

Системные баги под нагрузкой

Миша Жилин, Postgres Professional

Михаил Жилин

- Физтех, 2008 год выпуска
- 18 лет занимаюсь
производительностью программ
- Обожаю Open Source, Java,
PostgreSQL и FreeBSD
- Работаю 5,5 лет в Postgres
Professional



○ чём доклад

- ○ моём личном опыте



О чём доклад

- О моём личном опыте
- Нагрузочное тестирование с 2008 по 2020 года
 - Сервера приложений
 - Базы данных
 - Высоконагруженные решения



О чём доклад

- О моём личном опыте
- Нагрузочное тестирование с 2008 по 2020 года
 - Сервера приложений
 - Базы данных
 - Высоконагруженные решения
- В системном софте очень редко встречаются дефекты
 - Операционные системы работают как часы
 - Базы данных всегда возвращают корректный результат
 - Системный софт - надёжный, качественный, проверенный

О чём доклад

- Надежность: гарантия или иллюзия?



О чём доклад

- Надежность: гарантия или иллюзия?
- Поиск невозможного



О чём доклад

- Надежность: гарантия или иллюзия?
- Поиск невозможного
- Несколько историй из личного опыта
 - Включая много попыток поехать кукухой



О чём доклад

- Надежность: гарантия или иллюзия?
- Поиск невозможного
- Несколько историй из личного опыта
 - Включая много попыток поехать кукухой
- Много QR кодов (а может и немного)



Разгон

Intel SSD, 2019 год

Intel SSD, 2019 год

- Пробный период запуска мобильного оператора



Intel SSD, 2019 год

- Пробный период запуска мобильного оператора
- Отказ в распределенном дисковом пространстве CEPH



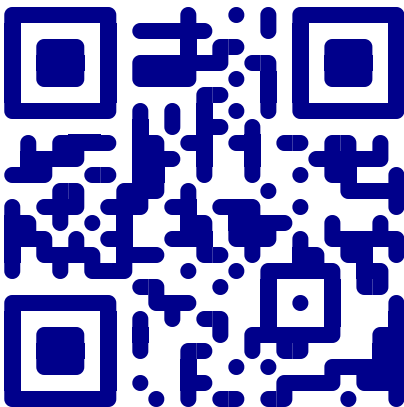
Intel SSD, 2019 год

- Пробный период запуска мобильного оператора
- Отказ в распределенном дисковом пространстве CEPH
- Зависание дисков Intel SSD S45210 и S4610



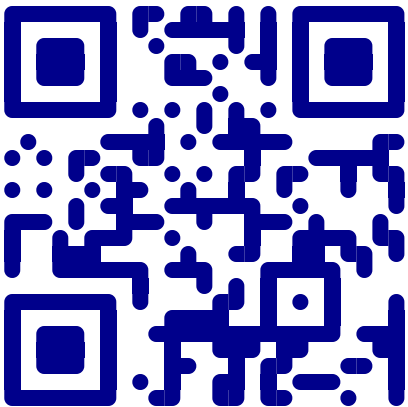
Intel SSD, 2019 год

- Intel SSD D3-S4510 and Intel SSD D3-S4610 series 1.92 TB and 3.84 TB drives become unresponsive after 1,700 cumulative idle power-on hours
- Currently addressed in the Maintenance Release 1 (MR1) of Intel firmware (version XCV10110) as of March 2019



Intel SSD, 2019 год

- Intel удалил все упоминания
- <https://forums.servethehome.com/index.php?threads/psa-errors-with-intel-r-solid-state-drive-dc-s4510-and-s4610-series.26196/>



WAL corruption, 2020 год

WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока



WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока
- Проблема от клиента



WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока
- Проблема от клиента
- База данных на 2 серверах: мастер и реплика



WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока
- Проблема от клиента
- База данных на 2 серверах: мастер и реплика
- Ошибка репликации
 - [2020-10-27 12:03:02 MSK u= d= h= p=14570 l=99]
 - LOG: incorrect resource manager data checksum in record at 61E7/ED365B20]

WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока
- Проблема от клиента
- База данных на 2 серверах: мастер и реплика
- Ошибка репликации
 - [2020-10-27 12:03:02 MSK u= d= h= p=14570 l=99]
 - LOG: incorrect resource manager data checksum in record at 61E7/ED365B20]
- Неверная контрольная сумма - повреждение WAL журналов

WAL corruption, 2020 год

- Смена работы
 - Инженер по отказоустойчивости в кластерной группе
 - 4ая неделя испытательного срока
- Проблема от клиента
- База данных на 2 серверах: мастер и реплика
- Ошибка репликации
 - [2020-10-27 12:03:02 MSK u= d= h= p=14570 l=99]
 - LOG: incorrect resource manager data checksum in record at 61E7/ED365B20]
- Неверная контрольная сумма - повреждение WAL журналов
- Материал для анализа: record at 61E7/ED365B20

003100 D0 BA D0 B0 D0 B7 D0 B0 D0 BD D0 B8 D0 B5 20 D1 83 D1 81 D0
003120 B6 00 72 07 12 00 30 40 37 31 38 38 20 30 33 30 32 20 30 38
003140 20 34 20 30 33 20 39 30 30 34 39 20 32 34 32 13 30 30 31 31 4 03 90049 242!!0011
003160 30 36 33 31 00 00 00 00 01 00 2F 00 1B 00 30 40 37 31 38 38 0631 © / ← 0@7188
003180 20 30 33 30 32 D1 82 D0 B8 D0 B7 D0 B8 D1 80 D0 BE D0 B2 D0 0302
003200 B0 D0 BD D0 BD D1 8B D1 85 20 D1 80 D0 B0 D0 B1 D0 BE D1 87

003100 D0 BA D0 B0 D0 B7 D0 B0 D0 BD D0 B8 D0 B5 20 D1 83 D1 81 D0
003120 B6 D1 83 D0 B3 20 D0 B2 20 D1 81 D1 84 D0 B5 D1 80 D0 B5 20
003140 D0 98 D0 9A D0 A2 20 D0 BF D0 BE 20 D0 B0 D1 82 D1 82 D0 B5
003160 D1 81 D1 82 D0 B0 D1 86 D0 B8 D0 B8 20 D0 B0 D0 B2 D1 82 D0
003180 BE D0 BC D0 B0 D1 82 D0 B8 D0 B7 D0 B8 D1 80 D0 BE D0 B2 D0
003200 B0 D0 BD D0 BD D1 8B D1 85 20 D1 80 D0 B0 D0 B1 D0 BE D1 87

WAL corruption, 2020 год

- Ошибка диска/OS: и всё что над ним RAID контролер, драйвера, блочной системы, файловой системы



WAL corruption, 2020 год

- Ошибка диска/OS: и всё что над ним RAID контролер, драйвера, блочной системы, файловой системы
- Ошибка СУБД (не исключена вероятность “гуляющего” указателя)

WAL corruption, 2020 год

- Ошибка диска/OS: и всё что над ним RAID контролер, драйвера, блочной системы, файловой системы
- Ошибка СУБД (не исключена вероятность “гуляющего” указателя)
- **Ошибка памяти:**
 - адрес повреждения в WAL файле – 0x366500 (выровнен на 256 байт)
 - размер повреждения - 64 байта (размер кэшлайна процессора)

WAL corruption, 2020 год

- Через месяц ещё одно падение



WAL corruption, 2020 год

- Через месяц ещё одно падение
- Потом ещё несколько



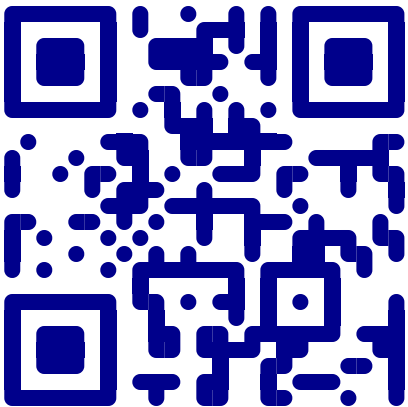
WAL corruption, 2020 год

- Через месяц ещё одно падение
- Потом ещё несколько
- Но всегда портился только WAL, данные проверяли - они все верные



WAL corruption, 2020 год

- Двойной WAL буфер - защита от disk/memory corruption
- Настройка `wal_sender_check_crc` в PostgresPro Enterprise



Наступил 2021 год

WAL corruption, 2021 год

- Изоляция проблемного сервера и доступ для экспериментов



WAL corruption, 2021 год

- Изоляция проблемного сервера и доступ для экспериментов
- BullSequana S series
 - Несколько материнских плат
 - 4 терабайта памяти
 - 8 процессоров



WAL corruption, 2021 год

- Изоляция проблемного сервера и доступ для экспериментов
- BullSequana S series
 - Несколько материнских плат
 - 4 терабайта памяти
 - 8 процессоров
- Недельные тесты
 - Ноль результатов



WAL corruption, 2021 год

- Изоляция проблемного сервера и доступ для экспериментов
- BullSequana S series
 - Несколько материнских плат
 - 4 терабайта памяти
 - 8 процессоров
- Недельные тесты
 - Ноль результатов
- MemTest
 - Ноль результатов



WAL corruption, 2021 год

- Полный ноль



WAL corruption, 2021 год

- Полный ноль
- Фрустрации



WAL corruption, 2021 год

- Полный ноль
- Фрустрации
 - А что ищем?



WAL corruption, 2021 год

- Полный ноль
- Фрустрации
 - А что ищем?
 - Как задать вопрос?



WAL corruption, 2021 год

- Полный ноль
- Фрустрации
 - А что ищем?
 - Как задать вопрос?
 - Что я не понимаю?



WAL corruption, 2021 год

- Полный ноль
- Фрустрации
 - А что ищем?
 - Как задать вопрос?
 - Что я не понимаю?
- Google it, опытные коллеги



WAL corruption, 2021 год

- Полный ноль
- Фрустрации
 - А что ищем?
 - Как задать вопрос?
 - Что я не понимаю?
- Google it, опытные коллеги
- Время менять подходы



WAL corruption, 2021 год

- Основная гипотеза - это память



WAL corruption, 2021 год

- Основная гипотеза - это память
- stress-ng - утилита нагрузки на железные ресурсы

- `stress-ng --vm-rw 1000 --vm-rw-bytes 2500G --verify --metrics-brief -t 60m`



WAL corruption, 2021 год

- Основная гипотеза - это память
- stress-ng - утилита нагрузки на железные ресурсы
 - `stress-ng --vm-rw 1000 --vm-rw-bytes 2500G --verify --metrics-brief -t 60m`
- Никаких ошибок программы



WAL corruption, 2021 год

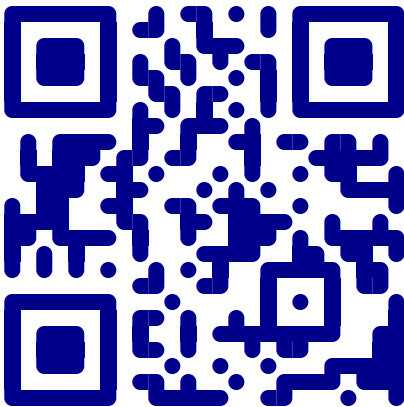
- Основная гипотеза - это память
- stress-ng - утилита нагрузки на железные ресурсы
 - `stress-ng --vm-rw 1000 --vm-rw-bytes 2500G --verify --metrics-brief -t 60m`
- Никаких ошибок программы
- Зато Fiber Channel начал мигать
 - `2021 Feb 15 23:43:57 dn-MDS-C9148S-1 %PORT-5-IF_DOWN_LINK_FAILURE: %$VSAN2221%$ Interface fc1/15 is down (Link failure loss of signal)`

WAL corruption, 2021 год

- Основная гипотеза - это память
- stress-ng - утилита нагрузки на железные ресурсы
 - `stress-ng --vm-rw 1000 --vm-rw-bytes 2500G --verify --metrics-brief -t 60m`
- Никаких ошибок программы
- Зато Fiber Channel начал мигать
 - `2021 Feb 15 23:43:57 dn-MDS-C9148S-1 %PORT-5-IF_DOWN_LINK_FAILURE: %$VSAN2221%$ Interface fc1/15 is down (Link failure loss of signal)`
- Время и идеи закончились

WAL corruption, 2021 год

- Через полгода статья от техподдержки производителя железа
 - <https://habr.com/ru/companies/atosservers/articles/562376/>
- Обновление драйвера FC линков (драйвер устройства операционной системы Linux).

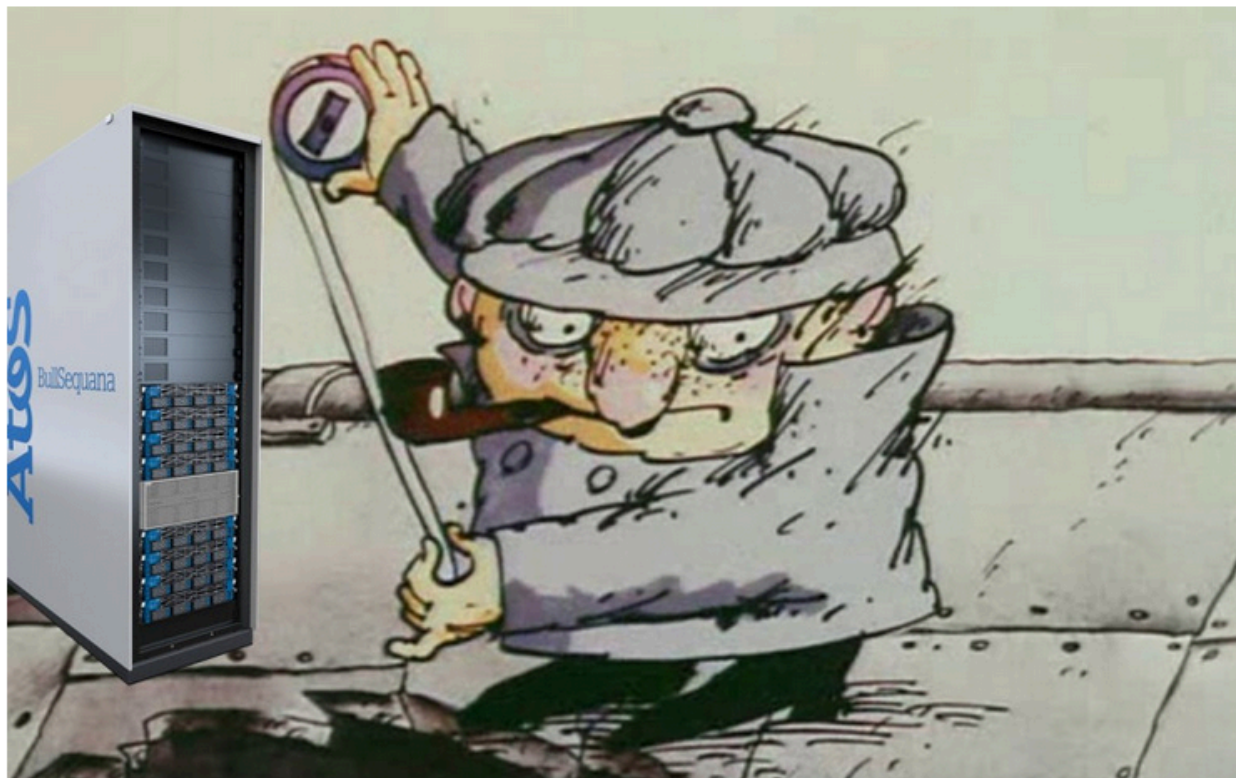


sharkattack 18 июн 2021 в 18:24

Следствие вели: пропажа FC-линков HBA Emulex на сервере Atos BullSequana S1600

🕒 6 мин 👁 3.8К

Блог компании Atos, Высоконагруженные системы*, Тестирование IT-систем*, IT-компании



WAL corruption, 2021 год

- Цитаты
 - "За всё время этой проблемы пошатали и потрогали всё – саму ОС, FW, прошивку HW сервера, настройки HW сервера, параметры GRUB, настройки фабрики, свитчей и линков..."
 - "Техподдержка врёт (ну или добросовестно заблуждается), поэтому приходится старательно все проверять самому"
 - "Трогать и шатать при траблшутинге надо всё!"

WAL corruption, 2021 год

- Итоги
 - Обновления драйвера и реализации двойного буфера
 - Больше повреждений WAL не было
 - Кто виноват - до сих пор загадка
 - Это успех или нет?
 - Негативный опыт - тоже опыт



Прошли годы

Прошли годы

- Меня выбросили из кластерной группы как непрофпригодного
 - "Иди делай группу нагрузочных исследователей" 😜

Прошли годы

- Меня выбросили из кластерной группы как непрофпригодного
 - "Иди делай группу нагрузочных исследователей" 😜
- Коллаборация с нашей техподдержкой
 - Они нам задачи
 - Мы им помощь в исследованиях



Прошли годы

- Меня выбросили из кластерной группы как непрофпригодного
 - "Иди делай группу нагрузочных исследователей" 😜
- Коллаборация с нашей техподдержкой
 - Они нам задачи
 - Мы им помощь в исследованиях
- История с подвисанием сервера на Linux 5.15

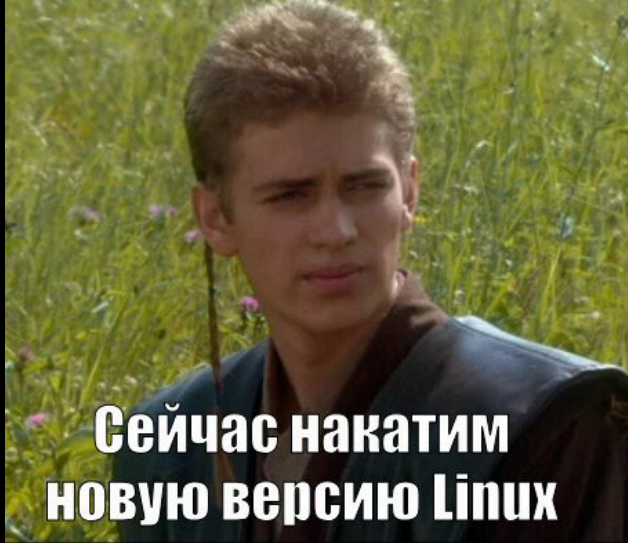


Прошли годы

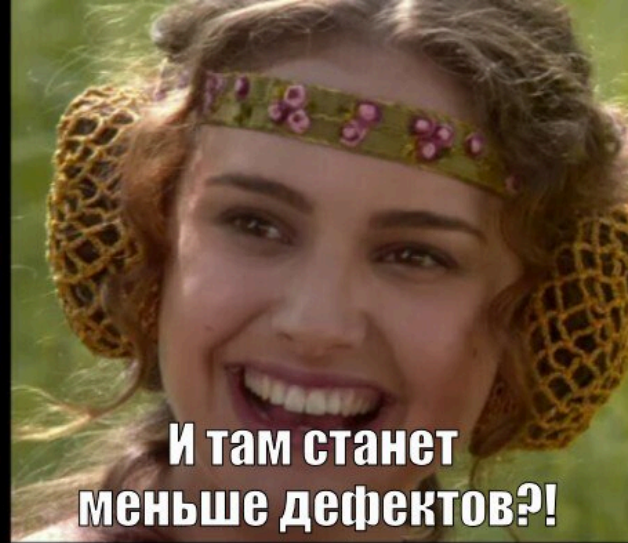
- Меня выбросили из кластерной группы как непрофпригодного
 - "Иди делай группу нагрузочных исследователей" 😜
- Коллаборация с нашей техподдержкой
 - Они нам задачи
 - Мы им помощь в исследованиях
- История с подвисанием сервера на Linux 5.15
- Проблема масштабирования на lookup-а файлов в XFS

Прошли годы

- Меня выбросили из кластерной группы как непрофпригодного
 - "Иди делай группу нагрузочных исследователей" 😜
- Коллаборация с нашей техподдержкой
 - Они нам задачи
 - Мы им помощь в исследованиях
- История с подвисанием сервера на Linux 5.15
- Проблема масштабирования на lookup-а файлов в XFS
- **Общая рекомендация:** обновитесь до новой версии операционной системы



**Сейчас накатим
новую версию Linux**



**И там станет
меньше дефектов?!**



segfault, 2024

segfault, 2024

- 30 декабрь 2024 года



segfault, 2024

- 30 декабрь 2024 года
- Миграция данных



segfault, 2024

- 30 декабрь 2024 года
- Миграция данных
- Упала база данных по segfault



segfault, 2024

- 30 декабрь 2024 года
- Миграция данных
- Упала база данных по segfault
- ```
[Mon Dec 30 17:49:50 2024] postgres[4092660]: segfault at 55edf40f20d0 ip
000055edf40f20d0 sp 00007ffc7a7773c8 error 15 in
postgres[55edf3f67000+2a9000] likely on CPU 40 (core 16, socket 1)
```

# segfault, 2024

## ■ checkpointer (4092660)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c1ada8, keyPtr=0x7ffc7a777480, action=HASH_ENTER, foundPtr=0x7ffc7a777480) at hashtable.c:131
#2 0x000055edf3dd2dd2 in RememberSyncRequest (ftag=0x7ffc7a777480, type=<optimized out>) at sync.c:565
#3 0x000055edf3dd2fb5 in RegisterSyncRequest (ftag=ftag@entry=0x7ffc7a777480, type=type@entry=SYNC_REQUEST, retryOnError=retryOnError@entry=0) at sync.c:611
```

## ■ startup (4092662)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0) at hashtable.c:131
#2 0x000055edf3b40603 in XLogPrefetcherIsFiltered (blockno=1523809, rnode=... , prefetcher=0x55edf5c375f8) at xlogprefetcher.c:101
#3 XLogPrefetcherNextBlock (pgsr_private=94480518837752, lsn=<optimized out>) at xlogprefetcher.c:693
#4 0x000055edf3b40f51 in lrq_prefetch (lrq=<optimized out>) at xlogprefetcher.c:256
#5 lrq_complete_lsn (lsn=67516543241680, lrq=0x55edf5c3aff8) at xlogprefetcher.c:294
```

# segfault, 2024

## ■ checkpointer (4092660)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c1ada8, keyPtr=0x7ffc7a777480, action=HASH_ENTER, foundPtr=0x7ffc7a777480) at hash.c:102
#2 0x000055edf3dd2dd2 in RememberSyncRequest (ftag=0x7ffc7a777480, type=<optimized out>) at sync.c:565
#3 0x000055edf3dd2fb5 in RegisterSyncRequest (ftag=ftag@entry=0x7ffc7a777480, type=type@entry=SYNC_REQUEST, retryOnError=retryOnError@entry=0) at sync.c:612
```

## ■ startup (4092662)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0) at hash.c:102
#2 0x000055edf3b40603 in XLogPrefetcherIsFiltered (blockno=1523809, rnode=... , prefetcher=0x55edf5c375f8) at xlogprefetcher.c:102
#3 XLogPrefetcherNextBlock (pgsr_private=94480518837752, lsn=<optimized out>) at xlogprefetcher.c:693
#4 0x000055edf3b40f51 in lrq_prefetch (lrq=<optimized out>) at xlogprefetcher.c:256
#5 lrq_complete_lsn (lsn=67516543241680, lrq=0x55edf5c3aff8) at xlogprefetcher.c:294
```

# segfault, 2024

## ■ checkpointer (4092660)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c1ada8, keyPtr=0x7ffc7a777480, action=HASH_ENTER, foundPtr=0x7ffc7a777480) at hashtable.c:102
#2 0x000055edf3dd2dd2 in RememberSyncRequest (ftag=0x7ffc7a777480, type=<optimized out>) at sync.c:565
#3 0x000055edf3dd2fb5 in RegisterSyncRequest (ftag=ftag@entry=0x7ffc7a777480, type=type@entry=SYNC_REQUEST, retryOnError=retryOnError@entry=0) at sync.c:605
```

## ■ startup (4092662)

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0) at hashtable.c:102
#2 0x000055edf3b40603 in XLogPrefetcherIsFiltered (blockno=1523809, rnode=..., prefetcher=0x55edf5c375f8) at xlogprefetcher.c:102
#3 XLogPrefetcherNextBlock (pgsr_private=94480518837752, lsn=<optimized out>) at xlogprefetcher.c:693
#4 0x000055edf3b40f51 in lrq_prefetch (lrq=<optimized out>) at xlogprefetcher.c:256
#5 lrq_complete_lsn (lsn=67516543241680, lrq=0x55edf5c3aff8) at xlogprefetcher.c:294
```

# segfault, 2024

- Два процесса одновременно



# segfault, 2024

- Два процесса одновременно
- `hash_search`
  - Фундаментальный код
  - Общий с PostgreSQL
  - Часто выполняемый aka горячий код



# segfault, 2024

- Два процесса одновременно
- hash\_search
  - Фундаментальный код
  - Общий с PostgreSQL
  - Часто выполняемый aka горячий код
- Трудность



# segfault, 2024

- Начинаем искать зацепки



# segfault, 2024

- Начинаем искать зацепки
- Вводные
  - Дамп памяти
  - Конечное состояние регистров
  - Ассемблеровский код



# segfault, 2024

- Начинаем искать зацепки
- Вводные
  - Дамп памяти
  - Конечное состояние регистров
  - Ассемблеровский код
- Поиск непротиворечивого сценария



# segfault, 2024

- Начинаем искать зацепки
- Вводные
  - Дамп памяти
  - Конечное состояние регистров
  - Ассемблеровский код
- Поиск непротиворечивого сценария
- Никаких вариантов, кроме одного нереального



# segfault, 2024

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0)
```

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi
0x000055edf3f5871c <+140>: add %rdi,%rsi
0x000055edf3f5871f <+143>: jmp *%rsi
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0)
```

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi
0x000055edf3f5871c <+140>: add %rdi,%rsi
0x000055edf3f5871f <+143>: jmp *%rsi
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0)
```

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi
0x000055edf3f5871c <+140>: add %rdi,%rsi
0x000055edf3f5871f <+143>: jmp *%rsi
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
#0 0x000055edf40f20d0 in ?? ()
#1 0x000055edf3f1a370 in hash_search (hashp=0x55edf5c458f8, keyPtr=0x7ffc7a7781ec, action=HASH_FIND, foundPtr=0x0)
```

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi
0x000055edf3f5871c <+140>: add %rdi,%rsi
0x000055edf3f5871f <+143>: jmp *%rsi
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq(%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq(%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq (%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

```
0x000055edf3f5870b <+123>: cmp $0xb,%r10d
0x000055edf3f5870f <+127>: ja 0x55edf3f586b0 <hash_bytes+32>
0x000055edf3f58711 <+129>: lea 0x1999b8(%rip),%rdi # %rdi = (%rip+0x1999b8) | %rdi = 0x55edf40f20d0
0x000055edf3f58718 <+136>: movslq(%rdi,%r10,4),%rsi # %rsi = mem(%rdi+(%r10*4)) | %r10 = 0
0x000055edf3f5871c <+140>: add %rdi,%rsi # %rsi = %rsi + %rdi | %rsi = 0x55edf40f20d0
0x000055edf3f5871f <+143>: jmp *%rsi # jump to %rsi | segfault
0x000055edf3f58721 <+145>: nopl 0x0(%rax)
```

# segfault, 2024

- Процессор зачитал значение равное **0** из адреса `%rdi+%r10*4`
- Должен был зачитать **0xffe667a7**

```
(gdb) p/x *(unsigned*)0x55edf40f20d0@12
$9 = {0xffe667a7, ... }
```

# segfault, 2024

- Процессор зачитал значение равное **0** из адреса `%rdi+%r10*4`
- Должен был зачитать **0xffe667a7**

```
(gdb) p/x *(unsigned*)0x55edf40f20d0@12
$9 = {0xffe667a7, ... }
```

# segfault, 2024

- 31 декабря, 2:30 ночи: документ с анализом



# segfault, 2024

- 31 декабря, 2:30 ночи: документ с анализом
- "Главное - мы видим аномалию в работе системы, по какой-то причине процессор зачитал нулевое значения из ненулевой памяти (которая постоянная, никогда не меняется)"



**Конечно никто не собирался  
нам поверить**

# segfault, 2024

- Документ ушёл в ТП вендора ОС



# segfault, 2024

- Документ ушёл в ТП вендора ОС
- Ответ



# segfault, 2024

- Документ ушёл в ТП вендора ОС
- Ответ
  - **"Причины SIGSEGV во всех случаях (внезапно!) не внешние, а со стороны userspace, вызванные процессом postgres"**

# segfault, 2024

- Документ ушёл в ТП вендора ОС
- Ответ
  - **"Причины SIGSEGV во всех случаях (внезапно!) не внешние, а со стороны userspace, вызванные процессом postgres"**
- Так заканчивался 2024 год...

segfault, 2025

# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти
- Гипотезы рождались как грибы



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти
- Гипотезы рождались как грибы
  - алгоритмы сжатия



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти
- Гипотезы рождались как грибы
  - алгоритмы сжатия
  - программные raid контроллеры



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти
- Гипотезы рождались как грибы
  - алгоритмы сжатия
  - программные raid контроллеры
  - прошивки



# segfault, 2025

- Новые падения (6 января) - случайный места, случайные программы (sshd)
- Больше доказательство о нулевой памяти
- Гипотезы рождались как грибы
  - алгоритмы сжатия
  - программные raid контроллеры
  - прошивки
  - FPU



# segfault, 2025

- Падения продолжались (28 января, 7 февраля)



# segfault, 2025

- Падения продолжались (28 января, 7 февраля)
- **rasdaemon**: RAS (Reliability, Availability and Serviceability) logging tool

# segfault, 2025

- Падения продолжались (28 января, 7 февраля)
- **rasdaemon**: RAS (Reliability, Availability and Serviceability) logging tool
  - Тук-тук, у вас битые планки памяти!

# segfault, 2025

- Падения продолжались (28 января, 7 февраля)
- **rasdaemon**: RAS (Reliability, Availability and Serviceability) logging tool
  - Тук-тук, у вас битые планки памяти!
  - Эх, опять мимо

# segfault, 2025

- Падения продолжались (28 января, 7 февраля)
- **rasdaemon**: RAS (Reliability, Availability and Serviceability) logging tool
  - Тук-тук, у вас битые планки памяти!
  - Эх, опять мимо
- Пали разные сервера



# segfault, 2025

- Падения продолжались (28 января, 7 февраля)
- **rasdaemon**: RAS (Reliability, Availability and Serviceability) logging tool
  - Тук-тук, у вас битые планки памяти!
  - Эх, опять мимо
- Падали разные сервера
- Делали анализ, писали отчёты, всегда одно и тоже - **обнуление памяти!**



**СЛОЖИТЬ  
ЛАПКИ**



**ПОИГРАТЬСЯ  
СО СТЕНДОМ**

# segfault, 2025

- Стенд от производителя железа



# segfault, 2025

- Стенд от производителя железа
- **Спасибо** за оперативность



# segfault, 2025

- Стенд от производителя железа
- **Спасибо** за оперативность
- Одно из падений: перестроение индексов в базе данных (rebuild index)



# segfault, 2025

- Стенд от производителя железа
- **Спасибо** за оперативность
- Одно из падений: перестроение индексов в базе данных (rebuild index)
- Делаем большую синтетическую базу



# segfault, 2025

- Стенд от производителя железа
- **Спасибо** за оперативность
- Одно из падений: перестроение индексов в базе данных (rebuild index)
- Делаем большую синтетическую базу
- Добавляем проверку памяти (memtester)



## ■ 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

bash упал 11:18:44 ✓

Что сделал:

1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```

2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```

3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов

bash упал 11:18:44 ✓

Что сделал:

- 1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:  

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```
- 2) На базе поставил maintenance\_work\_mem в 96GB:  

```
sql
alter system set maintenance_work_mem to '96GB';
```
- 3) В параллель запустил:  

```
copy
reindexdb -j 400
```

и  

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов
- 1000 итераций проверки 680 гигабайт памяти

bash упал 11:18:44 ✓

Что сделал:

1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```

2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```

3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов
- 1000 итераций проверки 680 гигабайт памяти
- Упал не postgres, а bash

bash упал 11:18:44 ✓

Что сделал:

1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```

2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```

3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов
- 1000 итераций проверки 680 гигабайт памяти
- Упал не postgres, а bash
- Воспроизводимый кейс

bash упал 11:18:44 ✓

Что сделал:

1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```

2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```

3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов
- 1000 итераций проверки 680 гигабайт памяти
- Упал не postgres, а bash
- Воспроизводимый кейс
- Маленькая победа!

bash упал 11:18:44 ✓

Что сделал:

- 1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```
- 2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```
- 3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

- 25 февраля

```
reindexdb -j 400
memtester 680G 1000
```

- 400 процессов перестроения индексов
- 1000 итераций проверки 680 гигабайт памяти
- Упал не postgres, а bash
- Воспроизводимый кейс
- Маленькая победа!
- Но что дальше?

bash упал 11:18:44 ✓

Что сделал:

- 1) поднял базу которая лежала в /MBD\_DATA/pgdata. Там 100GB нужно было в huge pages, их туда добавил:

```
shell
echo 51543 > /proc/sys/vm/nr_hugepages
/opt/pgpro/ent-15/bin/pg_ctl -D /MBD_DATA/pgdata start
```
- 2) На базе поставил maintenance\_work\_mem в 96GB:

```
sql
alter system set maintenance_work_mem to '96GB';
```
- 3) В параллель запустил:

```
copy
reindexdb -j 400
```

и

```
copy
memtester 680G 1000
```

Прошло буквально несколько секунд-минут и reindex отвалился. В этот момент всё остановил и увидел в dmesg что был crash самого bash

11:22:07 ✓

# segfault, 2025

- Проверка нового оборудования



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС
  - Увеличение памяти с 1TiB до 4TiB



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС
  - Увеличение памяти с 1TiB до 4TiB
  - Замена software RAID на hardware RAID



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС
  - Увеличение памяти с 1TiB до 4TiB
  - Замена software RAID на hardware RAID
- Ура! Никаких больше падений



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС
  - Увеличение памяти с 1TiB до 4TiB
  - Замена software RAID на hardware RAID
- Ура! Никаких больше падений
- Успешный выход проекта в мае-июне



# segfault, 2025

- Проверка нового оборудования
- Глобальное обновление железа (Апрель-Май)
  - Смена ОС
  - Увеличение памяти с 1TiB до 4TiB
  - Замена software RAID на hardware RAID
- Ура! Никаких больше падений
- Успешный выход проекта в мае-июне
- **Увы, никто тогда не знал что причина была не в железе...**



КОНЕЦ

"Конец, конец... Концы в воду!"



**"Иди, и без ёлки не возвращайся"**

# segfault, 2025

- Осень 2025



# segfault, 2025

- Осень 2025
- Другие клиенты начали жаловаться на падения
  - на других ОС
  - на другом железе
  - и разные клиенты
  - и не только PostgresPro, но PostgreSQL



# segfault, 2025

- Осень 2025
- Другие клиенты начали жаловаться на падения
  - на других ОС
  - на другом железе
  - и разные клиенты
  - и не только PostgresPro, но PostgreSQL
- А точно ли дефект был в железе?



# segfault, 2025

- Начинаем воспроизводить
  - Конфигурация как была у клиентов (вплоть до версий ядра)
  - Linux kernel 6.1, XFS, 2 TiB RAM



# segfault, 2025

- Начинаем воспроизводить
  - Конфигурация как была у клиентов (вплоть до версий ядра)
  - Linux kernel 6.1, XFS, 2 TiB RAM
- Сначала воспроизвелось на наших серверах (неожиданно)



# segfault, 2025

- Начинаем воспроизводить
  - Конфигурация как была у клиентов (вплоть до версий ядра)
  - Linux kernel 6.1, XFS, 2 TiB RAM
- Сначала воспроизвелось на наших серверах (неожиданно)
- Подкрутили тест



# segfault, 2025

- Начинаем воспроизводить
  - Конфигурация как была у клиентов (вплоть до версий ядра)
  - Linux kernel 6.1, XFS, 2 TiB RAM
- Сначала воспроизвелось на наших серверах (неожиданно)
- Подкрутили тест
- Воспроизвели на виртуалках с большим количеством vCPU и памяти




# segfault, 2025

- Начинаем воспроизводить
  - Конфигурация как была у клиентов (вплоть до версий ядра)
  - Linux kernel 6.1, XFS, 2 TiB RAM
- Сначала воспроизвелось на наших серверах (неожиданно)
- Подкрутили тест
- Воспроизвели на виртуалках с большим количеством vCPU и памяти
- Ещё подкрутили тест



# segfault, 2025

- Начинаем воспроизводить
    - Конфигурация как была у клиентов (вплоть до версий ядра)
    - Linux kernel 6.1, XFS, 2 TiB RAM
  - Сначала воспроизвелось на наших серверах (неожиданно)
  - Подкрутили тест
  - Воспроизвели на виртуалках с большим количеством vCPU и памяти
  - Ещё подкрутили тест
  - Воспроизвели на маленьких виртуалках
- 

# segfault, 2025

- Супер! Теперь можем что-нибудь поменять



# segfault, 2025

- Супер! Теперь можем что-нибудь поменять
- ext4 вместо XFS - **не** воспроизводится



# segfault, 2025

- Супер! Теперь можем что-нибудь поменять
- ext4 вместо XFS - **не** воспроизводится
- Debian вместо российской ОС - **не** воспроизводится



# segfault, 2025

- Супер! Теперь можем что-нибудь поменять
- ext4 вместо XFS - **не** воспроизводится
- Debian вместо российской ОС - **не** воспроизводится
- Магия!



# segfault, 2025

- Супер! Теперь можем что-нибудь поменять
- ext4 вместо XFS - **не** воспроизводится
- Debian вместо российской ОС - **не** воспроизводится
- Магия!
- Замена Linux kernel с вендорского на ванильное - **воспроизводится**

# segfault, 2025

- Давайте менять версии Linux kernel



# segfault, 2025

- Давайте менять версии Linux kernel
- 5.18 – 6.8



# segfault, 2025

- Давайте менять версии Linux kernel
- 5.18 – 6.8
- Fun fact



# segfault, 2025

- Давайте менять версии Linux kernel
- 5.18 – 6.8
- Fun fact
  - Поиск коммита в git

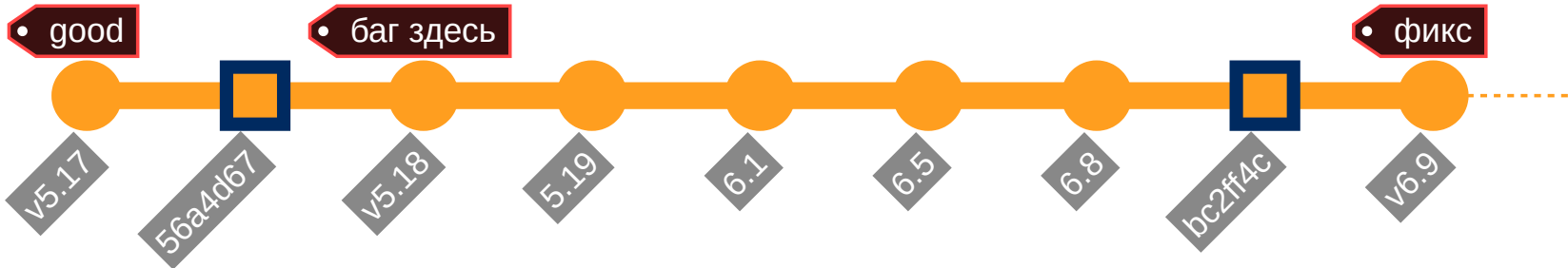


# segfault, 2025

- Давайте менять версии Linux kernel
- 5.18 – 6.8
- Fun fact
  - Поиск коммита в git
  - GitHub vs git-bisect



main



# segfault, 2025

- Перебор коммитов



# segfault, 2025

- Перебор коммитов
- Проблемный коммит в драйвере клавиатуры!



# segfault, 2025

- Перебор коммитов
- Проблемный коммит в драйвере клавиатуры!
- Чё-то не то...



# GITHUB

drivers: net: amd: lance: Remove this driver 

 lunn authored and kuba-moo committed 2 days ago

2fbd04d  

drivers: net: 3com: 3c589: Remove this driver 

 lunn authored and kuba-moo committed 2 days ago

4ff8d06  

drivers: net: 3com: 3c574: Remove this driver 

 lunn authored and kuba-moo committed 2 days ago

a7fbf27  

drivers: net: 3com: 3c515: Remove this driver 


 lunn authored and kuba-moo committed 2 days ago

082b2e0  

drivers: net: 3com: 3c509: Remove this driver 

 lunn authored and kuba-moo committed 2 days ago

91f3a27  

Merge patch series "eventpoll: fix ep\_remove() UAF and follow-up cleanup" 

 brauner committed 2 days ago

Verified

ac8777c  

eventpoll: drop vestigial epi->dying flag 

 brauner committed 2 days ago

Verified

07422c9  

eventpoll: drop dead bool return from ep\_remove\_epi() 

 brauner committed 2 days ago



Verified

3a4551e  

eventpoll: refresh eventpoll\_release() fast-path comment 

 brauner committed 2 days ago

Verified

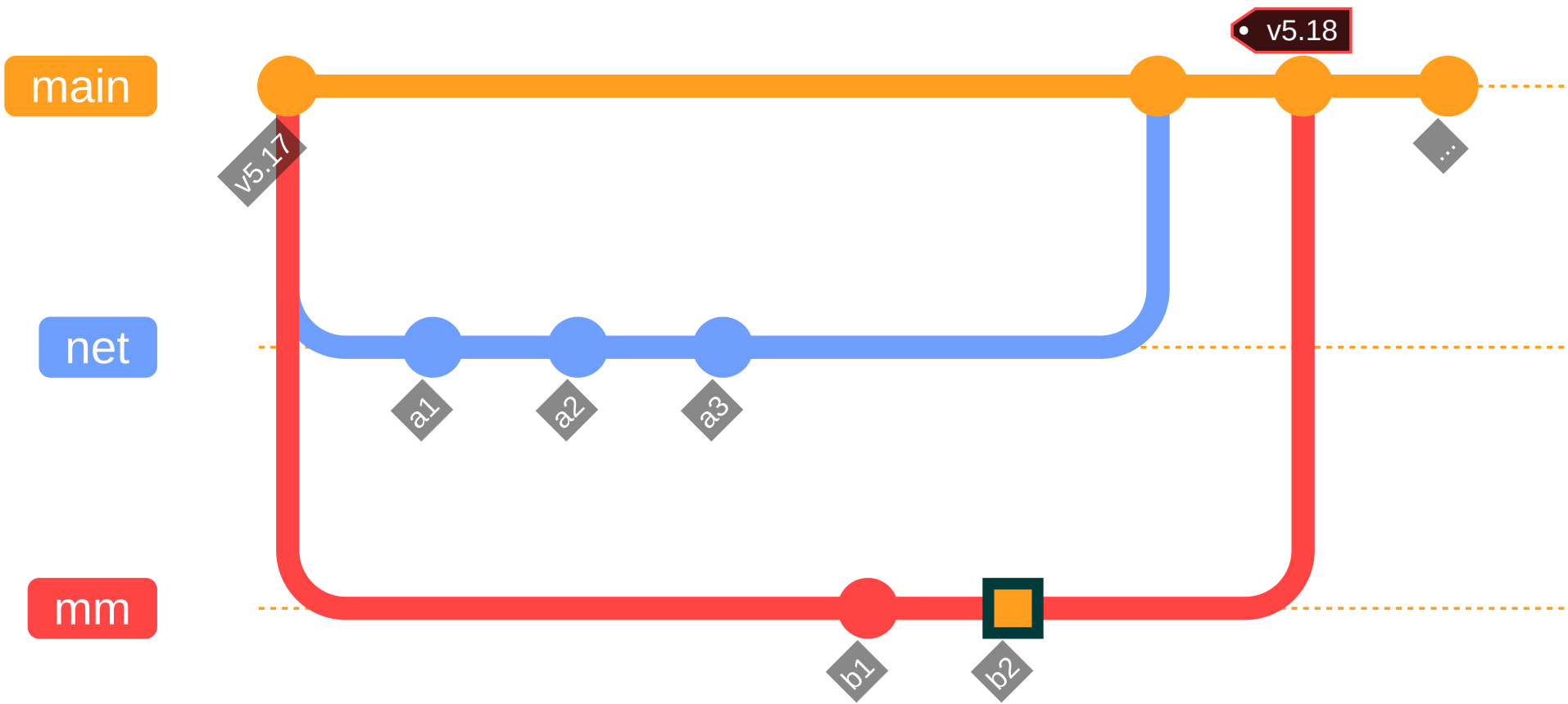
33e92e9  

eventpoll: move f\_lock acquisition into ep\_remove\_file() 

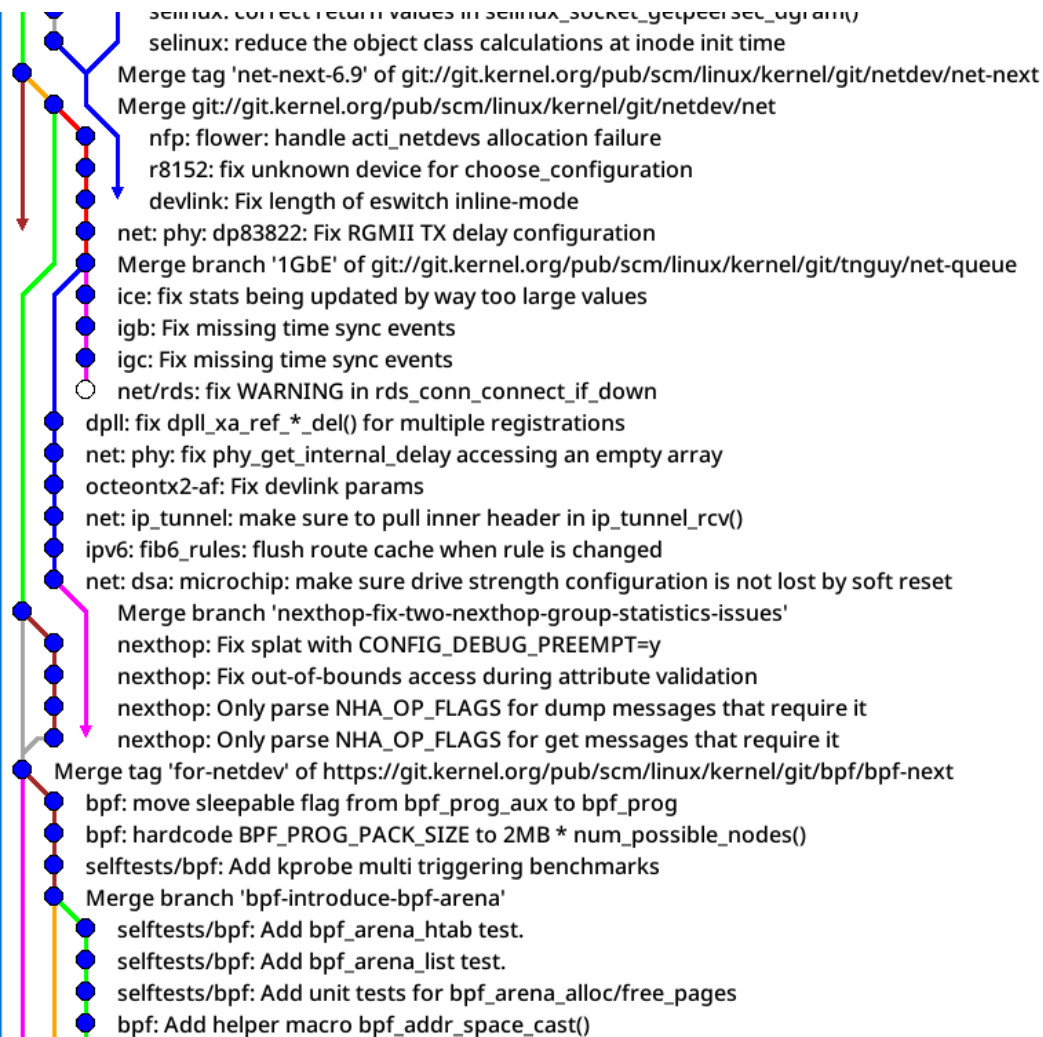
 brauner committed 2 days ago

Verified

d30deeb  







|                                      |                 |
|--------------------------------------|-----------------|
| Paul Moore <paul@paul-moore.com>     | 2024-01-19 00:4 |
| Paul Moore <paul@paul-moore.com>     | 2024-01-19 04:4 |
| Linus Torvalds <torvalds@linux-fou   | 2024-03-13 03:4 |
| Jakub Kicinski <kuba@kernel.org>     | 2024-03-12 06:3 |
| Duoming Zhou <duoming@zju.edu.c      | 2024-03-08 17:2 |
| Hayes Wang <hayeswang@realtek.c      | 2024-03-08 10:5 |
| William Tu <witu@nvidia.com>         | 2024-03-10 19:4 |
| Tim Pambor <tp@osasysteme.de>        | 2024-03-05 14:0 |
| Jakub Kicinski <kuba@kernel.org>     | 2024-03-08 22:3 |
| Przemek Kitszel <przemyslaw.kitsze   | 2024-02-27 17:3 |
| Vinicius Costa Gomes <vinicius.gom   | 2024-02-21 02:5 |
| Vinicius Costa Gomes <vinicius.gom   | 2024-02-21 02:5 |
| Edward Adam Davis <eadavis@qq.c      | 2024-03-05 03:1 |
| Jiri Pirko <jiri@nvidia.com>         | 2024-03-06 18:1 |
| Kévin L'hôpital <kevin.lhopital@savc | 2024-03-07 14:1 |
| Sunil Goutham <sgoutham@marvel       | 2024-03-07 13:5 |
| Eric Dumazet <edumazet@google.c      | 2024-03-07 13:0 |
| Shiming Cheng <shiming.cheng@m       | 2024-03-07 13:0 |
| Oleksij Rempel <o.rempel@pengutr     | 2024-03-04 16:5 |
| Jakub Kicinski <kuba@kernel.org>     | 2024-03-12 06:3 |
| Ido Schimmel <idosch@nvidia.com>     | 2024-03-11 19:2 |
| Ido Schimmel <idosch@nvidia.com>     | 2024-03-11 19:2 |
| Ido Schimmel <idosch@nvidia.com>     | 2024-03-11 19:2 |
| Ido Schimmel <idosch@nvidia.com>     | 2024-03-11 19:2 |
| Jakub Kicinski <kuba@kernel.org>     | 2024-03-12 04:0 |
| Andrii Nakryiko <andrii@kernel.org   | 2024-03-09 03:4 |
| Puranjay Mohan <puranjay12@gma       | 2024-03-11 15:2 |
| Jiri Olsa <jolsa@kernel.org>         | 2024-03-12 00:1 |
| Andrii Nakryiko <andrii@kernel.org   | 2024-03-12 01:3 |
| Alexei Starovoitov <ast@kernel.org   | 2024-03-08 04:0 |
| Alexei Starovoitov <ast@kernel.org   | 2024-03-08 04:0 |
| Alexei Starovoitov <ast@kernel.org   | 2024-03-08 04:0 |
| Alexei Starovoitov <ast@kernel.org   | 2024-03-08 04:0 |

# segfault, 2025

- git bisect: поиск бага по графу

```
git bisect start
git bisect good v5.17 # бага нет
git bisect bad v5.18 # баг есть
git bisect run ./test.sh # → 56a4d67c264e
```

# segfault, 2025

- Автор: Matthew Wilcox, Oracle
- Коммит который сломал (21 марта, 2022): mm/readahead: Switch to page\_cache\_ra\_order
  - Масштаб: 3 файла, +5 / -5 строк
- Коммит который исправил (5 марта, 2024): mm: free folios in a batch in shrink\_folio\_list()
  - Масштаб: 1 файла, +9 / -11 строк
- Знаем коммиты, хотим ли мы знать причину?
  - Не хотим

# segfault, 2025

- Но хотим понять почему на Debian не воспроизводится!



# segfault, 2025

- Но хотим понять почему на Debian не воспроизводится!
- Ядра одинаковые, но есть и параметры загрузки ядра. Grub



# segfault, 2025

- Но хотим понять почему на Debian не воспроизводится!
- Ядра одинаковые, но есть и параметры загрузки ядра. Grub
- `init_on_free`
  - 0 во всех зарубежных
  - 1 в российских



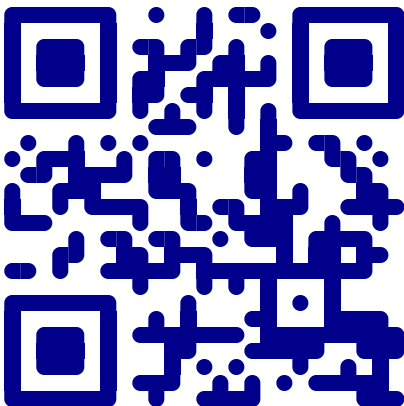
# segfault, 2025

- Проверьте свои серверы
  - Ядро 5.18 – 6.8 + XFS + init\_on\_free = обратитесь к вендору ОС
  - Фикс в 6.9, но не помечен как исправление
  - Бэкпорта нет: на LTS-ядрах фикс сам не придёт

# Заключение

# Заключение

- Декабрь 2025 года
- Технический бюллетень о проблеме
  - Передано российским вендорам
  - ИСП РАН
- Статья на Хабр, где куда больше деталей




# Заключение

- Благодарности
  - Своим коллегам - Алексей Махмутов, Тимофей Мелко
  - Коллегам из других компаний - Александр Бурцев и Ко (product owner МБД Скала^p)
  - И многих других!
  - Славе Смирнову за **ГИГАНСТКУЮ ПОМОЩЬ** в подготовке доклада!

# Заключение

## ■ Guideline

- Нужен стенд
  - Детальный анализ условий падений
    - что за процессы работали в этот момент, какие запросы/действия выполняли
  - Если тест не показывает бага -> не то нагружается или не достаточно
  - Гипер-усиливайте тест, чтобы сократить время воспроизведения
  - Уменьшайте ресурсы постепенно, переводите на виртуалки
  - Меняйте версии софта и параметры
  - git-bisect
- 

# Заключение

- Конец...
  - Ребята из одного банка воспроизвели и на **init\_on\_free=0**
  - Но ушло не 1 минута, а 50 часов!
  - Тикет в Linux bugzilla - я так и не завёл

Спасибо что выслушали!

А данный доклад закончился

**Вопросы?**

