УДК 004.85

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ APACHE SPARK С ПРИМЕНЕНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ ДАННЫХ БЕНЧМАРКА О ПРОИЗВОДИТЕЛЬНОСТИ

Апыхин А. А. (ИТМО)
Научный руководитель – доцент Платонов А. В.
(ИТМО)

Введение. Производительность приложений Apache Spark существенно зависит от настроек конфигурации и характеристик вычислительной среды. При этом пространство параметров велико, а оптимальные значения меняются в зависимости от типа нагрузки и объёма данных. Ручной подбор параметров требует большого числа прогонов и не масштабируется, поэтому актуальны подходы, использующие данные бенчмарков и методы машинного обучения для построения моделей времени выполнения и последующего поиска конфигураций [1].

Сбор данных для обучения является ресурсозатратным: даже при переборе 16 параметров конфигурации с ограниченным набором значений общее число комбинаций достигает нескольких тысяч, а выполнение одной задачи занимает от нескольких минут до часа. В рамках текущего этапа исследования основной датасет сформирован для нагрузки WordCount из набора HiBench; для каждой конфигурации выполнялось 6 запусков с последующим вычислением медианы времени выполнения.

Основная часть. Инфраструктура экспериментов включала Apache Spark, HiBench и HDFS, развёрнутые в Docker-контейнерах [2]. Для подготовки признаков был реализован препроцессор, выполняющий обработку категориальных параметров, нормализацию числовых значений и парсинг параметров памяти, что позволило унифицировать представление конфигураций для обучения моделей.

Для построения предиктора времени выполнения были реализованы baseline-модели (DummyRegressor, MLP), а качество оценивалось метриками MAE, RMSE и MAPE. Наиболее устойчивые и воспроизводимые результаты на собранном датасете показал Random Forest с автоматическим подбором гиперпараметров через RandomizedSearchCV [3].

Для дальнейших экспериментов Random Forest использовался как surrogate-модель, позволяющая быстро оценивать кандидатные конфигурации без повторных реальных запусков. Поиск в дискретном пространстве параметров выполнялся методом Simulated Annealing, допускающим вероятностное принятие ухудшающих шагов на ранних итерациях и снижающим риск застревания в локальных минимумах [4].

В дополнение к ансамблевым методам был реализован DNN-предиктор на PyTorch по мотивам работы [1] (архитектура 128->64->1, логарифмическая трансформация целевой переменной, early stopping и learning rate scheduling). Однако при текущем объёме и уровне шумности данных нейросетевой подход оказался менее стабильным и в среднем уступал Random Forest по согласованности предсказаний с фактическими значениями времени выполнения.

В качестве решающих алгоритмов были исследованы табличный Q-learning, Deep Q-Network (DQN), policy-based методы PPO и A2C, а также Bayesian Optimization (Optuna, TPE) [5]. На текущем датасете эти подходы проявили высокую чувствительность к гиперпараметрам обучения и качеству данных. При попытке подтвердить рекомендации повторными запусками Spark-задач на локальном кластере выявлено расхождение медианных значений порядка 20 % (в меньшую сторону), что указывает на ограничения текущего этапа и необходимость дальнейшей доработки экспериментов.

Ключевыми направлениями дальнейшей работы являются: увеличение объёма и репрезентативности датасета на выделенных вычислительных ресурсах; расширение

набора нагрузок HiBench (Sort, KMeans и др.); сбор дополнительных метрик из Spark event-log и систем мониторинга; очистка данных (фильтрация выбросов) и сравнение моделей (ансамбли, бустинг) с учётом неопределённости предсказаний.

Выводы. Проведён цикл экспериментов по обучению моделей, предсказывающих время выполнения Spark-задачи на основе параметров конфигурации и характеристик окружения. Показано, что на текущем датасете Random Forest обеспечивает наиболее устойчивое качество и подходит в роли базовой модели для последующего поиска конфигураций. Основным ограничением остаются объём и качество данных, поэтому дальнейшие исследования будут направлены на масштабирование сбора датасета и расширение набора нагрузок.

Список использованных источников :

1. Huang X., Zhang H., Zhai X. A Novel Reinforcement Learning Approach for Spark Configuration Parameter Optimization // Sensors. 2022. Vol. 22, No. 15.
2. Huang S. et al. The HiBench benchmark suite: Characterization of the MapReduce-based data analysis // ICDEW. 2010. P. 41–51.
3. Breiman L. Random Forests // Machine Learning. 2001. Vol. 45, No. 1. P. 5–32.
4. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing // Science. 1983. Vol. 220. P. 671–680.
5. Akiba T. et al. Optuna: A Next-generation Hyperparameter Optimization Framework // KDD. 2019. P. 2623–2631.