Storage Deduplication in Cloud Computing

João Paulo and José Pereira

University of Minho

July 2010

Cloud Computing Overview

Cloud Computing

- Cloud services allow clients to shift their data and applications into the “cloud“.
- These services run in a scalable and dependable infrastructure, which has a large server pool in several data centres.

Virtualization

- Virtualization is a key aspect to achieve the Elasticity provided by cloud computing.
- Virtual Machines (VMs) can be deployed/migrated in few minutes.
- VMs Isolation allows a better management of resources and failures.

Cloud Computing Overview

Cloud Computing

- Cloud services allow clients to shift their data and applications into the “cloud“.
- These services run in a scalable and dependable infrastructure, which has a large server pool in several data centres.

Virtualization

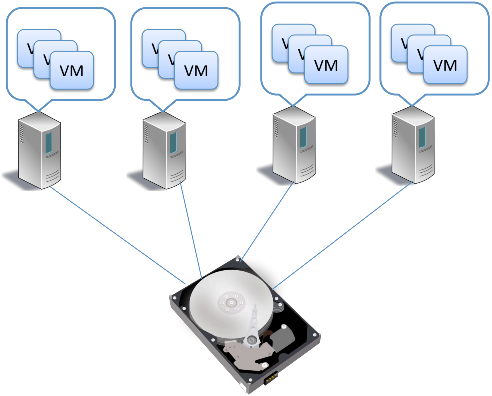
- Virtualization is a key aspect to achieve the Elasticity provided by cloud computing.
- Virtual Machines (VMs) can be deployed/migrated in few minutes.
- VMs Isolation allows a better management of resources and failures.

Deduplication

- Cloud services store client's data, applications and VMs images.
- Deduplication allows to:
  - Decrease storage's size.
  - Optimize the management of storage's data.
- Deduplication introduces overhead to the service.

Outline

Scenario



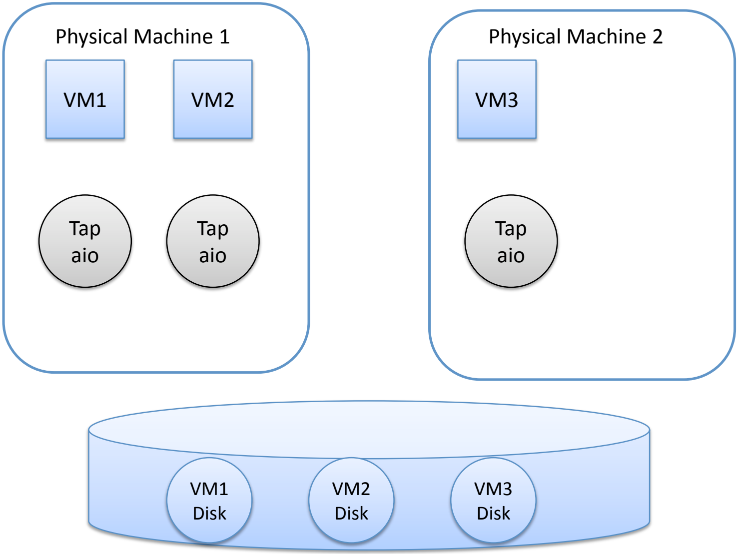
- Groups of VMs run in different physical machines.
- Each VM has its own virtual disk.
- Virtual disks are kept in a shared storage.

XEN Blktap mechanism

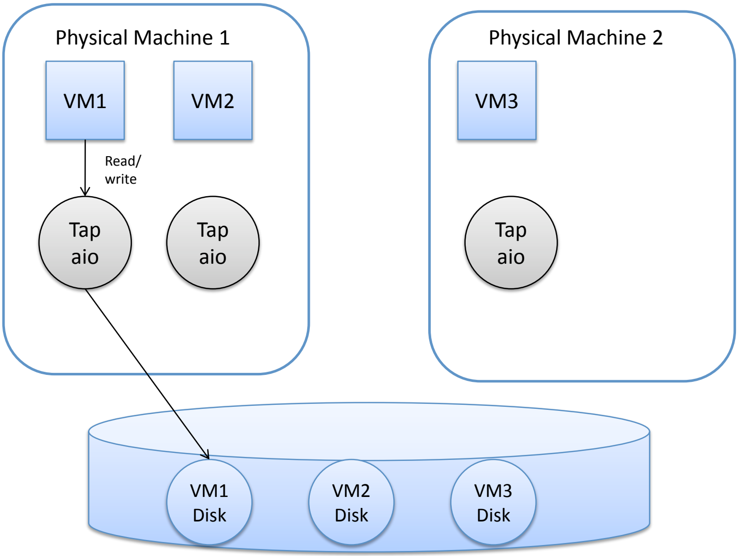
Blktap

- Implemented within Xen.
- Allows to implement virtual block devices for Virtual Machines.
- User-level disk I/O interface (Tapdisk).
- Allows to have independent per-disk handler processes.
- Easy to implement Copy-on-Write.

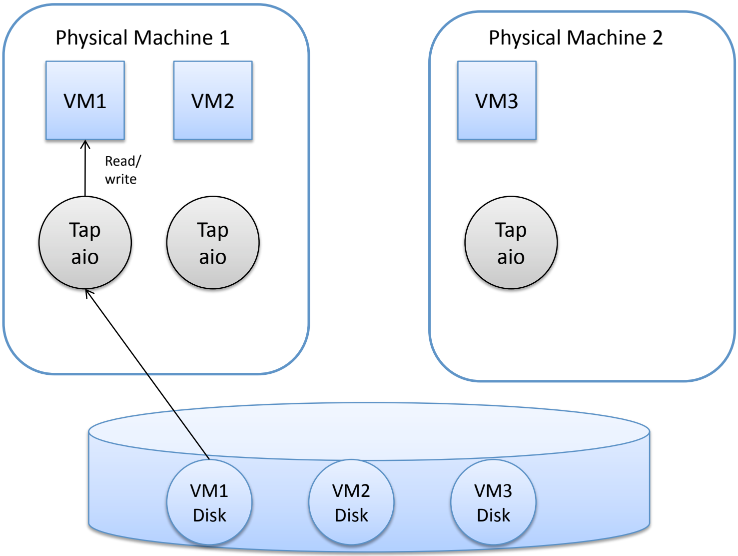
XEN Blktap mechanism



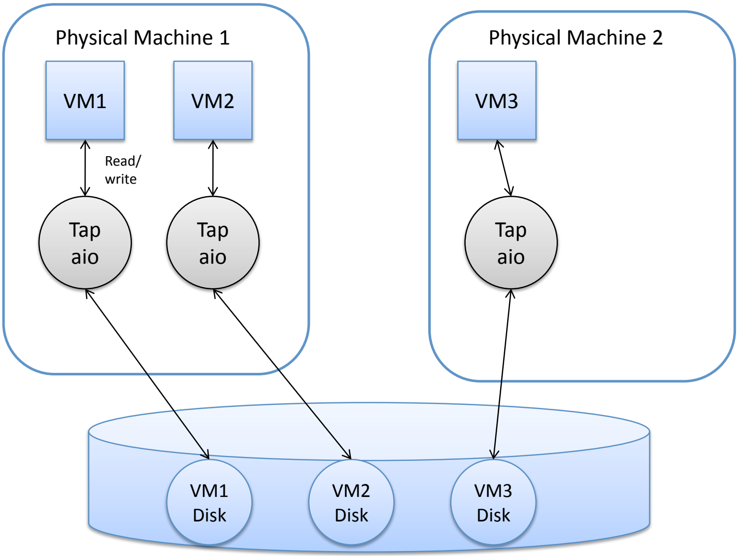
XEN Blktap mechanism



XEN Blktap mechanism



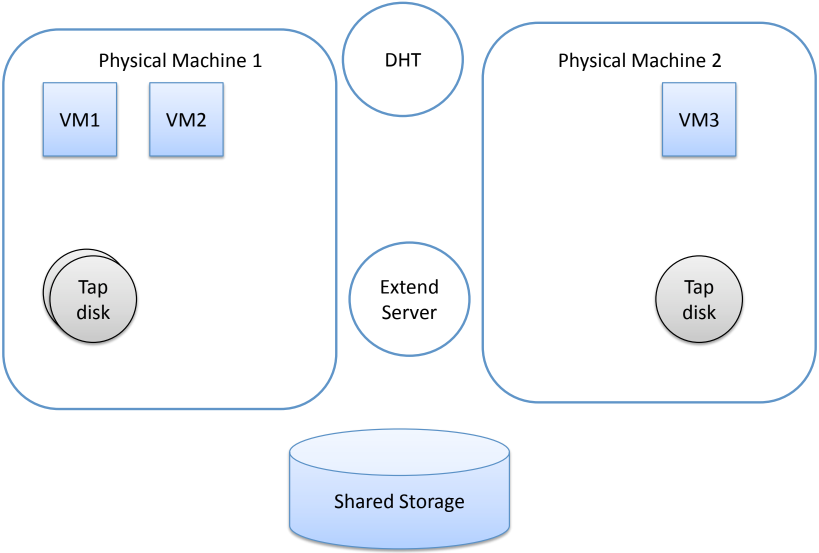
XEN Blktap mechanism



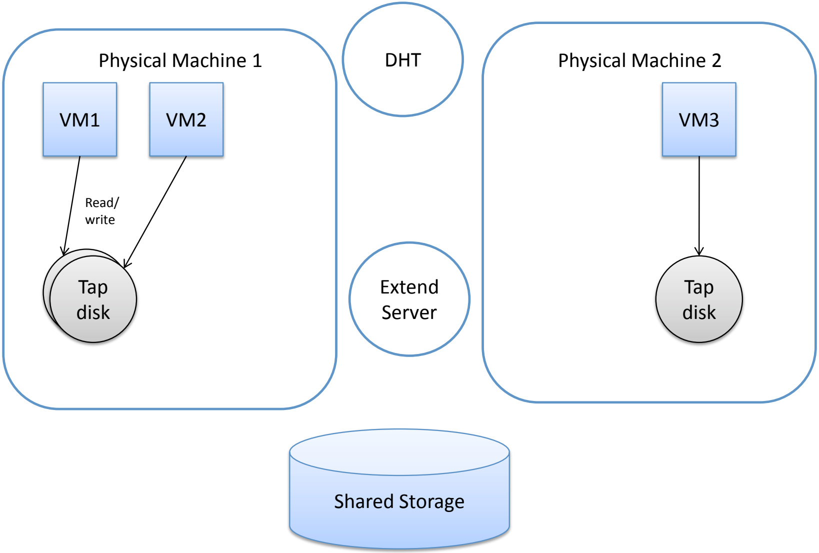
Deduplication Challenges

- Deduplication is usually used for backup scenarios where data is practically immutable.
- In a virtualized scenario where stored data changes constantly, we must have in account:
  - The overhead introduced by the deduplication algorithm.
  - The best approach to find duplicated data, which must be transparent to the VMs.
  - The metadata needed to share identical data.

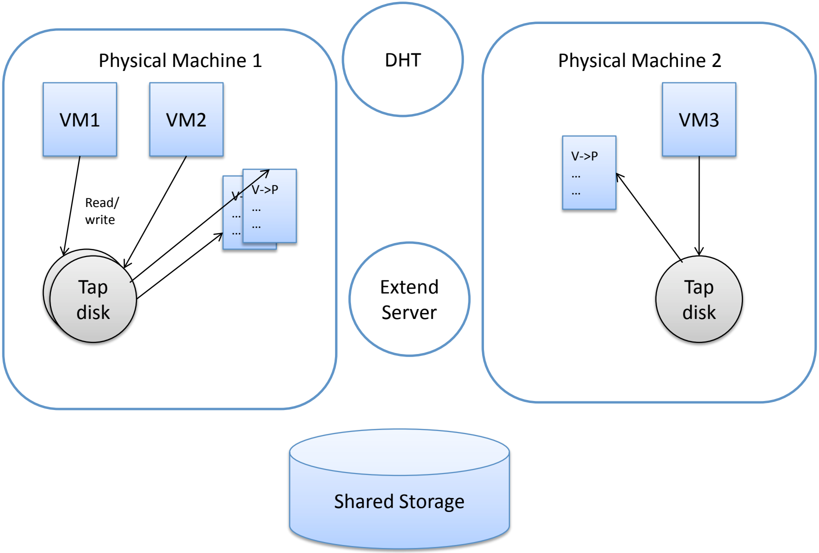
Deduplication Algorithm



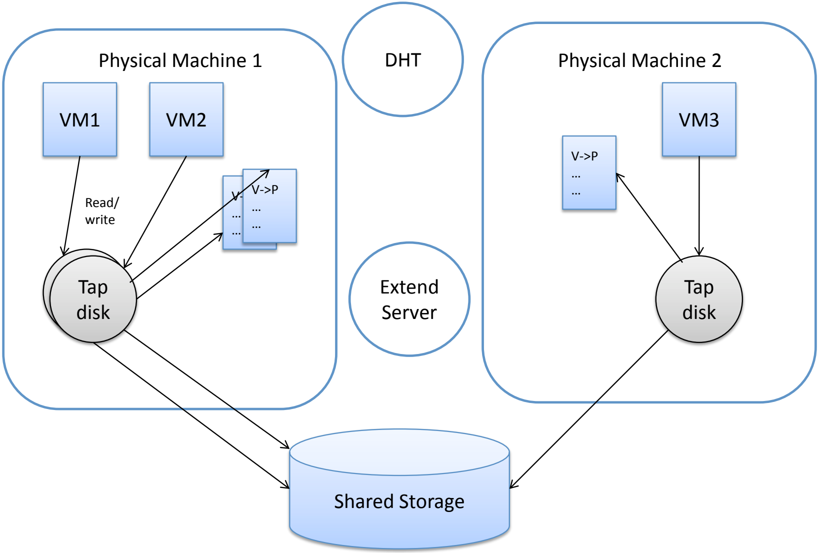
Deduplication Algorithm



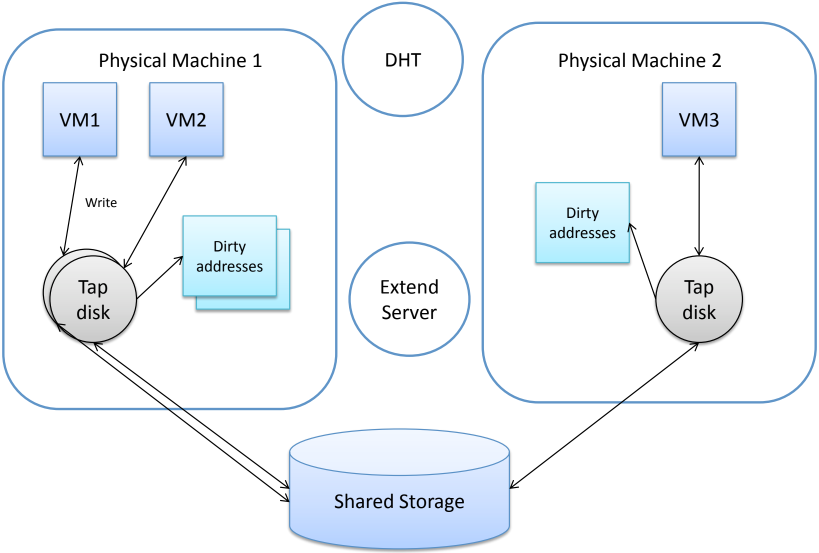
Deduplication Algorithm



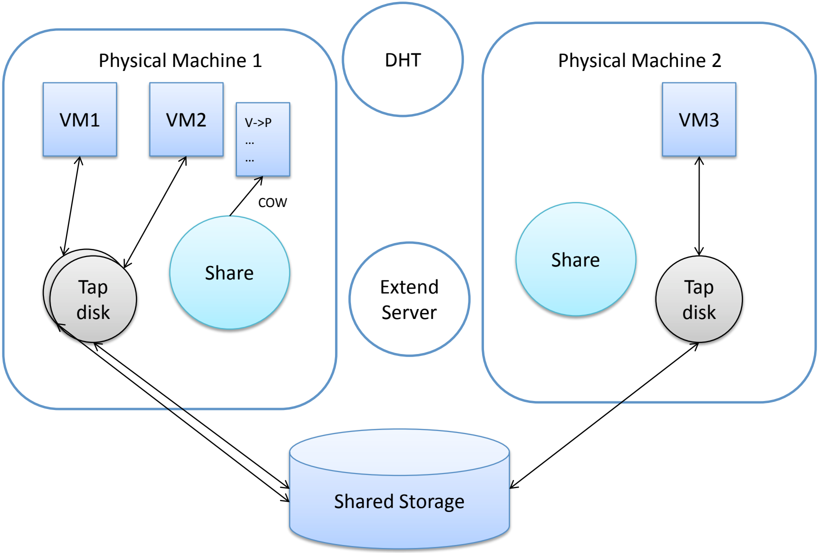
Deduplication Algorithm



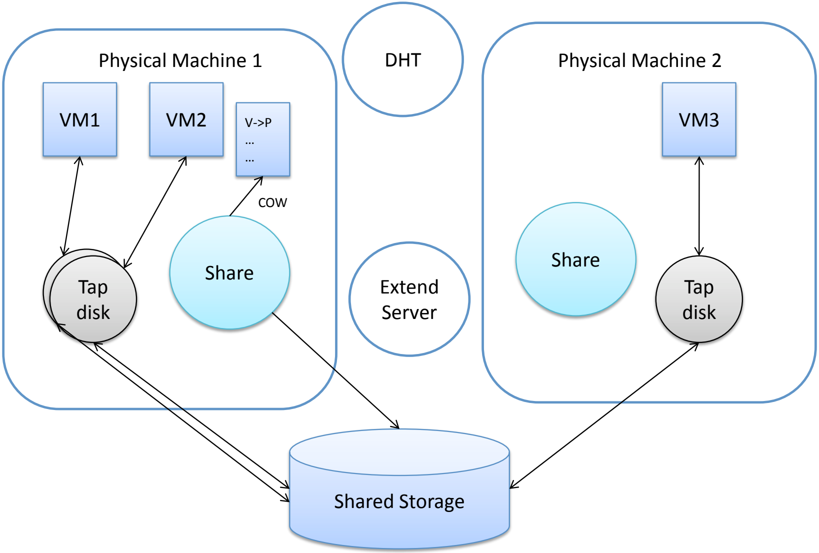
Deduplication Algorithm



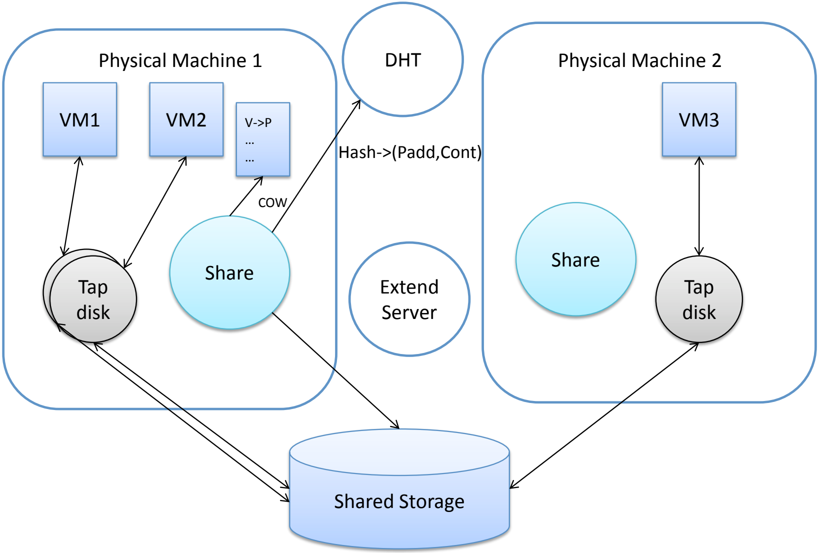
Deduplication Algorithm



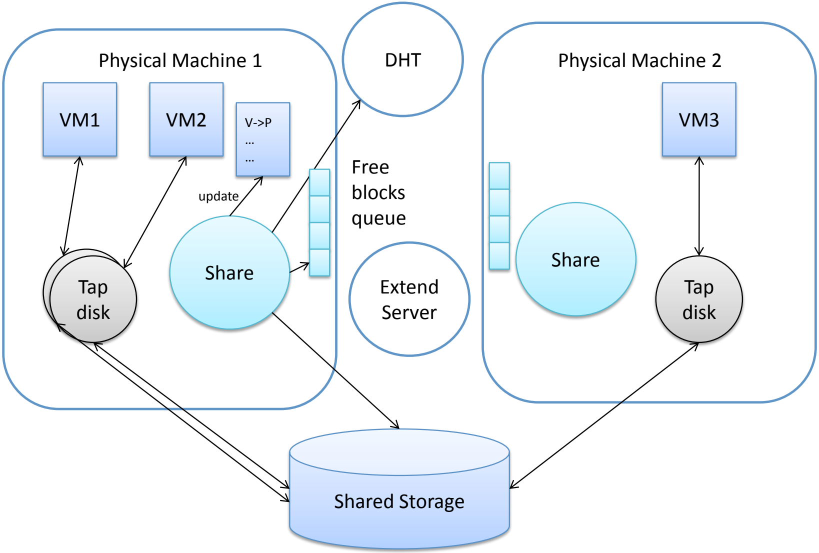
Deduplication Algorithm



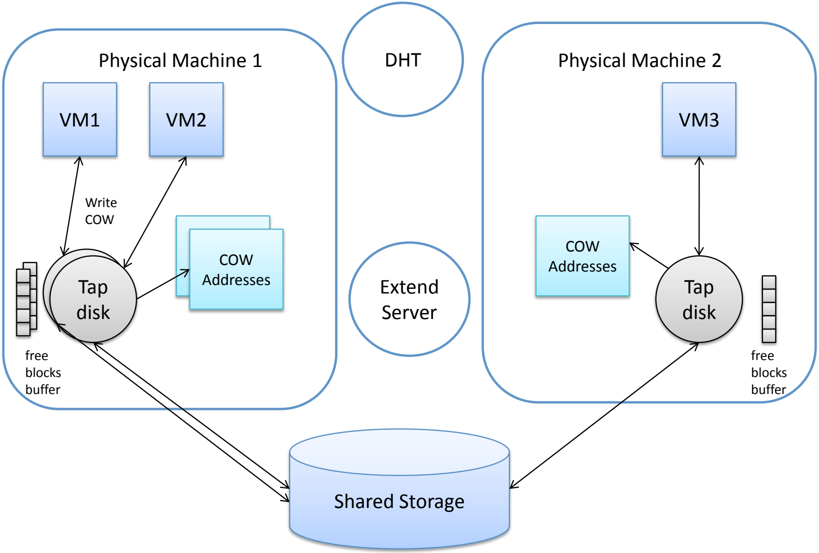
Deduplication Algorithm



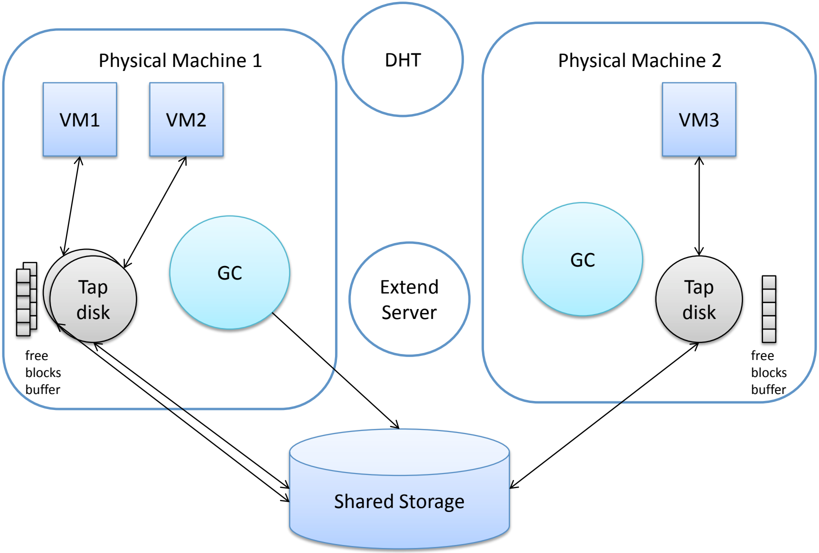
Deduplication Algorithm



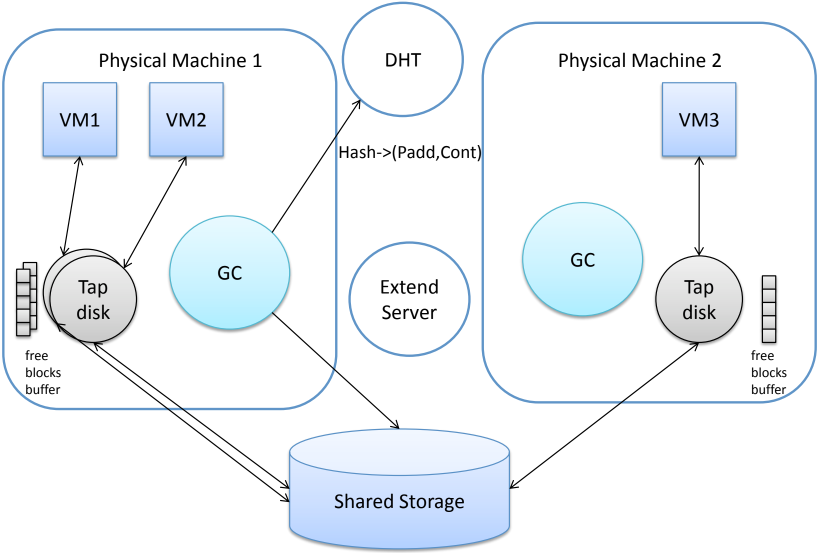
Deduplication Algorithm



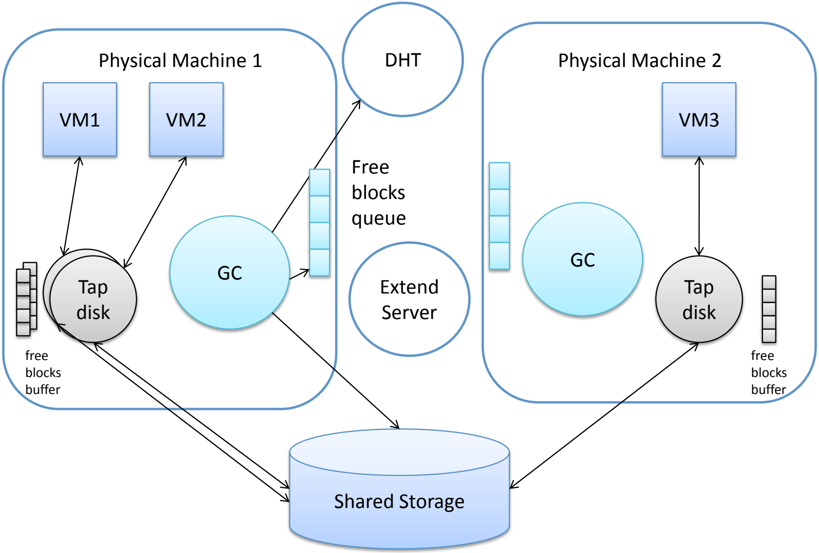
Deduplication Algorithm



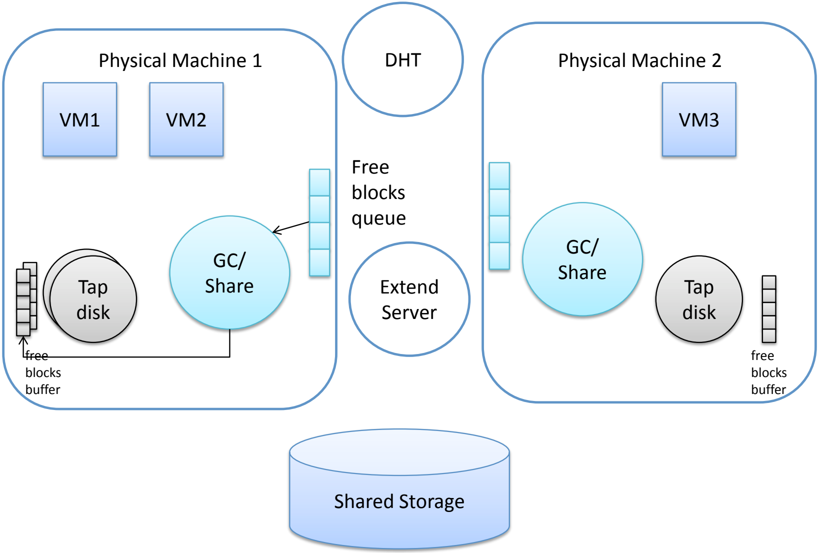
Deduplication Algorithm



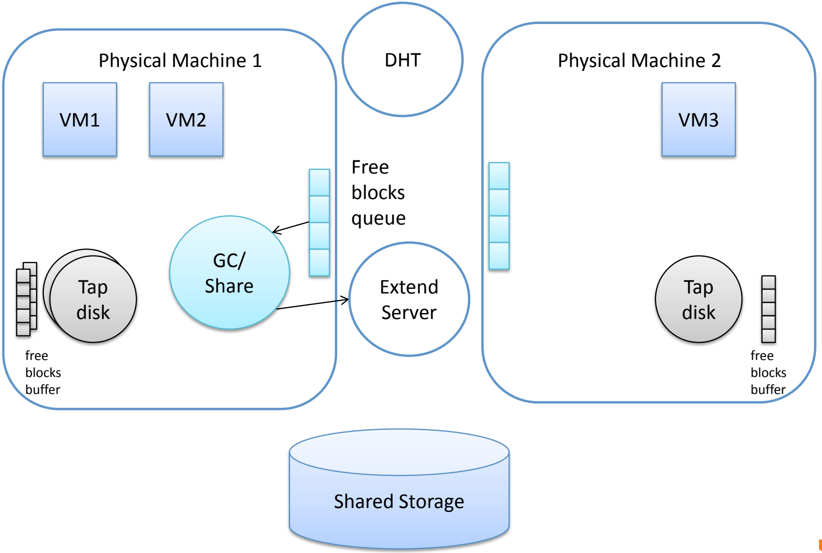
Deduplication Algorithm



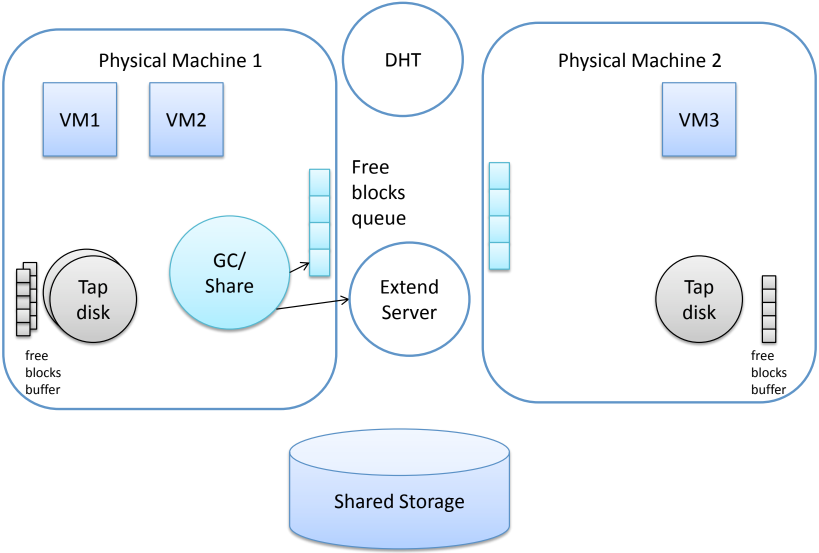
Deduplication Algorithm



Deduplication Algorithm

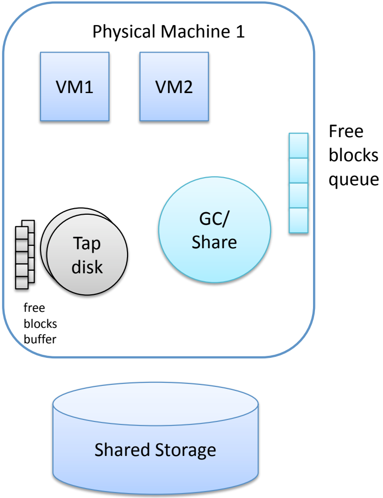


Deduplication Algorithm



Outline

Evaluated Prototype



- Without Distribution and Fault Tolerant design.
- Two Optimizations:
  - Set of mutexes for each VM's Translation table.
  - VM's free blocks buffer refilling granularity.

Benchmarks

Main Goals

- Measure the I/O and CPU overhead introduced by our prototype when compared to a default approach (Tap Aio).
- Measure the sharing rates achieved by our prototype.

Write and Read Benchmarks

- TPC-C NURand function is used to generate hotspots for write and read operations.
- A realistic distribution is used for generating the content of the blocks that are written.

Benchmarks

Main Goals

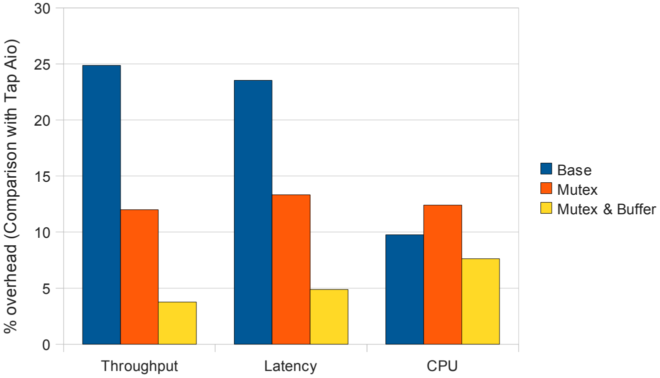
- Measure the I/O and CPU overhead introduced by our prototype when compared to a default approach (Tap Aio).
- Measure the sharing rates achieved by our prototype.

Write and Read Benchmarks

- TPC-C NURand function is used to generate hotspots for write and read operations.
- A realistic distribution is used for generating the content of the blocks that are written.

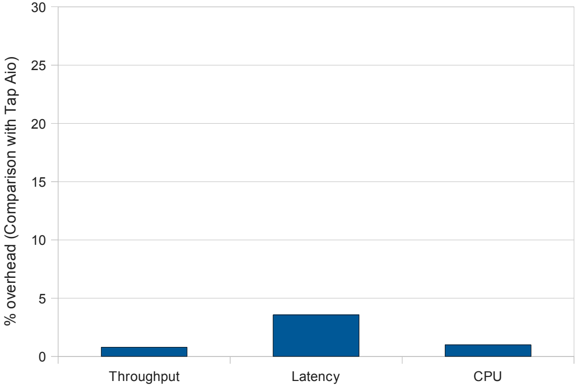
Write Benchmark

- The benchmark ran for 30 minutes in three VMs with 10 GB images.
- Additional RAM used was 250 MB for the fully optimized version.
- Approximately 28% of the written data (20 GB) was shared.



Read Benchmark

- The benchmark ran for 40 minutes in three VMs with 10 GB images.
- Additional RAM used was 200MB for the fully optimized version.
- Approximately 55% of the written data (4.5 GB) was shared.



Outline

Conclusions

- The evaluated prototype shares identical data without introducing a significant amount of overhead in the CPU usage and I/O requests.
- The asynchronous approach to share identical data, the dynamic detection of modified data and the prototype optimizations are key aspects to achieve these results.
- Project available at `http://www.holeycow.org/`

Outline

Future Work and Challenges

Replicated Databases

- HoleyCOW - Shared Storage Cluster.
- Can deduplication be used to improve this specific scenario?

Fault Tolerance

- Byzantine Faults are not contemplated.
  - Data corruption.
  - What level of redundancy should we keep?
  - Malicious attacks.

Resilient Databases (Red) - `http://red.lsd.di.uminho.pt/`

Future Work and Challenges

Replicated Databases

- HoleyCOW - Shared Storage Cluster.
- Can deduplication be used to improve this specific scenario?

Fault Tolerance

- Byzantine Faults are not contemplated.
  - Data corruption.
  - What level of redundancy should we keep?
  - Malicious attacks.

Resilient Databases (Red) - `http://red.lsd.di.uminho.pt/`

Future Work and Challenges

Replicated Databases

- HoleyCOW - Shared Storage Cluster.
- Can deduplication be used to improve this specific scenario?

Fault Tolerance

- Byzantine Faults are not contemplated.
  - Data corruption.
  - What level of redundancy should we keep?
  - Malicious attacks.

Resilient Databases (Red) - `http://red.lsd.di.uminho.pt/`

Future Work and Challenges

Distributed Storage

- Each server has its own disk.
- A new approach to find and share duplicated data is necessary. Epidemic Protocols?
- Garbage Collector also needs to be redesigned.