SkyML: Infra-less Machine Learning on Any Cloud

riselab UC Berkeley

Zongheng Yang, Michael Luo, Zhanghao Wu, Frank Luan, Wei-Lin Chiang, Romil Bhardwaj, Gautam Mittal, Siyuan Zhuang, Scott Shenker, Ion Stoica

{zongheng, michael.luo, zhwu, lsf, weichiang, romil.bhardwaj, gbm, source, shenker, istoica}@berkeley.edu

Tale of Two RISE Students

On-prem User

Pros: convenient Slurm queueing system, cheap access to GPUs

Cons: Frequently out of space, congested, and orphaned processes

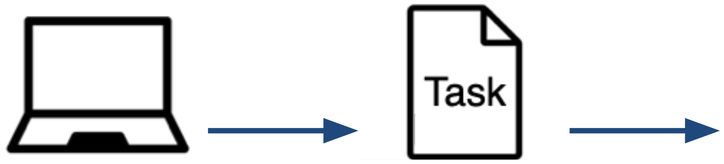
AWS User

Pros: Lots of accelerators and instance options

Cons: Hard to spin up new projects, limited availability, and under-utilized non-AWS credits

How can we run ML projects easily and transparently on any cloud?

Task Lifecycle



1 **User writes a task description with resource requirements on their laptop**

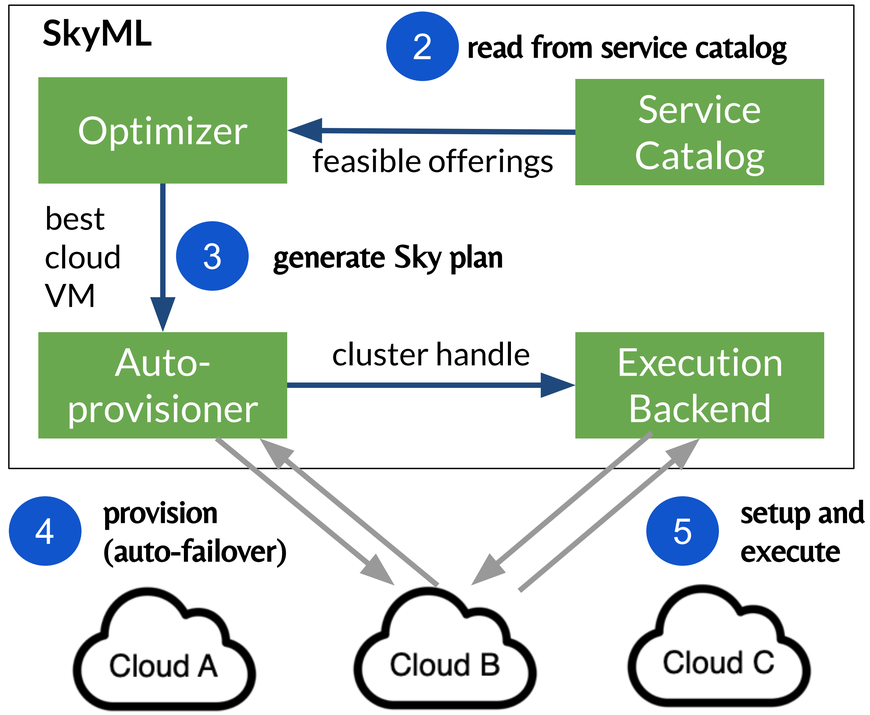
```
# my_task.yaml
workdir: ~/local/repo/path

num_nodes: 16
resources:
  accelerators:
    V100: 8

setup: “pip install -r requirements.txt”
run: “python train.py”
```

workdir automatically syncs

high-level resource requirements



Key Features

Any cloud	Declare task and resources – SkyML handles the rest
Job queue	Submit-and-go-to-bed behavior provides a pleasant workflow
Price-perf optimizer	Automated selection of region/ cloud satisfying accelerators reqs.
Data movement	Easily work with and move large datasets across clouds

User Workflow

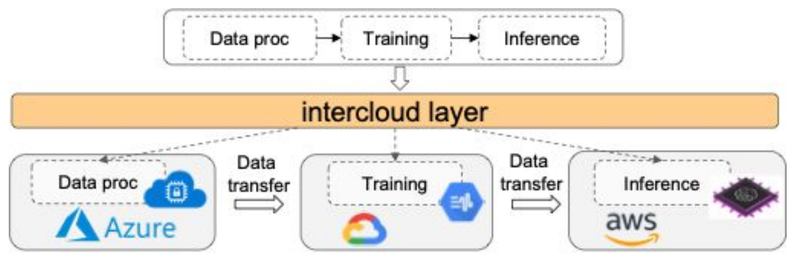
```
# Get a CPU/GPU/TPU node
sky gpunode --gpus V100:8
sky tpunode
# Launch task (provision + execute)
sky launch -c cluster my_task.yaml

# Run task (execution only)
sky exec cluster my_task.yaml

# View job status
sky queue cluster
```

Develop and run ML tasks anywhere with **no code changes**

Multi-cloud ML Pipelines



On an NLP pipeline, Sky execution plan is **61% cheaper and 47% faster** than single-cloud by using **TPU for training (GCP)** and **Inferentia for serving (AWS)**