Model loading from network storage and Local disk

Note:
1. A100 80Gb used for all experiments and network storage used for the Tensorizer & Hf loading.
2. *Results are average of 10 requests (*(Model loading time)*)*

Using Classic hugging face loading

Local disk(cache):
1x A100 80GB - Hf_CACHE

Llama-2-7b-chat-hf	Model loading time: 9.44 seconds
Llama-2-13b-chat-hf	Model loading time: 14.69 seconds

Network-Storage: Region- US-KS-1

Llama-2-7b-chat-hf	Model Loading TIme: 12.12 seconds.
Llama-2-13b-chat-hf	Model Loading time: 17.31seconds

Notes:

1. Major time taking factor is serialization/deserialization that occurs when loading from disk to the GPU.
2. Single thread for transferring tensors to GPU. (potential cause).
3. Classic hugging face loading uses torch.load() under the hood.
Which is loading from disk to cpu and then using map_location to GPU.
https://pytorch.org/docs/stable/generated/torch.load.html

4.Serialization speed matters. RAM and gpu bandwidth are so fast that they don't matter so using the fastest possible serializer is best.

coreweave/Tokensier lib - valid for both transformers and diffusers.
https://github.com/coreweave/tensorizer

Using Tensorizer (Faster serializer)

- *Results are average of 10 requests (*(Model loading time)*)*

Local disk(cache):

1x A100 80GB - Hf_CACHE

Llama-2-7b-chat-hf	Model Loading TIme: 4.72s
Llama-2-13b-chat-hf	Model Loading TIme: 8.90s

Network-Storage: Region- US-KS-1

Llama-2-7b-chat-hf	Model Loading TIme: 5.31s
Llama-2-13b-chat-hf	Model Loading TIme: 9.25s