ЯДЕРНАЯ ФИЗИКА И ИНЖИНИРИНГ, 2021, том 12, № 2, с. 95–99

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЯДЕРНЫХ ТЕХНОЛОГИЯХ

УДК 539.12

ИСПОЛЬЗОВАНИЕ РАСПАРАЛЛЕЛИВАНИЯ В ЗАДАЧАХ АНАЛИЗА ФИЗИЧЕСКИХ ДАННЫХ ЭКСПЕРИМЕНТА LHCb

© 2021 г. А. В. Егорычев[a], *, И. М. Беляев[a], Т. А. Овсянникова[a]

[a]НИЦ "Курчатовский Институт" – ИТЭФ, Москва, 117218 Россия

**E-mail: artem.egorychev@cern.ch*

Поступила в редакцию 28.12.2020 г.
После доработки 30.12.2020 г.
Принята к публикации 14.01.2021 г.

Технологии распараллеливания являются в настоящее время активно развивающейся сферой в методах разработки программного обеспечения. Адаптация программного обеспечения к существующим многоядерным и многопроцессорным системам, применяя распараллеливание, позволяет существенно повысить производительность вычислений и значительно уменьшить скорость обработки большого массива данных. Высокая скорость работы программного обеспечения важна как для предварительной обработки данных в режиме реального времени, так и для анализа данных на всех последующих этапах, включая и этап визуализации данных. Это требование является ключевым для экспериментов Большого адронного коллайдера (БАК). В докладе будут представлены результаты применения технологии распараллеливания в задачах анализа физических данных эксперимента LHCb, реализованных в программном обеспечении OSTAP, на базе программных пакетов pathos и pyROOT.

Ключевые слова: LHCb, ROOT, pyROOT, python, pathos, swan, OSTAP

DOI: 10.56304/S2079562920060196

ВВЕДЕНИЕ

Эксперимент LHCb один из экспериментов на большом адронном коллайдере по поиску и изучению распадов очарованных и прелестных адронов [1]. На данный момент эксперимент 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$^{-1}$. Поток данных, образующихся в pp столкновениях, очень велик. Коллайдер БАК может производить информацию до 50 Пбайт/с и хранение такого большого количества данных было бы очень ресурсо-затратно. К счастью, интересующая физическая информация занимает всего несколько процентов от общего потока данных и фильтруется с помощью индивидуальной онлайн триггерной системы для каждого эксперимента. Но даже после такого отбора и фильтрации записанная информация составляет несколько петабайт. Характерный объем данных, использующийся для индивидуального физического анализа, составляет порядка 1–100 Гбайт на 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 доступных систем с множеством вычислительных узлов и программных библиотек, поддерживающих распараллеливание процессов, что позволяет адаптировать программное обеспечение анализа данных к возможностям существующих мультипроцессорных и кластерных систем и существенно улучшить время выполнения задач.

СТАНДАРТНЫЕ ПАКЕТЫ ОБРАБОТКИ В ФЭЧ

Как правило, пакеты для анализа данных в физике высоких энергий реализуется на языках C++ и python на базе пакетов ROOT и pyROOT [2]. Программное обеспечение ROOT – пакет объектно-ориентированных программ и библиотек, разработанных в качестве платформы для обработки данных экспериментов физики высоких энергий. Он существует уже более 20 лет и широко используется для графического представления результатов анализа физических данных. Но, поскольку в настоящее время все большую популярность имеет интерпретируемый язык python, зарекомендовавший себя, как максимально удобный язык для скриптового программирования, то все большую популярность приобретает другой пакет обработки данных. Программное обеспечение PyROOT – это модуль расширения, основанный на языке python, который позволяет пользователям взаимодействовать с пакетом ROOT посредством интер-

претатора python. Такой пакет сочетает в себе простоту использования программного языка python с возможностями пакета ROOT. Так же 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 полученных индивидуальных результатов и получением общего, целевого результата.

Типовыми задачами физического анализа на эксперименте LHCb являются следующие проблемы: подавление комбинаторного фона и выделение сигнального процесса, сравнение характерных распределений и их параметров между физическими данными и данными математического моделирования, извлечение интересующих параметров распределений, оценка значимости сигнала и вычисление неопределенностей, получаемых физических результатов [3–6]. Некоторые из методов решения этих задач могут быть эффективно распараллелены. Например, подавление фона обычно выполняется с помощью мультивариативного анализа или с помощью фильтрации на основе критериев отбора. Другим характерным примером, в котором эффективно применяется метод распараллеливания ресурсов, может служить задача оценки систематических эффектов, для решения которой используется упрощенное математическое моделирование характерных распределений.

СРЕДА ПРОГРАММИРОВАНИЯ OSTAP

В эксперименте LHCb разработано уникальное программное обеспечение OSTAP [7], которое обеспечивает более удобное и интуитивно понятное для пользователя представление интерфейса для программного пакета ROOT и PyROOT. Оно позволяет осуществлять расширение существующей функциональности и включает в себя основные инструменты, необходимые пользователю для выполнения анализа.

Проект разработан в 2009 году и основывается на функциях языка программирования python. Многие функции пакета взяты из проекта Bender [8] – 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 программного обеспечения для эксперимента LHCb появилась в начале 2017 года. Преимуществами пакета OSTAP являются:

• простые манипуляции с объектами и классами программного обеспечения ROOT и RooFit: гистограммы, деревья, наборы данных и т.д.;

• простой и дружественный 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. На рис. 1 показана структура среды программирования OSTAP.

Программное обеспечение pathos [9] 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 был разработан удобный интерфейс, позволяющий пользователю распараллеливать любые типовые задачи по данным. Структура интерфейса показана на рис 2. Так же несколько дополнительных инструментов были реализованы для решения типовых задач:

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

2. ProjectTask – стандартный шаблон для проекции данных табличным представлением ntuple;

3. ChopperTraining, AddChopping, AddTMVA функции для вывода мультивариативного анализа;

4. FuncTask – стандартный шаблон для выполнения вызываемого объекта/функции;

5. 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 вывода и ввода данных I/O в процессе выполнения задачи:

1. Задача process, project – максимально задействован блок выводы и ввода данных;

2. Задача TMVA – степень загрузки блок выводы и ввода данных средняя;

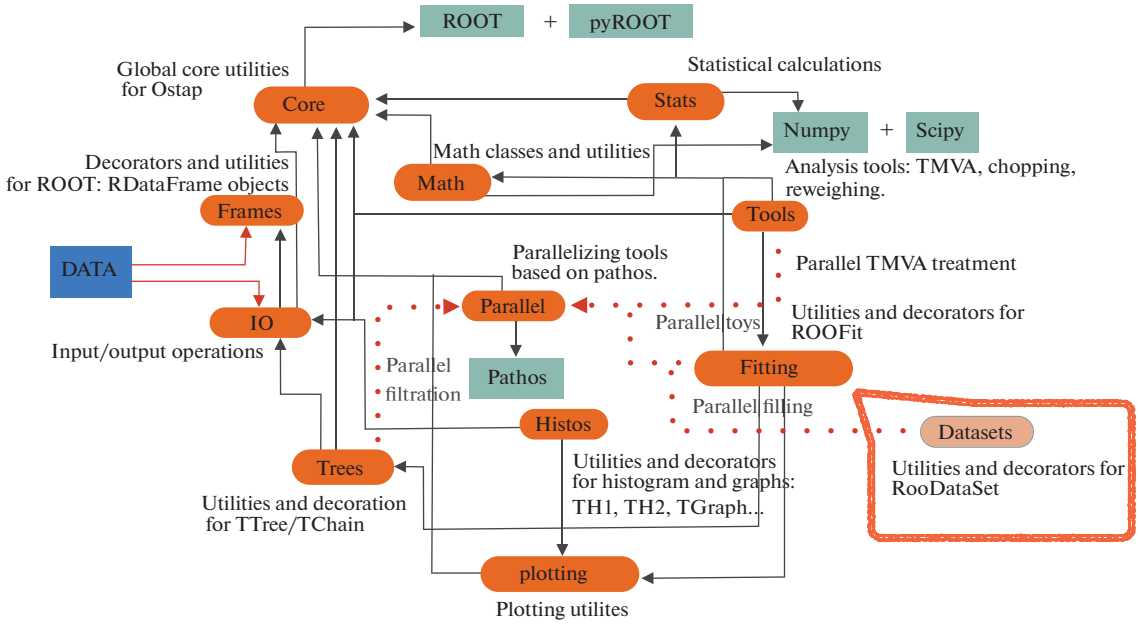

Рис. 1. Структура среды программирования OSTAP.

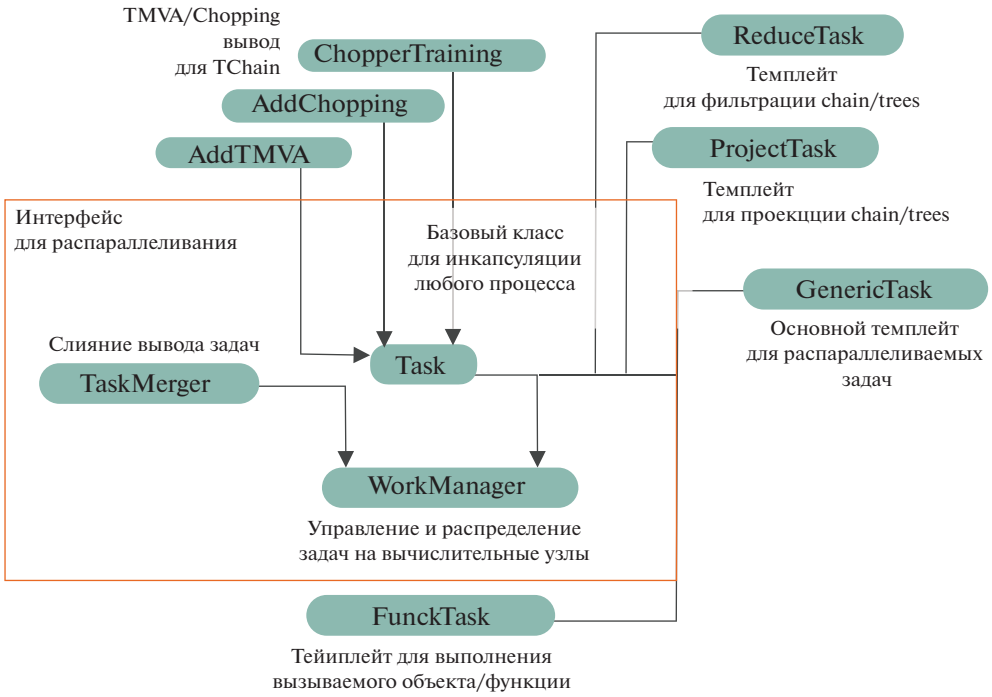

Рис. 2. Набор инструментов для распараллеливания.

3. Задача Toys – блок выводы и ввода данных практически не используется.

На рис. 3 показана зависимость ускорения выполнения от количества вычислительных узлов, на которых происходило распараллеливание. Видно, что зависимость практически линейна для всех типов задач. Результаты распараллеливании типовых задач на кластере, состоящим из 5 машин представлены в табл. 1. Анализ полученных численных значений подтверждает линейное улучшение времени исполнения задач от количества вычислительных узлов.

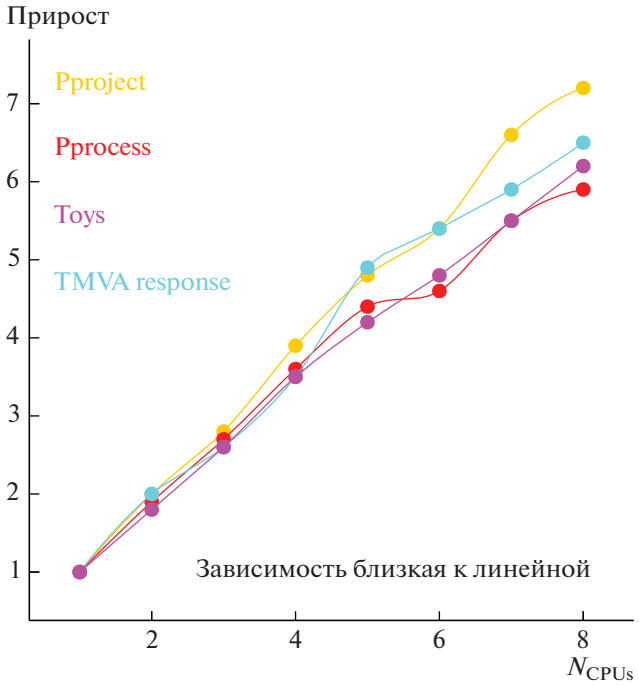

Рис. 3. Ускорение выполнения задач. Тесты мультипроцессинга на четырех видах задач: Pproject, Pprocess – максимальное задействован IO, TMVA response – среднее обращение с IO, Toys – минимальное IO.

Таблица 1. Распараллеливание на кластере. Статистика выполнения очереди задач 12:30 с против 08:22:10 с (прирост 39.6)

Статистика выполнения очереди задач 12:30c vs 08.22.10c (Gain 39.6)

# Задачи	%	Полное время	Время/Задачи	Сервер задач	N_{CPUs}
191	47.8	5421	28.38	Localhost	12
41	10.2	5487	133.8	Lxplus1	10
61	15.2	6898	113.1	Lxplus2	10
50	12.5	5631	112.6	Lxplus3	10
57	14.2	6692	117.4	Lxplus4	10

Стоит также отметить, что пакет OSTAP может использоваться не только на платформе Linux, но также на docker-контейнере, среде conda и на сервисе для веб-анализа данных CERN cloud/SWAN [10]. Программное обеспечение SWAN позволяет анализировать данные используя среду CERN cloud, на которой доступен многопроцессорный режим. В качестве интерфейса в таком случае используется Jupyter notebook. Сеансы пользователей помещаются в изолированные контейнеры, синхронизируются и хранятся через пакет CERNBox (сервис синхронизации и обмена файлами, построенный на основе среды Owncloud). Доступ к большим наборам данных пользователь получает в системе EOS – дисковой системе хранения данных ЦЕРН.

ЗАКЛЮЧЕНИЕ

Технологии распараллеливания являются в настоящее время активно развивающейся сферой в методах разработки программного обеспечения. Адаптация программного пакета OSTAP к многоядерным и многопроцессорным системам и кластерам позволила существенно повысить произво-

дительность вычислений и сократить время проведения анализа.

СПИСОК ЛИТЕРАТУРЫ/REFERENCES

1. *Aaij R. et al.* // Int. J. Mod. Phys. A. 2015. V. 30. P. 1530022.
2. *Brun R. and Rademakers F.* // Nucl. Instrum. Methods Phys. Res., Sect. A. 1997. V. 389. P. 81. https://doi.org/10.1016/S0168-9002(97)00048-X https://doi.org/10.5281/zenodo.3895860
3. *Aaij R. et al.* // J. High Energy Phys. 2020. V. 08. P. 123.
4. *Pereima D.* // Search for New Decays of Beauty Particles at the LHCb Experiment. PhD Thesis. NRC Kurchatov Inst., ITEP. 2020 . Moscow. CERN-THESIS-2020–204. https://cds.cern.ch/record/2745798?ln=en.
5. *Aaij R. et al.* // J. High Energy Phys. 2021. V. 02. P. 24. https://doi.org/10.1007/JHEP02(2021)024
6. *Aaij R. et al.* // J. High Energy Phys. 2018. V. 08. P. 131.
7. OstapHEP/ostap: v1.5.0.4. 2020. https://doi.org/10.5281/zenodo.4005683
8. *Belyaev I. et al.* // CERN Report CERN-LHCb-2004-089. https://inspirehep.net/literature/928906.
9. *McKerns M.M. et al.* // Building a Framework for Predictive Science. Proc. 10th Python in Science Conf., 2011. http://arxiv.org/pdf/1202.1056.
10. *Pipalo D. et al.* // J. High Energy Phys. 2021. V. 02. P. 24. https://doi.org/10.1007/JHEP02(2021)024; Future Gener. Comput. Syst. 2016. V. 78 (3). P. 1071. https://doi.org/10.1016/j.future.2016.11.035

Using Parallelization in the Analysis of Physics Data of the LHCb Experiment

A. V. Egorychev[1], *, I. M. Belyaev[1], and T. A. Ovsiannikova[1]

[1]*Alikhanov Institute for Theoretical and Experimental Physics, National Research Centre "Kurchatov Institute", Moscow, 117218 Russia*

**e-mail: artem.egorychev@cern.ch*

Received December 28, 2020; revised December 30, 2020; accepted January 14, 2021

Abstract—Usage of parallelised tools allows the efficient exploitation of the modern multicore and muptiprocessor systems and significant reduction of the processing time for the interactive data analysis. The development and performance of the parallelisation software within the OSTAP project, used for the interactive analysis of LHCb data, are reported. The efficient multicore and multiprocessing paralellisation is achieved via usage the pathos framework for heterogeneneous computing.

Keywords: LHCb, ROOT, pyROOT, python, pathos, swan, OSTAP