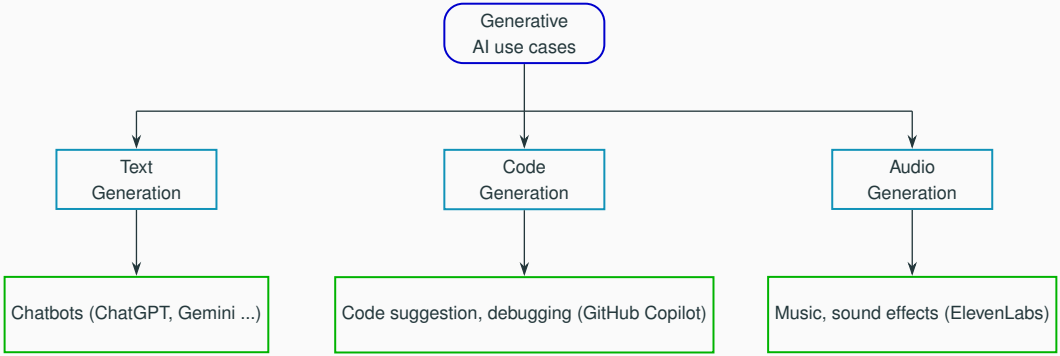
Deferred prefill for throughput maximization in LLM inference

Moonmoon Mohanty, Gautham Bolar, Preetam Patil
UmaMaheswari Devi, Felix George, Pratibha Moogi
Parimal Parag

31-Mar-2025

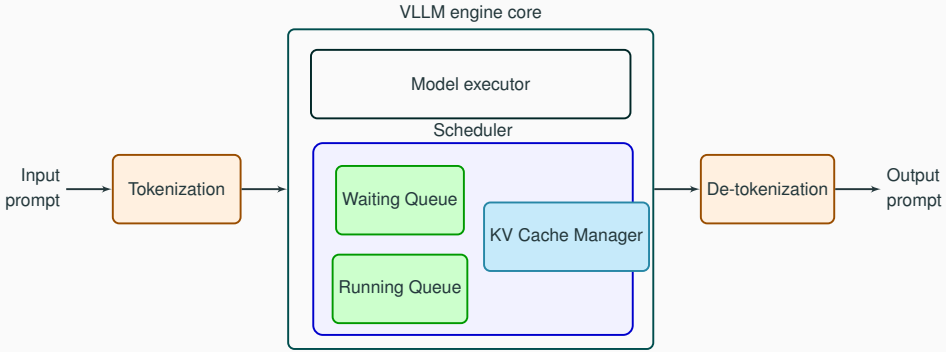
IISc and IBM Research

Motivation

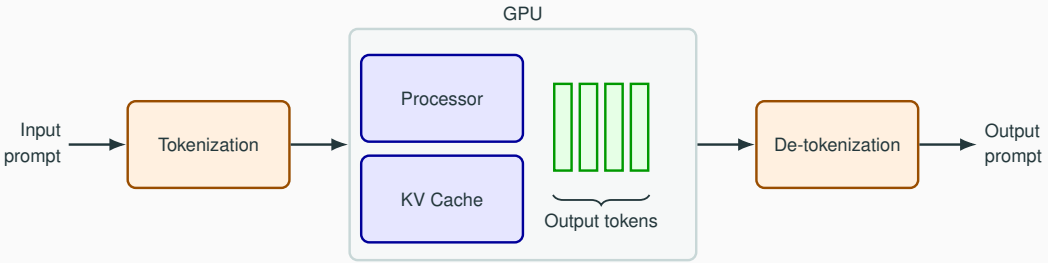


- **Cost, energy reduction**
- **Better user experience**

LLM inference server



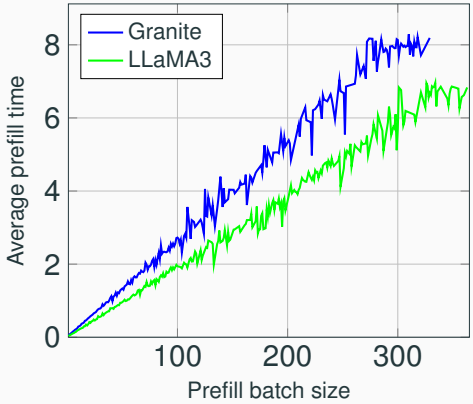
LLM inference phases



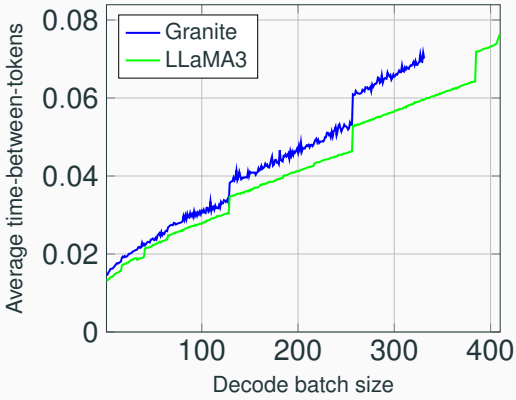
Compute memory interplay

- Prefill: Input tokens are *processed in parallel* and stored in KV cache
- Decode: Ouput tokens are *processed sequentially* from input tokens
- Output appended to input in KV cache and retrieved for each decode
- Batching of decodes to improve GPU core utilization

Prefill and Decode times(ShareGPT dataset)



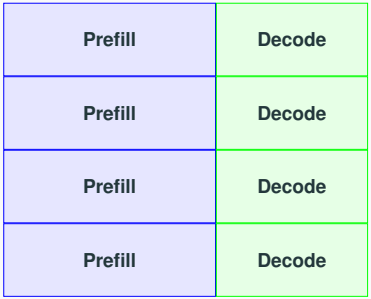
Prefill time



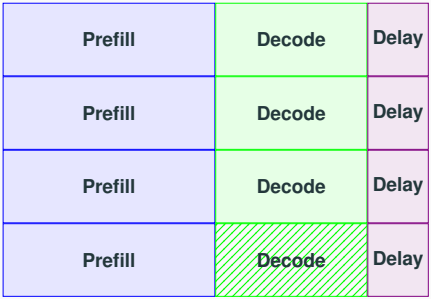
Decode time

Prompt processing policy

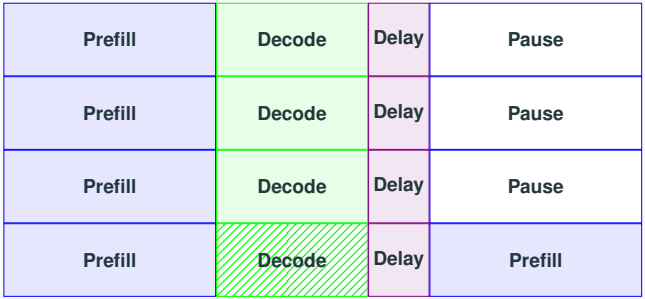
Prompt processing policy



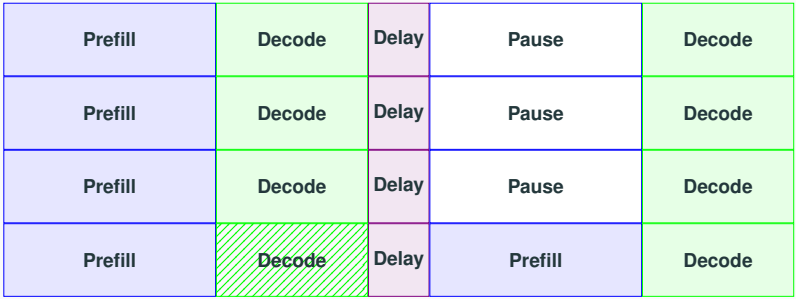
Prompt processing policy



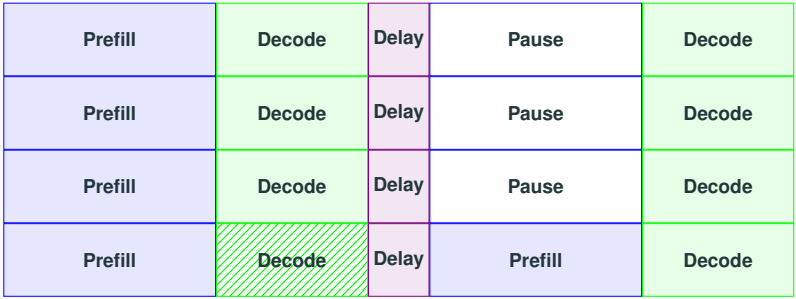
Prompt processing policy



Prompt processing policy



Prompt processing policy



Challenge: Frequent prefills = larger switching delay.
Solution: Prefill after multiple departures

Key question

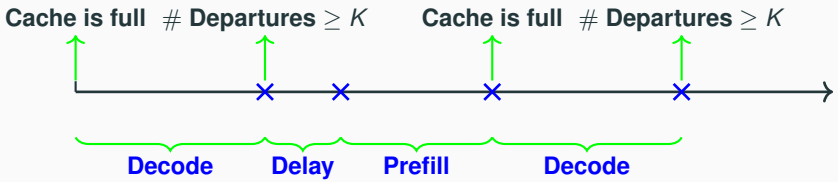


Figure 1: Timeline

Goal: Optimal departure threshold for average throughput maximization?

System Model

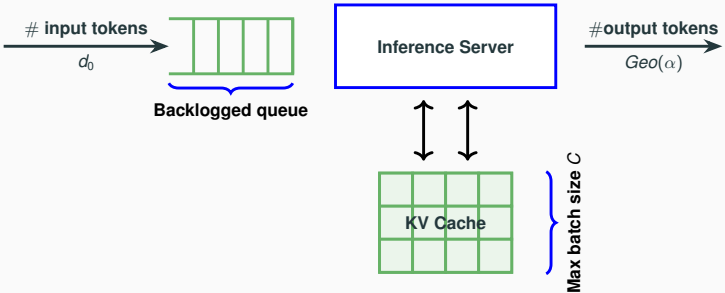
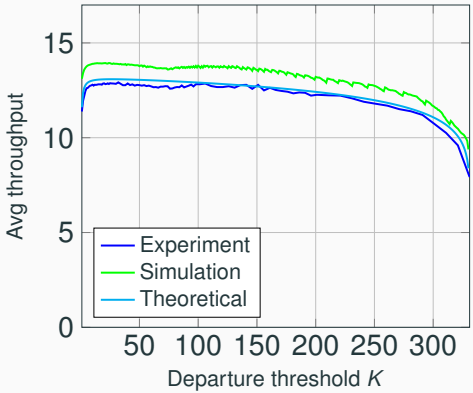


Figure 2: Timeline

$$\rho(K)^{-1} \approx \frac{1}{K}\left(c_p + c_d \frac{\ln(1-\frac{K}{C})}{\ln(1-\alpha)}\right) + \frac{t_d}{\alpha} + \frac{t_p d_0}{N}$$

K: departure threshold
c_p: scheduling overhead
t_p: i/p processing time
c_d: o/p token compute
t_d: memory slowdown

Experimental validation



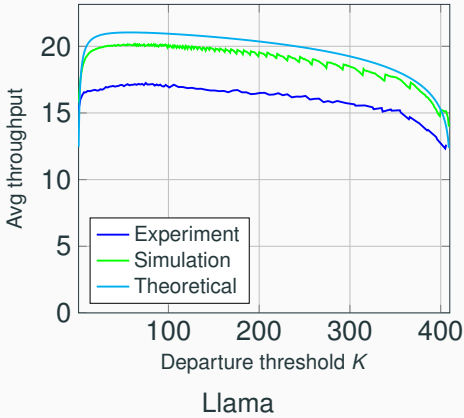


Figure 3: Performance metrics for different models with the ShareGPT dataset.

Conclusion

- Departure threshold-based scheduling algorithm.
- Analytical model for inference system, deduced closed form expression for throughput.
- Proved existence of optimal departure threshold that maximizes the system throughput.
- Characterization of LLM inference system for system parameters.
- Experimental validation with vLLM inference server and NVIDIA A100 GPU.
- **Key observation**: Proposed policy leads to 13% improvement in average throughput accompanied by 14% reduction in average prompt completion time.