

Устранение проблем производительности Firebird при управлении командой разработчиков



Шапошников Александр
Инженер данных ООО «ЦифромедЛаб»



Четыре Ответа на Четыре Вопросы

- **КОМУ?**

Целевая аудитория: ИТ директор, Технический Директор, Ведущий разработчик...

- **О ЧЕМ?**

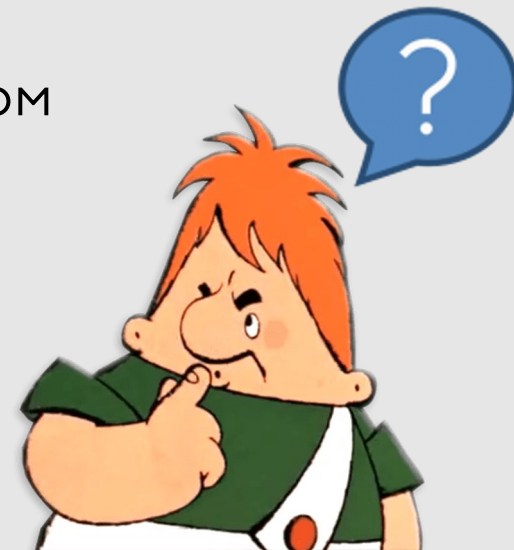
Как Вашей компании (и Вам) не стать заложником качества SQL-кода разработчиков

- **ЗАЧЕМ?**

Рассказать об эффективном в техническом и управленческом плане решении

- **И ЧТО?**

Сэкономьте себе время, деньги, ресурсы и нервы...



Условная градация SQL-кода разработчиков



- Грамотный, продуманный код
Деградация – несчастный случай



- Нормальный код
Деградация – маловероятна



- Некачественный код
Деградация – вполне вероятно

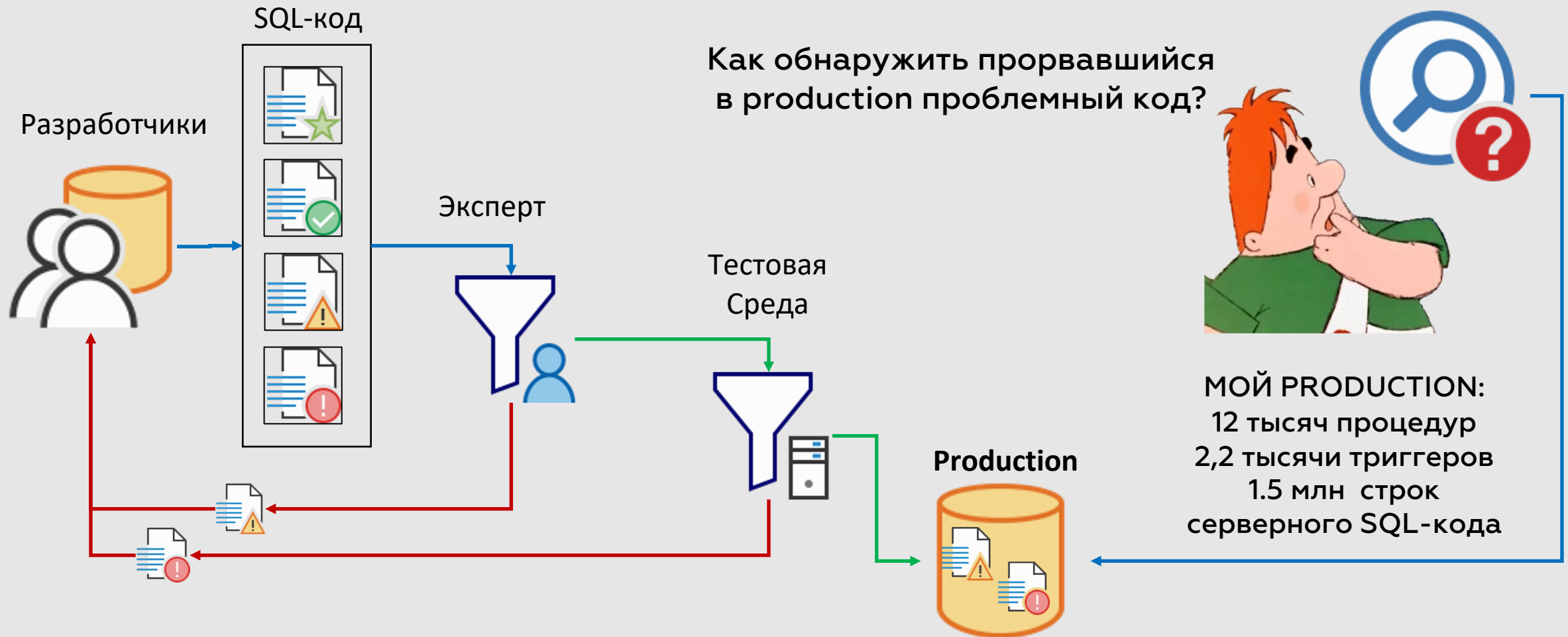


- Изначально проблемный
Деградация Разработчика (?)

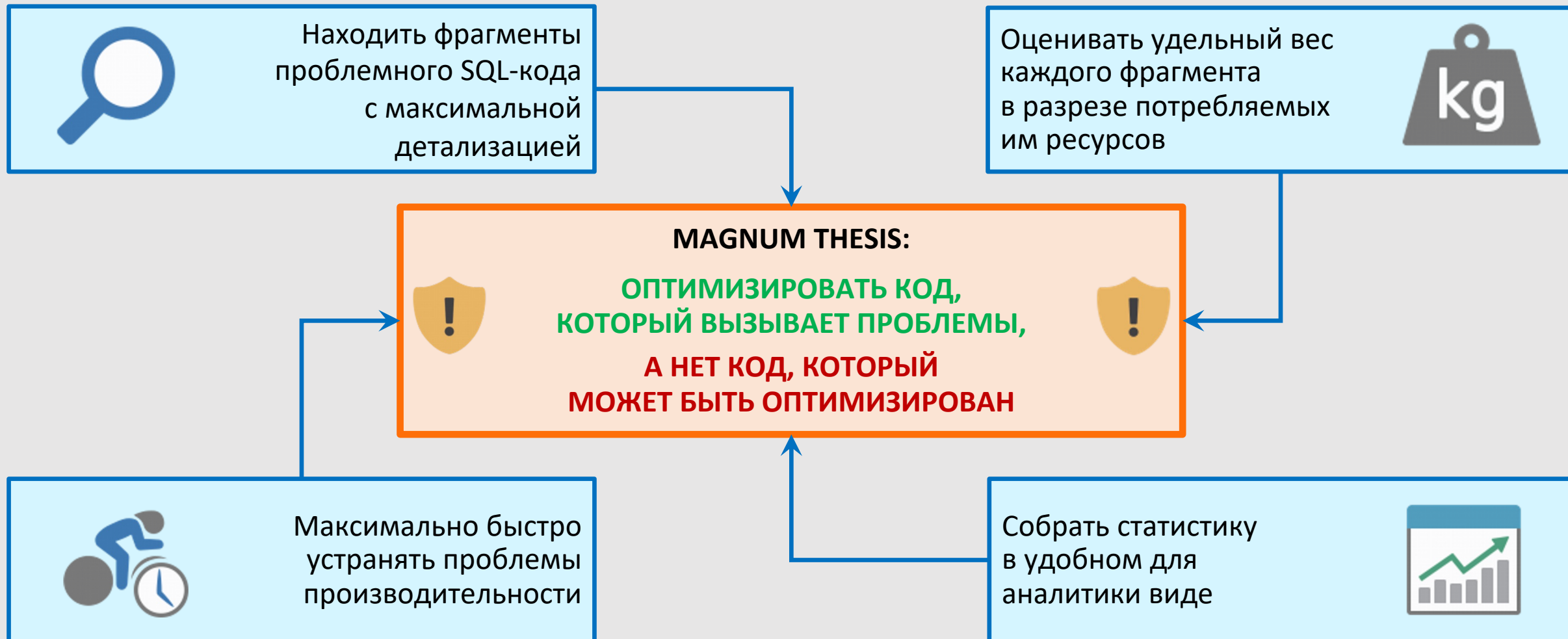


Как быстро и точно
обнаружить SQL-код,
вызывающий проблемы
производительности?

Типовой путь SQL-кода в Production



Чего хотим от решения?



Зачем изобретать велосипед?

Велосипед марки «FbTrace»

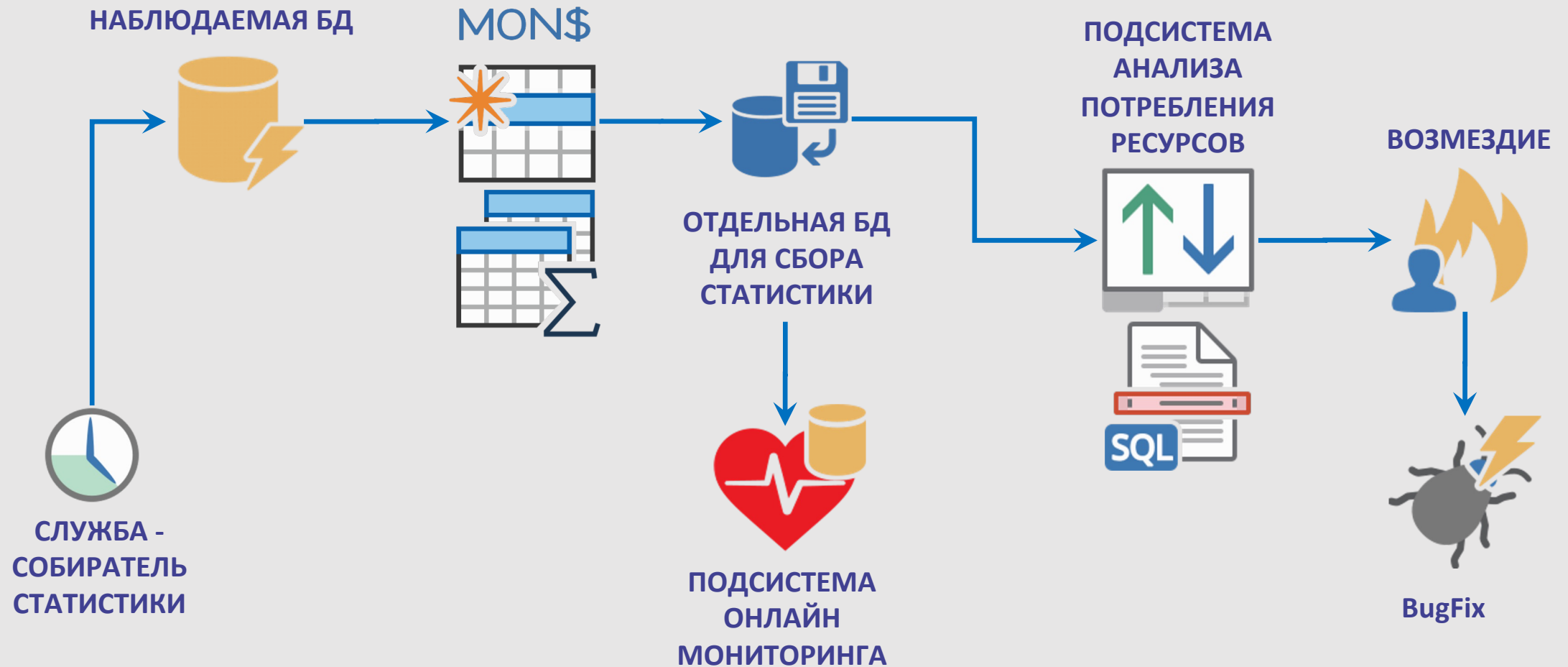
- Как выйти на проблемную строку в триггере?
- Как выйти на вложенную процедуру?
- Большой объем текстовых логов
- неудобство анализа логов

Велосипед марки «Профайлер»

- Появился недавно
- Опыт на примере работы с postgres: 2 точки наблюдений – мало!



ΔMON\$LOGGER. Принципиальный План



Потенциальные Проблемы Методики

- **ПОТЕРИ ПРИ СНЯТИИ СТАТИСТИКИ**

Требуются не абсолютные показатели статистики запросов, а их соотношение между собой

- **ЧАСТОЕ ОБРАЩЕНИЕ К MON\$ТАБЛИЦАМ**

Снятие статистики раз в 5 секунд не вызывает заметных затруднений (Fb2.5 Classic, Fb3 Classic&Super)

- **РАЗМЕРЫ БД СО СТАТИСТИКОЙ**

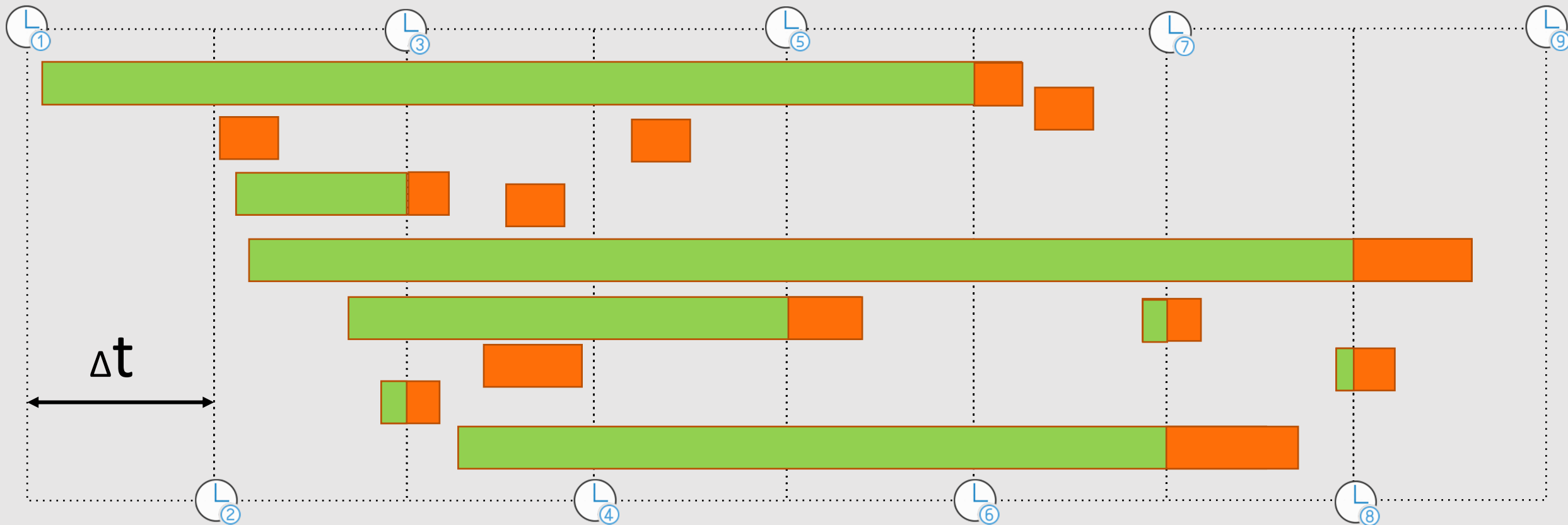
Удалось добиться приемлемых значений

- **МЕТОДИКА ОЦЕНКИ НАКОПЛЕННЫХ ДАННЫХ**

Консультации команды Firebird



Погрешность оценочной функции



 **Зафиксированная Статистика**
 **Потерянная Статистика**

Оффтоп. Метод Монте-Карло

Пример: расчет числа π (3,1415926...) методом Монте-Карло
указанным количеством измерений

	Каждые 60 сек, сутки 1 440	Каждые 15 сек, сутки 5 760	Каждые 5 сек, сутки 17 280	Каждые 5 сек, неделя 120 960
№1	3.08	3.14	3.15	3.14
№2	3.14	3.14	3.14	3.14
№3	3.16	3.19	3.14	3.14
№4	3.22	3.16	3.14	3.13
№5	3.20	3.12	3.13	3.14

АНАЛИЗ по MON\$STATEMENTS & MON\$CALL_STACK



ОСНОВНЫЕ ХАРАКТЕРИСТИКИ



mon\$record_seq_reads

Неиндексированное Чтение (ЧтН)



mon\$record_idx_reads

Индексированное чтение (ЧтИ)



mon\$page_fetches (PF)

Коррелирует с нагрузкой на ЦП



Общее время выполнения



Количество зафиксированных наблюдений

ДОПОЛНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ

Active Statements Count

АТ-Скорость (new AT/min)

ТХ-Скорость (new TX /min)

mon\$record_rpt_reads

mon\$record_inserts

mon\$record_updates

mon\$record_deletes

OIT, OAT

OST, NEXT

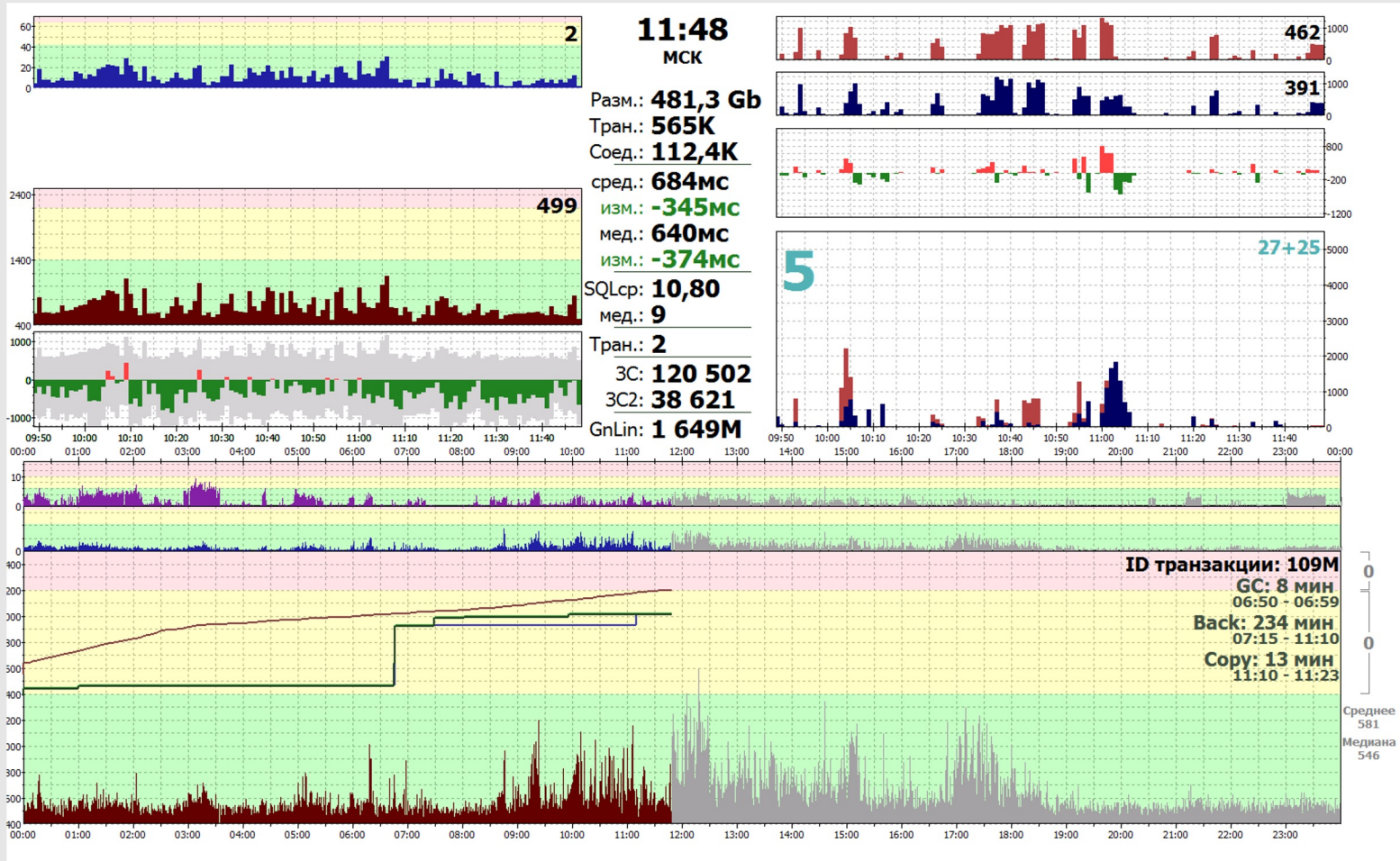
и еще куча

Всего...



фиксируется ВСЯ
статистика

Контрольный Мониторинг



На момент начала проекта
В компании был
реализован контрольный
мониторинг, что
значительно облегчило
начальную оценку
эффективности фиксов

Евгению Диденко –
пламенный привет
и моя глубочайшая
признательность за сотни
(или тысячи?) часов
разработки....

MON\$CALL_STACK. Неиндексированное чтение

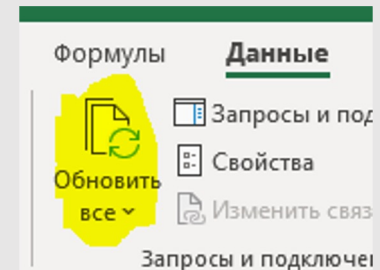


Компания ООО «М БУЛАК».

ПЕРВИЧНЫЙ ОСМОТР

Недельный Анализ в разрезе Mon\$CALL_STACK (120 тыс измерений)

- 4020 Различных зафиксированных различных «Процедура(Номер Строки)»
- Топ5 = 85% всех зафиксированных неиндексированных чтений
- Топ 10= 93%



Процедура (№ Строки)	ЧтН (тыс)	ЧтН%	Сек	Кол
HP_FSROBOT_MSGBOX_CLEANUP (#5)	445 898	37,58%	2 658,69	23
PLACES_BIO (#16)	187 217	15,78%	578,77	260
HP_DOCRECOG_CLEANUP (#19)	178 353	15,03%	713,45	18
AAREP_CHECK_FINPOLIS_TMP (#188)	145 216	12,24%	4 207,99	3080
HP_DOCRECOG_CLEANUP (#29)	51 375	4,33%	214,23	5
HP_CHECK_CLIENT_FIO_FT (#167)	39 690	3,34%	3 460,27	3287

Дело о «Триггере-Натурале»

PLACES_BIO (#16)	187 217	15,78%	578,77	260
------------------	---------	--------	--------	-----

```
1 ALTER TRIGGER PLACES BIO
2 as
3 declare variable c_code varchar(3);
4 declare variable pl_id type;
5 begin
6   new.pl_datetime = current_time;
7   if ((new.pl_id is null) or (new.pl_id <= 0)) then
8     begin
9       c_code = left(new.parent, 3);
10      if (f get db id() = 41701) then
11        begin
12          new.pl_id = :c_code * 1000000000 + gen_id(c
13        end
14      else
15        begin
16          select max(pl_id) + 1
17          from places
18          where pl_id starting with :c_code
19          into new.pl_id;
20        end
21      end
22
```



Показания:

Неиндексированное чтение на 16 строке объясняется условием сравнения начальных цифр числа со строкой

Фикс (индекс+приведение типа):
уменьшение времени выполнения
в 426,7 раза,
уменьшение чтения кэша **18,8 раза**

После Фикса триггер мониторингом ни разу не зафиксирован.

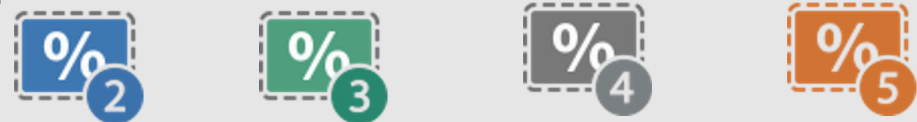
Вадим Жарков

MON\$CALL_STACK. Индексированное чтение.

ПЕРВИЧНЫЙ АНАЛИЗ в разрезе MON\$CALL_STACK

Период: Неделя. (120 тыс измерений)

- 4020 Различных зафиксированных различных «Процедура(Номер Строки)»
- Топ5= 67% всех зафиксированных неиндексированных чтений
- Топ 10=75%



	ЧТИ (тыс)	ЧТИ%	PF%	Сек	Кол
AAREP_SB_REPAYS (#188)	6 047 571	43,86%	45,85%	30 984,17	66
AAREP_CEO_PAR_REPORT_TMP (#221)	1 805 183	13,09%	8,81%	13 649,99	383
AAREP_CEO_PAR_REPORT_TMP (#240)	674 355	4,89%	3,56%	4 895,42	170
FM_LOANS_IN_PROCESS (#158)	418 967	3,04%	3,77%	3 621,17	1 191
FM_LOANS_IN_PROCESS (#57)	369 567	2,68%	3,35%	5 237,28	3 654
HP_BKI_GET_LIST_RU (#289)	303 891	2,20%	0,93%	1 585,06	11

Анализ по MON\$STATEMENTS не менее эффективен, но не так эффектен для демонстрации

Statements SQL Text	CS_CNT	Catched Num	Total Sec	Rec Seq Reads (K)	Rec Idx Reads (K)	Page Fetches (K)	App List	Tx Mode	AT_V ER_LI ST	SQL_SHA512
		3 002	9 144,48	6 292 352,1	0,1	6 653 709,7	GBAK.EXE	SN	P14	
execute block		144	214,52	52 400,3	0,4	53 401,7	FSSVC.EXE	RC_RV	P15	8m+5NnhD09d
Select P.RDB\$PARAMETER_NAME,F.RDB\$FIELD_N		17	12,52	32 807,3	7,8	33 679,8	FINSTART.EXE	RC_RV	P15	9nfMQBndU5PA
insert into combolist (cl_kind, cl_text, cl_category) selec		6	121,15	6 539,8	8,6	7 895,2	FSSVC.EXE	RC_RV	P15	Vpp/eV1P3Wzc
select bkl_object_kind, bkl_object_id, max(bkc_status)		1	83,53	6 445,3	6 477,0	58 617,1	IBEXPERT.EXE	RC_RV	P15	Wee20LpJUJL
select eil.*		1	12,74	5 927,5	0,0	6 095,7	IBEXPERT.EXE	RC_RV	P15	ruk1poK7D4Gxa
insert into combolist (cl_kind, cl_text) select distinct 21, t		6	53,86	5 069,8	0,0	6 050,0	FSSVC.EXE	RC_RV	P15	6PJTDMWDfgv



Анализ по тем же принципам, что и MON\$CALL_STACK

- Потребовалась разработка ПО для анализа
- Считаем SHA512 и по нему идентифицируем (спасибо за Firebird4!)
- Gbak и прочее – попадает в статистику, надо убирать
- Нельзя смешивать аналитику MON\$CALL_STACK и MON\$STATEMENTS!

Контроль Качества Исправления

Названия строк	Кол+	2	Чт	ЧТИ	ЧТИ%	2	1	3	2	1
⊕ AAREP_SB_MAKE_AUTO	3	1		43	0,00%					
⊕ CM_RU_0133_SQL			1							
⊖ AAREP_SB_REPAYS	45	6	2	3 273 098	410 555	172 608	19,10%	2,37%	1,10%	
188	44	1		3 214 800	67 324		18,76%	0,39%		
211	1		1	58 298		65 827	0,34%		0,42%	

До Фикса

После Фикса

```
186 begin
187     sum_zadol = null;
188     select sum(ox.ox_amount_plus - ox.ox_amount_minus - ox.ox_amount_writeoff)
189     from operextra ox
190     where ox.ol_is_active = 1 and
191           ox_object_id = :k_id
192     into sum_zadol;
```



Получили возможность
спустя определенный период
времени понять была ли
устранена проблема

Контроль Качества Исправления (ЧтИ,ЧТИ)

	Кол+	ЧтИ,К			ЧтИ%		
Названия строк	03_09 10_16 17_24	03_09	10_16	17_24	03_09	10_16	17_24
HP_FSROBOT_MSGBOX_CLEANUP(5)	25 27 25	487 629	531 603	495 864	32,40%	44,13%	41,37%
PLACES_BIO(16)	254	196 411			13,05%		
HP_DOCRECOG_CLEANUP(19)	19 20 17	186 933	195 749	170 130	12,42%	16,25%	14,19%
AAREP_CHECK_FINPOLIS_TMP(188)	3654 3967 3947	176 166	188 942	189 298	11,70%	15,69%	15,79%
PACKET_1C_ACC_BIO(6)	2440	114 964			7,64%		

Назв									
	Кол+	ЧтИ,К			ЧтИ%				
Названия строк	03_09 10_16 17_24	03_09	10_16	17_24	03_09	10_16	17_24		
AAREP_SB_REPAYS(188)	42	3 019 773			20,16%				
AAREP_CEO_PAR_REPORT_TMP(221)	450 465 444	1 767 557	2 128 687	1 884 114	11,80%	20,68%	19,12%		
AAREP_OSV_EPS_OSV_TMP(127)	37 5 3	958 211	126 848	74 071	6,40%	1,23%	0,75%		
AAREP_CEO_PAR_REPORT_TMP(240)	176 178 176	934 321	828 401	814 790	6,24%	8,05%	8,27%		
AAREP_OPER_CLASSIFIC(71)	3	708 715			4,73%				

Контроль Качества Исправления (PF, Time)

	Кол+			PF,K			PF%		
Названия строк	03_09	10_16	17_24	03_09	10_16	17_24	03_09	10_16	17_24
AAREP_OSV_EPS_OSV_TMP(127)	37	5	3	3 122 166	413 455	241 509	7,35%	1,51%	0,96%
AAREP_CEO_PAR_REPORT_TMP(221)	450	465	444	3 096 054	3 633 109	3 300 181	7,29%	13,27%	13,14%
AAREP_OSV_EPS_OBOR_TMP(43)	21		1	1 670 809		7 940	3,93%		0,03%
AAREP_CEO_PAR_REPORT_TMP(240)	176	178	176	1 597 446	1 505 955	1 462 612	3,76%	5,50%	5,82%
FM_LOANS_IN_PROCESS(158)	1269	1477	1420	1 538 243	1 873 254	1 796 371	3,62%	6,84%	7,15%

Назв									
	Кол+			Сек			Сек%		
Названия строк	03_09	10_16	17_24	03_09	10_16	17_24	03_09	10_16	17_24
AAREP_SB_REPAYS(188)	42			18 676,23			1,89%		
AAREP_CEO_PAR_REPORT_TMP(221)	450	465	444	13 295,17	16 593,46	15 216,28	1,35%	2,06%	1,56%
AAREP_CALC_PERC_ONE_CLS_TMP(60)	236	272	262	11 414,21	7 861,59	7 827,44	1,16%	0,98%	0,80%
HP_GET_CLIENT_TYPE(63)	44	5	1	9 124,17	1 008,20	0,92	0,92%	0,13%	0,00%
HP_GET_LO_STAGE(31)	1611	1813	1745	9 107,02	12 670,61	10 787,47	0,92%	1,57%	1,11%

Закон Парето - на службу Firebird!

Пример ABC-Анализа по характеристикам:

Время

Неиндексированное Чтение

Индексированное Чтение

Период 2 месяца
СОЧЕТАНИЙ (SP+LINE): 12 018

СЕССИЙ (ИЗМЕРЕНИЙ): 985 862
Записей категории 2A+ = 6

Процедура -Строка		ABC		Вес, %	Ранг
AAREP_CHECK_FINPOLIS(188) 31708	AAA	t	A	0,60	20
		RS	A	12,80	3
		RI	A	1,28	18
HP_CHECK_CLIENT_FIO_FT(167) 31381	ABA	t	A	0,44	31
		RS	A	3,21	5
		RI	B	0,30	52
AAREP_CHECK_CLIENT_FIO(430) 5482	ABA	t	A	0,14	127
		RS	B	1,53	13
		RI	B	0,26	60
HP_BKI_GET_LIST_RU(290) 142	ABA	t	A	0,32	45
		RS	B	1,80	10
		RI	A	0,70	29
HP_CHECK_CLIENT_FIO_FT(181) 4643	ABA	t	A	0,08	231
		RS	B	0,82	15
		RI	B	0,08	128
HP_BKI_GET_LIST_RU(289) 43	ABA	t	A	0,06	303
		RS	B	0,35	23
		RI	A	0,33	49

Дело о неиндексированном чтении в процедуре

Сумма по полю REC_SEQ_1000																										
Названия строк		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Общий итог
⊕ PERFMON\$GET_PING		266	178	227	205	215	223	240	39	147	216	66	107	167	89	162	185	499	877	372	256	390	266	197	334	5 923
⊕ REGL\$OFFERS_AUTOPRICE_FILL_LIB									3 820																	3 820
⊖ SP\$SEL_DOC6987_FROM_DOC6051																106		1 520								1 626
⊖ 47																106		1 520								1 626
Ветошкин																106		1 520								1 626

Сумма по полю REC_SEQ_1000	
Названия строк	Общий итог
⊕ PERFMON\$GET_PING	23,79%
⊕ REGL\$OFFERS_AUTOPRICE_FILL_LIB	15,35%
⊖ SP\$SEL_DOC6987_FROM_DOC6051	6,53%
⊖ 47	6,53%
Ветошкин	6,53%
⊕ SP\$MARKETING_TORG	5,86%
⊕ SP\$PRIEMKA_REPORT_DATA	5,48%

Результат анализа 121 тысячи
снятых за неделю метрик (ТОП-5)

Разработчик должен либо
исправить ситуацию к следующему
совещанию статистику, либо
доказать что данное поведение –
оправдано



Показания обвиняемого (из RedMine)

Запрос, получающий по документу "Довоз коробок" Путевой лист, Водителя и branch_id МДЛП ошибочен

Его некорректность в том, что did документа ищется в строках всех Путевых листов

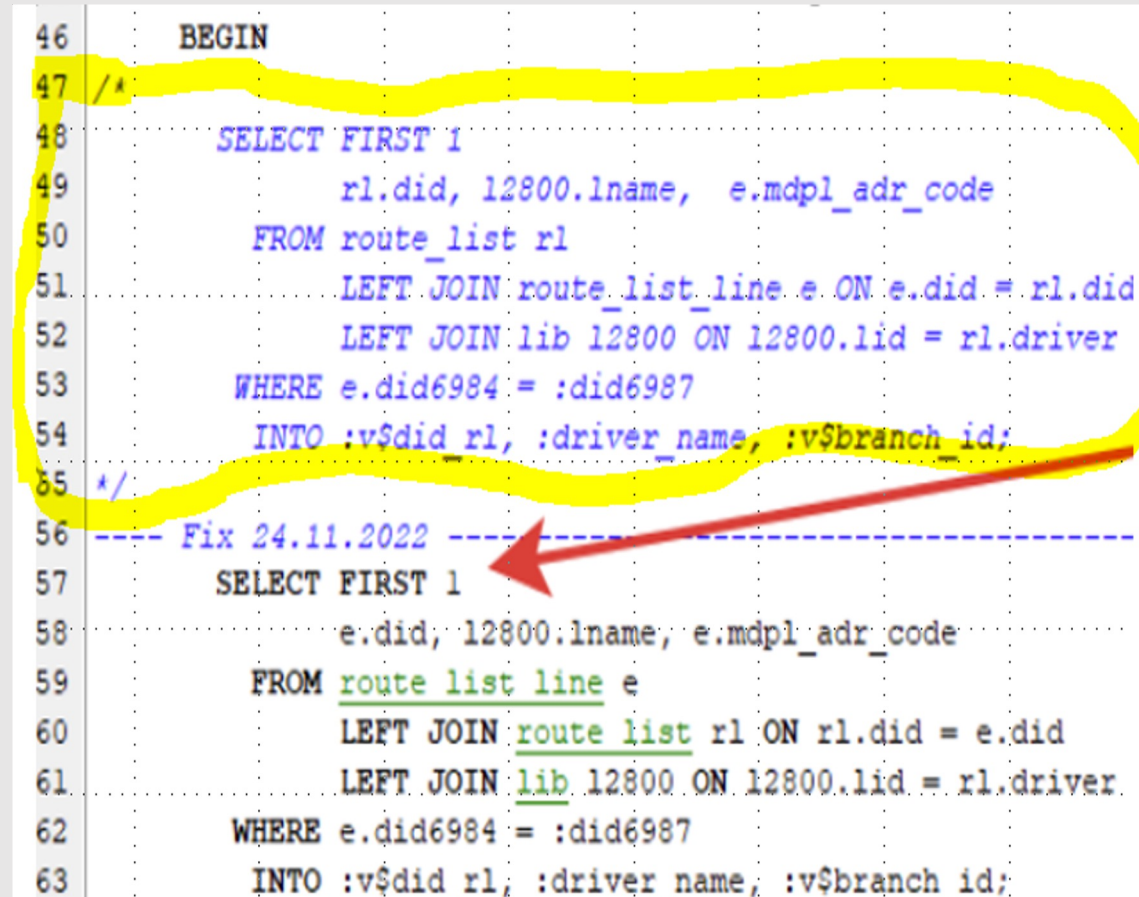
Проблемная процедура и строка мониторингом определена верно.

"Признаю свою вину, меру, тяжесть глубину..."

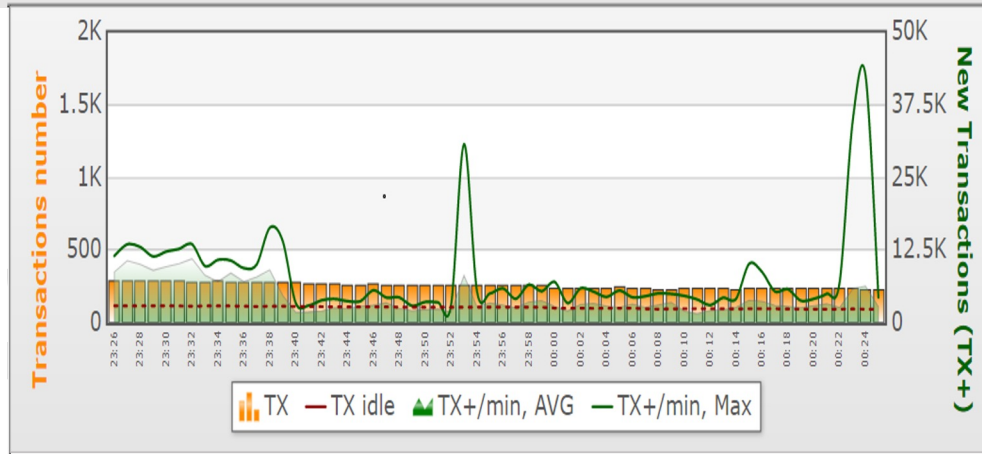
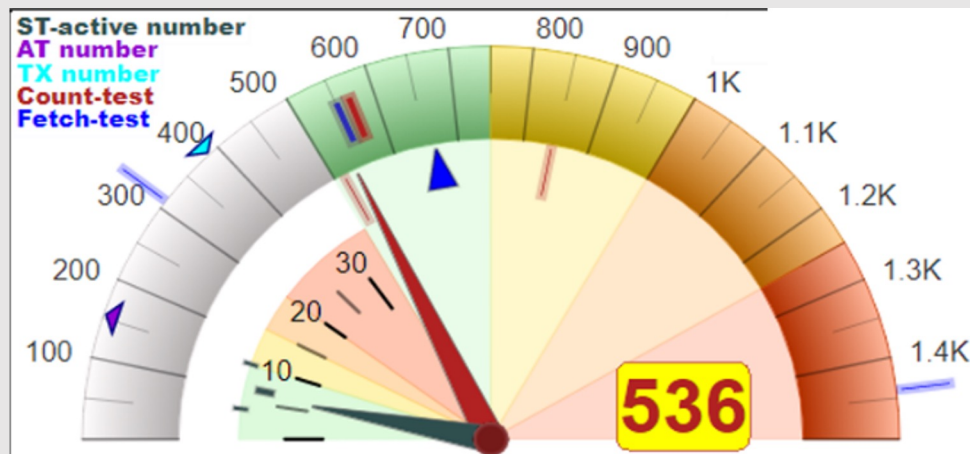
ИСПРАВЛЕНО.

Владимир Ветошкин

```
46 BEGIN
47 /*
48 SELECT FIRST 1
49     rl.did, 12800.lname, e.mdpl_adr_code
50 FROM route_list rl
51     LEFT JOIN route_list_line e ON e.did = rl.did
52     LEFT JOIN lib 12800 ON 12800.lid = rl.driver
53 WHERE e.did6984 = :did6987
54 INTO :v$did_rl, :driver_name, :v$branch_id;
55 */
56 --- Fix 24.11.2022 ---
57 SELECT FIRST 1
58     e.did, 12800.lname, e.mdpl_adr_code
59 FROM route_list_line e
60     LEFT JOIN route_list rl ON rl.did = e.did
61     LEFT JOIN lib 12800 ON 12800.lid = rl.driver
62 WHERE e.did6984 = :did6987
63 INTO :v$did_rl, :driver_name, :v$branch_id;
```



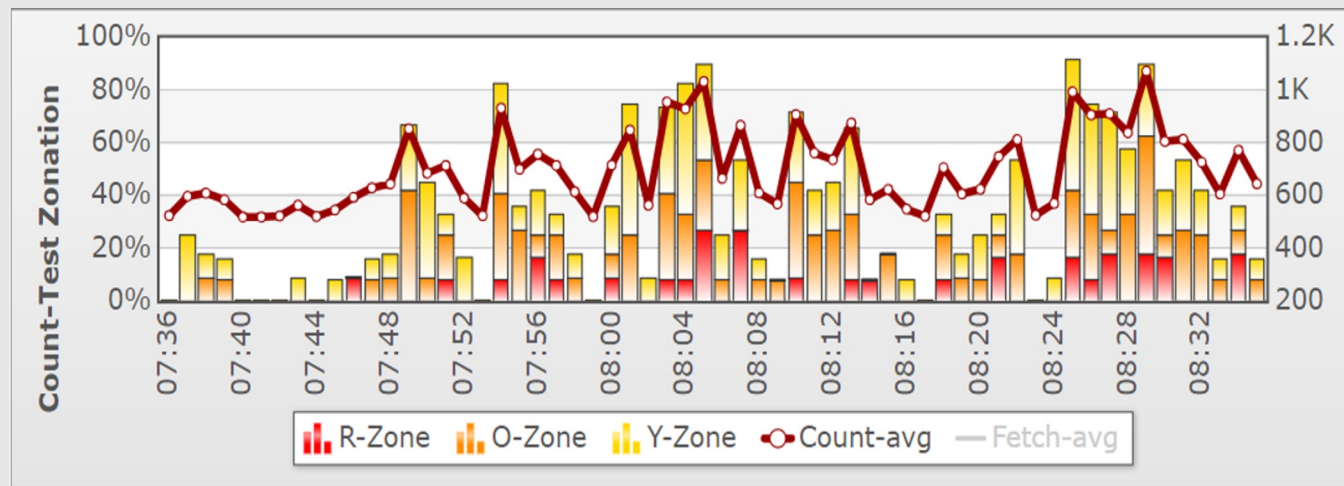
ОНЛАЙН-МОНИТОРИНГ, КАК ПОБОЧНЫЙ ЭФФЕКТ



Шире надо
смотреть

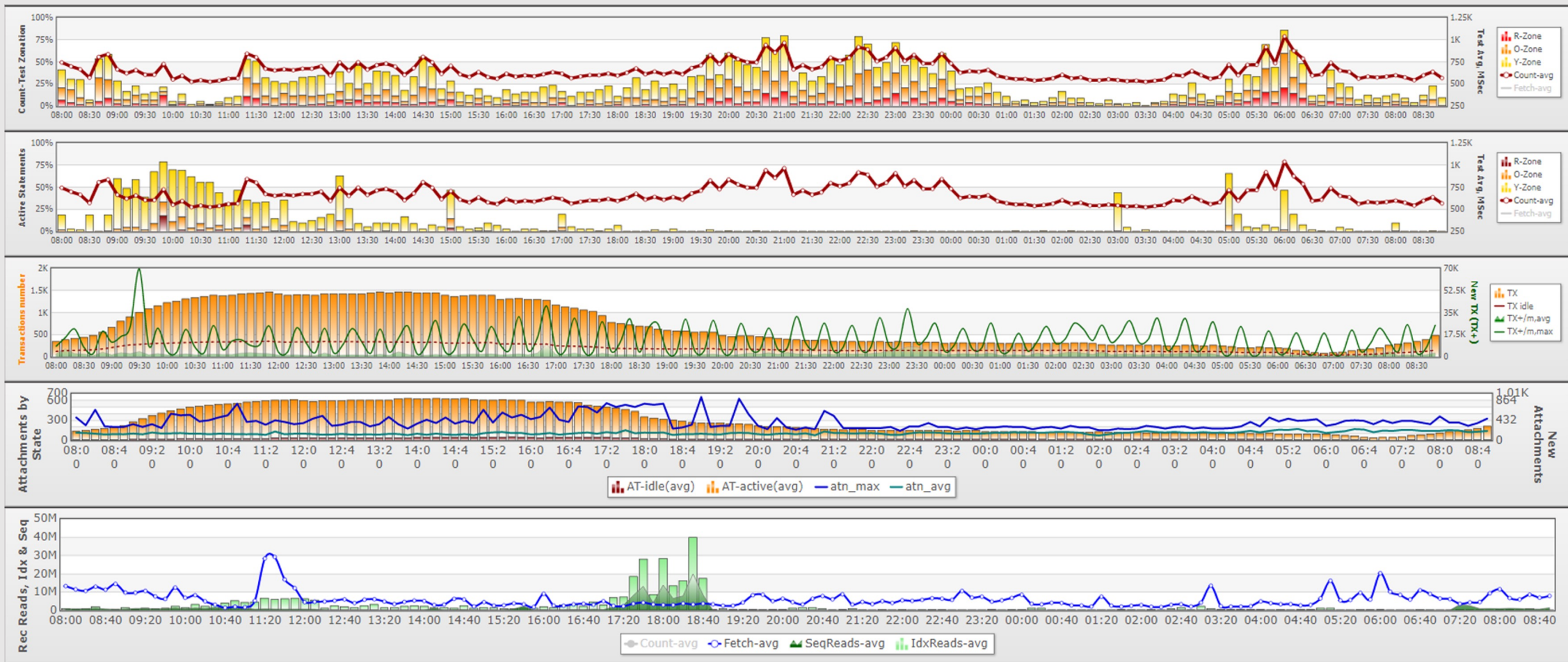


на работу с БД!

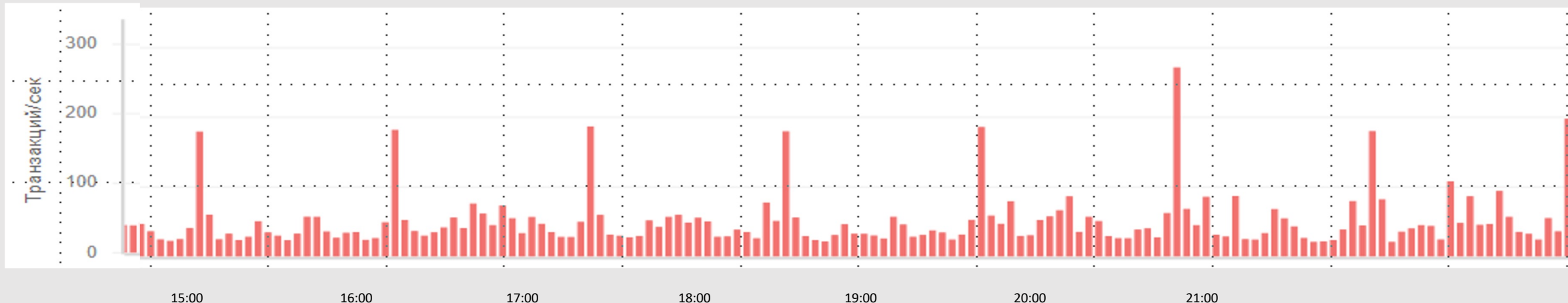


Онлайн-контроль основных
характеристик
скорости возникновения новых
подключений и транзакций,
Выполнение контрольных
COUNT и FETCH запросов

ОНЛАЙН-МОНИТОРИНГ. ПРИМЕРЫ



Дело о Генерации Транзакций



Из корпоративного ИТ-Чата

Какой-то ежечасный регламент, который отрабатывает 05 минут каждого часа (15:05, 16:05, 17:05 и плодит новые транзакции со всей пролетарской ненавистью...

Если завтра, в 11.00 не будет признательных показаний автора, то к 12.00 на основании снятого трейса буду знать виновника и раскрою ему глаза на качество его кода...

Снятие трейса не понадобилось, в 11.00 последовали признательные показания, к 13:00 был готов fix

ИНТЕРЕСНЫЕ ФАКТЫ

После появления PACKAGES в Firebird3 список полей индекса в таблице RDB\$PROCEDURE_PARAMETERS был расширен. (RDB\$PACKAGE_NAME, RDB\$PROCEDURE_NAME, RDB\$PARAMETER_NAME). Некоторые компоненты доступа (например: DevArt UniDac) перестали попадать в индекс при работе с этой таблицей и были замечены в топе по неиндексированному чтению

Разработчики часто не строят индекс по полю, в котором только 2 значения, не понимая важность соотношения этих значений. Какие то TASK-Manager-ы, у которых задача имеет два значения – «Новая» и «Выполнена» (1:10000). Выбираем всегда Задача=«Новая», но индекс не строим, ведь селективность = 0.5.

Мониторинг показывает ~~хреневую~~ неважную производительность самописных функций firebird при работе со строками. Например: у разных клиентов какие-то разные (и везде тривиальные) версии функции DateToStr, и у всех мониторинг ее "замечает" в CALL_STACK, если она вызывается из процедуры.

Во время пандемии функционал использовался для контроля удаленной работы сотрудников

mon\$record_rpt_reads блестяще определяет бессмысленное повторяющееся чтение (обнаружение бесконечно зачисляемого заказа)

ИСТИННАЯ МОТИВАЦИЯ



**Да нормальная у меня,
в общем-то, работа...**



**Я самый несчастный
ИТ-директор на Свете!**

У нас получилось – получится и у Вас!



ИЛИ ОБРАЩАЙТЕСЬ ЗА КОНСУЛЬТАЦИЕЙ

alex@shaposh.ru

skype: shaposh1974

https://t.me/shaposhnikov_av





**Спасибо
за внимание!**



**До встречи
в 2025!**