



Поддержка языка Python в системе автоматизации разработки Paddle

Автор: Смирнов Олег Евгеньевич, 18.Б11-мм

Научный руководитель: к. т. н., доцент Брыксин Т.А.

Консультант: программист ООО “Интеллиджей Лабс” Танков В.Д.

Рецензент: программист ООО “Интеллиджей Лабс” Сазанович В.В.

Санкт-Петербургский Государственный Университет
Кафедра системного программирования

4 июня 2022 г.

Введение

- Автоматизация сборки программных проектов — сложный процесс
- Системы сборки общего назначения требуют много усилий для конфигурации даже типовых задач разработки на конкретном языке
- Проекты на Python — не исключение
 - Нужна поддержка монорепозиториев
 - Есть свои особенности разработки: виртуальные окружения, PyPi-репозитории, и т.д.
 - Нужна интеграция системы со средой разработки (Integrated Development Environment, IDE)
- В компании JetBrains был спроектирован прототип системы автоматизации разработки Paddle и соответствующий плагин для IDE PyCharm

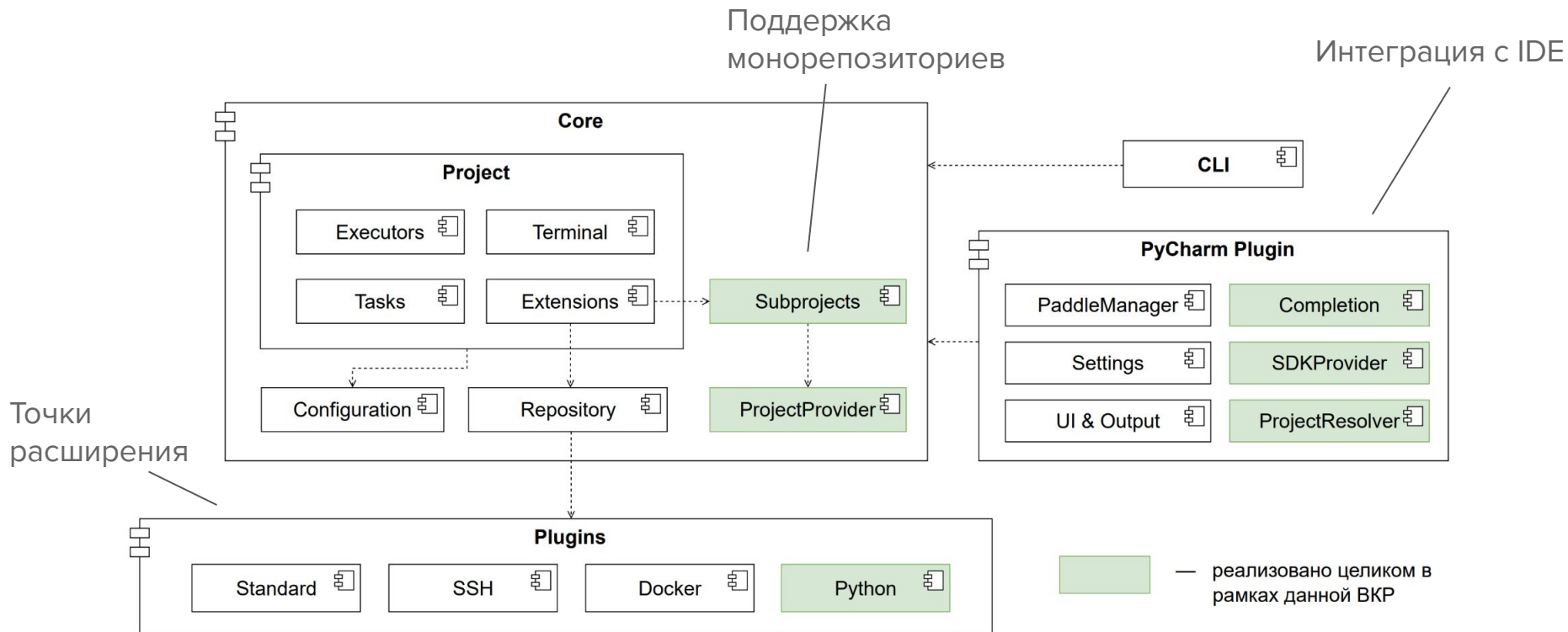
Цель и задачи

- Целью данной работы является расширение системы автоматизации разработки проектов Paddle для поддержки языка Python
- Задачи:
 - Провести обзор предметной области и существующих решений в сфере систем сборки проектов и автоматизации разработки на языке Python
 - Реализовать подсистему управления пакетами и виртуальными окружениями
 - Реализовать поддержку сборки монорепозитория с Python-проектами
 - Интегрировать реализованную функциональность в плагин для IDE PyCharm
 - Продемонстрировать возможности системы на нескольких существующих монорепозиториях с Python-проектами

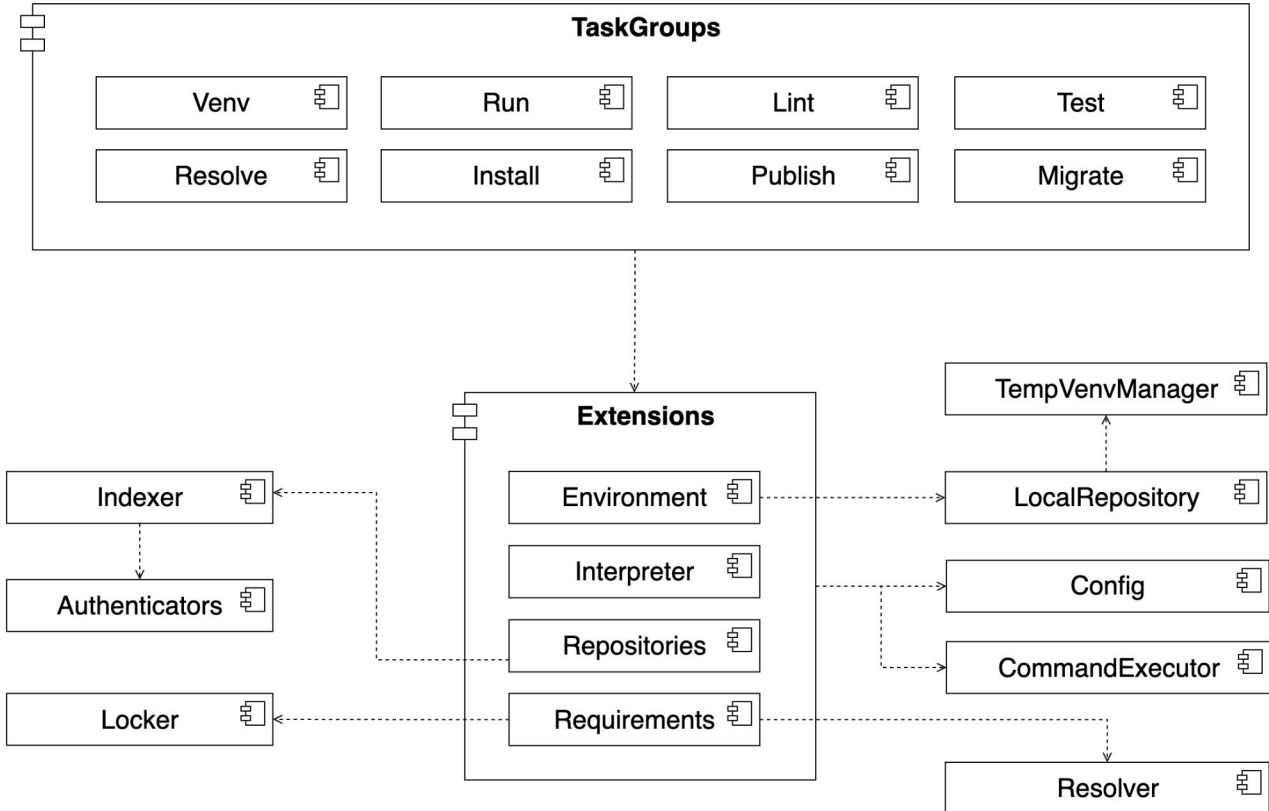
Обзор: системы автоматизации разработки

	Make	Bazel	Pants	Conda	Poetry	Paddle
Система сборки	✓	✓	✓	✗	✗	✓
Интеграция с IDE	✓	👎	✗	✓	✓	✓
Скорость сборки	✓	✓	✓	✗	✗	✗
Поддержка Python	✗	✓	✓	✓	✓	ВКР — ✓
Легкость конфигурации	✗	✗	✓	✗	✓	✓
Расширяемость	✗	✓	✓	✗	✗	✓
Оптимизация памяти при хранении виртуальных окружений	✗	✗	✗	✓	✗	ВКР — ✓

Архитектура системы Paddle



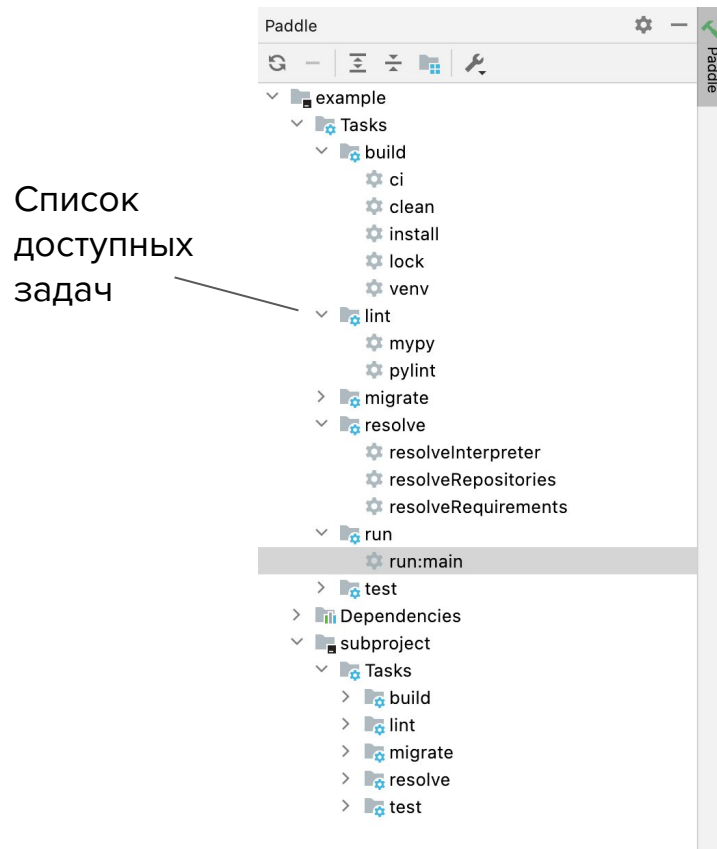
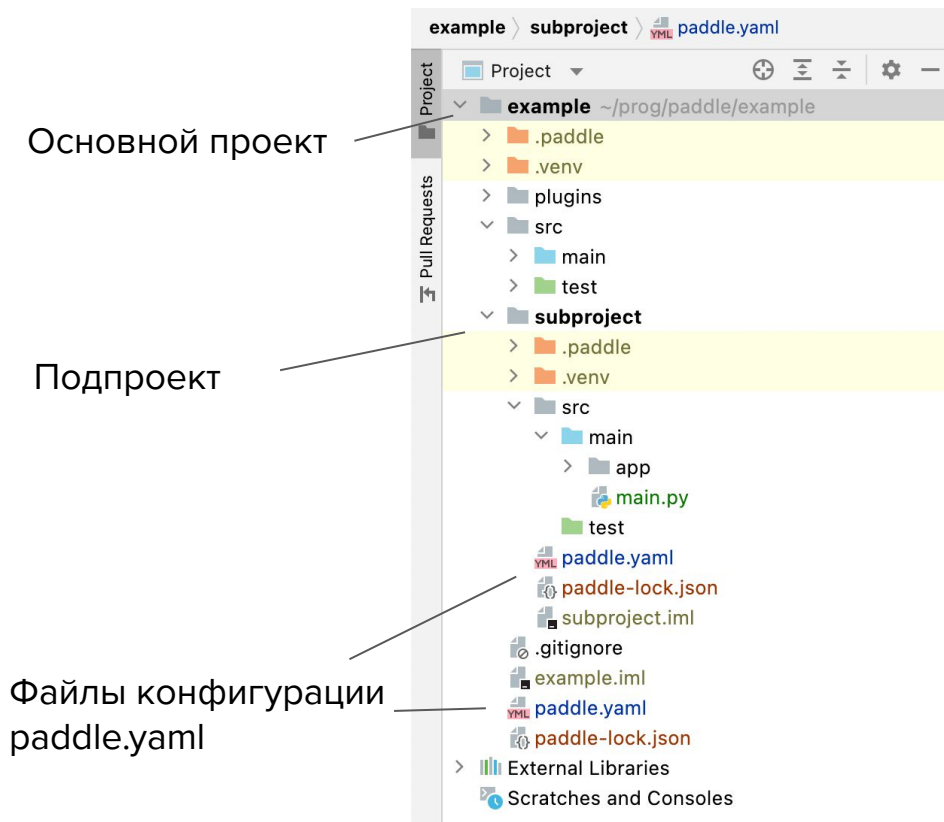
Архитектура Python-плагина



Особенности реализации Python-плагины

- Локальный репозиторий установленных пакетов (~/.paddle/cache)
 - Экономия дискового пространства за счёт символьных ссылок
- Установка пакетов из публичных и частных PyPi-репозиториях
 - Аутентификация с помощью keyring, .netrc, профилей пользователей
 - Построение индексов для имён и версий пакетов
- Поиск и установка произвольной версии интерпретатора Python
 - Поддержка Unix, MacOS
- Контроль целостности с помощью lock-файлов
- Встраивание в систему импортов Python и поддержка зависимостей на подпроекты

Примеры: структура проекта (1) и окно задач (2)



Примеры: файл конфигурации paddler.yaml (3)

Дескриптор проекта

Зависимости на подпроекты

Корневые директории

Виртуальное окружение

Удалённые репозитории пакетов

```
main.py x subproject/paddle.yaml x paddler.yaml x
1 descriptor:
2   name: example
3   version: 0.1.0
4
5 subprojects:
6   - subproject
7
8 roots:
9   sources:
10    - src/main
11   tests:
12    - src/test
13
14 environment:
15   path: .venv
16   python: 3.9
17
18 repositories:
19   - name: grazie-ml
20     url: https://packages.jetbrains.team/pypi/p/grazi/grazie-ml/simple
21     auth:
22       type: netrc
23       username: Oleg.Smirnov
24
```

Примеры: журнал исполняемых задач (4)

Создание lock-файла

```
Run: /Users/Oleg.Smirnov/prog/paddle/example [lock] x
✓ /Users/Oleg.Smirn 1 min, 8 sec, 90 ms 19:38:23: Executing 'lock'...

> Task :example:resolveInterpreter: EXECUTE
[INFO] Resolving interpreter...
[INFO] Found cached installation of Python-3.9.12: /Users/Oleg.Smirnov/.paddle/interpreters/3.9.12/Pytho
[INFO] Finished: 13ms
> Task :example:resolveInterpreter: DONE
> Task :example:venv: EXECUTE
[INFO] Creating virtual environment...
[INFO] /Users/Oleg.Smirnov/prog/paddle/example$ /Users/Oleg.Smirnov/.paddle/interpreters/3.9.12/Python-3
[INFO] Finished: 30115ms
> Task :example:venv: DONE
> Task :example:resolveRepositories: EXECUTE
[INFO] Resolving and indexing repositories...
[INFO] Finished: 770ms
> Task :example:resolveRepositories: DONE
```

Топологический порядок обхода графа задач

Апробация

- Для проведения апробации было выбрано 3 внутренних проекта по машинному обучению и анализу естественного языка команды Grazie в компании JetBrains
 - В монорепозитории проекта grazie-cloud с помощью локального репозитория пакетов удалось сократить использование дискового пространства с **32,1** Гб (при использовании venv) до **4,1** Гб
 - В проектах grazie-crowd и grazie-ml использовалось единое виртуальное окружение пакетов для всех подпроектов, что по словам разработчиков вызывало трудности при миграции отдельных зависимостей
- Были протестированы основные различные сценарии использования системы автоматизации Paddle и выделены направления дальнейшей работы

Результаты

- Проведен обзор систем сборки общего назначения и основных инструментов разработки на языке Python, рассмотрены системы управления пакетами и виртуальными окружениями
- Реализована подсистема для поддержки языка Python в системе автоматизации разработки Paddle:
 - Реализованы расширения для управления пакетами и виртуальными окружениями, поддержка удалённых PyPi-репозитория и lock-файлов
 - Реализовано расширение для работы с монорепозиториями Python-проектов
- Выполнена интеграция Python-расширения системы автоматизации Paddle в соответствующий плагин для IDE PyCharm
- Проведены эксперименты по использованию системы автоматизации Paddle в трёх проектах компании JetBrains в области машинного обучения, рассмотрены основные сценарии применения системы и выделены направления дальнейшей работы (поддержка императивных расширений на Python, кэши на уровне организации, распределённое исполнение задач)