Media Over QUIC
Internet-Draft
Intended status: Informational
Expires: 12 October 2025

H. Shi
Huawei Technologies
S. Yue
China Mobile

10 April 2025

KVCache over MoQT
draft-shi-moq-kvcache-01

Abstract

Large language model (LLM) inference involves two stages: prefill and decode. The prefill phase processes the prompt in parallel, generating the KVCache, which is then used by the decode phase to produce tokens sequentially. KVCache can be reused if the model and prompt is the same, reducing computing cost of the prefill. However, its large size makes efficient transfer challenging. Delivering these over architectures enabled by publish/subscribe transport like MoQT, allows local nodes to cache the KVCache to be later retrieved via new subscriptions, saving the bandwidth. This document specifies the transmission of KVCache over MoQT.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Media Over QUIC mailing list (moq@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/moq/.

Source for this draft and an issue tracker can be found at https://github.com/VMatrix1900/draft-moq-kvcache.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 October 2025.

Table of Contents

1. Introduction: KVCache in LLM inference

The inference process of large language models is typically divided into two distinct stages: prefill and decode. The prefill phase processes the input prompt in parallel, generating a KVCache, which serves as an essential input for the decode phase. The decode phase then utilizes the KVCache to generate output tokens sequentially, one at a time. Prefill is a computationally intensive process, whereas decoding is constrained by memory bandwidth. Due to their differing resource requirements, prefill and decode processes are often deployed on separate computing clusters using different hardware chips optimized for computational performance in prefill nodes and memory bandwidth efficiency in decode nodes, with KVCache transferred between them.

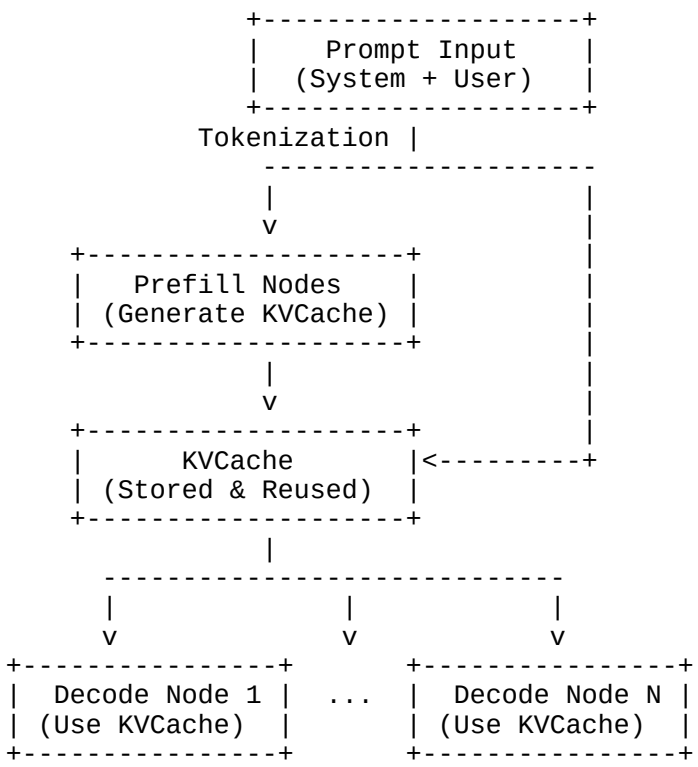

Figure 1: LLM inference process

KVCache is significantly large, with a single token requiring 160KB for a 70B model(8bit quantization). For a prompt of 1000 tokens, the KVCache size reaches 160MB. To reduce the size of KVCache, various quantization and compression algorithm are proposed such as [CacheGen]. Furthermore, KVCache can be reused across sessions if derived from the same prompt and model, as shown in Figure 1. The most basic reuse strategy is prefix caching, where KVCache is shared among prompts with a common prefix. More advanced methods, such as [CacheBlend], improve reuse efficiency by selectively reusing KVCache beyond prefix matching. To minimize transmission costs, a publish/subscribe architecture is required to distribute KVCache. This document defines how to send KVCache over MoQT[I-D.ietf-moq-transport].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the following terms:

* LLM: A large language model (LLM) that utilizes the attention mechanism to process and generate text efficiently by capturing long-range dependencies within input sequences.

* KVCache: A key-value cache storing intermediate representations used in LLM inference.

* Prompt: A prompt consists of two parts: the system prompt and the user prompt. The system prompt is predefined by the LLM model developer to guide the model’s behavior, while the user prompt is provided dynamically by the user to specify the task or request.

* Token: The smallest unit of processing in LLM inference, typically representing a word or subword.

3. KVCache Data Model

The KVCache data model is structured as follows.

Naming: The Track Namespace consisting of following tuples (moq://kvcache.moq.arpa/v1/),(modelName), (prompt) is defined in this specification. The track name identifies the compression level for the KVCache. Thus, a track name can be identified with the tuple (<compression>) and the full track name having the following format (when represented as a string):

moq://kvcache.moq.arpa/v1/<modelName>/<compression>

Following compressions are defined in this specification, along with their size:

Compression	Description	Size per Weight
FP16	Quantized using FP16	2 bytes
BF16	Quantized using BF16	2 bytes
FP8	Quantized using FP8	1 byte
Int8	Quantized using Int8	1 byte
FP4	Quantized using FP4	0.5 byte
Int4	Quantized using Int4	0.5 byte
AC (5x)	Compressed using Arithmetic Coding (5x ratio)	Variable

Table 1: Compression of KVCache

Group ID: Normally the tokens are split into chunks of uniform length(typical value is 128). The KVCache are organized into groups corresponding into token chunks. The ID of the group represents the index of a token group within the KVCache.

Object ID: An identifier for a specific token within a group.

Object Payload: The content of the KVCache, which varies based on the compression algorithm used for storage and transmission.

4. Security Considerations

TBD

5. IANA Considerations

TBD

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

6.2. Informative References

[CacheBlend] "CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion", 2024, <https://arxiv.org/abs/2405.16444>.

[CacheGen] "CacheGen: Fast Context Loading for Language Model Applications via KV Cache Streaming (SIGCOMM24)", 2024, <https://github.com/UChi-JCL/CacheGen>.

[I-D.ietf-moq-transport] Curley, L., Pugin, K., Nandakumar, S., Vasiliev, V., and I. Swett, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-10, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-10>.

Authors' Addresses

Hang Shi
Huawei Technologies
China
Email: shihang9@huawei.com

Shengnan Yue
China Mobile
China
Email: yueshengnan@chinamobile.com