Resource Aware GPU Scheduling in Kubernetes Infrastructure

Aggelos Ferikoglou, Dimosthenis Masouros, Achilleas Tzenetopoulos, Sotirios Xydis, Dimitrios Soudris

Microprocessors and Digital Systems Laboratory, ECE, National Technical University of Athens, Greece

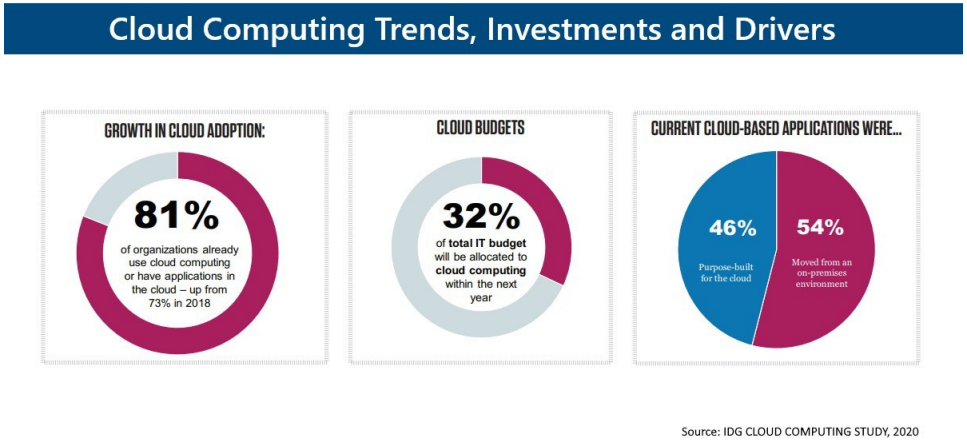
19-01-2021

Outline

- **Introduction & Trends**
- **Motivational Observations & Analysis**
- **Resource-aware GPU Scheduling**
- **Experimental Setup & Evaluation**
- **Conclusion**

Introduction & Trends

- **Trends: Cloud Migration & Machine Learning Hype**

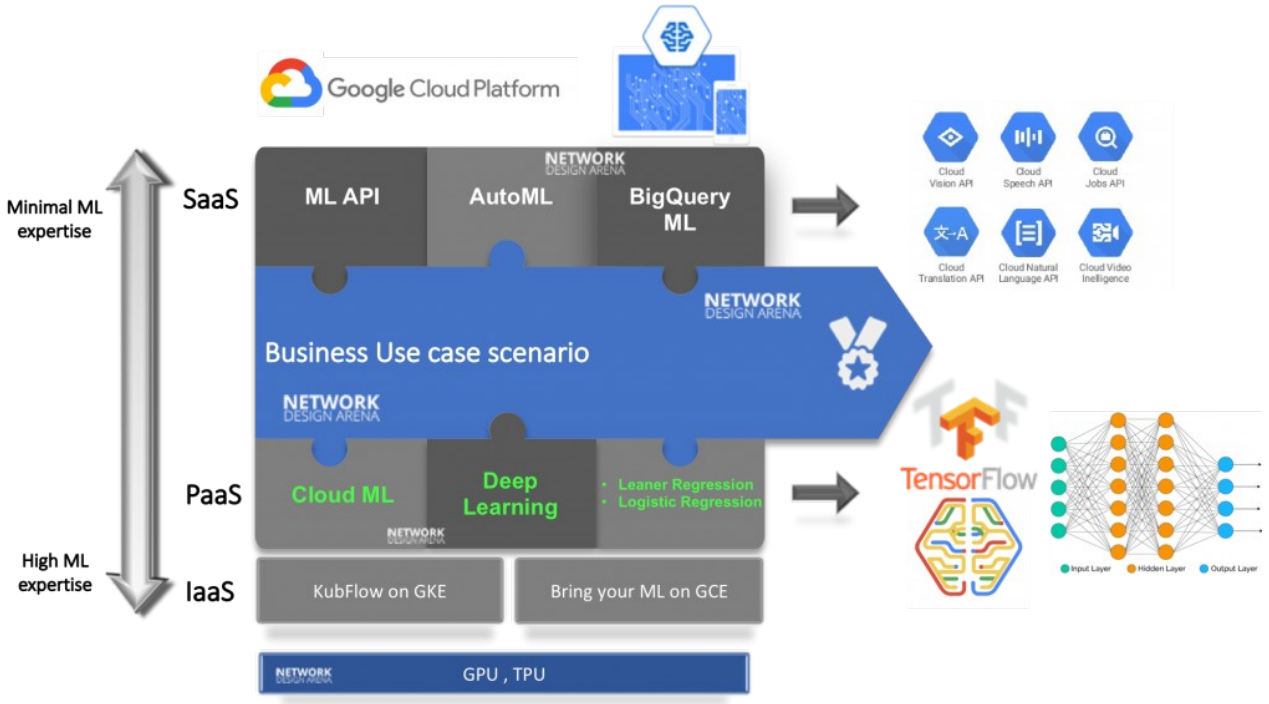


- **Key Technologies:**
  - Docker (Containerization)
  - Kubernetes (Container Orchestration)

Introduction & Trends

- **ML workloads on cloud**
  - How can we manage the computational demands ?

    USE OF ACCELERATORS (GPUs, FPGAs, TPUs, ASICs)



http://www.netdesignarena.com/index.php/2019/01/21/machine-learning-on-google-cloud-platform-simplified/

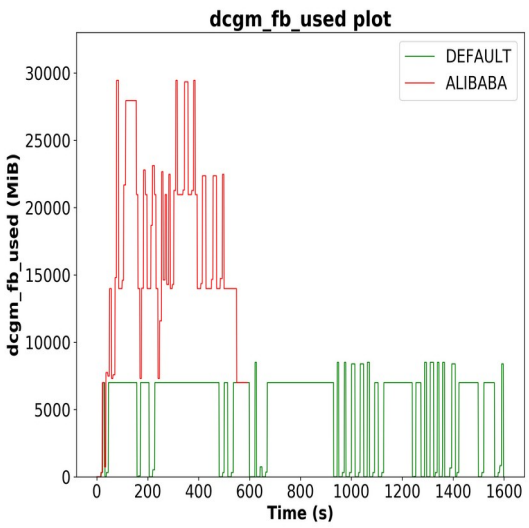
Motivational Observations & Analysis

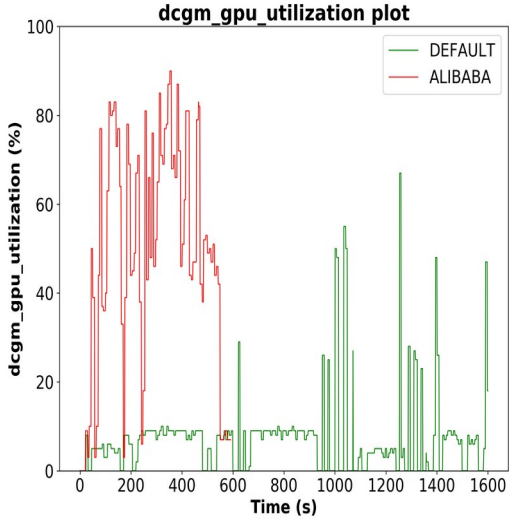
- **How does Kubernetes handle GPUs ?**
  - Binds the whole GPU to an application
- **How does Alibaba Cloud handle GPUs ?**
  - Exposes GPU memory as an extended resource in Kubernetes

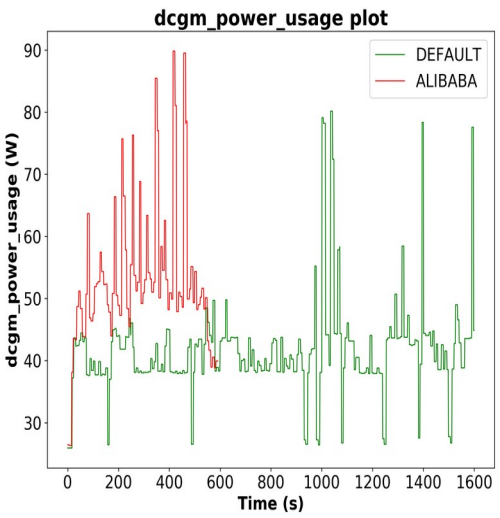
  ⇒ ENABLES GPU SHARING !

https://github.com/AliyunContainerService/gpushare-scheduler-extender

Motivational Example: GPU Sharing Advantages

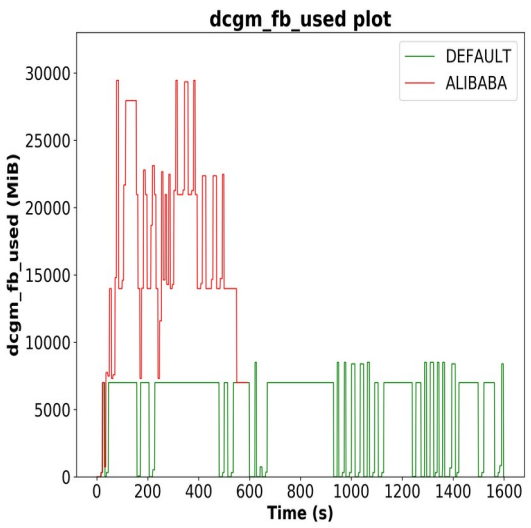


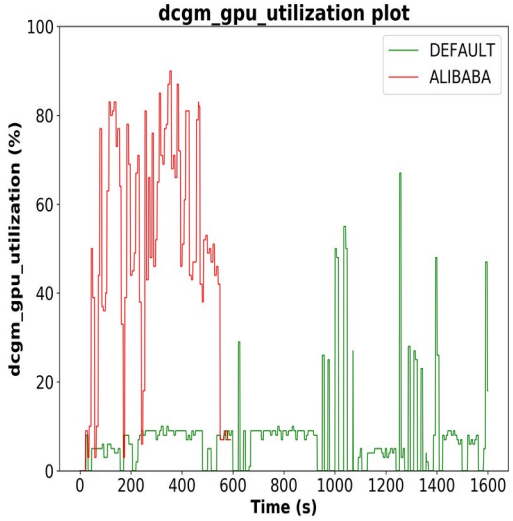


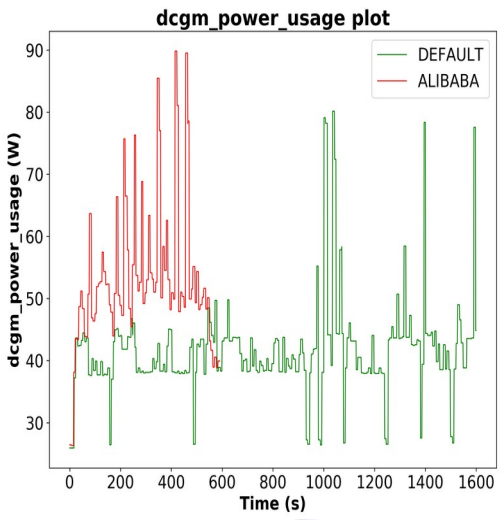


- x3.24 higher average memory usage
- x6.8 higher average utilization percentage
- x1.28 higher average power usage
- 52.8% decrease of the average energy consumption
- x2.67 faster workload execution

Motivational Example: GPU Sharing Advantages







- x3.24 higher average memory usage
- x6.8 higher average utilization percentage
- x1.28 higher average power usage
- 52.8% decrease of the average energy consumption
- x2.67 faster workload execution

What about GPU memory over-provisioning from users ?

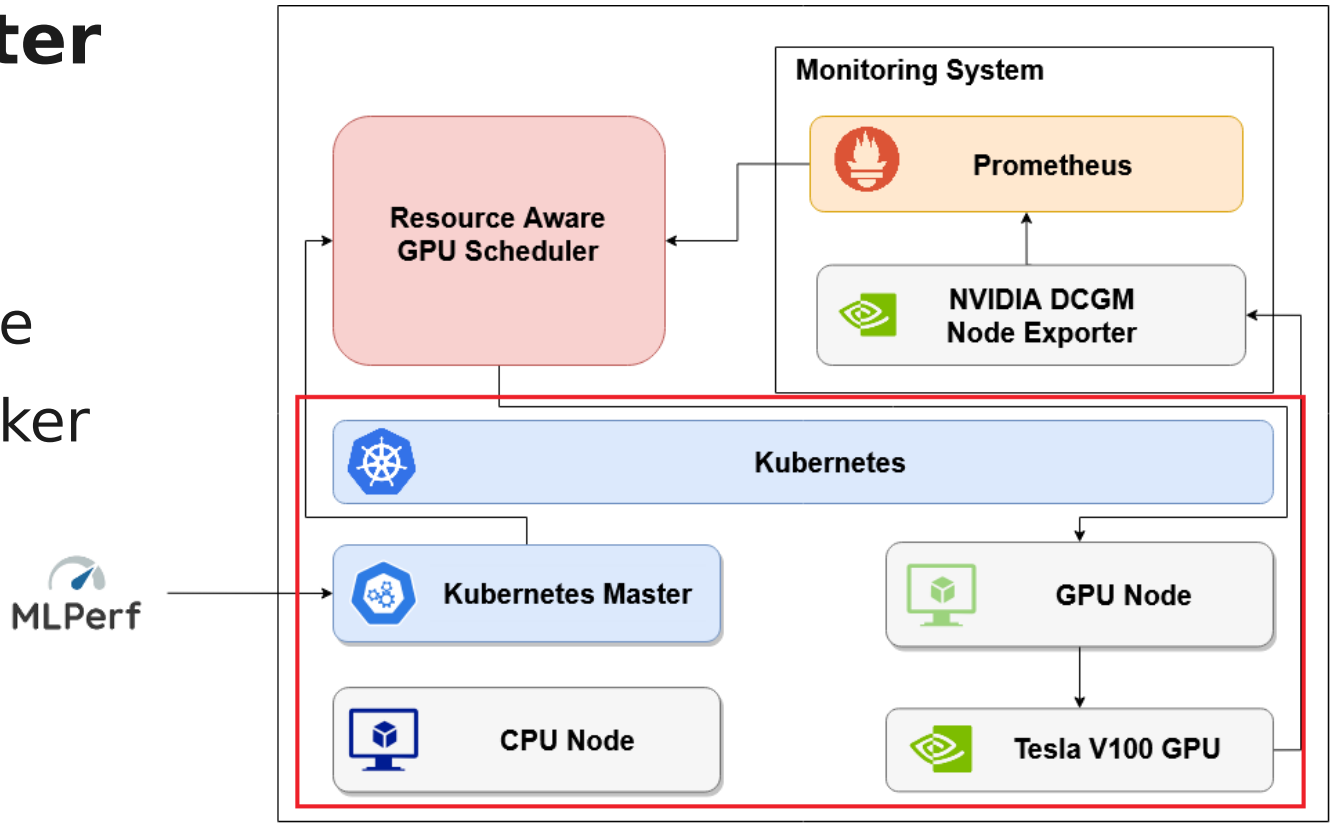
Not Only a Users' Problem...

- **State-of-the-art frameworks bind a GPU per application !**
  - Tensorflow by default binds the whole GPU per application
  - Spark 3.0 with Rapids plugin by default binds the 90% of the GPU memory per executor

+

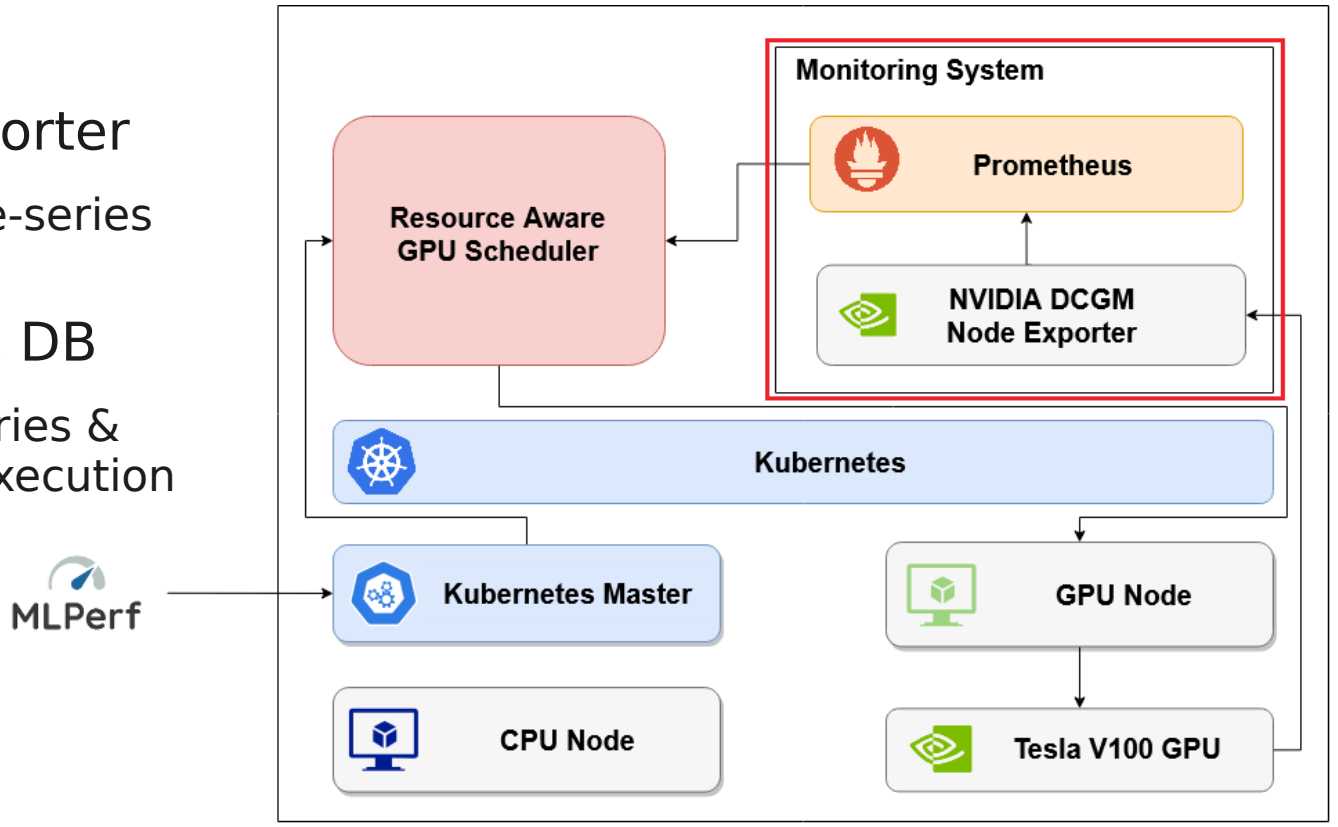
Resource Aware GPU Scheduling Framework

- **Kubernetes Cluster with 3 Nodes:**
  - Master node
  - CPU only worker node
  - GPU provisioned worker node



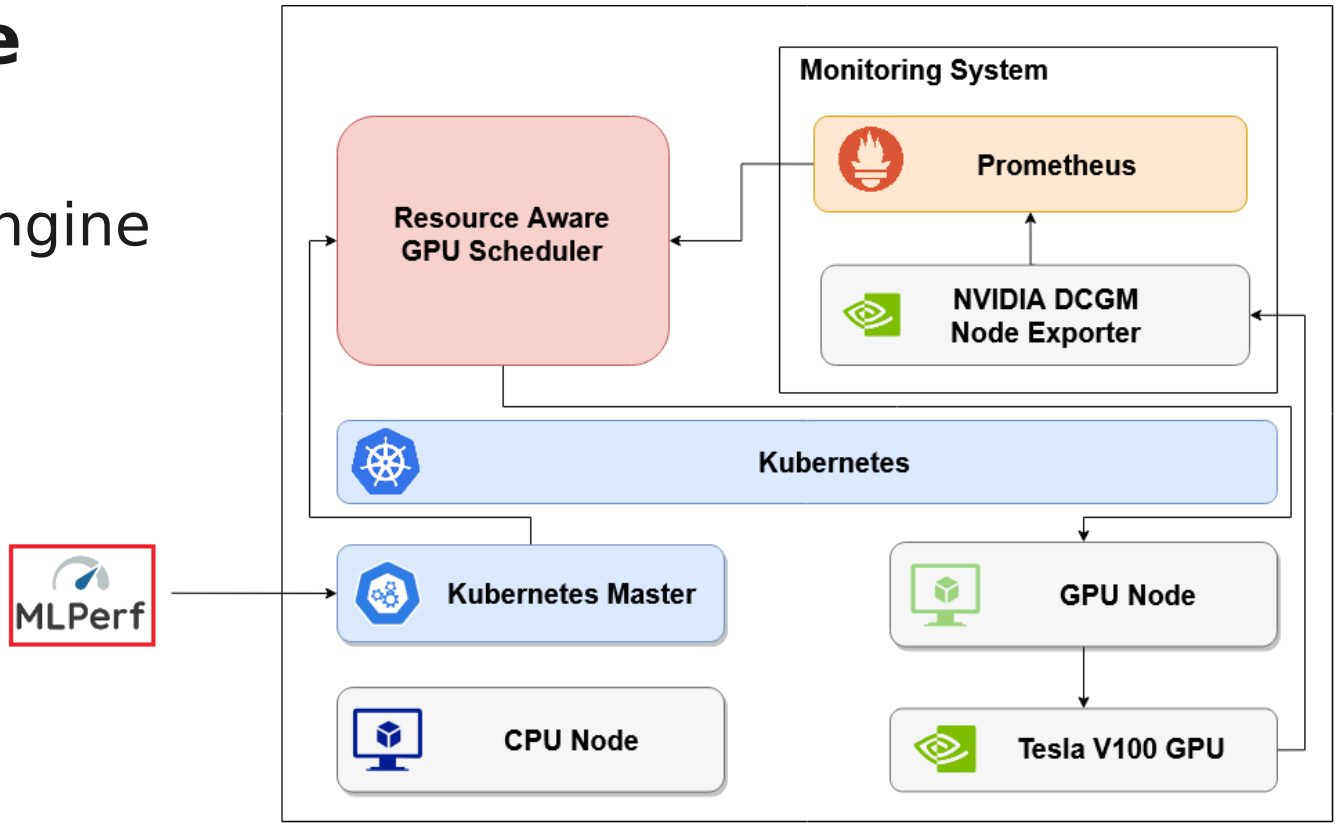
Resource Aware GPU Scheduling Framework

- **Monitoring System**
  - NVIDIA DCGM Node Exporter
    - Exports GPU metrics in time-series format
  - Prometheus Time-series DB
    - Stores GPU metrics time-series & provides PromQL for query execution

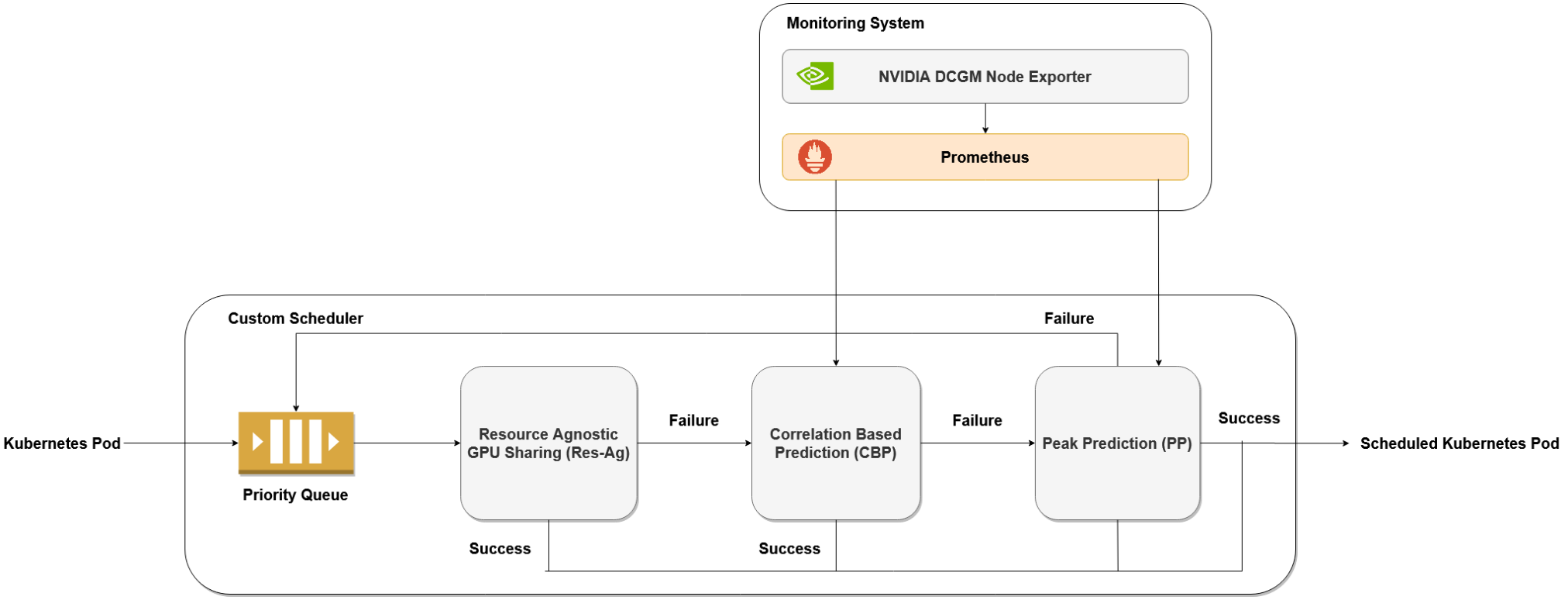


Resource Aware GPU Scheduling Framework

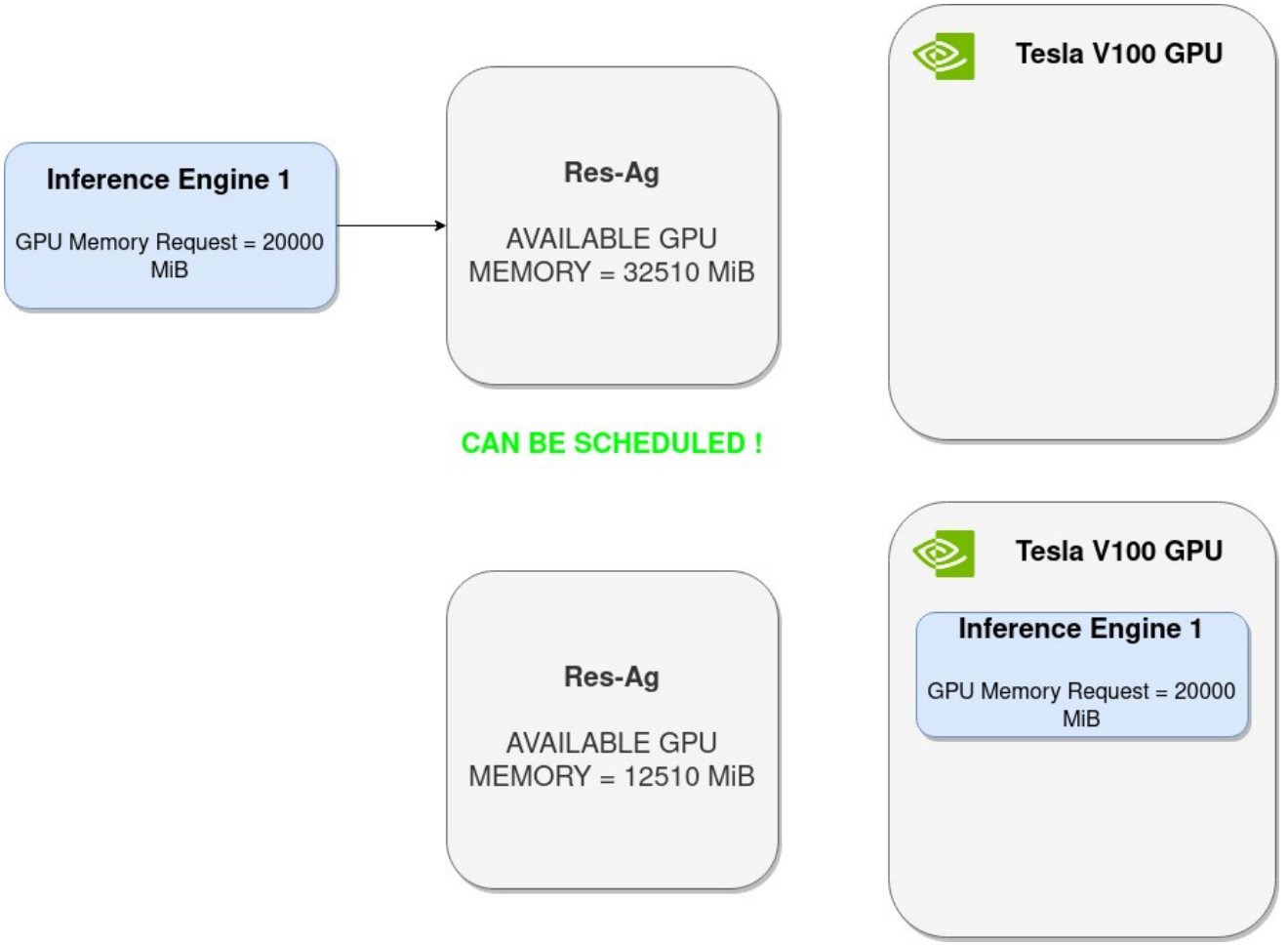
- **MLPerf Inference Benchmark Suite**
  - Used for inference engine workload creation



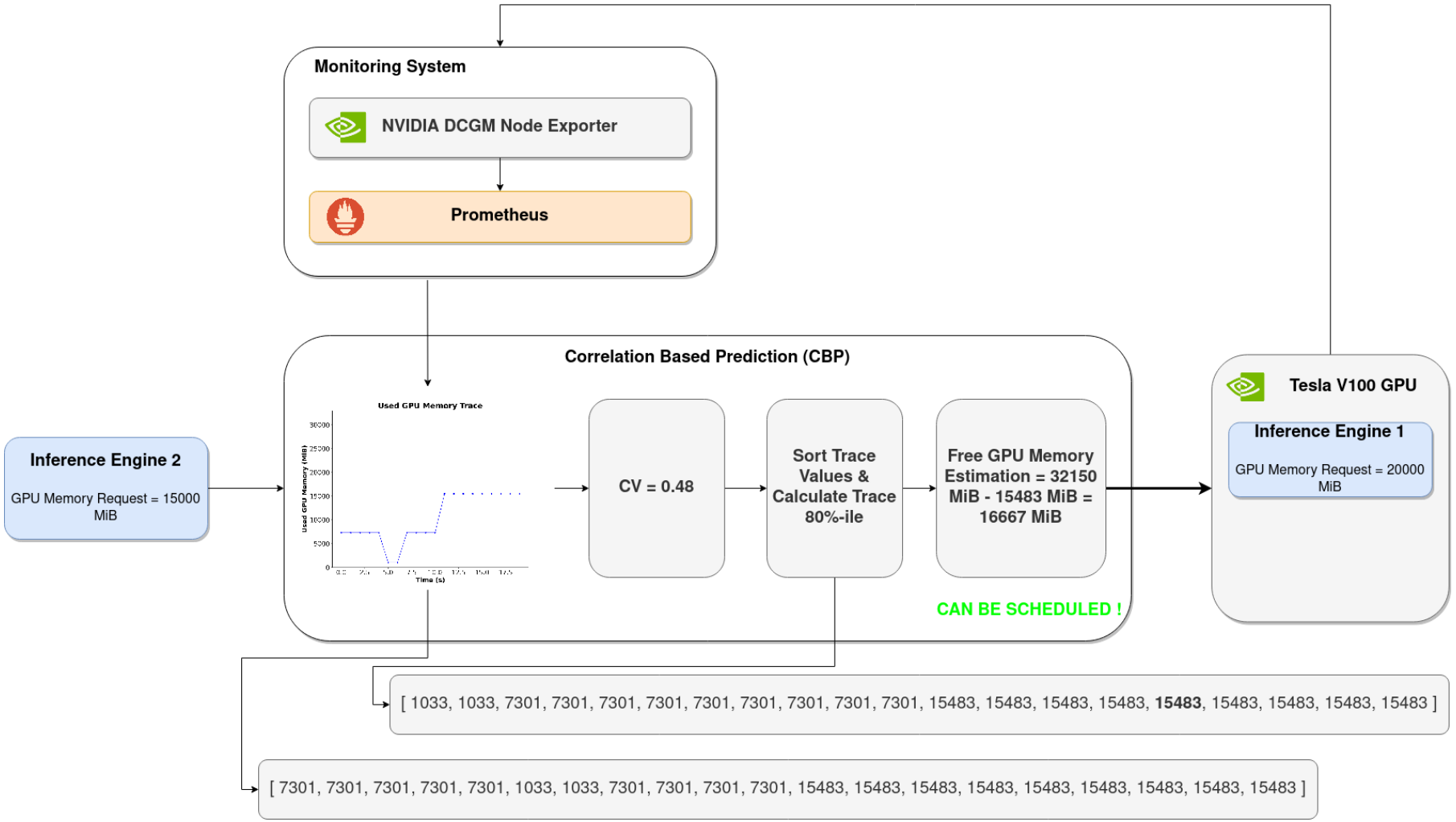
Resource Aware GPU Scheduler Overview



Resource Agnostic GPU Sharing (Res-Ag)

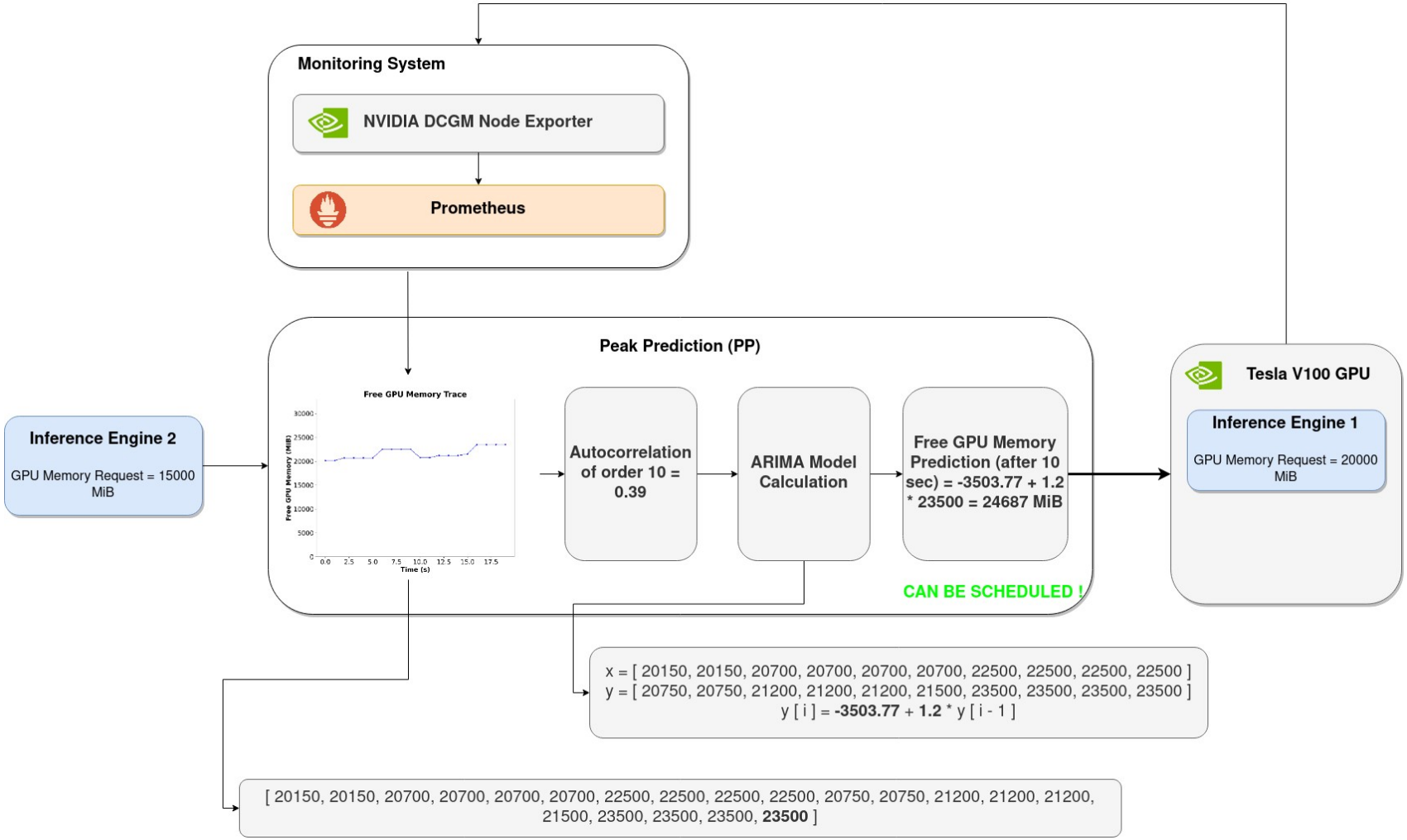


Correlation Based Prediction (CBP) *



* https://ieeexplore.ieee.org/document/8891040

Peak Prediction (PP) *



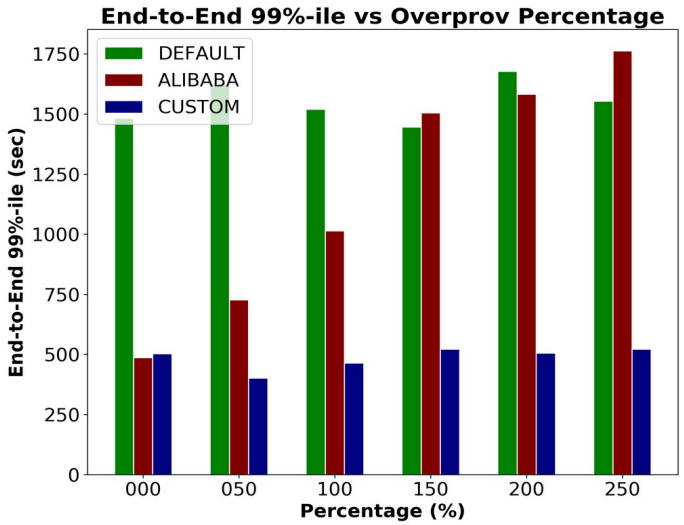
* https://ieeexplore.ieee.org/document/8891040

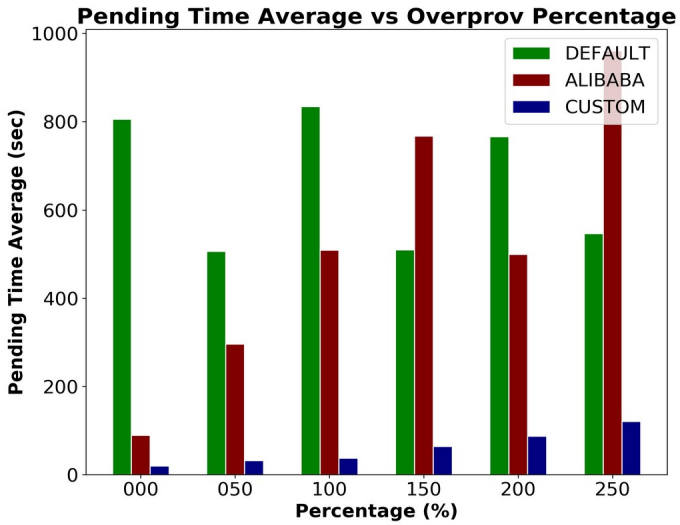
Experimental Setup & Evaluation

- **Evaluation through a rich set of comparative experiments**
- **A workload consists of:**
  - A set of different MLPerf Inference Engines
  - An interval that defines the Inference Engine arrival pattern
- **In each experiment the exact same workload, for different over-provisioning percentages, was fed to:**
  - Kubernetes GPU Scheduler Extension
  - Alibaba Cloud GPU Scheduler Extension
  - Resource-aware GPU Scheduler

Model	Dataset	Queries/Engine (#Engines)
mobilenet	Imagenet	1024(2), 2048(2)
mobilenet quantized	Imagenet	256(2), 512(2)
resnet50	Imagenet	4096(2), 8192(2)
ssd-mobilenet	Coco (Resized 300x300)	128(3), 1024(2)
ssd-mobilenet quantized finetuned	Coco (Resized 300x300)	64(2), 1024(2)
ssd-mobilenet symmetrically quantized finetuned	Coco (Resized 300x300)	512(2), 4096(2)

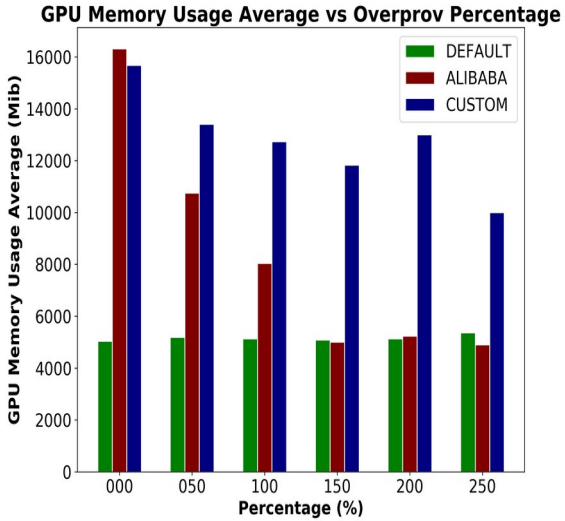
Quality of Service Metrics

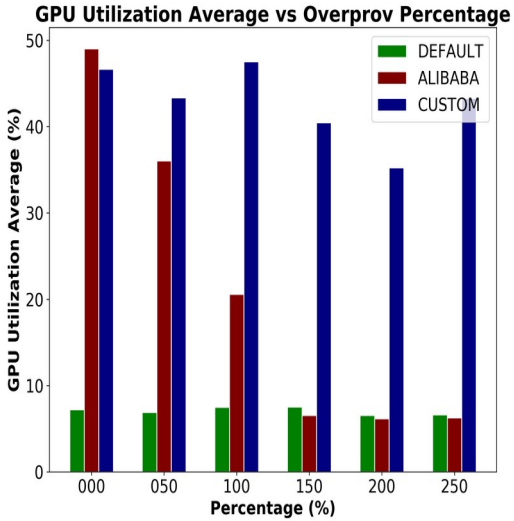


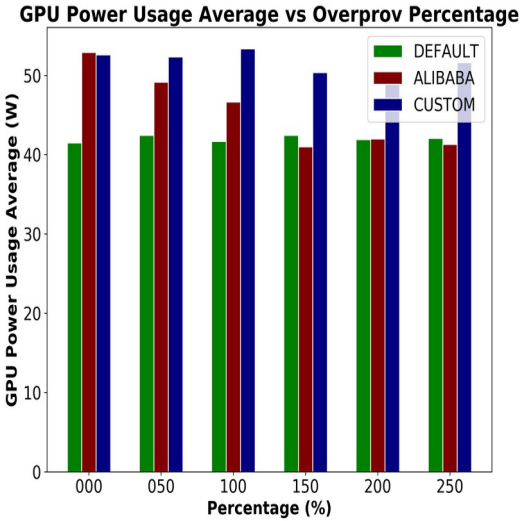


- **Offers lower End-to-End Inference Engine 99%-ile Execution**
  - **x3.2** from Kubernetes GPU scheduler on an average
  - **x2.4** from Alibaba Cloud GPU scheduler on an average
- **Offers lower Inference Engine Pending Time Average**
  - **x11** from Kubernetes GPU scheduler on an average
  - **x8.6** from Alibaba Cloud GPU scheduler on an average

GPU Resource Utilization Metrics







- **Offers higher GPU Memory Usage Average**
  - **x2.5** from Kubernetes GPU scheduler on an average
  - **x1.5** from Alibaba Cloud GPU scheduler on an average
- **Offers higher GPU Utilization Percentage Average**
  - **x6.1** from Kubernetes GPU scheduler on an average
  - **x2.1** from Alibaba Cloud GPU scheduler on an average
- **Leads to higher GPU Power Usage Average**
  - **x1.2** from Kubernetes GPU scheduler on an average
  - **x1.1** from Alibaba Cloud GPU scheduler on an average

Conclusion

- **Designed a resource-aware GPU scheduling framework for Kubernetes Inference clusters**
- **Our framework offers:**
  - **x2.4** lower end-to-end inference engine execution time 99%-ile
  - **x1.5** higher GPU memory usage average
  - **x2.1** higher GPU utilization percentage average

  from Alibaba Cloud GPU Scheduler Extension

Thank you, Questions ?