OPTIMIZĒTU XEN VIRTUĀLO MAŠĪNU IZMANTOŠANA

XEN VIRTUĀLO MAŠĪNU PIELĀGOŠANA ĀTRAI SĀKNĒŠANAI UN MINIMĀLAM ATMIŅAS PATĒRIŅAM UZ UBUNTU LINUX BĀZES

Dokumenta faila vārds:	**LG-nodevums-Xen-mini-bk.doc**
Aktivitāte:	**2.aktivitāte "Starpprogrammatūras rīku praktiskā realizācija"**
Projekta numurs:	**VPD1/ERAF/CFLA/05/APK/2.5.1./000055/027**
Organizācija:	**Latvijas Universitātes aģentūra "Latvijas Universitātes Matemātikas un informātikas institūts"**
Autori:	**Leo Trukšāns, Baiba Kaškina**

Anotācija:

Šajā dokumentā ir aprakstītas Xen izmantošanas un virtuālo mašīnu optimizēšanas iespējas, kā arī izstrādāti rīki vienkāršai un ātrai virtuālo mašīnu izveidei. Apskatīta Xen sistēmas uzbūve un lietošana, Xen virtuālo mašīnu īpašības. Tāpat aprakstīta Ubuntu Linux startēšanas sistēma Upstart un tās pielāgošanas iespējas vienkāršāku un optimālāku sistēmu izveidē. Doti trīs varianti ar piemēriem minimālus resursus prasošu virtuālo mašīnu izveidošanai.

Saturs

1. XEN VIRTUALIZĀCIJAS SISTĒMA

1.1. XEN PRODUKTS

Xen ir populārākā atklātā pirmkoda (*Open Source*) virtualizācijas sistēma. Tā jebkuram ir brīvi pieejama lietošanai, pielāgošanai un izplatīšanai. Xen ir pieejams arī komerciāls atbalsts no tās galvenā attīstītāja *XenSource Inc.* (http://www.xensource.com/).

Xensource uzņēmums uzsver galvenos Xen sistēmas pielietojumus:

- Serveru konsolidācija;
- Aparātiskā neatkarība;
- Vairāku OS izpilde uz viena datora izstrādei vai testēšanai;
- Kodola izstrāde;
- Klāsteru (cluster) skaitļošana;
- Aparātiskais nodrošinājums modificētām OS.

Xen sistēmu sāka veidot Kembridžas universitātē. Pirmā Xen versija 1.0 tika izlaista 2003. gada oktobrī. Pašlaik Xen produkts ietilpst Citrix uzņēmumā, kas agrāk bija populārs ar terminālserveru risinājumiem.

Papildus informāciju par Xen var iegūt sekojošās saitēs: http://en.wikipedia.org/wiki/Xen, http://www.xen.org/. Par Xen lietošanu ir izdota grāmata “Virtualization with Xen” (Syngress, 2007), pieejama ražotāja dokumentācija: “Xen user's manual” (XenSource, 2005) un “Xen interface manual” (XenSource, 2005). Šī nodaļa lielā mērā balstīta uz Xen lietotāja ceļveža un autora pieredzes.

1.2. XEN TERMINOLOĢIJA

VMM – programmatūra, kas ļauj vairākas virtuālās mašīnas multipleksēt vienā fiziskā.

Paravirtualizācija – virtualizācijas metode, kas prasa operētājsistēmas modificēšanu darbam virtuālā mašīnā.

Hipervīzors (Hypervisor) – alternatīvs VMM nosaukums – pārvalda ”supervīzorus”.

Pilna virtualizācija – virtualizācijas metode, kas neprasa nekādu virtuālās mašīnas sistēmas modificēšanu.

Shadow pagetables – metode datora atmiņas izvietojuma slēpšanai no virtuālajām mašīnām.

Virtuālā mašīna – vide, kurā strādā mitināta (*hosted*) operētājsistēma un kas piedāvā īpaši izdalīta datora abstrakciju.

Domēni – izpildes konteksti, kas satur virtuālo mašīnu.

Dom0 (Domain 0) – apzīmējums domēnam, kurš pārvalda sistēmu un citus domēnus.

DomU – apzīmējums viesim – ne-Dom0 domēnam.

Domēna ID – unikāls domēna identifikators.

1.3. XEN BŪTĪBA

Xen ir paravirtualizējošs virtuālo mašīnu monitors (VMM). Xen nenodrošina pilnu virtualizāciju. Viss domēnu darbs ar aparatūru iet caur hipervīzoru. Domēnu operētājsistēmām ir nepieciešams īpašs kodols, kas ir pielāgots Xen-am un kam jāspēj strādāt ārpus procesora 0-tā līmeņa, izsaukumus aparatūrai nododot hipervīzoram. Tā kā kodols ir viena no operētājsistēmas svarīgākajām un kritiskākajām sastāvdaļām, tā pielāgošana darbam Xen sistēmā ir atstāta attiecīgo operētājsistēmu autoriem. Xen

izstrādātāji ir izveidojuši līdzekļus un metodi Linux kodola pielāgošanai, kas ir visai viegli iekļaujami dažādos distributīvos. Linux viņiem ir kā vide izstrādei un prototipu demonstrēšanai. Pašlaik FreeBSD un NetBSD izstrādātāji strādā pie šo operētājsistēmu pielāgošanas Xen sistēmai. Visdrīzāk, ka tajās Xen atbalsts būs kā opcija vai papildu funkcija – līdzīgi, kā tas ir Linux-ā.

Xen sistēma spēj izmantot arī aparātisko virtualizāciju (AMD, Intel jaunakajos procesoros), ko sauc par ”asistēto virtualizāciju” un kas stipri vienkāršo DomU pielāgošanu. Tādā veidā var arī palaist Microsoft Windows operētājsistēmu kā DomU virs Linux Dom0. Tomēr kaut kādas modifikācijas DomU operētājsistēmā ir nepieciešamas. Gluži jebkuru sistēmu tādā veidā vēl nevarot palaist.

1.4. XEN IESPĒJAS

Domēni strādā ar minimālu veiktspējas zudumu. Visvieglāk iet ar skaitļošanas uzdevumiem. Tā kā domēni Dom0 hipervīzorā tiek plānoti (*schedule*) un “paralēli” izpildīti līdzīgi kā paralēlās programmas tradicionālā daudzuzdevumu operētājsistēmā, to konkurence par procesora resursu ir labi plānota un ar tik niecīgiem zudumiem, kas salīdzināmi ar mērījumu kļūdu.

Grūtāk ir ar domēniem, kuru programmas aktīvi izmanto datora perifērijas ierīces, piemēram, diskus vai tīkla kartes. Tā kā visi izsaukumi uz ierīcēm iet caur papildu emulācijas slāni, un daudzu domēnu vienlaicīgs darbs ar ierīci prasa īpašu ierīces koplietošanas plānošanu, šis process kopumā palēnina darbu jau par jūtamiem 5-10%. Lai gan ļoti bieži šo veiktspējas zudumu ar lielu pārsvaru atsvērs virtualizācijas izmantošanas priekšrocības.

DomU sistēmas ”diskam” var izmantot failu ar diska sadaļas (*partition*) attēlu vai LVM/EVM sadaļu. Linux satur visus nepieciešamos rīkus sadaļu attēlu radīšanai, formatēšanai, montēšanai, kā būs redzams tālāk šinī nodaļā. LVM/EVM sadaļas var kopēt strādājošam domēnam bez veiktspējas zudumiem un nezaudējot datu integritāti, nodrošinot biežas ”karstās” kopijas – no strādājošām sistēmām. LVM/EVM atbalsts ir visos mūsdienu Linux distributīvos, vairākos to var izmantot jau instalēšanas gaitā.

Tipisks DomU satur 3 failus:

1. Konfigurācijas fails – apraksta domēna parametrus.
2. Kodola fails – Linux kodols lietošanai DomU operētājsistēmā (tam nav nekāda sakara ar Dom0 sistēmu, tas ir tikai fails, kas pieder attiecīgajam DomU).
3. Saknes un apmaiņas (*swap*) failsistēmu attēli, katrai failsistēmai – savs attēla fails.

DomU sistēmu saglabāšanai un migrēšanai ir pieejami vairāki varianti:

- Apstādinātā veidā – līdzīgi kā, klonējot apstādinātu operētājsistēmu no viena diska uz otru.
- Hibernētā veidā (save/restore), kad strādājoša sistēma ir “iesaldēta”. Pēc atjaunošanas to atliek “pamodināt”, un tā turpina strādāt ar visām atvērtajām programmām.
- Dzīvajā (*live*) veidā (pārsūta strādājošu sistēmu), kad pilns strādājošas sistēmas stāvoklis tiek pārsūtīts uz citu datoru pat bez “iesaldēšanas” un uzreiz pēc tam to aizver vienā datorā un atver – otrā, kur tā turpina strādāt, it kā nekas nebūtu bijis, saglabājot atvērtas programmas, savienojumus utt. Xen autori apgalvo, ka viņu DomU migrēšanas metode nodrošina migrējamās sistēmas nepieejamību tikai uz 60-300ms.

1.5. DomU IZVEIDOŠANA

Lai radītu DomU sistēmu, jāizveido vismaz divi faili:

- Viesa konfigurācijas fails direktorijā /etc/xen;
- Failsistēmas attēla fails saknes (/) sadaļai ar ext3 vai citu Linux failsistēmu. Tā atrašanās vieta nav svarīga. Tajā jāiekopē vai jāieinstalē vēlamās operētājsistēmas faili.

Ieteicams izveidot arī vienu apmaiņas (*swap*) sadaļas attēla failu, bet tas nav obligāti. Turpmākajās darbībās autors to neizmantos.

Lai izveidotu nepieciešamās failsistēmas attēlus, lieto tālāk redzamās komandas.

Izveidot 700MB lielu failu /usr/local/serv1.ext3 saknes failsistēmas attēlam:

```
sudo dd if=/dev/zero of=/usr/local/serv1.ext3 bs=1024k seek=700 count=0
```

Formatēt:

```
sudo mkfs.ext3 /usr/local/serv1.ext3
```

```
sudo mkdir /mnt/serv1
```

Piemontēt:

```
sudo mount /usr/local/serv1.ext3 /mnt/serv1 -o loop
```

Ieinstalēt Ubuntu Feisty (7.04) bāzes sistēmu:

```
sudo debootstrap feisty /mnt/serv1 http://lv.archive.ubuntu.com/ubuntu
```

Iekopēt kodola moduļus:

```
sudo cp -a /lib/modules/2.6.19-4-server/ /mnt/serv1/lib/modules/
```

Pēc šī posma sistēma jau ir sāknējama. Bet vēl nepieciešams veikt dažas kritiskas noskaņošanas darbības.

Jāieraksta failā /mnt/serv1/etc/network/interfaces sekojošo informāciju interfeisiem lo un eth0:

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet dhcp
```

Iekopēt `hosts` failu:

```
sudo cp /etc/hosts /mnt/serv1/etc/hosts
```

Ierakstīt hostvārdu:

```
sudo nano /mnt/serv1/etc/hostname
```

Ierakstīt failā /mnt/serv1/etc/fstab vismaz divu svarīgāko failsistēmu datus:

```
proc            /proc  proc  defaults  0  0
```

```
/dev/sda1  /  ext3  defaults,errors=remount-ro 0  1
```

Atmontēt failsistēmas attēlu, lai attiecīgais DomU tiek pie tā rakstīšanas režīmā:

```
sudo umount /mnt/serv1
```

Tālāk jāveido DomU konfigurācijas fails /etc/xen/serv1.cfg ar sekojošu informāciju:

```
kernel = "/boot/vmlinuz-2.6.19-4-server"
```

```
ramdisk = "/boot/initrd.img-2.6.19-4-server"
```

```
builder='linux'
memory = 128
name = "serv1"
vcpus = 1
vif = [ 'bridge=xenbr0' ]
disk = [ 'file:/usr/local/serv1.ext3,ioemu:sda1,w' ]
root = "/dev/sda1 ro”
```

Ja vēlas virtuālajai tīkla kartei noteikt MAC adresi, jāmaina “vif” rindiņa pret šādu (kur “00:16:3e” ir Xen rezervēts OUI, bet “01:02:03” ir patvaļīgs seriālais numurs):

```
vif = [ 'bridge=xenbr0, mac=00:16:3e:01:02:03' ]
```

Lai domēns tiktu aktivizēts datora startēšanā, tā konfigurācijas failu (vai saiti) jāieliek direktorijā /etc/xen/auto/

Minētie parametri ir labi eksperimentiem, bet tie būtu jāmaina reālām sistēmām. Vismaz jāparedz lielāka operatīvā atmiņa (tā ir maksimālā vērtība, ko iegūs DomU) vai *swap* sadaļa.

Konfigurācijā var norādīt tieši izmantojamās PCI ierīces parametrus, padarot darbu ar ierīci ātrāku, bet ekskluzīvu dotajam DomU. Var izmantot īpašus dziņus, piemēram, ja tos neatbalsta bāzes operētājsistēma.

Failsistēmai var izmantot arī NFS eksportu. Tādā gadījumā ir citi “disk” parametri. Bet NFS izmantošana nav ieteicama drošības problēmu dēļ.

1.6. DOMĒNU DARBINĀŠANA

Lai apskatītu atvērto domēnu sarakstu, lieto komandu:

```
sudo xm list
```

Lai aktivizētu DomU:

```
sudo xm create /etc/xen/serv1.cfg
```

Lai aktivizētu viesi ar konsoles piesaisti (uzreiz pieslēdzas virtuālajai konsolei, redz startēšanas paziņojumus, var sākt strādāt):

```
sudo xm create /etc/xen/serv1.cfg -c
```

Pievienoties viesa konsolei jau strādājošam DomU:

```
sudo xm console serv1
```

Atvienoties no konsoles – Ctrl+5

Apstādināt viesi (*shutdown*):

```
sudo xm shutdown serv1
```

Pārtraukt viesi (”atvienot elektrību”):

```
sudo xm destroy serv1
```

1.7. DOMĒNU SĀKOTNĒJAIS STĀVOKLIS

Iepriekš aprakstītās metodes rezultātā DomU sistēmai sākumā ir tukša root parole. Tādēļ jāpiesakās ar root un jāuzstāda paroli:

```
passwd root
```

Intereses pēc autors apskatās aizņemto diska vietu un operatīvo atmiņu:

`df` – ~300MB izmantoti failsistēmā

`top` – ~28MB izmantotas operatīvās atmiņas

Ja tīklā, kam pieslēgts dators ir DHCP serveris, tīklam jau jāstrādā ar uzstādīto konfigurāciju. Lai par to pārliecinātos, izpilda šādas komandas:

Interfeisu parametru konstatēšanai:

`ifconfig` – redzam eth0 ar definēto MAC adresi un ar DHCP iegūtajiem IP parametriem

Pārbaudīt tīkla darbību:

```
ping www.lumii.lv
```

1.8. DOMĒNU PAPILDINĀŠANA

Šajā posmā būtu viesu sistēmās jāieinstalē un jānoskaņo tajās paredzētās lietojumu programmatūras.

Piemēra pēc autors uzstādīs kādu servera pakotni. Tā būs Tīmekļa servera programma lighttpd. Tā aizņem maz vietas un atmiņas. Tās faili ir zem /var/www/, bet žurnāls – /var/log/lighttpd/access.log. Tās instalēšana:

```
aptitude install lighttpd
```

Parasti serveros nepieciešama arī OpenSSH servera programma. To pieinstalē ar šādu komandu:

```
aptitude install openssh-server
```

Jāatceras, ka Ubuntu sistēmā OpenSSH serveris pēc noklusēšanas ļauj autentificēties root lietotājam. Tādēļ ieteicams to aizliegt un izveidot nepriviliģētu lietotāju, ielikt to admin grupā un lietot virtuālā servera administrēšanai.

2. UBUNTU 7.04 INICIALIZĀCIJAS SKRIPTI

2.1. UPSTART SISTĒMA UN SKRIPTI

Ubuntu izstrādātāji ir piedāvājuši alternatīvu Linux startēšanas sistēmu ar nosaukumu Upstart, kas aizvieto System V. Upstart izveides mērķis bija padarīt sistēmas ielādes procesu elastīgāku, vieglāk maināmu un atkļūdojamu. Viņi piedāvā atteikties no *runlevel* līmeņiem un to vietā izmantot darbus (*jobs*), kas ļauj sadalīt startēšanas procesu mazākos darbiņos un kombinēt tos pēc vajadzības. Katram darbam var definēt notikumus (*events*), kas izsauks tā palaišanu vai apstādināšanu. Tā, piemēram, darbam var pateikt palaisties pēc konkrēta cita darba pabeigšanās, vai ieejot kādā *runlevel* līmenī.

Upstart sistēmas komplektā ir alternatīvs */sbin/init* dēmons, kas regulē darbu izpildi, un t.s. darbu apraksti, kas atrodas direktorijā /etc/event.d/ un būtībā ir īpašā sintaksē aprakstīti skripti. Tā kā daudzas programmas un bibliotēkas Ubuntu operētājsistēmā vēl joprojām izmanto *runlevels* koncepciju, tā ir saglabāta savietojamībai. Un pašlaik Upstart darbi vienkārši palaiž visu to pašu, ko palaistu iešana pa System V *runlevels* līmeņiem. Tātad šobrīd Upstart vēl nav aizvietojis System V, bet ir ielicis papildus slāni un pārvietošanos pa līmeņiem ir pārtaisījis par notikumiem. Toties jau tagad ieliktajā Upstart slānī ir viegli mainīt startēšanas procesu.

2.2. UPSTART PIELĀGOŠANA 1 DARBA IZPILDEI

Tas būtu nepieciešams situācijās, kad kādai instalācijai jāveic tikai viena funkcija, kas realizējama kā viens izpildes fails – binārs vai skripts, iztiekot bez vairākiem paralēliem pakalpojumiem līdzīgi vienlietotāja (*single user*) režīmam. Tikai vienlietotāja režīms inicializēs sistēmas nepieciešamos skriptus un atvērs čaulu, bet autors šinī eksperimentā mēģina iztikt pat bez tā. Būtībā no šīs eksperimentālās instalācijas autors vēlās tikai divas lietas: lai palaiž kodolu ar dziņiem un vienu skriptu.

Tā kā *Upstart* sistēma ir visai viegli pielāgojama, autors pēc nelieliem eksperimentiem ar virtuālo mainīšanu ķērās klāt izdzēst vai nomainīt “start on” uz nereālu notikumu visiem darbiem. Un izveidoja jaunu darbu rc-noklus ar sekojošu saturu:

```
start on startup
stop on runlevel
console output
script
      echo rc-noklus 1
      /root/slodze.sh
      echo rc-noklus 2
      /sbin/halt -f
end script
```

Darbu rc-noklus Upstart sistēma aktivizēs uzreiz pie tās palaišanās, apstādinās pie jebkura *runlevel*. Tā saturiskā daļa ir izvadīt pa teksta rindiņai pirms un pēc vajadzīgās programmas (turpmāk – Programma) un apstādinās sistēmu.

Programmas saturs šoreiz ir vienkāršs – pamēģināt “papingot” vienu serveri un izdrukāt procesu koku:

```
#!/bin/bash
echo
/bin/ping -c 1 www.delfi.lv
```

```
echo

echo Talak seko pasreiz palaisto procesu saraksts:

pstree

echo
```

Gan šajā eksperimentā, gan turpmāk, autors izpildīja operētājsistēmas Xen virtuālajās mašīnās, lai process notiktu raitāk, un būtu vieglāk konstatēt sistēmas darbības rezultātus.

Šī sistēma deva labus rezultātus. Tiešām, pēc kodola ielādes un inicializēšanas (/scripts/..) izpildījās tikai darbs rc-noklus. Viss process no virtuālās mašīnas palaišanas līdz izzušanai ilga apmēram 5 sekundes, no kurām lielāko daļu aizņēma kodola un dziņu ielāde. Tas nozīmē, ka ar optimizētu kodolu visu sistēmas dzīves ciklu var saīsināt līdz pāris sekundēm vai mazāk (!).

Tālāk var redzēt virtuālās mašīnas ielādes pēdējās 30 rindas (pirmās 110 ir kodola ielādes paziņojumi):

```
   111 Begin: Running /scripts/local-premount ...
   112 Done.
   113 [18009.801648] kjournald starting.  Commit interval 5 seconds
   114
[18009.801665] EXT3-fs: mounted filesystem with ordered data mode.
   115
Begin: Running /scripts/local-bottom ...
   116 Done.
   117 Done.
   118 Begin: Running /scripts/init-bottom ...
   119 Done.
   120 rc-noklus 1
   121
   122 ping: unknown host www.delfi.lv
   123
   124 Talak seko pasreiz palaisto procesu saraksts:
   125 init-+-events/0
   126      |-khelper
   127      |-ksoftirqd/0
   128      |-kthread-+-aio/0
   129      |         |-kblockd/0
   130      |         |-kjournald
   131      |         |-kseriod
   132      |         |-kswapd0
   133      |         |-2*[pdflush]
   134      |         |-xenbus
   135      |         `-xenwatch
   136      |-migration/0
   137      |-sh---slodze.sh---pstree
   138      `-watchdog/0
   139
   140 rc-noklus 2
   141 [18011.885255] System halted.
```

Var redzēt, ka ping nav izdevies, jo nav inicializēs tīkls, un ka procesu koks sastāv no kodola procesiem un pProgrammas skripta slodze.sh. Kaut ko līdzīgu varētu lietot situācijās, kad nepieciešama tikai lokāla programmas izpilde bez datortīkla starpniecības.

2.3. UPSTART PIELĀGOŠANA 1 DARBA IZPILDEI AR TĪKLU

Lai Programmai būtu iespējams strādāt tīklā, piemēram, saņemot vai nosūtot darbības rezultātus vai strādājot kā serverim, nepieciešams pirms tās izsaukšanas inicializēt tīkla parametrus. Autors noteica 3 skriptus, kuri jāizsauc, lai pilnvērtīgi inicializētu tīkla parametrus. To izsaukumus (atvēršanu un aizvēršanu) autors pierakstīja darba rc-noklus skriptam, pie viena tajā ieliekot nepieciešamās programmas darbības:

```
start on startup

stop on runlevel
```

```
console output
script
        echo rc-noklus 1
        /etc/init.d/mountkernfs.sh start
        /etc/init.d/loopback start
        /etc/init.d/networking start
        echo rc-noklus 2
        ping -c 2 www.delfi.lv
        /etc/init.d/networking stop
        /etc/init.d/loopback stop
        echo rc-noklus 3
        /sbin/halt -f
end script
```

Šādas sistēmas palaišana jau izskatījās nedaudz savādāk:

```
   111 Begin: Running /scripts/local-premount ...
   112 Done.
   113 [20394.229314] kjournald starting.  Commit interval 5 seconds
   114
[20394.229458] EXT3-fs: mounted filesystem with ordered data mode.
   115
Begin: Running /scripts/local-bottom ...
   116 Done.
   117 Done.
   118 Begin: Running /scripts/init-bottom ...
   119 Done.
   120 rc-noklus 1
   121  Files under mount point '/var/run' will be hidden.
   122  * Starting basic networking...       [ OK ]
   123  * Configuring network interfaces...
[20394.464338] NET: Registered protocol family 10
   124
[20394.464596] lo: Disabled Privacy Extensions
   125
[ OK ]
   126 rc-noklus 2
   127 PING www.delfi.lv (62.85.117.94) 56(84) bytes of data.
   128 64 bytes from 62.85.117.94: icmp_seq=1 ttl=57 time=7.13 ms
   129 64 bytes from 62.85.117.94: icmp_seq=2 ttl=57 time=8.75 ms
   130
   131 --- www.delfi.lv ping statistics ---
   132 2 packets transmitted, 2 received, 0% packet loss, time 1007ms
   133 rtt min/avg/max/mdev = 7.130/7.942/8.755/0.817 ms
   134  * Deconfiguring network interfaces...       [ OK ]
   135  * Stopping basic networking...       [ OK ]
   136 rc-noklus 3
   137 [20397.642169] System halted.
```

3. NOBEIGUMS

Ubuntu Upstart sistēma ļauj viegli optimizēt sistēmas startēšanas procesu. Autors piedāvā divus minimašīnu pielietojumus:

- Ja izpildāmais Uzdevums ir lokāla programma, kas savu darbības rezultātu izvada standarta izvadē, nav nepieciešams inicializēt nevienu sistēmas skriptu vai procesu. Sistēma var sākt izpildīt Uzdevumu uzreiz pēc kodola ielādes. Un arī tās apstādināšana notiek praktiski momentāni ar komandu "`halt -f`", kas līdzinās strāvas atvienošanai datoram reālajās (nevirtuālajās) sistēmās. Uzdevuma pārcelšana uz šādi optimizētu DomU sistēmu prasa papildus apmēram 5 sekundes un 15MB operatīvās atmiņas. Toties ieguvums ir papildus drošība, resursu ierobežošana, un nav jānodrošina Uzdevuma savietojamība ar bāzes operētājsistēmu vai distributīvu.
- Ja Uzdevums ir tīkla programma, piemēram, Tīmekļa serveris, papildus nepieciešami daži inicializācijas skripti, kas palēnina ielādi par mazāk kā sekundi un paņem pāris MB vairāk operatīvās atmiņas. Arī tas autoram šķiet pilnīgi pieņemams resursu zudums, salīdzinot ar ieguvumiem, ko dod servera izmantošana atsevišķā virtuālajā mašīnā.

Virtualizācijas tehnoloģiju var ļoti veiksmīgi izmantot grid skaitļošanas tīklos. Nav nepieciešams katru skaitļošanas programmatūru pielāgot un noskaņot darbam grid sistēmā izmantotajam Linux distributīvam. Vajadzīgo programmatūru var ieinstalēt virtuālajā mašīnā kopā ar tai nepieciešamo operētājsistēmu, un šādu virtuālo mašīnu var piegādāt skaitļošanas mezglam kā vienu lielu failu. Tas atvieglotu specifisku programmu izmantošanu gridtīklā un uzlabotu to izolēšanas un kontrolēšanas iespējas. Šajā darbā piedāvātās minimašīnas labi parāda, ka laika un atmiņas zudums programmas palaišanai kopā ar visu operētājsistēmu ir niecīgs. Līdz ar to virtualizācijas izmantošanai grid infrastruktūrā ir vairāk priekšrocības, nekā trūkumi. Autors piedāvā ieviest virtualizāciju grid skaitļošanas klāsteros kā papildu iespēju specifisku programmu darbināšanai.