AWC: Agent Workload Characterization Tool

Project description

Large Language Models (LLMs) are increasingly being used as autonomous agents for tasks such as software development, research, and personal assistance. Unlike conventional chatbot workloads, agentic applications repeatedly invoke an LLM, execute external tools, access files, compile code, search for information, and interact with operating-system or application environments. Their performance therefore depends not only on LLM inference, but also on CPU execution, memory usage, storage and network activity, and interactions between different system components. Recent benchmarks primarily evaluate either the task-solving capability of agents or the performance of individual AI models. Although several recent studies have begun to examine agent execution from a systems perspective, reproducing and comparing complete agentic workloads remains difficult because an agent may follow a different sequence of actions each time it runs. This makes it challenging to determine which workload characteristics are stable and which system resources become bottlenecks. This project will investigate the system-level characteristics of a small set of representative, tool-using agent workloads, with an initial focus on applications such as coding or research agents. The student will construct a reproducible experimental environment and collect execution traces containing information such as LLM requests, tool calls, execution dependencies, token usage, tool duration, and CPU, memory and I/O activity. The study will analyse how these characteristics vary across repeated executions and across selected tasks. Based on the collected traces, the student will develop a lightweight method for converting selected agent executions into repeatable replay workloads. The replayed workloads will be compared with the original executions to evaluate how well they preserve important characteristics such as LLM–tool execution balance, context growth, execution time and system-resource pressure. The project will produce a small trace dataset, a reproducible profiling and replay tool, and an experimental characterization of selected agentic workloads. All core experiments will be designed to run on a standard PC using open-source software and appropriately sized models. Evaluation on additional AI-PC or edge platforms may be included as an optional extension, subject to hardware availability.

Deliverables

- A literature review covering LLM inference, agentic workflow, tool use and system profiling.
- A reproducible experimental setup for running and profiling LLM inference and agentic workflow.
- A tool that can extract agentic executions into a repeatable, stable, standard testbench dataset.
- A replay harness that can execute the extracted standard testbench, e.g. LangChain, LangGraph.
- A performance characterisation analysis that summarises the agentic workflow into a set of specific performance pressures.
- Source code, experimental scripts, collected results, and documentation.
- A final dissertation/report presenting the methodology, experimental results, analysis, and conclusions.

Suggested timeline

- **Months 1–2:** Literature review and reproducible experimental setup for running and profiling LLM inference and agentic workflows.
- **Months 2–3:** Trace collection and initial characterisation of selected agent workloads.
- **Months 3–4:** Extraction tool for converting agent executions into a standard testbench dataset and replay harness.
- **Months 4–5:** Comparison of replayed and original executions and performance characterisation analysis.
- **Months 5–6:** Final evaluation and dissertation writing.

Equipment/Software required to carry out this project

- Standard personal computer or laptop: existing equipment, £0 additional cost.
- Python: open source, £0.
- PyTorch: open source, £0.
- Hugging Face Transformers: open source, £0.
- Open-source LLMs, such as Llama-family compatible models, Qwen, Gemma, or TinyLlama models: £0.
- System profiling tools, such as Python profiling libraries, psutil, Linux monitoring utilities, PyTorch Profiler, or equivalent tools: £0.
- Git/GitHub: free educational/open-source use, £0.

Total additional equipment/software cost: £0.

The experiments will be designed so that they can be completed using either a normal PC with appropriately sized models or the available Spark computing resources.

Skills developed in carrying out this project

- Understanding of Large Language Model inference and Transformer architectures.
- Understanding of computer architecture, von-Neumann architecture, GPU and memory systems.
- Performance profiling and benchmarking.
- Experimental methodology and workload characterisation.
- Python and PyTorch programming.
- Use of modern LLM frameworks and open-source models.
- Data analysis and visualisation.
- Scientific writing and presentation of experimental results.

Skills required to carry out this project

- Basic programming experience, preferably in Python.
- Basic understanding of computer systems or computer architecture.
- Basic knowledge of machine learning is desirable but not essential.
- Basic knowledge and experience with LLM Agents, e.g. Claude Code, Codex, OpenClaw, Hermes Agent and so on.
- Ability to work with Linux command-line tools is desirable.
- Previous experience with LLMs, GPU programming, or specialised hardware is desired but not required.