

## Метод онлайн-визуализации параллельных программ

П.А. Васёв

Институт математики и механики им. Н.Н. Красовского УрО РАН, Екатеринбург

Онлайн-визуализация параллельных программ подразумевает: 1) возможность наблюдения за ходом вычисления, и 2) возможность влияния на ход вычисления. Такая визуализация несет существенные выгоды [1], но сопряжена с определенными техническими сложностями:

а) в параллельном программировании до сих пор не предложено стандартных средств формирования визуальных образов для визуализации по ходу вычисления;

б) при больших вычислениях рендеринг визуальных образов необходимо осуществлять на узлах, где происходит вычисление [2], что обычно требует особых компетенций.

Осмысление пункта (а) приводит к наблюдению, что в то же время в параллельных программах доступен оператор `print`, который крайне удобен и успешно выводит информацию в лог-файл программы или файлы. Но лог-записи и файлы медленны, и по своей природе не визуальны. В то же время существует подход *CinemaScience* [3], который частично решает задачу онлайн-визуализации и использует `print`-подобный подход. Возникает закономерный вопрос, почему `print` удобен и доступен, и ему до сих пор не найдено достойных альтернатив?

В настоящей работе предлагается следующий подход. Пусть параллельная программа это клиент графической среды *Grafix* [4], по аналогии с клиентом среды *Gradio*, используемой при обучении нейросетей. Параллельная программа при запуске соединяется с заданным веб-сервером, который отвечает за визуализацию интерфейса программы. В ответ от веб-сервера программа получает веб-ссылку, по которой сервер будет готов отобразить визуализацию и управляющий интерфейс программы. Параллельная программа печатает эту ссылку в выходной поток, и далее производит вычисления как обычно.

При необходимости пользователь открывает поданную ему в потоке ссылку, и параллельная программа получает от веб-сервера сигнал *gui\_attached*. В ответ на этот сигнал программа формирует графический интерфейс (в формате json, см. [4]) и отправляет его веб-серверу. Далее программа наполняет интерфейс актуальными данными о своем состоянии.

Пользователь наблюдает эти данные, и при необходимости формирует управляющие воздействия, которые определяются графическим интерфейсом программы. Сигналы об этих воздействиях поступают в параллельную программу, и она имеет возможность реагировать на них согласно своему устройству.

Следует отметить, что среда *Grafix* интегрирована со средой *Parallel Programming Kit* [5], но не требует ее использования. Любая параллельная программа может зарегистрироваться по заданным протоколам и представить свое состояние и варианты взаимодействия в предлагаемом веб-интерфейсе.

### Литература

1. Vasev P. Analyzing an Ideas Used in Modern HPC Computation Steering // 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT), Yekaterinburg, Russia, 2020. P. 1–4. DOI: 10.1109/USBREIT48449.2020.9117685.
2. Moreland K. The tensions of in situ visualization // IEEE Computer Graphics & Applications, March/April 2016. P. 5–9. DOI: 10.1109/MCG.2016.35.
3. Ahrens J. et al. An image-based approach to extreme scale in situ visualization and analysis // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '14). IEEE, 2014. P. 424–434. DOI: 10.1109/SC.2014.40.
4. Васёв П.А. Сетевая технология программирования визуальных интерфейсов // GraphiCon 2024: Материалы 34-й Международной конференции по компьютерной графике и машинному зрению, Омск, 17-19 сентября 2024 года. Омск: Омский государственный технический университет, 2024. С. 354–360. DOI: 10.25206/978-5-8149-3873-2-2024-354-360.
5. Васёв П.А. Среда параллельного программирования *Parallel Programming Kit* // Тезисы докладов Национального суперкомпьютерного форума (НСКФ-2024), 26-29 ноября 2024, Институт программных систем им. А.К. Айламазяна РАН, Переславль-Залесский.