

# Porównanie systemów przetwarzania w chmurze oraz wirtualizacji sprzętowej

Dariusz Czerwiński, Sławomir Przyłucki, Daniel Sawicki

**Przetwarzanie w chmurze stało się obecnie popularne wśród wielu instytucji oraz przedsiębiorstw związanych z branżą IT. Bardzo szybko wzrasta ilość użytkowników usług uruchamianych w chmurze, a korzystanie z możliwości przechowywania danych czy też uruchamiania aplikacji w środowiskach chmurowych staje się praktycznie standardem dzisiejszego przemysłu internetowego.**

**P**rzetwarzanie w chmurze to nowy i obiecujący model dostarczania usług IT jako narzędzi informatycznych. Chmury są przeznaczone do świadczenia usług dla użytkowników zewnętrznych, a dostawcy zapewniają dostęp do zasobów i rozwiązań chmurowych. Pomysł rynku zbytu dla zasobów informatycznych w branży IT istnieje już od dłuższego czasu. Wiele projektów badawczych, takich jak Sharp, Tycoon, Bellagio, Shirako i PlanetLab, bazuje na pojęciu handlu alokacjami zasobów [1–6].

Firmy w całym sektorze technologii IT zaczynają utożsamiać się z rozwiązaniami chmurowymi. Amazon.com Inc rozpoczął sprzedaż usługi Elastic Compute Cloud w 2006 r. Usługa EC2 polega na wynajęciu przez klienta zasobów gigantycznych komputerów firmy Amazon. Juniper Networks Inc, produkująca sprzęt do transmisji danych, nazwała swój ostatni projekt Stratus. Firmy Yahoo, Intel Corporation i kilka innych niedawno uruchomiły program badawczy o nazwie OpenCirrus [5].

Około 52% informatyków z 13 krajów objętych ogólnostanowym badaniem stwierdziło, że korzysta lub planuje skorzystać z przetwarzania w chmurze. Prognozy przewidują jeszcze wyższe wskaźniki popularności tej technologii w Brazylii (70%), Chinach (69%) i Indiach (76%) [7].

Wydajność jest elementem, który często ma kluczowe znaczenie dla wyboru odpowiedniego oprogramowania wykorzystującego wirtualizację. Jest to również element, który zapewnia, z punktu widzenia użytkownika, wysoki poziom funkcjonalności aplikacji. Należy jednak pamiętać, że wydajność poszczególnych rozwiązań może się znacznie różnić w zależności od rodzaju pracy. Na przykład niektóre systemy wirtualizacji mogą być szybsze w zastosowaniach sieciowych, a inne będą szybciej wykonywały zadania przy dużym obciążeniu CPU [8].

W pracach innych autorów można zauważyć, iż przeprowadzane testy koncentrowały się na sprawdzeniu wydajności rozwiązań wirtualizacyjnych w zastosowaniach serwerowych lub też na ocenie wydajności rozwiązań przetwarzania w chmurze [8–12].

W niniejszym artykule zaproponowano trochę inne podejście, mianowicie porównanie rozwiązań przetwarzania w chmurze z rozwiązaniami wirtualizacyjnymi, w typowych zastosowaniach aplikacyjnych.

**Streszczenie:** Wirtualizacja i przetwarzanie w chmurze (ang. *cloud computing*) mają coraz większe znaczenie w zakresie zastosowań przemysłowych. Technologie te zapewniają lepsze wykorzystanie zasobów serwera, zarządzania serwerem i efektywności energetycznej. Obie technologie są blisko związane i mogą zaoferować ten sam wspólny zbiór zasobów fizycznych dostępnych dla dużej grupy użytkowników o różnych potrzebach. Ponadto przetwarzanie w chmurze wydaje się uzupełnieniem takich rozwiązań, jak klastry i sieci superkomputerów. Podczas stosowania w środowisku przemysłowym czasami trudno jest stwierdzić, czy korzystniej jest stosować technologię przetwarzania w chmurze czy też prostą wirtualizację. W pracy przedstawiono porównanie skuteczności różnych środowisk wirtualizacji. Głównym celem jest porównanie wydajności tych środowisk z punktu widzenia oprogramowania jako całości procesów uruchomionych w środowisku sprzętowo-programowym. Porównanie jest prowadzone na dwóch wirtualnych systemach: pierwszy z uruchomioną maszyną wirtualną KVM, a drugi z uruchomionym środowiskiem chmurowym Open Eucalyptus.

**Słowa kluczowe:** przetwarzanie w chmurze, maszyna wirtualna KVM, Eucalyptus, wydajność systemów wirtualnych.

## COMPARISON OF CLOUD AND VIRTUALIZATION SYSTEMS

**Abstract:** *Virtual machine has a special matter in the computer industry. This technology provides better server utilization, better server management and better power efficiency. The paper presents performance comparison of 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:** *cloud computing, Eucalyptus, virtual systems efficiency.*

## Przetwarzanie w chmurze i wirtualizacja

Technologie chmurowe korzystają z rozwiązań wirtualizacji głównie w celu osiągnięcia korzyści wielokrotnienia przetwarzania i przechowywania danych. Można powiedzieć, że wirtualizacja jest jedynie narzędziem do budowy wydajnych, efektywnych i oszczędnych infrastruktur informatycznych działających w modelu przetwarzania w chmurze. Wpływ ma na to przede wszystkim konsolidacja i wirtualizacja zasobów.

## 1. Przetwarzanie w chmurze

Upraszczając, przetwarzanie w chmurze, według NIST (ang. *National Institute of Standards and Technology*) – określa pięć cech charakterystycznych [13]:

- samodzielność w korzystaniu z usług, tzw. usługa na żądanie (ang. *on-demand*) – użytkownicy mogą instalować i konfigurować oprogramowanie bez pomocy ekspertów;
- swobodny dostęp – usługa dostępna za pośrednictwem standardowych urządzeń sieciowych umożliwiających dostęp do sieci Internet;
- połączenie zasobów niezależne od ich lokalizacji;
- elastyczność – użytkownicy mogą zwiększać lub zmniejszać swoje zasoby w zależności od potrzeb;
- *Pay-per-use* – opłaty pobierane są od użytkowników w zależności od parametrów wykorzystywanej infrastruktury (np. mocy obliczeniowej procesorów, przepustowości łącza internetowego czy dostępnej przestrzeni dyskowej).

Proces przetwarzania w chmurze może być realizowany na trzech różnych poziomach, utożsamianych z trzema modelami usług. Są to odpowiednio: oprogramowanie jako usługa (ang. *Software as a Service* – SaaS), platforma jako usługa (ang. *Platform as a Service* – PaaS) oraz infrastruktura jako usługa (ang. *Infrastructure as a Service* – IaaS) [14, 15].

Model SaaS polega na tym, iż dostawcy mogą dostarczać w chmurze wyłącznie gotowe aplikacje, a zasoby niewykorzystywane w danym momencie przez jednego użytkownika są przydzielane drugiemu. Aplikacje w chmurze dostępne są wyłącznie za pośrednictwem sieci.

Model PaaS bazuje na założeniu, iż platforma daje swobodny dostęp aplikacjom do współdzielonych zasobów, ale wymaga pisania aplikacji przystosowanych do jej wymagań. Na ogół jest to ograniczenie co do stosowanego języka programowania lub wykorzystania API danej platformy.

Modelem najbardziej interesującym z punktu widzenia przedstawionych w artykule badań jest model IaaS. Model ten koncentruje się na podstawowym elemencie funkcjonalnym struktury chmury, jakim jest infrastruktura jako usługa. Rozwiązanie to umożliwia klientom uzyskanie dostępu do zarządcy maszyn wirtualnych, tym samym pozwalając na umieszczanie w chmurze dowolnych systemów operacyjnych i pełną kontrolę nad nimi.

Modelem IaaS skupia się na dostarczeniu urządzeń (serwerów, pamięci masowej, infrastruktury sieciowej) oraz oprogramowania (najczęściej użytkowego, które opiera się na wirtualizacji i rozproszonym systemie plików) w postaci usługi. Takie podejście pozwala na dostęp do zasobów w momencie ich wykorzystania. Przykładem tego modelu usługi jest Amazon Web Services Elastic Compute Cloud (EC2) oraz Secure Storage Service (S3), a także ich darmowy odpowiednik 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 z zasobów chmury publicznej oferowanej przez Amazon. Struktura systemu Eucalyptus składa się z następujących elementów [15, 16]:

- kontroler węzła NC (ang. *Node Controller*) – zasób fizyczny (najczęściej pojedyncze stanowisko), na którym uruchamiane są poszczególne instancje maszyn wirtualnych;
- kontroler klastra CC (ang. *Cluster Controller*) – spełnia trzy funkcje: przydziela zasoby NC dla zadań, zarządza instancjami maszyn wirtualnych, a także zbiera informacje o procesie realizacji zadań i wykorzystaniu zasobów;

- kontroler magazynu SC (ang. *Storage Controller*) – przechowuje obrazy maszyn wirtualnych i dane użytkowników;
- kontroler Walrus – implementuje skalowalną strukturę interfejsu przechowywania danych; obecna implementacja jest zgodna z S3 Amazona;
- kontroler chmury CLC (ang. *Cloud Controller*) – ten element odpowiada za dostęp do zasobów zgłaszanych przez użytkowników oraz za planowanie przydziału zasobów do zgłoszonych żądań.

Systemy chmurowe, tak jak każdy system komputerowy, opierają się o ściśle zdefiniowany model wykonywania zadań, przenoszenia i przechowywania danych oraz komunikacji pomiędzy elementami składowymi systemu. Elastyczność przetwarzania w chmurze wynika z odpowiednio zorganizowanego procesu multipleksowania zadań i przydziału zasobów odpowiedzi na żądania obsługi ze strony użytkowników. Aby móc zrealizować taki sposób przetwarzania danych, w chwili obecnej konieczne jest sięgnięcie po rozwiązania z obszaru wirtualizacji. W rozwiązaniach przetwarzania w chmurze wirtualizacja jest zatem sposobem osiągnięcia celów oraz wydajności sieci chmurowych.

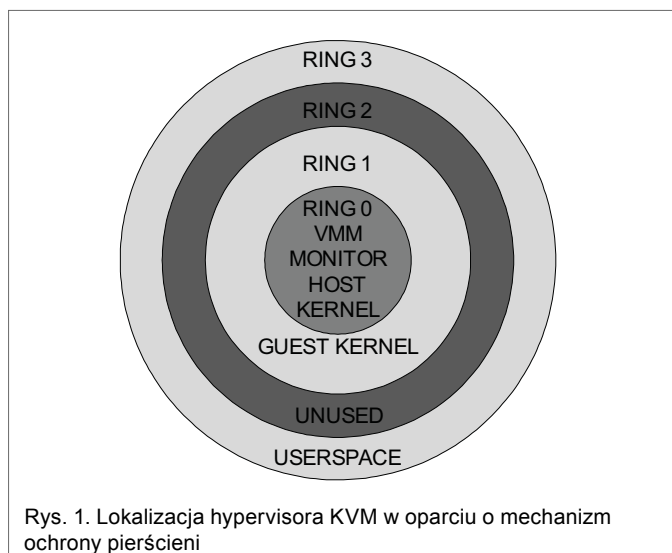
## 2. Wirtualizacja

Wirtualizacja – jest to szerokie pojęcie odnoszące się do rozwiązań programowych i/lub sprzętowych, umożliwiających oddzielenie wykorzystywanych zasobów sprzętowych od systemów i/lub aplikacji, które z tych zasobów korzystają.

Wirtualizacja polega na stworzeniu na systemie gospodarza (ang. *host*) kompletnego wirtualnego środowiska dla systemu gościa (ang. *guest*). Używa się do tego celu specjalnych technologii lub też rozszerzeń sprzętowych i emuluje się tylko to, czego nie da się zrealizować sprzętowo w celu zapewnienia jak największej wydajności systemu gościa. W niektórych zastosowaniach (np. sieci komputerowe) wydajność systemów wirtualnych jest porównywalna z wydajnością natywnie zainstalowanego systemu operacyjnego. Bardzo ważnym elementem struktury jest hypervisor, który jako programowe narzędzie pozwala na jednoczesne działanie wielu systemów operacyjnych na jednym fizycznym komputerze. Można wyróżnić dwa rodzaje hypervisorów:

- Typu 1 – tzw. natywny, który jest uruchomiony bezpośrednio na sprzęcie komputerowym, a w warstwie wyższej są uruchamiane systemy goście (rozwiązania oparte o VMware ESXi, Citrix XenServer czy Microsoft Hyper-V);
- Typu 2 – tzw. hostowany (ang. *hosted*), który jest uruchomiony w systemie operacyjnym gospodarza, a systemy goście są uruchamiane na trzecim poziomie powyżej sprzętu komputerowego [17].

Każdemu z wirtualizowanych systemów „wydaje się”, że ma bezpośredni dostęp do wszystkich podzespołów komputera, chociaż tak naprawdę to kontroluje i zarządza tym hypervisor. Udostępnia on podzespoły komputera, dzieląc je pomiędzy uruchomione systemy gości. Możliwości wirtualizacji są ograniczone do wykorzystania mechanizmu *protection rings*, więc w zależności od tego, jak jest ona realizowana, wykorzystuje ona różne pierścienie owego mechanizmu. KVM jest zarządcą typu 2, czyli działa pod kontrolą systemu operacyjnego, jak każda inna aplikacja. W tym wypadku to system operacyjny pełni rolę gospodarza, mając pod kontrolą cały sprzęt, i udostępnia go jedynie hypervisorowi. Rozpatrując model uprzywilejowanych pierścieni, w którym każdy system



operacyjny ma do wykorzystania cztery pierścienie numerowane od 0 do 3, zarządcą KVM jest umiejscowiony w pierścieniu ring 0 i posiada wszystkie uprawnienia – może bezpośrednio komunikować się ze sprzętem (rys. 1). Pierścieniem ring 3 przeznaczony jest zazwyczaj na tak zwany *userspace* (aplikacje), posiada najmniejsze uprawnienia i to tutaj działają aplikacje [18–20].

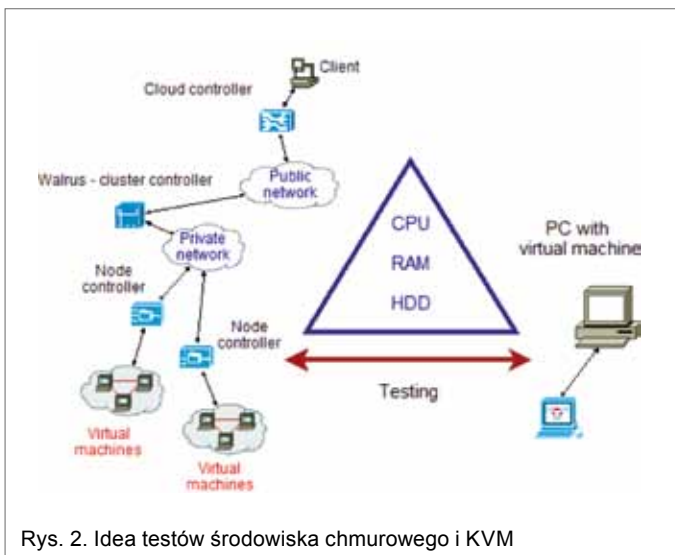
System uprzywilejowanych pierścieni charakteryzuje hierarchiczna budowa, która jest mechanizmem ochronnym zapewniającym bezpieczeństwo uruchamianych procesów.

Jedną z głównych metod realizowania wirtualizacji przez hypervisor typu 2 jest translacja binarna, czyli tłumaczenie instrukcji systemu gościa w locie na instrukcje, które będą wykonane bezpośrednio na procesorze za pomocą systemu gospodarza. Translacja odbywa się pomiędzy instrukcjami jądra gościa, które rezyduje w pierścieniu ring 1, a jądrem systemu gospodarza pracującym w pierścieniu ring 0. Aplikacje – zarówno gospodarza, jak i gościa standardowo urzędują w pierścieniu ring 3. Wiele zarządców implementuje również mechanizm pamięci podręcznej dla zapytań gościa, co jeszcze bardziej usprawnia działanie systemu gościa. Inną również często stosowaną techniką jest kompilacja *just in time*, gdzie instrukcja jest kompilowana podczas jej pierwszego wywołania, a każde następne jej wywołanie powoduje jedynie ponowne uruchomienie już skompilowanej wersji.

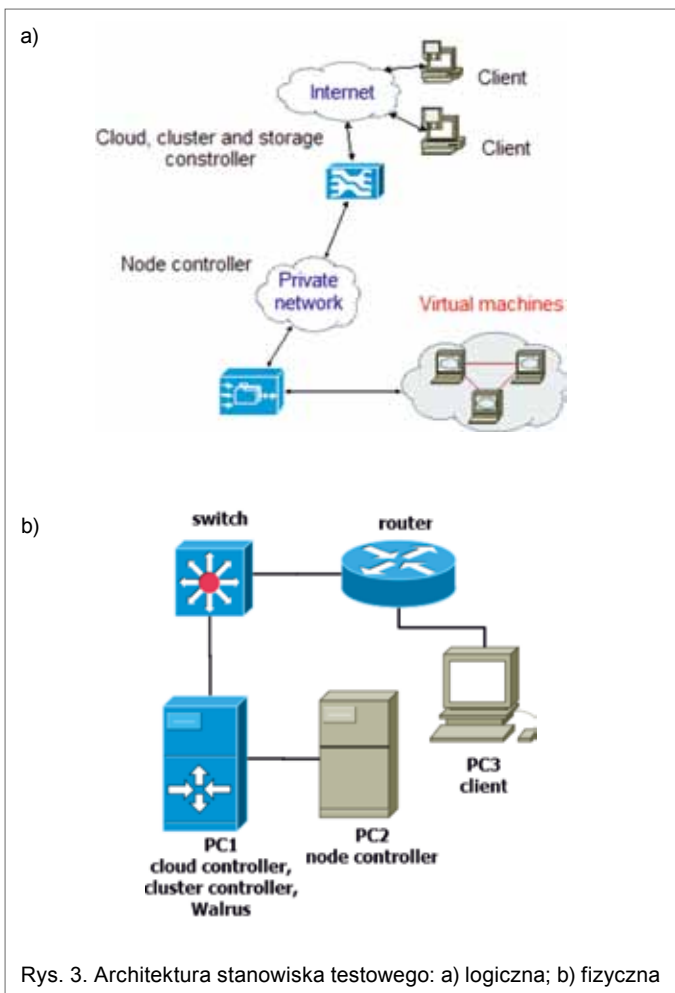
## Metodologia testów

Przeprowadzenie testów wydajnościowych w środowiskach chmurowych i wirtualizacyjnych często sprowadza się do przetestowania najbardziej istotnych elementów, czyli pamięci operacyjnej RAM, jednostki CPU oraz systemu plików. Elementy te są testowane w różny sposób. Mogą być to całosciowe testy systemu gościa typu SiSoft Sandra, Bonnie, LMBench, Phoronix suite, a mogą być to również testy ukierunkowane na pojedynczy element, takie jak: HDTune, IOMeter, RAMtest, Linpack [8–12, 21, 22].

Większość rozwiązań chmurowych opartych o model JaaS opiera się na technologii wirtualizacji i wykorzystaniu hypervisora do uruchamiania wirtualnych serwerów na tym samym sprzęcie komputerowym. Różne hypervisory wspierają całkiem inaczej metody współdzielenia i alokacji CPU. Wśród metod



Rys. 2. Idea testów środowiska chmurowego i KVM



Rys. 3. Architektura stanowiska testowego: a) logiczna; b) fizyczna

można wyróżnić takie, jak stałe, ważne, przerywane i wiele innych. Z tego też powodu trudno jest porównać rozwiązania chmurowe z rozwiązaniami wirtualizacyjnymi, jeżeli posiadają one różnego rodzaju hypervisory [23].

Istnieje też pewna trudność w doborze rodzaju testów oceniających wydajność rozwiązań wirtualnych i chmurowych. Trudno jest odnieść się i porównać wyniki testów w momencie, kiedy zostały one przeprowadzone w wirtualnych systemach operacyjnych, często niemających bezpośredniego dostępu do

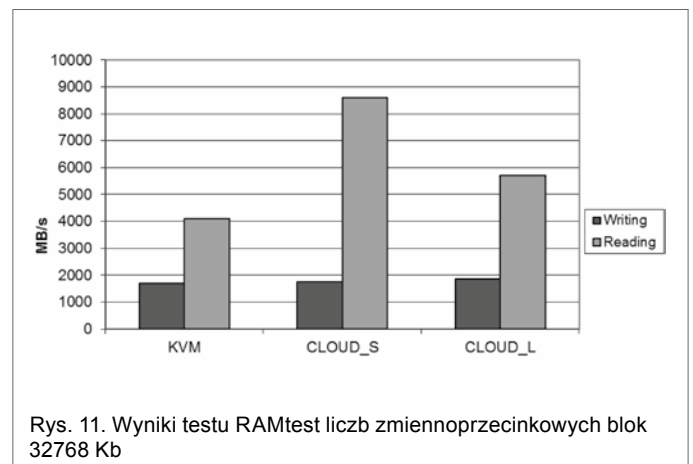
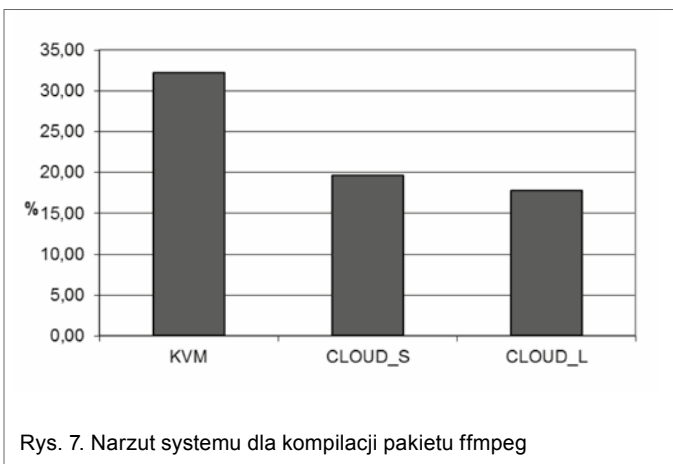
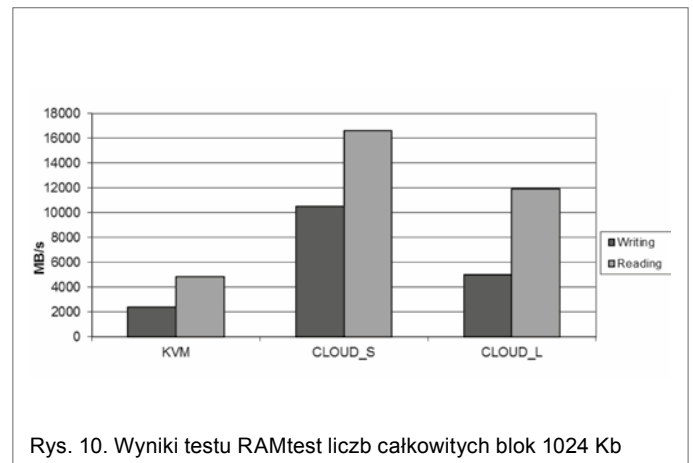
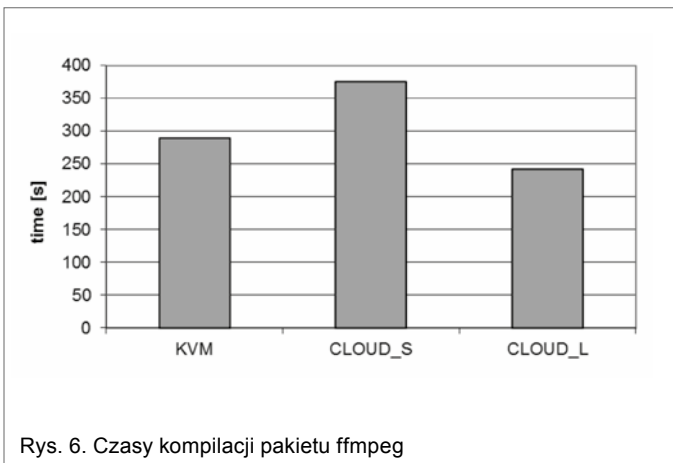
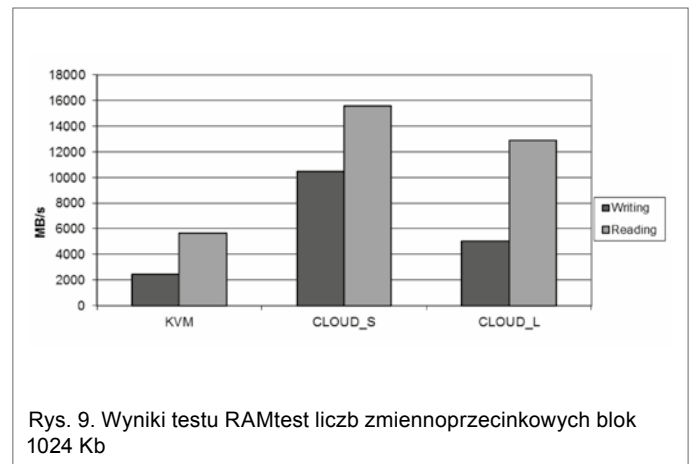
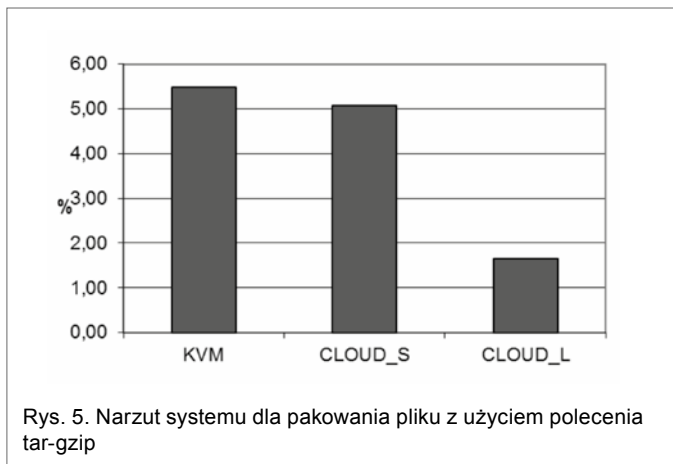
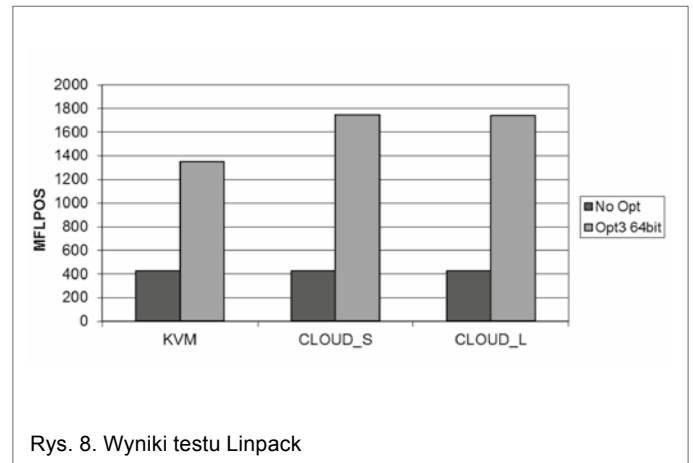
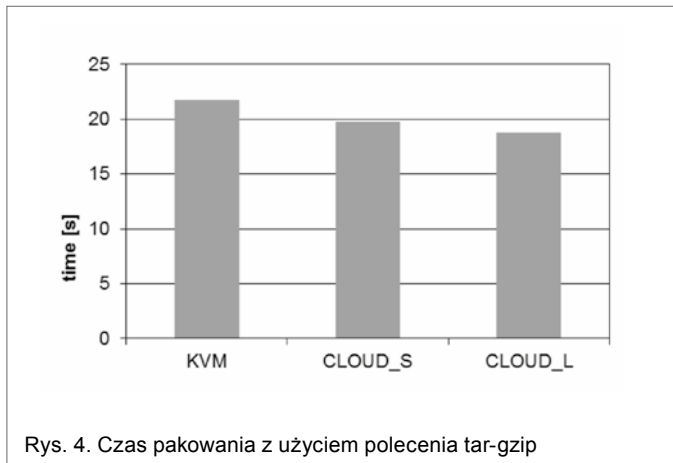




Tabela 1. Parametry sprzętowe i programowe

Sprzęt	KVM	CLOUD_S	CLOUD_L
RAM	1024 MB	128 MB	1024 MB
Core	2	1	2
HDD	10 GB	5 GB	10 GB
Platform	X86_64	X86_64	X86_64
Oprogramowanie	KVM	CLOUD_S	CLOUD_L
System	Ubuntu 9.04	Ubuntu 9.04	Ubuntu 9.04

sprzętu komputerowego. Z tego powodu w artykule wprowadzono pojęcie wirtualnego komputera w odniesieniu do rozwiązań przetwarzania w chmurze, jak i maszyn wirtualnych. Na wirtualny komputer składają się takie podstawowe elementy, jak: procesor CPU, pamięć operacyjna RAM oraz dysk twardej HDD i to one podlegały ocenie i porównaniu w systemach chmurowych i wirtualizacji sprzętowej.

Można wyróżnić tutaj dwa rodzaje testów: całościowe (ogólne) oraz ukierunkowane. Całościowe testy pozwalają ocenić ogólną wydajność systemu gościa uruchomionego w maszynie wirtualnej, część z testów posiada narzędzia pozwalające ocenić podstawowe elementy, takie jak: RAM, CPU oraz HDD.

Testy ukierunkowane na poszczególne elementy pozwalają na szczegółową ocenę podstawowych elementów, jakie składają się na wirtualny komputer, czyli pamięci operacyjnej, procesora oraz twardego dysku (rys. 2).

Na zbudowanej platformie testowej zdecydowano się przeprowadzić oba rodzaje testów.

Podstawą do budowy środowiska testowego był system Eucalyptus, a jego poszczególne elementy zostały zaimplementowane w oparciu o komputery klasy PC. Komputery te były wyposażone w dwurdzeniowy procesor firmy Intel ze wsparciem dla sprzętowej wirtualizacji, 2 GB pamięci RAM, pamięć dyskową o pojemności 250 GB oraz 2 interfejsy sieciowe. Podstawą realizacji wirtualizacji był pakiet KVM. Poszczególne elementy struktury testowego systemu Eucalyptus zostały przypisane do 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. 3.

Na tak zbudowanej platformie wykonano testy mające na celu zbadanie wydajności systemów wykorzystujących wirtualizację. Zestawienie testowanych systemów znalazło się w tabeli 1.

Ich wydajność testowana była na następujących urządzeniach:

- system Ubuntu 9.04 uruchomiony w maszynie wirtualnej KVM węzła środowiska Eucalyptus Cloud, VM (2 rdzenie), 1 GB RAM, 10 GB HDD SATA, system gospodarza Fedora 12 (x86\_64) – platforma testowa do badań oznaczonych jako „KVM”;

- 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\_S”;

- kontroler chmury Eucalyptus Cloud, Cloud (2 rdzenie), 1 GB RAM, 10 GB HDD SATA, system Ubuntu 9.04 (x86\_64) – platforma testowa do badań oznaczonych w dalszej części artykułu jako „CLOUD\_L”.

Testy wydajnościowe były uruchamiane odpowiednio:

- 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 zbadać bezpośrednio 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 testów

Wyniki pakowania pliku z użyciem polecenia tar przedstawia rys. 4. Czas pakowania określony został z użyciem polecenia time i na wykresie zaprezentowano rzeczywisty okres czasu, w jakim program był uruchomiony (ang. *real time*).

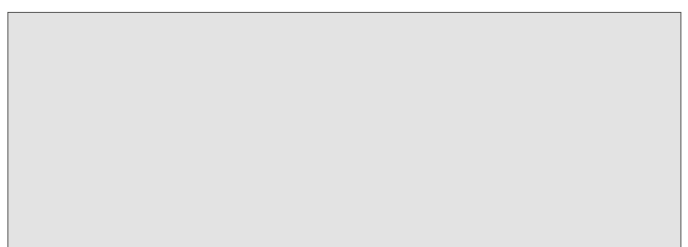
CLOUD\_L z tymi samymi parametrami sprzętowymi co KVM jest o 3 s szybszy od KVM. Odchylenie standardowe dla poszczególnych pomiarów wyniosło KVM = 0.41, CLOUD\_S = 0.11, CLOUD\_L = 0.23.

Wyraźnie widać, iż operacje plikowe w chmurze w przypadku CLOUD\_S zajmują o około 10% mniej czasu, natomiast w przypadku CLOUD\_L jest to już 16% szybciej niż w KVM. Potwierdzeniem tego faktu jest przedstawienie tak zwanego narzutu systemowego (rys. 5). 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 ( $\text{sys/user} \cdot 100\%$ ).

Kolejnym przeprowadzonym testem była kompilacja pakietu ffmpeg. Wyniki całkowitego czasu kompilacji przedstawiono na rys. 6.

Analizując wyniki czasów kompilacji, można zauważyć, że CLOUD\_L wykonał zadanie najszybciej w czasie 242 s, natomiast KVM był wolniejszy od CLOUD\_S o 132 s i działa jeszcze wolniej niż CLOUD\_L. Największe odchylenie standardowe uzyskano w pomiarze CLOUD\_S = 37.34, w pozostałych

reklama



odpowiednio: CLOUD\_L = 3.27, KVM = 1.84. Wśród systemów wirtualnych w procesie kompilacji architektura chmury ma przewagę w stosunku do systemu uruchomionego w maszynie wirtualnej, ale tylko dla porównywalnych zasobów sprzętowych. Na rys. 7 można zauważyć wyraźną różnicę pomiędzy systemami CLOUD i KVM. W przypadku chmury narzut systemu jest aż ok. 70% mniejszy dla CLOUD\_L w stosunku do maszyny wirtualnej.

Kolejnym przeprowadzonym testem był benchmark Linpack. Wyniki pomiarów zostały przedstawione na rys. 8.

Wyniki benchmarku bez optymalizacji są niemal identyczne (różnica jednego MFLOPS-a) dla systemów chmurowych, i wirtualnych. Wydajność w przypadku wykorzystania optymalizacji dla KVM jest gorsza o 25% niż dla systemów chmurowych. Odchylenie standardowe tych pomiarów zostało przedstawione w tabeli 2.

W celu zbadania wydajności pamięci operacyjnej RAM został przeprowadzony RAMtest. Przeprowadzone zostały testy na małych blokach alokowanych w pamięci RAM, co przedstawiono na rys. 9 i 10.

Największą wydajność posiadają małe obrazy uruchomione w środowisku Eucalyptus Cloud.

Ostatnim testem, który miał na celu sprawdzenie wydajności odczytu i zapisu danych na twardym dysku, był test z wykorzystaniem polecenia dd. Wyniki testów przedstawiono na rys. 13.

W tym teście rozwiązanie oparte o Eucalyptus Cloud jest wydajniejsze, pliki są zapisywane o 33% i odczytywane o 57% szybciej niż w przypadku KVM. Można to tłumaczyć tym, w jaki sposób jest obsługiwany system plików w chmurze w stosunku do obsługi systemu plików w maszynie wirtualnej KVM. W chmurze zadanie to powierzone jest oddzielnemu kontrolerowi, który bezpośrednio odwołuje się do systemu plików.

Natomiast w maszynie wirtualnej KVM to hypervisor odwołuje się pośrednio do twardego dysku systemu gospodarza, stąd też dodatkowe opóźnienie. Odchylenie standardowe dla odczytu wyniosło 2.27 dla KVM i 0.27 dla CLOUD\_L, natomiast dla zapisu wyniosło 0.12 dla KVM i 0.04 dla CLOUD\_L.

## Podsumowanie

W artykule dokonano porównania wydajności wirtualnego systemu oraz systemu uruchamianego w środowisku chmurowym. Test Gzip wykazał, że system chmurowy jest wydajniejszy od systemu wirtualnego odpowiednio dla obrazu CLOUD\_L o 16%, dla obrazu CLOUD\_S o 10%. W przypadku testów polegających na kompilacji pakietu ffmpeg najlepiej wypada obraz CLOUD\_L, później system wirtualny, a na końcu obraz CLOUD\_S. Jest to odpowiednio 55% i 19%. W przeprowadzonych testach narzut systemu w przypadku systemu uruchamianego w maszynie wirtualnej był zawsze największy i wynosił 32%.

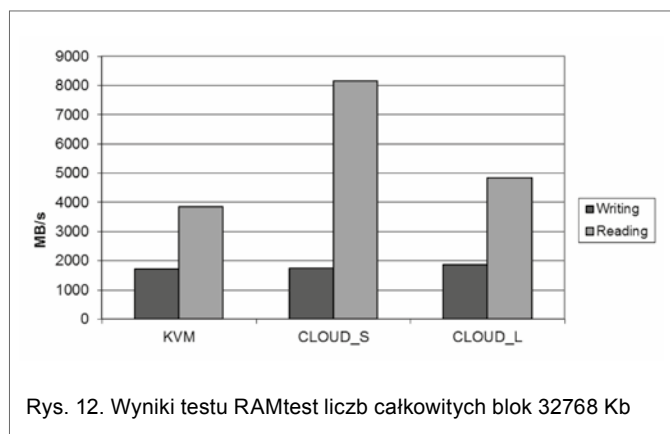
Benchmark Linpack przeprowadzony na platformie testowej wykazał przewagę środowiska chmurowego nad środowiskiem systemu wirtualnego o 25%, ale tylko z wykorzystaniem optymalizacji Opt3 64 bit. Bez optymalizacji różnica ta była na poziomie 1 MFLPOS.

RAMtest także wykazał przewagę systemów chmurowych nad wirtualnymi nawet do 50%.

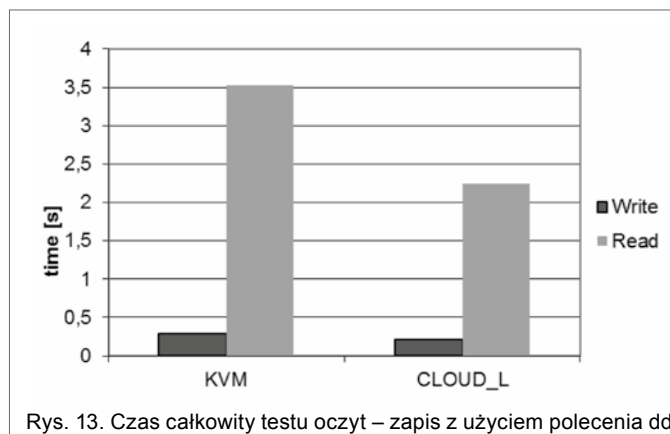
Test zapisu i odczytu za pomocą polecenia dd wykazał przewagę systemów chmurowych o 57% przy zapisie i o 33% przy odczycie danych.

Tabela 2. Odchylenie standardowe testu Linpack

Test	Parameter	KVM	CLOUD_S	CLOUD_L
Linpack	No Opt	1,15	1,78	1,76
	Opt3 64 bit	4,79	1,49	6,18



Rys. 12. Wyniki testu RAMtest liczb całkowitych blok 32768 Kb



Rys. 13. Czas całkowity testu odczyt - zapis z użyciem polecenia dd

## Bibliografia

- [1] FU Y., CHASE J., CHUN B., SCHWAB S., VAHDAT A.: *SHARP: an architecture for secure resource peering*. ACM SIGOPS Operating Systems Review, 37(5): 133–148, Dec. 2003.
- [2] LAI K., RASMUSSEN L., ADAR ZHANG E.L., HUBERMAN B.A.: *Tycoon: An implementation of a distributed, market-based resource allocation system*. Multiagent and Grid Systems, 1(3): 169–182, 2005.
- [3] AU YOUNG A., CHUN B., SNOEREN A., VAHDAT A.: *Resource allocation in federated distributed computing infrastructures*. Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT Infrastructure (OASIS 2004), Boston, USA, Oct. 2004, pp. 205–215.
- [4] IRWIN D.E., CHASE J.S., GRIT L.E., YUMEREFENDI A., BECKER R.D., YOCUM K.: *Sharing networked resources with brokered leases*. Proceedings of the 2006 USENIX Annual Technical Conference (USENIX 2006), Boston, USA, June 2006, pp. 199–212.
- [5] BUYYA R., YEO C.S., VENUGOPAL S.: *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. CCGRID '09 Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009, pp. 5–13.

- 
- [6] VERMA M., SINGH A., VANDANA, KAUR S.: *A Sight into Cloud Computing*. IJCST Vol. 2, Issue 2, June 2011.
- [7] Cisco Connected Technology World Report [Online], Available <http://newsroom.cisco.com>
- [8] DANTI G.: *Virtual machines performance comparison: vmware vs virtualbox vs kvm vs xen*. [Online], Available <http://www.ilsistemista.net>
- [9] SARDA K., SANGHRAJKA S., SION R.: *Cloud Performance Benchmark Series, Amazon EC2 CPU Speed Benchmarks*. Strony Brook University, 2010.
- [10] OSTERMANN S., IOSUP A., YIGITBASI N., PRODAN R., FAHRINGER T., EPEMA D.: *A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing*. Cloudcomp 2009, LNICST 34, pp. 115–131, 2010.
- [11] Parallels® Virtuozzo Containers: *Performance evaluation of a LAMP stack application in competing virtualized environments*. [Online], Available <http://www.parallels.com>
- [12] DANTI G.: *KVM vs Virtualbox 4.0 on RHEL 6* [Online], Available <http://www.ilsistemista.net>
- [13] 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, pp. 50–55.
- [14] FOSTER I., ZHAO Y., RAICU I., LU S.: *Cloud Computing and Grid Computing 360-Degree Compared*. Grid Computing Environments Workshop, 2008, pp. 1–10.
- [15] 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, pp. 124–131.
- [16] JOHNSON D., MURARI K., RAJU M., SUSEENDRAN R.B., GIRIKUMAR Y.: *Eucalyptus Beginner's Guide – UEC Editio*. CSS Corp. 2010, <http://www.csscorp.com/eucauecbook>, June 2010.
- [17] GOLDBERG R.P.: *Architectural Principles for Virtual Computer Systems*. Harvard University, Technical report 1973, pp. 22–26.
- [18] KARGER P.A., SAFFORD, D.R.: *I/O for Virtual Machine Monitors: Security and Performance Issues*. Security & Privacy, IEEE, Sept.-Oct. 2008, vol. 6, pp. 16–23.
- [19] HIRT T.: *KVM – The kernel-based virtual machine*. University of Applied Sciences Wiesbaden, 2010, Available <http://www.hs-rm.de>
- [20] Xen Intro, [Online], Available <http://wiki.xensource.com/xenwiki/XenIntro>
- [21] WARD J.S.: *A performace comparison of clouds, Amazon EC2 and Ubuntu Enterprise Cloud*. SICSA DemoFEST 2009 [Online], Available <http://www.cs.st-andrews.ac.uk/stacc>
- [22] LARABEL M.: *Amazon EC2 Cloud Benchmarks. Published December 2010* [Online], Available <http://www.phoronix.com>
- [23] Cloud harmony: *What is an ECU? CPU Benchmarking in the Cloud*. Published May 2010 [Online], Available <http://blog.cloudharmony.com>