

УДК: 004.432, 517.958

Экосистема ML/DL/HPC платформы HybriLIT (ЛИТ ОИЯИ): новые возможности для прикладных исследований

Ю. А. Бутенко¹, М. И. Зуев^{1,а}, М. Чосич², А. В. Нечаевский¹,
Д. В. Подгайный¹, И. Р. Рахмонов¹, А. В. Стадник¹, О. И. Стрельцова¹

¹ Международная межправительственная организация
Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, Московская обл., ул. Жолио-Кюри, д. 6

² Институт ядерных наук «Винча», Университет Белграда,
Сербия, Винча, Белград, 11351, Мике Петровича Аласа 12-14

E-mail: ^а zuevmax@jinr.ru

Получено дд.мм.гггг, *после доработки* — ддмм.гггг.

Принято к публикации дд.мм.гггг.

В работе представлены возможности по использованию экосистемы ML/DL/HPC, развернутой на гетерогенной вычислительной платформе HybriLIT (Лаборатория информационных технологий им М.Г. Мещерякова ОИЯИ) на базе JupyterHub и предоставляющей возможности решения задач не только в области машинного обучения и глубокого обучения, но и для удобной организации проведения расчетов и научной визуализации. Экосистема позволяет разрабатывать и реализовывать на Python программные модули, проводить методические расчеты. Актуальность разворачивания такой среды, прежде всего, связана с большой востребованностью программных модулей, которые предоставляются группе исследователей или научному сообществу, когда все этапы исследования могут быть воспроизведены, код модифицирован и использовался научным сообществом. На примере решения конкретной задачи по изучению динамики намагниченности в джозефсоновском ф0-переходе представлена методика разработки программных модулей, позволяющая не только проводить расчеты, но и визуализировать результаты исследования и сопровождать их необходимыми формулами и пояснениями. Показана возможность параллельной реализации алгоритма для проведения расчетов при различных значениях параметров модели на основе Python-библиотеки Joblib, и представлены результаты проведенных вычислительных экспериментов, показывающие эффективность параллельной обработки данных.

Ключевые слова: параллельные вычисления, джозефсоновские переходы, Jupyter Notebook

Работа выполнена при финансовой поддержке Российского научного фонда в рамках научного проекта № 22-71-10022

© 2022 Юрий Александрович Бутенко, Зуев Максим Игоревич, Марко Косик, Андрей Васильевич Нечаевский, Дмитрий Владимирович Подгайный, Илхом Рауфович Рахмонов, Алексей Викторович Стадник, Оксана Ивановна Стрельцова

UDC: 004.432, 517.958

ML/DL/HPC Ecosystem of the HybriLIT Heterogeneous Platform (MLIT JINR): New Opportunities for Applied Research

Yu. A. Butenko¹, M. I. Zuev^{1,a}, M. Ćosić², A. V. Nechaevskiy¹,
D. V. Podgainy¹, I. R. Rahmonov¹, A. V. Stadnik¹, O. I. Streltsova¹

¹ International Intergovernmental Organization

Joint Institute for Nuclear Research,

6, Joliot-Curie st., Dubna, Moscow Region, 141980, Russia

² Vinča Institute Of Nuclear Sciences, University of Belgrade

12-14 Mike Petrovića Alasa, Vinča, Beograd, 11351, Srbija

E-mail: ^a zuevmax@jinr.ru

Received 00.00.2022, after completion — 00.00.2022.

Accepted for publication 00.00.2022.

The work presents the possibilities for using the ML/DL/HPC ecosystem deployed on the HybriLIT Heterogeneous Platform (Meshcheryakov Laboratory of Information Technologies JINR) on top of JupyterHub, which provides opportunities for solving tasks not only in the field of machine learning and deep learning, but also for the convenient organization of calculations and scientific visualization. The ecosystem allows one to develop and implement program modules in Python, as well as to carry out methodical computations. The relevance of deploying such an environment is primarily associated with the great demand for software modules that are provided to a group of researchers or the scientific community, when all stages of the study can be reproduced; the code has been modified and used by the scientific community. Using the example of solving a specific problem to study the dynamics of magnetization in a Phi-0 Josephson Junction (Superconductor-Ferromagnet-Superconductor structure), a methodology for developing software modules is presented; it enables not only to carry out calculations, but also to visualize the results of the study and accompany them with the necessary formulas and explanations. The possibility of parallel implementation of the algorithm for performing computations for various values of parameters of the model based on the Joblib Python library is shown, and the results of computational experiments demonstrating the efficiency of parallel data processing are presented.

Keywords: parallel computing, Josephson junctions, Jupyter Notebook

Citation: *Computer Research and Modeling*, 2016, vol. 8, no. ?, pp. ???–??? (Russian).

The work was supported by the Russian Science Foundation grant, project no.22-71-10022.

Введение

С целью ускорения изучения систем, основанных на джозефсоновских переходах, имеющих критически важное значение для современной микроэлектроники, разрабатываются параллельные реализации алгоритмов для решения соответствующих систем нелинейных дифференциальных уравнений с использованием технологий параллельного программирования, а для предоставления удобной исследовательской среды разрабатываются информационные системы, позволяющие проводить расчеты на вычислительных платформах с гибридной архитектурой и в облачных средах [Виртуальная исследовательская среда...; Гетерогенная платформа HybriLIT; Облачная инфраструктура ОИЯИ; JINR cloud service...]. Отличительной особенностью таких информационных систем является агрегация разработанных моделей, которые позволяют пользователям менять различные параметры, а далее проводить расчеты в батч-режиме. Однако, на этапе исследования или для формирования лабораторных работ для студентов, представляет интерес возможность как модифицировать уравнения, так и последовательно воспроизводить все этапы расчетов: наглядно видеть уравнения, строить все промежуточные графики и отображать необходимые промежуточные расчеты. Такую возможность предоставляет среда, построенная на базе экосистемы Jupyter Notebook [Kong et al.; Project Jupyter], с языком программирования Python, активно развиваемая и включающая широкий набор математических библиотек, библиотек для научной визуализации и подходами к распараллеливанию алгоритмов. Отметим, что Jupyter Notebook сейчас активно используется для задач машинного и глубокого обучения, а накопленный инструментарий может применяться более широко и для научных исследований, требующих, в том числе численных расчетов. Данный подход в работе продемонстрирован на примере решения задачи по изучению возможности переворота магнитного момента в структурах сверхпроводник-ферромагнетик-сверхпроводник, в которых спин-орбитальное взаимодействие в ферромагнетике обеспечивает механизм прямой связи между магнитным моментом и джозефсоновским током – такие джозефсоновские контакты называются φ_0 -переходами [Atanasova et al., 2019]. Отметим, что исследование таких систем связано с многочисленными вычислительными экспериментами для различных моделей и для большого количества параметров моделей. Для ускорения вычислений особую актуальность представляет разработка параллельной реализации на Python.

Экосистема ML/DL/HPC

Гетерогенная вычислительная платформа HybriLIT [Гетерогенная платформа HybriLIT] является частью Многофункционального информационно-вычислительного комплекса, Лаборатории информационных технологий им. М.Г. Мещерякова ОИЯИ, г. Дубна [МИВК ОИЯИ]. Платформа состоит из суперкомпьютера «Говорун» [Суперкомпьютер «Говорун»], учебно-тестового полигона «HybriLIT» [Учебно-тестовый полигон] и экосистемы ML/DL/HPC [Экосистема ML/DL/HPC] для задач машинного обучения, глубокого обучения и анализа данных.

В состав экосистемы гетерогенной вычислительной платформы HybriLIT входят следующие вычислительные компоненты:

- jhub.jinr.ru – виртуальная машина, предназначенная для разработки моделей и алгоритмов на базе JupyterHub, многопользовательской платформы по работе с Jupyter Notebook: KVM vCPU (2.6 Ghz, 24 ядра), 128 GB RAM;
- jhub2.jinr.ru – для проведения ресурсоемких, массивно-параллельных расчетов, в том числе с использованием графических ускорителей NVIDIA: 2x Intel Xeon Gold 6148 (2.4 Ghz, 20 ядер), 4x NVIDIA Volta V100, 512 GB RAM;
- jlabhpc.jinr.ru – виртуальная машина для проведения расчетов на вычислительных узлах гетерогенной платформы HybriLIT с использованием планировщика задач SLURM, разработки

приложений, анализа данных и научной визуализации полученных результатов: Intel Xeon Gold 6126 (2.6 Ghz, 24 ядра), 32 GB RAM;

- studhub.jinr.ru – для проведения лекций, tutorиалов и мастер-классов для студентов, аспирантов и научных сотрудников ОИЯИ: 2x Intel Xeon Gold 6152 (2.1 Ghz, 22 ядра), 512 GB RAM.

На всех компонентах экосистемы установлены основные библиотеки и фреймворки для задач машинного и глубокого обучения, разработка алгоритмов и проведение исследований ведется на базе Jupyter Notebook.

Jupyter Notebook представляет собой среду программирования на языке Python с возможностью работать с отдельными блокнотами Jupyter, текстовыми файлами, изображениями различных форматов, датасетами. В блокноте есть возможность разбить код на отдельные фрагменты, записанные в виде отдельных ячеек, и выполнить их в произвольном порядке. Результат запуска каждой отдельной ячейки отображается в этом блокноте. Интерфейс Jupyter Notebook открывается в браузере, используемом по умолчанию. Все дополнительные библиотеки, не представленные с базовой версии Jupyter, можно установить в процессе работы с блокнотами.

Готовый файл ноутбука можно сохранить в удобном формате, например, HTML или PDF. При необходимости провести расчеты заново или пересчитать с новыми начальными параметрами достаточно установить окружение Jupyter через менеджер пакетов Anaconda [Anaconda] на рабочую машину, открыть в ней полученный файл и произвести запуск всех имеющихся в нем ячеек или отдельные части кода.

Так же есть возможность формировать отчеты-ноутбуки, содержащие помимо ячеек с кодом, ячейки в формате Markdown [Markdown], содержащие описание решаемой задачи с формулами, сопроводительные рисунки и графики. Помимо отдельных файлов, ноутбуки можно организовывать в отдельные Интернет-ресурсы с уникальной ссылкой, зайдя по которой пользователь может ознакомиться с описанием задачи, скопировать необходимый код и запустить его на своей рабочей машине [Jupyter Book].

Так же в рамках экосистемы есть возможность запускать расчеты в пакете MATLAB, используя интерфейс JupyterHub [MATLAB в среде JupyterHub].

Доступ на компоненты экосистемы ML/DL/HPC предоставляется зарегистрированным пользователям гетерогенной платформы HybriLIT.

Постановка физической задачи

В качестве примера работы в среде Jupyter Notebook рассмотрим задачу по расчету временных зависимости компонент магнитного момента при различных значениях параметров φ_0 - перехода, на основе которых можно установить интервалы параметров, где происходит его переворот от $m_z = 1$ к $m_z = -1$.

Основные уравнения представлены в работе [Atanasova et al., 2019]. Ниже приведена задача Коши для системы уравнений в безразмерном виде.

Динамика магнитного момента M рассматриваемой системы описывается уравнением Ландау-Лифшица-Гильберта:

$$\begin{aligned}\frac{dm_x}{dt} &= -\frac{1}{1+M^2\alpha^2} \left\{ m_y H_z - m_z H_y + \alpha [m_x(M, H) - H_x] \right\}, \\ \frac{dm_y}{dt} &= -\frac{1}{1+M^2\alpha^2} \left\{ m_z H_x - m_x H_z + \alpha [m_y(M, H) - H_y] \right\}, \\ \frac{dm_z}{dt} &= -\frac{1}{1+M^2\alpha^2} \left\{ m_x H_y - m_y H_x + \alpha [m_z(M, H) - H_z] \right\},\end{aligned}\tag{1}$$

где $M = [m_x, m_y, m_z]$ – компоненты магнитного момента, компоненты эффективного поля $H = [H_x, H_y, H_z]$ зависят от джозефсоновской разности фаз φ определяются следующим образом:

$$\begin{aligned} H_x(t) &= 0, \\ H_y(t) &= Gr \sin(\varphi(t) - rm_y(t)), \\ H_z(t) &= m_z(t), \end{aligned}$$

где y компонента зависит от джозефсоновской разности фаз. Уравнение на джозефсоновскую разность фаз φ определяется в рамках резистивной модели из выражения для электрического тока I через джозефсоновский переход, измеренный в единицах критического тока I_c :

$$I = \omega \left(\frac{d\varphi}{dt} - r \frac{dm_y}{dt} \right) + \sin(\varphi - rm_y),$$

а именно:

$$\frac{d\varphi}{dt} = -\frac{1}{\omega} \left(\sin(\varphi - rm_y) + r \frac{dm_y}{dt} \right) + \frac{1}{\omega} I, \quad (2)$$

со следующими параметрами модели: G – отношение энергии Джозефсона к энергии магнитной анизотропии, r – константа спин-орбитального взаимодействия, α – диссипация Гилберта, $\omega = 1$ – отношение частоты ферромагнитного резонанса на характеристической частоты джозефсоновского перехода. В наших расчетах полагается равной 1.

Начальные условия предполагают, что все компоненты магнитного момента, кроме m_z , равны нулю:

$$m_x(0) = 0, \quad m_y(0) = 0, \quad m_z(0) = 1, \quad \varphi(0) = 0. \quad (3)$$

В рассматриваемой задаче исследуется возможность переворота магнитного момента m_z под действием прямоугольного импульса тока $I = I(t)$, входящего в уравнение (2):

$$I = I(t) = \begin{cases} 0, & t \in [0, T_f - \Delta t/2], \\ A_s, & t \in [T_s - \Delta t/2, T_s + \Delta t/2], \\ 0, & t \in [T_s + \Delta t, T_t], \end{cases} \quad (4)$$

где A_s – амплитуда импульса, Δt – его длительность, T_s – момент времени начала его локализации, а T_f – время расчета, которое выбирается достаточным для большим для выхода в устойчивый режим.

Реализация вычислительной схемы

Для исследования возможности переворота магнитного в зависимости от параметров модели необходимо численно решить задачу Коши для системы нелинейных уравнений (1-2) с начальными условиями (3) при заданной временной зависимости тока (4).

Численное решение начальной задачи проводилось с использованием функции `scipy.integrate.solve_ivp` библиотеки SciPy, в которой есть возможность выбора метода решения: для нежестких задач доступны явные методы Рунге-Кутты “RK45” или “RK23” порядков 5 (4) и 3 (2) соответственно и методы решения жестких задач “Radau” – неявный метод Рунге-Кутты семейства Радо IIА 5-го порядка, неявный многошаговый метод переменного порядка “BDF” и др. [SciPy. Integration and ODEs]. Приведенные в статье результаты получены с использованием методов “RK45” и “BDF”.

Численное решение задачи Коши проводилось для различных значений параметров модели. На Рис. 1 приведена динамика переворота компоненты магнитного момента m_z при двух значениях параметра отношение энергии Джозефсона к энергии магнитной анизотропии G : при $G=9$ (линия 3) наблюдается переворот магнитного момента m_z , при $G=8$ (линия 2) переворот отсутствует. На Рис. 1 также представлен график приложенного импульсного тока (линия 1). Расчеты проведены при значениях параметров $r=0.1$, $\alpha=0.1$.

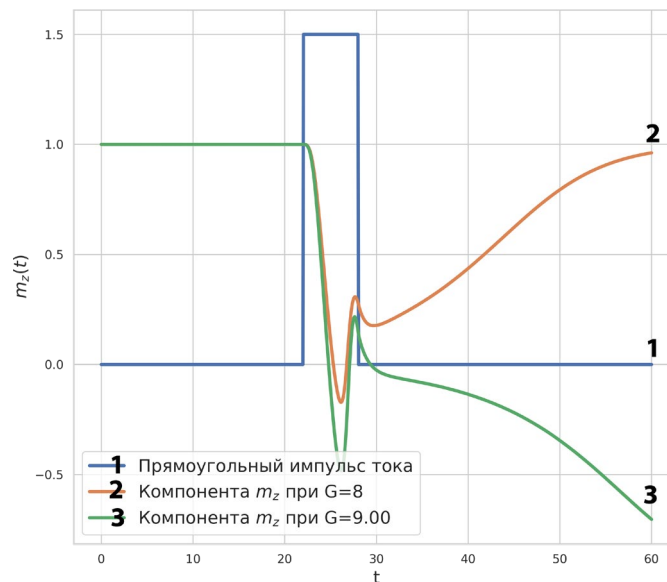


Рис. 1. Динамика переворота магнитного момента m_z при двух значениях параметра отношение энергии Джозефсона к энергии магнитной анизотропии G и график импульсного тока

Исследование показало, что переворот магнитного момента носит периодический характер, что можно показать, проводя расчеты с малыми шагами в плоскости параметров, например, (G, α) , что полностью согласуется с результатами, полученными в работе [Atanasova et al., 2019].

На Рис. 2 приведен график периодичности переворота магнитного момента m_z с увеличением значения параметра G в плоскости (G, α) . Результаты получены для значений параметров: $\alpha = 0.01 \dots 0.11$ с шагом 0.001 и $G = 1 \dots 101$ с шагом 1.

Отметим, что проведение многочисленных расчетов для различных параметров требует большого количества времени, например, для получения результатов, представленных на Рис. 2 необходимо решить 100^2 однотипных задач, а результат (положительное или отрицательное значение компоненты m_z при времени расчета T_f) записать в финальную матрицу значений. Однотипность решаемых задач и позволяет провести распараллеливание по данным, распределяя решения задач Коши при разных значениях параметров по разным процессам или потоком. Одна из возможных параллельных реализаций представлена ниже.

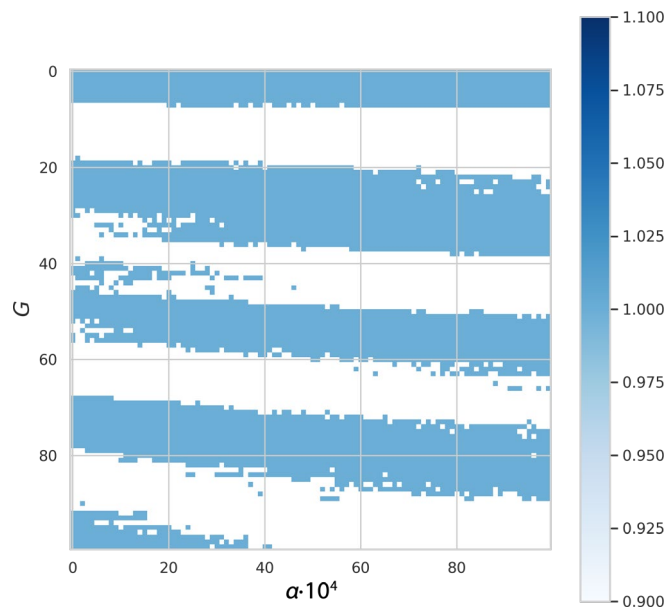


Рис. 2. График периодичности переворота магнитного момента m_z в плоскости параметров (G, α)

Параллельная реализация алгоритма

Для ускорения вычислений в языке Python существуют как отдельные библиотеки, позволяющие проводить расчеты на центральных процессорах и графических ускорителях: multiprocessing [Multiprocessing...], Joblib [Joblib...], mpi4py [MPI for Python], – так и в популярные библиотеки и фреймворки реализована возможность параллельных вычислений: TensorFlow [Tensorflow...], Keras [Keras...].

В данной работе параллельная реализация была сделана с применением библиотеки Joblib. Схема распараллеливания представлена на Рис. 3.

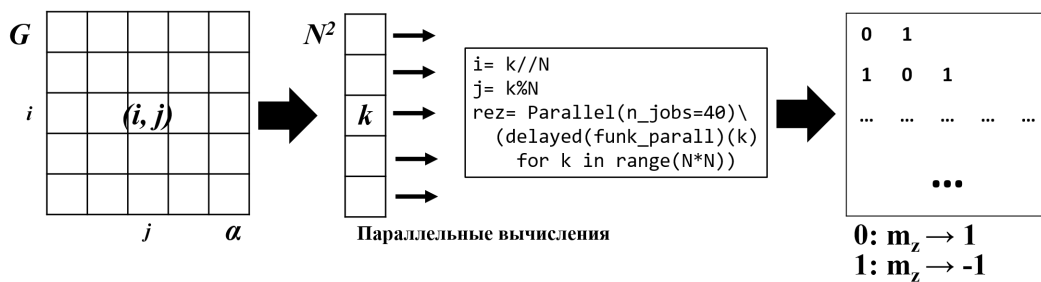


Рис. 3. Схема распараллеливания задачи с применением библиотеки Joblib

В процессе распараллеливания вычислительная область размерностью $N \times N = 100 \times 100$ распределялась между вычислительными потоками, значение которых задается параметром n_jobs . Для распараллеливания применяется класс Parallel библиотеки Joblib (см. Рис. 3).

Расчеты разработанного алгоритма проводились на двух компонентах экосистемы ML/DL/HPC, представляющие собой реальные вычислительные сервера. В процессе расчетов все полученные времена расчетов усреднялись по трем запускам.

На Рис. 4 – 5 приведены графики зависимости времени вычислений и ускорения от числа потоков на компоненте jhub2.jinr.ru. На графиках приведены значения до 50 потоков, в связи с тем, что при их дальнейшем увеличении время вычислений значительно не меняется. Из графиков видно, что в процессе расчетов было достигнуто ускорение в 21 раз на 42 потоках.

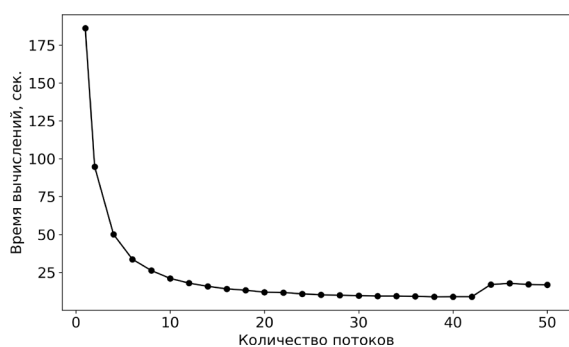


Рис. 4. График зависимости времени вычислений задачи от количества потоков на jhub2.jinr.ru

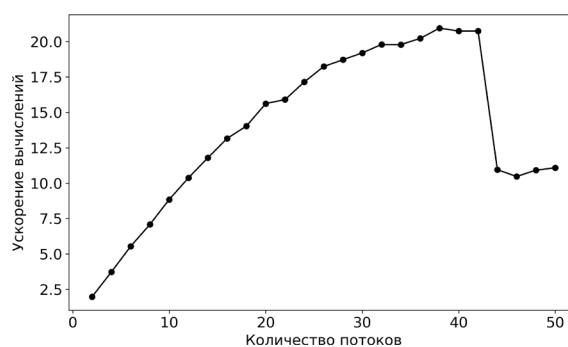


Рис. 5. График зависимости ускорения вычисления задачи от количества потоков на jhub2.jinr.ru

На Рис. 6 – 7 приведены графики зависимости времени вычислений и ускорения от числа потоков на компоненте studhub.jinr.ru. На графиках приведены значения до 60 потоков, в связи с тем, что при их дальнейшем увеличении время вычислений значительно не меняется. Из графиков видно, что в процессе расчетов было достигнуто ускорение в 22 раз на 50 потоках.

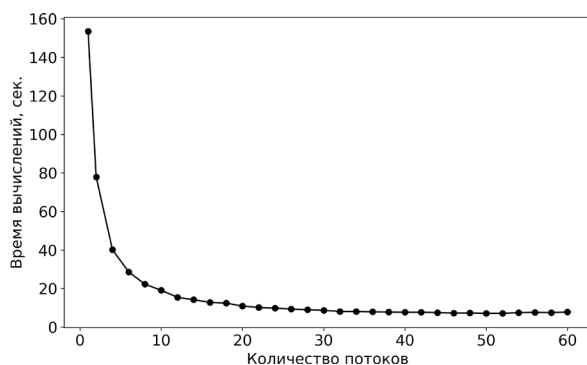


Рис. 6. График зависимости времени вычислений задачи от количества потоков на studhub.jinr.ru

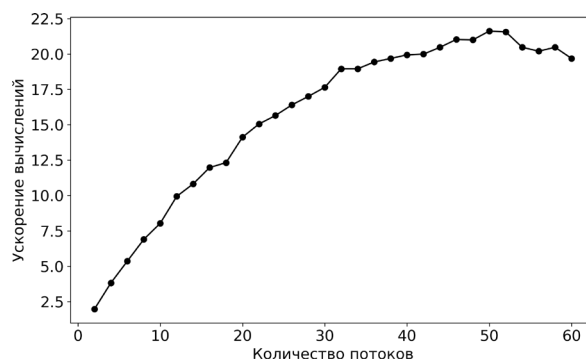


Рис. 7. График зависимости ускорения вычисления задачи от количества потоков на studhub.jinr.ru

Заключение

Использование языка Python и библиотеки Joblib, для распараллеливания вычислений, упростило процесс написания программы для решения задачи исследования возможности переворота магнитного момента в структурах сверхпроводник-ферромагнетик-сверхпроводник. На обеих компонентах экосистемы ML/DL/HPC были получены хорошие показатели ускорения расчетов в более чем 20 раз. Готовый Jupyter-ноутбук выложен в свободный доступ на ресурсах гетерогенной платформы HybriLIT и доступен по ссылке: <http://hlit.jinr.ru/spintronics>. Он будет использоваться в процессе проведения лабораторных и исследовательских работ как студентов, так и научных сотрудников института.

Список литературы (References)

Виртуальная исследовательская среда для моделирования физических процессов в гибридных наноструктурах состоящих из сверхпроводников и магнетиков [Электронный ресурс]: <http://sconduct.jinr.ru> (Дата обращения: 01.08.2022)

Virtual'naya issledovatel'skaya sreda dlya modelirovaniya fizicheskikh processov v gibridnykh nanostrukturah sostoyashchih iz sverhprovodnikov i magnetikov [Electronic resource]: <http://sconduct.jinr.ru> (accessed: 01.08.2022)

- Гетерогенная платформа HybriLIT [Электронный ресурс]: <http://hlit.jinr.ru> (Дата обращения: 01.08.2022)
Heterogeneous platform HybriLIT [Electronic resource]: <http://hlit.jinr.ru> (accessed: 01.08.2022)
- MATLAB в среде JupyterHub [Электронный ресурс]: <http://hlit.jinr.ru/matlab-in-jupyterhub-ru> (Дата обращения: 01.08.2022)
MATLAB v srede JupyterHub [Electronic resource]: <http://hlit.jinr.ru/matlab-in-jupyterhub-ru/> (accessed: 01.08.2022)
- МИВК ОИЯИ [Электронный ресурс]: <https://micc.jinr.ru> (Дата обращения: 01.08.2022)
JINR MICC [Electronic resource]: <https://micc.jinr.ru> (accessed: 01.08.2022)
- Облачная инфраструктура ОИЯИ [Электронный ресурс]: <https://micc.jinr.ru/?id=19> (Дата обращения: 01.08.2022)
JINR Cloud Infrastructure [Electronic resource]: <https://micc.jinr.ru/?id=19> (accessed: 01.08.2022)
- Суперкомпьютер «Говорун» [Электронный ресурс]: http://hlit.jinr.ru/supercomputer_govorun (Дата обращения: 01.08.2022)
“Govorun” supercomputer [Electronic resource]: http://hlit.jinr.ru/en/supercomputer_govorun (accessed: 01.08.2022)
- Учебно-тестовый полигон [Электронный ресурс]: http://hlit.jinr.ru/education_and_testing_polygon (Дата обращения: 01.08.2022)
Education and testing polygon [Electronic resource]: http://hlit.jinr.ru/education_and_testing_polygon (accessed: 01.08.2022)
- Экосистема ML/DL/HPC [Электронный ресурс]: http://hlit.jinr.ru/access-to-resources/ecosystem-for-ml_dl_bigdataanalysis-tasks (Дата обращения: 01.08.2022)
ML/DL/HPC ecosystem [Electronic resource]: http://hlit.jinr.ru/en/access-to-resources_eng/ecosystem-for-ml_dl_bigdataanalysis-tasks (accessed: 01.08.2022)
- Anaconda [Electronic resource]: <https://www.anaconda.com> (accessed: 01.08.2022)
- Atanasova P. Kh., Panayotova S. A., Rahmonov I. R., Shukrinov Yu. M., Zemlyanaya E. V. & Bashashin M. V.* Periodicity in the Appearance of Intervals of the Reversal of the Magnetic Moment of a $\phi 0$ Josephson Junction // JETP Letters. — 2019. — Vol. 110, No. 11. — С. 722–726.
- JINR cloud service for scientific and engineering computations [Electronic resource]: <https://joblib.readthedocs.io> (accessed: 01.08.2022)
- Joblib: running Python functions as pipeline jobs [Electronic resource]: <https://saas.jinr.ru> (accessed: 01.08.2022)
- Jupyter Book [Electronic resource]: <https://jupyterbook.org> (accessed: 01.08.2022)
- Keras: the Python deep learning API [Electronic resource]: <https://keras.io> (accessed: 01.08.2022)
- Kong Q., Siau T., Bayen A.* Python Programming and Numerical Methods: A Guide for Engineers and Scientists [Electronic resource]: <https://pythonnumericalmethods.berkeley.edu> (accessed: 01.08.2022)
- Markdown [Electronic resource]: <https://daringfireball.net/projects/markdown> (accessed: 01.08.2022)
- MPI for Python [Electronic resource]: <https://mpi4py.readthedocs.io/en/stable> (accessed: 01.08.2022)
- Multiprocessing – Process-based parallelism – Python documentation [Electronic resource]: <https://docs.python.org/3/library/multiprocessing.html> (accessed: 01.08.2022)
- Project Jupyter [Electronic resource]: <https://jupyter.org> (accessed: 01.08.2022)
- SciPy. Integration and ODEs [Electronic resource]: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html (accessed: 01.08.2022)
- TensorFlow: Use a GPU [Electronic resource]: <https://www.tensorflow.org/guide/gpu> (accessed: 01.08.2022)