Energy-Aware Scheduling in Disk Storage Systems

Jerry Chou, Jinoh Kim, and Doron Rotem

Lawrence Berkeley Lab.

Email: { jchou, jinohkim, d_rotem }@lbl.gov

Distributed Computing Systems (ICDCS), 2011
31st International Conference

Guo-Wei Chen

Outline

- Introduction
- Related work
- Energy-aware scheduling
- Simulation
- Conclusions

Introduction (1/3)

- Because of the faster rotation speed and the larger capacity of disks, disks cost more energy

- Currently it is estimated that disk storage systems consume about 35 percent of the total power used in data centers

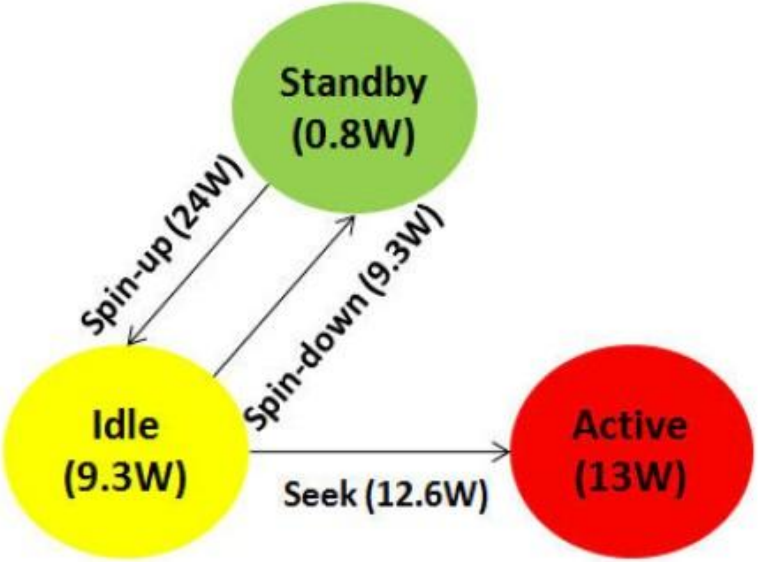
Introduction (2/3)

- Some energy saving techniques have been proposed like spinning down the disk

- But there are still some problems
  - Energy and response time penalty
  - Expected length of inactivity periods
  - Number if spin-up/down operations

Introduction (3/3)

- Power parameters from Seagate Barracuda specification



Description	Value	Description	Value
Idle power	9.3 W	Spin-up power	24 W
Active power	13 W	Spin-down power	9.3 W
Standby power	0.8 W	Spin-up time	15 sec
Breakeven time	54 sec	Spin-down time	10 sec

Related work

- There are some techniques related to the proposed scheme
  - Write off-loading
    - Minimize the energy consumed due to write requests
    - Newly written data is diverted to disks which is spinning
  - Replication for energy saving
    - Access data copies from spinning disks
    - Transition disks that contain redundant data to standby

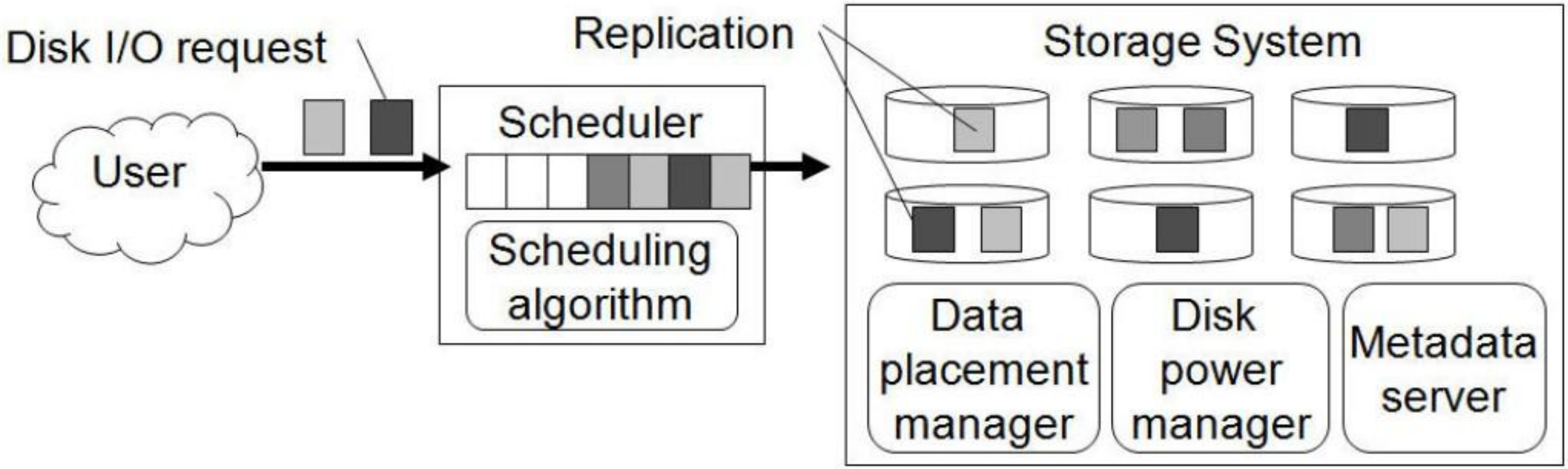
Energy-aware scheduling

- Storage system architecture

- Algorithms
  - Offline
  - Batch
  - Online

Storage system architecture

- Data spread across disks
- Data replicated for availability and performance
- Each request for a single data block (512B)



Scheduling algorithm

- Offline
  - A scheduler has a-priori knowledge of the arrival times of requests
- Batch
  - Queues requests and dispatches them all together to disks periodically at a scheduling interval
- Online
  - a scheduler immediately dispatches requests to disks upon their arrival

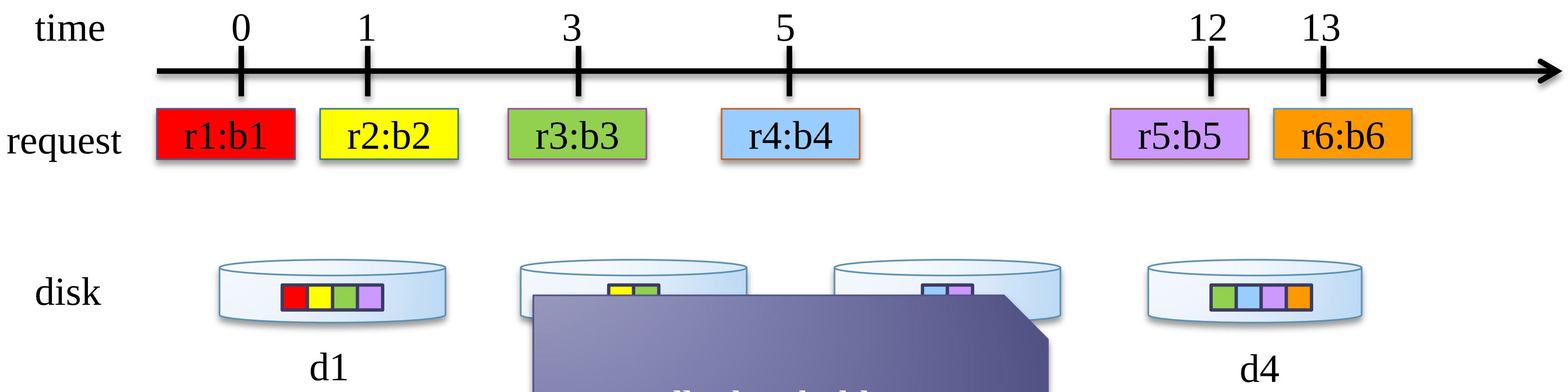
Offline scheduling

- The energy saving from any pair of requests is determined by their arrival time ***t***
- we can o… its successo… spun down

T_B: idle time threshold
P_I: idle power
E_{up}, E_{down}, T_{up}, T_{down}: the energy and time to spin up anddown

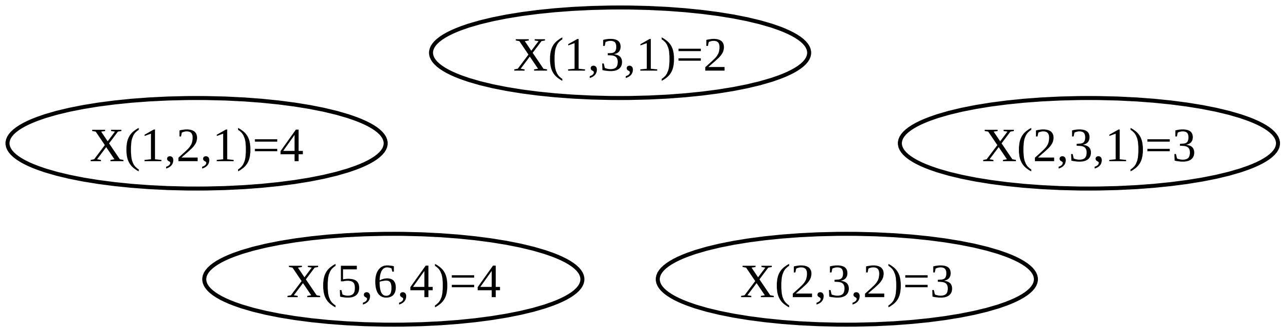
$$X(i,j,k) = \begin{cases} E_{up} + E_{down} + (T_B - (t_j - t_i)) * P_I, \\ \quad \text{if } 0 \leq t_j - t_i < T_B + T_{up} + T_{down} \\ 0, \quad \text{otherwise} \end{cases}$$

Offline scheduling algo. (1/3)

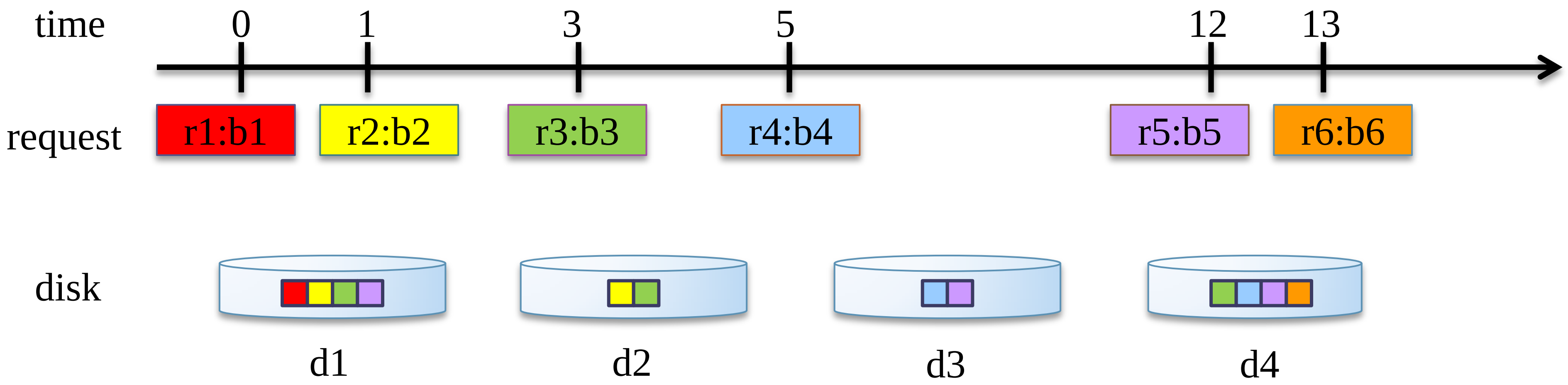


- **Operation flow:**
  - Step1: compute a ... ts

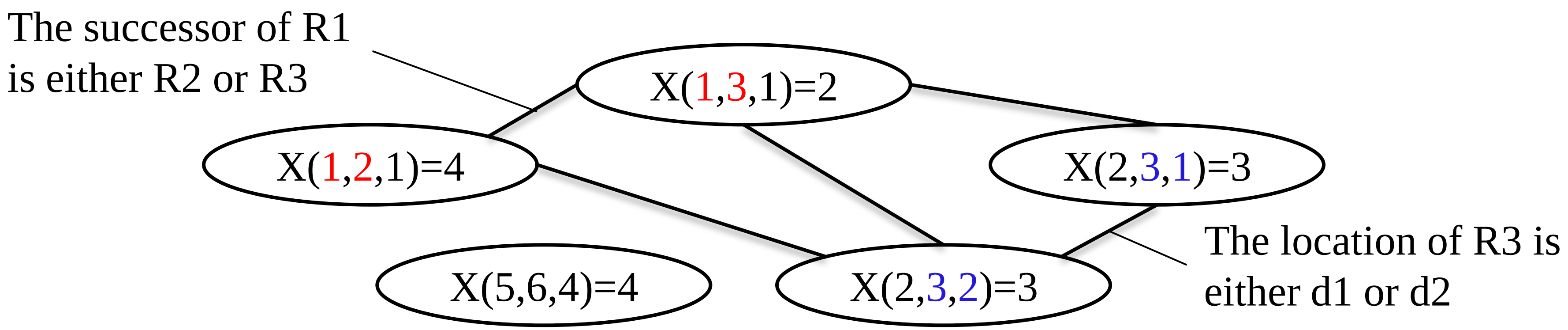
Idle threshold = 5
Disk power unit = 1



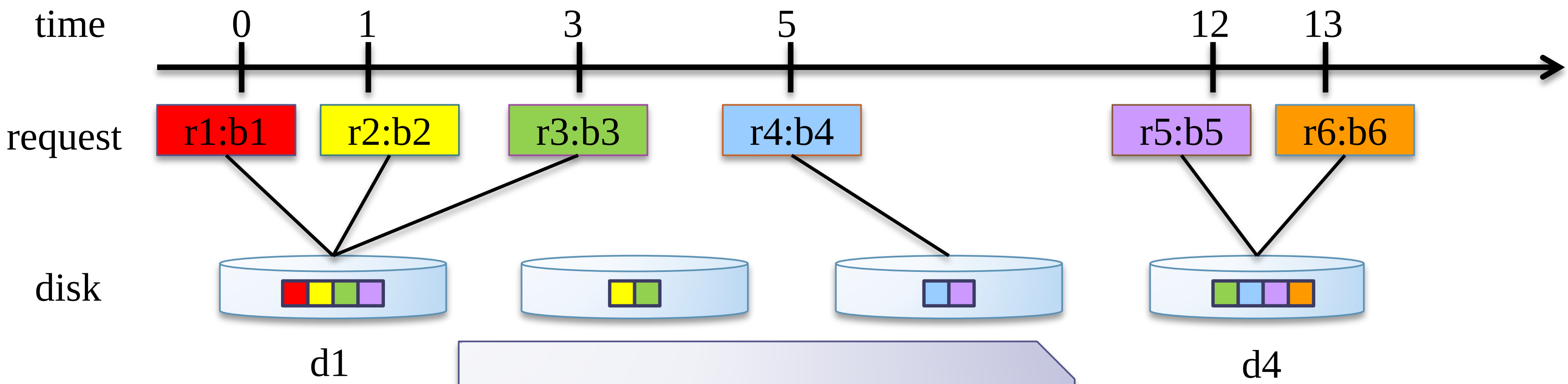
Offline scheduling algo. (2/3)



- **Operation flow:**
  - Step1: compute all energy saving from requests
  - Step2: add schedule constraints

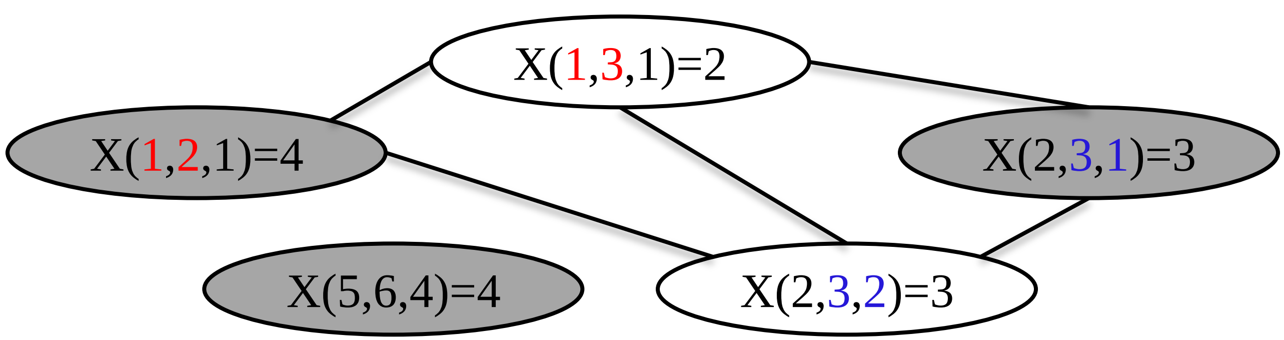


Offline scheduling algo. (3/3)



The saved energy compared to always-on disk is 4+4+3=11

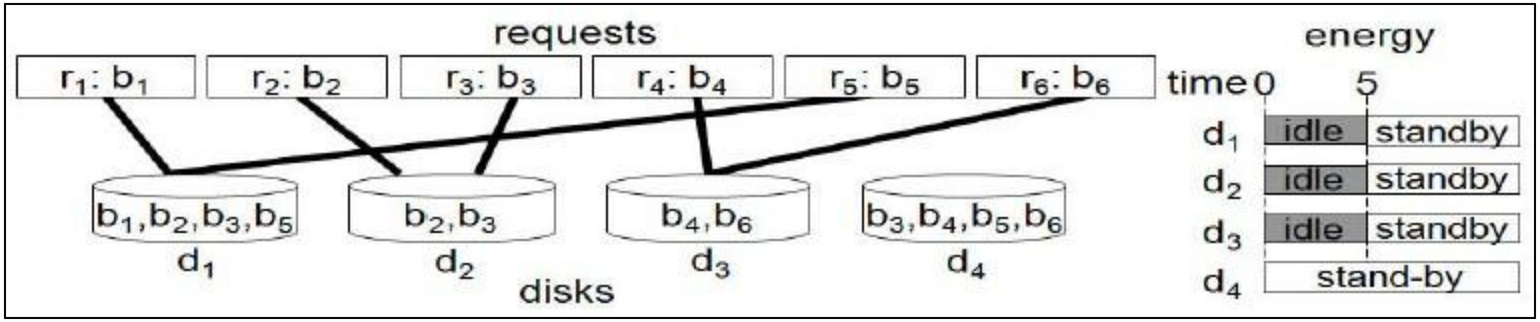
- **Operation flow:**
  - Step1: comp
  - Step2: add sc
  - Step3: find the maximum weighted independent set



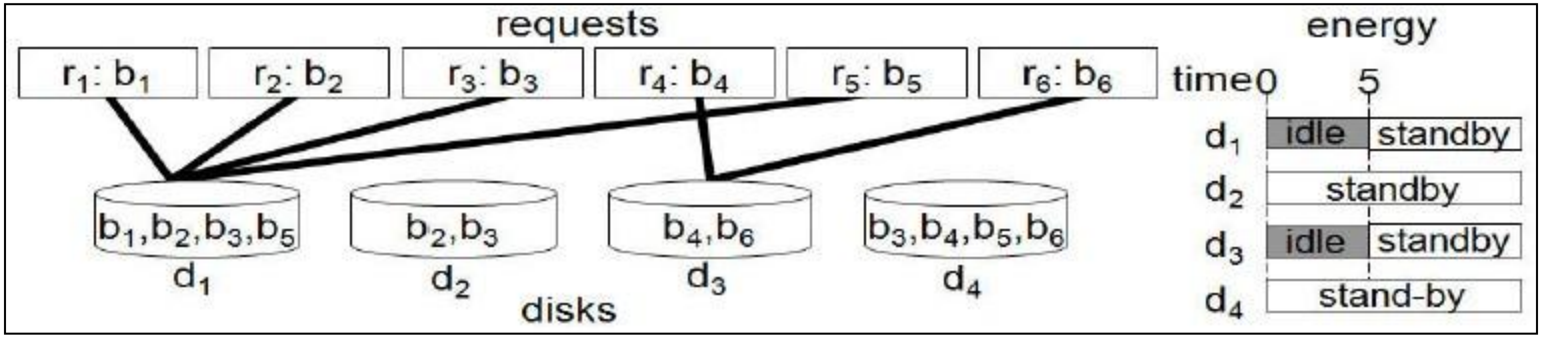
Batch scheduling (1/2)

- All requests access disks at the same time
- Energy consumption is proportional to the number of scheduled disks
- Minimize energy = minimize scheduled disks

Batch scheduling (2/2)



Energy cost = 5*3 = 15



Energy cost = 5*2 = 10

Online scheduling

- Schedule one request at a time
- The cost function:

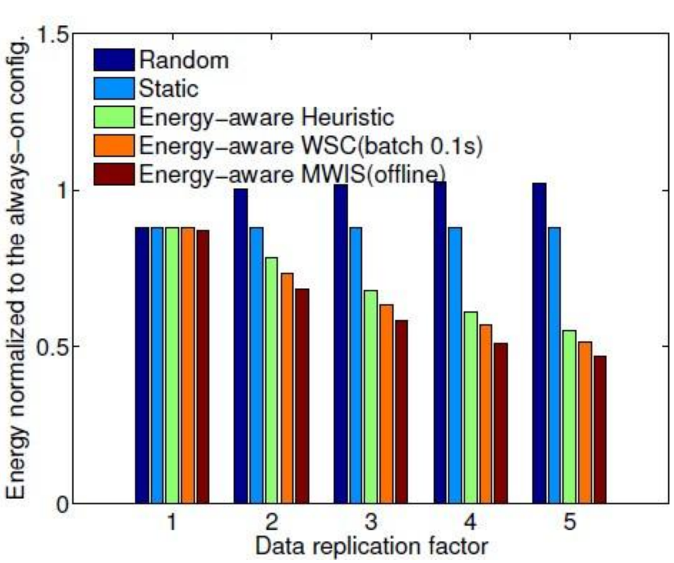
$$C(d_k) = E(d_k) * \frac{\alpha}{\beta} + P(d_k) * (1 - \alpha)$$

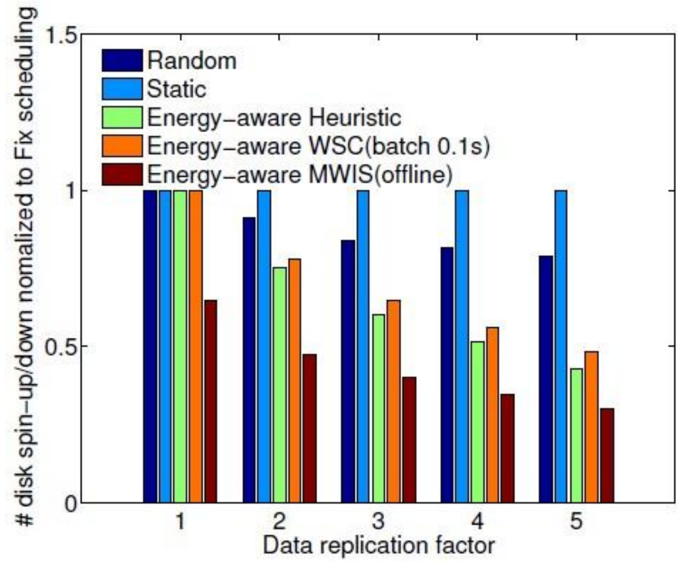
  - $E(d_k)$: Energy cost can be computed by disk idle time
  - $P(d_k)$: number of requests queued on disk d_k
  - α, β: Cost parameter

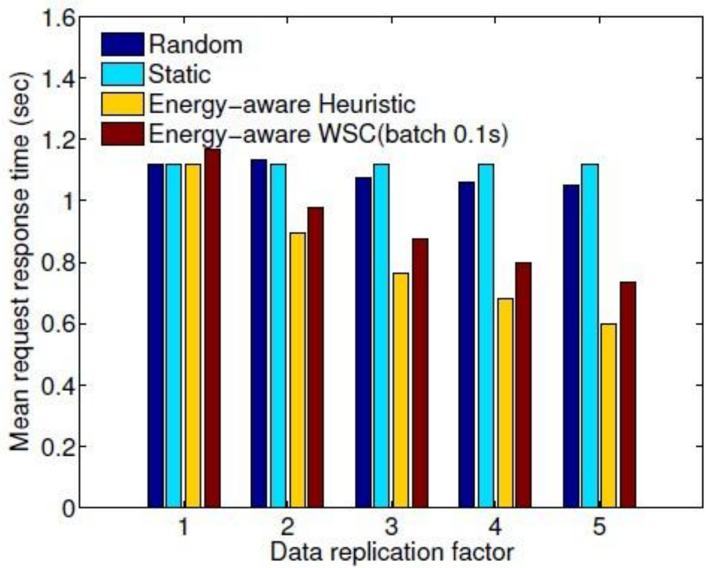
Simulation (1/4)

- Workload trace
  - Cello: collected by IBM
- Simulator
  - Omnet++ for system simulation
  - DiskSim for disk simulation
- Data placement
  - 180 disks
  - Original data is skewed distributed
  - Replicated data is uniform distributed

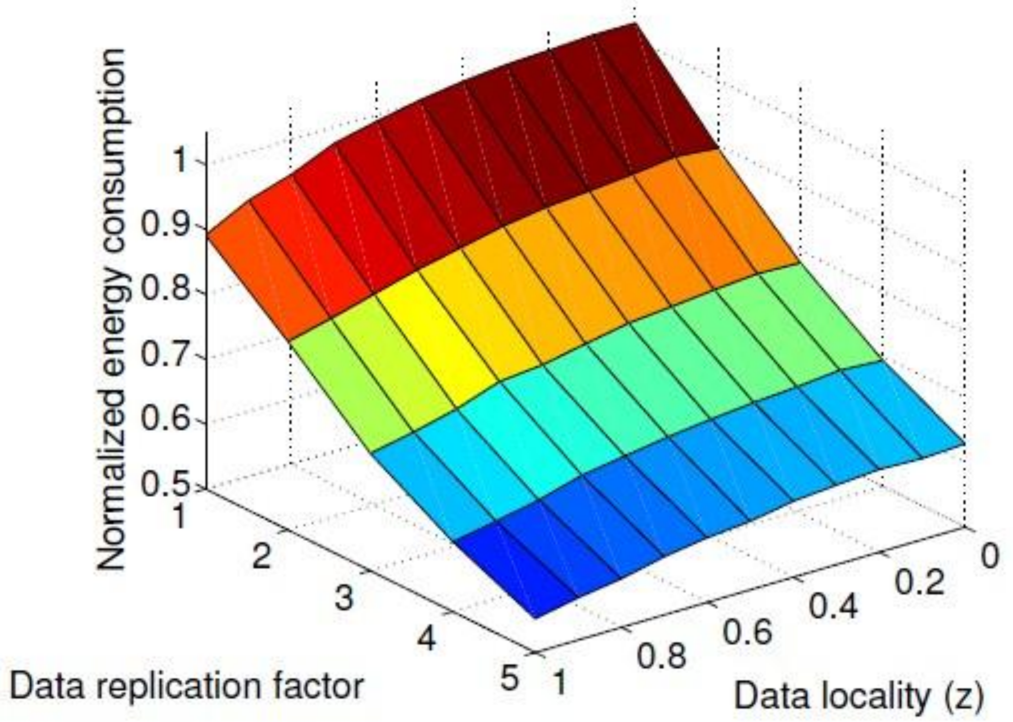
Simulation (2/4)



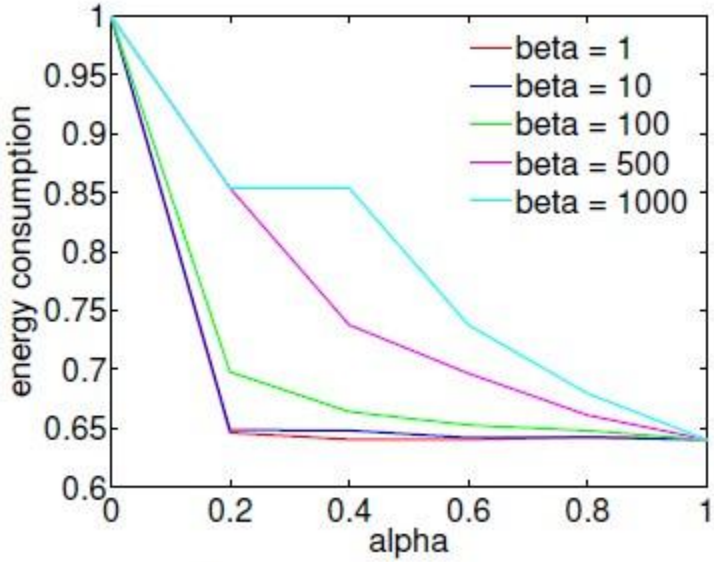




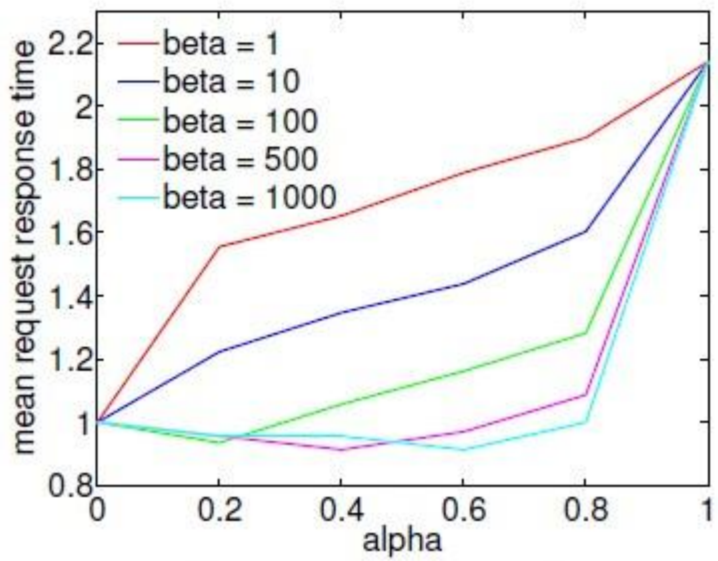
Simulation (3/4)



Simulation (4/4)



(a) Energy consumption



(b) Request response time

Conclusions

- Propose scheduling algorithms for online, batch and offline models
- Show significant performance and energy improvement using realistic traces
- Future work on better online scheduling algorithm