УДК 004.85

СОВМЕСТНАЯ ОПТИМИЗАЦИЯ PIPELINE PARALLELISM И НИЗКОБИТНОГО КВАНТИЗОВАНИЯ ВЕСОВ ДЛЯ УСКОРЕНИЯ ОБУЧЕНИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Балакин А. А.[1] / Поляков И. В.[1]
Научный руководитель — доктор технических наук, доцент Духанов А. В.[1]
[1]Университет ИТМО
BalakinAA2026@itmo.ru

Введение

Обучение больших языковых моделей (LLM) ограничивается памятью GPU и падением эффективности при масштабировании на кластер из-за коммуникаций и простоев. На практике pipeline parallelism и низкобитные представления весов применяются раздельно, из-за чего часть выигранной памяти не конвертируется в рост пропускной способности. Цель работы — предложить метод совместной настройки пайплайна и низкобитного режима весов, уменьшающий время обучения и требования к памяти при сохранении качества.

Основная часть

Предлагается подход, в котором pipeline parallelism (PP) и низкобитное квантизование весов рассматриваются как единая задача оптимизации [1 — 3].

Ключевая идея заключается в использовании сэкономленной при сжатии весов памяти для поддержания загруженности пайплайна, что и дает прирост к скорости обучения. Низкобитное хранение весов (например, 8-бит или 4-бит, либо режимы пониженной точности в современных GPU-стэках) уменьшает постоянную часть потребления памяти на стадиях пайплайна. Освободившийся ресурс целенаправленно используется не “в запас”, а для повышения заполнения конвейера: увеличения числа микробатчей в полёте и/или увеличения эффективного батча через аккумулирование градиентов [4 — 6]. Это снижает долю простоев (“пузырей”) и повышает утилизацию ускорителей [2, 3].

Конфигурация пайплайна подбирается с учётом того, что простои определяются дисбалансом времени стадий и накладными расходами на межстадийные передачи [2, 3]. Границы стадий выбираются так, чтобы приблизительно выровнять длительности стадий и уменьшить объём межстадийных коммуникаций [1 — 3]. Используется расписание исполнения (1F1B), уменьшающее простои при заданном числе стадий [1, 8]. Число же микробатчей подбирается так, чтобы обеспечить устойчивое заполнение пайплайна при соблюдении лимитов памяти под веса/активации/буферы [2, 8].

Квантизация вводится не как единая настройка “для всех весов”, а как практическая политика точности по компонентам модели: более агрессивная низкобитность применяется там, где выигрыш по памяти и по устойчивости максимален, а чувствительные компоненты (например, нормализации и отдельные проекции) сохраняются в более высокой точности [6]. Для LLM-практики ориентиром служат результаты по 4-битным режимам при сохранении качества для задач дообучения, а также результаты обучения в FP8 на современных GPU, демонстрирующие потенциал ускорения при корректной настройке точности [4, 5]. Дополнительно, снижение памяти может быть усилено низкобитными состояниями оптимизатора без существенной потери качества [7]. После включения низкобитных режимов выполняется повторная балансировка стадий, так как вычислительный профиль стадий может измениться [1, 6].

Метод реализуется как итеративный цикл: профилирование базовой PP-конфигурации (время стадий, коммуникации, простои, пик памяти), включение низкобитного режима весов и оценка высвобождения памяти/накладных расходов, перераспределение высвобождённой памяти на увеличение микробатчей и корректировку границ стадий/расписания, контроль качества на валидации и проверка устойчивости обучения [2,6,8]. Практическая реализуемость подтверждается наличием инструментов для пайплайнинга и расписаний в современных фреймворках распределённого обучения [8], а также наличием библиотек для низкобитных режимов [6].

Выводы

Предложен тезисный метод совместной оптимизации pipeline parallelism и низкобитного квантизования весов для ускорения обучения LLM. Практическая ценность подхода состоит в том, что он рассматривает pipeline parallelism и квантизацию как единый контур оптимизации: высвобождение памяти используется для уменьшения “пузырей” и повышения пропускной способности, а не только для формального снижения потребления памяти.

Литература

1. Narayanan D., Shoeybi M., Casper J. et al. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM // SC 2021. 2021. — Режим доступа: https://arxiv.org/abs/2104.04473 (дата обращения: 20.02.2026).
2. Huang Y., Cheng Y., Bapna A. et al. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism // arXiv. 2018. — Режим доступа: https://arxiv.org/abs/1811.06965 (дата обращения: 20.02.2026).
3. Harlap A., Narayanan D., Phanishayee A. et al. PipeDream: Pipeline Parallelism for DNN Training // arXiv. 2018. — Режим доступа: https://arxiv.org/abs/1806.03377 (дата обращения: 02.02.2026).
4. Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L. QLoRA: Efficient Finetuning of Quantized LLMs // NeurIPS 2023. 2023. — Режим доступа: https://arxiv.org/abs/2305.14314 (дата обращения: 22.02.2026).
5. Peng H. et al. FP8-LM: Training FP8 Large Language Models // arXiv. 2023. — Режим доступа: https://arxiv.org/abs/2310.18313 (дата обращения: 15.02.2026).
6. NVIDIA. Using FP8 and FP4 with Transformer Engine // Transformer Engine Documentation. — Режим доступа: https://docs.nvidia.com/deeplearning/transformer-engine/user-guide/examples/fp8_primer.html (дата обращения: 15.02.2026).
7. Dettmers T. et al. 8-bit Optimizers via Block-wise Quantization // ICLR 2022. 2021. — Режим доступа: https://arxiv.org/abs/2110.02861 (дата обращения: 15.02.2026).
8. PyTorch. Pipeline Parallelism (torch.distributed.pipelining) // PyTorch Documentation. — Режим доступа: https://docs.pytorch.org/docs/2.9/distributed.pipelining.html (дата обращения: 02.02.2026).

Балакин А. А. (автор) Подпись _____.

Поляков И. В. (соавтор) Подпись _____.

Духанов А. В. (научный руководитель) Подпись _____.