BlockFlex: Enabling Storage Harvesting with Software-Defined Storage*

Benjamin Reidys‡ Jinghan Sun‡ Anirudh Badam† Shadi Noghabi† Jian Huang

University of Illinois at Urbana-Champaign †Microsoft Research

Abstract

Cloud platforms today make efficient use of storage resources by slicing them among multi-tenant applications on demand. However, our study discloses that cloud storage is still seriously underutilized for both allocated and unallocated storage. Although cloud providers have developed harvesting techniques to allow evictable virtual machines (VMs) to use unallocated resources, these techniques cannot be directly applied to storage resources, due to the lack of systematic support for the isolation of space, bandwidth, and data security in storage devices.

In this paper, we present BlockFlex, a learning-based storage harvesting framework, which can harvest available flash-based storage resources at a fine-grained granularity in modern cloud platforms. We rethink the abstractions of storage virtualization and enable transparent harvesting of both allocated and unallocated storage for evictable VMs. BlockFlex explores both heuristics and learning-based approaches to maximize the storage utilization, while ensuring the performance and security isolation between regular and evictable VMs at the storage device level. We develop BlockFlex with programmable solid-state drives (SSDs) and demonstrate its efficiency with various datacenter workloads.

1 Background and Motivation

In modern cloud platforms, storage devices such as flash-based solid-state drives (SSDs) have been virtualized as system-wide shared resources to provide storage services across multiple application instances [2, 4]. This enables efficient use of storage capacity and bandwidth by slicing them among multi-tenant virtual machines (VMs). However, our study of the event traces collected from popular cloud platforms [1, 3] reveals that storage I/O is still significantly underutilized for both unallocated (unsold) and allocated storage.

To improve resource efficiency in the cloud, providers offer evictable VMs (i.e., Spot VMs or Harvest VMs). These evictable VMs allow users to use unallocated resources with low priority, i.e., the resources of evictable VMs can be reclaimed by regular VMs at any time. Recent studies [2] advanced this technique by improving the resource allocation and scheduling for evictable VMs with heuristic-based harvesting approaches.

However, prior work on resource harvesting mainly focused on CPU and memory resources, which cannot be directly applied to cloud storage for three reasons. First, current cloud storage virtualization approaches do not support storage harvesting, and dynamic reallocation of resources is not feasible. Second, cloud storage usually stores sensitive application data, which requires careful management for storage allocation and deallocation. Third, cloud storage can suffer from significant harvesting overhead due to the block erasure and metadata updates, which requires specific optimizations.

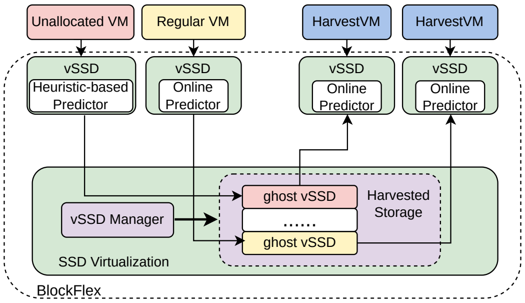

Figure 1: System overview of BlockFlex.

2 Characterization of Storage Harvesting

We first conduct a characterization study of storage resources that could be harvested in cloud platforms. Although storage virtualization is widely deployed in cloud platforms, we observe that storage devices are still significantly underutilized for allocated and unallocated storage resources.

Allocated storage resources. We conduct the storage utilization study based on the open-source cloud traces from Google [3]. The cumulative distribution of storage capacity across the VMs of Google Cloud [3] (see Baseline in Figure 2). We find that 20% of the VMs almost did not use their allocated storage capacity, 50% of the VMs used only 26.4% of the allocated storage capacity on average. Although different VMs may allocate different storage capacities, our study shows that their capacity utilization is surprisingly low.

The low utilization of allocated cloud storage resources is mainly due to two reasons. First, cloud platforms usually allocate storage resource associated with each VM at a coarse-grained granularity for simplified storage management. Second, storage allocation is usually conducted in a static manner, while the storage usage of the workloads running in each VM changes dynamically. Therefore, the user of a VM has to over-provision sufficient storage for the peak demand upon VM creation.

Unallocated (unsold) storage resource. Unallocated (unsold) storage in cloud platforms is another source of storage underutilization. This is because cloud providers usually over-provision VMs in their resource pool to satisfy the elasticity requirement from customers [2]. As each unsold VM consumes a fixed amount of resources (e.g., processor cores, memory, and storage), it will result in unallocated storage resources.

To further understand unallocated storage, we analyze traces of unsold storage resources from Azure [2]. Nearly 70% of cloud servers have unsold storage resources, 50% of the servers have an average of 17.3% of their storage unallocated. Given that a datacenter has thousands of servers, the unallocated storage is another critical source of storage underutilization.

3 Design and Implementation

To overcome storage underutilization, we present BlockFlex (Figure 1), which enables transparent and fine-grained storage

* This work has been published at OSDI 2022 [5].

‡ Co-primary Authors.

Table 1: Exception handling for different scenarios.

ID	Harvestable Storage	Demanded Storage	Possible Exceptions
❶	Over-predict	Over-predict	Waste or Early Reclamation or N/A
❷	Over-predict	Under-predict	Under-Harvest or Early Reclamation
❸	Under-predict	Over-predict	Waste
❹	Under-predict	Under-predict	Under-Harvest or Waste or N/A

harvesting for both allocated and unallocated storage while ensuring data privacy for users with low harvesting overhead.

New Abstraction for Storage Harvesting. To enable transparent and fine-grained storage harvesting, we rethink the abstractions of storage virtualization for flash-based SSDs. The recent development of software-defined flash (SDF) in datacenters [4] allows VMs to map their storage to dedicated flash channels. We build on top of the SDF abstraction and propose a new class of virtualized SSDs (vSSD), named *ghost vSSD* (gSSD). A gSSD is created by harvesting free flash blocks from either unallocated or allocated but unused storage. Its block interface is the same as that of the regular vSSD. Similar to vSSDs, each gSSD has a block-level mapping table to index the mappings of logical block addresses to physical block addresses, and a free block list to manage the free flash blocks.

Management of the gSSDs. BlockFlex manages the lifecycle of each gSSD with a gSSD pool. BlockFlex supports the following operations: (1) *Creation*: A vSSD creates a gSSD when its predictor predicts that it will have available storage resources for harvesting. In order to create a new gSSD, BlockFlex will harvest free blocks from the vSSD and create a mapping table for them; (2) *Lookup*: To facilitate fast gSSD lookup, we organize gSSDs in a set of lists in the vSSD manager with considering the sorting in three dimensions: storage bandwidth, capacity, and time available for harvesting; (3) *Expiration*: For the expired gSSDs that have not been allocated to any harvest VM, BlockFlex will remove them from the list; (4) *Harvesting*: Upon receiving a request for storage harvesting, BlockFlex will check the gSSD pool to identify a best-fit match for the requested storage capacity, bandwidth, and time available for harvesting. BlockFlex uses the best-fit matching policy to minimize the waste of storage resources. These requested parameters are obtained from the predictors deployed in the vSSD of the corresponding harvest VM; (5) *Reclamation*: When a harvest VM finishes its jobs, the harvested gSSDs will be reclaimed to the gSSD pool. Upon gSSD reclamation, the corresponding entries in the address mapping table of the vSSD will be removed.

Prediction of Storage Availability and Demand. To best utilize harvestable resources, we use predictions. For the unallocated (unsold) VMs, we use a heuristic-based approach, using our study to characterize the unallocated storage. For allocated storage, we use a lightweight online learning approach. We predict the harvestable storage resources for allocated VMs, and demanded storage resources for harvest VMs. Since the predictions for allocated VMs and harvest VMs are both determined by their workloads, they use the same learning-based approach but different learning parameters. The inputs are statistical measures gathered from the bandwidth, IOPS, and storage utilization. The predictors generate the predicted bandwidth (in channels), capacity (in GB), and duration of resource demands.

Exception Handling in Storage Harvesting. Since we use predictions to improve resource utilization, BlockFlex must also handle mispredictions. As shown in Table 1, mispredictions could mainly cause three exceptions: *waste of storage resources*, *early resource reclamation*, and *under-harvesting*.

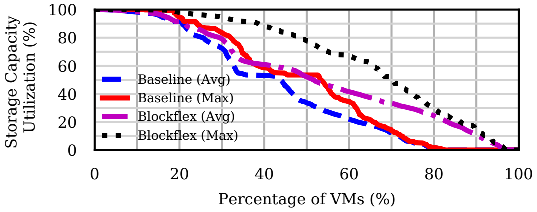

Figure 2: The capacity utilization of allocated cloud storage.

First, BlockFlex could *waste storage resources* when mispredictions leave them unused. As we over-provision demanded storage in the harvest VMs to avoid reclamations, it is inevitable to cause some waste of storage resources. To handle *early reclamation*, BlockFlex migrates data between gSSDs at block granularity to minimize the impact on running applications. BlockFlex reclaims the old gSSD while ensuring its flash blocks are erased before being used by the regular vSSD. Finally, to handle *under-harvesting*, BlockFlex will harvest new gSSDs until meeting the demand. However, if no gSSD is available, BlockFlex will report an exception to the harvest VM, resulting in a termination or delay of job execution in the harvest VM.

Key Contributions. We summarize the contributions below:

- We conduct a characterization study of the storage efficiency in different cloud platforms, motivating storage harvesting.
- We rethink the abstractions of storage virtualization in modern cloud platforms for enabling fine-grained storage harvesting with software-defined flash.
- We build a learning-based storage harvesting framework named BlockFlex that can harvest both unallocated and allocated storage resources.
- We develop predictors that can make efficient predictions for both storage demand and availability in terms of storage capacity, bandwidth, and the time available for harvesting.

BlockFlex Implementation. We implement the gSSD abstraction of BlockFlex using a programmable SSD, whose controller allows read/write/erase operations against the raw flash resources. Each model is implemented with one hidden LSTM layer fully connected with the input and output layers.

BlockFlex Evaluation. Our evaluation demonstrates that: (1) BlockFlex gives 1.25x improvement in storage utilization for cloud platforms by leveraging both underutilized and unallocated storage resources (see Figure 2 for improvement in storage capacity); (2) BlockFlex improves the performance of harvest VMs by up to 60% while minimizing the impact on regular VMs; (3) BlockFlex introduces negligible overhead to storage management. Please see the detailed evaluation in [5].

References

[1] "Alibaba Cluster Trace.." `https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/trace_2018.md`.

[2] P. Ambati, I. Goiri, F. Frujeri, A. Gun, K. Wang, B. Dolan, B. Corell, S. Pasupuleti, T. Moscibroda, S. Elnikety, M. Fontoura, and R. Bianchini, "Providing slos for resource-harvesting vms in cloud platforms," in *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20)*, Nov. 2020.

[3] "Google Cluster Trace.." `https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md`.

[4] J. Huang, A. Badam, L. Caulfield, S. Nath, S. Sengupta, B. Sharma, and M. K. Qureshi, "Flashblox: Achieving both performance isolation and uniform lifetime for virtualized ssds," in *Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST'17)*, Feb. 2017.

[5] B. Reidys, J. Sun, A. Badam, S. Noghabi, and J. Huang, "BlockFlex: Enabling storage harvesting with Software-Defined flash in modern cloud platforms," USENIX Association, July 2022.