Problem

- Underutilization of GPUs in exclusive deployment of DL jobs
  - Commodity GPUs have fixed compute and memory capacity
  - DL jobs have varying resource requirements
  - Mismatch required and available resources

- Sharing of resources between DL jobs improves resource utilization
  - Uncontrolled sharing can result in unpredicted performance
  - SLOs of some jobs may get violated

Problem Statement: How to share GPUs between DL jobs, increasing GPU efficiency while maintaining SLOs?

Challenges

- Resource utilization can vary significantly during a given DL job
  - Idle/Partially utilized GPU → Sharing opportunities
  - Fully utilized GPU → Performance degradation when shared

- Over-sharing can result in errors: e.g. Out of Memory (OOM)
- Under-sharing can result in wastage of resources

Solution

- Estimate resource footprint for various operations in a DL job
  - Profile control flow graphs (negligible overhead)

- Our Solution: Herald (fine-grained space-time GPU sharing)
  - Identify “light” compute operations for spatial sharing
  - Avoid spatial-sharing for compute-intensive (“heavy”) operations
  - Time-share for “heavy” operations: Prioritize SLO-sensitive jobs

Evaluation

- RNNLM Training + Transformer Inference
  - Training: \( \uparrow 4.4 \times \)
  - Inference: \( \uparrow 1.2 \times \)

- VGG16 Training + Transformer Inference
  - Training: \( \uparrow 3.4 \times \)
  - Inference SLO: \( \uparrow 1.2 \times \)

Integration directly with TensorFlow source for ease of use in production