

# СНІТОН – ФРЕЙМВОРК ДЛЯ РАЗРАБОТКИ НАУЧНЫХ ВЕБ-СЕРВИСОВ

Е.В. Соболев, М.И. Пятков, О.В. Соболев

Часто результатом научных исследований, связанных с разработкой новых методов обработки экспериментальных данных или моделированием, являются реализующие их вычислительные программы. Новые научные программы в итоге нужно внедрять и популяризовать. Классическим способом распространения компьютерных программ является оформление их в виде отдельного продукта, который можно выслать по электронной почте или разместить на файловом сервере. В этом случае обычно приходится разрабатывать пользовательский интерфейс (командной строки или графический), инструкцию по установке, запуску и интерпретации результатов. Часто бывает необходимо предусмотреть работоспособность программы под управлением различных операционных систем.

Чтобы воспользоваться таким продуктом, пользователю также необходимо совершить ряд действий, например удовлетворить программно-аппаратные требования, установить программу, изучить документацию и т.п. При этом пользователь несет риски, связанные с безопасностью, потому что новая программа может быть нестабильной или даже опасной. Очевидно, что если пользователю необходимо просто ознакомиться с новой программой или применять ее от случая к случаю, затраты становятся очень большими по сравнению с ожидаемым эффектом. Часто это приводит к тому, что большинство ученых, которым новый инструмент мог бы оказаться полезным, отказываются от попыток ознакомиться с ним, отдавая предпочтение тем продуктам, которые уже зарекомендовали себя.

Поиск решения этой проблемы привел к появлению парадигмы «облачных вычислений», которые предполагают удаленную обработку и хранение данных. В рамках этой парадигмы сформулирована модель обслуживания «программа как сервис», которая вместо распространения самой программы предлагает предоставлять доступ к ней через Интернет. Это может значительно расширить круг пользователей программы, особенно если предлагаемая программа не является необходимой для решения задач пользователя. У крупных научных учреждений, как правило, есть целый набор веб-сервисов. Примером могут служить Европейский институт биоинформатики [1] и банк белковых структур [2]. Отдельные научные группы обычно поддерживают одну или несколько программ [3, 4].

Распространение программы в виде веб-сервиса обладает и другими преимуществами по сравнению с классическим способом. Наиболее значимым для разработчика является то, что:

- нет необходимости создавать мультиплатформенное приложение, а достаточно добиться стабильной работы программы только на сервере;
- нет необходимости распространять новые версии программы, достаточно обновить копию, связанную с веб-интерфейсом;
- сопровождать программы проще, так как вся информация об ошибках собирается непосредственно разработчиком.

Недостаток «облачной» модели заключается в необходимости создать веб-интерфейс программы. Эта задача требует значительных усилий, знаний в области веб-программирования и сетевых технологий. При этом часто такая деятельность не является приоритетной для разработчика, не ведет к научным открытиям и выполняется по остаточному принципу. Поэтому авторы чаще останавливаются на классической схеме распространения в виде отдельной программы. Избавиться от этого недостатка можно, если удастся сократить затраты на создание веб-интерфейса и сделать их сопоставимыми с затратами на распространение программы по классической схеме.

На наш взгляд, наиболее эффективный способ решения этой задачи заключается в создании шаблона веб-сервиса, так называемого фреймворка, который хорошо подходит для построения подавляющего большинства конкретных реализаций и настраивается под каждую путем доопределения небольшого количества ее особенностей. Преимущества фреймворка заключаются в наличии богатого набора функций по умолчанию и возможности его расширения. Полноценный фреймворк для разработки веб-интерфейсов должен обеспечивать:

1. Сбор входных данных и параметров через веб-интерфейс соответственно их типам и проверку на допустимые значения.
2. Передачу входных файлов на вычислитель.
3. Формирование задачи и постановку ее в очередь на вычислителе.
4. Отслеживание и отображение статуса задачи.
5. Копирование выходных данных с вычислителя на файл-сервер.
6. Отображение в наглядном виде результатов расчетов.

В качестве вычислителя может использоваться как компьютер, управляющий веб-интерфейсом, так и удаленный компьютер или кластер, к которому возможен доступ по локальной сети или сети Интернет.

Если зафиксировать исполняющую часть, в частности, протоколы доступа к вычислителю, тип очереди и т.п., то первые пять функций будут иметь практически идентичную реализацию для большей части

вычислительных программ. Всего несколько реализаций каждой из этих функций позволяют покрыть подавляющее большинство возможных вариантов исполняющей части.

Наглядное отображение результатов расчетов является наиболее трудной задачей для унификации, однако значительная часть способов сохранения информации в файле и способов ее отображения являются типовыми. Наиболее простой случай – отображение текстовых результатов работы программы (лог-файлов) на веб-странице. В более сложных случаях в файлах могут записываться данные в определенном формате. Например, часто табличные данные записываются в текстовый файл по строкам, ячейки в которых разделены некоторым символом. Отображение табличных данных возможно непосредственно в виде таблицы, или в виде диаграмм и графиков. Графические файлы, видео файлы и широко распространенные научные форматы, такие как PDB, MOL, CIF, CUBE, могут отображаться непосредственно браузером или уже разработанными для их визуализации небольшими приложениями (апплетами).

Кроме основных функций, необходимых для реализации веб-интерфейса вычислительной программы, можно выделить сопутствующие функции, которые делают интерфейс более удобным и позволяют получать дополнительную информацию его создателю.

К настоящему моменту нами разработан ряд различных инструментов: моделирование гидратации биополимеров [4], поиск альтернативных конформаций в процессе кристаллографического уточнения [5], поиск протяженных повторов в ДНК [6]. Автоматизация разработки веб-интерфейса с помощью фреймворка позволило бы организовать их распространение.

CHITON – разрабатываемый нами фреймворк для создания веб-интерфейса вычислительной программы. Для его разработки используется язык программирования Python и его расширение Django. Фреймворк разрабатывается согласно принципам объектно-ориентированного программирования. Базовая архитектура фреймворка построена на классах, описывающих вычислительную программу, форму сбора входных данных, протокол доступа к вычислителю для передачи файлов и постановки задач в очередь, используемую очередь для запуска асинхронных задач и класс формирования отчетов. Конкретная реализация этих классов, путем использования фильтров и функций обратного вызова, настраивает фреймворк под конкретную аппаратно-программную часть. Предусмотрена библиотека реализаций фильтров и функций обратного вызова для различных часто встречающихся вариантов. Для случаев, отсутствующих в библиотеке, можно написать свою реализацию.

Класс, описывающий приложение, в своих методах должен содержать описания входных и выходных данных, их количество и типы. Для ряда типов предусмотрены стандартные описатели, которые позволяют фреймворку автоматически построить классы формы сбора данных и формирования отчета.

Класс, описывающий вычислитель, требует задать методы передачи входных и выходных файлов и метод постановки задания в очередь. Предусмотрены стандартные реализации этих методов. На данный момент существуют реализации для локального доступа и доступа по протоколу SSH и очередей Celery и PBS.

В минимальном варианте разработчик веб-интерфейса для своей программы должен сформировать классы, описывающие вычислительную программу и вычислитель с помощью стандартных реализаций методов. Такое описание занимает несколько десятков строк на языке Python и его достаточно, чтобы все функции веб-интерфейса заработали в стандартном режиме.

На данный момент CHITON выполняет ряд основных функций: формирование веб-страницы с пояснениями для ввода данных, в том числе с возможностью передачи файлов; подготовка запуска программы на локальном или удаленном вычислителе; постановка задачи в очередь; информирование о текущем статусе активной задачи; формирование страницы результатов с отображением ряда стандартных форматов данных (текст, таблица, график, рисунок).

Внедрение CHITON способно значительно сократить затраты на создание веб-интерфейса для научных расчетных программ. При этом применение отлаженного фреймворка, созданного согласно современным парадигмам, обеспечивает высокое качество построенных на его основе веб-интерфейсов.

В данный момент CHITON находится в стадии активной разработки, тестовые версии доступны у авторов по запросу (egor@impb.psn.ru).

Работа выполнена при поддержке РФФИ, проекты № 12-04-31096 и № 12-07-31085.

#### ЛИТЕРАТУРА:

1. H. McWilliam, F. Valentin, M. Goujon et al. // *Nucleic Acids Research*. 2009. V. 37. P. W6–W10.
2. H.M. Berman, J. Westbrook, Z. Feng et al. // *Nucleic Acids Research*. 2000. V. 28. P. 235–242.
3. J.C. Gordon, J.B. Myers, T. Folta et al. // *Nucleic Acids Research*. 2005. V. 33. P. W368–W371.
4. E.V. Sobolev, O.V. Sobolev, D.A. Tikhonov. // *SAR and QSAR in Environmental Research*. 2008. V. 19. P. 303–315.
5. Sobolev O.V., Lunin V.Y. // *Acta Crystallographica Section D: Biological Crystallography*. 2012. V. 68. P. 1118–1127.
6. A. Pankratov, M. Pyatkov, F. Dedus, et. al. // *Pattern Recognition and Image Analysis*. 2009. V.19. P.678–692.