PACEMAKER: Avoiding HeART attacks in storage clusters with disk-adaptive redundancy

Saurabh Kadekodi, Francisco Maturana,
Suhas Jayaram Subramanya, Juncheng Yang,
K. V. Rashmi, Gregory R. Ganger

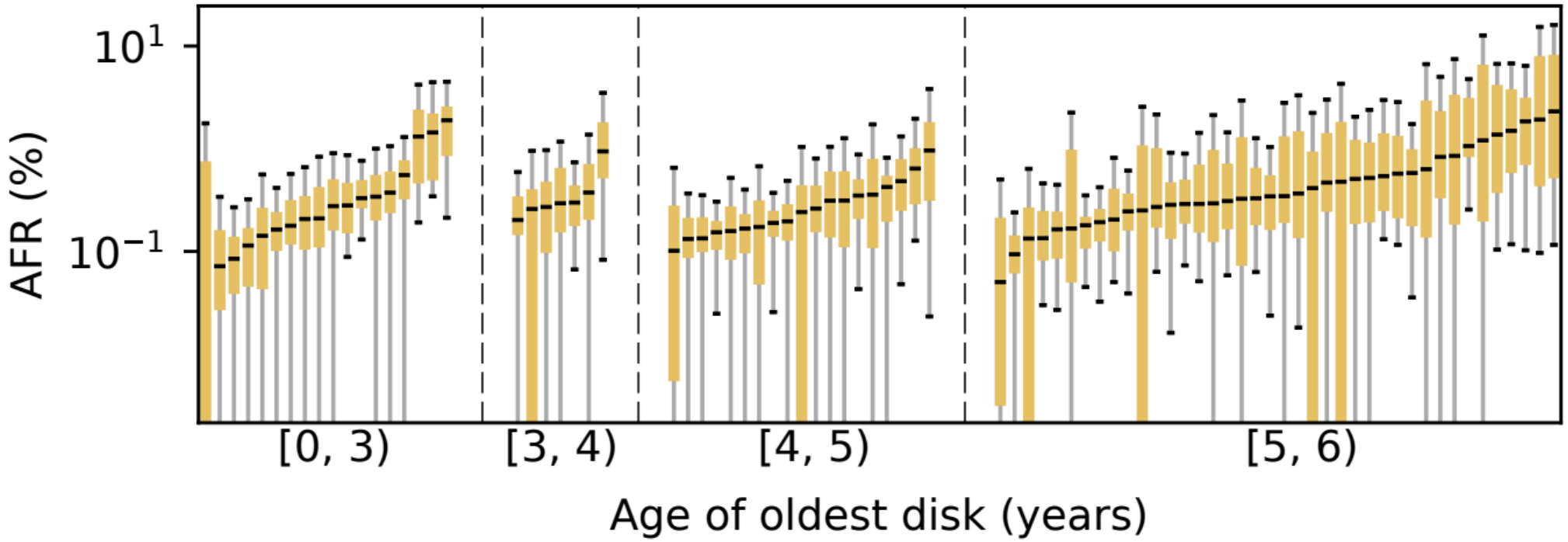
OSDI 2020

Background

- Storage clusters are made up of disks from a mix of makes/models acquired over time
- Data redundancy scheme store auxiliary 'parity chunks,' to provide the ability of data recovery
- Tuning redundancy schemes to observed disk failure rates can provide balance between space-saving and data-protection.

AFR

AFR describes the expected fraction of disks that experience failure in a typical year.
AFR is used to describe disk failure rates

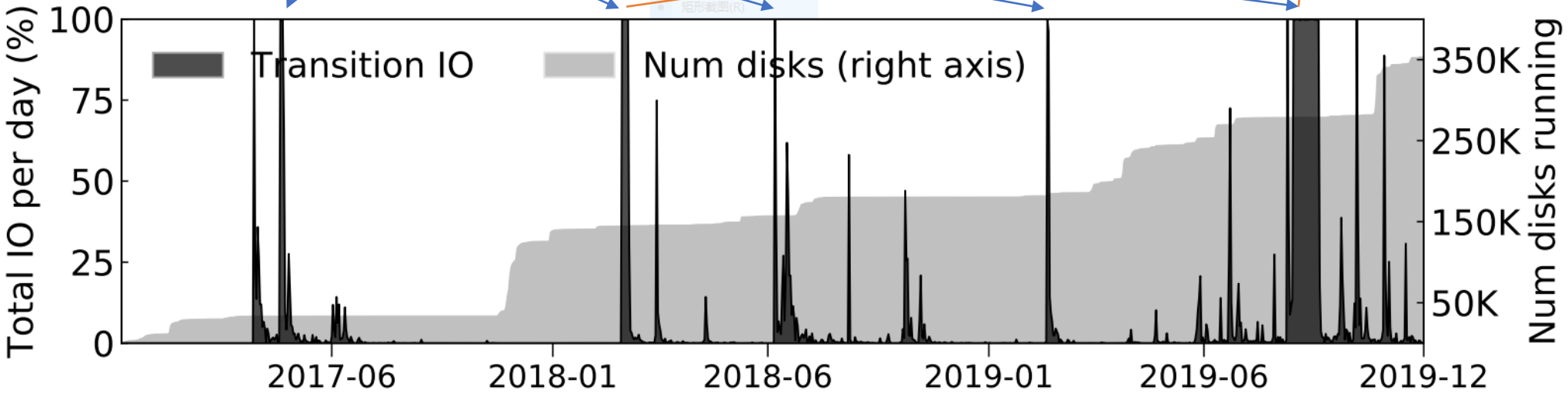


(a) Spread of make/model AFRs

The last Design(HeART): Unusable

redundancy scheme transition use all the IO : Transition Overload

data is already under-protected until transition is over



(a) Transition IO for HeART [27] on Google Cluster1.

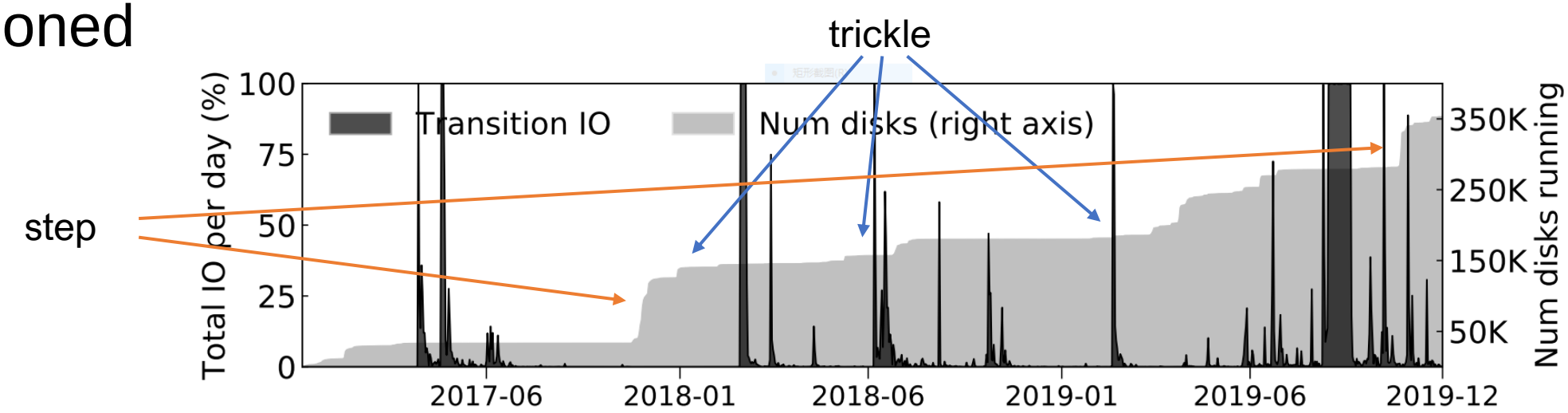
Address the problem

- Longitudinal production trace analyze
- 5.3 million HDDs from several clusters
- trickle-deployed disks(slow rise)
- step-deployed disks(massive disks at once).

Causes of transition overload

- Trickle-deployed disks need more time to make enough disks identified as in need of transition.

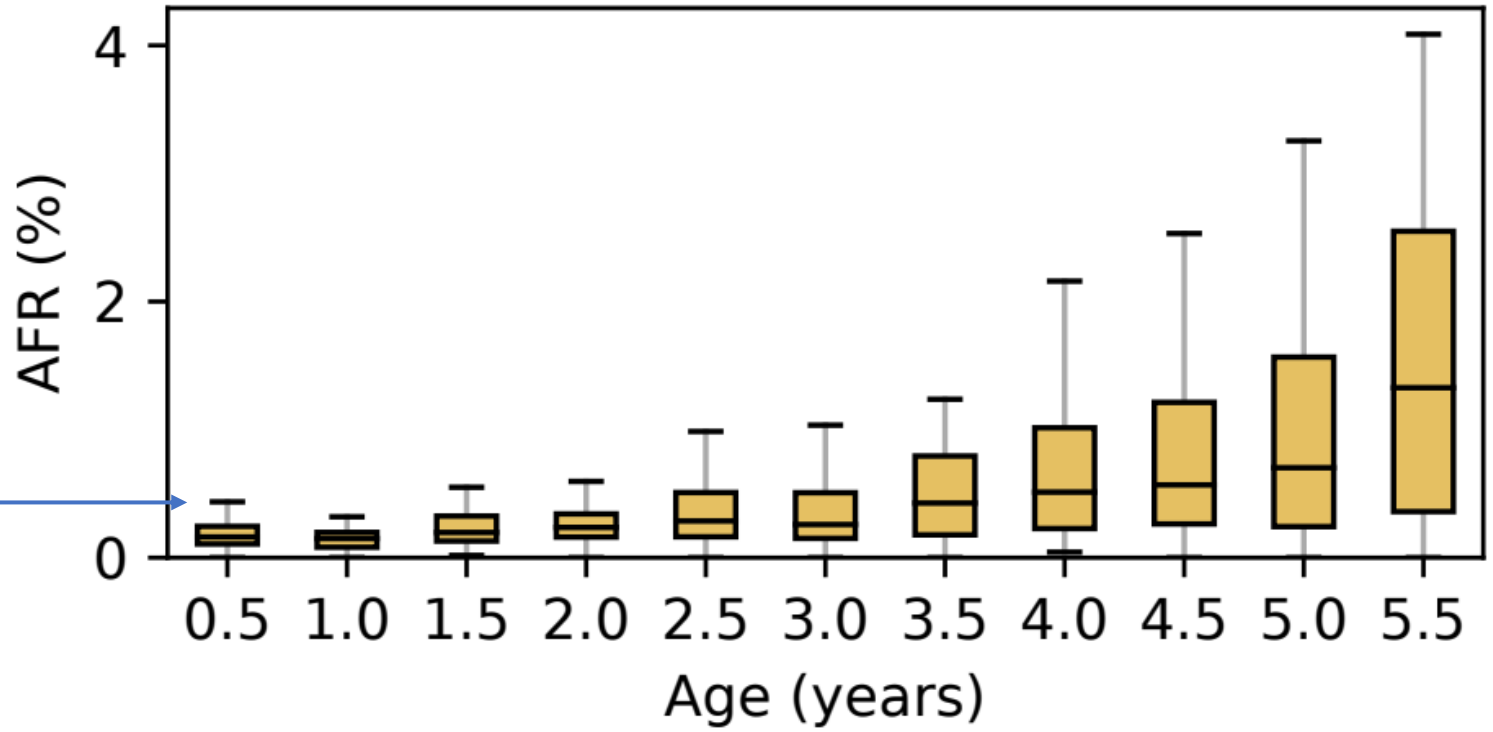
- Step-deployed disks, all large-number disks should be transitioned



(a) Transition IO for HeART [27] on Google Cluster1.

AFRs rise gradually over time

Short-lived infancy

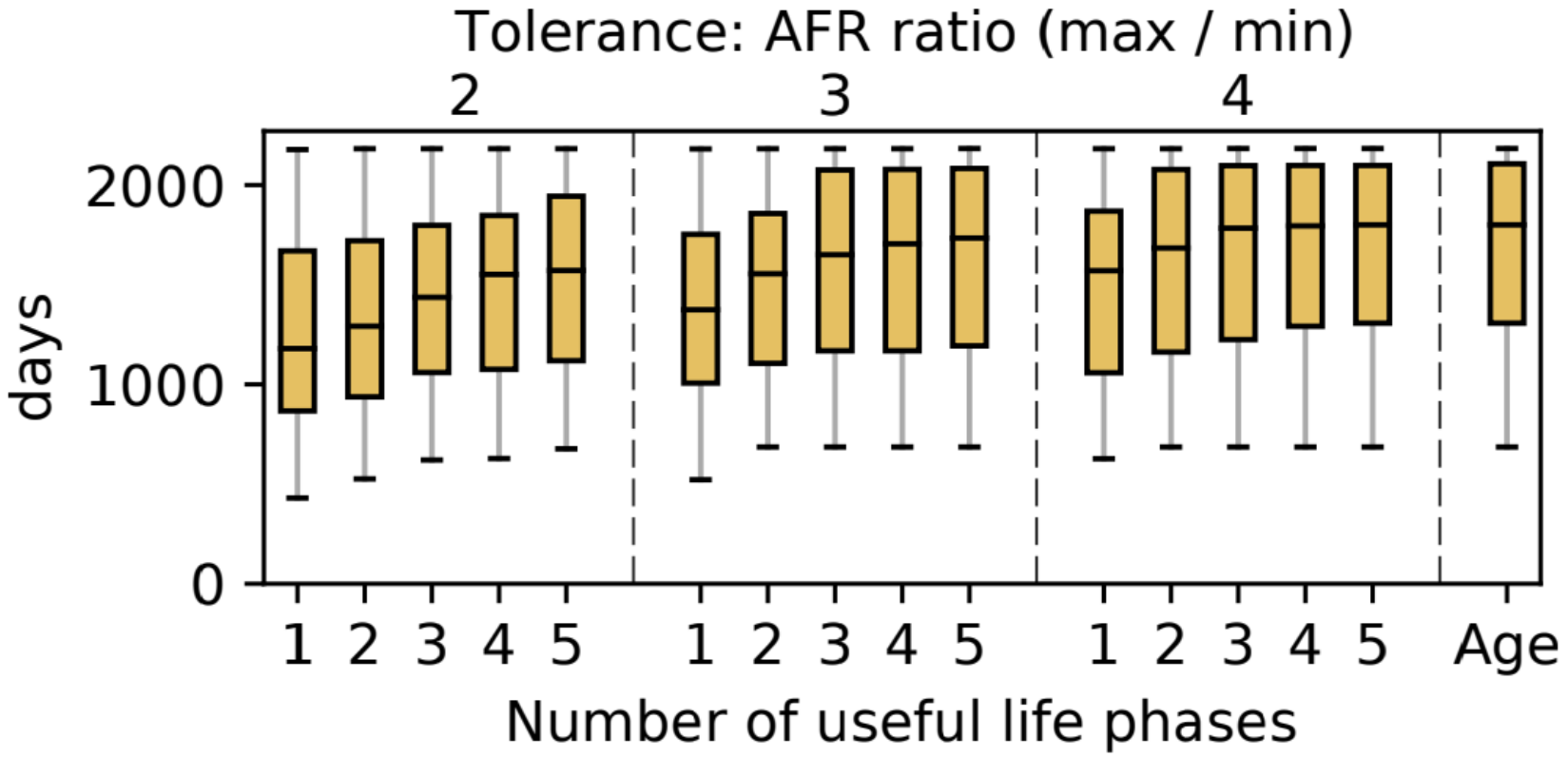


(b) AFR distribution over disk life

Multi-phased disk life

Considering more than one phase will extend the length of useful life

Approximate length of useful life changes by little when considering four or more phases



(c) Approximate useful-life length

Constraints

1. Reliability constraint on all data all the time
2. Failure reconstruction IO constraint on all disks all the time
3. Peak-IO constraint on all disks all the time
4. Average-IO constraint on all disks over time

Solution: PACEMAKER

- Proactively initiate transitions
- Selecting the right redundancy schemes
- transition the disks in the most IO-efficient way

Design

- Each disk under PACEMAKER simultaneously belongs to one Dgroup and one Rgroup
- Dgroup is used to differentiate disk makes/models
- Each Rgroup has an associated redundancy scheme

- Multiple Rgroups can use the same redundancy scheme
- No stripe may span across Rgroups
- The Dgroup of a disk never changes, but a disk may transition through multiple Rgroups

- At the time of deployment (or “birth”), the disk belongs to Rgroup0, use the default redundancy scheme
- The first transition any disk undergoes is an RDn transition(to a lower level of redundancy)
- Then all the next transitions are RUp transition(to a higher level of redundancy)

Components

Determine when to transition disks using the AFR curves and the disk deployment information

Choose the Rgroup to which the disks should transition

Address how to transition the disks to the planned Rgroup in the most IO-efficient way

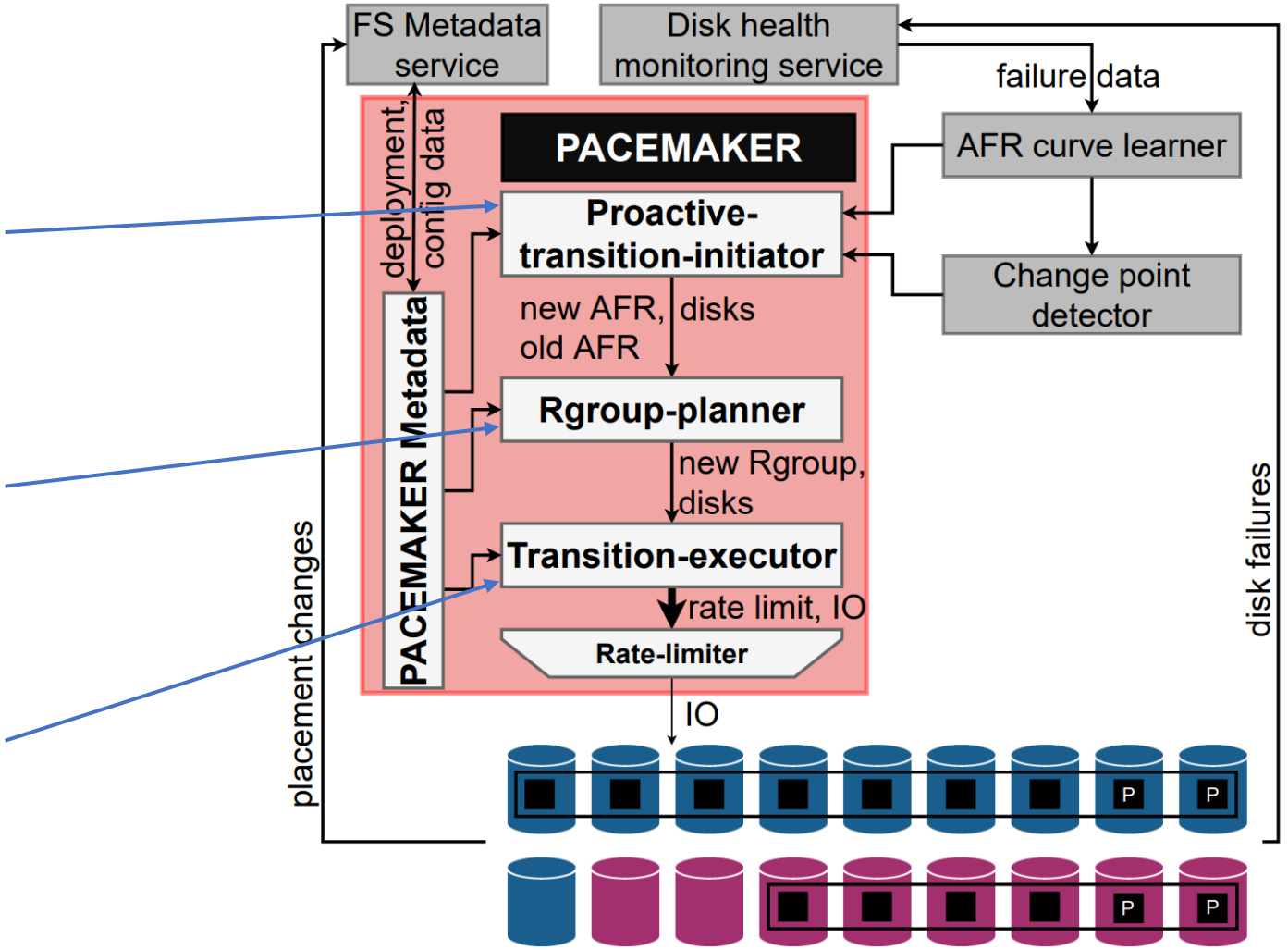


Figure 3: PACEMAKER architecture.

Proactive-transition-initiator

- Decide when to RDn a disk
  - the AFR has decreased sufficiently, and is stable
  - only once in a disk's lifetime
- Decide when to RUp a disk
  - For trickle-deployed disks:
    - the first-deployed small number (e.g., 3000) of disks is used to observe the AFR curve of the same mark/model, called canary disks, they remain in default redundancy scheme
    - the same model disks deployed later will be transitioned proactively
  - For step-deployed disks:
    - PACEMAKER sets a threshold, termed threshold-AFR, which is a (configurable) fraction of the tolerated-AFR of the current redundancy scheme
    - when the observed AFR crosses the threshold-AFR, initiates a proactive RUp transition

Rgroup-planner

- To avoid transition overload, IO constraints should be met when the Rgroup-planner estimates whether the transition to a scheme is worth it.
- If all the constraints are met, Rgroup-planner chooses the one that provides the highest space-savings.
- For trickle-deployments, Rgroup-planner creates a new Rgroup for a redundancy scheme if and only if one does not exist already to prevent too many small-sized Rgroups.
- For step-deployments, Rgroup-planner creates a new Rgroup for each step-deployment, even if there is already one or more Rgroups that employ the chosen scheme.

Transition-executor

- Conventional re-encoding: reading–re-encoding–writing all the stripes whose chunks reside on each transitioning disk
- If a small percentage of a Rgroup's disks are being transitioned, retain the transitioning disks' contents, and simply move (copy) data to other disks within the current Rgroup
- If a large fraction of disks in a Rgroup need to transition together, it is more efficient to transition the entire Rgroup rather than only the disks that need a transition at that time, because the data chunks have to be only read for computing the new parities, but they do not have to be re-written
- Almost always, trickle-deployed disks use Type 1 because they transition a-few-at-a-time, and step-deployed disks use Type 2 because Rgroup-planner maintains each step in a separate Rgroup

Implementation in HDFS

- Easy
- Because the components required for the transition-executor are already present and adequately modular.

A natural mechanism to realize Rgroups in HDFS is to have one DNMgr per Rgroup

The NN is usually a high-end server compared to the DNs, and an additional tens of threads shouldn't affect performance

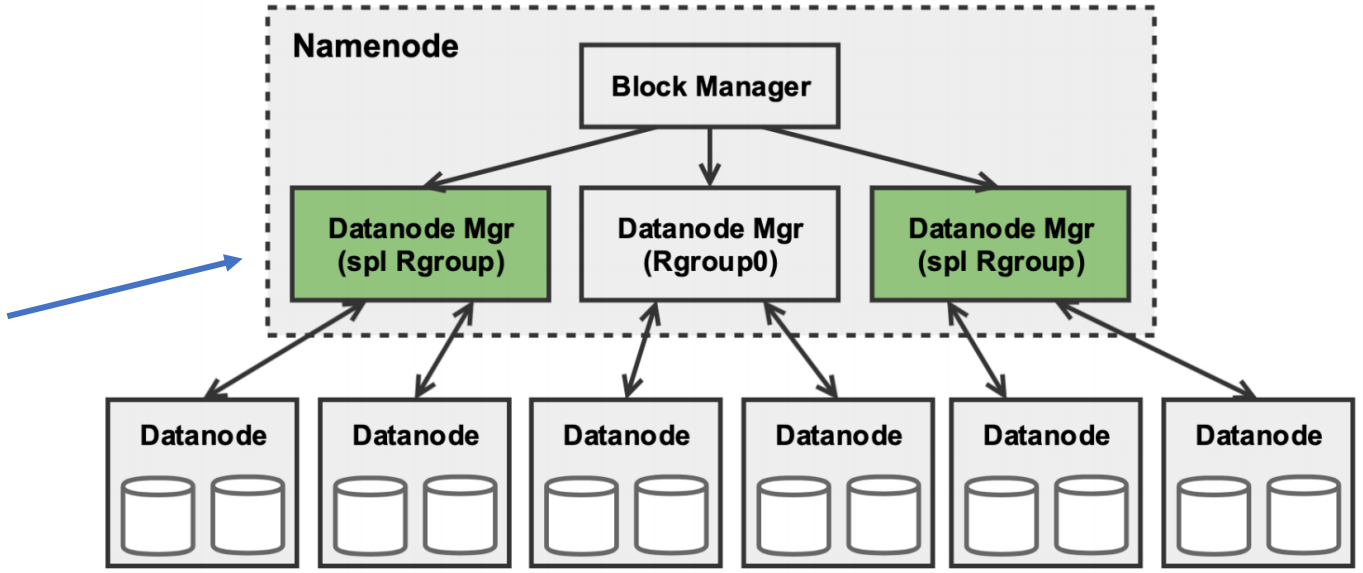
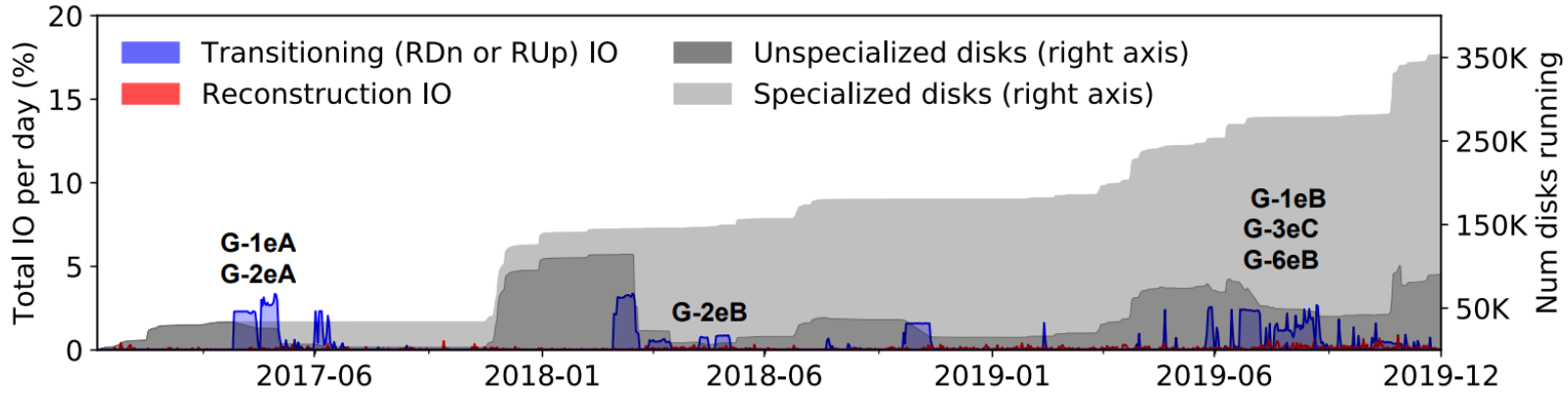


Figure 4: PACEMAKER-enhanced HDFS architecture.

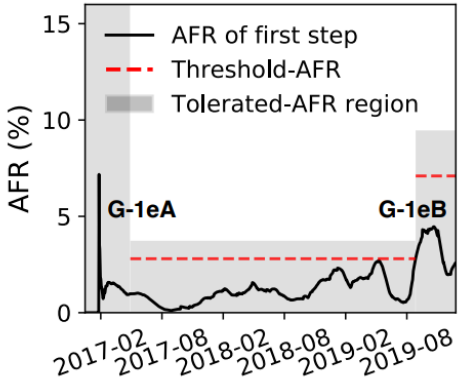
Evaluation methodology

- Simulated chronologically for each of the four cluster logs
- For each simulated date, the simulator changes the cluster composition according to the disk additions, failures and decommissioning events in the log
- IO bandwidth needed for each day's redundancy management is computed as the sum of IO for failure reconstruction and transition IO requested, and is reported as a fraction of the configured cluster IO bandwidth (100MB/sec per disk, by default)

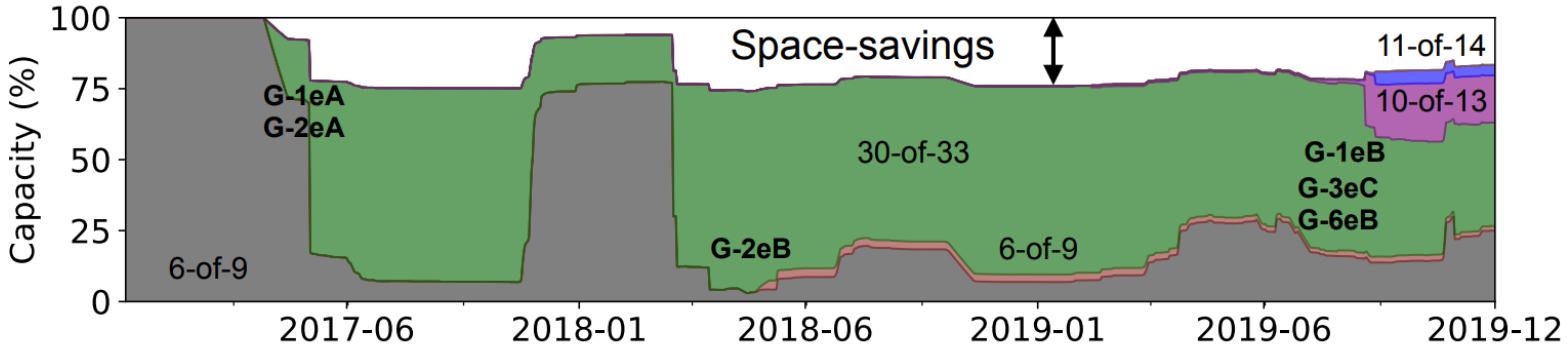
Google Cluster1 Evaluation



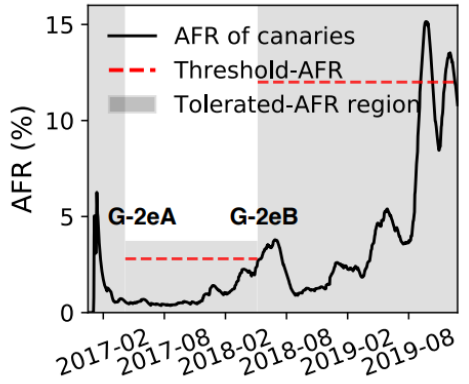
(a) Redundancy management IO due to PACEMAKER over its 2.5+ year lifetime broken down by IO type. This identical to Fig. 1b with the left Y axis only going to 20% to show the detailed IO activity happening in the cluster.



(b) G-1 (step) AFR curve



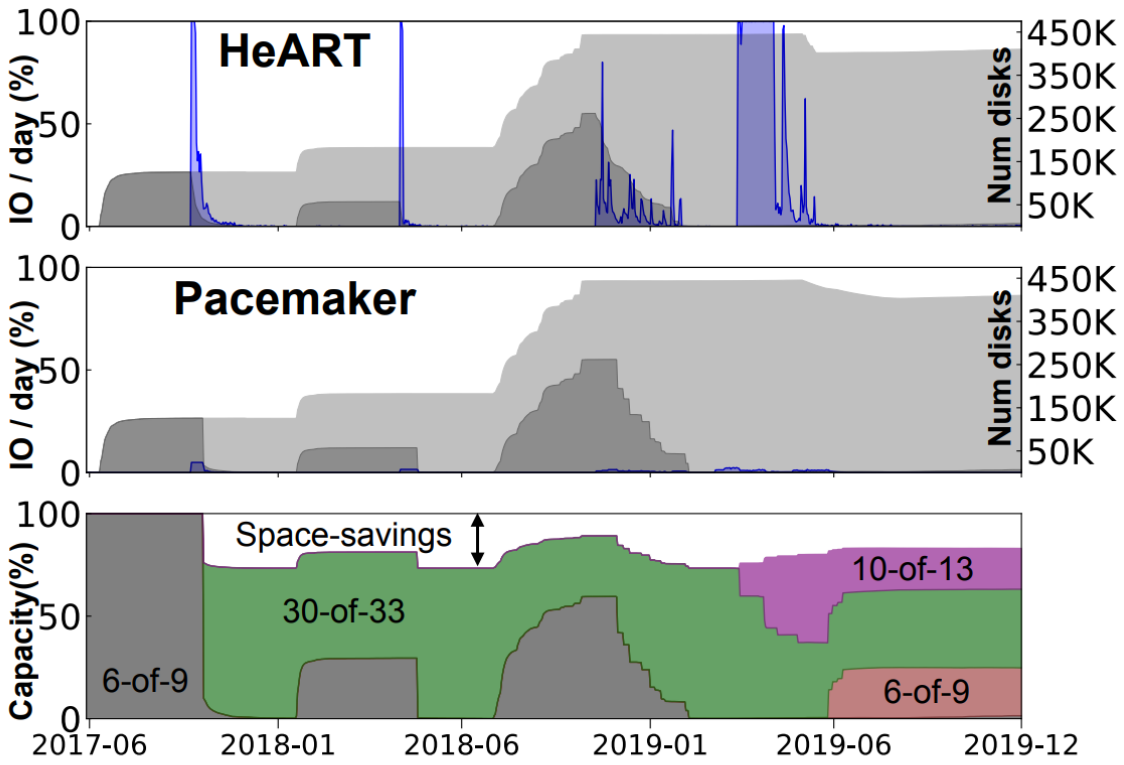
(c) Space-savings due to PACEMAKER. Each colored region represents the fraction of cluster capacity that is using a particular redundancy scheme. 6-of-9 is the default redundancy scheme (Rgroup0's).



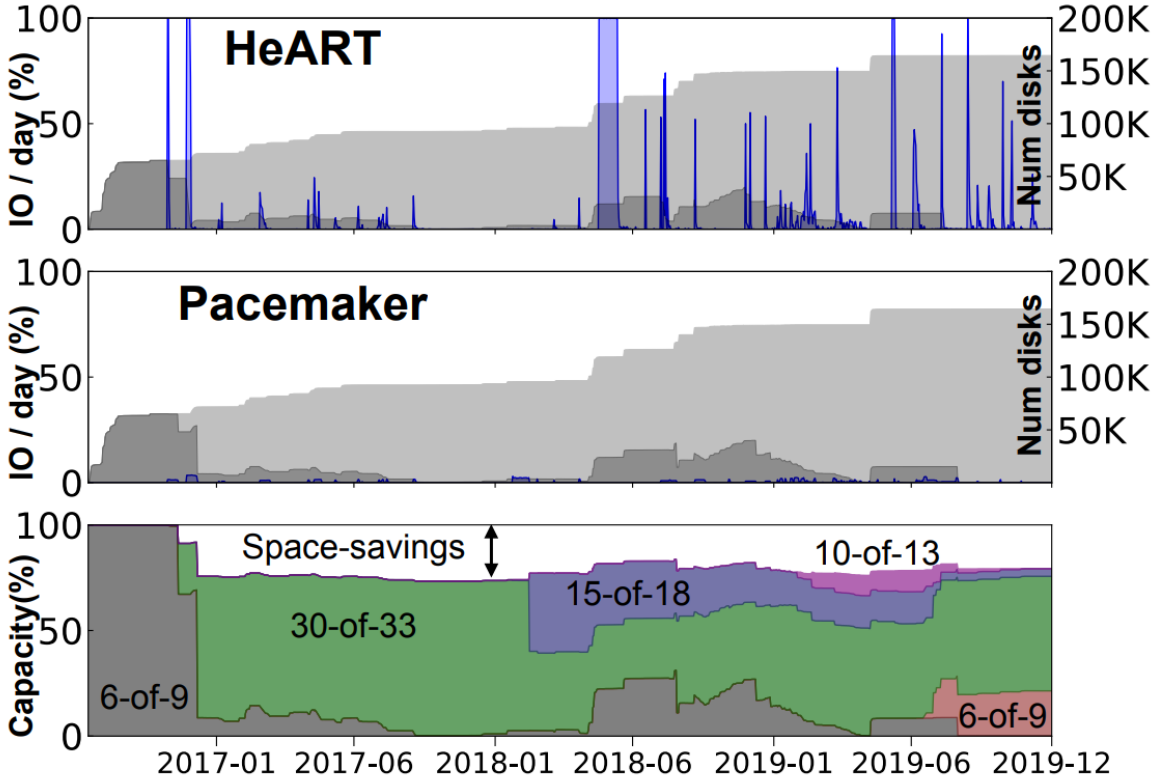
(d) G-2 (trickle) AFR curve

Figure 5: Detailed IO analysis and space savings achieved by PACEMAKER-enabled adaptive redundancy on Google Cluster1.

Other Evaluation

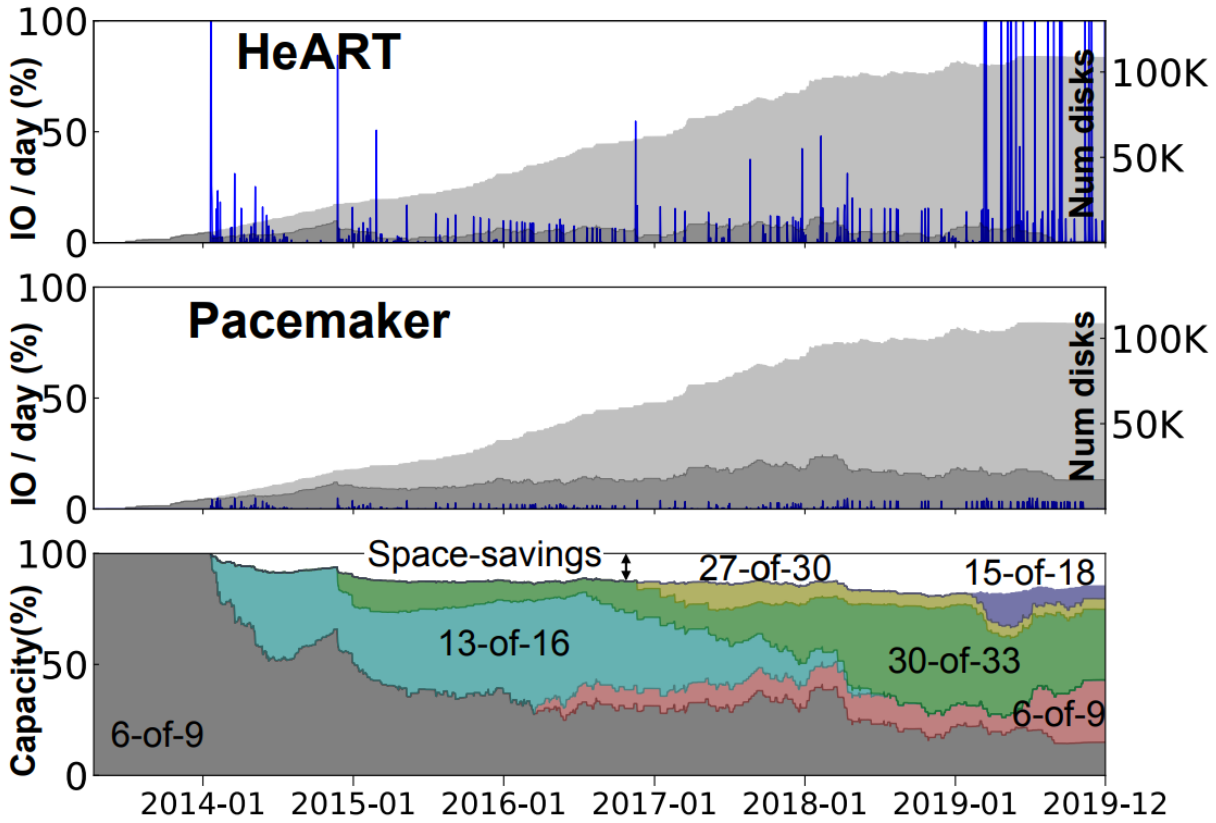


(a) Google Cluster2



(b) Google Cluster3

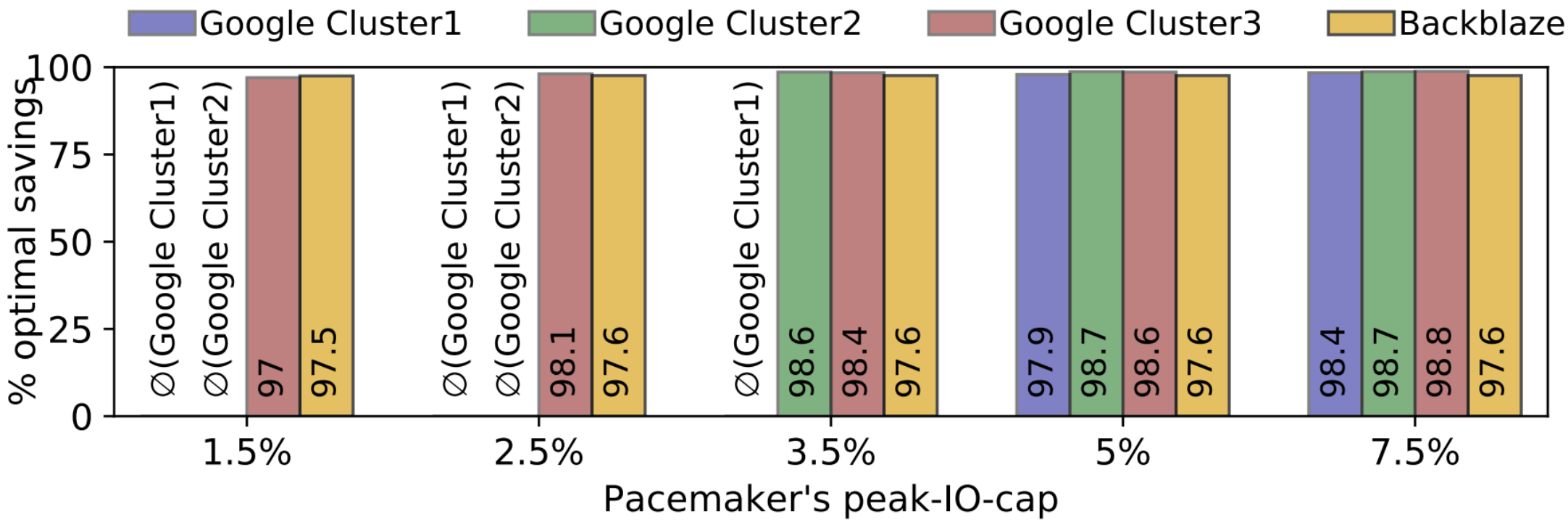
Other Evaluation



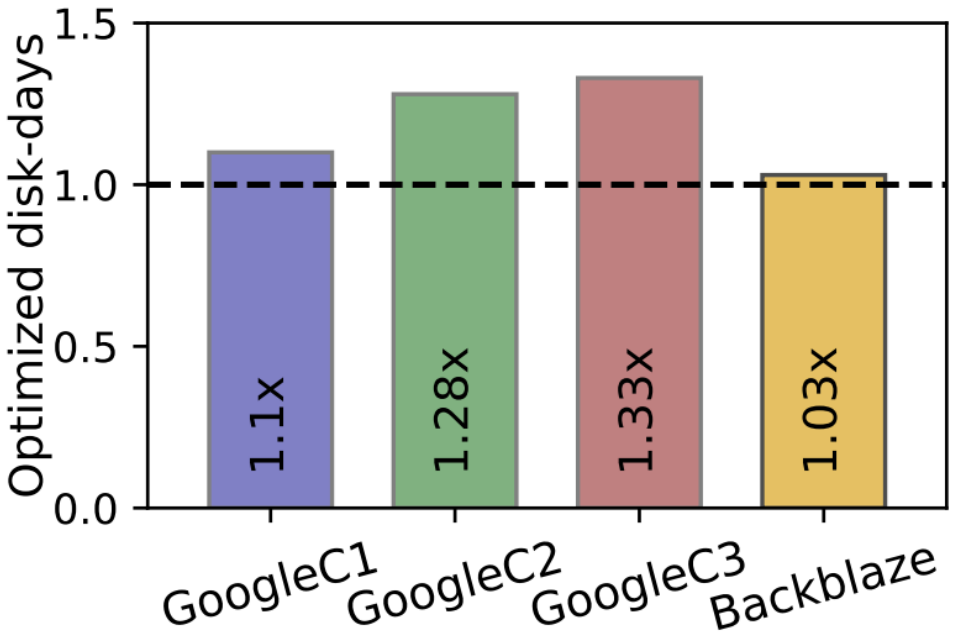
(c) Backblaze

Ablation studies

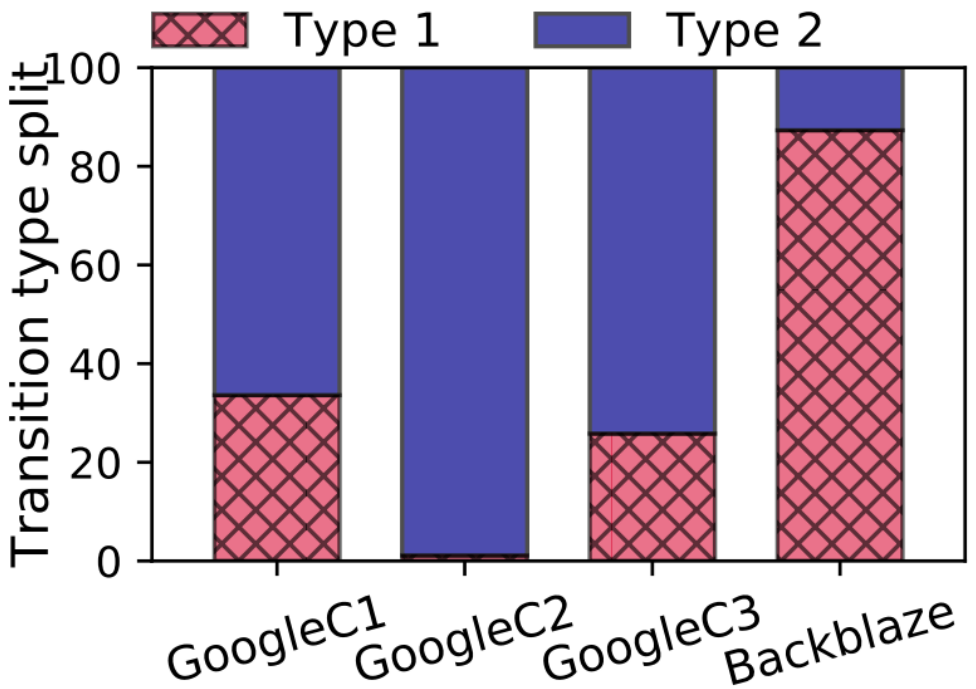
Sensitivity to IO constraints



(a) PACEMAKER's sensitivity to the peak-IO constraint.



(b) Multiple useful life phases



(c) Transition type distribution

Evaluation for HDFS

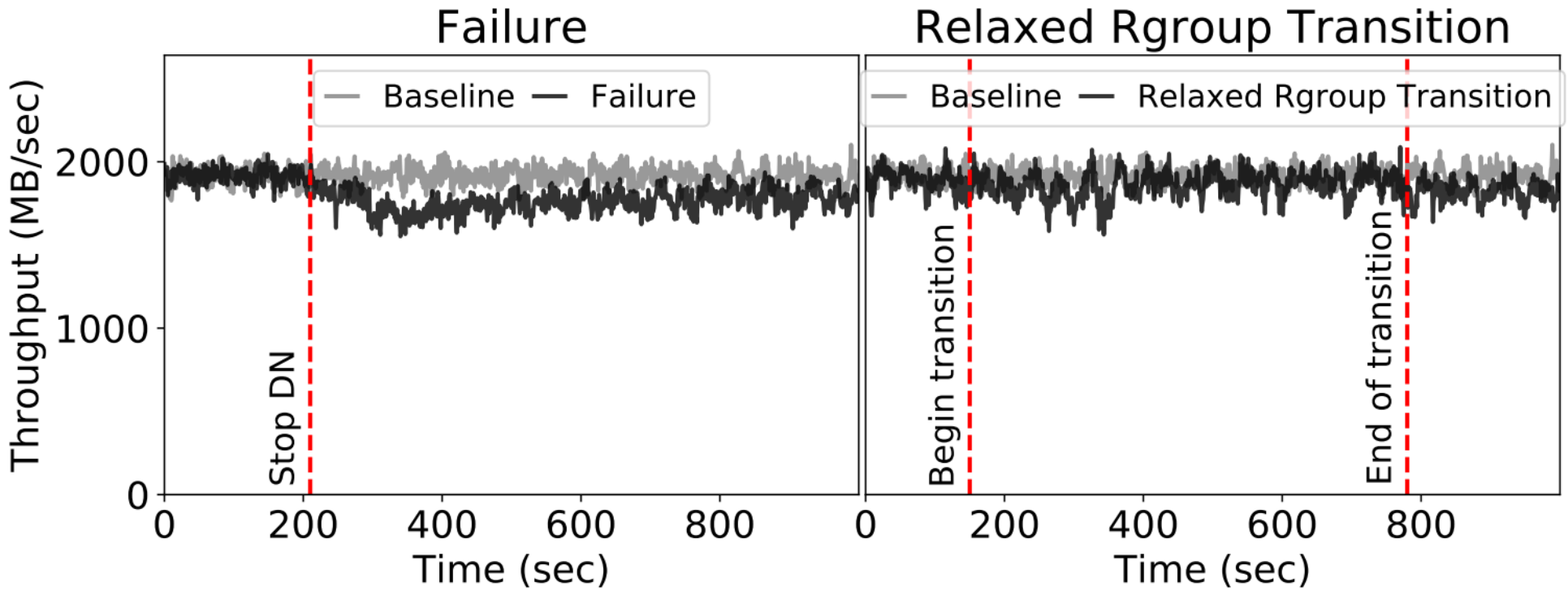


Figure 8: DFS-perf reported throughput for baseline, with one DN failure and one Rgroup transition.

- Strengthens:
  - PACEMAKER performed well, solved the problem entirely, and achieved a balance between space-saving and preventing transition overload.
  - Because multiple phases of disk life are determined and used, PACEMAKER contributed more space-saving than past design.
- Weaknesses:
  - This design is only for the case where storage clusters are designed for a single dedicated storage service, consisting of a large number of disks. It can't be used on multiple different distributed storage services or a smaller number of disks.
  - PACEMAKER needs a better way to set parameters, such as threshold-AFR and peak-IO constraints, to prevent possible failure.
- Comments:
  - The research in this article is based on the author's previous research. In fact, it solves the remaining problems that caused the previous design to be unable to be applied in the actual production environment. Although it does solve the problem perfectly, The new design is not so much brand new as it is a supplement to the old design.

Thank You