Autoscaling of AI services

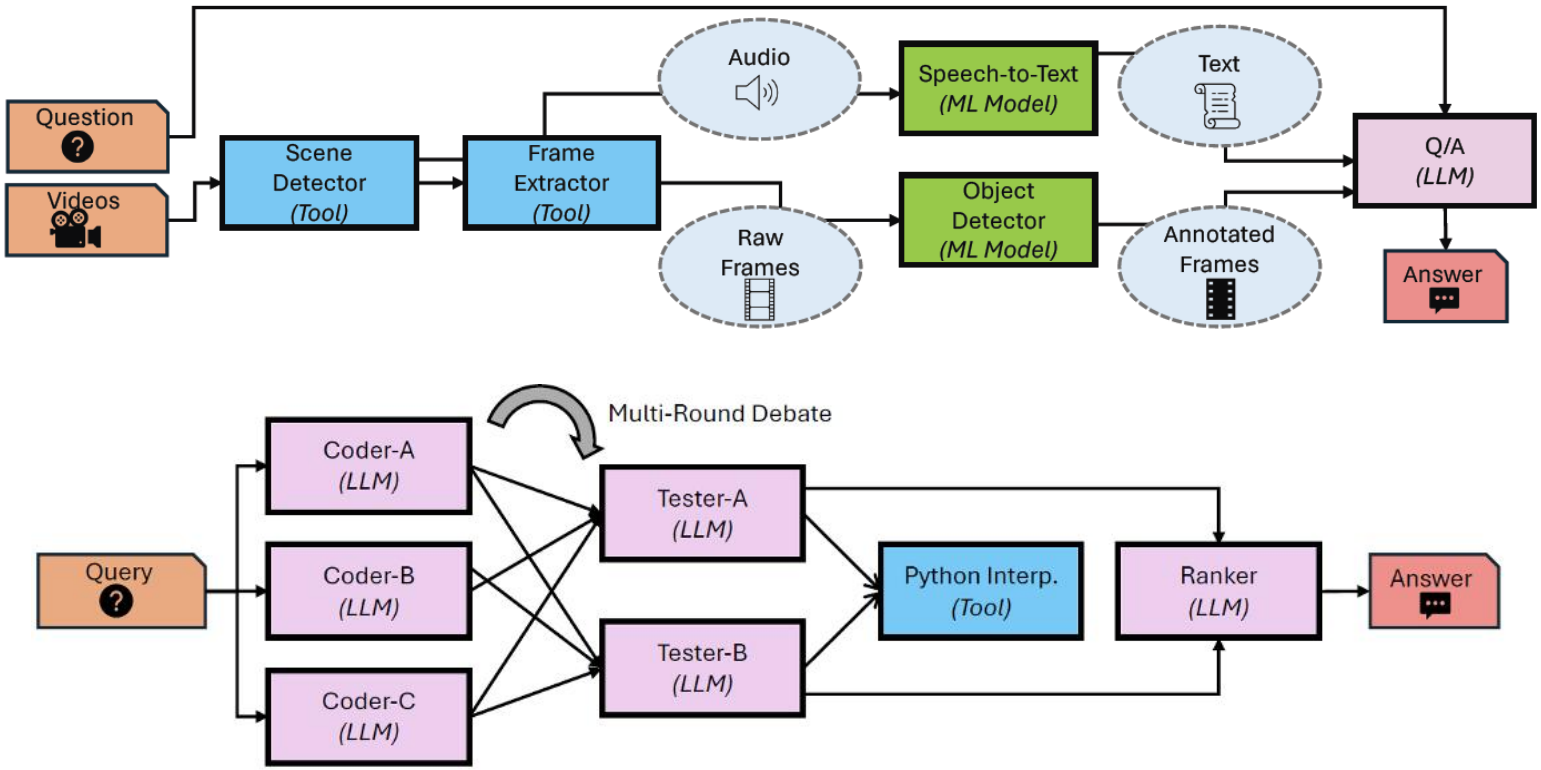
Kostas Magoutis
Professor and Chair, CSD, University of Crete &
Affiliated researcher, FORTH-ICS
magoutis@ics.forth.gr
http://www.csd.uoc.gr/~magoutis

FORTH
INSTITUTE OF COMPUTER SCIENCE

UOC
CSD

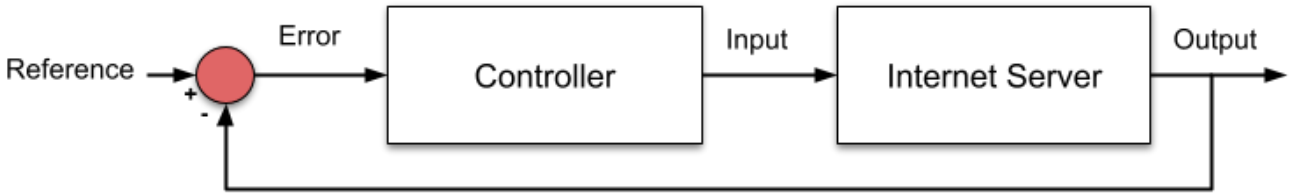
OLYMPIA
AI DATA
INFRASTRUCTURE
SUMMIT 2026

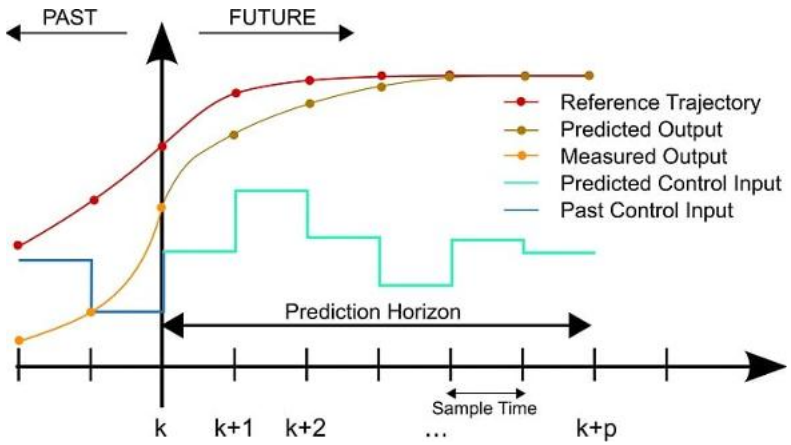
Agentic AI workflows



Chaudhury et al., Murakkab: Resource-Efficient Agentic Workflow Orchestration in Cloud Platforms, OSDI'26

Autoscaling infrastructure services





Autoscaling LLM inference servers

- Operation & optimizations (SOTA)
- References (SLOs)
- Control inputs (knobs)
- Performance prediction
- Dynamic reconfiguration

Autoscaling Internet servers [ACSOS'24]

Background

AI infrastructures

Connectivity to other compute and storage nodes

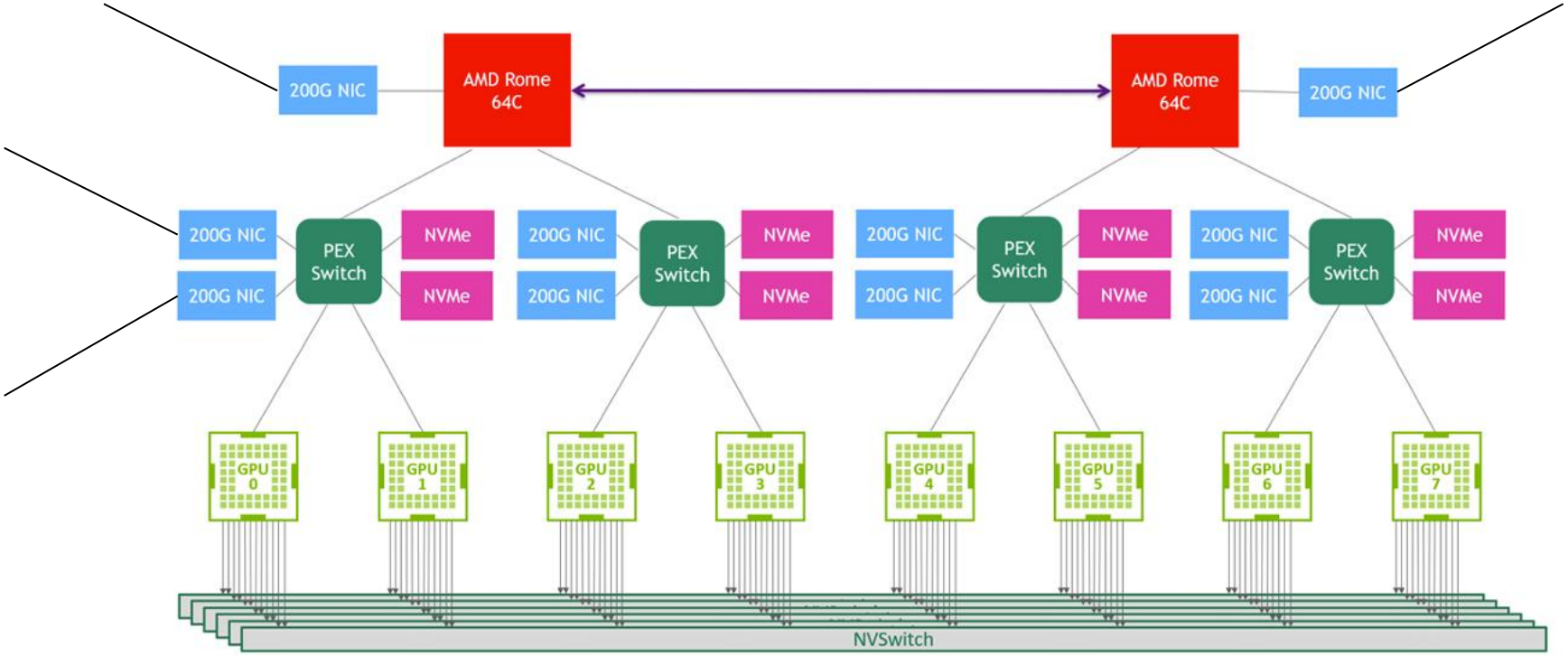
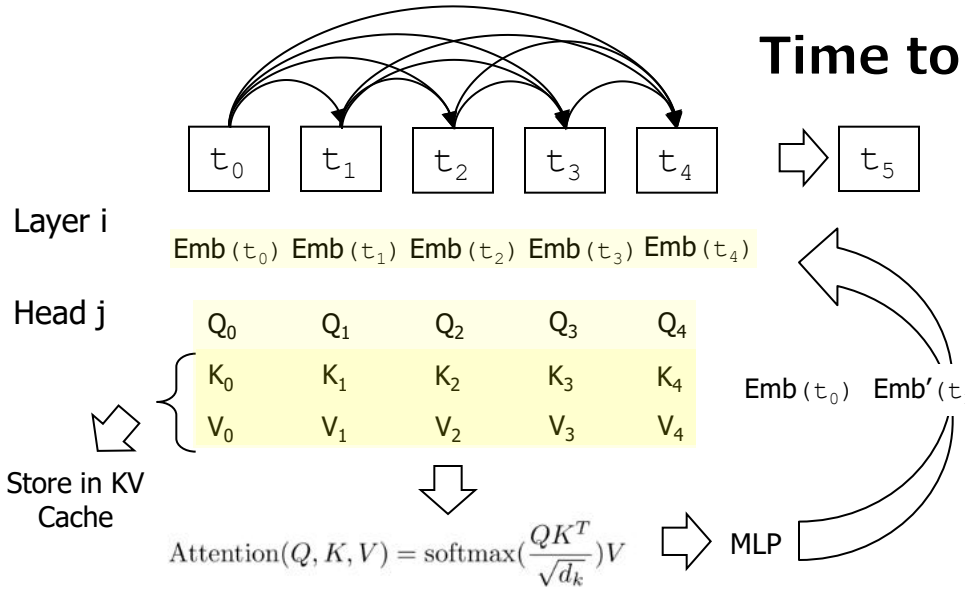


Image source: NVIDIA A100 Tensor Core GPU Architecture In-Depth (Whitepaper)

Transformer phases: Prefill, Decode

PREFILL

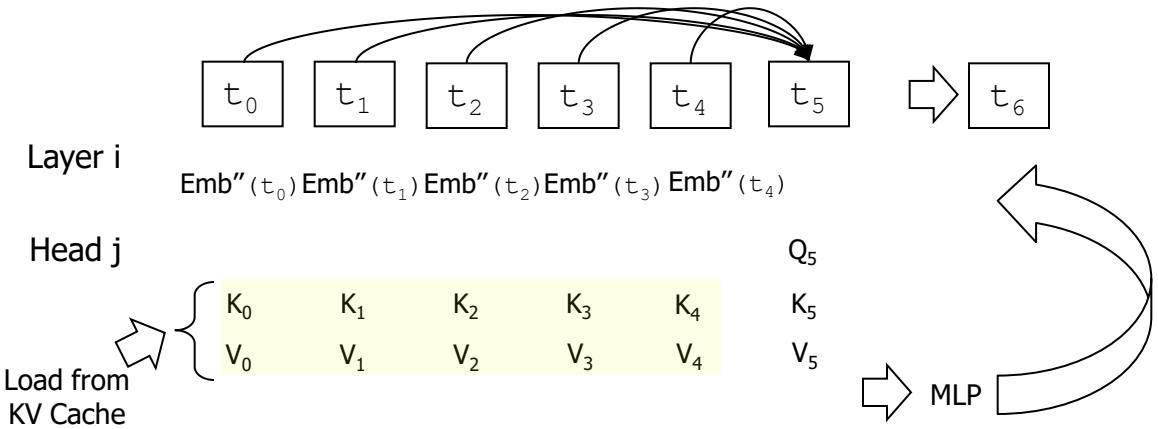


Time to First Token (TTFT)

- Separate prefill & decode servers
- GPU parallelization
  - Layers (pipelining)
  - Tensor

Cache lookup key: (prefix (tokens), layer, head)

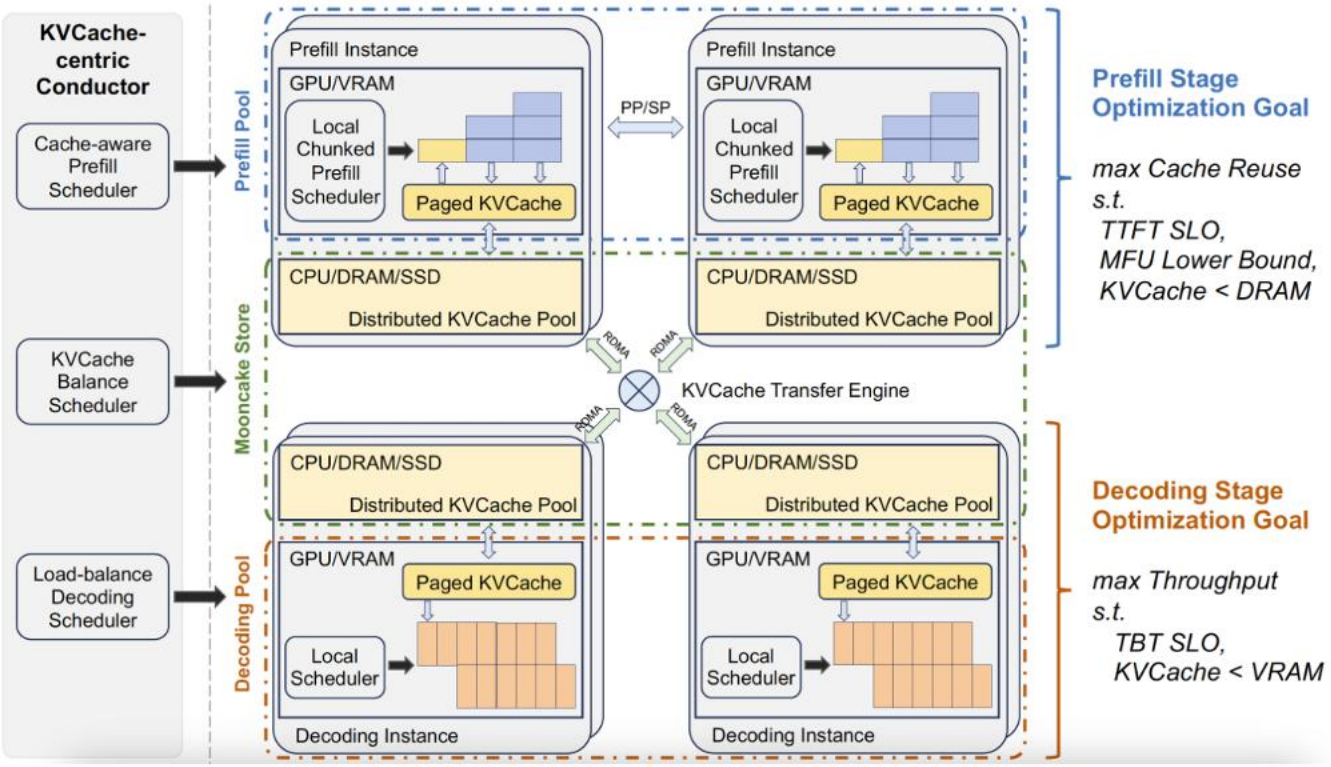
DECODE



Time between Tokens (TBT)

- Caching
  - Improve cache hit ratios
  - Increase cache size
  - Multi-level caching
  - High-speed transfers

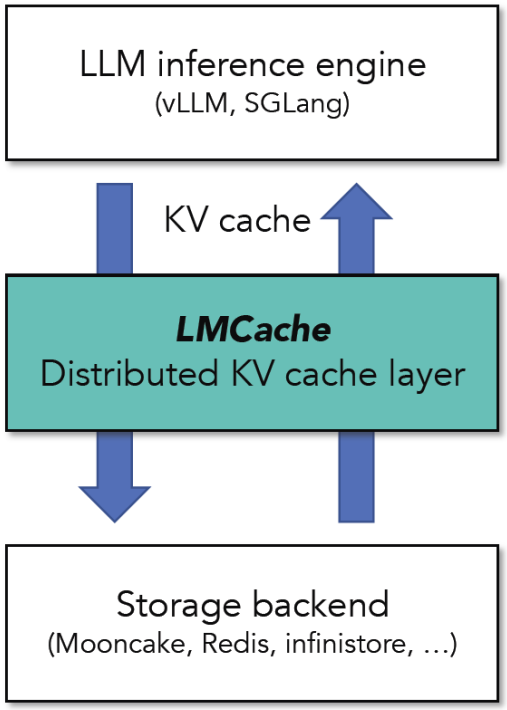
Disaggregated KVCache-centric architecture



Qin et al. Mooncake: Trading More Storage for Less Computation — A KVCache-centric Architecture for Serving LLM Chatbot, FAST'25

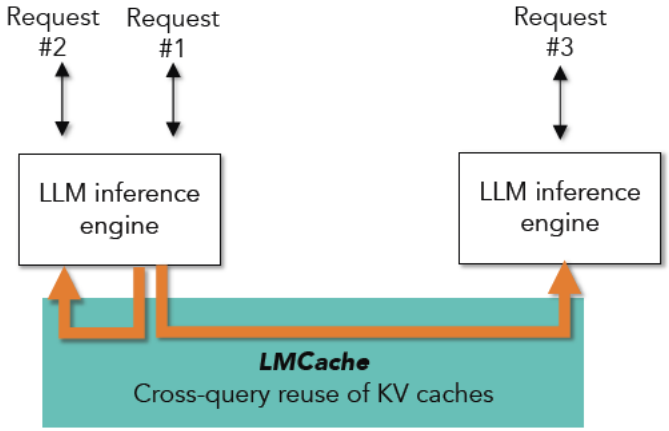
See also: Kwon et al., Efficient Memory Management for Large Language Model Serving with PagedAttention, SOSP'23 (vLLM)

Context (prefix) caching

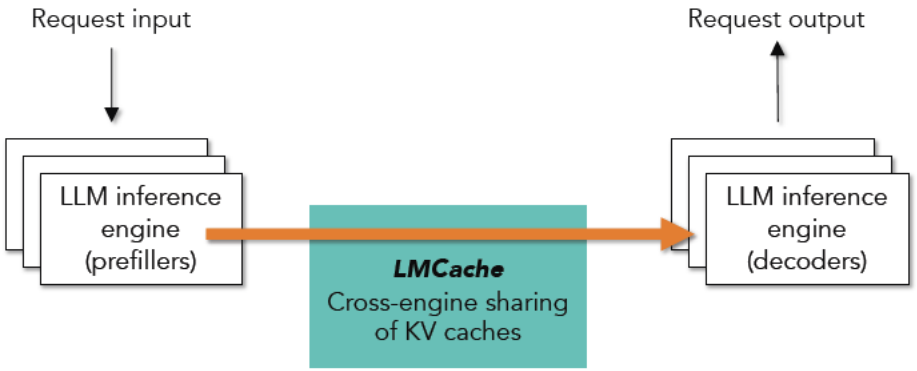


Liu et al., LMCache: An Efficient KV Cache Layer for Enterprise-Scale LLM Inference, arXiv'25

Context (prefix) caching & Prefill-Decode disaggregation



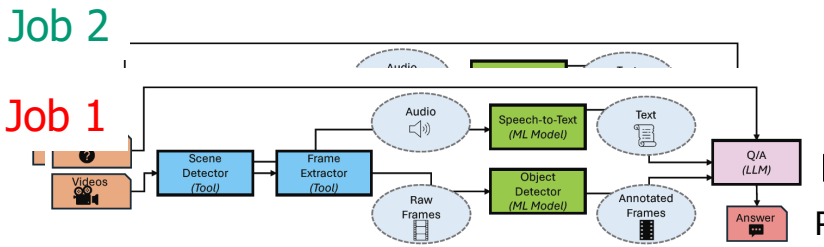
(a) Context caching: Cross-query reusing via CPU/disk offloading of KV cache



(b) PD disaggregation: Cross-engine sharing of KV caches

Liu et al., LMCache: An Efficient KV Cache Layer for Enterprise-Scale LLM Inference, arXiv'25

Autoscaling architecture



Prompt/context

Policies:
- Prompt-aware routing
- Scale-up/down (based on SLO)
- Reconfigure vLLM

Router/controller

Telemetry

SLO (TTFT, TBT targets)

Prompt/context

Scale, reconfigure (reshard)

Job 1 query

Job 2 query

vLLM

Llama3 70B

KVCache

RDMA

Job 1 query

Job 2 query

Performance modeling
Online / Offline
(benchmarks, e.g., MLPerf,
vLLM bench, AIPerf)

Prefill GPU server pool

Decode GPU server pool

External APIs

TTFT, TBT
measurements

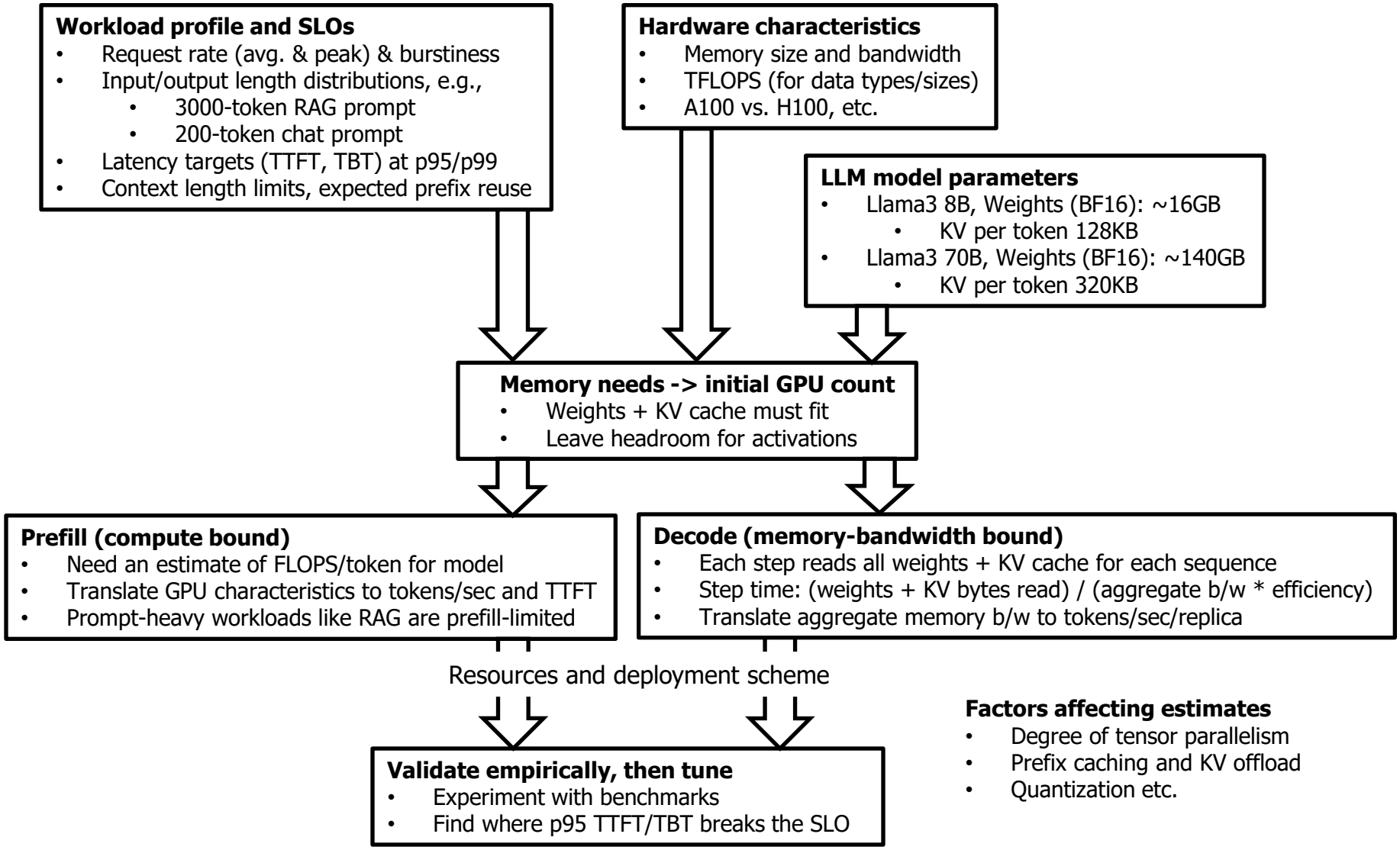
KVCache store

First-class data objects: Manage and share for long-context, multi-turn, and agentic AI inference

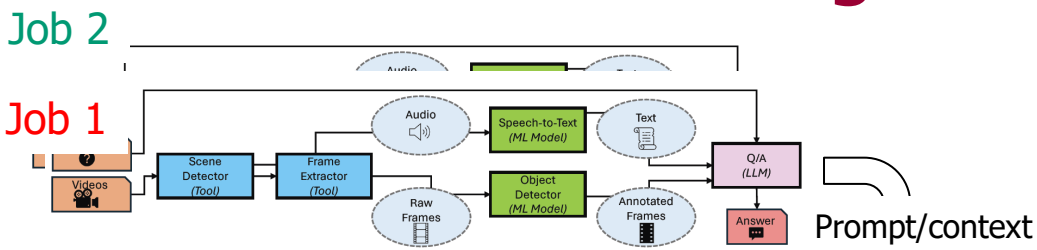
Challenges

- AI workloads are bursty
  - LLM server pools may reach capacity over time periods
- Reconfiguration is costly – typically checkpoint-restart
  - Re-sharding at model scale (100s GBs of weights) takes a long time
  - TENPLEX [SOSP'23], BYTECHECKPOINT [NSDI'25]
  - Rapid reconfiguration of KVCache (AQUEDUCT [SRDS'26])
- Need performance models
  - Complex behavior, sensitive on workload, caches

Capacity estimation



Autoscaling architecture



Prompt/context

Policies:
- Prompt-aware routing
- Scale-up/down (based on SLO)
- Reconfigure vLLM

Router/controller

SLO (TTFT, TBT targets)

Telemetry

Prompt/context

Pool 1

Pool 2

KVCache

RDMA

Pool 1

Pool 2

Performance modeling
Online / Offline
(benchmarks, e.g., MLPerf,
vLLM bench, AIPerf)

Prefill GPU server pool

Decode GPU server pool

vLLM

Llama3 70B

TTFT, TBT
measurements

Shared KVCache

KVCache store

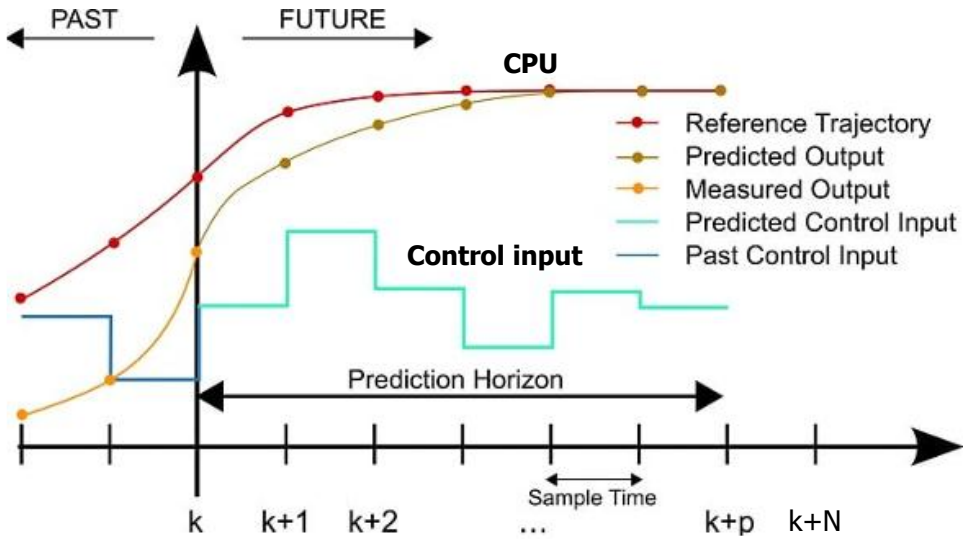
Model Predictive Control (MPC)

Advanced control strategy, optimizes control actions over a time horizon

$$y_{k+1} = ay_k + bu_k$$

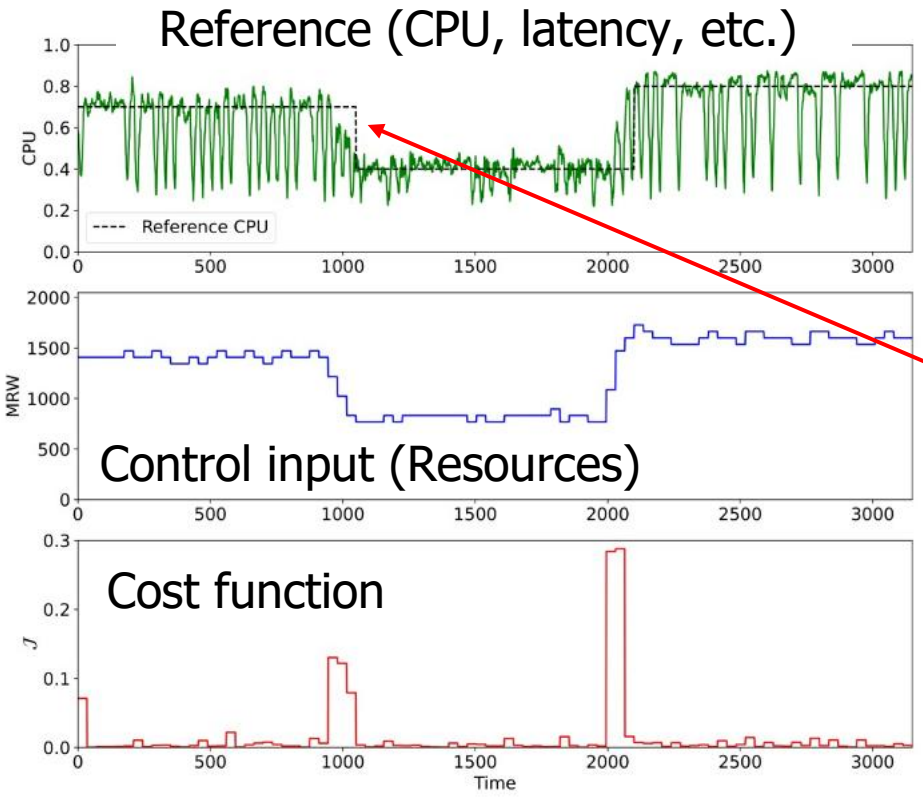
$$\mathbf{Y} = \mathbf{F}y_k + \mathbf{GU}$$

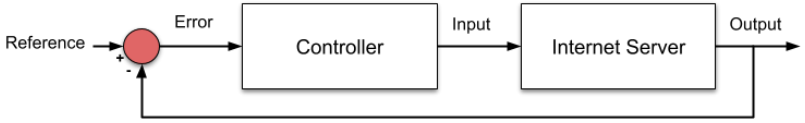
Cost Function: $$\mathcal{J} = (\mathbf{R} - \mathbf{Y})^T \mathcal{Q}(\mathbf{R} - \mathbf{Y}) + \mathbf{\Delta U}^T \mathcal{M} \mathbf{\Delta U}$$

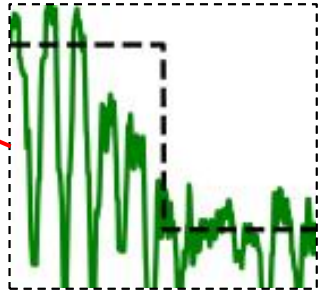


Term	Description
R	References
Y	Control Outputs
U	Control Inputs
Q, M	Weights

Controlling Internet Services (MPC)







- Reference Points: 0.7, 0.4 and 0.8
- Prediction Horizon N = 3

Model Predictive Control of Internet servers using QC [ACSOS'24]

Takeaways

- Autoscaling of LLM inference servers
  - Feedback control at router, reactive or proactive
  - Triggers: Resource utilization, tail latency
  - Key policy: prefix-cache-aware routing
  - Provision sufficient server pools to absorb bursts
- Caching is key
  - Multiple tiers, high speed transfers over the network
  - However, cached KV entries have limited lifetime
  - Encourage prefix reuse as much as possible
  - Exploit synergy between routing and caching policy

Thank you / Questions?



George T. Stamatiou, PhD



E. Papageorgiou, PhD candidate



E. Ntallaris, PhD candidate

QUADS

Quantum-Enhanced Adaptive Computer Systems

H.F.R.I.
Hellenic Foundation for Research & Innovation

FORTH
INSTITUTE OF COMPUTER SCIENCE

UOC
CSD

We thankfully acknowledge funding by the Hellenic Foundation for Research and Innovation (H.F.R.I.), under the 3rd Call for Research Projects to Support Faculty Members & Researchers (HFRI-FM3-25389)