

Модель системы онлайн-визуализации

П. А. Васёв

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

Онлайн-визуализация (и её частный случай, *insitu*-визуализация) является актуальной задачей современных суперкомпьютерных вычислений [1]. В настоящей работе предлагается решение этой задачи в форме специальной модели параллельных вычислений (которая развивает идеи из [2]). Будем считать, что параллельная счётная программа использует доменную декомпозицию и выполняет вычисление, используя итерации – это типичный сценарий расчётов. Необходимо получать данные от такой программы по ходу счёта и преобразовывать их, включая и параллельный рендеринг, согласно заданному конвейеру визуализации.

Для реализации связи с параллельной программой и для программирования параллельных алгоритмов обработки распределённых данных предлагается использовать обещания (англ. *promises*). Обещание — это объект, который соответствует данным, которые будут вычислены *когда-нибудь*. Обещание можно создать в одном процессе, выполнить (то есть заполнить данными) в другом, а реагировать на выполнение — в-третьих. Обещания можно создавать явно или неявно. Один из удобных методов неявного создания и использования обещаний — это их увязка с асинхронным выполнением задач. Для этого модель предлагает операцию вычислить: код, параметры → обещание. На вход операции поступает *код* функции на некотором языке программирования и *параметры*, которые следует передать этой функции. В результате своего действия операция *вычислить* ставит *код* в очередь на параллельное исполнение, и немедленно возвращает объект *обещания*. Обещания можно использовать в параметрах других операций вычислить. В этом случае их вычисление запускается только когда все обещания, перечисленные в параметрах, окажутся выполнены (они заменяются на результаты вычислений кодов).

Удобство обещаний заключается в том, что оперировать ими можно в любой момент времени, ещё до того, как получены результаты вычислений. Это позволяет описывать параллельные алгоритмы обработки данных с помощью последовательных кодов, см. рис. 1.

Модель предполагает, что итерация цикла параллельной счётной программы формирует структуру из обещаний, соответствующую используемой доменной декомпозиции. Обещания из структуры выполняются счётными процессами по мере их завершения итерации. Этот же экземпляр структуры обещаний подаётся на вход алгоритму конвейера визуализации. Он формирует необходимую обработку данных в интересах визуализации аналогично примеру рис. 1.

```
1: let filenames = ["1.dat", "2.dat", ..., "50.dat"]
2: let blocks = filenames.map( x => load!(x) )
3: let images = blocks.map( b => render!(b, camera_position) )
4: let final_image = recursive_merge( images )
```

Рис. 1. Решается задача параллельного рендеринга регулярных сеток, см. [3]. Имена со знаком “!” – это обращения к операции вычислить, которая возвращает обещание, что запускает серию связанных процессов. Анимация работы алгоритма и пояснения доступны на странице youtu.be/XnV318hw8QE.

Литература

1. "Leveraging Production Visualization Tools In Situ." Kenneth Moreland, Andrew C. Bauer, Berk Geveci, Patrick O'Leary and Brad Whitlock. In *In Situ Visualization for Computational Science*. Springer, 2022. DOI 10.1007/978-3-030-81627-8_10.
2. М.О. Бахтерев, П.А. Васёв, А.Ю. Казанцев, И.А. Альбрехт, Методика распределенных вычислений RiDE // ПаВТ'2011: труды международной научной конференции (Москва, 28 марта – 1 апреля 2011 г.) – Челябинск: Издательский центр ЮУрГУ, 2011, с. 418–426.
3. Потехин, А. Л., Об одном способе организации структур данных в системах научной визуализации // Вопросы атомной науки и техники. Серия: Математическое моделирование физических процессов. – 2022. – № 4. – С. 64-71. – DOI: 10.53403/24140171_2022_4_64.