

Система видеоаналитики с интеллектуальными модулями OMV  
Инструкция по установке

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

## Аннотация

Настоящий документ представляет собой руководство по установке системы видеоаналитики с интеллектуальными модулями OMV.

Име № подл.		Взамен инв. №		Име № дубл.		Подл. и дата	
Подл. и дата							
						<i>Система видеоаналитики с интеллектуальными модулями OMV</i>	
	<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		
	<i>Разраб.</i>						
	<i>Пров.</i>						
	<i>Н. контр.</i>						
						<i>Инструкция по установке</i>	
						Лит	Лист
						1	35

## Оглавление

1	Системные требования.....	4
1.1	Системные требования .....	4
2	Подготовка к работе .....	6
2.1	Предварительные условия перед установкой системы .....	6
3	Установка системы.....	7
3.1	Установка сервиса бизнес логики.....	7
3.2	Установка сервиса Видеоанализа .....	9
3.2.1	Установка программного обеспечения .....	9
3.2.2	Конфигурация программного обеспечения .....	10
3.2.3	Управление программным обеспечением.....	34
4	Запуск системы .....	35
4.1	Проверка работы компонентов .....	35
4.2	Запуск приложения .....	35

Инь № подл.	Взамен инв. №	Инь № дубл.	Подп. и дата

# 1 Системные требования

## 1.1 Системные требования

Для запуска системы необходимо иметь следующую аппаратную инфраструктуру:

### Сервер Видеоанализа

Позиция	Минимальные требования
Процессор	2xIntel Xeon Gold 6248R
Память (RAM)	256GB DDR4-3200
Диски (SSD)	480ГБ SSD SATA
RAID	LSI 9460-8I
Видеокарта	NVIDIA RTX A5000 (24GB) x 4шт.
Сетевая карта	не хуже Intel X710 2 port 10GbE
Корпус	4U

### Сервер бизнес логики

Позиция	Минимальные требования
Процессор	2xIntel Xeon Silver 4410Y
Память (RAM)	128GB DDR4-3200
Диски (SSD)	480ГБ SSD SATA
RAID	LSI 9460-8I
Видеокарта	встроенная
Сетевая карта	не хуже Intel X710 2 port 10GbE
Корпус	2U

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

**Сервер рестриминга**

<b>Позиция</b>	<b>Минимальные требования</b>
Процессор	2xIntel Xeon Silver 4410Y
Память (RAM)	128GB DDR4-3200
Диски (SSD)	480ГБ SSD SATA
RAID	LSI 9460-8I
Видеокарта	встроенная
Сетевая карта	не хуже Intel X710 2 port 10GbE
Корпус	2U

**Сервер рестриминга**

<b>Позиция</b>	<b>Минимальные требования</b>
Процессор	Intel Core i5
Память (RAM)	16GB
Диски (SSD)	480 GB
HDD	1TB
Видеокарта	PCI-E nVidia T1000 8GB GDDR6 128bit 4*mDP
Сетевая карта	1Gbs

**Клиентская часть**

<b>Программный продукт</b>	<b>Версия</b>
Операционная система	Ubuntu
Браузер	Google Chrome 110 и выше

Инь № дубл.	Инь № дубл.	Инь № дубл.	Инь № дубл.
Взамен инв. №	Взамен инв. №	Взамен инв. №	Взамен инв. №
Подп. и дата	Подп. и дата	Подп. и дата	Подп. и дата
Инь № подл.	Инь № подл.	Инь № подл.	Инь № подл.

## 2 Подготовка к работе

### 2.1 Предварительные условия перед установкой системы

Для работы системы предварительно должны быть установлены:

Для сервиса Видеоанализа:

- операционная система Ubuntu версии 20.04 без предустановленных пакетов и дополнительных конфигураций.

Для сервиса бизнес логики:

- операционная система Ubuntu версии 20.04 без предустановленных пакетов и дополнительных конфигураций.
- дистрибутивный пакет бизнес логики `css-host_1.0.0.0_amd64.deb`
- Сервер СУБД PostgreSQL
- Ffmpeg. Необходим в работе сервиса комплексной системы безопасности в части арг

Инв. № подл.	Подп. и дата	Взамен инв. №	Инв. № дубл.	Подп. и дата	Инструкция по установке					Лист				
										5				
										Изм.	Лист.	№ документа	Подп.	Дата.

### 3 Установка системы

#### 3.1 Установка сервиса бизнес логики

##### 1. Установка сервера СУБД PostgreSQL и БД

###### 1.1. Обновление списков пакетного менеджера

- `sudo apt-get update`

###### 1.2. Установка сервера

- `sudo apt-get install postgresql`

- `sudo -u postgres psql`

- `\password`

• Задаем пароль для учетки postgres (например admin). Вводим 2 раза.

- `\q`

###### 1.3. Создание БД

- `sudo -u postgres psql`

- `create database "CSS_DB";`

- `\q`

###### 1.4. Установка начальной структуры БД с бэкапа

• Скопировать файл бэкапа БД `css_db.backup` в текущий домашний (home) каталог

• `pg_restore --host "localhost" --port "5432" --username "postgres" --dbname "CSS_DB" --verbose "css_db.backup"`

Будет запрошен пароль от учетки postgres (например, admin)

##### 2. Установка ffmpeg

- `sudo apt-get install ffmpeg`

##### 3. Установка пакета сервиса бизнес логики.

3.1. Скопировать файл `css-host_1.0.0.0_amd64.deb` в текущий домашний (home) каталог

Инь № дубл.	Подп. и дата
Взамен инв. №	
Инь № дубл.	Подп. и дата
Инь № подл.	

Изм.	Лист.	№ документа	Подп.	Дата.

### 3.2. Установка пакета

- `sudo apt-get install ./css-host_1.0.0.0_amd64.deb`

3.3. Удаление пакета (при необходимости, если надо удалить ранее установленный)

- `sudo apt-get remove css-host`

### 3.4. Настройка конфигурации сервиса

- Необходимо отредактировать файл `/opt/CSS_Host/appsettings.json` (см.ниже)

### 3.5. Перезапустить сервис

- `sudo systemctl -f -q restart CSS_Host`

#### **Параметры файла конфигурации сервиса бизнес логики:**

// Порт прослушки, на котором работает сервис

```
"urls": "http://*:60111",
```

// Включить репликацию событий через SignalR для OmegaBI

```
"OmegaEvents": false,
```

// Строка соединения с БД сервиса

```
"ConnectionStrings": {
```

```
  "ConStr": "Server=localhost;Port=5432;User
```

```
Id=postgres;Password=admin;Database=CSS_DB;CommandTimeout=240"
```

```
},
```

// Период пинга (сек) для устройства по умолчанию, если в устройстве

включено пингование, но не задан конкретно период

```
"PingTime": 120,
```

// Включение в сервисе сваггера (в промышленной эксплуатации стоит

выключить)

```
"SwaggerEnable" : true,
```

// Путь к исполняемому файл ffmpeg

```
"FFMpegFile": "/usr/bin/ffmpeg",
```

Подп. и дата	
Инь № дубл.	
Взамен инв. №	
Подп. и дата	
Инь № подл.	

						<i>Инструкция по установке</i>	Лист
Изм.	Лист.	№ документа	Подп.	Дата.			7

Параметры, которые не указаны здесь, не менять. Большинство параметров, если другое ПО установлено по умолчанию, менять не требуется.

### 3.2 Установка сервиса Видеоанализа

#### 3.2.1 Установка программного обеспечения

##### 1. Копирование файлов поставки в операционную систему

Необходимо перенести файлы поставки с цифрового носителя в директорию сервера:

```
/var/sinaps-system
```

Предварительно создав директорию командой:

```
sudo mkdir -p /var/sinaps-system
```

##### 2. Запуск установочного скрипта

Необходимо перейти в директорию ПО:

```
cd /var/sinaps-system
```

Необходимо установить права на запуск скрипта:

```
chmod +x control.sh
```

Запускаем процесс установки командой:

```
sudo ./control.sh install
```

В процессе установки скрипт запросит конфигурационные данные:

Переменная	Значение (необходимо заменить вручную)
<b>\$SIGNALR_ENDPOI</b> <b>NT_HOST</b>	IP-адрес или доменное имя сервера SignalR.

Изм.	Лист.	№ документа	Подп.	Дата.

<b>\$BL_SERVER_ENDP</b>	IP-адрес или доменное имя сервера
<b>OINT_HOST</b>	БЛ.

3. Дожидаемся перезагрузки сервера после установки

### 3.2.2 Конфигурация программного обеспечения

1. Актуализирование нейросетевых моделей

Необходимо для обновления нейросетевых моделей, в процессе установки можно пропустить данный шаг – модели уже присутствуют в дистрибутиве.

Требуется доступ к публичному ресурсу: <https://cloud-api.yandex.net:443>.

```
sudo chmod +x /var/sinaps-system/sinaps-trt-server/fetch_models.sh
```

```
cd /var/sinaps-system/sinaps-trt-server/ && sudo ./fetch_models.sh
```

2. Конфигурация интеграции с сервером бизнес-логики

Установочный скрипт автоматически заполняет данные конфигурационные файлы на основании предоставленных данных в момент установки.

```
sudo nano /var/sinaps-system/signalr/appsettings.json
```

После изменения конфигурации: `cd /var/sinaps-system && sudo docker-compose build --no-cache signalr`

**Содержимое конфигурационного файла:**

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
}
```

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						9

```

"URL": "http://$SIGNALR_ENDPOINT_HOST:60111/api/device/events",
"Channel": "Requests",
"InnerURL": "tcp://0.0.0.0:5555",
"RetryTimeout": 0,
"ExchangeMode": 2,
"MockMode": false
}

```

Переменная	Значение (необходимо заменить вручную)
<b>\$SIGNALR_ENDPOINT_HOST</b>	IP-адрес или доменное имя сервера SignalR.

```
sudo nano /var/sinaps-system/triton_client-variables.env
```

### Содержимое конфигурационного файла:

```

# Логин, пароль и адрес сервера бизнес логики
BL_LOGIN=admin
BL_PASSWORD=admin
BL_BASE_URL=http://$BL_SERVER_ENDPOINT_HOST:60111
# Адрес очереди сообщений от сервера бизнес логики
SIGNALR_URL=tcp://172.33.0.13:5555
# Логин, пароль и адреса компонентов базы данных лиц
ES_LOGIN=admin
ES_PASSWORD=admin
ES_API_SERVICE_BASE_URL=http://172.33.0.24:8080
ES_CLUSTER_BASE_URL=https://172.33.0.21:9200

```

Инь № дубл.	Взамен инв. №	Инь № дубл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						10

ES\_KIBANA\_BASE\_URL=http://172.33.0.23:5601

# Адрес сервера моделей машинного обучения

TRITON\_HTTP\_HOST=172.33.0.11

TRITON\_HTTP\_PORT=8000

# Адреса рестримера

RESTREAMER\_RTSP\_BASE\_URL=rtsp://172.33.0.25:8554

RESTREAMER\_HTTP\_BASE\_URL=http://172.33.0.25:9997

Переменная	Значение (необходимо заменить вручную)
<b>\$BL_SERVER_ENDP</b> <b>OINT_HOST</b>	IP-адрес или доменное имя сервера БЛ.

sudo nano /var/sinaps-system/video\_reader\_service-variables.env

### Содержимое конфигурационного файла:

# Адрес алгоритмического модуля

CLIENT\_HOST=172.33.0.12

### 3. Конфигурация пути для сохранения видеоархива

В файле /var/sinaps-system/docker-compose.yml необходимо указать путь до директории хранилища в секции rtsp-restreamer->volumes (выделено на скриншоте ниже), а также в секции video\_reader\_service->volumes по аналогии:

Инь № дубл.	Подп. и дата
Инь № дубл.	Подп. и дата
Взамен инь. №	Подп. и дата
Инь № подл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

```

rtsp-restreamer:
  image: omegalab/sinaps-rtsp-restreamer:latest
  build:
    context: ./sinaps-rtsp-restreamer
    dockerfile: Dockerfile
  container_name: rtsp-restreamer
  restart: always
  volumes:
    - /var/lib/libvirt/sinaps-rtsp-restreamer-records/dvr:/dvr
  networks:
    omega_net:
      ipv4_address: 172.33.0.25
  ports:
    - 9997:9997
    - 9998:9998
    - 8888:8888
    - 8554:8554

```

## 4. Конфигурация аппаратных ускорителей (GPU)

### 4.1. Трекер объектов

<b>Системное наименование:</b>	DeepSort_tensorflow
--------------------------------	---------------------

```

sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/DeepSort_tensorflow/config.pbtxt

```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						12

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного GPU
```

```
gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров
```

```
}
```

```
]
```

```
...
```

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

Подп. и дата
Инв № дубл.
Взамен инв. №
Подп. и дата
Инв № подл.

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

## 4.2. Распознавание лиц

<b>Системное наименование:</b>	arcface_onnx
--------------------------------	--------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/arcface_onnx/config.pbtxt
```

### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного GPU
```

```
gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров
```

```
}
```

Инь № подл.	Подп. и дата			
Взамен инв. №	Инь № дубл.			
Подп. и дата	Подп. и дата			
Изм.	Лист.	№ документа	Подп.	Дата.

]

...

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

### 4.3. Детектор лиц

<b>Системное наименование:</b>	face_detection_tensorflow
--------------------------------	---------------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/face_detection_tensorflow/config.pbtxt
```

Инь № подл.	Подл. и дата
Взамен инв. №	Инь № дубл.
Подл. и дата	
Инь № подл.	

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						15

## Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается  
количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного  
GPU
```

```
gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-  
экземпляров
```

```
}
```

```
]
```

...

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

Инь № подл.	Подп. и дата	Взамен инв. №	Инь № дубл.	Подп. и дата
-------------	--------------	---------------	-------------	--------------

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

#### Горизонтальное масштабирование:

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

#### Вертикальное масштабирование:

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

### 4.4. Считыватель особенностей лица

<b>Системное наименование:</b>	face_landmarks_onnx
--------------------------------	---------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/face_landmarks_onnx/config.pbtxt
```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

Инь № дубл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата
Инь № подл.	Инь № подл.

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						17

`count: 1 # <- для горизонтального масштабирования указывается количество = 1`

`gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)`

`},`

`# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)`

`{`

`kind: KIND_GPU`

`count: 2 # <- тут указывается количество экземпляров модели для одного GPU`

`gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров`

`}`

`]`

`...`

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

Изм.	Лист.	№ документа	Подп.	Дата.

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						18

«count»: указываем всегда значение = 1;

«gpus»: указываем массив ID GPU для использования;

### Вертикальное масштабирование:

«count»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«gpus»: указываем конкретный ID GPU для использования в формате числа;

### 4.5. Детектор ГРЗ

<b>Системное наименование:</b>	number_plate_detector_tensorrt
--------------------------------	--------------------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/number_plate_detector_tensorrt/config.pbtxt
```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

Инь № подл.	Подп. и дата			
Взамен инв. №	Инь № дубл.			
Подп. и дата	Подп. и дата			
Изм.	Лист.	№ документа	Подп.	Дата.

**count: 2 #** <- тут указывается количество экземпляров модели для одного GPU

**gpus: 2 #** <- тут указывается ID GPU, на которой будет создано count-экземпляров

}

]

...

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

#### 4.6. Считыватель ГРЗ

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						20

<b>Системное наименование:</b>	number_plate_reader_tensorrt
--------------------------------	------------------------------

sudo nano /var/sinaps-system/sinaps-trt-server/model\_repository/number\_plate\_reader\_tensorrt/config.pbtxt

**Область “интереса” в конфигурационном файле:**

...

instance\_group [

# Horizontal scaling by num of GPUs (attr: gpus)

{

kind: KIND\_GPU

count: 1 # <- для горизонтального масштабирования указывается количество = 1

gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)

},

# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)

{

kind: KIND\_GPU

count: 2 # <- тут указывается количество экземпляров модели для одного GPU

gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров

}

]

...

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке

Лист
21

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

#### Горизонтальное масштабирование:

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

#### Вертикальное масштабирование:

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

#### 4.7. Детектор объектов

<b>Системное наименование:</b>	object_detector_onnx
--------------------------------	----------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/object_detector_onnx/config.pbtxt
```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						22

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается  
количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного  
GPU
```

```
gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-  
экземпляров
```

```
}
```

```
]
```

```
...
```

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

Инь № подл.	Подл. и дата
Взамен инв. №	Инь № дубл.
Подл. и дата	Подл. и дата

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«count»: указываем всегда значение = 1;

«gpus»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«count»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«gpus»: указываем конкретный ID GPU для использования в формате числа;

**4.8. Детектор СИЗ**

<b>Системное наименование:</b>	ppe_detector_tensorflow
--------------------------------	-------------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/ppes_detector_tensorflow/config.pbtxt
```

**Область “интереса” в конфигурационном файле:**

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						24

```

    },
    # Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
    {
        kind: KIND_GPU

        count: 2 # <- тут указывается количество экземпляров модели для одного
        GPU

        gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-
        экземпляров
    }
]
...

```

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист 25

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

#### 4.9. Детектор текста

<b>Системное наименование:</b>	text_detector_onnx
--------------------------------	--------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/text_detector_onnx/config.pbtxt
```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного GPU
```

Инь № дубл.	Инь № дубл.	Взамен инв. №	Подп. и дата	Подп. и дата
-------------	-------------	---------------	--------------	--------------

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						26

**gpus: 2 #** <- тут указывается ID GPU, на которой будет создано **count-экземпляров**

}

]

...

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “**интереса**” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

#### 4.10. Детектор атипичного поведения

Изм.	Лист.	№ документа	Подп.	Дата.

Подп. и дата

Инь № дубл.

Взамен инв. №

Подп. и дата

Инь № подл.

<b>Системное наименование:</b>	atypical_behavior_estimator_tensorrt
--------------------------------	--------------------------------------

sudo nano /var/sinaps-system/sinaps-trt-server/model\_repository/atypical\_behavior\_estimator\_tensorrt/config.pbtxt

**Область “интереса” в конфигурационном файле:**

...

instance\_group [

# Horizontal scaling by num of GPUs (attr: gpus)

{

kind: KIND\_GPU

count: 1 # <- для горизонтального масштабирования указывается количество = 1

gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)

},

# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)

{

kind: KIND\_GPU

count: 2 # <- тут указывается количество экземпляров модели для одного GPU

gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров

}

]

...

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке

Лист
28

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

#### Горизонтальное масштабирование:

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

#### Вертикальное масштабирование:

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

### 4.11. Детектор огня

<b>Системное наименование:</b>	gun_fire_rifle_detector_tensorrt
--------------------------------	----------------------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/gun_fire_rifle_detector_tensorrt/config.pbtxt
```

#### Область “интереса” в конфигурационном файле:

...

```
instance_group [
```

Инь № дубл.	Подп. и дата
Инь № дубл.	Подп. и дата
Взамен инв. №	Подп. и дата
Инь № подл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						29

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается  
количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

```
},
```

```
# Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 2 # <- тут указывается количество экземпляров модели для одного  
GPU
```

```
gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-  
экземпляров
```

```
}
```

```
]
```

```
...
```

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.
------	-------	-------------	-------	-------

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«count»: указываем всегда значение = 1;

«gpus»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

«count»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«gpus»: указываем конкретный ID GPU для использования в формате числа;

**4.12. Детектор дыма**

<b>Системное наименование:</b>	smoke_detector_tensorrt
--------------------------------	-------------------------

```
sudo nano /var/sinaps-system/sinaps-trt-server/model_repository/smoke_detector_tensorrt/config.pbtxt
```

**Область “интереса” в конфигурационном файле:**

...

```
instance_group [
```

```
# Horizontal scaling by num of GPUs (attr: gpus)
```

```
{
```

```
kind: KIND_GPU
```

```
count: 1 # <- для горизонтального масштабирования указывается количество = 1
```

```
gpus: [ 0, 1 ] # <- тут указываем ID GPU (массив)
```

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						31

```

    },
    # Vertical scaling by num of instances for specific GPU (attrs: gpus, count)
    {
        kind: KIND_GPU

        count: 2 # <- тут указывается количество экземпляров модели для одного GPU

        gpus: 2 # <- тут указывается ID GPU, на которой будет создано count-экземпляров
    }
]
...

```

Массив «instance\_group» принимает произвольное количество вложенных объектов.

Объект состоит из трех обязательных атрибутов:

«**kind**»: значение всегда KIND\_GPU;

«**count**»: количество экземпляров (число);

«**gpus**»: ID GPU (массив или число);

В области “интереса” приведен пример конфигурации, в котором отражены два возможных варианта масштабирования нейросетевой модели на произвольное количество аппаратных ускорителей GPU:

**Горизонтальное масштабирование:**

«**count**»: указываем всегда значение = 1;

«**gpus**»: указываем массив ID GPU для использования;

**Вертикальное масштабирование:**

Инь № подл.	Подп. и дата
Взамен инв. №	Инь № дубл.
Подп. и дата	

Изм.	Лист.	№ документа	Подп.	Дата.	Инструкция по установке	Лист
						32

«**count**»: указываем количество экземпляров исходя из объема оперативной памяти GPU;

«**gpus**»: указываем конкретный ID GPU для использования в формате числа;

### 3.2.3 Управление программным обеспечением

#### 1. Запуск:

```
cd /var/sinaps-system
```

```
sudo ./control.sh start
```

#### 2. Остановка:

```
cd /var/sinaps-system
```

```
sudo ./control.sh stop
```

#### 3. Перезагрузка:

```
cd /var/sinaps-system
```

```
sudo ./control.sh restart
```

#### 4. Вывод отладочной информации:

```
cd /var/sinaps-system
```

```
sudo ./control.sh logs
```

Инь № подл.	Подп. и дата	Взамен инв. №	Инь № дубл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.

## 4 Запуск системы

### 4.1 Проверка работы компонентов

Убедиться, что запущены все компоненты, установка которых проходила в предыдущих пунктах и имеется доступ к БД.

### 4.2 Запуск приложения

Для входа в Систему необходимо выполнить следующие действия:

1. Запустить программу, используя ярлык Системы, расположенный на рабочем столе пользователя.

2. При этом откроется окно для прохождения авторизации пользователя.

Пример окна авторизации на Рисунок 1.

#### Вход в систему

Логин

Пароль

Войти

Рисунок 1 - Окно авторизации пользователя

3. В поле Логин указать имя пользователя Системы (логин и пароль должны быть переданы Администратором Системы).

4. В поле Пароль указать пароль пользователя Системы.

Инь № подл.	Подп. и дата	Взамен инв. №	Инь № дубл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.

5. Нажать кнопку Войти.

В результате выполнения указанных действий произойдет авторизация пользователя и вход в Систему. Внешний вид главного окна Системы представлен на Рисунок 2.

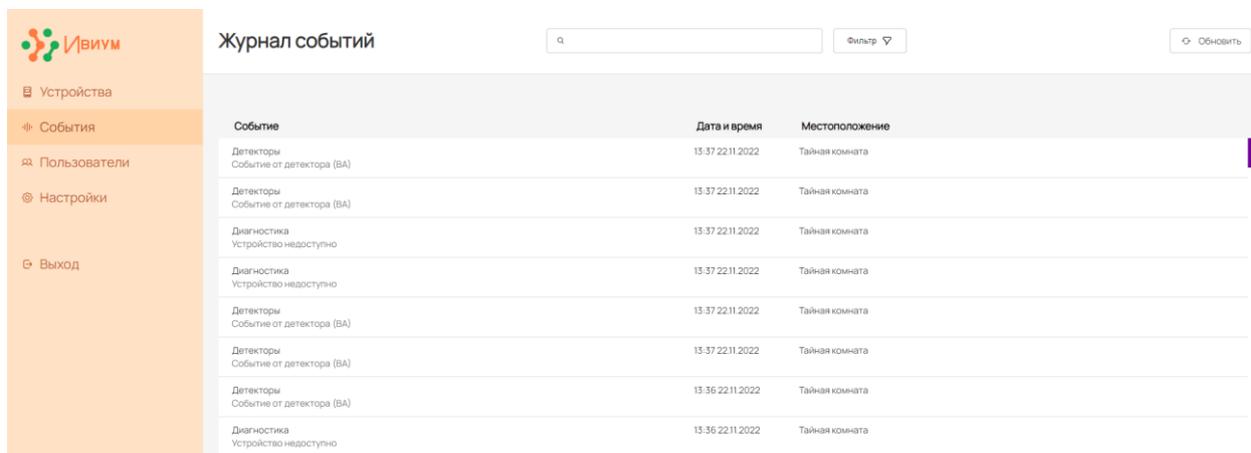


Рисунок 2 - Главное окно Системы

Инь № подл.	Подп. и дата	Взамен инв. №	Инь № дубл.	Подп. и дата

Изм.	Лист.	№ документа	Подп.	Дата.