

ЭКСПЕРИМЕНТЫ С JUPYTERHUB В САНКТ-ПЕТЕРБУРГСКОМ ГОСУДАРСТВЕННОМ УНИВЕРСИТЕТЕ

Андрей Ерохин ^a, Андрей Зароченцев ^b

*Санкт-Петербургский Государственный Университет, Университетская набережная 7-9,
Санкт-Петербург, 199034, Россия*

E-mail: ast016885@student.spbu.ru, ba.zarochentsev@spbu.ru

Описывается опыт использования JupyterHub, расширения функциональности Jupyter-ноутбуков и то, как можно использовать JupyterHub для задач, которых он не предназначался. Проект начинался как клон CERN SWAN, но сейчас развивается независимо. Тем не менее, так же, как в SWAN, используется CVMFS для загрузки ядер Jupyter и прочего программного обеспечения и EOS для хранения пользовательских домашних каталогов.

Ключевые слова: jupyter, kubernetes, удалённый рабочий стол

© 2018 Андрей Ерохин, Андрей Зароченцев

Предпосылки использования и развития

При решении современных научных задач, практически во всех случаях, используются специализированные программные пакеты. Это могут быть и широко распространённые пакеты математической обработки данных с широкой специализацией (MatLab, Mathematica, Ansys и др.), так и узкоспециализированные программные продукты (например, пакеты AliRoot, FairRoot, разработанные для конкретных экспериментальных установок в области релятивистской ядерной физики). Установка этих пакетов на локальные ресурсы может быть связана с такими трудностями как лицензионные ограничения, технические ограничения по «мощности» или спецификации локальных ресурсов, технические сложности в установке. Для решения этих проблем существуют компьютерные центры коллективного пользования. Но в этом случае возникает вопрос об удалённом авторизованном доступе к этим компьютерным центрам. Зачастую, этот доступ тоже требует дополнительного клиентского программного обеспечения. Это может быть RDP или SSH клиент, что не всегда по умолчанию реализовано на всех операционных системах. В тоже время, полноценные сессии с визуализацией графических приложений могут требовать довольно широкого канала связи и, опять же, локальных ресурсов на клиентском персональном компьютере.

Прекрасным решением описанной проблемы является предоставление доступа к ресурсам с помощью браузерных приложений. В этом случае решается вопрос с локальным ПО (браузерами обеспечены все существующие операционные системы) и вопрос с нагрузкой на локальное оборудование и канал - визуализировать надо только конкретное приложение, а не всю сессию. Из существующих и существовавших ранее реализаций браузерного, пользовательского доступа к ресурсам авторы выбрали пакет JupyterHub [1] и созданный на его основе в CERN пакет SWAN [2].

История развития проекта

Проект SWAN (Service for Web based ANalysis) предлагается к использованию в CERN уже в течение двух лет. В его основе лежит независимый проект Jupyter.

Jupyter Notebook представляет из себя:

- стандартизированный формат файлов с возможностью хранить в нём программный код, математические формулы и визуальные данные, например, графики и гистограммы ROOT [3];
- веб-приложение для интерактивной работы с данным форматом документа;
- стандартизированный протокол связи с так называемыми ядрами (Jupyter kernels) — приложениями, которые обрабатывают содержащийся в файле программный код и могут возвращать результаты вычислений в виде графиков, текста и т.д. Существует множество реализаций ядер для различных языков программирования.

Веб-приложение (сервер) Jupyter Notebook является однопользовательским. Для поддержки многопользовательской работы существует проект JupyterHub. JupyterHub представляет из себя централизованную систему для аутентификации пользователей и запуска (spawn) однопользовательских веб-приложений. Интерфейс классов, реализующих аутентификацию и запуск, является документированным, что обеспечивает возможность подключать различные модули для аутентификации и запуска. Чаще всего используются модули для аутентификации с использованием PAM (аутентификация локальных UNIX-пользователей), протокола LDAP, протокола OAuth. Простейшим способом запуска однопользовательских веб-приложений Jupyter является порождение подпроцесса. Для изоляции пользовательских процессов друг от друга используется контейнеризация Docker. Для запуска контейнеров на многих вычислительных узлах используется оркестратор контейнеров Kubernetes.

Итак, SWAN представляет из себя JupyterHub для аутентификации пользователей и запуска их однопользовательских серверов. Файловой системой для домашних каталогов пользователей является CERN EOS. Ядра для Jupyter и ряд других программ и библиотек загружаются с CVMFS – распределённой файловой системы, используемой в WLCG для распространения собранных приложений.

В СПбГУ был развёрнут аналогичный проект, опирающийся на те же идеи – загрузку ПО из CVMFS и хранение домашних каталогов в EOS.

Расширение для Jupyter

«Интерактивность» в Jupyter означает, что пользователю при разработке кода не нужно повторять цикл редактирование-компиляция-запуск. «Интерактивность» в Jupyter близка к интерактивности в смысле REPL (Read-Eval-Print Loop), с наглядным отображением результатов вычислений введённого кода и сохранением истории ввода вместе с результатами в файле. Но подобной интерактивности не всегда достаточно, иногда возникает необходимость использовать полноценные GUI-приложения, например, визуализировать геометрию детектирующей установки в GEANT4 [4]. Поэтому возникла задача предоставить возможность запускать графические приложения и при этом сохранить преимущества подхода в смысле отсутствия необходимости устанавливать ПО, в том числе клиент удалённого доступа к рабочему столу, на пользовательские машины.

С появлением стандарта HTML5 с такими возможностями как элементы canvas, позволяющими рисовать на веб-странице и протоколом WebSocket, позволяющим организовать полноценное двустороннее (full-duplex) соединение с сервером из браузера, ряд серверов доступа к удалённому рабочему столу реализовали браузерный клиент. Было решено разработать расширение для Jupyter Notebook, позволяющее пользователям запускать и останавливать сервер удалённого рабочего стола, получать к нему доступ. В качестве сервера удалённого доступа к рабочему столу был выбран проект Xpra [5], имеющий HTML5-клиент.

Jupyter использует веб-сервер Tornado, написанный на Python. Пользователь, работая со своим домашним каталогом и Jupyter-ноутбуками из браузера, обращается к этому веб-серверу. Для доступа к серверу удалённого рабочего стола, работающему в контейнере с веб-сервером Tornado и реализующему доступ с использованием технологий HTML5, достаточно организовать проксирование трафика от сервера рабочего стола через веб-сервер Tornado до браузера клиента. К счастью, подобный прокси-сервер для Jupyter уже существует: nbserveprxou. Этот прокси-сервер является ключевой зависимостью для разработанного расширения. Прокси-сервер был переписан для исправления ряда ошибок и придания ему большей настраиваемости.

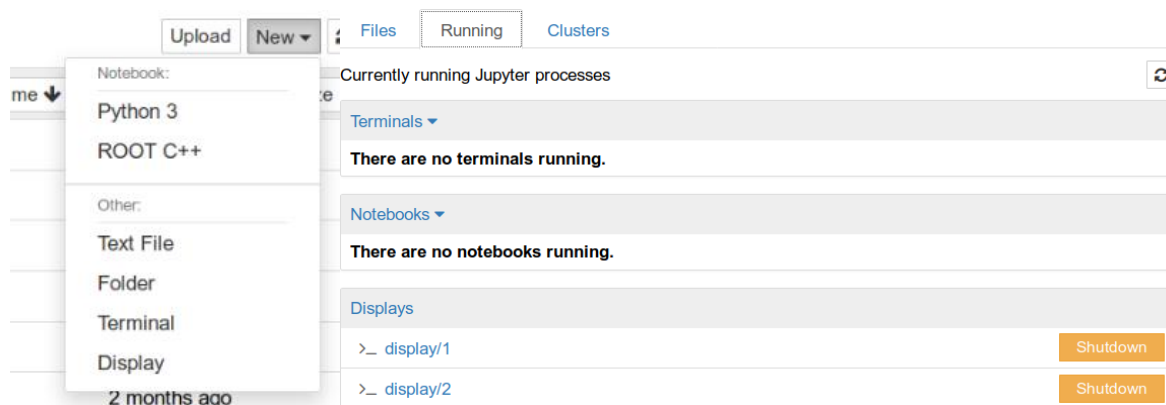


Рисунок 1. Кнопка (Display) запуска нового сервера удалённого рабочего стола и список запущенных серверов с кнопками для их завершения

Разработанное расширение состоит из серверной и клиентской частей. Серверная часть написана, естественно, на python, клиентская – на JavaScript. Серверная часть предоставляет API для запуска, остановки и перечисления запущенных серверов рабочего стола, следит за статусом процессов, уведомляет клиентов об их завершении или запуске новых серверов, настраивает проксирование. Клиентская часть модифицирует интерфейс веб-страницы Jupyter (см. Рисунок 1), добавляя элементы для запуска и остановки серверов рабочего стола, просмотра их списка, и взаимодействует с серверной частью через API для выполнения соответствующих функций.

Использование JupyterHub для запуска сессий удалённого рабочего стола

Многопользовательский сервер JupyterHub, в принципе, не зависит от того, какой процесс или контейнер он порождает для пользователя. Достаточно, чтобы процесс после запуска ожидал подключений на указанном порту и корректно интерпретировал передаваемые через переменные окружения параметры, такие как имя аутентификационной HTTP cookie, URL-префикс и т.п. Порождаемый процесс не обязательно должен быть однопользовательским сервером Jupyter. Это свойство JupyterHub даёт возможность использовать его преимущества – настраиваемый способ аутентификации и порождения процессов/контейнеров – для запуска различных интерактивных серверов. Например, многим пользователям не нужна функциональность Jupyter Notebook-ов, а нужен только доступ к удалённому рабочему столу.

Было реализовано приложение на основе веб-сервера Tornado и расширения nbservedproху, запускающее подпроцесс Xpra, настраивающее прокси и корректно взаимодействующее с JupyterHub: принимающее подключения на указанном порту и т.д. Таким образом, JupyterHub с KubeSpawner-ом — компонентом для запуска контейнеров в Kubernetes-кластере – позволяет запускать для пользователей сессии удалённого рабочего стола в Kubernetes-облаке. Приложение работает самостоятельно, а не как расширение Jupyter Notebook и, следовательно, не требует запуска последнего.

Нерешённые проблемы

У некоторых пользователей возникают проблемы с соответствием между раскладкой клавиатуры на локальной машине и в удалённой сессии рабочего стола.

Планы

В рамках проекта Jupyter развивается новый, более удобный интерфейс JupyterLab. Планируется доработать клиентскую часть нашего расширения для Jupyter Notebook для работы в JupyterLab.

Благодарности

Эта работа была поддержана Российским Научным Фондом, грант №17-72-20045.

Ссылки

- [1] JupyterHub [online resource]: <https://jupyterhub.readthedocs.io/>, (accessed 09.11.2018).
- [2] The Swan Service [online resource]: <https://swan.web.cern.ch/>, (accessed 09.11.2018).
- [3] ROOT is ... [online resource]: <https://root.cern/>, (accessed 09.11.2018).
- [4] Geant4 [online resource]: <https://geant4.web.cern.ch/>, (accessed 09.11.2018).
- [5] Xpra [online resource]: <https://www.xpra.org/>, (accessed 09.11.2018).

EXPERIMENTS WITH JUPYTERHUB AT THE SAINT PETERSBURG STATE UNIVERSITY

Andrey Erokhin ^a, Andrey Zarochentsev ^b

St. Petersburg State University, 7-9 Universitetskaya Emb., St Petersburg 199034, Russia

E-mail: ast016885@student.spbu.ru, ba.zarochentsev@spbu.ru

The article focuses on our experience with JupyterHub, the ways we extend Jupyter and how we abuse JupyterHub to spawn something other than single-user Jupyter notebook servers. Our project has started as a copy of CERN SWAN environment, but it evolves independently. However, we are still using CVMFS to load Jupyter kernels and other software and EOS to store user home directories.

Keywords: jupyter, kubernetes, remote desktop

© 2018 Andrey Erokhin, Andrey Zarochentsev