

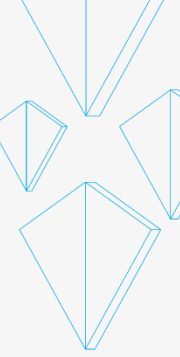
Новый инструмент для визуального анализа системных трасс: возможности и примеры использования

Усов Дмитрий

Разработчик

Отдел инструментальных средств

TraceKey – новый инструмент анализа трасс



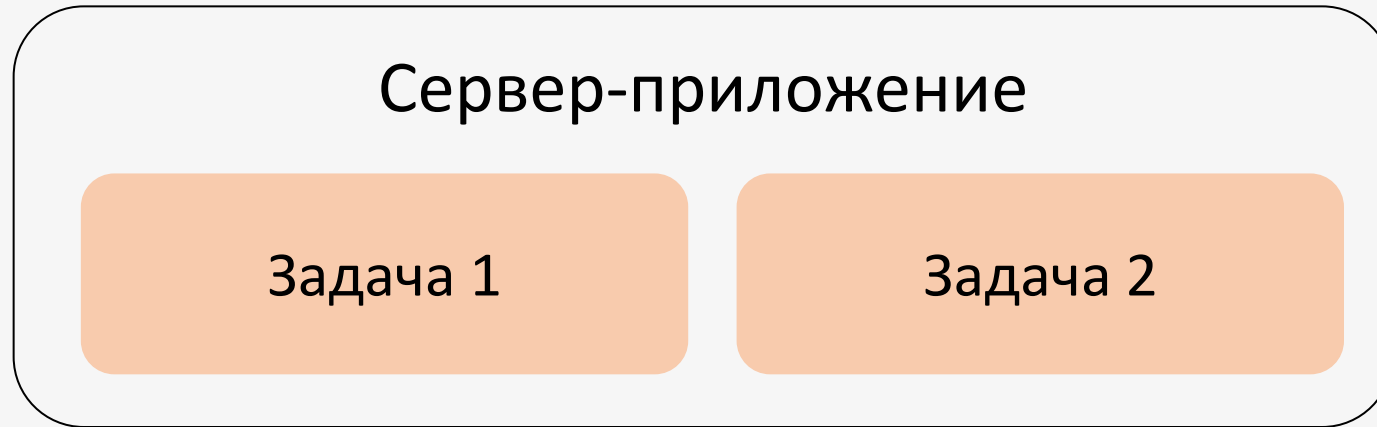
◆ Для чего?

- ◆ TraceKey позволяет упростить анализ системных трасс, собранных на узлах под управлением ЗОСРВ «Нейтрино» для дальнейшей диагностики, отладки и поиска ошибок в работе приложений или системы

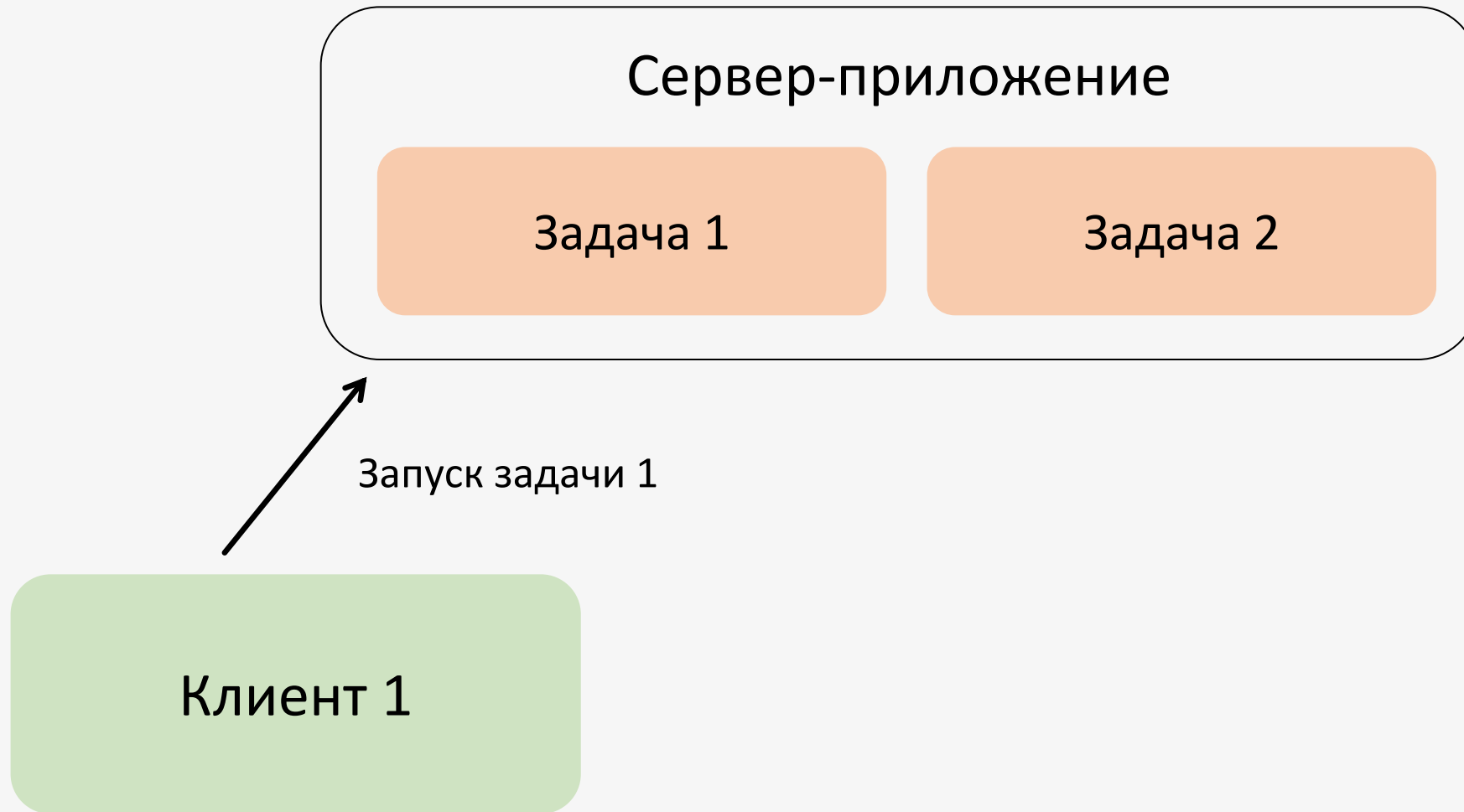
◆ Примеры использования

- ◆ Анализ характеристик реального времени приложений
- ◆ Исследование запуска системы
- ◆ Изучение межпроцессного взаимодействия программ (IPC)
- ◆ Нахождение ошибок или аномального поведения (задержки прерываний, ошибки в многопоточных приложениях, нарушения обмена сообщениями)

Проблема в многопоточном приложении



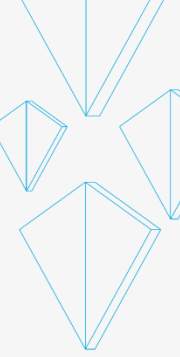
Проблема в многопоточном приложении



Проблема в многопоточном приложении



Проблема в многопоточном приложении



◆ Что происходит?

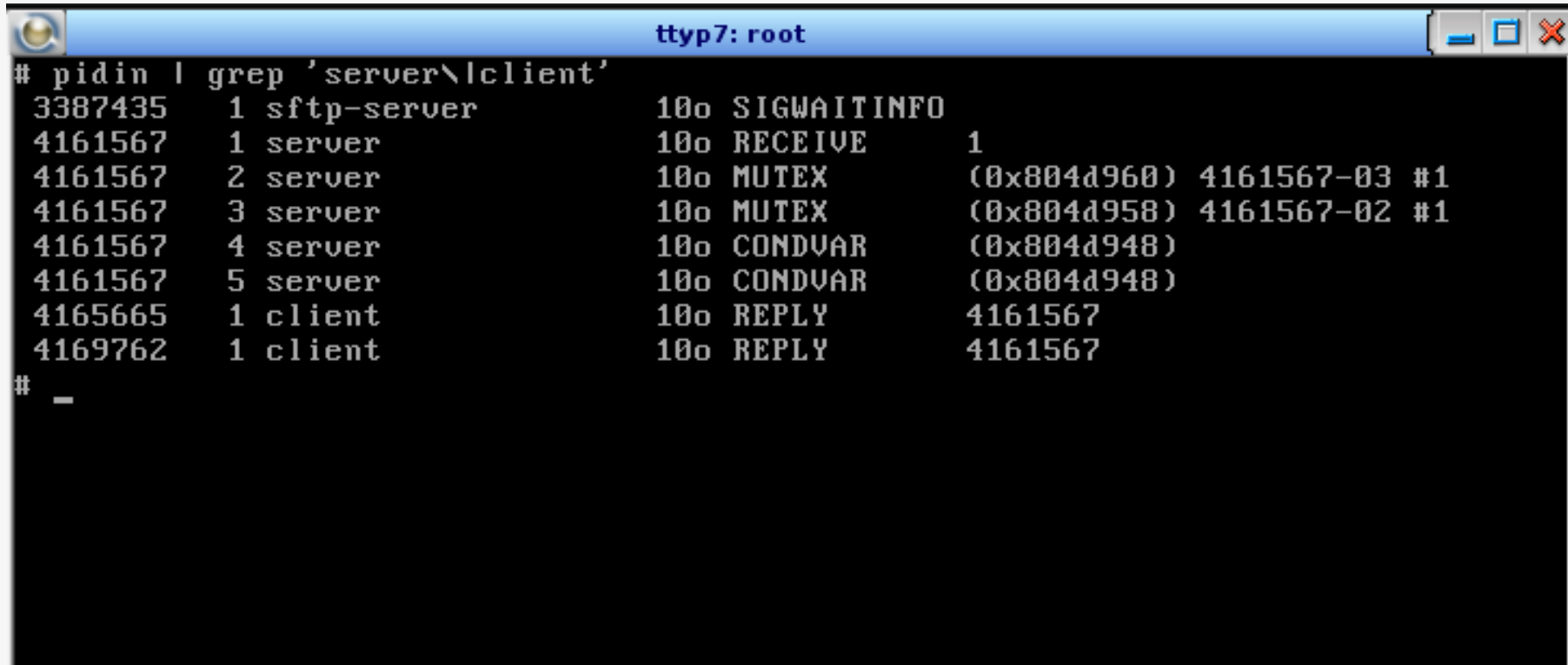
- ◇ Сервер перестает отвечать
- ◇ Клиенты не получают результат выполнения задачи

◆ Первые шаги анализа проблемы

- ◇ Определить в каком состоянии находятся потоки приложений

Анализ проблемы

- ◆ Для получения информации о состоянии потоков приложений сервера и клиента воспользуемся командой **pidin**



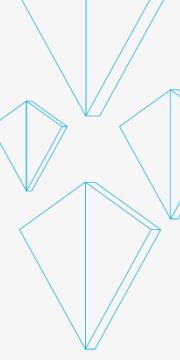
```
ttyp7: root
# pidin | grep 'server\|client'
3387435  1 sftp-server      10o SIGWAITINFO
4161567  1 server           10o RECEIVE         1
4161567  2 server           10o MUTEX            (0x804d960) 4161567-03 #1
4161567  3 server           10o MUTEX            (0x804d958) 4161567-02 #1
4161567  4 server           10o CONDVAR          (0x804d948)
4161567  5 server           10o CONDVAR          (0x804d948)
4165665  1 client           10o REPLY            4161567
4169762  1 client           10o REPLY            4161567
# _
```

Запись системной трассы

- ◆ В случае возможности воспроизведения проблемы, мы можем записать системную трассировку для дальнейшего анализа
- ◆ Запись системной трассы производится с помощью программы **tracelogger**
- ◆ После запуска **tracelogger**'а воспроизводим проблему, запуская сервер и клиенты

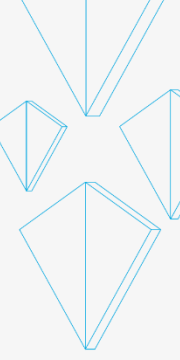
```
# tracelogger -w -s 15 -f ./client_server_1.kev
```


Запуск TraceKev

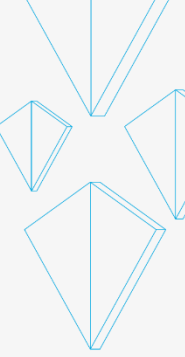


- ◆ Теперь перейдем в программу для анализа системных трасс «**TraceKev**» и откроем новый проект с собранным **kev** файлом

Промежуточные итоги анализа



- ◆ Клиенты и сервер находятся в блокировке
 - ◇ Клиенты находятся в состоянии **REPLY**
 - ◇ Потоки сервера переходят в состояние **MUTEX** во время работы над задачами
- ◆ Необходимо определить, какие мьютексы блокируются



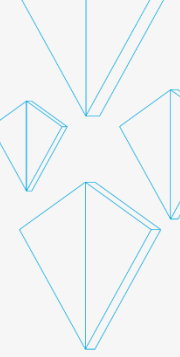
Дополнение кода сервера

- ◆ POSIX функции **pthread_mutex_lock()** и **pthread_mutex_unlock()** не во всех случаях приводят к системному вызову
- ◆ Оборачиваем блокировку и разблокировку мьютексов функцией **trace_logf()** для добавления пользовательских событий в лог системных событий
- ◆ Повторно собираем трассу с помощью **tracelogger**

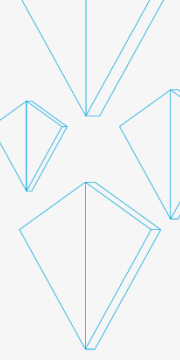
```
29 | trace_logf(rcvid, "res_1_lock started"); | 45 | trace_logf(rcvid, "res_1_unlock started");  
30 | resource1_mutex.lock(); | 46 | resource1_mutex.unlock();  
31 | trace_logf(rcvid, "res_1_lock finished"); | 47 | trace_logf(rcvid, "res_1_unlock finished");  
32 |
```

Итоги

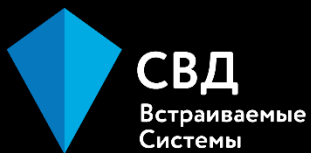
- ◆ На сервере запускается задача 1
 - ◇ Блокируется мьютекс 1
- ◆ На сервере запускается задача 2
 - ◇ Блокируется мьютекс 2
- ◆ Задача 1 ожидает разблокировки мьютекса 2 и его захвата
- ◆ Задача 2 ожидает разблокировки мьютекса 1 и его захвата
- ◆ Происходит **deadlock**, и клиенты остаются в заблокированном состоянии (**REPLY**)



Планы развития TraceKey



- ◆ Выпуск первой публичной версии
 - ◇ В составе обновления текущей редакции *Комплекта Разработчика*
 - ◇ Ориентировочные сроки – **4-й квартал 2024 года**
- ◆ Расширение функциональности
 - ◇ Обновление пользовательского интерфейса
 - ◇ Функции удаленного сбора трассы с целевых систем
 - ◇ Подробная информация о межпроцессных сообщениях (**IPC**)
 - ◇ Дополнительная информация о переключении контекста
 - ◇ Статистика нагрузки на процессор
 - ◇ Настройка пользовательского интерфейса (цветовые темы, разделяемые окна)
 - ◇ и т. д.



Спасибо за внимание!

Усов Дмитрий

Разработчик

Отдел инструментальных средств

ул. Кузнецовская, д. 19,

г. Санкт-Петербург

+7 (812) 346-89-56

www.kpda.ru

support@kpda.ru