

Оптимизация проверки задач для Linux* курсов

Волков Д., Заславский М.

Computer Science Center

20 декабря 2017 г.



Разработка модулей ядра Linux

Информация

Отзывы

Содержание

Комментарии

Новости

О курсе

Данный курс посвящен программированию в ядре Linux. Слушатели узнают об архитектуре ядра, научатся разрабатывать драйверы простейших символьных устройств, узнают о

Зачем?

- ▶ Облегчить задачу составителям курсов
- ▶ Отделение проверяющей системы от тестов
- ▶ Сокращение времени ожидания вердикта пользователем

Цели и задачи

Цель проекта: оптимизация структуры и производительности проверяющей системы.

Задачи:

- ▶ Архитектурное разделение проверяющей системы и сценариев проверки отдельных заданий.
- ▶ Профилирование проверки решений
- ▶ Ускорение проверки решений

На старте

Проблема, связанная с взаимосвязью проверяющей системы

Tree: a66f91a985 ▾ [mooc-lp-optimization](#) / [tests](#) / [checker_pathes.rb](#)

 **zmm** initial commit

1 contributor

18 lines (17 sloc) | 419 Bytes

```
1  case TASK_ID
2  when "1"
3      TYPE_PATH = "type1" + "/"
4      CHECKER_SCRIPT_NAME = 'type1_ch.rb'
5  when "1000"
6      TYPE_PATH = "deploy" + "/"
7      CHECKER_SCRIPT_NAME = 'deploy.rb'
8  when "2000"
9      TYPE_PATH = "stub" + "/"
10     CHECKER_SCRIPT_NAME = 'stub.rb'
11  when "kernel_module_1_load"
12     TYPE_PATH = "kernel_module_1_load/"
13     CHECKER_SCRIPT_NAME = 'kernel_checker.rb'
```

⏮ ⏪ ⏩ ⏭ ⏮ ⏪ ⏩ ⏭ 🔍 ↺

На старте

Виртуализация KVM + Libvirt

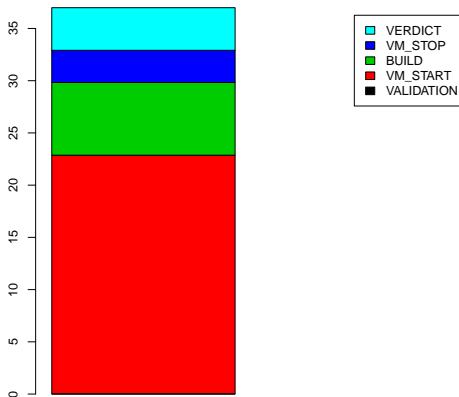
```
[2017-10-27 09:04:12.639634860+03:00][21877]: VERDICT
[2017-10-27 09:04:12.641058603+03:00][21877]: {"submission_id"=>"fb433823-cfe3-463b-bd3a-de1302a4dbab", "fail_reason"=>"make -C /lib/modules/4.9.0-0.bpo.3-amd64/build M=/vagrant modules\nmake[1]: Entering directory '/usr/src/linux-headers-4.9.0-0.bpo.3-amd64'\nBuilding modules, stage 2.\nMODPOST 1 modules\nCC [M] /vagrant/solution.o\nSolution build succeed.\n", "comp_exec"=>"Load module successfully\nModule was successfully removed\n=== Solution logs from printk (only containing \"kernel module loaded\")\nlocalhost kernel: [ 15.918547] [kernel_mooc] Hello world 1.\nOct 27 06:04:05 localhost kernel: [ 15.934098] [kernel_mooc] Bye\nor_message"=>"", "status"=>"VERDICT"}
[2017-10-27 09:04:12.642265318+03:00][21877]: submission_check_statistics:task_id=kernel_module_1_load,submission_bab,execution_time=33.056095142
[2017-10-27 09:04:12.643689996+03:00][21877]: fb433823-cfe3-463b-bd3a-de1302a4dbab check finished, exit code = 0
```

Оптимизация архитектуры

- ▶ Рефакторинг кодовой базы
- ▶ Отделение чекеров от проверяющей системы

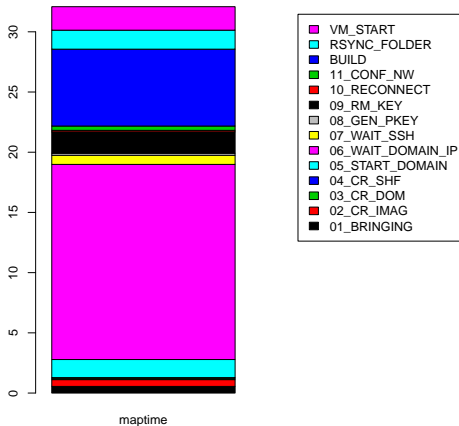
Профилирование

Парсинг логов *pdaemon.sh*



Профилирование

Парсинг логов *pdaemon.sh*



Оптимизация по времени

- ▶ Конфигурирование сетевых настроек загружаемого образа
- ▶ Создание постоянной конфигурации со статическим IP для Libvirt

Технологии

- ▶ ОС (гостевая и хостовая): **GNU/Linux**
- ▶ Виртуализация: **QEMU/KVM, Vagrant + (Docker || Libvirt)**
- ▶ Языки: **Ruby, Python, Bash, R**
- ▶ Система контроля версий: **Git**
- ▶ Удаленный доступ: **SSH**
- ▶ Коммуникация: **Slack**

Результаты

- ▶ Проверяющая система отделена от тестовых сценариев
- ▶ Создание автоматических тестов производительности системы
- ▶ Ускорение проверки!

Перспективы

Проект не завершен в полной мере, ещё есть направления для развития:

- ▶ Интеграция изменений
- ▶ Автоматические тесты производительности могут быть совершеннее
- ▶ Дальнейший рефакторинг кода
- ▶ Потенциально возможно дальнейшее ускорение проверки
- ▶ Автоматизация конфигурирования оптимальных настроек

Ссылки

- ▶ Курс: <https://stepik.org/course/2051>

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻