



CONFERENCE

—NORTH AMERICA 2026—

Faster LLM Serving Startup with ***fastsafetensors***

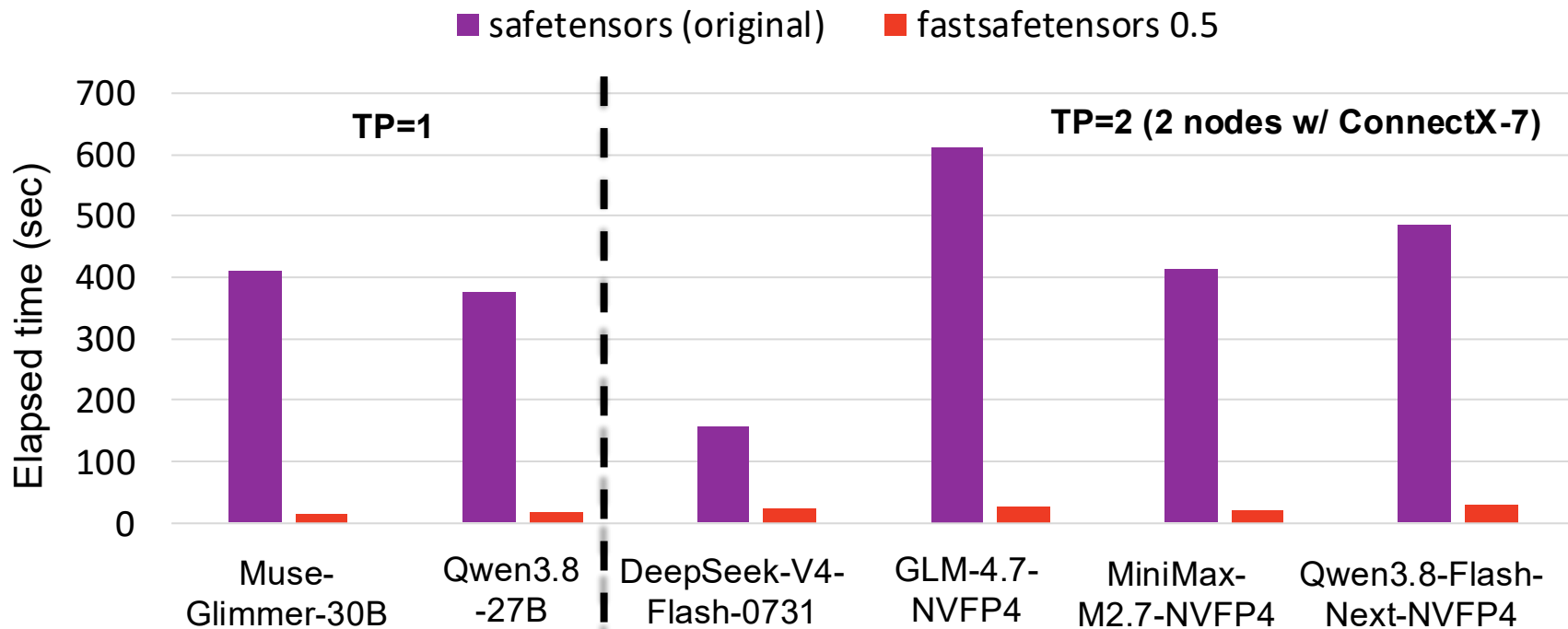
Takeshi Yoshimura, IBM Research - Tokyo

```

(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:347]
(APIServer pid=1386788) INFO 10-07 01:55:01 [api_utils.py:286] non-default args: {'model_tag': 'Qwen/Qwen3.8-72B', 'model': 'Qwen/Qwen3.8-72B', 'max_model_len': 32768, 'enforce_eager': True, 'load_format': 'safetensors', 'safetensors_load_strategy': 'lazy', 'gpu_memory_utilization': 0.9}
(APIServer pid=1386788) INFO 10-07 01:55:02 [model.py:699] Resolved architecture: Qwen3_5ForConditionalGeneration
(APIServer pid=1386788) INFO 10-07 01:55:02 [model.py:2109] Using max model len 32768
(APIServer pid=1386788) INFO 10-07 01:55:04 [config.py:643] Mamba cache mode is set to 'align' for Qwen3_5ForConditionalGeneration by default when prefix caching is enabled
(APIServer pid=1386788) WARNING 10-07 01:55:04 [vllm.py:1706] Enforce eager set, disabling torch.compile, CUDAGraphs, and JIT kernel warmup. This is equivalent to setting --cc.mode=none --cc.cudagraph_mode=none and --kernel_config.enable_jit_warmup=False
(APIServer pid=1386788) WARNING 10-07 01:55:04 [vllm.py:1730] Inductor compilation was disabled by user settings, optimizations settings that are only active during inductor compilation will be ignored.
(APIServer pid=1386788) INFO 10-07 01:55:04 [kernel.py:498] Final IR op priority after setting platform defaults: IrOpPriorityConfig(rms_norm=['vllm_c', 'native'], fused_add_rms_norm=['vllm_c', 'native'], gelu_and_mul_sparse=['triton', 'native'])
(APIServer pid=1386788) INFO 10-07 01:55:04 [vllm.py:1940] Cudagraph is disabled under eager mode
(APIServer pid=1386788) INFO 10-07 01:55:04 [compilation.py:333] Enabled custom fusions: norm_quant, act_quant
(EngineCore pid=1386938) INFO 10-07 01:55:15 [core.py:129] Initializing a V1 LLM engine (v0.1.dev22015+g036d6b435.d20261006) with config: model='Qwen/Qwen3.8-72B', speculative_config=None, tokenizer='Qwen/Qwen3.8-72B', skip_tokenizer_init=False, tokenizer_revision=None, revision=None, trust_remote_code=False, dtype=torch.bfloat16, max_seq_len=32768, download_dir=None, load_format=safetensors, tensor_parallel_size=1, pipeline_parallel_size=1, data_parallel_size=1, decode_only=False, ext_parallel_size=1, dcp_comm_backend=auto_rs, disable_custom_all_reduce=False, quantization=None, quantization_config=None, enforce_eager=True, aux_output_config=AuxOutputConfig(enable_return_routed_experts=False, max_bytes=None), kv_cache_dtype=auto, device_config=cuda, structured_outputs_config=StructuredOutputsConfig(backend='auto', disable_any_whitespace=False, disable_additional_properties=False, reasoning_parser=None, reasoning_parser_plugin=None, enable_in_reasoning=False, observability_config=ObservabilityConfig(show_hidden_metrics_for_version=None, otlp_traces_endpoint=None, collect_detailed_traces=None, per_request_spec_decode_metrics=None), kv_cache_metrics=False, kv_cache_metrics_sample=0.01, cudagraph_metrics=False, enable_layerwise_nvtx_tracing=False, enable_mfu_metrics=False, enable_mm_processor_stats=False, enable_logging_iteration_details=False, jit_monitor_mode='warn', jit_monitor_verbose=False), seed=0, served_model_name=Qwen/Qwen3.8-72B, enable_prefix_caching=True, enable_chunked_prefill=True, pooler_config=None, compilation_config={'mode': '<compilation_mode.NONE: 0>', 'debug_dump_path': None, 'cache_dir': '', 'compile_cache_save_format': 'binary', 'backend': 'inductor', 'custom_ops': ['all'], 'ir_enable_torch_wrap': False, 'splitting_ops': [''], 'compile_mm_encoder': False, 'cudagraph_mm_encoder': False, 'encoder_cudagraph_token_budgets': [], 'encoder_cudagraph_max_vision_items_per_batch': 0, 'encoder_cudagraph_max_frames_per_batch': None, 'compile_sizes': [], 'compile_ranges_endpoints': [2048], 'inductor_compile_config': {'enable_auto_functionalized_v2': False, 'combo_kernels': True, 'benchmark_combo_kernel': True, 'inductor_passes': {}}, 'cudagraph_mode': <CUDAGraphMode.NONE: 0>, 'cudagraph_num_of_warmups': 0, 'cudagraph_capture_sizes': [], 'cudagraph_copy_inputs': False, 'cudagraph_specialize_lora': True, 'use_inductor_graph_partition': False, 'pass_config': {'fuse_norm_quant': True, 'fuse_act_quant': True, 'fuse_attn_quant': False, 'enable_sp': False, 'fuse_gemm_comms': True, 'fuse_all_reduce_rms': False, 'enable_gk_norm_rope_fusion': False, 'fuse_rope_kv_cache_cat_mla': False, 'fuse_act_padding': False, 'fuse_gk_norm_rope_kv_cache': False}, 'max_cudagraph_capture_size': 0, 'dynamic_shapes_config': {'type': 'dynamic_shapes_type.BACKED', 'backed': 'evaluate_guards': False, 'assume_32_bit_indexing': False}, 'local_cache_dir': None, 'fast_moe_cold_start': False, 'static_all_moe_layers': [], kernel_config=KernelConfig(ir_op_priority=IrOpPriorityConfig(rms_norm=['vllm_c', 'native'], fused_add_rms_norm=['vllm_c', 'native'], gelu_and_mul_sparse=['triton', 'native']), enable_flashinfer_autotune=True, enable_cudatoolkit_warmup=True, enable_jit_warmup=False, moe_backend='auto', sparse_indexer_topk_backend='auto', linear_backend='auto', linear_backend_per_quant=None)
(EngineCore pid=1386938) INFO 10-07 01:55:15 [parallel_state.py:1817] world_size=1 rank=0 local_rank=0 distributed_init_method=file:///tmp/vllm_dist_5b5be8389237446fb5e41a2f3927077 backend=nccl
(EngineCore pid=1386938) INFO 10-07 01:55:15 [parallel_state.py:2261] rank 0 in world size 1 is assigned as DP rank 0, PP rank 0, PCP rank 0, TP rank 0, ETP rank 0, EP rank N/A, EPLB rank N/A
(EngineCore pid=1386938) INFO 10-07 01:55:15 [gpu_worker.py:479] Using V2 Model Runner
(EngineCore pid=1386938) INFO 10-07 01:55:16 [model_runner.py:374] Loading model from scratch...
(EngineCore pid=1386938) INFO 10-07 01:55:18 [cuda.py:585] Using backend AttentionBackendEnum.FLASH_ATTEN for vit attention
(EngineCore pid=1386938) INFO 10-07 01:55:18 [mm_encoder_attention.py:372] Using AttentionBackendEnum.FLASH_ATTEN for MMEncoderAttention.
(EngineCore pid=1386938) INFO 10-07 01:55:18 [qwen_gdn_linear_attn.py:177] Using FlashInfer GDN prefill kernel (requested=auto, head_k_dim=128).
(EngineCore pid=1386938) INFO 10-07 01:55:18 [qwen_gdn_linear_attn.py:533] GDN decode kernel: cuda
(EngineCore pid=1386938) INFO 10-07 01:55:18 [cuda.py:526] Using FLASH_ATTEN attention backend out of potential backends: ['FLASH_ATTEN', 'FLASHINFER', 'TRITON_ATTEN', 'FLEX_ATTENTION'].
(EngineCore pid=1386938) INFO 10-07 01:55:18 [flash_attn.py:1170] Using FlashAttention version 2
(EngineCore pid=1386938) INFO 10-07 01:55:21 [weight_utils.py:1069] Filesystem type for checkpoints: EXT4. Checkpoint size: 51.75 GiB. Available RAM: 61.47 GiB.
MMLoading safetensors checkpoint shards: 0% Completed | 0/18 [00:00:00, ?it/s]
[transformers] Qwen3VL video processing does not apply the per-frame pixel cap the reference implementation (qwen-vl-utils) applies, so some videos cost far more tokens than they would there. In v5.22 the capped behavior will become the default and 'cap_pixels_per_frame' will be removed. Pass 'cap_pixels_per_frame=True' to adopt the reference behavior now, or 'False' to keep the current behavior and silence this warning.
(APIServer pid=1386788) INFO 10-07 01:55:25 [base.py:255] Multi-modal warmup completed in 13.421s
MMLoading safetensors checkpoint shards: 6% Completed | 1/18 [00:29:08:20, 29.42s/it]
MMLoading safetensors checkpoint shards: 11% Completed | 2/18 [00:48:06:11, 23.20s/it]
MMLoading safetensors checkpoint shards: 17% Completed | 3/18 [01:07:05:24, 21.60s/it]
MMLoading safetensors checkpoint shards: 22% Completed | 4/18 [01:38:05:53, 25.22s/it]
MMLoading safetensors checkpoint shards: 28% Completed | 5/18 [01:52:04:33, 21.08s/it]
MMLoading safetensors checkpoint shards: 33% Completed | 6/18 [02:23:04:54, 24.54s/it]
MMLoading safetensors checkpoint shards: 39% Completed | 7/18 [02:36:03:49, 20.86s/it]
MMLoading safetensors checkpoint shards: 44% Completed | 8/18 [03:07:03:59, 23.99s/it]
MMLoading safetensors checkpoint shards: 50% Completed | 9/18 [03:21:03:06, 20.73s/it]
MMLoading safetensors checkpoint shards: 56% Completed | 10/18 [03:53:03:15, 24.40s/it]

```

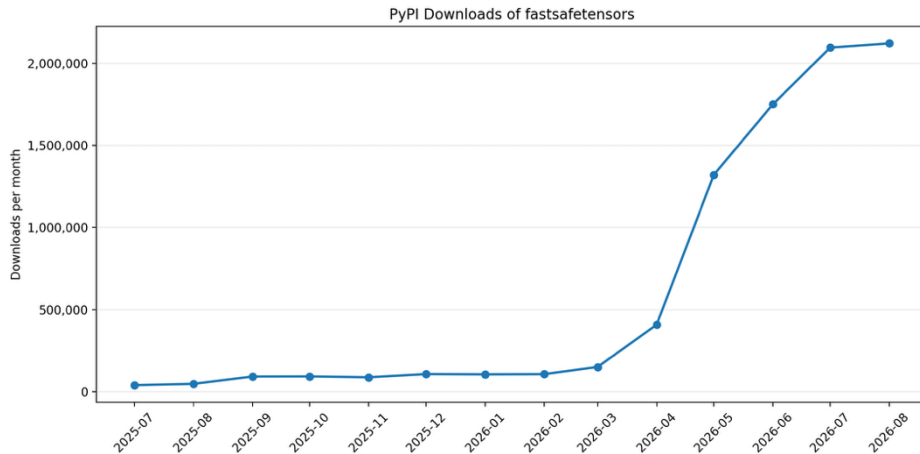

Problem: safetensors load time (DGX Spark)



What is fastsafetensors (fast-safe-tensors)?

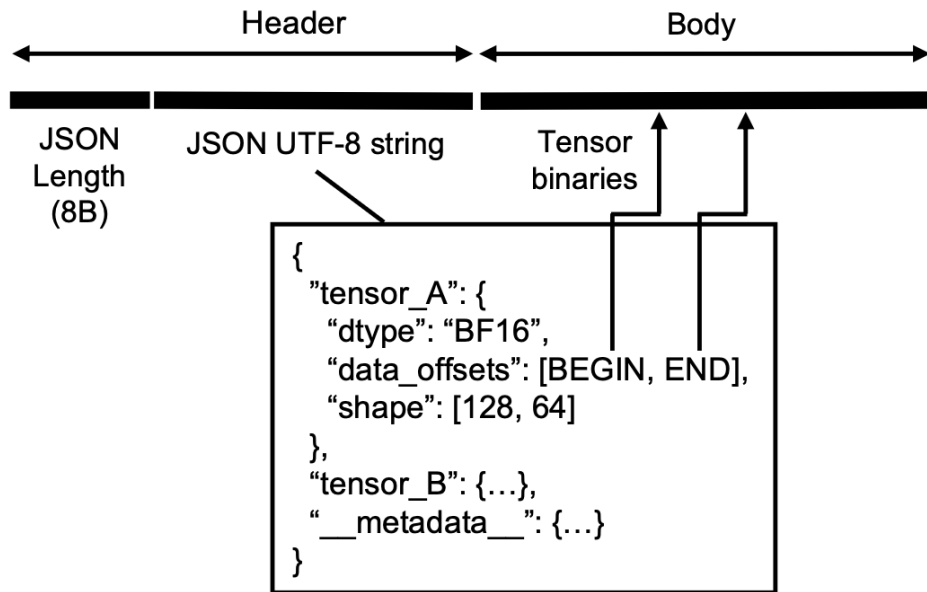
- Open library for safetensors load
 - Optimize DMA pipelines to load large and many checkpoint files
 - Latest release 0.5 in October add new optimization
- Example:
 - `vllm serve --load-format fastsafetensors`

>2M PyPi downloads Aug 2026



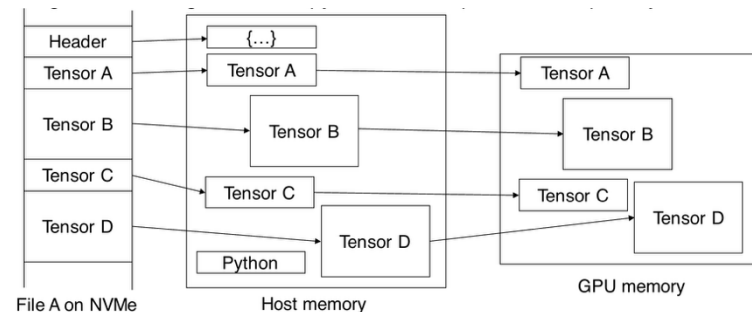
safetensors

- Simple and efficient file format for checkpoints
 - Often used for models in Hugging Face
- Deserialization *was* suboptimal
 - mmap caused high host memory pressure on Linux
 - 0.9.0rc1 adds new feature **prefetch!**

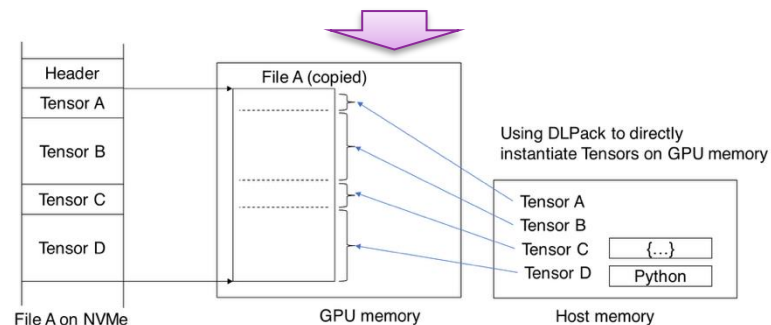


Key design

- Do not run sequential `model.to()`
 - Copy tensor chunks in files to device memory at first
 - Then, map device tensor objects
- Extensibility with *copier* interface
 - Can take different copy paths specific to the underlying platform
 - Unify pipeline strategy for any copiers



Original safetensors: `to()` after mmap tensor objects

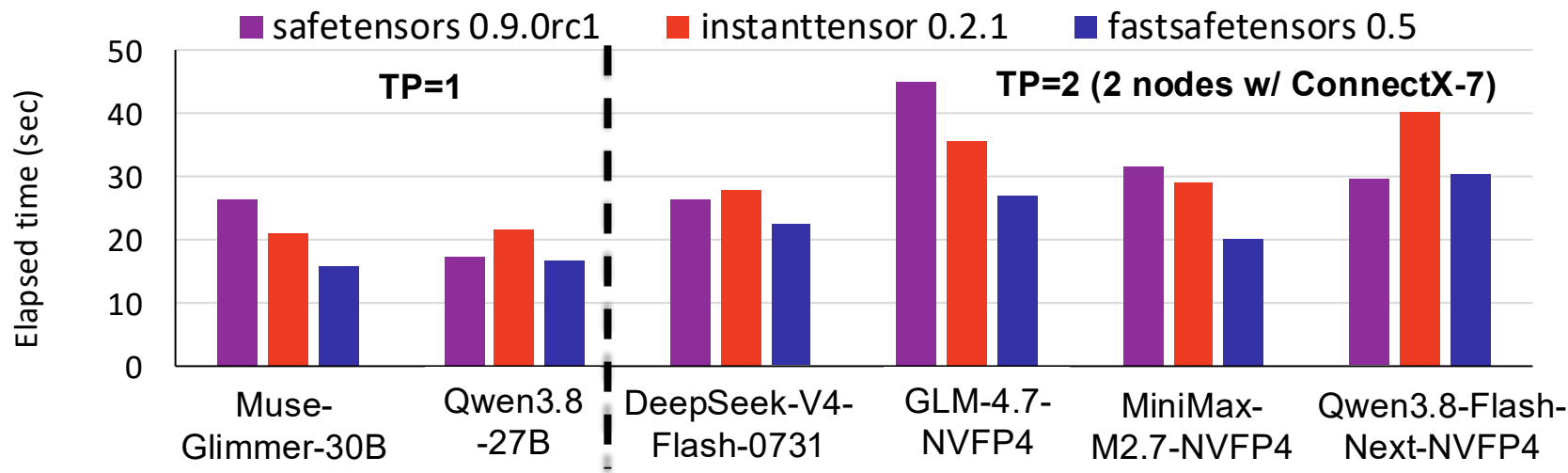


fastsafetensors: copy first and map with DLPack

Optimization for various platforms

- NVIDIA GPUDirect Storage
- AMD ROCm hipFile
- Unified Memory System, e.g., DGX Spark
- Windows DirectStorage
- DeepSeek 3FS
- Fast GDS (<https://github.com/Storage-and-OS-for-AI/fgds>)
- IBM Spyre (plan)

Model load time on vLLM draft PRs w/ DGX Spark



- safetensors 0.9.0rc1 (prefetch) achieved similar load time to others
- fastsafetensors 0.5 showed *slightly* better performance than others
- A100/H100 showed similar trends

Summary

- Safetensors load issue is almost over on Linux CUDA
 - Other environments (e.g., IBM Spyre) still need optimization
- Code: <https://github.com/foundation-model-stack/fast safetensors>
 - Contact: Takeshi Yoshimura tyos@jp.ibm.com or @takeshi-yoshimura
- Acknowledgement:
 - @ABNER-1, @gitbisector, @tjtanaa, @zeroRains, @lixin010, @kurapov-peter, @wjabbour, @yinjuncheng, @noobHappyLife, @SystemPanic, @ZhiweiYan-96, @bet0x, @njhill, @esmeetu, @bukejiyu, @spzala, @manish-sethi, @chinazhangchao, @BabyChouSr