

Одиннадцатая независимая научно-  
практическая конференция  
«Разработка ПО 2015»  
22 - 24 октября, Москва



# Сжатие данных на блочном уровне в Linux "За и против"

К.Кринкин, Н.Плохой  
СПбГЭТУ "ЛЭТИ"

# Мотивация

- большой размер файлов с “сырыми” данными
- требуется on-line сжатие без потерь
- необходима независимость от ФС

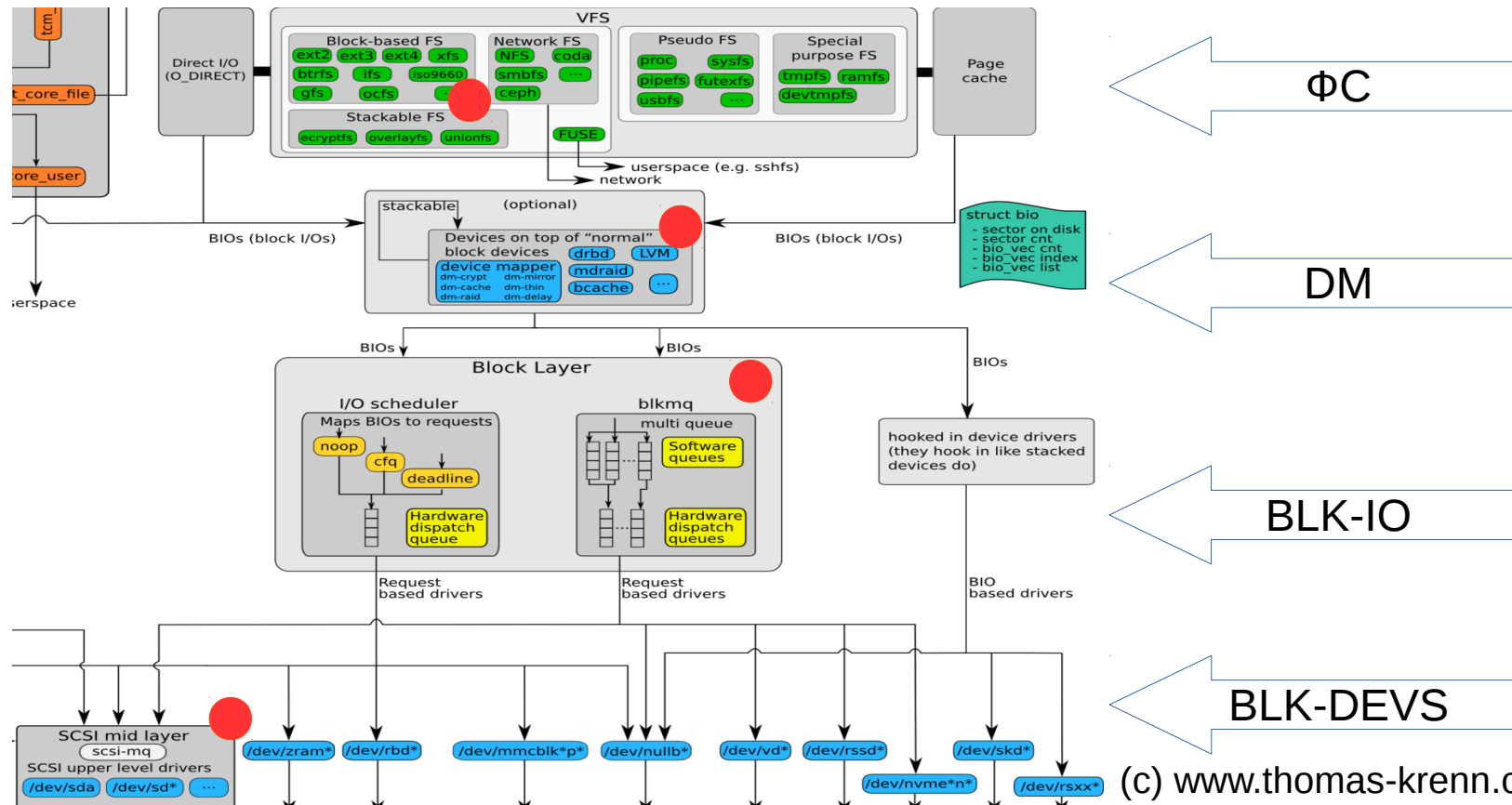
*Тестовый набор данных:* результаты медицинского 3D сканирования

*Сценарий:* однократная запись, множественное чтение

# Задачи

- Оценить возможности для компрессии блочного уровня в Linux
- Качественно сравнить file-level и block-level сжатие
- Подобрать алгоритмы сжатия блоков
- Разработать схему трансляции

# Блочный ввод-вывод в Linux



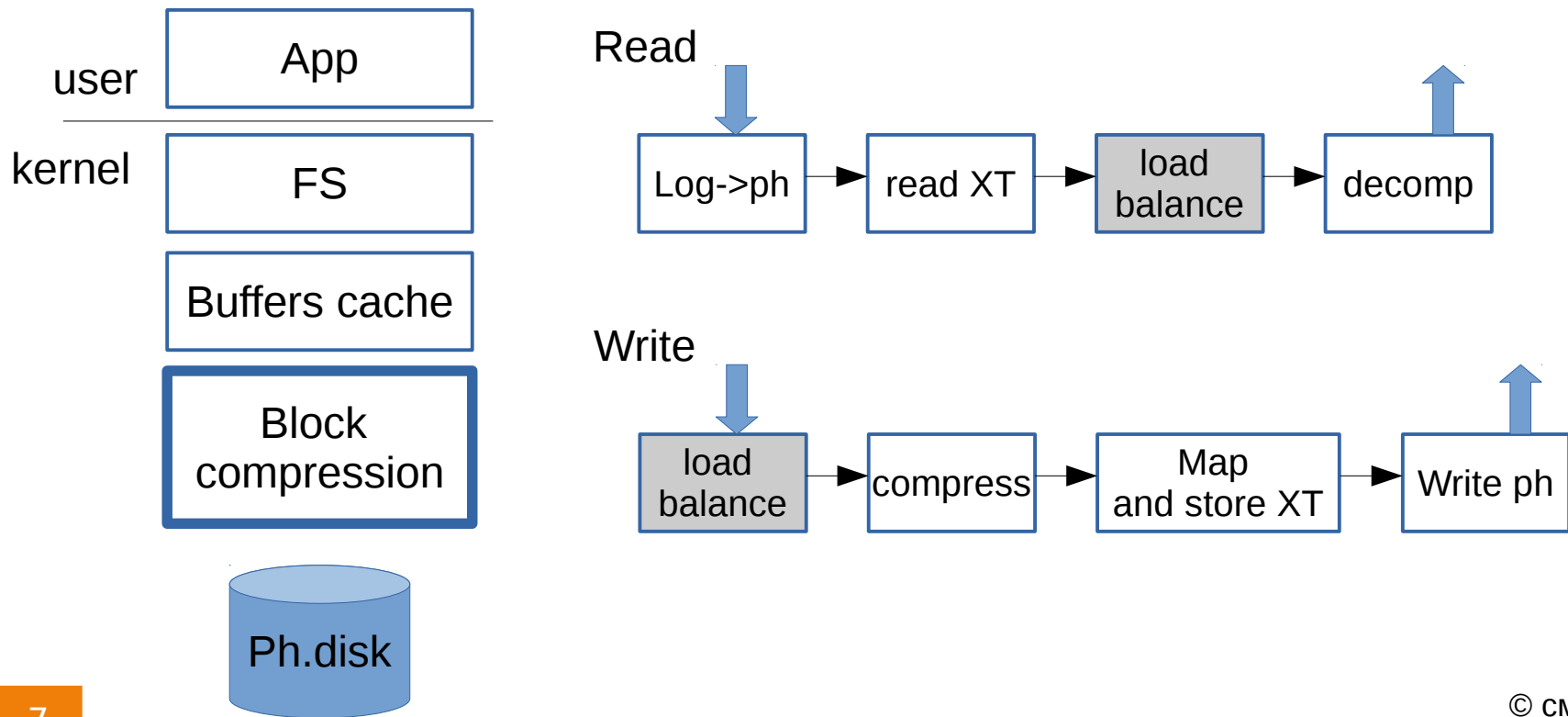
# Существующие работы

- Th. Makatos et al. **ZBD** – compressed block device, 2010-2012
- Y. Klonatos et al. **Azor** (HDD/SDD compression scheme), 2011
- Y. Cao et al. Block level memory compression, 2015

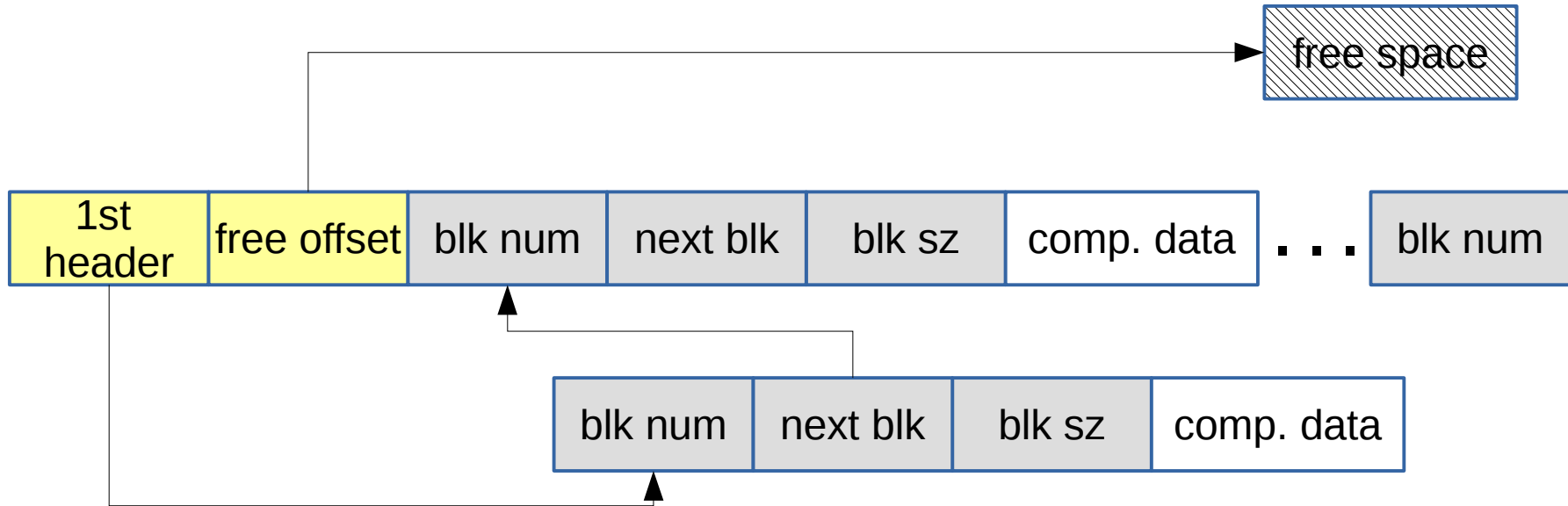
# Проблемы сжатия блоков

- Разный размер после сжатия
- Не все блоки нужно сжимать
- Требуется уровень трансляции
- Увеличение количества ввода-вывода (read-before-write)
- Фрагментация данных на диске

# Архитектура I/O

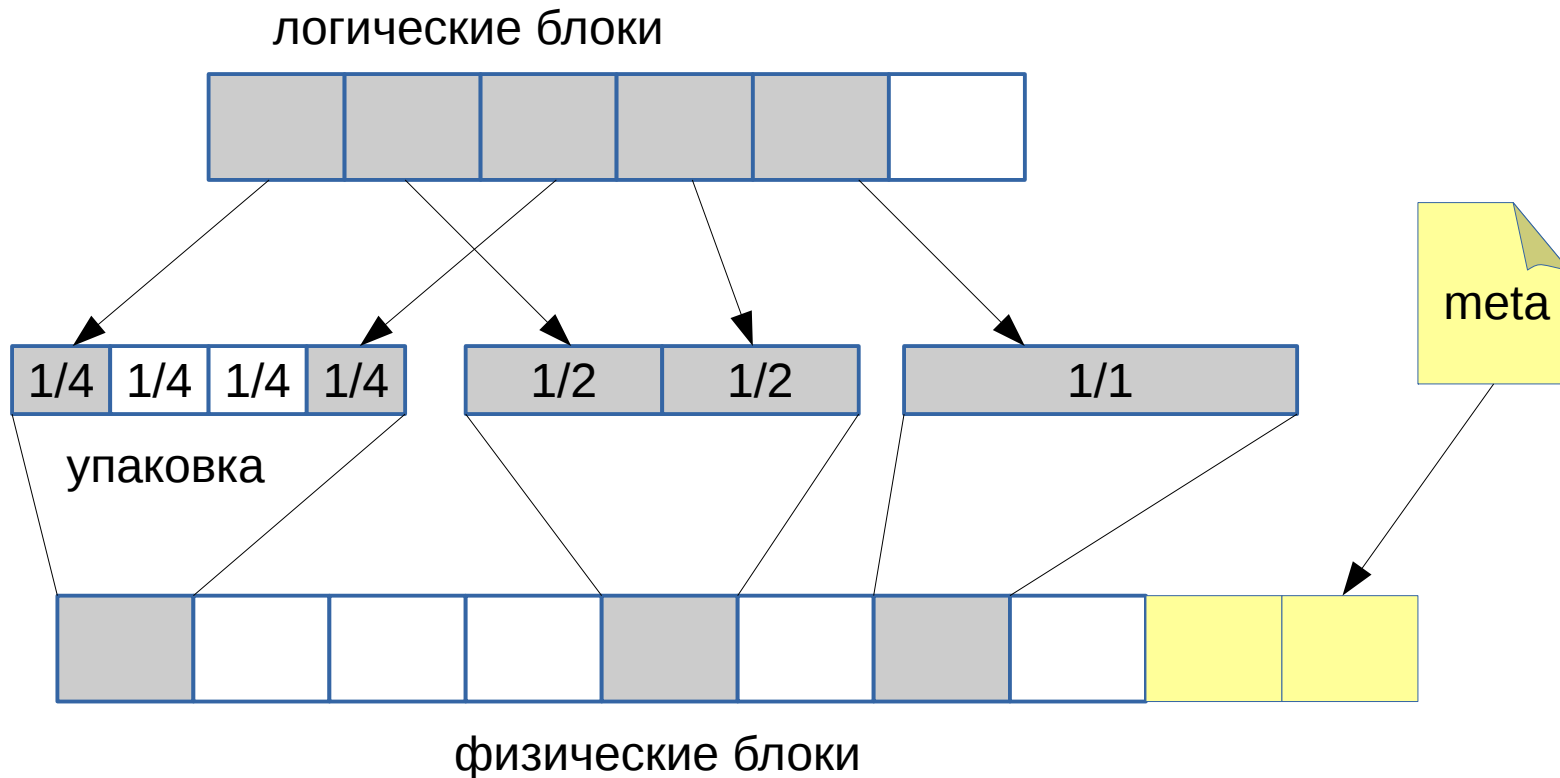


# Трансляция блоков: extents





# Трансляция блоков: упаковка

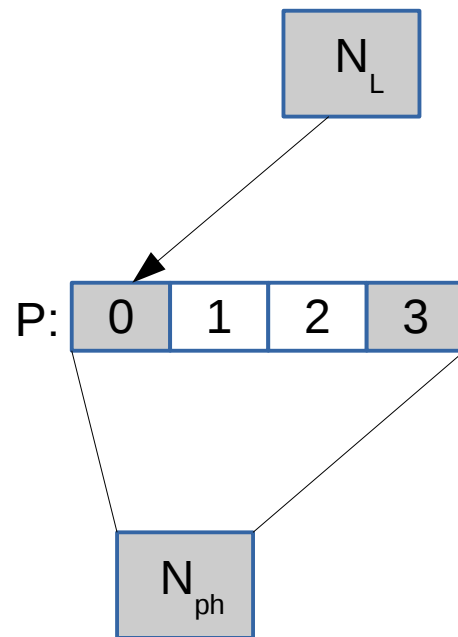


# Координатное пространство

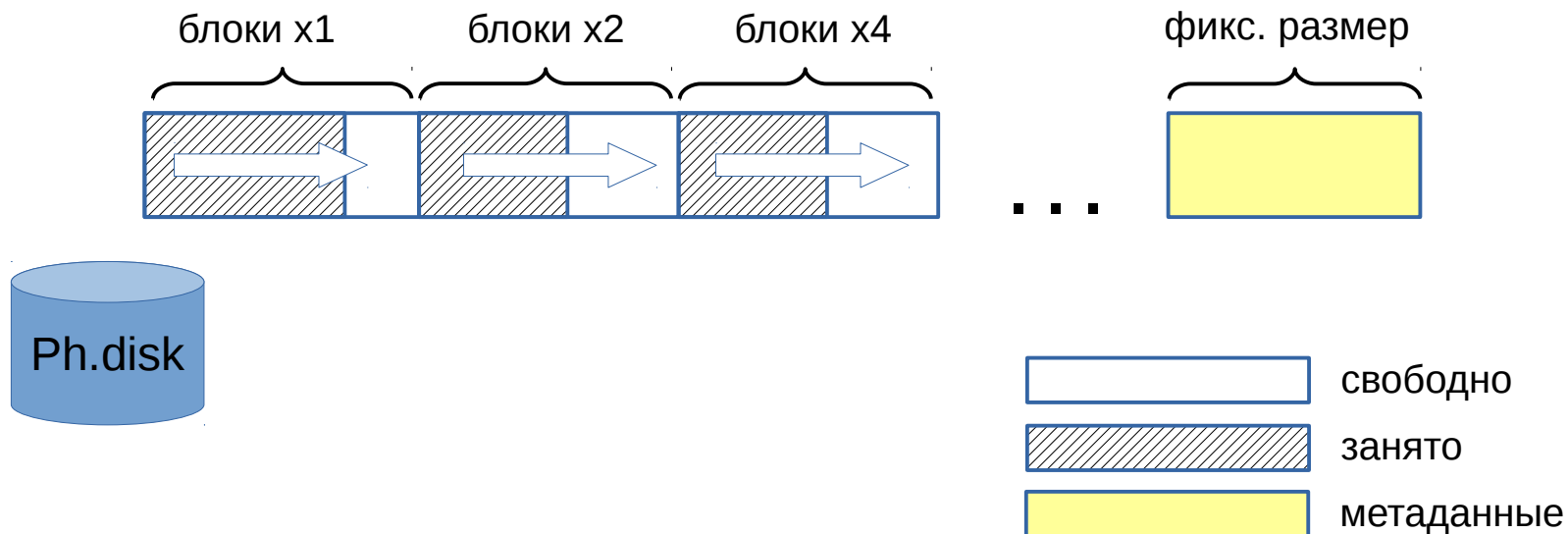
$$N_L \rightarrow \{N_{ph}, P\}$$

Где:

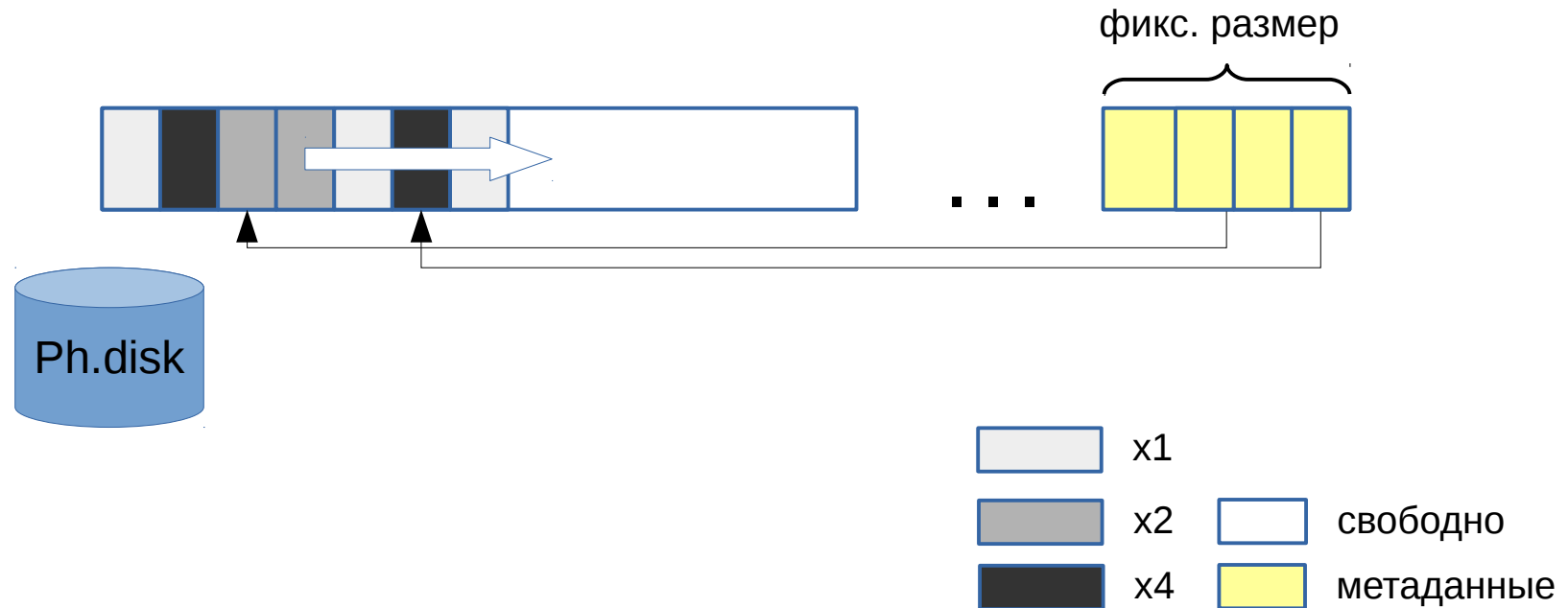
- $N_L$  - логический номер блока
- $N_{ph}$  - физический номер блока
- $P$  - позиция внутри физического блока



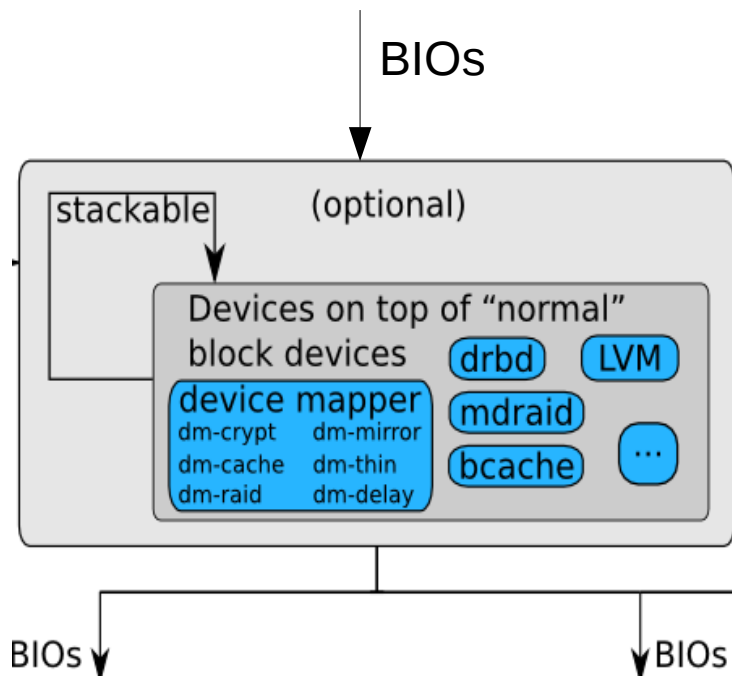
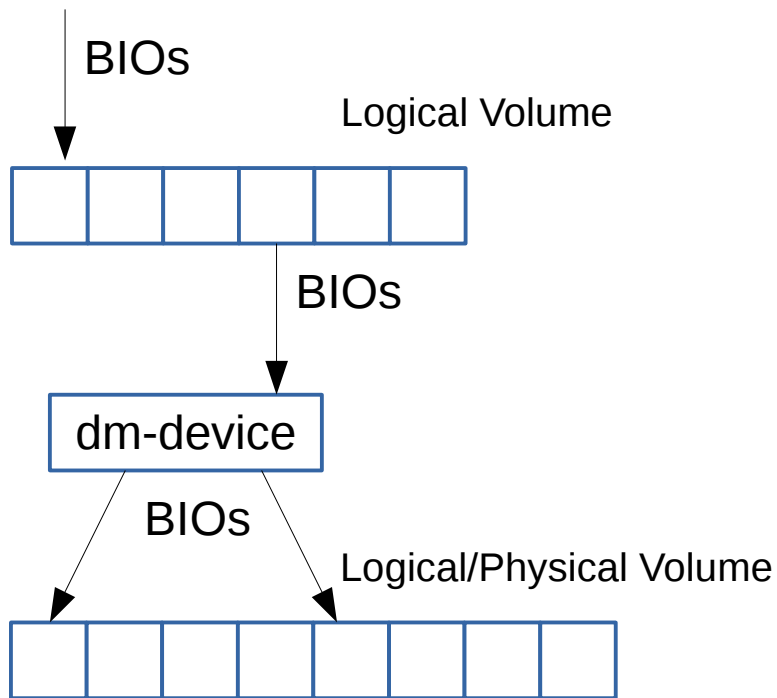
# Статическая разметка диска



# Динамическая разметка диска\*



# Device Mapper



# Device mapper iface

```
static struct target_type cbd_target = {  
    .name = "cbd_target",  
    .version = {1,0,0},  
    .ctr = cbd_ctr,  
    .dtr = cbd_dtr,  
    .map = cbd_map,  
};
```

```
static int cbd_ctr(struct dm_target *ti,  
                  unsigned int argc, char **argv)
```

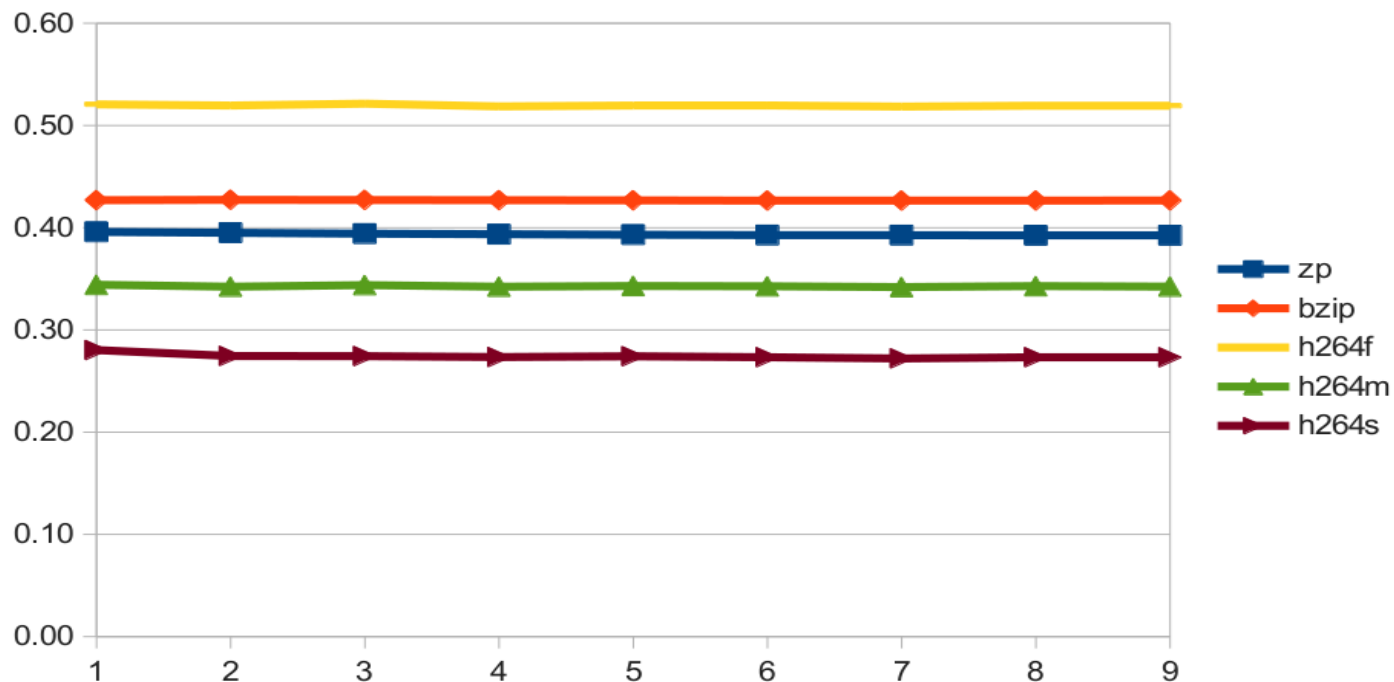
```
static void cbd_dtr(struct dm_target *ti)
```

```
static int cbd_map(struct dm_target *ti,  
                  struct bio *bio)
```

```
dm_target  
: dm_dev
```

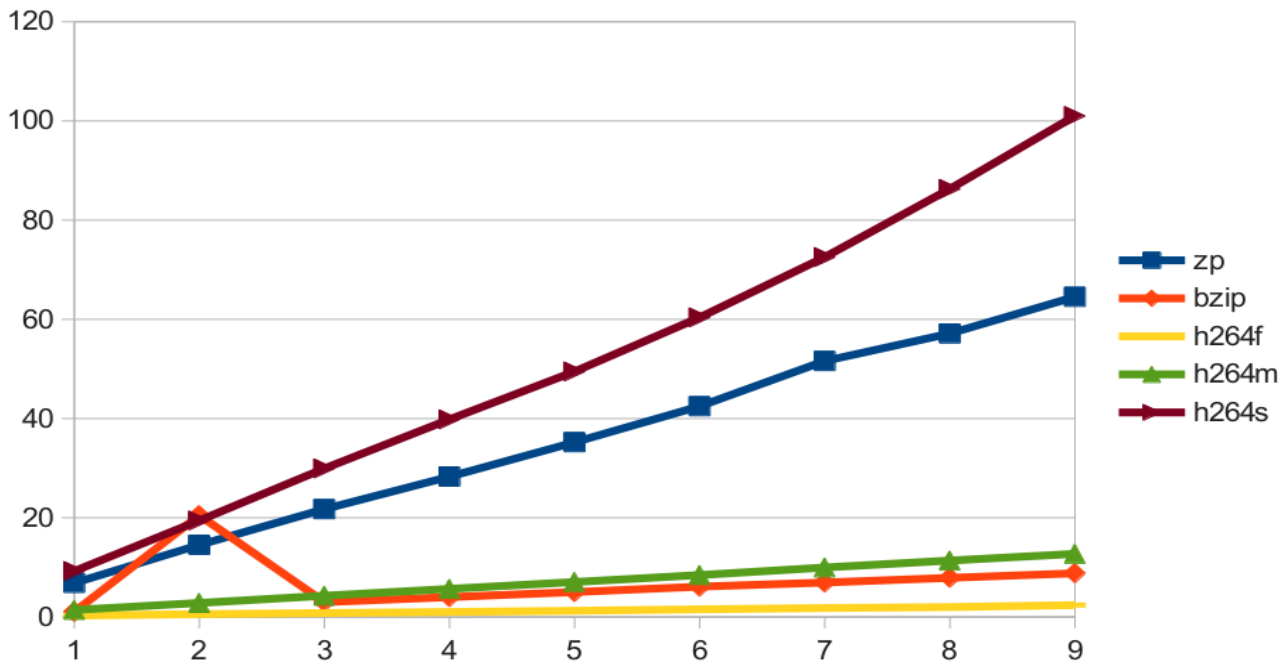
```
124 struct dm_dev {  
125     struct block_device *bdev;  
126     fmode_t mode;  
127     char name[16];  
128 };
```

# Компрессоры: степень сжатия



#	1	2	3	4	5	6	7	8	9
Sz	5M	10M	15M	20M	25M	30M	35M	40M	46M

# Компрессоры: время сжатия



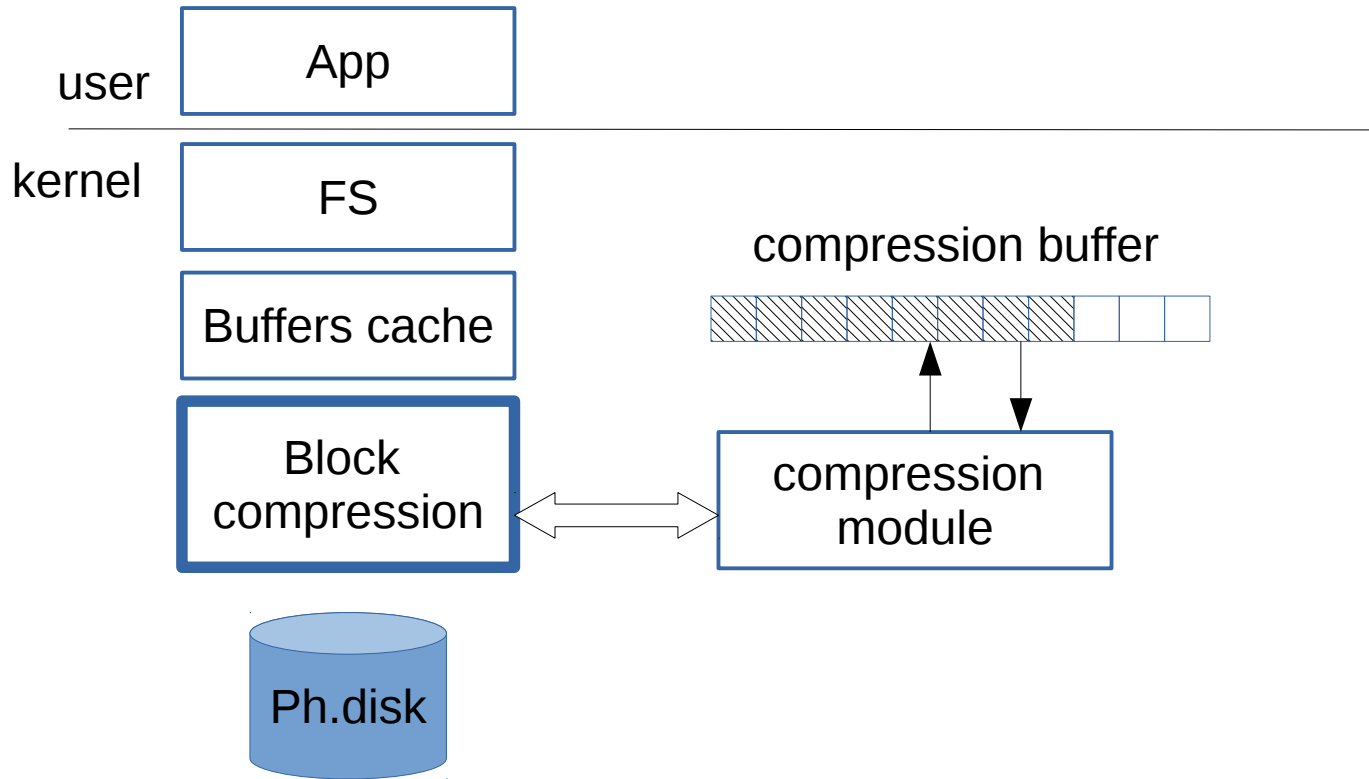
#	1	2	3	4	5	6	7	8	9
Sz	5M	10M	15M	20M	25M	30M	35M	40M	46M



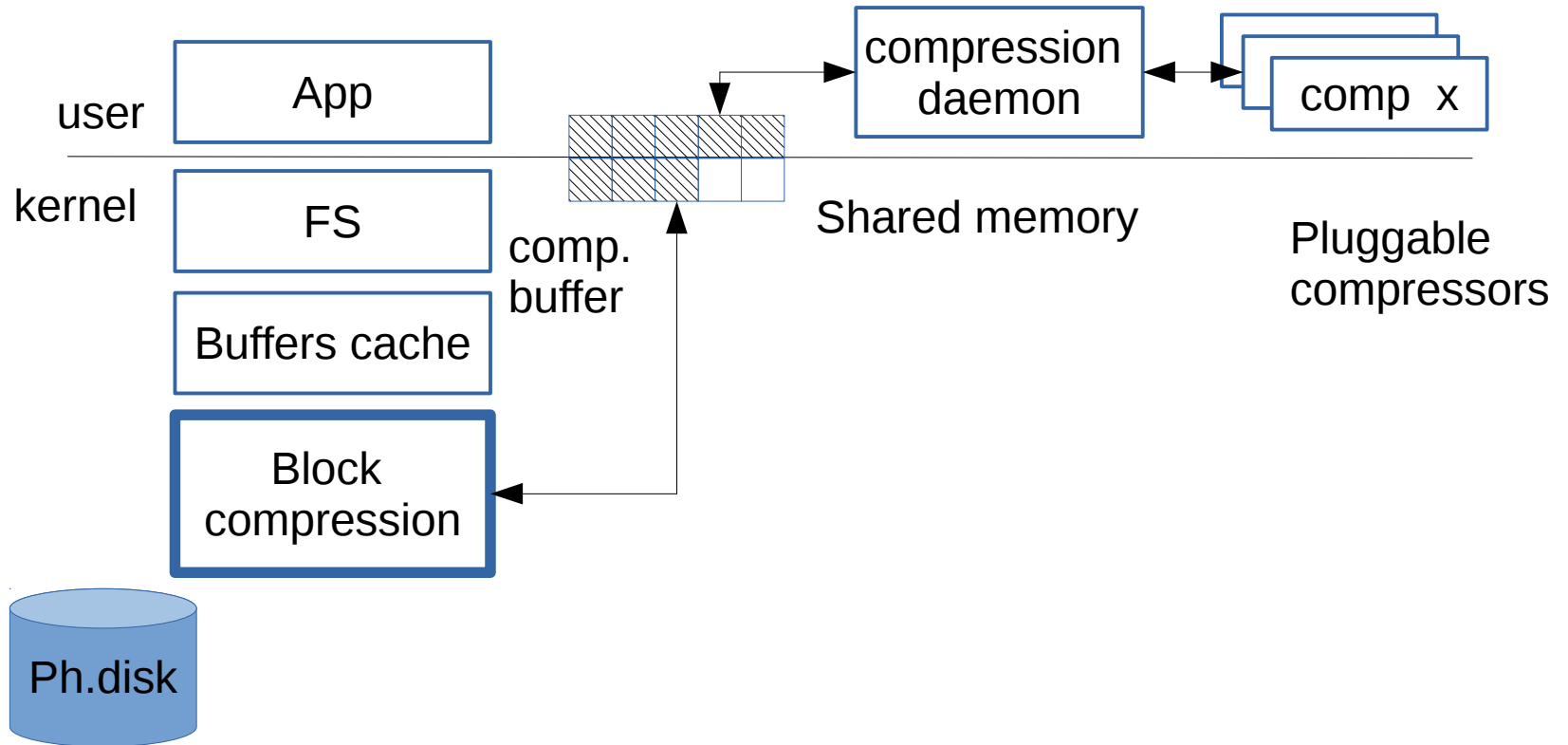
# Подключение компрессоров

- in kernel
- Shared memory + User space daemon

# In-kernel compressor



# User space compressor



# Сжатие блоков: “за” и “против”

- За:
  - прозрачное увеличение дискового пространства
  - лучшая (по сравнению с ФС) гранулярность объектов сжатия
  - скорость доступа\*
- Против:
  - фрагментация\*
  - затраты на хранение мета информации
  - отсутствие интеграции с ФС

# Развитие

- Балансировка нагрузки
- Тестирование производительности (HDD, SSD)
- Эффективное кэширование
- Использование аппаратного ускорения сжатия

# ИСТОЧНИКИ

- [1] Y. Klonatos et al. Transparent Online Storage Compression at the Block-Level. ACM Trans. Storage 8, 2, Article 5 (May 2012)
- [2] ZBD: Using Transparent Compression at the Block Level to Increase Storage Space Efficiency // 2010 International Workshop on Storage Network Architecture and Parallel I/Os
- [3] Y. Cao Flexible Memory: A Novel Main Memory Architecture with Block-level Memory Compression // Networking, Architecture and Storage (NAS), 2015 IEEE International Conference on
- [4] [www.thomas-krenn.com](http://www.thomas-krenn.com)

Вопросы?

*Кирилл Кринкин*  
*kirill.krinkin@gmail.com*

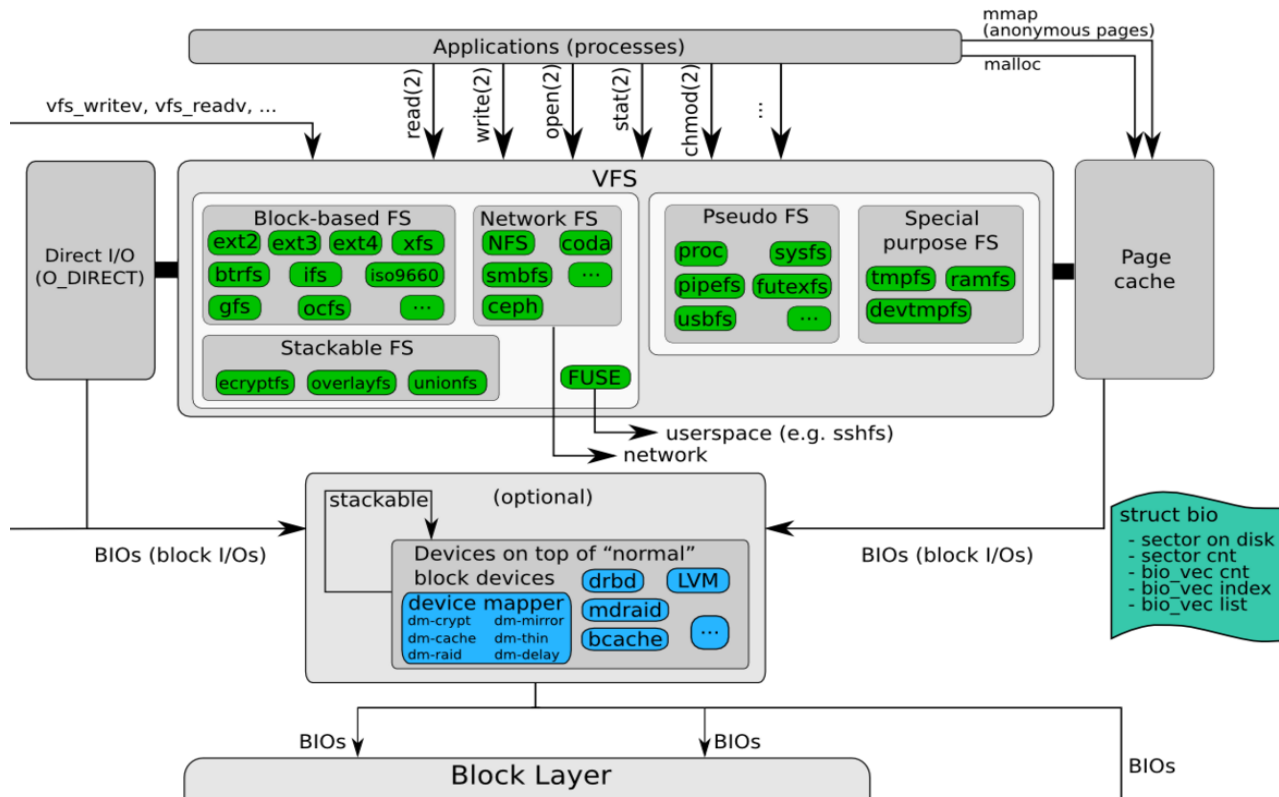
запасные слайды



# Intel DEFLATE

Vinodh Gopal et al. High Performance DEFLATE  
Compression on Intel ® Architecture Processors  
(igzip library) Nov 2011

# Возможности для компрессии



# Уровень BIOs

