

СУБД для вычислительных комплексов семейства «Эльбрус»

Докладчик: Королев Сергей Дмитриевич

АО «МЦСТ»

30 ноября 2016

- Разработка ключевых компонентов компьютера
 - Процессоры
 - Контроллеры
 - BIOS
 - Ядро ОС
 - Дистрибутив ОПО «Эльбрус»
- Собственные средства разработки
- Доверенная платформа



Эволюция процессоров Эльбрус



Эльбрус
2007г.



Эльбрус-S
2010г.



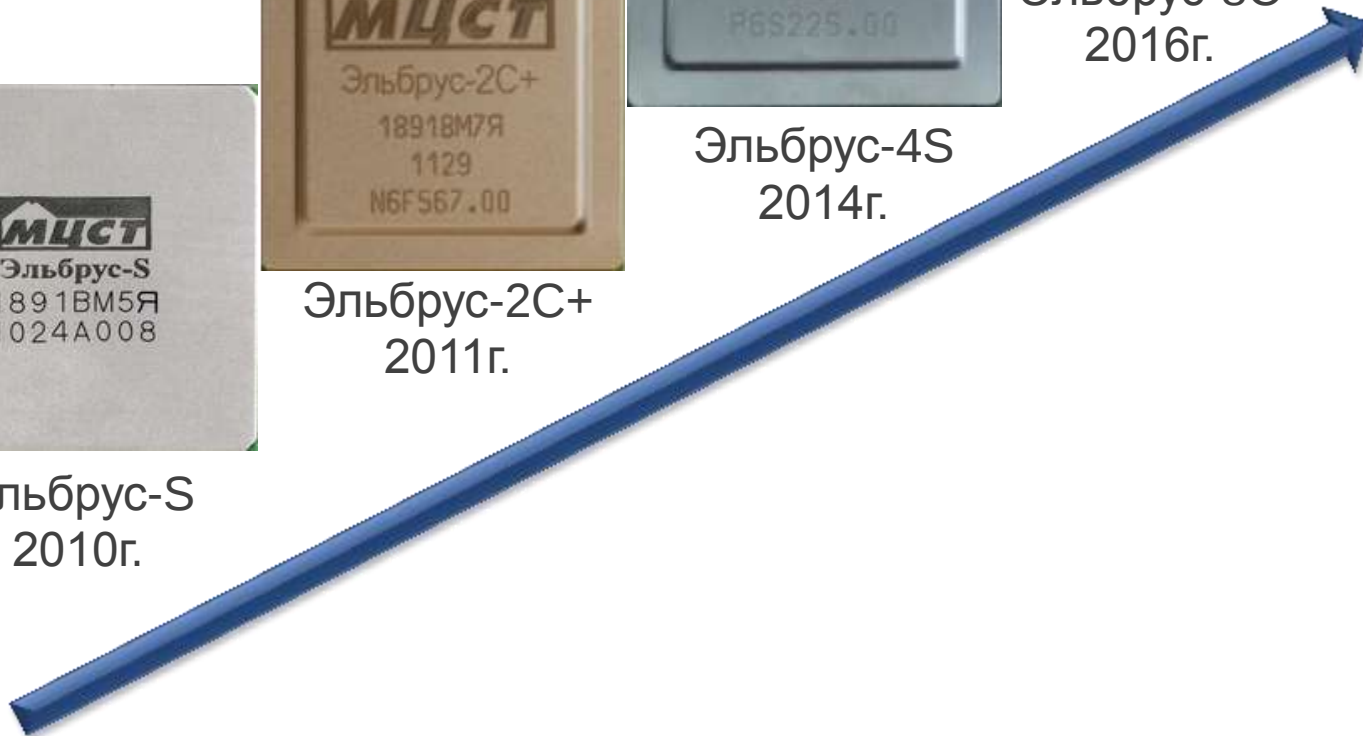
Эльбрус-2С+
2011г.



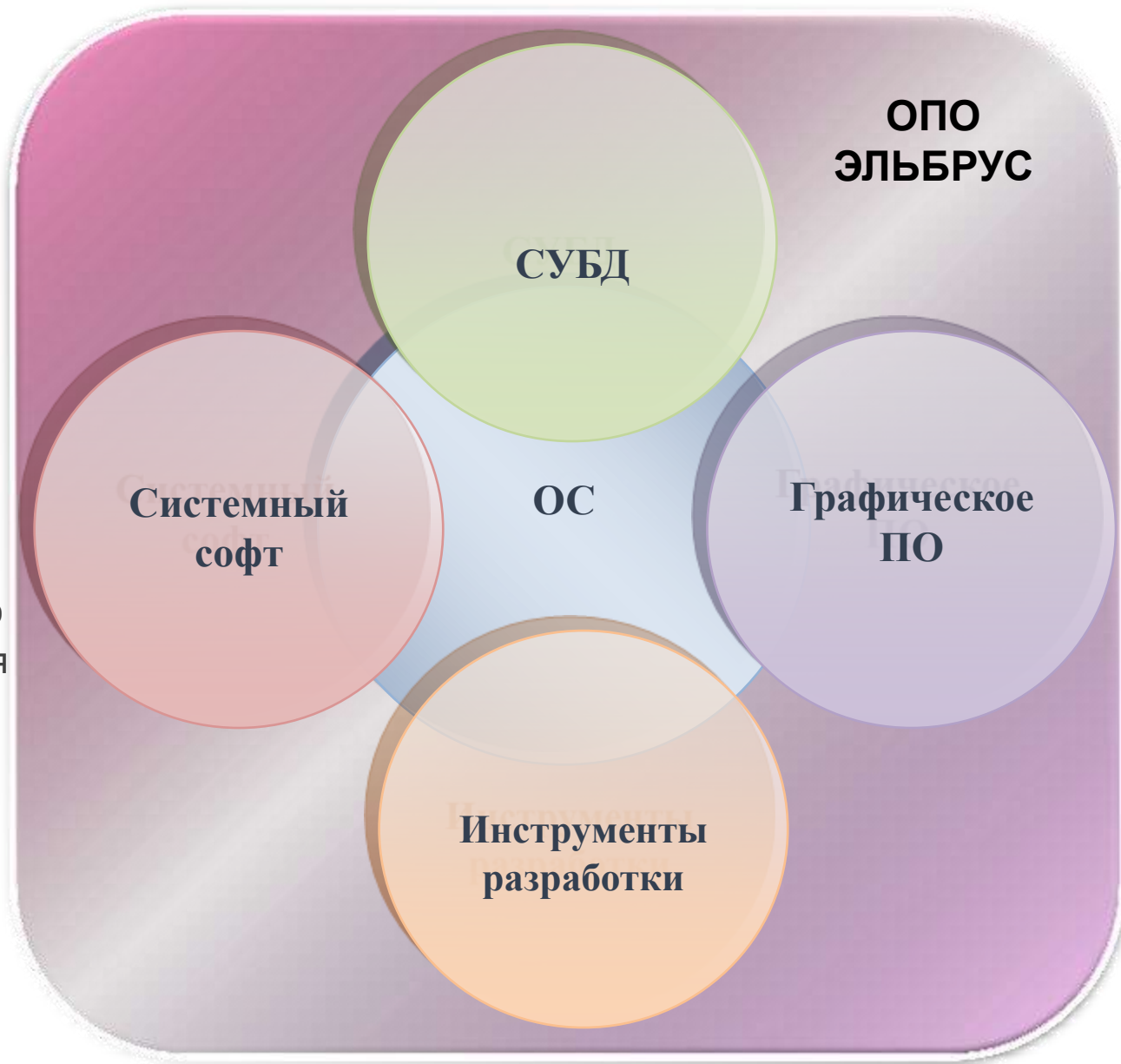
Эльбрус-4С
2014г.



Эльбрус-8С
2016г.



Состав ОПО Эльбрус



- ✓ Базовые системные библиотеки и инструменты
- ✓ СУБД
- ✓ Средства обработки документов
- ✓ Средства работы с сетью
- ✓ Языки программирования и средства разработки
- ✓ Мультимедиа

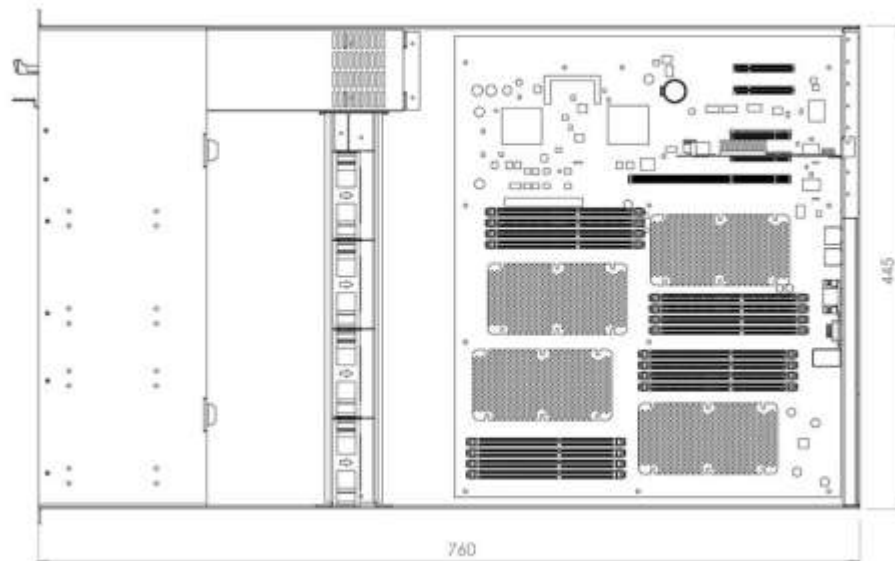
СУБД в ОПО Эльбрус

- ❑ PostgreSQL-9.5.2 (9.2.3, 9.6.1)
- ❑ MySQL-5.5.49
- ❑ Postgres-xl-9.5
- ❑ MongoDB-2.4.0
- ❑ SQLite3-3.8.7.1
- ❑ Gpdb-4.3
- ❑ Memcached-1.4.33
- ❑ Redis-3.0.2

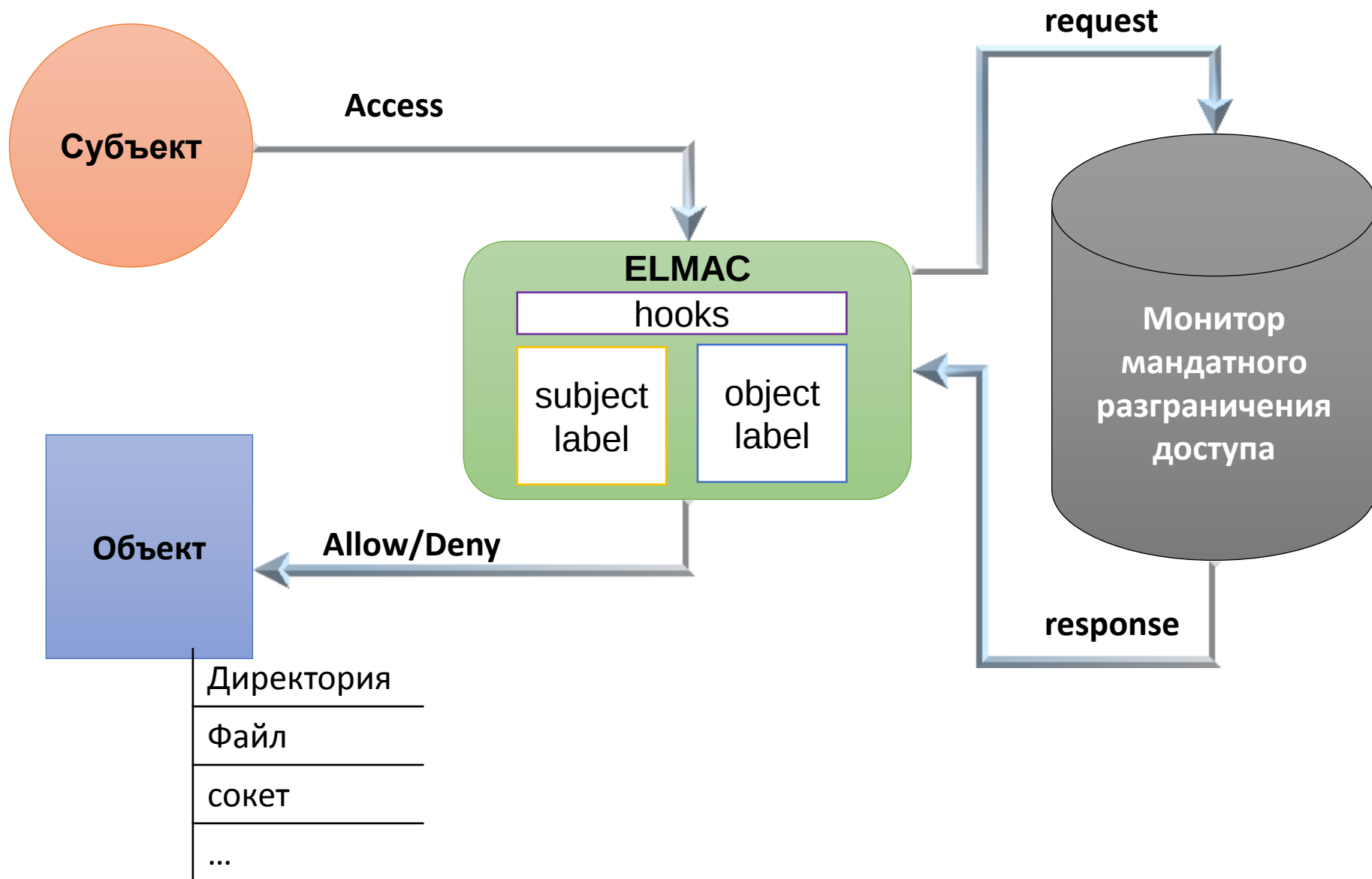


Разработка «Эльбрус-4.4 БД»

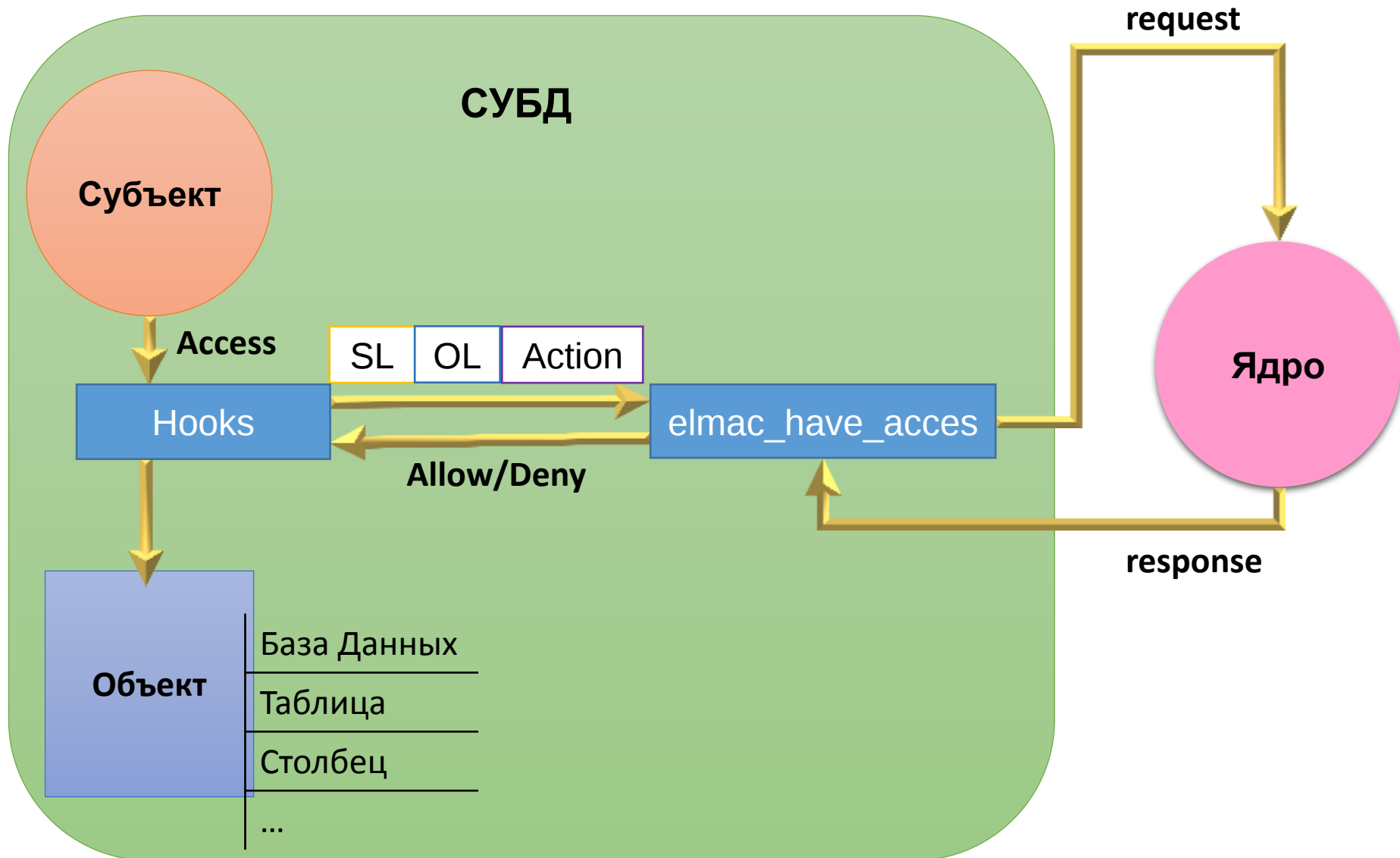
- ❑ Кол-во дисков: 16 дисков 2.5 с режимом горячей замены
- ❑ Контроллер: LSI 12 Gb/s SAS 9300
- ❑ Чипсет: Эльбрус КПИ
- ❑ Процессор: 4 процессора Эльбрус 4С, 4ядра 800MHz
- ❑ ОП: до 384GB DDR3 ECC
- ❑ Питание: 800W
- ❑ Интерфейсы: 2 x PCI-E 1.0
- ❑ x8, 2 x PCI
- ❑ Сеть: 2 x 1 Gb/s, FC(опционально)



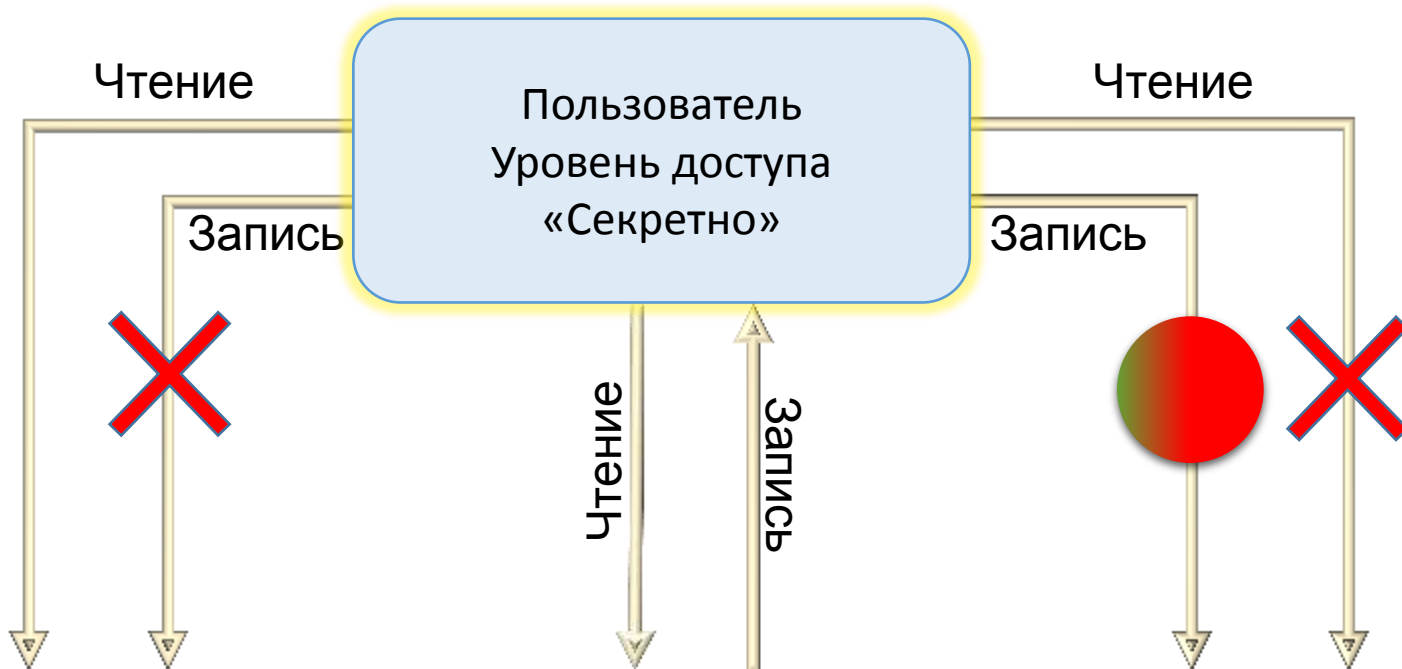
Реализация мандатного разграничения доступа в ОПО Эльбрус



Реализация мандатного разграничения доступа в СУБД PostgreSQL-9.2.3

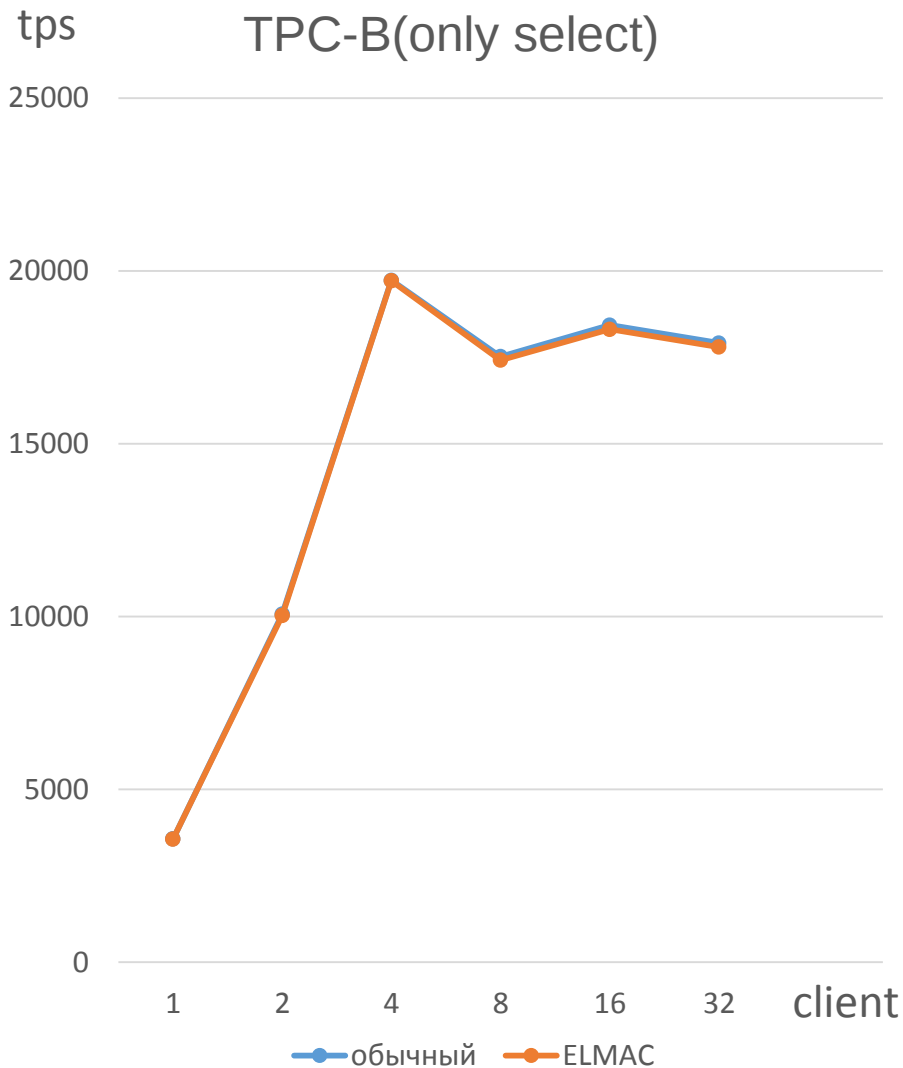
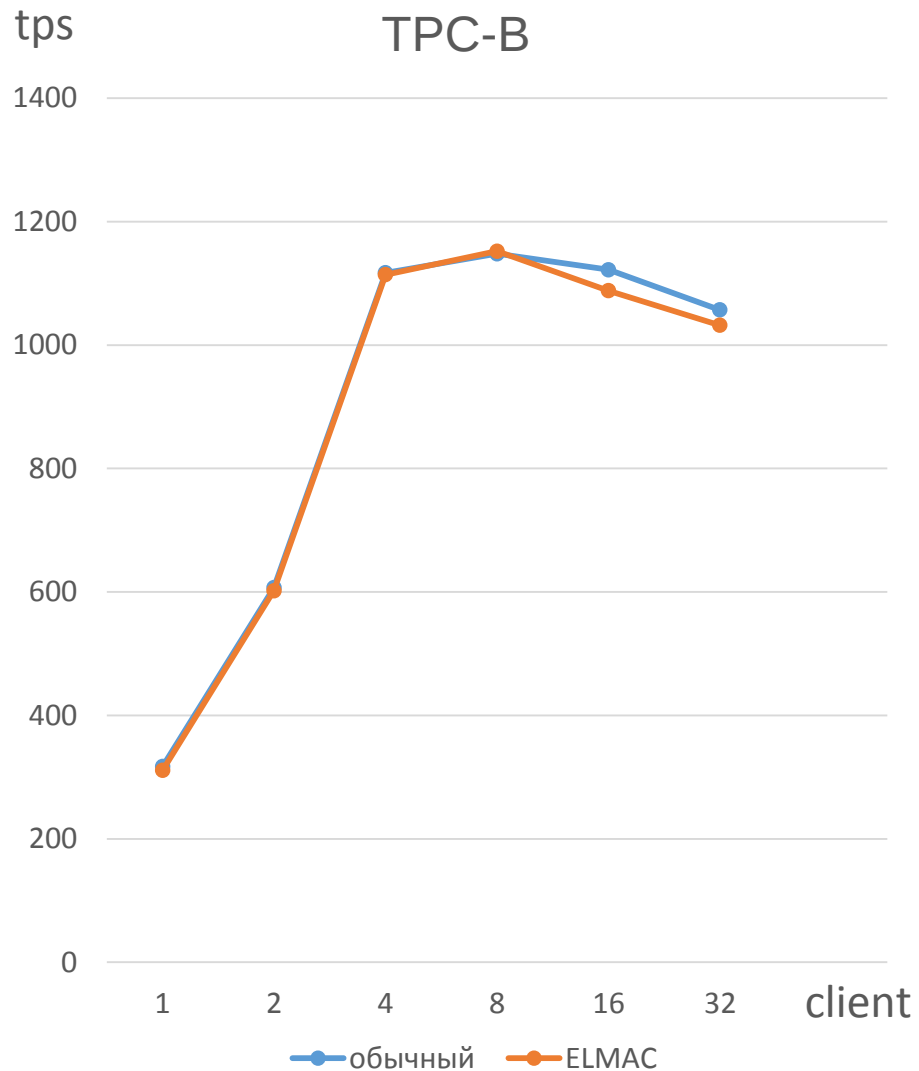


Модель Белла-Лападулы в ОПО Эльбрус



Не секретно	Секретно	Совершенно секретно
...	...	...

Производительность PostgreSQL-9.2.3 с мандатным разграничением доступа



- ❑ Архитектурно-зависимые части
- ❑ Код требующий оптимизации для платформы Эльбрус
- ❑ Использование не библиотечных функций



Архитектурно зависимые части

```
#if defined (__e2k__)
#define HAS_TEST_AND_SET
#define TAS(lock) tas(lock)
typedef int slock_t;
static __inline__ int
tas(volatile slock_t *lock)
{
    return __sync_lock_test_and_set(lock, 1);
}
#define S_UNLOCK(lock) __sync_lock_release(lock)
#endif
```

“занято”
“свободно”

Sync_lock_test_and_set, sync_lock_release

```
register <val_type> prev_val, value, *addr;  
asm volatile  
( "1: <ld>,0  %[addr], %[prev], mas = 0x7"  
  "  {"  
    "    <st>,2  %[addr], %[value], mas=0x2"  
    "    rbranch 1b"  
    "  }"  
: [prev] "=&r" (prev_val), [addr] "+m" (*addr)  
: [value] "r" (value)  
: "memory");  
return prev;
```

```
register <type> *addr, value;  
asm volatile  
( "wait ld_c = 1, st_c = 1"  
  "<st> %[addr], %[value]"  
: [addr] "=m" (*addr)  
: [value] "r" (value)  
: "memory");
```

Overhead в профиле

3,05%	postgres	[.] LWLockAttemptLock
2,53%	postgres	[.] _bt_compare
1,85%	[kernel]	[k] __schedule
1,83%	libc-2.23.so	[.] _int_malloc
1,60%	pgbench	[.] dopr
1,57%	libc-2.23.so	[.] malloc
1,55%	postgres	[.] hash_search_with_hash_value
1,28%	postgres	[.] PinBuffer
1,26%	pgbench	[.] doCustom
1,24%	postgres	[.] dopr

Tps(no dopr) = 34207

Tps(dopr) = 35502

Улучшение на 4%

2,38%	postgres	[.] LWLockAttemptLock
2,32%	postgres	[.] _bt_compare
1,66%	libc-2.23.so	[.] _int_malloc
1,60%	libc-2.23.so	[.] malloc
1,34%	postgres	[.] LWLockRelease
1,29%	[kernel]	[k] __schedule
1,25%	libc-2.23.so	[.] _int_free
1,14%	postgres	[.] hash_search_with_hash_value
1,01%	libc-2.23.so	[.] vfprintf
1,01%	pgbench	[.] doCustom

Подсказки компилятору

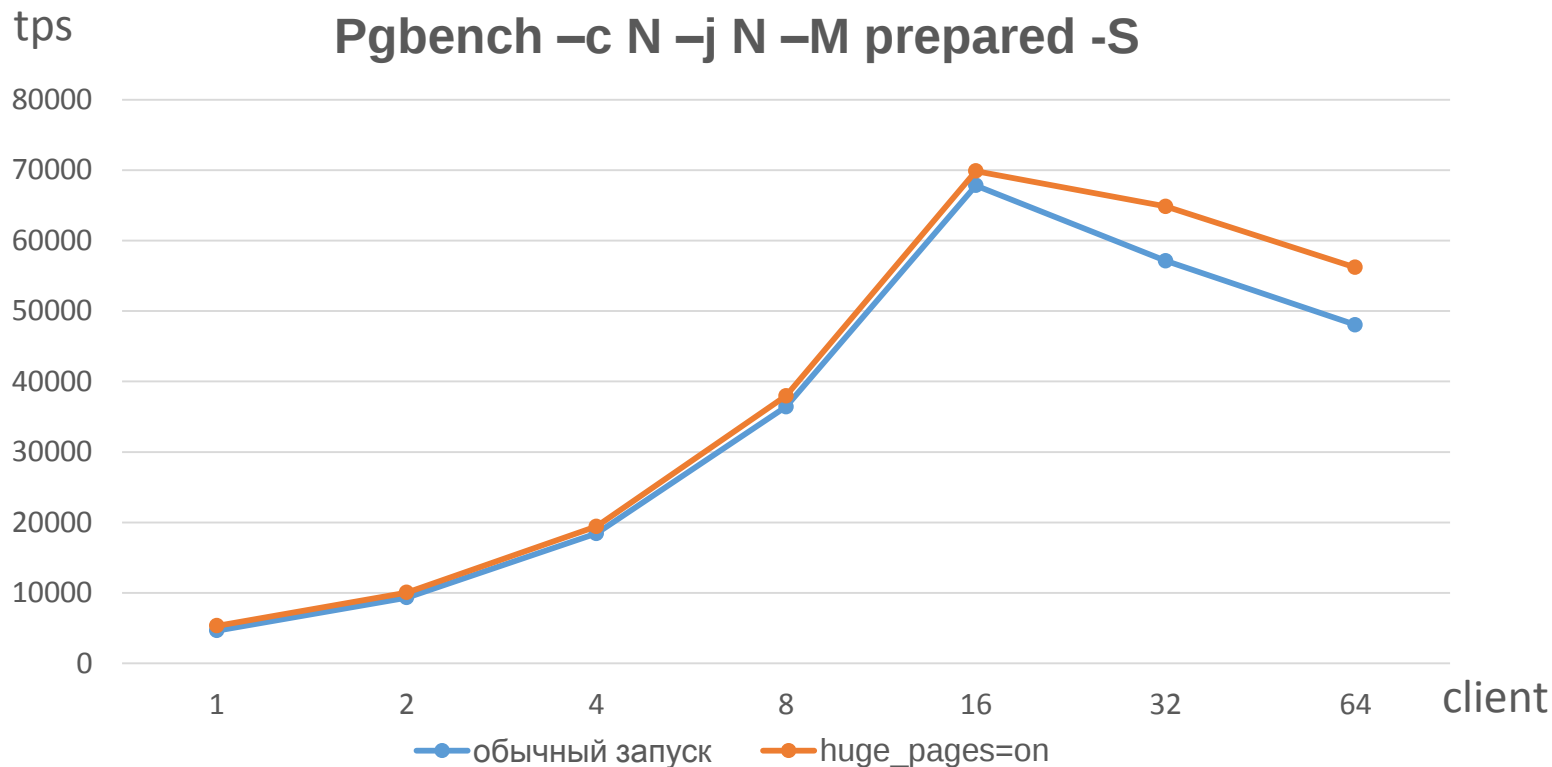
- ❑ Атрибуты
- ❑ Builtin'ы
- ❑ Прагмы
- ❑ Подсказки о маловероятных альтернативах (`__builtin_expect`)



```
static void tuplesort_heap_siftup(Tuplesortstate *state, bool checkIndex){
    SortTuple *memtuples = state->memtuples; SortTuple *tuple; int i,n;
    if (--state->memtupcount <= 0)
        return;
    CHECK_FOR_INTERRUPTS();
    n = state->memtupcount;
    tuple = &memtuples[n];          /* tuple that must be reinserted */
    i = 0;                          /* i is where the "hole" is */
    for (;;) {
        Int j = 2 * i + 1;
        if (j >= n)
            break;
        if (j + 1 < n && HEAPCOMPARE(&memtuples[j], &memtuples[j + 1]) > 0)
            j++;
        if (HEAPCOMPARE(tuple, &memtuples[j]) <= 0)
            break;
        memtuples[i] = memtuples[j];
        i = j;
    }
    memtuples[i] = *tuple;
}
```



Тестирование PostgreSQL-9.5.2



MySQL OLTP Test

ELBRUS

queries performed:

read: 1400322
 write: 500115
 other: 200046
 total: 2100483

transactions: 100023 (597.17 per sec.)

deadlocks: 0 (0.00 per sec.)
 read/write requests: 1900437(11346.19 per sec.)
 other operations: 200046 (1194.34 per sec.)

General statistics:

total time: 167.4957s
 total number of events: 100023
 total time taken by event execution: 2671.7325
 response time:
 min: 12.88ms
avg: 26.71ms
 max: 838.78ms
 approx. 95 percentile: 45.55ms

Threads fairness:

events (avg/stddev): 6251.4375/83.61
 execution time (avg/stddev): 166.9833/0.02

INTEL

queries performed:

read: 1408512
 write: 502984
 other: 201193
 total: 2112689

transactions: 100585 (850.74 per sec.)

deadlocks: 0 (0.00 per sec.)
 read/write requests: 1911496(16167.27 per sec.)
 other operations: 201193 (1701.67 per sec.)

General statistics:

total time: 118.2325s
 total number of events: 100585
 total time taken by event execution: 1891.0395
 response time:
 min: 1.73ms
avg: 18.80ms
 max: 521.94ms
 approx. 95 percentile: 18.36ms

Threads fairness:

events (avg/stddev): 6286.5625/4.25
 execution time (avg/stddev): 118.1900/0.01

Существующие проблемы:

- Не реализован JIT-компилятор
- Особенности сборки

ELBRUS	INTEL
Test update(16 threads) 15000 update/sec	Test update(16 threads) 139772 update/sec
Test insert(16 threads) 4800 update/sec	Test insert(16 threads) 90000 update/sec
Test queries(16 threads) 2289 queries/sec	Test queries(16 threads) 43147 queries/sec

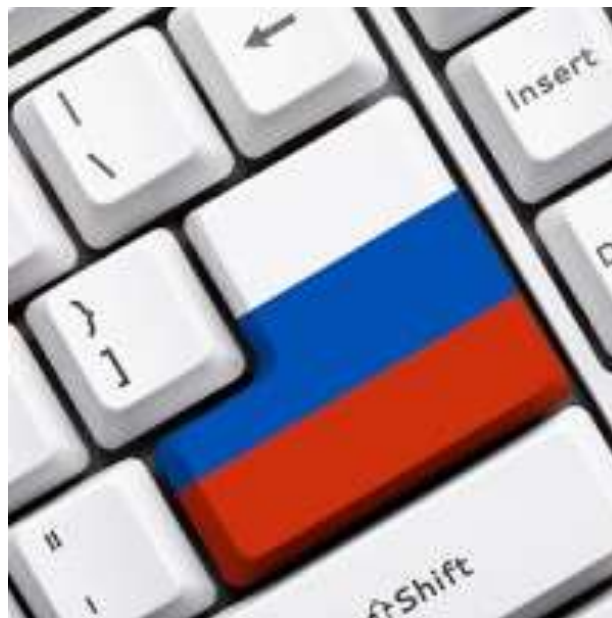
Тестирование СУБД Redis

redis-benchmark -q -n 100000 -c 50 -P 12

Операция	ELBRUS (request per. sec.)	INTEL (request per. sec.)
SET	92081.03	230946.89
GET	119760.48	282485.88
INCR	103842.16	336700.34
LPUSH	88105.73	226244.34
LPOP	99900.09	296735.91
SADD	101010.10	325732.88
SPOP	127877.23	348432.06
MSET (10 keys)	82802.02	144717.80

Кто использует СУБД на Эльбрусе?

- ФГУП НИИ «Восход»
- Пенсионный фонд России
- Министерство обороны



- Портирование нереляционных СУБД
- Перенос популярных реляционных СУБД
- Повышение производительности существующих СУБД
- Реализация JIT-компилятора для MongoDB



Спасибо за внимание!

Вопросы?

Докладчик: Королев Сергей Дмитриевич

30 ноября 2016

Контакты:

Тел: +7-968-949-35-75, e-mail: Sergey.D.Korolev@mcst.ru