УДК 004.4(043.2)

В.А. Рябоконь, В.В. Клобуков,
Ю.В. Максименко
(Національний авіаційний університет, Україна, м. Київ)

ВІРТУАЛІЗАЦІЯ РОБОЧИХ СТАНЦІЙ НА БАЗІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВІДКРИТИМ ПРОГРАМНИМ КОДОМ

Розглянуто основні компоненти для системи віртуалізації робочих станцій з використанням програмного забезпечення з відкритим програмним кодом

Віртуалізація робочих станцій останнім часом займає потужний шар програмного забезпеченння. Такі потужні гравці на ринку програмного забезпечення, як Vmware, Microsoft, Citrix мають в своєму портфелі рішення для організації систем віртуалізації робочих станцій. Лідером серед таких програм безперечно є програмне забезпечення VMware View. Свій вклад в розвиток віртуалізації робочих станцій вносить і спільнота програмного забезпечення з відкритим прграмним кодом. Так, компанія RedHat пропонує програмне забезпечення Red Hat Enterprise Virtualization for Desktops, побудоване на базі програмного забезпечення з відкритим прграмним кодом Ovirt. Ovirt використовує такі основні програмні засоби, як операційна система Linux, гіпервізор KVM (Kernel-based Virtual Machine), емулятор QEMU, протокол SPICE(Simple Protocol for Independent Computing Environments), система управління libvirt, розподілена файлова система GlusterFS та інше програмне забезпечення з відкритим програмним кодом.

KVM

KVM (Kernel-based Virtual Machine) являє собою модуль ядра Linux для процесорів з розширеними функціями з підтримки віртуалізації Intel VT або AMD-V. KVM також може працювати на системах на базі сучасних процесорів PowerPC та S390. З використанням KVM можливий запуск багатьох віртуальних машин з немодифікованими операційними системами. Підтримуються операційні системи Windows, Linux, FreeBSD, OpenBSD, OpenSolaris та інші. Для емуляції віртуальних машин і надання доступу до віртуальних машин використовується емулятор з відкритим програмним кодом QEMU. Перевагою KVM є те, що цей гіпервізор є частиною ядра Linux і тому використовує всі переваги Linux. Одним з важливих компонентів KVM є система паравіртуалізаціїї пристроїв virtio. Повна емуляція пристроїв вводу виводу потребує великих затрат від процесорів системи. Для зменшення цих затрат використовуєть паравіртуалізація пристроїв з використанням virtio. Virtio являє собою систему абстракції пристроїв. Ця система складається з back-end і front-end драйверів. Back-end драйвери працюють на стороні гіпервізора і надають стандартизований набір обладнання для роботи гостьових систем. front-end драйвери надають доступ гостьовій операційній системі до пристроїв, які були надані з використанням back-end драйверів. Крім back-end і front-end драйверів virtio визначає два рівні для взаємодії між гіпервізором і гостьовою системою. На верхньому рівні(який називають virtio) діє інтерфейс віртуальних черг, який приєднує front-end драйвери до back-end драйверів віртуальної машини. Драйвери можуть використовувати декілька черг, або не

використовувати їх взагалі, в залежностиі від необхідності. Наприклад, драйвер мережевої карти virtio використовує дві черги – одну для отримання даних, іншу – для передачі.
Сучасні процесори включають функцію передачі систем вводу-виводу(Intel (VT-d *(Virtualization Technology for Directed I/O)*) і AMD (IOMMU (*I/O Memory Management Unit)*), яка дозволяє відображати адреси фізичних пристроїв вводу-виводу PCI в віртуальні середовища. За допомогою цієї технології віртуальна машина отримує доступ до фізичного пристрою вводу-виводу.

Рис. 1. Структура роботи пристроїв virtio

Важливою функцією сучасних систем віртуалізації, що підтримується KVM є міграція гостьових систем з одного сервера віртуалізації на інший без зупинки цих систем і їх сервісів. Це необхідно при побудові систем віртуалізації, що складаються з декількох гіпервізорів і обслуговують велику кількість гостьових систем а також для побудови систем хмарних обчислень. В процесі такої міграції віртуальна машина зупиняється, виконується передача параметрів віртуальної машини з гіпервізора-передавача на гіпервізор – отримувач, потім передається образ оперативної пам’яті з гіпервізора-передавача на гіпервізор – отримувач, після чого на гіпервізорі – отримувачі запускається передана віртуальна машина з переданим образом оперативної пам’яті. Такий процес міграції може бути непомітним для користувача, оскільки займає небагата часу, хоча це залежить від того, який об’єм оперативної пам’яті використовується віртуальною машиною, а також швидкістю передачі даних між гіпервізорами.

QEMU

KVM надає можливість віртуалізації на рівні ядра. Але для надання доступу до віртуальної машини, а також для емуляції пристроїв вводу-виводу,

що не підтримуються систтмою virtio, або передаванням фізичних пристроїв з використанням технології VT-d або IOMMU необхідне використання додаткового програмного забезпечення. Сьогодні основним програмним забезпеченням, що виконує такі функції є ємулятор з відкритим програмним кодом QEMU.
QEMU — це машинний емулятор, за допомогою якого можна запускати віртуальні машини різних конфігурацій і платформ. QEMU підтримує емуляці таких архітектур як x86, ARM, ETRAX CRIS, MIPS, MicroBlaze, PowerPC та SPARC. З використанням KVM на обладнанні, що підтримує функції віртуалізації, QEMU використовує ці функції віртуалізації. З використанням QEMU можливий запуск різних операційних систем на гостьовій віртуальній машині: Linux, Microsoft Windows, FreeBSD, DOS, Solaris.
Крім повної емуляції віртуальної машини QEMU може працювади в користувацькому режимі, коли запускається лише одна програма, що створена для роботи на іншій платформі і операційній системі:
qemu-i386 -L / /bin/ls
Існує можливість запуска Linux-систем не з образу диска, а, наприклад ядра, що знаходиться на файловій системі, що можна викоритовувати для тестування нових функцій ядра системи.
qemu-system-i386 -kernel arch/i386/boot/bzImage -hda root-2.4.20.img -append "root=/dev/hda"
Як вже було сказано вище, QEMU використовує інтерфейс virtio для надання гостьовим системам високопродуктивних пристроъв вводу виводу. Крім того в разі необхідності QEMU може емулювати інші пристрої : USB-hub, мережеві карти, графічні планшети, відеоадаптери, дискові пристрої. Для роботи мережевих пристроїв і з’єднання мережі віртуальної машини з реальною мережею QEMU викоритовує такі технології як VLAN(Virtual Private Network) і TUN/TAP. За допомогою цих технологій виконується мережевий зв’язок між віртуальними машинами, віртуальними машинами і фізичною машиною та між віртуальними машинами і реальною мережею. Для налаштування мережі можуть використовуватись вбудовані DHCP-сервер та TFTP-сервер.

Glusterfs

Для розгортання системи віртуалізації з викоритсанням декількох серверів необхідно використовувати систему зберігання інформації, що дозволяє надавати доступ з усіх серверів системи. В системі ovirt використовується файлова система GlusterFS. GlusterFS - це масштабована, розподілена файлова система з можливістю захисту від збоїв. GlusterFS працюючи поверх TCP / IP (і інших підтримуваних мережевих транспортних протоколів) об'єднує виділені дискові простору вузлів мережі в одну паралельних мережеву файлову систему розміри якої можуть нарощуватися до декількох петабайт. GlusterFS працює в користувацькому оточенні ОС через інтерфейс FUSE (Filesystem in Userspace). Це полегшує розробку і налагодження FS, а також дозволяє захистити ядро ОС від помилок в реалізації драйверів ФС. GlusterFS використовує рівень абстракції поверх конкретної

реалізації файлової системи, іншими словами використовує реальну файлову систему всередині себе. Серед пригідних до використання файлових систем, поверх яких працює GlusterFS - ext3, ext4, ZFS, xfs, jfs. Архітектура GlusterFS розділена на клієнтську і серверну частини. Складається з таких частин:brick (цегла) - локальна файлова система виділена під том зберігання; client (клієнт) - машина яка монтує томи (може виступати одночасно і як сервер); server (сервер) - машина яка надає частину своєї файлової системи для клієнта; subvolume (підтем) - brick після обробки як мінімум одним транслятором, надається клієнтові; voluve (тому) - Готова ФС зібрана на клієнті з підтомів. Ключовим елементом архітектури GlusterFS є так звані "транслятори" - програмні модулі обробляючі та модифікуючі запити на файлові операції. По своєму функціональному призначенню транслятори діляться на:
- Транслятори зберігання даних (відповідають за взаємодію з базовою ФС на вузлах);
- Транслятори передачі даних (мережева взаємодія);
- Транслятори підвищення продуктивності (кешування);
- Транслятори додаткової функціональноті (блокування, права доступу);
- Транслятори кластеризації - найбільш важливий клас трансляторів, який реалізує алгоритми розподілу даних по вузлам.

Транслятори підключаються один за одним утворюючи різні режими роботи, основними з яких є:
distribute - розподіляє файли за наявними підтомами за допомогою хеш-функції. Наприклад один файл записується на першу ноду, другий на другу, третій на третю і так далі. З мінусів можна відзначити що при виході з ладу одного сервера, вся інформація що зберігається на ньому стає недоступною. З плюсів - чим більше додається серверів тим швидше стає доступ до файлів; збільшення тому відбувається за рахунок додавання нового серверу.
replicate - мережевий RAID-1. У цьому режимі кожен сервер зберігає весь набір файлів, запис стає трохи повільніше, зате швидкість така ж як з локального диска, плюс - додається відмовостійкість.
stripe - мережевий RAID-0. Файл розбивається на шматочки які рівномірно розподіляються по підтомам. Плюс - висока швидкість, що особливо відчутно на великих файлах. Мінус - не можливо додати новий підтом.

Висновки

В системі віртуалізації робочих місць Ovirt використовуються компоненти, що дозволяють створити потужну і надійну систему. Всі компоненти системи мають відкрити код, що дозволяє використовувати їх без додаткового навантаження на бюджет і модифікувати під власні потреби.

Список літератури

1. *C. Arnold, M. Rode.* KVM Best Practices.- Берлін, dpunkt.verlag -2012.-301с.
2. Gluster File System 3.3.0, 2012-134с.