



VISIONLABS LP5 INDEX

Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации ПО

Оглавление

1 Глоссарий.....	3
2 Введение.....	4
3 Аппаратные и программные требования.....	5
3.1 Минимальные аппаратные требования.....	5
3.2 Программные требования.....	5
4 Описание работы Основные положения.....	6
4.1 Индекс.....	6
4.1.1 Связь индекса с версией биометрического шаблона.....	6
4.1.2 Структура индекса.....	6
4.1.3 Процесс создания задачи на построение индекса.....	7
4.1.4 Процесс создания индекса.....	7
4.2 Сравнение.....	8
4.2.1 Запросы на сравнение.....	8
4.2.2 Процесс сравнения.....	8
5 Взаимодействие сервисов.....	10

1 Глоссарий

Термин	Определение
LUNA PLATFORM 5	<p>Система распознавания лиц и тел.</p> <p>ПО VisionLabs LUNA PLATFORM 5 включено в Единый реестр российских программ для электронных вычислительных машин и баз данных</p> <p>Запись в реестре №12557 от 14.01.2022 произведена на основании поручения Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 14.01.2022 по протоколу заседания экспертного совета от 10.01.2022 №П10-12пр</p>

Сокращенное название	Расшифровка
VL-LP5-I	VisionLabs LP5 Index
LP	VisionLabs LUNA PLATFORM 5
DB	База данных
API	LUNA PLATFORM 5 API
Faces	LUNA PLATFORM 5 Faces
Python Matcher	LUNA PLATFORM 5 Python Matcher
Python Matcher Proxy	LUNA PLATFORM 5 Python Matcher Proxy
Admin	LUNA PLATFORM 5 Admin
Configurator	LUNA PLATFORM 5 Configurator
Licenses	LUNA PLATFORM 5 Licenses

2 Введение

VisionLabs LP5 Index (Далее – Модуль VL-LP5-I) – это дополнительный функциональный модуль продукта LUNA PLATFORM 5, используемый для построения индекса на основе существующих БШ для их последующего быстрого сравнения.

3 Аппаратные и программные требования

3.1 Минимальные аппаратные требования

Для работы Модуля VL-LP5-I необходим запущенный экземпляр LUNA PLATFORM 5.

Следующие минимальные системные требования необходимы для установки программного пакета LUNA PLATFORM 5:

- CPU Intel, минимум 4 физических ядра с тактовой частотой 2.0 GHz или выше. Требуется поддержка набора инструкций AVX2 для CPU;
- RAM DDR3 (рекомендуется DDR4), 8 Гб или выше.
- Свободное место на диске - минимум 80 Гб.

Рекомендуется использование SSD для баз данных и хранилищ Image Store.

- Доступ к Интернету (для контейнеров и дополнительных загрузок ПО).

Примечание Приведенная выше конфигурация обеспечит минимальную мощность для работы ПО, но для использования системы в продуктивном контуре этого недостаточно. Требования для использования системы в продуктивном контуре рассчитываются в зависимости от предполагаемой нагрузки.

GPU

Для ускорения GPU необходим NVIDIA GPU. Поддерживаются следующие архитектуры:

- Pascal или более новые.

Требуется минимум 6Гб оперативной или выделенной видеопамяти. Рекомендуется 8 Гб VRAM или более.

На сервере должно быть установлено ПО CUDA версии 11.2. Рекомендуемые драйверы NVIDIA - r450, r455.

Для корректной работы приложения аппаратное обеспечение должно отвечать следующим минимальным требованиям:

- CPU с частотой 2 ГГц и выше;
- 4 Гб оперативной памяти и выше;
- 400 Мб свободного места на жестком диске.

3.2 Программные требования

Для запуска LUNA PLATFORM с Модулем VL-LP5-I может использоваться RedOS (ПЕД ОС) версии 7.3 и выше, CentOS версии 7.8 и выше.

4 Описание работы Основные положения

Дополнительный модуль индексирования VisionLabs LP5 Index:

- отправляет запросы на индексацию списка с БШ;
- выполняет построение индекса;
- загружает индекс в память и выполняет сравнение.

VL-LP5-I содержит следующие сервисы:

- Index Manager - управляет задачами на построение индекса и координирует сервис Indexer;
- Indexer - строит индексы на основе списка БШ;
- Indexed Matcher - выполняет аппроксимированное сравнение ближайших соседей (БШ) с помощью индекса.

Для работы с модулем требуется сервис Python Matcher Proxy с встроенным плагином сравнения, позволяющим определять к какому сервису будут направляться запросы из сервиса LUNA API - к сервису Python Matcher или к сервису Indexed Matcher. Также для работы сервисов Index Manager и Indexed Matcher требуется наличие БД Redis.

Все сервисы VL-LP5-I масштабируются, что означает возможность использования нескольких экземпляров.

4.1 Индекс

Индекс - коллекция предоставленного пользователем набора БШ, развернутых вместе для выполнения аппроксимированного сравнения. Он строится как граф зависимостей, вершинами которого являются БШ. Поиск зависимостей (биометрических шаблонов) в этом графе выполняется при перемещении по его вершинам (см. раздел “Процесс сравнения” ниже).

Построение индекса требует много ресурсов в течение длительного времени и является достаточно медленным процессом, поэтому нужно грамотно выставлять настройку периода автоматического перестроения индекса при появлении изменений в списке (см. ниже).

Размер списка с БШ определяет соотношение скорости и точности при построении индекса и поиске. Более высокие значения приводят к более точному, но более медленному поиску. Для настройки этих параметров следует использовать настройки “ef_construction” секции “LIM_MANAGER_INDEXING” и “ef_search” секции “LIM_MATCHING”.

4.1.1 Связь индекса с версией биометрического шаблона

VisionLabs LP5 Index учитывает изменение версии биометрических шаблонов лиц. Сервис Indexer выполняет построение индекса из биометрических шаблонов версии, указанной в настройке “DEFAULT_FACE_DESCRIPTOR_VERSION” сервиса Index Manager”. Сервис Index Manager автоматически перестраивает индекс, если в нем не содержится информация о версиях биометрических шаблонов. Сервис Indexed Matcher загружает только те индексы, которые содержат биометрические шаблоны версии, указанной в настройке “DEFAULT_FACE_DESCRIPTOR_VERSION” сервиса Index Manager”.

4.1.2 Структура индекса

Индекс состоит из следующих файлов:

- файла meta.json, который содержит метаинформацию об индексе, в том числе о версии биометрического шаблона и о том, какие объекты индексируются;
- файла index.dat, который содержит двоичные данные индекса;
- файла ids.dat, который содержит упорядоченный список идентификаторов объектов в индексе.

Каждый индекс имеет уникальное имя, которое используется в качестве имени ключа/папки.

Стандартная директория для сохранения индекса указывается для каждого сервиса VL-LP5-I в настройке “index_storage_local” секции “OTHER” сервиса Configurator. Обратите внимание, что для всех трех сервисов директория должна быть одинаковой.

4.1.3 Процесс создания задачи на построение индекса

Индексация набора БШ осуществляется посредством постановки задач на индексирование в очередь. Такие задачи создаются в сервисе Index Manager. Существует два типа задач построения индекса - разовый и фоновый.

Разовый тип позволяет единоразово создать задачу на построение индекса с помощью HTTP-запроса “create task” к сервису Index Manager. В теле запроса необходимо указать требуемый “list_id”.

Фоновый тип позволяет создавать задачи на построение индекса в фоновом режиме, где:

- набор списков явно задается в настройке “indexing_list” секции “LIM_MANAGER_INDEXING” сервиса Configurator;
- динамически индексируются все существующие в LP списки, у которых количество лиц превышает количество, заданное в настройке “min_indexing_list_size” секции “LIM_MANAGER_INDEXING” сервиса Configurator. При этом значение настройки “indexing_list” должно принимать значение “dynamic”. Значение по умолчанию - 50000 лиц.

При использовании фонового типа сервис Index Manager отслеживает изменение количества лиц в списках, взаимодействуя с сервисом Faces. Если количество лиц изменилось, то во внутреннюю очередь будет отправлена новая задача.

Одна задача обрабатывает только один список.

4.1.4 Процесс создания индекса

Ниже описан процесс создания индекса:

1. Для начала индексации сервис Index Manager отправляет запрос сервису Indexer с необходимыми параметрами - “list_id” и “task_id”. Сервис Indexer преобразует данные параметры в “label” (далее - метка для сравнения) и “index_id” соответственно.
2. При получении запроса на индексацию сервис Indexer запускает отдельный процесс для индексации. На данном этапе Indexer устанавливает свой статус на “indexing”.
3. Когда процесс индексации запущен, сервис Indexer извлекает БШ из сервиса Faces. Выборка выполняется партиями по 1000 элементов.
4. После того, как все БШ были извлечены и загружены в память, сервис Indexer начинает построение индекса. Строится ориентированный граф зависимостей БШ (см. раздел “Индекс”).

5. Затем, когда индексация завершена, сам индекс сохраняется в хранилище (файловую систему). В хранилище индекс представляет собой каталог, содержащий некоторые файлы (см. раздел “Структура индекса”).
6. После успешного сохранения индекса процесс индексации останавливается. На данном этапе Indexer устанавливает свой статус на “success”. Если же процесс индексации закончился ошибкой, то Indexer установит свой статус на “error”.

Информацию о сохраненных индексах можно получить с помощью запросов “get indexes” или “get most relevant indexes” к сервису Index Manager.

Посмотреть статус сервиса Indexer можно с помощью запроса “get tasks” к сервису Index Manager.

Через некоторое время после сохранения индексов все запущенные экземпляры Indexed Matcher автоматически (повторно) загружают эти индексы в память. После загрузки индексов в память можно отправлять запросы на сравнение индексированных наборов БШ с указанной меткой для сравнения.

4.2 Сравнение

Сервис Indexed Matcher загружает более релевантные индексы из хранилища и обрабатывает запросы на сравнение. Поскольку хранилище индексов может содержать несколько версий индексов с определенной меткой для сравнения, сервис Indexed Matcher всегда стремится выполнить сравнение с более новой (т.е. более релевантной) версией.

Индекс устаревает, как только в LUNA PLATFORM 5 создаются или удаляются БШ.

Для синхронизации индексов в памяти сервиса Indexed Matcher с хранилищем индексов предназначен периодический фоновый процесс, называемый перезагрузкой индекса (см. подробную информацию в разделе “Перезагрузка индекса”).

Если в исходный список были внесены какие-то изменения, то сервис Indexed Matcher обновляет соответствующие индексы у себя в памяти, путем постепенного добавления небольшого количества новых биометрических шаблонов в кэшированный индекс (см. подробную информацию в разделе “Обновление кэшированного индекса”).

4.2.1 Запросы на сравнение

Запросы на сравнение поступают из сервиса API в сервис Python Matcher Proxy, который использует плагин сравнения для перенаправления запроса в сервис Indexed Matcher. Сервис Indexed Matcher принимает запросы на сравнение через потоки Redis, выполняет сравнение и отправляет результат сравнения в канал Redis, откуда результат перенаправляется в сервис Python Matcher Proxy, а затем в сервис API.

Для запросов для каждой соответствующей метки для сравнения существует поток с именем метки. Несколько запущенных экземпляров Indexed Matcher с загруженным индексом являются группой потребителей для этого потока.

4.2.2 Процесс сравнения

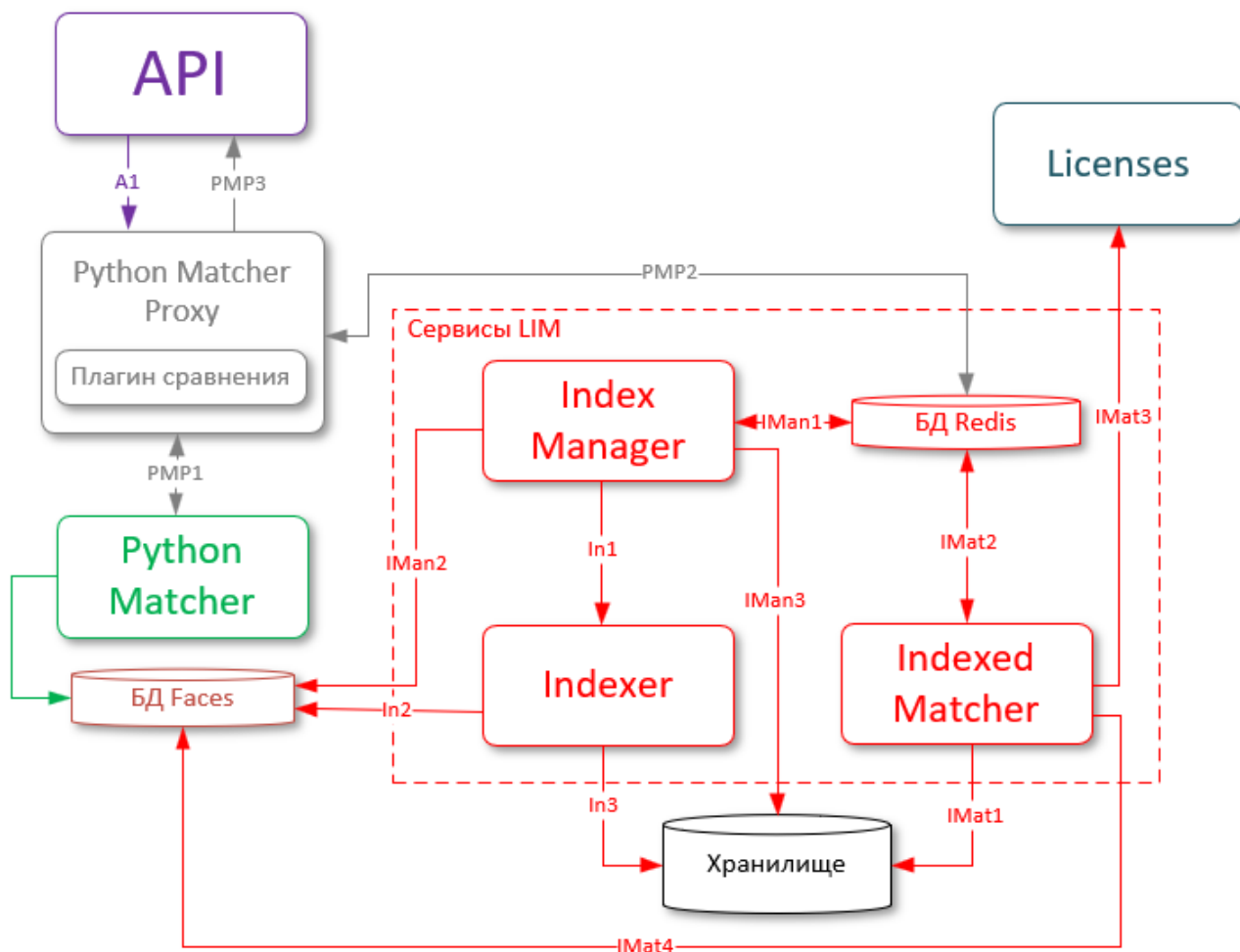
Сервис Indexed Matcher перемещается по вершинам графа зависимостей (индекса).

После перехода к первой вершине входящий БШ сравнивается со всеми вершинами, связанными с текущей вершиной. Когда найдена наиболее похожая вершина, выполняется следующее сравнение с вершинами, связанными с ней. После нескольких итераций находится наиболее похожая вершина (т.е. БШ с наибольшим показателем сходства). Количество операций при таком поиске значительно сокращается, что увеличивает производительность поиска в сотни раз.

5 Взаимодействие сервисов

Ниже приведена схема взаимодействия сервисов модуля индексирования.

Перед началом работы с VL-LP5-I пользователь должен создать список и прикрепить к нему необходимое количество лиц.

*Взаимодействие сервисов*

Перед тем как отправить запрос на сравнение из сервиса API, пользователь должен создать индекс, который создается с помощью создания задачи на его построение. Задача создается в сервисе Index Manager и может быть разового типа или фонового типа.

После создания, задача отправляется в очередь в БД Redis (**IMan1**).

Кроме очереди, БД Redis выступает хранилищем для всех задач на построение индексов. Там же используется механизм RedLock, обеспечивающий работу нескольких экземпляров сервиса Index Manager.

Сервис Index Manager взаимодействует с БД Faces для отслеживания изменений в списках (**IMan2**) если была создана задача фонового типа.

Далее Index Manager отправляет запрос в сервис Indexer на построение индекса (**In1**). После получения запроса сервис Indexer извлекает БШ из БД Faces (**In2**) и начинает построение индекса. Построенный индекс сохраняется в хранилище (**In3**).

После успешного создания индекса, статус задачи изменяется на “success”. Перед отправкой запроса на сравнение нужно проверить статус индекса с помощью запроса “get tasks” к сервису Index Manager. При отправке пользовательских запросов к сервису Index Manager на получение информации об индексах, сервис будет взаимодействовать с хранилищем индексов (**IMan3**).

Indexed Matcher кэширует индекс (IMat1), загружает метку для сравнения (содержит “list_id” кэшированного индекса) в БД Redis (IMat2) и слушает поток Redis пока там не появится запрос на сравнение.

Сервис Indexed Matcher постоянно следит за изменениями в списках с лицами, взаимодействуя с базой данных Faces (**IMat4**). В случае внесения новых изменений в список, сервис Indexed Matcher постепенно добавляет новые биометрические шаблоны в соответствующий кэшированный индекс. При этом индекс, находящийся в хранилище, остается неизменным до его перестроения. При необходимости данный функционал можно отключить.

После того, как индекс создан и загружен в память сервиса Indexed Matcher, пользователь выполняет запрос на сравнение в сервисе API, прикрепляя изображение эталона. Данный запрос перенаправляется в сервис Python Matcher Proxy (A1), где плагин сравнения генерирует запрос определенного формата, содержащий метку для сравнения и биометрический шаблон эталона, и определяет загружена ли метка для сравнения в БД Redis сервисом Indexed Matcher. Далее выполняется выбор сервиса для выполнения сравнения: - если метка для сравнения не загружена в БД Redis, то запрос направляется в Python Matcher (PMP1). В таком случае сравнение будет выполнено с помощью классического метода перебора биометрических шаблонов. - если метка для сравнения загружена в БД Redis, то запрос направляется в виде сообщения в поток Redis. Сервис Indexed Matcher считывает сообщение из потока Redis (IMat2). После этого сервис Indexed Matcher проверяет наличие лицензии LUNA PLATFORM 5 на возможность выполнения сравнения, взаимодействуя с сервисом Licenses (IMat3).

После окончания сравнения сервис Indexed Matcher записывает результаты сравнения в канал Redis (IMat2). Плагин сравнения сервиса Python Matcher Proxy считывает результаты сравнения и возвращает их пользователю в сервисе API (PMP2).