VastSky

Cluster storage system for XCP

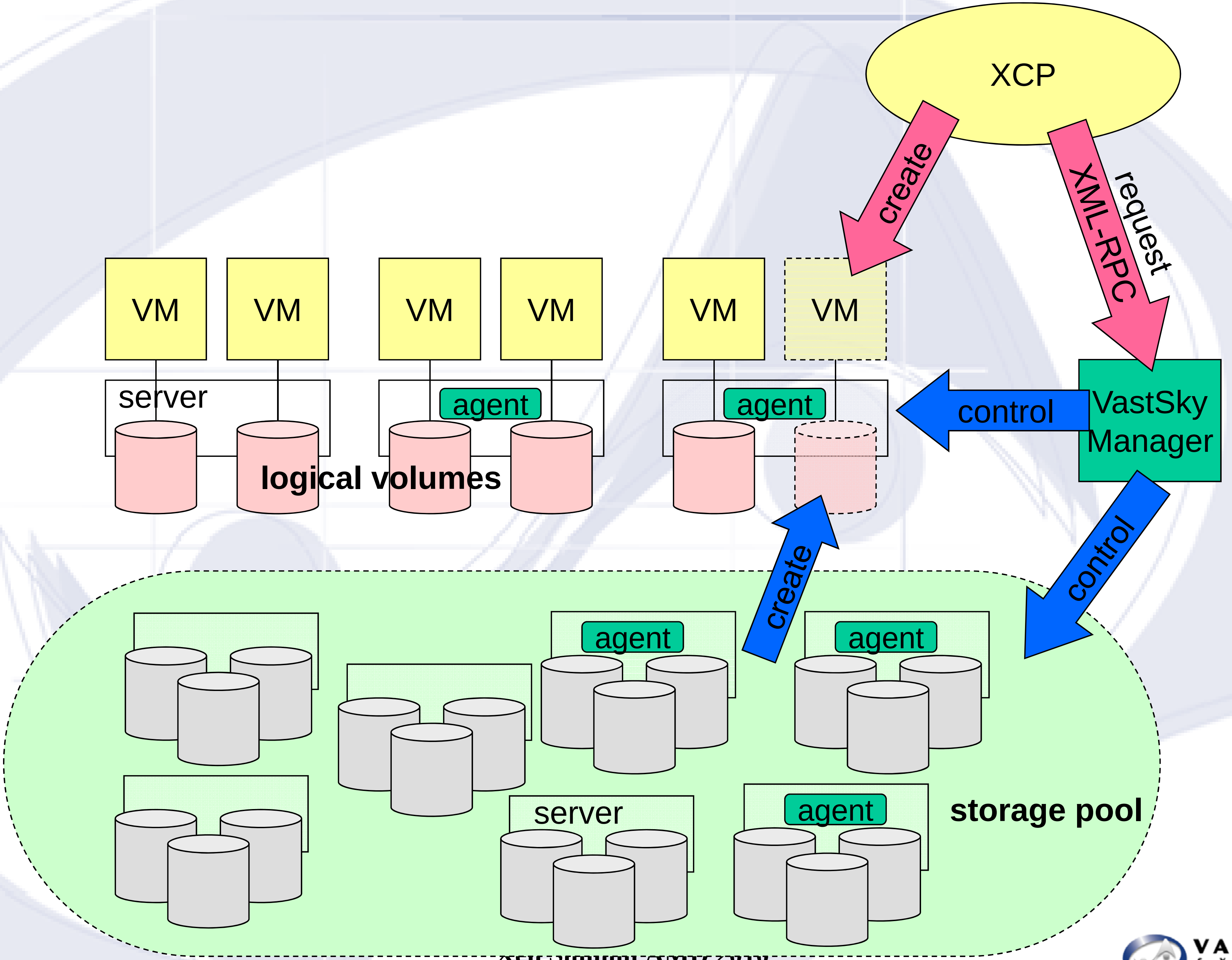
Apr. 28th, 2010

VA Linux Systems Japan K.K.
Hirokazu Takahashi
Tomoaki Sato
Takashi Yamamoto

Xen Summit AMD 2010

What is VastSky all about?

- VastSky is a cluster storage system made up of a lot of servers and disks, from which VastSky Manager creates logical volumes for VMs
- VMs can directly run on VastSky, which XCP can control
- VastSky is scalable, high availabile and has a good performance


XCP
create
XML-RPC
request
VM
VM
VM
VM
VM
VM
server
agent
agent
logical volumes
control
VastSky Manager
create
control
agent
agent
server
agent
storage pool

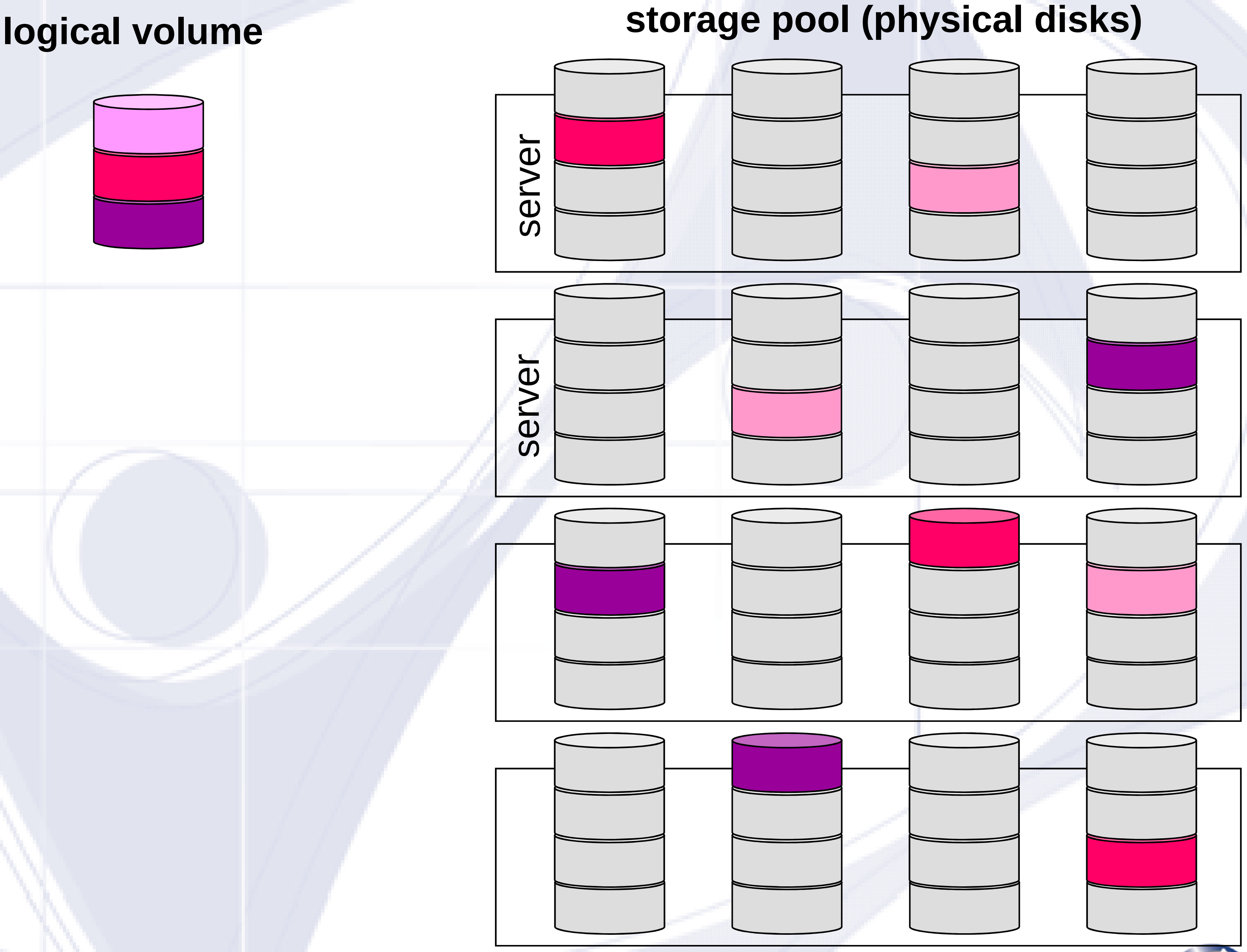
Announcement

- The code of VastSky has become open at http://sf.net/projects/vastsky/
  - Some more work is needed to be done before the first release.

Basic Design

- A logical volume is a set of several mirrored disks, each of which consists of several physical disk chunks on different servers.
  - The logical volume won’t lose its data whether a physical disk or a storage server in the storage pool has broken.
  - All I/O requests, including read, write and even re-synchronizing requests of mirrored devices will be distributed to all the physical disks.

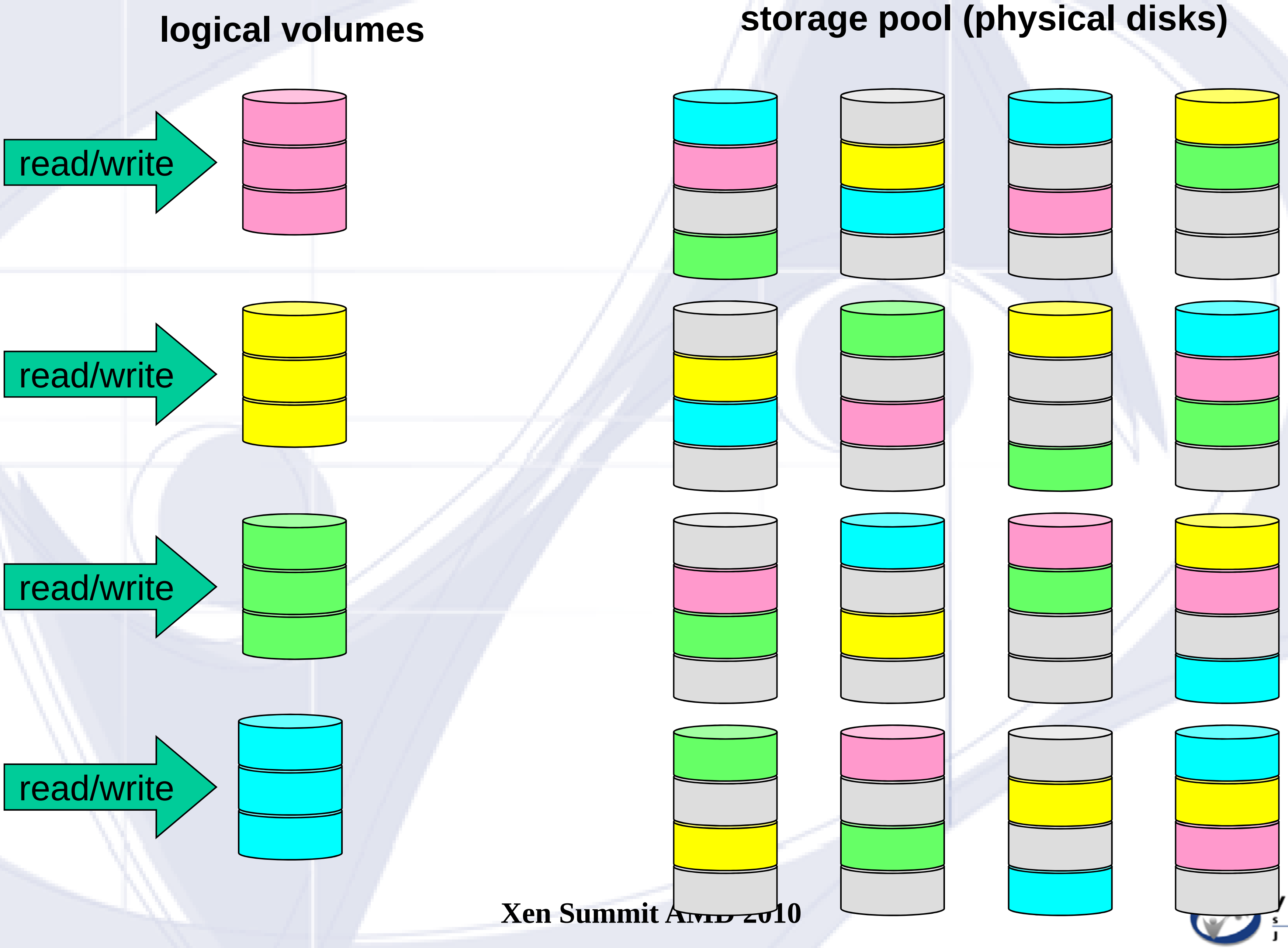
The way of making a logical volume



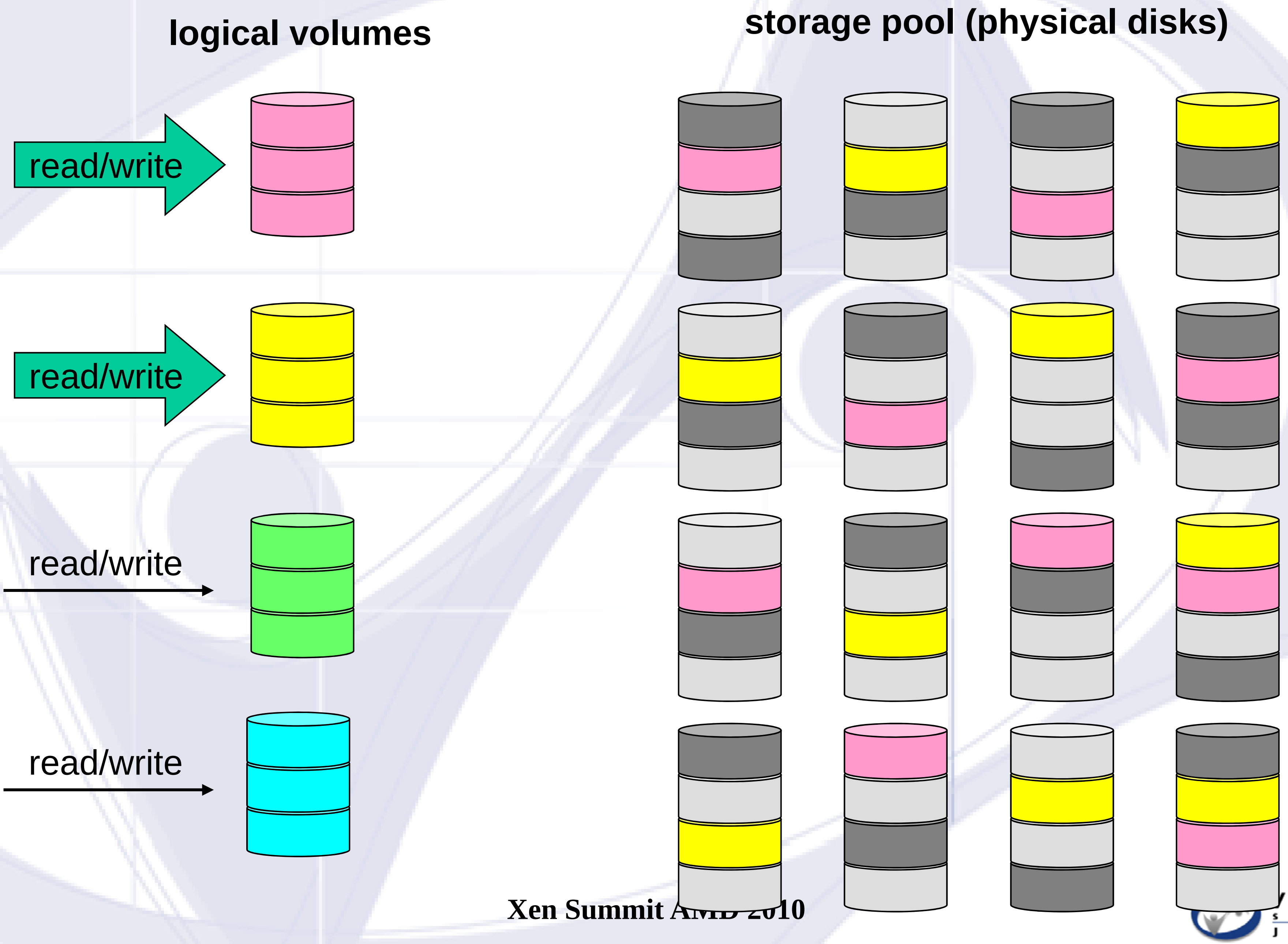
Good performance

- All I/O operations will be done in the linux kernel without any VastSky Manager interactions.
- I/O loads of logical volumes, which can be extremely unbalanced, will be equalized between the physical disks.
- I/O requests to rebuild mirrored devices are also distributed across a lot of physical disks.

Load balancing of read/write requests



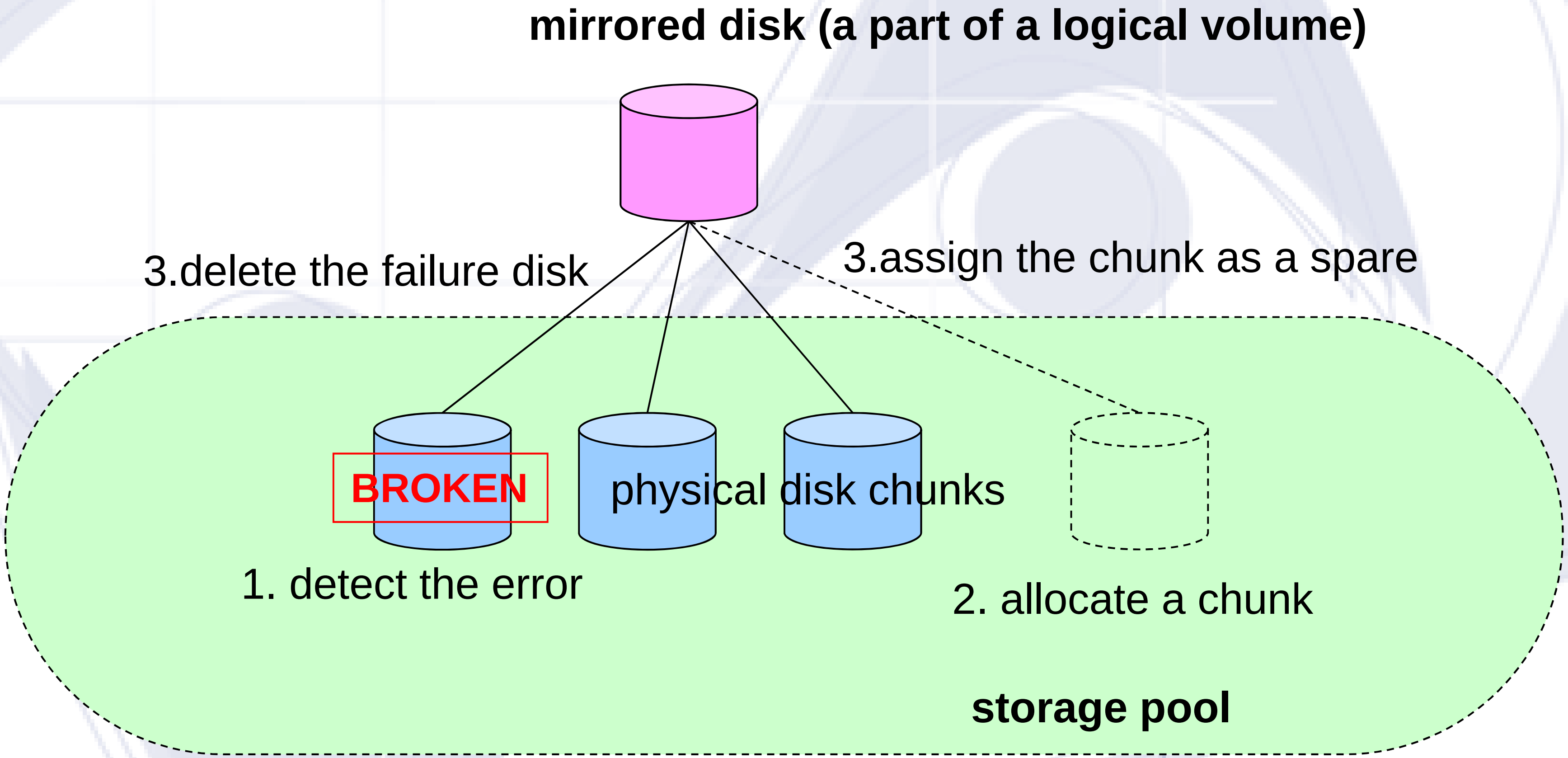
Load balancing of read/write requests



Mirrored disk recovery

- Each mirrored disk doesn’t have its own spare disk.
- When VastSky detects one of the physical disk chunks of the mirrored disk has caused an error, VastSky Manager allocates a new chunk form the storage pool and assigned it to the mirrored disk as a spare.
  - The manager schedules when it should be assinged, so two or more re-sync operations won’t work on the same physical disk.
- The mirrored disk starts re-synchronizing the disk chunks right after the spare is assigned.

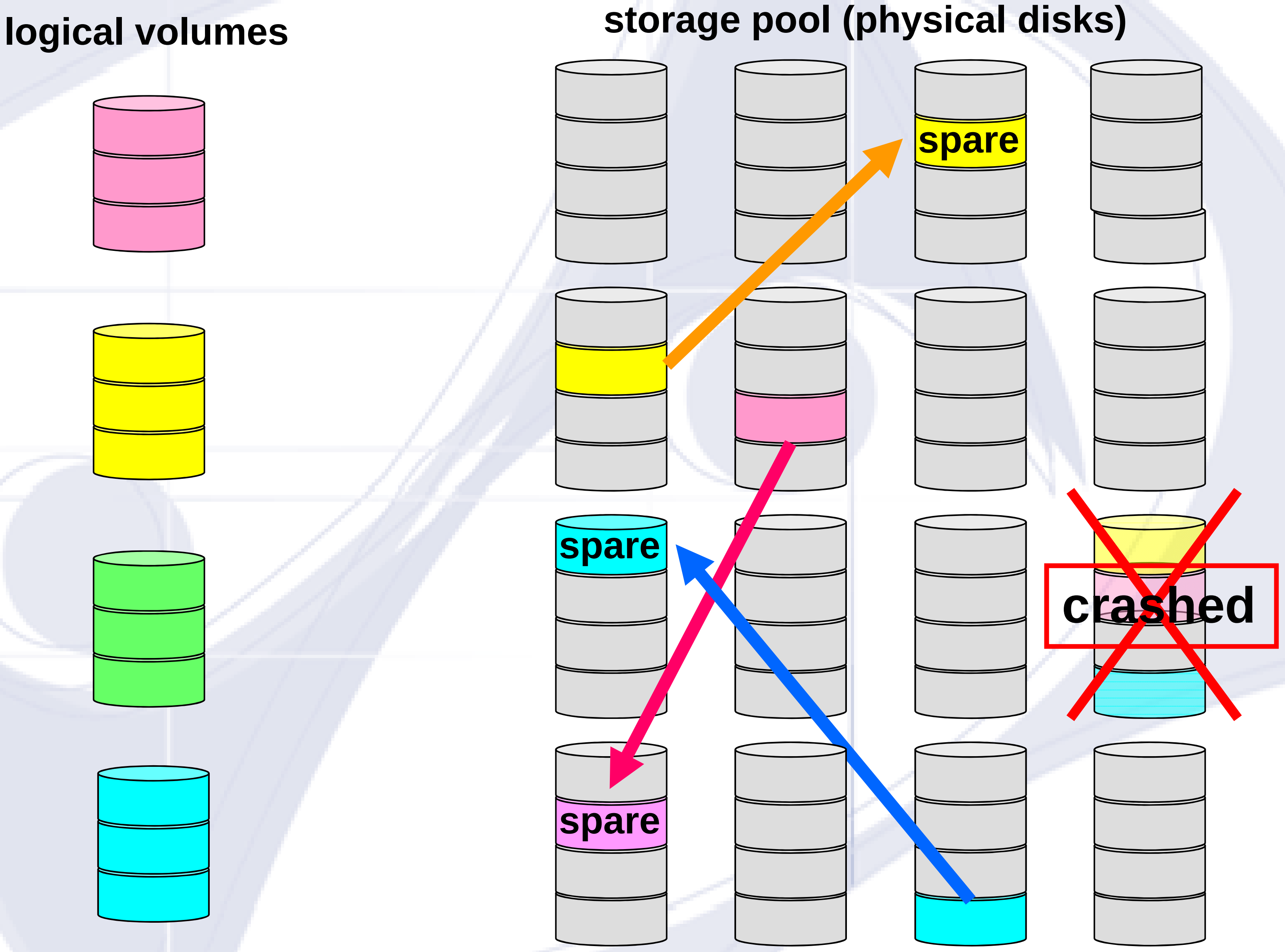
Mirrored disk recovery


mirrored disk (a part of a logical volume)
3.delete the failure disk
3.assign the chunk as a spare
BROKEN
physical disk chunks
1. detect the error
2. allocate a chunk
storage pool

Load balancing when re-synchronizing the mirrored devices

- When a certain physical disk gets broken, VastSky tries to rebuild the mirrored disks related to the physical disk simultaneously since the disk chunks belong to different mirrored disks.
  - No need to rebuild if the disk chunk is unsed.

Load balancing when re-synchronizing the mirrored devices



Scalable

- Each volume can handle its I/O operations independently.
  - VastSky Manager doesn’t care about it.
- Servers can be added to the system dynamically.

How to setup

- VastSky should be installed with VM management software such as XCP to take care about VM life-cycle.
- Networking redundancy should be implemented outside VastSky, such as using a bonding device.
- Hardware health check should also work outside VastSky and hopefully it can tell VastSky which server or disk should be removed.
- The current implementation of VastSky requires HA cluster software to detect its manager down to be restarted.

API

- VastSky supports XML-RPC interface and CUI like:
  - Define a logical volume.
  - Attach the logical volume on a specified server.
  - Detach the volume.
  - Notify which disk or server has gone.
  - Add a new server or a physical disk.
  - Delete the server or the physical disk.

ToDo (1)

- XCP integration
  - Under development.
- Improve scalability.
  - Network topology aware volume allocation. When creating a new logical volume, physical disk chunks should be allocated from storage servers close to the server that owns the logical volume.
- Logical volume expansion feature.
- Snapshot feature for Guest volumes.

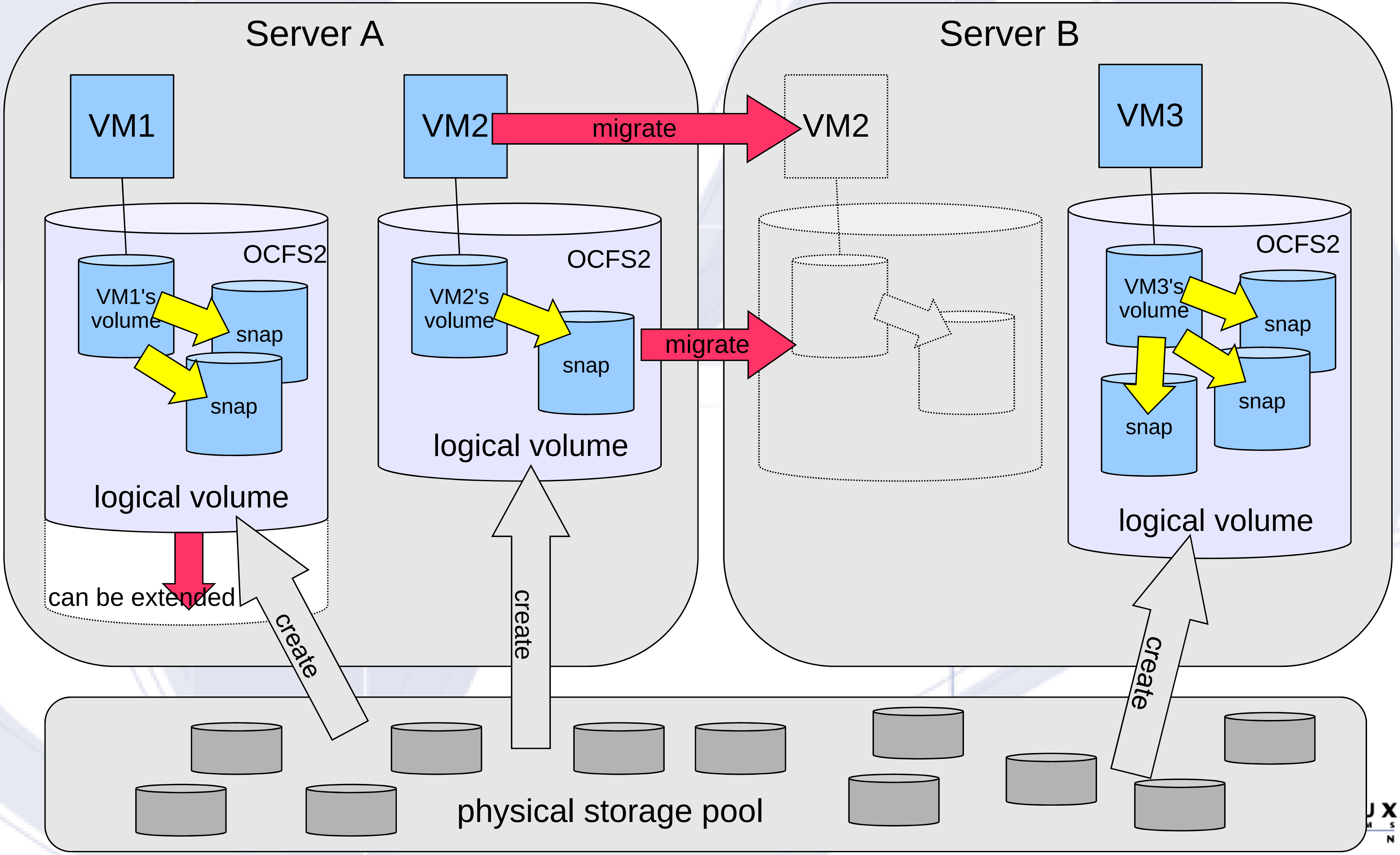
Ideas of how to implement volume snapshot feature

- Use dm-snap. It is the easiest way but works slow.
- Implement a completely new implementation like Parallax does but it will take long time.
- Use OCFS2, which has rich features but it will be a bit heavy.

An idea of using OCFS2

- **If you place only one VM's volume placed in an OCFS2 on a logical volume on a head-server, you can obtain:**
  - Better snapshot mechanism using an ocfs2's new feature reflink.
  - Thin provisioning.
  - The volume can still be moved to another server.

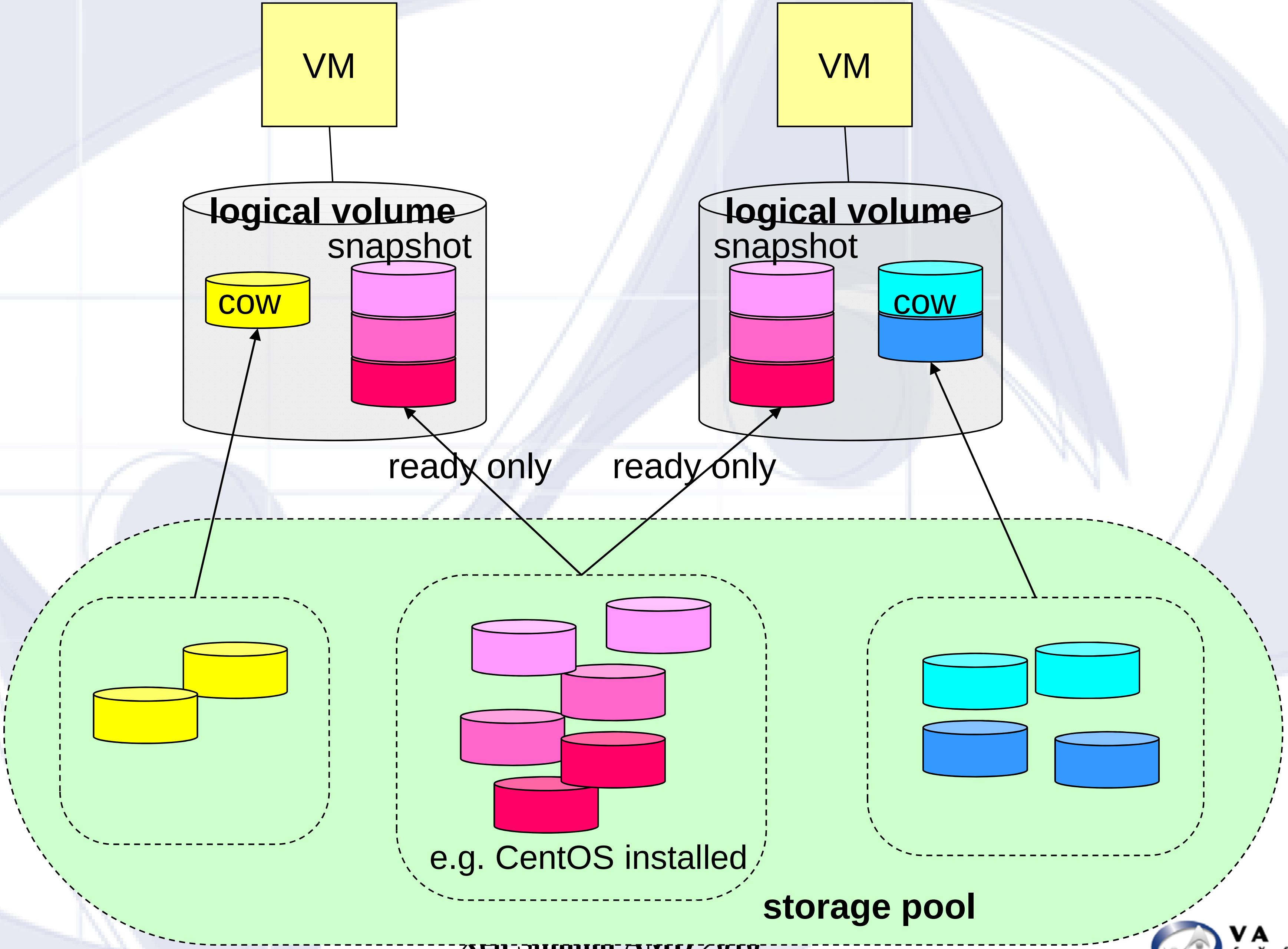
Place only one guest's volume in an ocfs2 filesystem



ToDo (2)

- Shared storage for VMs, which some type of active/active clustering software requires. The point is the way of rebuilding the mirrored devices.
  - The way to determine which server should take the job to rebuild the mirrored device.
  - Make the rebuilding job and write access to the device exclusively.
- Fast VM deploying and cloning. This can be done with the combination of “shared storage” and “snapshot” features.

Fast VM deployment



ToDo (3)

- **Make one server be able to manage both VMs and a lot of physical disk.**
  - Do you really want this feature?
- **Improve the disk chunk allocation algorithm.**
  - Make it disk performance aware.
- **Graceful server termination.**
  - The copies of the chunks in the server should be prepared before the termination.
- **Make VastSky Manager be able to run in a VM.**
  - Need some trick. The info to create the volume of the VM for VastSky is stored in this volume.

Roadmap

- First version release
  - XCP integration
  - Make it stable
  - Performance test
  - Write documents
  - The target date is this coming June.
- Second version and after
  - The rest on the Todo list. What should we do first?

Thank you!