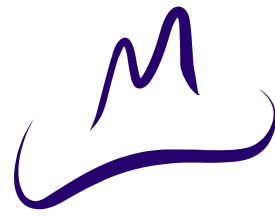


MSE 07

Iteration №4





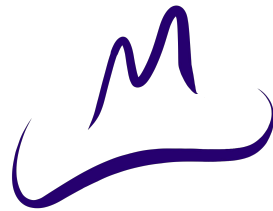
Состав команды

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)



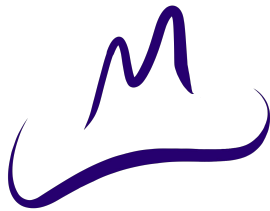
План на текущую итерацию

В итерации 3 было зафиксировано, что планами на текущую являются:

- 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)

Продолжить анализ получаемых при сборке образов данных

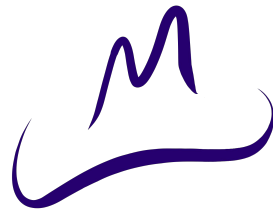
Данный пункт не был принят в работу, поскольку заказчик попросил сосредоточиться на кэшировании и серверах кэша.



Полученный результат (1)

- Реализованы тесты которые проверяют все, что используется при сборке образа, а именно:
 - Тестирование сборки образа виртуального окружения для Yocto
 - Тестирование частичной сборки образа Yocto
 - Тестирование полной сборки образа Yocto
 - Тестирование запуска Qemu

Демонстрации работы тестов



```
SUCCESS: docker is installed.  
SUCCESS: docker compose (v2.25.0) is installed.
```

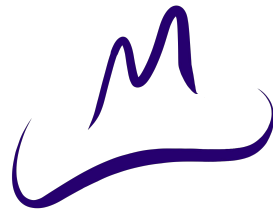
```
[+] Building 2.2s (11/11) FINISHED  
=> [yocto_project internal] load build defin  
=> => transferring dockerfile: 2.24kB  
=> [yocto_project internal] load metadata fo  
=> [yocto_project internal] load .dockerigno  
=> => transferring context: 2B  
=> [yocto_project requirements 1/2] FROM doc  
=> [yocto_project internal] load build conte  
=> => transferring context: 518B  
=> CACHED [yocto_project requirements 2/2] R  
=> CACHED [yocto_project base 1/4] RUN pytho  
=> CACHED [yocto_project base 2/4] RUN rm /b  
=> CACHED [yocto_project base 3/4] WORKDIR /  
=> [yocto_project base 4/4] COPY ./assembly/  
=> [yocto_project] exporting to image  
=> => exporting layers  
=> => writing image sha256:ed8fb0b54b8a22601  
=> => naming to docker.io/library/yocto-imag
```

```
SUCCESS: test is passed.
```

```
STAGE: check image  
STAGE: start container  
[+] Running 1/1  
✓ Container yocto_project Started  
STAGE: start build yocto  
STAGE: stop container  
[+] Stopping 1/0  
✓ Container yocto_project Stopped  
STAGE: check exit code  
STAGE: check dirs  
Required dirs were found  
y.  
Required dirs were found  
y/build_yocto.  
STAGE: check files  
Required files were found  
STAGE: check logs  
SUCCESS: test is passed.
```

```
STAGE: check image  
STAGE: choose check time  
STAGE: start container  
[+] Running 1/1  
✓ Container yocto_project Started  
STAGE: start build yocto  
STAGE: stop container  
[+] Stopping 1/0  
✓ Container yocto_project Stopped  
STAGE: check dirs  
Required dirs were found by path /home/  
y.  
Required dirs were found by path /home/  
y/build_yocto.  
STAGE: check logs  
STAGE: reboot building  
[+] Running 1/1  
✓ Container yocto_project Started  
STAGE: stop container  
[+] Stopping 1/0  
✓ Container yocto_project Stopped  
STAGE: check logs
```

```
SUCCESS: test is passed.
```



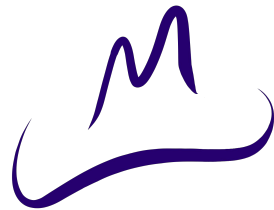
Полученный результат (2)

- Изучены переменные среды сборки BitBake, которые регулируют максимальные объем ресурсов который хотим выделить под сборку
 - BB_PRESSURE_MAX_CPU – максимальная нагрузка процессора
 - BB_PRESSURE_MAX_IO – максимальный порог нагрузки ввода-вывода
 - BB_PRESSURE_MAX_MEMORY – максимальный порог нагрузки на память

Демонстрация потребления ресурсов

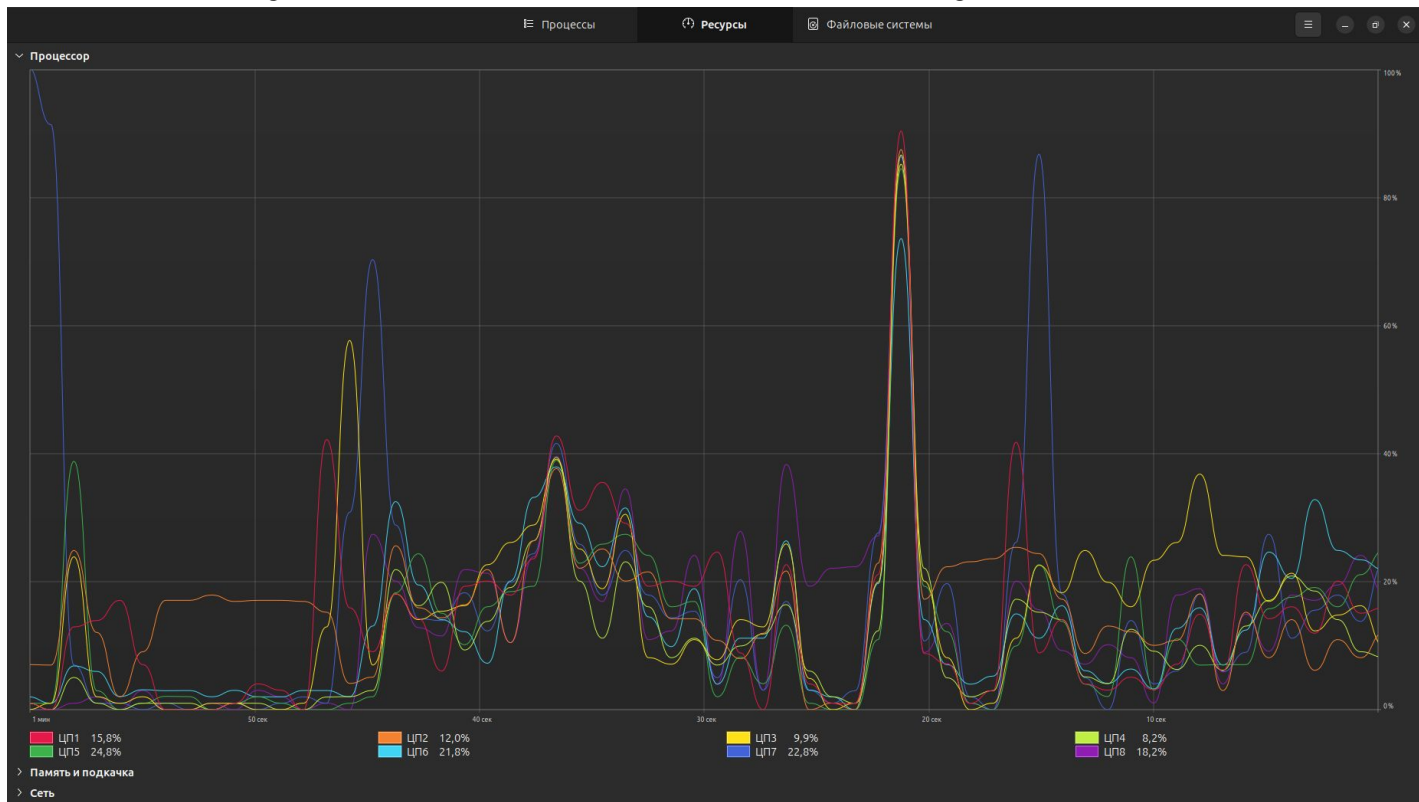
Без ограничения нагрузки – видно, что нагрузка высокая

высокая



Демонстрация потребления ресурсов

Без ограничения нагрузки – видно, что нагрузка ниже



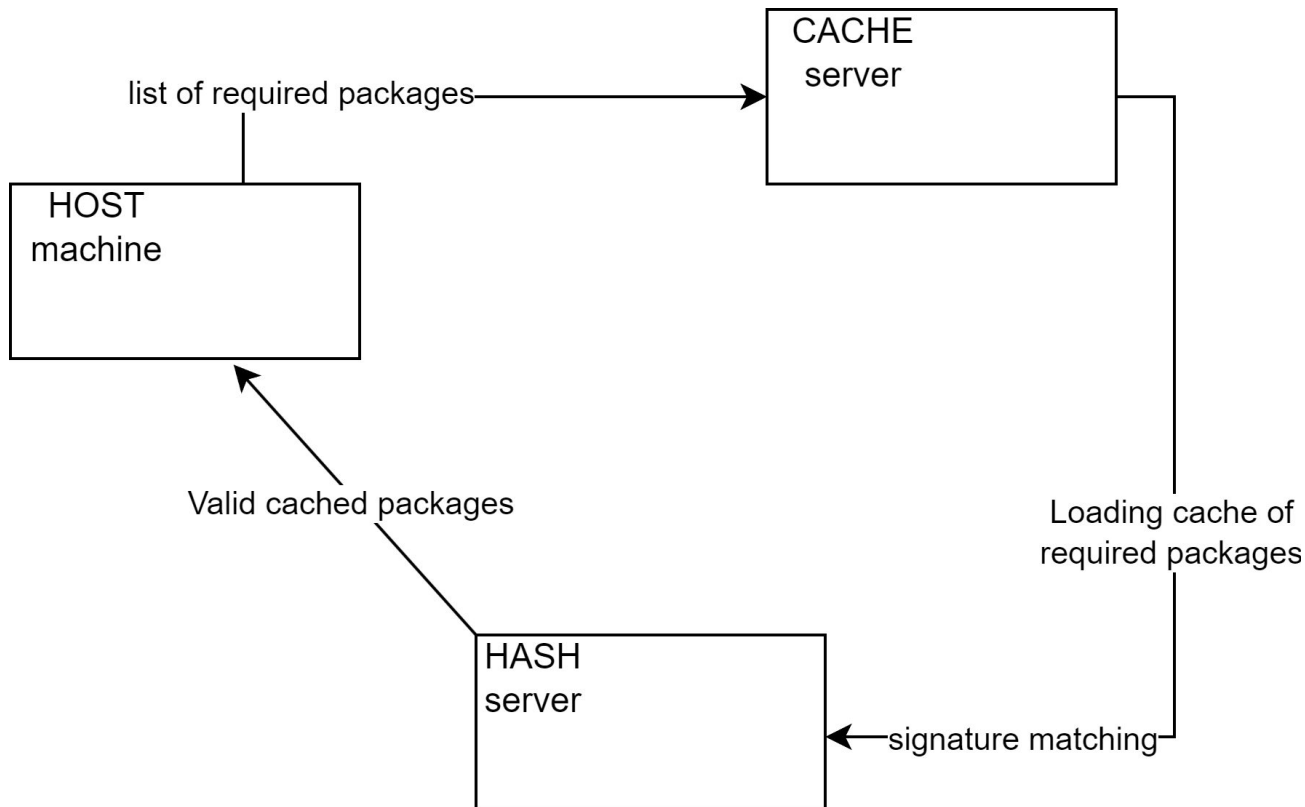
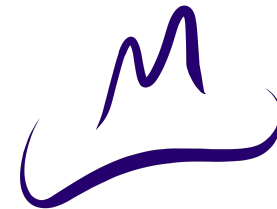


Полученный результат (3)

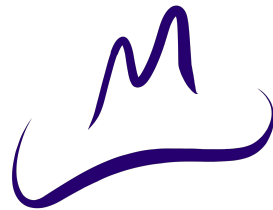
- Изучены принципы работы с кэшем
 - Изучены принципы создания кэш серверов
 - Изучены принципы использования кэш серверов с помощью технологии зеркалирования
 - С помощью сервера кэша автоматизирована сборка
- Изучены принципы работы с хэш сервером
 - Изучены принципы создания хэш серверов
 - Изучены принципы использования хэш серверов в конфигурации конкретной сборки

Ко всему вышеописанному написана инструкция – за счет использования связка кэш + хэш было загружено из кэша 99% задач сборки

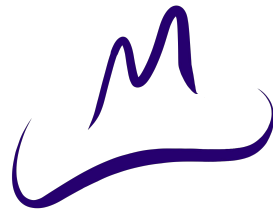
Схема загрузки кэша состояний и пакетов



Оптимизация сборки через использование кэш и хэш сервера (99% match)



```
NOTE: Fetching uninnative binary shim http://downloads.yoctoproject.org/releases/uninnative/4.4/x86_64-nativesdk-libc-4.4.tar.xz;sha256sum=d81c5428
will check PREMIRRORS first)
Checking sstate mirror object availability: 100% |#####| Time: 0:01:41
Sstate summary: Wanted 1863 Local 0 Mirrors 1857 Missed 6 Current 0 (99% match, 0% complete)
NOTE: Executing tasks
Setscene tasks: 260 of 1864
```



Итоги проекта

Опишем в хронологической последовательности полученные результаты

- Каждым участником проекта собран образ Linux
- Изучены утилиты трассировки системных ресурсов
- Проанализированы логи сборки
- Сборка обернута в Docker
- Реализован набор программа для парсинга логов сборки
- По полученным данным построены графики и построен граф зависимостей слоев и пакетов сборки
- Изучен механизм кэширования, описано, как можно создавать сервера кэш и хэш, произведены эксперименты
- Реализована система тестирования сборки

Четвертая итерация. Команда MSE 07

