

# **Создание СХД с томами «тонкой» настройки на базе дистрибутивов Linux**

Общее описание, модель хранения данных,  
основные переменные для СХД с томами  
«тонкой» настройки на базе дистрибутивов  
Linux.

Доклад на LVEE 2019.

# Основная задача

- Создать инфраструктуру проекта с учетом всех требований к системе хранения данных (СХД), с перспективой дальнейшего расширения, но с минимальными затратами на старте.
- Развернуть проект с выделением требуемого объема дискового пространства превышающие текущее значение свободного места в хранилище.

# “Thick” or “Thin” volume

- **“Thick” volume** – том параметры которого определены заранее и содержит фиксированное количество блоков для хранения данных, без возможности изменения их числа в процессе работы.
- **“Thin” volume** – том параметры которого определены заранее, но изначально содержит небольшое число блоков для хранения данных, и их количество «растет» по мере его заполнения до максимального значения.

# Недостатки “**thick**” volume

- Суммарный объем “**thick**” томов не может превышать общий выделенный для них объем в хранилище данных
- Нерациональное использование выделенного объема из хранилища данных;
- Изменение параметров тома хранения данных в процессе работы зависит от возможностей операционной и файловой системы и, как правило, проводятся системным администратором в ручную

# Достоинства “thin” volume

- Можно заранее определить максимально возможные параметры тома хранения данных для пользователя
- Блоки для хранения данных в том могут добавляться из хранилища в динамическом режиме
- Позволяют рационально использовать доступную емкость хранилища данных
- Суммарный объем “thin” томов может превышать общий выделенный для них объем в хранилище данных

# Виртуальные абстракции

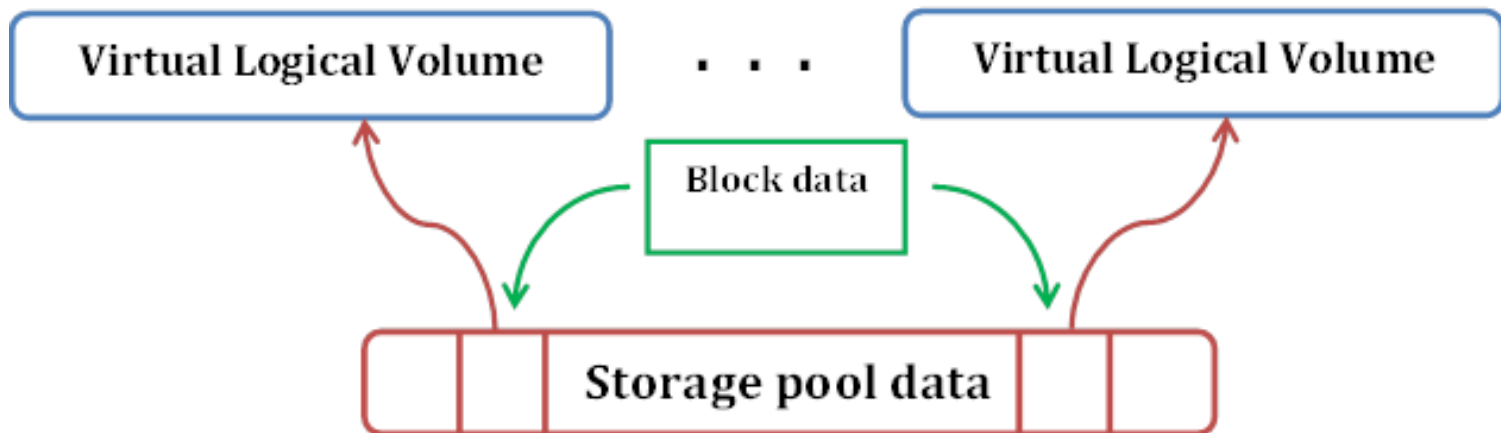
- **Файл (физический объект)**
  - Внутри содержит полную структуру сходную с форматом жесткого диска
  - Размер фиксированный или динамический
- **Блочное устройство (Device mapper)**
  - Одноуровневая модель
  - Двухуровневая модель

# Одноуровневая модель

- Реализуется как виртуальная абстракция жесткого диска (пула данных)
- Может использоваться только для «**thick**» **volume**
- По умолчанию возможно только добавление новых блоков (динамически «растет»)
- Общая емкость диска должна быть кратна размеру одного блока
- Не может использоваться в составе RAID массивов и сложных групп
- Изменение параметров возможна только в **offline** режиме

# Двухуровневая модель

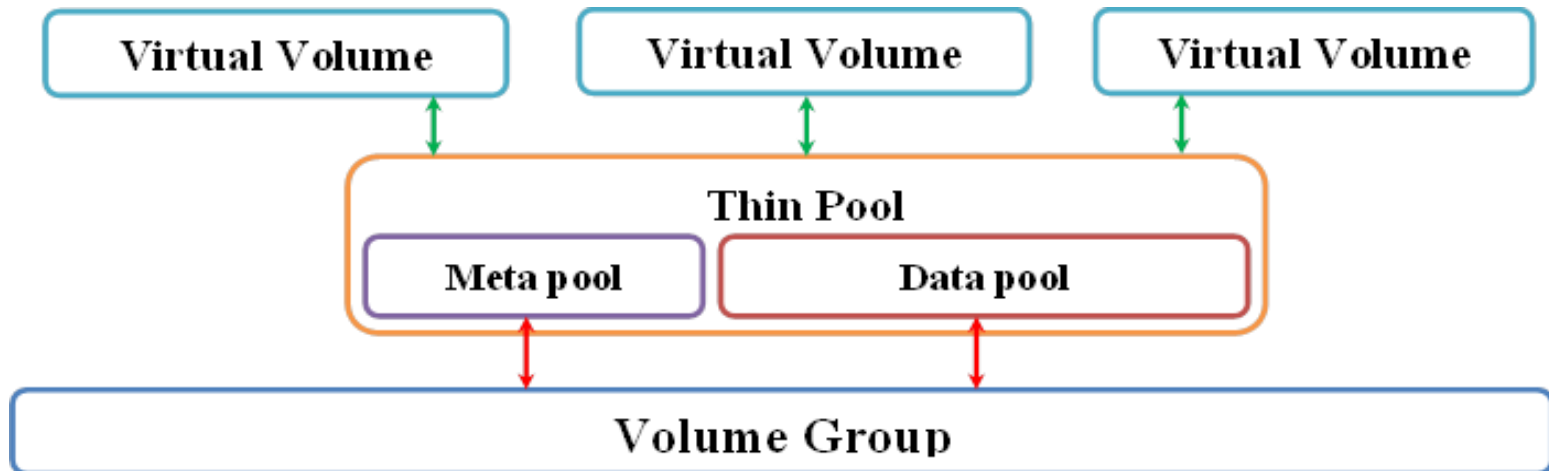
- Две виртуальные абстракции:
  - пул хранения данных (ПХД)
  - виртуальные логические тома (ВЛТ)
- Динамическое распределение блоков из пула в виртуальные логические тома





# Типовая структурная схема СХД

- Основана на двухуровневой модели
- Основная подсистема - **Device Mapper**
- Пул хранения данных (thin pool) состоит из пулов метаданных и данных
- Менеджер томов - **LVM2**



# Возможности LVM2

- Поддержку многослойного режима работы с внутренними и внешними хранилищами (один слой в режиме чтения, второй в чтения/записи)
- Поддержку динамической агрегации метаданных при помощи демона `lvm2metad`;
- Поддержку технологии LVM Cache для общих пулов хранения данных
- Поддержку RAID массивов для общего пула хранения данных
- Поддержку работы с снапшотами (снимками)

# Ключевые параметры «thin» томов

- Размер виртуальных томов (virtual volume)
- Размер фрагмента выделения (chunk size)
- Размер тома с метаданными
- `$low_water_mark` – переменная определяет предельный уровень свободного места в пуле хранения данных

# Пример расчета

- Условие: Необходимо создать хранилище данных для 100 виртуальных машин с томами размером по 50GB для каждой, при наличии физического дискового пространства в 200GB и одного накопителя `sdd` емкостью 128GB.
- Определим значение переменной `$data_dev_size_max`:
- $\$data\_dev\_size\_max = 100 * 50 * 1073741824 = 5\,368\,709\,120\,000$  байт
- Определим значение переменной `\$data_dev_size`, установив его значение в 98% от максимально возможного (рекомендовано):
- $\$data\_dev\_size = (200 * 1073741824) * 0.98 = 210\,453\,397\,504$  байт
- Определим размер фрагмента выделения значением по умолчанию в 128 блоков по 512 байт:
- $\$data\_block\_size = 128 * 512 = 65\,536$  байт
- Определим размер тома для хранения метаданных:
- $\$metadata\_dev\_size = 48 * 5\,368\,709\,120\,000 / 65\,536 = 3\,932\,160\,000$  байт или примерно 3,67GB
- Повторим предыдущие два вычисления, изменив исходные данные.
- Определим размер фрагмента выделения значением в 256 блоков по 512 байт:
- $\$data\_block\_size = 256 * 512 = 131\,072$  байт
- Определим размер тома для хранения метаданных:
- $\$metadata\_dev\_size = 48 * 5\,368\,709\,120\,000 / 131\,072 = 1\,966\,080\,000$  байт или примерно 1,83GB
- Как видно чем больше размер фрагмента выделения, тем меньше будет необходим том для хранения метаданных.
- Определим значения переменной `\$low_water_mark`, установив, ее значение равной 1024 блока:
- $\$low\_water\_mark = 1024 * 131\,072 = 134\,217\,728$  байт или примерно 128МБ
- Если это значение критично, то может его увеличить, например до 65 536 блоков:
- $\$low\_water\_mark = 65\,536 * 131\,072 = 8\,589\,934\,592$  байт или примерно 8ГБ.