



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

TESTOVACÍ SADA SLOUŽÍCÍ PRO ANALÝZU TUNED PROFILŮ

TESTSUITE FOR ANALYSIS OF PROPERTIES OF TUNED PROFILES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

BRANISLAV BLAŠKOVIČ

VEDÚCI PRÁCE

SUPERVISOR

Ing. ALEŠ SMRČKA, Ph.D.

BRNO 2013

Abstrakt

Výtah (abstrakt) práce v slovenskom jazyku.

Abstract

Výtah (abstrakt) práce v anglickom jazyku.

Klíčová slova

tuned, testování, linux, fedora

Keywords

tuned, testing, linux, fedora

Citácia

Branislav Blaškovič: Testovací sada sloužící pro analýzu Tuned profilů, bakalářská práce, Brno, FIT VUT v Brně, 2013

Testovací sada sloužící pro analýzu Tuned profilů

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval sám pod vedením pána...

.....

Branislav Blaškovič

11. mája 2013

PodĎakovanie

PodĎakovanie.

© Branislav Blaškovič, 2013.

Táto práca vznikla ako školské dielo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

1	Úvod	3
2	Popis komponenty tuned	4
2.1	História tuned	4
2.2	Tuned profily	4
2.2.1	Predkonfigurované profily	4
2.2.2	Manuálne vytváranie profilov	5
3	Emulácia diskových zariadení a sieťových služieb	7
3.1	Sieťové služby	7
3.2	Diskové zariadenia	7
3.2.1	Emulácia bez využitia virtuálneho systému	8
3.2.2	Emulácia s využitím virtuálneho systému	8
4	Testovanie diskových zariadení a sieťových služieb	10
4.1	Testy pre sieťové služby	10
4.2	Testy diskových zariadení	11
5	Plán testovania pre Fedora Linux	15
5.1	Úvod do testovania	15
5.2	Riziká testovania výkonu na virtualizovanom stroji	17
5.2.1	Výhody	17
5.2.2	Nevýhody	17
5.2.3	Zhrnutie	17
6	Príprava prostredia a implementácia testov	18
6.1	Nastavenia systému	18
6.2	Implementácia testov	19
6.2.1	Riešenie problému so zlyhaním testu	20
7	Výsledky testovania	21
7.1	Automatizované spracovanie výsledkov	21
7.2	Vyhodnotenie testov	21
7.2.1	Testovanie s profilom <i>balanced</i>	21
7.2.2	Testovanie s profilom <i>latency-performance</i>	22
7.2.3	Testovanie s profilom <i>powersave</i>	23
7.2.4	Testovanie s profilom <i>throughput-performance</i>	24
7.2.5	Testovanie s profilom <i>virtual-guest</i>	25

8	Bočné produkty práce	27
8.1	Problém s <i>libvirt</i>	27
8.2	Power Management Test Day	27
8.3	Nájdené chyby	28
9	Záver	29
9.1	Najrýchlejší súborový systém a najväčšie zrýchlenie	29
9.2	Ďalší možný rozvoj testovania	29
A	Obsah CD	31

Kapitola 1

Úvod

Každý linuxový server alebo osobný počítač môže slúžiť na niečo iné. Preto je veľmi náročné vytvoriť linuxovú distribúciu, ktorá by pokrývala požiadavky každého a zároveň bola optimalizovaná pre všetky operácie. Preto je potrebné systém nastaviť tak, aby presne vyhovoval našim potrebám a získali sme maximálny výkon pre naše potreby. Keďže sa jedná a množstvo druhov nastavení, vznikol balíček *tuned* [7], ktorý ich zahrňuje.

Cieľom tejto práce je priblížiť *tuned*, zhrnúť jeho hlavné funkcie, popísať profily, implementovať sadu testov pre diskové operácie nad najpoužívanejšími súborovými systémami a diskovými zariadeniami. Taktiež si popíšeme možnosti už existujúcich nástrojov na testovanie a popíšeme spôsob emulácie diskových zariadení.

Do práce bude zahrnutých aj niekoľko bočných tém, ktoré nie sú síce predmetom práce ale úzko súvisia s *tuned* a jeho testovaním, ako napríklad testovací deň *Fedora 19*.

V kapitolách bude postupne najprv predstavené *tuned*, jeho história, náhľad do profilov aj popísaná možnosť vytvorenia vlastného profilu. Nasledujúca kapitola sa zaoberá možnosťami emulácie diskových zariadení a sieťových služieb. Táto kapitola uvádza základné prístupy k problematike, ako postupovať v prípade emulácie na virtualizovanom systéme, ale aj bez neho. Ďalej sa táto práca bude zaoberať testovaním diskových zariadení a sieťových služieb. Uvedieme si niekoľko existujúcich nástrojov, príklad ich použitia a krátky popis, na čo presne slúžia. Potom si popíšeme plán testovania, aké sú riziká pri testovaní, čo budeme testovať a aké výsledky sa budeme snažiť dosiahnuť. Nasleduje príprava testovacieho prostredia a implementácia testov. Obrázkami a textom bude vysvetlené fungovanie testovacej sady a jej prepojenosť s automatickým spracovaním výsledkov. Ďalej práca obsahuje popis automatického spracovania výsledkov, ich vyhodnotenie v tabuľkách a prirovnaním nameraných hodnôt k jednotlivým profilom a ich nastavení. Počas testovania som narazil na niekoľko chýb, ktoré sú zhrnuté v tabuľke a taktiež popísaný problém s *libvirt*, na ktorý som narazil. Na záver si zhrnieme výsledky testov, určíme najefektívnejší profil pre optimalizáciu diskových operácií a popíšeme možnosti ďalšieho pokračovania.

Kapitola 2

Popis komponenty tuned

Balíček *tuned* je primárne napísaný pre linuxovú distribúciu Fedora [4] a Red Hat Enterprise Linux. Démon *tuned* neustále beží, skenuje systém a upravuje nastavenia podľa potreby. Napríklad najväčšia záťaž na disk je pri štarte systému alebo pri ukladaní dát na disk (napríklad filmov). Inak je disk skoro nečinný. *tuned* dokáže optimalizovať zápis práve v tej dobe, keď je to potreba. Rovnako je to aj pri sieťových operáciach.

Súčasťou *tuned* je aj program *tuned-adm*, ktorý nám dovoľuje prepínať medzi profilmi. Každý z profilov slúži na iné zameranie a napriamo podľa toho upravuje systém, čím dosahujeme ešte lepšie výsledky. Tieto profily sú bližšie popísané v sekcii 2.2.

2.1 História tuned

Komponenta *tuned* je vyvíjaná od roku 2008. Prvými autormi boli *Philip Knirsh* a *Thomas Woerner*. Dnes sú najväčšími prispievateľmi *Ján Včelák*, *Jaroslav Škarvada* a *Ján Kaluža*. *tuned* je aktuálne vo verzii 2. Medzi verziami 1 a 2 je veľký rozdiel, pretože bol celý kód od základov prepísaný. Prechod na túto verziu nastal len nedávno, približne s vydaním operačného systému *Fedora 17*. Jedna z najväčších zmien je v používaní *D-BUS* ¹. Ďalšia zmena je v profiloch. Niektoré profily boli zmenené alebo odobraté. Pre zachovanie spätnej kompatibility vznikol preto balík *tuned-profiles-compatible*, ktorý obsahuje všetky profily z verzie *tuned 1*.

2.2 Tuned profily

Profily sú hlavne zamerané na optimalizáciu CPU, diskov, sieťových zariadení a FSB ². Samotný balíček obsahuje niekoľko predpripravených profilov a ako základný profil je po spustení démona profil *balanced*.

2.2.1 Predkonfigurované profily

Profily *tuned* sa nachádzajú v adresári `/usr/lib/tuned`. V tomto adresári sa taktiež nachádza súbor s funkciami, ktoré tieto profily využívajú. Práve tieto súbory sú najviac vyvíjané a menené.

¹Systém pre jednoduché zasielanie správ a komunikáciu medzi aplikáciami

²Front-Side Bus - datová zbernica, ktorá zaisťuje komunikáciu medzi CPU a hardvérom. Využíva sa v procesoroch *Intel*

Prehľad profilov zo základného balíčka *tuned*. Zoznam je platný pre verziu **tuned-2.2.2-1** na *Fedora 18*:

- **balanced** – predvolený profil pre väčšinu systémov s výnimkou virtuálnych
- **latency-performance** – znižuje latenciu systému
- **powersave** – na zníženie odberu energie
- **throughput-performance** – vylepšuje celkovú priepustnosť systému
- **virtual-guest** – predvolený profil pre virtuálne systémy
- **virtual-host** – vhodný pre systémy, na ktorých sa prevádzkujú virtuálne systémy

Balíček *tuned-profiles-compat* rozširuje zoznam o tieto ďalšie profily:

- **default**
- **desktop-powersave**
- **enterprise-storage**
- **laptop-ac-powersave**
- **laptop-battery-powersave**
- **server-powersave**
- **spindown-disk**

Medzi najčastejšie operácie profilov patrí menenie governoru³ procesoru medzi **ondemand**⁴ a **performance**⁵. Viac o tejto vlastnosti je možné dočítať sa na wiki stránkach Arch Linuxu [1].

Ďalej je to nastavovanie plánovačov diskov. Toto nastavenie sa mení v súbore `/sys/block/<dev>/queue/scheduler`. Niektoré profily ho prepínajú z predvoleného plánovača na *deadline*⁶. Viac o plánovačoch diskov je možné sa dočítať na stránkach dokumentácie *OpenSuse* [2] alebo v článku [9].

Niektoré profily taktiež vypínajú bariéry⁷ pri pripojovaní diskov.

2.2.2 Manuálne vytváranie profilov

Profily si môžeme aj sami vytvárať. Ak si nie sme istý, čo je potrebné upraviť, môžeme využiť odporúčania z programu *powertop* [5] a za pomoci skriptu *powertop2tuned* si nechať profil vytvoriť automaticky.

Vytvorenie takéhoto profilu a jeho čiastočný obsah je na obrázku 2.1.

³Rýchlosť procesora je možné meniť a tým šetriť energiu v čase, keď ho naplno nevyužívame. Túto rýchlosť ovplyvňujú rôzne druhy plánovačov.

⁴Nastavuje rýchlosť procesora podľa využitia.

⁵Nastaví rýchlosť procesora na najvyššiu hodnotu bez ohľadu na využitie.

⁶Tento plánovač sa používa pre zníženie latencie. Obsahuje dve fronty a každá požiadavka má deadline.

⁷Vypnutím bariér hrozí poškodenie dát pri odpojení napájania diskov, pokiaľ disk nemá záložnú batériu.

```
# /bin/powertop2tuned test-profile
Generating shell script /etc/tuned/test-profile/script.sh
Generating Tuned config file /etc/tuned/test-profile/tuned.conf
# cat /etc/tuned/test-profile/tuned.conf
# Automatically generated by powertop2tuned tool

[main]
include=/usr/lib/tuned/virtual-host/tuned.conf

[powertop_script]
type=script
replace=1
script=script.sh
# cat /etc/tuned/test-profile/script.sh
#!/bin/sh

. /usr/lib/tuned/functions

start() {
    # NMI watchdog should be turned off
    #echo '0'>'/proc/sys/kernel/nmi_watchdog';

    # VM writeback timeout
    #echo '1500'>'/proc/sys/vm/dirty_writeback_centisecs';
    ...
}
```

Obrázok 2.1: Príklad vytvorenia profilu za pomoci powertop2tuned

Z obrázka je vidieť, že takto vytvorený profil sa už nenachádza v zložke `/usr/lib/tuned` ale v `/etc/tuned`. Taktiež je potrebné si po vytvorení profil otvoriť a ručne skontrolovať, pretože má všetky nastavenia zakomentované a je na používateľovi, ktoré povolí.

Takýto profil si môžeme vytvoriť aj ručne, bez použitia nástroja *powertop2tuned*, pokiaľ presne vieme, čo by bolo vhodné upraviť.

Kapitola 3

Emulácia diskových zariadení a sieťových služieb

V tejto kapitole sa budem venovať emulácií diskových zariadení a sieťových služieb v teoretickej rovine. Je potrebné si uvedomiť, že na to, aby bolo testovanie na 100 % validne nám emulácia nestačí a bolo by potrebné všetky testovacie prostriedky mať k dispozícii, čo nie je možné.

Aj keby tomu tak bolo, program nie je možné dokonale otestovať, pretože väčšinou existuje nekonečne veľa možností vstupu. V prípade testovania hardvéru sa táto skutočnosť ešte násobi, keďže na neho môže vplývať ešte viac okolností ako na softvér. Tejto myšlienke sa venuje aj *Ron Patton* v Knihe *Software Testing* [8], kde opisuje testovanie obvyčajnej kalkulačky.

3.1 Sieťové služby

Ak chceme testovať rôzne typy sietí, *Linux* nám k tomu poskytuje sieťové skripty, ktorými dokážeme vytvárať pomerne zložité návrhy sietí v rámci jedného systému – napríklad mosty medzi zariadeniami, virtuálne podsiete a podobne.

V prípade, že využijeme virtuálne systémy, dajú sa vytvoriť aj pokročilejšie situácie a pri testovaní môžeme využívať služby na rôznych strojoch. Na webový, emailový a DNS server môžeme použiť oddelené systémy a testovať ich navzájom.

Existuje množstvo hardvérových riešení pre emuláciu sietí. Tieto zariadenia ale nie sú relevantné pre komponentu *tuned*, pretože vyžaduje operačný systém *Fedora* alebo jemu podobné. Pri týchto riešeniach sa jedná ale väčšinou o rôzne vstavané systémy.

Testovanie sietí je veľmi obsiahla téma a nie je zameraním tejto bakalárskej práce.

3.2 Diskové zariadenia

K emulácii diskových zariadení sa dá pristupovať dvoma spôsobmi.

1. Bez využitia virtuálneho systému
2. S využitím virtuálneho systému

3.2.1 Emulácia bez využitia virtuálneho systému

Ak nechceme využiť virtuálny systém, potrebujeme vytvoriť blokové zariadenie diskou bez toho, aby sme v skutočnosti pripojili fyzický disk. K tomu slúži napríklad balíček `scsi-target-utils`, ktorý obsahuje nástroj `/usr/bin/tgtadm`. Tento nástroj dokáže vytvoriť *SCSI* cieľový disk aj zo súbora.

Ďalej budeme potrebovať spustiť službu `tgtd`, ktorá nám takto pridaný disk sprístupní na sieť. Tento prístup sa používa vtedy, keď potrebujeme takýto disk nazdieľať po lokálnej sieti. Samozrejme, môžeme ho pripojiť aj s využitím *localhost* PC.

Keď máme takýto disk prichystaný, pripojíme ho za pomoci príkazu uvedeného na obrázku 3.1. Tento nástroj sa nachádza v balíčku `iscsi-initiator-utils`.

```
/usr/sbin/iscsiadm -m discovery -t st -p 127.0.0.1
```

Obrázok 3.1: Vyhľadávanie diskov na sieti pomocou `iscsiadm`

Pre korektnosť pripojenia je vhodné premazať všetky nepoužívané mapy zariadení. Na to nám posluží nástroj `/sbin/multipath` z balíčka `device-mapper-multipath`. Aby sme ho mohli využiť, je taktiež potrebné zaviesť do jadra modul `dm-multipath`. Následne je potrebné reštartovať celú službu `multipathd`. Celý postup je uvedený na obrázku 3.2.

```
modprobe dm-multipath
/sbin/multipath -F
/etc/init.d/multipathd restart
multipath -ll
```

Obrázok 3.2: Vyhľadanie zariadení pomocou `multipathd`

Keď som testoval tento typ pripojenia diskov, narazil som na problém s načasovaním. Je nevyhnutné nechať medzi príkazmi krátke časové intervaly na to, aby všetko prebehlo korektne. Ak sa tieto príkazy spustia za sebou, je vysoká pravdepodobnosť, že sa disky nenájdu a nepripoja.

3.2.2 Emulácia s využitím virtuálneho systému

S virtuálnym systémom sa situácia zjednodušuje. Virtualizácia je v dnešnej dobe veľmi pokročilá. Ak sa rozhodneme používať *libvirt* a *qemu-kvm*, tak existuje viacero nástrojov, ktoré nám uľahčujú prácu s virtuálnymi systémami.

Pre tých, ktorí majú radi klikacie nástroje, existuje skoro dokonalý *virt-manager*. Vytvoriť si v ňom virtuálny systém a podrobne si ho nastaviť je otázka niekoľkých ťahov myšou.

Automatické testy ale nemôžu používať klikacie GUI nástroje. Preto existuje nástroj `/bin/virsh` z balíka `libvirt-client`, ktorý dokáže zjednodušene pracovať s virtuálnymi systémami z príkazovej riadky.

Nastavenia virtuálneho stroja cez *libvirt* sa ukladajú ako *XML*¹ dokument. Tieto konfiguračné súbory sú uložené štandardne v zložke `/etc/libvirt/qemu` a majú ľahko pochopiteľnú štruktúru.

Ak chceme pridať nový disk, tak pridáme nového potomka k uzlu `<devices>`, ktorý môže vyzeráť napríklad ako na obrázku 3.3.

```
<disk type='file' device='disk'>
  <driver name='qemu' type='raw'/>
  <source file='/home/user/Virtuals/F18.img'/>
  <target dev='sda' bus='sata'/>
  <address type='drive' controller='0' bus='0' target='0' unit='0'/>
</disk>
```

Obrázok 3.3: Príklad diskového zariadenia pre libvirt

Niektoré typy zariadení je možné pripájať dokonca aj počas behu virtuálneho systému. Na to sa používa príkaz uvedený v 3.4.

```
/bin/virsh attach-device <názov stroja> súbor.xml
```

Obrázok 3.4: Pripojenie zariadenia bez priamej úpravy XML súbora

Treba brať na vedomie ale to, že takto pridané zariadenie je pripojené len do najbližšieho vypnutia stroja, pretože tento príkaz neovplyvňuje hlavný XML súbor uložený v `/etc/libvirt/qemu`.

¹eXtensible Markup Languagei – rozšíriteľný značkovací jazyk

Kapitola 4

Testovanie diskových zariadení a sieťových služieb

Či už zariadenia máme fyzicky pripojené alebo ich len simulujeme, nemalo by to mať vplyv na spôsob ich testovania. Preto už nie je potrebné rozlišovať virtualizovaný a nevvirtualizovaný systém.

4.1 Testy pre sieťové služby

V prvom rade je potrebné si rozmyslieť, čo konkrétne chceme zo sieťových služieb testovať a uvedomiť si, čo všetko má na testovanie vplyv a obsiahnuť všetky možnosti je nad rámec bakalárskej a asi aj diplomovej práce.

Ako príklad môžeme uviesť jednoduchý test – meranie rýchlosti/priepustnosti LAN siete. Najjednoduchšia varianta, čo by mohla niekoho napadnúť je, merať rýchlosť sťahovania pripraveného súboru cez HTTP protokol. Na jednom uzly siete sa vytvorí server a na druhom uzly začneme sťahovanie. Toto riešenie ale nie je úplne presné, keďže samotný protokol má taktiež nejakú réžiu. Preto treba ísť na čo najnižšiu vrstvu a ideálne si spraviť vlastný merací nástroj, ktorý bude pracovať priamo s paketmi. Na meranie ale má vplyv každé zariadenie, ktoré je v ceste od jedného bodu k druhému.

Nástroj iperf

Na meranie výkonnosti siete existuje napríklad nástroj *iperf*¹ vyvíjaný *NLANR/DAST*, ktorý dokáže merať priepustnosť siete, stratu paketov a iné. Projekt NLANR bohužiaľ skončil pred pár rokmi a preto je projekt neaktívny. Posledná zmena na ňom bola v roku 2011.

Iperf je potrebné spustiť na jednom systéme ako server a na druhom ako klient. Príklad najjednoduchšieho použitia je na obrázku 4.1.

```
/bin/iperf -s # Spustenie ako server  
/bin/iperf -c <host> # Spustenie ako klient, kde <host> je IP adresa servera
```

Obrázok 4.1: Príklad použitia iperf

¹Domovská stránka – <http://iperf.sourceforge.net>

Výsledok takéhoto jednoduchého príkladu môže vyzeráť napríklad ako na obrázku 4.2. Ako je vidieť, server mal IP adresu 192.168.0.103 a program nameral rýchlosť 168 Mbits/sec.

```
-----
Client connecting to 192.168.0.103, TCP port 5001
TCP window size: 22.9 KByte (default)
-----
[ 3] local 192.168.0.120 port 60573 connected with 192.168.0.103 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-10.0 sec 201 MBytes 168 Mbits/sec
```

Obrázok 4.2: Výstup z programu iperf

Nástroj TTCP

Ďalší nástroj, ktorý je možné použiť je *PCATTCP*². Tento program má podporu aj pre *IPv6* a je vyvíjaný pre Windows. Jeho unixová varianta sa volá *Unix Test TCP*.

4.2 Testy diskových zariadení

Meranie rýchlosti zápisu alebo čítania disku nie je až tak problematické narozdiel od sietí, ak si dáme pozor na to, či zápis prebehol v skutočnosti alebo ešte len čaká vo vyrovnávacej pamäti.

Nástroj dd

Najjednoduchší spôsob merania zápisu môžeme previesť nástrojom `/bin/dd`. Po spustení kopíruje zo zdroja do výstupu podľa zadaných argumentov (ako napríklad veľkosť načítaného bloku dát) a po ukončení mimo iné vypíše veľkosť prenesených dát s rýchlosťou operácie. Pokiaľ nás zaujímajú priebežné štatistiky počas kopírovania, program po odchytení signálu `SIGUSR1` vypíše aktuálne údaje o prenose. Príklad použitia a jeho výsledku je možné vidieť na obrázku 4.3.

```
$ /bin/dd if=/dev/zero of=file bs=1M count=300 conv=fdatasync
300+0 records in
300+0 records out
314572800 bytes (315 MB) copied, 2.96816 s, 106 MB/s
```

Obrázok 4.3: Príklad použitia a výsledok nástroja dd

Voľba `conv=fdatasync` nám zabezpečuje to, že *dd* po zápise vyžiada synchronizáciu systému a tým sú dáta v skutočnosti naozaj zapísané na disk.

²PCAUSA Test TCP – <http://www.pcausa.com/Utilities/pcattcp.htm>

Nástroj **hdparm**

Ďalšou možnosťou je použiť program `/usr/sbin/hdparm`. Konkrétne nás na účely testovania rýchlosti disku budú zaujímať prepínače uvedené v tabuľke 4.1.

Argument	Popis
<code>-t</code>	Testuje rýchlosť čítania cez vyrovnávaciu pamäť disku
<code>-T</code>	Testuje rýchlosť čítania priamo z linuxovej vyrovnávacej pamäte

Tabuľka 4.1: Popis argumentov `hdparm` pre testovanie rýchlosti

Existuje aj argument `-f`, ktorý pred koncom synchronizuje systém a vyprázdni vyrovnávaciu pamäť. Pri použití prepínača `-t` alebo `-T` to nie je potrebné, pretože to robia implicitne.

```
$ /usr/sbin/hdparm -t -T /dev/sda
/dev/sda:
Timing cached reads: 22722 MB in 1.98 seconds = 11476.62 MB/sec
Timing buffered disk reads: 386 MB in 3.01 seconds = 128.18 MB/sec
```

Obrázok 4.4: Príklad použitia nástroja `hdparm`

Nástroj **bonnie++**

Medzi pokročilejšie merače rýchlosti môžeme zaradiť napríklad *Bonnie++*. Program sa spúšťa bez argumentov. Testovanie trvá niekoľko minút a počas testov by sa na počítači nemalo pracovať pre čo najpresnejšie výsledky. Po skončení testov sa nám zobrazí tabuľka výsledkov s hodnotami latencie pre jednotlivé diskové operácie.

```

$ /usr/sbin/bonnie++
Writing a byte at a time...done
Writing intelligently...
done
Rewriting...done
Reading a byte at a time...done
Reading intelligently...done
start 'em...done...done...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version 1.96 -----Sequential Output----- --Sequential Input- --Random-
Concurrency 1 -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
hostname 31984M 1118 96 95761 7 46341 5 4279 78 115086 8 216.4 2
Latency 9476us 17262ms 13180ms 42596us 258ms 1516ms
Version 1.96 -----Sequential Create----- -----Random Create-----
hostname -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
      files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
      16 17190 37 +++++ + + + + + + + 28574 59 +++++ + + + + + + +
      Latency 90450us 226us 252us 91us 29us 151us
      1.96,1.96,laura.blaskovic.sk,1,1367789311,31984M
      ,,1118,96,95761,7,46341,5,4279,78,115086,8,216.
      4,2,16,,,,,17190,37,+++++,+,+,+++++,+,+,28574,59,+++++,+,+,
      +++++,+,+,9476us,17262ms,13180ms,42596us,258ms,1516ms,90450us
      ,226us,252us,91us,29us,151us

```

Obrázok 4.5: Príklad použitia nástroja Bonnie++

Na obrázku 4.5 je vidieť, aké testy približne *bonnie++* prevádza. Jedná sa o zápis do súboru, jeho prepisovanie, čítanie, mazanie a iné. Oproti vyššie uvedeným nástrojom ako *dd* a *hdparm* je *bonnie++* priamo určený na meranie rýchlosti pevného disku a preto sú jeho výsledky pravdepodobne viac relevantné.

Nástroj iozone

Ďalší z rady meračov rýchlosti diskov má názov *IOzone Filesystem Benchmark*. Podobne ako *bonnie++* testuje zápis, čítanie, opätovný zápis, opätovné čítanie, náhodné čítanie a iné. Tento nástroj sa nenachádza v bežných³ repozitároch systému *Fedora* a je ho potrebné stiahnuť na jeho domovskej stránke <http://www.iozone.org>.

Pre bližšiu špecifikáciu toho čo chceme otestovať, je možné presne definovať, aké testy sa majú spustiť. *IOzone* rozlišuje presne 13 testov, ktoré sa dajú nastaviť argumentom *-i*. Prehľad testov je v tabuľke 4.2.

³Základné repozitáre a rpmfusion repozitár.

Číslo	Názov testu
0	write/rewrite
1	read/re-read
2	random-read/write
3	Read-backwards
4	Re-write-record
5	stride-read
6	fwrite/re-fwrite
7	fread/Re-fread,
8	random mix
9	pwrite/Re-pwrite
10	pread/Re-pread
11	pwritev/Re-pwritev
12	preadv/Re-preadv

Tabuľka 4.2: Prehľad dostupných testov programu IOzone

Tento nástroj ma naozaj širokú škálu nastavení. Medzi zaujímavé možnosti patrí napríklad ukladanie výsledkov do tabuľky vo formáte pre *Microsoft Excel* alebo *Open Office Calc*.

Kapitola 5

Plán testovania pre Fedora Linux

5.1 Úvod do testovania

Na testovanie *tuned* využijeme pomocnú knižnicu *beakerlib* [3] pre jednoduchšie písanie testov a prehľadnejšiu interpretáciu dosiahnutých výsledkov. Cieľom testovania je zistiť, nakoľko *tuned* ovplyvňuje rýchlosť diskových operácií.

Testovacie položky

Napísané testy budú overovať správnu funkcionálnu *tuned* démona a taktiež profilov v zameraní na diskové operácie. Všetky testy budú pripravené pre linuxovú distribúciu Fedora 18 [4].

Overí sa rýchlosť zápisu na bežných aj RAID diskoch a s najpoužívanějšími súborovými systémami uvedenými v zozname nižšie.

- **ext2** – Pôvodne navrhnutý francúzskym vývojárom *Rémy Card*-om. Tento súborový systém prišiel ako náhrada za *ext*.
- **ext3** – Jeho vývoj viedol *Stephen Tweedie* a hlavný rozdiel oproti *ext2* je v pridanej žurnálovacej vrstve.
- **ext4** – Bol zahrnutý do linuxového jadra verzie 2.6.19. Zachováva spätnú kompatibilitu s *ext2* a *ext3*.
- **xfs** – pôvodne slúžil ako predvolený súborový systém v operačnom systéme *IRIX*¹ a až neskôr bol portovaný do linuxového jadra.
- **jfs** – Žurnálovací súborový systém, ktorý uschováva plánované zmeny žurnálu pred zápisom. Tým predchádza poškodeniu dát v prípade havárie.
- **ReiserFS** – oproti *ext2* a *ext3* je rýchlejší pre súbory menšie ako 4 KiB vďaka *tail packingu*².
- **btrfs** – Experimentálny B–strom súborový systém s podporou obrazov disku.

¹Operačný systém vyvíjaný spoločnosťou Silicon Graphics, postavený na MIPS architektúre.

²Princíp redukcie internej fragmentácie.

Hostiteľský systém bude spúšťať predpripravené obrazy virtualizovaného systému Fedora 18. K virtualizovanému systému sa budú pripájať ďalšie disky. Tieto nové disky budú formátované na najpoužívanejšie súborové systémy a testované ich rýchlosti pri rôznych profiloch *tuned*.

Na testovacie účely použijeme najnovšiu verziu *tuned* z repozitára *Fedora 18*.

Softvérové riziká

V prípade zlyhania niektorých testov môže prísť k poškodeniu už pripojených diskov. Preto je vhodné spúšťať sadu testov na virtuálnom stroji. V prípade vydania novej verzie *tuned* alebo inej použitej komponenty je tu riziko, že testy nebudú stabilné a môžu sa správať nepredvídateľne. V tomto prípade ale môžeme uvažovať o nájdení chyby (dokonca regresie) v *tuned*.

Čo sa nebude testovať

Pretože testy bežia na virtualizovanom hardvéri, výsledky môžu byť trochu skreslené. Viac o testovaní na virtualizovanom systéme v sekcii 5.2.

Kritéria pre splnenie testov

Počas testovania so zapnutým démonom *tuned* by všetky I/O operácie diskov mali byť rýchlejšie alebo aspoň tak rýchle ako s vypnutým *tuned*. Žiadna operácia by nemala skončiť s chybou a disky by sa nemali poškodiť. Zápis so zapnutým *tuned* musí mať rovnaké výsledky ako s vypnutým.

Výnimku tvoria len tie profily, ktoré nemajú za úlohu zvyšovať výkon. Medzi tieto profily patrí napríklad *powersave*, ktorý naopak môže diskové operácie spomaľovať.

Kritéria na prerušenie testovania

Ak zlyhá operácia pripájania disku k virtuálnemu stroju, ďalšie testovanie stráca význam. Preto je potrebné počas testovania kontrolovať, či táto kľúčová operácia dopadla úspešne. Rovnaká situácia môže nastať, ak zlyhá nahratie uloženého obrazu disku systému pred testovaním.

Prvý prípad (pripájanie obrazu disku) sa rieši opakovaním testu. Takto dosiahneme vždy požadované výsledky, aj keď operácia zlyhá. Problém pripájania diskov je bližšie popísaný v sekcii 8.1.

Čo obsahuje plán testovania a jeho výsledky

Celé testovania sa skladá z niekoľkých súčastí:

- Plán testovania.
- Zdrojové kódy jednotlivých testov.
- Knižnica na spracovanie výsledkov.
- Zoznam chýb, ktoré nastali.
- Vyhodnotenie výsledkov.

Výsledky testov majú podobu nameraných hodnôt v tabuľkách (generujú sa automaticky) a písomného popisu týchto hodnôt.

5.2 Riziká testovania výkonu na virtualizovanom stroji

Testovanie diskových operácií prebieha vo virtuálnom stroji za použitia *qemu-kvm* pod *libvirt*. Tento spôsob testovania som zvolil pre minimalizáciu problémov, ktoré môžu nastať pri testovaní (popísané v sekcii 5.1) a ochránenie hostiteľského operačného systému.

5.2.1 Výhody

Medzi hlavné výhody testovania na virtualizovanom systéme patrí:

- Jednoduchšia správa obrazov celého systému.
- Pohodlné pripájanie a odpájanie diskov.
- Dokonalé prispôsobenie systému pre požiadavky testov – na hostiteľskom systéme moc nezáleží a dá sa využiť skoro akákoľvek Linuxová distribúcia.

5.2.2 Nevýhody

Toto testovanie prináša ale aj množstvo nevýhod.

- Hostiteľský systém môže obmedzovať virtualizovaný systém.
- Medzi systémami môže nastať kešovanie (ako je zobrazené aj na obrázku 6.2). Toto sa ale dá do veľkej miery eliminovať a nemalo by mať vplyv na výsledky testov.
- Virtualizovaný systém a taktiež aj virtuálne disky budú vždy odlišné od fyzických. Aj keď virtualizácia beží na úrovni jadra, stále su v systémoch malé rozdiely. Preto je celé testovanie experimentálne a výsledky sa budú zbierať a priemerovať z viacerých iterácií testovania.

5.2.3 Zhrnutie

Virtuálny systém nebude nikdy pre testovanie výkonu úplne ideálny. Výkon by bolo najvhodnejšie testovať priamo bez virtualizácie a najvhodnejšie na viacerých druhoch hardvéru a v prípade testu diskových operácií aj s viacerými typmi diskových zariadení od rôznych výrobcov a výsledky nakoniec zpriemerovať.

Aj napriek všetkým nevýhodám je testovanie na virtualizovanom systéme vhodné pre naše použitie. Výsledky testov sú porovnávacie (so zapnutým a vypnutým *tuned*) – to znamená, že aj keď by v skutočnosti časy mohli byť odlišné, rozdiel medzi *tuned* a bez *tuned* verziou ostáva rovnaký.

Taktiež treba poznamenať, že je dnes virtualizácia využívaná veľmi často a preto virtualizácia pri testovaní nie je prekážkou.

Kapitola 6

Príprava prostredia a implementácia testov

Táto kapitola popisuje testovacie prostredie, ako správne nakonfigurovať systém, na čo si pri testovaní dať pozor a na záver približuje implementáciu testov, ich parametrizáciu a popis riešenia problémov, ktoré môžu nastať.

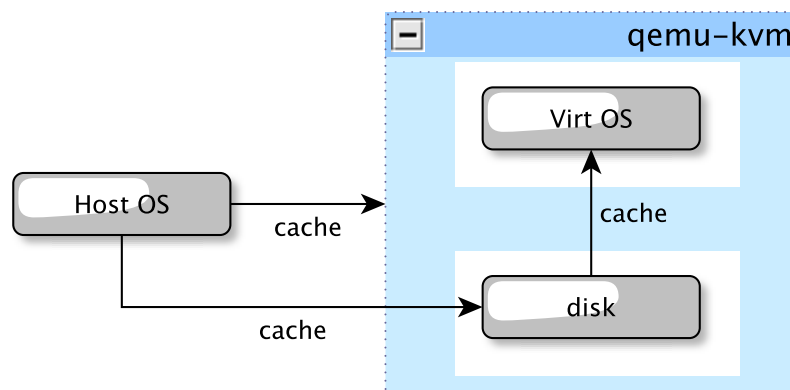
6.1 Nastavenia systému

Pred testovaním je potrebné pripraviť si nainštalovaný systém *Fedora 18* ako obraz disku. Tento obraz sa bude spúšťať cez *qemu-kvm*. Disk, na ktorom sa bude testovať rýchlosť zápisu a čítania by mal mať v ideálnom prípade vypnutú vyrovnávaciu pamäť. Pred testovaním aj po testovaní je potrebné hostiteľský aj virtualizovaný systém synchronizovať a zmazať stránky vo vyrovnávacej pamäti (Obrázok 6.1).

```
/bin/sync  
echo 3 > /proc/sys/vm/drop_caches
```

Obrázok 6.1: Synchronizácia systému

Každé miesto, kde je možnosť, že by systém si uchovával nejaké dáta vo vyrovnávacej pamäti, ktoré by mohli ovplyvniť výsledky testovania je potrebné poznať (Obrázok 6.2).



Obrázok 6.2: Vyznačenie miest, kde môže nastať ukladanie do vyrovnávacej pamäte

Virtualizovaný systém musí obsahovať všetky potrebné balíčky, ktorých programy sa používajú v testoch. Ďalej by mali byť vypnuté všetky nepotrebné služby na pozadí (napríklad *abrt* alebo *ntp*). Treba si dať pozor aj na zoznam *cron*¹ úloh, ktoré môžeme nájsť v adresároch */etc/cron.**.

6.2 Implementácia testov

Na implementáciu testov je využitý prevažne jazyk *bash*. Testovanie riadi hlavný súbor *test_disk.sh*, v ktorom sa nastavujú parametre testov. Tieto testy su popísane v tabuľke 6.1. Konkrétne podtesty začínajú prefixom *inc.**. Virtuálny systém sa pred každým testom vypne, obnoví zo zálohy disku a znova zapne.

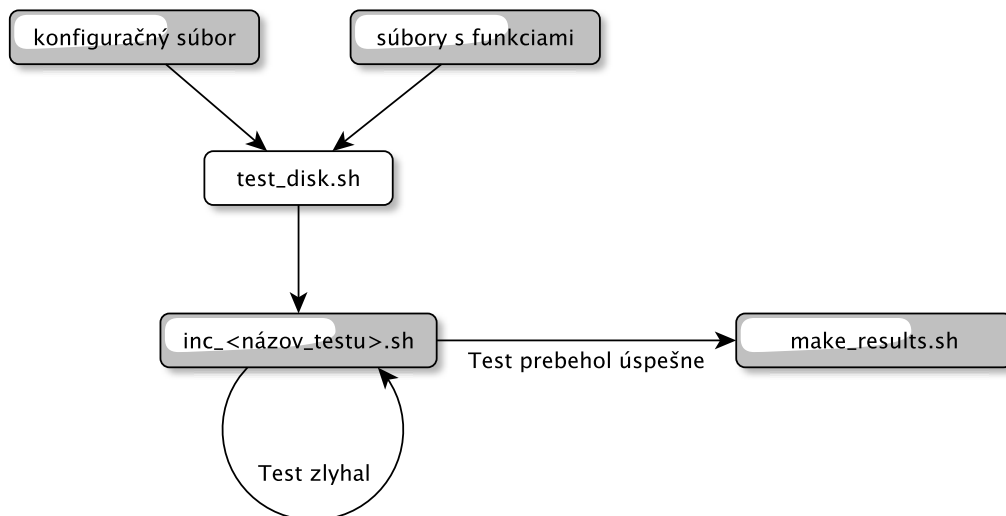
Názov premennej	Popis
TEST_COMMAND	Sada príkazov pre diskové operácie
FS	Asociatívne pole, obsahujúce názov súborového systému a príkazu na jeho vytvorenie
TO_TEST	Zoznam testov, ktoré sa prevedú
TUNED_PROFILES	Profily <i>tuned</i> , ktoré sa zahrnú v testoch
MACHINE_NAME	Názov virtuálneho systému pre ovládanie cez <i>virsh</i>
MACHINE_IP	IP adresa virtuálneho systému
MACHINE_DISK_SOURCE	Obraz predpripraveného disku s nainštalovaným systémom
MACHINE_DISK	Miesto, kde <i>libvirt</i> očakáva hlavný disk systému
LOG_FILE	Súbor s výsledkami
FAILED_RUN_LOG	Súbor s príznakom chyby

Tabuľka 6.1: Popis premenných pre parametrizáciu testov.

Disky sa pripájajú podľa toho, ako to vyžaduje aktuálny test. Kešovanie diskov medzi hostiteľským a virtualizovaným systémom je vypnuté na miestach, ktoré su uvedené na

¹Časovo závislý plánovač úloh. Každý užívateľ si môže naplánovať vykonávanie príkazov v daných časových intervaloch.

obrázku 6.2 a zároveň neovplyvňujú prácu *tuned*.



Obrázok 6.3: Pribeh testovania

6.2.1 Riešenie problému so zlyhaním testu

Test môže zlyhať z nepredvídateľných príčin, ako je napríklad chyba *libvirt* popísaná v kapitole 8.1. Ak v teste zlyhá operácia, ktorá má vplyv na výsledky, zavolá sa funkcia `failedRunSave`, ktorá nastaví príznak chyby. Funkciou `failedRunCheck` vieme overiť tento príznak. Ak napríklad nastala chyba pripájania diskov, test zápisu sa už nespustí. Hlavný súbor `test_disk.sh` taktiež využíva túto funkciu a daný test opakovane spúšťa, ak je to potreba.

Kapitola 7

Výsledky testovania

7.1 Automatizované spracovanie výsledkov

Po skončení testov sa spúšťa skript `make_results.sh`, ktorý automatizovane spracováva výsledky testov a generuje výstup vo formáte \LaTeX . Výsledky sú v tabuľkách a rozdelené do sekcií. Ak chceme niektorú sekciu dodatočne okomentovať, tento \LaTeX kód očakáva dodatočné súbory s menom `obsah-test-<názov tuned profilu>.tex`. Ak tento súbor existuje, jeho text sa vloží pre tabuľku s výsledkami.

Skript priebežne počíta aj priemerné časy a priemerné percentuálne zlepšenie pre každý súborový systém zvlášť a na záver pre celý profil.

7.2 Vyhodnotenie testov

7.2.1 Testovanie s profilom *balanced*

Profil *balanced* je prednastavený po spustení *tuned*. Mal by rovnomerne optimalizovať systém a je vhodným začiatkom pre bežnú prácu.

V základnom nastavení ale nerobí zásahy do diskov, ale iba *CPU*, *audio* a *video*. Pre ladenie diskov by bolo potrebné ho upraviť, ale toto testovanie sa zaoberá základnými profilmi.

Rozdiely v rýchlosti diskových operácií su preto minimálne, dokonca až horšie ako bez použitia *tuned*. Z testu najväčšie zrýchlenie dosiahol súborový systém *ReiserFS*. *Ext3* znamenal naopak zhoršenie až o viac ako 3 %.

Test	bez tuned	s tuned	rozdiel	rozdiel [%]
simple.disk ext3	105 s	107 s	-2 s	-1.90 %
raid0 ext3	117 s	117 s	0 s	0 %
raid1 ext3	108 s	119 s	-11 s	-10.18 %
Priemery	110.00 s	114.33 s	-4.33 s	-3.67 %
simple.disk ext2	123 s	121 s	2 s	1.63 %
raid0 ext2	116 s	120 s	-4 s	-3.44 %
raid1 ext2	121 s	121 s	0 s	0 %
Priemery	120.00 s	120.67 s	-0.67 s	-0.33 %
simple.disk ext4	117 s	117 s	0 s	0 %
raid0 ext4	114 s	116 s	-2 s	-1.75 %
raid1 ext4	118 s	120 s	-2 s	-1.69 %
Priemery	116.33 s	117.67 s	-1.33 s	-0.67 %
simple.disk jfs	123 s	122 s	1 s	.82 %
raid0 jfs	123 s	118 s	5 s	4.07 %
raid1 jfs	129 s	131 s	-2 s	-1.55 %
Priemery	125.00 s	123.67 s	1.33 s	1.67 %
simple.disk reiserfs	128 s	119 s	9 s	7.04 %
raid0 reiserfs	138 s	136 s	2 s	1.45 %
raid1 reiserfs	194 s	199 s	-5 s	-2.57 %
Priemery	153.33 s	151.33 s	2.00 s	2.67 %
simple.disk btrfs	108 s	110 s	-2 s	-1.85 %
raid0 btrfs	123 s	116 s	7 s	5.70 %
raid1 btrfs	103 s	113 s	-10 s	-9.70 %
Priemery	111.33 s	113.00 s	-1.67 s	-1.33 %
simple.disk xfs	105 s	108 s	-3 s	-2.85 %
raid0 xfs	113 s	111 s	2 s	1.77 %
raid1 xfs	116 s	116 s	0 s	0 %
Priemery	111.33 s	111.67 s	-0.33 s	0.00 %
Celkové priemery	121.76 s	121.76 s	-0.71 s	-0.24 %

Tabuľka 7.1: Výsledky testov pre profil balanced

7.2.2 Testovanie s profilom *latency-performance*

Úlohou profilu *latency-performance* je čo najviac znížiť latenciu systému. Disky *sd**, *cciss**, *dm-**, *vd** sú ovplyvnené, ak je tento profil aktívny.

Rýchlosť diskových operácií sa zlepšila až o približne 8 %. Prekvapením je experimentálny súborový systém *btrfs*, ktorý pri použití s *raid0* zaznamenal zrýchlenie o viac ako 13 %.

Test	bez tuned	s tuned	rozdiel	rozdiel [%]
simple.disk ext3	105 s	100 s	5 s	4.77 %
raid0 ext3	117 s	107 s	10 s	8.55 %
raid1 ext3	108 s	107 s	1 s	.93 %
Priemery	110.00 s	104.67 s	5.33 s	5.00 %
simple.disk ext2	123 s	99 s	24 s	19.52 %
raid0 ext2	116 s	111 s	5 s	4.32 %
raid1 ext2	121 s	113 s	8 s	6.62 %
Priemery	120.00 s	107.67 s	12.33 s	10.67 %
simple.disk ext4	117 s	97 s	20 s	17.10 %
raid0 ext4	114 s	104 s	10 s	8.78 %
raid1 ext4	118 s	110 s	8 s	6.78 %
Priemery	116.33 s	103.67 s	12.67 s	11.33 %
simple.disk jfs	123 s	110 s	13 s	10.57 %
raid0 jfs	123 s	118 s	5 s	4.07 %
raid1 jfs	129 s	111 s	18 s	13.96 %
Priemery	125.00 s	113.00 s	12.00 s	10.00 %
simple.disk reiserfs	128 s	134 s	-6 s	-4.68 %
raid0 reiserfs	138 s	136 s	2 s	1.45 %
raid1 reiserfs	194 s	166 s	28 s	14.44 %
Priemery	153.33 s	145.33 s	8.00 s	4.33 %
simple.disk btrfs	108 s	100 s	8 s	7.41 %
raid0 btrfs	123 s	106 s	17 s	13.83 %
raid1 btrfs	103 s	96 s	7 s	6.80 %
Priemery	111.33 s	100.67 s	10.67 s	9.67 %
simple.disk xfs	105 s	94 s	11 s	10.48 %
raid0 xfs	113 s	110 s	3 s	2.66 %
raid1 xfs	116 s	109 s	7 s	6.04 %
Priemery	111.33 s	104.33 s	7.00 s	7.00 %
Celkové priemery	111.33 s	111.33 s	9.71 s	8.29 %

Tabuľka 7.2: Výsledky testov pre profil latency-performance

7.2.3 Testovanie s profilom *powersave*

Ak potrebujeme ušetriť energiu, profil *powersave* je na to najvhodnejší. Na rýchlosť diskových operácií má ale negatívny efekt. Jeho aktivovaním sa nastaví hodnota ALPM¹ na *min_power*.

Výsledky sú približne totožné, ako s vypnutým *tuned*. V niektorých prípadoch sú do-

¹Aggressive Link Power Management - technika, ktorá pomáha šetriť energiu. Má tri stavy: *min_power*, *medium_power* a *max_performance*

konca horšie.

Test	bez tuned	s tuned	rozdiel	rozdiel [%]
simple.disk ext3	105 s	108 s	-3 s	-2.85 %
raid0 ext3	117 s	112 s	5 s	4.28 %
raid1 ext3	108 s	126 s	-18 s	-16.66 %
Priemery	110.00 s	115.33 s	-5.33 s	-4.33 %
simple.disk ext2	123 s	118 s	5 s	4.07 %
raid0 ext2	116 s	111 s	5 s	4.32 %
raid1 ext2	121 s	125 s	-4 s	-3.30 %
Priemery	120.00 s	118.00 s	2.00 s	2.33 %
simple.disk ext4	117 s	113 s	4 s	3.42 %
raid0 ext4	114 s	118 s	-4 s	-3.50 %
raid1 ext4	118 s	127 s	-9 s	-7.62 %
Priemery	116.33 s	119.33 s	-3.00 s	-2.00 %
simple.disk jfs	123 s	119 s	4 s	3.26 %
raid0 jfs	123 s	119 s	4 s	3.26 %
raid1 jfs	129 s	134 s	-5 s	-3.87 %
Priemery	125.00 s	124.00 s	1.00 s	1.67 %
simple.disk reiserfs	128 s	166 s	-38 s	-29.68 %
raid0 reiserfs	138 s	142 s	-4 s	-2.89 %
raid1 reiserfs	194 s	151 s	43 s	22.17 %
Priemery	153.33 s	153.00 s	0.33 s	-2.67 %
simple.disk btrfs	108 s	111 s	-3 s	-2.77 %
raid0 btrfs	123 s	118 s	5 s	4.07 %
raid1 btrfs	103 s	101 s	2 s	1.95 %
Priemery	111.33 s	110.00 s	1.33 s	1.67 %
simple.disk xfs	105 s	107 s	-2 s	-1.90 %
raid0 xfs	113 s	124 s	-11 s	-9.73 %
raid1 xfs	116 s	115 s	1 s	.87 %
Priemery	111.33 s	115.33 s	-4.00 s	-3.00 %
Celkové priemery	122.14 s	122.14 s	-1.10 s	-0.90 %

Tabuľka 7.3: Výsledky testov pre profil powersave

7.2.4 Testovanie s profilom *throughput-performance*

Tento profil ovplyvňuje disky podobne ako *latency-performance*. Veľkým rozdielom ale je v *transparent huge pages*, ktoré *latency-performance* nastavuje na **never** a *throughput-performance* na **always**.

Zrýchlenie diskových operácií s týmto profilom dosahuje hodnoty aj cez 10 % a tým sa stáva ideálnym profilom, ak chceme optimalizovať prácu s diskami.

Najlepšie zrýchlenie dosiahol na súborovom systéme *JFS*, avšak najrýchlejší čas mal s *btrfs* a s použitím *raid1*.

Test	bez tuned	s tuned	rozdiel	rozdiel [%]
simple_disk ext3	105 s	100 s	5 s	4.77 %
raid0 ext3	117 s	107 s	10 s	8.55 %
raid1 ext3	108 s	116 s	-8 s	-7.40 %
Priemery	110.00 s	107.67 s	2.33 s	2.33 %
simple_disk ext2	123 s	105 s	18 s	14.64 %
raid0 ext2	116 s	110 s	6 s	5.18 %
raid1 ext2	121 s	119 s	2 s	1.66 %
Priemery	120.00 s	111.33 s	8.67 s	7.67 %
simple_disk ext4	117 s	100 s	17 s	14.53 %
raid0 ext4	114 s	108 s	6 s	5.27 %
raid1 ext4	118 s	113 s	5 s	4.24 %
Priemery	116.33 s	107.00 s	9.33 s	8.67 %
simple_disk jfs	123 s	106 s	17 s	13.83 %
raid0 jfs	123 s	104 s	19 s	15.45 %
raid1 jfs	129 s	108 s	21 s	16.28 %
Priemery	125.00 s	106.00 s	19.00 s	15.67 %
simple_disk reiserfs	128 s	123 s	5 s	3.91 %
raid0 reiserfs	138 s	136 s	2 s	1.45 %
raid1 reiserfs	194 s	156 s	38 s	19.59 %
Priemery	153.33 s	138.33 s	15.00 s	8.67 %
simple_disk btrfs	108 s	102 s	6 s	5.56 %
raid0 btrfs	123 s	107 s	16 s	13.01 %
raid1 btrfs	103 s	90 s	13 s	12.63 %
Priemery	111.33 s	99.67 s	11.67 s	11.00 %
simple_disk xfs	105 s	97 s	8 s	7.62 %
raid0 xfs	113 s	102 s	11 s	9.74 %
raid1 xfs	116 s	107 s	9 s	7.76 %
Priemery	111.33 s	102.00 s	9.33 s	8.67 %
Celkové priemery	110.29 s	110.29 s	10.76 s	8.95 %

Tabuľka 7.4: Výsledky testov pre profil throughput-performance

7.2.5 Testovanie s profilom *virtual-guest*

Virtual-guest by mal byť najvhodnejší profil pre virtuálny systém – to znamená, aj pre naše testovacie prostredie. Na diskoch upravuje *readahead* hodnotu a nastavuje ju na 4 krát väčšiu. Je ale dobré vedieť, že taktiež zároveň načítava nastavenia z profilu *throughput-performance*.

Práve pri tomto profile su najviac viditeľné rozdiely medzi súborovými systémami. S použitím *ext3* sme dosiahli zrýchlenie niečo cez 1 %, ale s použitím *ext2* je to až cez 10 %.

Toto nastavenie taktiež môžeme označiť za veľmi vhodné pre zrýchlenie diskových operácií a znova je najväčšie zrýchlenie pri použití s *JFS*, ale najrýchlejší pri použití kombinácie *btrfs* a *raid1* ako aj v predchádzajúcom prípade.

Test	bez tuned	s tuned	rozdiel	rozdiel [%]
simple_disk ext3	105 s	101 s	4 s	3.81 %
raid0 ext3	117 s	109 s	8 s	6.84 %
raid1 ext3	108 s	116 s	-8 s	-7.40 %
Priemery	110.00 s	108.67 s	1.33 s	1.33 %
simple_disk ext2	123 s	100 s	23 s	18.70 %
raid0 ext2	116 s	109 s	7 s	6.04 %
raid1 ext2	121 s	115 s	6 s	4.96 %
Priemery	120.00 s	108.00 s	12.00 s	10.33 %
simple_disk ext4	117 s	105 s	12 s	10.26 %
raid0 ext4	114 s	105 s	9 s	7.90 %
raid1 ext4	118 s	111 s	7 s	5.94 %
Priemery	116.33 s	107.00 s	9.33 s	8.33 %
simple_disk jfs	123 s	98 s	25 s	20.33 %
raid0 jfs	123 s	115 s	8 s	6.51 %
raid1 jfs	129 s	118 s	11 s	8.53 %
Priemery	125.00 s	110.33 s	14.67 s	12.33 %
simple_disk reiserfs	128 s	118 s	10 s	7.82 %
raid0 reiserfs	138 s	125 s	13 s	9.43 %
raid1 reiserfs	194 s	169 s	25 s	12.89 %
Priemery	153.33 s	137.33 s	16.00 s	10.33 %
simple_disk btrfs	108 s	108 s	0 s	0 %
raid0 btrfs	123 s	105 s	18 s	14.64 %
raid1 btrfs	103 s	89 s	14 s	13.60 %
Priemery	111.33 s	100.67 s	10.67 s	9.67 %
simple_disk xfs	105 s	94 s	11 s	10.48 %
raid0 xfs	113 s	104 s	9 s	7.97 %
raid1 xfs	116 s	107 s	9 s	7.76 %
Priemery	111.33 s	101.67 s	9.67 s	9.00 %
Celkové priemery	110.52 s	110.52 s	10.52 s	8.76 %

Tabuľka 7.5: Výsledky testov pre profil virtual-guest

Kapitola 8

Bočné produkty práce

8.1 Problém s *libvirt*

Pri pripájaní diskov k virtuálnemu systému za použitia príkazu `virsh attach-device`, sa systém občas náhodne vypne. Nepodarilo sa mi spoľahlivo zreprodukovať túto chybu do takeho stavu, aby som ju mohol nahlásiť vývojárom, pretože nastávala nepravidelne.

O páde systému som nenašiel žiadnu zmienku v logoch virtualizovaného ani hostovacieho systému.

Tento problém som riešil tým, že kontrolujem úspešnosť operácií a test sa automaticky opakuje, kým neskončí korektne. Táto problematika je presnejšie popísaná v kapitole 6.2.1.

8.2 Power Management Test Day

Dňa 17. apríla 2013 sa konal deň otvorených dverí Brnenskej pobočky firmy *Red Hat*. Súčasťou programu bol aj *Power Management Test Day* najnovšej verzie operačného systému *Fedora 19*, ktorého som bol spoluorganizátor.

Mimo iné sa testovalo aj *tuned* samotné. Najväčším prekvapením bolo testovanie *tuned* profilu *powersave*, ktorý by mal znižovať elektrický odber. Na testovanie prišlo množstvo ľudí s rôznymi notebookmi. Konkrétne výsledky sú uvedené v tabuľke 8.1.

Typ notebooku	Odber bez tuned	Odber s tuned	Zlepšenie
Lenovo ThinkPad X230	4.540 Wh	4.110 Wh	9.47%
HP Elitebook 8540w	7.566 Wh	7.493 Wh	0.97%
Lenovo T60 laptop	4.720 Wh	4.500 Wh	4.66%
Dell Optiplex 990	8.500 Wh	7.700 Wh	9.41%
Lenovo ThinkPad T61	4.880 Wh	4.340 Wh	11.07%
Lenovo ThinkPad T520	5.387 Wh	4.815 Wh	10.62%
Lenovo T510	3.310 Wh	2.820 Wh	14.80%
ThinkPad T430	5.690 Wh	5.370 Wh	5.62%
Samsung N210 (Intel Atom)	1.402 Wh	1.394 Wh	0.58%
Priemerné zlepšenie	7.47%		

Tabuľka 8.1: Odber rôznych notebookov s vypnutým a zapnutým tuned.

Už na prvý pohľad je viditeľné, že tento *tuned* profil si robí svoju prácu. V priemere sa odber znížil o 7.47 %.

8.3 Nájsené chyby

Počas písania tejto bakalárskej práce, narábanie s *tuned* a testovaní som narazil na niekoľko chýb, ktoré som nahlasoval do oficiálnej *Red Hat* a *Fedora* bugzilly [6]. Chyby sa nachádzali väčšinou priamo v balíku *tuned*, ale taktiež v balíku *selinux-policy*.

Tieto chyby boli väčšinou opravené do pár dní. Ich prehľad je v tabuľke 8.2.

Číslo chyby	Názov	Status
BZ#907065	tuned-adm can crash because of unknown variable log	opravené
BZ#953128	tuned does not start if active profile is missing	opravené
BZ#953132	AVCs after starting tuned daemon	opravené
BZ#901689	Typo in tuned-adm.py	opravené
BZ#911138	tuned-adm profile virtual-guest/host AVC denied	zatiaľ neopravené
BZ#958575	mkfs.btrfs segfaults	opravené

Tabuľka 8.2: Zoznam nájsených chýb

Tieto chyby neovplyvnili testovanie, pretože neboli až tak závažné alebo ich vývojár rýchlo opravil. Jedine chyba v `mkfs.btrfs` znemožnila testovanie tohoto súborového systému po dlhý čas a bola opravená len pár dní pred ukončením tejto práce.

Na množstvo ďalších chýb som narazil, ktoré už ale boli nahlásene aj opravené, len sa nedostali do repozitárov. Preto bolo potrebné niekedy aktualizovať balíček z testovacieho repozitára alebo priamo z upstreamu.

Kapitola 9

Záver

Z výsledkov testovania (kapitola 7) je možné potvrdiť zrýchlenie diskových operácií pre niektoré *tuned* profily.

Medzi najvhodnejšie profily patrí *throughput-performance* a *latency-performance*. Pri použití na virtuálnom systéme by bolo asi najvhodnejšie použiť profil *virtual-guest*, ktorý kopíruje *throughput-performance* a dodatočne vylepšuje ešte ďalšie nastavenia vhodné pri virtualizácii.

9.1 Najrýchlejší súborový systém a najväčšie zrýchlenie

Najväčšie zrýchlenie po aplikovaní *tuned* profila zaznamenal pomerne neznámy súborový systém *JFS*. Narozdiel od *extN* nikdy nezaznamenal zhoršenie so zapnutým *tuned* a preto je možné *tuned* bez obáv použiť, ak ho máme.

Zrýchlenie je ale relatívne a neznamená to, že na *JFS* boli diskové operácie najrýchlejšie. Experimentálny súborový systém *btrfs* jednoznačne vyhráva v rýchlosti. V kombinácii s *raid1* jednoznačne víťazí nad ostatnými a vďaka jeho schopnosti zaznamenávania obrazov diskov určite stojí za vyskúšanie.

9.2 Ďalší možný rozvoj testovania

Aby sme dosiahli ešte lepšie výsledky, môžeme použiť program *powertop*, ktorý nám navrhuje, čo by sa dalo na systéme vylepšiť a programom *powertop2tuned* si nechať na základe odporúčaní vygenerovať profil na mieru tak, ako bolo popisované v sekcii 2.2.2.

Na toto riešenie je ale potrebné dokonale rozumieť problematike a obsiahnuť všetky možnosti nastavení pre automatizované testovanie by bolo veľmi výpočetne náročné.

Literatúra

- [1] *CPU Frequency Scaling*, [online]. 2013 [cit. 2013-04-20].
URL https://wiki.archlinux.org/index.php/CPU_Frequency_Scaling
- [2] *Tuning I/O Performance*, [online]. 2013 [cit. 2013-04-20].
URL http://doc.opensuse.org/products/draft/SLES/SLES-tuning_sd_draft/cha.tuning.io.html
- [3] BeakerLib - domovská stránka. <https://fedorahosted.org/beakerlib>.
- [4] Fedora - domovská stránka. <http://fedoraproject.org>.
- [5] PowerTOP - domovská stránka. <https://01.org/powertop>.
- [6] Red Hat bugzilla. <https://bugzilla.redhat.com>.
- [7] Tuned - domovská stránka. <https://fedorahosted.org/tuned>.
- [8] Ron Patton: *Software testing Second edition*. SAMS, 2005, iISBN 0-672-32798-8.
- [9] Stanovich, M.; Baker, T.; Wang, A.-I.: Throttling On-Disk Schedulers to Meet Soft-Real-Time Requirements. In *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE*, 2008, ISSN 1545-3421, s. 331–341, doi:10.1109/RTAS.2008.30.

Dodatok A

Obsah CD

Priložené CD obsahuje testovacie skripty a súbor README s návodom na ich použitie.