UNITECH 2017 GABROVO

INTERNATIONAL SCIENTIFIC CONFERENCE
17-18 November 2017, GABROVO

СРАВНИТЕЛЕН АНАЛИЗ НА ПЛАТФОРМИ ЗА ВИРТУАЛИЗАЦИЯ И КОНТЕЙНЕРИЗАЦИЯ ПРИ ОБРАБОТКА НА ГОЛЕМИ ОБЕМИ ДАННИ

PERFORMANCE STUDY OF VIRTUALIZATION AND CONTAINERIZATION PLATFORMS FOR BIG DATA PROCESSING

Daniel Trifonov
Technical University of Varna

Hristo Valchanov
Technical University of Varna

Abstract

For the processing of unstructured, semi-structured and structured data as well as research data of historical and statistical nature, the traditional relational database management systems are not a rational choice and have been replaced by other solutions such as systems to work with large volumes of data - the Big data. For this purpose, large-scale distributed data processing systems such as Apache Hadoop have been widely used. Building such a system requires the availability of multiple machines, which is a serious investment even for the large companies. The use of virtualization platforms can solve a number of existing problems. Increasingly, however, an attractive technology emerges as an alternative to virtualization - the containerization technology. Containers solve some of the problems typical of hypervisors and virtual machines. This paper presents a study of the performance of large data processing system implemented on virtual machines and containers.

Keywords: Big data, Apache Hadoop, Virtualization, Containerization

ВЪВЕДЕНИЕ

Обемът на данните, които се генерират и съхраняват ежедневно расте с все по-бързи темпове. Цялото това количество информация е необходимо да бъде не само съхранявано, но и обработвано. Данните стават все по-разнообразни по вид и все по-малко структурирани. В случаите на обработка на неструктурирани, полу-структурирани и структурирани данни, както и при данни от научни изследвания, исторически и статистически характер, все по-често традиционните релационни системи за управление на бази от данни не са рационален избор и са заместувани от други решения като системите за работа с големи обеми от данни – Big data. Big data се използва не само от всички софтуерни гиганти като Google, Microsoft, AWS, Facebook, Ebay, Booking, New York Stock Exchange, но и от много на брой помалки компании и изследователски центрове [1].

Десктоп инструментите за анализ и релационните бази от данни често имат проблеми с обработката на големи количества разнородни данни, затова на помощ идват силно-разпределените системи за обработка на данни като Apache Hadoop [2]. Hadoop представлява клъстерна система от малък брой управляващи и голям брой изчислителни и съхраняващи данните възли. Тази силна разпределеност позволява изчисленията върху огромни обеми от данни да стават за изключително кратко време, като всеки възел обработва само намиращите се локално при него данни.

Изграждането на такава една система е сериозна инвестиция дори и за големите компании, като се има предвид, че за да има истинска полза от разпределената обработка са необходими няколко десетки машини. Използването на виртуализационни платформи (ВП) може да разреши редица от съществуващите проблеми. Това може да бъде реализирано в няколко насоки. Тези платформи предоставят икономични от финансова страна среди изчисления, редуцират необходимостта от инвестиции в ново оборудване. В същото време ВП намаляват и цената на поддръжката, изразяваща се в

преинсталиране на операционни системи и софтуер. Все по-широко обаче, навлиза атрактивна технология, явяваща се алтернатива на ВП - контейнеризационната технология. Контейнерите решават някои от проблемите, типично характерни за хипервайзорите и виртуалните машини. Поради тяхната опростена архитектура те предоставят по-добра производителност от виртуалните машини. Същевременно позволяват по-бързо и гъвкаво предоставяне на ресурси и наличност на нови инстанции на приложенията.

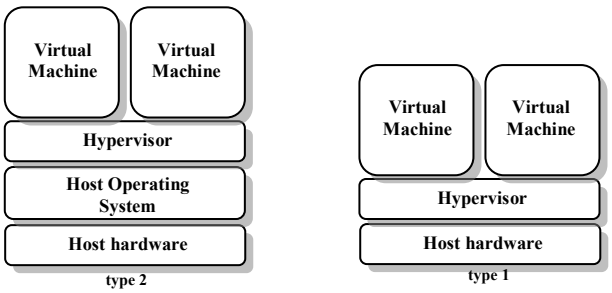
Настоящият доклад представя сравнителен анализ на производителността на системи за обработка на големи обеми данни, изградени на базата на виртуални машини и на контейнери.

ВИРТУАЛИЗАЦИЯ

Виртуализацията се дефинира като „абстрактно представяне на физическите ресурси на компютъра във виртуален компютър с помощта на специализиран софтуер" [3]. Виртуализационната платформа виртуализира хардуерните ресурси на компютрите. Тя създава напълно функционална виртуална машина (ВМ), върху която може да се изпълняват самостоятелна операционна система (ОС) и приложения, точно както при реален компютър.

Виртуализацията работи с помощта на специален софтуер – хипервайзор, който разпределя хардуерните ресурси на машината динамично и прозрачно между виртуалните машини. Няколко операционни системи могат да работят едновременно върху един физически компютър и да споделят хардуерни ресурси един с друг. Чрез капсулиране на цялата машина, включително процесор, памет, операционна система и мрежови устройства, виртуалната машина е напълно съвместима с всички стандартни x86 операционни системи, приложения и драйвери.

Известни са два основни типа виртуализационни среди: hosted и bare-metal среди (фиг. 1).



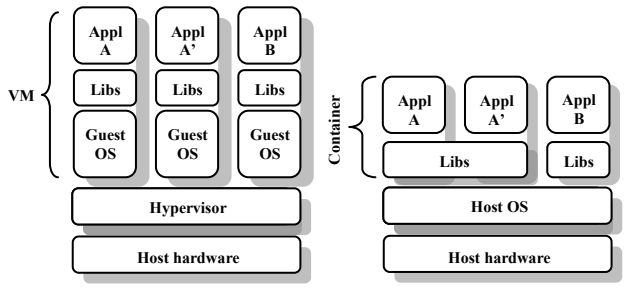
***Фиг. 1**. Виртуализационни среди*

В hosted среди хипервайзорите (type 2) се явяват софтуерни приложения, работещи в рамките на операционната система на компютъра. Хипервайзорът контролира ресурсите, които са заделени от операционната система на долното ниво. Този тип хипервайзори основно се използват в системи, където има нужда от различни входно/изходни устройства, които могат да бъдат поддържани от хост операционната система и в клиентски системи с ниска ефективност. Пример за такъв тип хипервайзори са: Microsoft Virtual Server [4], VMware Server и VMware Workstation [5].

В bare-metal среди хипервайзорите (type 1) са софтуерни системи, които работят директно върху хардуера на хоста. По този начин се постига по-висока виртуална ефективност и производителност. Този тип хипервайзори са предпочитан подход за виртуализация. Пример за подобни хипервайзори са: Citrix XenServer [6], VMware ESXi [7], Microsoft Hyper-V [8].

КОНТЕЙНЕРИЗАЦИЯ

Разликата между виртуализация и контейнеризация е основно в мястото на виртуализационния слой и начинът, по който системните ресурси се използват. Контейнеризацията, наричана още „контейнерно базирана виртуализация", "паравиртуализация" или "виртуализация на приложения", представлява виртуализационен метод за внедряване и изпълнение на разпределени приложения на нивото на операционната система, без необходимостта от стартиране на цяла виртуална машина за всяко приложение. Вместо това, множество изолирани системи, наречени контейнери, са изпълнявани на един единствен хост и достъпват ядрото на операционната система (фиг. 2).



***Фиг. 2**. Системи с ВМ и с контейнери*

Контейнерът представлява лек, самостоятелен, изпълним пакет софтуер, който включва всичко необходимо за изпълнението му: код, библиотеки, системни приложения и настройки. Софтуерът вътре в контейнера се изпълнява по един и същи начин, без значение от средата. Контейнерите изолират софтуера от обкръжаващата среда, например между средата за разработка и работната среда. Те спомагат за намаляване на конфликтите между екипите, които използват различен софтуер върху една и съща инфраструктура.

Тъй като контейнерите споделят едно и също ядро на операционната система заедно с хост машината, то те могат да бъдат по ефективни от виртуалните машини, които пък изискват отделни операционни системи. Хост операционната система ограничава достъпа на контейнера до физическите ресурси като процесор и памет, така един контейнер не може да изконсумира всички системни ресурси.

Контейнерите работещи на една и съща машина споделят ядрото на операционната система. Те се стартират по-бързо и използват по-малко изчислително време и памет в сравнение с виртуалните машини. Контейнерите споделят общи файлове което намалява употребата на дисково пространство.

Съществуват редица контейнерно- базирани решения, като Linux-VServer [9], OpenVZ [10], Docker [11].

ОБРАБОТКА НА ГОЛЕМИ ОБЕМИ ДАННИ

Една от широко използваните системи за обработка на големи обеми данни е Hadoop. Hadoop е софтуер (framework) с отворен код, разработен на Java. В Hadoop има множество инструменти за изпълнение на код и скриптове на различни програмни езици. Той се състои от две основни части - част за съхранение и част за обработка на данни. Частта за съхранение е разпределената файлова система Hadoop Distributed File System (HDFS). Тя има блокова организация, като всеки блок е с размер от 256MB. Блоковете, които съставляват един файл са разпределени по всички възли на разпределената система. С цел постигане на резервираност Hadoop поддържа репликации на блоковете. Хранилището на данни е Hive. Hive предлага лесно обобщаване на данни, временни заявки и други анализи на големи количества данни. За заявките се използва език, подобен на SQL, известен като HiveQL.

Частта за обработка на данни е MapReduce. Това е програмен модел за паралелна обработка на множество задачи. Един от основните аспекти на работата на MapReduce програмирането е, че MapReduce разделя задачите по такъв начин, че позволява тяхното паралелно изпълнение върху разпределена система от изчислителни възли. Противоположно на традиционните системи за управление на релационни бази от данни, които не могат да нараснат, за да обработят големи количества данни, програмирането в средата на Hadoop MapReduce позволява на организациите да изпълняват приложения върху огромен брой машини, което също включва и обработката на хиляди терабайти данни.

ЕКСПЕРИМЕНТАЛНИ ИЗСЛЕДВАНИЯ И РЕЗУЛТАТИ

Като платформа за виртуализация е избран хипервайзорът VMware ESXi 6.5, а за контейнеризация – Docker. Изборът на тези две платформи е продиктуван от тяхното широко използване, високата им производителност и възможности.

Тестовата среда е изградена върху сървърна система DELL CS24-TY, която разполага със следния хардуер:

- 2 броя, 4-ядрени Xeon L5520 с Hyper Threading;
- 72 GB оперативна памет;
- 1,8 TB дисково пространство.

На тази машина се инсталират последователно двете платформи и върху тях се из-

пълняват съответните тестове за производителността на Big data системата.

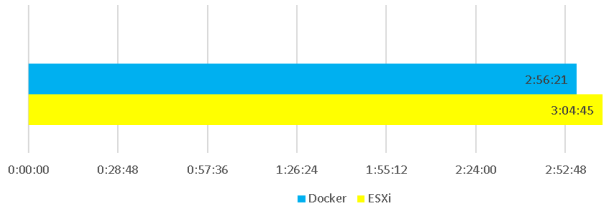
Върху VMWare са инсталирани 5 отделни Ubuntu 16.4 LTS (64-битови) виртуални машини. Във виртуалните машини има инсталирана Hadoop система, която е разпределена на 3 изчислителни възела (data nodes), 1 главен (name node) и един спомагателен (secondary name node).

Docker е инсталиран върху Ubuntu 16.4 LTS (64-битова) операционна система, като са създадени 5 отделни контейнера. В контейнерите е поместена същата Hadoop система.

Тестовете са така подбрани, че да покрият изискванията към различните хардуерни устройства: процесор, памет и устройства за съхранение.

Първият тест тества производителността на системата при обработка на данни в полуструктуриран вид. Такава обработка се налага постоянно в Big data системите, понеже входните данни доста често идват от разнородни източници и са в различни формати. Тестът е Java приложение (WordCount), което се изпълнява директно от Hadoop и извиква MapReduce. MapReduce обхожда файла и разделя текста на отделни думи, като премахва пунктуацията – map фазата на задачата. Следващата фаза е reduce. Тя редуцира резултата до файл със съдържание от двойки име–стойност, указващи за всяка срещната дума колко пъти е открита. Данните за обработка са един от последните архиви на Wikipedia (EN) като е използвано само съдържанието на английски език. То се предоставя като bz2 архив, в който има един xml файл с размер приблизително 62GB.

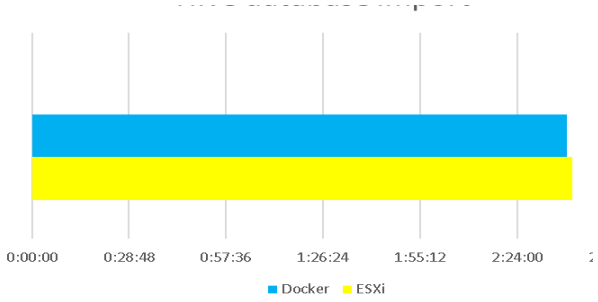
На фиг. 3 са показани резултатите от теста.



***Фиг. 3**. Резултати от тест WordCount*

Резултатите показват предимство от ~4,5% за контейнеризацията и Docker. Това е очаквано, имайки предвид, че по време на теста се използва максимално количество памет, а всяка виртуална машина заема по приблизително 1,3GB памет, която иначе се използва от Hadoop. Достъпът до дисковото пространство при виртуализацията е потенциално незначително по-бавен, което също леко увеличава преднината на Docker. Друг важен момент е дългото време за изпълнение на теста. Това се дължи на ниската производителност на дисковете на системата и е възможно при по-високоскоростни устройства, резултатите да се различават от получените в тази тестова постановка.

Вторият тест е комплексен по отношение на процесорна обработка и входно/изходни операции – Hive dataset import. Този тест използва отново MapReduce. Той прочита съдържанието на текстовия файл *enwiki-20170701-pages-articles-multistream.xml* и го импортира в таблица на нерелационна база от данни – MongoDB. Колоните на таблицата са 2, като първата е заглавието на статията, а втората е целия текст. Обработката на текста изисква процесорно време и памет, докато четенето от файла и записът в базата от данни натоварват дисковете. Записът изисква 3 пъти повече входно-изходни операции, заради репликационния фактор. Резултатите са показани на фиг. 4.

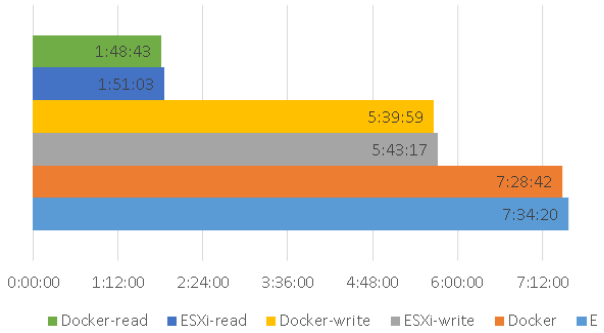


***Фиг. 4**. Резултати от тест Hive dataset import*

Следва да се отбележи, че този тест не дава ясен резултат за предимство на виртуализацията или контейнеризацията. Противна по-бързо от първия тест, понеже в базата от данни текстът се съхранява в различен формат, а и обработката на данните се свежда единствено до извличане от текста и запис в базата. Липсват фазите на сортиране и комбиниране на сортираните резултати. Разликата във времето за изпълнение на WordCount и Hive database import не е мно-

го голяма отново заради спецификата на дисковото пространство. Този тест би трябвало да приключи много по-бързо, ако за всеки data node има отделен твърд диск или дори SSD диск. В подобно изследване с тестови системи, изградени с реални и виртуални машини, при които върху една реална се стартират две виртуални, виртуалните се представят по-добре, понеже уплътняват използването на процесора, докато при реалните той не е натоварен напълно във всеки един момент от теста. В настоящото изследване не се наблюдава подобно поведение, тъй като използваният хардуер е една единствена компютърна система и натоварването и в двата случая е пълно.

Третият тест служи за оценяване на системата за съхранение. Той се изпълнява в две части: TestDFSIO-write записва 100GB данни във файловата система на Hadoop – HDFS, а TestDFSIO-read ги прочита обратно. Поради репликационния фактор, тестът за запис прави 3 пъти повече входно-изходни операции отколкото теста за четене и генерира значителен мрежов трафик. Резултатите са представени на фиг. 5.



***Фиг. 5**. Резултати от тест TestDFSIO*

При този тест резултатите са доста по-интересни. Ясно се вижда, че тестовете със запис са около 3 пъти по-бавни от тези с четене. Това е нормално и е така именно заради репликационния фактор и спецификата на тестовия клъстър. Факторът за репликация е 3, а точно толкова са и data node-овете, т.е. всяка изчислителна машина поема целия обем данни на своя диск, докато при четенето се чете едновременно и от трите машини. В случай че репликационния фактор остане 3 а изчислителните възли са повече, то разликата във времената за четене и запис ще бъде пропорционално по-малка. При големи системи със стотици изчислителни машини реално няма разлика между скоростите на четене и запис, въпреки, че репликационния фактор там може да достигне дори 5. Леко предимство има отново постановката с контейнеризация поради факта, че обръщенията към диска през виртуалния дисков контролер се забавят малко повече отколкото през интерфейса, предоставен от контейнеризацията.

ЗАКЛЮЧЕНИЕ

На базата на проведените тестове се установява, че контейнеризацията дава малко по-добри резултати в повечето случаи.

Ако вече се разполага с инсталиран сървър с хипервайзор, то не е нужно да се преинсталира с една единствена операционна система и контейнери, тъй като разликата в производителността не е толкова голяма. Освен това, в хипервайзора могат да се ползват и други виртуални машини, докато клъстерът не се използва. Следва да се отбележи, че като цяло тестовете преминават много по-бавно от очакваното. Причината е в споделените 2 диска за всичките 5 виртуални машини в единия случай, и контейнери в другия. Като препоръка при реализация на система от този тип е добре да бъдат ползвани поне по един диск и по-точно SSD диск за всяка виртуална машина или съответно контейнер. Тогава би се усетила по-реално производителността на Hadoop системата. Основната разлика в производителността идва от наличието на по-голямо количество използвана памет при системата с контейнери, защото в противния случай тя е заета от операционните системи на виртуалните машини (5 x 1,3GB) и от хипервайзора (2,26GB).

При изграждането на нови системи е изгодна реализация с контейнери, тъй като ефективността на използване на хардуера е малко по-добра, а освен това се улеснява и администрирането на системата – само една операционна система за поддръжка.

REFERENCE

[1] Want To Use Big Data? https://www.forbes.com/sites/bernardmarr/2017/08/

14/want-to-use-big-data-why-not-start-via-google-facebook-amazon-etc/#5dd460173d5d.
[2] T. White. Hadoop: The Definitive Guide. O'Reilly, 2015.
[3] J. Drews. Going Virtual, Network Computing, Vol. 17, No. 9, p. ES 5, 2006.
[4] Microsoft Virtual Server. https://www.microsoft.com/windowsserversystem/virtualserver/.
[5] VMware. https://www.vmware.com.
[6] Xen Server. https://xenserver.org/
[7] ESXi. https://www.vmware.com/products/esxi-and-esx.html.
[8] Server Virtualization. http://www.microsoft.com.
[9 Linux-VServer. http://linux-vserver.org/.
[10] OpenVZ Linux Containers Wiki. http://openvz.org/.
[11] Docker – Build, Ship, and Run Any App, Anywhere. http://www.docker.com/.