

# Lab7 Report

112164513 陳彥凱

## 實驗流程：

使用 run\_DPP.sh 執行分散式訓練，觀察不同 GPU 數量的 scalability，而因為 hades server 每個 node 上面就只有兩張卡，所以我們就測試 GPU 數量為 1 和 2 的情況。參數 --nproc\_per\_node 指定每個 node 上的處理器數量（在這裡是 GPU 數量），這個 script 被運行了兩次，一次是設置為 1，另一次設置為 2。

--nproc\_per\_node 1

Output:

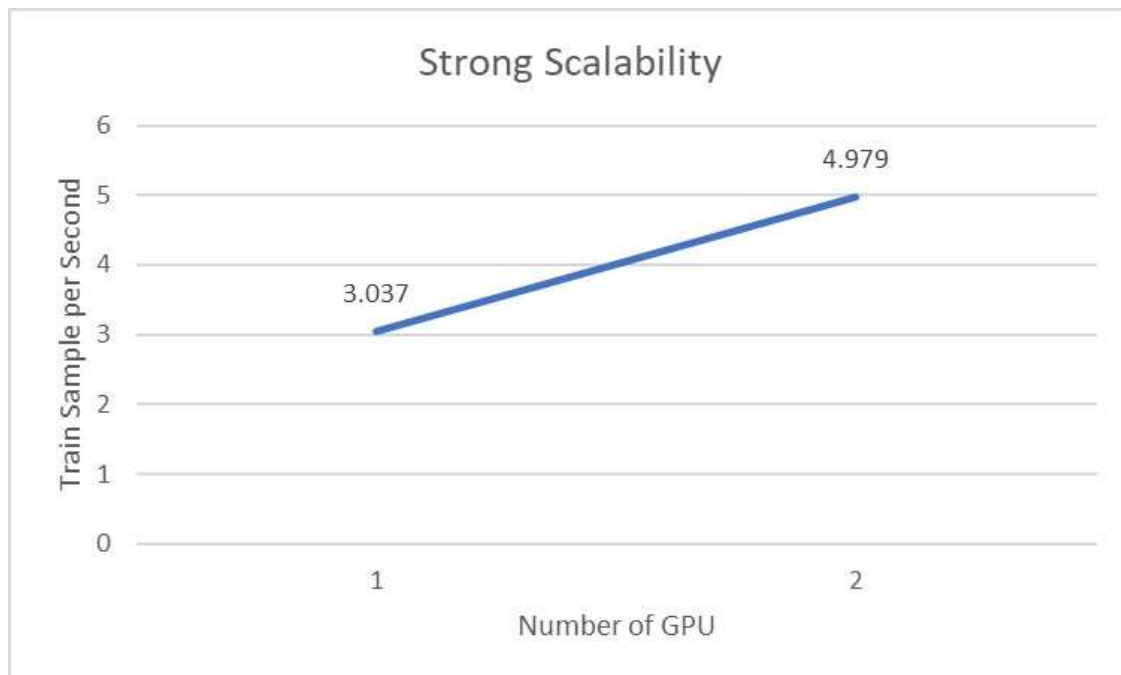
```
***** train metrics *****
epoch                =          0.09
train_loss           =          3.3877
train_runtime        = 0:01:05.85
train_samples        =          2318
train_samples_per_second =          3.037
train_steps_per_second  =          3.037
```

--nproc\_per\_node 2

Output:

```
***** train metrics *****
epoch                =          0.17
train_loss           =          3.3826
train_runtime        = 0:01:20.33
train_samples        =          2318
train_samples_per_second =          4.979
train_steps_per_second  =          2.49
```

## Strong Scalability:



### 實驗結果解釋：

單 GPU (nproc\_per\_node=1):

- train\_runtime: 訓練耗時大約 65.85 秒。
- train\_samples\_per\_second: 每秒處理大約 3.037 個樣本。
- train\_steps\_per\_second: 每秒執行大約 3.037 個訓練步驟。

雙 GPU (nproc\_per\_node=2):

- train\_runtime: 訓練耗時大約 80.33 秒。
- train\_samples\_per\_second: 每秒處理大約 4.979 個樣本。
- train\_steps\_per\_second: 每秒執行大約 2.49 個訓練步驟。

分析：

執行時間 (train\_runtime): 使用兩個 GPU 的執行時間比一個 GPU 長，這可能是因為額外的通信開銷。在分散式訓練中，多個 GPU 之間的同步可能導致額外的延遲。

樣本處理速度 (train\_samples\_per\_second): 使用兩個 GPU 的樣本處理速度更高，這是預期的，因為有更多的計算資源。

每秒訓練步驟次數 (train\_steps\_per\_second): 單 GPU 的每秒訓練步驟次數實際上比雙 GPU 的要高。這可能是因為並行處理時的通信和同步開銷減慢了步驟的執行速度。在分散式訓練中，增加節點數並不總是能夠 linear speed up，尤其是在通信成本較高或訓練任務本身不足夠大，以致於不能充分利用所有計算資源時。

結論：

這個實驗展示了增加平行處理節點 ( GPU ) 並不總是導致線性的性能提升。在分散式訓練中，需要平衡計算能力和通信開銷。針對特定的訓練任務和硬體設定，可能需要調整分散式策略以達到最佳性能。