CS 744: NEXUS

Shivaram Venkataraman
Fall 2022
Course Project Proposals
- Due Oct 26!
- See Piazza for template

Midterm details
- Oct 27th: 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

Training

Inference

Training Data

Learn

Model

Adapt

Feedback

Query

Prediction

Application
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)
SCHEDULING GOAL: HIGH GPU UTILIZATION

Placement

Batching

\[ batch_{\text{lat}}(b) = \alpha b + \beta, \]
SCHEDULING BATCHED EXECUTION

Target tputs A: 64, B: 32, C: 32 req/sec. SLO: 250ms

<table>
<thead>
<tr>
<th></th>
<th>Model A</th>
<th></th>
<th>Model B</th>
<th></th>
<th>Model C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Batch</td>
<td>Lat</td>
<td>Req/s</td>
<td>Batch</td>
<td>Lat</td>
</tr>
<tr>
<td>Model A</td>
<td>4</td>
<td>50</td>
<td>80</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>Model B</td>
<td>8</td>
<td>75</td>
<td>107</td>
<td>8</td>
<td>90</td>
</tr>
<tr>
<td>Model C</td>
<td>16</td>
<td>100</td>
<td>160</td>
<td>16</td>
<td>125</td>
</tr>
</tbody>
</table>
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 $L$

\[
\begin{align*}
\text{minimize} & \quad \sum_{\nu} R_{\nu} \frac{l_{\nu}(b_{\nu})}{b_{\nu}} \\
\text{subject to} & \quad \sum_{u: M_{\text{root}} \sim M_{\nu}} l_u(b_u) \leq L \quad \forall \nu \in \text{leaf}
\end{align*}
\]
Clipper: Adapt the batch size based on the oldest request in the queue

\[
\text{batch\_lat}(b) = \alpha b + \beta,
\]
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*

![Graph showing throughput vs. alpha (ms)]
OTHER FEATURES

Prefix Batching

GPU Multiplexing

Overlapping CPU and GPU computation
NEXUS ARCHITECTURE

Cluster Manager

Global Scheduler
- Every epoch
- Monitor if workload changes
- Epoch scheduling
  - Update latency split from query latency SLO
  - Determine whether models can be prefix batched
  - Perform squishy bin packing

Workload stats

Application Container
- Nexus Library

User requests

Frontend

Model Database

Model ingest

Data flow

Control flow

Backend
- prefix model
  - suffix1
  - suffix2
  - common prefix
- GPU scheduler
- GPU

GPU

model A

model B
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
DISCUSSION

https://forms.gle/PtEaiF4casfZm2JY6
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?
Next class: SQL
Coming soon
  Project Introductions
  Midterm I