University of Hong Kong

AUTOFUZZ: AN LLM-DRIVEN FRAMEWORK FOR AUTONOMOUS FUZZING

BY MA TIN YU, MARTIN
SUPERVISED BY PROF CHEN, HO.

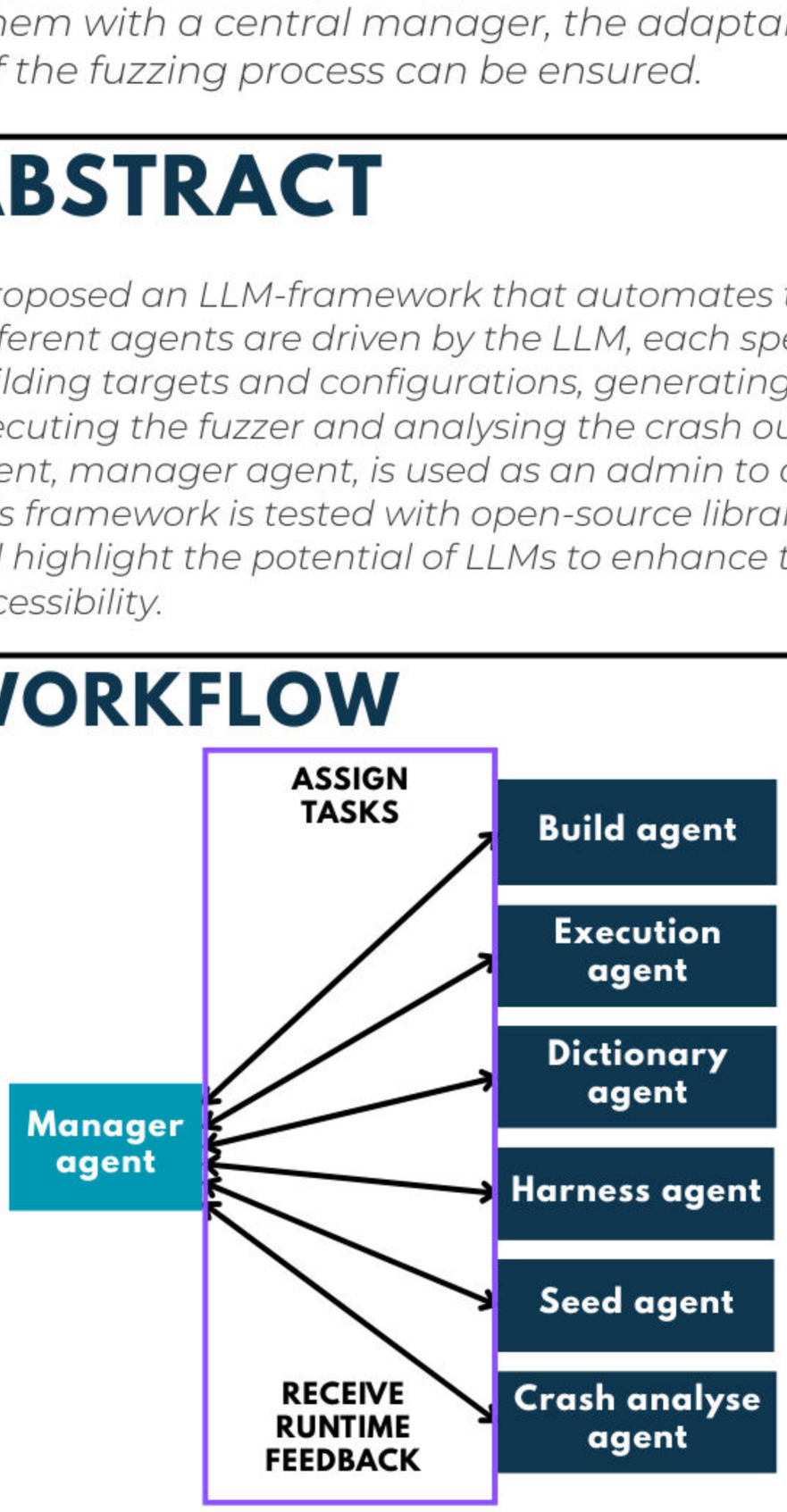
INTRODUCTION

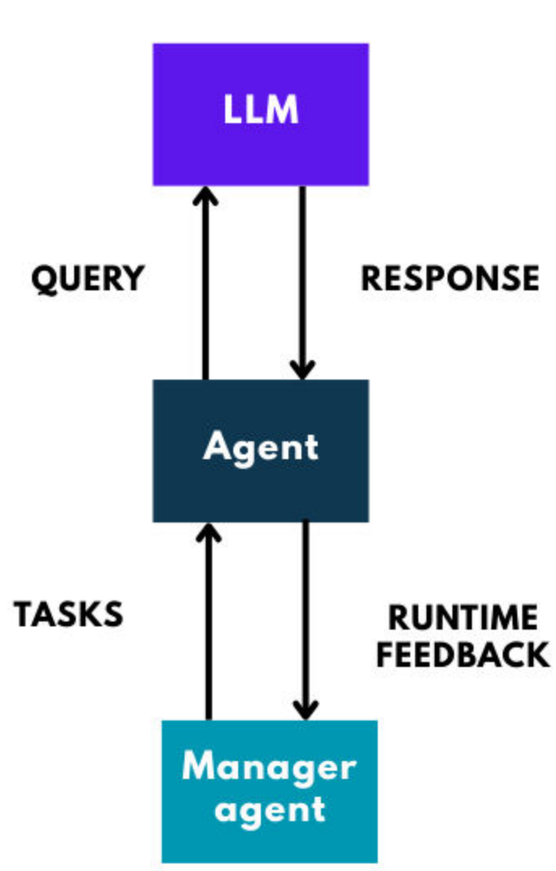
Fuzzing is a brand-new technique for finding programming bugs and security bugs from the software through executing the program with automated generated seeds. Traditionally, developers have to build up a pipeline for the whole fuzzing process, which are manually selecting the appropriate fuzzer, builiding files and program targets, creating harness, dictionary and seeds, analysing the crash output. These tasks have increase the complexitiy of the whole workflow, limiting the accessbility.
Therefore, this project aims to reduce the complexity of fuzzing workflow by introducing the concept of Lareg Language Model (LLM) Agents. By dividing the pipeline into different sub-tasks and leading them with a central manager, the adaptability and the performance of the fuzzing process can be ensured.

ABSTRACT

I proposed an LLM-framework that automates the entire fuzzing workflow. Different agents are driven by the LLM, each specialized in fuzzing task : building targets and configurations, generating harness, seeds and dictionary, executing the fuzzer and analysing the crash output. Particularly, an extra agent, manager agent, is used as an admin to orchestrated those agents. This framework is tested with open-source libraries, such as jHead. Our result will highlight the potential of LLMs to enhance the fuzzing efficiency and accessibility.

WORKFLOW





RESULTS & DISCUSSION

We evaluated our automated fuzzing framework on the jhead library, which is an image metadata processing library. The pipeline built using agents are stable, resulting in discovering at least one crash.

However, our findings also revealed areas for improvement. The performance of the LLM agents significantly depend on the structure of the prompts, meaning that general prompts would lead to incorrect execution.

Moreover, the agents also rely on the capabilities of the LLM, where high quality model, such as Google Gemini would give a better result than Quwen model.

FUTURE WORK

To further enhane the efficiency of the framwork, an opimization in prompt construction are needed by introducing a more task-oriented prompt structure.