Scaling Large Language Model Serving Infrastructure at Meta

A comprehensive recipe to turn LLMs into LLM serving infrastructure

Ye (Charlotte) Qi Al Inference @ Meta







The AI Gold Rush



Inference Scaling and Compound Systems Are Coming



@DrJimFan

Background about Myself



Ye (Charlotte) Qi 🔿 🕸 AlInfra

Had been running model services for 6.5 years

- Ads model serving
- LLaMa serving

Machine translation research before Meta

We Support Product Backend for Meta AI

500M Monthly active users

Q А



Behind the making of LLaMa

Liama 2 Llama-2-7B Llama-2-73B Llama-2-70B

405B

8

В

Meet Llama 3.1

70B

INTRODUCING Lightweight Llama models

Introducing Meta Llama 3

and multimodal



safety

"Should I run my own LLM services?"

Let's Build This Step By Step



NEWS

Uber Achieves Significant Storage Savings with InfoQ Dev Summit Munich: In-Memory Java **MyRocks Differential Backups** Database EclipseStore Delivers Faster Data Processing MATT SAUNDERS ON NOV 11 KARSTEN SILZ ON NOV 11 OpenJDK News Roundup: Instance Main Methods, Flexible Constructor Bodies, Module Import Transforming Gaming Worlds Declarations DANIEL DOMINGUEZ ON NOV 10 MICHAEL REDLICH ON NOV 11 Cloudflare Advocates for Broader Adoption of security.txt Standard for Vulnerability Reporting RENATO LOSIO ON NOV 10 STEEF-JAN WIGGERS ON NOV 09

Amazon Aurora Introduces Global Database Writer Endpoint for Distributed Applications

RENATO LOSIO ON NOV 09

Decart and Etched Release Oasis, a New AI Model

Microsoft Unveils Enhanced AI Tools for Developers at GitHub Universe

Optimizing Uber's Search Infrastructure: Upgrading to Apache Lucene 9.5

ADITYA KULKARNI ON NOV 08

TRENDING

_	10 days			40 days	6 months
	1	Đ	Microsoft for .NET: / Cloud	Introduces Moderr Accelerating App Mo	Web App Pattern odernization to the
	2	Đ	Cloudflar Eliminatir	e Introduces Short- ng the Need for SSH	Lived SSH Access, I Credentials
	3	Ð	Apache To Threads a	omcat 11.0 Delivers and Jakarta EE 11	Support for Virtual
	4	Đ	Amazon A Writer En	Aurora Introduces G dpoint for Distribut	lobal Database ed Applications
	5		Java-Base Bootstrap	ed No-Code and Lov oping Tools Review	v-Code Application
	6	ត	Namee Ol How They	berst on Small Lang / are Enabling AI-Po	uage Models and wered PCs
	7	ត	Generally Godfathe	AI - Season 2 - Epis rs of Programming	sode 6: the and AI

MORE NEWS >

Summarize Charlotte's posts and ask follow-ups

Challenge 1 Fitting

Challenge 2

Challenge 3

Challenge 4

STEP 1

Find a good runtime

Isn't that just grabbing eval code?

triggers one model.forward

prefill

Llama is an artificial intelligence model developed by Meta, designed to process and generate human-like language. Like other large language models, Llama uses natural language processing to generate text. It works by taking a sequence of words as input and predicting the next word to create coherent and natural-sounding



what is llama



working on it!!!

Continuous Batching



Use this







Imagine this sentence being

generated by an LLM. KV

tensors for yellow parts are

cached in GPU memory at

128KiB/tok (LLaMa3-8B) under

bf16.



KV Cache

> How does KV cache work?

320KiB/tok (LLaMa3-70B),

TensorRT-LLM



STEP 2

Understand hardware resources





Let's Only Worry About Model Loading

40/80GB

80/96GB NVIDIA H100

192GB



STEP 3: Start fitting some models with 80GB H100x8



STEP 3: Single-Card Inference

LLaMa3-8B

bf16: 16GB < 80GB

STEP 3: Distributed Inference: Tensor Parallelism

LLaMa3-70B

bf16: 140GB < 80GB x 2



Partitioning Weights

STEP 3: Distributed Inference: Pipeline Parallelism

LLaMa3-405B

bf16: 810GB < 80GB x 16 bf16: 810GB < 192GB x 8





Partitioning Weights More

Or Find GPUs With Bigger HBM

Takeaways

Use tensor/pipeline parallelism to fit your models

Find a specialized LLM runtime as your starting point

Understand system resources available on AI hardware

Let's Build This Step By Step (Part 2!)



Optimizing Uber's Search Infrastructure:

Upgrading to Apache Lucene 9.5

ADITYA KULKARNI ON NOV 08

RENATO LOSIO ON NOV 10

Amazon Aurora Introduces Global Database Writer Endpoint for Distributed Applications

RENATO LOSIO ON NOV 09

MORE NEWS >

1	Đ	Microsoft Introduces Modern Web App Pattern for .NET: Accelerating App Modernization to the Cloud
2	Đ	Cloudflare Introduces Short-Lived SSH Access, Eliminating the Need for SSH Credentials
3	Đ	Apache Tomcat 11.0 Delivers Support for Virtual Threads and Jakarta EE 11
4	Đ	Amazon Aurora Introduces Global Database Writer Endpoint for Distributed Applications
5	Ē	Java-Based No-Code and Low-Code Application Bootstrapping Tools Review
6	ត	Namee Oberst on Small Language Models and How They are Enabling AI-Powered PCs
7	ត	Generally AI - Season 2 - Episode 6: the Godfathers of Programming and AI



Challenge 1 Fitting

Challenge 2 It's Slow

Challenge 3

Challenge 4

Throw capacity at the problem?

reddit.com/r/ProgrammerHumor/comments/hnlge5/why solve problems when you can just throw money/

credit:

SCALABILITY





Add more replicas



Buy faster hardware



(Image credit: nvidia)

usage: vllm serve [-h] [--host HOST] [--port PORT] [--uvicorn-log-level {debug,info,warning,error,critical,trace}] [--allow-credentials] [--allowed-origins ALLOWED_ORIGINS] [--allowed-methods ALLOWED_METHODS] --allowed-headers ALLOWED_HEADERS] [--api-key API_KEY] -lora-modules LORA_MODULES [LORA_MODULES ...]] --prompt-adapters PROMPT_ADAPTERS [PROMPT_ADAPTERS ...]] -chat-template CHAT_TEMPLATE] -response-role RESPONSE_ROLE] [--ssl-keyfile SSL_KEYFILE] -ssl-certfile SSL_CERTFILE] [--ssl-ca-certs SSL_CA_CERTS] --ssl-cert-reqs SSL_CERT_REQS] [--root-path ROOT_PATH] -middleware MIDDLEWARE] [--return-tokens-as-token-ids] -disable-frontend-multiprocessing] -enable-auto-tool-choice] --tool-call-parser {granite-20b-fc,granite,hermes,internlm,jamba,llama3_j --tool-parser-plugin TOOL_PARSER_PLUGIN] [--model MODEL] -task {auto,generate,embedding}] [--tokenizer TOKENIZER] -skip-tokenizer-init] [--revision REVISION] -code-revision CODE_REVISION] -tokenizer-revision TOKENIZER_REVISION] -tokenizer-mode {auto,slow,mistral}] -chat-template-text-format {string,openai}] --trust-remote-code] --allowed-local-media-path_ALLOWED_LOCAL_MEDIA_PATH] --download-dir DOWNLOAD_DIR] --download-dir DOWNLOAD_DIR] --load-format {auto,pt,safetensors,npcache,dummy,tensorizer,sharded_state --config-format {auto,hf,mistral}} --dtype {auto,half,float16,bfloat16,float32}] -max-model-len MAX_MODEL_LEN] -guided-decoding-backend {outlines, lm-format-enforcer}] --distributed-executor-backend {ray,mp} [--worker-use-ray] --pipeline-parallel-size PIPELINE_PARALLEL_SIZE] -tensor-parallel-size TENSOR_PARALLEL_SIZE] -max-parallel-loading-workers MAX_PARALLEL_LOADING_WORKERS] ---ray-workers-use-nsight] [---block-size {8,16,32,64,128}] --enable-prefix-caching] [--disable-sliding-window] -use-v2-block-manager] -num-lookahead-slots NUM_LOOKAHEAD_SLOTS] [--seed SEED] --swap-space SWAP_SPACE] [--cpu-offload-gb CPU_OFFLOAD_gB] --gpu-memory-utilization GPU_MEMORY_UTILIZATION] --num-gpu-blocks-override NUM_GPU_BLOCKS_OVERRIDE] --max-num-batched-tokens MAX_NUM_BATCHED_TOKENS] --max-num-seqs MAX_NUM_SEQS] [--max-logprobs MAX_LOGPROBS] -disable-log-stats] --quantization {aqlm,awq,deepspeedfp,tpu_int8,fp8,fbgemm_fp8,modelopt,mar --rope-scaling ROPE_SCALING] [--rope-theta ROPE_THETA] --hf-overrides HF_OVERRIDES] [--enforce-eager] --max-seq-len-to-capture MAX_SEQ_LEN_TO_CAPTURE] --disable-custom-all-reduce] --tokenizer-pool-size TOKENIZER_POOL_SIZE] -tokenizer-pool-type TOKENIZER_POOL_TYPE] -tokenizer-pool-extra-config TOKENIZER_POOL_EXTRA_CONFIG] --limit_mm_per_prompt LIMIT_MM_PER_PROMPT] --mm_processor-kwargs MM_PROCESSOR_KWARGS] [--enable-lora] -max-loras MAX_LORAS] [--max-lora-rank MAX_LORA_RANK] -lora-extra-vocab-size LORA_EXTRA_VOCAB_SIZE] -lora-dtype {auto,float16,bfloat16,float32}] -long-lora-scaling-factors LONG_LORA_SCALING_FACTORS] -max-cpu-loras MAX_CPU_LORAS] [--fully-sharded-loras] -enable-prompt-adapter] -max_prompt-adapters MAX_PROMPT_ADAPTERS] -max_prompt-adapter-token MAX_PROMPT_ADAPTER_TOKEN] -device {auto,cuda,neuron,cpu,openvino,tpu,xpu,hpu}] -num-scheduler-steps NUM_SCHEDULER_STEPS] --multi-step-stream-outputs [MULTI_STEP_STREAM_OUTPUTS]] --scheduler-delay-factor SCHEDULER_DELAY_FACTOR] --enable-chunked-prefill [ENABLE_CHUNKED_PREFILL]] --speculative-model SPECULATIVE_MODEL] -speculative-model-quantization {aqlm,awq,deepspeedfp,tpu_int8,fp8,fbgen -num-speculative-tokens NUM_SPECULATIVE_TOKENS] --speculative-disable-mqa-scorer] --speculative-draft-tensor-parallel-size SPECULATIVE_DRAFT_TENSOR_PARALLE --speculative-max-model-len SPECULATIVE_MAX_MODEL_LEN] --speculative-disable-by-batch-size SPECULATIVE_DISABLE_BY_BATCH_SIZE] --specutative_disable-oy-batch_size_specutative_disable_by_batch_size} --ngram-prompt-lookup-max NGRAM_PROMPT_LOOKUP_MAX] --ngram-prompt-lookup-min NGRAM_PROMPT_LOOKUP_MIN] --spec-decoding-acceptance-method {rejection_sampler,typical_acceptance_s --typical-acceptance-sampler-posterior-threshold_TYPICAL_ACCEPTANCE_SAMPL -typical-acceptance-sampler-posterior-alpha TYPICAL_ACCEPTANCE_SAMPLER_F -disable-logprobs-during-spec-decoding [DISABLE_LOGPROBS_DURING_SPEC_DEC -model-loader-extra-config MODEL_LOADER_EXTRA_CONFIG] -ignore-patterns IGNORE_PATTERNS] -preemption-mode PREEMPTION_MODE] -served-model-name SERVED_MODEL_NAME [SERVED_MODEL_NAME ...]] -qlora-adapter-name-or-path QLORA_ADAPTER_NAME_OR_PATH] -otlp-traces-endpoint OTLP_TRACES_ENDPOINT] -collect-detailed-traces COLLECT_DETAILED_TRACES] -disable-async-output-proc] --override-neuron-config OVERRIDE_NEURON_CONFIG] --scheduling-policy {fcfs,priority}] --pooling-type {LAST,ALL,CLS,STEP,MEAN}] [--pooling-norm] --no-pooling-norm] [--pooling-softmax] -no-pooling-softmax] --pooling-step-tag-id POOLING_STEP_TAG_ID] --pooling-returned-token-ids POOLING_RETURNED_TOKEN_IDS [POOLING_RETURNEL --disable-log-requests] [--max-log-len MAX_LOG_LEN] [--disable-fastapi-docs]

But it doesn't seem to help ...



credit: https://docs.vllm.ai/en/latest/serving/openai compatible server.html#command-line-arguments-for-the-server

Why is That?... Let's Understand System Bottleneck a Bit Deeper..



Prefill = read O(10GB) weight, compute O(1) - O(100) TFLOPs! **Decode** = read O(10GB) weight, compute O(0.01) - O(0.1) TFLOPs for O(10)-O(100) times! And generate O(100KB) KV cache per token.



Make LLM faster

- = Fit LLM operations within system resources
- = Fit GPU compute + fit memory bandwidth + fit memory capacity

Prefill

Decode

Which scales ~ model sizes, sequence length, batch size

But each GPU has fixed configuration for system resource ratio

fit memory capacity KV Cache

What Type of Slowness are You Talking About...



Understand what's reasonable expectation. Look at public benchmark data by





Quality Trade-off

You are unhappy with average generation speed



As context window gets longer



STEP 1 Throw Capacity More Wisely – Using Disaggregated Prefill/Decode

3 PREFILL HOSTS

Disaggregated prefill/decode improve latency-bound throughput and reduce tail decode latency

IDECODE HOST IDECODE HOST

→ Replicating Weights

STEP 1 Throw Capacity Wisely – Using Context Parallelism

Input lism



3+3 PREFILL HOSTS

1 min for 128K input without parallelism

1 DECODE HOST 1 DECODE HOST

Replicating Weights Partitioning Context



STEP 2 Make Your Problems Smaller



Shorter Prompts

Be a good and concise communicator to LLM

Smaller Models

Fine-tuning small generalist

Distillation / pruning



Quantization

Many components to choose (FFN / KV / ATTN)

Different data types (INT8 / FP8 / INT4 / NF4)

Different policies to choose (W8A8 / W4A16 / tensor-wise / row-wise / group-wise)



Memory Hierarchy with Bandwidth & Memory Size



With Prefix Caching

```
Typically ...
```

Technique	TTFT	ттіт
Speculative Decoding	••	
Chunked Prefill		•••
Attention Kernels	··· –	•• – ••
Token Sparsity		

for your unique workload requirement, every 🔗 🙂 😨 😶 may be flipped







Domain Specific Inference Optimizations

Decoding Algorithms

Serving Engine



Kernels



Model Architecture

"Did you do all of these at Meta?"

Let's Build This Step By Step (Part 3!)



Microsoft Unveils Enhanced AI Tools for

Optimizing Uber's Search Infrastructure:

Developers at GitHub Universe

Upgrading to Apache Lucene 9.5

STEEF-JAN WIGGERS ON NOV 09

ADITYA KULKARNI ON NOV 08

Cloudflare Advocates for Broader Adoption of security.txt Standard for Vulnerability Reporting

RENATO LOSIO ON NOV 10

Amazon Aurora Introduces Global Database Writer Endpoint for Distributed Applications

RENATO LOSIO ON NOV 09

MORE NEWS >

	10 0	lays	40 Uay	3	omonti	15
1	ß	Microsoft for .NET: / Cloud	Introduces M Accelerating A	lodern We opp Moder	b App Patter nization to t	rn he
2	Ð	Cloudflar Eliminatir	e Introduces S ng the Need fo	Short-Live or SSH Cre	d SSH Acces dentials	s,
3	Đ	Apache To Threads a	omcat 11.0 De Ind Jakarta EE	livers Sup 11	port for Virt	ual
4	Đ	Amazon A Writer En	urora Introdu dpoint for Dis	ices Globa stributed A	l Database pplications	
5	Ð	Java-Base Bootstrap	ed No-Code ar oping Tools Re	nd Low-Co eview	de Applicati	on
6	្ល	Namee Ol How They	berst on Smal are Enabling	ll Languag Al-Power	e Models an ed PCs	d
7	ត	Generally Godfathe	AI - Season 2 rs of Program	- Episode ming and	6: the Al	



Challenge 1 Fitting

Challenge 2 It's Slow

Challenge 3 Production

Challenge 4



Input length and traffic change over time





The Hard Lessons from Reality





Hardware FLOPS Kernel Benchmark FLOPS Latency-bound FLOPS Running FLOPS

STEP 1: Understand If Anything Can be Traded Off

Trade-offs!

- Reliability:
 - Understand the cost of different 9s

• Latency: average adult

- Reads English @ 200-300
 words per minute
- Speaks English @ 150-200
 words per minute



RELIABILITY





THROUGHPUT

STEP 2: Look at E2E Latency - Network



Note: This data is made up and does not represent real latency at Meta.

Business Logic

400ms







Compute & transfer overlap, system scheduling optimization





product x infra co-design to maximize cache hit rate

BUT,

STEP 3: Look at E2E Quality - Quantization

YES,



Build you own product eval

BUT,



STEP 3: Look at E2E Quality - CI/CD

Category Benchmark	Llama 3.1 405B	Llama 3.1 70B	Llama 3.1 8B
General MMLU (0-shot, CoT)	88.6	86.0	73.0
MMLU PRO (5-shot, CoT)	73.3	66.4	48.3
IFEval	88.6	87.5	80.4
Code HumanEval (0-shot)	89.0	80.5	72.6
MBPP EvalPlus (base) (0-shot)	88.6	86.0	72.8
Math GSM8K (8-shot, CoT)	96.8	95.1	84.5
MATH (0-shot, CoT)	73.8	68.0	51.9
Reasoning ARC Challenge (0-shot)	96.9	94.8	83.4
GPQA (0-shot, CoT)	51.1	46.7	32.8
Tool use BFCL	88.5	84.8	76.1
Nexus	58.7	56.7	38.5
Long context ZeroSCROLLS/QuALITY	95.2	90.5	81.0
InfiniteBench/En.MC	83.4	78.2	65.1
NIH/Multi-needle	98.1	97.5	98.8
Multilingual Multilingual MGSM (0-shot)	91.6	86.9	68.9





Credit: https://www.mabl.com/blog/what-is-cicd

Non-inference latency is substantial

Infra x product co-design to optimize effective cache hit rate

Continuously test acceleration techniques using product signals

Takeaways

We lose theoretical performance in production environment A LOT

Let's Build This Step By Step (Part 4!)



Developers at GitHub Universe

Upgrading to Apache Lucene 9.5

Optimizing Uber's Search Infrastructure:

STEEF-JAN WIGGERS ON NOV 09

ADITYA KULKARNI ON NOV 08

Amazon Aurora Introduces Global Database

Bootstrapping Tools Review

Writer Endpoint for Distributed Applications

Java-Based No-Code and Low-Code Application

Namee Oberst on Small Language Models and

How They are Enabling AI-Powered PCs

Generally AI - Season 2 - Episode 6: the

Godfathers of Programming and AI

-

5

6 6

6

Cloudflare Advocates for Broader Adoption of security.txt Standard for Vulnerability Reporting

RENATO LOSIO ON NOV 10

Amazon Aurora Introduces Global Database Writer **Endpoint for Distributed Applications**

RENATO LOSIO ON NOV 09

MORE NEWS >



Challenge 1 Fitting

Challenge 2 It's Slow

Challenge 3 Production

Challenge 4 Scaling

Let's build a rocket and make it fly!



What Will Scale?

- Number of Deployments
- Number of GPUs
- Number of Developers
- Number of Models



STEP 1: Scale Number of Deployments - Allocation at Scale (Physical Availability)



BUT,

Region 1



Region 2



STEP 1: Scale Number of Deployments - Allocation at Scale (Shared Fate)



STEP 1: Scale Number of Deployments - Allocation at Scale (Capacity Constraints)



Your Capacity Limit



Maintenance Event

STEP 2 Inference Optimization At Scale



Inference Menu

ιсу	Quality	Quality Complexity			
		0	\$\$		
	0	0	\$		
			\$\$\$		

YES,

How to use these free capacity?

BUT,

achievable with inference optimizations

STEP 2: Inference Optimization At Scale (Where to Focus)		tail	tail	tail	tail	tail	tail	tail	tail	tail	tail
Guess		tail	tail	tail	tail	tail		tail	tail	tail	
		tail	tail	tail	tail	tail Reality tail			tail	tail	
		tail	tail	tail	tail	tail	tail	tail	tail	tail	tail
		tail	tail	tail							tail
	Head	tail	tail	tail							tail
	TICCU	tail	tail	tail		Head				tail	tail
		tail	tail	tail							tail
	tail tail tail tail	tail	tail	tail					tail	tail	
		tail	tail	tail	tail	tail	tail	tail	tail	tail	tail
		tail	tail	tail	tail	tail	tail	tail	tail	tail	tail
		tail	tail	tail	tail	tail	tail	tail	tail	tail	tail

STEP 2: Inference Optimization At Scale (Data Driven Cost Reduction)



Used Throughput



Charlotte's post is boring. Let users customize their subscriptions

Putting together everything we talked so far ... You'll get a scalable LLM serving infrastructure!

THE LLM SERVING ICEBERG





Meta

