ALGORITHMIA

Sharing a GPU Among Multiple Containers

Patrick McQuighan, Senior Software Engineer

About Me

- Joined Algorithmia in 2015
- Machine Learning model deployment, serving, scaling, and management
- 100,000+ users
- Millions of API calls to tens of thousands of models per day
- Sharing GPUs since 2016

Is sharing GPU to multiple containers feasible? #52757

Open **tianshapjq** opened this issue on Sep 19, 2017 · 63 comments

tianshapjq commented on Sep 19, 2017 Contributor •••

Is this a BUG REPORT or FEATURE REQUEST?: feature request
/kind feature

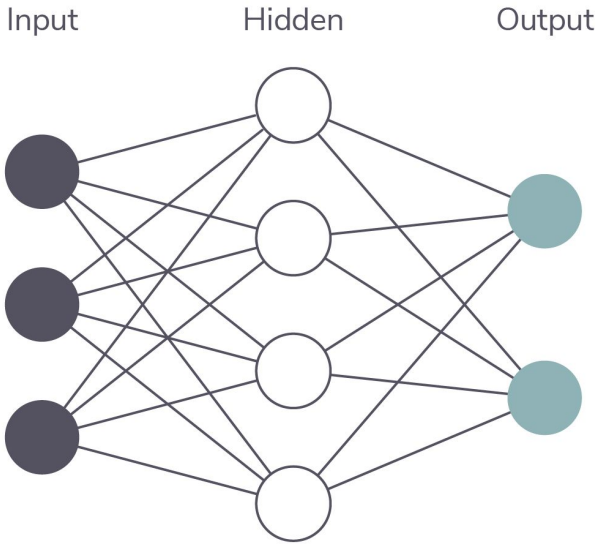
What happened:
As far, we do not support sharing GPU to multiple containers, one GPU can only be assigned to one container at a time. But we do have some requirements on achieving this, is it feasible that we manage GPU just like CPU or memory?

What you expected to happen:
sharing GPU to multiple containers just like CPU and memory.

👍 109 🎉 3 ❤️ 30 🚀 3

Why GPUs?

- Neural networks on CPUs
  - Days to train
- GPUs can parallelize these operations



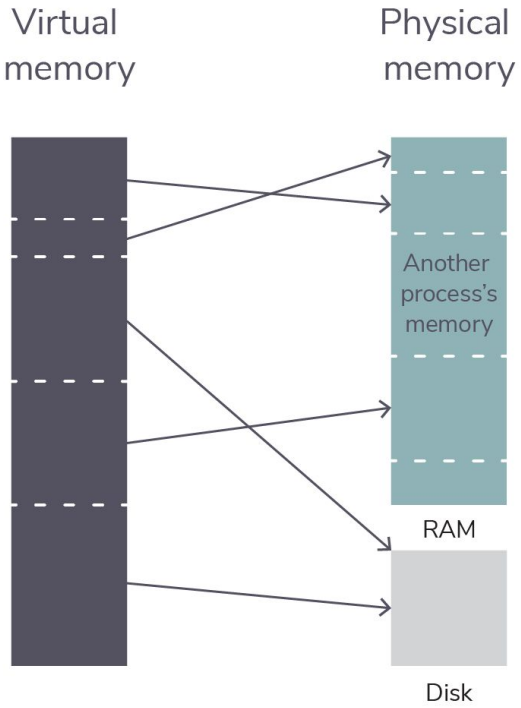
Why sharing?

- **2-10x** more expensive!
- Maximizing hardware utilization
  - Limited GPU instance capacity in cloud providers

What’s the problem?

Virtualization

- GPU does not have virtual memory
- Host RAM does:
  - Isolation between processes
  - Can overprovision



Parallelism

- Kernel **control groups** (cgroup) allow limiting resources
  - Mechanism used to enforce pod limits
  - No cgroup for GPU resources
- On GPU, scheduling is **time-sliced**
  - Different contexts must be swapped on/off the GPU
  - Overhead cost to run multiple jobs

NVIDIA Multi-Process Service (MPS)

- Single shared context for GPU
  - Lower scheduling overhead
  - No swapping
  - **Volta** architecture can have memory isolation
- Errors from one client will halt activity of **all** clients
- Great for **some** workloads

Kubernetes Support

Device Plugins

- Hosts must have all drivers and runtimes setup
- Nodes declare resources
- Pods request resources
- Does not allow sharing devices

https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/

Kubernetes Extensions

- Kubernetes is **flexible**
  - Extended resources
  - Scheduler extensions

https://kubernetes.io/docs/tasks/administer-cluster/extended-resource-node/

https://kubernetes.io/docs/concepts/extend-kubernetes/extend-cluster

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

Should you share devices?

- Sharing devices is not a magic wand
- May save money but cause operational nightmares
  - Performance monitoring
  - Tracking actual resource consumption
  - Additional safeguards and monitoring

Wrap-Up

- Isolation and parallelism are **hard**
- **Flexibility** of Kubernetes makes anything possible
- Monitoring your workloads is **critical** before you can fully benefit

Thank you!

Visit us at our booth S102

Download a whitepaper:
http://bit.ly/AlgoKubeCon

Contact us:
info@algorithmia.com