

НОВЫЙ ПОДХОД К ВЗАИМОДЕЙСТВИЮ ОБЪЕКТОВ В ПРОГРАММНОМ ПАКЕТЕ ДЛЯ ПОДДЕРЖКИ РАБОТЫ В ПРОГРАММНОМ КОМПЛЕКСЕ OCTOSHELL

Паокин Андрей Викторович

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: andrejpaokin@yandex.ru

Научный руководитель — Никитенко Дмитрий Александрович

Octoshell – система управления доступом к суперкомпьютерным центрам [1]. На данный момент она используется в самом крупном в России Суперкомпьютерном центре Московского Государственного Университета имени М.В.Ломоносова. Octoshell работает удаленно с суперкомпьютерами по SSH и может работать без доступа к вычислительной системе.

Octoshell v2.5 заменяет собой сразу несколько инструментов в едином веб-интерфейсе для организации доступа, техподдержки, рассылок, для информирования пользователей о статусе задач, для управления доступом к программному обеспечению. Octoshell v2.5 построен на платформе веб-приложений Ruby on Rails v5, которая следует архитектурному шаблону Model-View-Controller, и функциональность системы распределена между модулями. Модульность позволяет достичь между модулями слабой связности, а внутри них – сильной связности, что упрощает разработку системы [2]. Обычные подходы для платформы Ruby on Rails рассматривают модуль как подсистему, которая взаимодействует только с основным приложением, что противоречит многомодульной организации Octoshell. В результате классы из разных модулей взаимодействуют точно так же, как и классы в одном модуле, что делает межмодульные зависимости жесткими.

В общем случае предлагается блоки кода, используемые для межмодульного взаимодействия, оборачивать if-конструкциями, которые будут осуществлять проверку того, что необходимые ассоциации и модели присутствуют в Octoshell. С другой стороны, этот подход не решает проблемы модулей, которые встраиваются в функциональность других модулей. Например, при просмотре информации о конкретном пользователе может понадобиться вся информация, распределенная между модулями: проекты, отчеты, опросы, SSH-ключи. В таких случаях используются хуки, т.е. конкретные места в веб-интерфейсе получает имя, которые модули используют для встра-

ивания. В работе выделены наиболее частые типы межмодульного взаимодействия, для каждого из которых разработано своё решение.

Работа выполнена при поддержке РФФИ (грант № 18-29-03230).

Литература

1. Nikitenko D.A., Voevodin Vad.V., Zhumatiy S.A. Driving a Petascale HPC Center with Octoshell Management System // Lobachevskii Journal of Mathematics. 2019. Т. 40, № 11. С. 1817–1830
2. Официальный сайт Ruby on Rails: <https://rubyonrails.org/>