CS 744: NEXUS

Shivaram Venkataraman

Fall 2021

ADMINISTRIVIA

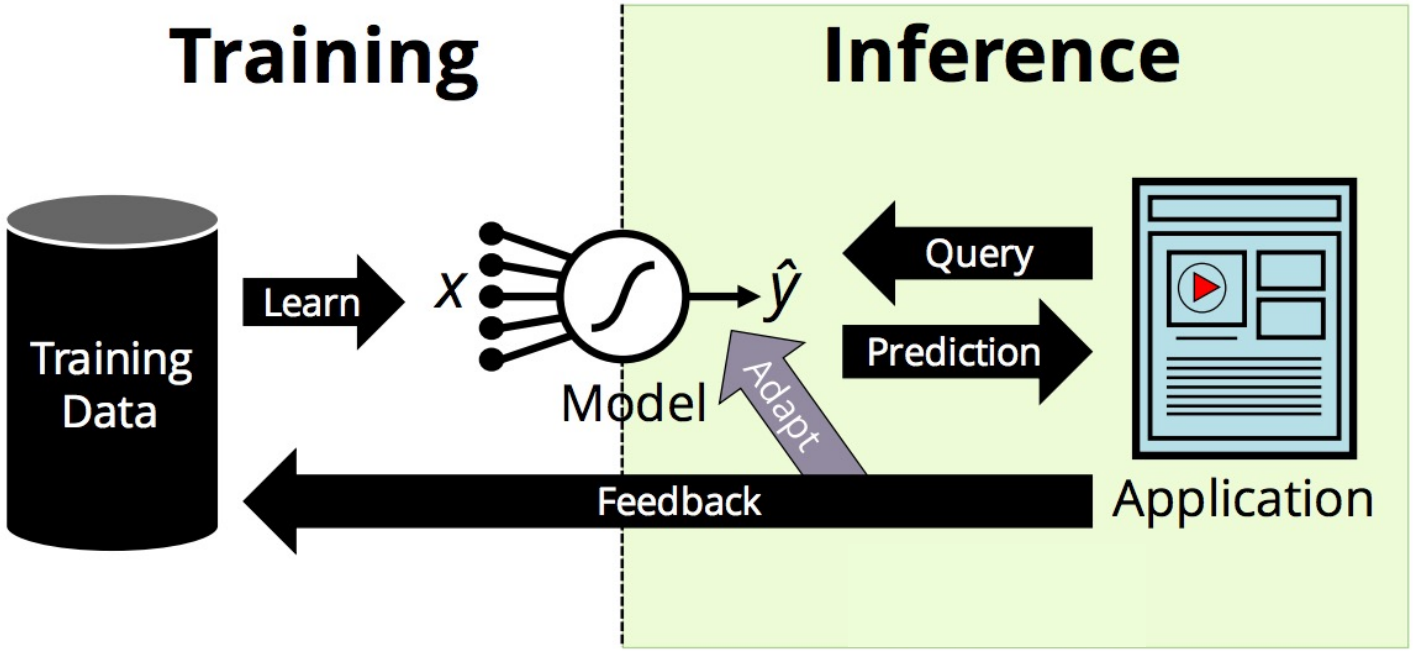
Course Project Proposals

- Due Oct 25!
- See Piazza for template

Midterm details

- Oct 28th: Includes papers from Datacenter as a Computer to Nexus
- Open book, open notes
- Held in class time 9.30-10.45am Central Time

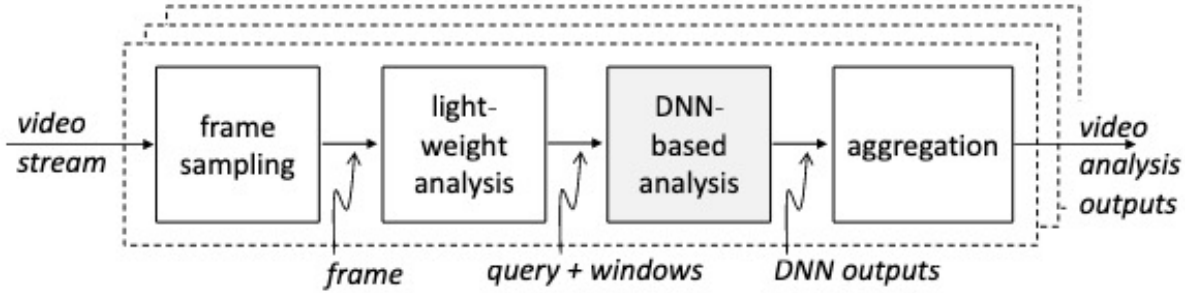
MACHINE LEARNING: INFERENCE



EXAMPLE APPLICATION

Video analysis service

- Thousands of streams, thousands of tenants
- Each stream is processed by a DNN-based "query"
- Latency SLOs (10s to 100s of ms)



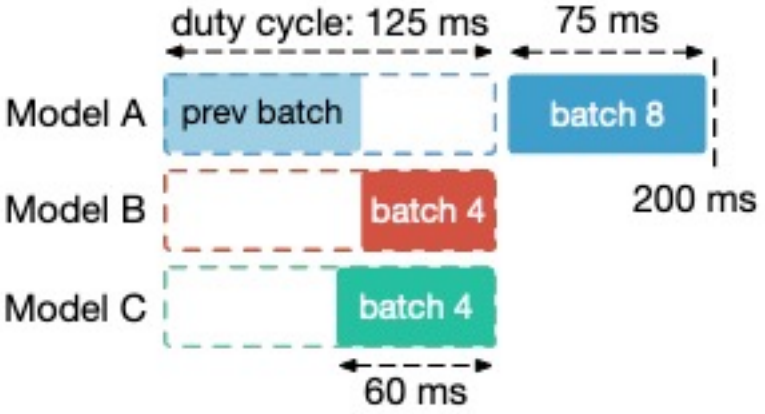
GOAL: HIGH GPU UTILIZATION

Placement

Batching

$$\text{batch\_lat}(b) = \alpha b + \beta,$$

SCHEDULING BATCHED EXECUTION



(b) Residual workload

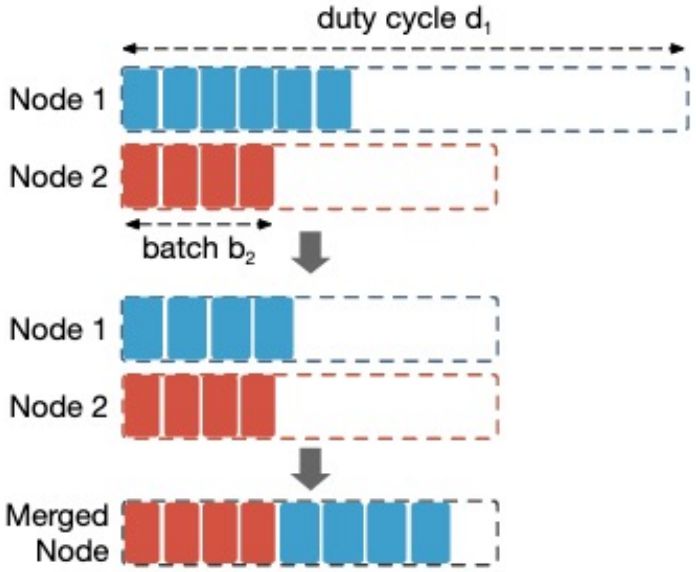
Model A			Model B			Model C		
Batch	Lat	Req/s	Batch	Lat	Req/s	Batch	Lat	Req/s
4	50	80	4	50	80	4	60	66.7
8	75	107	8	90	89	8	95	84
16	100	160	16	125	128	16	125	128

BATCH-AWARE SCHEDULING

Inputs: Request rate, SLO for each model, Profiles at batch size

Approach: Allocate "full" GPUs based on load. Handle residuals

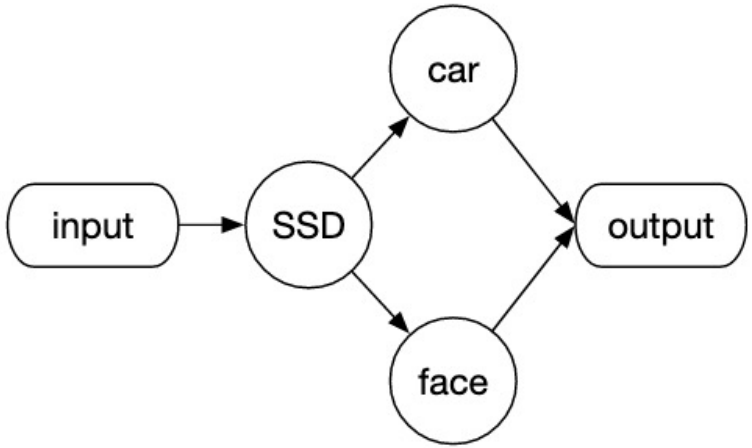
Greedy Approximation



HANDLING COMPLEX QUERIES

Challenge:

How do we set latency SLOs for complex queries?



SCHEDULING COMPLEX QUERIES

Query Analysis to determine latency SLO splits

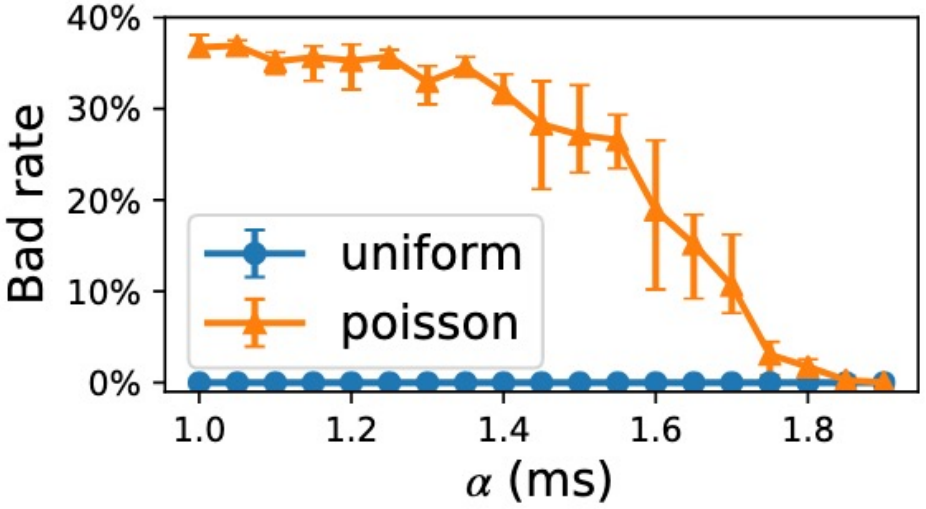
Inputs: Models with request rate R_i latency SLO L

$$\underset{\{b_v\}}{\text{minimize}} \quad \sum_v R_v l_v(b_v)/b_v$$

$$\text{subject to} \quad \sum_{u:M_{\text{root}} \rightsquigarrow M_v} l_u(b_u) \leq L \qquad \forall v \in \text{leaf}$$

ADAPTIVE BATCHING

Clipper: Adapt the batch size based on the oldest request

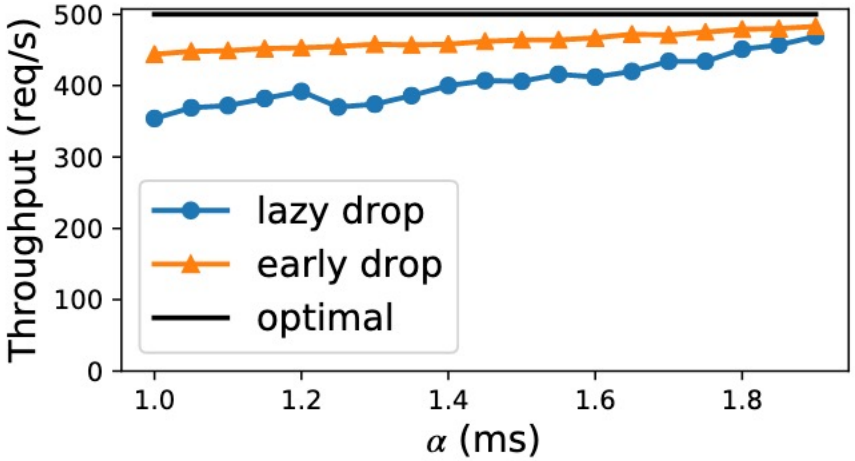


BATCH-AWARE DISPATCH

Early-dropping scheme

1. Scans queue using sliding window of batch size

2. Stop at the first request with that can execute *entire window*



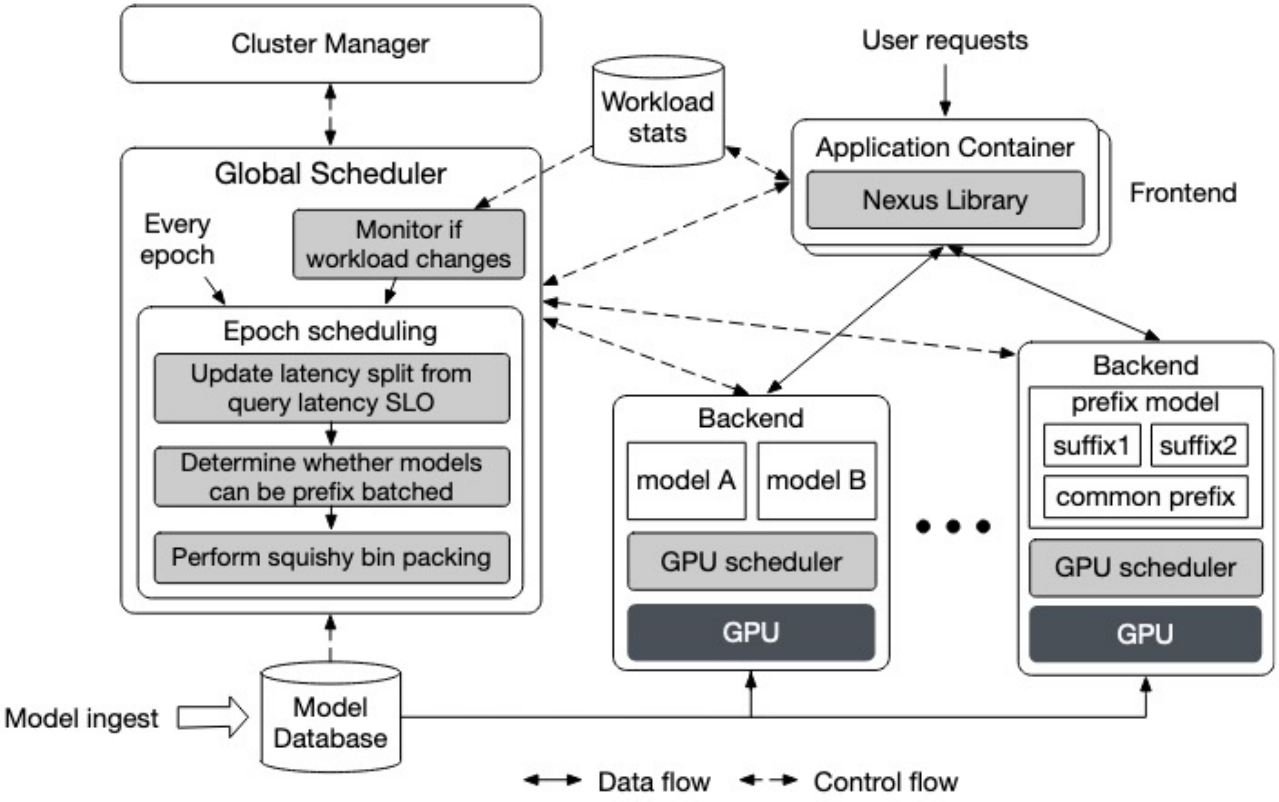
OTHER FEATURES

Prefix Batching

GPU Multiplexing

Overlapping CPU and GPU computation

NEXUS ARCHITECTURE



SUMMARY

- ML Inference goals: latency SLO, GPU utilization
- Nexus: Handle multiple tenants, multiple DNNs
- Schedule using squishy bin packing
- Breakdown SLO for complex queries, adaptive batching

Pytorch Distributed

DataParallel Training API
Overlap compute, communication

PipeDream

Generalize parallelism: Pipeline parallel
Reduce network, maintain consistency

Ray

Reinforcement learning applications
Actors and tasks, Local and global scheduler

Pollux

Scheduler ML training jobs in a cluster
Co-adaptive scheduling to set batch size, LR

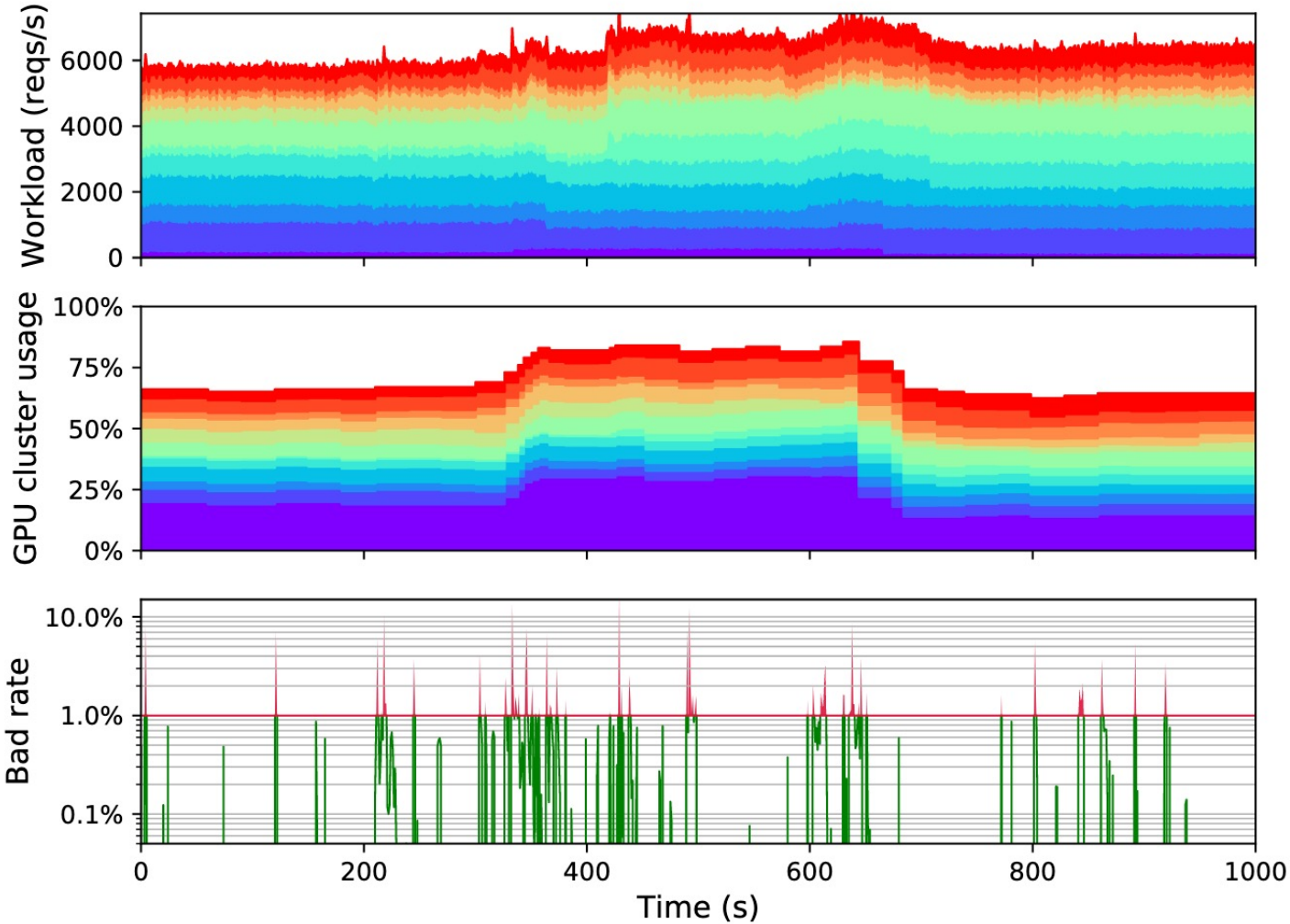
Nexus

System for ML Inference
Meet latency SLOs while ensuring high utilization

DISCUSSION

https://forms.gle/XQ4CfNzTTFsSrVv7A

Consider a scenario where you have a model that takes variable amount of time depending on the input. For example if a frame contains 100 cars it takes 250ms to process but if the frame has 1 car then it finishes in 10ms. What could be one shortcoming in using Nexus to schedule this model?


Workload (reqs/s)
6000
4000
2000
0
GPU cluster usage
100%
75%
50%
25%
0%
Bad rate
10.0%
1.0%
0.1%
0
200
400
600
800
1000
Time (s)

Next class: SQL