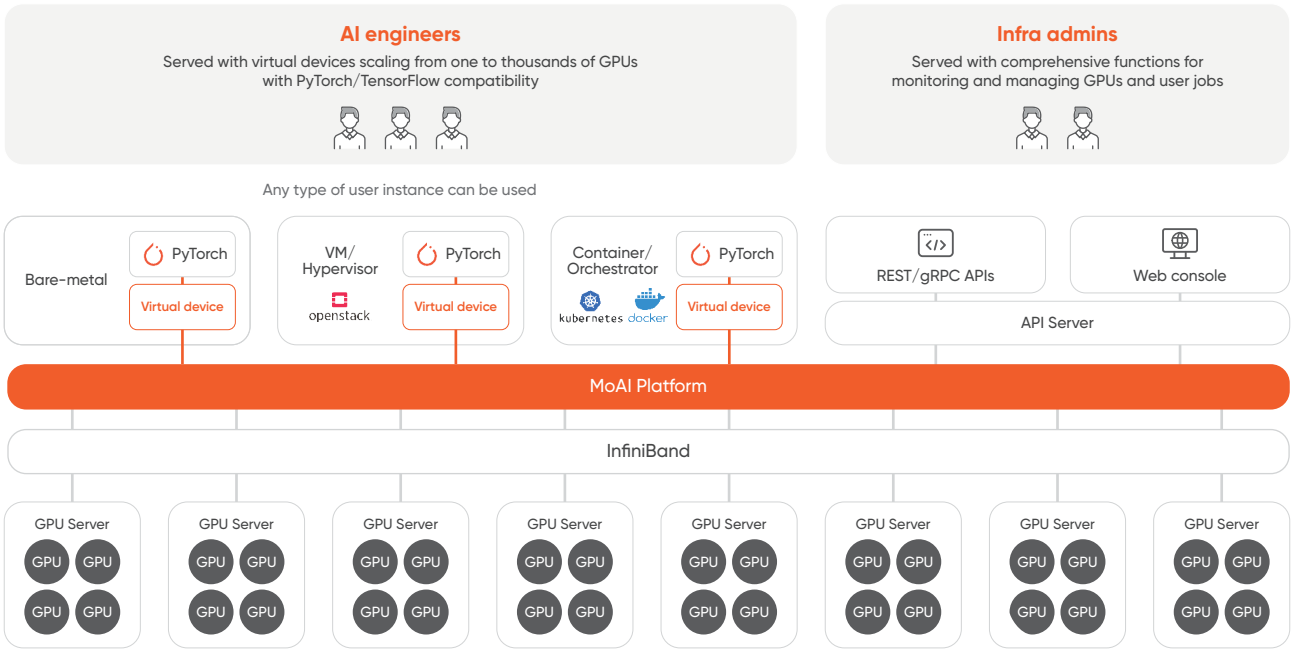
MoAI Platform

for Infrastructure Builders

MOREH

moreh.io

A virtualized and scalable GPU infrastructure for hyperscale AI



■ PyTorch/TensorFlow compatibility for AI engineers

End users of the MoAI platform can enjoy the same user experience as a physical GPU system and traditional AI software stacks. They do not need to insert or modify even a single line of existing source code. They also do not need to change the method of running a program. The architectural changes in the platform are completely hidden from the users.

■ GPU virtualization for higher utilization and failover

User applications are isolated from physical GPUs and run on virtual devices. The MoAI platform manages the mapping between virtual and physical devices. Its efficient resource scheduling drastically improves the average utilization of the cloud infrastructure. Its fault tolerance mechanism ensures GPU hardware failures do not interrupt user processes. The virtualization is supported by a lightweight software layer and has negligible impact on performance.

■ AMD GPU support

The MoAI platform is not bound to a specific hardware vendor and supports various GPU backends. Especially it has been largely optimized for AMD Instinct GPUs. AI infrastructure can be built in a cost-effective way without concerning software compatibility.

AMD ROCm **MoAI Platform primarily supports AMD ROCm**

The MoAI platform is now powering >2,000 GPUs for commercial IaaS/PaaS clouds and on-premise LLM training systems

Read the success story of Korea Telecom

Key Features of MoAI Virtual Device

- AI frameworks
  - PyTorch support (as cuda:0)
  - TensorFlow support (as GPU:0)
  - Triton Inference Server support with LLM optimizations
- Performance and scalability
  - Scalable to hundreds of petaflops and hundreds of terabytes of memory capacity using thousands of physical GPUs together
  - Automatic parallelization for any multi-billion or multi-trillion parameter model, for any (existing) PyTorch/TensorFlow application
  - Automatic performance optimization for individual GPUs and clustering
- Additional features
  - PyTorch extension libraries for LLMs such as FlashAttention
  - Automatic failover for GPU failures
  - CLI tools for device query and manipulation
- LLM development
  - Bundled with a rich set of LLM and multimodal model implementations written in PyTorch
  - Technical support and co-development for LLM training

```
(pytorch) ubuntu@vm:~$ moreh-smi
+-----------------------------------------------------------------------------------------------+
|                                                Current Version: 23.10.0  Latest Version: 23.10.0  |
+-----------------------------------------------------------------------------------------------+
|  Device  |         Name          |       Model      | Memory Usage  |  Total Memory  |  Utilization  |
+===============================================================================================+
|  * 0     |  MoAI Virtual Device  |  xlarge.512gb  |  214201 MiB    |  524160 MiB    |   86 %        |
+-----------------------------------------------------------------------------------------------+

Processes:
+-----------------------------------------------------------------------+
|  Device  |  Job ID  |    PID    |          Process          | Memory Usage  |
+=======================================================================+
|       0  |  268511  |  734106   |  python train_llama2.py   |  214201 MiB    |
+-----------------------------------------------------------------------+
```

```
(pytorch) ubuntu@vm:~$ python
>>> import torch
>>> torch.__version__
'2.0.1+cu118.moreh23.10.0'
>>> torch.cuda.device_count()
[info] Requesting resources for MoAI Virtual Device from the server...
[info] Initializing the worker daemon for MoAI Virtual Device...
[info] [1/1] Connecting to resources on the server (192.168.80.10:24162)...
[info] Establishing links to the resources...
[info] MoAI Virtual Device is ready to use.
1
>>> torch.cuda.get_device_name()
'MoAI Virtual Device'
>>> torch.cuda.get_device_properties(0)
_MorehDeviceProperties(name='MoAI Virtual Device', major=0, minor=0, total_memory=549621596160,
multi_processor_count=8)
>>> quit()
```

```
(pytorch) ubuntu@vm:~$ cd resnet
(pytorch) ubuntu@vm:~/resnet$ python train.py --save-model model.pt -b 384 --num-workers 8
...
[info] Requesting resources for MoAI Virtual Device from the server...
[info] Initializing the worker daemon for MoAI Virtual Device...
[info] [1/1] Connecting to resources on the server (192.168.80.14:24157)...
[info] Establishing links to the resources...
[info] MoAI Virtual Device is ready to use.
| INFO     | __main__:train:305 - Epoch 1/42 start
...
```

Key Features of MoAI Platform

- System architecture
  - AMD GPU support
  - Support for any off-the-shelf GPU server
  - Cost effective and scalable InfiniBand network topology enabled by software optimizations
  - Support for heterogeneous accelerators
- Cloud environment
  - Support for various virtual machines and hypervisors (including KVM, OpenStack, and VMware)
  - Support for Docker and Kubernetes
  - Technical support to build a cloud environment based on Kubernetes and Ceph on top of the MoAI platform
- GPU resource management
  - Built-in dynamic GPU resource allocation
  - User management and billing
  - Customizable resource allocation policy
  - Integrated system health and usage monitoring
  - Hardware fault detection
  - Dashboards for statistics and insights
- Interfaces
  - REST APIs, gRPC APIs, and the web console