STUDIA INFORMATICA 2018
Volume 23 Number 2 (47)

Zygmunt BOK
Szpital Specjalistyczny im Prof. E. Michałowskiego, MEDHOLDING S.A.

METODY KREACJI MASZYN WIRTUALNYCH W ŚRODOWISKU VMWARE ESXI

Streszczenie. W niniejszym artykule opisano metody kreacji maszyn wirtualnych na nowych serwerach lub klastrach *VMware ESXi*. Opisano kilka metod, w tym: metodę natywną dostępną w oprogramowaniu *VMware vSphere*, metodę opartą na przenoszeniu maszyn wirtualnych z zastanych serwerów *VMware ESXi* i metodę wykorzystującą oprogramowanie *VMware vCenter Converter Standalone*. W artykule zaproponowano metodę tworzenia maszyn wirtualnych wykorzystującą metodę natywną, połączoną z funkcjonalnością oprogramowania typu "*Backup and Disaster Recovery*" o nazwie „*Acronis Backup Advanced for PC"* firmy *ACRONIS.*

METHODS FOR CREATING VIRTUAL MACHINES IN THE ENVIRONMENT VMWARE ESXI

Summary. In this article the creation of virtual machines on the new WMware ESXi servers or clusters has been described. Several methods has been described, including: native method available in VMware vSphere, method based on the moving virtual machines from existing servers VMware ESXi and the method of using VMware vCenter Converter Standalone. In this article a method to create virtual machines has been proposed using the native method combined with the functionality of the "Backup and Disaster Recovery" type software named "Acronis Backup Advanced for PC" from ACRONIS company.

1. Wprowadzenie

1.1. Technologia wirtualizacji

Wirtualizacja [1,2], rozpoczęta w latach 60-tych na komputerach typu *mainframe*, swój początek miała w 1960r., kiedy to *IBM* wprowadził logiczną partycję w swojej maszynie *VM/370* [3]. Wirtualizacja jest technologią stosowaną do współdzielenia zasobów i możliwości komputerów fizycznych, poprzez dzielenie zasobów pomiędzy systemy operacyjne i różne aplikacje. Koncepcja maszyn wirtualnych *VM's* (*Virtual Machines*) [4,5] rozpoczęła się w 1964 wraz z projektem *IBM* o nazwie *System CP/CMS* (*Control Program*/*Cambridge Monitor System*), czyli systemu operacyjnego typu *time-sharing*, dzielącego zasoby obliczeniowe wśród wielu użytkowników, w tym samym czasie, za pomocą wielozadaniowości. *CP/CMS* ewoluował w następnych latach do maszyny wirtualnej *Virtual Machine Facility/370* [6]. W tamtych latach, komponent *CP* był systemem operacyjnym maszyny obliczeniowej, zdolnym wykorzystywać wiele kopii tej maszyny w tym samym czasie. Każda kopia (*VM*) była kontrolowana przez jej własny *OS* (*Operating System – komponent CMS*). W 1973r. Ronald J. Srodawa oraz Lee A. Bates zademonstrowali sposób tworzenia maszyn wirtualnych [7] na systemie *IBM OS/360*.

IBM-owski projekt i wysiłki środowiska naukowego w latach 70-tych określiły podstawy wirtualizacji niskopoziomowej, w szczególności prace [8,9], które dały jasne wyjaśnienie koncepcji i korzyści odnoszących się do wirtualizacji. W następnych latach zainteresowanie technologią wirtualizacji spadało, aż do roku 1999, kiedy to firma *VMware Inc.* zaprezentowała swój produkt *VMware Virtual Platform* [10,11] dla architektury x86-32. Inny produkt tej firmy, znany pod nazwą *VMware Player*, umożliwia tworzenie i uruchamianie maszyn wirtualnych na stacji roboczej, wykorzystując do tego celu technikę typu *paravirtualization* [12], wspierając tym samym szeroki zakres gościnnych systemów operacyjnych (*guest OS)*. Kluczową korzyścią wirtualizacji jest zdolność uruchamiania wielu gościnnych systemów operacyjnych, w postaci maszyn wirtualnych, na pojedynczej fizycznej maszynie zwanej hostem i współdzielenia jej zasobów sprzętowych, znana też jako wirtualizacja typu *partitioning* [13]. Wirtualizacja może być zastosowana do różnych warstw systemowych, włączając w nie wirtualizację typu: (*i*) *hardware-level virtualization operating system*, (*ii*) *operating system-level virtualization* oraz (*iii*) *high-level language virtual machines*.

Równolegle do produktu komercyjnego jakim był *VMware Virtual Platform,* pojawił się projekt typu *Open Source*, znany pod nazwą *Xen* [14,15,16]. Zapoczątkowany został w *University of Cambridge Computer Laboratory* w późnych latach 90-tych, którego celem

było stworzenie efektywnej platformy dla przetwarzania rozproszonego. *Xen*, uważany jest za standard przemysłowy, zapewniający abstrakcyjny poziom pomiędzy *hardwar*'em, a gościnnymi systemami operacyjnymi. Jest monitorem maszyn wirtualnych *VMM* (*Virtual Machine Monitor*), pozwalający na fizycznej maszynie *boot*'ować różne systemy operacyjne w architekturze x86-32. *Xen* używa techniki typu *paravirtualization* pomiędzy *hipervisor*'em a gościnnymi systemami operacyjnymi, w celu zapewnienia ujednoliconego i bezpiecznego dostępu do zasobów sprzętowych. W 2004r. założono firmę *XenSource*, której zadaniem było wprowadzenie *hypervisor*'a *Xen* na rynek. W 2005r. firmy *RedHat*, *Novell* (przejęty przez *Micro Focus*) oraz *Sun* (przejęty przez *Oracle*) dodały *hypervisor*'a *Xen* do oferowanych przez nich rozwiązań, wprowadzając go tym samym do głównego nurtu wirtualizacji. Dwa lata później firma *Citrix Systems* zakupiła firmę *XenSource*, w celu zapewnienia kompletności oferowanych rozwiązań.

Wśród innych projektów typu *Open Source* [16,17] wskazać można na:
(1) ***QEM***, który jest szybkim emulatorem typu *multihost* i *multitarget* oraz wirtualizatorem, napisanym przez *Fabrice Bellard'a*. Dostępny jest w postaci wolnego oprogramowania; umożliwia jednoczesne uruchomienie kilku gościnnych niemodyfikowanych systemów operacyjnych na jednej maszynie. *QEM* może pracować na maszynach w architekturze *x86*, *x86-64*, *PowerPC* i emulować systemy w architekturze *x86*, *x86-64*, *ARM*, *SPARC*, *PowerPC*, *MIPS*. Dla większości z nich może być uruchamiany w dwóch trybach: (*i*) *full-system emulation* – który emuluje kompletny system komputerowy, począwszy od podstawowego systemu wejścia-wyjścia *BIOS*, do kart video-dźwiękowych, (*ii*) *user-mode emulation* – pozwalający na uruchomienie linuksowego kodu binarnego, skompilowanego do innej architektury,
(2) ***VirtualBox***, który jest swobodnie dostępnym produktem firmy *Oracle*, o dużych możliwościach, bogatej funkcjonalności oraz wysokiej wydajności. Przeznaczony jest na platformy o architekturze *x86* i *AMD64/Intel64*, dla klientów przemysłowych oraz domowych zastosowań. Obecnie *VirtualBox* uruchamiany jest na hostach typu: *Windows*, *Linux*, *Macintosh*, *Solaris*. Wspiera dużą liczbę gościnnych systemów operacyjnych: *Windows* (*NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10*), *DOS/Windows 3.x*, *Linux* (2.4, 2.6, 3.x, 4.x), *Solaris* and *OpenSolaris*, *OS/2*, *OpenBSD*.

Obecnie technologia wirtualizacji obejmuje bardzo szerokie spektrum technik wirtualizacyjnych [1,3,15,16,18,19,20,21,22,23] obejmujących:

- **wirtualizację sprzętową** - *hardware virtualization* - do której zalicza się następujące typy wirtualizacji, tj.: (1) *full virtualization* - prawie kompletna symulacja aktualnego *hardware*'u, pozwalająca uruchomić *software*, w postaci gościnnego systemu operacyjnego, bez żadnych zmian w środowisku wirtualnym. *Hypervisor* dokonuje translacji i emulacji każdego elementu z fizycznej warstwy sprzętowej, (2) *hardware-*

assisted virtualization – która jest sposobem poprawy ogólnej wydajności wirtualizacji, (3) *partial virtualization* – jest typem wirtualizacji, polegającej na symulacji części docelowego środowiska; niektóre programy gościnne mogą wymagać modyfikacji, w celu umożliwienia ich uruchomienia w środowisku wirtualnym, (4) *paravirtualization* – w tym typie wirtualizacji, środowisko *hardware*'owe nie jest symulowane, jednak programy gościnne są uruchamiane w ich własnych domenach, tak, jak gdyby pracowały w odseparowanym systemie, (5) *operating system-level virtualization* – typ wirtualizacji wspierający uruchamianie (*execution*) całych systemów operacyjnych, (6) *hypervisor* lub *VMM Virtual Machine Monitor* [9] - jest platformą wirtualnego menadżera, która działa powyżej warstwy fizycznej serwera, ale zarazem poniżej systemu operacyjnego; uruchamia i zarządza jedną lub więcej gościnnych maszyn wirtualnych wraz z ich własnym systemem operacyjnym. Jest warstwą *software*'ową, która wirtualizuje wszystkie zasoby maszyny fizycznej, a tym samym definiuje i wspiera uruchomienie (*execution*) wielu maszyn, (7) *hardware virtualization disaster recovery* - jest środowiskiem wirtualizacji mogącym zapewnić wysoki poziom dostępności, w sytuacjach zakłócających normalne operacje biznesowe, (8) emulacja (symulacja) wirtualnego środowiska systemu operacyjnego hosta w maszynie wirtualnej,

- **wirtualizację desktopów** - *desktop virtualization* - jest koncepcją separacji logicznych desktopów od fizycznych maszyn i stanowi bardziej zaawansowaną formą wirtualizacji sprzętowej (*hardware virtualization*); zamiast interakcji z hostem poprzez klawiaturę, mysz lub monitor, użytkownik współdziała z hostem używając innego desktopu lub urządzenia mobilnego, za pomocą połączenia sieciowego typu *LAN*, *Wireless LAN*, Internet,
- **wirtualizację zagnieżdżoną** - *nested virtualization* – która odnosi się do możliwości uruchomienia maszyny wirtualnej w innej wirtualnej maszynie; innymi słowy, zagnieżdżona wirtualizacja, która umożliwia uruchomienie jednego lub więcej *hypervisor*'ów w środowisku innego *hypervisor*'a,
- **inne typy wirtualizacji**: (1) wirtualizacja aplikacji - *AV - Application Virtualization* oraz przestrzeni roboczych - *WV - Workspace Virtualization*, (2) wirtualizacja usług - *SV - Service Virtualization*, (3) wirtualizacja pamięci - *MV - Memory Virtualization*, (4*) wirtualizacja magazynów pamięci - SV - Storage Virtualization,* (5) wirtualizacja danych - *DV - Data Virtualization,* (6) wirtualizacja sieci - *NV - Network Virtualization*.

W zakresie wsparcia wirtualizacji umożliwiające uruchamiania (*execution*) całych systemów operacyjnych istnieje kilka technik wirtualizacji [12,24,25]. Klasyfikuje się je z punktu widzenia gościnnego systemu operacyjnego. Wśród technik, które uruchamiają:

- **zmodyfikowane systemy operacyjne** - wyróżnić można: (a) technika typu: *operating system-level virtualization* oraz (b) technika typu: *para-virtualization*,
- **niezmodyfikowane systemy operacyjne** – do nich zaliczyć można: (c) technika typu *binary translation* oraz (d) technika typu *hardware assisted*.

a) Technika wirtualizacji typu *operating system-level virtualization* [25,26,27,28,29] - jest metodą wirtualizacji serwera, w której jądro systemu operacyjnego pozwala na utworzenie i zarządzanie wiele izolowanych instancji (wystąpień) przestrzeni użytkownika (*VSM - Server Virtualization Methods*) zamiast jednej. Technika narodziła się w systemach operacyjnych typu *Linux* i jest zaawansowaną implementacją standardowego mechanizmu *chroot*. Za pomocą tej techniki wirtualizuje się host fizyczny na poziomie systemu operacyjnego. Znana jest także pod nazwą techniką konteneryzacji, odnosząca się do funkcjonalności systemu operacyjnego, dzięki której jądro systemu operacyjnego pozwala na istnienie wielu izolowanych instancji typu *user-space*, zwanych czasami kontenerami albo *software*'owymi kontenerami, partycjami, silnikami wirtualizacji *VE* (*virtualization engines*) lub tzw. *chroot jail*, będącym sposobem pozwalającym na izolowania procesu i jego podprocesów typu *children* od reszty systemu operacyjnego. Za pomocą tego sposobu tworzy się nowe drzewo katalogowe, do którego kopiuje się wszystkie pliki systemowe, wymagane do pracy procesu, a następnie używa się polecenia systemowego *chroot()*, w celu: (1) zmiany *root directory*, mający wskazywać na podstawę nowego drzewa katalogowego oraz (2) inicjuje się proces działający w środowisku *chroot*.

b) Technika typu: *paravirtualization* – w tej technice tworzy się specjalny zbiór instrukcji zwanych *hypercalls*, które zamieniają instrukcje ze zbioru realnych instrukcji maszynowych procesora [14,25,30] na instrukcje *hypercalls* [31]. W tej technice, środowisko *hardware*'owe nie jest symulowane, jednak programy gościnne są uruchamiane w ich własnych domenach, tak, jak gdyby pracowały w odseparowanym systemie. Technika typu *paravirtualization* różni się od techniki typu *full virtualization* tym, że niemodyfikowany system operacyjny nie wie, że jest zwirtualizowany, a wywołania (instrukcje) systemu operacyjnego są poddane translacji binarnej w czasie ich wykonywania. Instrukcje te są obsługiwane w czasie kompilacji, kiedy niezwirtualizowane instrukcje systemu operacyjnego zamieniane są na instrukcje *hypercalls*.

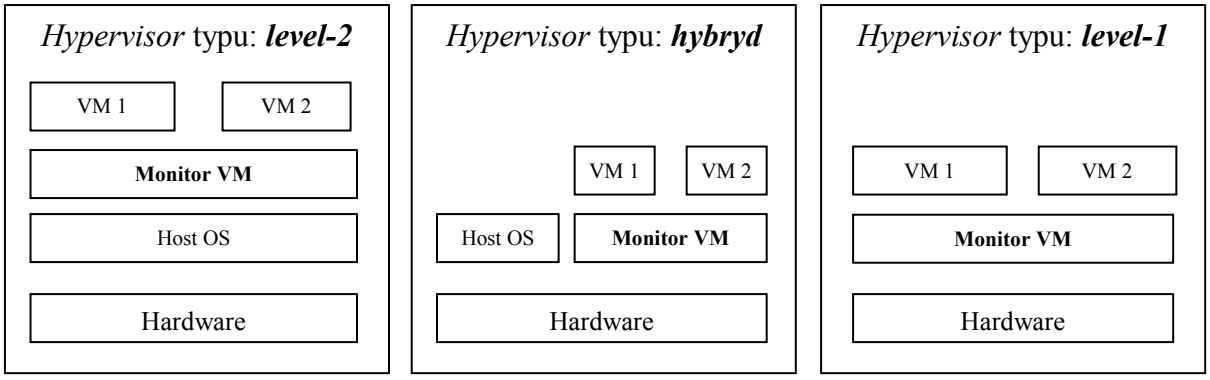
c) Technika typu *BT - Binary Translation* – jest techniką używaną w celu emulacji architektury procesora ponad architekturą innego procesora [25,32]. Umożliwia wykonanie (*execution*) niezmodyfikowanego gościnnego systemu operacyjnego, poprzez emulację jednego zbioru instrukcji za pomocą innego zbioru, poprzez translację kodu.

d) Technika typu *hardware assisted* – została po raz pierwszy zaimplementowana, kiedy w 1972r. wirtualna maszyna *Virtual Machine Facility/370* stała się dostępna dla systemu

IBMSystem/370 [6]. Później, w roku 2006, została wprowadzona przez firmę *AMD* oraz *INTEL* dla ich linii produktowej. Jest sposobem, który poprawia ogólną wydajność wirtualizacji; wzmacnia *CPU* do wspierania wirtualizacji, bez potrzeby użycia techniki typu *binary translation* lub *paravirtualization* [25].

Wspomniana wcześniej jedna z technik, z szerokiego spektrum technik wirtualizacyjnych, tj. wirtualizacja serwerów *SV - Server Virtualization* [32,34,35,36,37,38,39,40,41], reprezentuje obecnie podstawową technologię adresowaną do ciągle rozrastających się i bardziej złożonych systemów *IT*, czyniąc coraz trudniejszy wybór techniki wirtualizacyjnej oraz optymalną rozbudowę struktury *IT*, w gwałtownie zmieniającym się środowisku biznesowym. Pewną pomoc przy wyborze techniki wirtualizacyjnej dla konkretnych zadań mogą przynieść prace [19,42,43]. Według raportu *Gartne*r'a z 2011r., wirtualizacja serwerów wzrosła i ciągle rośnie. Już w roku 2012r., według *Forrester'*a, 52% środowisk *IT* pracowało w środowisku wirtualnych serwerów. Na podstawie jego badań, ok. 85% firm *IT* używa, bądź planuje używać środowiska wirtualnych serwerów w najbliższej przyszłości (*Forrester* 2012r.). Wirtualizacja serwerów w 2016r., na podstawie raportu [44], zbliży się do wartości ok. 80%., natomiast według raportu *Gartner*'a z 2011r. osiągnie swój szczyt w roku 2018r.

Techniki wirtualizacji serwerów mogą być skategoryzowane w czterech grupach [11,45,47], tj. wirtualizacja typu: (1) *Full*, (2) *Para*, (3) *OS*, (4) *hardware*. Wszystkie typy wirtualizacji, z wyjątkiem wirtualizacji *OS*, używają jednej z trzech poniższych typów technologii *hypervisor'*a [1,2,16,46], pokazanych na Rys. 1, tj.: (*i*) *hypervisor* typu *level-1*, (*ii*) *hypervisor* typu level-2 oraz (*iii*) *hypervisor* typu *hybryd*.



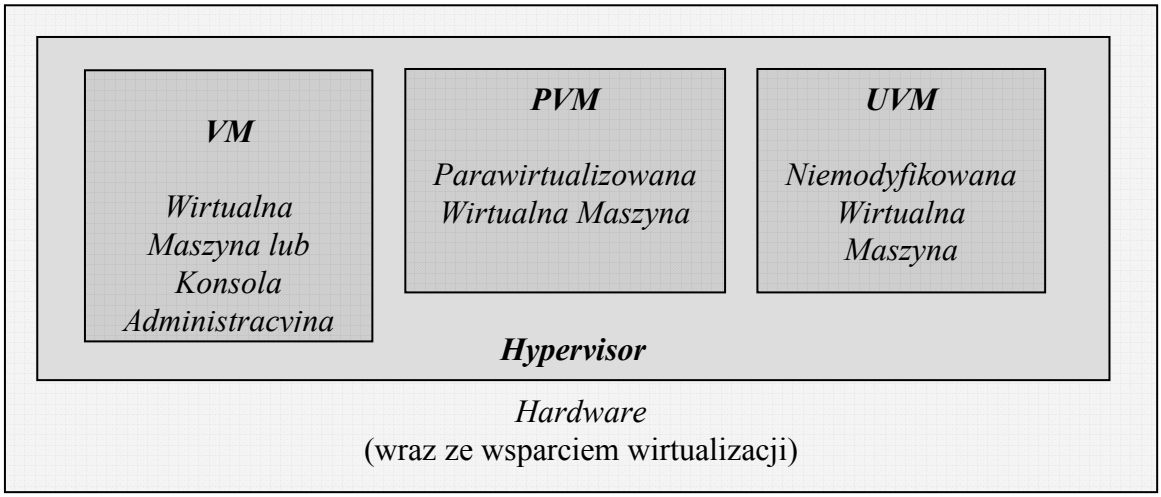
Rys. 1. Typy technologii *hypervisor'*a
Fig. 1. Hypervisor's technology types

Hypervisor typu *level-1* oraz *Hypervisor* typu *hybryd* używany jest do wirtualizacji sprzętu; w typie *level-1*, funkcja *hypervisor*'a (*VMM - Virtual Machine Monitor*) znajduje się

ponad sprzętem i działa bezpośrednio na tym sprzęcie. Ponieważ w tym typie wirtualizacji, nie ma żadnej warstwy pomiędzy sprzętem a *VMM*, dlatego nazywany jest też typem wirtualizacji zwaną *bare-metal*. Uważany jest za bardziej bezpieczny niż *level-2*; uszkodzenie jednej maszyny wirtualnej nie wpływa na inne maszyny wirtualne, jak również na *hypervisor*'a [16]. Poziom *hybryd* emuluje cały sprzęt. Typ *level-2*, posiada warstwę *hypervisor'*a ulokowaną ponad warstwą systemu operacyjnego *host'a*. Korzyść płynąca z używania tego typu wirtualizacji jest taka, że może wspierać różny sprzęt (*hardware)*, ponieważ jego oprogramowanie jest dziedziczone przez *hypervisor'*a wprost z systemu operacyjnego *host*'a.

1). Wirtualizacja typu *Full* - używa *hypervisor*'a typu *level-1*, gdzie niemodyfikowany system operacyjny (*OS*) hosta leży na jego szczycie. W tym podejściu, gościnny *OS* (*VM1, VM2*) nie potrzebuje być zmieniany, a architektura *hardware*'owa może się różnić od architektury fizycznej.

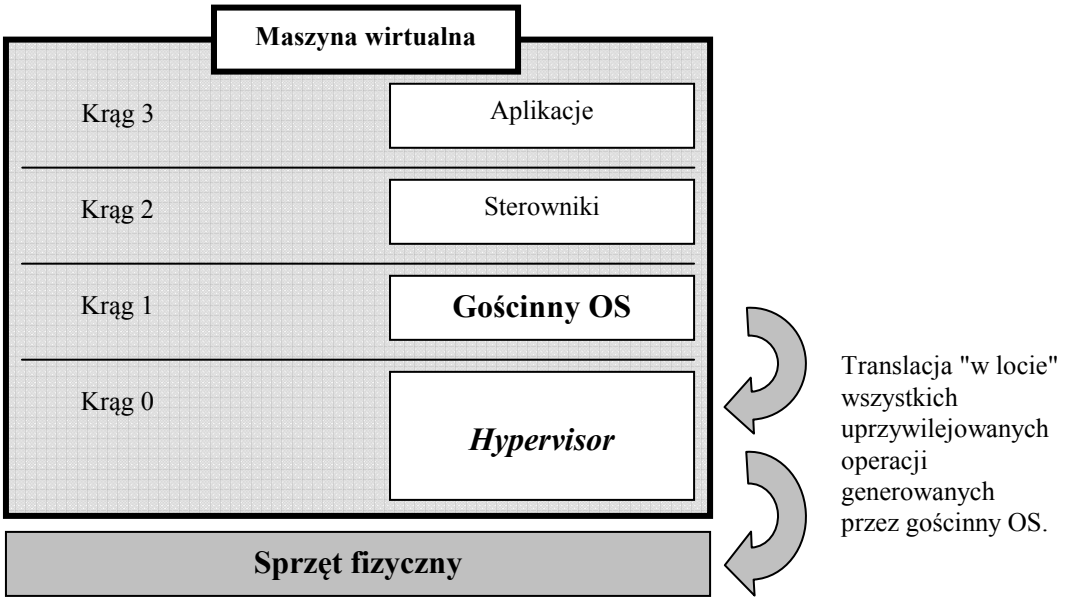
2). Wirtualizacja typu *Para* - którą pokazano schematycznie na Rys. 2 - jest podejściem stosowanym do wirtualizacji serwerów; wprowadzone przez *Xen* w 2005r., które zostało zaadaptowane przez innych wytwórców – w tym przez *VMware Inc*. W tym typie wirtualizacji *hypervisor* uruchamiany jest bezpośrednie ze sprzętu.



Rys. 2. Technika parawirtualizacji
Fig. 2. Paravirtualization technique

W technice parawirtualizacji, znanej także jako *OS-assisted virtualization* i schematycznie pokazanej na Rys. 3 [47], gościnny system operacyjny, musi być wstępnie jawnie przetworzony w czasie *build-time* lub *load-time*, aby mógł pracować na warstwie *hypervisor*'a. To wymaga zastąpienia bezpośredniego wywołania uprzywilejowanych

instrukcji, które normalnie byłyby uruchamiane w kręgu 0 procesora, z jawnym wywołaniem interfejsów *API* funkcji *hypervisor*'a, znanych jako *hypercalls*.



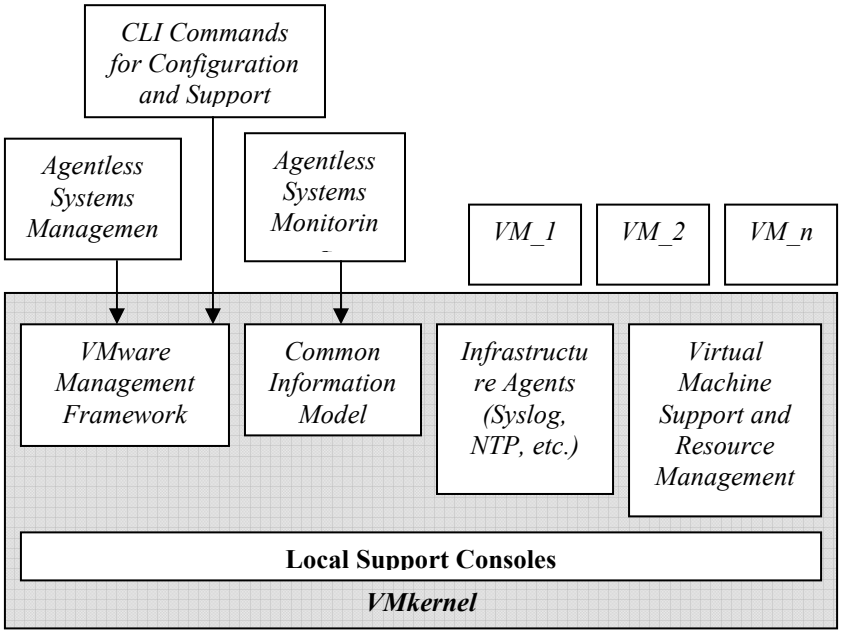
Rys. 3. Parawirtualizacja
Fig. 3. Paravirtualization

W zakresie strategii wirtualizacji, obecnie istnieje wiele dostępnych rozwiązań proponowanych przez różnych producentów systemów operacyjnych, promujących swoje specyficzne rozwiązania. Dotyczą one wirtualizacji odnoszącej się do tych systemów, jak chociażby rozwiązania: *Solaris Zones*, *BSD jails in FreeBSD*, *HP-UX Containers* oraz *PowerVM* na systemie *IBM AIX*. Innym przykładem technologii wirtualizacji serwerów może być technologia zaproponowana przez f-mę *Fujitsu* [48], która umożliwia elastyczne konstruowanie wirtualnych serwerów, prawie bez żadnych ograniczeń sprzętowych, co w konsekwencji redukuje ogólny koszt posiadania *TCO* (*Total Cost of Ownership*) oraz czyni łatwe w użyciu wirtualnych serwerów w zmieniającym się środowisku biznesowym. Firma *Fujitsu* dostarcza technologię wirtualizacji serwerów w formie produktu *software*'owego, zwanego *Virtual Machine Function*, który adoptuje technologię *Xen* typu *open-source*, dając jednocześnie swój wkład do rozwoju tej technologii, w szczególności wzmocnienie funkcjonalności dla systemów typu *mission-critical*, bazując na technologiach dla wirtualizacji komputerów typu *mainframe*.

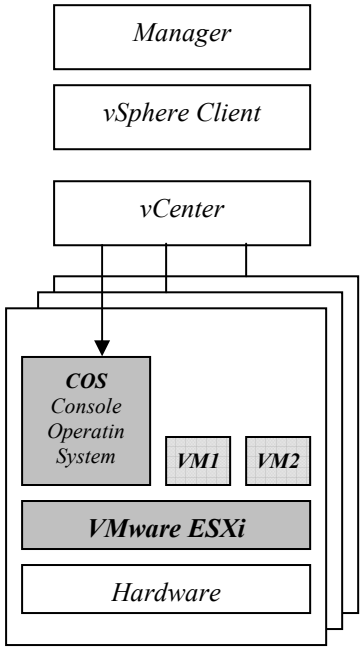
W niniejszej pracy, w dużym skrócie, opisano technologie wirtualizacji serwerów w architekturze *x86*, w szczególności w trzech technologiach, reprezentujących blisko 100% rynku. Są to rozwiązania zaproponowane przez: (*i*) *VMware ESXi Server,* (*ii*) *Citrix Xen*, (*iii*) *Microsoft Hyper-V*. Spośród dużej liczby rozwiązań wirtualizacji serwerów w innych technologiach opisano wirtualizację serwerów w technologii *Oracle*'a.

1.1.1. Wirtulizacja serwerów w technologii VMware ESXi Server

Technologia *VMware ESXi Server*, która dziesięć lat po pierwszym pojawieniu się technologii *ESX*, według raportu *Gartner*'a posiada blisko 85% rynku [16]. Służy do tworzenia wirtualnej infrastruktury informatycznej wraz z wirtualnymi serwerami w postaci maszyn wirtualnych pracujących środowisku *VMware ESXi* [13,19], którego uproszczoną architekturę, według [16], pokazano na Rys. 4, natomiast schemat blokowy, według [49], przedstawiono na Rys. 5.



Rys.4. Architektura *VMware ESXi*
Fig. 4. The VMware ESXi architekture

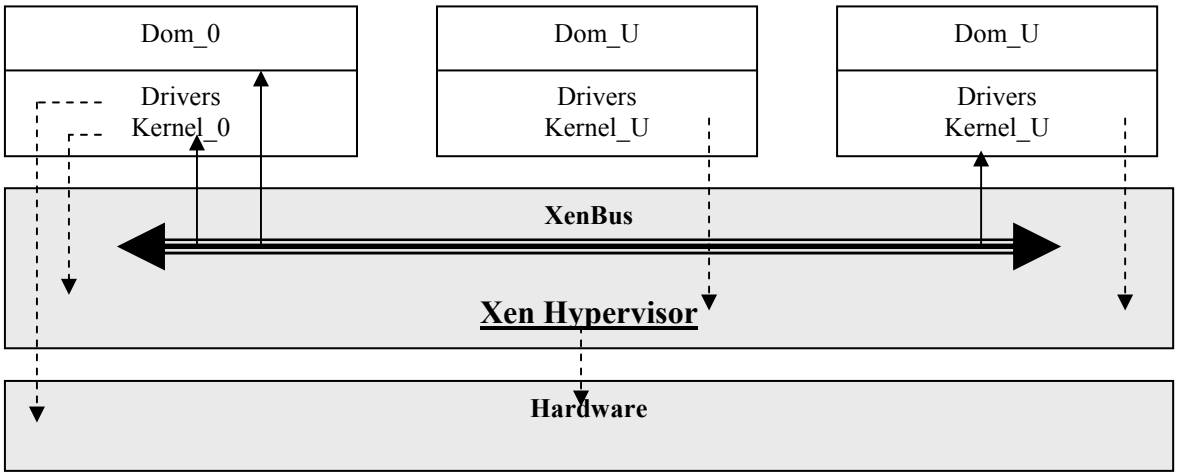


Rys.5. Schemat blokowy *ESXi*
Fig. 5. The ESXi block schema

Jądro *vmkernel* zawiera wszystkie procesy niezbędne dla: (i) wsparcia maszyn wirtualnych, (*ii*) zarządzania *hardware*'em i dostępnymi zasobami, (*iii*) wsparcia usług infrastrukturalnych, np. czasu, logowania, (*iv*) integracji z innymi natywnymi *VMware*'owskimi narzędziami zarządzającymi lub firm trzecich, np. sterownikami lub monitorami sprzętowymi. Ten model wirtualizacji stanowi jedną z największych różnic pomiędzy *VMware Inc*., a wieloma rozwiązaniami innych firm. Jak wspomniano na wstępie, *VMware Inc.,* jako pierwsza na rynku w zakresie wirtualizacji maszyn w architekturze x86-32, zyskała możliwość rozwoju swojej technologii i zaproponowania użytkownikom dojrzałe funkcjonalności i możliwości, które inni producenci posiadali w postaci elementarnej lub w ogóle nie mieli. Przykładem może być wprowadzona w 2004r. funkcjonalność znana pod nazwą *VMotion*, *HA* (*High A Availability*) oraz *FT* (*fault tolerance*) [50,51,52]. Są one przykładem szerokiego zakresu funkcjonalności oferowanych przez *VMware ESXi*.

1.1.2. Wirtulizacja serwerów w technologii Citrix Xen

Jak już wcześniej wspomniano, *Citrix Systems* zakupił *XenSource* w celu zapewnienia kompletności oferowanych rozwiązań, a w szczególności *hypervisor*'a, którego architekturę przedstawiono na Rys. 6 [16].



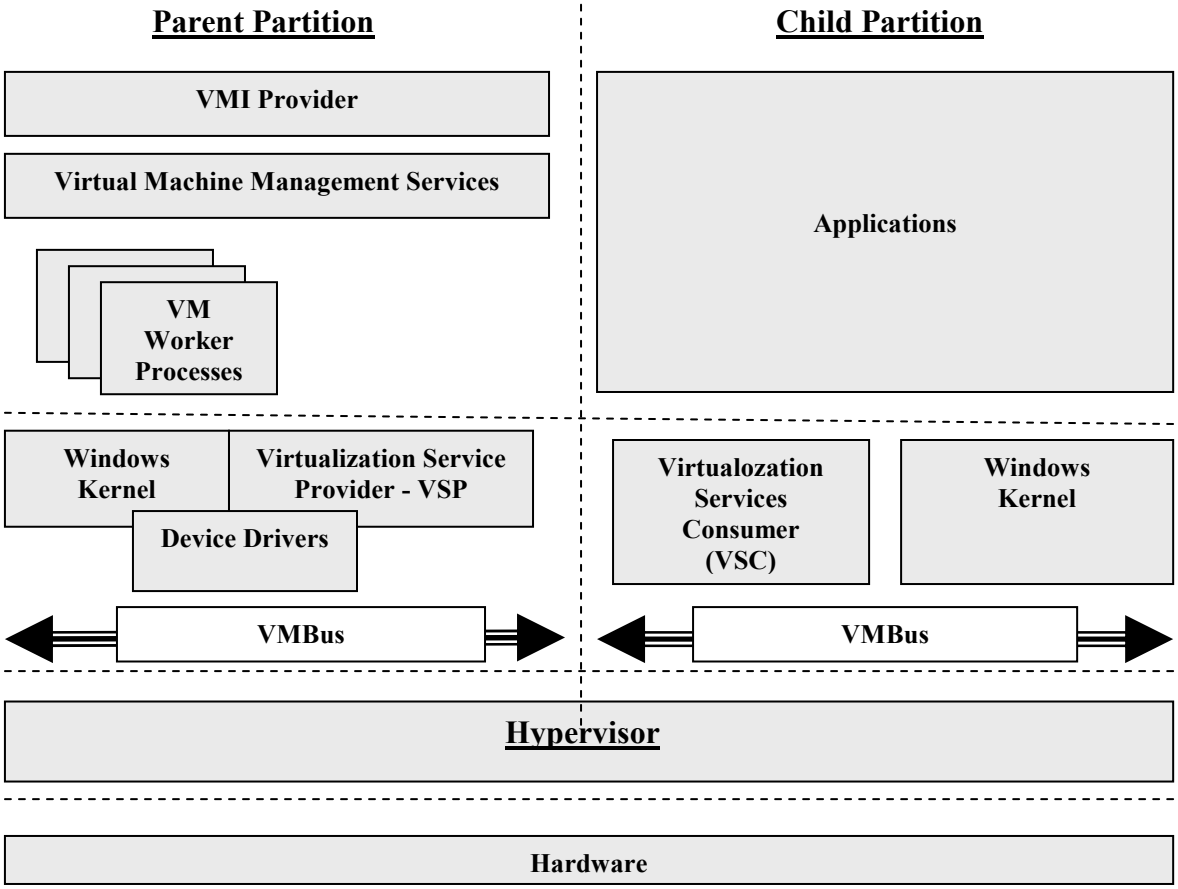
Rys. 6. Architektura *Citrix Xen*
Fig. 6. The *Citrix Xen* architekture

W tej architekturze, *hypervisor* jest typu *bare-metal*, osadzony bezpośrednio na warstwie sprzętowej. Jego implementacja pokazuje pewne różnice w stosunku do architektury *VMware ESX*, pokazanej na wcześniejszym Rys. 5. Architektura *hypervisor*'a *Xen* posiada specjalną domenę *Dom_0*, wraz z gościnnym *systemem operacyjnym*, która jest *boot*'owana wtedy, kiedy *boot*'owany jest *hypervisor*. Ta domena posiada inne uprawnienia zarządzające niż pozostałe domeny (*Dom_U*) oraz bezpośredni dostęp do warstwy sprzętowej. Pozwala to jej na obsługę wszystkich instrukcji *I/O* dla pozostałych domen (gościnnych systemów operacyjnych). Kiedy dodatkowe domeny formułują żądania do zasobów w niżej leżącej warstwie zasobów sprzętowych, wówczas kierowane są najpierw do *hypervisor*'a, który następnie przekazuje je do domeny *Dom_0* i za jej pośrednictwem trafiają do warstwy sprzętowej.

1.1.3. Wirtulizacja serwerów w technologii Microsoft Hyper-V

Microsoft rozpoczął swój udział w wirtualizacji serwerów w 2005r., wraz z wprowadzeniem na rynek produktu *Virtual Server*, po tym, jak wcześniej nabył rozwiązanie od firmy *Connectix*. Produkt *Virtual Server* jest *hypervisor*'em typu *level-2*, natomiast *Microsoft Hyper-V*, udostępniony w 2008r., jako instalowalna część (rola) w *Windows Server 2008 Operating System* [53], jest *hypervisor*'em typu *level-1*. Podobnie jak w modelu *Xen*, *Microsoft Hyper-V* [54, 55] wymaga specjalnej partycji (*parent*), która posiada bezpośredni dostęp do warstwy zasobów sprzętowych. Pomimo tego, że *Microsoft* relatywnie późno

zaoferował swój produkt, zyskał ok. 10% udziału w rynku wirtualizacji serwerów. Poniżej, na Rys. 7, przedstawiono architekturę tego rozwiązania [16].



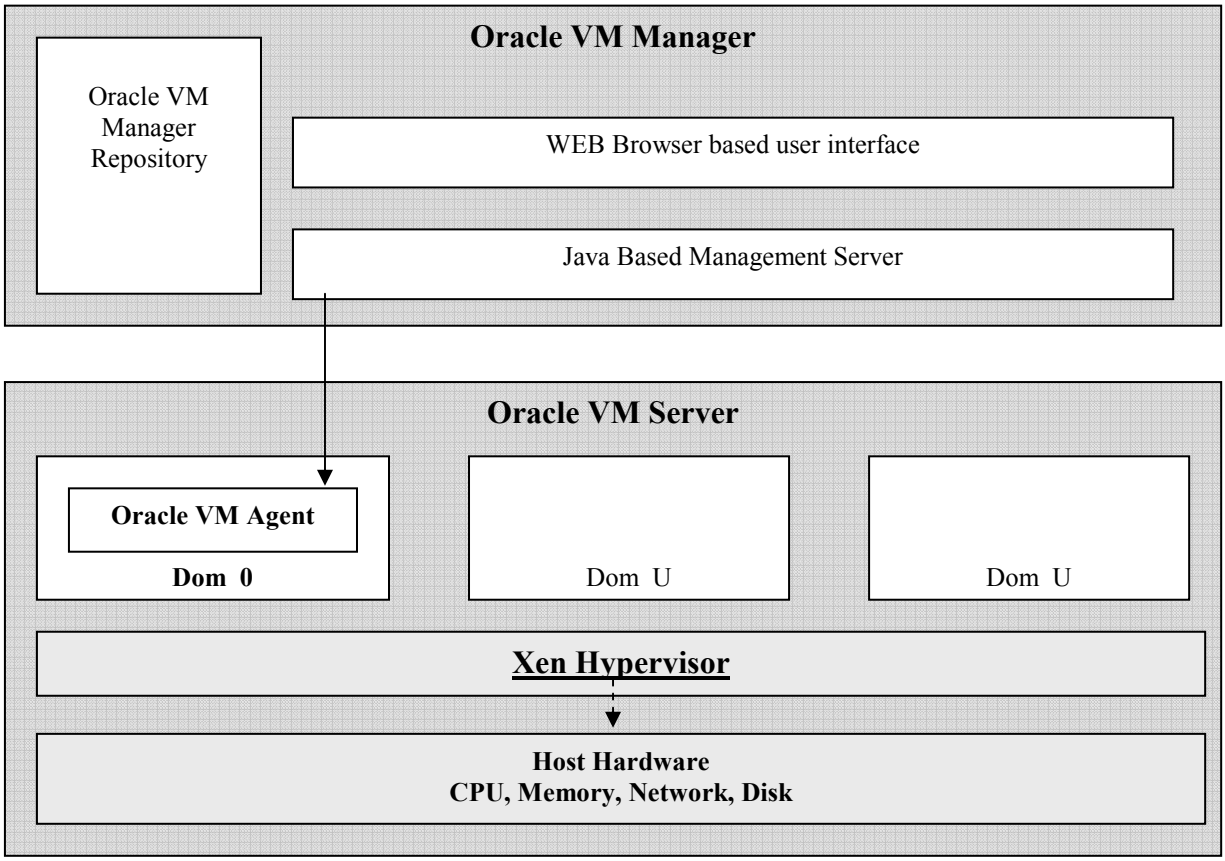
Rys. 7. Architektura *Microsoft Hyper-V*
Fig. 7. The *Microsoft Hyper-V* architekture

1.2. Wirtualizacja serwerów w innych technologiach

W uzupełnieniu ww. wirtualizacji serwerów, istnieje duża grupa rozwiązań innych producentów. Większość z nich bazuje na oryginalnym kodzie technologii *Xen* typu *Open Source*. Wśród nich znajdują się rozwiązania zaproponowane przez firmę *Oracle*. Wprowadzony w 2007r. produkt *Oracle VM* jest *hypervisor'*em typu *bare-metal*. W 2009r. *Oracle* nabył *Virtual Iron*, następne rozwiązanie bazujące na technologii *Xen*, w celu integracji z *Oracle VM*. Przejęcie przez *Oracl*'a w 2010r. firmy *Sun Microsystems*, przyniosło wraz z nią szereg dodatkowych rozwiązań wirtualizacji, które *Sun* opracował lub nabył, w tym także technologii *Solaris*'owych [56] oraz zorientowanego w architekturze *x86* popularnego dla deweloperów narzędzia *VirtualBox* typu *Workbench*.

1.2.1. Wirtualizacja serwerów w technologii Oracle'a

Maszyny wirtualne w technologii *Oracle*'a mogą być tworzone za pomocą rozwiązania *Oracle On Demand Managed Services Grid with Oracle VM* [57]. Jest to produkt w pełni wspierający aplikacje *Oracle*, jak i innych producentów, oferujący skalowalne serwery wirtualizacyjne. Poniżej, na Rys. 8, przedstawiono architekturę *hypervisor*'a *Oracle*'a [58].



Rys. 8. Architektura hypervisor'a *Oracle*'a

Fig. 8. The Oracle hypervisor architecture

1.3. Maszyny wirtualne w technologii *VMware ESXi Server*

Produkowane obecnie przez *VMware Inc*. oprogramowanie *VMware ESXi Server* [59] klasy *enterprise*, dla organizacji i podmiotów gospodarczych różnej wielkości, opiera się na własnym jądrze *vmkernel* [60,61] oraz konsoli zarządzającej, którą jest zmodyfikowany system operacyjny *RedHat Linux* [62,63], posiadający własne sterowniki i obsługujący specyficzny sprzęt komputerowy. System operacyjny *RedHat Linux* to jedna z najstarszych i w swoim czasie najpopularniejszych dystrybucji *Linuks*'a, tworzona przez firmę *RedHat*, która obecnie rozwija się w dwóch gałęziach: niekomercyjny projekt *Fedora* i komercyjna dystrybucja *RedHat Enterprise Linux*.

1.3.1. Tworzenie maszyn wirtualnych w środowisku VMWARE ESXi

Kreację maszyn wirtualnych pracujących w środowisku serwerowym lub klastrowym *VMWARE ESX* [64,65] można dokonać kilkoma metodami: (1) metoda natywna dostępna w oprogramowaniu *VMware vSphere Client*, (2) metoda oparta na przenoszeniu maszyn wirtualnych, (3) metoda wykorzystująca oprogramowanie *VMware vCenter Converter Standalone*, (4) metoda będąca przedmiotem niniejszej pracy, łączącą metodę natywną z funkcjonalnością oprogramowania typu *Backup and Disaster Recovery* o nazwie *Acronis Backup Advanced for PC 11.5*.

Pierwsza metoda, określana jako metoda natywna, dostępna jest w oprogramowaniu *VMware vSphere Client* [66] instalowanym na konsoli zarządzającej klastra, która wykorzystuje wewnętrzny kreator maszyn wirtualnych. Przeznaczona jest do tworzenia maszyn wirtualnych dla systemów *Windows, Linux, Apple Mac, FreeBSD, IBM OS, Novell NetWare, Oracle Solaris, Sun Microsystems, SCO OpenServer, SCO UnixWare* oraz innych systemów 32 oraz 64 bitowych. Za pomocą kreatora tej metody można tworzyć maszyny wirtualne z dyskami wirtualnymi, jak i bez dysków. Kreator pozwala na utworzenie maszyn wirtualnych w wersji 4,7,8,9, z dyskami wirtualnymi o pojemności określanej przez następujące jednostki alokacyjne, tj.: do 2 TB - w przypadku *VMWARE ESXi ver. 5.5* oraz powyżej 2 TB - w przypadku *VMWARE ESXi ver. 6.0*.

W wyniku działania kreatora tworzenia dysku wirtualnego, po udzieleniu kilku odpowiedzi dotyczących między innymi typu dysku wirtualnego (zanim system operacyjny maszyny wirtualnej go użyje), jak również jego wielkości, miejsca utworzenia, następuje jego utworzenie w zasobach serwera *ESXi*. Dostępne typy dysku wirtualnego są następujące:

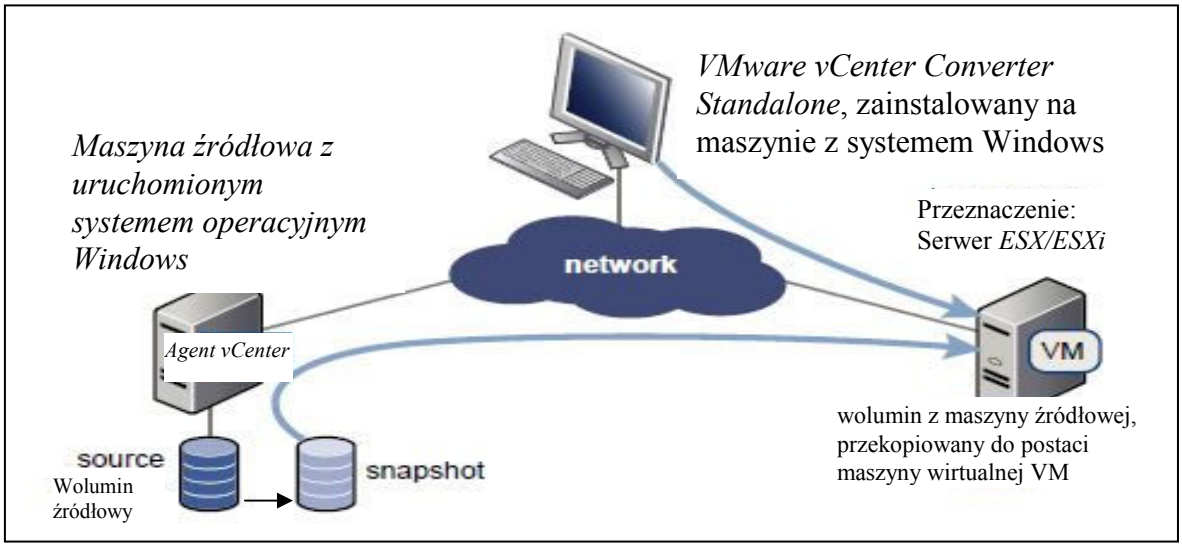
1. *Thin Provisioning* – jest typem dysku maszyny wirtualnej, niezajmującego całej przestrzeni pamięci masowej, która została mu przydzielona w chwili tworzenia dysku przez serwer *ESXi*, a specyficzne sektory dyskowe są wyczyszczone (*zeroed*) z jakiejkolwiek poprzedniej zawartości przy pierwszym zapisie,
2. *Thick Provisioning* - jest typem alokacji w przestrzeni pamięci masowej, w której wielkość pojemności dysku wirtualnego jest zawczasu alokowana w pamięci masowej w chwili tworzenia dysku przez serwer *ESXi*. Rozróżnia się typ: (*i*) *Thick Provision Lazy Zeroed* oraz (*ii*) *Thick Provision Eager Zeroed*, w których specyficzne sektory dyskowe muszą być wyczyszczone z jakiejkolwiek poprzedniej zawartości w następujący sposób:
   - *Thick Lazy* - alokowane zawczasu i zerowane przy pierwszym zapisie,
   - *Thick Eager* - zawczasu alokowane i zerowane.

Po utworzeniu maszyny wirtualnej (z dyskiem wirtualnym) dla konkretnego systemu operacyjnego, w następnym kroku należy zainstalować system operacyjny. Instalacja systemu operacyjnego, mającego działać na tej maszynie wirtualnej, dokonuje się dopiero po jej urucho-

mieniu z poziomu oprogramowania klienckiego *vSphere Client*. Następnie, po uruchomieniu wewnętrznego natywnego kreatora maszyn wirtualnych oraz po wskazaniu mu pliku z obrazem typu **.iso*, zawierającym *boot*'owalną wersję wcześniej wybranego systemu operacyjnego, sterowanie zostaje przekazane kreatorowi systemu operacyjnego, który dokonuje jego instalacji na wcześniej utworzonym dysku wirtualnym.

Druga metoda oparta jest na przenoszeniu maszyn wirtualnych z innych serwerów z systemem *VMWARE ESXi* [67]. Wykorzystuje polecenie *scp,* wbudowane w ten system, pracujący na osnowie *Red Hat Linux*. Istota metody sprowadza się do wykonania dwóch kroków. W kroku pierwszym należy przekopiować, za pomocą polecenia *scp*, odpowiednie pliki typu **.vmdk* (stanowiące wirtualne twarde dyski) z maszyny wirtualnej hosta źródłowego *ESXi* do hosta *ESXi* docelowego. W kroku drugim, na docelowym hoście *ESXi*, należy utworzyć nową maszynę wirtualną (bez wirtualnych dysków twardych), a następnie, jako jej wirtualne dyski twarde należy przyporządkować do niej uprzednio skopiowane pliki typu **.vmdk*. Generalnie, aby móc skorzystać z tej metody, niezbędne jest uzyskanie dostępu do konsoli zarówno źródłowego, jak i docelowego serwera *ESXi*. Dostęp do konsoli serwera źródłowego i docelowego *ESXi*, uzyskuje się z pomocą menu dostępnego na serwerze z systemem *VMWARE ESXi* lub oprogramowania klienckiego *VMware vSphere*.

Trzecia metoda wykorzystuje oprogramowanie *VMware vCenter Converter Standalone v. 6.2* [68], służące do konwersji: (*i*) maszyn fizycznych, z systemami operacyjnymi typu *Windows* i *Linux* oraz (*ii*) zastanych maszyn wirtualnych, konwertowanych do postaci maszyn wirtualnych pracujących w środowisku *VMWARE*. Dla maszyn z systemem operacyjnym *Windows*, zasadę działania oprogramowania *VMware vCenter Converter Standalone* przedstawiono na Rys. 9.

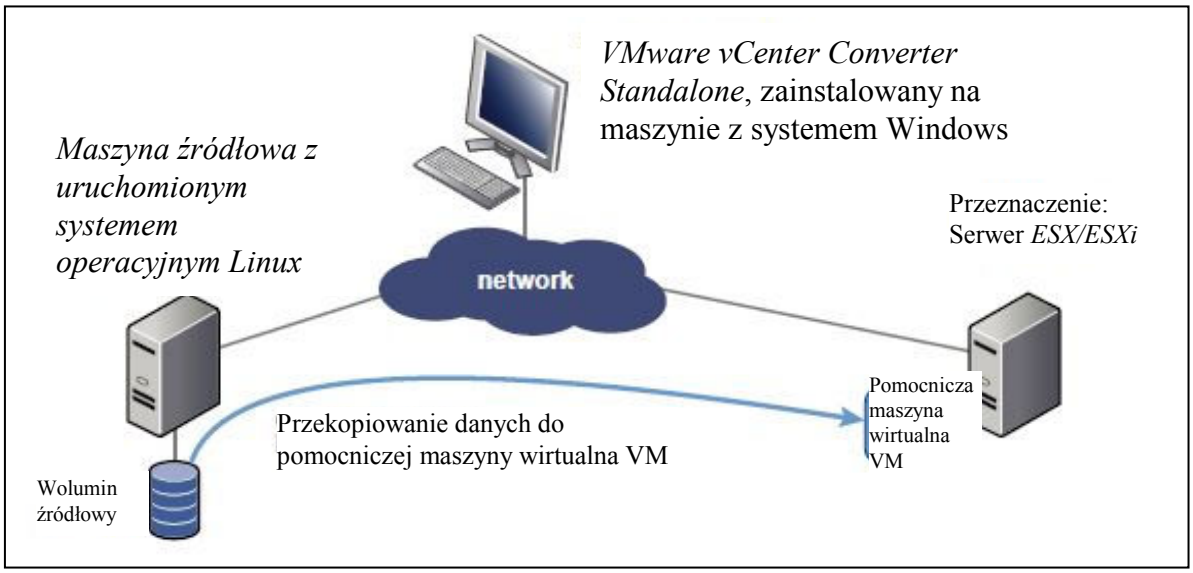


Rys. 9. Zdalne "gorące" klonowanie pracującej maszyny źródłowych z systemem Windows

Fig. 9. Remote hot cloning of powered on source machine that are running Windows

W tym przypadku konwersja następuje w czterech krokach: (1) - *vCenter Converter* przygotowuje maszynę źródłową do konwersji; instaluje agenta na tej maszynie, który wykonuje chwilowy obraz typu *snapshot* woluminu źródłowego, (2) *vCenter Converter* przygotowuje maszynę wirtualną na komputerze docelowym, (3) *vCenter Converter* kompletuje proces konwersji; agent instaluje wymagane sterowniki na maszynie docelowej, pozwalające na restart maszyny wirtualnej oraz jej personalizację (np. zmiana adresu IP), (4) *vCenter Converter* deinstaluje agenta na maszynie źródłowej (opcjonalnie); maszyna wirtualna jest gotowa do pracy na serwerze przeznaczenia.

Dla maszyn z systemem operacyjnym *Linux*, zasadę działania oprogramowania *vCenter Converter* przedstawiono na Rys. 10. W tym przypadku konwersja następuje w trzech krokach: (1) - *vCenter Converter*, używając protokołu *SSH* łączy się z maszyną źródłową, a następnie odczytuje źródłowe informacje, (2) *vCenter Converter* – na podstawie wcześniej odczytanych informacji źródłowych tworzy na maszynie przeznaczenia pomocniczą maszynę wirtualną, która następnie jest na nim uruchamiana, (3) pomocnicza maszyna wirtualna łączy się, za pomocą protokołu *SSH*, z maszyną źródłową i rozpoczyna pobieranie z niej danych.



Rys. 10. Zdalne "gorące" klonowanie pracującej maszyny źródłowych z systemem Linux
Fig. 10. Remote hot cloning of powered on source machine that are running Linux

Ponadto, oprogramowanie *VMware vCenter Converter Standalone* ułatwia konwersję maszyn wirtualnych pomiędzy następującymi produktami: *VMware Workstation, VMware Fusion, VMware Player* oraz wirtualnymi maszynami w środowisku *ESX/ESXi*. W pracy [69] przedstawiono sposób konwersji, za pomocą narzędzia *VMware vCenter Converter Standalone*, rzeczywistego zastanego systemu operacyjnego typu *Microsoft Windows* oraz *Linux* zainsta-

lowanego na fizycznej maszynie, do postaci wirtualnej. To oprogramowanie można użyć także do wykonania innych zadań, tj.:

1. import działających zdalnych fizycznych wirtualnych maszyn do środowiska *ESX/ESXi*,
2. import maszyn wirtualnych obsługiwanych przez *VMware Workstation* lub *Microsoft Hyper-V Server* do hostów *ESX/ESXi*,
3. Eksport maszyn wirtualnych zarządzanych przez hosty serwera *vCenter* do innych formatów maszyny wirtualnej *VMware*.

Czwarta metoda, opisana w następnym Rozdziale 2, będąca przedmiotem niniejszej pracy, służąca do tworzenia maszyn wirtualnych w środowisku *VMWARE ESXi*, stanowi połączenie metody natywnej, dostępnej w oprogramowaniu *VMware vSphere*, z funkcjonalnością oprogramowania typu *Backup and Disaster Recovery* o nazwie *Acronis Backup Advanced for PC 11.5* [70,71,72,73,74] firmy *ACRONIS*.

Zaproponowana metoda stanowi odpowiedź na istniejącą potrzebę: (*i*) skrócenia czasu przestoju systemu informatycznego spowodowaną operacją migracji zastanego systemu informatycznego do postaci wirtualnej działającej na hoście z zainstalowanym oprogramowaniem *hypervisor*'a *WMWare ESXi* lub klastrze, (*ii*) zbudowania testowego środowiska laboratoryjnego umożliwiającego: (*1*) bezpieczne testowanie współpracujących ze sobą różnych wirtualnych kopii systemów informatycznych, odseparowanych od rzeczywistego środowiska produkcyjnego, (*2*) przetestowania, na wirtualnej postaci produkcyjnego serwera fizycznego, nowego oprogramowania mającego działać na serwerze produkcyjnym, (3) szybkiego sprawdzenia, czy eksploatacja wirtualnej wersji fizycznego serwera spełnia wymagania użytkownika i czy jest w ogóle sens podjęcia jego eksploatacji w wersji wirtualnej, (4) wielokrotnego tworzenia maszyn wirtualnych, w przypadku konieczności wykonania zmian konfiguracyjnych, koncepcyjnych itp. dotyczących maszyn ESXi działających samodzielnie lub w klastrze.

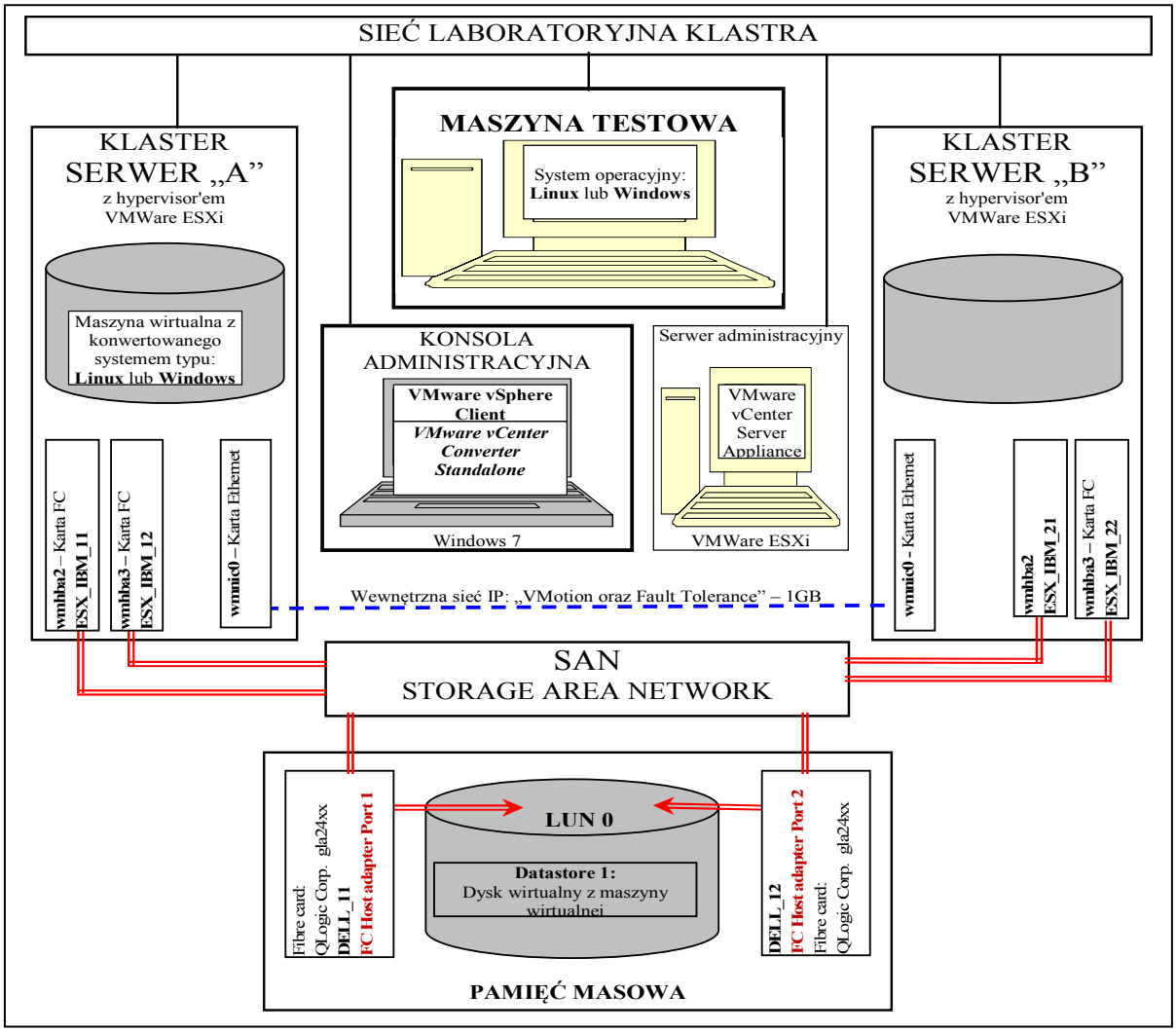
Istota tej metody sprowadza się do wykorzystania funkcjonalności oprogramowania *Acronis Backup Advanced for PC 11.5* w charakterze instalatora systemu operacyjnego w nowo tworzonej maszynie wirtualnej, na podstawie wcześniej wykonanej kopii fizycznego rzeczywistego serwera.

Celem niniejszej pracy było wykazanie, że zaproponowana metoda:

1. nadaje się do wirtualizacji zastanych rzeczywistych fizycznych serwerów, dla których możliwe jest wykonanie kopii całego serwera za pomocą programu typu *Backup and Disaster Recovery* o nazwie *Acronis Backup Advanced for PC 11.5* [75]. W pracy wykazano możliwość wirtualizacji zastanych rzeczywistych fizycznych serwerów, na przykładzie

szpitalnych systemów informatycznych z serwerami *Windows Server 2012R2* oraz *Oracle Linux 6.6* używanych w Szpitalu Specjalistycznym im. Prof. E. Michałowskiego MEDHOLDING S.A., do postaci maszyn wirtualnych działających w laboratoryjnej infrastrukturze klastrowej, przedstawionej na Rys. 11.

2. pozwala na skrócenie czasu przestoju systemu informatycznego spowodowaną operacją migracji do postaci wirtualnej, która w połączeniu z metodą przedstawioną w artykule [76], dotycząca powiększania pojemności dysku wirtualnego, skutkuje radykalnym skróceniu czasu przestoju migrowanego systemu informatycznego, w porównaniu z czasami przestojów spotykanymi w przypadku innych metod.



Rys. 11. Laboratoryjna infrastruktura klastrowa
Fig. 11. Clustered Lab Infrastructure

Ponadto, w pracy dokonano analizy porównawczej zaproponowanej metody, z metodą wykorzystującą oprogramowanie *VMware vCenter Converter Standalone*.

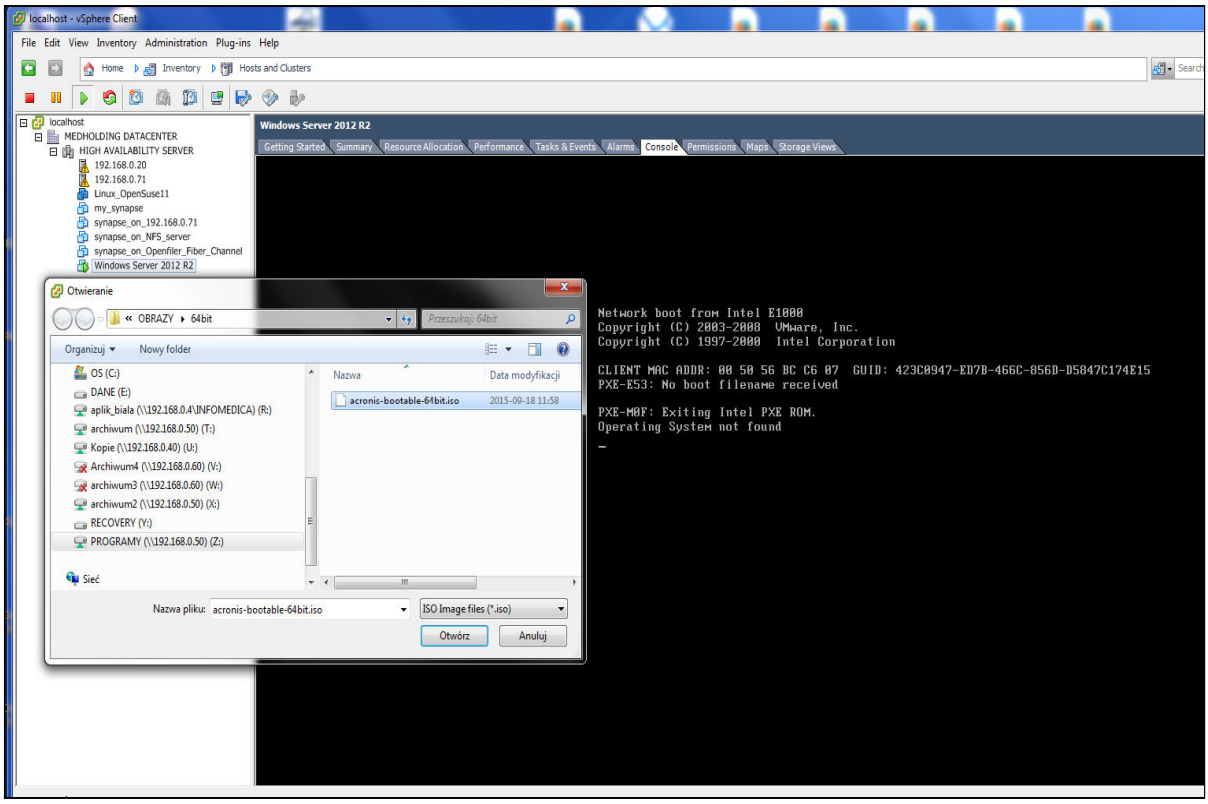
W analizie porównawczej dokonano porównania znormalizowanych czasów tworzenia maszyn wirtualnych w środowisku badawczym dla wybranych systemów operacyjnych typu *Linux* oraz *Microsoft*. Konkretne systemy operacyjne instalowano kolejno na maszynie testowej, którą poddawano konwersji do postaci wirtualnej, za pomocą wymienionych metod, mierząc czasy tworzenia maszyn wirtualnych, które następnie normalizowano.

2. Metoda kreacji maszyn wirtualnych wykorzystująca oprogramowanie typu *Backup and Disaster Recovery*

Metoda kreacji maszyn wirtualnych w środowisku *VMWARE ESXi*, będąca połączeniem metody natywnej, dostępnej w oprogramowaniu *VMware vSphere*, z funkcjonalnością oprogramowania typu *Backup and Disaster Recovery* o nazwie *Acronis Backup Advanced for PC 11.5* firmy *ACRONIS*, polega na wykonaniu następujących operacji:

1. Wykonania kopii rzeczywistego serwera, za pomocą *Acronis Backup Advanced for PC*,
2. Uruchomienia, z poziomu oprogramowania klienckiego *vSphere Client*, wewnętrznego kreatora maszyn wirtualnych i utworzenie nowej maszyny wirtualnej. Następnie, po uruchomieniu tej maszyny (bez systemu operacyjnego – *Operating System not found*) – co pokazano na Rys. 12, należy wskazać kreatorowi maszyn wirtualnych, zamiast pliku **.iso* zawierającego *boot*'owalną wersję systemu operacyjnego mającego być zainstalowanym w maszynie wirtualnej, *boot*'owalną wersję programu *Acronis Backup Advanced for PC*, w celu wykorzystania tego oprogramowania w charakterze instalatora systemu operacyjnego, wraz z danymi, w nowo utworzonej maszynie wirtualnej w środowisku *VMWARE ESXi*, na podstawie wcześniej wykonanej kopii fizycznego rzeczywistego serwera.

Wewnętrzny kreator maszyn wirtualnych dostępny jest pod ikoną o nazwie *Connect/disconect the CD/DVD devices of the virtual machine*. Po wskazaniu tej ikony, wybrano opcję *Connect to ISO image on local disk* i wskazywano lokalizację *boot*'owalnej wersji programu *Acronis Backup Advanced for PC*. Z chwilą, kiedy z poziomu oprogramowania klienckiego *vSphere Client* została uruchomiona *boot*'owalna wersja oprogramowani *Acronis Backup Advanced for PC 11.5,* wówczas wybrano opcję *Recover*, a następnie wskazywano miejsce w sieci komputerowej, gdzie zeskładowano *backup*'ową wcześniej wykonaną kopię zastanego rzeczywistego serwera.
Po wskazaniu dysków, wraz z rekordem *MBR* (*Master Boot Record*), które mają być odtworzone oraz niezbędnych opcji, w tym, w zakładce *Universal Restore for Windows* wskazywano opcję umożliwiającą *boot*'owanie odtworzonego systemu na innym sprzęcie.



Rys. 12. Uruchomienie z poziomu kreatora programu *Acronis Backup Advanced*
Fig. 12. Starting from the wizard Acronis Backup Advanced software

2.1. Tworzenie maszyn wirtualnych z kopii zastanych rzeczywistych serwerów

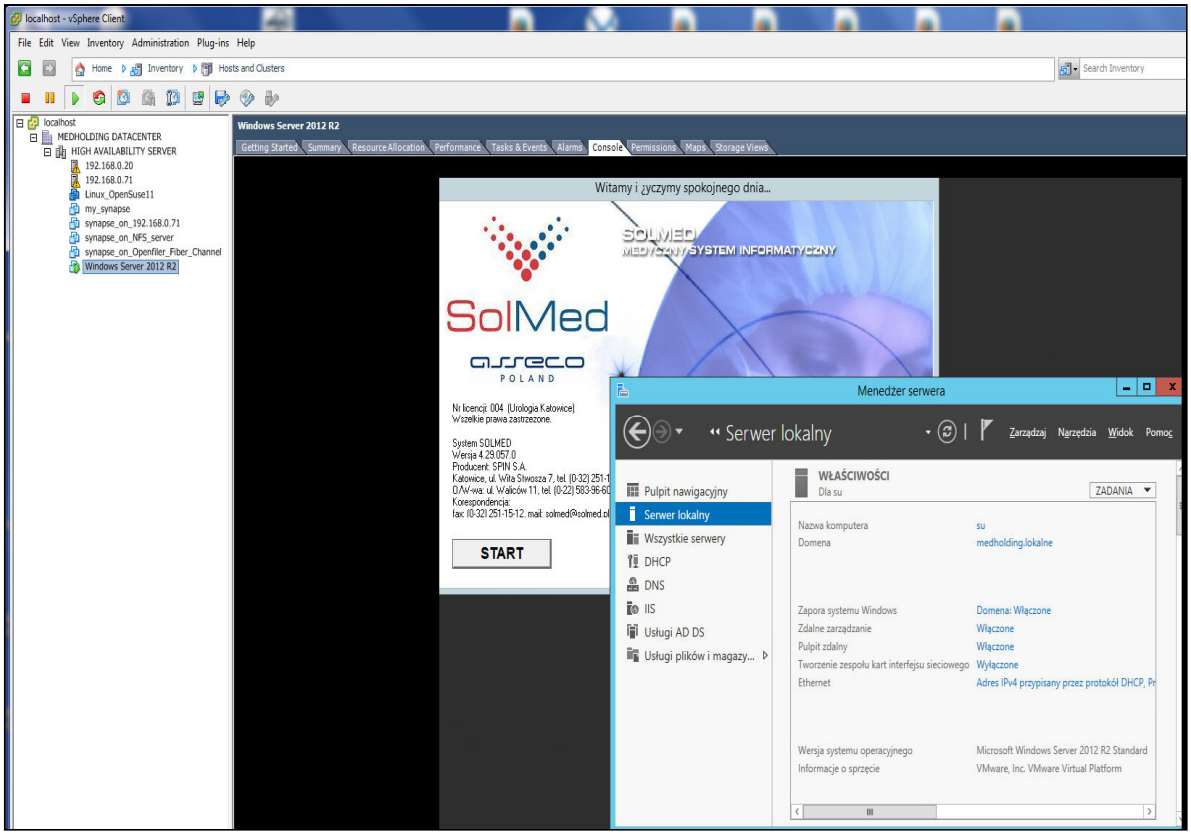
Metodę kreacji maszyn wirtualnych w środowisku *VMWARE ESXi* będącej połączeniem metody natywnej dostępnej w oprogramowaniu *VMware vSphere*, z funkcjonalnością oprogramowania *Acronis Backup Advanced for PC 11.5*, zastosowano do przetestowania w środowisku badawczym, pod kątem możliwości uruchomienia, wirtualnych kopii szpitalnych serwerów wraz z uruchomionymi na nich usługami oraz systemami informatycznymi.

Za pomocą tej metody dokonano wirtualizacji zastanych rzeczywistych fizycznych serwerów z systemem operacyjnym *Windows*, *Linux*, dla których możliwe było wykonanie kopii całego serwera za pomocą programu *Acronis Backup Advanced for PC 11.5*, tj.:

1. *SERWER DELL PowerEdge 320*, z systemem *Windows Serwer 2012 R2* wraz z zainstalowanym na nim kontrolerem *Active Directory,* usługami *DNS, DHCP* oraz zastanym szpitalnym systemem medycznym *Infomedica* i archiwalnym systemem *Solmed* współpracujący z bazą danych *Microsoft SQL Server 2000 Enterprise*,
2. *SERWER IBM System x3400*, z systemem operacyjnym *Oracle Linux 6.6* wraz z zainstalowaną bazą danych *Oracle Database 11g – 64 bit Production* ze szpitalnego systemu informatycznego *Infomedica*.

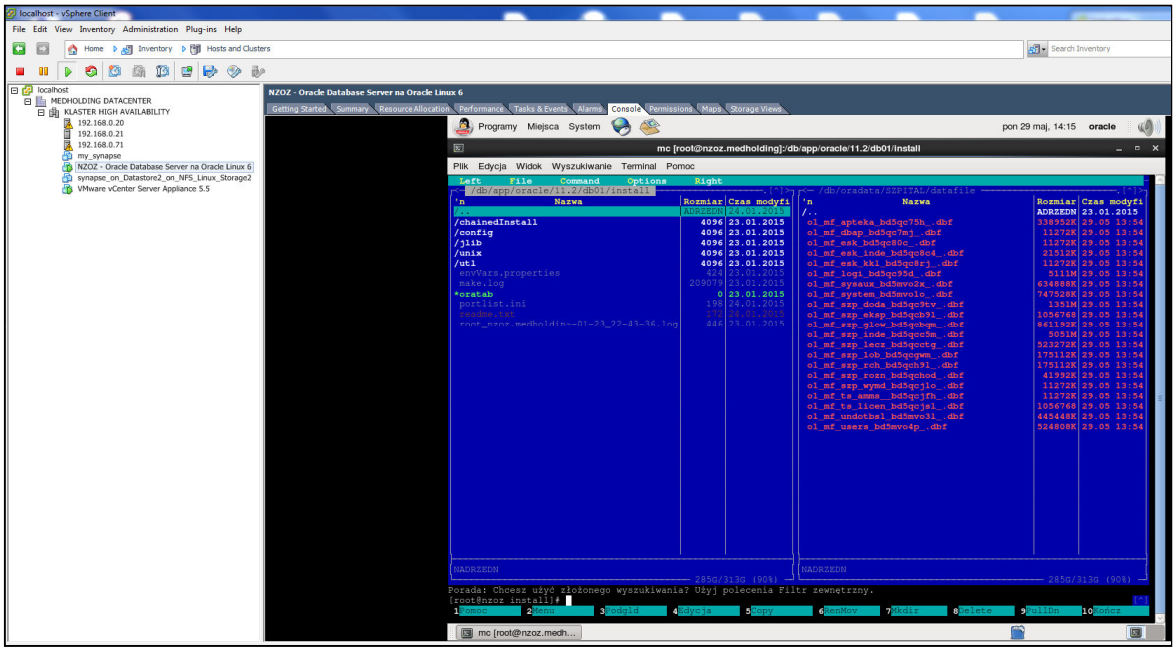
W tym celu, na jednym z hostów *VMWARE ESXi* środowiska badawczego, utworzono oraz uruchomiono nowe maszyny wirtualne z dyskami wirtualnymi większymi niż dyski serwerów fizycznych. Po uruchomieniu, z poziomu oprogramowania klienckiego *vSphere Client*, oprogramowania *Acronis Backup Advanced for PC 11.5*, wybierano opcję *Recover* oraz wskazywano miejsca w sieci komputerowej, gdzie zeskładowano *backup*'owe kopie zastanych fizycznych rzeczywistych serwerów działających na ww. serwerach.

Po uaktywnieniu procesu odzyskiwania danych, utworzone zostały wirtualne maszyny wraz z systemami operacyjnymi, usługami, aplikacjami, bazami danych oraz zwykłymi danymi pochodzącymi z tych serwerów. I tak, jako przykład, zaprezentowany na Rys. 13, pokazano działającą maszynę wirtualną, zawierającą zwirtualizowany fizyczny serwer *DELL PowerEdge 320*, z systemem *Windows Serwer 2012 R2* wraz z zainstalowanym na nim kontrolerem *Active Directory*, usługami *DNS*, *DHCP*, zastanym szpitalnym systemem medycznym *Infomedica* oraz archiwalnym systemem medycznym *Solmed*, współpracujący z bazą danych *Microsoft SQL Server 2000 Enterprise*. Na Rys. 14 pokazano działającą maszynę wirtualną, zawierającą zwirtualizowany fizyczny serwer *IBM System x3400*, z systemem operacyjnym *Oracle Linux 6.6* wraz z zainstalowaną bazą danych *Oracle Database 11g – 64 bit Production*, ze szpitalnego systemu informatycznego *Infomedica*.



Rys. 13. Zwirtualizowany *SERWER DELL PowerEdge 320* z *Windows Server 201R2*

Fig. 13. Virtualized SERWER DELL PowerEdge 320 with Windows Server 201R2



Rys. 14. Zwirtualizowany *SERWER IBM System x3400* z *Oracle Linux 6.6*

Fig. 14. Virtualized SERWER IBM System x3400 with Oracle Linux 6.6

Powyższe przykłady dokumentują wykonanie, za pomocą zaproponowanej metody, wirtualizacji zastanych fizycznych serwerów. Ilustrują możliwość szybkiego zbudowania testowego środowiska laboratoryjnego, umożliwiającego bezpieczne testowanie współpracujących ze sobą różnych wirtualnych kopii systemów informatycznych, które odseparowane są od rzeczywistego środowiska produkcyjnego. Użytkownik pracujący na stacji roboczej w tym środowisku, ze względu na to, że uruchomiono w nim zwirtualizowane fizyczne serwery z usługami *DNS*, *DHCP* oraz *Active Directory*, ma możliwość zalogowania się do sieci testowej, tak jak gdyby logował się do sieci produkcyjnej - z tym samym loginem i tym samym hasłem. Po zalogowaniu się do sieci testowej, ma możliwość podjęcia pracy i wykonania swoich testów na szpitalnym systemie informatycznym typu klient-serwer, współpracujący z bazą danych *Oracle*, działającą na zwirtualizowanym serwerze *IBM System x3400*. Ponadto w tym przykładzie widać, że użytkownik ma możliwość:

1. wielokrotnego tworzenia maszyn wirtualnych, w przypadku konieczności wykonania zmian konfiguracyjnych, koncepcyjnych itp. dotyczących maszyn *ESXi* działających samodzielnie lub w klastrze,
2. szybkiego sprawdzenia, czy eksploatacja wirtualnej wersji fizycznego serwera spełnia wymagania użytkownika i czy jest w ogóle sens podjęcia jego eksploatacji w wersji wirtualnej,
3. przetestowania, na wirtualnej postaci produkcyjnego serwera fizycznego, nowego oprogramowania mającego działać na serwerze produkcyjnym.

3. Analiza porównawcza

Jak wspomniano we wprowadzeniu, w pracy dokonano analizy porównawczej czasów tworzenia maszyn wirtualnych w środowisku dla *hypervisor*'a *VMware ESXi*, według zaproponowanej metody wykorzystującej oprogramowanie *Acronis Backup Advanced for PC 11.5* oraz za pomocą metody wykorzystującej oprogramowanie *VMware vCenter Converter Standalone ver.6.2.0*. W analizie porównawczej dokonano porównania znormalizowanych czasów tworzenia gościnnych maszyn wirtualnych na *hypervisor*'ach *ESXi* w środowisku badawczym, dla wybranych systemów operacyjnych typu *Linux* oraz *Microsoft*.

3.1. Metodyka pomiarowa

Metodyka pomiarowa polegała na instalowaniu na maszynie testowej, pokazanej wcześniej na Rys. 11, kolejnych wybranych systemów operacyjnych. Po ich zainstalowaniu, poddawano je konwersji, za pomocą wymienionych metod, do postaci maszyny wirtualnej alokowanej na hoście z zainstalowanym oprogramowaniem *hypervisor*'a *ESXi*. Podczas konwersji, odczytywano czasy tworzenia maszyn wirtualnych, za pomocą funkcjonalności oferowane przez te programy, które następnie normalizowano.
Konwersje wybranych systemów operacyjnych do postaci wirtualnej, za pomocą oprogramowania *VMware vCenter Converter Standalone ver.6.2.0* zainstalowanego na konsoli zarządzającej zarządzającej, dokonywano podczas ich pracy. Dzięki funkcjonalności tego oprogramowania alokowano je wprost na hoście z *hypervisor*'em *ESXi*.

W przypadku zaproponowanej metody, wykorzystującej oprogramowanie *Acronis Backup Advanced for PC 11.5*, tworzono najpierw ich kopie, a następnie, przed skorzystaniem z tej metody, określano niezbędną wielkość dysku wirtualnego dla nowo tworzonej maszyny wirtualnej na hoście z *hypervisor*'em *ESXi*.

W celu oszacowania wartości parametru określającego wielkość tego dysku wirtualnego dla konwertowanego systemu operacyjnego, pod uwagę brano rzeczywistą pojemność: (1) dysku systemowego (w przypadku zastanego systemu operacyjnego *Windows*), (2) partycji systemowych (w przypadku zastanego systemu *Linux*). Po określeniu niezbędnej wielkości dysku wirtualnego dla nowo tworzonej maszyny wirtualnej na hoście z *hypervisor*'em *ESXi*, uruchamiano proces odczytu danych z kopii, zgodnie z zaproponowaną metodą, którą umieszczano na jednym z dysków konsoli administracyjnej.

Wśród innych parametrów, które ustawiano przed rozpoczęciem procesu zaczytywania danych do dysku wirtualnego w nowo tworzonej maszynie wirtualnej, to: (1) zaznaczano pole typu *check box* w opcji *MBR* (*Master Boot Record*) określające, czy rekord *MBR* ma być przekopiowany do dysku wirtualnego, (2)) zaznaczano odpowiednie pola typu *check box* w

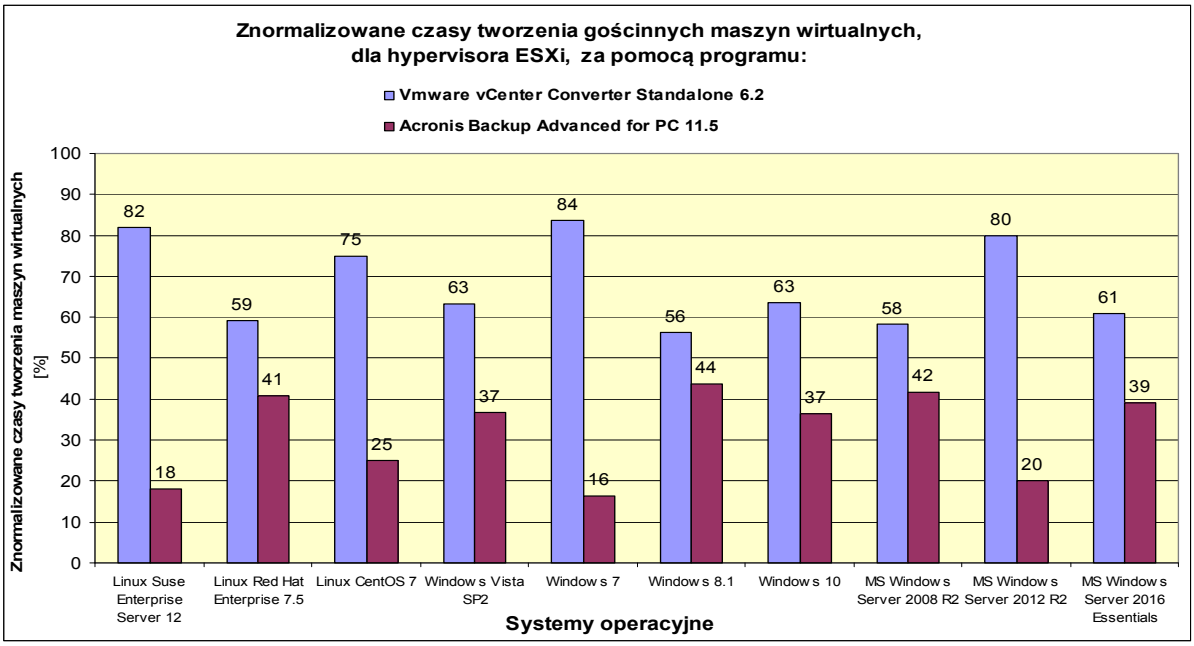
opcji *Basic*, określające, które dyski z kopii zapasowej biorą udział w procesie zaczytywania danych.

Wyniki pomiarów czasu odczytu danych z kopii, podczas tworzenia gościnnej maszyny wirtualnej za pomocą zaproponowanej metody, dla *hypervisor*'a *ESXi*, pokazano w Tab. 1, natomiast znormalizowane czasy odczytu, w postaci graficznej, pokazano na Rys. 15.

CZASY TWORZENIA MASZYN WIRTUALNYCH DLA WYBRANYCH SYSTEMÓW OPERACYJNYCH				
System operacyjny	Czasy tworzenia maszyn wirtualnych dla hypervisor'a typu ESXi za pomocą:			
	Vmware vCenter Converter Standalone 6.2		Acronis Backup Advanced for PC 11.5	
	[min]	[%]	[min]	[%]
Linux Suse Enterprise Server 12	9	**82**	2	**18**
Linux Red Hat Enterprise 7.5	16	**59**	11	**41**
Linux CentOS 7	9	**75**	3	**25**
Windows Vista SP2	12	**63**	7	**37**
Windows 7	36	**84**	7	**16**
Windows 8.1	9	**56**	7	**44**
Windows 10	59	**63**	34	**37**
MS Windows Server 2008 R2	7	**58**	5	**42**
MS Windows Server 2012 R2	12	**80**	3	**20**
MS Windows Server 2016 Essentials	14	**61**	9	**39**

Tab. 1. Czasy tworzenia maszyn wirtualnych dla *hypervisor*'a *ESXi*

Tab. 1. Times for creating virtual machines for ESXi hypervisor



Rys. 15. Znormalizowane czasy tworzenia maszyn wirtualnych dla *hypervisor*'a *ESXi*

Fig. 15. Standarized times for creating virtual machines for ESXi hypervisor

Obserwacja 1.

Z przedstawionego wykresu znormalizowanych czasów tworzenia maszyn wirtualnych dla *hypervisor*'a *ESXi* widać, że czas tworzenia maszyn wirtualnych za pomocą zaproponowanej metody, wykorzystujące oprogramowanie *Acronis Backup Advanced for PC 11.5*, jest krótszy niż w przypadku oprogramowania *VMware vCenter Converter Standalone ver.6.2.0*.

4. Podsumowanie i Wnioski

4.1. Podsumowanie

W niniejszej pracy wykazano praktyczną możliwość wirtualizacji rzeczywistego fizycznego środowiska informatycznego do postaci maszyn wirtualnych działających w środowisku *VMWARE ESXi*, za pomocą zaproponowanej metody, będącej połączeniem metody natywnej, dostępnej w oprogramowaniu *VMware vSphere*, z funkcjonalnością oprogramowania typu *Backup and Disaster Recovery* o nazwie *Acronis Backup Advanced for PC 11.5*. Wirtualizacja za pomocą tej metody możliwa jest dla rzeczywistych fizycznych serwerów z różnymi systemami operacyjnymi i systemami informatycznymi, dla których możliwa jest do wykonania kopia całego serwera za pomocą tego oprogramowania.

Wirtualizacja rzeczywistych serwerów za pomocą tej metody daje następujące korzyści:

1. radykalne skrócenia czasu procesu tworzenia wirtualnych wersji działających rzeczywistych serwerów, w stosunku do czasu potrzebnego do ręcznego przeniesienia ich systemów operacyjnych, usług, oprogramowania i aplikacji, baz danych oraz zwykłych danych pochodzących z tych serwerów.
2. zmniejszenie kosztów instalacyjnych; nie trzeba powtórnie ponosić kosztów związanych z procesem ręcznej instalacji systemów operacyjnych i systemów użytkowych.

Do zalet tej metody zaliczyć można możliwość:

1. wielokrotnego tworzenia maszyn wirtualnych, w przypadku konieczności wykonania zmian konfiguracyjnych, koncepcyjnych itp. dotyczących maszyn *ESXi* działających samodzielnie lub w klastrze,
2. szybkiego sprawdzenia, czy eksploatacja wirtualnej wersji fizycznego serwera spełnia wymagania użytkownika i czy jest sens podjęcia jego eksploatacji w wersji wirtualnej,
3. przetestowania, na wirtualnej postaci produkcyjnego serwera fizycznego, nowego oprogramowania mającego docelowo działać na serwerze produkcyjnym,

4. szybkiego zbudowania testowego środowiska laboratoryjnego, umożliwiającego bezpieczne testowanie współpracujących ze sobą różnych wirtualnych kopii systemów informatycznych, odseparowanych od rzeczywistego środowiska produkcyjnego.

4.2. Wnioski

1. Na podstawie przeprowadzonych pomiarów wynika, że znormalizowane czasy tworzenia maszyn wirtualnych dla *hypervisor*'a *WMWare ESXi* za pomocą zaproponowanej metody, wykorzystujące oprogramowanie *Acronis Backup Advanced for PC 11.5*, są krótsze niż w przypadku oprogramowania *VMware vCenter Converter Standalone ver.6.2.0*.

LITERATURA

1. Niemi J.: Empowering IT Solutions witch server virtualization, Turku University of Applied Sciences, 2012.
2. Metzler J.: Virtualization: Benefits, Challenges, and Solutions, Riverbed Technology, 2011.
3. Suresh S.: A Study on System Virtualization Techniques, International Journal of Advanced Research in Computer Science & Technology, ISSN: 2347–9817, 2014.
4. Daniels J.: Server Virtualization Architecture and Implementation, Crossroads, Fall 2009/Vol. 16, No.1, 2009.
5. Daniluk D.: Wirtualne serwery na bazie oprogramowania VMware GSX/ESX wspomaganego przez VMware VirtualCenter, 2006r.
6. Creasy R. J.: The origin of the vm/370 time-sharing system, IBM Journal of Research and Development 25 (5) 483–490, 1981.
7. Srodawa R. J., Bates L. E.: An efficient virtual machine implementation. In Proceedings of ACM SIGARCH-SIGOPS Workshop on Virt. Comp. Systems, 1973.
8. Goldberg R. P.: Architecture of virtual machines, in: Proceedings of the workshop on Virt. Comp. Systems, ACM Press, New York, USA, pp. 74–112, 1973.
9. Popek G. J., Goldberg R. P.: Formal requirements for virtualizable third generation architectures, Commun. ACM 17 (7) 412–421, 1974.
10. VMWare Inc.: Introducing VMware virtual platform, technical white paper, 1999.
11. Rose R.: Survey of System Virtualization Techniques, 2004.
12. VMware white paper: Virtualization overview. http://www.hubtech.com/resource/vmware-virtualization-overview-white-paper/.
13. VMware white paper: Virtualization: Architectural Considerations And Other Evaluation Criteria.
14. Barham P., Dragovic B., Fraser K., Hand S., Harris T., Neugebauer A., Pratt I., Warfield A.: Xen and the art of virtualization, in: SOSP ’03: Proceedings of the nineteenth ACM symposium on Operating systems principles, ACM Press, New York, USA, pp. 164–177, 2003.
15. Vait M.: Platform Virtualization for F2F Computing, University of Tartu, Institute of Computer Science, 2011.

16. Portnoy M.: Virtualization-essential, Copyright by John Wiley & Sons, Inc., Indianapolis, Indiana, ISBN: 978-1-118-17671-9, 2012.
17. Dueñas J., Ruiz J., Cuadrado F., García B., Hugo A., Parada G.: System Virtualization Tools to the Rescue of Software Developers, Universidad Politécnica de Madrid.
18. Sodhi B.: Topics in Virtualization and Cloud Computing, Dept. of Computer Science and Engineering, IIT Ropar PB 140001 India.
19. Jayaraman A., Rayapudi P.: Comparative Study of Virtual Machine Software Packages with Real Operating System, Blekinge Institute of Technology, 2012.
20. Pearce M., Zeadally S., Hunt R.: Virtualization: Issues, Security Threats, and Solutions, University of Canterbury, University of The District of Columbia, University of Canterbury, ACM Computing Surveys, Vol. 45, No. 2, Article 17, 2013.
21. Soni L.: Case Study of Virtualization in Existing System, International Journal of Advanced Research in Computer Science.
22. The Center for Internet Security: Virtual Machine Security Guidelines, Editor: Joel Kirch, WBB Consulting, 2007.
23. Zaidenberg N.J.: Applications of Virtualization in Systems Design, JYVÄSKYÄ STUDIES IN COMPUTING 153, UNIVERSITY OF JYVÄSKYLÄ, 2012.
24. Lee H.: Virtualization Basics: Understanding Techniques and Fundamentals, Bloomington, IN 47408, Indiana University.
25. Rodr´ıguez-Haro F., Freitag F., Navarro L., Hern´andez-S´anchez E., Far´ıas-Mendoza N., Guerrero-Ib´a˜nez J.A., Gonz´alez-Potes A.: A summary of virtualization techniques, The Iberoamerican Conference on Electronics Engineering and Computer Science, 2012.
26. Yu Y.: OS-level Virtualization and Its Applications, Computer Science Stony Brook University, 2007.
27. Laadan O., Nieh J.: Operating System Virtualization: Practice and Experience, Columbia University, 10027, New York, 2010.
28. Soltesz S. , Pötzl H., Fiuczynski M., Bavier A., Peterson L.: Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors, EuroSys'07, ACM 978-1-59593-636-3/07/0003, 2007.
29. Padhy R., Patra M. Satapathy S.: Virtualization techniques &Technologies: State-of-the-Art, Journal of Global Research in Computer Science, ISSN-2229-371X, Volume 2, No. 12, 2011.
30. Whitaker A., Shaw M, Gribble S.D.: Lightweight virtual machines for distributed and networked applications, Tech. Rep., Feb. 08, 2002.
31. Sanjay P. Ahuja: Full and Para Virtualization, Fidelity National Financial Distinguished Professor of CIS, School of Computing, UNF.
32. Bellard F.: QEMU, a fast and portable dynamic translator, in: USENIX, Annual Technical Conference, USENIX, pp. 41–46, 2005.
33. Jin Y., Wen J., Chen Q., Zhu Z.: An Empirical Investigation of the Impact of Server Virtualization on Energy Efficiency for Green Data Center, Nanyang Technological University, Yangtze Delta Institute of Tsinghua University, University of Science and Technology of China, 2013.
34. Ghorpade Y., Ghorpade S., Bennur T., Acharya H.S.: Server virtualization: a cost effective and green computing approach towards educational infrastructure management, International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, 2013.

35. Ghorpade Y., Kamatchi R.: A Study to Develop Server Virtualization Infrastructure in Educational Institutions with Cost Effective and Green Computing Approach, Computer Science Department, Bharathiar University, Tamilnadu, Amity School of Engineering & Technology, India, International Journal of Computer Science Trends and Technology (I JCST) – Volume 4 Issue 2, 2016.
36. Ronkainen N.: Server Virtualization, Lappeenranta University of Technology, 2003.
37. Shrithi H., Vanamala C.: Applications of Server Virtualization Technology, International Journal of Computer Science and Information Technologies, Vol. 5 (3), 4214-4217, 2014.
38. Skubch N., Klausnitz R.: Trends in Virtualization and their implications, JSC Managementunt Technologleberatung AG, 2012.
39. Badger L.: Server Virtualization Techniques, The Volgenau School of Information Technology and Engineering.
40. Herrmann J., Novich L., Zimmerman Y., East J., Parker D., Radvan S.: Red Hat Enterprise Linux 6 Virtualization Getting Started Guide, Red Hat, Inc., 2017.
41. Trends: Trends In Server Virtualization, Arcserve (USA), LLC and its affiliates and subsidiaries, 2017.
42. Graniszewski W., Arciszewski A.: Performance analysis of selected hypervisors, INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2016, VOL. 62, NO. 3, PP. 231–236 Manuscript received August 12, 2016; revised September, 2016.
43. Pousa D., Rufino J.: EVALUATION OF TYPE-1 HYPERVISORS ON DESKTOP-CLASS VIRTUALIZATION HOSTS, IADIS International Journal on Computer Science and Information Systems, Vol12, No.2, ISSN: 1646-3692.
44. Bittman T., Dawson P., Warrilow M.: Magic Quadrant for x86 Server Virtualization Infrastructure, ID: GOO289889, 2016.
45. Durairaj M., Kannan P.: A study On Virtualization Techniques And Challenges In Cloud Computing, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSN 2277-8616, ISSUE 11, 2014.
46. Kašpar J., Tvrdík P.: Virtualization Techniques II, Department of Computer Systems Faculty of Information Technology Czech Technical University in Prague, 2011.
47. Truong P., Mathieu B., Aguessy F., Bettan O., Nguyen T., Ploix T., Aoun M., Doyen G., Lahmadi A, Cholez T.: Virtualization Techniques: Analysis and Selection, Project ANR-14-CE28-0001, 2015.
48. Oguchi Y., Yamamoto T.: Server Virtualization Technology and its Latest Trends, Fujitsu Sci. Tech. J.,44,1, p.46-52, 2007.
49. Timalsen P.: A Study of The impact of Virtualization on Computer Networks, University Of Oslo, 2013.
50. vSphere 5 Documentation Center - Creating a vSphere HA Cluster, https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.avail.doc_50%2FGUID-E90B8A4A-BAE1-4094-8D92-8C5570FE5D8C.html.
51. vSphere 5 Documentation Center - vSphere Availability, EN-000916-00, 2012.
52. BOK Z.: Analiza wpływu zdarzeń krytycznych na pracę klastra VMware ESXi i maszyn wirtualnych", Zeszyty Naukowe Politechniki Śląskiej, seria Studia Informatica, artykuł zgłoszony do druku w 2017.
53. Microsoft, Windows Server 2012: Server Virtualization.

54. Shah Z.H.: Windows Server 2012 Hyper-V: Deploying Hyper-V Enterprise Server Virtualization Platform, Packt Publishing, ISBN 978-1-84968-834-5, UK, 2013.
55. Orin T.: Microsoft Certification Exam 74-409: Server Virtualization with Windows Server Hyper-V and System Center, https://piermick.files.wordpress.com/2015/11/orinthomas_1-8_chapter_study_guide-4.pdf.
56. Ozawa A., Suzuki K., Taoda M.: Oracle Solaris Virtualization Technologies, FUJITSU Sci. Tech. J., Vol. 47, No. 2, 2011.
57. An Oracle White Paper: Oracle On Demand Infrastructure: Virtualization with Oracle VM, 2007.
58. Oracle Inc., Quick Start Guide, Release 2.2, E15445-01, 2009.
59. vSphere ESXi Bare-Metal Hypervisor, VMware Polska, 2015.
60. How To Install and Update A Redhat Linux Kernel RPM, 2004.
61. Ball B.: Red Hat Linux 7.3, "Księga Experta", Tłumaczenie: Pasternacki M., ISBN: 83-7197-787-5, Wydawnictwo HELION, 2002.
62. Red Hat Linux, https://pl.wikipedia.org/wiki/Red_Hat_Linux.
63. Red Hat Enterprise Linux 7 System Administrator's Guide, 2016.
64. VMware vSphere 5.1 Documentation Center, VMware, Inc., 2017.
65. vSphere 5 Documentation Center - Creating a vSphere HA Cluster, https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vsphere.avail.doc_50%2FGUID-E90B8A4A-BAE1-4094-8D92-8C5570FE5D8C.html.
66. VMware vSphere 5.5 Documentation Center, VMware, Inc., 2017.
67. Brzeżek I.: VMWare ESXi przenoszenie maszyn wirtualnych, 2013.
68. VMware vCenter Converter Standalone User's Guide, EN-001951-00, 2016.
69. Ali I., Meghanathan N.: Virtual machines and networks – installation, performance, study, advantages and virtualization options, Jackson State University, International Journal of Network Security & Its Applications, Vol.3, No.1, 2011.
70. Acronis Backup & Recovery 11, Copyright © Acronis, Inc., 2011.
71. Acronis Backup & Recovery 11.5, Podręcznik Instalacji, 2012.
72. Acronis Backup Advanced 11.5 Solution Guide and Best Practices, 2015.
73. Acronis Backup Advanced Version 11.5 Update 6, User Guide, Acronis GmbH, 2015.
74. Acronis Backup Advanced 11.7 Update 1, User Guide, Acronis GmbH, 2016.
75. Acronis Backup Advanced Version 11.5 Update 6, Instrukcja szybkiego rozpoczęcia pracy, Acronis GmbH, 2015.
76. BOK Z.: Heterogeniczne dyski wirtualne w środowisku VMWare ESXi, Studia Informatica, Politechnika Śląska, artykuł zgłoszony do druku w 2018.

Recenzent:

Wpłynęło do Redakcji 30 sierpnia 2018 r.

Abstract

In this article the creation of virtual machines on the new WMware ESXi servers or clusters has been described. In work described several methods, including: native method available in VMware vSphere, method based on the moving virtual machines from existing servers VMware ESXi, the method of using VMware vCenter Converter Standalone. In the work a method to create virtual machines has been proposed using the native method combined with the functionality of the "Backup and Disaster Recovery" type software named "Acronis Backup Advanced for PC" from ACRONIS company, which that functionality has been used as the operating system installer in newly created virtual machine. This method has been used to create a virtual machines running on a VMWARE ESXi environment.

Based on conducted measurements he results, that standardized times of creating virtual machines for ESXi hypervisor with the suggested method, using the Acronis software Advanced Backup for 11.5 PC, are shorter than in case of the VMware vCenter Converter Standalone ver.6.2.0 software.