STUDIA INFORMATICA 2011
Volume 32 Number 3A (98)

Dariusz CZERWIŃSKI, Sławomir PRZYŁUCKI, Daniel SAWICKI
Politechnika Lubelska, Wydział Elektrotechniki i Informatyki

WYDAJNOŚĆ SYSTEMÓW OPERACYJNYCH W ŚRODOWISKACH WYKORZYSTUJĄCYCH WIRTUALIZACJĘ

Streszczenie. Artykuł prezentuje porównanie wydajności systemów operacyjnych uruchamianych w różnych środowiskach wirtualizacji. Główny nacisk położony jest na porównanie wydajności wirtualnego systemu oraz systemu uruchamianego w środowisku chmurowym.

Słowa kluczowe: wydajność systemu, wirtualizacja, sieci chmurowe, Eucalyptus

OPERATING SYSTEMS EFFICIENCY IN VIRTUALIZATION ENVIRONMENTS

Summary. The paper presents performance comparison of operating systems running on different virtualization environments. The main focus is to compare the performance of a virtual system and the system running in an cloud computing environment.

Keywords: systems efficiency, virtualization, cloud computing, Eucalyptus

1. Wprowadzenie

Przetwarzanie w chmurze (ang. *Cloud computing*) stało się obecnie popularne wśród naukowców i inżynierów zajmujących się tematyką IT. Bardzo szybko wzrasta liczba użytkowników usług uruchamianych w chmurze, a korzystanie z możliwości przechowywania danych czy też uruchamiania aplikacji w środowiskach chmurowych staje się praktycznie standardem dzisiejszego Internetu.

Proces przetwarzanie w chmurze może być realizowany na trzech różnych poziomach, utożsamianych z trzema modelami usług. Są to odpowiednio: oprogramowanie jako usługa

(ang. *Software as Service – SaaS*), platforma jako usługa (ang. *Platform as Service – PaaS*) oraz infrastruktura jako usługa (ang. *Infrastructure as Service – IaaS*) [2, 4].

Struktura i zależności, jakie występują pomiędzy wymienionymi poziomami przetwarzania w chmurze są przedstawione na rys. 1. W niniejszym artykule skoncentrowano się na podstawowym poziomie funkcjonalnym w strukturze chmury, jakim jest infrastruktura jako usługa. Poziom ten, w tym modelu usług, skupia się na dostarczeniu urządzeń (serwerów, pamięci masowej i infrastruktury sieciowej) i oprogramowania (najczęściej użytkowego, które opiera się na wirtualizacji i rozproszonym systemie plików) w postaci usługi. Takie podejście pozwala na dostęp do zasobów w momencie, kiedy są one najbardziej potrzebne. Przykładem tego modelu usługi jest Amazon Web Services Elastic Compute Cloud (EC2) oraz Secure Storage Service (S3), a także ich darmowy odpowiednik Eucalyptus.

Rys. 1. Warstwy przetwarzania w chmurze jako usługi
Fig. 1. Cloud computing levels as services

2. Rozwiązania systemów chmurowych

Systemy chmurowe, tak jak każdy inny system komputerowy, opierają się na ściśle zdefiniowanym modelu wykonywania zadań, przenoszenia i przechowywania danych i komunikacji pomiędzy elementami składowymi systemu. Elastyczność przetwarzania w chmurze, a także związana z nią iluzja nieograniczonych zasobów wynikają z odpowiednio zorganizowanego procesu multipleksowania zadań i przydziału zasobów odpowiedzi na żądania obsługi ze strony użytkowników. Aby móc zrealizować taki sposób przetwarzania danych, w obecnej chwili, konieczne jest sięgnięcie po rozwiązania z obszaru wirtualizacji. Wirtualizacja jest zatem podstawą osiągnięcia celów, ale również wydajności sieci chmurowych i jest to najbardziej widoczne w realizacji zadań objętych definicją usług IaaS.

W tabeli 1 przedstawiony jest różny sposób podejścia do przetwarzania w chmurze, w zależności od sposobu wirtualizacji, metod obsługi pamięci masowej i algorytmów łączności w najpopularniejszych obecnie środowiskach chmurowych [4].

Tabela 1

Zestawienie wybranych środowisk wirtualnych opartych na chmurach

Firma	Amazon	Microsoft	Google
Nazwa	Amazon Web Service	Azure	AppEngine
Specyficzne cechy sprzętowo-programowe	Architektura x86 (32 i 64 bitowa), wirtualizacja oparta o środowiska KVM lub Xen	Maszyny wirtualne oparte o CLR (ang. *Microsoft Common Language Runtime*)	Wykorzystuje zamkniętą architekturę aplikacji i definicji zasobów
Model obsługi pamięci masowych	Cała gama modeli, od EBS (ang. *Elastic Block Store*) do zaawansowanych modeli przechowywania typu key/ blob (SimpleDB, S3) wraz z zestawem dedykowanych API	SQL Data Services (zmodyfikowany SQL Server) lub Azure storage service (rozwiązanie zamknięte)	MegaStore/BigTable (rozwiązanie w praktyce zamknięte)
Algorytmy komunikacji pomiędzy elementami chmury	Wykorzystanie grup bezpieczeństwa, Elastic IP dostarczający publicznych adresów IP dla każdego żądania dostępu do zasobów, wirtualizacja sieci na poziomie warstw 2 oraz 3	Automatycznie organizowana na podstawie opisu komponentów żądanej aplikacji	Stała topologia komunikacyjna, ukierunkowana na 3-warstwowy model struktury aplikacji Web

2.1. System Eucalyptus

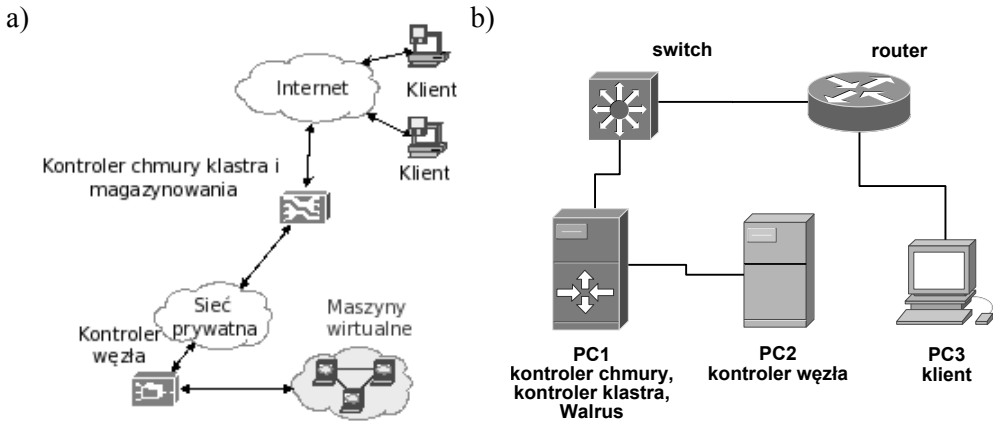
System Eucalyptus pozwala na budowę struktur chmur prywatnych, zgodnych ze standardem Amazon EC2. Dzięki temu, użytkownicy tego systemu korzystają z zasobów chmury w ten sam sposób, jak zasoby chmury publicznej, oferowanej przez Amazon. Pozwala to na prostą integrację tego rozwiązania w ramach projektów chmur hybrydowych [3]. Struktura systemu Eucalyptus składa się z czterech elementów [3, 5]:

- kontrolera węzła NC (ang. Node Controller) – jest to zasób fizyczny (najczęściej pojedynczy host), na którym uruchamiane są poszczególne instancje maszyn wirtualnych,
- kontrolera klastra CC (ang. Cluster Controller) – urządzenie to spełnia trzy funkcje: przydziela zasoby NC dla zadań, zarządza instancjami maszyn wirtualnych, a także zbiera informacje o procesie realizacji zadań i wykorzystaniu zasobów,
- kontrolera magazynowania Walrus (ang. Storage Controller) – ten element przechowuje obrazy maszyn wirtualnych i dane użytkowników,
- kontrolera chmury CLC (ang. Cloud Controller) – ten element odpowiada za dostęp do zasobów zgłaszanych przez użytkowników oraz za planowanie przydziału zasobów do zgłoszonych żądań.

Podstawą do budowy środowiska testowego były system Eucalyptus, a jego poszczególne elementy zostały zaimplementowane za pomocą komputerów klasy PC. Komputery te były wyposażone w dwurdzeniowy procesor firmy Intel, ze wsparciem dla sprzętowej wirtualizacji, 2 GB pamięci RAM, pamięć dyskową o pojemności 250GB oraz 2 interfejsy sieciowe. Podstawą realizacji wirtualizacji był pakiet KVM. Poszczególne elementy struktury testowego systemu Eucalyptus zostały przypisane do, pokazanego na rys. 2b, sprzętu komputerowego w następujący sposób:

- kontroler klastra, kontroler magazynu, kontroler chmury oraz Walrus (CC, SC, CLC, WS) – komputer PC1 (2 rdzenie, 4 procesory logiczne, współdzielone), 2 GB RAM, 250 GB HDD SATA, system Fedora 12 (x86_64),
- kontroler węzła (NC) – komputer PC2 (2 rdzenie, 4 procesory logiczne, współdzielone), 2 GB RAM (współdzielona z VM), 250 GB HDD SATA, system Fedora 12 (x86_64),
- klient – komputer PC3 (1 rdzeń, 2 procesory logiczne, współdzielone), 512 MB RAM, 8 GB HDD SATA, system Fedora 12 (x86_32).

Architektura środowiska testowego została przedstawiona na rys. 2.



Rys. 2. Architektury środowiska testowego: a) logiczna, b) fizyczna
Fig. 2. Testbed architecture: a) logical, b) physical

3. Wyniki pomiarów

Na tak zbudowanej platformie wykonano testy, mające na celu zbadanie wydajności systemów wykorzystujących wirtualizację. Ich wydajność testowana była na następujących urządzeniach:

- komputer klasy PC (2 rdzenie), 2 GB RAM, 250 GB HDD SATA, system Fedora 12 (x86_64)) – platforma testowa do badań oznaczonych w dalszej części artykułu jako „Fedora”,

- system Ubuntu 9.04 uruchomiony w maszynie wirtualnej KVM węzła środowiska Eucalyptus Cloud, VM (1 rdzeń), 0.12 GB RAM, 5 GB HDD SATA, system gospodarza Fedora 12 (x86_64) – platforma testowa do badań oznaczonych jako „VM”,
- kontroler chmury Eucalyptus Cloud, Cloud (1 rdzeń), 0.12 GB RAM, 5 GB HDD SATA, system Ubuntu 9.04 (x86_64) – platforma testowa do badań oznaczonych w dalszej części artykułu jako „Cloud”.

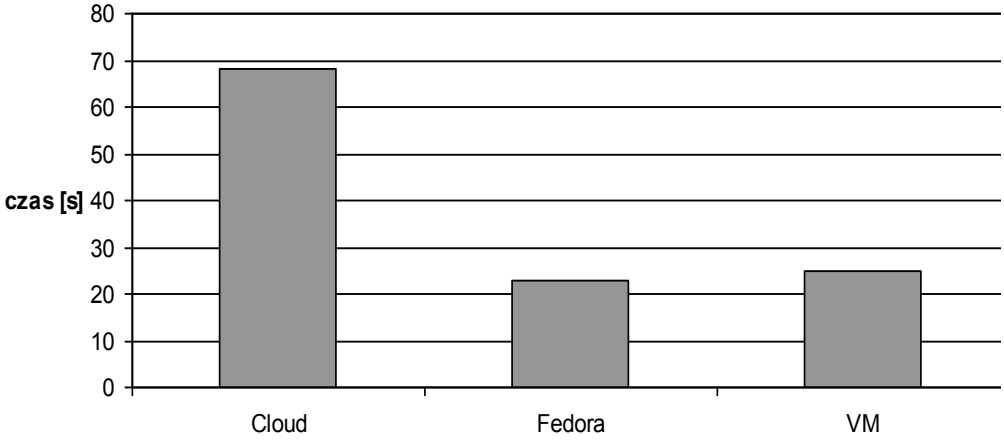
Testy wydajnościowe były uruchamiane odpowiednio:

- na komputerze z zainstalowanym system operacyjnym Fedora, wyniki testów posłużyły jako punkt odniesienia w stosunku do systemów wykorzystujących wirtualizację,
- w zainstalowanej chmurze Eucalyptus Cloud, skonfigurowanej w architekturze jednego kontrolera oraz jednego węzła tak, aby wyeliminować wpływ dodatkowych węzłów,
- na systemie Ubuntu uruchomionym w maszynie wirtualnej, tak aby bezpośrednio zbadać wydajność takiego rozwiązania.

W czasie testów stanowiska nie były obciążone dodatkowymi procesami. Autorzy dobrali kilka rodzajów testów tak, aby uzyskać miarodajne wyniki wydajności poszczególnych systemów operacyjnych. Wybrane testy to:

- pakowanie pliku o rozmiarze 318 MB, z użyciem poleceń tar oraz gzip,
- kompilacja pakietu ffmpeg,
- benchmark Dhrystones,
- benchmark Linpack.

Wyniki pakowania pliku z użyciem polecenia *tar* przedstawia rys. 3. Czas pakowania określony został z użyciem polecenia *time* i na wykresie zaprezentowano rzeczywisty okres, w jakim program był uruchomiony (ang. *real time*).

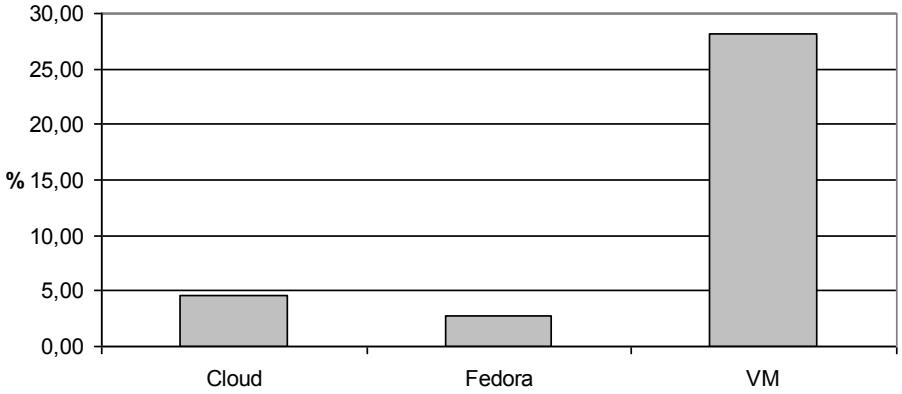


Rys. 3. Czasy pakowania pliku z użyciem poleceń tar oraz gzip
Fig. 3. Packing time of the file with tar-gzip command

Wyraźnie widać, iż operacje plikowe wejścia/wyjścia w chmurze zajmują znacznie więcej czasu, aniżeli w pozostałych systemach. Potwierdzeniem tego faktu jest przedstawienie tak

zwanego narzutu systemowego. Narzut systemowy został zdefiniowany jako procentowa wartość, w postaci stosunku czasu pracy procesora w trybie systemu do czasu, jaki procesor rzeczywiście poświęcił na przetwarzanie programu w trybie procesu uruchomionego, gdzie procesor pracował w trybie użytkownika (sys/user·100%).

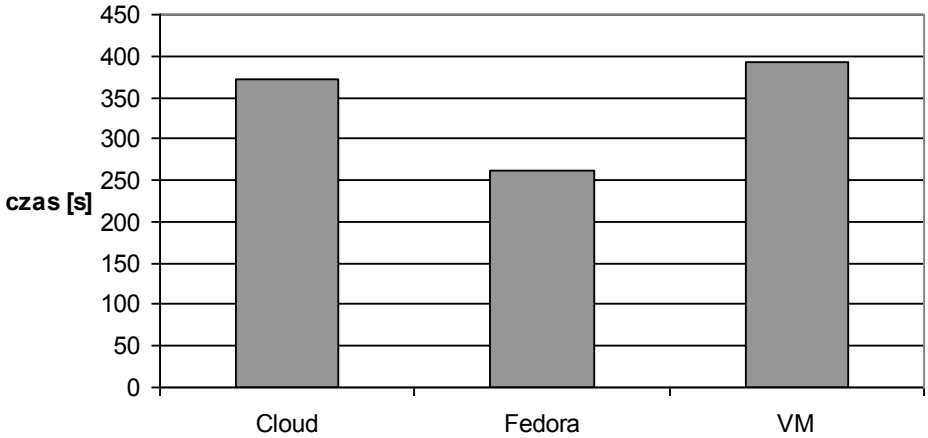
W przypadku tak sporządzonego zestawienia widać (rys. 4), że dla systemów Fedora i Cloud narzut systemowy jest najmniejszy i wynosi od 2% do 4%.



Rys. 4. Narzut systemu dla pakowania pliku z użyciem polecenia tar-gzip
Fig. 4. Overhead of the system for packing the file using tar-gzip command

Kolejnym przeprowadzonym testem była kompilacja pakietu *ffmpeg*. Kompilację przeprowadzono w konfiguracji sprzętowej, opisanej w punkcie 2.1. Wyniki całkowitego czasu kompilacji przedstawiono na rys. 5.

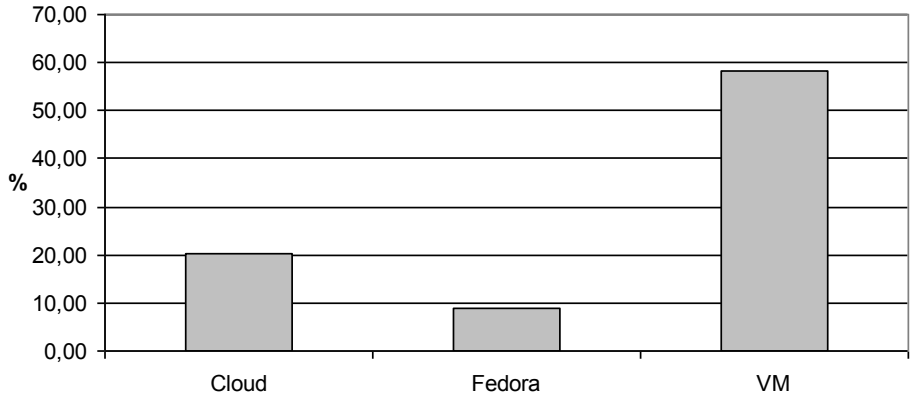
Analizując wyniki czasów kompilacji można zauważyć, że systemy wirtualne są o około 30% wolniejsze w stosunku do natywnie zainstalowanego systemu operacyjnego.



Rys. 5. Czasy kompilacji pakietu ffmpeg
Fig. 5. Total compilation time of the ffmpeg package

Wśród systemów wirtualnych w procesie kompilacji architektura chmury również ma przewagę w stosunku do systemu uruchomionego w maszynie wirtualnej. Na rys. 6 można

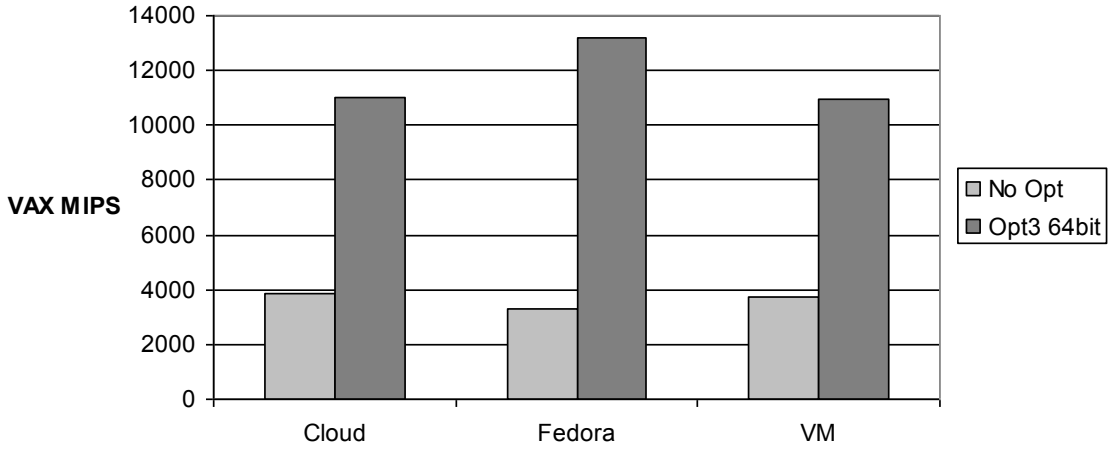
zauważyć wyraźną różnicę pomiędzy systemami Cloud i VM, w przypadku chmury narzut systemu jest około 40% mniejszy w stosunku do maszyny wirtualnej.



Rys. 6. Narzut systemu dla kompilacji pakietu ffmpeg
Fig. 6. Overhead of the system for ffmpeg compilation

W narzut systemowy nie wchodzą operacje wejścia-wyjścia, stąd widać, że architektura chmury Eucalyptus pod względem wydajności operacji, które nie wymagają częstego zapisu danych na nośnikach trwałych, jest często zbliżona do wydajności systemu Fedora, a na pewno znacznie bardziej wydajna w stosunku do samej maszyny wirtualnej.

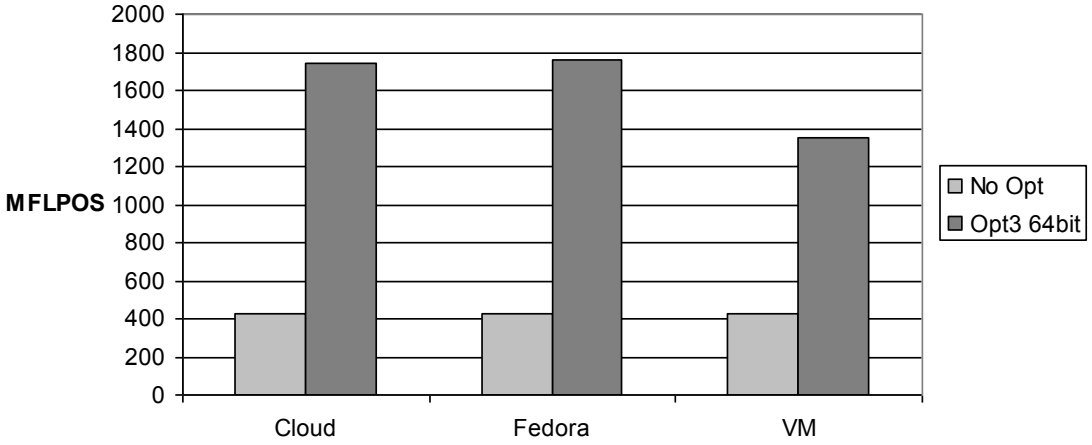
Kolejnym benchmarkiem przeprowadzonym na tak zestawionej platformie testowej był Dhrystones. Na rys. 7 przedstawiono zestawienie otrzymanych wyników dla tego testu.



Rys. 7. Wyniki testu Dhrystones 2.1
Fig. 7. Results of Dhrystones 2.1 test

W obu konfiguracjach (Cloud oraz VM) systemy wirtualne w teście z optymalizacją na architekturę 64-bitową są niewiele gorsze (około 16%) od systemu operacyjnego Fedora. Natomiast w teście bez optymalizacji to Fedora się okazała być gorsza o około 14% w stosunku do pozostałych systemów.

Ostatnim przeprowadzonym testem był benchmark Linpack. Wyniki pomiarów zostały przestawione na rys. 8.



Rys. 8. Wyniki testu Linpack
Fig. 8. Results of Linpack benchmark

Wyniki benchmarku bez optymalizacji są niemal identyczne (różnica jednego MFLOPS-a), natomiast w przypadku optymizacji wyraźnie gorszy jest system uruchomiony w maszynie wirtualnej.

4. Podsumowanie

W artykule dokonano porównania wydajności wirtualnego systemu oraz systemu uruchamianego w środowisku chmurowym. Wydajności te porównano z wydajnością lokalnie zainstalowanego systemu operacyjnego. Po przeprowadzeniu testów można wyciągnąć następujące wnioski.

Operacje obsługi plików w środowisku chmurowym nie są obsługiwane zbyt dobrze. Pakowanie plików z użyciem poleceń tar-gzip jest lepiej obsługiwane przez system uruchamiany w maszynie wirtualnej. System taki jest niewiele wolniejszy w czasie tego testu od lokalnie zainstalowanego systemu operacyjnego Fedora, różnica wyniosła 2,5 sekundy.

Ciekawym aspektem jest w tym teście największy narzut systemu operacyjnego uruchamianego wirtualnie i wynosi on około 28%. Może być to spowodowane obsługą wirtualizacji przez KVM – łączny czas pracy procesora w trybie systemu jest znacznie dłuższy niż w pozostałych przypadkach.

W przypadku testów polegających na kompilacji pakietu ffmpeg czasy są bardzo zbliżone, najlepiej wypada lokalny system Fedora później środowisko chmurowe, a na końcu system wirtualny. W przypadku systemu wirtualnego kompilacja jest wykonywana niewiele dłużej od środowiska chmurowego i jest to różnica około 5%. W przeprowadzonych testach narzut systemu w przypadku systemu uruchamianego w maszynie wirtualnej był zawsze największy i mieścił się w granicach od 20% do 60%.

Benchmark Dhrystones przeprowadzony na platformie testowej wykazał przewagę środowiska chmurowego nad środowiskiem systemu wirtualnego, różnica ta nie była zbyt duża i wynosiła 3%.

Systemy wirtualne dzisiejszych czasów stają się coraz bardziej wydajne, co potwierdzają wyniki benchmarku Linpack. W teście bez optymalizacji wszystkie systemy osiągnęły prawie identyczne wyniki na poziomie 428 MFLOPS, natomiast z optymalizacją systemy Fedora oraz środowisko chmurowe są na zbliżonym poziomie, a system wirtualny miał w tym teście o 22% gorszą wydajność.

BIBLIOGRAFIA

1. Vaquero L. M., Rodero-Merino L., Caceres J., Linder M.: A Break in the Clouds: Towards a Cloud Definition, ACM SIGCOMM Computer Communication Review, Vol. 39, No. 1, 2009, s. 50÷55.
2. Foster I., Zhao Y., Raicu I., Lu S.: Cloud Computing and Grid Computing 360-Degree Compared, Grid Computing Environments Workshop, 2008, s. 1÷10.
3. Nurmi D., Wolski R., Grzegorczyk Ch., Obertelli G., Soman S., Youseff L., Zagorodnov D.: The Eucalyptus Open-source Cloud-computing System, 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), Vol. 0, 2009, s. 124÷131.
4. Armbrust M., Fox A., Griffith R., Joseph A. D., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I., Zaharia M.: Above the Clouds: A Berkeley View of Cloud Computing, UC Berkeley Reliable Adaptive Distributed Systems Laboratory, EECS Department, University of California, Technical Report EECS-2009-28.
5. Johnson D., Murari K., Raju M., Suseendran R. B., Girikumar Y.: Eucalyptus Beginner's Guide – UEC Edition, CSS Corp. 2010, http://www.csscorp.com/eucauecbook, June 2010.
6. Eucalyptus Administrator's Guide (2.0), http://open.eucalyptus.com/wiki/, September 2010.

Recenzenci: Prof. dr hab. inż. Bolesław Pochopień
Prof. dr hab. inż. Tadeusz Wieczorek

Wpłynęło do Redakcji 19 marca 2011 r.

Abstract

Cloud system as any other computer system needs models of computation, storage and communication, respectively. Elasticity and the illusion of infinite capacity, the cloud computing is famous for, come from, generally speaking, the statistical multiplexing of possessed and required resources, according to service demands. That is a reason why these resources need, in most cases, to be virtualized. That concerns all mentioned above models, so the virtualization has strong influence on the efficiency of all IaaS implementations. That is why the paper presents performance comparison of operating systems running on different virtualization environments. The main focus is to compare the performance of a classical virtual system and the system running in an IaaS cloud computing environment. All tests, presented in the article were conducted on the Eucalyptus system. The architecture of the Eucalyptus is simple, flexible and modular with a hierarchical design reflecting common resource environments found in many enterprise cloud computing systems. Moreover, users of Eucalyptus interact with the system using the exact same tools and interfaces that they use to interact with Amazon EC2, one the most popular and well-known cloud solution today.

The testbed consists of all typical elements of private IaaS cloud, Node Controller (NC), Cluster Controller (CC), Storage Controller (Walrus) and Cloud Controller (CLC). The virtualization of the operating systems based on KVM packet and the 2.6 Linux kernel. Four different tests were planed and conducted in order to asses virtualized systems efficiency. They were: files compression based of tar algorithm, ffmpeg compilation, Dhrystones anf Linpack benchmarks. All data from the tests were collected on single host with Fedora Linux, virtualized instance of Ubuntu Linux running on top of Fedora and Ubuntu Linux runimg inside the private Eucalyptus cloud. The performance, of the encountered at the present time solutions of the virtual systems, is considerably closer to the operating system installed locally. The results of the carried out tests show that in some areas the system installed locally performs better, but in the high throughput computing the cloud environment points out.

Adresy

Dariusz CZERWIŃSKI: Politechnika Lubelska, Instytut Podstaw Elektrotechniki i Elektrotechnologii, ul. Nadbystrzycka 38a, 20-618 Lublin, Polska, d.czerwinski@pollub.pl

Sławomir PRZYŁUCKI: Politechnika Lubelska, Katedra Elektroniki, ul. Nadbystrzycka 38a, 20-618 Lublin, Polska, spg@politechnika.lublin.pl

Daniel SAWICKI: Politechnika Lubelska, Katedra Elektroniki, ul. Nadbystrzycka 38a, 20-618 Lublin, Polska, sawi@politechnika.lublin.pl