УДК 539.12

DOI: 10.15827/0236-235X.133.189-194

Дата подачи статьи: 15.12.20

2021. Т. 34. № 1. С. 189–194

Распараллеливание в задачах анализа физических данных эксперимента LHCb

*А.В. Егорычев¹, лаборант, Artem.Egorychev@cern.ch**И.М. Беляев¹, к.ф.-м.н., старший научный сотрудник, Ivan.Bellyaev@cern.ch**Т.А. Овсянникова¹, младший научный сотрудник, Tatiana.Ovsiannikova@cern.ch*

¹ *Институт теоретической и экспериментальной физики им. А.И. Алиханова
Национального исследовательского центра «Курчатовский институт»,
г. Москва, 117218, Россия*

Общий прогресс в эксплуатационных характеристиках оборудования с 1990-х годов резко расширил возможности сборки информационных систем из готовых компонентов и сделал доступными свободно распространяемые программные инструменты конструирования систем программирования, в том числе поддерживающие организацию параллельных процессов если не на уровне языка, то на уровне библиотечных компонент.

В работе представлены результаты применения метода распараллеливания в задачах физического анализа данных эксперимента LHCb, реализованных с помощью программного пакета OSTAP, на базе широко используемого в физике элементарных частиц пакета ROOT. Объемы данных, получаемые в реальном времени в экспериментах Большого адронного коллайдера, требуют высокой производительности вычислений и скорости принятия решений триггерной системой эксперимента. Высокая производительность ПО также является ключевым требованием для анализа данных, поступающих в систему хранения информации, полученных на последующих этапах работы эксперимента. Адаптация ПО к существующим многоядерным и многопроцессорным системам позволяет достичь необходимой вычислительной мощности для эффективного решения задач обработки данных.

Программный пакет OSTAP имеет удобный и доступный для пользователя интерфейс, реализованный на языке Python. Язык Python также зарекомендовал себя как удобное средство разработки распределенных систем и сетевого программирования. Параллельный алгоритм может быть реализован по частям на множестве различных устройств с последующим объединением полученных результатов и получением целевого результата. Мультипарадигматические языки, такие как Python, показывают хорошие результаты в программировании сетевых процессов для многопроцессорных комплексов и привлекают большое число сторонников.

Ключевые слова: LHCb, ROOT, pyROOT, Python, Pathos, SWAN, OSTAP.

Одним из экспериментов на Большом адронном коллайдере (Large Hadron Collider, LHC) по поиску и изучению распадов очарованных и прелестных адронов является LHCb [1]. На данный момент LHCb проводит анализ данных, набранных при энергии протон-протонных (pp) столкновений в системе центра масс 7, 8 и 13 ТэВ и соответствующих интегральной светимости около 9 фб^{-1} [2]. Поток данных, образующихся в pp столкновениях, очень велик. Коллайдер LHC производит информацию до 50 Птбайт/сек., и хранение такого большого объема данных затратно с точки зрения необходимых объемов памяти. К счастью, интересующая физическая информация занимает всего несколько процентов от общего потока данных и отбирается с помощью онлайн триггерной системы каждого эксперимента. Но даже после такого отбора записанная информа-

ция составляет несколько петабайт. Характерный объем данных, использующийся для индивидуального физического анализа, составляет порядка 1–100 Гбайт в эксперименте LHCb.

Таким образом, физический анализ представляет собой достаточно затратный с точки зрения вычислительного ресурса и времени исполнения комплекс задач. На данный момент существует большое количество систем и вычислительных центров с множеством вычислительных узлов и программных библиотек, поддерживающих распараллеливание процессов, что позволяет адаптировать ПО анализа данных к возможностям имеющихся мультипроцессорных и кластерных систем и значительно улучшить показатели времени выполнения задач. Для эффективного решения описанных задач были разработаны набор инструментов и интерфейс для использования распарал-

леливания на многопроцессорных и кластерных системах в рамках программного пакета OSTAP [3], используемого для анализа данных в эксперименте LHCb.

Стандартные пакеты обработки

Как правило, программы для анализа данных в физике высоких энергий реализуются на языках C++ и Python на базе пакетов ROOT и pyROOT [4]. ПО ROOT – пакет объектно-ориентированных программ и библиотек, разработанных в качестве платформы для обработки данных экспериментов физики высоких энергий. Он существует более 20 лет и широко используется для графического представления результатов анализа физических данных. Но, поскольку в настоящее время растет популярность интерпретируемого языка Python, зарекомендовавшего себя как максимально удобный язык для скриптового программирования, все более популярным становится и другой пакет обработки данных. ПО pyROOT – это модуль расширения, основанный на языке Python, позволяющий пользователям взаимодействовать с пакетом ROOT посредством интерпретатора Python. Такой пакет сочетает в себе простоту использования программного языка Python с возможностями пакета ROOT. Также язык Python зарекомендовал себя как удобное средство разработки распределенных систем и сетевого программирования. Параллельный алгоритм может быть реализован по частям на множестве различных устройств с последующим объединением индивидуальных результатов и получением общего, целевого результата.

Типовыми задачами физического анализа в эксперименте LHCb являются следующие: подавление комбинаторного фона и выделение сигнального процесса, сравнение характерных распределений и их параметров между физическими данными и данными математического моделирования [2, 5, 6], извлечение интересных параметров распределений, оценка значимости сигнала и вычисление неопределенностей получаемых физических результатов [7–9]. Некоторые из методов решения этих задач могут быть эффективно распараллелены. Например, подавление фона обычно осуществляется с помощью мультивариативного анализа или фильтрации на основе критериев отбора. Другим характерным примером эффективного применения метода распараллеливания ресурсов может служить задача оценки систематических эффектов, для решения которой использу-

ется упрощенное математическое моделирование характерных распределений.

Среда программирования OSTAP

В эксперименте LHCb разработано уникальное ПО OSTAP, обеспечивающее более удобное и интуитивно понятное для пользователя представление интерфейса для программных пакетов ROOT и PyROOT. Оно позволяет расширять существующую функциональность и включает в себя основные инструменты, необходимые пользователю для выполнения анализа.

Проект разработан в 2009 году и основан на функциях языка программирования Python. Многие функции пакета взяты из проекта Bender [10] – среды анализа физических данных на основе языка Python, используемой в эксперименте LHCb. До осени 2016 года проект входил в состав программного комплекса LHCb и с большим успехом был использован для подготовки около 30 статей по обработке физических данных. Автономная независимая версия ПО для эксперимента LHCb появилась в начале 2017 года. Преимущества пакета OSTAP:

- простые манипуляции с объектами, классами ПО ROOT и RooFit (гистограммы, деревья, наборы данных и т.д.);
- простой и дружелюбный интерфейс для оборудования пакета RooFit;
- расширенный набор моделей плотности распределения вероятности случайной величины для подпрограммы RooFit;
- интерактивная среда анализа OSTAP для описания сигнальной компоненты подгонки расширения с помощью программы RooFit.

Одним из важных преимуществ пакета OSTAP является возможность параллельного запуска нескольких задач на серверах кластера lxplus, который используется для обработки данных. В проект были включены полезные утилиты для поддержки многопроцессорности и параллельной обработки задач на основе пакета Pathos [11]. На рисунке 1 показана структура среды программирования OSTAP.

ПО Pathos реализовано как согласованный высокоуровневый интерфейс для настройки и запуска параллельных вычислений на различных ресурсах. Пакет Pathos предоставляет пользователю настраиваемые программы запуска для параллельных и распределенных вычислений, где каждая программа запуска содержит синтаксическую логику для настройки и запуска заданий в среде выполнения.

Интерфейс для распараллеливания в программе OSTAP

В рамках программных пакетов проекта OSTAP [1] разработан удобный интерфейс,

позволяющий пользователю распараллеливать любые типовые задачи по данным. Структура интерфейса показана на рисунке 2. Также были реализованы несколько дополнительных инструментов для решения типовых задач.

Рис. 1. Структура среды программирования OSTAP

Fig. 1. OSTAP package structure

Рис. 2. Набор инструментов для распараллеливания

Fig. 2. The parallelization tool-kit

Распараллеливание на кластере

The test of parallelization on the cluster system

# Jobs	%	total time	time/jobs	Jobs server	ncpus
191	47.8	5421	28.38	localhost	12
41	10.2	5487	133.8	remotehost1	10
61	15.2	6898	113.1	remotehost2	10
50	12.5	5631	112.6	remotehost3	10

- ReduceTask – стандартный шаблон для фильтрации и хранения данных табличным представлением ntuple и гистограмм.

- ProjectTask – стандартный шаблон для проекции данных табличным представлением ntuple.

- ChopperTraining, AddTMVA, AddChopping – функции для вывода мультивариативного анализа.

- FuncTask – стандартный шаблон для выполнения вызываемого объекта/функции.

- GenericTask – основной управляющий блок для распараллеливания задач.

Проверка процедуры ускорения выполнения задач была проведена для разного количества вычислительных узлов на трех типах примеров, ориентируясь на степень использования блока ввода и вывода данных I/O в процессе выполнения задачи: process, project – максимально задействован блок ввода и вывода данных, TMVA – средняя степень загрузки блока, Toys – блок не используется.

Зависимость ускорения выполнения от количества вычислительных узлов, на которых происходило распараллеливание, показана на рисунке 3. Полученная зависимость практически линейна для всех типов задач. Результаты распараллеливания типовых задач на кластере, состоящем из 5 машин, представлены в таблице. Время выполнения очереди задач на кластере находится на уровне 8 секунд, что почти в 40 раз меньше времени выполнения на одном вычислительном узле. Анализ полученных численных значений подтверждает линейное улучшение времени исполнения задач в зависимости от количества вычислительных узлов.

Стоит также отметить, что была реализована возможность использования дополнительных вычислительных ресурсов для анализа данных – облачный сервис CERN cloud, на котором доступен многопроцессорный режим. Для этого в программный пакет OSTAP интегрирована возможность использования ПО SWAN [12], обеспечивающего запуск задач и интерактивный режим работы на CERN cloud [12]. В качестве интерфейса в таком слу-

чае используется Jupyter notebook. Сеансы пользователей помещаются в изолированные контейнеры, синхронизируются и хранятся через пакет CERNBox [13] (сервис синхронизации и обмена файлами, построенный на основе среды Owncloud [14]). Доступ к большим наборам данных пользователь получает в системе EOS [15] – дисковой системе хранения данных в Европейском центре ядерных исследований.

Рис. 3. Ускорение выполнения задач

Fig. 3. The problems gain

Заключение

Высокая производительность ПО является ключевым требованием для анализа данных. В рамках программного пакета OSTAP, использующегося для анализа данных в эксперименте LHCb, был разработан набор инструментов и интерфейсов, позволяющих применять распараллеливание на многоядерных и многопроцессорных системах, что значительно сократило время вычисления и проведения анализа данных. Тестирование разработанного расширения пакета OSTAP показало практически линейную зависимость времени исполнения задач от количества вычислительных данных для всех типовых задач анализа.

Литература

1. Aaij R., Adeva B., Adinolfi M., Affolder A., Ajaltouni Z., Akar S. et al. LHCb detector performance. *IJMPA*, 2015, vol. 30, no. 7, art. 1530022. DOI: 10.1142/S0217751X15300227.
2. Aaij R., Beteta C.A., Ackernley T. et al. Study of the $\psi_2(3823)$ and $\chi_{c1}(3872)$ states in $B^+ \rightarrow (J/\psi \pi^+ \pi^-) K^+$ decays. *JHEP*, 2020, vol. 08, pp. 123–150.
3. Belyaev I., Egoruchev A., Mazurov A., Ovsiannikova T. OstapHEP/ostap: v1.5.0.4. URL: <https://zenodo.org/record/4005683> (дата обращения: 05.12.2020).
4. Brun R., Rademakers F. ROOT – An object oriented data analysis framework. *NUCL INSTRUM METH A*, 1997, vol. 389, pp. 81–86. DOI: 10.1016/S0168-9002(97)00048-X.
5. Перейма Д.Ю. Поиск новых распадов прелестных частиц в эксперименте LHCb. Атореф. НИЦ «Курчатовский институт» ИТЭФ, 2020. 26 с.
6. Aaij R. et al. Study of $B^0_s \rightarrow J/\psi p^+ K^- K^-$ decays. URL: <https://arxiv.org/abs/2011.01867> (дата обращения: 05.12.2020).
7. Aaij R., Adeva B., Adinolfi M., Aidala C.A., Ajaltouni Z., Akar S. et al. Observation of the decay $\Lambda^0_b \rightarrow \psi(2S) p \pi^-$. *JHEP*, 2018, vol. 8, pp. 131–147. DOI: 10.1007/jhep08(2018)131.
8. The LHCb collaboration, Aaij R., Abellán Beteta C., et al. Observation of the $\Lambda^0_b \rightarrow \chi_{c1}(3872) p K^-$ decay. *JHEP*, 2019, vol. 09, pp. 28–47. DOI: 10.1007/JHEP09(2019)028.
9. The LHCb collaboration, Aaij R., Abellán Beteta C., et al. Study of the line shape of the $\chi_{c1}(3872)$ state. *PRD*, 2020, vol. 102, no. 9, art. 092025. DOI: 10.1103/PhysRevD.102.092025.
10. Belyaev I., Frank M., Mato P., Barrand G., Tsaregorodtsev A.Yu., De Oliveira E. Python-based physics analysis environment for LHCb. *Proc. Computing in High Energy Physics and Nuclear Physics*, 2004, pp. 377–380. URL: <https://inspirehep.net/literature/928906> (дата обращения: 05.12.2020).
11. McKerns M., Strand L., Sullivan T., Fang A., Aivazis M. Building a framework for predictive science. *Proc. SCIPY*, 2011. URL: <http://arxiv.org/pdf/1202.1056> (дата обращения: 05.12.2020).
12. Piparo D., Tejedor E., Mato P., Mascetti L., Moscicki J., Lamanna M. SWAN: A service for interactive analysis in the cloud. *FGCS*, 2016, vol. 78, pp. 1071–1078. DOI: 10.1016/j.future.2016.11.035.
13. Mascetti L., Labrador H.G., Lamanna M., Mościcki J.T., Peters A.J. CERNBox+ EOS: end-user storage for science. *JPCS*, 2015, vol. 664, is. 6, art. 062037. DOI: 10.1088/1742-6596/664/6/062037.
14. Mościcki J.T., Lamanna M. Prototyping a file sharing and synchronization service with ownCloud. *JPCS*, 2014, vol. 513, is. 4, art. 042034. DOI: 10.1088/1742-6596/513/4/042034.
15. Peters A.J., Janyst L. Exabyte scale storage at CERN. *JPCS*, 2011, vol. 331, is. 5, art. 052015. DOI: 10.1088/1742-6596/331/5/052015.

Software & Systems
DOI: 10.15827/0236-235X.133.189-194

Received 15.12.20
2021, vol. 34, no. 1, pp. 189–194

Parallelization in the analysis of physical data of the LHCb experiment

A.V. Egorychev¹, Laboratory Assistant, Artem.Egorychev@cern.ch

I.M. Belyaev¹, Ph.D. (Physics and Mathematics), Senior Researcher, Ivan.Belyaev@cern.ch

T.A. Ovsiannikova¹, Junior Researcher, Tatiana.Ovsiannikova@cern.ch

¹ Institute for Theoretical and Experimental Physics named by A.I. Alikhanov of National Research Centre "Kurchatov Institute", Moscow, 117218, Russian Federation

Abstract. The general progress in hardware performance since the 1990s has completely expanded the ability to build information systems from ready-made components and made available freely distributed software tools for designing programming systems, including those that support the organization of parallel processes, if not at the language level itself, then at the level of library components.

The paper presents the application results of parallelization in physical data analysis problems of the LHCb experiment. The current realization is implemented in the OSTAP framework based on the ROOT and python packages. The amount of the data obtained in proper time in experiments at the Large Hadron Collider require a high speed of preprocessing, which means high computing performance. The high processing speed is also a major requirement for analyzing the data obtained in the subsequent stages. Adaptation of the software to

modern multi-core and multiprocessor systems makes it possible to achieve the necessary computing power for efficiently solving the data analysis problems in experiments of elementary particle physics.

The OSTAP software package has a user-friendly interface which is implemented by using the Python. The Python has also established itself as a powerful tool for developing distributed systems and network programming. The parallel algorithm can be implemented in parts on many different devices with the subsequent combination of the obtained results and obtaining the target result. Multiparadigmatic languages, such as Python, show excellent results in programming network processes for multiprocessor systems and attract many supporters.

Keywords: LHCb, ROOT, pyROOT, Python, Pathos, SWAN, OSTAP.

References

1. Aaij R., Adeva B., Adinolfi M., Affolder A., Ajaltouni Z., Akar S. et al. LHCb detector performance. *IJMPA*, 2015, vol. 30, no. 7, art. 1530022. DOI: 10.1142/S0217751X15300227.
2. Aaij R., Beteta C.A., Ackernley T. et al. Study of the $\psi_2(3823)$ and $\chi_{c1}(3872)$ states in $B^+ \rightarrow (J/\psi \pi^+ \pi^-) K^+$ decays. *JHEP*, 2020, vol. 08., pp. 123–150.
3. Belyaev I., Egoruchev A., Mazurov A., Ovsiannikova T. *OstapHEP/ostap: v1.5.0.4*. Available at: <https://zenodo.org/record/4005683> (accessed December 05, 2020) (in Russ.).
4. Brun R., Rademakers F. ROOT – An object oriented data analysis framework. *NUCL INSTRUM METH A*, 1997, vol. 389, pp. 81–86. DOI: 10.1016/S0168-9002(97)00048-X.
5. Pereima D. *Search for New Decays of Beauty Particles at the LHCb Experiment*. Ph.D. thesis, NRC Kurchatov Inst. ITEP, 2020, 26 p. (in Russ.).
6. R. Aaij et al., Study of $B^0_s \rightarrow J/\psi p^+ K^- K^-$ decays. URL: <https://arxiv.org/abs/2011.01867> (accessed December 05, 2020).
7. Aaij R., Adeva B., Adinolfi M., Aidala C.A., Ajaltouni Z., Akar S. et al. Observation of the decay $\Lambda_b^0 \rightarrow \psi(2S) p \pi^-$. *JHEP*, 2018, vol. 8, pp. 131–147. DOI: 10.1007/jhep08(2018)131.
8. The LHCb collaboration., Aaij R., Abellán Beteta C., et al. Observation of the $\Lambda_b^0 \rightarrow \chi_{c1}(3872) p K^-$ decay. *JHEP*, 2019, vol. 09, pp. 28–47. DOI: 10.1007/JHEP09(2019)028.
9. The LHCb collaboration, Aaij R., Abellán Beteta C., et al. Study of the line shape of the $\chi_{c1}(3872)$ state. *PRD*, 2020, vol. 102, no. 9, art. 092025. DOI: 10.1103/PhysRevD.102.092005.
10. Belyaev I., Frank M., Mato P., Barrand G., Tsaregorodtsev A.Yu., De Oliveira E. Python-based physics analysis environment for LHCb. *Proc. Computing in High Energy Physics and Nuclear Physics*, 2004, pp. 377–380. Available at: <https://inspirehep.net/literature/928906> (accessed December 05, 2020).
11. McKerns M., Strand L., Sullivan T., Fang A., Aivazis M. Building a framework for predictive science. *Proc. SCIPY*, 2011. Available at: <http://arxiv.org/pdf/1202.1056> (accessed December 05, 2020).
12. Piparo D., Tejedor E., Mato P., Mascetti L., Moscicki J., Lamanna M. SWAN: A service for interactive analysis in the cloud. *FGCS*, 2016, vol. 78, pp. 1071–1078. DOI: 10.1016/j.future.2016.11.035.
13. Mascetti L., Labrador H.G., Lamanna M., Mościcki J.T., Peters A.J. CERNBox+ EOS: end-user storage for science. *JPCS*, 2015, vol. 664, is. 6, art. 062037. DOI: 10.1088/1742-6596/664/6/062037.
14. Mościcki J.T., Lamanna M. Prototyping a file sharing and synchronization service with ownCloud. *JPCS*, 2014, vol. 513, is. 4, art. 042034. DOI: 10.1088/1742-6596/513/4/042034.
15. Peters A.J., Janyst L. Exabyte scale storage at CERN. *JPCS*, 2011, vol. 331, is. 5, art. 052015. DOI: 10.1088/1742-6596/331/5/052015.

Для цитирования

Егорычев А.В., Беляев И.М., Овсянникова Т.А. Распараллеливание в задачах анализа физических данных эксперимента LHCb // Программные продукты и системы. 2021. Т. 34. № 1. С. 189–194. DOI: 10.15827/0236-235X.133.189-194.

For citation

Egorychev A.V., Belyaev I.M., Ovsiannikova T.A. Parallelization in the analysis of physical data of the LHCb experiment. *Software & Systems*, 2021, vol. 34, no. 1, pp. 189–194 (in Russ.). DOI: 10.15827/0236-235X.133.189-194.