



ООО «Базальт СПО»

Российский разработчик
операционных систем «Альт»

basealt.ru

Интеграция OpenQA с Proxmoх Virtual Environment

Сергей Иванов

Инженер по тестированию программного обеспечения

zagagyka@basealt.ru

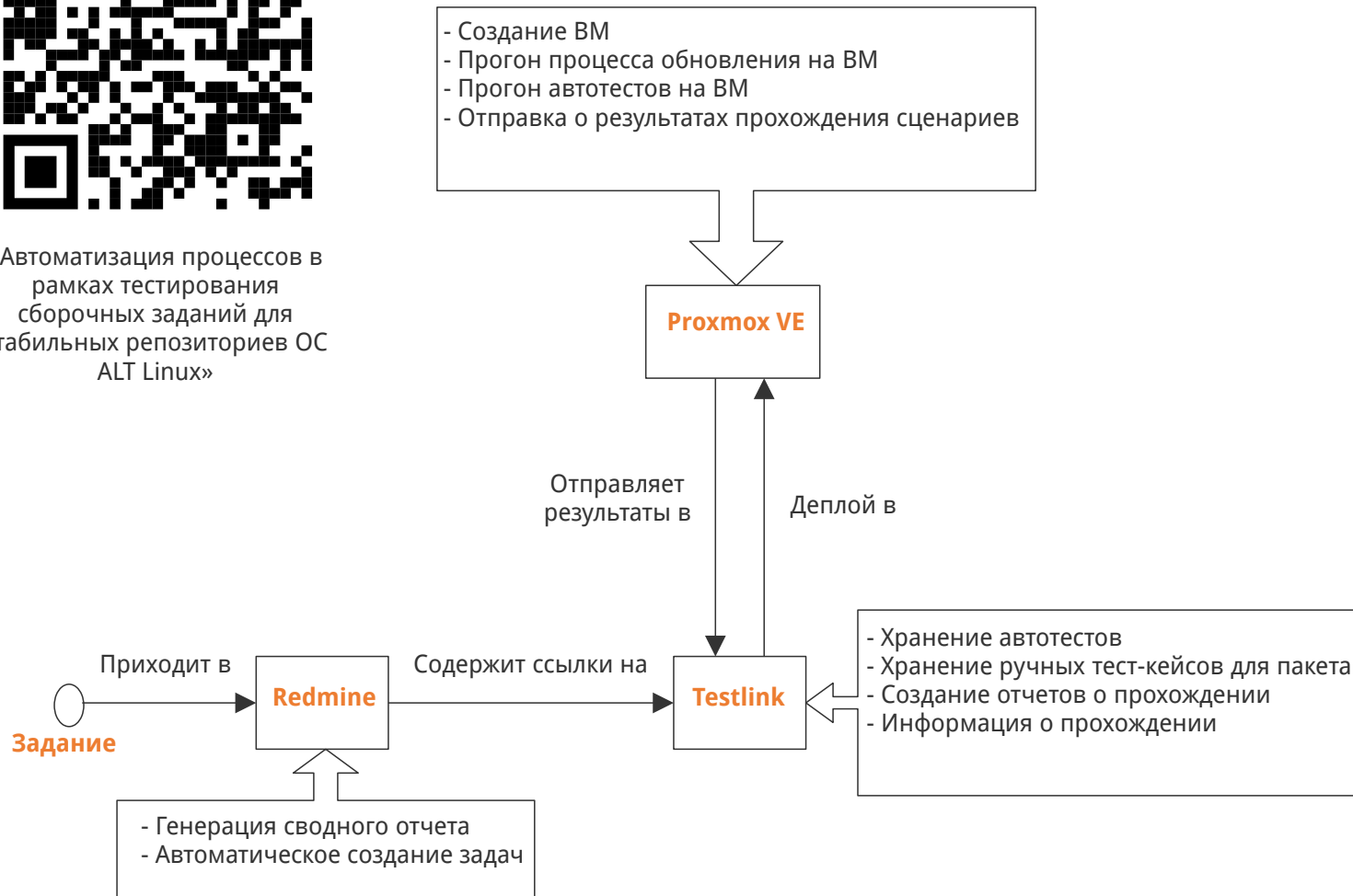




Схема взаимодействия компонентов при тестировании сборочных заданий



«Автоматизация процессов в рамках тестирования сборочных заданий для стабильных репозитивов ОС ALT Linux»



OpenQA

- Открытый тестовый фреймворк
- Запускает собранный образ (.iso) или диск (.qcow2) с установленной системой в виртуальной машине
- Эмулирует нажатия клавиш и движения мыши
- Сравнивает полученные скриншоты с помощью OpenCV





OpenQA

alt education

- [Build20230914-task327699](#) (5 days ago) ✓

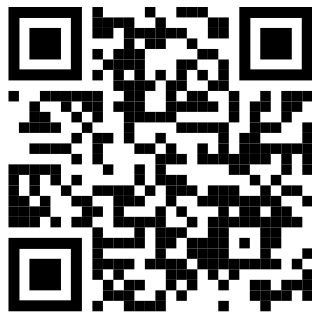
alt kworkstation

- [Build10.2-20230901](#) (7 days ago) ✓
- [Build20230911-task327699-try3-5seksleep](#) (8 days ago) ✓
- [Build20230911-task327699](#) (7 days ago) ✓



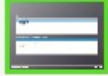



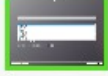
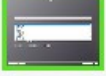
alt server

- [Build20230912-task327699](#) (7 days ago) ✓
- [Build20230906-task328350](#) (8 da
- [Build10.2-beta1](#) (a day ago) ✓

Flavor: DVD



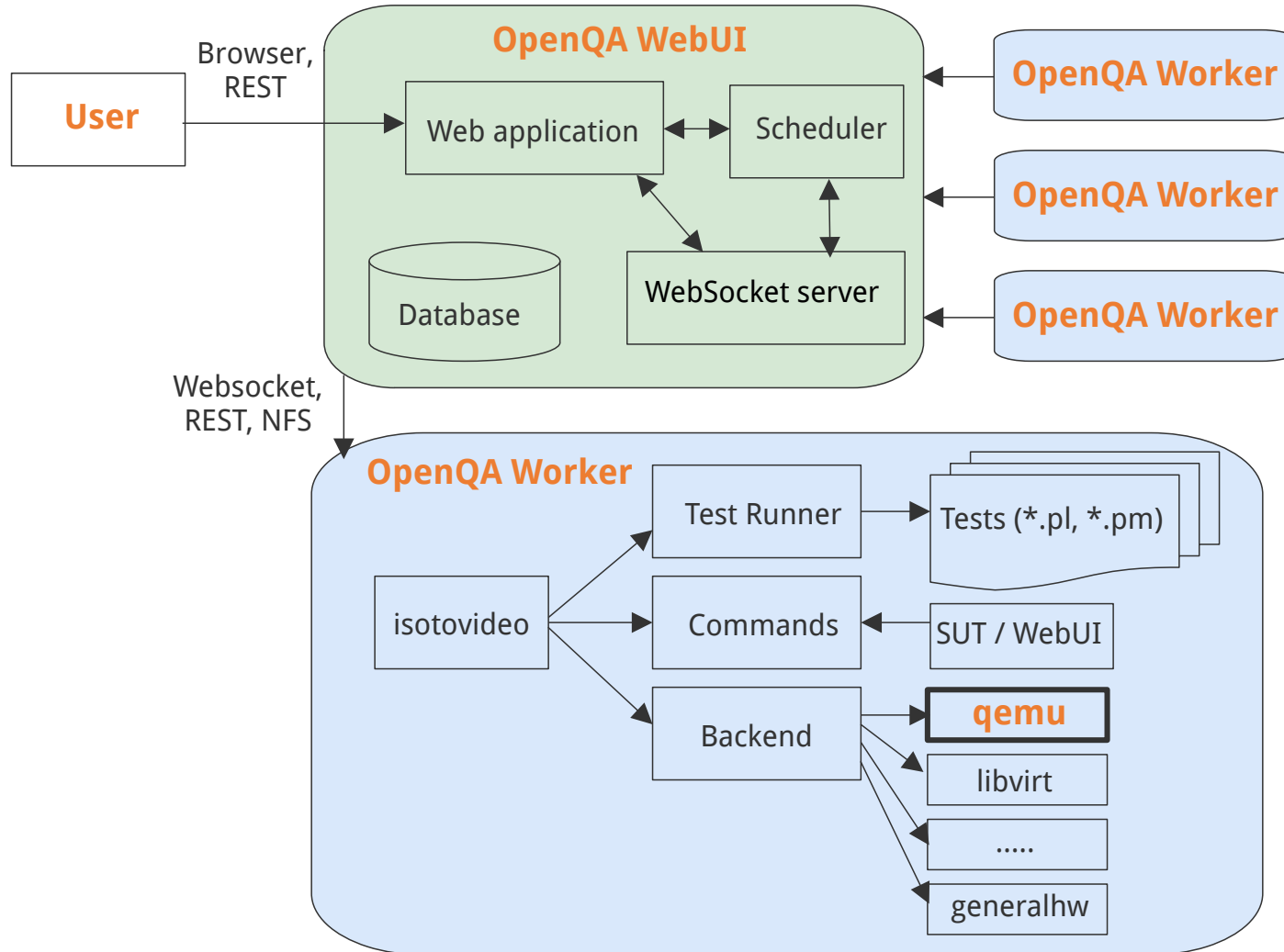
Автоматическое тестирование дистрибутивов ОС Альт с использованием OpenQA

Test	Result	References
install		
Bootloader_start 33s ↗	passed	 
Language 13s ↗	passed	 
License_agreement 16s ↗	passed	 
Date_Time 15s ↗	passed	 

Test	x86_64
BTRFS_ROOT_BPO	↻ ●
BTRFS_ROOT_BPO@64bit_uefi	↻ ●
BTRFS_ROOT_HOME_BPO	↻ ●
BTRFS_ROOT_HOME_BPO@64bit_...	↻ ●
BTRFS_ROOT_HOME_WBP	↻ ●
BTRFS_ROOT_WBP	↻ ●
LUKS_HOME_BPO	↻ ●



Архитектура OpenQA





QEMU backend

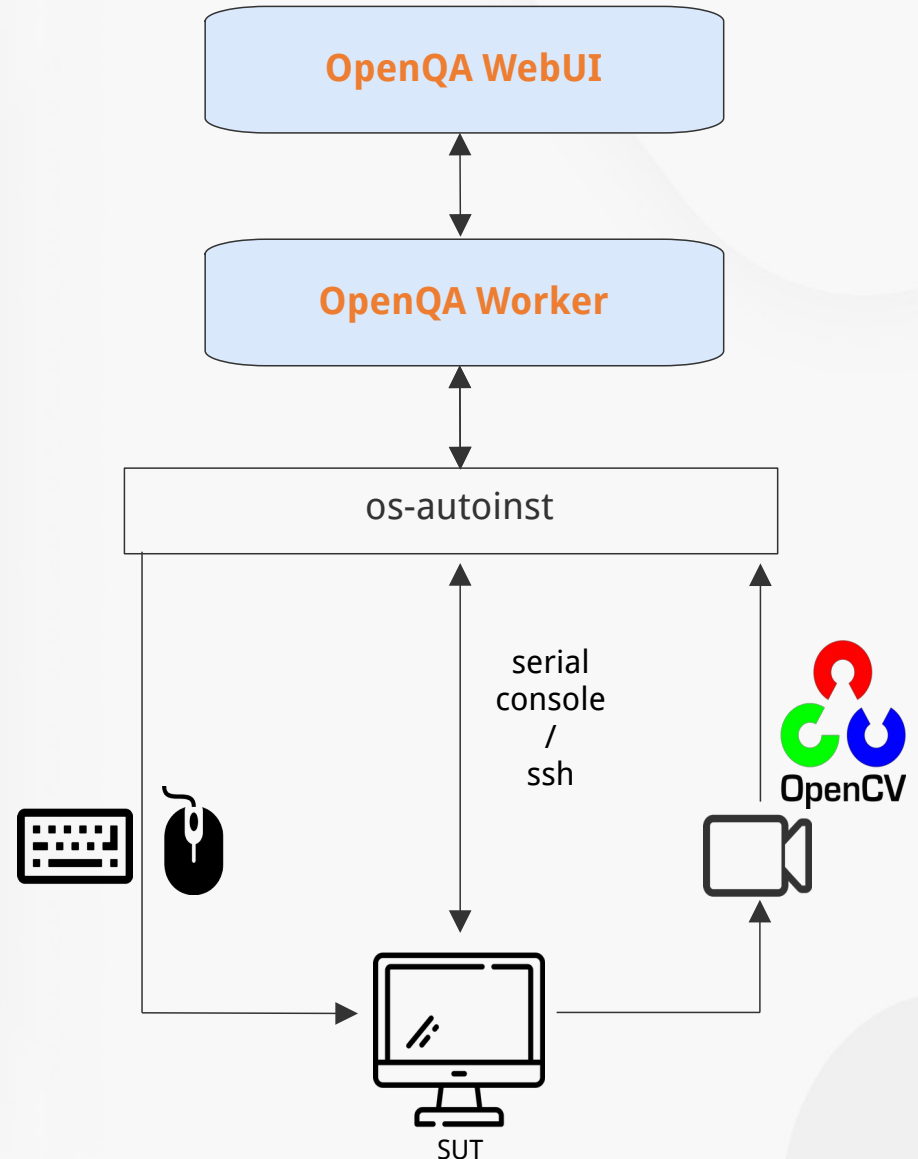
Проблемы при использовании QEMU backend'a

- Под OpenQA Worker требуется отдельный высокопроизводительный сервер.
- Ограниченный функционал менеджмента виртуальных машин
- Сложности при настройке сети в мультимашинных тестах

OpenQA Worker backends

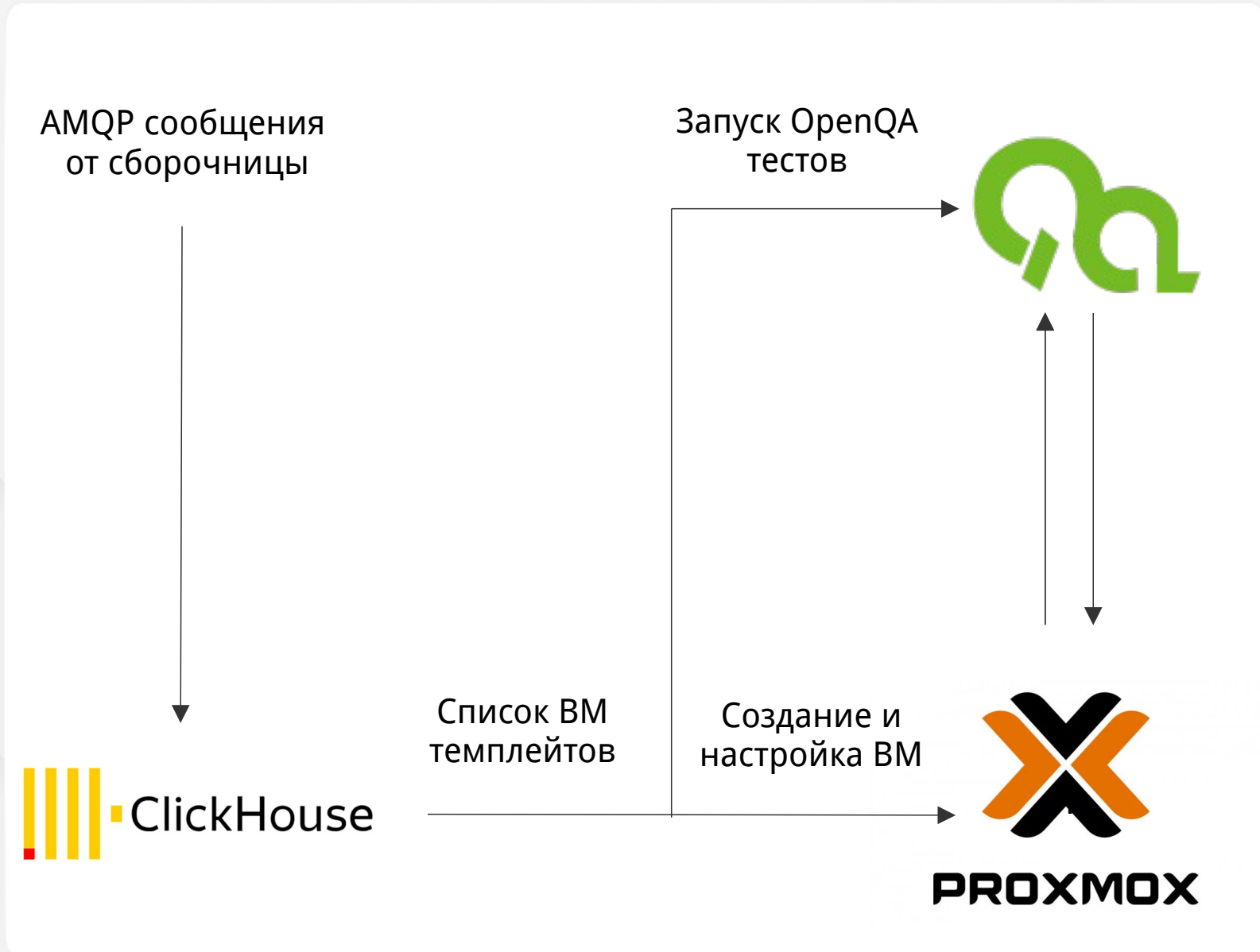
Бекенд generalhw:

- Создает на порядок меньшую нагрузку на OpenQA worker (т. к. только отправляет и обрабатывает команды)
- Подключается к уже созданной виртуальной / реальной машине
- Может включать и выключать машину, **НО** не осуществляет их менеджмент (создание / удаление / настройка)





Общая схема интеграции





Скрипт интеграции

```
$ ./prepare_autotest_list  
<METHOD(["method=<Queue.DeclareOk(['message_count=0', 'queue=qa-task-list'])>"]>  
<METHOD(["method=<Queue.DeclareOk(['message_count=0', 'queue=qa-autotest-list'])>"]>
```

- В очередь **qa-task-list** приходят AMQP сообщения с номером сборочного задания
- Выполняется запрос к БД по данному taskу.
- В результате запроса получаем таблицу содержащую списки бинарных пакетов, тестов и темплейтов виртуальных машин.








branch	binary_name	groupUniqArray(tve_template_id)
p10	rpm_package_1	[3329,3343]
p10	rpm_package_2	[3360]
p10	rpm_package_3	[3363,3329,3360,3361,3343]
p10	rpm_package_4	[3363,3329,3360,3361,3343]

Скрипт интеграции

- Получившийся результат отправляется в очередь **qa-autotest-list** отдельными AMQP сообщениями по каждой строчке

```
{'id': '329785', 'binary_name': ['php8.0-pgsql', 'php8.0-ldap'], 'platform': 3361}  
{'id': '329785', 'binary_name': ['php8.0-devel'], 'platform': 3360}
```

✓ Template

-  3295 (p9-server-9.2-x86-64-office)
-  3296 (p9-education-9.2-x86-64-default)
-  3297 (p9-education-9.2-i586-default)
-  3368 (p10-workstation-10.1-x86-64-efi)
-  3369 (p10-server-10.1-x86-64)
-  3370 (p10-server-10.1-x86-64-efi)
-  3371 (p10-server-10.1-x86-64-minimal)

Темплейты в Proxmox VE



Скрипт интеграции

```
$ ./main.py  
<METHOD(["method=<Queue.DeclareOk(['message_count=0', 'queue=qa-autotest-list'])>"])>
```

- Слушает очередь **qa-autotest-list** и на каждое AMQP сообщение выполняет действия
- Создание виртуальной машины из темплейта
`vmid = cloneVM(binary_name, task, vmIdTemplate)`
- Настройка виртуальной машины
`futures = executor.submit(configAT, vmid, binary_name)`

```
{  
  "data": {  
    "vga": "none",  
    "args": "-vnc 0.0.0.0:42228 -device VGA,edid=on,xres=1024,yres=768"  
  }  
}
```



Скрипт интеграции

- 32978500 (autotest-list-ts1-329785-created-p10-kworkstation-10.1-x86-64) pveapi
- 32978501 (autotest-list-ts1-329785-created-p10-education-10.1-i586) pveapi
- 32978502 (autotest-list-ts1-329785-created-p10-workstation-10.1-i586) pveapi

Созданные виртуальные машины в Proxmox VE

- Выполнение запуска OpenQA тестов

```
$ openqa-cli api -X POST jobs \  
  DISTRI=kworkstation          VERSION=10  
  FLAVOR=HDD                   ARCH=x86_64  
  MACHINE=64bit                BUILD=329785  
  GENERAL_HW_VNC_DEPTH=24     GENERAL_HW_VNC_IP=<pve_ip>  
  GENERAL_HW_VNC_PORT=48128  GENERAL_HW_NO_SERIAL=1  
  SUT_IP=<VM_IP>              TEST=php8.0-ldap  
  VIDEO_STREAM_PIPE_BUFFER_SIZE='1024*768*0+0'
```



Найденные в процессе разработки проблемы

- Настройка VNC подключения (vnc color depth)
- Настройка serial console (/dev/sshserial вместо /dev/ttyS0)

```
# wait_serial expected: "apt-get update; echo 0EtDD-\$?-"  
# Result:  
apt-get update; echo 0EtDD-$?-
```

Выполнение команды: assert_script_run("apt-get update");

```
rm -f /dev/sshserial; mkfifo /dev/sshserial; chmod 666 /dev/sshserial;  
tail -fn +1 /dev/sshserial
```

- Поведение курсора мыши (не выходит за край экрана, в отличии от qemu backend'a)



Преимущества использования связки ProxmoX VE и OpenQA

- Увеличение общей скорости прохождения тестов
- Скорость работы сети стала в разы быстрее
- Возможность сохранения виртуальной машины с не прошедшими тестами
- Более простое конфигурирование виртуальных машин



Дальнейшее развитие

- Связать логи и результаты тестов OpenQA с TestLink'ом
- Подключение с помощью generalhw к реальным машинам



Спасибо за внимание