SRCMap: Energy Proportional Storage using Dynamic Consolidation

By: Akshat Verma, Ricardo Koller, Luis Useche, Raju Rangaswami
Presented by: James Larkby-Lahet

Motivation

- storage consumes 10-25% of datacenter power (higher ratio at lower loads)
- storage virtualization is happening already (in part driven by OS virtualization)
- can we use create energy proportionality in virtualized storage systems the same way as in OS virtualization?
  - workload variability exists, migration is more expensive

Virtualized Storage

- Virtualized Storage: a set of 'logical volumes' provided across SAN (Fiber Channel, iSCSI) to compute nodes
- In this case, storage is provided by a set of physical volumes, e.g. RAID arrays
- virtualization server maps logical to physical at some granularity (volumes == low metadata and performance overhead, blocks == efficient utilization)

Overall Design

- Claims: working set is small and stable, workload intensity varies within and across volumes
- pre-replicate working set to other volumes and offload writes
- virtualization redirects accesses away from spun-down disks
- create close to N power levels with N volumes

Design Goals

- fine-grained energy proportionality: many power levels
- low space overhead: .25x is reasonable, coarse granularity & disks *are* cheap, however Nx is not
- reliability: on-off duty cycles are limited
- workload shift adaptation: must maintain proportionality while adapting
- heterogeneity: multi-vendor & multi-generation datacenters

Existing Solutions

Design Goal	Write offloading	Caching systems	Singly Redundant	Geared RAID
Proportionality	~	X	X	~
Space overhead	✓	✓	X	X
Reliability	X	X	✓	✓
Adaptation	X	✓	✓	✓
Heterogeneity	~	~	~	X

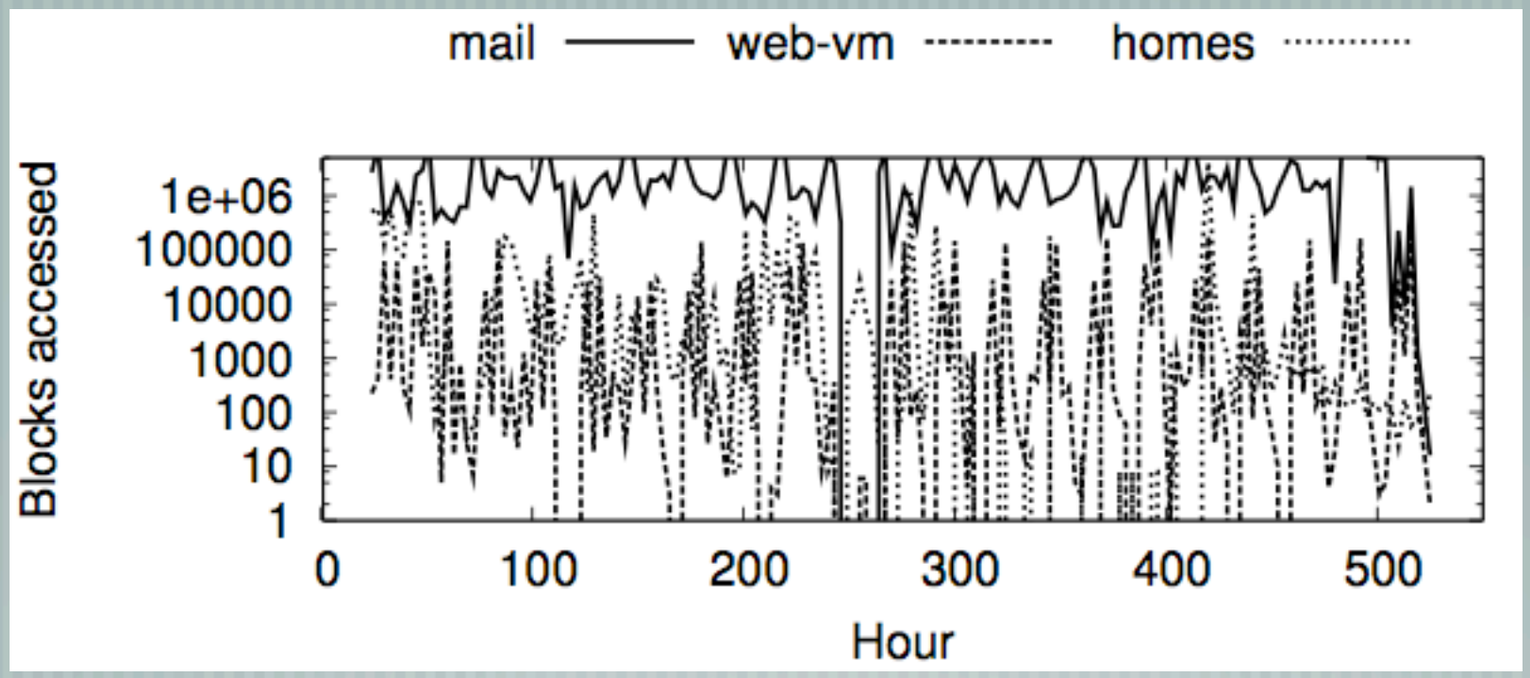
- singly redundant schemes: able to power off a (parity) disk
- geared RAID: skewed striping of replicated data
- caching: disk or SSD cache of popular data, PDC
- write offloading: cache writes somewhere persistent

Workload Characteristics: 1

Workload Volume	Size [GB]	Reads [GB] Total	Reads [GB] Uniq	Writes [GB] Total	Writes [GB] Uniq	Volume accessed
mail	500	62.00	29.24	482.10	4.18	6.27%
homes	470	5.79	2.40	148.86	4.33	1.44%
web-vm	70	3.40	1.27	11.46	0.86	2.8%

- active dataset is typically a small fraction of the total data

Workload Characteristics: 2



- There is a significant variability in workload intensity

Workload Characteristics: 3

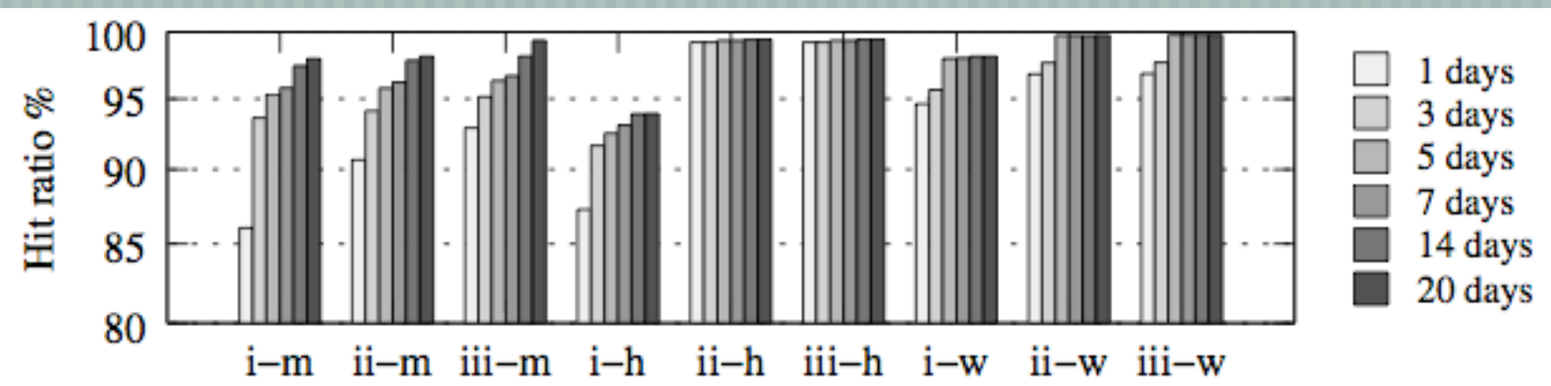
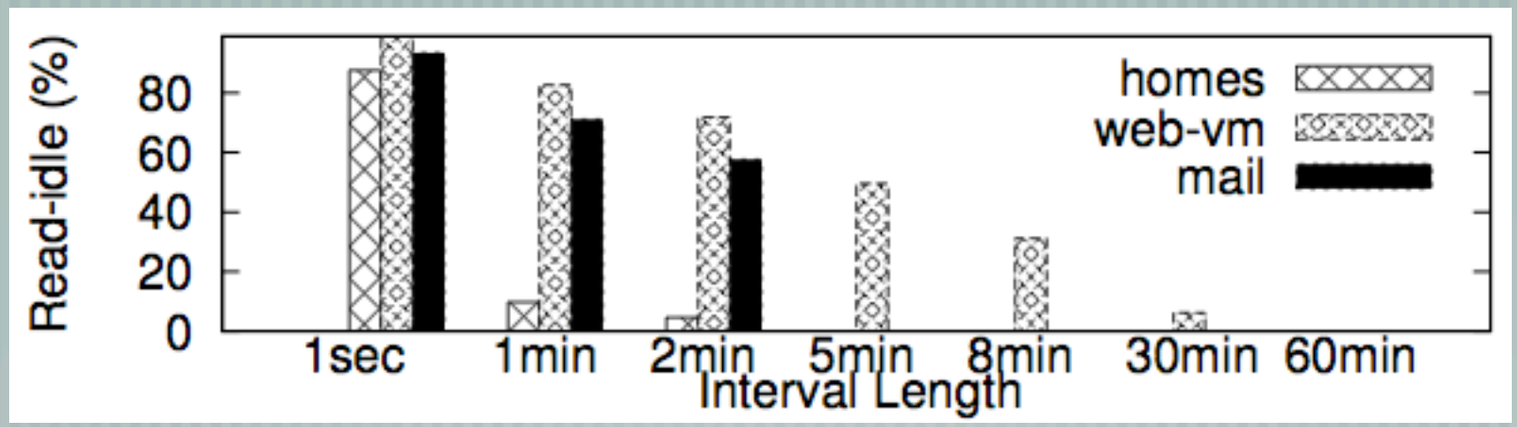


Figure 2: **Overlap in daily working sets for the mail (m), homes (h), and web-vm (w) workloads.** *(i) Reads and writes against working set , (ii) Reads against working set and (iii) Reads against working set, recently offloaded writes, and recent missed reads.*

- data usage is skewed toward popular and recent data

Workload Characteristics: 4

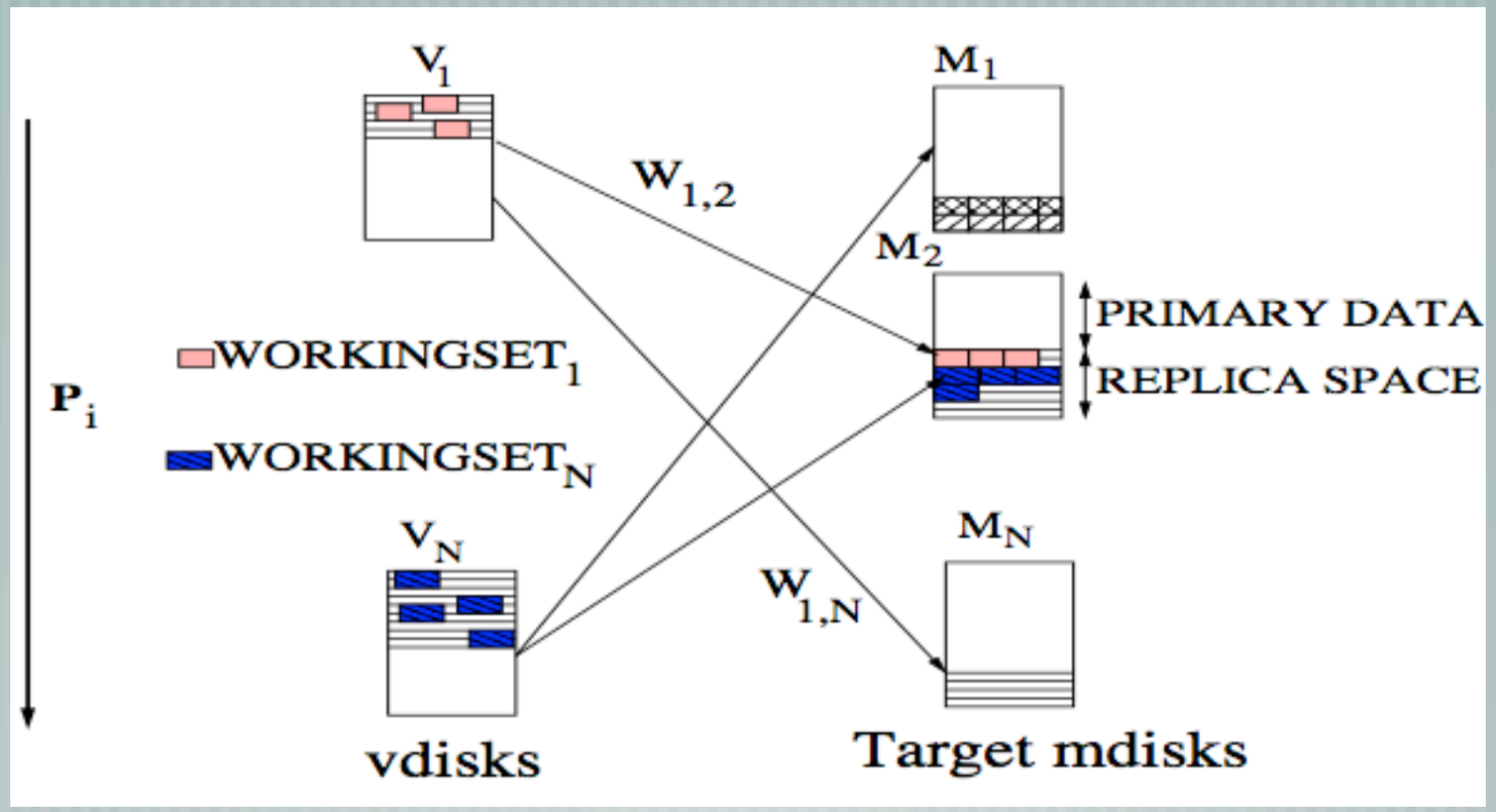


- read-idle time is dominated by small durations
- write-offloading and spin-down alone won't save much power and will significantly increase # of duty cycles, reducing reliability

SRCMap

- want to power only a subset of disks (volumes)
- migration is expensive
  - assign each logical volume (vdisk) to a physical volume (mdisk)
  - store working sets of other vdisks in the free space of mdisks
  - power the minimum subset of mdisk to serve all vdisks at acceptable performance

SRCMap Illustrated



Rationale

- multiple replica targets - during peak load, the primary mdisk is active, under lower load, multiple replicas are required to provide fine grained energy proportionality
- sampling - replicating entire vdisks is impractical, working sets are much smaller and reasonable to replicate
- ordered replica placement - space is still an issue, not all replicas are equal. prefer to replicate idle and small

Rationale, continued

- dynamic vdisk -> mdisk mapping - workload varies and some vdisks are more highly replicated. must decide online
- dual data sync - update replica on read miss to adapt to workload shifts, lazy incremental sync with non-active replicas on active mdisks
- coarse grained power cycling - consolidation interval (on the order of hours) where the active mdisks don't change (except replica misses)

SRCMap Overview

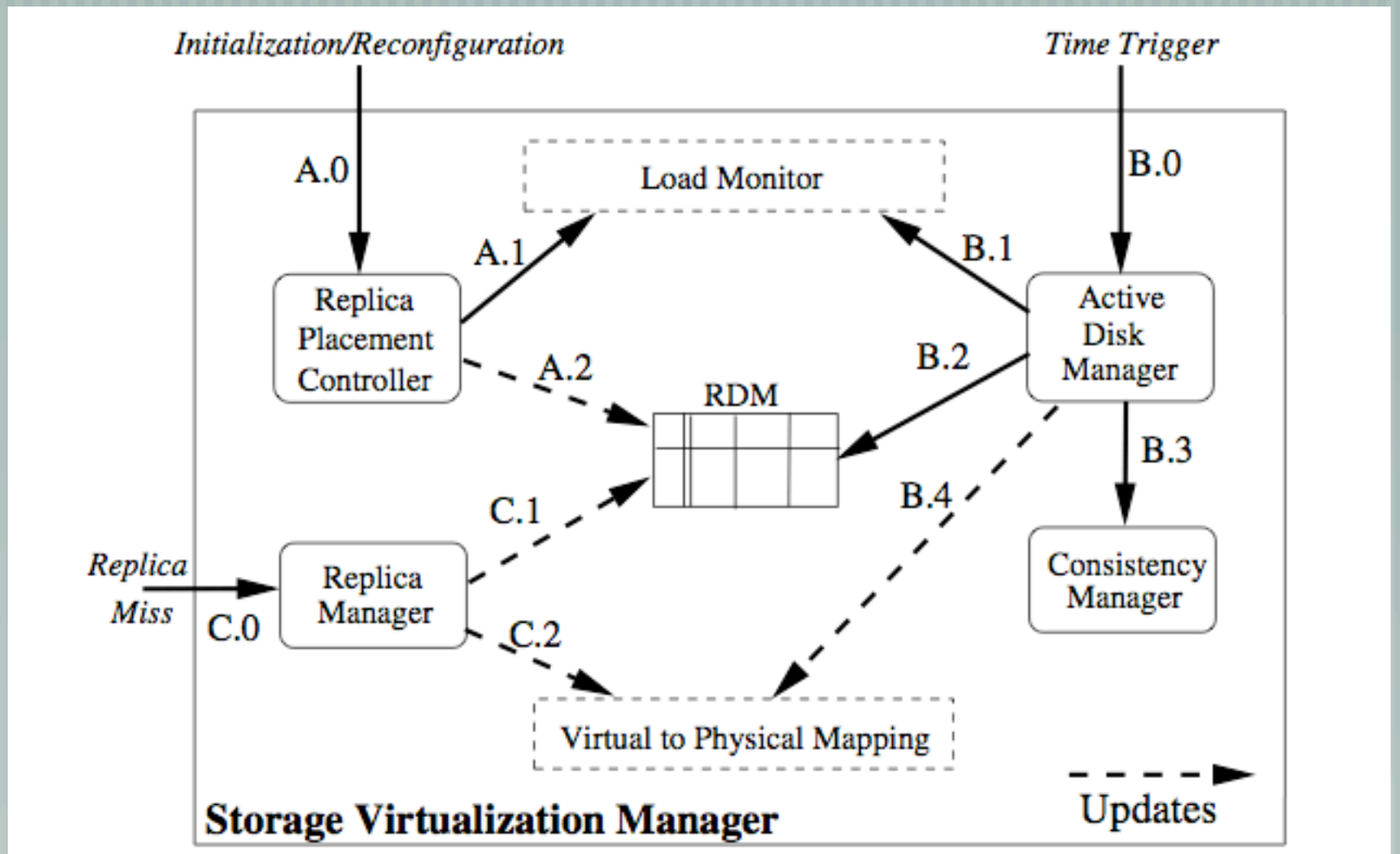


Figure 4: **SRCMap integrated into a Storage Virtualization Manager.** *Arrows depict control flow. Dashed/solid boxes denote existing/new components.*

Replica Placement Algo.

- better vdisks to replicate - smaller working set, stable working set (lower replica miss rate), small average load, hosted on a less power efficient volume
- *Ordering Property*: if vdisk V_i is more likely than V_j to require an replica during Active Disk Selection, V_i is more likely than V_j to find a replica among the active mdisks

Replica Placement Algo. 2

- order vdisks based on cost-benefit tradeoff
- create a bipartite graph that reflects this ordering
  - iteratively create one source-target mapping that respects ordering
  - recalibrate edge weights to respect *Ordering Property*

Initial vdisk Ordering

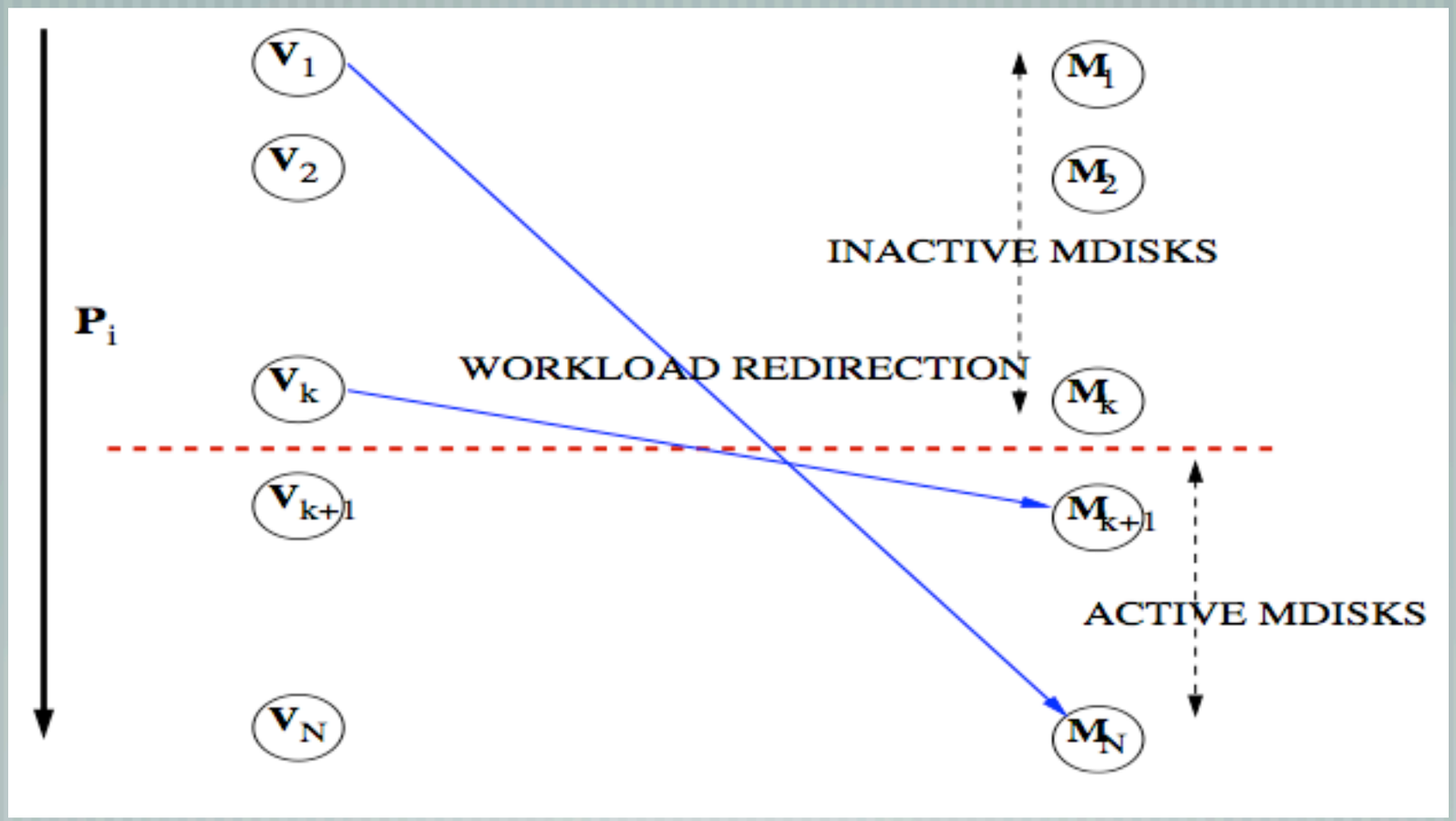
$$P_i = \frac{w_1 WS_{min}}{WS_i} + \frac{w_2 PPR_{min}}{PPR_i} + \frac{w_3 \rho_{min}}{\rho_i} + \frac{w_f m_{min}}{m_i} \quad (1)$$

- Pi = probability that vdisk i's primary mdisk is spun down
- w = tunable weights
- WS = size of working set
- PPR = ratio between peak I/O bandwidth and peak power
- ρ = average IOPS
- m = number of read misses in working set

Bipartite Matching

- maps Vdisks to Mdisks with weight as cost-benefit of replication
- Vdisks sorted by inverse P, aligned with primary Mdisk
  - match is made by allocating a replica to topmost Mdisk from Vdisk with highest edge weight
  - weights for selected Vdisk are multiplied by probability of target Mdisk, and next iteration begins

Active Disk Selection Goal



Active Disk Selection

- run every interval, or if performance degrades
- 1) estimate load for each vdisk as load from prior interval
- 2) determine minimum number of mdisks to meet aggregate load (and select the mdisks with smallest P to be active)
- 3) for any vdisk whose mdisk is not selected, find a replica with spare bandwidth on the active mdisks
- 4) if no replica can be found, increase number of active mdisks and repeat 3

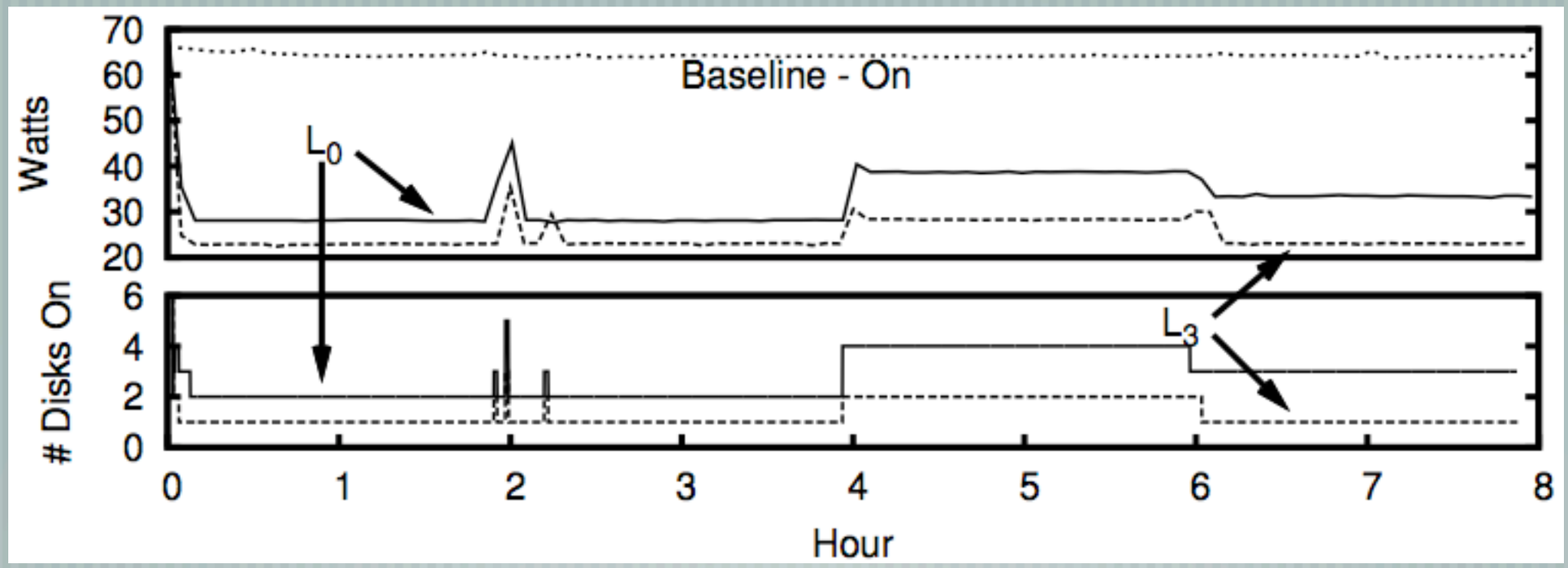
Optimizations

- sub-volumes - sub-divide vdisks for easier replica packing
- replica scratch space - for write buffering and missed reads

Evaluation

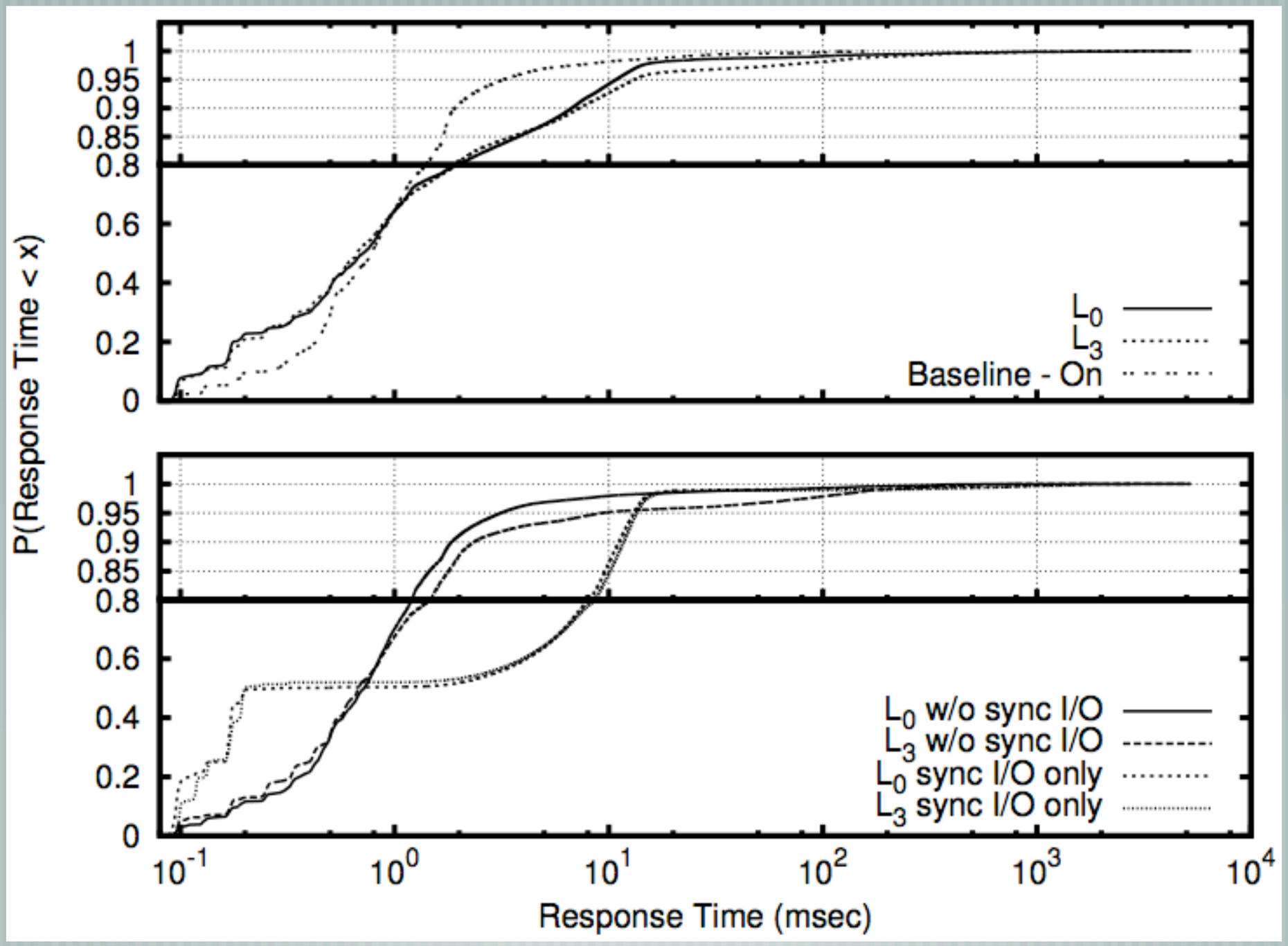
- testbed system with 8 SATA channels, single disks acting as mdisks
- *Watts up?* power meter monitoring disks power
- simulator seeded with testbed values for longer running traces
- workloads: block traces of volume request for webserver, home directories, svn, wikis

Prototype - Peak 8 Hours

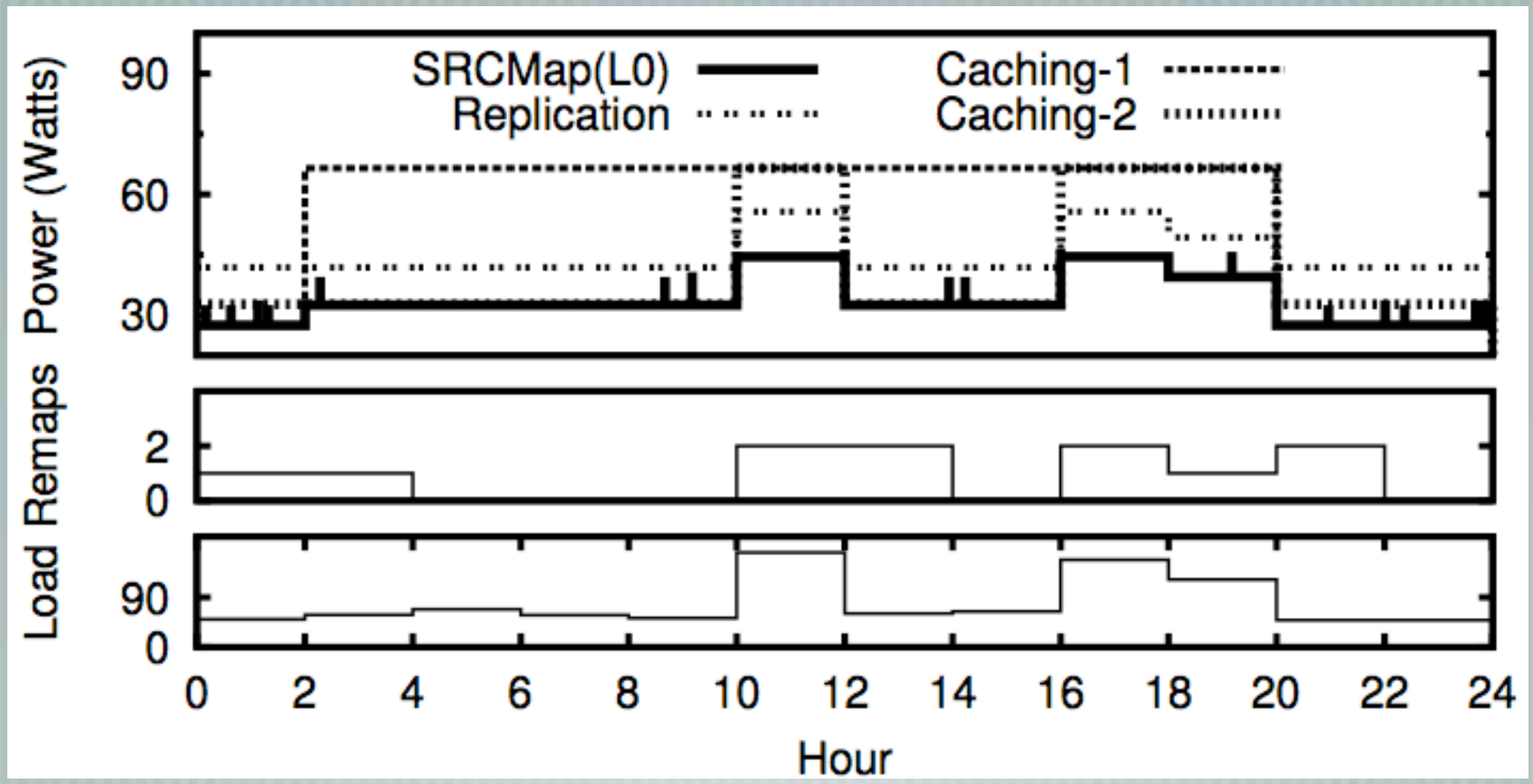


- L0 - 35.5% average power reduction L3 - 56.6%
- .0003% requests suffer read miss spin-up delays

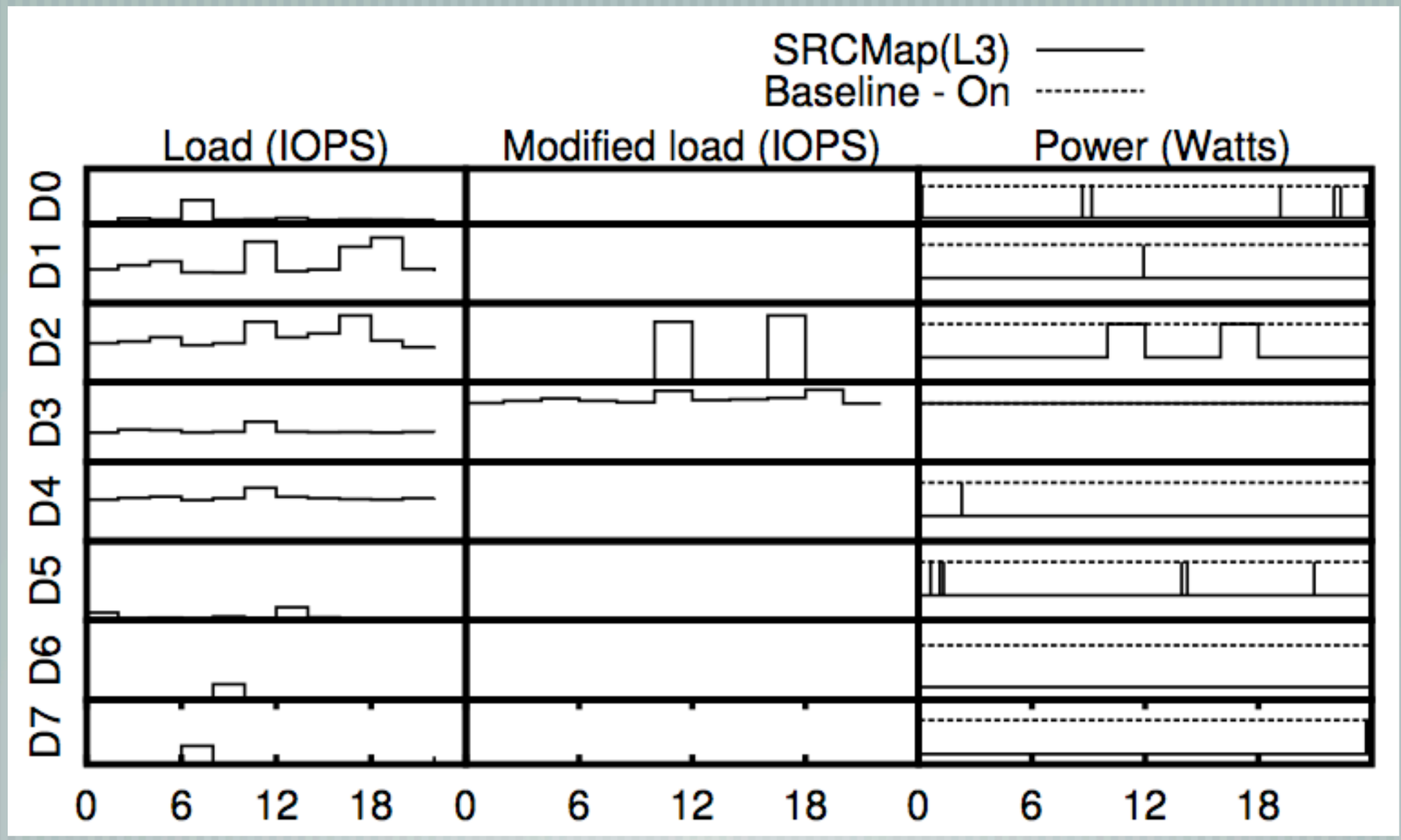
Prototype - Peak 8 Hours



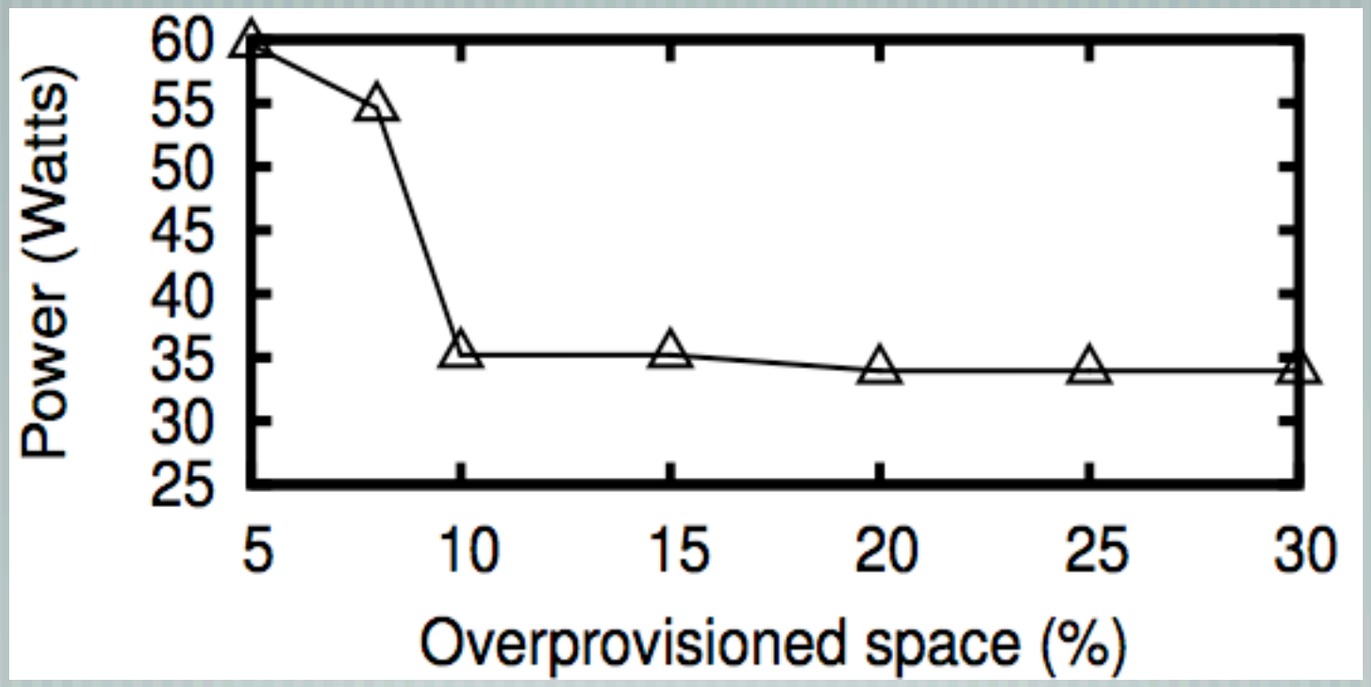
Simulation - Competitors



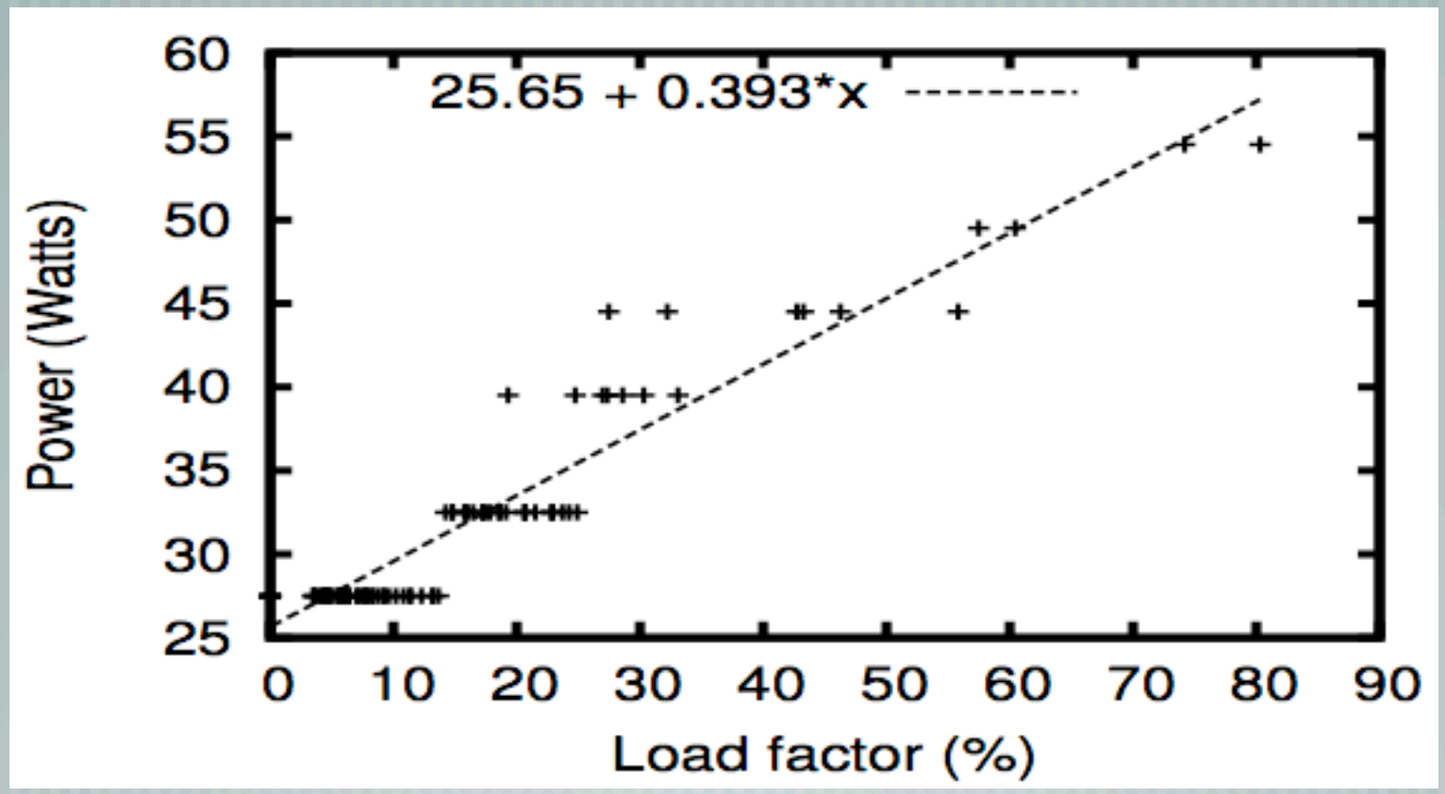
Aggressive Load Consolidation



Sensitivity to Free Space



Energy Proportionality



Overhead

- Per block map - current avtice replica, version, write redirect
- Volumes * Size * % replica space * 13 / 4k
- 10 10TB volumes with 10% over-provisioning
  - 3.2GB metadata

Conclusion

- feasible to build dynamically consolidated, energy-proportional storage system
- meets goals of fine-grained proportionality, low space overhead, reliability, workload adaptation, and heterogeneity support
- TODO: better synchronization I/O scheduling