

Шаблоны инженерных и научных расчетов

А. А. Прохоров^{1,a}, А. М. Назаренко^{1,b}, А. В. Давыдов^{2,c}

¹ Институт проблем передачи информации РАН им. А.А. Харкевича,
125051, г. Москва, Б. Каретный пер., 19

² ООО «ДАТАДВАНС»,
117246, Москва, Научный проезд, д. 17

E-mail: ^a prokher@gmail.com, ^b nazar@phystech.edu, ^c andrey.davydov@datadvantage.net

Работу над современными инженерными и научными задачами невозможно представить без проведения вычислительных экспериментов и компьютерного моделирования. Зачастую решение одной задачи требует применения нескольких расчетных средств. С точки зрения автоматизации, такие вычислительные задачи являются композицией слабо связанных (в смысле интенсивности взаимодействия) типовых подзадач, решение каждой из которых производится специализированным расчетным средством. Для автоматизации такого рода расчетов уже несколько десятилетий разрабатывается и применяется специализированное программное обеспечение – интеграционные среды и интеграционные платформы.

Несмотря на многообразие программных решений, далеко не все они подходят для автоматизации расчетов, необходимых для решения актуальных задач. Это объясняется тем, что если ранее вычислительный эксперимент использовался в основном для проверки гипотез и контроля характеристик разрабатываемых изделий, то сегодня компьютерное моделирование все чаще используется для комплексного исследования пространства параметров, в частности, для автоматизированного поиска оптимальных конфигураций исследуемых расчетных моделей. Кроме того, задачи, стоящие сегодня перед специалистами, зачастую являются междисциплинарными, а проводимые расчеты, в свою очередь, характеризуются довольно сложными схемами взаимодействия расчетных средств.

С целью выявления требований к интеграционному программному обеспечению, предназначенному для автоматизации инженерных и научных расчетов, было проанализировано множество расчетных схем, в результате чего выявлены типовые вычислительные подзадачи и описаны соответствующие им расчетные схемы – шаблоны. В данной работе описывается семь базовых типов расчетов (шаблонов): последовательный, параллельный, альтернативный, пакетный, итеративный, итеративно-пакетный и составной. Описание каждого из шаблонов сопровождается характерным примером, указываются релевантные шаблоны потоков работ, затрагивается вопрос организации хранения данных, получаемых в процессе расчета.

Представленные шаблоны расчетных схем, могут быть использованы в качестве элементов декомпозиции сложных вычислительных задач. Кроме того, возможность реализации перечисленных шаблонов может рассматриваться в качестве критерия применимости интеграционного программного обеспечения к решению актуальных научных и инженерных задач.

Ключевые слова: автоматизация расчетов, интеграция расчетов, интеграционная платформа, поток работ, шаблоны потоков работ.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 15-29-07043. Условия для выполнения работ по проекту предоставлены ООО "ДАТАДВАНС".

© 2016 Александр Александрович Прохоров, Алексей Михайлович Назаренко, Андрей Владимирович Давыдов

1. Введение

Работу над современными инженерными и научными задачами невозможно представить без проведения вычислительных экспериментов и компьютерного моделирования. Зачастую решение одной задачи требует применения нескольких расчетных средств (РС). С точки зрения автоматизации, такие вычислительные задачи являются композицией слабо связанных (в смысле интенсивности взаимодействия) типовых подзадач, решение каждой из которых производится специализированным РС. Для автоматизации такого рода расчетов уже несколько десятилетий разрабатывается и применяется специализированное программное обеспечение – интеграционные среды и интеграционные платформы, в частности описанные в [Радченко, 2009; Deelman, Gannon, ..., 2009; Бурнаев, Губарев, ..., 2012; Knyazkov, Kovalchuk, ..., 2012; Sukhoroslov, Volkov, Afanasiev, 2015].

Несмотря на многообразие программных решений, далеко не все они подходят для автоматизации расчетов, необходимых для решения актуальных задач. Это объясняется тем, что если ранее вычислительный эксперимент использовался в основном для проверки гипотез и контроля характеристик разрабатываемых изделий, то сегодня компьютерное моделирование все чаще используется для комплексного исследования пространства параметров, в частности, для автоматизированного поиска оптимальных конфигураций исследуемых расчетных моделей [Соболь, Статников, 2006]. Кроме того, задачи, стоящие сегодня перед специалистами, зачастую являются междисциплинарными, а проводимые расчеты, в свою очередь, характеризуются довольно сложными схемами взаимодействия РС [Kodiyalam, Sobieszczanski-Sobieski, 2001; Martins, Lambe, 2013].

С целью выявления требований к интеграционному программному обеспечению, предназначенному для автоматизации инженерных и научных расчетов, было проанализировано множество расчетных схем, в результате чего выявлены типовые вычислительные подзадачи и описаны соответствующие им расчетные схемы – шаблоны. В данной работе описывается семь базовых типов расчетов (шаблонов): последовательный, параллельный, альтернативный, пакетный, итеративный, итеративно-пакетный и составной. Описание каждого из шаблонов сопровождается характерным примером, указываются релевантные шаблоны потоков работ [van Der Aalst, Wil, ..., 2003], затрагивается вопрос организации хранения данных, получаемых в процессе расчета. Более подробно этот вопрос рассматривается в [Nazarenko, Prokhorov, 2015; Назаренко, Пересторонин, Прохоров, 2016].

2. Шаблоны расчетов

2.1. Последовательный расчет

Последовательный расчет является наиболее простым типом расчета. При последовательном расчете происходит поочередный запуск РС, причем РС, запущенные позднее, используют результаты работы РС, завершившихся до их запуска. Правила организации хранения данных зависят от конкретной задачи. В общем случае, некоторые РС в составе последовательного расчета используют единую рабочую директорию, другие же выполняются изолированно. (Для простоты изложения здесь и далее будем считать, что данные, получаемые в процессе расчета, сохраняются РС в файлы, создаваемые в текущей рабочей директории.)

Характерный пример последовательного расчета — газогидродинамический расчет, состоящий из последовательного запуска системы трехмерного моделирования, построителя сетки, препроцессора, решателя и постпроцессора (рис. 1).

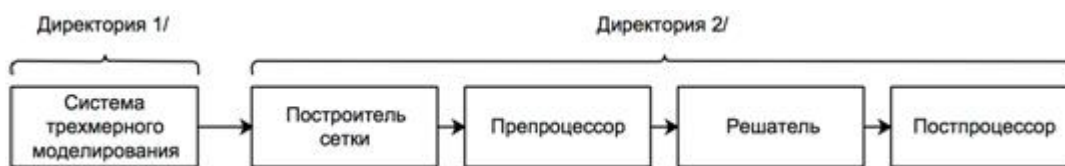


Рис. 1. Пример последовательного расчета

В терминах шаблонов потоков работ последовательный расчет является шаблоном «последовательность».

2.2 Параллельный расчет

Параллельный расчет состоит из двух или более РС, работающих независимо. В общем случае исходные данные для работы всех РС поступают из одного источника, результат же можно считать полученным тогда, когда работа всех РС завершена. Ввиду независимости работы РС, входящих в состав параллельного расчета, все они должны выполняться в изолированных рабочих директориях.

В качестве примера можно привести расчет некоего интегрального показателя (КПД), который вычисляется по характеристикам, получаемым в процессе моделирования разных режимов работы одного исследуемого объекта. Ввиду того, что расчеты характеристик разных режимов не связаны между собой, их следует проводить параллельно. Заметим, что параллельные расчеты могут использовать одни и те же исходные данные, в данном примере — расчетную сетку (рис. 2).



Рис. 2. Примеры параллельного (слева) и альтернативного (справа) расчетов

В терминах шаблонов потоков работ параллельный расчет состоит из двух шаблонов: «параллельное расщепление» и «синхронизация».

2.3. Альтернативный расчет

Альтернативный расчет состоит из двух или более РС, представляющих различные пути получения результата. При каждом запуске альтернативного расчета используется только одно из РС, причем его выбор зависит от данных, поступающих на вход альтернативного расчета. Очевидно, что для хранения файлов альтернативного расчета используется единственная рабочая директория.

В качестве примера приведем расчет, моделирующий обтекание аэродинамического объекта в широком диапазоне скоростей, включающем как дозвуковое, так и сверхзвуковое движение. Решатель выбирается в зависимости от скорости потока, задаваемого параметром — числом Маха. На основании полученных характеристик производится расчет механических нагрузок (рис. 2).

В терминах шаблонов потоков работ альтернативный расчет состоит из двух шаблонов: «исключающий выбор» и «простое соединение».

2.4. Пакетный расчет

Пакетный расчет использует одно РС для обработки множества наборов входных данных, причем на момент начала работы пакетного расчета все наборы входных данных уже определены. Пакетный расчет считается завершенным тогда, когда все входные данные обработаны. Так как каждый набор данных обрабатывается независимо, то и соответствующие рабочие директории должны быть изолированы. Структура хранения файлов должна однозначно определять к какому набору входных данных относится та или иная рабочая директория. Например, они могут быть пронумерованы в соответствии с порядком следования входных наборов данных.

Характерный практический пример пакетного расчета – многовариантное исследование расчетной модели на заданном плане эксперимента [Fisher, 1960; Volkov, Sukhoroslov, 2015] (рис. 3).

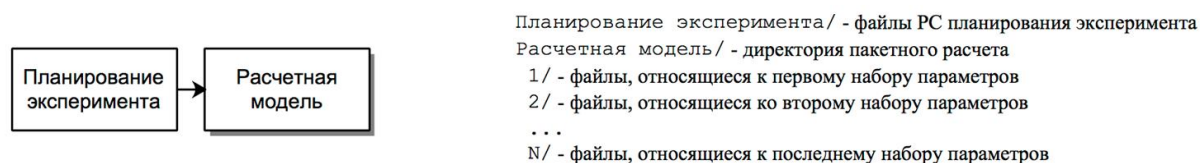


Рис. 3. Пример пакетного расчета: схема и структура рабочих директорий

В терминах шаблонов потоков работ пакетный расчет является шаблоном “мультидействие с известным до начала выполнения количеством экземпляров действия”.

2.5. Итеративный расчет

Итеративный расчет также использует одно РС для обработки множества наборов входных данных, однако в отличие от пакетного расчета, эти наборы данных и их количество не определены до начала вычислений. В общем случае входные данные для последующих расчетов (итераций) зависят от результатов предыдущих, а условие завершения всего итеративного расчета – от результатов всех вычислений. Так как каждый набор данных обрабатывается независимо, то и соответствующие рабочие директории должны быть изолированы. Как и при проведении пакетного расчета, структура хранения файлов должна однозначно определять к какому набору входных данных (к какой итерации) относится та или иная рабочая директория. Например, они могут быть пронумерованы в хронологическом порядке.

Характерным практическим примером итеративного расчета является поиск параметров, соответствующих оптимальным значениям характеристик некоторой расчетной модели, т.е. параметрическая оптимизация (рис. 4).

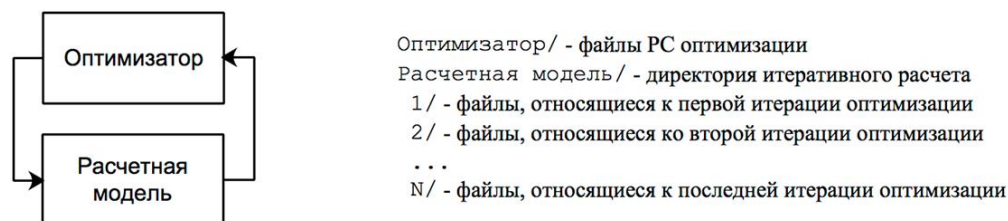


Рис. 4. Пример итеративного расчета: схема и структура рабочих директорий

В терминах шаблонов потоков работ итеративный расчет является шаблоном “мультидействие с неизвестным до начала выполнения количеством экземпляров действия”.

2.6. Итеративно-пакетный расчет

Итеративно-пакетный расчет состоит из комбинации итеративного и пакетного расчетов. Как и в итеративном расчете, данные, на которых проводятся вычисления, определяются по результатам предыдущих вычислений. Однако в отличие от итеративного расчета, на каждой итерации обрабатываются несколько наборов данных, т.е. фактически производится пакетный расчет. Можно сказать, что итеративный и пакетный расчеты являются частными случаями итеративно-пакетного расчета. Первое справедливо когда количество наборов данных для пакетной обработки на каждой итерации равно единице. Второе – когда весь расчет состоит лишь из одной итерации. Структура хранения файлов должна однозначно определять к какому набору входных данных относится та или иная рабочая директория. Например, можно применить сквозную нумерацию итераций и наборов данных пакетного расчета.

В качестве примера приведем процесс поиска оптимальных параметров расчетной модели с помощью алгоритма оптимизации, основанного на суррогатных моделях [Queiro, Haftka, ..., 2005]. Работа таких алгоритмов характеризуется тем, что целевые функции вычисляются в нескольких точках пространства параметров одновременно (рис. 5).

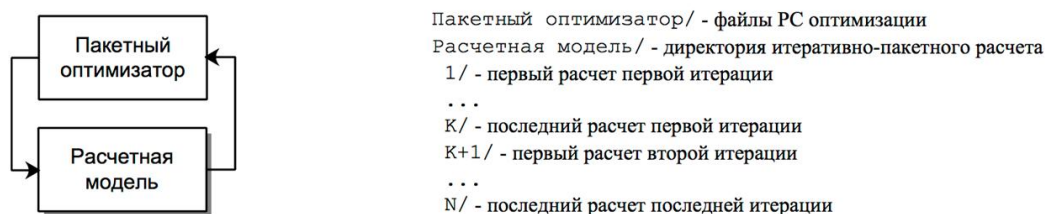


Рис. 5. Пример итеративно-пакетного расчета: схема и структура рабочих директорий

В терминах шаблонов потоков работ итеративно-пакетный расчет является шаблоном “мульти-действие с известным до начала выполнения количеством экземпляров действия”, вложенным в шаблон “мульти-действие с неизвестным до начала выполнения количеством экземпляров действия”.

2.7. Составной расчет

Очевидно, что при решении реальных вычислительных задач, требуется уметь комбинировать приведенные выше типовые расчеты. Для этого удобно использовать понятие составного расчета. Составным расчетом будем называть расчет, который может входить в состав любого из перечисленных типовых расчетов в виде РС, при этом сам он состоит из произвольного набора типовых расчетов, включая другие составные расчеты. Композиция типовых расчетов с помощью составного позволяет описывать расчетные схемы практически любой сложности. Здесь прослеживается аналогия с шаблоном “компоновщик” [Gamma, 1995], используемым в объектно-ориентированном проектировании (ООП). Действительно, составной расчет ничем не отличается от РС, т.е. они, выражаясь в терминах ООП, имеют общий интерфейс. Ввиду того, что составной расчет может содержать произвольную расчетную схему, необходимо обеспечить изоляцию хранения файлов от расчета, частью которого он является.

3. Заключение

В работе описаны семь базовых шаблонных схем инженерных и научных расчетов. Приведены поясняющие примеры. Установлено соответствие между указанными шаблонами и шаблонами потоков работ. Затронуты вопросы организации хранения данных, получаемых в процессе вычислений.

Представленные шаблоны расчетных схем, могут быть использованы в качестве элементов декомпозиции сложных вычислительных задач. Кроме того, возможность реализации перечисленных шаблонов может рассматриваться в качестве критерия применимости интеграционного программного обеспечения к решению актуальных научных и инженерных задач.

Список литературы

- Бурнаев Е.В., Губарев Ф.В., Морозов С.М., Прохоров А.А., Хоминич Д.С.* Автоматизация инженерных расчетов, анализ данных и оптимизация с помощью программного комплекса PSE/MACROS // Межотраслевая информационная служба. – 2012. – № 4 (165). – С. 41-50.
- Burnaev E. V., Gubarev F. V., Morozov S. M., Prokhorov A. A., Khominich D. S.* Avtomatizaciya inzhenernyh raschetov analiz dannyh i optimizaciya s pomoshchyu programmnoho kompleksa PSE/MACROS. [Process integration, design optimization and data analysis with PSE/MACROS framework] // Interindustry information service. – 2012. – No. 4 (165). – P. 41-50 (in Russian).
- Назаренко А.М., Пересторонин Н.О., Прохоров А.А.* Управление файлами в рамках модели потоков данных для распределенных вычислений // Научный вестник МГТУ ГА. – 2016. – Т. 19, № 05. – С. 161-172.
- Nazarenko A.M., Perestoronin N.O., Prokhorov A.A.* Upravlenie fajlami v ramkakh modeli potokov dannykh dlya raspredelennykh vychislenij. [File management in distributed dataflow-based workflows] // The Scientific Bulletin of MSTU GA. – 2016. – Vol. 19, No. 05. – P. 161-172 (in Russian).
- Радченко Г.И.* Грид-система CAEBeans: интеграция ресурсов инженерных пакетов в распределенные вычислительные среды. // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2009. – № 6-1.
- Radchenko G.I.* Grid-sistema CAEBeans: integraciya resursov inzhenernyh paketov v raspredelennye vychislitelnye sredy. [CAEBeans grid: CAE software integration for distributed calculations] // Vestnik of Lobachevsky University of Nizhni Novgorod. – 2009. – No. 6-1 (in Russian).
- Соболь И.М., Статников Р.Б.* Выбор оптимальных параметров в задачах со многими критериями. – М.: Дрофа. 2006. – 175 с.
- Sobol I.M., Statnikov R.B.* Vybora optimalnykh parametrov v zadachakh so mnogimi kriteriyami. [Choosing optimal parameters in problems with many criteria] – М.: Drofa, 2006 – 175 p. (in Russian).
- Deelman E., Gannon D., Shields M., Taylor I.* Workflows and e-Science: An overview of workflow system features and capabilities // Future Generation Computer Systems. – 2009. – Vol. 25, No. 5. – P. 528-540.
- Fisher R.A.* The design of experiments. – 1960.
- Gamma E.* Design patterns: elements of reusable object-oriented software. – Pearson Education India, – 1995.
- Кныазков К.В., Ковалчук С.В., Тчуоров Т.Н., Марин С.В., Буххановский А.В.* CLAVIRE: e-Science infrastructure for data-driven computing // Journal of Computational Science. – 2012. – Vol. 3, No. 6. – P. 504-510.
- Kodiyalam S., Sobieszczanski-Sobieski J.* Multidisciplinary design optimisation – some formal methods, framework requirements, and application to vehicle design // International Journal of Vehicle Design. – 2001. – Vol. 25, No. 1-2. – P. 3-22.
- Martins J.R.R.F., Lambe A.B.* Multidisciplinary Design Optimization: A Survey of Architectures // AIAA journal. – 2013. – Vol. 51, No. 9. – P. 2049-2075.
- Nazarenko A.M., Prokhorov A.A.* Hierarchical Dataflow Model with Automated File Management for Engineering and Scientific Applications // Procedia Computer Science. – 2015. – Vol. 66. – P. 496-505.

- Sukhoroslov O., Volkov S., Afanasiev A.* A Web-Based Platform for Publication and Distributed Execution of Computing Applications // 14th International Symposium on Parallel and Distributed Computing (ISPDC). – IEEE, 2015. – P. 175-184.
- van Der Aalst, Wil MP, Arthur HM Ter Hofstede, Bartek Kiepuszewski, Alistair P. Barros* Workflow patterns // Distributed and parallel databases. – 2003 – Vol. 14, No. 1. – P. 5-51.
- Volkov S., Sukhoroslov O.* A Generic Web Service for Running Parameter Sweep Experiments in Distributed Computing Environment // Procedia Computer Science. – 2015. – Vol. 66. – P. 477-486.
- Queipo N.V., Haftka R.T., Shyy W., Goel T., Vaidyanathan R., Tucker P.K.* Surrogate-based analysis and optimization // Progress in aerospace sciences. – 2005. – Vol. 41, No. 1. – P. 1-28.

Patterns of engineering and scientific workflows

A. A. Prokhorov^{1,a}, A. M. Nazarenko^{1,b}, A. V. Davydov^{2,c}

¹ Institute for information transmission problems,
19, Bolshoy Karetnyy per., Moscow, 127051, Russia

² DATADVANCE LLC,
17, Nauchny proezd, Moscow, 117246, Russia

E-mail: ^a prokher@gmail.com, ^b nazar@phystech.edu, ^c andrey.davydov@datadvance.net

Simulation experiments and computer models are essential for modern scientific and engineering studies. Often solving some problem requires using multiple software tools. In the context of automation such computational tasks are compositions of weakly coupled (in terms of communication intensity) typical subtasks, each being performed by a dedicated software tool. In an effort to automate computations of this kind, a special type of software – integration environments and platforms – have appeared and developed in the recent decades.

Despite the current variety of such software solutions, not all of them can support automation of such computations that are needed in present day tasks. This is a consequence of transition from using computational experiments mainly for hypothesis verification and quality control of designed products to using simulation experiments in global design space exploration, in particular for the optimization of computer models in question. Moreover, many of today's research and engineering tasks are multidisciplinary problems that involve complex schemes of communication between the software tools that perform simulation steps.

In order to determine the requirements for integration software intended to automate scientific and engineering calculations we have analyzed a great number of computational workflows, identified typical computational subtasks and described corresponding computing patterns. This paper explains seven basic patterns: sequential, parallel, alternative, batch, iterative, batched iterative, and composite. Each description includes a typical example, notes relevant workflow patterns, addresses questions of storing the data obtained during computation.

The presented patterns can be used in decomposition of complex computational tasks. Also, the capability to implement these patterns can be considered as a criteria of applicability of a certain integration software to solving modern research and engineering problems.

Keywords: process integration, process automation, workflow, workflow patterns.

The study is supported by the Russian Foundation for Basic Research (RFBR), research project No. 15-29-07043. Research facilities are provided by DATADVANCE LLC.