24春学期总结汇报

郭天宇

腾讯实习

- 时间 2024.1-6 地点 深圳
  - 解锁新的人生副本：一个人前往陌生的城市租房住

腾讯实习

- □ 时间 2024.1-6 地点 深圳
  - ➢ 解锁新的人生副本：一个人前往陌生的城市租房住
- □ 深圳是一个生活节奏很快的城市
  - ➢ 路上行人脚步很快，人行道的非机动车很快，日常的消费也很快

腾讯实习

- □ 时间 2024.1-6 地点 深圳
  - ➢ 解锁新的人生副本：一个人前往陌生的城市租房住
- □ 深圳是一个生活节奏很快的城市
  - ➢ 路上行人脚步很快，人行道的非机动车很快，日常的消费也很快
- □ 腾讯是一家很成熟的互联网企业
  - ➢ 入职培训(高压线)，组织架构，规范流程，工作环境

腾讯实习

- 时间 2024.1-6 地点 深圳
  - 解锁新的人生副本：一个人前往陌生的城市租房住
- 深圳是一个生活节奏很快的城市
  - 路上行人脚步很快，人行道的非机动车很快，日常的消费也很快
- 腾讯是一家很成熟的互联网企业
  - 入职培训(高压线)，组织架构，规范流程，工作环境
- 不同年龄段的人关注点相差很大
  - 买房，结婚，带娃

腾讯实习

- □ 时间 2024.1-6 地点 深圳
  - ➢ 解锁新的人生副本：一个人前往陌生的城市租房住
- □ 深圳是一个生活节奏很快的城市
  - ➢ 路上行人脚步很快，人行道的非机动车很快，日常的消费也很快
- □ 腾讯是一家很成熟的互联网企业
  - ➢ 入职培训(高压线)，组织架构，规范流程，工作环境
- □ 不同年龄段的人关注点相差很大
  - ➢ 买房，结婚，带娃
- □ 学习收获
  - ➢ 入门大语言模型推理

Overhead of CPU operations

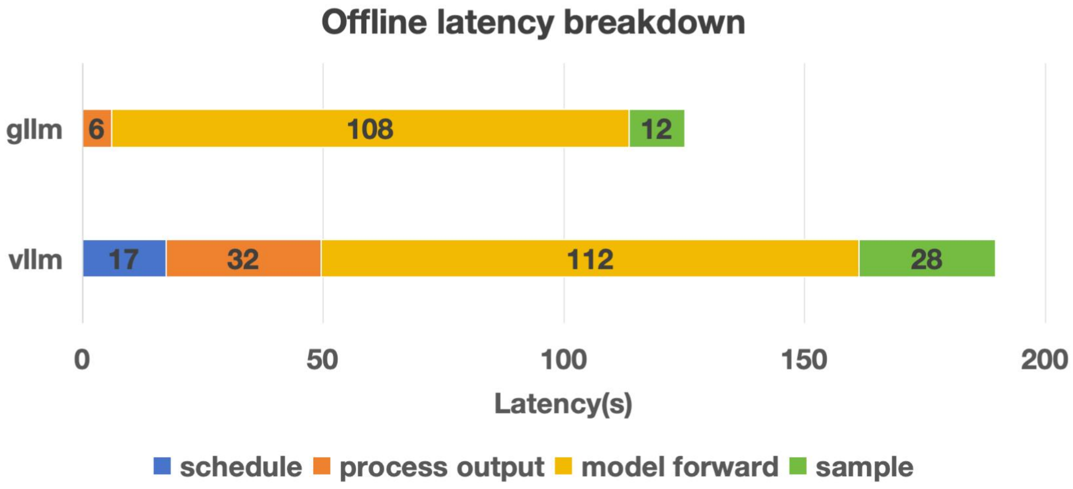
- Current LLM inference framework expect to **maximize** the utilization of GPU gLLM

Overhead of CPU operations

□ Current LLM inference framework expect to **maximize** the utilization of GPU

□ **Overhead** of operations on CPU like schedule and output process are **nonneglectable**

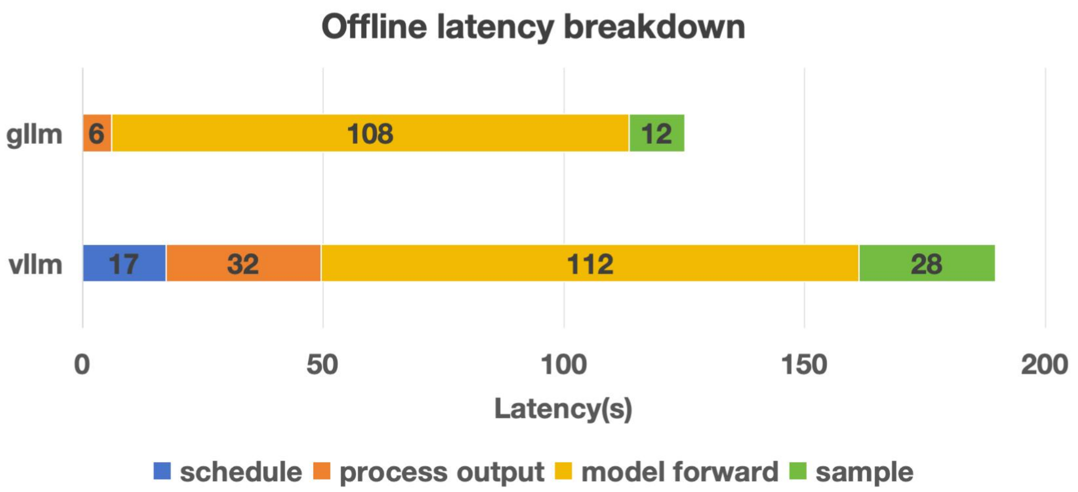
□ About **25%** of the time is spent on CPU operations

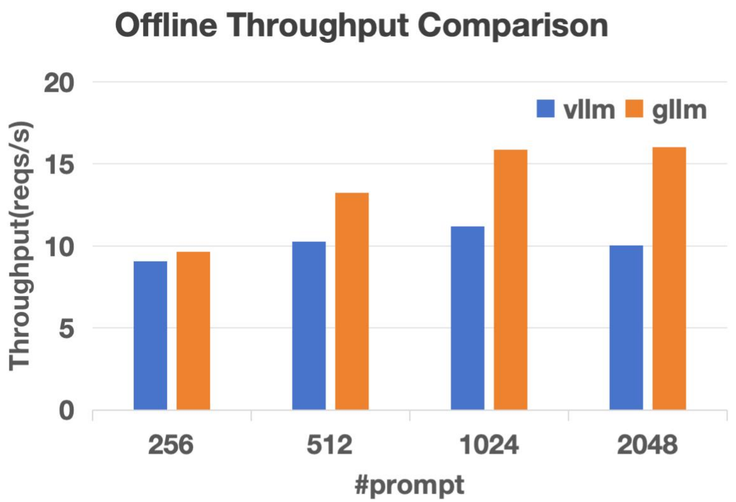


Overhead of CPU operations

gLLM

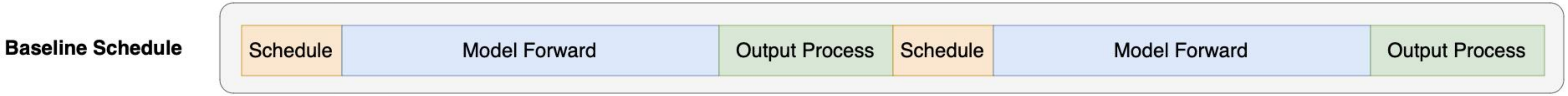
- □ Current LLM inference framework expect to **maximize** the utilization of GPU
- □ **Overhead** of operations on CPU like schedule and output process are **nonneglectable**
- □ About **25%** of the time is spent on CPU operations
- □ Reduce overhead of CPU opearations can improve performance


Offline latency breakdown


Offline Throughput Comparison

Pipeline Schedule

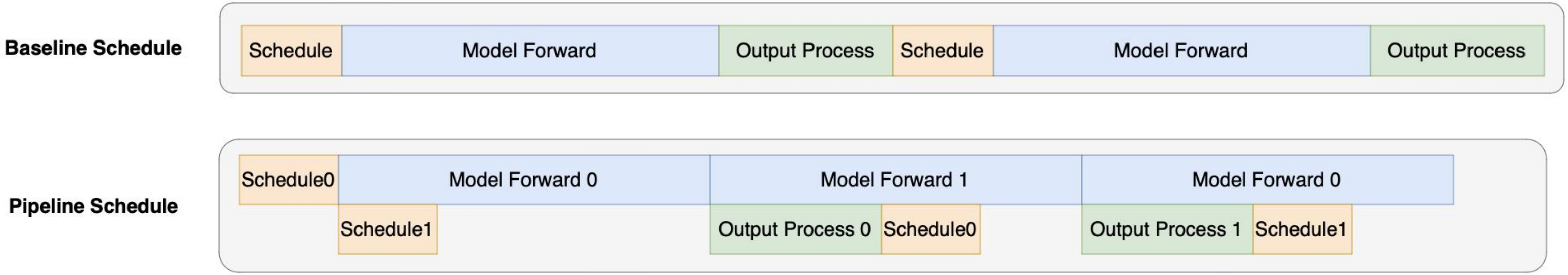
- Baseline schedule will **serialize** the execution of schedule, model forward and output process



Pipeline Schedule

□ Baseline schedule will **serialize** the execution of schedule, model forward and output process

□ Motivated by pipeline parallelism, pipeline schedule is introduced to **overlap** the execution between model forward and other procedure



Problems in Pipeline Schedule

- **Simultaneously** schedule **2 batches** of requests that can be processed by GPU
  - prefill-prefill, prefill-decode, decode-decode

Problems in Pipeline Schedule

- □ **Simultaneously** schedule **2 batches** of requests that can be processed by GPU
  - ➢ prefill-prefill, prefill-decode, decode-decode
- □ Careful orchestration addresses **dependencies** between execution phases
  - ➢ schedule => model forward => output process => schedule

Problems in Pipeline Schedule

□ **Simultaneously** schedule **2 batches** of requests that can be processed by GPU

- ➢ prefill-prefill, prefill-decode, decode-decode

□ Careful orchestration addresses **dependencies** between execution phases

- ➢ schedule => model forward => output process => schedule

□ Launch **multi-process** to implement execution overlaps

- ➢ Master process: schedule and output process
- ➢ Worker process: model forward

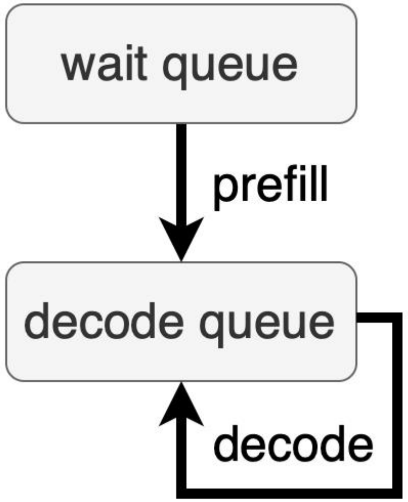
Simultaneously Batching

- There are three types of simultaneously batching
  - prefill-prefill
  - prefill-decode
  - decode-decode

Simultaneously Batching

- □ There are three types of simultaneously batching
  - ➢ prefill-prefill
  - ➢ prefill-decode
  - ➢ decode-decode

□ As the prefill only relys on wait queue, prefill-prefill and prefill-decode acts like serial schedule

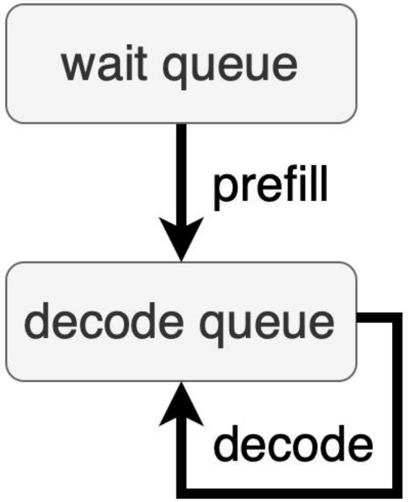


Simultaneously Batching

□ There are three types of simultaneously batching

- ➢ prefill-prefill
- ➢ prefill-decode
- ➢ decode-decode

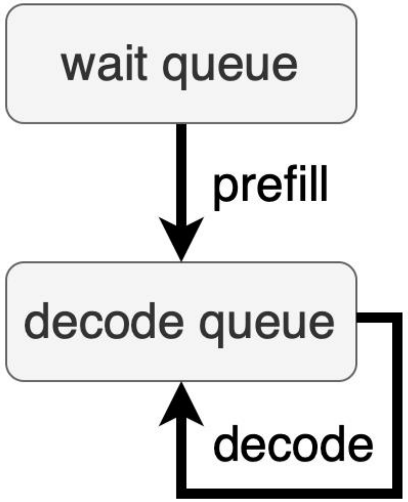
□ As the prefill only relys on wait queue, prefill-prefill and prefill-decode acts like serial schedule



□ Decode-decode may exhibit unbalanced batching: one schedule 225 seqs and the other schedule 31 seqs

Simultaneously Batching

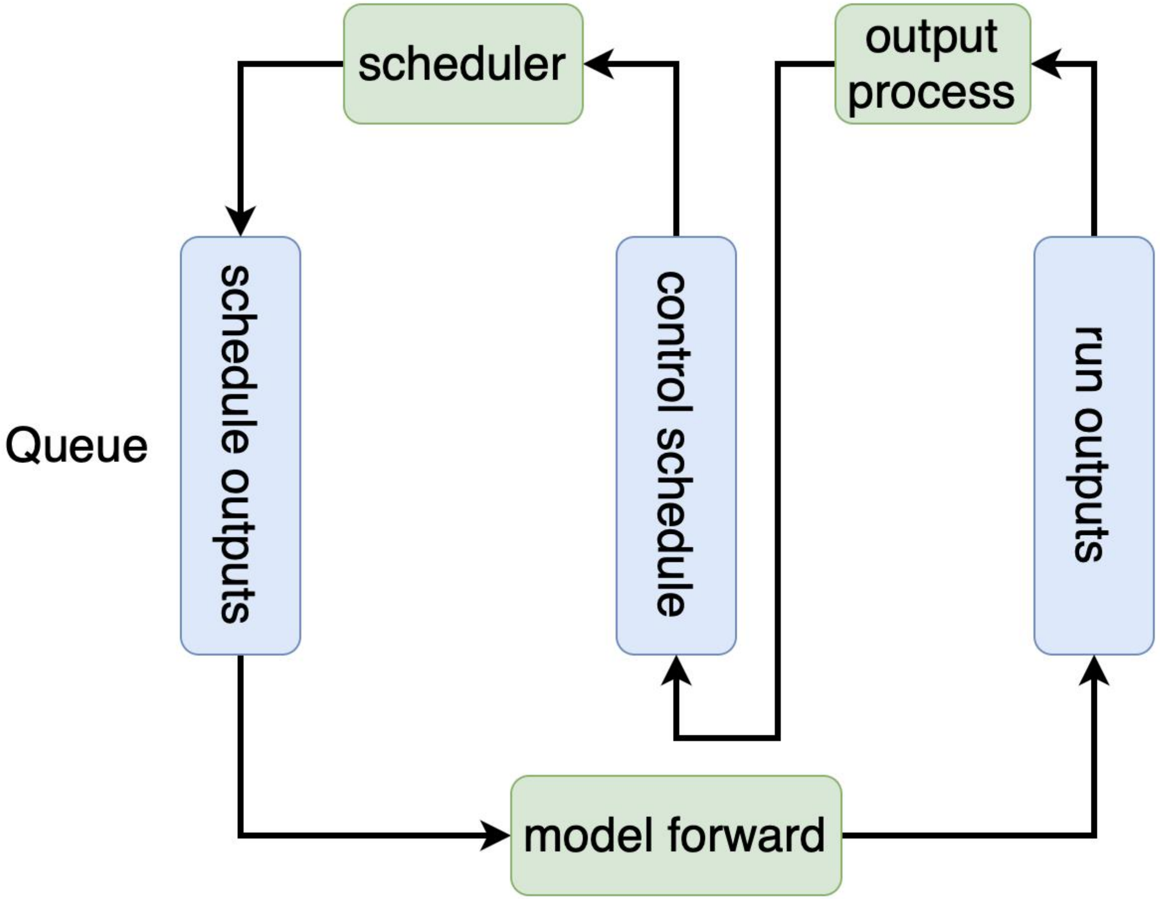
- □ There are three types of simultaneously batching
  - ➢ prefill-prefill
  - ➢ prefill-decode
  - ➢ decode-decode
- □ As the prefill only relys on wait queue, prefill-prefill and prefill-decode acts like serial schedule



- □ Decode-decode may exhibit unbalanced batching: one schedule 225 seqs and the other schedule 31 seqs
- □ **Delay scheduling** is proposed to merge two unbalanced schedule

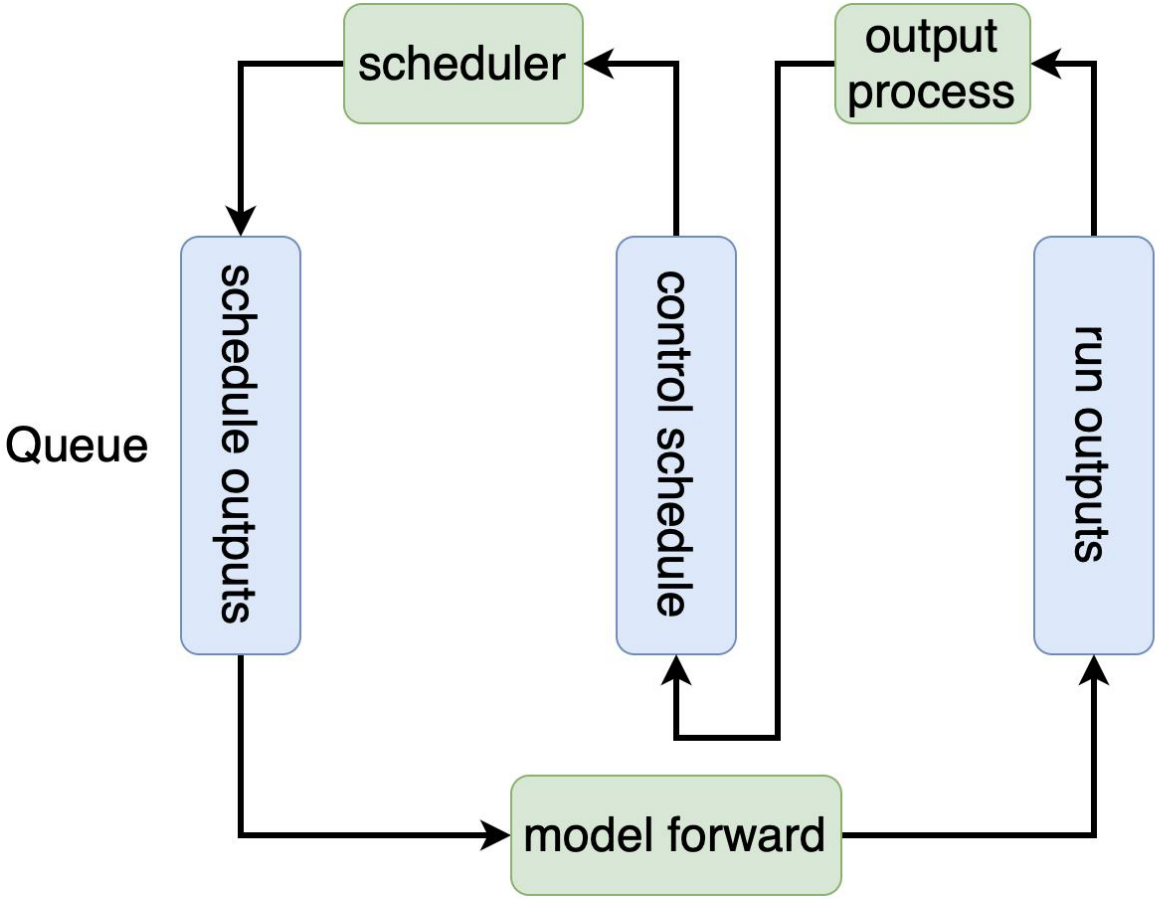
Dependency Control

- We can use three queues to control dependency
  - Schedule outputs
  - Run outputs
  - Control schedule



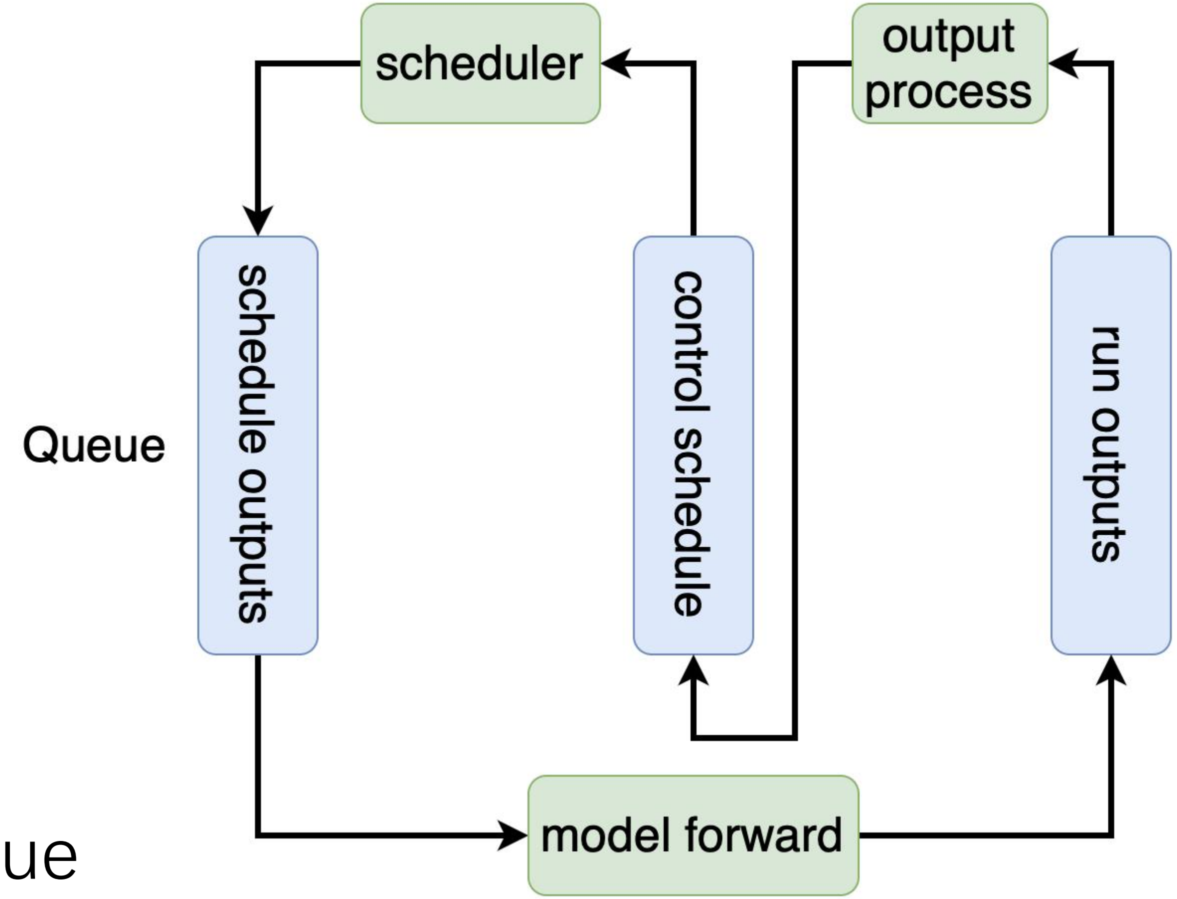
Dependency Control

- □ We can use three queues to control dependency
  - ➢ Schedule outputs
  - ➢ Run outputs
  - ➢ Control schedule
- □ Each module get an element from the queue



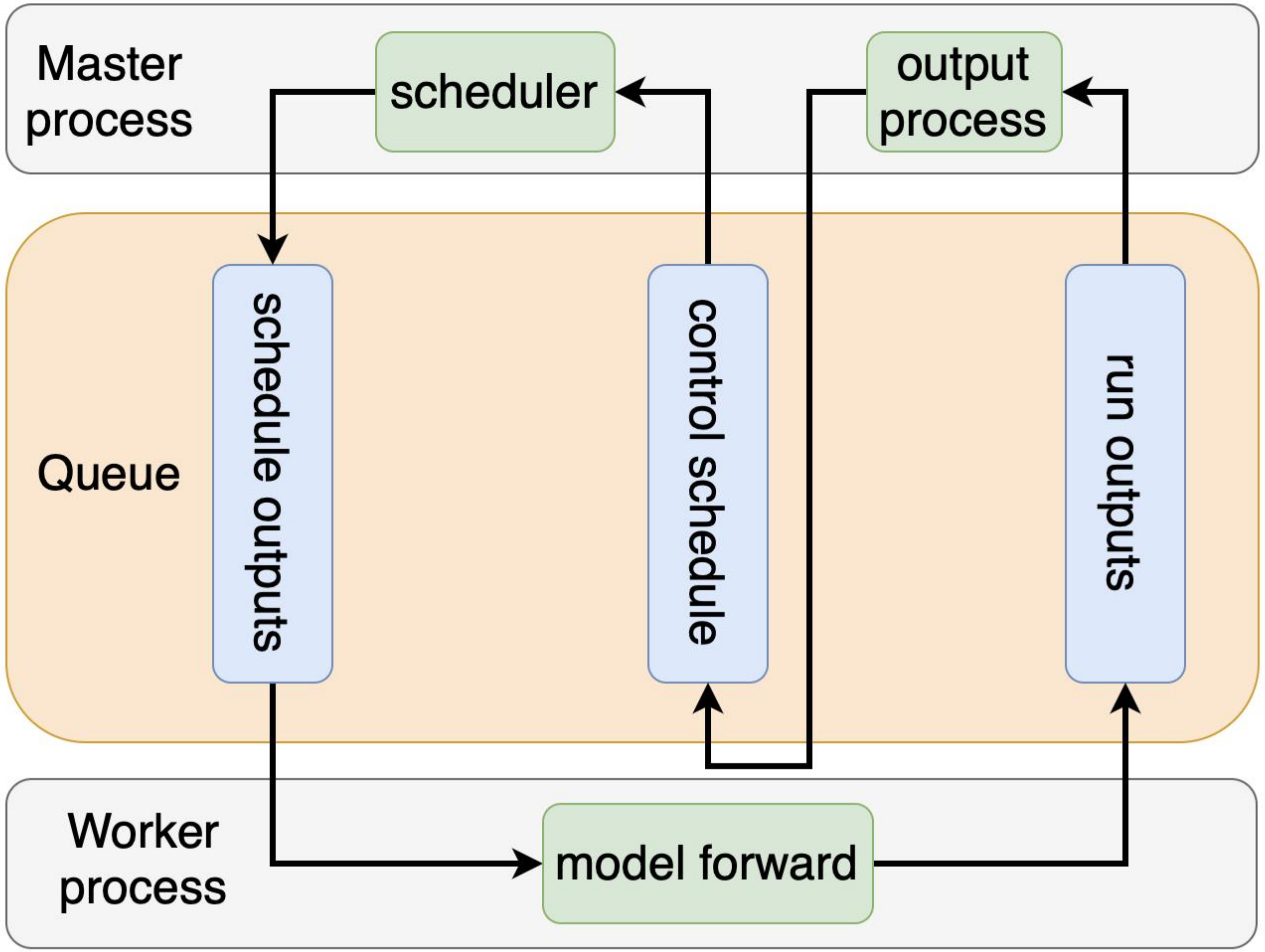
Dependency Control

- □ We can use three queues to control dependency
  - ➢ Schedule outputs
  - ➢ Run outputs
  - ➢ Control schedule
- □ Each module get an element from the queue
- □ The module is blocked when there isn't any element in the queue



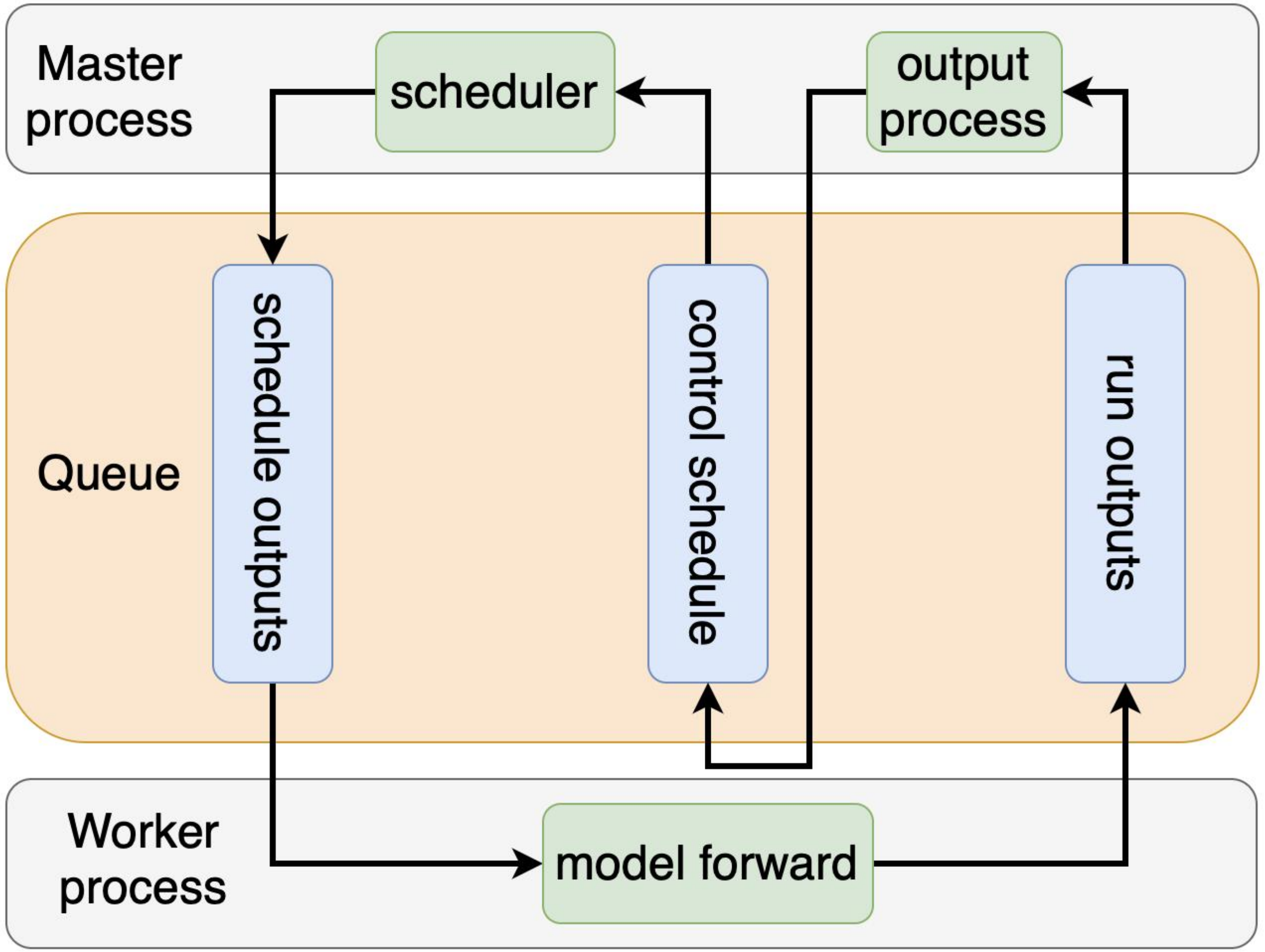
Multi-process

- Master process
  - Generate schedule outputs
  - Process model outputs



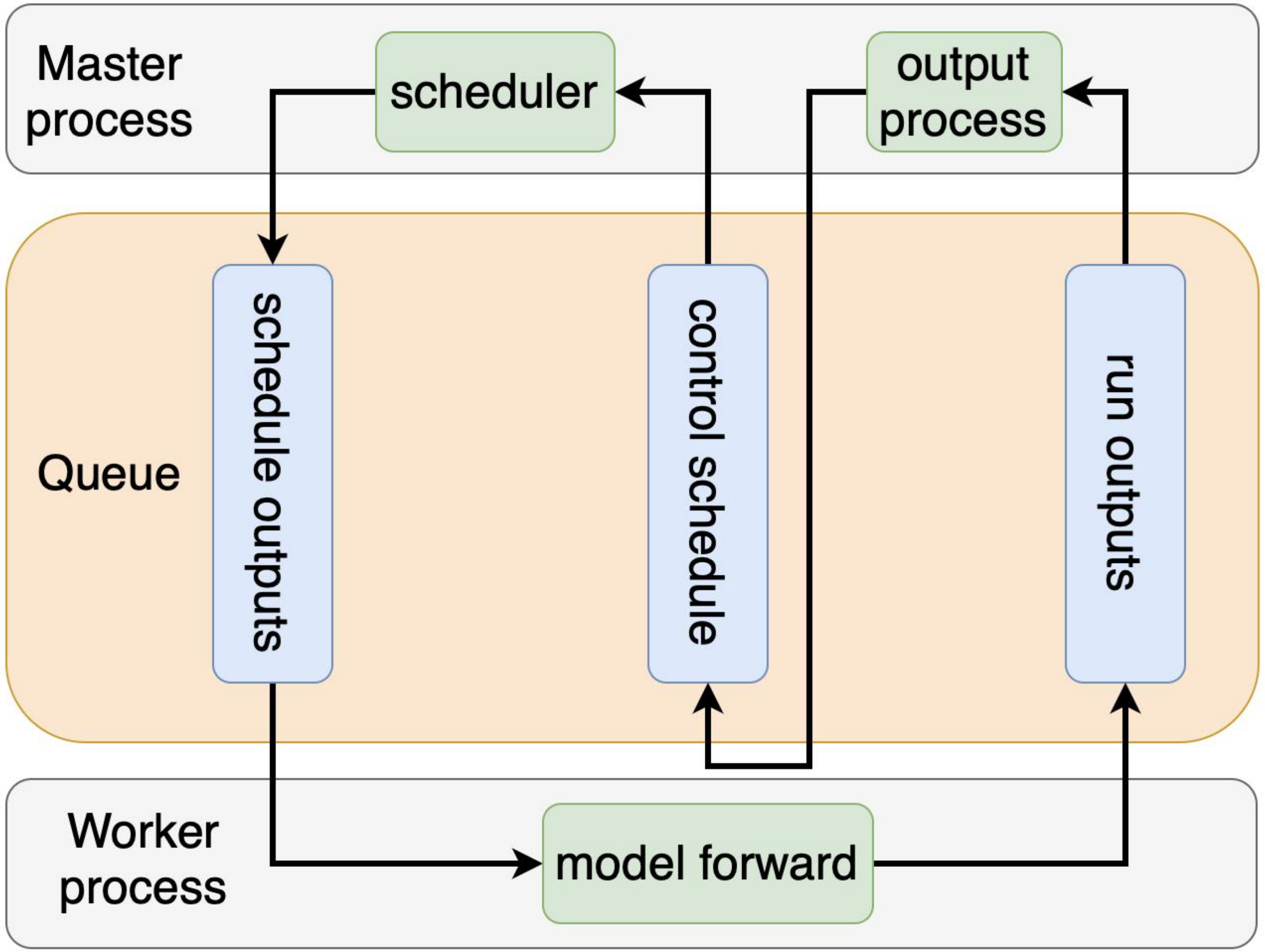
Multi-process

- □ Master process
  - ➢ Generate schedule outputs
  - ➢ Process model outputs
- □ Worker process
  - ➢ Generate model outputs



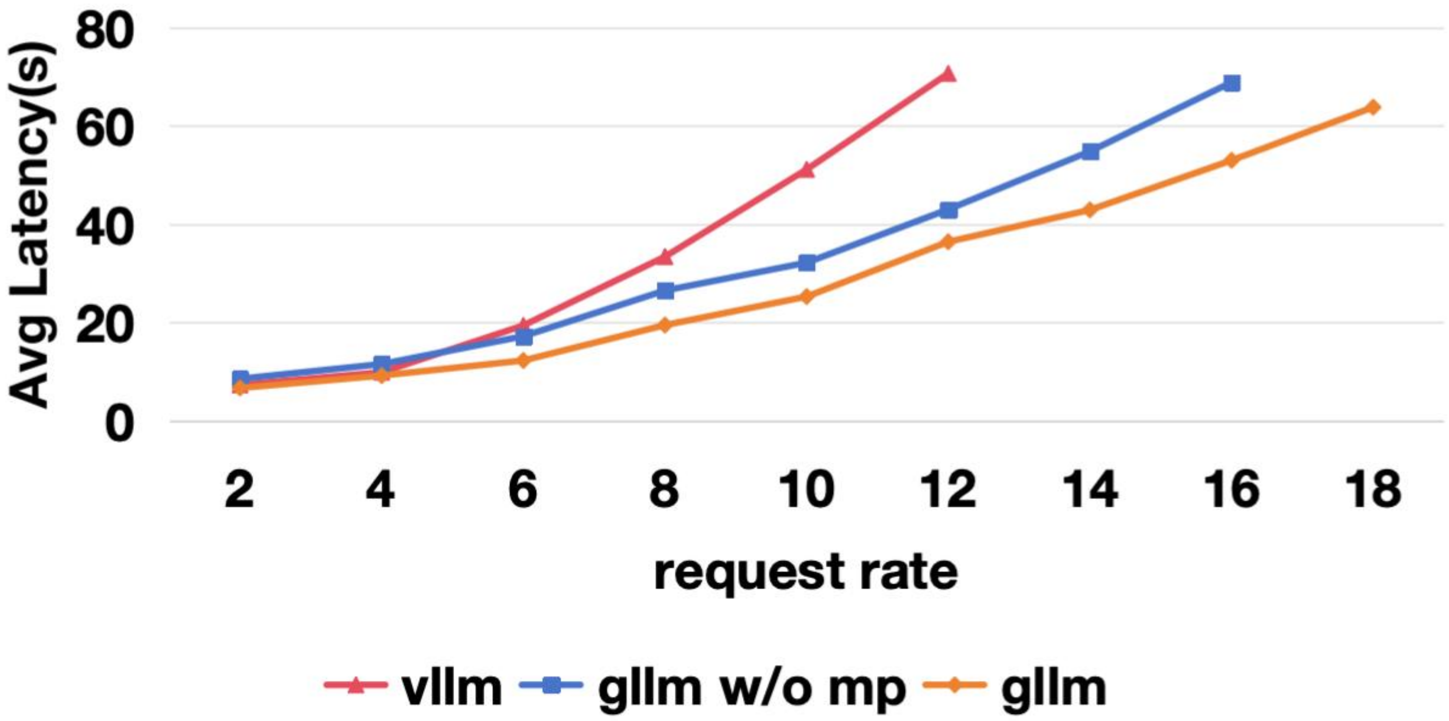
Multi-process

- □ Master process
  - ➢ Generate schedule outputs
  - ➢ Process model outputs
- □ Worker process
  - ➢ Generate model outputs
- □ Inter-process communication
  - ➢ Schedule outputs
  - ➢ Model outputs



Evaulated Result

- □ We implement pipeline schedule in gLLM
- □ We use shareGPT to benchmark online serving performance between vLLM, gLLM w/o pipeline schedule and gLLM on L20


Avg Latency(s)
request rate
vllm
gllm w/o mp
gllm