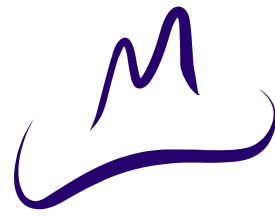


MSE 07

Iteration №3





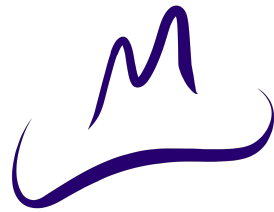
Состав команды

Магистр

- 1) Тиняков Сергей Алексеевич (9304)

Бакалавры

- 2) Кондратенко Константин Евгеньевич (1384)
- 3) Денисова Ольга Константиновна (1381)
- 4) Бутыло Егор Алексеевич (1303)
- 5) Андреева Елизавета Алексеевна (1303)



План на текущую итерацию

В итерации 2 было зафиксировано, что планами на текущую являются:

- Дополнить информацию о процессе кэширования сборки образа
- Модифицировать Docker образ для автоматического сбора статистики сборки образов, предоставить образу к конфигурации ядра хоста (для глубокой трассировки)
- Произвести первичный анализ логов на выявление явных симптомов потребления ресурсов



Полученный результат (1)

- Дополнение информации о создании кэша
 - Изучено, где располагаются следующие виды кэша
 - Кэш **процесса сборки** (кэш состояния)
 - Кэш **утилит** сборки
 - Загруженный **исходный код**
 - Изучено, как можно настроить расположение локального кэша
 - Изучено, как можно использовать локальный кэш
 - Изучено, как создать FTP и HTTP сервера с кэшем

← → ↻

🔒 📁 localhost:8000

Directory listing for /

- [downloads/](#)
- [sstate-cache/](#)

```
user@yadrolab-wp1:~/tester_ftp$ python3 connect_to_ftp.py
//downloads/Error-0.17029.tar.gz
//downloads/Error-0.17029.tar.gz.done
//downloads/Mako-1.2.4.tar.gz
//downloads/Mako-1.2.4.tar.gz.done
//downloads/Mako-1.3.2.tar.gz
//downloads/Mako-1.3.2.tar.gz.done
//downloads/MarkupSafe-2.1.3.tar.gz
```



Полученный результат (2)

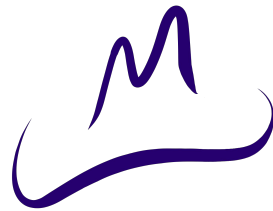
- Модификация Docker образа
 - Исправлены непредвиденные ошибки сборки с прошлой итерации
 - Добавлено автоматическое логирование процесса сборки
 - Написана инструкция для запуска системы

Instructions

Для образа потребуется не менее 90Gb памяти. Процесс сборки займёт более 2 часов.

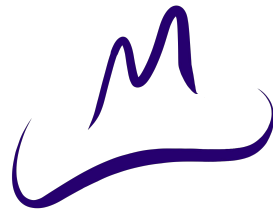
Для выполнения следующих шагов необходим `Docker` и `Docker compose v2` !





Полученный результат (3)

- Произведен первичный анализ логов сборки
 - Изучена скорость пересборки образов без некоторых кэш компонентов - без кэша состояний, без кэша загрузок и вообще без кэша
 - Экспортирован граф зависимостей пакетов
 - Граф проанализирован и построен



Полученный результат (4)

- Реализованы тесты текущих наработок
 - Тестирование docker сборки
 - Тестирование построения графиков
 - Анализ кода с помощью линтера



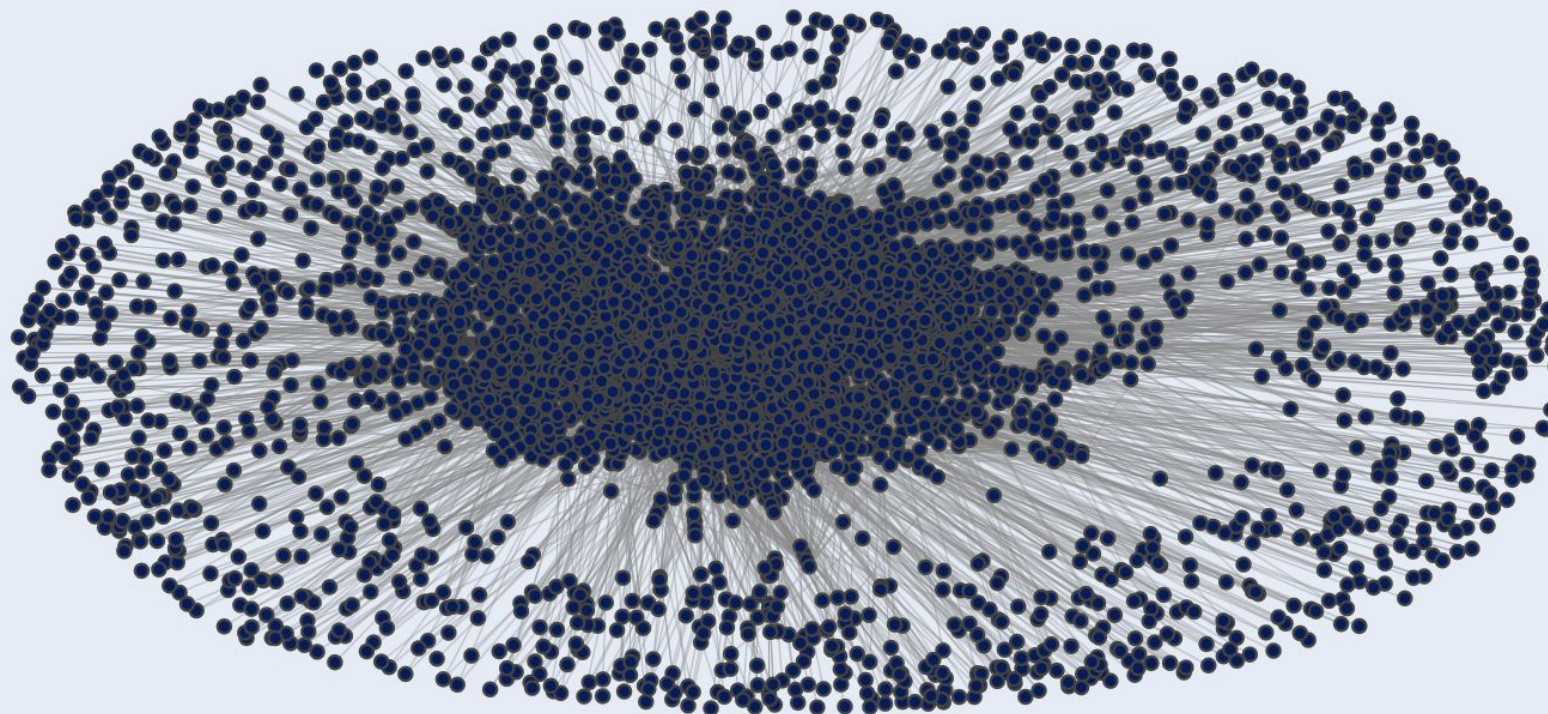
План на следующую итерацию

- Подготовить инструкцию для создания сервера кэша (сервера, который способен распространять кэш локальной сборки по сетевым протоколам)
- Продолжить анализ получаемых при сборке образов данных
- Усовершенствовать и дополнить тесты (покрытие большего количества реализованного функционала - интеграция тестирования в процесс сборки)
- Продолжить автоматизацию (автоматизация процессов кэширования и сборки логов)

Построение графа зависимостей



Network graph made with Python



Конфигурация локаций различного рода кэш



▼ build

> cache

> conf

> experiment_downloads_experiment

> experiment_sstate-cache_experiment

> tmp

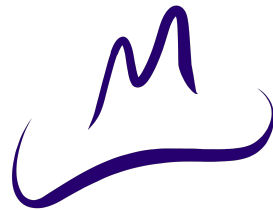
Распространение локального кэша локально



Изучено, как передать кэш одной сборки другой.

На рисунке показаны логи запуска новой сборки с предоставленным кэшем старой сборки - после загрузки кэша становится видно, что до начала активной фазы сборки, в готовом состоянии находятся 56% задач:

```
Sstate summary: Wanted 2013 Local 1138 Mirrors 0 Missed 875 Current 0 (56% match, 0% complete)#####  
Initialising tasks: 100% | #####  
NOTE: Executing Tasks  
NOTE: Tasks Summary: Attempted 4573 tasks of which 1863 didn't need to be rerun and all succeeded.
```



Реализована инструкция по запуску проекта

Instructions

Для образа потребуется не менее 80Gb памяти. Процесс сборки займёт более 2 часов.

Для выполнения следующих шагов необходим `Docker` и `Docker compose v2` ! Проверить соответствующие зависимости можно с помощью скрипта `/src/docker-check.sh` .

Необходимо клонировать репозиторий. Это можно выполнить следующей командой:

```
git clone https://github.com/moevm/os_profiling.git
```



Все дальнейшие шаги происходят в директории `<image_name>-build` . Тут и далее `image_name` может принимать только одно из следующих значений: `yocto` , `buildroot` .

- **Yocto**

i. Чтобы создать образ требуется выполнить команду:

```
docker compose build --build-arg UID=$(id -u) --build-arg GID=$(id -g) --build-arg CORE=$(uname -m)
```

ii. После удачной сборки образа, можно запустить контейнер командой:

```
docker compose up
```



Когда произойдёт автоматическое закрытие контейнера (с кодом 0) -- всё будет установлено.

Третья итерация. Команда MSE 07

