

Freeswitch HA,
sofia recovery,
как не терять
~~вызовы~~

Обо мне :



- Образование - высшее, техническое
- Инженер информационных технологий в телекоммуникациях
- Разработка, внедрение, интеграция VOIP решений
- Опыт работы над проектами на базе Freeswitch более 10 лет

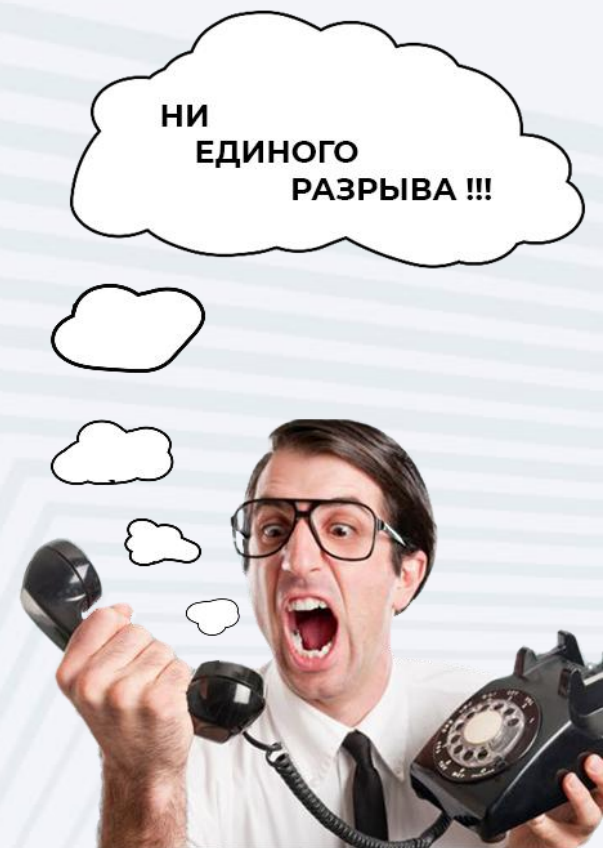
О чем доклад

- Нулевая толерантность к потере вызовов, зачем
- Построение HA кластера Freeswitch
- Практические демонстрации sofia recovery
- Анализ механизмов работы, ограничения, особенности, настройки
- Обсуждение архитектуры и настроек ПО и окружения
 - ❖ Opensips
 - ❖ RTPEngine
 - ❖ SQLite и PG SQL
 - ❖ Keepalived

Нулевая толерантность к потере вызовов

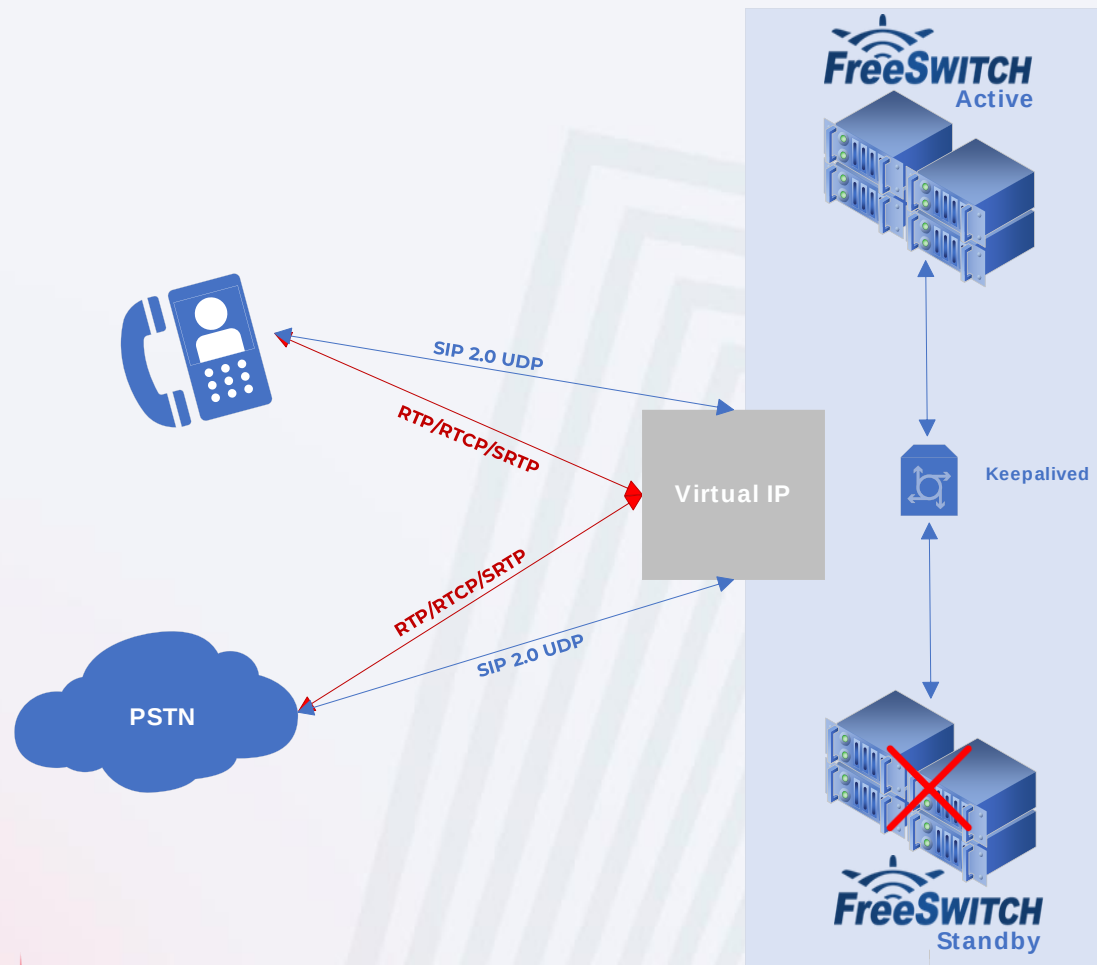
Потеря вызовов на коммуникационном оборудовании имеет ряд негативных последствий:

- Упущенные клиенты и прибыль
- Поломанные рабочие процессы внутри организации
- Снижается уровень качества обслуживания клиентов
- Падает лояльность текущих клиентов
- Снижается выгода от инвестиций



Построение HA кластера FreeSwitch

Базовая схема FreeSwitch HA Cluster.



Построение HA кластера Freeswitch

Недостатки базовой схемы:

- Passive-сервер простаивает, не обрабатывает трафик
- Консистентность конфигураций со временем станет проблемой
- В нужный момент Passive сервер может оказаться неработоспособным
- Транспорт для сигнализации только UDP
- Невозможность горизонтального масштабирования
- Производительности инстанса Freeswitch

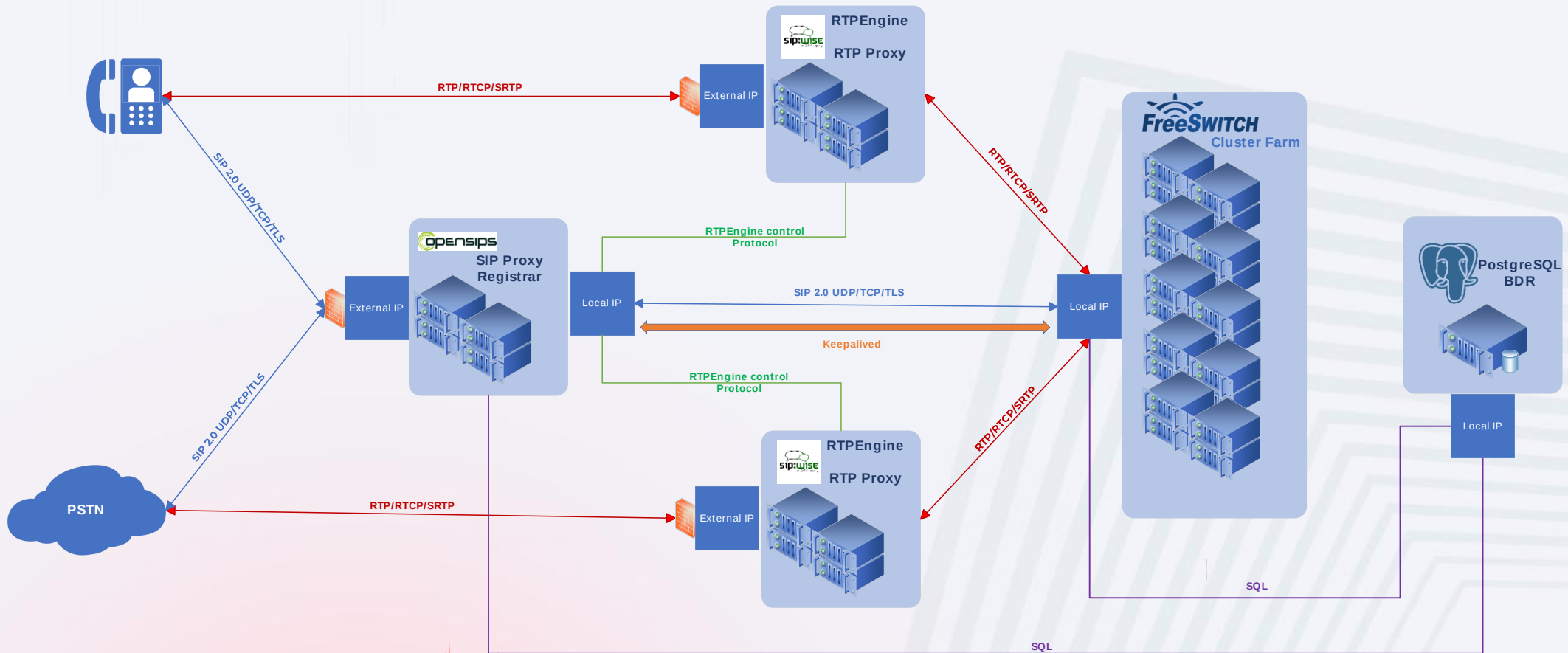
Построение HA кластера Freeswitch

Пути решения проблем:

- Использование фермы серверов Freeswitch
- Использование проксирующих серверов, как для сигнализации (Opensips/Kamailio), так и для media (RTPEngine)
- Использование единого кластера БД
- Использование ClusterFS
- Внедрение DevOps практик CI/CD
- Логика бизнес-приложений выносится за контур сервера Freeswitch

Построение HA кластера Freeswitch

Модифицированная схема кластера:



Recover calls, подготовка конфигураций

Конфигурирование Freeswitch

В параметрах sip профиля включаем

```
<param name="track-calls" value="true"/>  
<param name="odbc-dsn" value="odbc://dsn:username:password"/>
```

Freeswitch в autoload_configs/switch.conf.xml

```
<param name="switchname" value="freeswitch01"/>  
<param name="core-db-dsn" value="odbc://dsn:username:password"/>
```

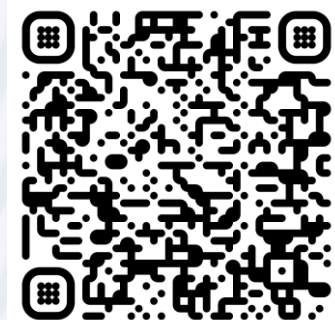
Freeswitch записывает в таблицу recovery в БД информацию о ответных вызовах, так же состояние канальных переменных

Для восстановления вызовов требуется

```
fs_cli -x "fsctl recover"
```

Для краш-теста инстанса Freeswitch выполняется:

```
fs_cli -x "fsctl crash"
```



Recover calls, подготовка конфигураций

Конфигурирование RTPengine

/etc/rtpengine/ rtpengine.conf

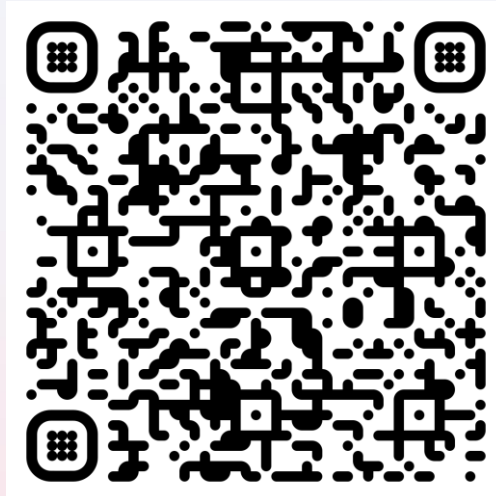
```
1 [rtpengine]
2
3 table = 0
4
5 interface = external/95.174.94.114;internal/10.0.0.13
6
7 listen-ng = 10.0.0.13:4000
8 listen-cli = 127.0.0.1:4001
9 listen-http = 127.0.0.1:9080
10
11 timeout = 60
12 silent-timeout = 3600
13 delete-delay = 5
14
15 port-min = 30000
16 port-max = 40000
17 max-sessions = 5000
18
19 log-level = 0
20 log-stderr = false
21 log-facility = local1
```

Recover calls, подготовка конфигураций

Конфигурирование keeralived

Скрипты для отслеживания состояния инстансов Freeswitch , для примера можно воспользоваться скриптом в документации. Может быть сгенерировано на Bash и прочих, отслеживая состояние и исполняя процедуру в БД и обращаясь через `fs_cli`

Для восстановления сессий.



Recover calls, особенности работы

	A-Z runtime_uuid	A-Z technology	A-Z profile_name	A-Z hostname	A-Z uuid	A-Z metadata
1	b4c08662-3f46-11ed-8659-29519313bb95	sofia	external	freeswitch01	f420ae4e-3f47-11ed-865a-29519313bb95	<cdr core-uuid='b4c08662-3f46-11ed-8659-29519313bb95' swit

БД freeswitch содержит в себе таблицу с информацией о отвеченных вызовах, для каждого switch_name каждого профиля.

В поле metadata содержится в XML-формате данные о вызове:

- Канальные переменные
- Флаги, временные метки
- Лог выполненных приложений диал-плана и сам диал-план
- Информацию о медиа-сессии

Recover calls, грязные лайфхаки

В случае замены в соответствующих полях записей таблицы `recovery` в поле `metadata`, и поле `hostname` значений на значения другого хоста (`switchname` и `ip address`) и выполнить нем **`fsctl recover`** – выбранный хост через `Re-INVITE` поднимет звонок.

- Дает возможность перебросить вызовы на другой хост или несколько других хостов в случае сбоя сервера
- Дает возможность проводить технические работы на серверах за счет переноса вызовов моментально на другой хост не дожидаясь окончания вызовов
- Требуется особой внимательности к операциям с плечами одного и того же звонка, со звонками участников одной и той же конференции.

Recover calls, грязные лайфхаки

Конфигурирование БД

Создадим и заполним

Вспомогательную таблицу с хостами и их адресами.

```
1 CREATE TABLE public.recoverhosts (  
2     switchname VARCHAR(32) NOT NULL,  
3     ipaddr VARCHAR(32) NOT NULL,  
4     CONSTRAINT recoveryhosts_t_switchname_idx UNIQUE (switchname),  
5     CONSTRAINT recoveryhosts_t_ipaddr_idx UNIQUE (ipaddr)  
6 );  
7  
8 INSERT INTO public.recoverhosts (switchname, ipaddr) VALUES('fre SWITCH01', '10.0.0.10');  
9 INSERT INTO public.recoverhosts (switchname, ipaddr) VALUES('fre SWITCH02', '10.0.0.11');  
10 INSERT INTO public.recoverhosts (switchname, ipaddr) VALUES('fre SWITCH02', '10.0.0.14');
```

Создадим функцию, которая заменяет в таблице recovery один хост на другой, по входным параметрам.

```
1 CREATE OR REPLACE FUNCTION public.switch_recover(hostsrc character varying, hostdst character varying)  
2 RETURNS void  
3 LANGUAGE plpgsql  
4 AS $function$  
5 #variable_conflict use_variable  
6 DECLARE  
7     ipsrc varchar(32);  
8     ipdst varchar(32);  
9 BEGIN  
10  
11     SELECT public."recoverhosts".ipaddr FROM public."recoverhosts" WHERE public."recoverhosts".switchname = hostsrc INTO ipsrc;  
12     SELECT public."recoverhosts".ipaddr FROM public."recoverhosts" WHERE public."recoverhosts".switchname = hostdst INTO ipdst;  
13  
14     UPDATE public."recovery" SET hostname = hostdst, metadata =  
15         (SELECT REGEXP_REPLACE(  
16             (SELECT REGEXP_REPLACE(  
17                 (SELECT public."recovery".metadata FROM public."recovery"), hostsrc, hostdst, 'g')  
18             ), ipsrc, ipdst, 'g')  
19         )  
20     WHERE hostname = hostsrc;  
21 END;  
22 $function$  
23 ;  
24
```

Итоги

- Детально рассмотрели HA Cluster Freeswitch
- Ознакомились с механизмами работы и нестандартными возможностями.
- Использование HA freeswitch не всегда требует Standby – ноды.
- Можно использовать не только для предотвращения отказа, но и для проведения плановых работ на сервере с уводом вызовов на другой хост.
- Мечтаем когда-нибудь получить возможность перехватить вызовы прямо из консоли.
- Функционал HA совершенно не заслуженно не используется инженерами и администраторами. Требуется настройка окружения, но результат может оправдать усилия.

Спасибо за внимание!

Буду рад ответить на все ваши вопросы
сейчас или свяжитесь со мной в
будущем:



Крюков Андрей

`AnAKryukov.a@yandex.ru`

`tg://NotUrBruddah`