

Моделирование нестационарных гидродинамических течений с помощью резервуарных вычислений

Анна-Мария Амелькина

2022.06.17: Удаленный запуск кода Waleffe_flow.Fortran

1 Описание структуры программы Waleffe_flow.Fortran

Структура директорий исходных кодов и запусков программы Waleffe_flow.Fortran изображена на рисунке [1](#).

2 Удаленный запуск кода

Для подключения к удаленной машине используется SSH – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений [\[1\]](#).

Для запуска программы Waleffe_flow.Fortran на удаленной машине необходимо выполнить следующие команды:

1. Подключение к удаленной машине [\[2\]](#):

```
ssh user@server
```

В данной работе использовалась виртуальная машина с именем hpc.rk6.bmstu.ru и пользователь amamelkina, поэтому команда выглядела следующим образом:

```
ssh amamelkina@hpc.rk6.bmstu.ru
```

2. Переход в директорию с программой [\[3\]](#):

```
cd /home/amamelkina/MWF-master
```

3. Чтобы настроить среду для использования библиотек MPI, нужно загрузить соответствующий модуль окружающей среды:

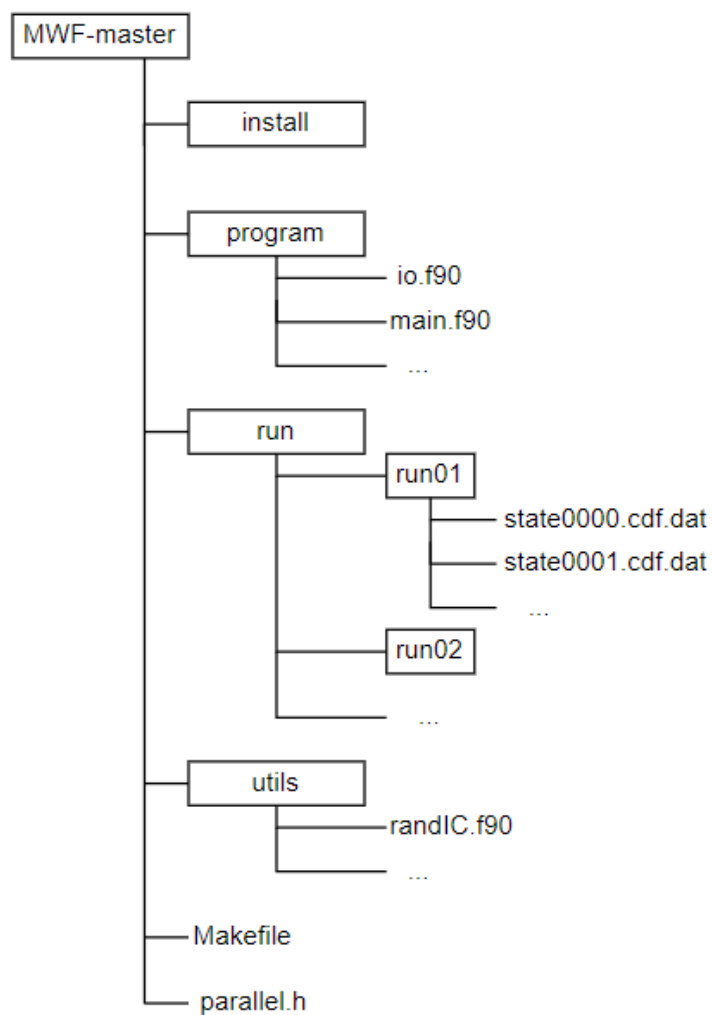


Рис. 1. Структура программы Waleffe_flow.Fortran

```
module load mpi
```

4. Сборка:

```
make
make install
```

5. Создание начальных условий:

```
make util
./randIC.out
```

- Далее необходимо создать папку, в которую будут записываться результаты. Такие наборы результатов хранятся в папке run. В папке run создается папка с номером (например run10), в нее копируются файлы /install/main.info, /install/main.out.

Файл `state0000.cdf.dat`, полученный в результате выполнения пункта 5, должен быть переименован в `state.cdf.in` и тоже скопирован в новую папку результатов.

7. Затем нужно перейти в созданную папку и запустить программу:

```
./main.out
```

8. После остановки программы, когда будет получено необходимое количество данных, можно отключаться от удаленной машины с помощью команды:

```
exit
```

9. Последний шаг – скопировать результат с удаленной машины на локальную [4]:

```
scp -r user@server:<адрес/откуда/копируем>  
      <адрес/куда/копируем>
```

Видно, что запуск данной программы на удаленной машине требует выполнения множества команд. Упростим запуск до одной команды – запуска `python`-скрипта, в котором будет реализован процесс удаленного запуска.

3 Программная реализация

Для реализации необходимого `python`-скрипта была использована библиотека `fabric`. `Fabric` – это библиотека `Python` и инструмент командной строки для оптимизации использования `SSH` для развертывания приложений или задач системного администрирования [5].

Для работы с `fabric` первым делом нужно создать `fabfile` и разместить в структуре файлов так, как показано на рисунке 2.

`Fabfile` – это то, что контролирует то, что выполняет `fabric`. Он называется `fabfile.py` и запускается командой `fab`. Все функции, определенные в этом файле, будут отображаться как подкоманды `fab`. Они выполняются на одном или нескольких серверах. Эти серверы могут быть определены либо в `fabfile`, либо в командной строке [6].

Добавим сервер в `fabfile`, определив его в переменной окружения `env` (листинг 1) [7].

Листинг 1. Добавление сервера в переменную окружения `env`

```
1 env.hosts = ['hpc.rk6.bmstu.ru']
```

`Fabric` по умолчанию использует локальное имя пользователя при подключении `SSH`, но при необходимости его можно переопределить, используя `env.user`. Предоставим пользователю программы возможность ввести имя (листинг 2). Это реализовано

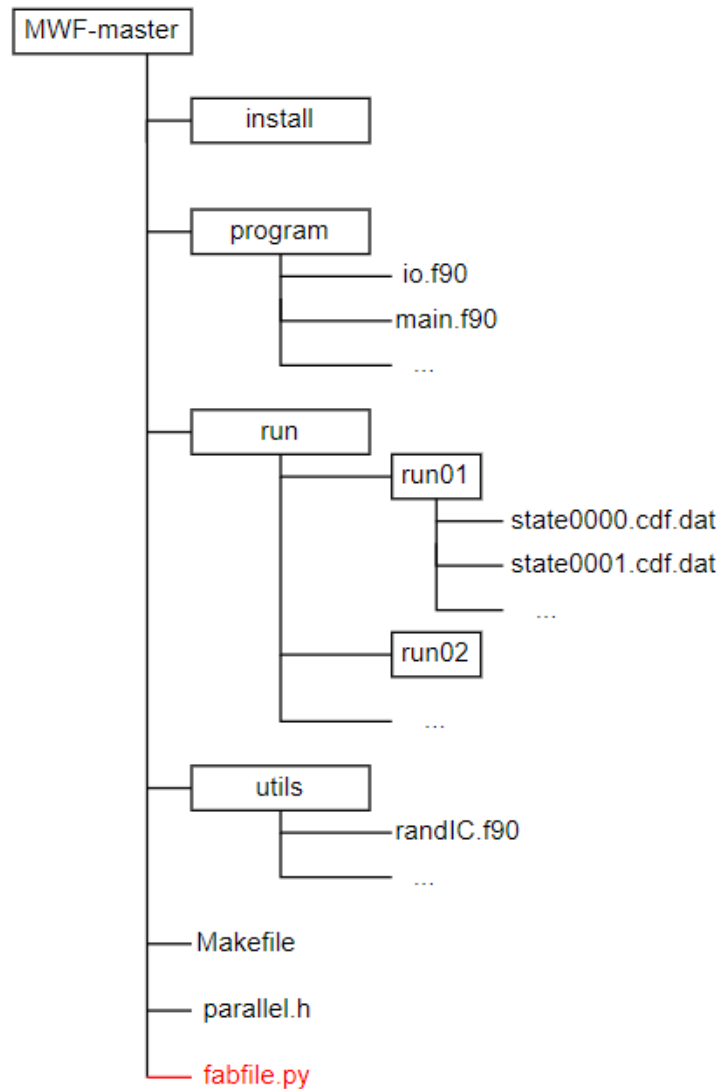


Рис. 2. Размещение fabfile

с помощью функции `prompt(text, default='', ...)` [8]. Данная функция выдаёт пользователю запрос с текстом `text` и возвращает полученное значение. Для удобства к `text` будет добавлен одиночный пробел. Если передан параметр `default`, то он будет выведен в квадратных скобках и будет использоваться в случае, если пользователь ничего не введёт (т.е. нажмёт Enter без ввода текста). По умолчанию значением `default` является пустая строка.

Листинг 2. Добавление имени пользователя в переменную окружения `env`

```
1 user = prompt("Enter username", default='amamelkina')
2 env.user = user
```

Также пользователю нужно ввести время в секундах, в течение которого будет

выполняться программа.

Затем с помощью функции `run(command, ...)`, которая запускает команду оболочки на удалённом узле, выполняется запуск программы `Waleffe_flow.Fortran`.

Также используются функции `put(local_path=None, remote_path=None, ...)` и `get(remote_path, local_path = None)` для загрузки файлов на удаленный сервер и скачивания файлов с удалённого сервера соответственно.

Листинг кода python-скрипта представлен в Приложении.

Для запуска `fabric` предоставляет команду `fab`, которая считывает свою конфигурацию из файла `fabfile.py`.

В листинге 3 представлена простая функция, с помощью которой будет продемонстрировано, как использовать `fabric` [9]. Эта функция сохранена как `fabfile.py` в текущем рабочем каталоге.

Листинг 3. Пример функции

```
1 def hello():
2     print("Hello!")
```

Функция приветствия может быть выполнена с помощью `fab` инструмента следующим образом:

```
fab hello
```

В результате выполнения этой команды будет выведено "Hello!".

Таким образом, для запуска программы `Waleffe_flow.Fortran` на удаленной машине теперь необходимо выполнить только одну команду:

```
fab run_fortran
```

В результате работы данного python-скрипта на персональном компьютере в папке `run` появится новая папка с результатами запуска.

Список литературы

- [1] SSH. // ru.wikipedia.org – URL: <https://ru.wikipedia.org/wiki/SSH>. (Дата обращения 6.05.2022).
- [2] Памятка пользователям ssh. // habr.com – URL: <https://habr.com/ru/post/122445/>. (Дата обращения 6.05.2022).
- [3] Управляем файлами и директориями в Linux. // infobox.ru – URL: <https://infobox.ru/community/blog/linuxvps/310.html>. (Дата обращения 6.05.2022).
- [4] Копирование файлов и папок через SSH. // ru.flamix.software – URL: <https://ru.flamix.software/about/news-article/kopirovanie-faylov-i-papokok-cherez-ssh/>. (Дата обращения 6.05.2022).

- [5] Fabric documentation. // docs.fabfile.org – URL: <https://docs.fabfile.org/en/1.11/tutorial.html>. (Дата обращения 16.05.2022).
- [6] Fabric. // runebook.dev – URL: <https://runebook.dev/ru/docs/flask/patterns/fabric/index>. (Дата обращения 16.05.2022).
- [7] The environment dictionary, env. // docs.fabfile.org – URL: <https://docs.fabfile.org/en/1.11/usage/env.html>. (Дата обращения 16.05.2022).
- [8] Python-блог: Fabric: Операции. // python-lab.blogspot.com – URL: <http://python-lab.blogspot.com/2013/02/fabric.html>. (Дата обращения 16.05.2022).
- [9] Начало работы с библиотекой Fabric Python. // coderlessons.com – URL: <https://coderlessons.com/articles/programmirovanie/nachalo-raboty-s-bibliotekoi-fabric-python>. (Дата обращения 16.05.2022).