RLlib：基于 Ray 和 tune 实现

Ray 主要功能：① 多进程
② 进程间的内存的数据传输

多进程

普通函数 def regular_function():
return 1

⇒ for _ in range(4):
regular_function()
串行执行

加了 ray 的装饰器的函数
@ray.remote
def remote_function():
return 1

⇒ for _ in range(4):
remote_function.remote()
可并行执行
是4个进程同时执行

进程间的通信

普通方法：需要知道各个端口，序列化传流，receive, send.

ray 方法：

```
@ray.remote
def remote_chain_function(value):
    return value + 1
y1_id = remote_function.remote()
assert ray.get(y1_id) == 1
```

得到的不是简单数字，而是个 objectID

RLlib 使用场景：RL 算法.

```
import ray
from ray import tune
ray.init()
tune.run(
        "PPO",
        stop = {"episode_reward_mean": 200},
        config = {
            "env": "CartPole-v0",
```

```
"num_gpus": 0,
"num_workers": 1,
"lr": tune.grid_search([0.01, 0.001, 0.0001]),
"eager": False,
},
)
```