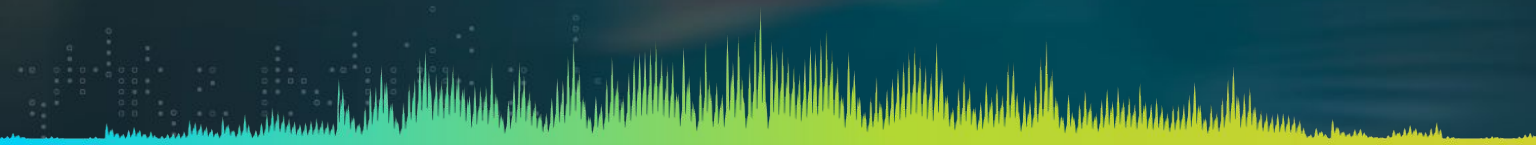


Стреляем в ногу: наш опыт замедления Kamailio



Обо мне:



- Компания IQTEK
- Разработка и внедрение на базе OSS VoIP
- Начиная с Asterisk 0.99
- Asterisk DCAP/DCAA

О чем доклад

- Все любят историю факапов?
- Высокопроизводительные компоненты ничего не гарантируют
- Математика и физика всегда работают
- История художественная на основе реальных событий

Строим распределенную систему

Компоненты:

- Kamailio
- Rtpengine
- Rtpengine-recording
- Redis
- Приложение
 - Получает данные о новых вызовах
 - Обрабатывает PCM

Проблема

- Производительность минимум **до 100cps**
- При тестах у клиентов упираемся **в 30cps**

Внешнее проявление:

- Проявление - **re-transmit**
- LA - **0.5**
- Запаздывающий 100 Trying

172.27.12.57:5060	172.27.12.67:7000	172.27.12.160:5060
15:40:08.023474 +5.005301		100 Trying
15:40:13.028775 +0.000276		180 Ringing
15:40:13.029051 +1.000377	180 Ringing	
15:40:14.029428 +0.000266		180 Ringing
15:40:14.029694 +1.999091	180 Ringing	
15:40:16.028785 +0.000321		180 Ringing
15:40:16.029106 +4.000518	180 Ringing	
15:40:20.029624 +0.000287		180 Ringing
15:40:20.029911 +3.007198	180 Ringing	
15:40:23.037109 +0.008800		200 (172.27.12.160:12222)
15:40:23.045909 +0.001023	200 (172.27.12.206:27146)	
15:40:23.046932 +0.496723	200 OK	
15:40:23.543655 +0.003962		200 (172.27.12.160:12222)
15:40:23.547617 +0.002933	200 (172.27.12.206:27146)	
15:40:23.550550 +0.993247	200 OK	
15:40:24.543797 +0.003136		200 (172.27.12.160:12222)
15:40:24.546933 +0.007221	200 (172.27.12.206:27146)	
15:40:24.554154 +0.509616	200 OK	
15:40:25.063770 +0.000247	BYE	
15:40:25.064017 +0.000602		BYE
15:40:25.064619 +0.000421		200 OK
15:40:25.065040 +1.477847	200 OK	
15:40:26.542887 +0.004496		200 (172.27.12.160:12222)

Rtpengine?

- Отключение RTPENGINE - помогает
- Замечаем что kamailio под нагрузкой начинает терять RTPENGINE
 - `journalctl -u kamailio -f`
- Увидели что заканчиваются RTP порты, все 25тыс. портов выделены
 - `netstat -nlp |grep rtpengine |wc -l`

Нашли!!? Но? нет...

- Порты зависали из-за не настроенных таймаутов при перезапуске kamailio
 - `rtpengine.cfg` —> **final-timeout**

Rtpengine-recording?

- Версия:
 - при выключении *rtpengine* перестаёт обрабатываться трафик
 - каким-то образом *rtpengine-recording* блокирует *rtpengine*, а тот в свою очередь перестаёт отвечать на запросы *kamailio*
- Отключение *tls-forwarding* помогает

Нашли!!? Но? нет...

- запись файлов происходит с высокими CPS
- при записи в тестовый *tls*-сервер проблем не наблюдается

Rtpengine-recording (архитектура)

- inotify
- **rtpengine** пишет метаданные в файл
- **rtpengine-recording** читает
- процессы не блокируют друг друга

Kernel?

- Ясно! Виновато ядро: мы упираемся под большим трафиком в сокеты
- Оптимизируем `sysctl`

Никаких изменений

Приложение?

- Точно!
- Идём к разработчикам **(Первая правильная идея!!)**
- Тормозит сокет
- Добавьте отладку, чтобы видеть время поступления трафика, общий трафик и время закрытия сокета

Результат:

- Всё нормально
- Для многих вызовов не было информации из Redis

Камаилио?

- Хотя не верится, но пробуем оптимизировать kamailio
- Отключаем публикацию в REDIS
- Работает!
- Приложение получает трафик, но без данных о вызове
- Увеличить число воркеров?
- Асинхронные операции?
- Чистим конфиг

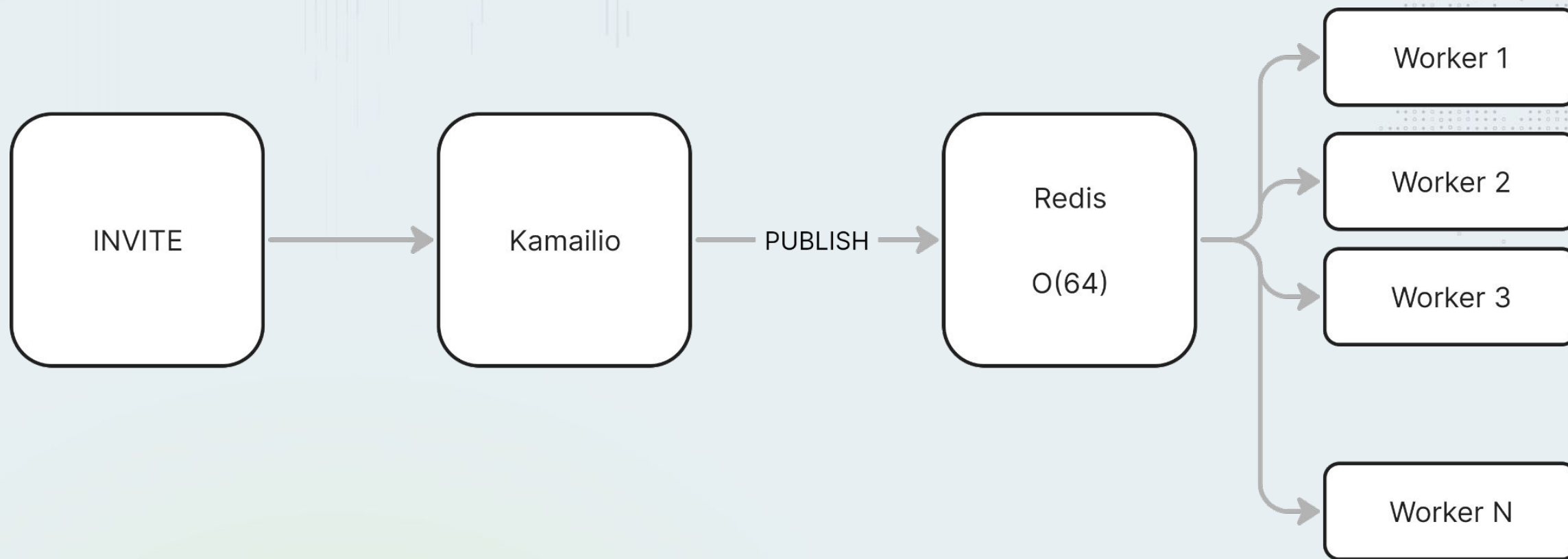
И всё же приложение?

- 8 серверов / 8 процессов
- **!!!** пул подписок (10 соединений в пуле)
- 640! подключений

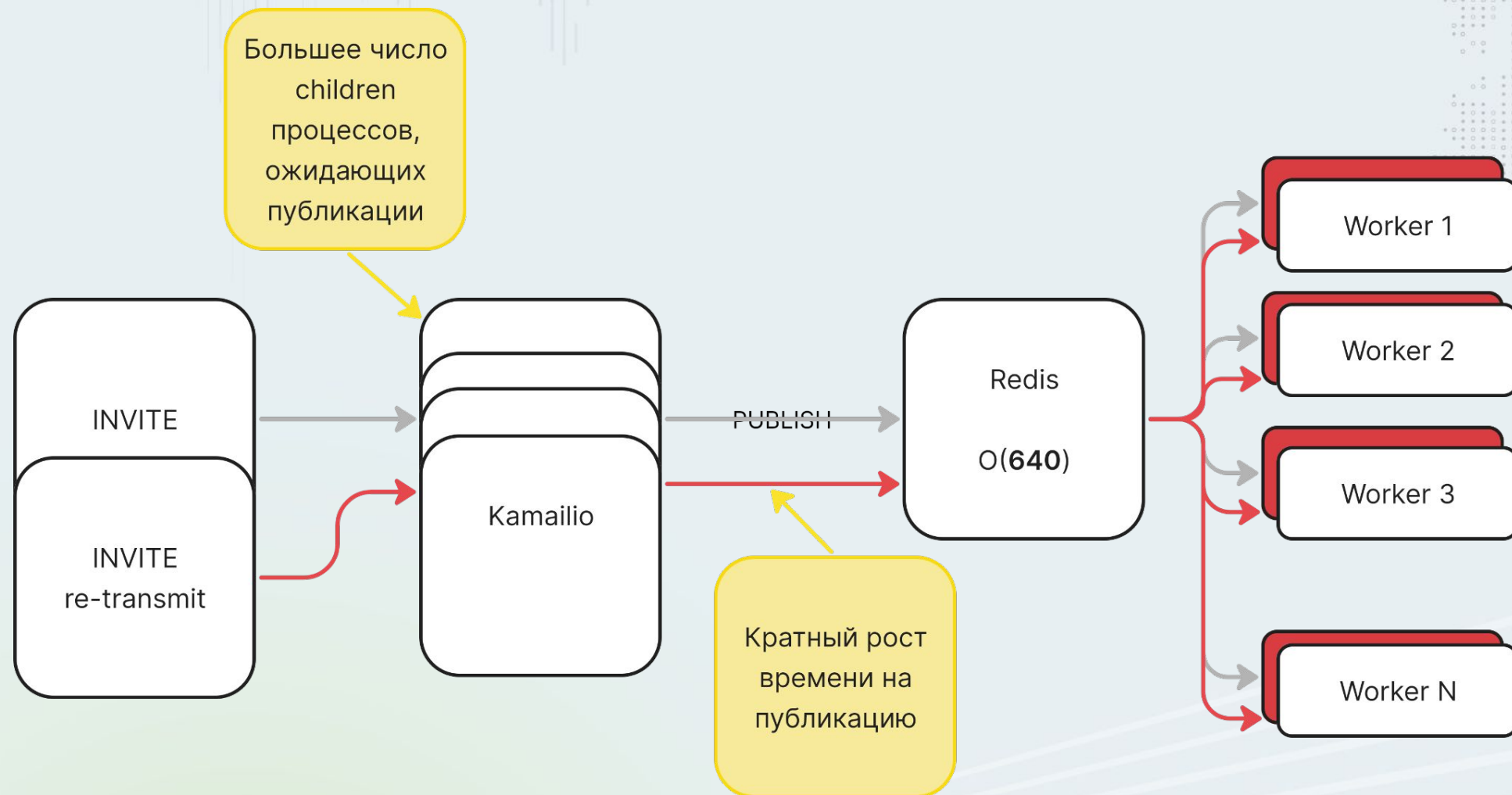
Redis:

- Скорость запроса - $O(M+N)$
- Блокировка на время PUBLISH процесса kamailio
- Повтор INVITE: время публикации достигло таймера

Было



Стало



Решение

- модификация сценария публикации
- Комбинация PUBLISH и RPUSH

```
# Публикация информации о вызове
redis_cmd("srvtm","PUBLISH call:$ci $var(dialog)","r");
|
if ($redis(r=>value) == 0) { # Нет подписчиков на канал
|   redis_cmd("srvtm","RPUSH call-mq:$ci $var(dialog)","r");
| }
if ($redis(r=>value) == 1) { # Устанавливаем время жизни списка
|   redis_cmd("srvtm","EXPIRE call-mq:$ci 3600","r");
```

Итоги

- **re-transmit:** забыли отследить необходимость отдельной обработки
- **Непрерывное тестирование** необходимо, в том числе для метрик производительности
- **программистам и телефонистам** полезно общаться, чтобы обсуждать детали задач и их взаимное влияние
- **Производительность компонента** не отменяет учёта сложности операций.
- **Развитие проекта** предполагает постоянный пересмотр принятых решений

Спасибо за внимание!

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



Гончаровский Игорь



igorg@iqtek.ru



[@IgorrG](https://t.me/IgorrG)