

Разработка программного инструментария для высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах

Работу выполнила: Налепова Елизавета Денисовна

Руководитель: Городничев Максим Александрович, ст. преп. каф. ПВ ФИТ НГУ

Актуальность

Нейрофизиологи собирают большие объемы данных, которые необходимо обрабатывать на высокопроизводительных вычислительных системах (ВВС).

Выполнение обработки данных научных исследований – сложный процесс:

- методов обработки много, на их выбор влияют разные факторы,
- данные могут передаваться между исследованиями,
- взаимодействие с ВВС и написание программ (в т.ч. параллельных) для обработки данных требует владения навыками системного программирования.

Таким образом, существует проблема организации вычислений прикладными специалистами.

Цель

Цель работы – разработать проект и прототип программного инструментария для высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах.

Задачи

Для достижения цели были поставлены следующие задачи:

- 1) Изучить способы организации вычислений на суперкомпьютерах и имеющиеся для этого программные средства;
- 2) Сформулировать требования к программному инструментарию;
- 3) Разработать архитектуру программного инструментария для высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах;
- 4) Реализовать прототип предметно-ориентированной среды для решения задач обработки нейрофизиологических данных;
- 5) Провести испытания на реальной задаче.

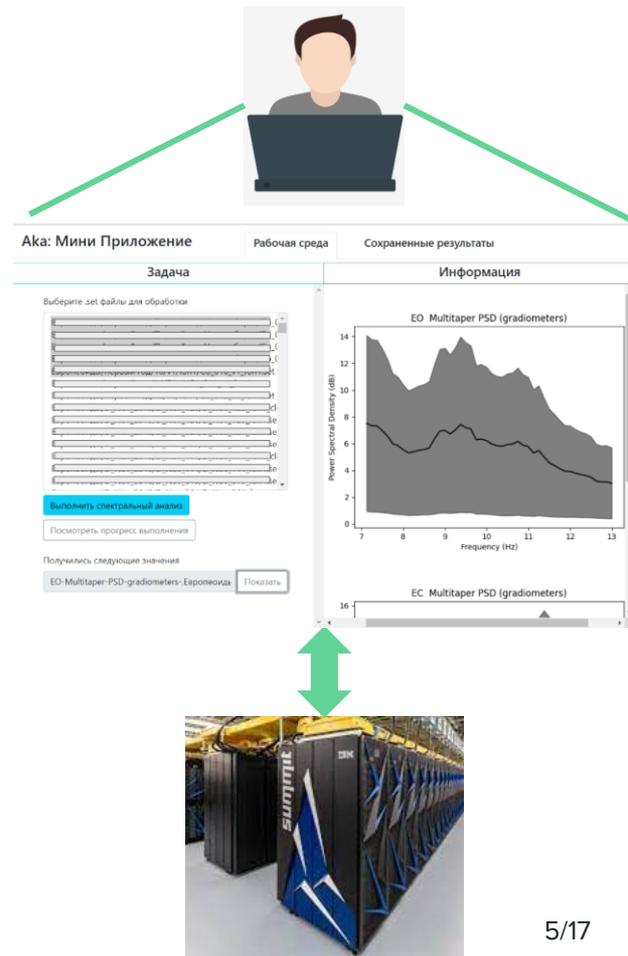
Требования

Разрабатываемый программный инструментарий должен:

- обеспечить взаимодействие конечного пользователя с ВВС в терминах его предметной области (нейрофизиология),
- обеспечить эффективное решение задач на ВВС,
- управлять данными,
- предполагать легкую расширяемость функциональности: должен быть обеспечен механизм добавления средств решения новых задач.

Высокоуровневая обработка исследовательских данных на суперкомпьютерах

- Прикладному специалисту функциональность инструментария предоставляется в виде *мини-приложений*.
- **Мини-приложение** – это совокупность пользовательского интерфейса для задания параметров конкретной задачи и визуализации результатов и программных компонентов, которые обеспечивают выполнение расчетов на ВВС.
- Предусматривается **метод и инструментарий для генерации мини-приложений** на основе базы знаний о сценариях расчетов, реализации вычислительных операций и форм ввода / представления данных.



Обоснование подхода к разработке мини-приложений

Инструментарий для генерации мини-приложений концептуально решает 3 проблемы:

1. Отсутствие удобного инструментария для работы прикладных специалистов без навыков системного программирования.

Решается: высоким уровнем абстракции.

1. Сложность конструирования интерфейсов под потребности конкретной группы исследователей.

Решается: выбором нужного сценария обработки данных из множества накопленных сценариев и генерацией интерфейса по спецификации.

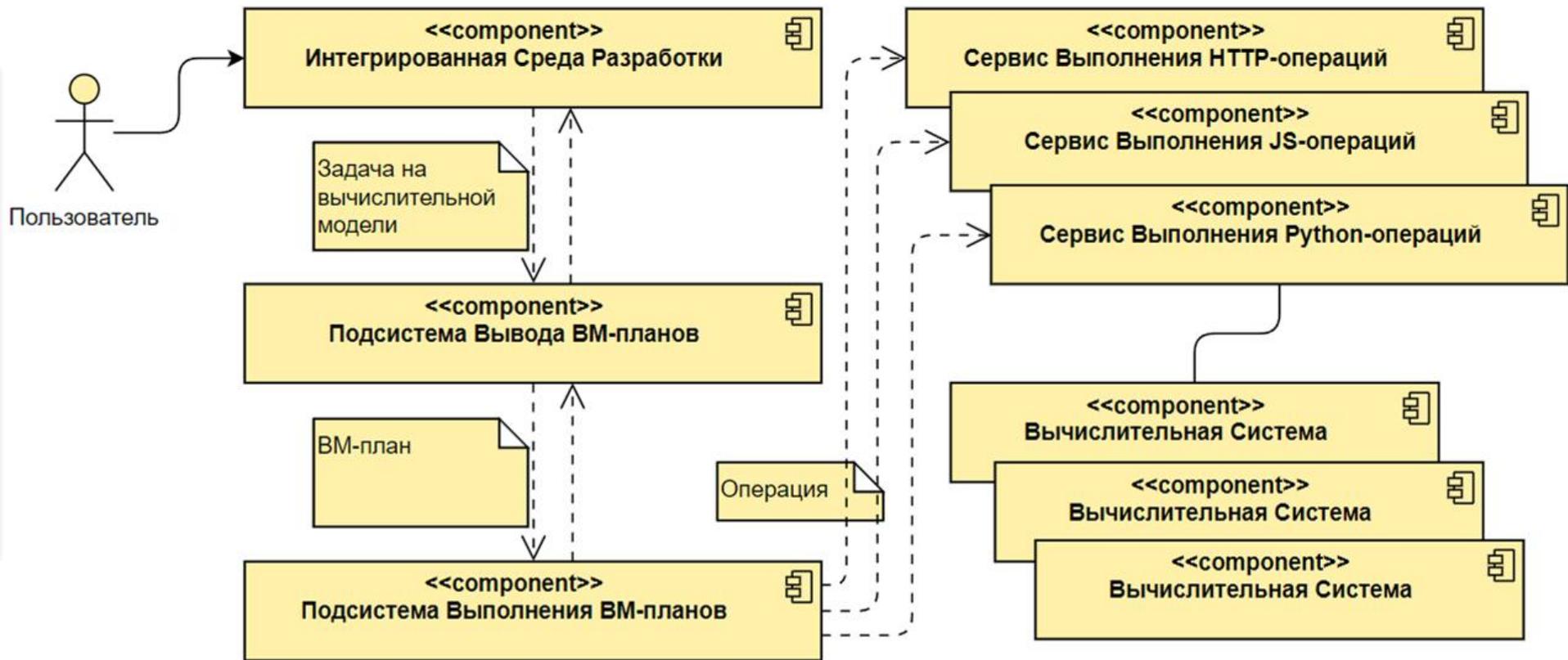
1. Отсутствие систематического накопления и переиспользования программных модулей, сценариев обработки данных.

Решается: созданием библиотеки программных модулей, реализующих операции VM, построением VM.

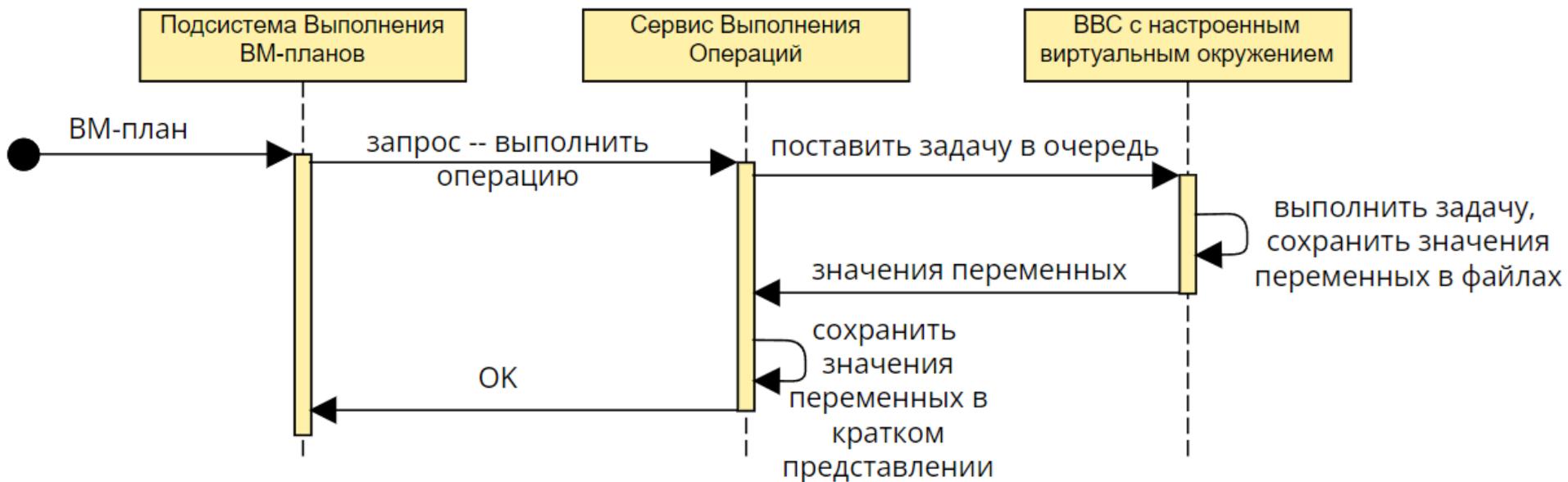
Формализация подхода к разработке мини-приложений

- Знания о возможных вариантах расчетов в предметной области формализуются **вычислительной моделью** (VM) – ориентированным двудольным графом без простых ориентированных циклов, у которого вершины соответствуют переменным и операциям предметной области.
- Требования к генерации мини-приложений формализуются **спецификацией задачи на VM** – заданными множествами входных и выходных переменных и нефункциональными требованиями к выводу **VM-плана**.
- **VM-план** – подграф вычислительной модели, соответствующий некоторому алгоритму вычисления выходных переменных по входным. Когда пользователь задает значения входных переменных и ставит задачу на расчет, происходит вывод VM-плана.
- Программные модули для реализации операций сценариев и формы ввода/вывода задаются при создании вычислительной модели и накапливаются в коллекции модулей.

Проект: архитектура инструментария



Проект: протокол взаимодействия компонентов системы



Реализация: программный интерфейс сервиса выполнения Python-операций

Разработан сервис в соответствии с архитектурным стилем REST с использованием фреймворка Django версии 3.2.12.

Функциональность сервиса:

1. выполнение операций на кластере ССКЦ,
2. сохранение значений переменных в собственном контексте,
3. предоставление значений переменных Ака для дальнейшего отображения пользователю,
4. освобождение контекста.

jobs

GET	/jobs/{job_id}/
GET	/jobs/{job_id}/vars/
POST	/jobs/{job_id}/vars/
DELETE	/jobs/{job_id}/vars/
GET	/jobs/{job_id}/vars/{var_name}/
DELETE	/jobs/{job_id}/vars/{var_name}/
POST	/jobs/
POST	/jobs/{job_id}/operations/

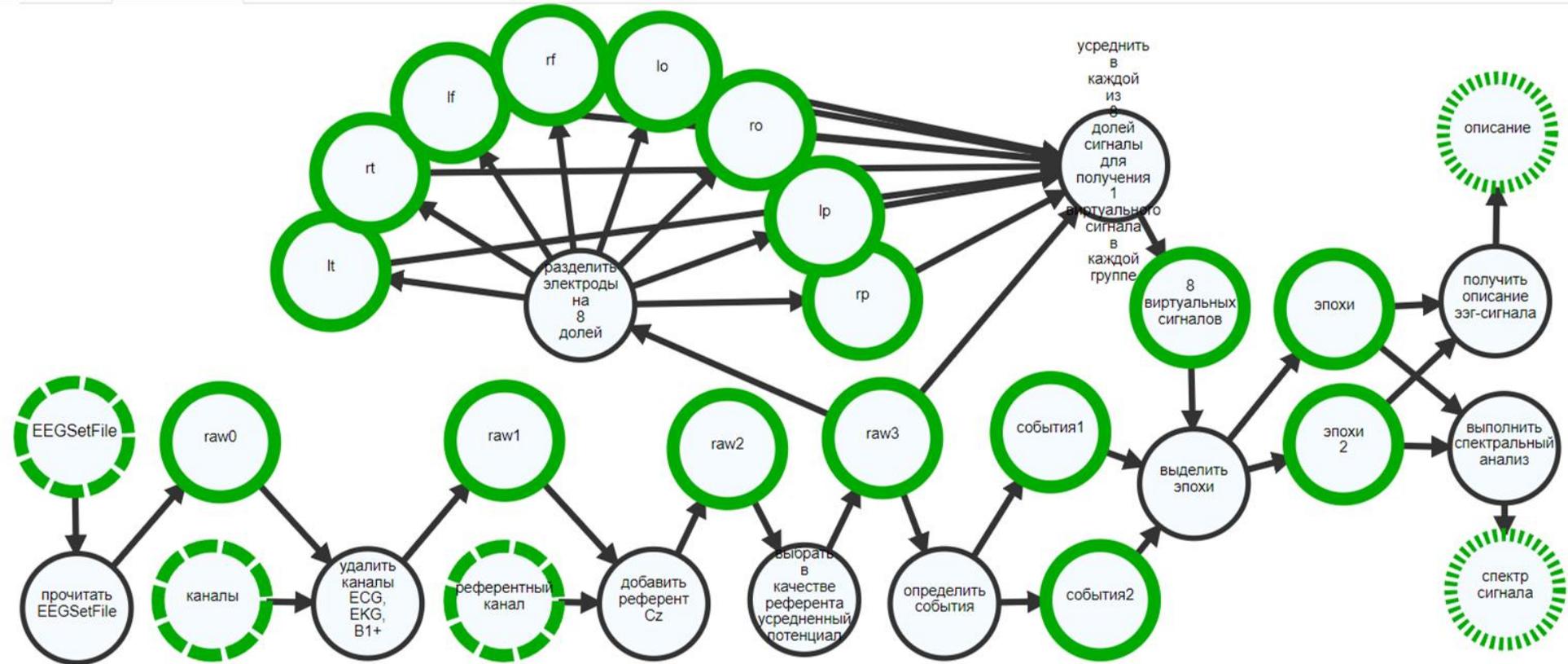
paths

GET	/paths/
-----	---------

images

GET	/images/{image_path}
-----	----------------------

Реализация: вычислительная модель



Реализация: графический пользовательский интерфейс [1]

Ака: Мини Приложение

Рабочая среда

Сохраненные результаты

Задача

Информация

Выберите .set файлы для обработки

Первый год/Co_016_v1_fon1_1.set
Первый год/Co_011_v1_fon1.set
Первый год/Co_012_v1_fon1.set
Первый год/Co_016_v1_fon1.set
первый год/16/v1/fon1/Co_016_v1_fon1.set
первый год/11/Fon1/Co_011_v1_fon1.set
первый год/12/v1/fon1/Co_012_v1_fon1.set
D_Nov_2013/D_Nov_126/D_Nov_126_fon1_clean.set
D_Nov_2013/D_Nov_126/D_Nov_126_fon1.set
D_Nov_2013/D_Nov_021/D_Nov_021_fon1.set
D_Nov_2013/D_Nov_056/D_Nov_056_fon1.set
D_Nov_2013/D_Nov_114/D_Nov_114_fon1_clean.set
D_Nov_2013/D_Nov_114/D_Nov_114_fon1.set
D_Nov_2013/D_Nov_019/D_Nov_019_fon1.set

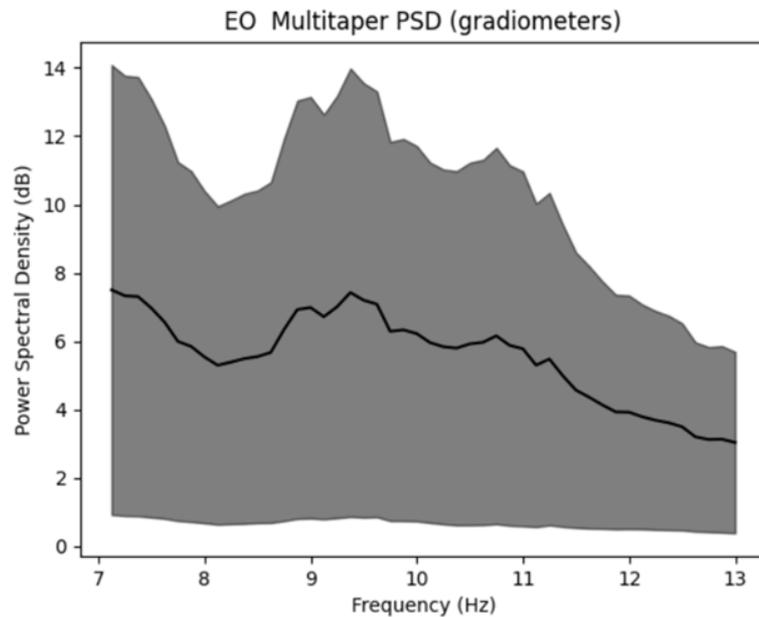
Выполнить спектральный анализ

Посмотреть прогресс выполнения

Получились следующие значения

EO-Multitaper-PSD-gradiometers-,Европеоиды,первый-год,Первый-год-1

Показать



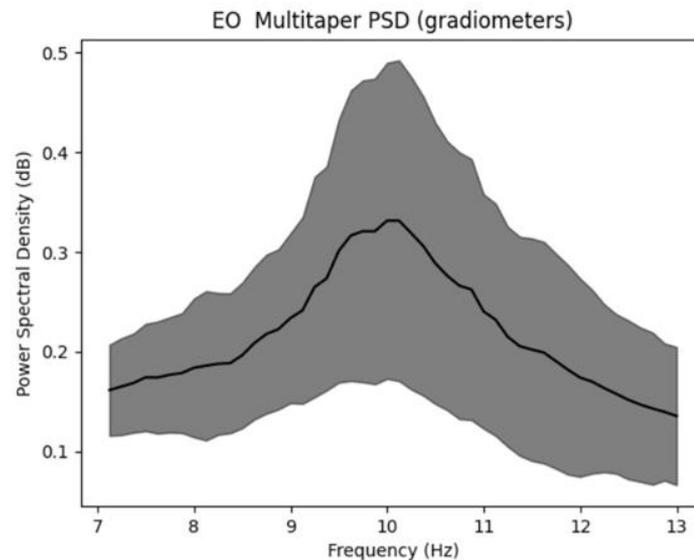
Реализация: графический пользовательский интерфейс [2]

Ака: Мини Приложение

Рабочая среда

Сохраненные результаты

Путь к файлу	Дата обработки
Первый год/Co_016_v1_fon1_1.set	Jun 16 23:38
Первый год/Co_011_v1_fon1.set	Jun 15 23:22
Первый год/Co_012_v1_fon1.set	Jun 15 18:14
Первый год/Co_016_v1_fon1.set	Jun 15 18:15
первый год/16/v1/fon1/Co_016_v1_fon1.set	Jun 15 18:17
первый год/11/Fon1/Co_011_v1_fon1.set	Jun 15 22:57
D_Nov_2013/D_Nov_126/D_Nov_126_fon1_clean.set	Jun 15 23:02
D_Nov_2013/D_Nov_126/D_Nov_126_fon1.set	Jun 15 18:43
D_Nov_2013/D_Nov_021/D_Nov_021_fon1.set	Jun 15 22:30
D_Nov_2013/D_Nov_056/D_Nov_056_fon1.set	Jun 15 22:33
D_Nov_2013/D_Nov_060/D_Nov_060_fon1.set	Jun 15 22:39
D_Nov_2013/D_Nov_041/D_Nov_041_fon1.set	Jun 15 22:41
D_Nov_2013/D_Nov_031/D_Nov_031_fon1.set	Jun 15 23:12



EC Multitaper PSD (gradiometers)



Результаты [1]

Разработан проект:

- концептуальная основа для обеспечения высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах – вычислительные модели;
- системная часть – система Ака (ИВМиМГ СО РАН);
- предложено расширение Ака системой сервисов для реализации вычислительных операций на высокопроизводительных вычислительных системах.

Результаты [2]

Разработан прототип предметно-ориентированной среды для решения задач обработки нейрофизиологических данных на суперкомпьютерах:

- реализован сервис выполнения Python-операций на BBC;
- программные модули, реализующие операции обработки данных, включены в коллекцию;
- разработана вычислительная модель для решения одной из задач обработки нейрофизиологических данных;
- реализован графический интерфейс для взаимодействия пользователя с задачей;
- приложение протестировано на задаче массовой обработки файлов.

Варианты развития программной части проекта

1. Реализация генерации UI по спецификации задачи на вычислительной модели.
2. Настройка взаимодействия с базой данных.
3. Развитие отказоустойчивости.
4. Добавление многопользовательского режима.
5. Улучшение UI/UX.
6. Реализация параллельного выполнения задач на кластере.
7. Добавление новых сценариев обработки данных в нейрофизиологии и других предметных областях.

Ссылки

Код сервиса:

<https://github.com/aeradcs/aka-service-api>

Код веб-приложения:

<https://github.com/aeradcs/aka-mini-app>

Руководство пользователя веб-приложения:

<https://disk.yandex.ru/i/IZo6-8rbKMwCjw>

Видео-знакомство-объяснение кода проектов:

- 1) <https://disk.yandex.ru/i/6sCnPwZpcCgk8A> – веб-приложение
- 2) https://disk.yandex.ru/i/aW6lcEp5_HYisg – сервис

Концептуальное обоснование идеи (ВКР):

<https://disk.yandex.ru/i/u9FmjRZEOo5FyA>

Контакты

<https://t.me/aeradcs>

или

e.nalepova@yandex.ru