УДК 004.41

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ И ОТКАЗОУСТОЙЧИВОСТИ МОДУЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА КОДА OPEN SOURCE ADVISOR

Шиянов А.О. (ИТМО)
Научный руководитель – Гетманов А.Н.

Введение. Современная разработка программного обеспечения все чаще опирается на инструменты класса AI4SE (Artificial Intelligence for Software Engineering) для автоматизации рутинных задач. Одним из таких инструментов является Open Source Advisor (OSA) — мультиагентная система на базе больших языковых моделей (LLM) для аудита, документирования и улучшения репозиториев. Однако, по мере роста сложности анализируемых проектов, критическими факторами становятся производительность и надежность системы. Синхронная обработка файлов при взаимодействии с внешними API (LLM-провайдеры, GitHub) приводит к простоям вычислительных ресурсов, а отсутствие гранулярной обработки ошибок делает автоматизированные пайплайны неустойчивыми к сетевым сбоям. Целью данной работы является архитектурная оптимизация инструмента OSA для обеспечения масштабируемости и отказоустойчивости.

Основная часть. Для устранения узких мест производительности («бутылочного горлышка» ввода-вывода) была переработана архитектура модулей валидации кода. Вместо последовательной обработки файлов реализована асинхронная модель на базе библиотеки asyncio. Внедрен механизм конкурентной отправки запросов к LLM с использованием семафоров для соблюдения квот API (Rate Limits). Ресурсоемкие операции, такие как парсинг PDF-документов, вынесены в отдельные потоки (asyncio.to_thread) для предотвращения блокировок основного цикла событий (Event Loop).

Для повышения надежности (Reliability) разработан модуль классификации ошибок взаимодействия с системой контроля версий (Git). Реализован парсинг потока ошибок (stderr) и кодов ответа API, позволяющий системе различать критические сбои (ошибки авторизации 401, отсутствие доступа 404) и временные проблемы (лимиты запросов 429). Внедрена стратегия самовосстановления для случаев повреждения индекса Git при автоматических коммитах.

Дополнительно усовершенствован алгоритм генерации зависимостей проекта (requirements.txt). Разработан гибридный подход: инструмент pipreqs используется для сканирования импортов, а LLM выполняет интеллектуальное слияние полученного списка с существующим файлом зависимостей, сохраняя зафиксированные версии библиотек и очищая неиспользуемые.

Выводы. В результате проведенной оптимизации время полного цикла валидации крупных репозиториев (на примере фреймворка Scrapy, >100 файлов) сократилось в 9 раз (с 14 минут до 1.5 минут). Внедренная система обработки ошибок обеспечила стабильную работу агента в условиях нестабильной сети и ограничений API, а новый модуль работы с зависимостями позволил сохранить контекст версионирования при автоматическом обновлении проектов.

Список использованных источников:

1. Hou X. [et al.]. Large Language Models for Software Engineering: A Systematic Literature Review // ACM Transactions on Software Engineering and Methodology. – 2024. – Vol. 33, No. 5. – P. 1–32.
2. Xi Z. [et al.]. The Rise and Potential of Large Language Model Based Agents: A Survey // arXiv preprint arXiv:2309.07864. – 2023.
3. Fowler M. Python Concurrency with asyncio. – Shelter Island : Manning Publications, 2022. – 376 p.
4. Open Source Advisor (OSA) Repository [Электронный ресурс]. – Режим доступа: https://github.com/aimclub/OSA (дата обращения: 17.02.2026).