even more
- Utilize understanding of workload shaping to develop a similar estimation framework
  - High priority has high demands
  - Low priority has small demands
    - When to break a high priority job while meeting QoS