УДК 004.8

ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ И ПОТРЕБЛЕНИЯ РАСПРЕДЕЛЕННЫХ РЕСУРСОВ СБОРОЧНЫХ ЛИНИЙ ДЛЯ ОБУЧЕНИЯ БОЛЬШИХ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Баркалов И. О.
Научный руководитель – канд. техн. наук, доцент Лабковская Р. Я.
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
bio.ejiki@gmail.com

Введение

Современные проекты в области машинного обучения подошли к беспрецедентному росту сложности моделей. Большие языковые модели LLM, такие как GPT-5, Grok 4 и их аналоги требуют колоссальных вычислительных ресурсов для обучения и последующей эксплуатации.

Традиционные подходы к автоматизации CI/CD эффективны для моделей умеренного размера, становятся неприменимыми в условиях работы с моделями, содержащими миллиарды параметров.

Масштабирование сборочных линий MLOps включает в себя такие проблемы как экспоненциальный рост времени обучения, неэффективное использование дорогостоящих вычислительных ресурсов (GPU и DDR) и сложность в оркестрации распределенных вычислений. Все это ведет к непомерно высоким затратам, в столь тяжелый период дефицита ресурсов и высокой конкуренции.

Основная часть

В данной работе предлагается комплексное решение описанных проблем через автоматизацию процессов путем построения сборочных линий и внедрения современных методов распределенных вычислений.

Разработана многоуровневая архитектура для сборочной линии, позволяющая воспроизводить различные стратегии распределения вычислений.

Основные компоненты можно разбить на оркестратора исполнения сборочных линий, менеджер ресурсов, система мониторинга и распределенное хранилище данных.

В качестве оркестратора были использован Jenkins совместно с плагином для Kubernetes. В роли менеджера ресурсов Kubernetes с поддержкой GPU-приложений. Для наглядности и мониторинга за процессами Prometheus в связке с Grafana. MinIO как S3-подобное распределенное хранилище, обеспечивающее надежность и репликацию данных.

Для оптимизации процесса обучения больших моделей были реализованы и протестированы такие подходы как Distributed Data Parallel (DDP) и Fully Sharded Data Parallel (FSDP).

Выводы

Реализованные в рамках сборочной линии подходы DDP и FSDP подтвердили эффективность, позволяя оптимально использовать доступные ресурсы в работе с большими языковыми моделями.

Данная работа показала, что архитектура на базе Jenkins и Kubernetes способна обеспечить как воспроизводимость процессов распределенного обучения, так и гибкое переключение между стратегиями параллельного обучения в зависимости от доступной инфраструктуры. Разработанное решение позволит минимизировать простой вычислительных ресурсов и затраты.

Список использованных источников

1. Лукша М. Kubernetes in action – Pact, 2019. — 672 с. — ISBN 978-5-97060-657-5.
2. Хьюен, Ч. Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications / Ч. Хьюен. – O'Reilly, 2022. – 386 p. – ISBN 978-1-098-11058-7.