

УДК 519.688
DOI: 10.15827/0236-235X.137.124-131

Дата подачи статьи: 08.11.21, после доработки: 10.12.21
2022. Т. 35. № 1. С. 124–131

Использование симулятора планировщика заданий для оценки эффективности предсказания времени работы задания

С.С. Шумилин¹, старший научный сотрудник, shumilin@jssc.ru
М.Ю. Воробьев¹, младший научный сотрудник, nordmike@jssc.ru

¹ Межведомственный суперкомпьютерный центр РАН, г. Москва, 119991, Россия

В настоящей работе исследуется эффективность планирования очереди с помощью предобученных моделей. Суперкомпьютерный кластер применяет планировщик для распределения поступающего потока заданий между доступными вычислительными ресурсами. Для размещения задания в очереди планировщик использует данные, указанные пользователем, в том числе заказанное время работы программы. Однако пользователи часто неверно оценивают время работы и указывают верхнюю оценку. Если задание завершается раньше, чем указано, то планировщику необходимо перепланировать очередь. Большое количество подобных событий может снижать эффективность распределения ресурсов.

В последнее время появилось множество работ, посвященных применению машинного обучения для предсказания времени работы задания. Это позволяет в процессе планирования использовать время работы, вычисленное с помощью предобученной модели. Однако все полученные модели содержат ошибку. Проблема заключается в необходимости оценки эффективности планирования при заданном значении ошибки модели.

В данном исследовании для оценки эффективности планирования очереди с помощью предобученных моделей сравниваются два сценария: планировщик использует указанное пользователем время (сценарий 1) и реальное время работы задания (сценарий 2). С помощью симулятора планировщика SLURM была произведена симуляция на статистических данных суперкомпьютера МВС-10П ОП2, установленного в Межведомственном суперкомпьютерном центре РАН. Результаты показали, что среднее время ожидания в сценарии 2 уменьшилось на 25 %, а приведенное время ожидания – на 50 %. Утилизация ресурсов изменилась незначительно.

Данные эксперимента доказывают целесообразность применения алгоритмов машинного обучения для предсказания времени работы заданий, поступающих на суперкомпьютерный кластер. Таким образом, в статье приводится оценка предельной оптимизации, так как эксперимент предполагает стопроцентную точность предсказания, что на сегодняшний день не демонстрирует ни одна из представленных работ по предсказанию времени работы заданий.

Ключевые слова: предсказательная аналитика, SLURM, планирование заданий, симулятор, МВС-10П.

Эффективность распределения ресурсов суперкомпьютерного кластера зависит от точности указанной пользователем оценки времени выполнения задания. Предсказать время работы задания довольно трудно, так как оно зависит от множества факторов. Чтобы подавать на вход планировщику более точную оценку, требуется на основе имеющейся информа-

ции предсказать время выполнения задания (рис. 1). Для различных систем планирования заданий возможно накопление самой разной информации о запусках, поэтому можно производить предсказание на основе данных о характеристиках самой задачи (аргументы командной строки, имя программы), профиля ее исполнения на кластере (загруженность про-

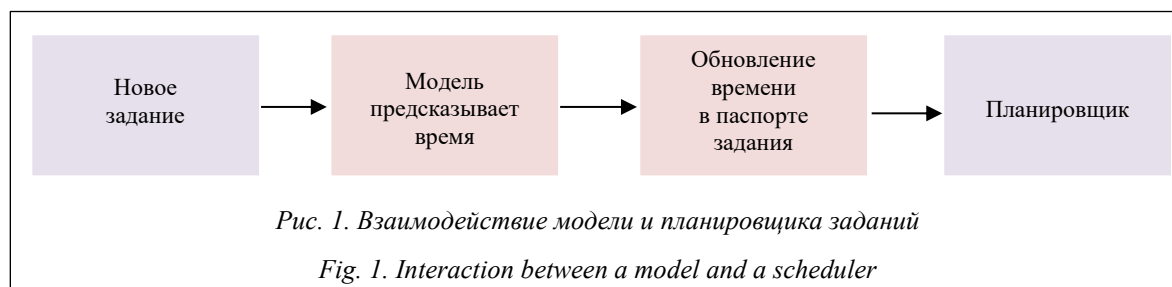


Рис. 1. Взаимодействие модели и планировщика заданий

Fig. 1. Interaction between a model and a scheduler

цессора, количество используемой оперативной памяти), данных пользователя (группа пользователя, организация) и других внешних данных (время постановки задачи в очередь в часовом поясе пользователя, день недели и т.д.). Для этого могут применяться алгоритмы машинного обучения.

Современные публикации [1–4] свидетельствуют о том, что результаты предсказания времени выполнения заданий существенно зависят от статистической БД, на которой производилось машинное обучение. При этом остается открытым вопрос о влиянии точности предсказания времени выполнения на эффективность планирования заданий с точки зрения таких показателей, как утилизация ресурсов, время ожидания заданий в очереди и других. В настоящей работе сравнивается эффективность планирования в двух сценариях: использование планировщиком времени, указанного пользователем (сценарий 1), и реального времени работы задания (сценарий 2).

Сопутствующие работы

Ряд исследований, в частности [5, 6], посвящены обучению модели на основе имеющихся данных и предсказанию времени на выполнение новых заданий.

Первые попытки применить модели машинного обучения были предприняты еще два десятилетия назад [7, 8]. За это время алгоритмы машинного обучения получили существенное развитие и стали общедоступными, как и большие вычислительные мощности.

Основой данного подхода является пользовательское моделирование [9, 10], то есть использование данных о поведенческих характеристиках пользователей и предсказание на их основе дальнейших действий.

В работе [11] исследуется возможность совмещения планировщика заданий и алгоритма, предсказывающего время выполнения задания, вводится метрика для оценки точности предсказания. Авторы [12] предлагают метрику для сравнения результатов работы симулятора планировщика и реальных данных.

В [13] используются алгоритмы машинного обучения для предсказания циклов CPU, требуемой RAM и бинарного показателя того, досчитается ли задание до конца. Для обучения моделей авторы применяют дополнительные признаки, отражающие опыт пользователя. В работе описаны несколько алгоритмов ма-

шинного обучения для задач регрессии и классификации и делается вывод, что добавление новых признаков способствует увеличению оценочных метрик моделей.

Аналогичный подход применяется в работах [1, 2, 6]: выбираются несколько алгоритмов машинного обучения для задач классификации и регрессии и сравниваются по выбранной метрике.

В [14] предложено использовать оценки модели в качестве рекомендации для пользователей.

Исследование [15] посвящено использованию сверточных нейронных сетей, трансформирующих скрипт с описанием задания в изображение и преобразующих его в оценку времени выполнения и входного/выходного потока данных.

Некоторые модели применяют исходный код для анализа ресурсов программы [16].

Авторы большинства работ делают выбор в пользу алгоритма случайного леса и используют данные, накопившиеся за некоторый период пользования кластером. Эти данные применяются для обучения моделей.

Важной задачей является подбор признаков задания для использования в модели. Паспорт задания содержит информацию о поступившем на кластер задании. Разные системы формируют разные паспорта, поэтому данные для обучения требуется формировать из доступных в системе параметров. Например, на установленном в Межведомственном суперкомпьютерном центре РАН (МСЦ РАН) суперкомпьютере МВС-10П ОП2 (раздел Cascade Lake, <https://www.jscc.ru/resources/vvod-v-ekspluatatsiyu-razdela-mvs-10p-op-cascade-lake/>) доступны следующие исходные признаки задания: идентификатор пользователя (userid), заказанные ядра (nproc), заказанное время (ntime), квант времени (quantum), идентификатор группы (gid), область исследований (domain), идентификатор организации (orgid).

Это те признаки, которые формируются при подаче пользователем задания. Однако их набор может быть расширен.

В работе [2] рассмотрены различные алгоритмы машинного обучения в применении к предсказанию времени выполнения пользовательских заданий на суперкомпьютерах МСЦ РАН. Сформированы новые признаки на основе существующей статистики пользователя, такие как средняя длительность задания, средний заказанный ресурс, среднее время работы

задания и др. Также на основе времени поступления задания сформированы категориальные признаки, указывающие на время поступления задания, такие как день, месяц, день недели. Эти признаки добавлены к исходным. Выполнен отбор признаков на основе различных методов, позволяющих отбирать наиболее информативные.

В качестве критерия сравнения регрессионных моделей был использован коэффициент детерминации R^2 . Самой лучшей моделью оказался случайный лес с коэффициентом детерминации 0,68. Для этой модели наиболее информативными признаками стали заказанное время, среднее время выполнения задания, средняя ошибка в оценке времени выполнения (в виде доли от заказанного времени).

Однако для исследования влияния модели на планирование заданий необходимо оценить несколько вариантов работы планировщика. Для этого нужно смоделировать поток заданий с помощью симулятора планировщика заданий. В настоящей статье рассматриваются два варианта работы планировщика, при которых ему на вход подаются характеристики заданий: t_{wc} – заказанное время работы, указанное пользователем (сценарий 1); t_r – действительное время работы задания (сценарий 2).

Симуляция проводилась на данных суперкомпьютера МВС-10П ОП2 (раздел Cascade Lake). Исходными данными была информация об использовании ресурсов за промежуток времени с января по июнь 2021 года. В исходную таблицу вошли 8,5 тыс. строк, каждая из которых – информация об уникальном задании, в том числе данные, указанные пользователем при постановке задания в очередь.

Целью настоящей работы является проведение симуляции на основе сформированного потока с помощью симулятора планировщика SLURM [17, 18]. Предложенный сценарий позволяет оценить максимальную оптимизацию планирования, так как использование действительного времени выполнения задания является идеальным случаем. Предполагается следующая последовательность действий: формирование модельного потока заданий на основе реального, выполнение симуляции по двум сценариям, сравнение результатов по принятым метрикам.

Формирование потока заданий

Если на вход поступает задание с точной на 100 % оценкой времени выполнения, то плани-

ровщик единожды располагает задания в очереди. Ни одна из полученных в работе [2] моделей не обладала точностью 100 %. Для оценки работоспособности модели имеет смысл как эталон использовать поток заданий, в котором в качестве t_{wc} использовано t_r . Анализ такой ситуации был сделан в работах [3, 19] и показал, что в некоторых случаях время ожидания может быть сильно уменьшено точным предсказанием времени работы задания.

В результате анализа данных выяснилось, что на суперкомпьютере МВС-10П ОП2 основную часть заданий составляют задания небольшой длительности (рис. 2). Среднее время ожидания составляет 6 часов.

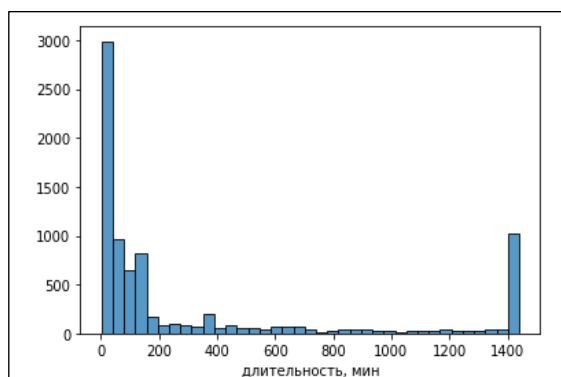


Рис. 2. Распределение заданий по длительности

Fig. 2. Distribution of jobs by runtime

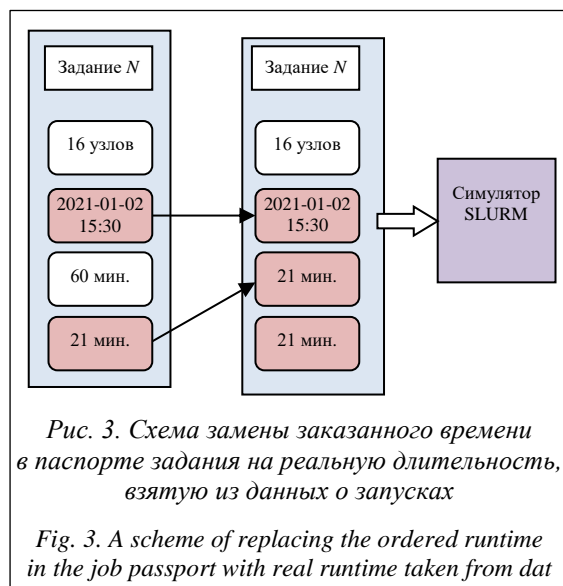
В связи с трудностями в моделировании, например, перерыв на профилактические работы, сравнение модельного потока с реальным в данном случае было бы некорректно.

Необходимо учесть, что в рассматриваемый период времени конфигурация кластера сильно менялась. Его отдельные ресурсы монополично использовались определенными пользователями и потому были недоступны для других. В силу этого было принято решение о создании модели с усредненными постоянными характеристиками.

Для сравнения сценариев сформирован модельный поток заданий на основе данных МСЦ РАН с суперкомпьютера МВС-10П ОП2 (раздел Cascade Lake). В качестве симулятора применен симулятор планировщика заданий SLURM.

При подготовке симуляции для каждого задания были заданы два времени: t_{wc} – время, указанное при постановке задания в очередь, и t_r – реальное время работы программы, взятое из статистических данных.

Смоделированы два варианта запуска симулятора планировщика: поток заданий без изменений и поток, где в качестве заказанного времени указано t_r (рис. 3). То есть второй вариант отражает ситуацию, когда пользователь абсолютно точно указывает время выполнения. Данный вариант показывает, какой эффективности работы планировщика можно достичь при помощи точного предсказания времени работы заданий.



Разные задания имеют различную площадь исполнения (количество ядер, умноженное на количество часов выполнения задания) и разное время выполнения.

Представим конфигурацию раздела Cascade Lake суперкомпьютера МВС-10П ОП2: количество процессоров в узле – 2, количество ядер CPU в процессоре – 24, количество узлов в кластере – 152, всего ядер CPU в системе – 7 296.

На ее основе была создана конфигурация для симуляции. Характеристики узла оставлены без изменений, а число узлов уменьшено до 64 – максимальное число узлов, указанное в тестовом потоке.

Так как выполнение заданий симулируется ожиданием в течение t_r и все узлы системы имеют одинаковые характеристики, учитывать производительность узлов в модели не требуется.

Симулятор SLURM

Планировщики заданий имеют множество настроек, влияющих на их поведение и качество планирования. Качество планирования за-

висит от архитектуры вычислительной системы, настроек планировщика и потока задач пользователей. Качество планирования может измеряться разными метриками, такими как время прохождения задач, утилизация вычислительных ресурсов и т.п. Для достижения оптимальных значений целевой метрики нужно перебрать разные сочетания настроек планировщика и сравнить результаты. Делать это на реальной системе нецелесообразно, так как тратятся вычислительные ресурсы, которые могли бы быть использованы для решения задач пользователей. Кроме того, расходуется время на ожидание завершения тестов, что ограничивает количество вариантов настроек для перебора. Избавиться от этих проблем помогают симуляторы планировщиков пользовательских заданий.

Симулятор не тратит вычислительные ресурсы и выполняет тесты за более короткое время. Это достигается за счет того, что реальные программы, выполняющие вычисления, не запускаются.

Одним из таких симуляторов является Slurm Simulator [20]. Он представляет собой модифицированный планировщик SLURM и скрипты для запуска симулятора и анализа полученных результатов. Симулятор использует вместо реального системного времени модельное, взятое из модельного потока заданий. Кроме того, время масштабируется для уменьшения времени симуляции. Вместо запуска реальных заданий планировщик лишь генерирует события запуска и остановки в соответствии с входным потоком. Это, а также исключение некоторых других действий исходного планировщика позволяют симулятору значительно уменьшить время симуляции.

Результаты

Результаты запусков симулятора планировщика пользовательских задач представлены на рисунке (см. <http://www.swsys.ru/uploaded/image/2022-1/2022-1-dop/1.jpg>).

Разница между временем окончания последнего задания для обоих вариантов составляет 65 часов.

Планировщик SLURM эффективно распределяет задания, несмотря на большую плотность потока. Кластер МВС-10П ОП2 обладает достаточной вычислительной мощностью, чтобы справиться с такой плотностью.

Описанные выше модели алгоритмов машинного обучения имеют точность ниже

100 %, поэтому они заведомо хуже точной оценки времени выполнения задания.

В таблице представлены метрики очереди, позволяющие сравнить два сценария (величины округлены до целого).

Полученные результаты согласуются с работами по данной тематике [3, 19]. Например, в работе [19] авторы на симуляторе Системы управления прохождением параллельных заданий, применяемой в МСЦ РАН, получили схожие значения метрик.

Так как рассмотрен идеальный случай, результаты показывают предельные значения оптимизации при данной конфигурации кластера. Можно сделать вывод, что имеет смысл применять модели машинного обучения для предсказания времени работы задания.

Сравнение сценариев

Scenario comparisons

Метрика	Сценарий 1	Сценарий 2
Среднее время ожидания в очереди, ч	60	45
Приведенное время ожидания, ч	33	15
Утилизация, %	90	92

Из таблицы видно, что применение t_r в качестве времени работы задания позволяет сократить среднее время ожидания в очереди на 15 часов.

Приведенное время ожидания [21], которое вычисляется как $T_{sd} = \frac{T_w + T_r}{T_r}$, где T_w – время от поступления задания до начала его выполнения, T_r – время работы задания, также снижается. Это позволяет заключить, что задания запускаются быстрее.

Утилизация ресурсов увеличивается на 2 пп., что не представляет существенного прироста в масштабах рассматриваемого кластера, и вычисляется как $U = S_{заданий}/S_{кластера}$, где $S_{заданий}$ – суммарная площадь всех заданий в ядро-часах, $S_{кластера}$ – площадь кластера в ядро-часах. Знаменатель в этой формуле ограничен временем завершения последнего задания, а числитель одинаков для обоих сценариев. В сценарии 2 задания завершаются на 65 часов раньше, что и дает большую утилизацию. По этим причинам существенное уменьшение и среднего, и приведенного времени ожидания практически не влияет на утилизацию.

Заключение

В настоящей статье рассмотрена возможность применения симулятора Slurm Simulator для оценки эффективности предсказательных моделей времени работы пользовательских заданий.

Симулятор был применен для сценария, в котором пользователь абсолютно точно указывает время работы задания. Для создания потока заданий использована статистика суперкомпьютера МВС-10П ОП2 (раздел Cascade-Lake), установленного в МСЦ РАН.

Сравнивались два сценария: в одном в паспорт задания вносится указанное пользователем время работы, в другом – точное время работы задания, взятое из статистических данных.

Результаты показывают, что точная оценка времени выполнения задания позволяет существенно снизить среднее время ожидания в очереди и приведенное время ожидания, отражающие скорость запуска задания после поступления.

Работа выполнена в МСЦ РАН как часть государственного задания 0580-2021-0016, а также при поддержке гранта РФФИ № 18-29-03236.

Литература

1. Savin G.I., Shabanov V.M., Nikolaev D.S. et al. Jobs runtime forecast for JSCC RAS supercomputers using machine learning methods. Lobachevskii J. Math., 2020, vol. 41, no. 12, pp. 2593–2602. DOI: 10.1134/S1995080220120343.
2. Рыбаков А.А., Шумилин С.С. Сравнение алгоритмов машинного обучения для предсказания времени работы пользовательских заданий в рамках оптимизации использования ресурсов суперкомпьютерного кластера МСЦ РАН // Тр. НИИСИ РАН. 2020. Т. 10. № 2. С. 4–13.
3. Klusacek D., Soysal M. Walltime prediction and its impact on job scheduling performance and predictability. In: Job Scheduling Strategies for Parallel Processing, 2020, pp. 127–144. DOI: 10.1007/978-3-030-63171-0_7.
4. Баранов А.В., Николаев Д.С. Применение машинного обучения для прогнозирования времени выполнения суперкомпьютерных заданий // Программные продукты и системы. 2020. № 2. С. 218–228. DOI: 10.15827/0236-235X.130.218-228.

5. Andresen D. et al. Machine learning for predictive analytics of compute cluster jobs. ArXiv, 2018, art. 1806.01116v1. URL: <http://arxiv.org/abs/1806.01116> (дата обращения: 28.10.2021).
6. Rezaei M., Salnikov A. Machine learning techniques to perform predictive analytics of task queues guided by slurm. GloSIC, 2018, pp. 1–6. DOI: 10.1109/GLOSIC.2018.8570130.
7. Smith W., Foster I., Taylor V. Predicting application run times using historical information. In: Job Scheduling Strategies for Parallel Processing, 1998, pp. 122–142. DOI: 10.1007/BFb0053984.
8. Smith W., Taylor V., Foster I. Using run-time predictions to estimate queue wait times and improve scheduler performance. In: Job Scheduling Strategies for Parallel Processing, 1999, pp. 202–219. DOI: 10.1007/3-540-47954-6_11.
9. Webb G.I., Pazzani M.J., Billsus D. Machine learning for user modeling. User Modeling and User-Adapted Interaction, 2001, vol. 11, no. 1, pp. 19–29.
10. Yao Y., Zhao Y., Wang J., Han S. A model of machine learning based on user preference of attributes. In: Rough Sets and Current Trends in Computing, 2006, pp. 587–596. DOI: 10.1007/11908029_61.
11. Tsafir D., Etsion Y., Feitelson D.G. Backfilling using system-generated predictions rather than user runtime estimates. IEEE Transactions on Parallel and Distributed Systems, 2007, vol. 18, no. 6, pp. 789–803. DOI: 10.1109/TPDS.2007.70606.
12. Baranov A., Telegin P., Shabanov B., Lyakhovets D. Measure of adequacy for the supercomputer job management system model. Proc. FedCSIS, 2019, vol. 18, pp. 423–426. DOI: 10.15439/2019F186.
13. Tanash M., Dunn B., Andresen D., Hsu W., Yang H., Okanlawon A. Improving HPC system performance by predicting job resources via supervised machine learning. Proc. PEARC, 2019, pp. 1–8. DOI: 10.1145/3332186.3333041.
14. Rodrigues E.R., Cunha R.L.F., Netto M.A.S., Spriggs M. Helping HPC users specify job memory requirements via machine learning. Proc. III Intern. Workshop HUST, 2016, pp. 6–13.
15. Wyatt M.R., Herbein S., Gamblin T., Moody A., Ahn D.H., Taufer M. PRIONN: Predicting runtime and IO using neural networks. Proc. 47th Int. Conf. on Parallel Processing, 2018, pp. 1–12. DOI: 10.1145/3225058.3225091.
16. Cheon H., Ryu J., Ryou J., Park C.Y., Han Y. ARED: automata-based runtime estimation for distributed systems using deep learning. Cluster Computing, 2021, pp. 1–13. DOI: 10.1007/s10586-021-03272-w.
17. ShedMD. Lucero A. Slurm Simulator. 2011. URL: https://slurm.schedmd.com/slurm Ug_2011/slurm_simulator_phoenix.pdf (дата обращения: 28.10.2021).
18. GitHub. Ubccr-slurm-simulator/slurm_simulator. URL: https://github.com/ubccr-slurm-simulator/slurm_simulator (дата обращения: 28.10.2021).
19. Baranov A.V., Lyakhovets D.S. Accuracy comparison of various supercomputer job management system models. Lobachevskii J. Math., 2021, vol. 42, no. 11, pp. 2510–2519. DOI: 10.1134/S199508022111007X.
20. Simakov N.A., Innus M., Jones M., Deleon R., White J., Gallo S., Patra A., Furlani T. A slurm simulator: Implementation and parametric analysis. In: Lecture Notes in Computer Science, 2017, pp. 197–217. DOI: 10.1007/978-3-319-72971-8_10.
21. Mualem A.W., Feitelson D.G. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. IEEE Transactions on Parallel and Distributed Systems, 2001, vol. 12, no. 6, pp. 529–543. DOI: 10.1109/71.932708.

Using job scheduler simulator to evaluate the effectiveness of job run time prediction

*S.S. Shumilin*¹, Senior Researcher, shumilin@jscs.ru
*M.Yu. Vorobev*², Junior Researcher, nordmike@jscs.ru

¹ Joint Supercomputer Center of RAS, Moscow, 119991, Russian Federation

Abstract. The paper investigates the efficiency of queue scheduling using pre-trained models. A supercomputer cluster uses a scheduler to distribute the incoming job flow among the available computing resources. In order to place a job in the queue, the scheduler uses the data specified by a user, including the ordered program runtime. However, users often misjudge the runtime and choose an upper estimate. If the job completes earlier than specified, then the scheduler needs to reschedule the queue. A large number of such events can reduce the efficiency of resource allocation.

Recently, there have been many papers describing the use of machine learning to predict the job run time. This allows using the run time calculated by a pre-trained model during the scheduling process. However, all the models contain an estimation error. Therefore, the problem is the need to assess the efficiency of planning for a given value of the model error.

This paper investigates the effectiveness of the proposed approach by comparing the scheduling efficiency in two scenarios: 1) the scheduler uses the time specified by a user and 2) the scheduler uses the real job runtime. For this purpose, the SLURM scheduler simulator performs simulation on the statistical data of the MVS-10P OP2 supercomputer installed at the Joint Supercomputer Center of the Russian Academy of Sciences. The results show that average waiting time in scenario 2 reduced by 25 %. Slowdown reduced by 50 %. Resource utilization did not change significantly.

The experimental results indicate the practicability of using machine learning algorithms to predict the running time of jobs arriving at a supercomputer cluster. Thus, the article provides an estimate of the ultimate optimization, since the experiment assumes a hundred percent prediction accuracy, which to date is not demonstrated by any of the presented works on runtime prediction.

Keywords: predictive analytics, SLURM, job scheduling, simulator, MVS-10P.

Acknowledgements *The work has been done at the JSCC RAS as part of the state assignment 0580-2021-0016 and also supported with the grant from the Russian Foundation for Basic Research no. 18-29-03236.*

References

1. Savin G.I., Shabanov B.M., Nikolaev D.S. et al. Jobs runtime forecast for JSCC RAS supercomputers using machine learning methods. *Lobachevskii J. Math.*, 2020, vol. 41, no. 12, pp. 2593–2602. DOI: 10.1134/S1995080220120343.
2. Rybakov A.A., Shumilin S.S. Comparison of machine learning algorithms for predicting the user job runtime in the framework of optimizing the use of resources of supercomputer cluster of JSCC RAS. *Proc. SRISA RAS*, 2020, vol. 10, no. 2, pp. 4–13 (in Russ.).
3. Klusacek D., Soysal M. Walltime prediction and its impact on job scheduling performance and predictability. In: *Job Scheduling Strategies for Parallel Processing*, 2020, pp. 127–144. DOI: 10.1007/978-3-030-63171-0_7.
4. Baranov A.V., Nikolaev D.S. Machine learning to predict the supercomputer jobs execution time. *Software & Systems*, 2020, no. 2, pp. 218–228. DOI: 10.15827/0236-235X.130.218-228 (in Russ.).
5. Andresen D. et al. Machine learning for predictive analytics of compute cluster jobs. *ArXiv*, 2018, art. 1806.01116v1. Available at: <http://arxiv.org/abs/1806.01116> (accessed October 28, 2021).
6. Rezaei M., Salnikov A. Machine learning techniques to perform predictive analytics of task queues guided by slurm. *GloSIC*, 2018, pp. 1–6. DOI: 10.1109/GLOSIC.2018.8570130.
7. Smith W., Foster I., Taylor V. Predicting application run times using historical information. In: *Job Scheduling Strategies for Parallel Processing*, 1998, pp. 122–142. DOI: 10.1007/BFb0053984.
8. Smith W., Taylor V., Foster I. Using run-time predictions to estimate queue wait times and improve scheduler performance. In: *Job Scheduling Strategies for Parallel Processing*, 1999, pp. 202–219. DOI: 10.1007/3-540-47954-6_11.
9. Webb G.I., Pazzani M.J., Billsus D. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 2001, vol. 11, no. 1, pp. 19–29.
10. Yao Y., Zhao Y., Wang J., Han S. A model of machine learning based on user preference of attributes. In: *Rough Sets and Current Trends in Computing*, 2006, pp. 587–596. DOI: 10.1007/11908029_61.
11. Tsafir D., Etsion Y., Feitelson D.G. Backfilling using system-generated predictions rather than user runtime estimates. *IEEE Transactions on Parallel and Distributed Systems*, 2007, vol. 18, no. 6, pp. 789–803. DOI: 10.1109/TPDS.2007.70606.
12. Baranov A., Telegin P., Shabanov B., Lyakhovets D. Measure of adequacy for the supercomputer job management system model. *Proc. FedCSIS*, 2019, vol. 18, pp. 423–426. DOI: 10.15439/2019F186.
13. Tanash M., Dunn B., Andresen D., Hsu W., Yang H., Okanlawon A. Improving HPC system performance by predicting job resources via supervised machine learning. *Proc. PEARC*, 2019, pp. 1–8. DOI: 10.1145/3332186.3333041.
14. Rodrigues E.R., Cunha R.L.F., Netto M.A.S., Spriggs M. Helping HPC users specify job memory requirements via machine learning. *Proc. III Intern. Workshop HUST*, 2016, pp. 6–13.
15. Wyatt M.R., Herbein S., Gamblin T., Moody A., Ahn D.H., Tauber M. PRIONN: Predicting runtime and IO using neural networks. *Proc. 47th Int. Conf. on Parallel Processing*, 2018, pp. 1–12. DOI: 10.1145/3225058.3225091.

16. Cheon H., Ryu J., Ryou J., Park C.Y., Han Y. ARED: automata-based runtime estimation for distributed systems using deep learning. *Cluster Computing*, 2021, pp. 1–13. DOI: 10.1007/s10586-021-03272-w.
17. *ShedMD*. Lucero A. *Slurm Simulator*. 2011. Available at: https://slurm.schedmd.com/slurm_ug_2011/slurm_simulator_phoenix.pdf (accessed October 28, 2021).
18. *GitHub*. *Ubccr-slurm-simulator/slurm_simulator*. Available at: https://github.com/ubccr-slurm-simulator/slurm_simulator (accessed October 28, 2021).
19. Baranov A.V., Lyakhovets D.S. Accuracy comparison of various supercomputer job management system models. *Lobachevskii J. Math.*, 2021, vol. 42, no. 11, pp. 2510–2519. DOI: 10.1134/S199508022111007X.
20. Simakov N.A., Innus M., Jones M., Deleon R., White J., Gallo S., Patra A., Furlani T. A slurm simulator: Implementation and parametric analysis. In: *Lecture Notes in Computer Science*, 2017, pp. 197–217. DOI: 10.1007/978-3-319-72971-8_10.
21. Mualem A.W., Feitelson D.G. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. *IEEE Transactions on Parallel and Distributed Systems*, 2001, vol. 12, no. 6, pp. 529–543. DOI: 10.1109/71.932708.

Для цитирования

Шумилин С.С., Воробьев М.Ю. Использование симулятора планировщика заданий для оценки эффективности предсказания времени работы задания // Программные продукты и системы. 2022. Т. 35. № 1. С. 124–131. DOI: 10.15827/0236-235X.137.124-131.

For citation

Shumilin S.S., Vorobev M.Yu. Using job scheduler simulator to evaluate the effectiveness of job run time prediction. *Software & Systems*, 2022, vol. 35, no. 1, pp. 124–131 (in Russ.). DOI: 10.15827/0236-235X.137.124-131.