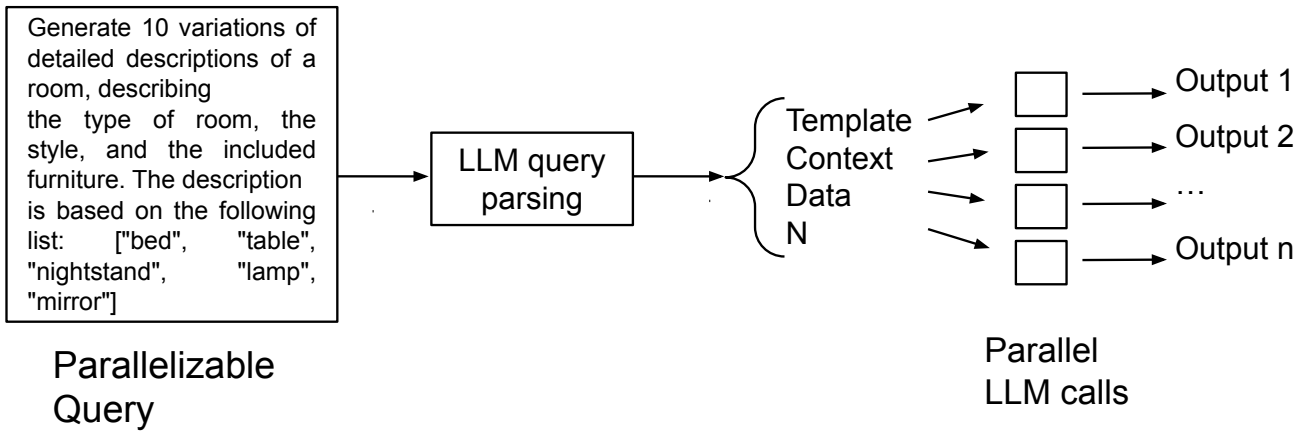
Extracting Parallelism from Large Language Model Queries

Steven Kolawole [1] Keshav Santhanam [2] Virginia Smith [1] Pratiksha Thaker [1]

[1]Carnegie Mellon University [2]Stanford University



Motivation

- LLM serving systems often treat queries as black boxes, missing chances to optimize tasks embedded within them.
- Common decomposable subtasks (e.g., generating multiple outputs or answering multiple questions) can drastically reduce latency (and potentially improve quality) if handled in parallel.

Challenges

- Identifying parallelizable subtasks in raw natural language queries.
- Converting queries into structured formats without user intervention.

Future Work

- Improve robust handling of queries requiring specific output formats or expecting independent content across parallel execution.
- Explore post-processing (i.e., an extra LLM step) for complex queries in need of assembling parallel outputs into required formats or filtering redundancies.

Method & Implementation

Approach: Identify queries with decomposable subtasks (e.g., *repeated generation*, *reading comprehension*, *keyword extraction*) from **LMSYS-chat-1M-dataset**. Built a prototype system with C++ that:

1. Uses LLMs to extract parallel structure from raw queries.
2. Parses queries into structured schemas for parallel execution (e.g., JSON).
3. Executes subtasks concurrently using data-parallel LLM calls.

Performance Gains

Up to **5.7× speedup** for parallelized execution compared to serial execution, with significant latency reductions.

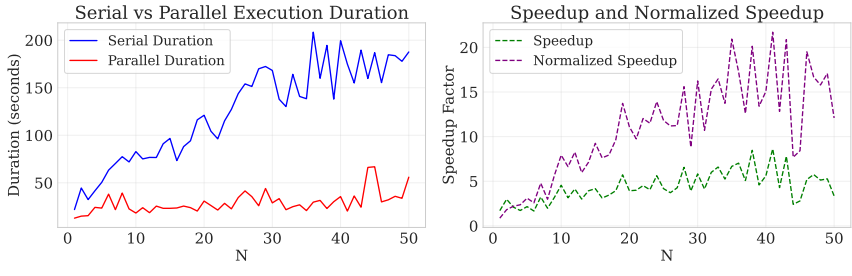
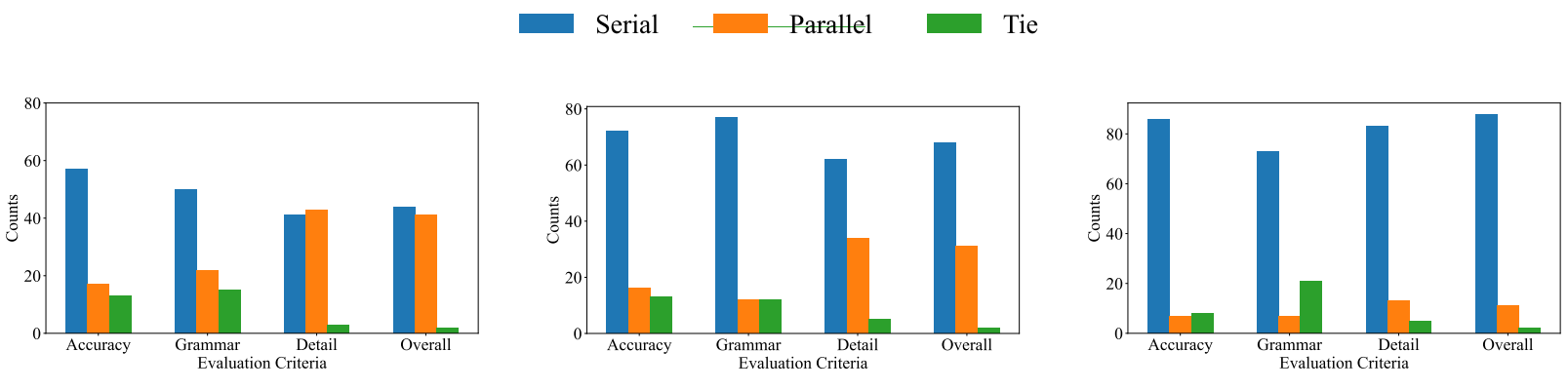


Figure 1. Scaling parallelization: varying n

Quality Comparison

An LLM judge (GPT-4o) was used to judge the two versions of the generations according to their *accuracy*, *grammar*, and *specificity*, as well as *an overall preference*.



Reading comprehension | Keyword extraction | Repeated generation