Hosted at Santa Clara University School of Engineering

33rd International Conference on Massive Storage Systems and Technology (MSST 2017)
May 15 — 19, 2017
Sponsored by Santa Clara University,

Technically Co-Sponsored by IEEE computer society

Hosted at Santa Clara University

Content-aware Trace Collection and I/O Deduplication for Smartphones

Bo Mao[1], Suzhen Wu[1], Hong Jiang[2], Xiao Chen[1], Weijian Yang[1]

{maobo, suzhen}@xmu.edu.cn, hong.jiang@uta.edu

[1] Xiamen University (http://astl.xmu.edu.cn/)

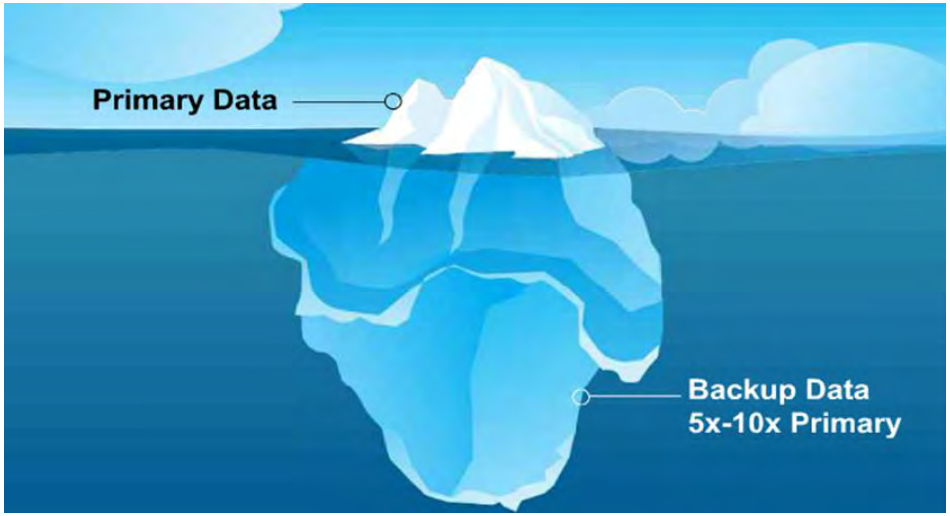
[2] University of Texas at Arlington

Outline

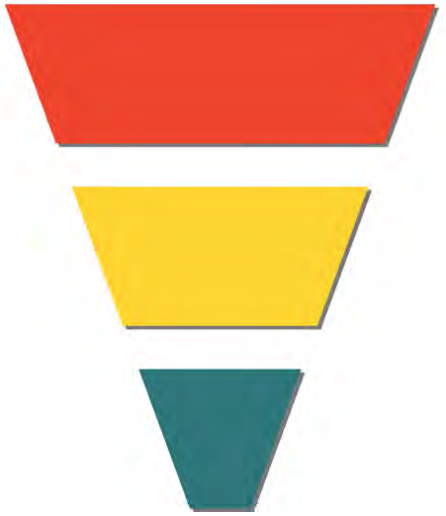
- Introduction and challenges
- Trace collection and observations
- System overview and design
- Performance evaluations
- Conclusion

Why data deduplication?

- Backup media changed from Tape to HDDs



Capacity

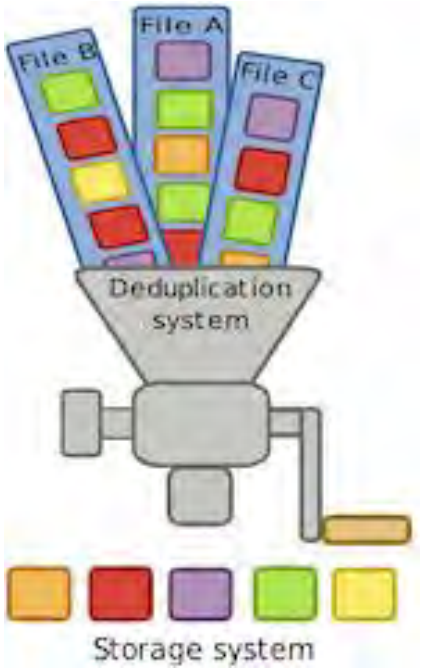


Cost

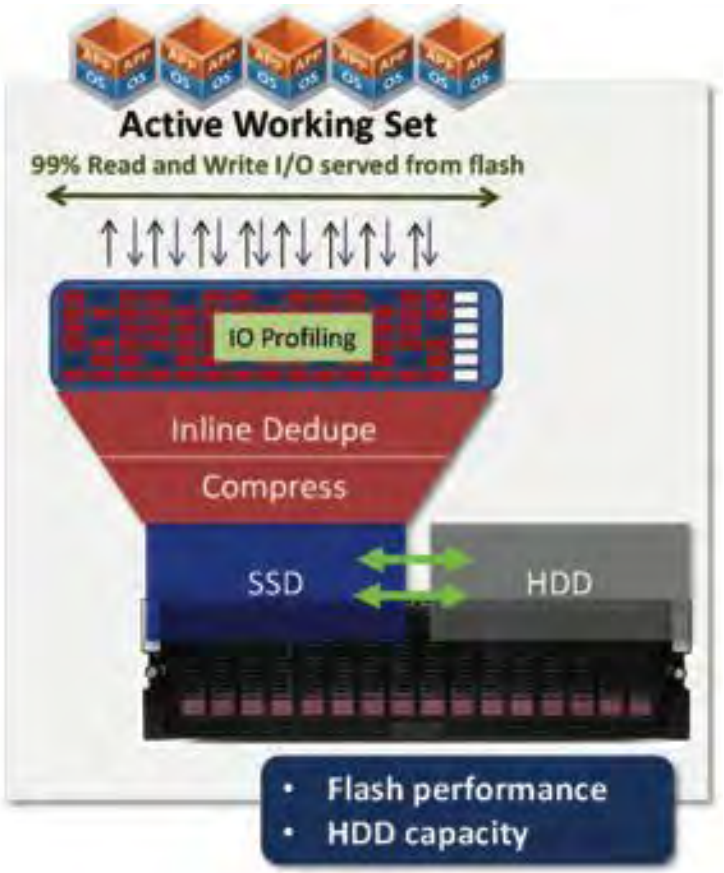
- In primary storage systems, it is also important to shrink the data volume.

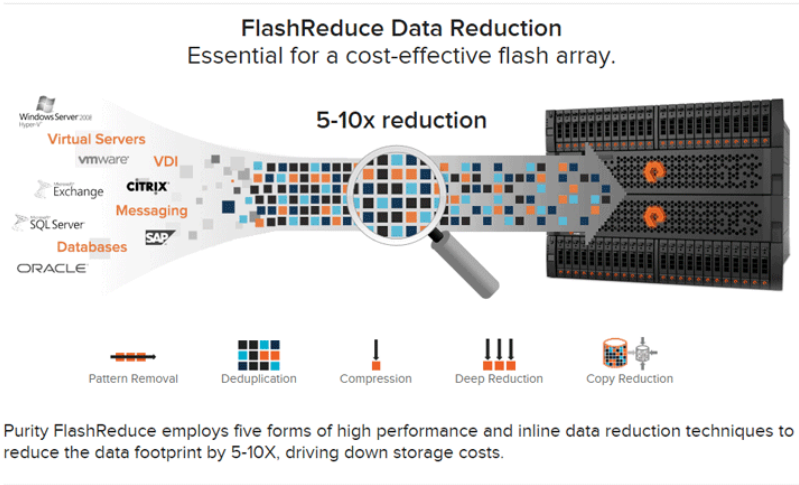
Data deduplication

- Data deduplication is widely deployed in secondary storage systems to:
  - ✓ Reduce backup time
  - ✓ Improve storage space efficiency
  - ✓ Improve network bandwidth
  - ✓ ……
- In primary storage systems:
  - ✓ VM-based storage systems (Linux KSM)
  - ✓ Flash storage products (Nimble storage, Tintri, Pure Storage …)
  - ✓ ……



Deduplication + Flash

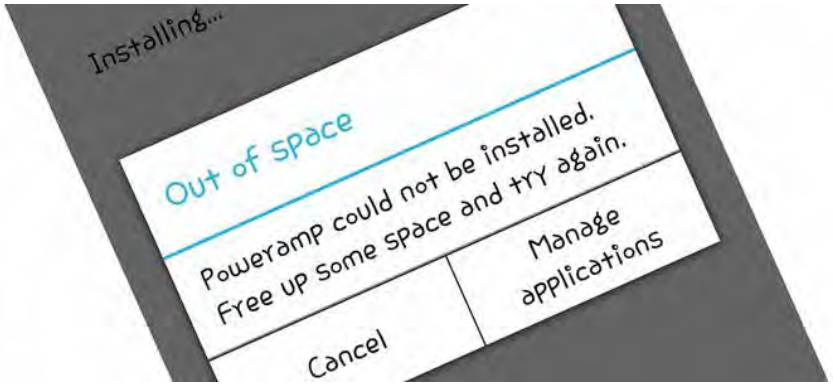


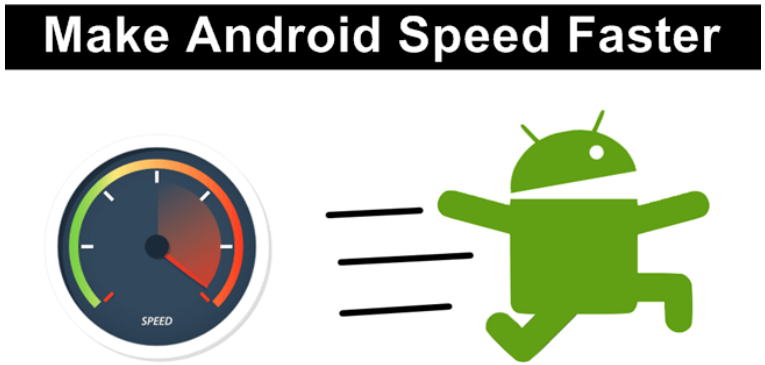


- CA-FTL (FAST'11)
- CA-SSD (FAST'11)

Deduplication has become an important feature for flash-based storage!

Flash within Smartphones





- Flash (eMMC or UFS) in Smartphones:
  - ✓ Performance tends to degrade after repeated usages.
  - ✓ Limited life cycles affect the Smartphones' reliability.
  - ✓ The cost of upgrading flash capacity is high.

How about applying data deduplication on flash storage within Smartphones?

Workflow of data deduplication

Fixed chunk, CDC, FastCDC…

Write data → Data Chunking (Method, Size et. al)

StoreGPU, Shredder …

Hash Computing (GPU, Multicore et. al)

DDFS, ChunkStash, SiLo …

Hash Index (Disk-Index, Structure et. al)

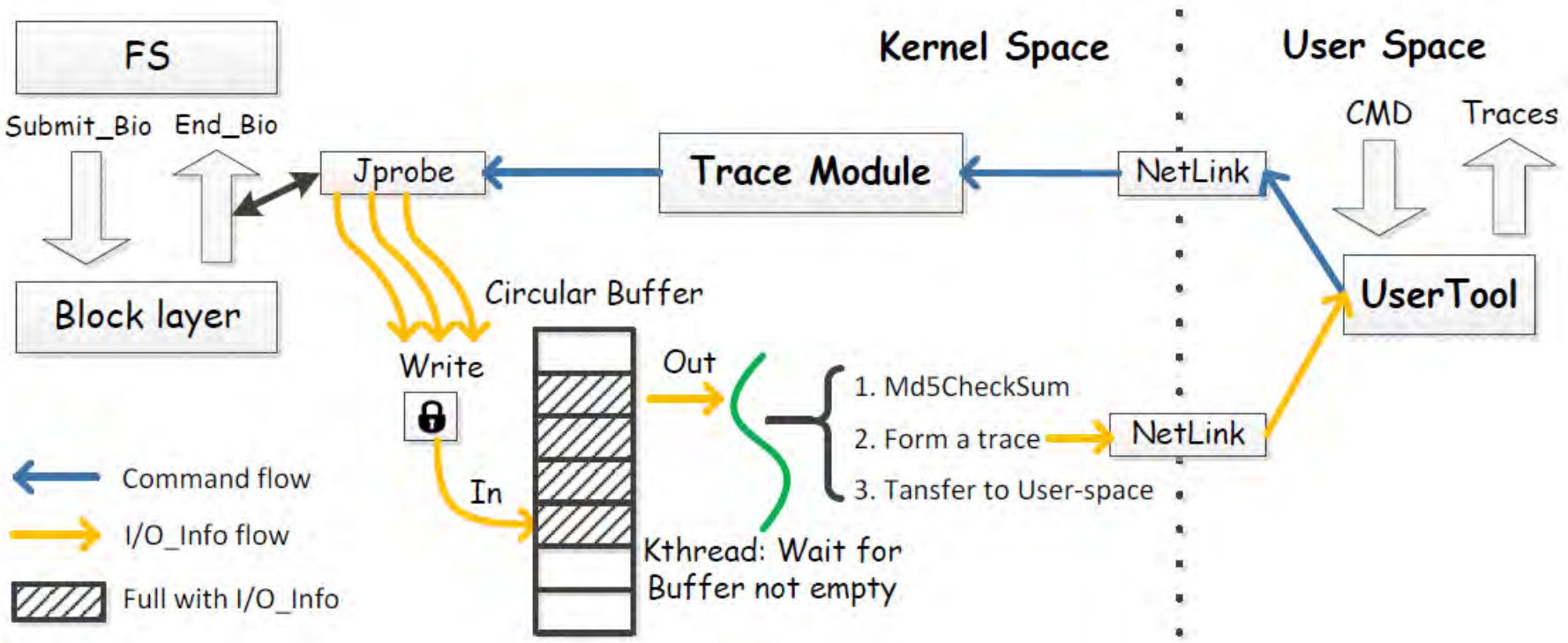
Index and Metadata Update (Update, Distribution et. al) → Devices

CPU and Memory Overhead!

Challenges

- Resources in Smartphones:
  - CPU utilizations affect power consumption.
  - Limited memory capacity.
  - Mobile APP usages.
- Is data deduplication feasible and how to?
  - How to investigate the redundancy within Smartphones?
  - How much data redundancy in mobile APPs?
  - How to design a lightweight data deduplication engine?

Content-aware trace collection



Obs 1: Redundancy characteristics*

APPs	Qiu	58City	Baidu	Game2048	Meitu	Moji
Qiu	**40.5%**	0.2%	1.0%	0.1%	0.3%	0.0%
58City	0.3%	**32.2%**	3.3%	1.3%	3.6%	3.2%
Baidu	0.4%	6.5%	**39.5%**	0.2%	2.4%	5.5%
Game2048	0.3%	0.7%	1.0%	**31.7%**	1.0%	2.0%
Meitu	0.3%	3.1%	2.2%	1.3%	**31.3%**	2.2%
Moji	0.1%	2.7%	4.7%	2.8%	2.3%	**32.4%**

- ✓ **Moderate to high data redundancy exists within mobile APPs.**
- ✓ **Amount of data redundancy shared between any two different mobile APPs is minimal[#].**

*Detailed results and analysis for all the 15 mobile APPs can be found in our paper.

[#]Y. Fu, H. Jiang, N. Xiao, L. Tian, F. Liu, AA-Dedupe: An Application-Aware Source Deduplication Approach for Cloud Backup Services in the Personal Computing Environment, in Proceedings of IEEE Cluster 2011, Austin, Texas, Sept. 2011.

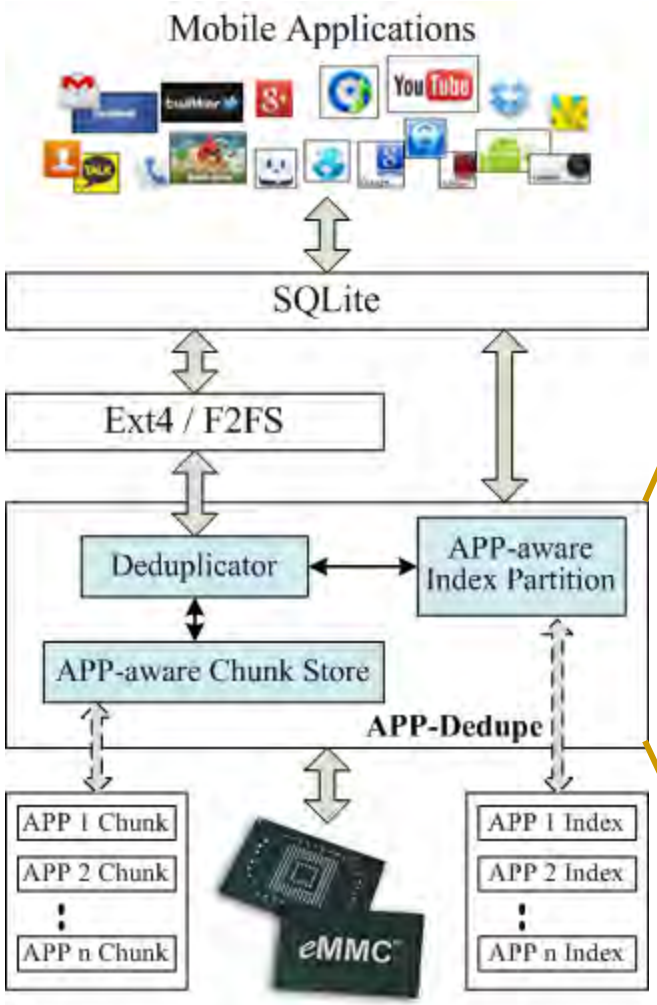
Obs 2: Lower IOPS

APPs	*Redundancy*	*IOPS*	*Size (MB)*	*Write Ratio*
Qiu	40.5%	1.6	55.0	87.9%
58City	32.2%	2.6	153.3	69.4%
Baidu Tieba	39.5%	4.3	213.5	88.6%
Game2048	31.7%	1.7	63.2	89.2%
Meitu	31.3%	5.6	131.6	41.9%
Moji Weather	32.4%	2.6	82.1	63.4%
Opera Browser	33.1%	3.0	72.7	57.5%
Fruit Cool	36.6%	6.0	159.9	47.1%
Sohu News	32.2%	2.0	138.2	84.2%
Tencent	34.1%	1.5	82.6	95.4%
Pea Pod	35.4%	2.1	237.0	67.8%
Wechat	30.0%	4.6	345.5	90.4%
Weibo	29.0%	1.4	97.0	88.5%
Xiami Music	27.3%	7.4	122.9	39.4%
Youdao Dict	31.9%	9.2	136.5	35.2%

The I/O intensity is low for most APPs (IOPS)*

*D. Zhou, W. Pan, W. Wang, and T. Xie, I/O Characteristics of Smartphone Applications and Their Implications for eMMC Design, in Proceedings of IISWC 2015, Atlanta,, USA, Oct. 2015.

System overview

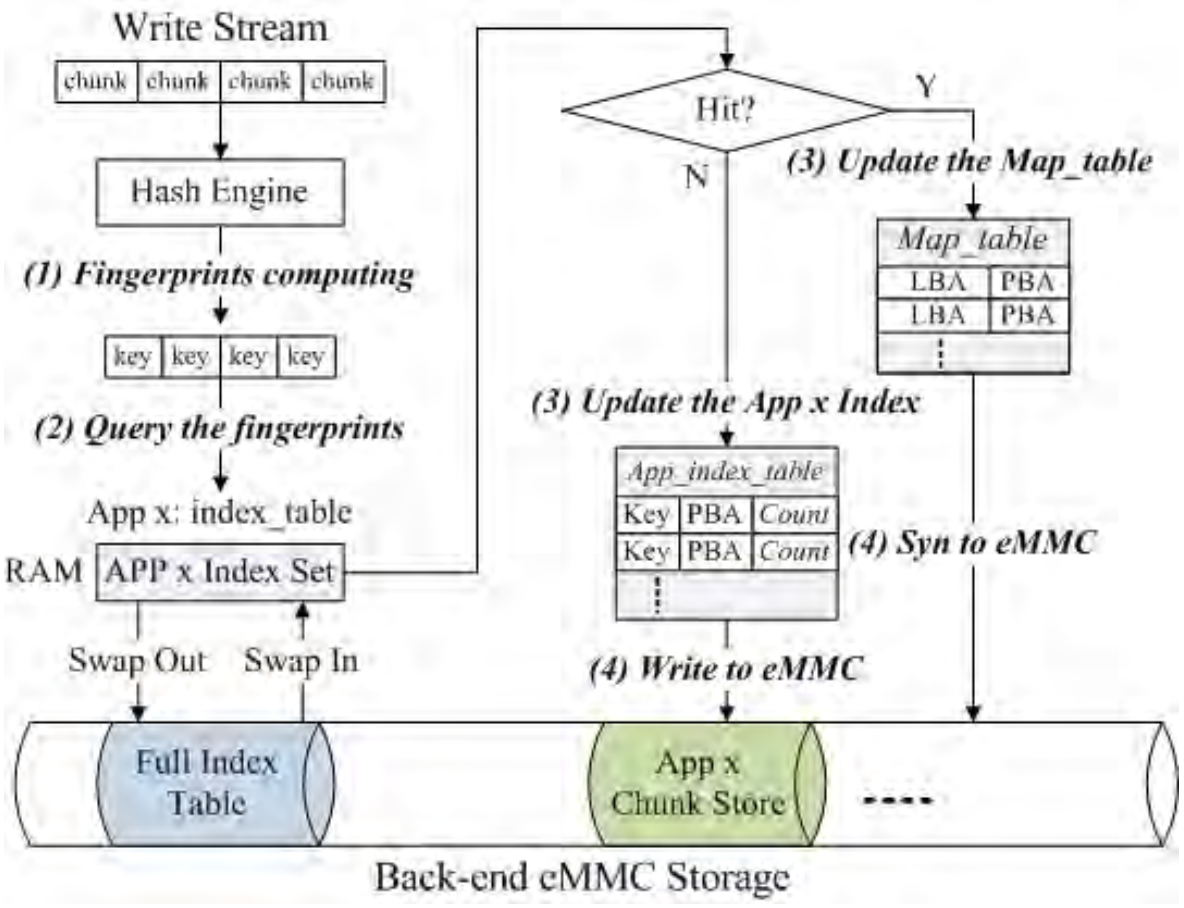


- Independent of upper file systems
- Low overhead design choice:
  - ✓ MD5 hash computing
  - ✓ Fixed chunking (4KB)
- Two optimizations:
  - ✓ Index partition
  - ✓ Chunk store

Design and Optimizations

- APP-aware Index Partition (AIP):
  - → *Memory overhead associated with big hash index table*.
  - ✓ Grouping the hash index according to the APPs.
  - ✓ Swap In/Out between memory and Flash.
- APP-aware Chunk Store (ACS):
  - → *Data fragmentation associated with data deduplication*.
  - ✓ Storing the data chunks according to the APPs (LBAs).
  - ✓ Concentrating the read accesses to a single container.

Write workflow in APP-Dedupe

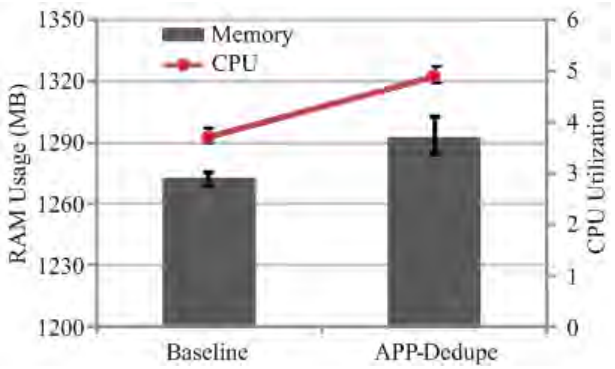


Experimental setup

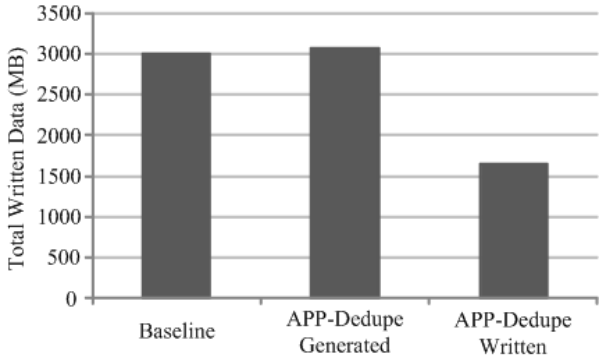
- Google Nexus 5 Smartphone:
  - → *Real system study*.
  - ✓ Qualcomm MSM8974 Quadcore 2.3 GHz, 2 GByte DRAM, 16 GByte eMMC storage.
  - ✓ Android 5.0.1 with Linux Kernel 3.4.
  - ✓ Benchmarks: Monkey tool and A1 SD Bench.
- SSD-based DiskSim simulator:
  - → *Simulation study*.
  - ✓ Replay the traces collected from real system.
  - ✓ Evaluate response time and GC count within flash.

Results and analysis

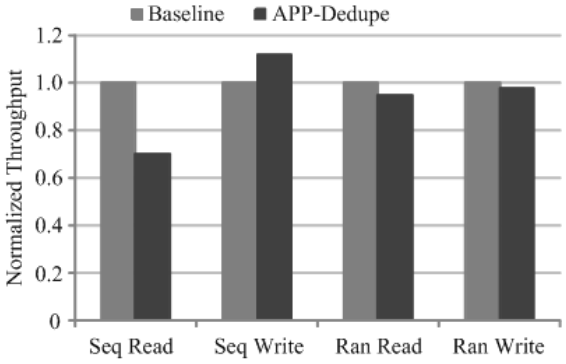
(1) Memory and CPU usages



(2) Total written data



(3) Throughput

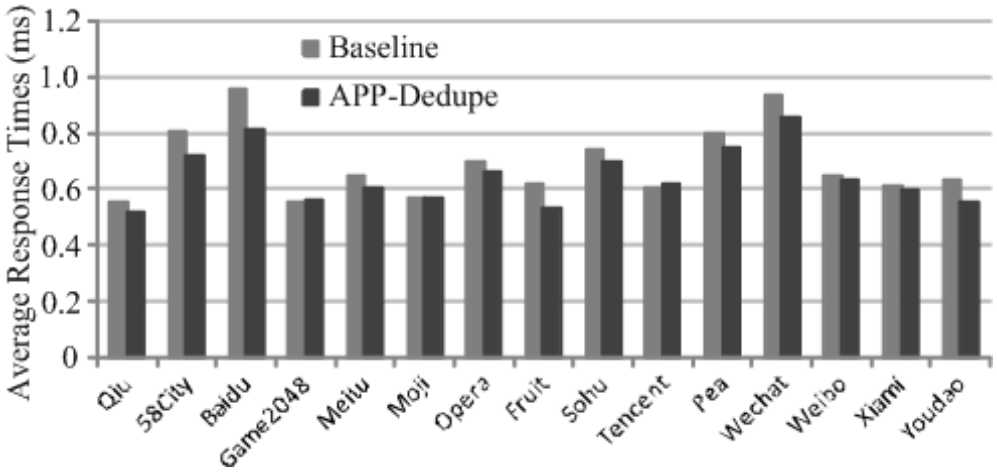


→ APP-Dedupe incurs very little memory and CPU overhead, by less than 3%.

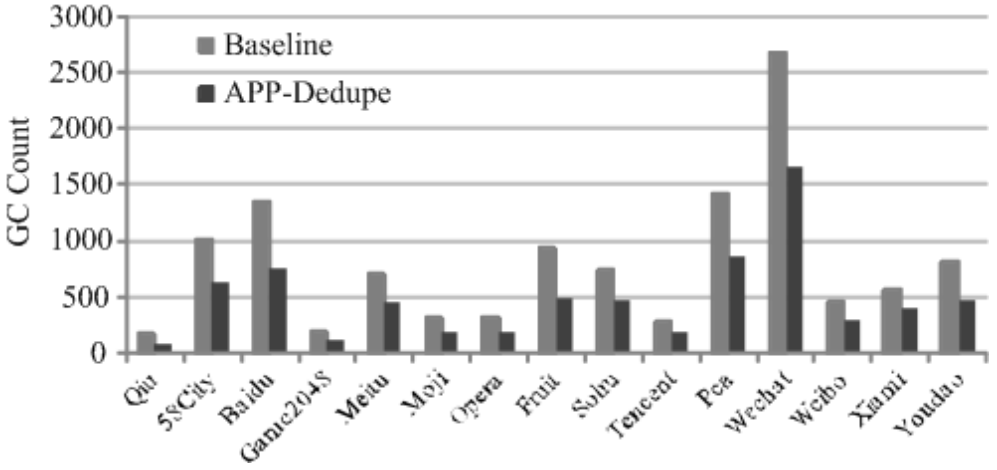
→ APP-Dedupe reduces the amount of write data to the back-end eMMC storage by an average of 45.2%.

→ System throughput performance is complicated.

Results and analysis



By up to 15.4% with an average of 6.2%



By up to 58.7% with an average of 41.5%

Conclusion

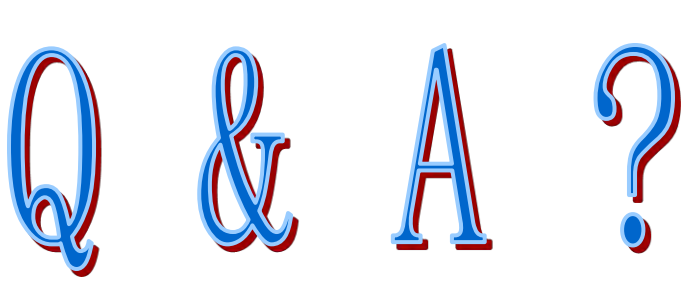
- Performance of the storage subsystem in Smartphones plays an important role in the application performance.
- We investigate the data redundancy characteristics within Smartphones and propose APP-Dedupe that detects and eliminates the I/O redundancy by exploiting the mobile applications' redundancy characteristics.
- APP-Dedupe reduces the GC overhead by an average of 41.5%, reduces the response times by up to 15.4% and saves the storage capacity by an average of 45.2%.

Content-aware Trace Collection and I/O Deduplication for Smartphones

Bo Mao[1]*, Suzhen Wu[1], Hong Jiang[2], Xiao Chen[1], Weijian Yang[1]

[1] Xiamen University (http://astl.xmu.edu.cn/)
[2] University of Texas at Arlington

* Please feel free to contact me: maobo@xmu.edu.cn for any questions!



UNIVERSITY OF TEXAS ARLINGTON