

Technická dokumentácia

Meno: Ján Nemčík

Predmet: APS

Ak. rok: 2019/2020

Obsah

- [1. Zadanie projektu](#)
- [2. Návrh](#)
- [3. Implementácia](#)
- [4. Testovanie](#)
- [5. Používateľská príručka](#)

1. Zadanie projektu

Zadaním tohto projektu bolo otestovanie konkrétneho scenáru na rôznych typoch virtualizácií a porovnať výsledky medzi jednotlivými typmi. Pre tento účel som si ako scenár zadefinoval build linux kernel-u. V tomto projekte som použil definíciu pre zariadenie [Odroid X-2](#). Pôvodným zámerom bolo buildenie kernelu pre debian, čo sa ale ukázalo pri softvérovej virtualizácii ako nevhodný scenár, nakoľko trval rádovo dni. Taktiež som mal v úmysle buildovať kernel na nasledujúcich typoch virtualizácie:

- hardvérová
- softvérová
- paravirtualizácie
- kontajnerizácia

Z týchto typov som ale vylúčil paravirtualizáciu, nakoľko by bolo potrebné zakomponovať softvér, ktorý paravirtualizáciu podporuje do grub loaderu môjho počítača, čo som považoval za príliš nebezpečné.

2. Návrh

Scripty som sa snažil navrhnuť tak, aby boli čo najviac flexibilné a zároveň čo najjednoduchšie. Z tohto dôvodu som sa rozhodol použiť *Shell*, nakoľko je použiteľný cross viaceré linux distribúcie. Cieľom bolo taktiež nutnosť inštalácie ďalších nadbytočných balíčkov a programov, preto som defaultne použil [KVM](#) na provisioning virtuálnych strojov, ktorý je natívne súčasťou kernelu.

Získavanie výsledkov

Nakoľko scripty budú bežať v rôznych prostrediach, čo znamená, že budú spúšťané priamo na konkrétnych virtuálnych strojoch, poprípade ako docker container-y, tak som vytvoril jednoduchú aplikáciu na verejne prístupnom serveri, kde sú posielané výsledky zo všetkých vykonaných buildov kernelu. Viac info v sekcii [používateľskej príručky](#).

3. Implementácia

4. Testovanie

5. Používateľská príručka

Tento projekt pozostáva z nasledujúcich scriptov:

- [build-kernel.sh](#)
- [build-docker.sh](#)
- [run-docker.sh](#)
- [get-result.sh](#)
- [virt-install.sh](#)

Každý z vyššie vymenovaných scriptov, okrem *build-docker*, obsahuje svoj vlastný *help*, a preto budú v nasledujúcich podsekciiach iba zhrnuté ich úlohy, *help* je možné vykonať nasledovne: `<script_name> -h|--help`.

build-kernel.sh

Tento script nainštaluje potrebné dependencies pre build kernelu a pre spracovanie výsledkov. Súčasťou scriptu je taktiež stiahnutie a rozbalenie samotného kernelu. Po týchto krokoch sa spustí samotný build pomocou príkazu `make`. Po úspešnom vykonaní buildu sa odstránia stiahnuté súbory a výsledky sa odošlú na server. Používateľ má taktiež možnosť po vykonaní buildu odstrániť všetky doinštalované dependencies (viac v *help-e* scriptu). Script taktiež umožňuje definovať počet jadier, na ktorých build prebiehať (viac v *help-e* scriptu). **Script je potrebné spúšťať ako `sudo user`, poprípade s userom, ktorý má práva inštalovať dependencies ako root!**

Inštalované dependencies:

```
build-essential
libncurses-dev
bison
flex
libssl-dev
libelf-dev
bc
time
curl
jq
wget
```

build-docker.sh

Tento script builduje docker image z Dockerfile v aktuálnom adresári a taguje ho ako `aps:latest`. Využitelný v prípade, že nie je dostupný image z [Docker hub-u](#), je možné po zbuildovaní spustiť container priamo lokálne pomocou príkazu `docker run` alebo spustením scriptu `run-docker.sh`.

run-docker.sh

Tento script zabezpečuje spustenie docker containera podľa predpripravenej definície, ktorá už obsahuje všetky potrebné dependencie aj samotný priečinok s kernel modulmi. Pri spustení containera sa automacitky spustí build kernelu. Po ukončení sa výsledky odošlú na server a container sa zastaví a vymaže. Script obsahuje možnosť `-r --results` s hodnotami *auto* alebo *load*, kde *auto* znamená, že výsledky sa automaticky zobrazia pre posledný vykonaný build v docker containeri. Možnosť *load* hovorí o tom, že používateľ chce výsledky zobrazit manuálne, viac informácií o možnostiach scriptu na sa nachádzajú v *help*-e.

get-result.sh

Tento script slúži na získanie výsledkov buildov kernelu v rámci projektu. Ponúka rôzne možnosti filtrovania a formátovania výstupu. Všetky prípady použitia sú popísané v *help*-e scriptu. **Dependencie potrebné pre získanie výsledkov boli nainštalované v rámci scriptu build-kernel.sh a môžu byť donštalované vykonaním príkazu `./get-result.sh -i`, ktorý musí byť spustený ako sudo príkaz.**

virt-install.sh

Tento script zabezpečí inštaláciu KVM virtuálneho stroja s Debian 10 OS. Všetky možnosti použitia sú popísané v *help*-e scriptu.

<https://github.com/JanNemcik/aps-project>

<https://hub.docker.com/r/jany15/aps>