FlexInfer: Flexible LLM Inference Using CPU Computation

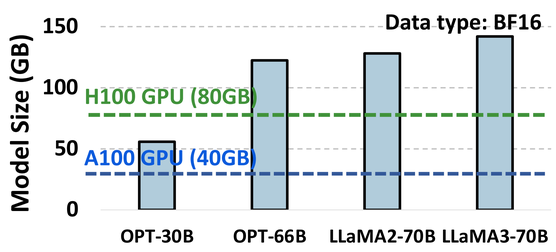
Seonjin Na[1], Geonhwa Jeong[2], Byung Hoon Ahn[3], Aaron Jezghani[1], Jeffery Young[1], Christopher J. Huges[4], Tushar Krishna[1], Hyesoon Kim[1]

[1]Georgia Institute of Technology, [2]Meta, [3]University of California San Diego, [4]Intel Labs

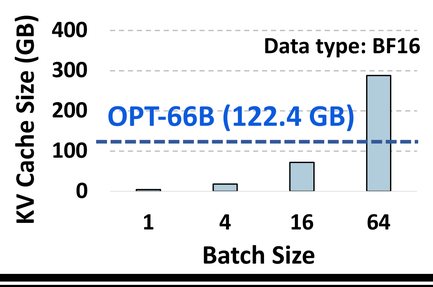
Problem

LLM Inference requires significant amount of memory

1. Size of LLMs



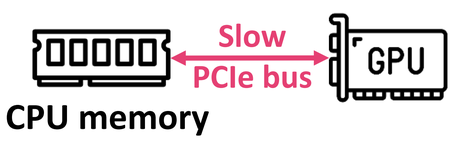
2. Size of KV Cache



Offloading-based LLM Inference suffer from PCIe data transfer overhead

Offloading-based LLM Inference

Offload data to CPU memory
(model weights, act., KV cache)



H100 with Offloading Exec. Breakdown

(Input length: 1024, output length: 32)

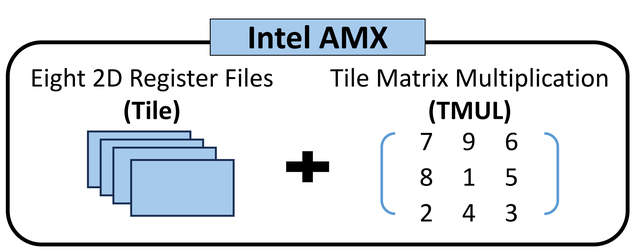
Batch Size	Data Load (PCIe)	GPU Compute
1	**95.8**	4.2
4	**94.4**	5.6
16	**89.3**	10.7

Significant data transfer overhead

Motivation

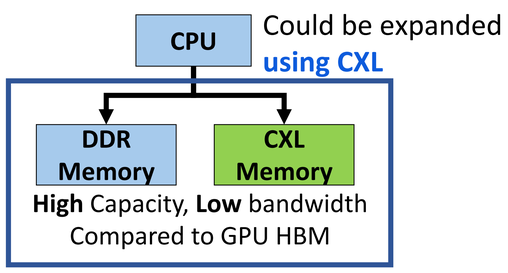
Key Opportunities in Recent CPUs

1. Dedicated GEMM Accelerator



Supported data type: **BF16,INT8**

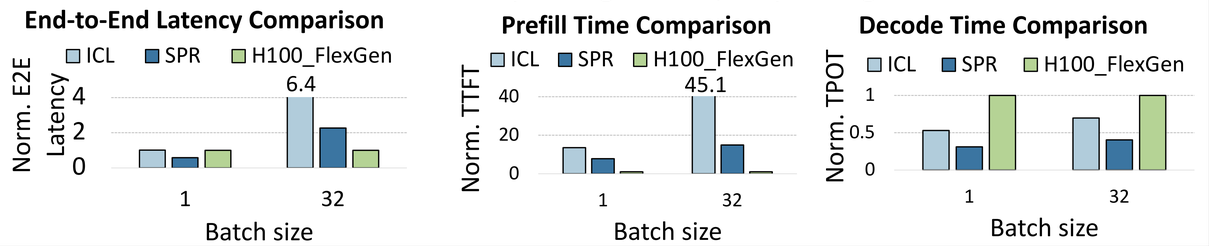
2. Larger Memory Capacity



CPU Computation vs GPU with Offloading

CPU Compute Limited in Prefill with longer seq. larger batch, Beneficial in Decode

Config: ICL CPU(without AMX) , SPR CPU (with AMX), H100 FlexGen
Model: OPT-66B Input length: 1024, output length: 32

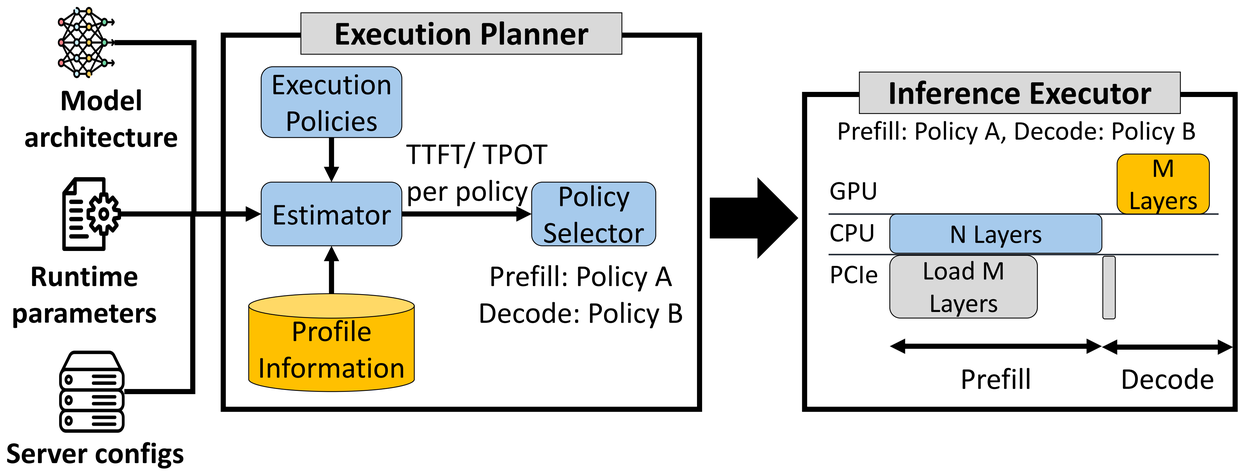


FlexInfer

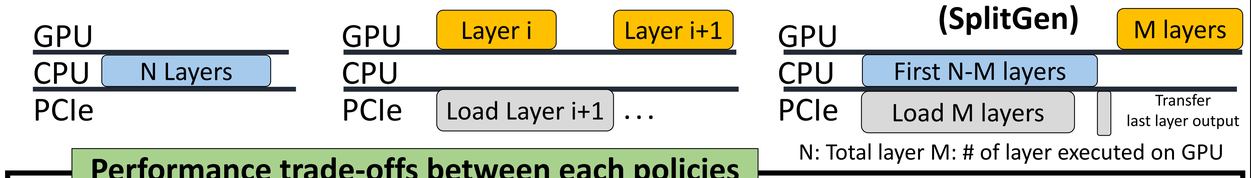
Overview of FlexInfer

Goal: FlexInfer dynamically select execution policies based on perf. estimator

Two key components: (1) Execution Planner (2) Inference Executor



Baseline Execution Policies in FlexInfer and Concrete Examples



Performance trade-offs between each policies

	1. CPU-Only	2. GPU with Offloading (FlexGen)	3. CPU-GPU Partitioning (SplitGen)
+	No need to transfer data	Efficiently process longer seq. & larger batch in prefill	Low data transfer volume, exploit both CPU and GPU
−	Limited compute throughput for longer seq. & larger batch	Slowdown due to data transfer	Slowdown due to CPU compute bottleneck

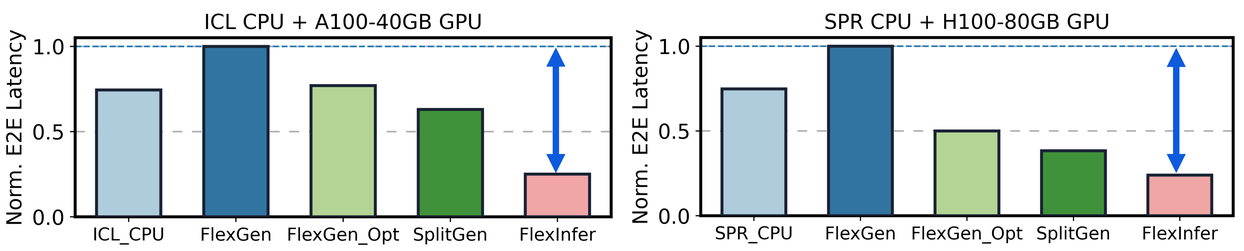
Sceneario1 Long input seq. Large batch
- Prefill → GPU with offloading
- Decode → SplitGen

Sceneario2 Short input seq. Smaller batch
- Prefill → CPU-Only
- Decode → CPU-Only

Evaluation Results and Summary

Performance Comparison Results (Input/Output length : 512/32)

FlexInfer reduces end-to-end latency by 75%, 76% on two different servers



Summary

Problem
- LLM inference requires significant memory exceeding recent GPUs memory capacity
- Offloading-based LLM inference suffers from **significant data transfer overhead**

Key observation & Idea
- CPU computation offers potential but struggles with compute-heavy prefill phase
- FlexInfer **dynamically select execution policies** based on HW configs/ runtime params

Evaluation results
- FlexInfer **reduces inference latency by 75%, 76%** on average across different servers