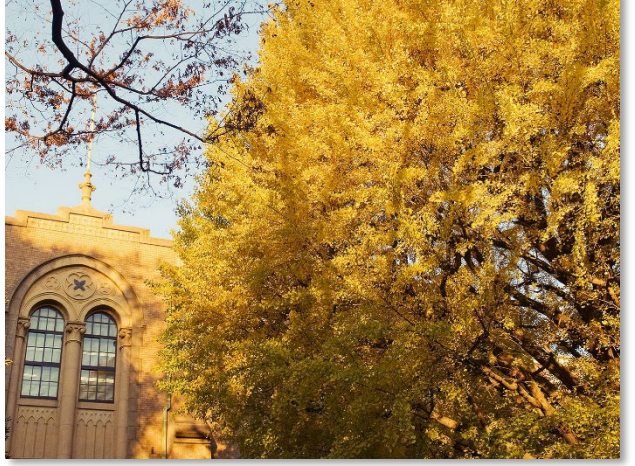
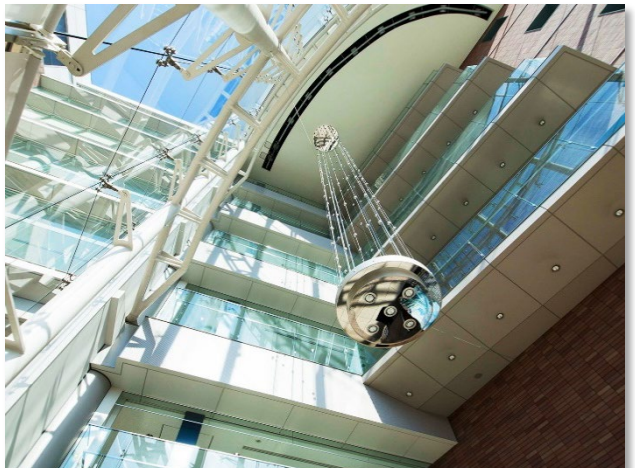


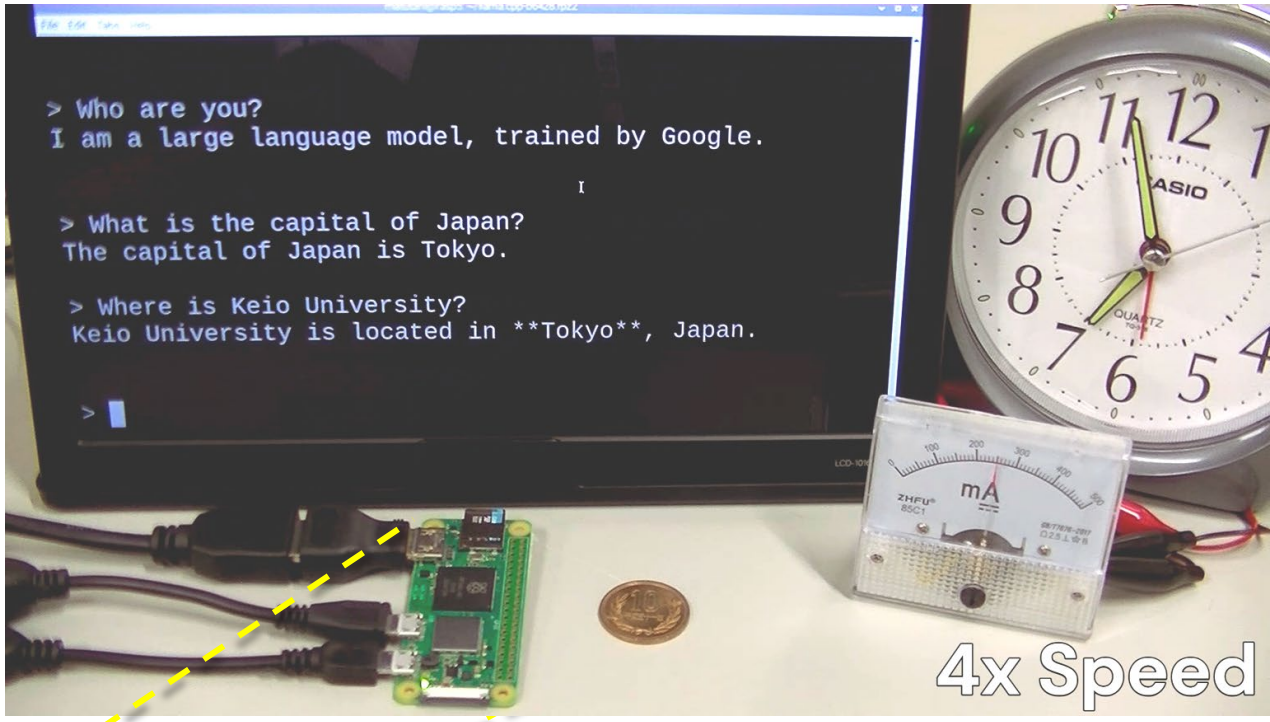
Accelerating Local LLMs on Resource-Constrained Edge Devices via Distributed Prompt Caching

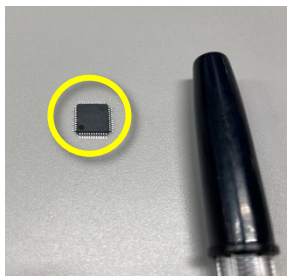
Hiroki Matsutani, Naoki Matsuda, Naoto Sugiura (Keio University, Japan)

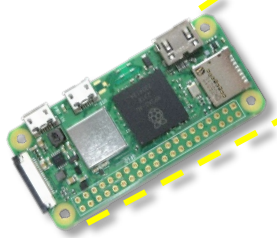
Motivation: Local LLM on low-cost devices

- **Latency in tens of seconds ☹**
  - RPi Zero 2W (price: $15)
  - Gemma3-270M model
  - Llama.cpp
  - Functional, but very slow…
- **Widespread low-cost LLMs**
  - What if KV cache [1] is shared among edge devices?

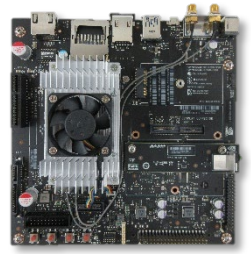



20-30mW


0.2-1W


2-10W

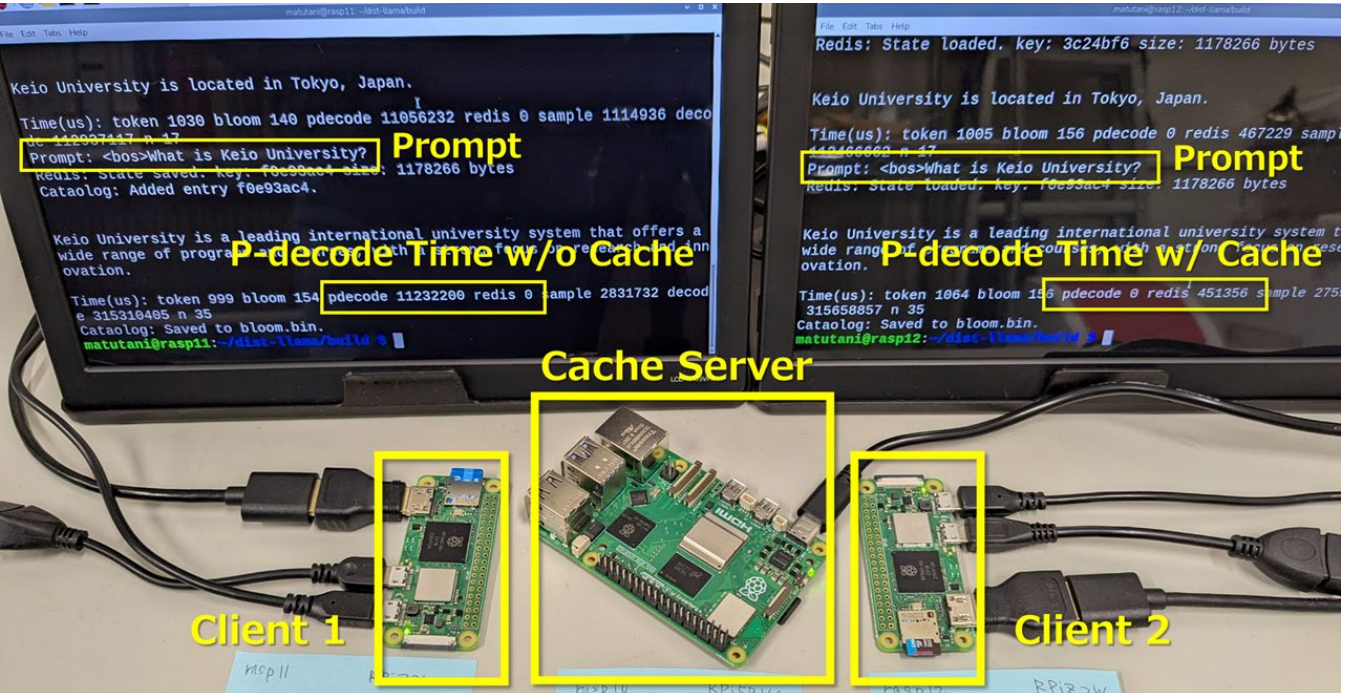

15W+

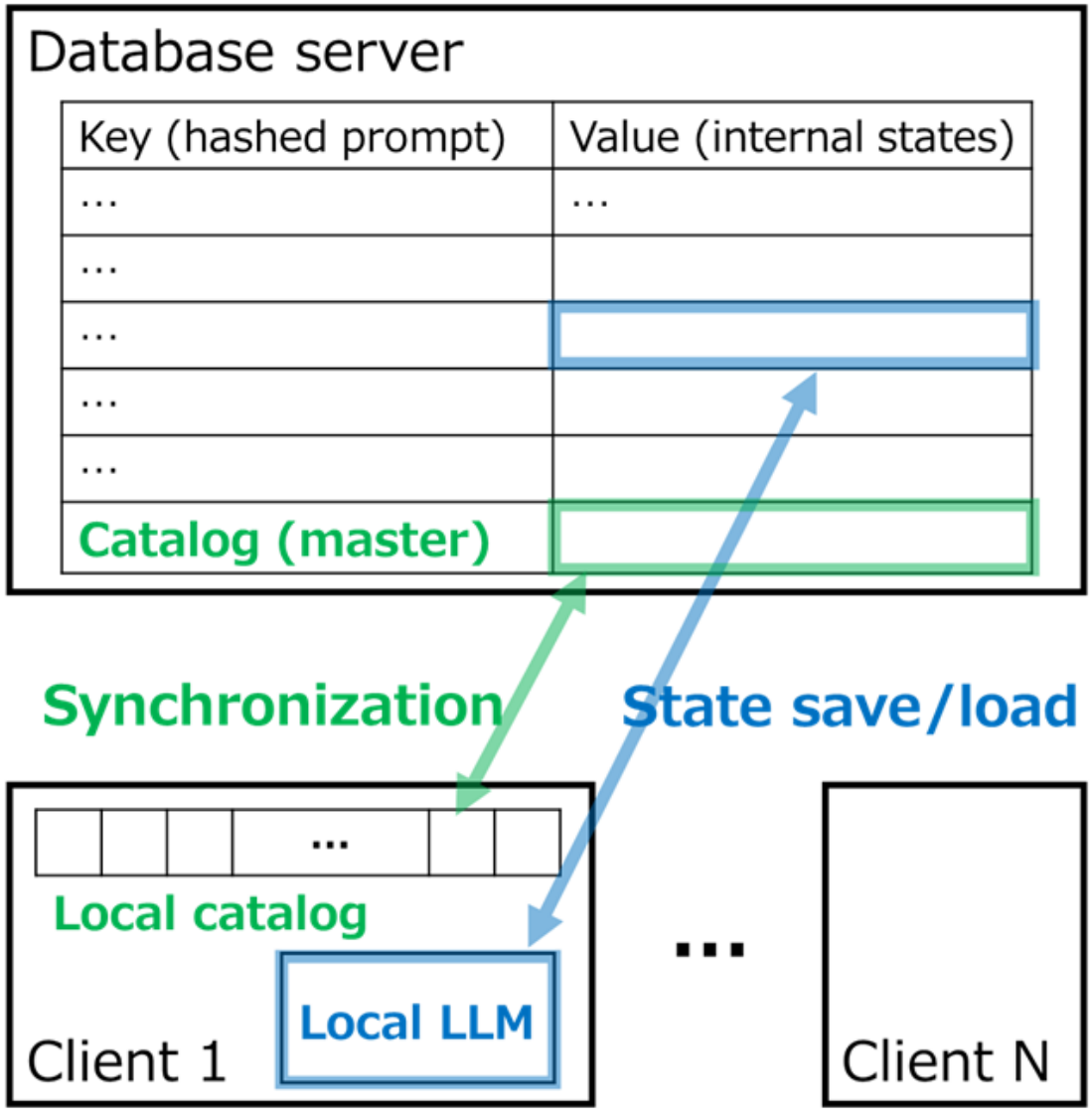

200W+

[1] In Gim et al., "Prompt Cache: Modular Attention Reuse for Low-Latency Inference", MLSys'24.

Approach: Sharing KV cache w/ edge devices

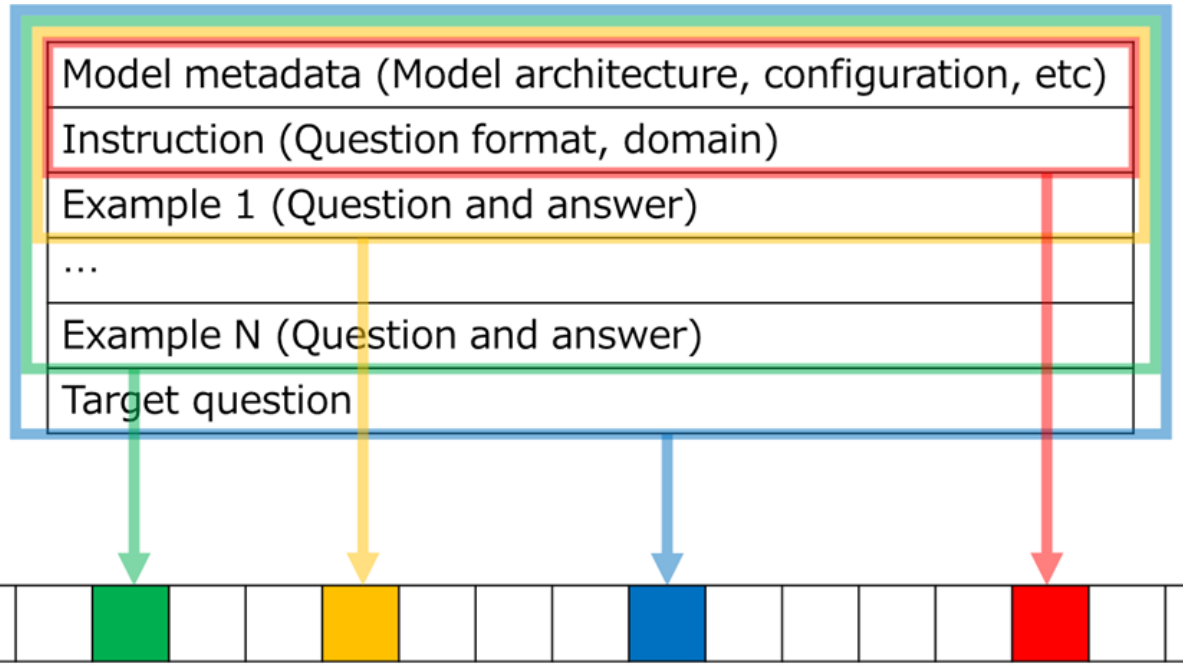
- **Client-server architecture**
  - **Server: stores KV cache**
  - **Clients: use KV cache for local LLM**
- **A critical issue**
  - ***How can clients find out if the server has the required KV cache?***



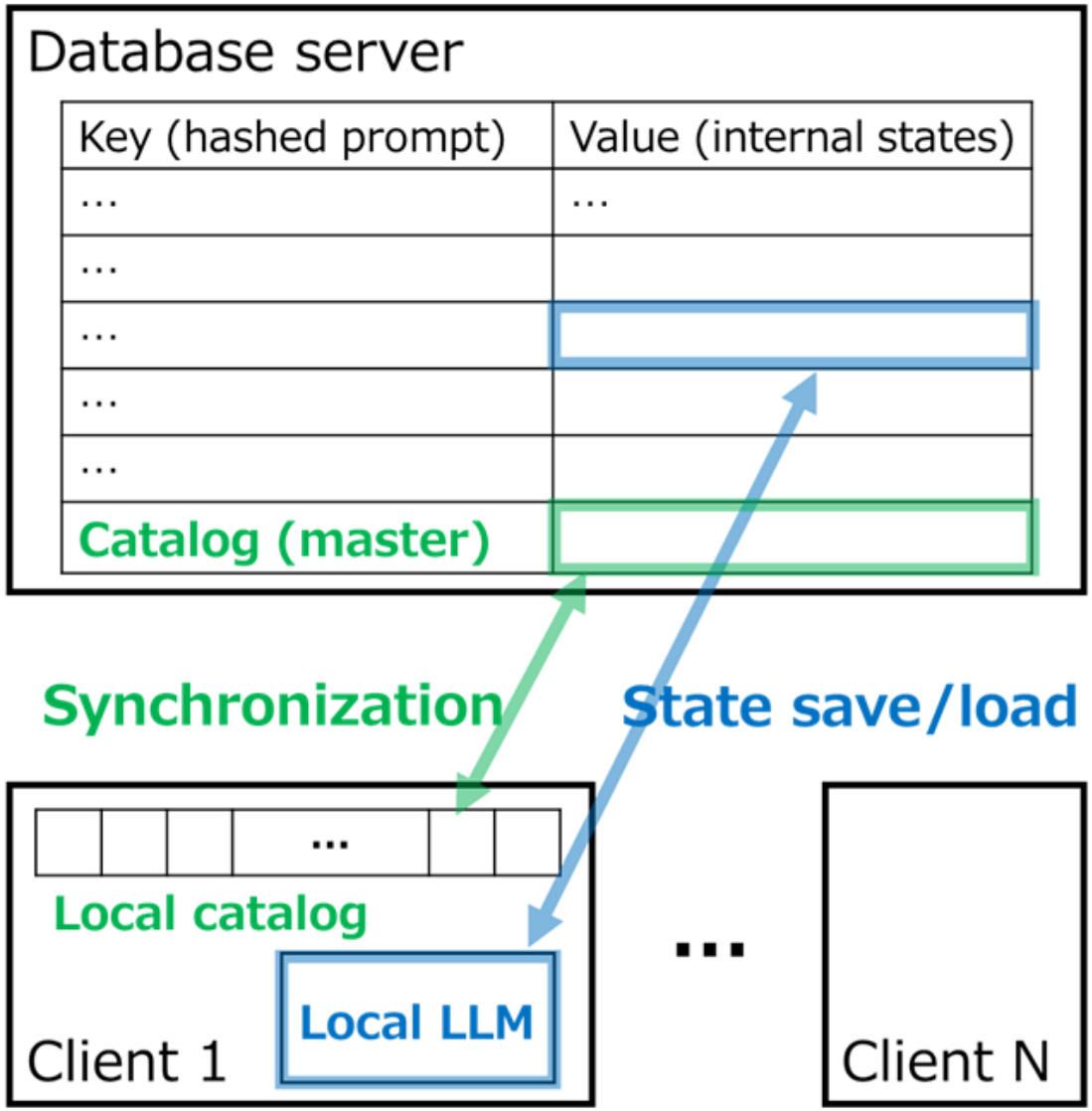


Approach: Sharing KV cache w/ edge devices

- **Bloom filter-based "catalog" to summarize stored KV cache**
  - **Server: maintains the catalog and periodically distributes it to clients**
  - **Clients: determine if the server has required KV cache w/ the catalog**



Catalog example for multi-choice question



Approach: Sharing KV cache w/ edge devices

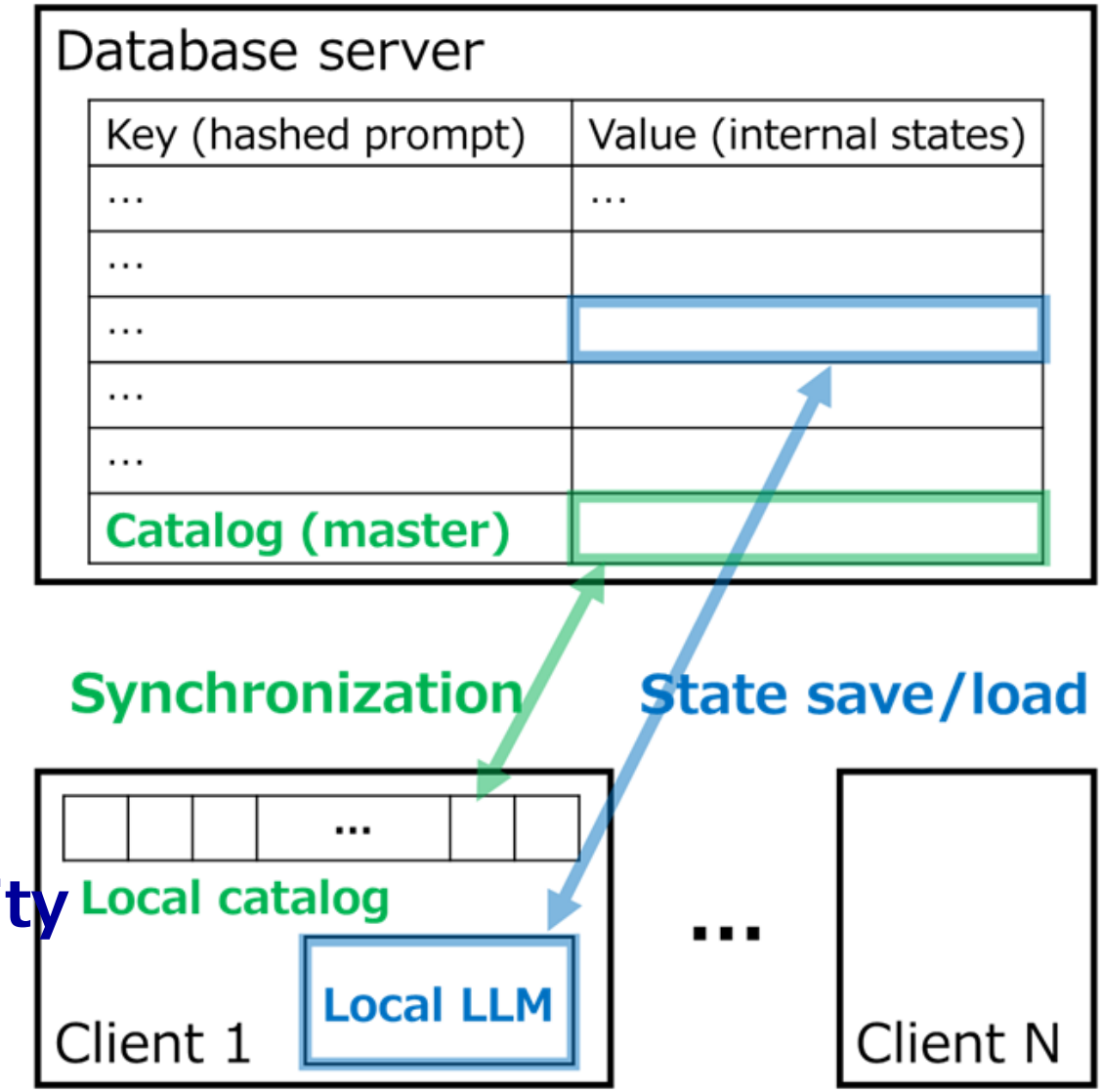
- Bloom filter-based "catalog" to summarize stored KV cache
  - Server: maintains the catalog and periodically distributes it to clients
  - Clients: determine if the server has required KV cache w/ the catalog

Processing flow

1. Prompt tokenization
2. **Prompt decoding**
3. Output decoding + sampling

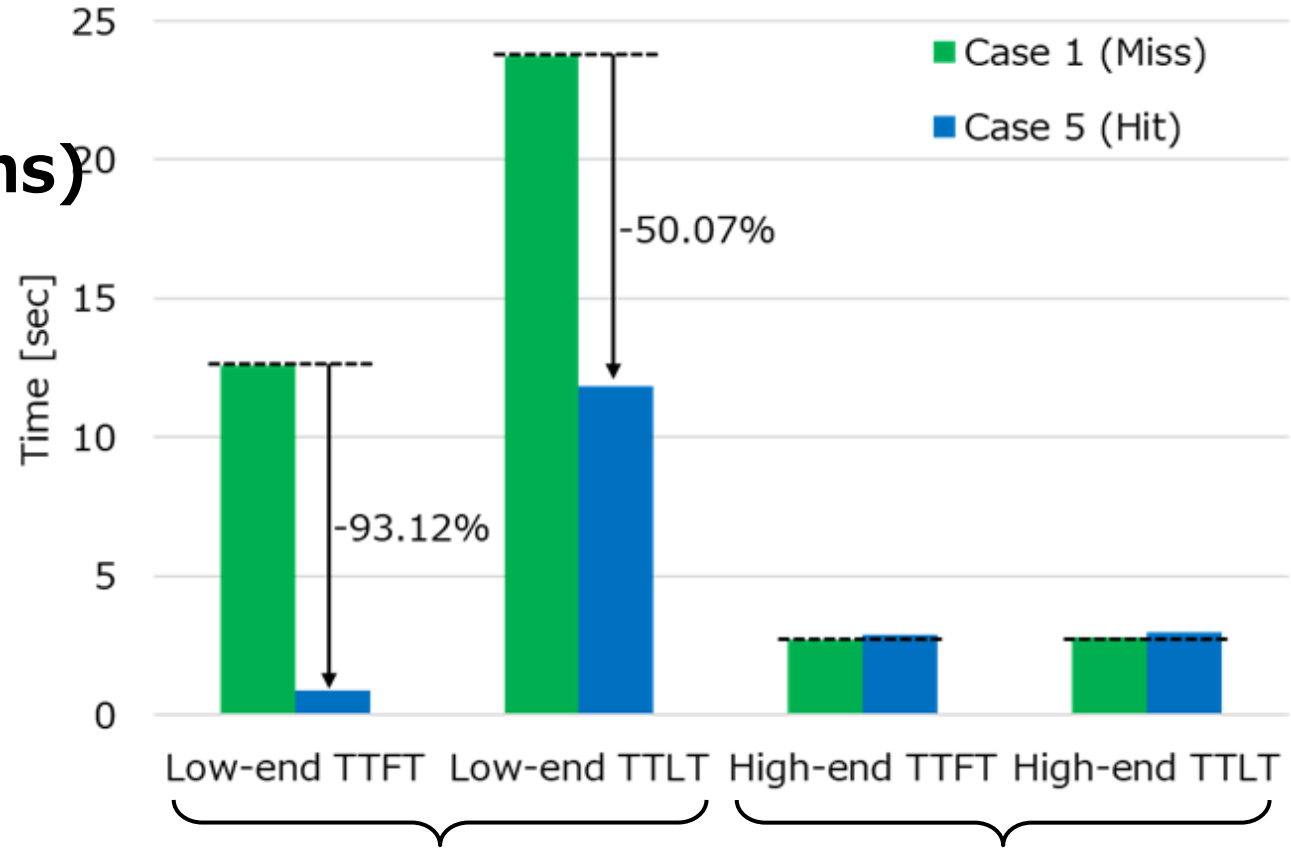
⇩

1. Prompt tokenization
2. **Catalog check for server's KV availability**
3. **Remote DB access (KV retrieval)**
4. Output decoding + sampling



Latency results: Full hit vs. cache miss

- **Evaluation setup**
  - MMLU bench (4-choice questions)
  - TTFT (time to first token)
  - TTLT (time to last token)
- **Low-end case**
  - Raspberry Pi Zero 2W
  - Gemma3-270M
  - 1 example (average 65 tokens)
- **High-end case**
  - Raspberry Pi 5B
  - Gemma3-1B
  - 5 examples (average 334 tokens)



RPi Zero2W

RPi 5B

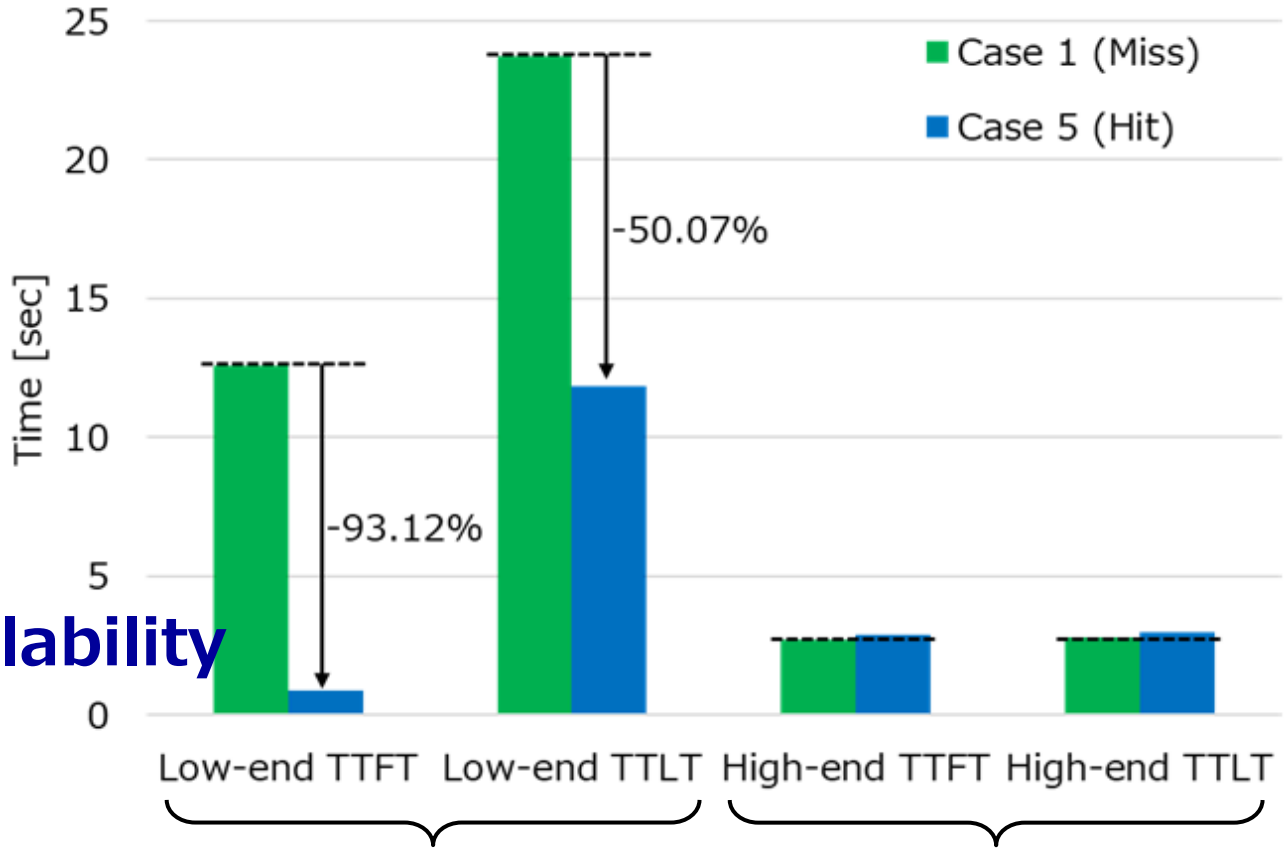
Latency breakdown: Full hit vs. cache miss

Processing flow

1. Prompt tokenization
2. **Prompt decoding**
3. Output decoding + sampling

⇩

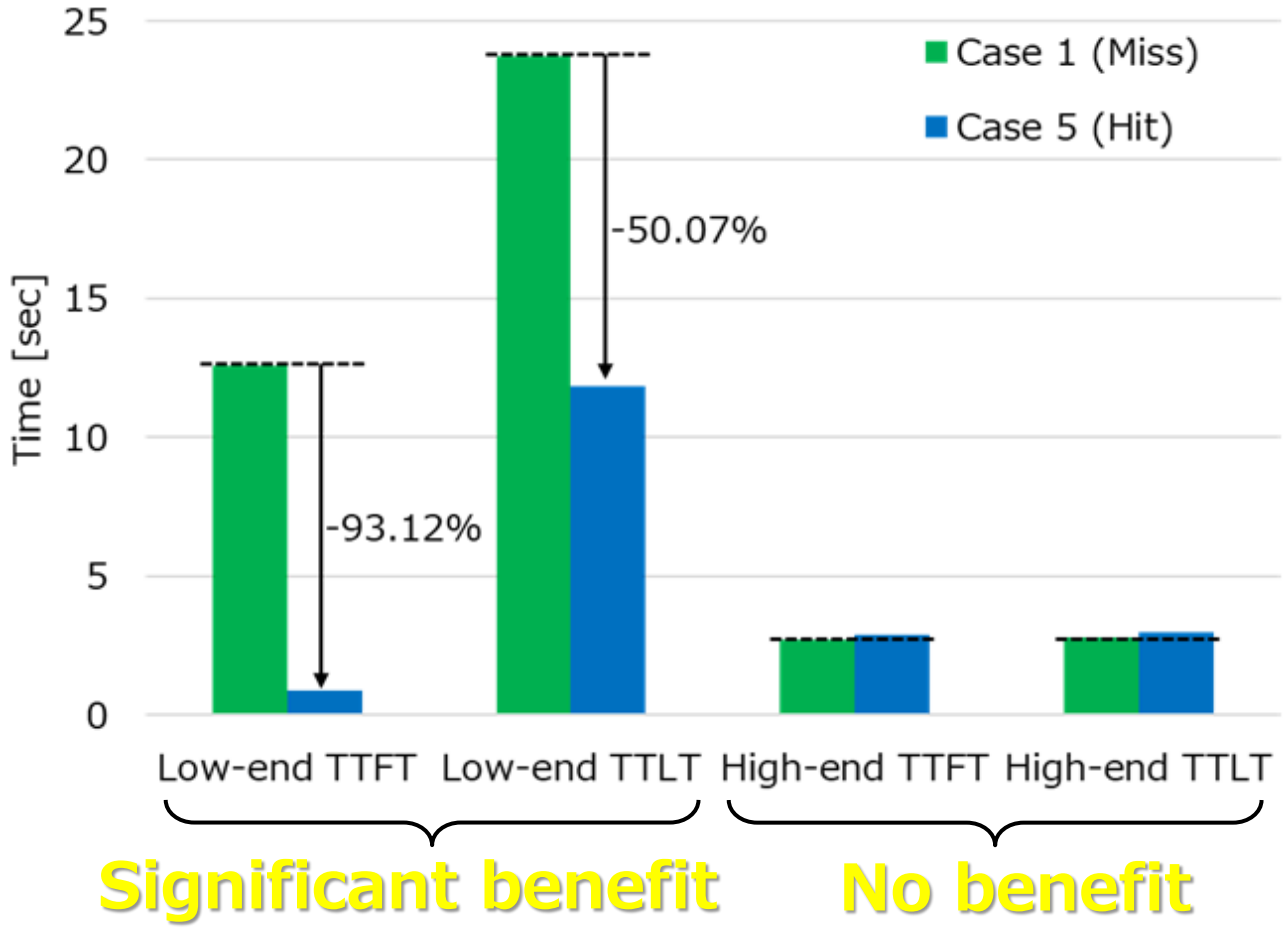
1. Prompt tokenization
2. **Catalog check for server's KV availability**
3. **Remote DB access (KV retrieval)**
4. Output decoding + sampling



	Token	Bloom	P-decode	Redis	R-decode	Sample	N	# tokens	State size [MB]
Low-end (Case 1)	3.46	0.30	12580.85	2.42†	11061.04	95.69	1	65.27	2.25
Low-end (Case 5)	3.46	0.19	0.00	861.92	10904.67	84.82	1	65.27	2.25
High-end (Case 1)	1.61	0.00	2688.17	7.84†	72.59	1.45	5	334.11	9.94
High-end (Case 5)	1.56	0.00	0.00	2887.04	78.12	1.67	5	334.11	9.94

Latency breakdown: Full hit vs. cache miss

- **Big benefit for low-end case**
  - Prefill (P-decode) takes 14.6x longer than remote Redis access overhead
  - Substantially improves TTFT
- **No benefit for high-end case**
  - Remote Redis access overhead is comparable to Prefill (P-decode) time

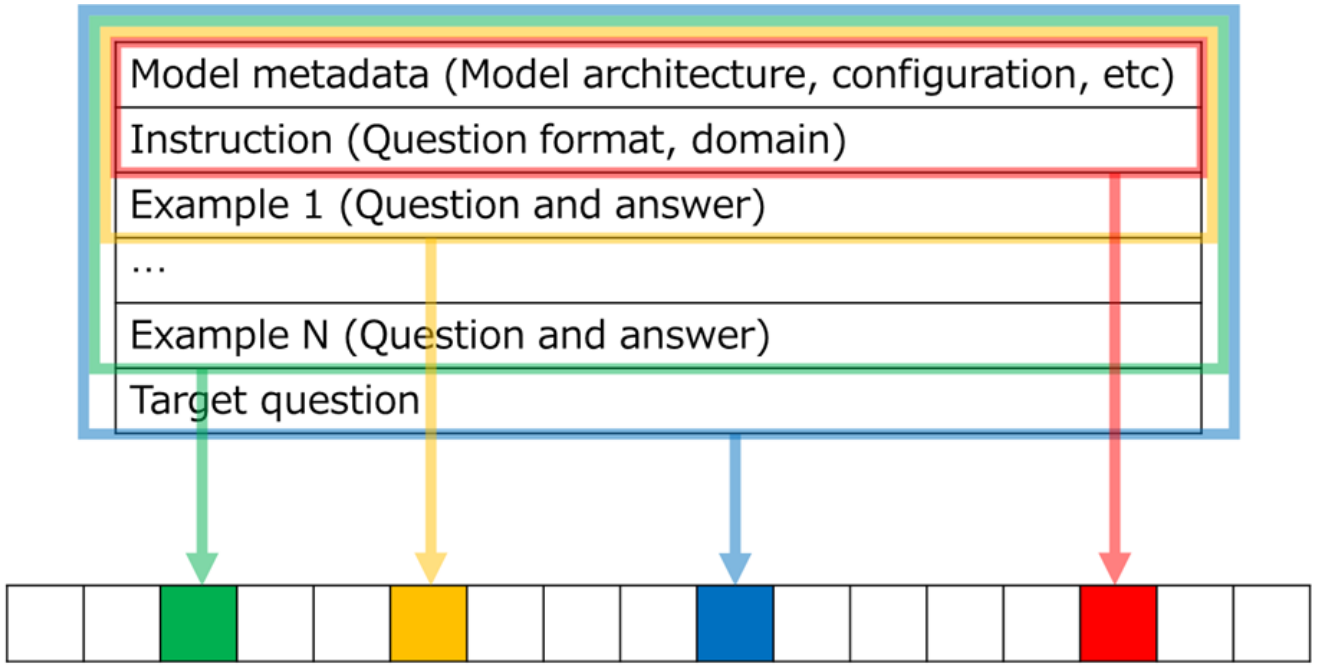


	Token	Bloom	P-decode	Redis	R-decode	Sample	N	# tokens	State size [MB]
Low-end (Case 1)	3.46	0.30	12580.85	2.42†	11061.04	95.69	1	65.27	2.25
Low-end (Case 5)	3.46	0.19	0.00	861.92	10904.67	84.82	1	65.27	2.25
High-end (Case 1)	1.61	0.00	2688.17	7.84†	72.59	1.45	5	334.11	9.94
High-end (Case 5)	1.56	0.00	0.00	2887.04	78.12	1.67	5	334.11	9.94

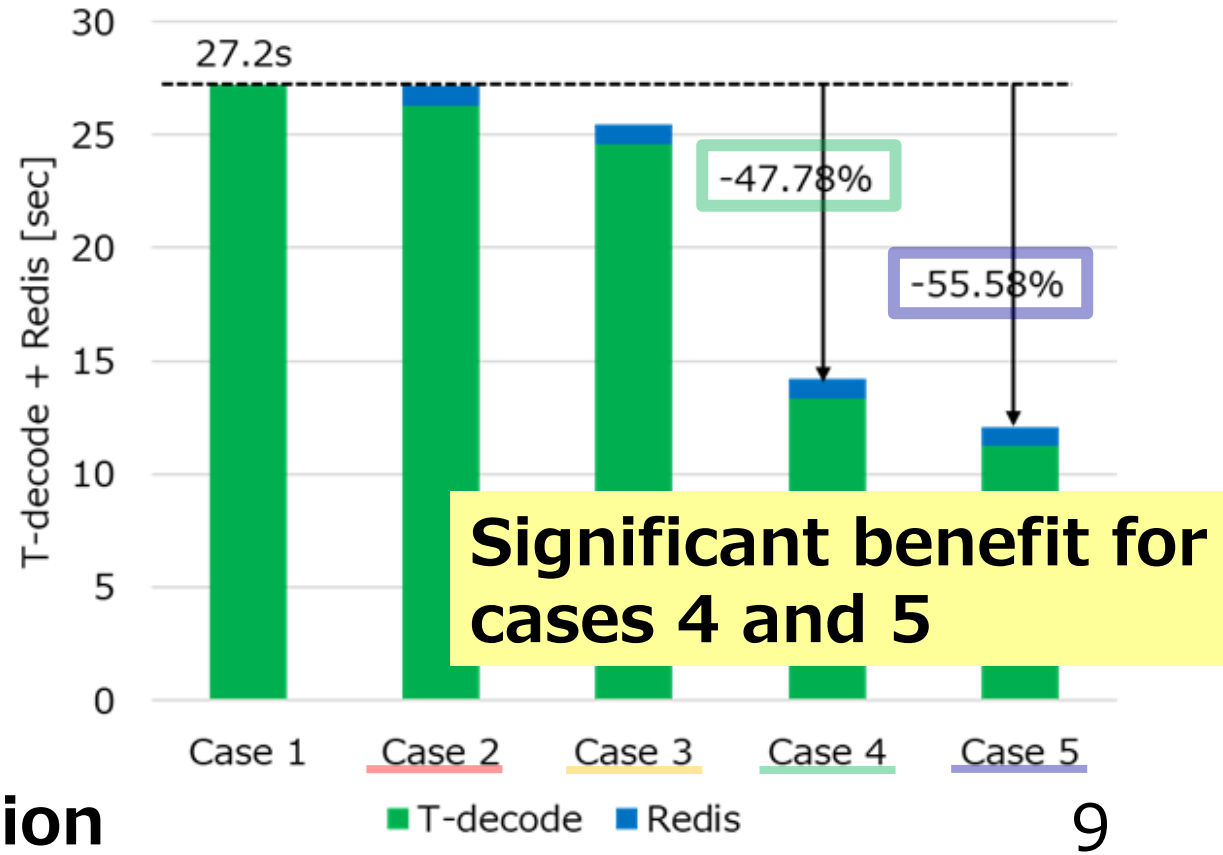
Latency results: Partial hit cases

- **Partial matches on KV cache**
  - **Case 1: miss**
  - **Case 5: full hit**
  - **Case 2: up to instruction hits**
  - **Case 3, 4: 1 or 5 examples hit**

	# matched	% matched	T-decode
Low-end (Case 1)	1	0.25	27203.96
Low-end (Case 2)	10	2.47	26288.23
Low-end (Case 3)	57	14.07	24590.09
Low-end (Case 4)	340	83.95	13344.96
Low-end (Case 5)	405	100.00	11220.95



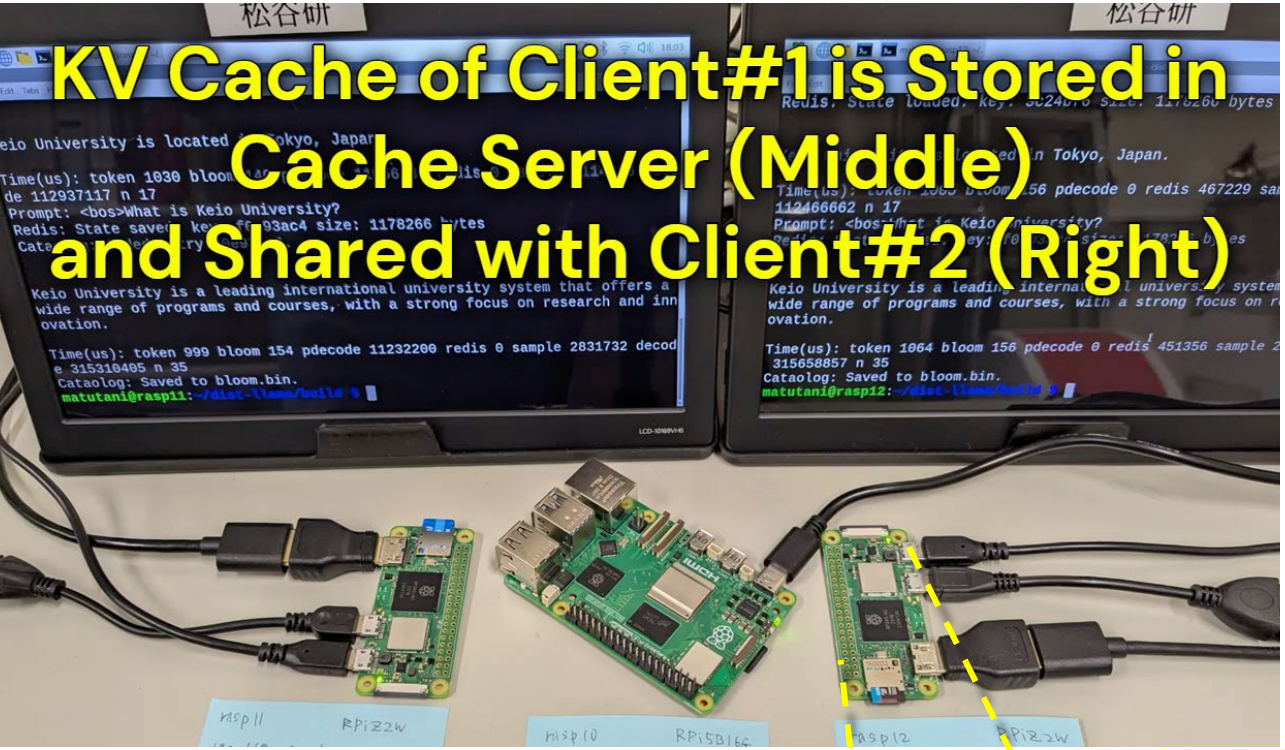
Catalog example for multi-choice question

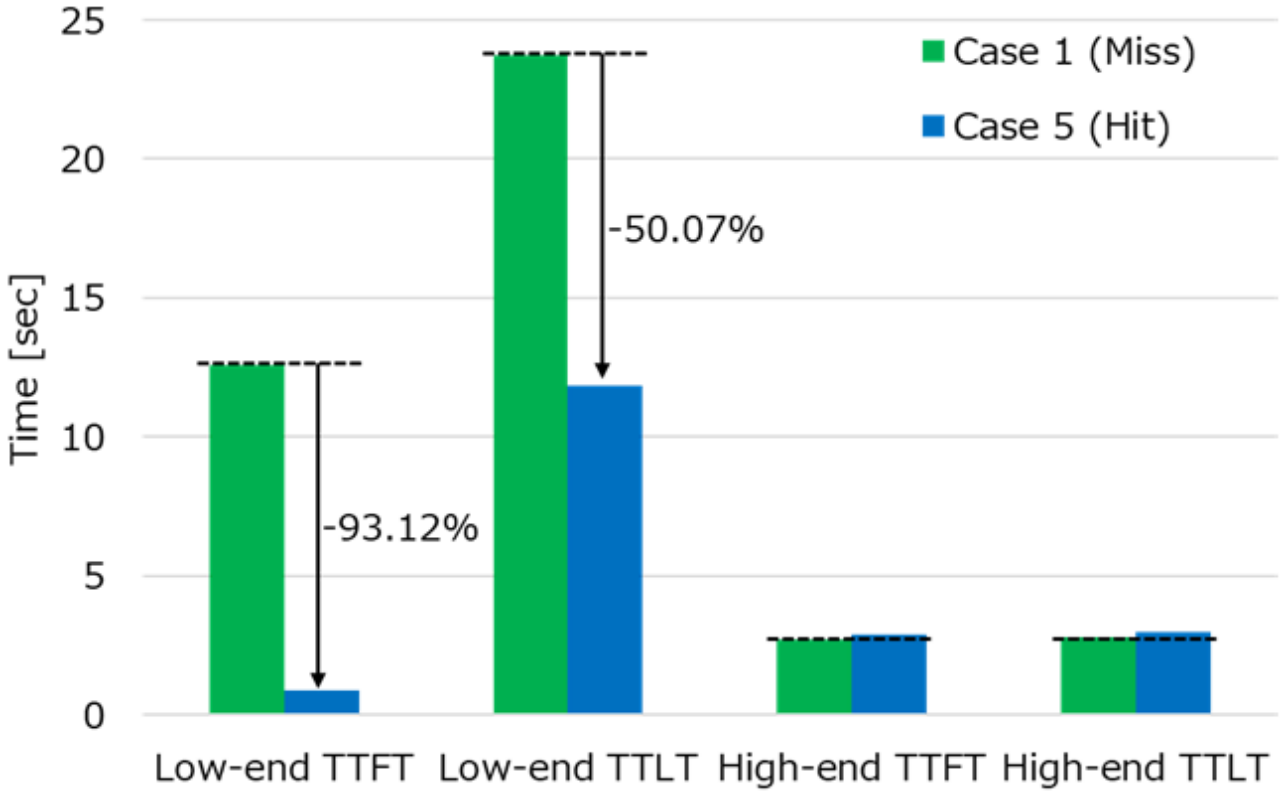


Significant benefit for cases 4 and 5

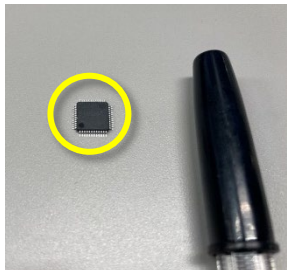
Summary: Sharing KV cache w/ edge devices

We introduce KV cache sharing to accelerate local LLM on low-cost edge





Results demonstrate significant benefit for RPi Zero2W but no benefit for RPi 5B



20-30mW

0.2-1W