Performance Evaluation of Tensorflow Parallel Strategies for Deep Learning Training on HPC Clusters

Kenichi Brumati, Marcelo Cardoso Oliveira Gulart, Rayan Raddatz de Matos

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INF01146 – Comp. Sys. Perf. Analysis – 2025/2

.inf INSTITUTO DE INFORMÁTICA UFRGS

Agenda

Recapitulação

Custom Training Loop

Reproducibilidade e Ambiente de Testes

Projeto Experimental

Resultados

Modelagem

Conclusão

Recapitulação

Mudanças desde a última etapa

- Focamos apenas no TensorFlow

Mudanças desde a última etapa

- Focamos apenas no TensorFlow
- Adicionamos outro modelo menor (EfficientNetB0)

Mudanças desde a última etapa

- Focamos apenas no TensorFlow
- Adicionamos outro modelo menor (EfficientNetB0)
- Focamos em entender o tempo de treino e descartamos métricas secundárias

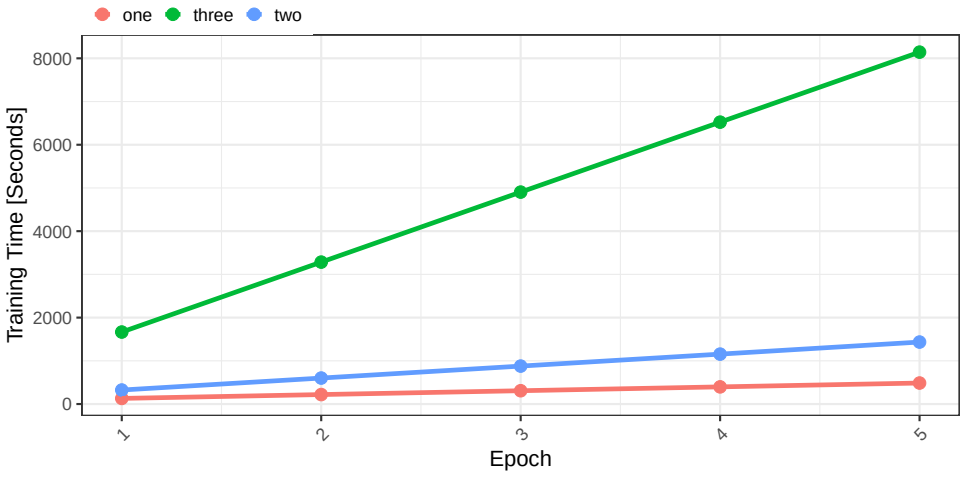
Mudanças desde a última etapa

- Focamos apenas no TensorFlow
- Adicionamos outro modelo menor (EfficientNetB0)
- Focamos em entender o tempo de treino e descartamos métricas secundárias
- Mudamos o dataset para o TinyImageNet-200

Mudanças desde a última etapa

- Focamos apenas no TensorFlow
- Adicionamos outro modelo menor (EfficientNetB0)
- Focamos em entender o tempo de treino e descartamos métricas secundárias
- Mudamos o dataset para o TinyImageNet-200
- Mudamos o hardware (downgrade)

Resultados anteriores



Fenômeno Observado

- Degradação de performance (aumento do tempo por época) ao adicionar novos workers.

Causa Técnica

- O treinamento distribuído síncrono exige que todos os nós atualizem os pesos simultaneamente.

- A operação de AllReduce (soma de gradientes) trafega dados pela rede a cada batch, assim, a latência de comunicação superou o ganho do paralelismo

Custom Training Loop

Estratégia

- Utilização de um Custom Training Loop.

Técnica Aplicada

- Implementação de acumulação de gradientes.

- Sincronização a cada 4 batches.

Impacto no Desempenho

- Aumenta o batch size efetivo antes de disparar a sincronização.

- Melhora a razão Computação/Comunicação, reduzindo o tráfego de rede e destravando o escalonamento.

Reproducibilidade e Ambiente de Testes

Limitação do GNU Guix

- A base oficial distribui estritamente Software Livre.

- Incompatibilidade com drivers proprietários essenciais (NVIDIA CUDA Toolkit e cuDNN).

Restrição de Infraestrutura

- Impossibilidade de adicionar canais não-oficiais (nonguix) no ambiente PCAD.

Solução Adotada

- Uso de **Python Virtual Environment** (venv).

- Isolamento de versões via pip garante reprodutibilidade sem violar restrições de licenciamento de drivers de GPU.

Hardware

Anteriormente estávamos utilizando máquinas da partição Tupi, por razões de **alta concorrência** mudamos para a partição Poti.

- Hardware anterior: Intel(R) Core(TM) i9-14900KF e **NVIDIA GeForce RTX 4090**
- Hardware UTILIZADO: Intel(R) Core(TM) i7-14700KF e **NVIDIA GeForce RTX 4070**

Tivemos alguns problemas como a configuração da rede da **poti4** e quedas repentinas da **poti2** que atrasaram os experimentos.

Projeto Experimental

Tipos de treinamento

Executamos dois tipos de treinamento:

- "sync" -> Sincronismo total com `model.fit()`
- "ctl" -> Sincronismo a cada 4 batches com Custom Training Loop

Modelo

Escolhemos além do **ResNet50** com **~25.6M** parâmetros um modelo menor: **EfficientNetB0** com **~5.3M** parâmetros.

Outros fatores

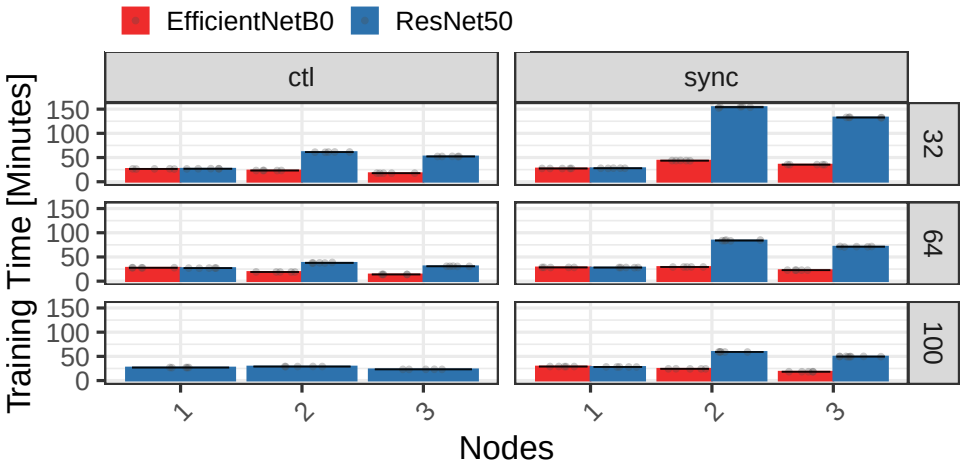
- **Batch size:** 32, 64, 100
- **Quantidade de nodos:** 1, 2 e 3

Replicações

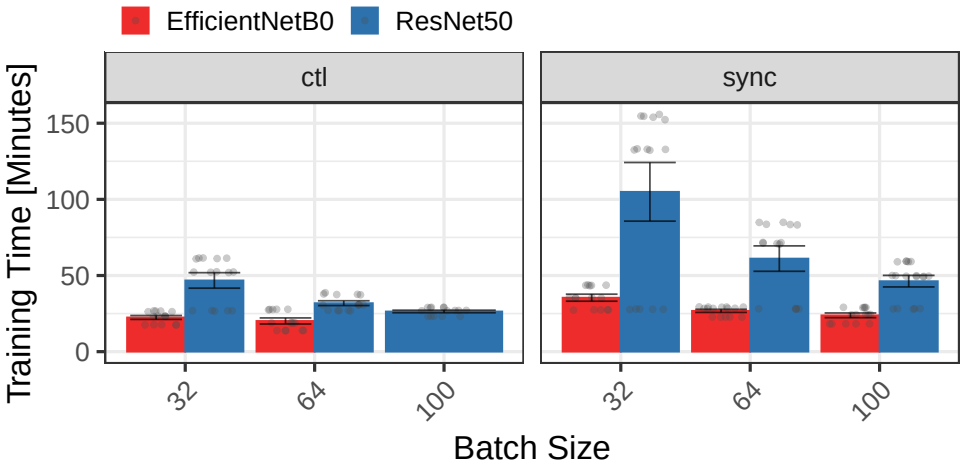
- 2x2x3x3 = **36 configurações**
- Cada uma foi **replicada 5 vezes**.
- Total de **180** execuções!

Resultados

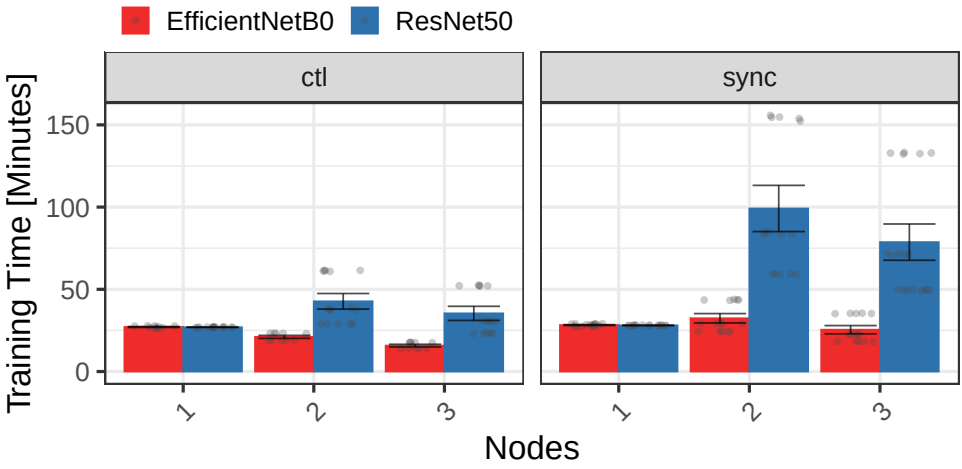
Aglomerado final dos dados



Resultados por batch

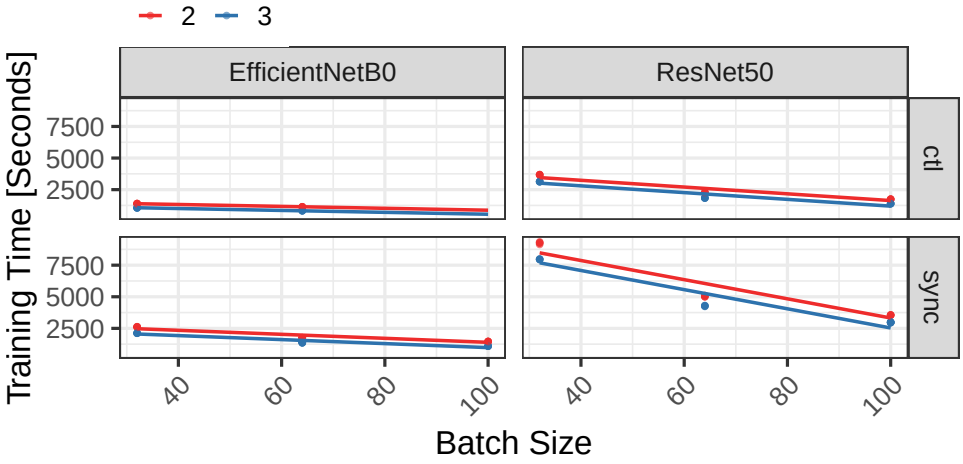


Resultados por número de nodos

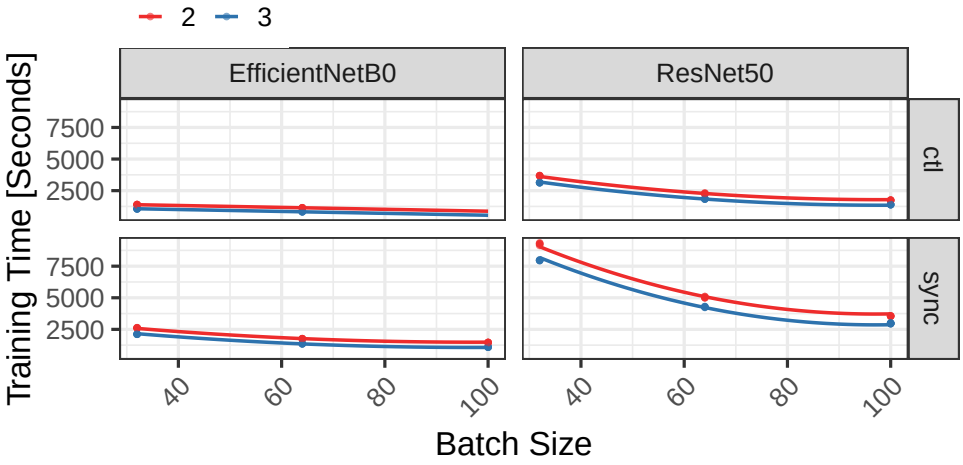


Modelagem

Nosso modelo linear



Nosso modelo quadrático



Conclusão

Conclusão

- O tamanho do modelo assim como a rede é um fator crítico para o treino distribuído síncrono.

- Mesmo com uma rede de baixo throughput é possível obter ganhos através de uma boa configuração do ambiente e do modelo.

Trabalhos futuros

Prever o overhead de comunicação baseado no tamanho do modelo e das condições da rede.

Perguntas?

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf
INSTITUTO DE INFORMÁTICA
UFRGS