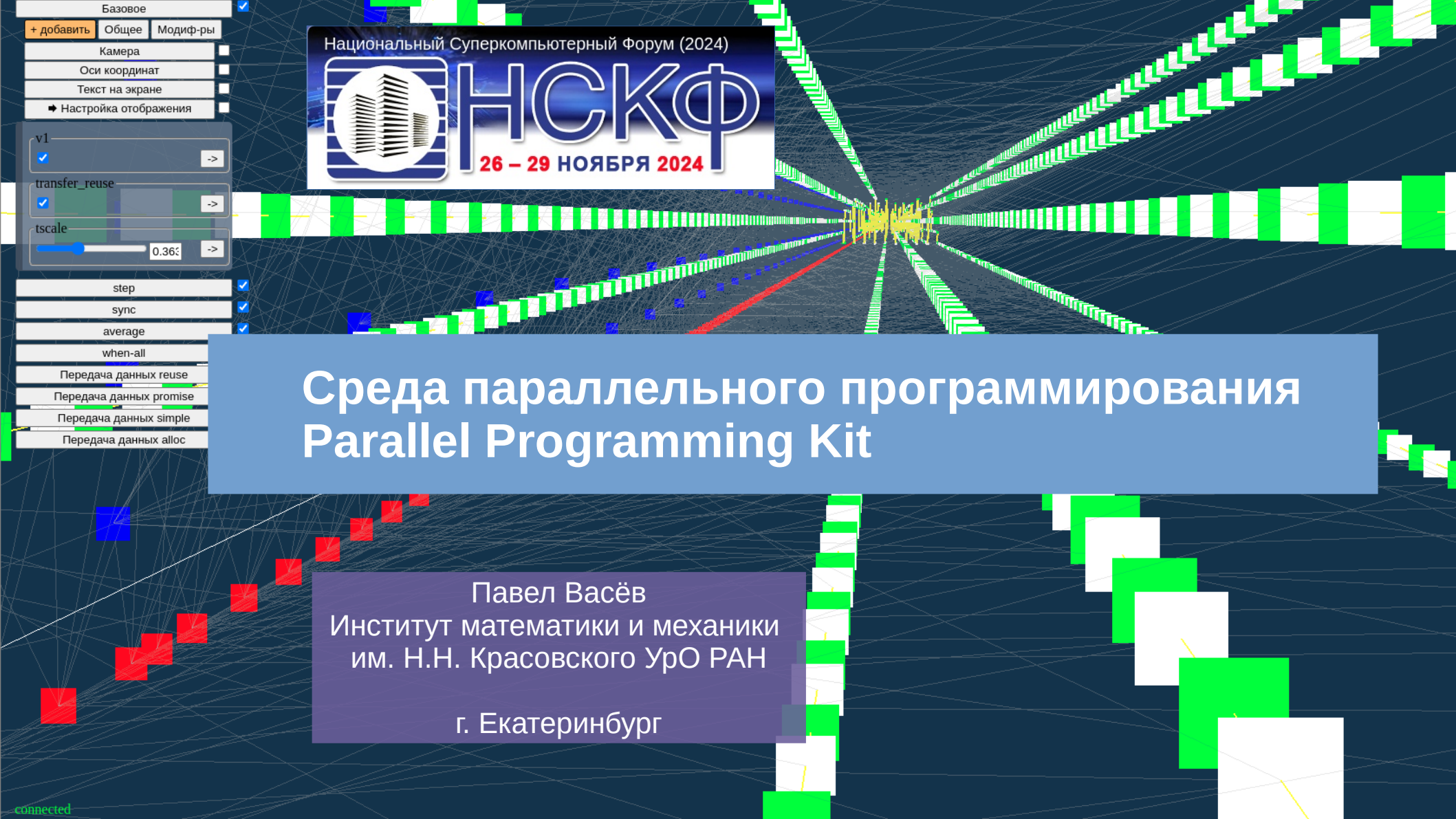


Национальный Суперкомпьютерный Форум (2024)

НСКФ

26 – 29 НОЯБРЯ 2024

Среда параллельного программирования Parallel Programming Kit

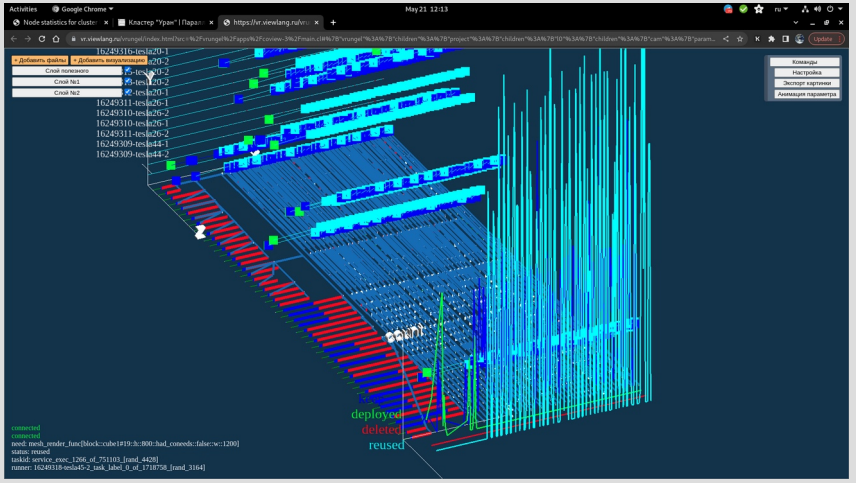
Павел Васёв
Институт математики и механики
им. Н.Н. Красовского УрО РАН

г. Екатеринбург

Тезисы

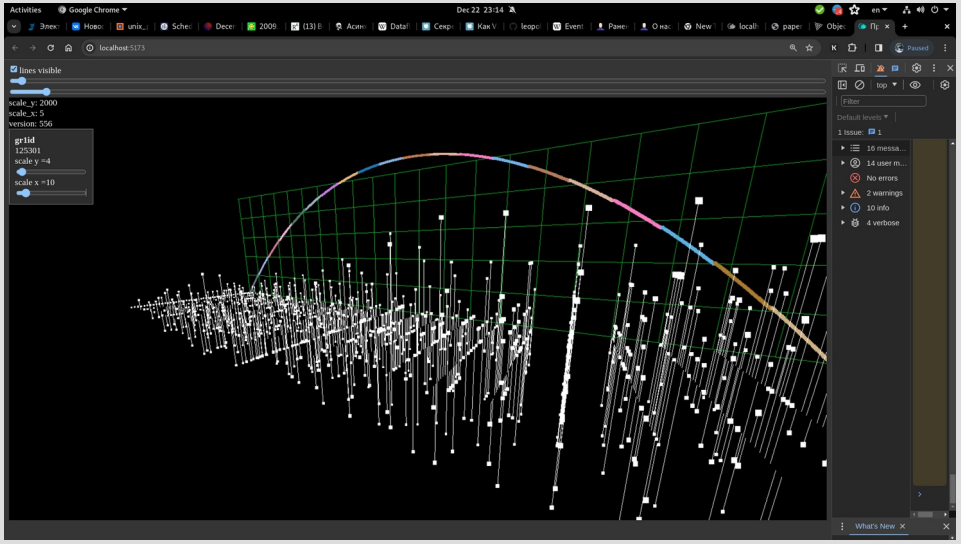
- Параллельное программирование — сложное.
- И более того, нам нужны интерактивные параллельные программы (онлайн/insitu визуализация). Параллельное вычисление как процесс! (а не функция)

- Но есть модели, которые упрощают ситуацию. Например простые:
  - map-reduce
  - мастер-рабочий
  - bulk syncronous parallel
- Есть и более интересные и богатые модели. Например:
  - суперподходы DVM, OpenMP, SYCL
  - поток данных (dataflow)
  - граф задач (task graph)



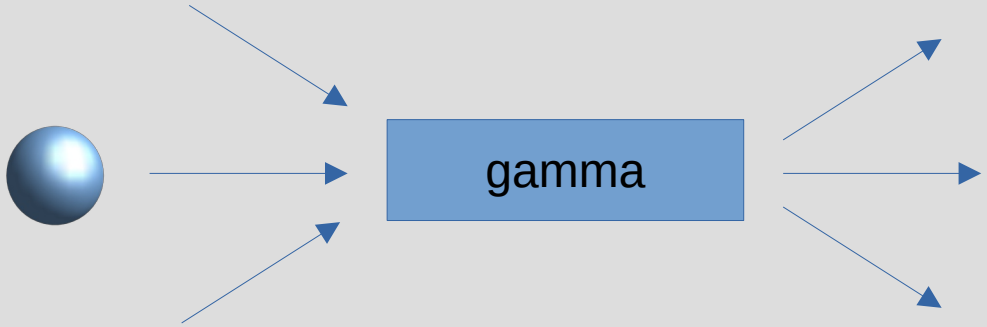
Что сделано

- Разработана модель коммуникации и программная среда, которая позволяет писать параллельные программы в разных моделях программирования (некоторых классов).



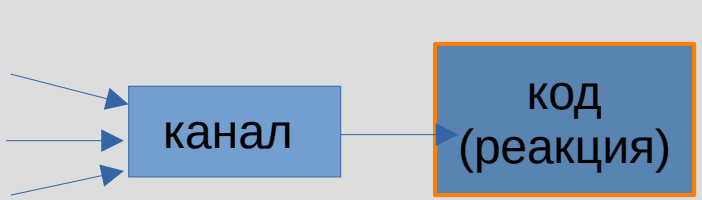
Модель. Сущности.

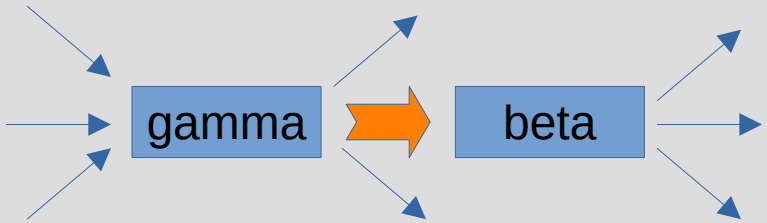
- **Канал** = метод коммуникации
- **Сообщение** = словарь + нагрузка
- **Окружение** = пространство имён каналов



Модель. Операции.

- Открыть канал: имя, чтение | запись → ch
- Отправить сообщение: ch, словарь + нагрузка → none
- Подписаться на сообщения в канале: ch, код → undo_fn
- Добавить связь между каналами: имя1, имя2 → undo_fn



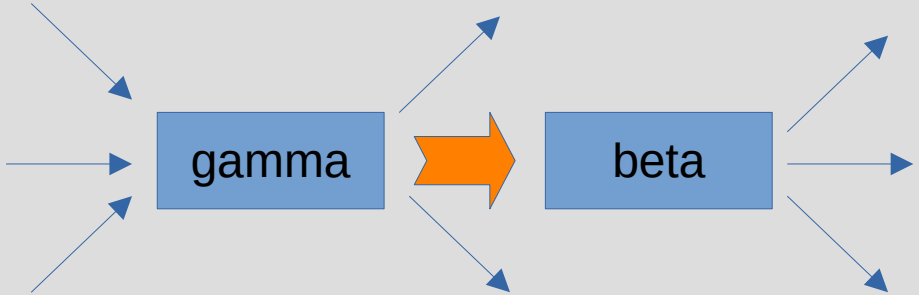


Особенности реализации

Прямая передача сообщений между отправителем и получателями (без посредников/брокеров, как например в Kafka и других). Посредством "мета-брокера".

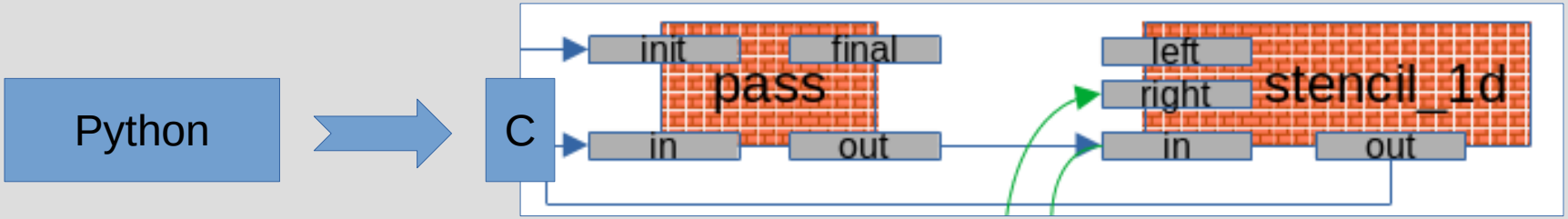
Передача сообщений по связям сразу на отправителе. Таким образом связи есть, но они могут и не привести к сетевой пересылке.

Передача ссылок вместо данных при локальной передаче.



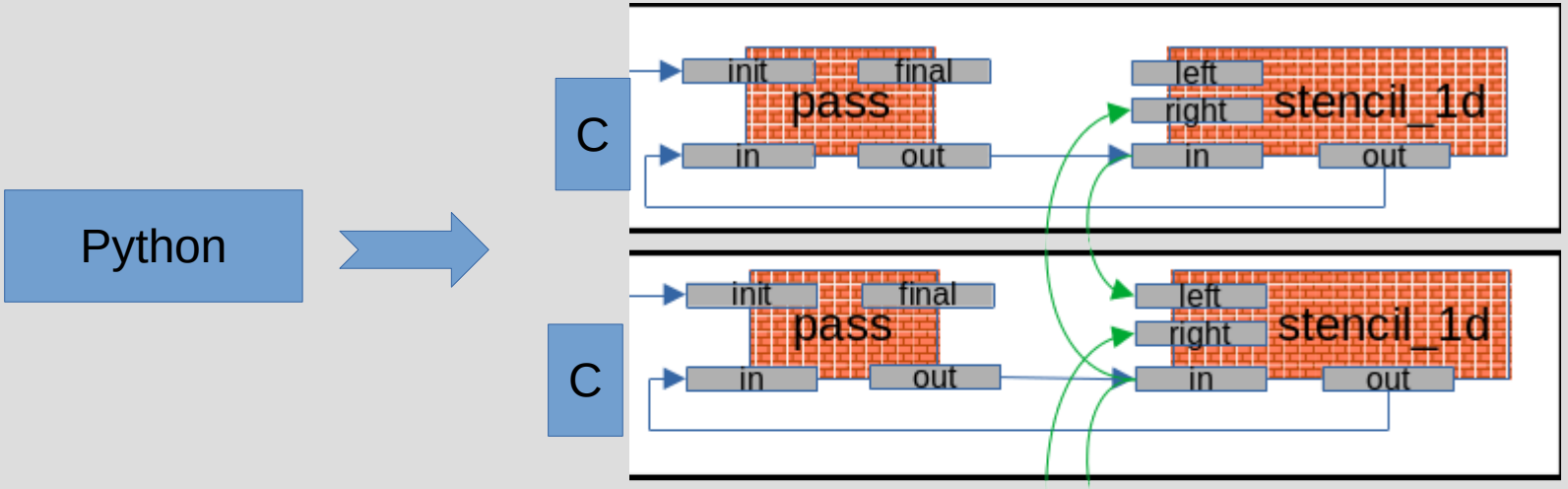
Интересные моменты

- Среда сетевая, не зависит от языков и т. п.
- Но при этом языки могут "взаимодействовать", если модель позволяет это.
- Например, можно порождать граф задач на Питоне, а фактически порождаемые узлы могут выполняться на C или на GPU.
- Это включает и взаимодействия по каналам / связям — связь созданная в Питоне приведет, например, к передаче данных внутри GPU.



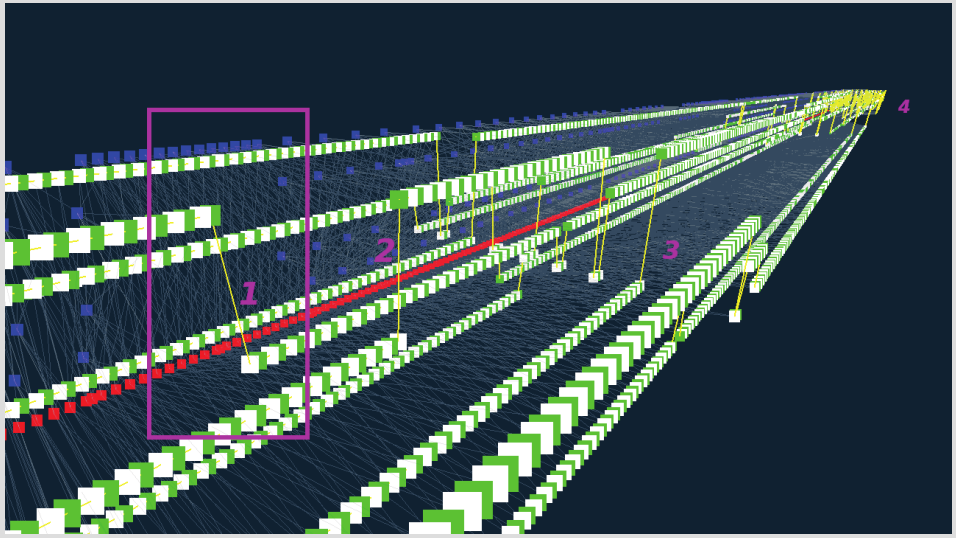
Интересные моменты

- Разработана программа "**исполнитель**", которая является однопоточным процессом уровня ОС. Она создает в своём адресном пространстве логические процессы (**акторы**) заданных разных типов, согласно управляющим сообщениям. Исполнитель может быть написан на любом языке, и подразумевается что акторы создаваемые им реализованы на языке исполнителя.
- Получается, что команды на создание акторов можно посылать хоть из Питона, хоть из Баш, а работать эти акторы будут уже в своей среде.



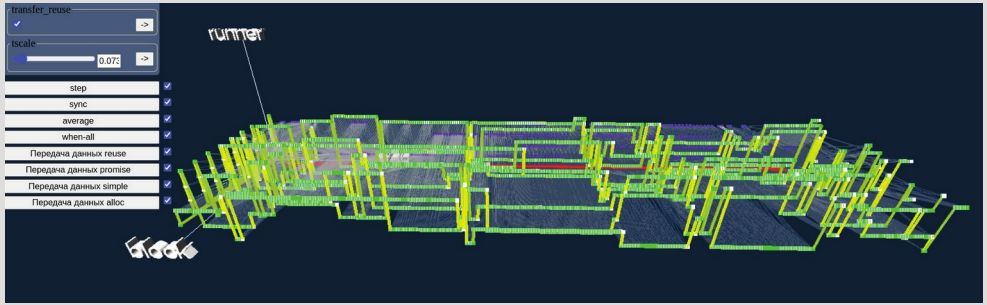
Опробованные модели

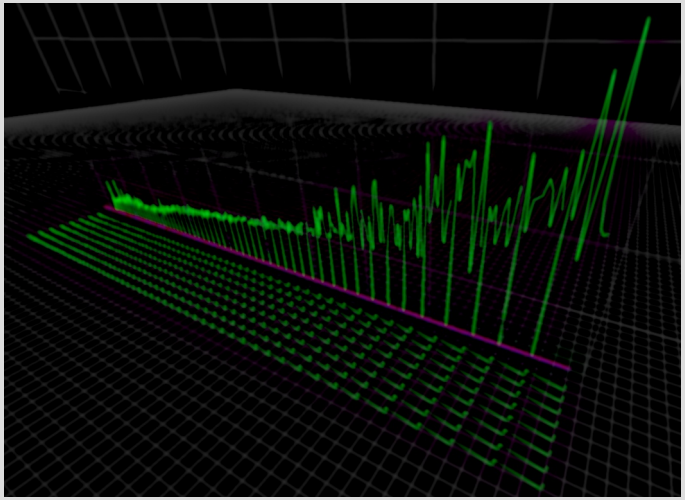
- Обмен сообщениями между логическими процессами (акторами)
- Граф задач
- Метод итераций для графов задач
- Метод схем



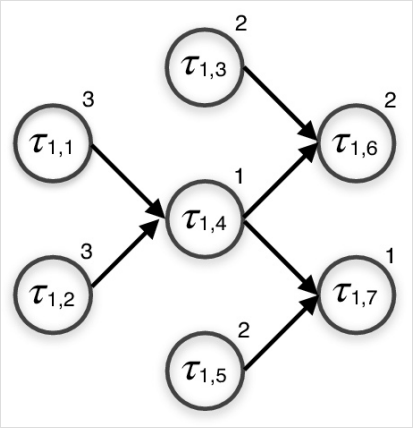
Полученные программы

- Интерактивная воксельная фильтрация (cpu) и визуализация (gpu) на кластере.
- Параллельная версия метода аппроксимации динамики нелокального уравнения неразрывности нелинейной марковской цепью.
- Параллельная версия обучения с подкреплением методом **PPO** пакета Stable-baselines3.
- Пользовательские приложения.





Граф задач

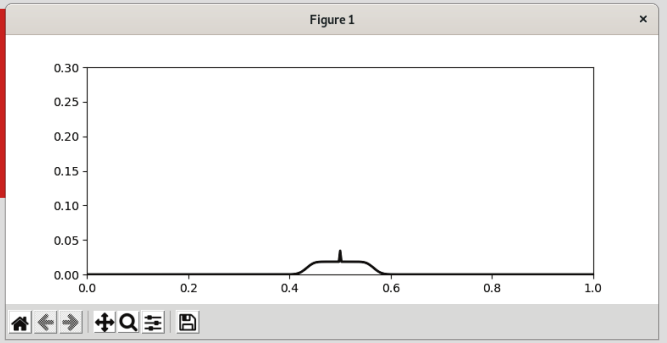


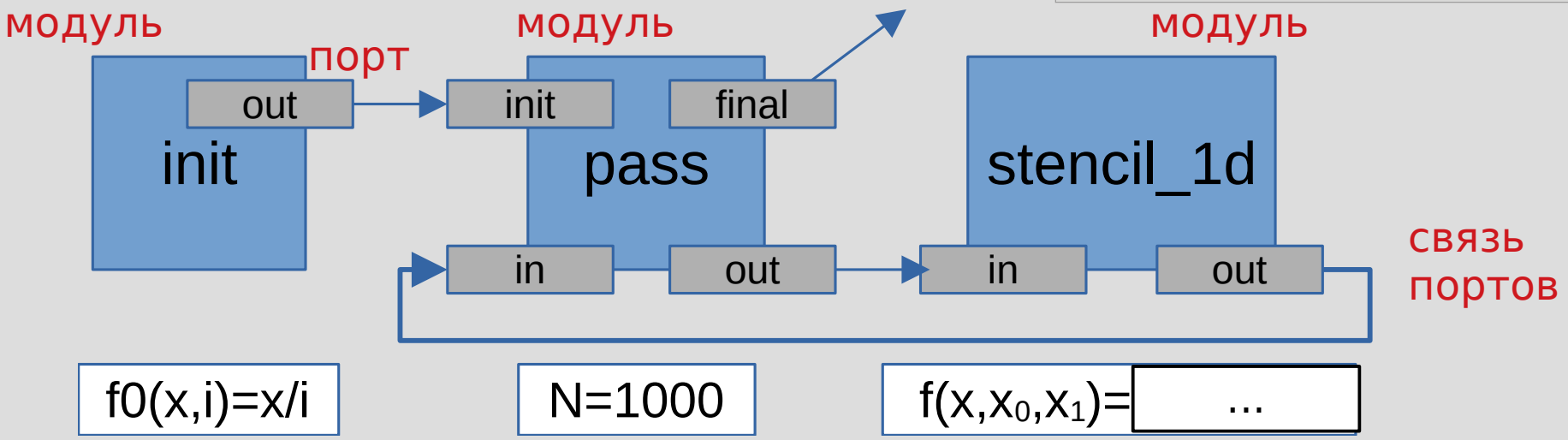
- Вершина = задача = разовое однопоточное вычисление
- Ребро = связь между задачами = выход одной на вход другим.
- Удобно: **обещания!** Генерируем граф **динамически**. Связываем задачи через обещания.

```
p = compute0()
q1 = compute1(p)
q2 = compute2(p)
print(q1,q2)
```

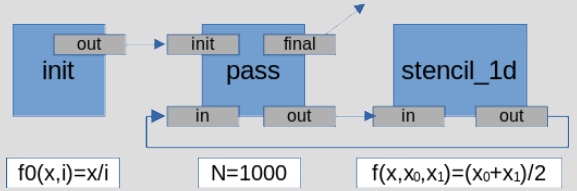
- **Плюсы**: параллельную программу можно описать последовательным алгоритмом, например, и это удобно. Автобалансировка. И другие.
- **Минусы**: накладные расходы. Модель не работает не мелкозернистых задачах.

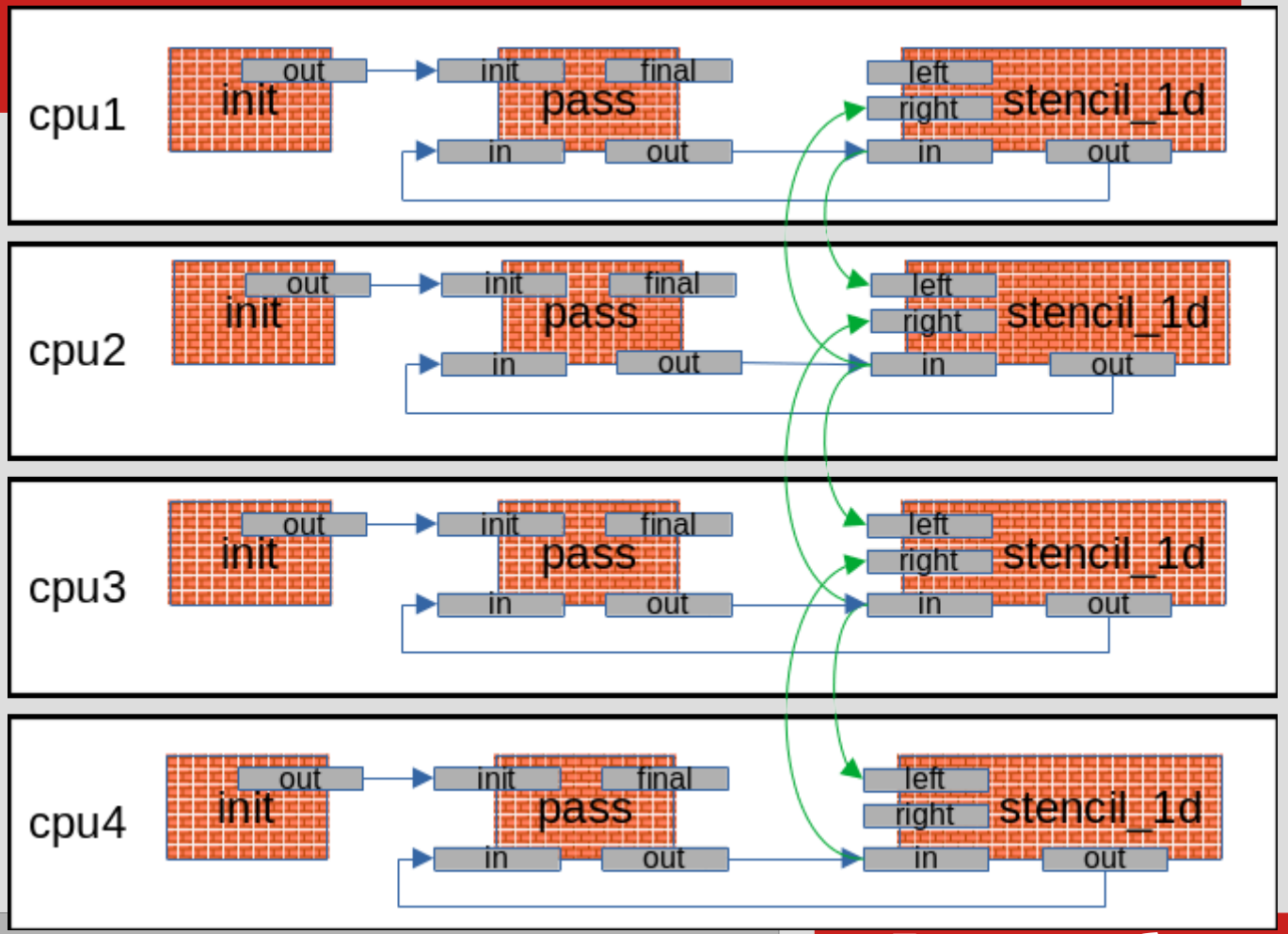
Метод схем. Схема программы



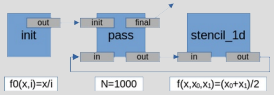


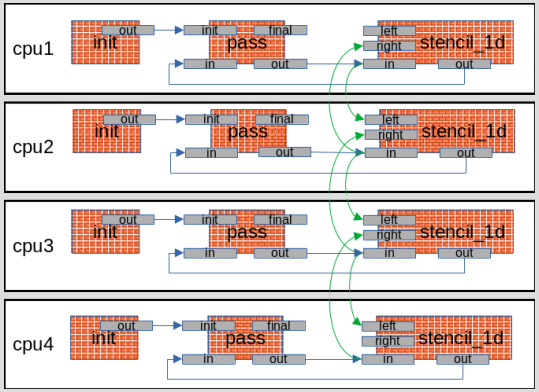
Порождённые процессы –

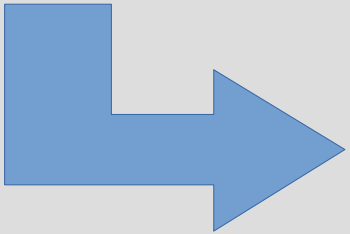


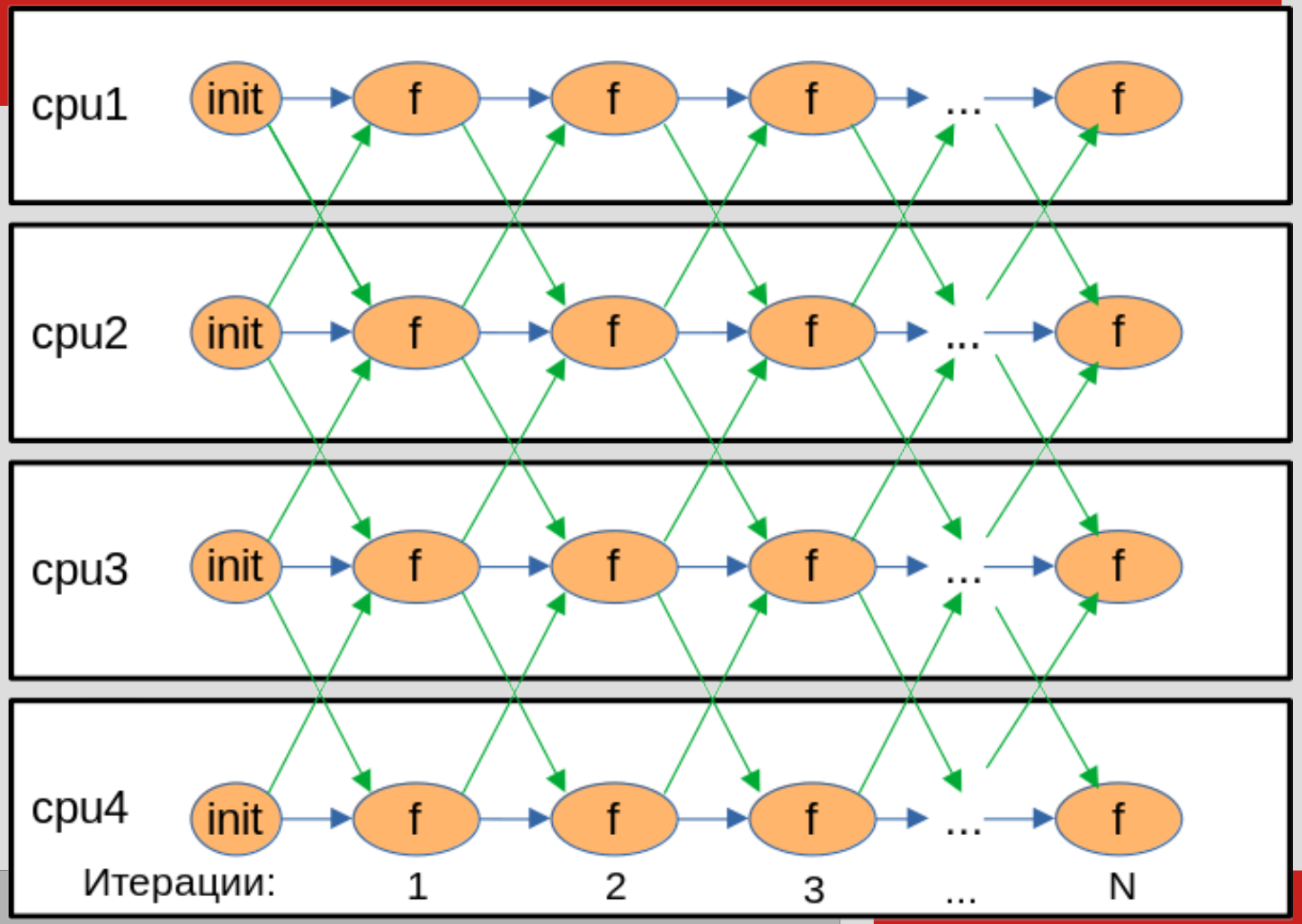


Граф задач –


init
pass
stencil_1d
f0(x,i)=x/i
N=1000
f(x,x0,x1)=(x0+x1)/2


cpu1
cpu2
cpu3
cpu4
init
pass
stencil_1d




cpu1
cpu2
cpu3
cpu4
init
f
...
Итерации:
1
2
3
...
N

ВЫВОДЫ

1. Параллельное программирование "напрямую" — сложное.

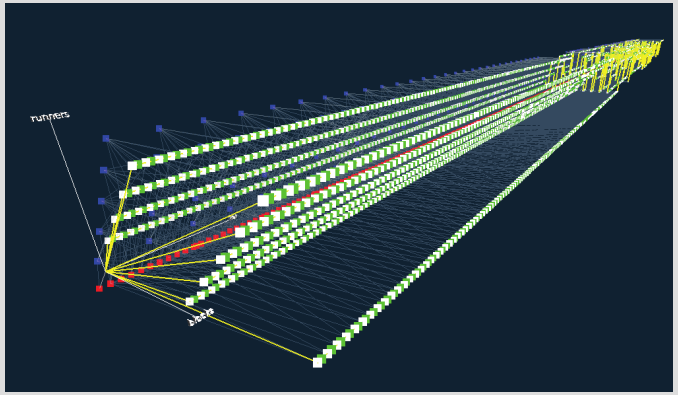
2. Но можно применять разные модели, которые упрощают такое программирование.

3. Разработана промежуточная модель и система, которая позволяет реализовывать разные модели некоторых классов.

ПЕРСПЕКТИВЫ, например:

1. Получить такую архитектуру, в которую можно было бы относительно легко добавлять разные улучшения. Например, предложить возможность оптимизации размещения логических процессов с учётом пересылок между ними.

2. Кодогенерация ИИ



Национальный Суперкомпьютерный Форум (2024)

НСКФ

26 – 29 НОЯБРЯ 2024

Среда параллельного программирования Parallel Programming Kit / Павел Васёв

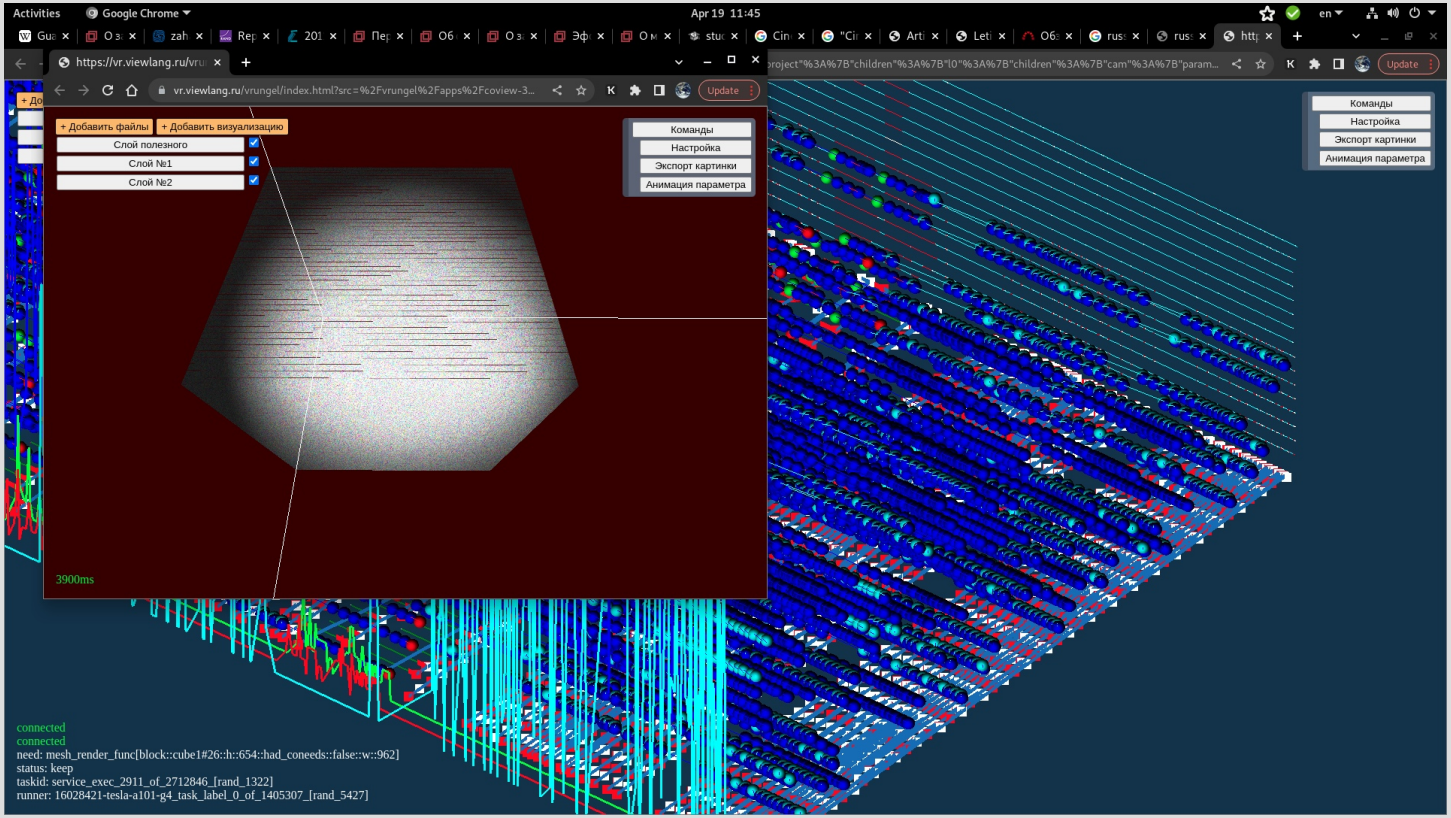
Благодарю за внимание!

github.com/pavelvasev/ppk

Документация на стадии оформления

Посоветуйте задачи!

vasev@imm.uran.ru



Справочно

Граф задач:
Pavel Vasev, A Computational Model for Interactive Visualization of High-Performance Computations // In: Voevodin, V., Sobolev, S., Yakobovskiy, M., Shagaliev, R. (eds) Supercomputing. RuSCDays 2023. Lecture Notes in Computer Science, vol 14389. Springer, Cham. https://doi.org/10.1007/978-3-031-49435-2_9

Метод итераций:
Васёв П.А., Метод итераций для графов задач // Параллельные вычислительные технологии – XVIII всероссийская конференция с международным участием, ПаВТ'2024, г. Челябинск, 2–4 апреля 2024 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2024. С. 182. ISBN 978-5-696-05450-6. DOI: https://doi.org/10.14529/pct2024

Метод схем:
П. А. Васёв,Один подход к онлайн-визуализации параллельных вычислений // Тезисы XIX Международной конференции СУПЕРВЫЧИСЛЕНИЯ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, 20 – 24 мая 2024 г., г. Саров, С. 100. URL: https://www.cv.imm.uran.ru/e/3241938