MAPX: Controlled Data Migration in the Expansion of Decentralized Object-Based Storage Systems

Li Wang
laurence.liwang@gmail.com
Didi Chuxing

Yiming Zhang
sdiris@gmail.com
(Corresponding)
NiceX Lab, NUDT

Jiawei Xu
titan_xjw@cs.sjtu.edu.cn
SJTU

Guangtao Xue
xue-gt@cs.sjtu.edu.cn
SJTU

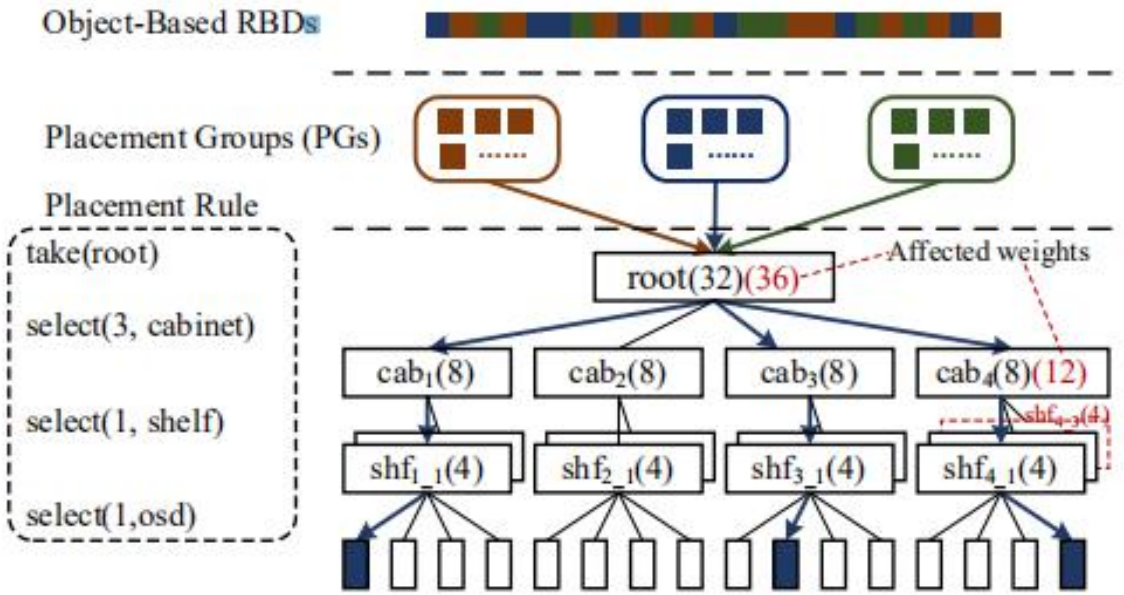
FAST 2020

Background

➢ Object-based storage systems

- view various datas as different objects
- data placement scheme: Centralized VS Decentralized

➢ CRUSH



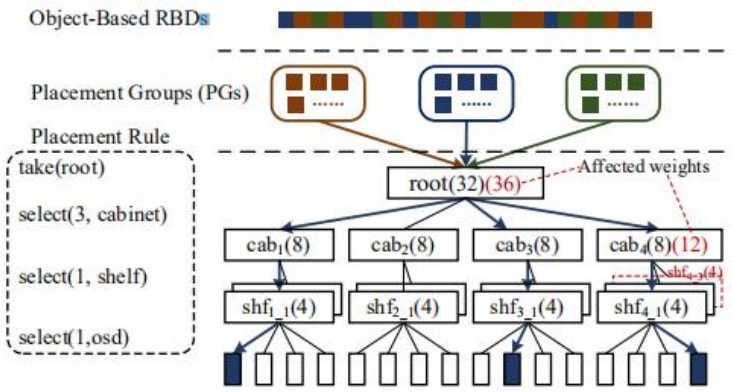
➢ **Two steps:**

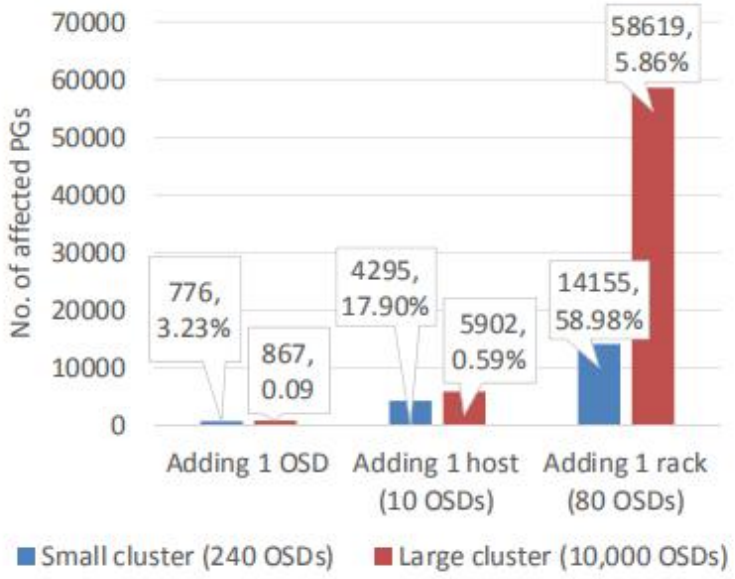
- **Object to PG**
- **PG to OSD**

Background

➢**CRUSH**

- high scalability, robustness, and performance
- uncontrolled data migration after expanding the clusters causes significant performance degradation





Motivation

➢ **Improving performance while enjoying CRUSH's advantage**

- Based on CRUSH

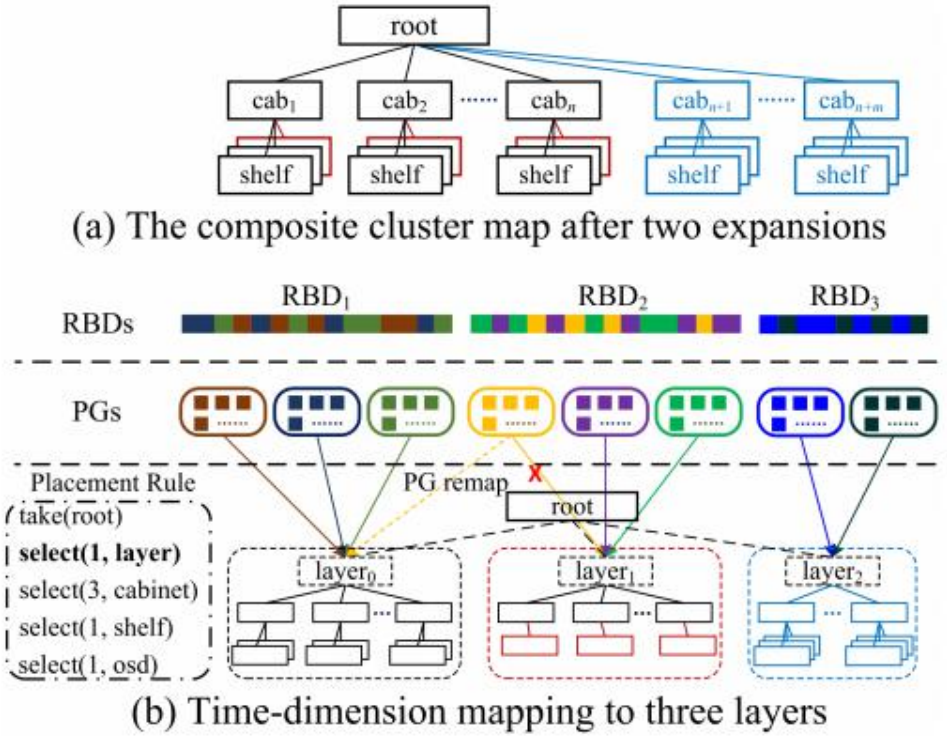
- centralized placement methods

— to keep existing objects unaffected during expansions and place only new objects onto the newly-added OSDs

Main Idea

➢Layer

- Each expansion is viewed as a new layer of the CRUSH map
- Layer is represented by a virtual node beneath the CRUSH root.
- Layer realizes the distinction between new and old clusters.



(a) The composite cluster map after two expansions

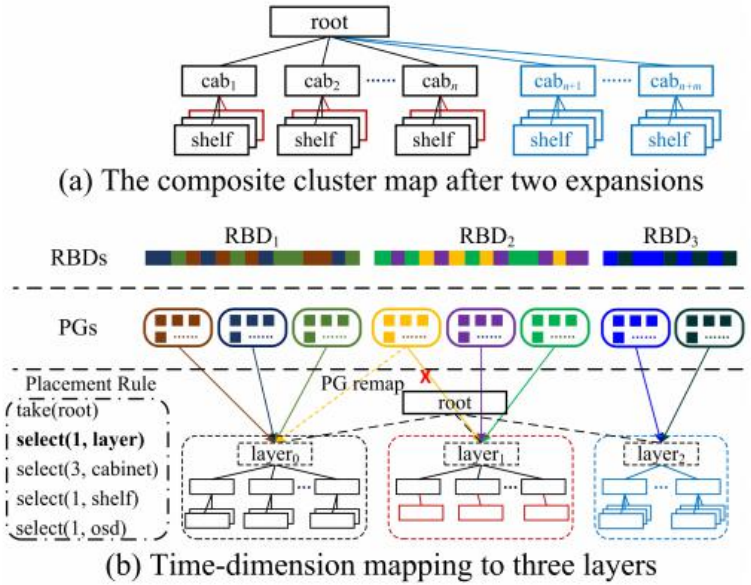
(b) Time-dimension mapping to three layers

➢Timestamps

- an extra dimensional mapping

Macro design

- **Migration-Free Expansion**
  - mainly application scenario,provides load balancing within each layer
- **Migration Control**
  - removals of objects, failures of OSDs, or workload changes……

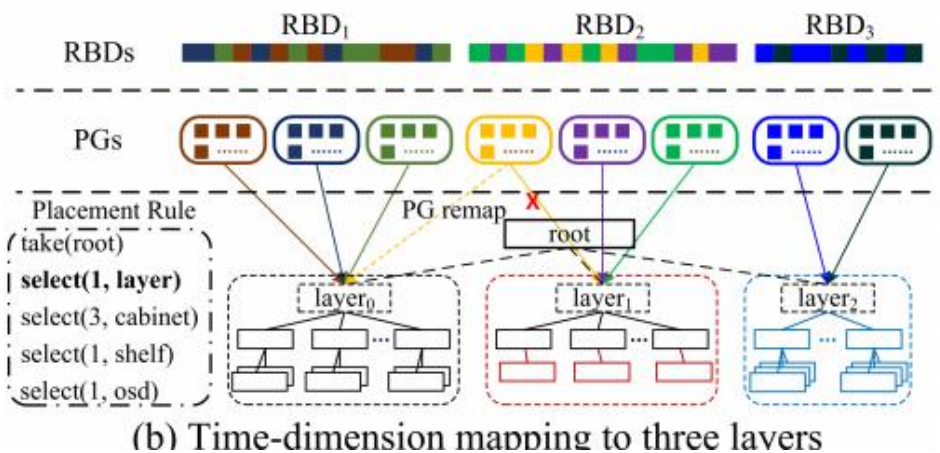


Migration-Free Expansion

➢ Mapping objects to PGs

- The new layer is assigned with a certain number of newly-created PGs
- Each Pg has a timestamp (tpgs) equal to the layer's expansion time (tl)
- Write/Read :Compute object's pgid

The calculation of the pgid is simply a modulo of the number of PGs in the layer by the HASH value + the conversion of the total number of PGs corresponding to all previous layers



(b) Time-dimension mapping to three layers

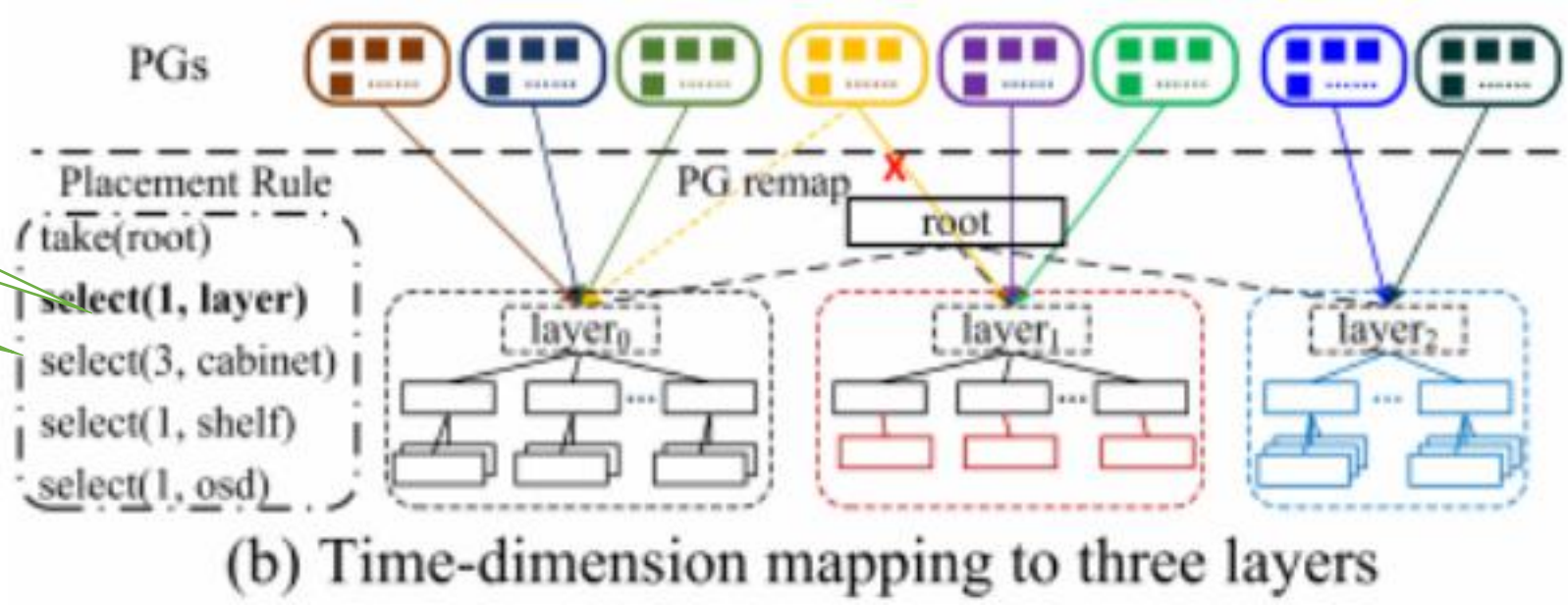
Migration-Free Expansion

➢ Mapping PGs to OSDs

- Similar to CRUSH, MAPX maps a PG onto a list of OSDs following a sequence of operations in a user-defined placement rule
- Note:

Algorithm 1,timestamp

if it has two cabs in the new layer



(b) Time-dimension mapping to three layers

Migration Control

➢PG remapping

- two timestamps :

  a static timestamp (tpgs) 、 a dynamic timestamp (tpgd)

- tpgd that could be set to any layer’s expansion time

➢Cluster shrinking

- view removing the layer’s devices as an inverse operation of expansions
- PG remapping

➢Layer merging

- just change layer's expansion time.

Implement

- **Applicable scene**
  - not suitable for general object stores,but a large variety of object based storage systems because of timestamp
- **Ceph-RBD**
  - the metadata-based timestamp retrieval mechanism: add timestamp in the rbd_header structure
- **CephFS**
  - through it's inode

Evaluation

➢ Major verification issues

- performance compared with CRUSH

➢ Environment

- hardware: dual 20-core Xeon E5-2630 2.20GHz CPU、128G RAM、10GbE NIC、5.5 TB HDDs
- software: three machines run the Ceph OSD storage servers and one runs the client.

OS：CentOS 7.0 Ceph：12.2 Luminous，BlueStore，Monitor co-located with one of the storage servers Client：fio benchmark

➢ Other

- OSD_max_backfills:characterize migration priority

Evaluation

➢Three storage machines, two OSDs per machine, three copies, 128 PGs, adding one or two OSDs to each machine, and then test the corresponding performance under IMAP and CRUSH (I/O latency and IOPS), the I/O size is 4KB, and the IOdepth of FIO is 1 and 128, corresponding to latency and IOPS test

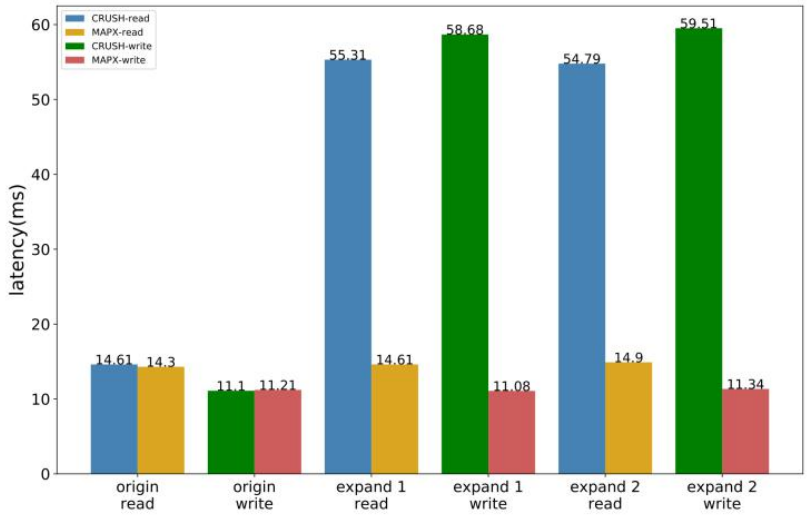


Figure 4: 99th percentile I/O latency of MAPX and CRUSH (during cluster expansions).

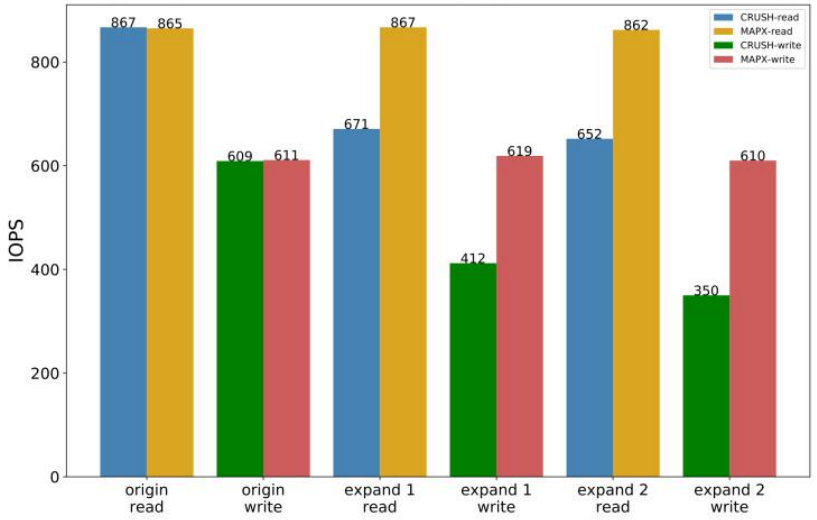


Figure 5: IOPS of MAPX and CRUSH (during cluster expansions).

➢**MAPX has better performance in expansion read/write.**

Evaluation

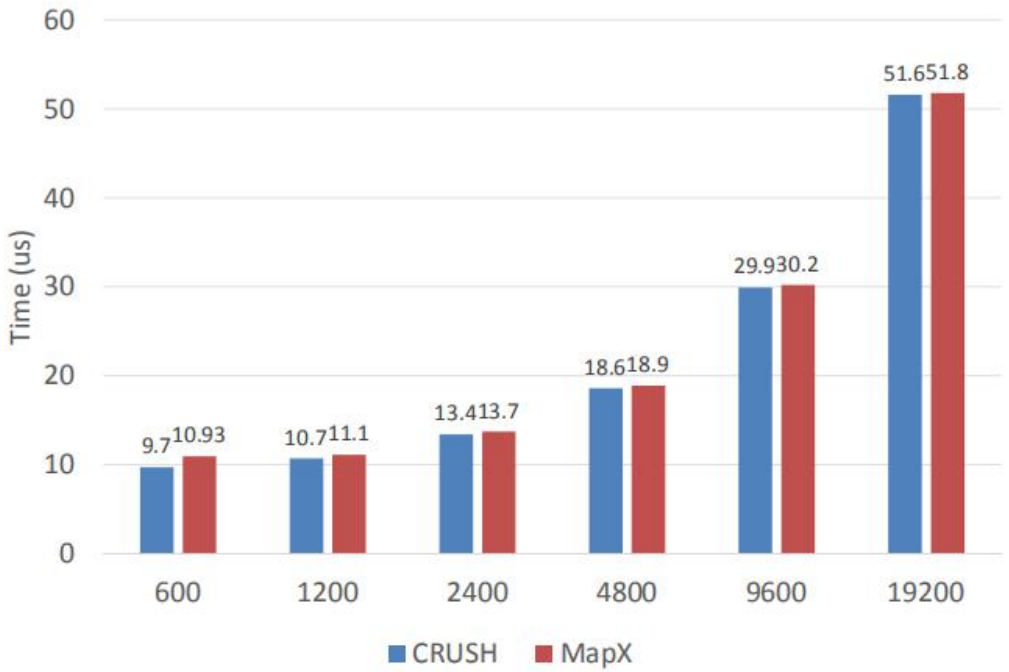


Figure 6: Computation overhead of MAPX and CRUSH.

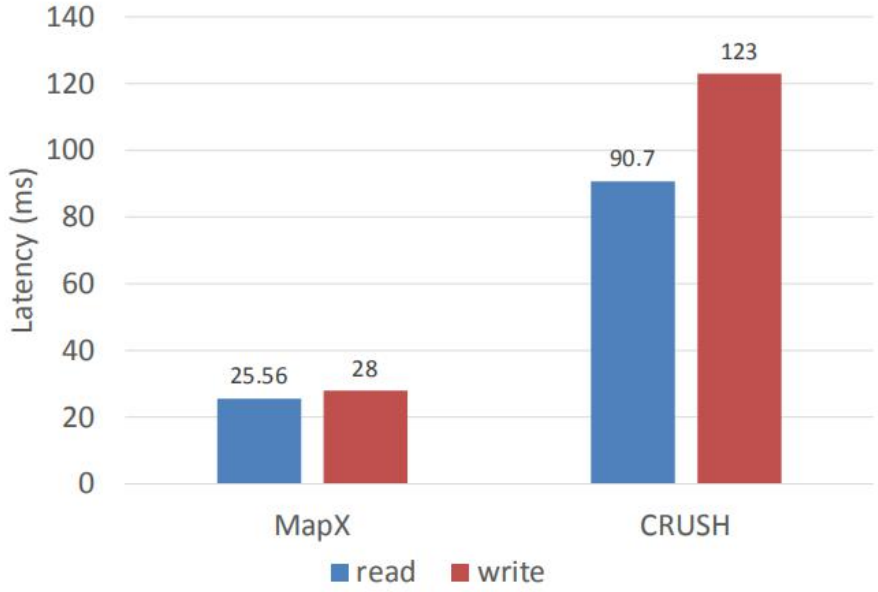


Figure 7: 99th percentile I/O latency of MAPX and CRUSH (during cluster shrinking).

- **a Ceph cluster of different numbers of OSDs**
- **the higher comes from the computation of the time-dimension mapping beneath the root**
- **Three machines, three OSDs per machine, first adding an OSD to each machine, then subtracting an OSD from each machine, setting the number of concurrently migrated PGs to 8 for controling the migration speed.**

Evaluation

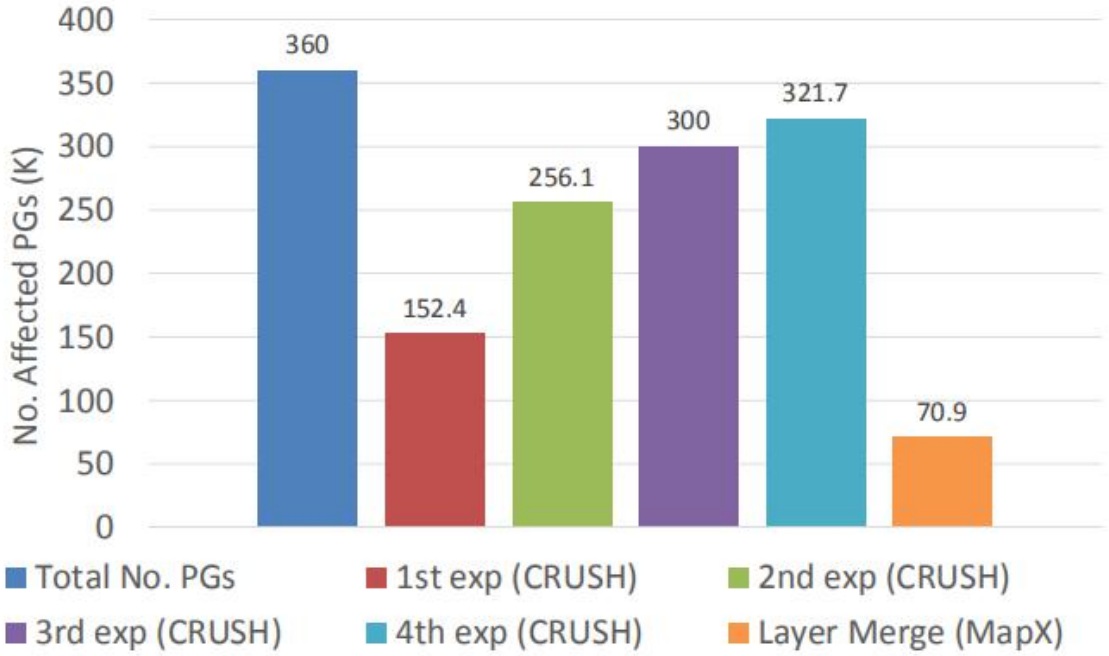


Figure 8: Number of affected PGs in layer merging in MAPX (after four expansions). Since CRUSH does not support merging, for reference we measure the number of affected PGs after each expansion in CRUSH.

- **Initially the storage cluster consists of 5 racks each having 20 machines. One machine has 20 OSDs. There are totally 100 machines and 2000 OSDs, storing 200,000 PGs.**
- **adding a new layer of one rack everytime.**
- **The relatively high ratio of affected PGs in layer merging of MAPX is decided by the nature of CRUSH**

Conclusion

- **MAPX:a novel extension to CRUSH that embraces the best of both decentralized and centralized methods**