Программный модуль «Speech to text»

**Руководство системного программиста**

**Листов \_15**

Инв. № подл.

Подп. и дата

Взам. инв. №

Инв. № дубл.

Подп. и дата

2023

**АННОТАЦИЯ**

Документ содержит описание структуры, процедуры установки, и проверки компонентов специального программного обеспечения «Программный модуль «Speech To Text»» и предназначен для системных программистов, осуществляющих подготовку программы к эксплуатации.

Настоящий документ предназначен для персонала, имеющего квалификацию системного администратора, обладающего знаниями и навыками в области системного администрирования, в том числе установки операционных систем на базе ядра Linux, например, ОС CentOS Linux версии 8 Stream, работы с СУБД Redis, работы с Ruby on Rails, Golang.

Документ не является руководством пользователя по установке и настройке, операционной системы, и прочего системного и технологического программного обеспечения.

**Содержание**

[1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ 4](#_Toc135234284)

[2. СТРУКТУРА ПРОГРАММЫ 5](#_Toc135234285)

[2.1. «Приложение сбора и потокового распознавания данных». 5](#_Toc135234286)

[2.2. «Система чтения трафика». 6](#_Toc135234287)

[3. УСТАНОВКА ПРОГРАММЫ 7](#_Toc135234288)

[3.1. Порядок установки 8](#_Toc135234289)

[3.1.1. Установка необходимого СПО 8](#_Toc135234290)

[3.1.2. Подготовка инфраструктуры 8](#_Toc135234291)

[3.1.3. Установка буфера данных 8](#_Toc135234292)

[3.1.4. Установка Ruby 9](#_Toc135234293)

[3.1.5. Установка torch 9](#_Toc135234294)

[3.1.6. Подготовка приложения 10](#_Toc135234295)

[3.1.7. Запуск приложения 12](#_Toc135234296)

[4. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 13](#_Toc135234297)

# 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

**1.1. Программа «Приложение сбора и потокового распознавания данных».**

Программа «Приложение сбора и потокового распознавания данных» обеспечивает выполнение функций, описанных в документе «Программный модуль «Speech To Text» Описание программы». Состав программы приведен в п. 2 настоящего документа.

Для обеспечения выполнения программы необходимы следующие технические средства:

Аппаратно-программные ресурсы выделенного сервера:

* не менее одного графического ускорителя, поддерживающего CUDA, с объемом видеопамяти не менее 6 Гб;
* операционная система на базе ядра Linux, например ОС CentOS Linux версии 8 Stream;
* СУБД: Redis версии не ниже 5;
* СПО: sox, git, rvm, ruby, gcc, CUDA, cuDNN, Python 3, libtorch.

**1.2. Система чтения трафика**

Система реализует механизм чтения трафика SIP, RTSP, SDP, MRCPv2, RTP/RTCP его разбор и передачу по REST API в сервис распознавания и преобразования услышанного близко к литературному написанию.

# 2. СТРУКТУРА ПРОГРАММЫ

Структурно программа состоит из 2 основных частей

## 2.1. «Приложение сбора и потокового распознавания данных».

Структура программы «Приложение сбора и потокового распознавания данных»:

Таблица 1 – Структура программы

|  |  |
| --- | --- |
| Название | Наименование |
| Буфер данных | Системная библиотека |
| Встроенный сервер | Программный сервис |
| HTTP-frontend | Программный модуль |
| GRPC-frontend | Программный модуль |
| Inference | Программный сервис |

Буфер данных - построен на базе Redis v5 или v6. Для очередей блоков данных и уведомлений используются очереди. Для хранения контекстов простое key-value хранилище.

Встроенный сервер – процесс, порождающий при старте свои форки, в идеале общим числом по числу ядер ЦПУ машины. Процесс выполняет следующие шаги:

* вычитывает с TCP-стека блоки данных размером до 64Киб;
* выполняет декодирование из формата, указанного в content-type заголовке, в pcm\_s16le;
* записывает декодированные pcm-фреймы в очередь блоков потока в буфер;
* записывает уведомление о появлении данных в очередь инферера по правилу inferrer\_id = request.uuid % num\_inferers;
* записывает метаданные аудиофайла, в частности – число каналов;
* асинхронно слушает очередь Inference и записывает результаты потоково в тело ответа.

HTTP – **frontend** – модуль поддержки HTTP-протокола для потокового распознавания аудио.

GRPC – **frontend** – модуль поддержки gRPC протокол для потокового распознавания аудио.

Inference – процесс, порождающий при старте свои копии (форки), в идеале, общим числом по количеству ГПУ машины. Процесс выполняет следующие шаги:

* слушает, блокируя очередь уведомлений;
* при появлении обновлений в каком-либо потоке читает из буфера его блоки данных и контекст;
* склеивает блоки данных и конвертирует бинарную строку во float32-тензоры фреймов поканально;
* вычисляет вывод ML-модели плавающим по данным тензорам окном размером в 1 с (размер окна зависит от конкретной модели);
* складывает хвосты тензоров в буфер в контекст потока для последующей склейки с новыми блокам данных.

## 2.2. «Система чтения трафика».

Механизм чтения трафика SIP, RTSP, SDP, MRCPv2, RTP/RTCP разбор и передача по rest-api в сервис распознавания [url](http://172.20.70.70:9999/v1/inference/callhold).

В трафике получаем данные из sip заголовков:

1. CALLERID

2. ACCOUNTCODE

3. X-CardNum

4. X-IntNum

5. X-ExtNum

6. CHANNELCIDNUM

7. CALACCOUNT

8. CHANNEL

9. UNIQUEID

И данные передаваемы из биллинга. В с помощью настроек сетевых интерфейсов, где разворачивается Asterisk, данные передаются на IP и порт указанный в конфигах пример ниже

```

iptables -t mangle -A PREROUTING -i $IN\_INT -p udp --dport 5060 -j TEE --gateway 127.0.0.1

iptables -t nat -A PREROUTING -i $IN\_INT -p udp --dport 5060 -j DNAT --to-destination IP\_DST:5060

sysctl net.ipv4.ipforward=1

```

Где:

- $IN\_INT - ip и порт текущего устройства

- %IP\_DST - ip и порт сервера, где работает sipsniff

# 3. УСТАНОВКА ПРОГРАММЫ

Перед развертыванием компонентов «Приложения сбора и потокового распознавания данных» на сервере должна быть установлена ОС на базе ядра Linux версии не ниже 4.18. К примеру: Centos Linux 8 Stream. Компоненты должны быть установлены на одном сервере.

Для системы чтения трафика

## 3.1. Порядок установки

## 3.1.1. Установка необходимого СПО

Для развертывания и эксплуатации «Приложения сбора и потокового распознавания данных» необходимо установить следующее СПО:

* git,
* gcc,
* sox,
* CUDA,
* cuDNN,
* Python 3 (конкретная версия зависит от требований libtorch).

## 3.1.2. Подготовка инфраструктуры

Для работы «Приложения сбора и потокового распознавания данных» в системе должен быть создан пользователь, от которого будет запущен системный сервис. В дальнейшем в данном руководстве для этих целей будет использован пользователь deploy. Для размещения исходного кода приложения созданы директории, доступные для записи пользователю deploy:

* /srv/inference\_app/current – для размещения приложения;
* /srv/inference\_app/bundle – для размещения модулей (гемов) Ruby;
* /var/log/stt/inference – для размещения логов.

## 3.1.3. Установка буфера данных

На сервере должна быть установлена СУБД Redis версии не ниже 5. Например, для сервера с ОС CentOS Linux версии 8 Stream это делается с помощью команды, выполняемой с административными правами (root или sudo):

# dnf module enable redis:5

# dnf install redis

После установки активировать и запускать службу redis не нужно.

## 3.1.4. Установка Ruby

«Приложения сбора и потокового распознавания данных» требуется версия Ruby не ниже 3.0. В настоящее время для ОС CentOS Linux версии 8 Stream данная версия Ruby доступна к установке через RVM:

# gpg --keyserver hkp://pool.sks-keyservers.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB

# \curl -sSL https://get.rvm.io | bash

# rvm install ruby-3.0.0

Для дальнейшей пакетной установки модулей Ruby требуется установить bundler:

# gem install bundler

## 3.1.5. Установка torch

В процессе работы «Приложение сбора и потокового распознавания данных» использует libtorch, исходный код которого доступен по ссылке:

https://github.com/pytorch/pytorch/archive/refs/heads/master.zip

Процесс компилирования libtorch может отличаться от версии к версии. Актуальная документация по его установке доступна по ссылке:

https://github.com/pytorch/pytorch/blob/master/docs/libtorch.rst

В дальнейшем в данной инструкции предполагается, что компилированная версия libtorch находится в «/opt/libtorch»

Сервис реализован на golang v1.15 находится в репозиториях

[sipsniffer](https://git.zonatelecom.ru/stt/sipsniffer), разборщик и проксировщик [sipdaemon](https://git.zonatelecom.ru/stt/sipdaemon). Код должен собираться и запускаться на сервере в среде linux debian 10, как системная служба пример ниже.

```

service sip\_sniffer start

## 3.1.6. Подготовка приложения

Исходный код приложения копируется в папку «/srv/inference-app/current», которой назначаются права доступа на запись пользователем deploy.

Для подготовки приложения к развертыванию необходимо произвести настройку bundle, для чего необходимо создать файл «.bundle/config» следующего содержания:

---

BUNDLE\_PATH: "/srv/inference\_app/bundle"

BUNDLE\_BUILD\_\_TORCH-RB: "--with-torch-include=/opt/libtorch/include/ --with-torch-lib=/opt/libtorch/lib"

BUNDLE\_BUILD\_\_TORCHAUDIO: "--with-torch-include=/opt/libtorch/include/ --with-torch-lib=/opt/libtorch/lib --with-sox-include=/usr/include/sox/"

Для подготовки приложения к запуску необходимо установить необходимые модули (гемы) Ruby, что делается командой от пользователя, под которым

$ bundle install

Сборка исходного файла

CGO\_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o sip\_sniffer

На выходе будет bin файл который необходимо добавить как службу.

Описание сетевой части. На хосте осуществляем сбора зеркалированного трафика

# /etc/network/interfaces

iface ens18 inet static

post-up /bin/sysctl -w net.ipv4.conf.${IFACE}.arp\_ignore=1 || true

auto ens18\_macvlan0

iface ens18\_macvlan0 inet manual

hwaddress 02:01:72:20:70:22

pre-up /sbin/ip link add link ens18 name ${IFACE} type macvlan mode private

up /sbin/ip address add 172.20.70.22/32 dev ${IFACE}

up /sbin/ip link set up dev ${IFACE}

post-up /bin/sysctl -w net.ipv4.conf.${IFACE}.arp\_ignore=1 || true

post-down /sbin/ip link del ${IFACE} || true

Перенаправление входящего трафика в приложение для анализа

iptables -t nat -A PREROUTING -i ens18\_macvlan0 -p udp -m udp --dport 5060 -j DNAT --to-destination 172.20.70.22:5061

Проверка, при которой если приложение не запущено, данные не перенаправляются в сеть.

iptables -t filter -A INPUT -i ens18\_macvlan0 -p udp -m conntrack --ctstate DNAT --ctorigdstport 5060 -m socket --nowildcard -j ACCEPT

iptables -t filter -A INPUT -i ens18\_macvlan0 -p udp -m conntrack --ctstate DNAT --ctorigdstport 5060 -j DROP

Для подготовки приложения к развертыванию необходимо произвести настройку

# /etc/systemd/system/sip\_sniffer.service

[Unit]

Requires=network.target

After=network-online.target

[Service]

Type=simple

WorkingDirectory=/opt/sip\_sniffer

ExecStart=/bin/bash -c '/opt/sip\_sniffer/server >> /home/mlagutin/sip\_sniffer.txt'

Restart=always

[Install]

WantedBy=multi-user.target

## 3.1.7. Запуск приложения

Для автоматического запуска приложения при перезагрузке системы используется сценарий systemd, располагаемый в «/etc/systemd/system/stt-inference.service»:

[Unit]

Description=STT Backend Inference

[Service]

Type=simple

User=deploy

Environment=RAILS\_LOG\_TO\_STDOUT=true

Environment=CONSOLE\_OUTPUT=XTerm

WorkingDirectory=/srv/inference\_app/current

ExecStart=/usr/local/rvm/wrappers/ruby-3.0.0/bundle exec ruby app.rb -n 8 -b http://0.0.0.0:9999

StandardOutput=append:/var/log/stt/inference/app.log

StandardError=append:/var/log/stt/inference/app.log

SyslogIdentifier=stt-inference

TimeoutSec=15

Restart=always

[Install]

WantedBy=multi-user.target

Для запуска приложения и активирования автоматического запуска при перезагрузке необходимо выполнить:

# systemctl daemon-reload

# systemctl start stt-inference.service

# systemctl enable stt-inference.service

# 4. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС - операционная система

СУБД - система управления базами данных

СПО - специальное программное обеспечение

МИНОС- Личный кабинет сотрудника ФСИН России.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц)  в докум. | №  документа | Входящий № сопрово-дительного документа и дата | Подп. | Дата |
| Изм. | изменен-ных | заменен-ных | новых | аннули-рован-ных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |