ADSL Reading Group (October 21)

ServeGen: Workload Characterization and Generation of Large Language Model Serving in Production

Yuxing Xiang[1,2] Xue Li[2] Kun Qian[2]
Wenyuan Yu[2] Ennan Zhai[2] Xin Jin[1]

[1] *School of Computer Science, Peking University* [2] *Alibaba Group*

https://arxiv.org/abs/2505.09999

Presenter：Zijian Dai

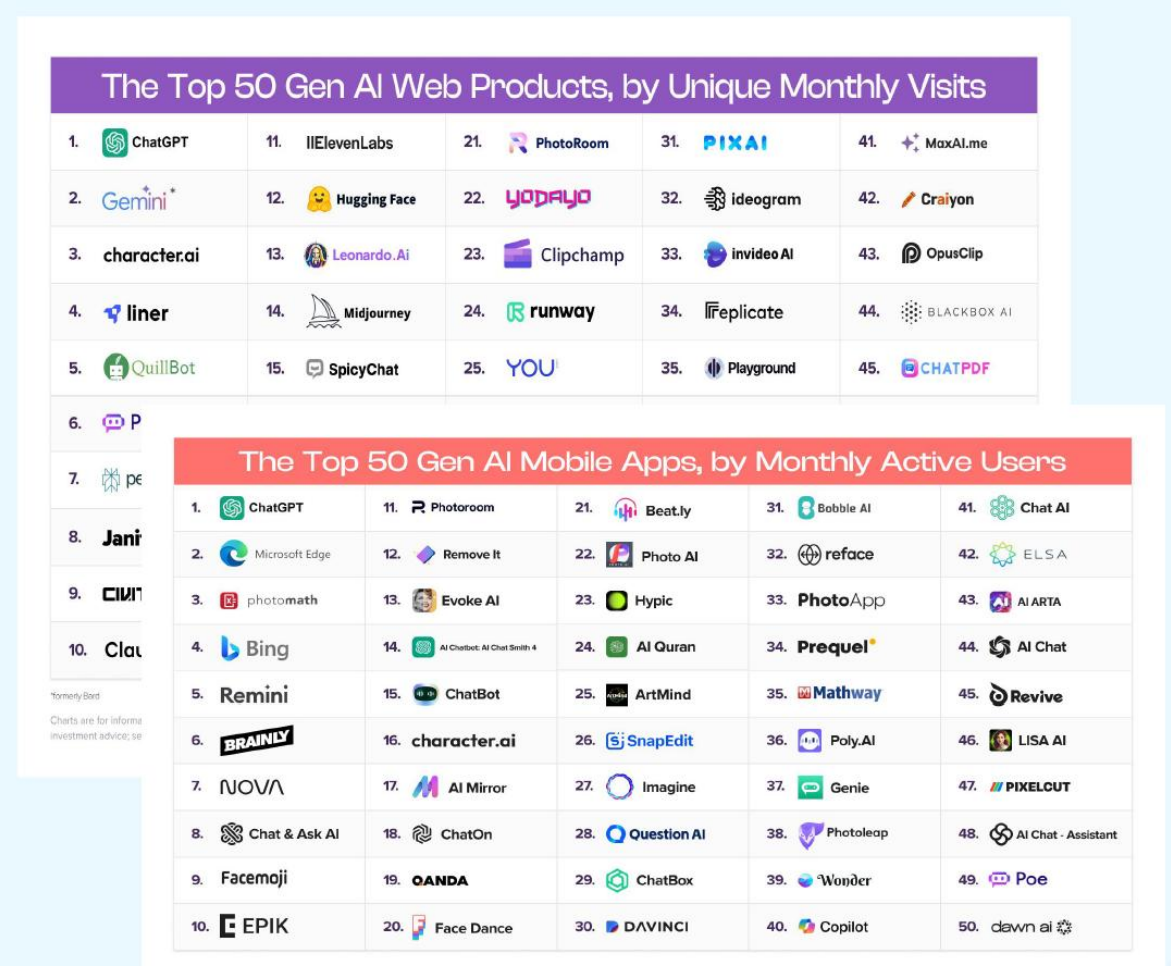
Contents

1

Background

Background

- The increasingly application of large models is driving a growing demand for inference services.



- Inference engines are constantly evolving.

vLLM

TensorRT-LLM

SGL

高吞吐

Transformers

最好兼容性

LLaMA C++

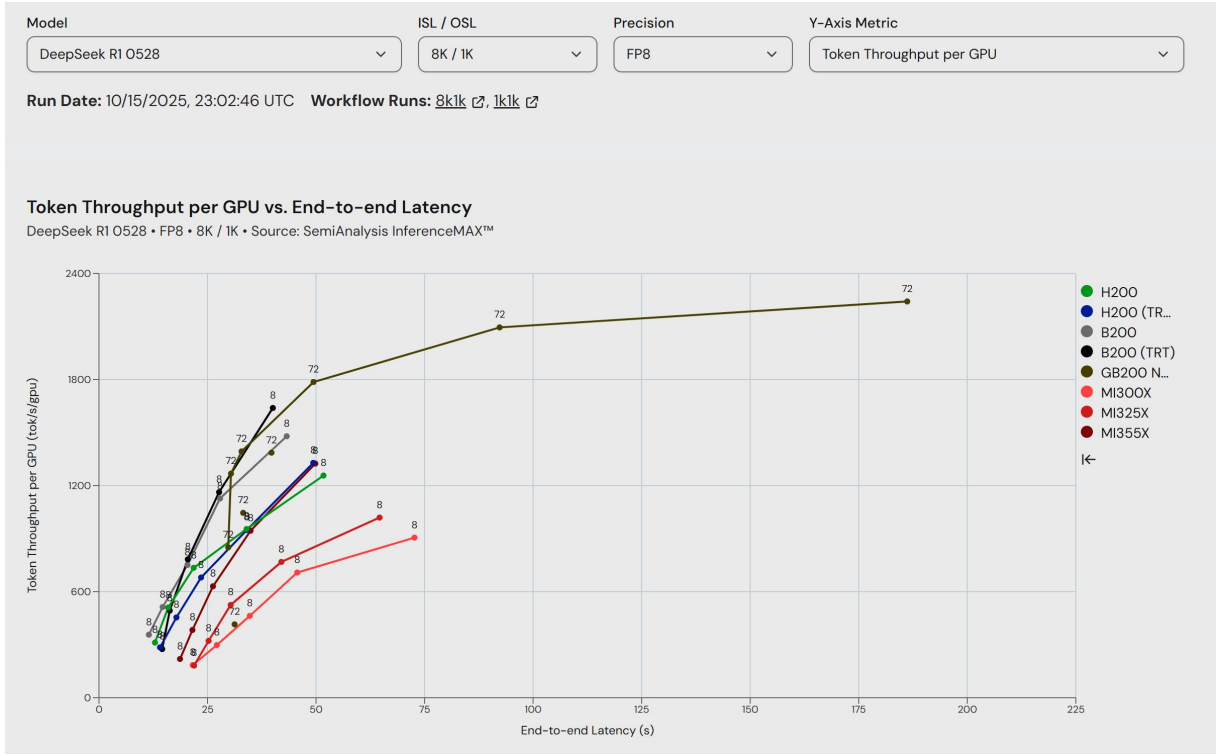
OpenVINO™

MLX

工作在端侧

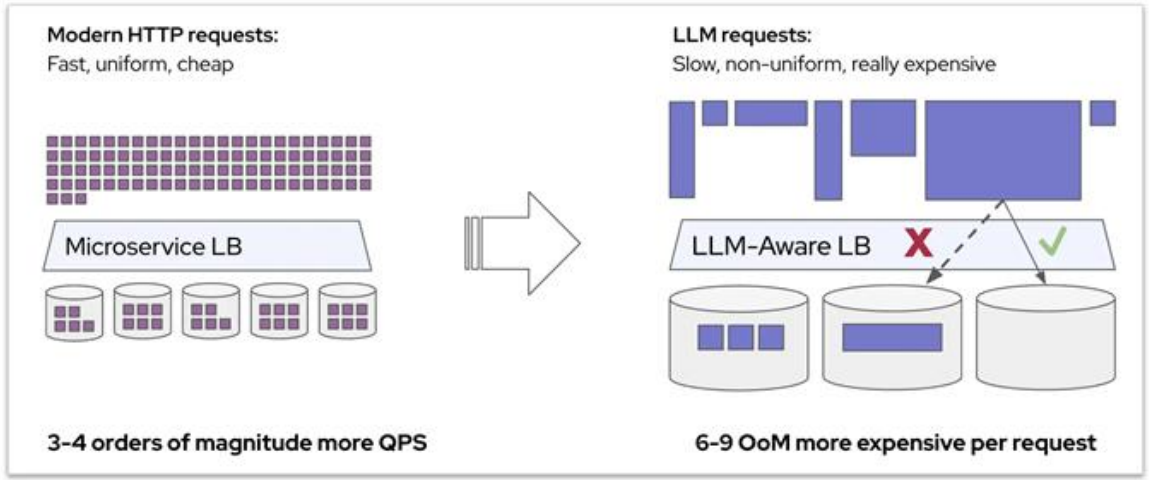
Background

- Benchmarking is critical for motivating LLM serving techniques and systems



https://inferencemax.semianalysis.com/

- Existing understanding of real-world LLM serving workloads is limited due to the lack of a comprehensive workload characterization.



Intro

- Many studies use Naïve approach to generate workloads， For request Inter-arrival time：

1. Sampled from Poisson or Gamma Processes
2. Scaled from published traces with datasets

- Analysis and experiments on real-world workload show that benchmarks based on these Naïve methods are inaccurate, leading to 50% under-provisioning in instance provisioning.

Open source workload of LLM serving

- Qwen-Bailian Anonymous Dataset

Scenarios: ChatGPT-like service, task automation with API calling

```
{
  "chat_id": 159,
  "parent_chat_id": 55,
  "timestamp": 61.114,
  "input_length": 521,
  "output_length": 132,
  "type": "text",
  "turn": 2,
  "hash_ids": [1089, 1090, 1091, 6326, ..., 13148]
}
```

1. https://github.com/alibaba-edu/qwen-bailian-usagetraces-anon
Optimizing KVCache cache design. (KVCache@ATC'25)

- Mooncake

Scenarios: conversation, tool&agent , synthetic

Listing 2: Request samples.

```
{
    "timestamp": 27000,
    "input_length": 6955,
    "output_length": 52,
    "hash_ids": [46, 47, 48, 49, 50, 51, 52,
                 53, 54, 55, 56, 57, 2111, 2112]
}
{
    "timestamp": 30000,
    "input_length": 6472,
    "output_length": 26,
    "hash_ids": [46, 47, 48, 49, 50, 51, 52,
                 53, 54, 55, 56, 57, 2124]
}
```

2. https://github.com/kvcache-ai/Mooncake/tree/main/FAST25-release/traces (Mooncake@FAST'25)

- BurstGPT

Scenarios: Azure Openai Conversation

Timestamp	Model	Request tokens	Response tokens	Total tokens	Log Type
5	ChatGPT	472	18	490	Conversation log
45	ChatGPT	1087	230	1317	Conversation log
118	GPT-4	417	276	693	Conversation log
185	ChatGPT	1360	647	2007	Conversation log
214	ChatGPT	185	215	400	Conversation log
233	GPT-4	586	293	879	Conversation log
261	ChatGPT	37	1656	1693	Conversation log
267	ChatGPT	54	503	557	Conversation log
410	ChatGPT	1528	414	1942	Conversation log
535	ChatGPT	89	370	459	Conversation log
560	GPT-4	549	362	911	Conversation log
638	ChatGPT	172	69	241	Conversation log
686	GPT-4	969	206	1175	Conversation log
741	ChatGPT	97	137	234	Conversation log
771	GPT-4	1574	501	2075	Conversation log
821	ChatGPT	253	239	492	Conversation log
966	ChatGPT	23	344	367	Conversation log
974	ChatGPT	509	207	716	Conversation log
1028	ChatGPT	387	349	736	Conversation log
1072	ChatGPT	733	180	913	Conversation log
1106	ChatGPT	0	0	0	Conversation log
1129	ChatGPT	1305	477	1782	Conversation log
1278	ChatGPT	17	382	399	Conversation log
1449	ChatGPT	601	557	1158	Conversation log

3.https://github.com/HPMLL/BurstGPT/tree/main (BurstGPT)

2

Characterizing Language Workloads

Workload source: Alibaba Bailian Platform

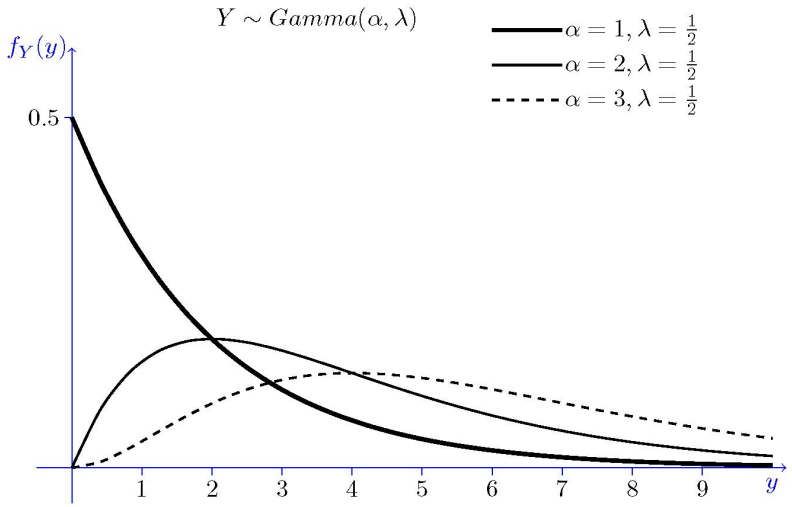
Duration: from January to April 2025
Models: 12
Requests: 3.54B
GPUs: 10K distributed in dozens of regions and zones

Table 1. The list of workloads and models in our study.

Category	Name	Model	Description	Workload Information
Language	`M-large`	Qwen-Max	Largest, general-purpose	February
	`M-mid`	Qwen-Plus	Balanced, general-purpose	February
	`M-small`	Qwen-Turbo	Cheapest, general-purpose	February
	`M-long`	Qwen-Long with a 10M context length	Long-document comprehension	January (one week)
	`M-rp`	Tongyi-Xingchen	Role-playing	January (one week)
	`M-code`	Tongyi-Lingma	Code completion	January (one week)
Multimodal	`mm-image`	Qwen2.5-VL-72B	Image & text input	March
	`mm-audio`	Qwen2-Audio-7B	Audio & text input	March
	`mm-video`	Qwen2.5-VL-72B	Video & text input	March
	`mm-omni`	Qwen2.5-Omni-7B	Omni-modal input	April (one week)
Reasoning	`deepseek-r1`	DeepSeek-R1-671B	Full reasoning model	March (one week)
	`deepqwen-r1`	DeepSeek-R1-distill-Qwen-32B	Distilled reasoning model	March (one week)

随机过程相关知识

时间间隔的随机分布：（Exponential / Gamma / Weibull）



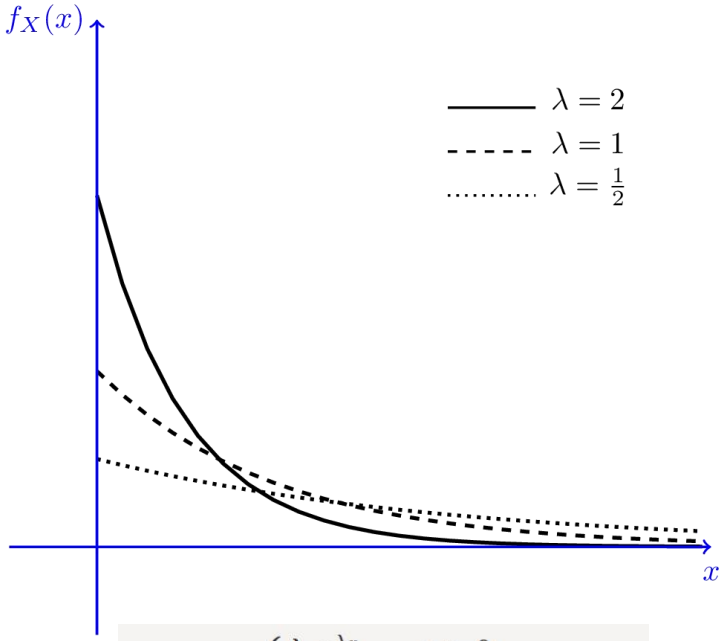
$$f_X(x) = \begin{cases} \frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

gamma分布

α是事件发生的次数

例如：商店第α个顾客到来、第α个热线电话呼入、第α个人口出生的时间间隔

α =1时退化 ⟹



$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

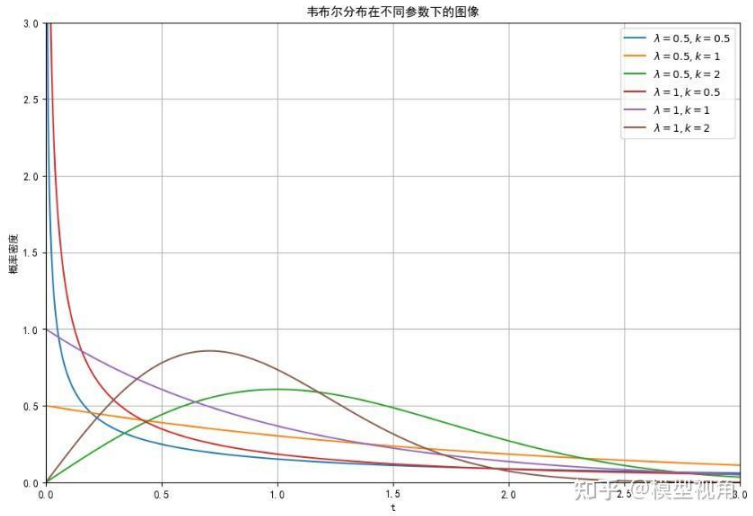
Exponential分布

例如：商店下1个顾客到来、下1个热线电话呼入、下1个人口出生的时间间隔

均值 $\mu = 1/\lambda$

标准差 $\sigma = \frac{1}{\lambda}$

变异系数CV=$\frac{\sigma}{\mu} = 1$



$$f(t) = k\lambda t^{k-1} e^{-\lambda t^k} \quad \text{for } t \geq 0$$

Weibull分布

例如：机器损坏时间

⟸ k=1时退化

Request Arrival Pattern

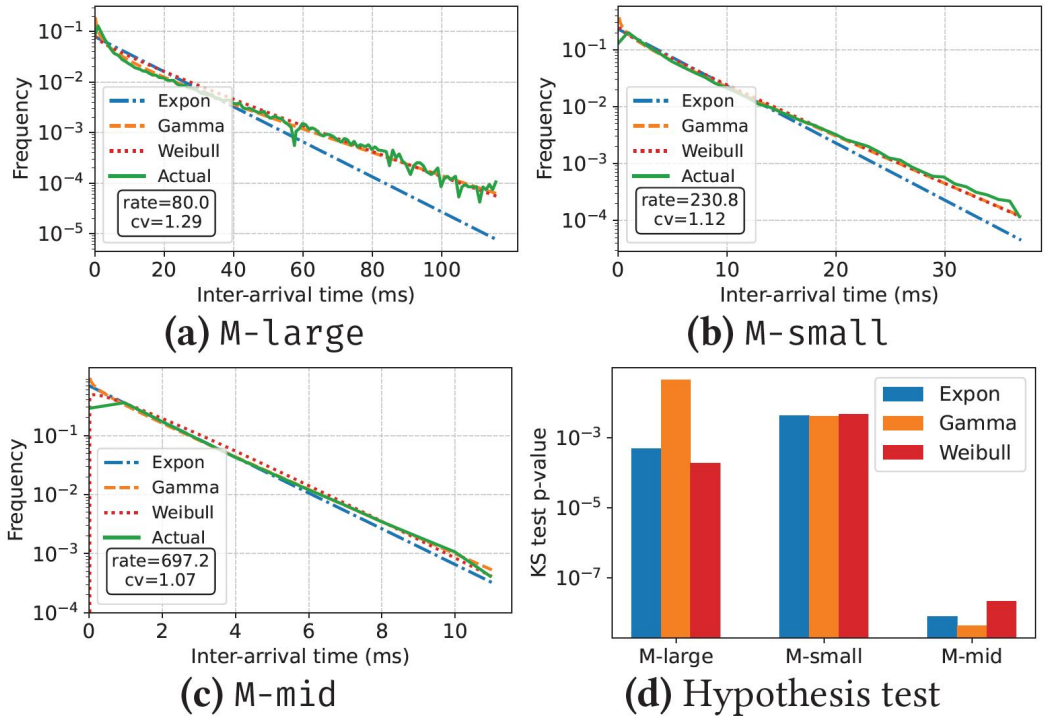


Figure 1. Inter-arrival time characterization.

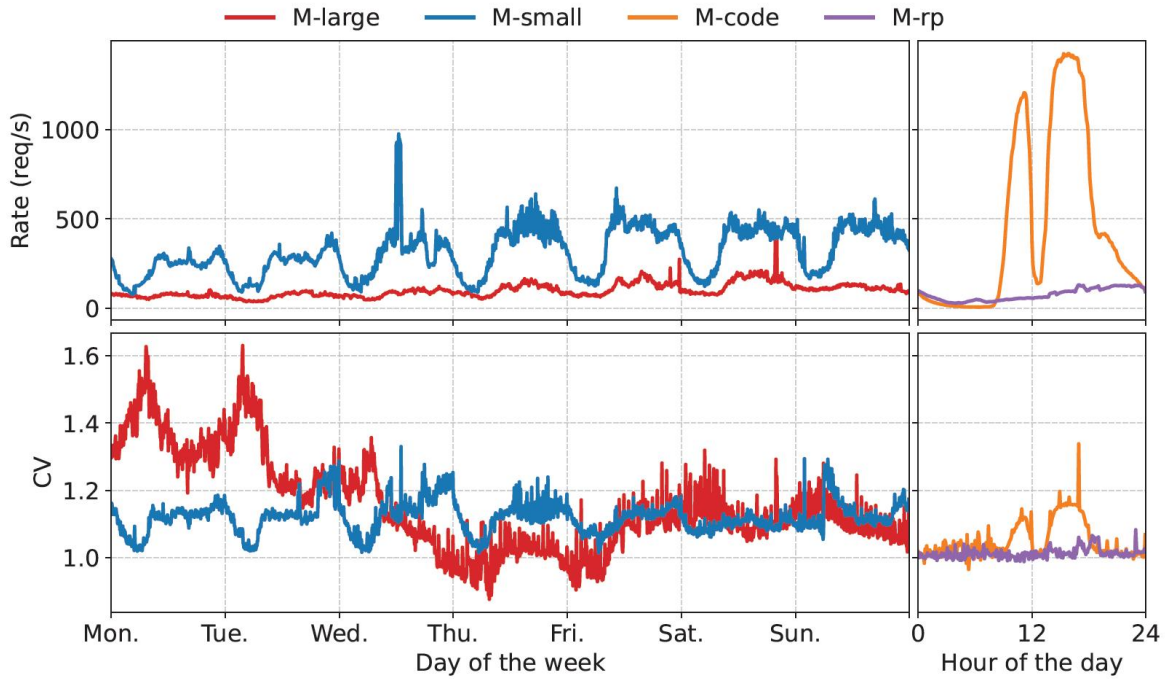


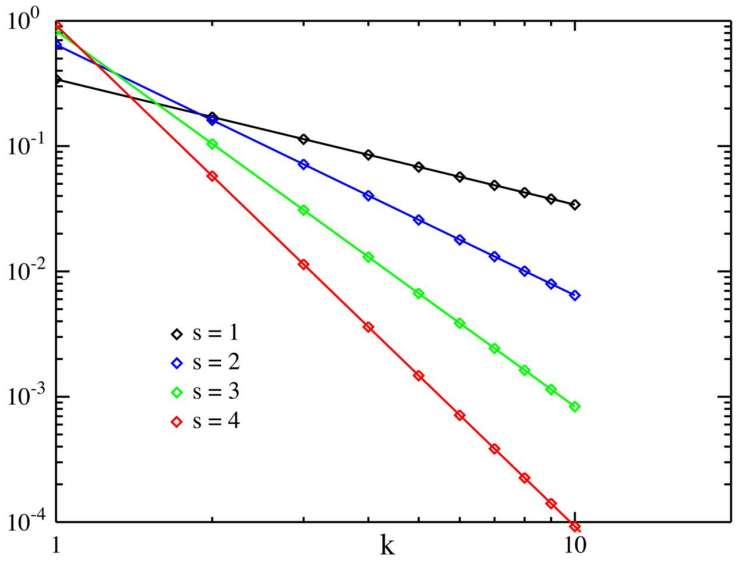
Figure 2. Long-term rate and CV shifts.

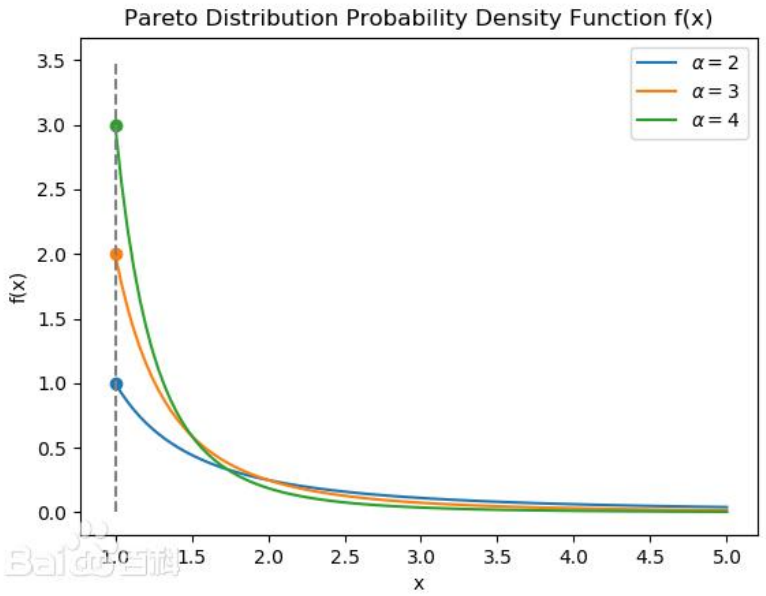
Finding 1: The short-term arrival of LLM requests is often **bursty (CV > 1)**, exhibiting complex patterns beyond any single stochastic process.

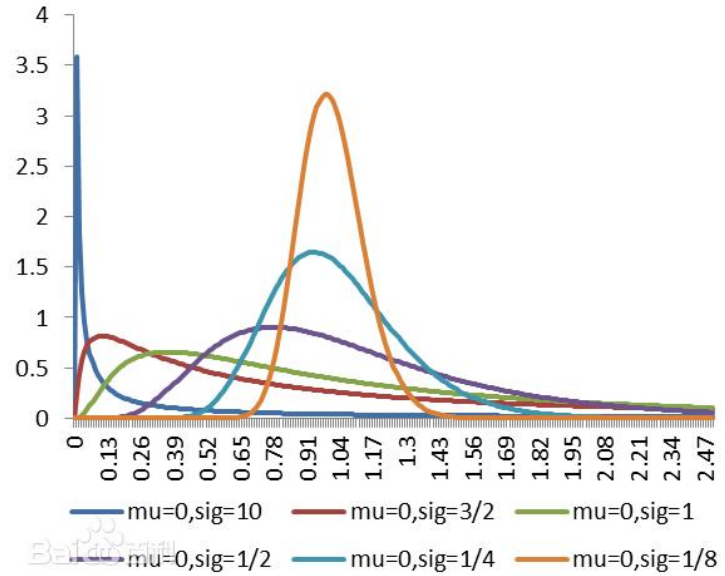
Finding 2: The arrival of LLM serving requests shows a diverse shifting pattern in terms of rate and burstiness, calling for adaptive system design.

随机过程相关知识

请求prompt长度的随机分布：（Pareto/Log-normal/zipf）







$$p(x) = \begin{cases} 0, & \text{if } x < x_{\min}; \\ \frac{k\, x_{\min}^{k}}{x^{k+1}}, & \text{if } x > x_{\min}. \end{cases}$$

$$f(x,\mu,\sigma) = \begin{cases} \frac{1}{x\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2\sigma^2}(lnx-\mu)^2\right], x>0 \\ 0, x\leq 0 \end{cases}$$

Zipf分布

例如：语料库里，一个单词出现的频率与它在频率表里的排名成反比

Pareto分布

例如：财富在个人之间的分布（即20%的人拥有80%的财富）

Log-normal分布

例如：收入、人口的分布

Input and Output Length Distribution

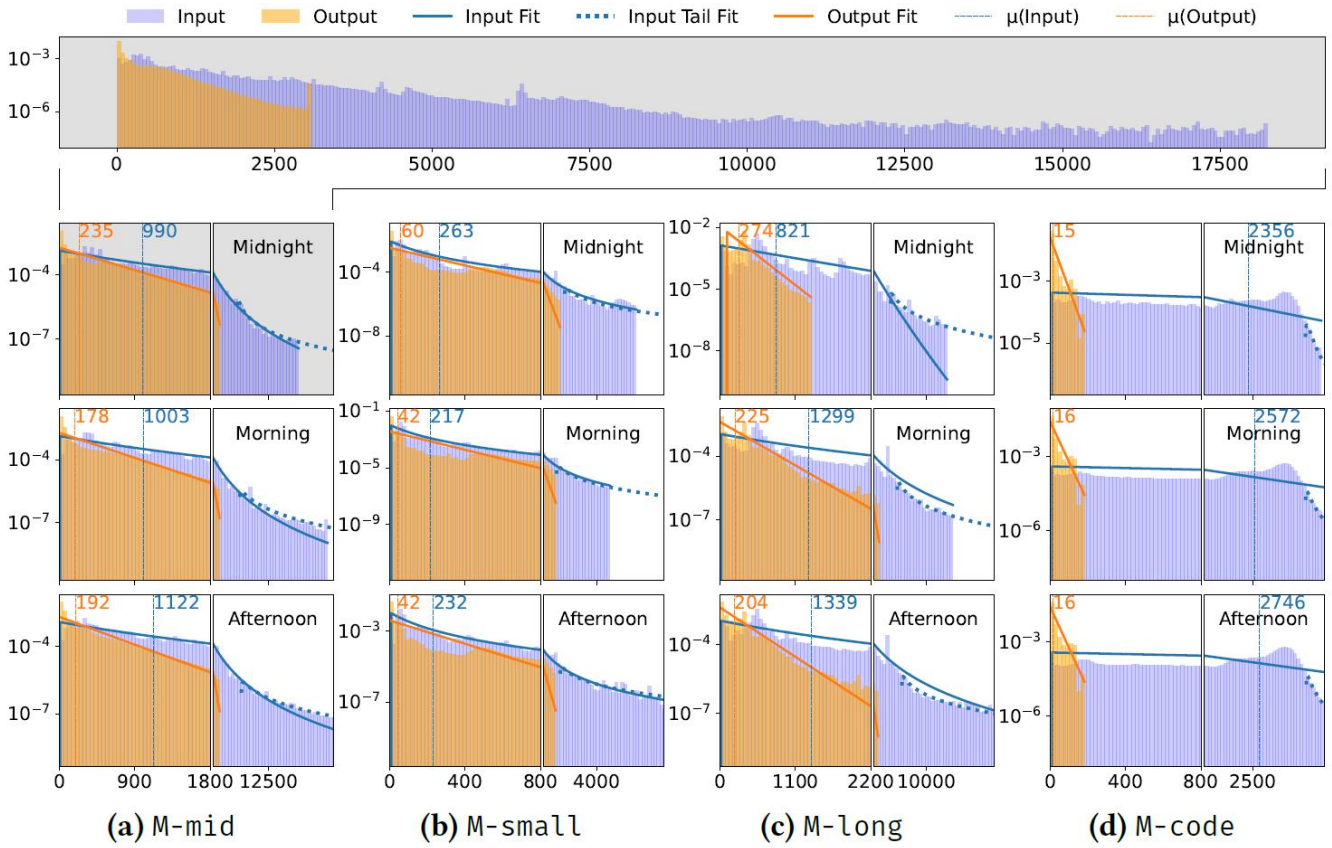


Figure 3. Input and output length distribution. *x-axis*: # tokens; *y-axis*: frequency. Each subfigure corresponds to a specific workload and time period, split to two consecutive x-scales to better visualize the shift in average lengths (left) as well as the tail distribution (right).

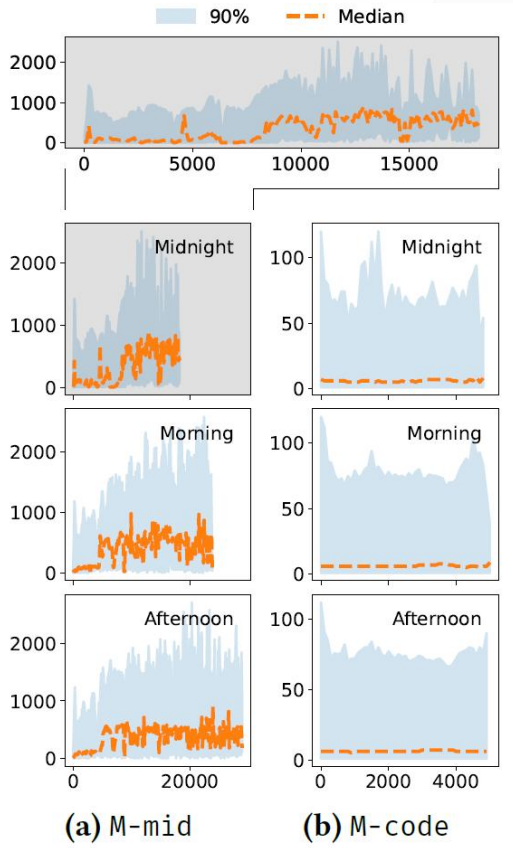
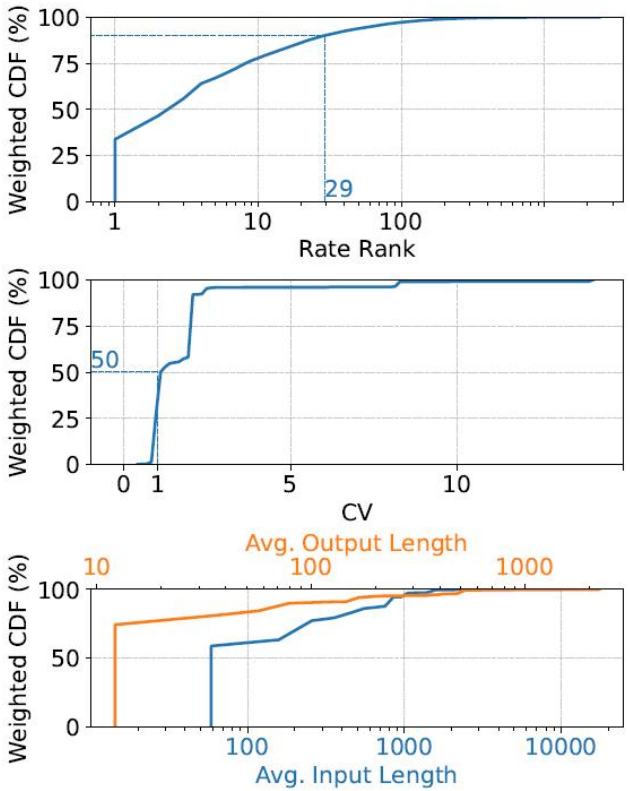


Figure 4. Input and output length correlation. *x-axis*: input length; *y-axis*: output length.

Finding 3: The input length distribution can be modeled with a mixture of Pareto and Log-normal distributions, and the output with Exponential distributions.

Finding 4: Correlation between input and output lengths is weak. The input and output length distributions shift dynamically and independently over time, leading to diverse load

Client Decomposition



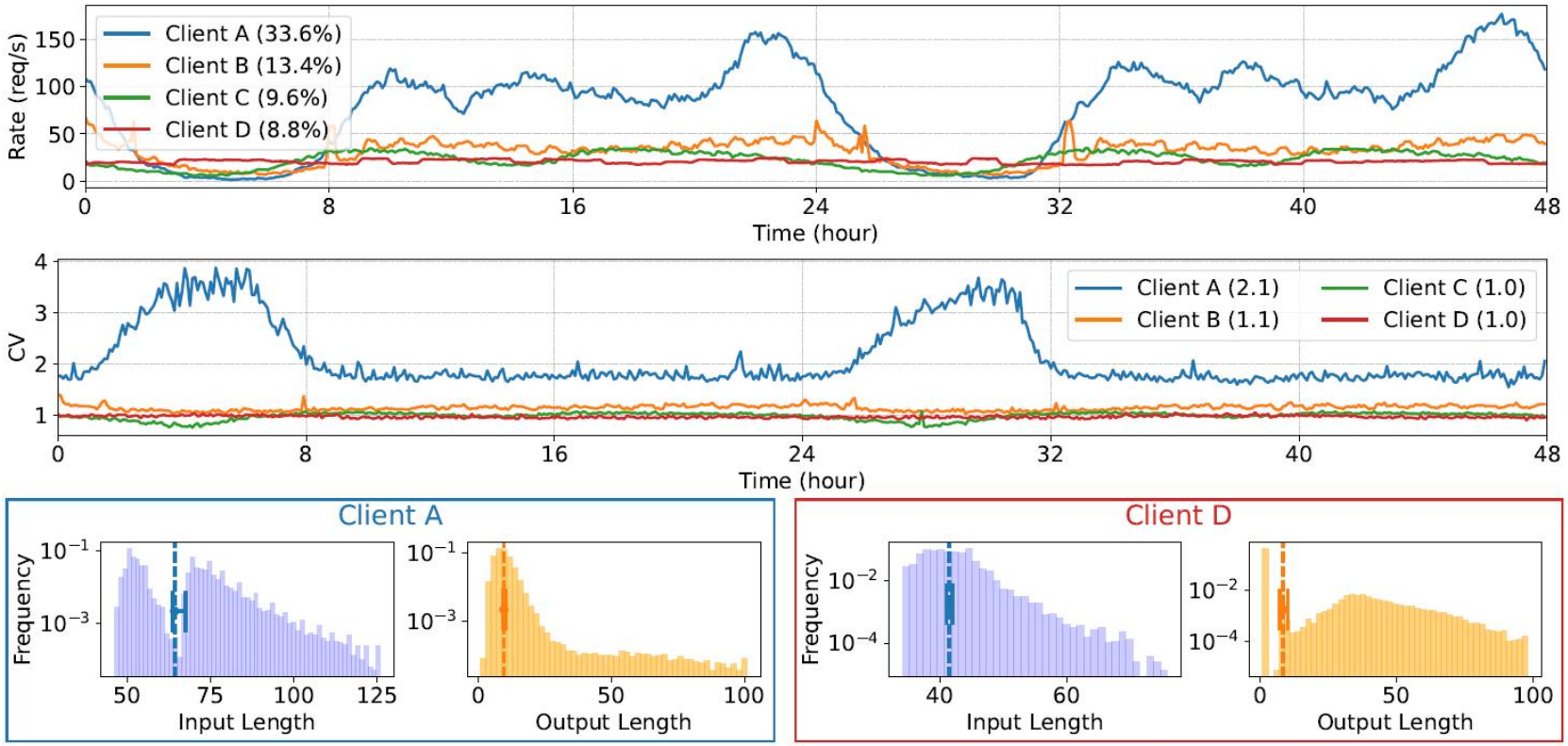


Figure 5. Client heterogeneity in terms of rate, *etc.* All CDFs are weighted by client rates.

Figure 6. Characterization of the top four clients in the `M-small` workload in isolation, using the first 48-hour data from Figure 2. Vertical lines in the last-row subfigures indicate average input/output lengths, and error bars show the range of average lengths in 1-hour windows.

observed 2,412 clients in 48-hour

Finding 5: Real-world workloads consist of heterogeneous clients with skewed arrival rates. The top clients and their rate fluctuations largely explain the shifting workload patterns

3

Characterizing Multimodal Workloads

Modality Load Variance

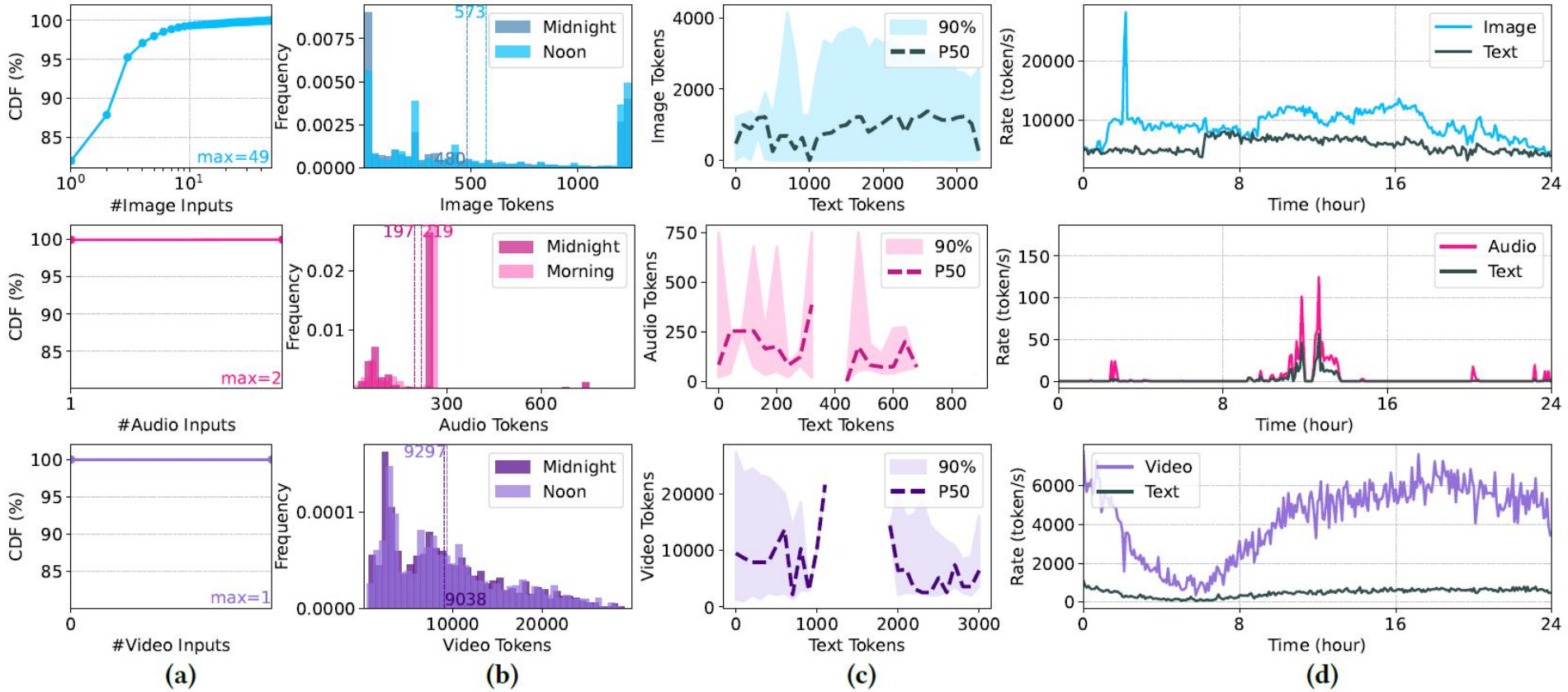


Figure 7. Characterization of multimodal inputs in three different workloads. *Rows*: `mm-image`, `mm-audio`, and `mm-video`, respectively. *Columns*: (a) number of multimodal inputs per request; (b) tokenized length distribution of multimodal inputs; (c) correlation between text tokens and multimodal tokens; (d) overall arrival rate of multimodal and text tokens.

- *multimodal inputs are more likely to have standard sizes (depending on upstream applications)*
- correlation between text tokens and multimodal tokens is weak

Modality Load Variance

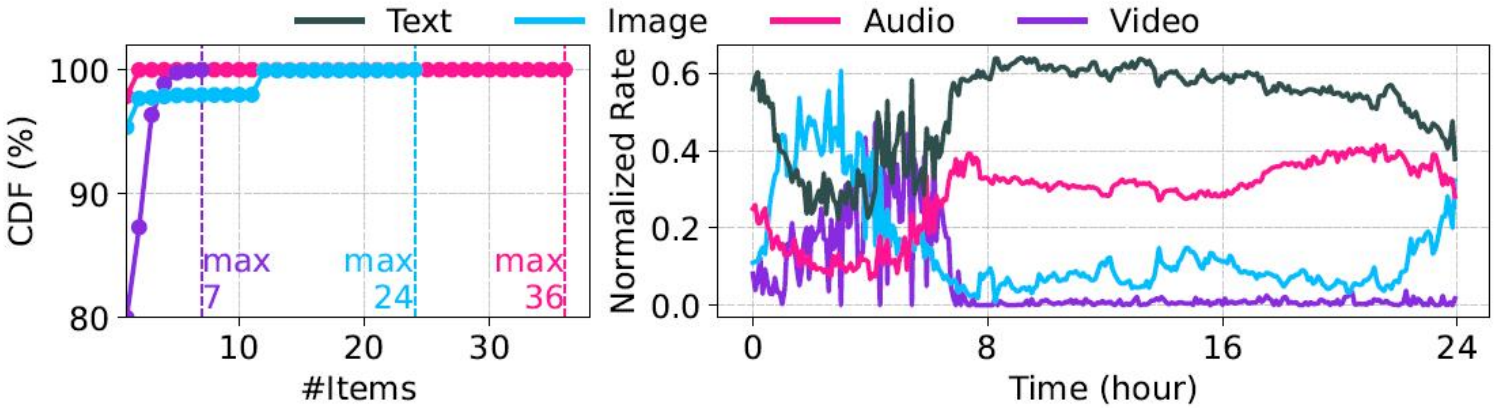


Figure 8. Characterization of omni-modal inputs in `mm-omni`. *Left*: number of multimodal inputs per request. *Right*: arrival rate of multimodal and text tokens, normalized by the total input rate.

- omni-modality workload exhibits more complex variability

Finding 6: *Multimodal data distributions exhibit irregular and independent shifts, underscoring significant load variance across modalities.*

Modality Load Variance

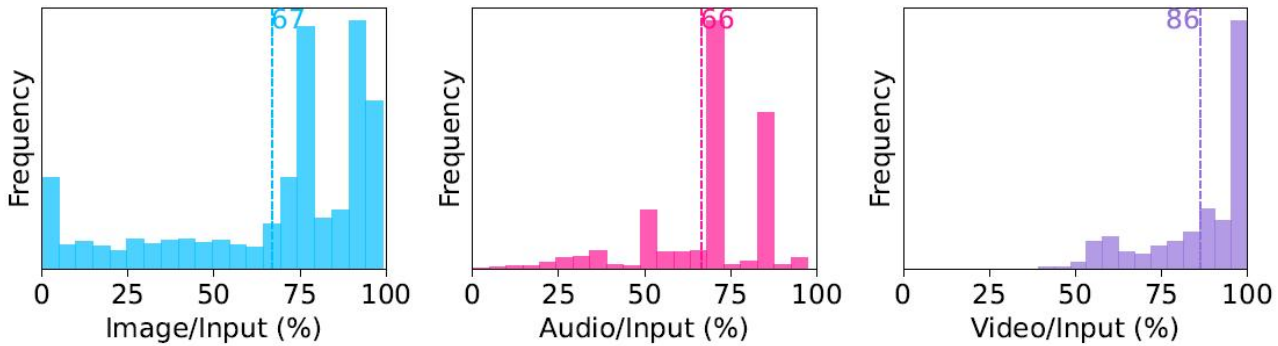
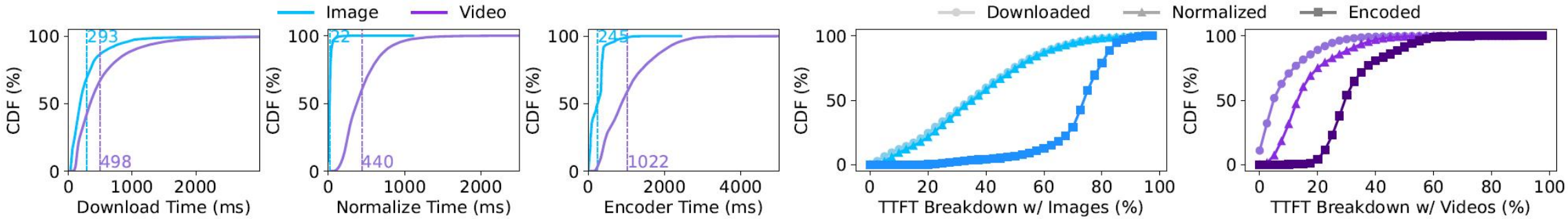


Figure 9. Ratio of multimodal input tokens per request in `mm-image`, `mm-audio`, and `mm-video`. Numbers indicate the average ratio.

- real multimodal requests are heterogeneous

Finding 7: Multimodal requests are heterogeneous with diverse ratios of multimodal inputs per request

Request Heterogeneity



(a) Per-stage time during first-token generation. **(b)** CDF of cummulative time after each stage.

Figure 10. Breakdown of first-token time when serving requests with image or video inputs (`mm-image` and `mm-video`).

Finding 8:
the download, normalization, and encoding stages for tokenizing multimodal inputs all contribute to considerable extra overhead,
which leads to prolonged TTFTs that necessitate tailored optimizations.

Request Heterogeneity

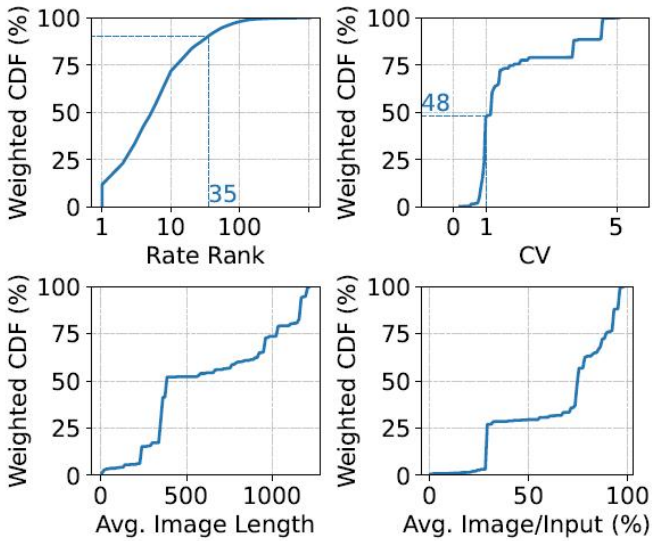


Figure 11. Client characterization for `mm-image`. CDFs are weighted by rates.

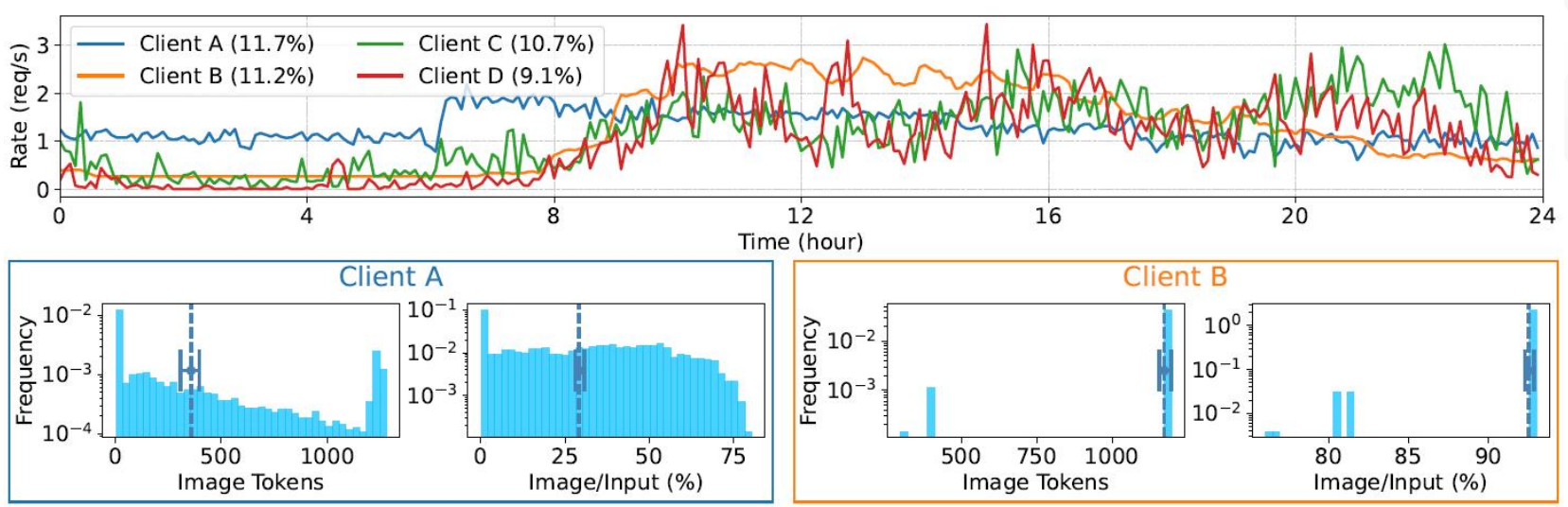


Figure 12. Behavior of top clients in `mm-image` in isolation. Vertical lines in the last-row subfigures indicate average lengths, and error bars show the range of average lengths within a day.

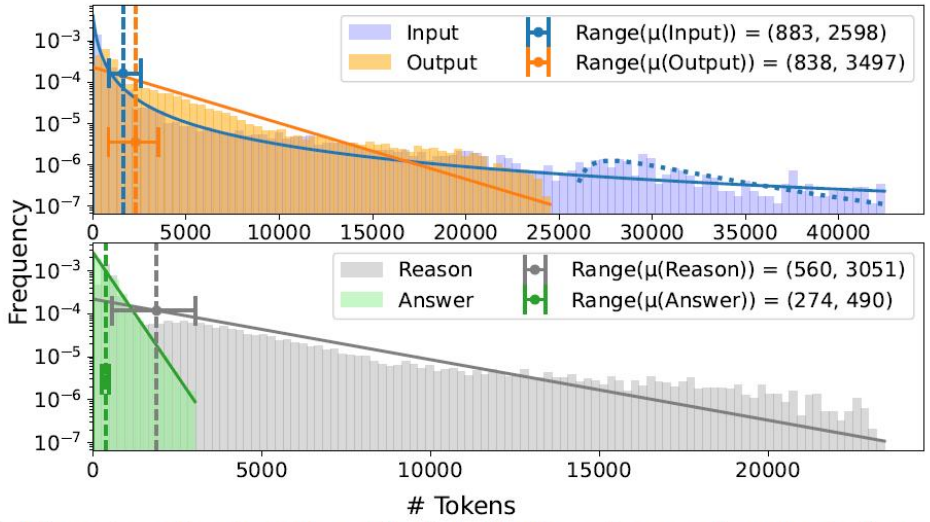
Among 1,036 multimodal clients

Finding 8: Top clients in multimodal workloads exhibit diverse behaviors, and characterizing them helps explain the overall workload patterns.

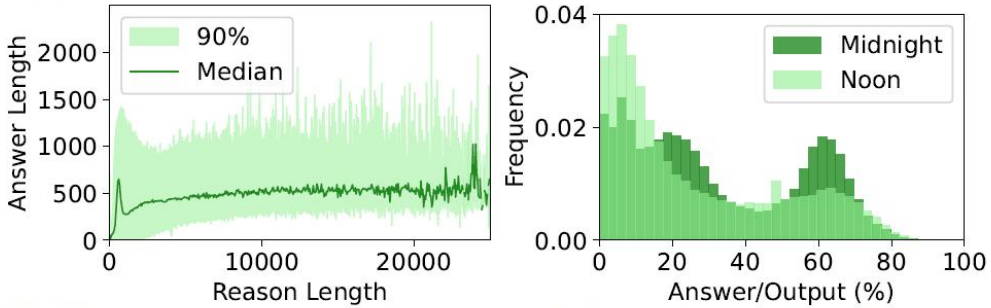
4

Characterizing Reasoning Workloads

Characterizing Reasoning Workloads



(a) Input and output length distribution, in one-hour windows.



(b) Reason-answer correlation. (c) Output length breakdown.

Figure 13. Characterization of input and output lengths for the `deepseek-r1` workload in one day. Error bars in (a) indicate the range of average lengths over the day.

Finding 9: Reasoning workloads exhibit longer and more variable output lengths, due to the reason tokens. In relation, reason and answer lengths display stronger positive correlation, as well as a unique bimodal ratio.

Characterizing Reasoning Workloads

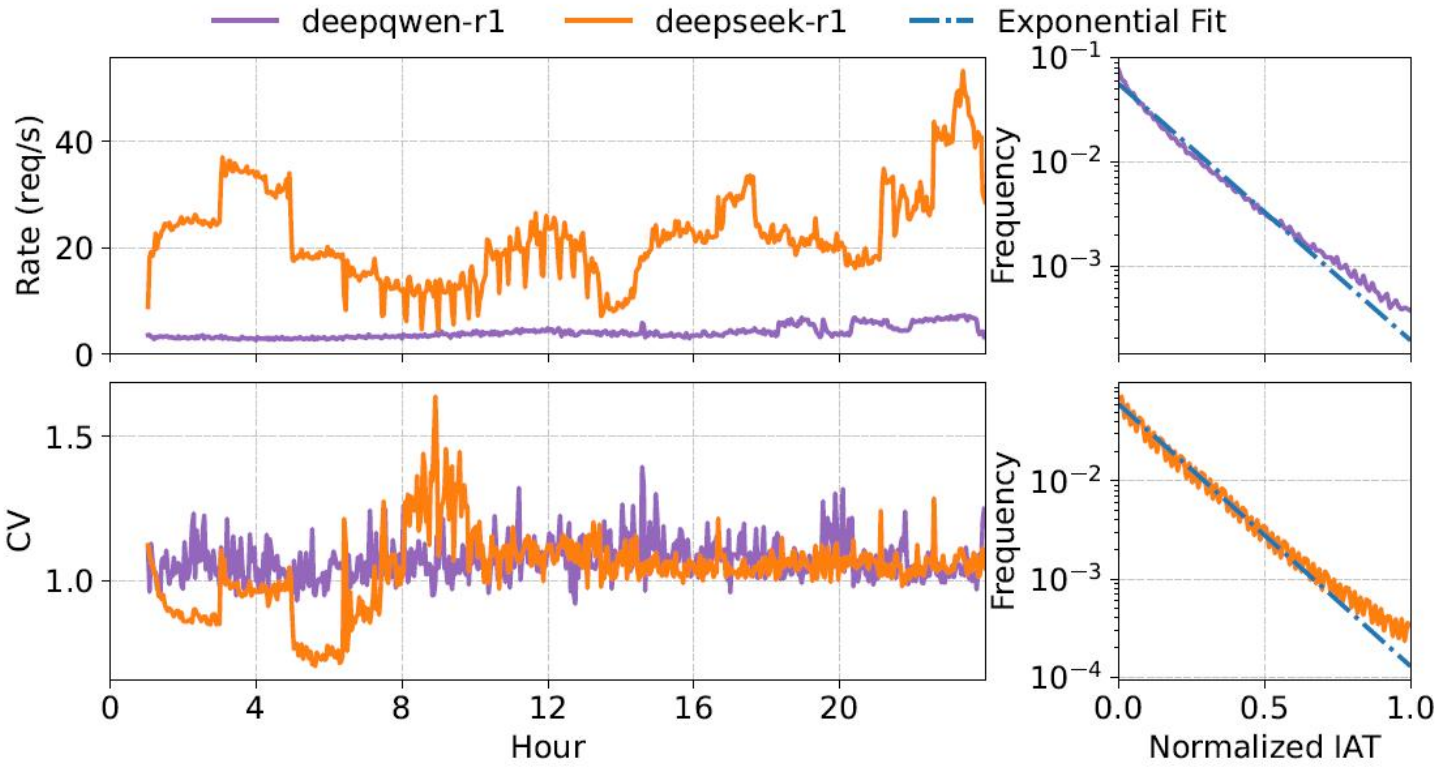
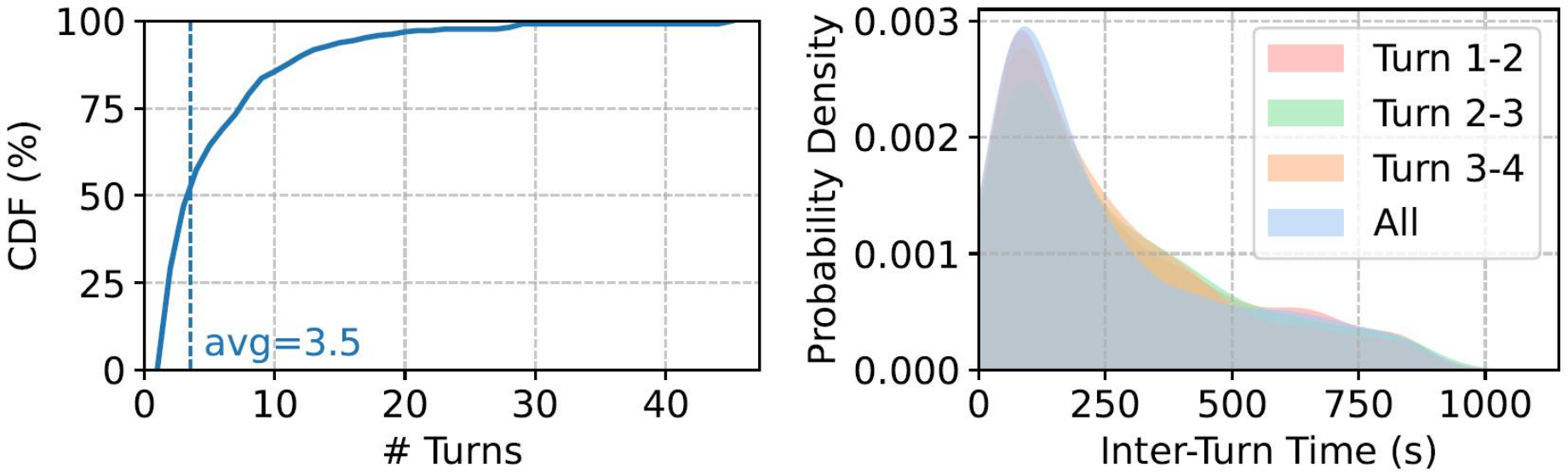


Figure 14. Characterization of request arrival patterns in `deepseek-r1` and `deepqwen-r1`. *Left*: Rate and burstiness shifts over a day. *Right*: Normalized inter-arrival time distributions.

- Exponential distribution
- Less bursty arrivals（compared with Non-reasoning model in Figure 2）

Characterizing multi-turn conversations



(a) CDF of conversation lengths. **(b)** PDF of inter-turn times.

Figure 15. Characterization of conversations in `deepseek-r1`.

- average conversation lengths = 3.5
- Similar distribution of inter-turn time

Characterizing Reasoning Workloads

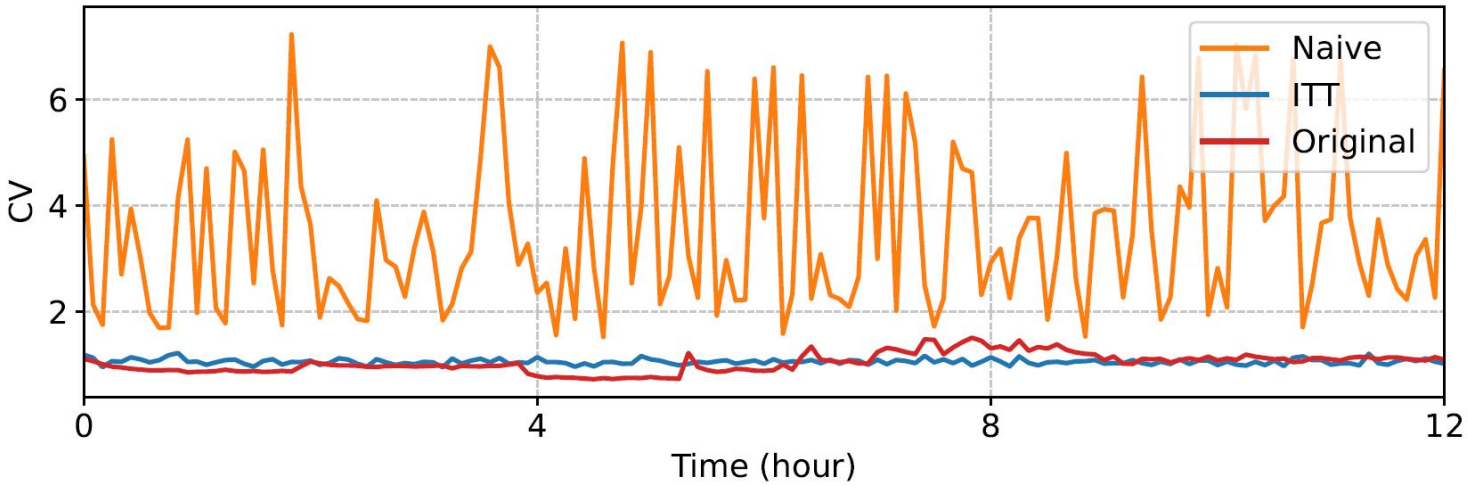


Figure 16. Comparison of two upsampling methods for a workload containing only multi-turn requests.

multi-turn requests constitute almost 10% of the deepseek-r1 workload

Use two upsampling methods （scale from multi-turn requests workload to original workload）
Naive： agnostic about the conversations (Figure15b),scales the inter-arrival time
ITT： scaling the arrival time between conversations, leaving the ITT distribution unchanged

Finding 10: Request arrival in reasoning workloads is impacted by the reoccurring pattern of multi-turn conversations and appears less bursty.

Characterizing Reasoning Workloads

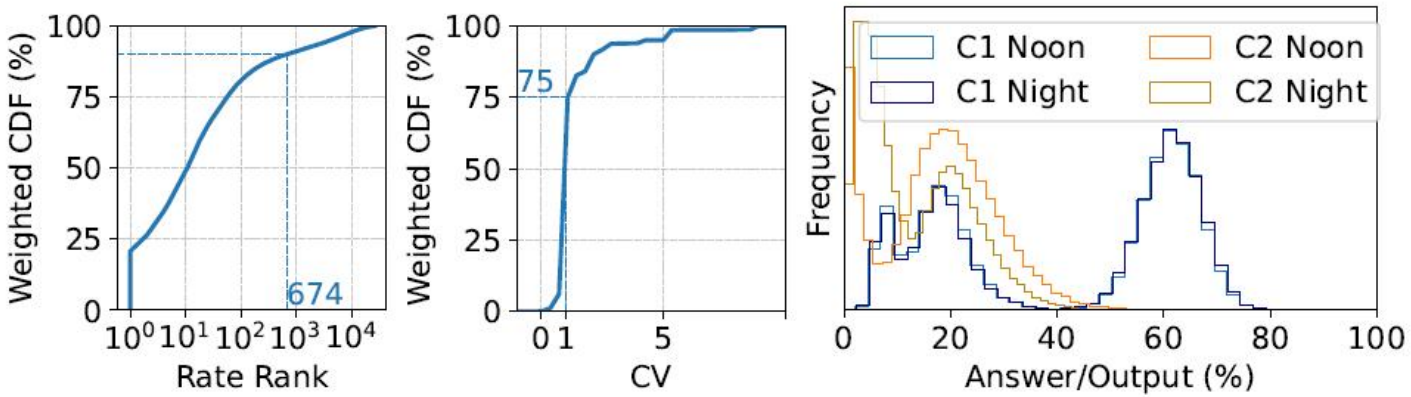


Figure 17. Client decomposition for `deepseek-r1`. (a) weighted CDF of client arrival rate. (b) weighted CDF of client burstiness. (c) output length breakdown of top clients (C1 and C2).

Among 25,913 clients

- Top clients in deepseek-r1 are shown to be less substantial
- The proportion of non-bursty clients is also significantly higher
- Bimodal distribution in the breakdown of request output lengths

Finding11 : Clients in reasoning workloads exhibit less skewed rates and less bursty arrivals, while also showing the bimodal pattern in terms of data distributions.

5

Workload Generation

Workload Generation：ServeGen Framework

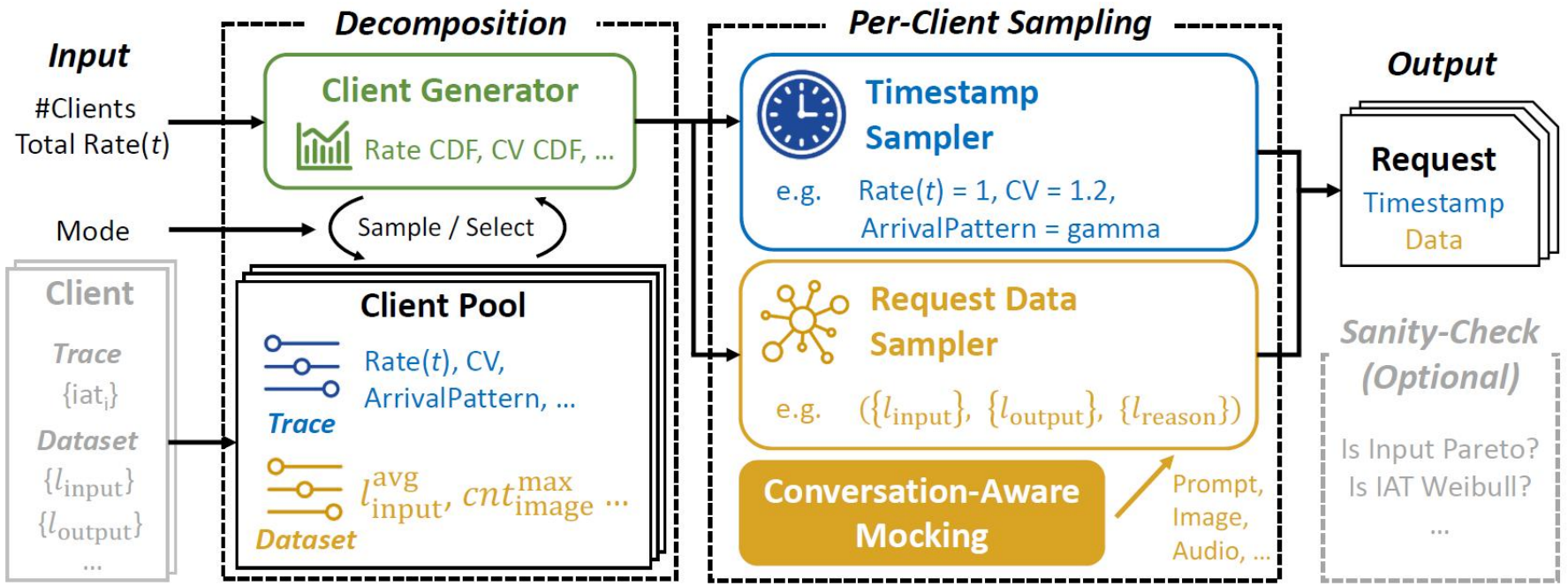


Figure 18. Overview of the ServeGen framework. The color gray indicates optional requirements; *e.g.*, users can still use ServeGen without providing additional client information.

https://github.com/alibaba/ServeGen

Workload Generation：ServeGen Framework

deepseek-r1 72 clients


```
{
  "0": {
    "input_tokens": "{45: 0.009666283084004603, 46: 0.011737629459148446, 47: 0.0039125431
    "output_tokens": "{64: 0.00023014959723820482, 77: 0.00023014959723820482, 82: 0.000466
    "reason_ratio": "{0.0: 0.0, 0.001996007984031936: 0.0, 0.003992015968063872: 0.0, 0.005
  },
  "21600": {
    "input_tokens": "{17: 0.001718213058419244, 45: 0.005154639175257732, 46: 0.0120274914
    "output_tokens": "{122: 0.001718213058419244, 130: 0.001718213058419244, 191: 0.001718
    "reason_ratio": "{0.0: 0.0, 0.001996007984031936: 0.0, 0.003992015968063872: 0.0, 0.005
  },
  "43200": {
    "input_tokens": "{}",
    "output_tokens": "{}",
    "reason_ratio": "{}"
  },
  "64800": {
    "input_tokens": "{}",
    "output_tokens": "{}",
    "reason_ratio": "{}"
  }
}
```


1	0	0	0		0	0
2	600	0	0		0	0
3	1200	0	0		0	0
4	1800	0	0		0	0
5	2400	0.085	6.69324386291654	Gamma	0.02232167101916883	73.01783087211976
6	3000	0.39	0.9137465123848778	Gamma	1.1977013397371243	2.074553310327594
7	3600	0.4216666666666667	1.0022801337212446	Gamma	0.9954552823877275	2.385870887198335
8	4200	0.43333333333333335	0.9140982433836469	Gamma	1.1967798028260763	1.911871171139863
9	4800	0.33666666666666667	1.0008881807649188	Gamma	0.9982260023520739	2.31995639433158
10	5400	0	0		0	0
11	6000	0	0		0	0
12	6600	0	0		0	0
13	7200	0	0		0	0
14	7800	0.35333333333333333	1.9996034406619674	Gamma	0.25009916934658716	11.012446779262042
15	8400	0.30833333333333335	1.152778512053871	Gamma	0.752503033285289	4.2769508383437875
16	9000	0.32666666666666666	0.8970360969242444	Weibull	1.1167491378227001	3.1569971349039734
17	9600	0.3333333333333333	1.0543935635041897	Gamma	0.8994862062683567	3.331600442916639

Request length distribution

chunk-0-dataset.json

Request Arrival Pattern

chunk-0-trace.csv

https://github.com/alibaba/ServeGen

Comparison of workload generation accuracy

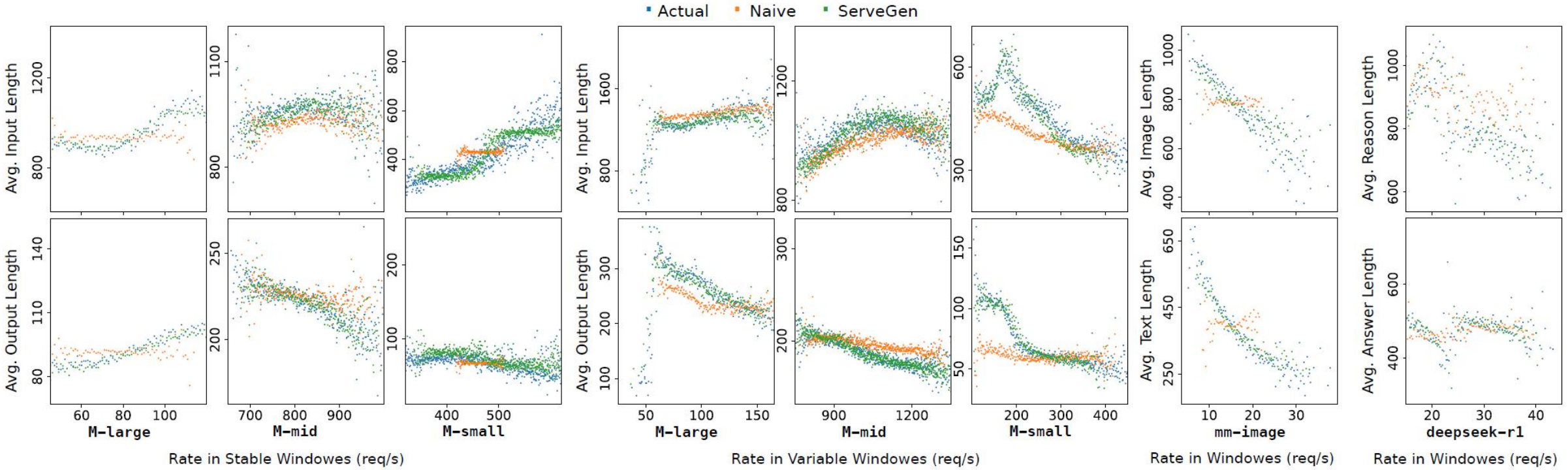
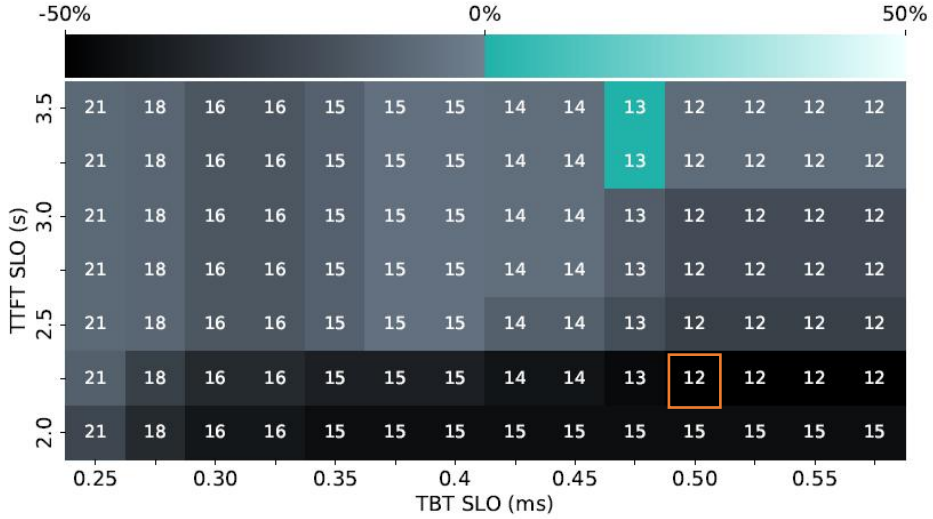


Figure 19. Comparison of workload generation accuracy.

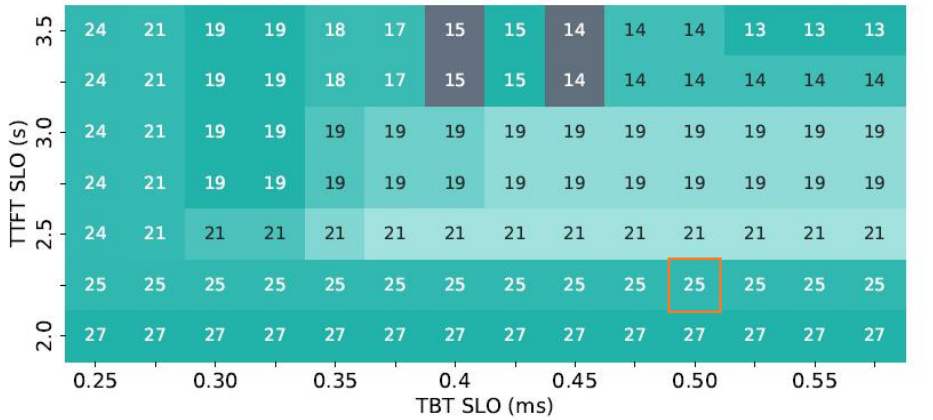
- ServeGen matches the actual plot much better

- Naïve barely capture the correlation between rates and data distributions
  (large or small short-term rates are likely caused by bursty top clients, and the workload data distributions are expected to shift correspondingly toward or away from the client data distributions.)

UseCase: Instance Provisioning



(a) Provisioning with NAIVE.



(b) Provisioning with ServeGen.

Figure 20. Provisioning results using the NAIVE approach and ServeGen. In each cell, the number indicates the provisioned instances, while the color shows the over-provisioning percentage.

- Workload:
10-minute period of M-large comprising 30,000 requests (generated by NAÏVE and Servegen approach)

- Each Instance：
2 * NVIDIA A100 (80GB) GPUs
vLLM(Pipeline parallelism) + Qwen2.5-14B model

- Method:
benchmark one instance using both NAÏVE and Servegen approach, determine the maximum rate that meets SLO targets (P99), and derive the required instance count

- Results:
TTFT 2.25s TBT 0.5s
ServeGen: 25 instances (4% over the actual)
Naïve: 12 instances (50% under-provisioning)

6

Discussion

Key takeaway

1. Real-world LLM Serving Workload request arrivals exhibit a complex bursty pattern that goes beyond any single stochastic process

2. Most nondeterministic patterns in request arrivals (e.g., bursts) and length distributions (e.g., high dynamics over time) are caused by several top clients, while the behaviors of most clients remain stable and predictable

3. Multi-turn conversations have Similar distribution of inter-turn time

4. A significant portion of TTFT stems from preprocessing (i.e., downloading, normalization, and encoding). highlights the importance of conducting full-stack optimizations

Discussion

1. LLM serving with plugin calls/function call (i.e., web searches, database queries, or calling external APIs):

For future work

2. prefix KV cache:

Not investigated, due to user privacy considerations.

3. Generalization:

Could record customize workload and reproduce.

Thank you !

13. Workload Generation：ServeGen Framework

	ServeGen	**Naive**
Sampling method	Resamples based on **client decomposition**	Resamples the **entire workload as a whole**, ignoring individual clients.
Realism of generated workload	Produces more **realistic** workloads — scatter plots closely match actual workload.	Workload diverges from real patterns in both rate and data distribution.
Variability (request rate)	Accurately reflects real variability — includes wide horizontal spread in scatter plots.	Often **less variable** in stable periods despite global burstiness.
Rate–data correlation	Captures correlation between **request rate and data distribution**, as real top clients cause correlated bursts.	Fails to capture such correlation — rate and data distribution are independent.