ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ И АТОМНОГО ЯДРА
2024. Т. 55, вып. 3. С. 580–584

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ЗАПУСКА ЗАДАЧИ ПУТЕМ СТАТИСТИЧЕСКОГО АНАЛИЗА ИСТОРИИ ОЧЕРЕДИ ЗАДАЧ

*И. С. Ященко *, А. Н. Сальников ***

Московский государственный университет им. М. В. Ломоносова, Москва

Предсказание времени запуска задачи без помещения ее в очередь задач в современных вычислительных центрах очень сложно. Предлагается усовершенствовать созданный ранее инструмент прогнозирования за счет использования более современных программных интерфейсов для системы массового обслуживания SchedMD (Slurm), а также за счет использования различных методов математического прогнозирования времени запуска задачи. Для сравнения и оценки моделей машинного обучения выбран файл SWF с данными истории выполнения задач на вычислительном кластере Люксембургского университета с 59 715 задачами. Исследованы следующие методы: линейная регрессия с L2-регуляризацией, метод опорных векторов, случайный лес, LightGBM, CatBoost, LightGBM с оптимизацией параметров, CatBoost с оптимизацией параметров. Для проверки точности прогноза построен тестовый стенд на базе симулятора Slurm (центр вычислительных исследований SUNY при Университете Буффало, США), который работает на основе выполнения задач и организации очереди сохраненных логов.

Predicting the start time of a task without placing it in a task queue is very difficult in modern data centers. The paper proposes to improve the previously created forecasting tool through the use of more modern software interfaces to the SchedMD queuing system (Slurm), as well as the use of various mathematical forecasting methods to predict the moment of task launch. To compare and evaluate machine learning models, we selected an SWF file with task execution history data on the University of Luxembourg computing cluster with 59,715 tasks. The following methods were investigated: linear regression with L2 regularization, support vector machine, random forest, LightGBM, CatBoost, LightGBM with parameter optimization, CatBoost with parameter optimization. To check the correctness of the prediction, a test bench was built based on the Slurm simulator (SUNY Center for Computational Research at the University at Buffalo, USA), which works based on executing tasks and organizing a queue of saved logs.

PACS: 89.20.Ff; 07.05.Tp

* E-mail: yashch.igor@gmail.com
** E-mail: salnikov@cs.msu.ru

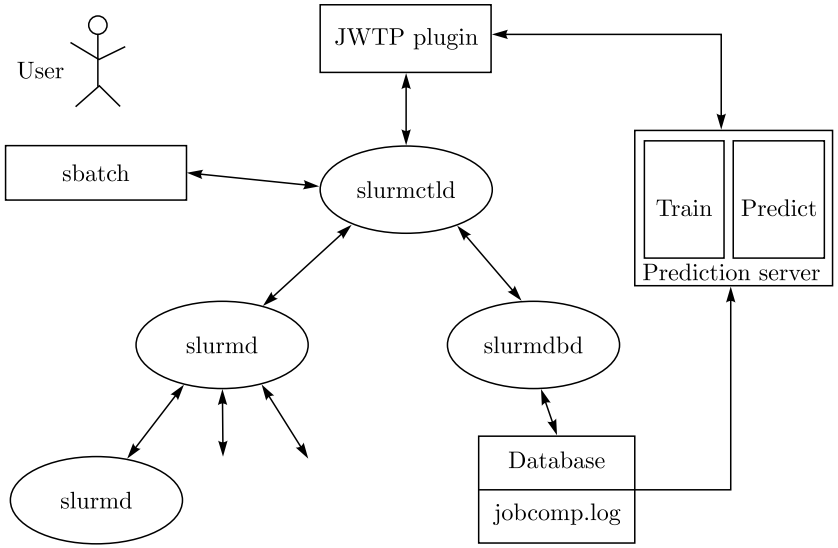
ВВЕДЕНИЕ

Современные суперкомпьютеры представляют собой сложные многокомпонентные и многопользовательские системы, которые обычно имеют кластерную структуру и используют различные системы управления очередями задач. Примером таких систем ведения очередей могут быть Slurm Workload Manager [1], Platform LSF, LoadLeveler и др. Как правило, очереди задач на вычислительных комплексах достаточно длинны, и иногда до старта задачи на выполнение приходится ожидать днями или даже неделями. Пользователям вычислительных комплексов необходимо иметь прогноз времени ожидания своих задач в очереди и на его основе осуществлять выбор наилучшего кластера под конкретное задание. Информация о времени нахождения задачи в очереди до начала ее выполнения позволяет пользователю вносить модификации в запрашиваемые ресурсы задачи с целью приближения времени запуска задачи на кластере. Некоторые системы ведения очередей способны выдать такой прогноз, сопоставив множество ресурсов системы, множество ресурсов запрашиваемых задач с запланированными интервалами запуска и завершения задач. Однако такая оценка часто не является адекватной, она может быть завышенной или заниженной из-за политики администраторов. В долгосрочный период времени с точки зрения управления потоками задач пользователей вычислительный кластер можно считать стационарной системой, в которой движение в очередях можно моделировать на основе статистических данных. Собрав статистику за длительный период времени о прохождении задач пользователей, мы сможем создать модель поведения системы для потока задач пользователей, на основе которой будем строить прогноз длительности нахождения задачи в очереди.

Спроектирована и разработана система, предоставляющая конкретному пользователю прогноз времени ожидания задачи в очереди с возможностью интеграции новых моделей и алгоритмов прогнозирования. Система расширяет функционал системы ведения очередей Slurm. Разработаны методы прогнозирования, основанные на подходах машинного обучения на реальной истории запусков задач пользователей. По результатам сравнения методов и оценки эффективности выбран наилучший метод для интеграции его в системе.

РЕАЛИЗАЦИЯ СИСТЕМЫ

Архитектура системы прогнозирования времени ожидания в очереди включает в себя два основных модуля: клиентский плагин JWTP и сервер Prediction Server (рисунок). Клиентский плагин JWTP (Job Waiting Time Prediction) создан на языке программирования C с использованием кода Slurm (job submit plugin API) и оформлен как файл .so. Этот плагин принимает данные о задаче, направленной на кластер, проверяет наличие ключа predict-time в файле «паспорта задачи» и далее перенаправляет



Архитектура системы прогнозирования времени ожидания в очереди

информацию на сервер через текстовый запрос HTTP. Получив ответ с прогнозом от сервера, плагин отправляет его на устройство пользователя через стандартный поток ошибок stderr. Сервер Prediction Server — автономная программа-демон с доступом к базе данных и историческим данным о выполнении задач на вычислительном кластере. Сервер читает Slurm Database и jobcomp.log, осуществляет переобучение прогнозирующей модели на полученных данных через равные промежутки времени. Сервер по запросу пользователя с информацией о ресурсах потенциальной задачи применяет обученную и выбранную заранее модель машинного обучения и возвращает ответ созданному плагину. Сервер реализован на языке программирования Python3.

Для сравнения и оценки моделей машинного обучения выбран файл формата Standard Workload Format (SWF) [2] с данными истории выполнения задач на вычислительном кластере Люксембургского университета [3]. Данные истории представляют табличный набор данных в 59 715 строках, каждая строка отнесена к конкретной задаче пользователя. Применен следующий подход к выделению набора данных из файла SWF: осуществляется выбор задач только для одной очереди (Queue Number), добавляется информация о пользователе. Для каждой задачи выделяются статистические данные (на момент ее постановки в очередь). Эти данные берутся из множеств задач: стоящие в очереди на текущий момент, закончившие выполнение за некоторый интервал к текущему моменту, выполняющиеся в данный момент. Добавляются поля с запрашиваемыми ресурсами для задачи. Каждый набор некоторым обра-

зом преобразуется в вектор признаков, которые потом конкатенируются в единый вектор признаков.

Рассмотрены следующие модели машинного обучения: линейная регрессия с L2-регуляризацией, метод опорных векторов, случайный лес, LightGBM, CatBoost, LightGBM с оптимизацией параметров, CatBoost с оптимизацией параметров. Обучение моделей проводилось на исторических данных, собранных на вычислительном кластере Люксембургского университета. По совокупности метрик, использованных для оценки качества моделей, была выбрана CatBoost с предварительным отбором оптимальных признаков.

ТЕСТИРОВАНИЕ СИСТЕМЫ

В связи со сложностью тестирования плагина на реальных данных было принято решение использовать симулятор Slurm, работающий на основе сохраненных логов выполнения задач, разработанный в центре вычислительных исследований SUNY Университета Буффало (США) [4]. Данный симулятор обладает следующими преимуществами: код симулятора размещен в открытом доступе на GitHub [5], разработчики осуществляют непрерывную поддержку и обновление версий симулятора.

Симулятор работает на рабочей станции или одном узле высокопроизводительных вычислений, что позволяет ускорить моделирование рабочих нагрузок на ресурсы высокопроизводительных вычислений. Исходная версия симулятора была специфически модифицирована его разработчиками путем удаления вызовов функций, использующих job submit plugin API. После общения с разработчиками симулятора стало понятно, что для интеграции плагина JWTP нужно явно добавить вызов необходимых функций, в частности функцию validate_job_create_req, которая вызывалась в момент получения новой задачи контрольным демоном и до отправки задачи в очередь.

Для автоматической реализации модификации в рабочей среде написан patch-файл simulator.diff, который применяется к исходному коду симулятора с дальнейшей пересборкой файлов Slurm, что обеспечивает удобство применения разработанной системы для широкого круга пользователей.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Таким образом, в результате выполненной работы для предсказания времени запуска задачи в очереди задач разработан сервер Prediction Server с доступом к базе данных и данным истории выполнения задач на вычислительном кластере. Для сравнения моделей машинного обучения выделен датасет признаков из файла Standart Workload Format (SWF). В результате сравнения и оценки моделей выбрана оптимальная модель — модель градиентного бустинга с использованием библиотеки

CatBoost — и интегрирована в систему прогнозирования. Для системы ведения очередей Slurm разработан клиентский плагин JWTP (Job Waiting Time Prediction) с использованием инструментария исходного кода Slurm. Экспериментальный стенд в окружении Docker собран для тестирования системы прогнозирования на симуляторе Slurm. Исходный код работы представлен в репозитории [6].

СПИСОК ЛИТЕРАТУРЫ

1. Slurm Workload Manager. https://slurm.schedmd.com (accessed 22.08.2023).
2. The Standard Workload Format. https://www.cs.huji.ac.il/labs/parallel/workload/swf.html (accessed 22.08.2023).
3. The University of Luxembourg Gaia Cluster log. https://www.cs.huji.ac.il/labs/parallel/workload/l_unilu_gaia/index.html (accessed 22.08.2023).
4. *Simakov N. A., Innus M. D., Jones M. D., DeLeon R. L., White J. P., Gallo S. M., Patra A. K., Furlani T. R.* A Slurm Simulator: Implementation and Parametric Analysis // High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation. 2018. P. 197–217.
5. Slurm Simulator. https://github.com/ubccr-slurm-simulator/slurm_simulator#readme (accessed 22.08.2023).
6. Project source code. https://github.com/IgorYashch/slurm_waittime_prediction_plugin (accessed 22.08.2023).