HW2

Tguo67

Question(s):

In this homework, read the paper of SGLang, answer the following questions.

1. How does RadixAttention work? Use an example different from the one in the paper to explain.
2. What benchmarks does the paper use for evaluating SGLang? What kinds of workloads do they represent? You can discuss several benchmarks together if they represent the same kind of workload. Hint: check the `benchmark` directory of SGLang source code.

Related links:
SGLangPaper:
https://proceedings.neurips.cc/paper_files/paper/2024/file/724be4472168f31ba1c9ac630f15dec8-Paper-Conference.pdfLinks to an external site.
SGLang Git Repo:
https://github.com/sgl-project/sglang

Answers:

1. **RadixAttention Basic concept:**
   Keep all already-computed KV tensors in a *radix tree* keyed by token prefixes; schedule new requests so they hit the longest shared prefixes first.
   Example:
   If the inputs are "Translate tagline to Japanese + Chinese",
   a) The long system prompt (same for all products) becomes a single long edge near the root. Its KV pages are reused *across the entire batch*.
   b) The fixed JSON schema tokens (regex-like structure) are another shared edge. Every product reuses those KV pages; for the actual constrained decoding, SGLang's compressed FSM can emit multiple tokens per step, speeding decoding itself.
   c) The interpreter sends prefix hints; the runtime inserts a node for the common prefix and then *forks the childrens.*
   d) After the task was done, Leaf-first LRU was performed, if memory fills, it drops old *leaf* branches (e.g., completed Tagline nodes) but preserves their ancestors (system/schema), maximizing reuse for upcoming products.
2. These are the benchmarks the paper uses:
   a) MMLU and HellaSwag
   b) ReAct agents, Generative agents
   c) Tree-of-Thought, Skeleton-of-Thought
   d) LLM Judge with Branch-Solve-Merge
   e) JSON decoding with a regex-specified schema
   f) Multi-turn chat (short vs. long outputs)
   g) DSPy RAG (official example)