

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

ANALIZA WPLYWU ZDARZEŃ KRYTYCZNYCH NA PRACĘ KLASTRA VMWARE ESXI I MASZYN WIRTUALNYCH

Streszczenie. W artykule dokonano analizy wpływu zdarzeń krytycznych na pracę maszyn wirtualnych, pod kątem ciągłości działania i oceny dostępności, w klastrze *VMware ESXi* wysokiej dostępności z pełną redundancją połączeń światłowodowych, na przykładzie fragmentu specjalistycznego szpitalnego Systemu Radiografii Pośredniej *Synapse* firmy *FUJIFILM Medical Systems USA, Inc.*, używanego w Szpitalu Specjalistycznym im. Prof. E. Michałowskiego MEDHOLDING S.A. W celu dokonania oceny dostępności tego systemu, uruchomionego w postaci maszyny wirtualnej na jednym z hostów laboratoryjnej infrastruktury klastrowej *VMWARE ESXi*, zaproponowano metodę pomiaru czasu potrzebnego na restart tej maszyny wirtualnej na innym hoście w tym klastrze, w sytuacjach wystąpienia zdarzeń krytycznych, powodującej awarię hosta na której została uruchomiona.

ANALYSIS OF THE INFLUENCE OF CRITICAL EVENTS ON THE WORK OF THE VMWARE ESXI CLUSTER AND VIRTUAL MACHINES

Summary. In the article they made analysis of the impact of critical events in the work of the virtual machines, for business continuity and the assessment of the availability, in VMware ESXi cluster high availability with full redundancy fiber connectivity on the example of the portion of the specialized hospital System of indirect Radiography *Synapse* FUJIFILM Medical Systems USA, Inc., used in Specialized Hospital named Prof. E. Michalowski MEDHOLDING S.A. In order to assess the availability of the system that is running as a virtual machine on one of the host in *VMWARE ESXi* laboratory cluster infrastructure, a method for measuring the time needed to restart this virtual machine on another host in this cluster, in situations of critical events that caused the failure of the host at which it was running, has been proposed.

1. Wprowadzenie

Technologia *VMware ESXi* firmy *VMware Inc.* [1], wspiera wiele różnych i bardzo drogie systemów pamięci masowych typu *SAN (Storage Area Network)* [2,3] w różnych konfiguracjach. Ogólnie można powiedzieć, że firma *VMware Inc.* tworzy i rozwija systemy *ESXi* [4,5] wraz ze wspieranymi przez nie systemami kosztownych pamięci masowych innych producentów [6], tj.: *EMC, IBM, HP, HITACHI, LENOVO, NetApp*, w zakresie ich podstawowej funkcjonalności, odporności na uszkodzenia *FT (Fault Tolerance)* [7] i awarie *HBA (Host Bus Adapters)* [8]. Z technologią *VMware ESXi* współpracuje również ekonomiczna pamięć masowa typu *Fiber Channel SAN Storage* [9,10,11,12,13,14,15,16,17,18], pracująca pod kontrolą systemu operacyjnego *Linux Openfiler ESA* [19,20,21,22,23,24,25]. Współpracuje ona z klastrem wysokiej dostępności *VMware ESXi HA (High Availability)* [26,27] w zakresie podstawowej zdolności przyłączeniowej *BC (Basic Connectivity)*, odporności na uszkodzenia *FT* oraz pełnej redundancji połączeń światłowodowych pomiędzy pamięcią masową a klastrem. *Linux Openfiler ESA* jest systemem operacyjnym, który zapewnia obsługę plikowych (*file-based*) typu *NAS (Network-Attached Storages)* oraz blokowych (*blok-based*) pamięci masowych typu *SAN Storage*.

W skład oprogramowania systemu operacyjnego *VMware ESXi* wchodzi własne jądro systemu operacyjnego *VMkernel* [28,29], które jest zmodyfikowanym systemem operacyjnym *Red Hat Linux*, przeznaczonym do wsparcia i zarządzania wirtualnymi serwerami, maszynami wirtualnymi (*VM*) i aplikacjami uruchamianymi z poziomu maszyn wirtualnych, z własnymi sterownikami obsługującymi specyficzny sprzęt komputerowy oraz konsola zarządzająca. Ze względu na ograniczone możliwości administracji *VMware ESXi*, monitoringu i wglądu w środowisko systemu [30] oraz maszyn wirtualnych na nim działających, pojawiły się pakiety serwerowego oprogramowania monitorującego firm trzecich, jak np.: *Monitoring ESXi Performance Using Server & Application Monitor* firmy *SolarWinds* [31], *ServerView Event Manager* firmy *Fujitsu* [32], czy też *Seagate Nytro vCenter Plug-In Solution* [33].

Wraz z wprowadzeniem *VMware Infrastructure 3* [34], *VMware Inc.* rozszerza ewolucję wirtualnej infrastruktury i wirtualnych maszyn, zapoczątkowaną przez *ESX Server v1.0*, wprowadzając nowy zbiór usług dla optymalizacji zasobów, wysokiej dostępności, ochrony danych oraz dostarczaniem nowych możliwości, które przedtem wymagały złożonych i drogie rozwiązań bazujących na maszynach fizycznych. *VMware ESX Server* wprowadza warstwę wirtualizacyjną, umiejscowioną pomiędzy warstwą sprzętową a systemem operacyjnym, która uogólnia procesor, pamięć operacyjną i masową, jak również zasoby sieciowe do postaci maszyn wirtualnych pracujących na tym samym fizycznym serwerze. Każda wirtualna maszyna reprezentuje kompletny system z procesorem, usługami sieciowymi, pamięcią oraz

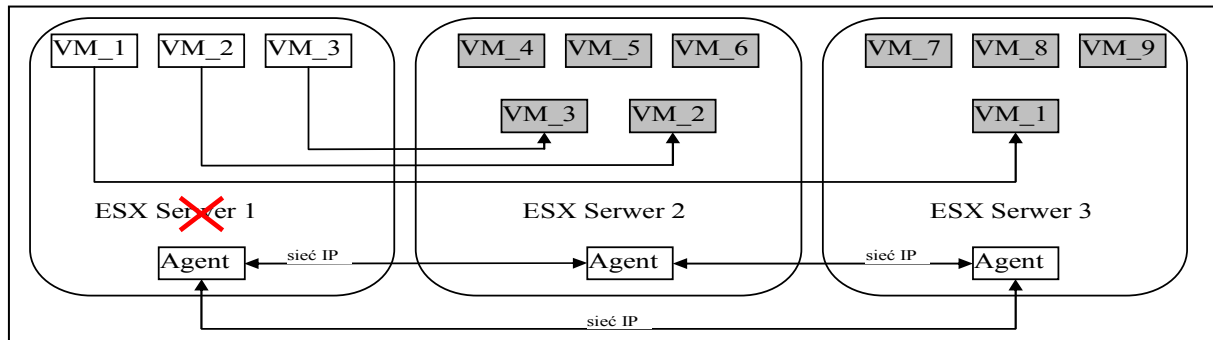
BIOS'em w taki sposób, że systemy operacyjne oraz aplikacje mogą w nich działać bez żadnych modyfikacji.

Klastry, będące kolejną koncepcją w zarządzaniu wirtualną infrastrukturą, dają duże możliwości w uproszczeniu zarządzania wieloma hostami jako pojedynczej jednostki. Wsparcie klastrowe w *VMware Infrastructure 3* redukuje złożoność zarządzania poprzez łączenie wolno stojących hostów w jeden klastrer z dzielonymi zasobami i z natury większą dostępnością. Klastry *VMware* pozwalają na agregowanie zasobów sprzętowych indywidualnych hostów *ESX* i zarządzanie nimi jak gdyby rezydowały na pojedynczym hoście. *VMware Infrastructure 3* dostarcza dwie usługi, tj.: *VMware DRS (Distributed Resource Scheduler)* oraz *VMware HA (High Availability)* i *FT (Fault Tolerance)* w celu ułatwienia zarządzania klastrami *VMware*. Usługa *VMware DRS* pozwala na automatyczne początkowe umiejscowienie maszyny wirtualnej na pewnym hoście w klastrze oraz automatyczną relokację i optymalizację zasobów w sytuacjach dodawania lub usuwania hostów z klastra, natomiast usługa *VMware HA* jest jedną z najlepszych funkcjonalności *ESX*, stanowiącą ekonomiczną alternatywę dla tradycyjnych serwerów klastrowych, zapewniających wysoką dostępność poprzez szybkie odzyskiwanie maszyn wirtualnych z *host'ów* które uległy awarii.

Usługa *VMware vSphere HA* bazuje na zmodyfikowanej wersji *EMC/Legato Automated Availability Manager (AAM) 5.1.2.*, zakupionego przez *VMware* dla *VMware VI3*. *VMware vSphere HA* monitoruje wszystkie fizyczne hosty w klastrze za pomocą agentów [35] umieszczonych na każdym z hostów klastra, których zadaniem jest utrzymywanie mechanizmu wzajemnej wymiany informacji o stanie innych hostów w klastrze, tzw. *heartbeats* oraz monitorowanie aktywności hostów w klastrze. Jeśli agenci nie otrzymają informacji poprzez *heartbeats* z pewnego hosta, jak również nie mogą do niego wysłać polecenia *ping*, wówczas uznają go jako host, który uległ awarii. Awaria hosta jest wykrywana po tym, jak usługa *HA* zainstalowana na hoście zaprzestaje wysyłania *heartbeats* do innych hostów w klastrze. Host zaprzestaje wysyłania *heartbeats*, jeśli doszło do wystąpienia jednego z poniższych zdarzeń krytycznych, tj. jeśli: (i) host jest izolowany od sieci, tzn. kiedy host ciągle działa ale nie może komunikować się z innymi hostami w klastrze, (ii) host się zawiesił, (iii) host jest całkowicie wyłączony z powodu awarii sprzętu. W przypadku wystąpienia awarii hosta, dzięki funkcjonalności *VMware vSphere HA* doprowadza się do inicjacji procesu restartu wszystkich maszyn wirtualnych z tego hosta na innym hoście.

Przedstawiony na Rys. 1. klastrer *VMware ESX HA FT*, posiada oprogramowanie klastrowe *VMware HA* wraz z agentami, zainstalowane na każdym hoście, które przy wykorzystaniu wewnętrznej sieci IP w ciągły sposób monitoruje wszystkie hosty w klastrze, w celu detekcji awarii. Jeśli jeden z nich ulegnie awarii, wówczas usługa *VMware HA*

natychmiast reaguje, poprzez automatyczny restart każdej dotkniętej awarią wirtualnej maszyny na innym hoście.



Rys. 1. Klaster VMware HA

Fig. 1. Cluster VMware HA

Oprogramowanie klastrowe pozwala, pracującym na określonym hoście wirtualnym maszynom, na automatyczny restart na innych hostach w zasobach klastra, w przypadku awarii maszyny lub wystąpieniu zdarzeń krytycznych, zapewniając tym samym wysoką dostępność dla wszystkich aplikacji uruchomionych w wirtualnych maszynach oraz minimalizując czas ich postoju.

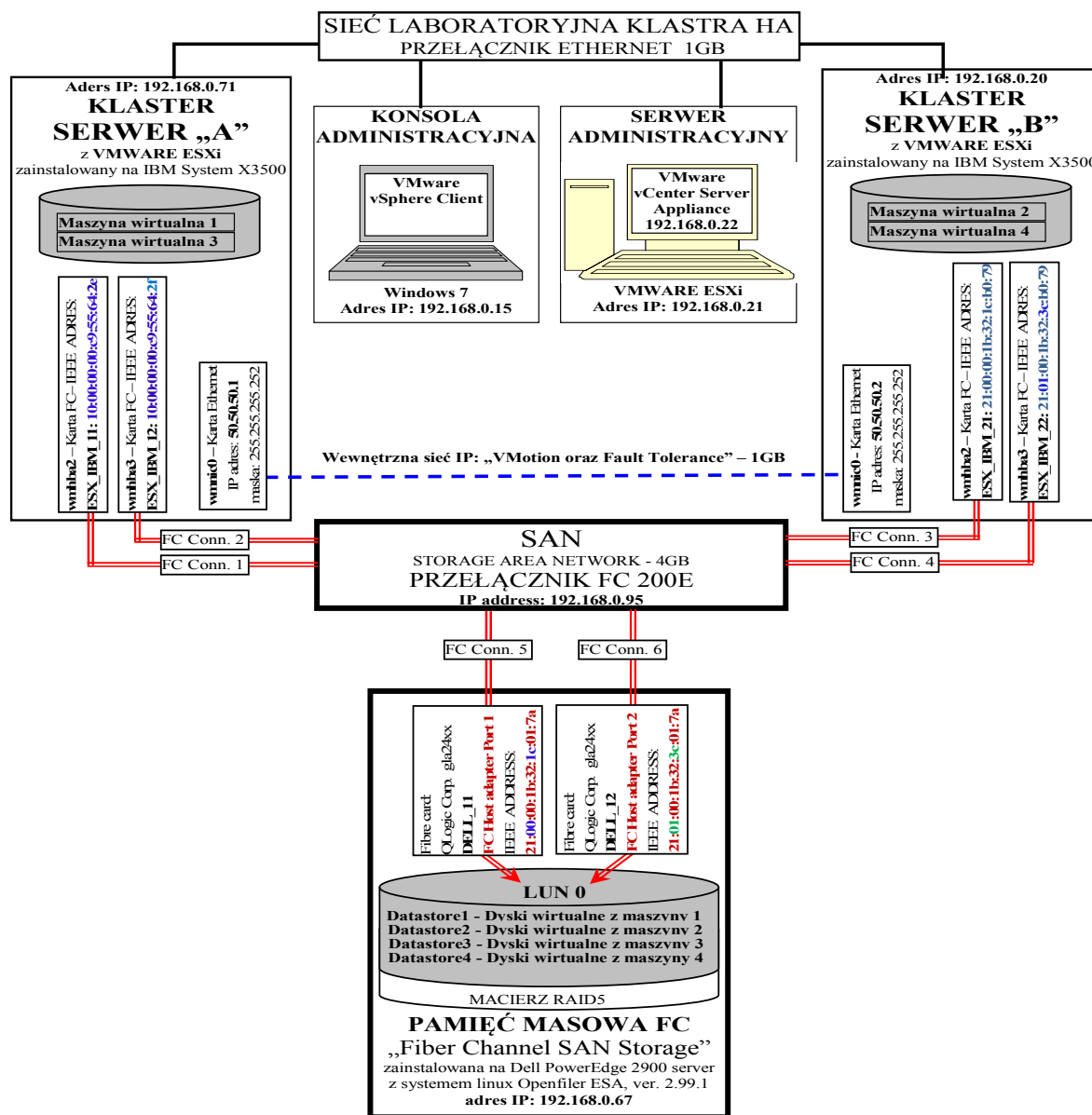
Usługa *VMware vSphere HA* dostarcza także funkcjonalności *Virtual Machine Monitoring*, monitorującej status maszyn wirtualnych w klastrze. Jeśli maszyna wirtualna nie generuje *heartbeats* w określonym czasie na pewnym hoście z powodu jego awarii, wówczas funkcjonalność *Virtual Machine Monitoring* identyfikuje ją i doprowadza do inicjacji procesu jej restartu oraz wszystkich innych, dotkniętych awarią tego hosta, maszyn wirtualnych na innych hostach. W tym celu usługa *VMware vSphere High Availability* stale monitoruje dostępność niezbędnych zasobów w klastrze, aby umożliwić restart wirtualnych maszyn na innej fizycznej maszynie w przypadku awarii pewnego hosta.

Bezpieczny restart maszyny wirtualnej możliwy jest za pomocą tzw. mechanizmu blokowania w pamięci masowej hosta *ESXi*, który pozwala na jednoczesny dostęp do tego samego pliku maszyny wirtualnej kilku hostom *ESXi*. Typowo, jak pokazano na Rys. 2., pliki maszyn wirtualnych rezydują we współdzielonej pamięci typu *VMSF (Virtual Machine File System)* zlokalizowanej w pamięci masowej typu *SAN*. *VMFS Lock Mechanism* jest mechanizmem blokowania plików maszyn wirtualnych, który ogranicza dostęp do danych typu *meta data* w określonym czasie tylko dla jednego hosta, oraz zapobiega wielu hostom na jednoczesny zapis i zapewnia ochronę danych przed uszkodzeniem.

Usługa *VMware vSphere FT* na platformie *ESXi* zapewnia ciągłą dostępność podstawowych (*primary*) maszyn wirtualnych, poprzez ich ochronę za pomocą wtórnych (*secondary*) maszyn wirtualnych, które pracują jednocześnie na innym hoście. Akcje i zdarzenia (kliknięcie myszy, naciśnięcie klawisza klawiatury) wykonywane na podstawowej maszynie wirtual-

nej, są zapisywane i następnie powtarzane na wtórnych maszynach wirtualnych, co zapewnia pozostawanie obydwu maszyn w identycznym stanie i przejęcie zadań maszynie wtórnej bez zakłóceń i utraty danych.

W niniejszym artykule przedstawiono analizę wpływu zdarzeń krytycznych na pracę klastra *VMWARE ESXi HA*, z pełną redundancją połączeń światłowodowych, współpracującego z systemem pamięci masowej typu *Fiber Channel SAN Storage* pracującej pod kontrolą systemu operacyjnego *Linux Openfiler ESA*. W celu wykonania analizy wpływu zdarzeń krytycznych na pracę klastra *VMWARE ESXi*, przygotowano środowisko badawcze w postaci laboratoryjnej infrastruktury klastrowej przedstawionej na Rys. 2.



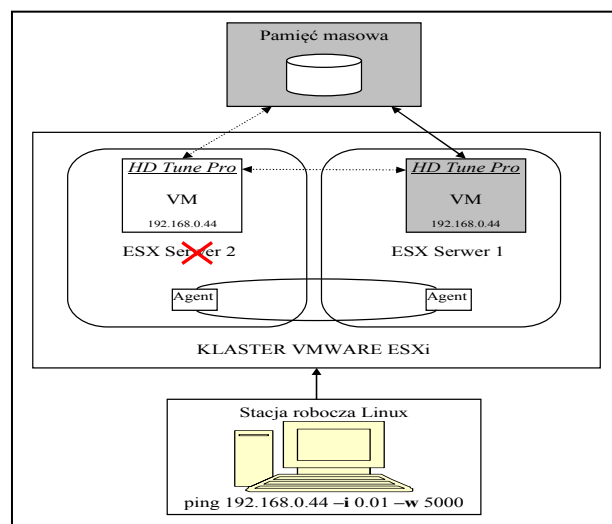
Rys. 2. Klastrowa Wysokiej dostępności VMware ESXi z pamięcią masową FC
 Fig. 2. Cluster HA VMware ESXi with "Fiber Channel SAN Storage"

Przygotowane środowisko badawcze wykorzystano do przetestowania, pod kątem ciągłości działania i oceny dostępności w sytuacjach wystąpienia zdarzeń krytycznych, fragmentu specjalistycznego szpitalnego Systemu Radiografii Pośredniej *Synapse* klasy *enterprise* firmy *FUJIFILM Medical Systems USA, Inc.*, działającego w postaci maszyny wirtualnej, używanego w Szpitalu Specjalistycznym im. Prof. E. Michałowskiego MEDHOLDING S.A., służącego do gromadzenia oraz prezentacji rentgenogramów (RTG) i tomogramów (TK).

Celem pracy była ocena dostępności tego systemu do diagnostyki medycznej poprzez pomiar czasu potrzebnego na restart tej maszyny wirtualnej na innym hoście, w sytuacjach wystąpienia zdarzeń krytycznych, powodującej awarię tej maszyny wirtualnej lub hosta *ESXi* na której została uruchomiona. Do pomiaru tego czasu zaproponowano prostą metodę polegającą na wykorzystaniu polecenia *ping* oraz programu typu „*Benchmark*” o nazwie „*HD Tune Pro*” f-my *EFD Software*, służącego do testowania dysków poprzez pomiar operacji prędkości odczytu dysków.

2. Metoda pomiaru czasu restartu maszyny wirtualnej

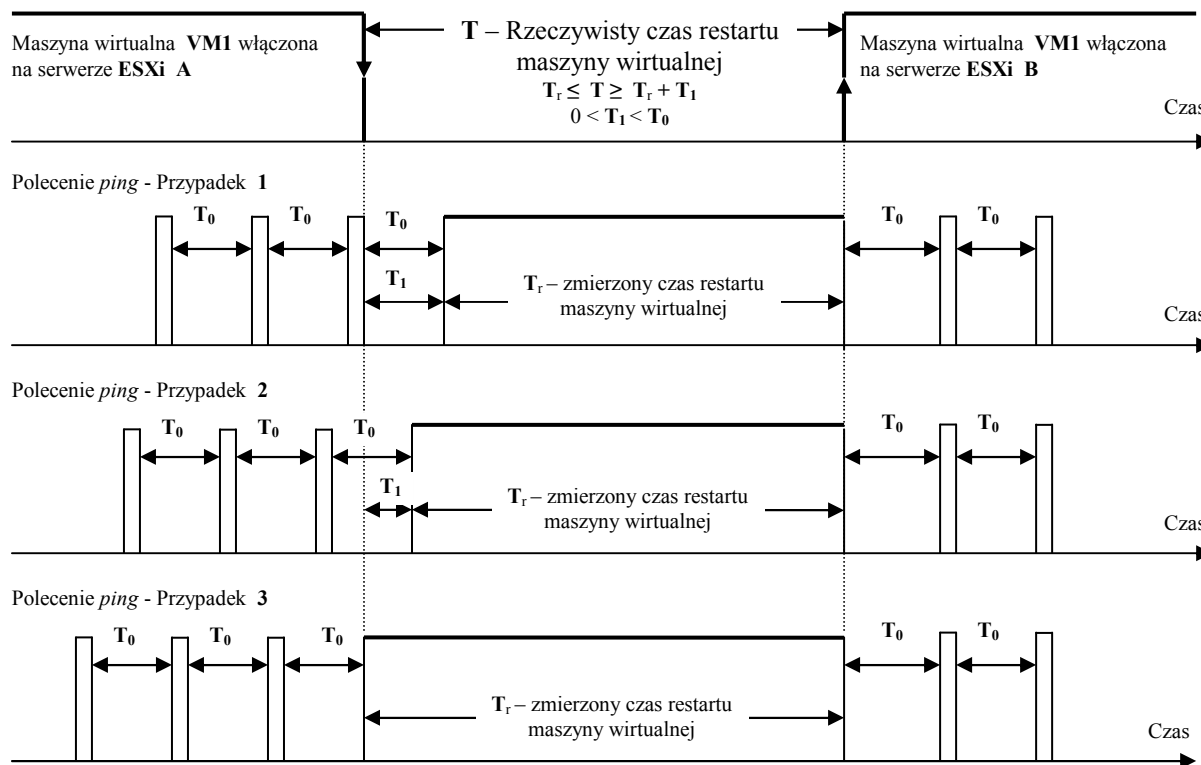
Istota zaproponowanej metody pomiaru czasu restartu maszyny wirtualnej na innym hoście w klastrze, pokazanej na Rys. 3, polega na testowaniu aktywności maszyny wirtualnej za pomocą polecenia *ping* z włączonym parametrem *-w 5000*, określającym limit czasu oczekiwania na odpowiedź w milisekundach.



Rys. 3. Metoda pomiaru czasu restartu maszyny wirtualnej za pomocą polecenia *ping*
 Fig. 3. The time measuring method to restart the virtual machine by using the *ping* command

W sytuacji wystąpienia zdarzenia krytycznego, w postaci symulowanej awarii hosta *ESXi* na którym ta maszyna wirtualna była uruchomiona, klaster dzięki funkcjonalności *FT* przełą-

cza tę maszynę na drugi host *ESXi*. Jeśli w trakcie wykonywania polecenia *ping*, w czasie oczekiwania na odpowiedź, pojawi się sygnał odpowiedzi z maszyny wirtualnej przełączanej na drugi host, wówczas wyświetlany czas odpowiedzi z polecenia *ping* będzie w przybliżeniu określał rzeczywisty czas *T* restartu maszyny wirtualnej, co pokazano na Rys. 4.



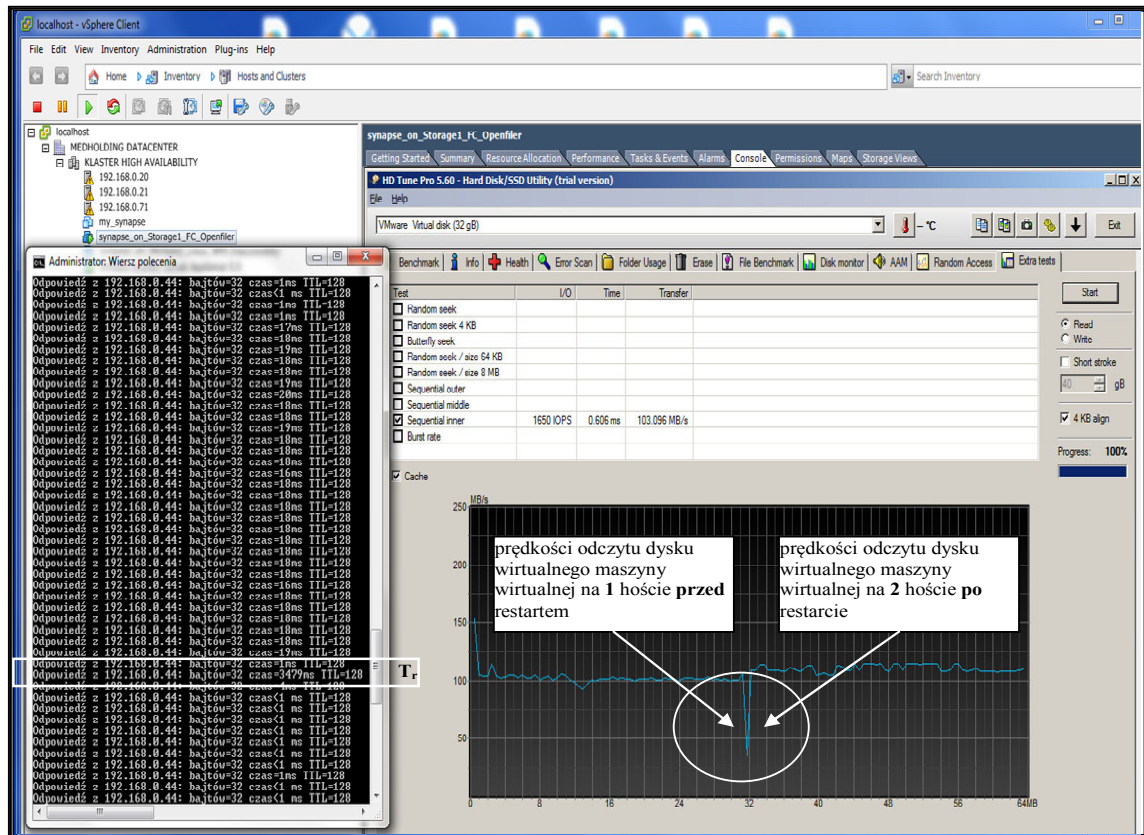
Rys. 4. Zależności czasowe podczas pomiaru czasu *T* restartu maszyny wirtualnej
 Fig. 4. Depending on the time when measuring restart time *T* of virtual machine

Jak widać z zależności $T_r \leq T \leq T_r + T_1$ oraz $0 \leq T_1 \leq T_0$, zmniejszenie czasu T_0 skutkuje zwiększeniem dokładności „podziałki przyrządu pomiarowego” i powoduje zwiększenie dokładności pomiaru czasu *T* restartu maszyny wirtualnej. Czas T_1 określa błąd pomiaru.

Symulacji zdarzenia krytycznego, w postaci awarii hosta *ESXi* na którym maszyna wirtualna była uruchomiona, dokonywano za pomocą oprogramowania administracyjnego *VMware vSphere Client ver.5.5*, przy wykorzystaniu opcji *Fault Tolerance->Test Failover*. W celu wizualizacji pomiaru czasu restartu maszyny wirtualnej użyto programu typu *Benchmark* o nazwie *HD Tune Pro* firmy *EFD Software*, pokazany na Rys. 5., z pewną ustaloną częstotliwością odczytów, możliwą do zmiany tylko poprzez zmianę parametru *Test speed/accuracy*, zmieniającego się w granicach od wartości *Accurate* do wartości *Fast*.

Metodyka pomiaru czasu restartu T_r maszyny wirtualnej była następująca: Uruchamiano program *HD Tune Pro* mierzący prędkość odczytu systemowego dysku wirtualnego maszyny wirtualnej, a następnie uruchamiano polecenie *ping 192.168.0.44 -w 5000*. Kiedy kreślony przez program *HD Tune Pro* wykres prędkości odczytu dysku systemowego osiągnie połowę

obszaru wyświetlania, wtedy symulowano awarię hosta *ESXi*, na którym uruchomiona była maszyna wirtualna, poprzez włączenie na tym gości opcji *Fault Tolerance->Test Failover*. Pomiar kończono, kiedy wykres osiągnie koniec obszaru wyświetlania.



Rys. 5. Wizualizacja metody pomiaru czasu restartu maszyny wirtualnej
Fig. 5. Visualization method of measuring the virtual machine restart time

Widoczny na tym rysunku gwałtowny spadek, w postaci ostrego piku, prędkości czytania jednego z dysków wirtualnych z maszyny wirtualnej znajdującej się w współdzielonej pamięci masowej, jest graficzną reprezentacją procesu przełączenia maszyny wirtualnej z jednego hosta *ESXi* na drugi host. Z lewej strony tego spadku pokazywane są prędkości odczytu dysku wirtualnego maszyny wirtualnej znajdującej się na pierwszym gości przed restartem, natomiast z prawej strony pokazywane są prędkości odczytu dysku wirtualnego z maszyny wirtualnej znajdującej się na drugim gości po restarcie. Szerokość podstawy tego piku jest miarą czasu restartu maszyny wirtualnej. Jej liczbową wartość (w opisywanym pomiarze - wynoszącą 3479 ms) odczytywano z okna *Administrator: Wiersz polecenia*, wyświetlającego czasy odpowiedzi z polecenia *ping* dostępnego w implementacji systemu operacyjnego *Windows7*, które wykonywane było co 1s ($T_0 = 1s$). Odpowiedzi leżące ponad tą wartością dotyczą odpowiedzi na polecenia *ping* od maszyny wirtualnej uruchomionej na pierwszym gości przed jej restartem, natomiast odpowiedzi pod tą wartością od maszyny wirtualnej znajdującej się na drugim gości po jej restarcie.

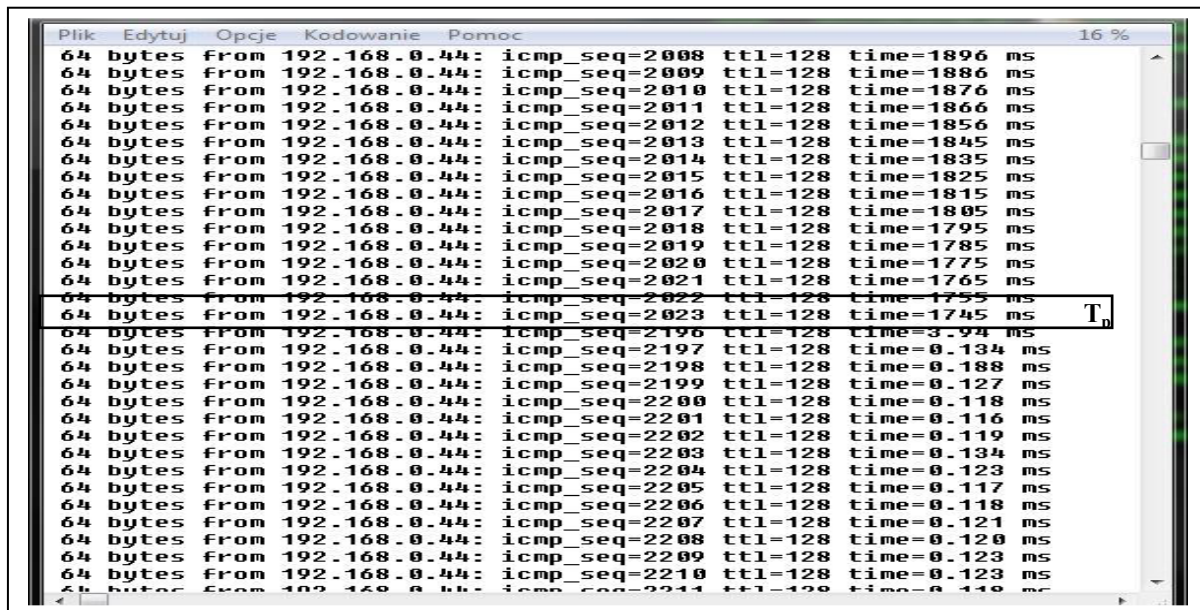
W celu zwiększenia dokładności pomiaru czasu T restartu maszyny wirtualnej należy zmniejszyć (zagęścić) „podziałki przyrządu pomiarowego”, poprzez zmniejszenie czasu T_0 . Ze względu na to, że zmniejszenie czasu T_0 jest możliwe w implementacjach polecenia *ping*, dostępnych na maszynach linuxowych, w których jest możliwość zmiany wartości parametru $-i$ określającego interwał czasu generowania kolejnych poleceń *ping* i osiągnięcia większej dokładności pomiarów, w niniejszej pracy wykorzystano polecenie *ping* z tych systemów. Dokładność „podziałki przyrządu pomiarowego” dla tej metody pomiarowej ustalono dla parametru $-i = 0.01s$, czyli $T_0 = 10$ ms. Końcowa postać polecenia *ping* miała składnię:

```
# ping 192.168.0.44 -i 0.01 -w 5000 > czas_restartu.txt
```

Czasy odpowiedzi z maszyny wirtualnej podczas jej restartu rejestrowano w plikach tekstowych *czas_restartu.txt*. Przy ww. dokładności „podziałki przyrządu pomiarowego” oraz w zależności od czasu trwania pomiarów, pliki te zawierały ok. tysiąca odczytów czasów odpowiedzi z maszyny wirtualnej, których początkowy fragment jednego z nich pokazano na Rys. 6, natomiast końcowy na Rys. 7. Na fragmencie odczytów pokazanych na Rys. 6, podobnie jak na poprzednim Rys. 5, widać nagłą zmianę wartości czasów odczytu odpowiedzi z maszyny wirtualnej, lecz w przeciwieństwie do niego, każdy następny odczyt jest mniejszy od poprzedniego, tworzące obszar przejściowy z zarejestrowanymi odczytami. Pierwszy z nich (zaznaczony na rysunku, którego wartość wynosi 4022 ms), jest czasem restartu T_r maszyny wirtualnej, ostatni zaś, pokazany na Rys. 7, jest czasem powrotu T_p z obszaru pracy przejściowej do obszaru pracy normalnej.

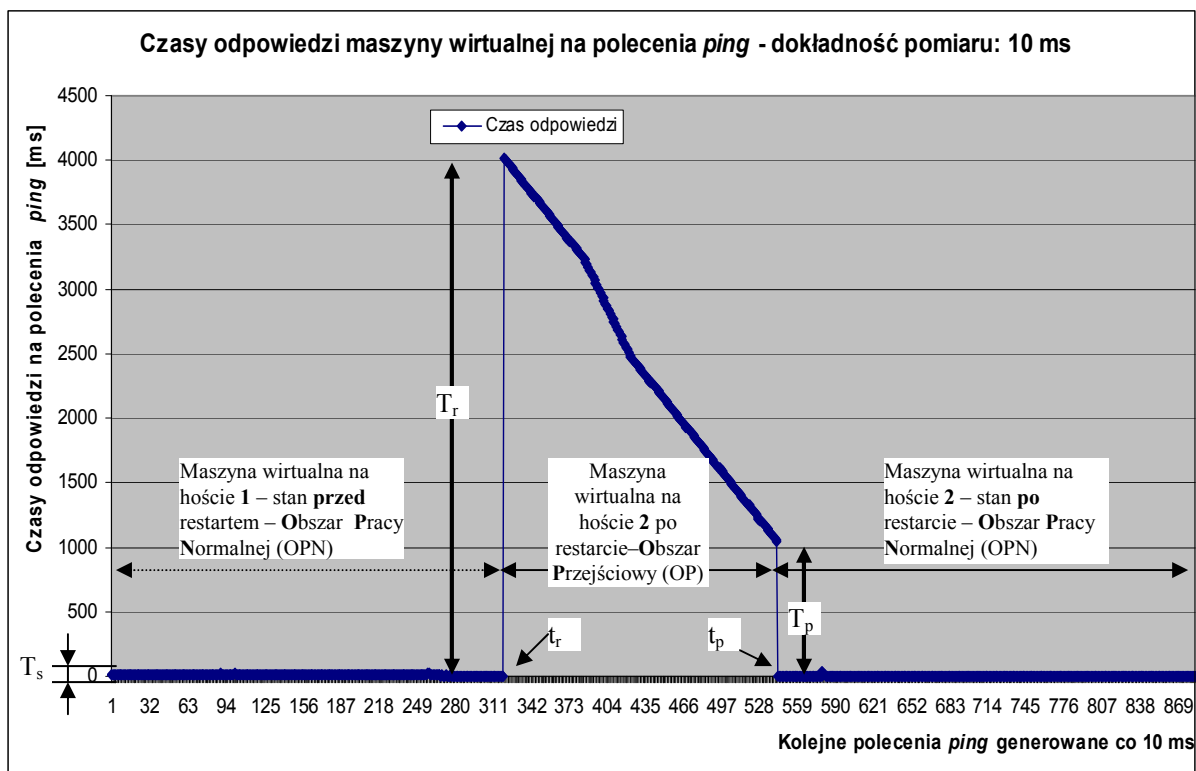
Plik	Edytuj	Opcje	Pomoc								13 %
64	bytes	from	192.168.0.44:	icmp_seq=1778	ttl=128	time=0.164	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1779	ttl=128	time=0.153	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1780	ttl=128	time=0.130	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1781	ttl=128	time=0.133	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1782	ttl=128	time=0.130	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1783	ttl=128	time=0.130	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1784	ttl=128	time=0.143	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1785	ttl=128	time=0.152	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1786	ttl=128	time=0.131	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1787	ttl=128	time=0.130	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1788	ttl=128	time=0.131	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1789	ttl=128	time=0.132	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1790	ttl=128	time=0.146	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1791	ttl=128	time=0.130	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1792	ttl=128	time=0.149	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1793	ttl=128	time=0.169	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1794	ttl=128	time=0.126	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1795	ttl=128	time=0.118	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1796	ttl=128	time=0.145	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1797	ttl=128	time=4022	ms				T_r
64	bytes	from	192.168.0.44:	icmp_seq=1798	ttl=128	time=4002	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1799	ttl=128	time=4002	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1800	ttl=128	time=3992	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1801	ttl=128	time=3982	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1802	ttl=128	time=3972	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1803	ttl=128	time=3962	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1804	ttl=128	time=3952	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1805	ttl=128	time=3942	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1806	ttl=128	time=3931	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1807	ttl=128	time=3921	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1808	ttl=128	time=3911	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1809	ttl=128	time=3901	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1810	ttl=128	time=3891	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1811	ttl=128	time=3881	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1812	ttl=128	time=3871	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1813	ttl=128	time=3861	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1814	ttl=128	time=3851	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1815	ttl=128	time=3841	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1816	ttl=128	time=3831	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1817	ttl=128	time=3821	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1818	ttl=128	time=3811	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1819	ttl=128	time=3800	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1820	ttl=128	time=3790	ms				
64	bytes	from	192.168.0.44:	icmp_seq=1821	ttl=128	time=3780	ms				

Rys. 6. Czas restartu T_r maszyny wirtualnej na polecenia *ping* podczas jej restartu
Fig. 6. Response time T_r of the virtual machine to *ping* command during her restart



Rys. 7. Czas powrotu T_p z obszaru pracy przejściowej do obszaru pracy normalnej
 Fig. 7. The T_p time return from the transitional work area to the normal work area

Po zaimportowaniu całej zawartości tego pliku do arkusza kalkulacyjnego otrzymano wykres czasów odpowiedzi maszyny wirtualnej na polecenia *ping*, przedstawiony na Rys. 8, dla ustalonej dokładności „podziałki przyrządu pomiarowego”.



Rys. 8. Wykres czasów odpowiedzi maszyny wirtualnej na polecenia *ping* podczas jej restartu
 Fig. 8. Graph of response times of the virtual machine to *ping* command during her restart

Widać na nim grupę odczytów (ok. 220) czasów odpowiedzi maszyny wirtualnej na polecenia *ping* podczas jej restartu, zawartych w obszarze przejściowym (OP) pomiędzy chwilami t_r i t_p , w których następowały zdarzenia polegające na nagłych zmianach wartości czasów odczytów. Odczyty pomiędzy chwilami t_r i t_p rozpoczynały się nagłym skokiem czasu odpowiedzi od wartości średniej z obszaru pracy normalnej (OPN) do wartości maksymalnej T_r , następnie stopniowo malały do pewnej wartości minimalnej T_p , i po nagłym skoku powracały do wartości średnich czasów T_s odpowiedzi na polecenie *ping* z obszaru pracy normalnej.

W celu oszacowania wskaźnika dokładności pomiaru czasu restartu T maszyny wirtualnej pomiędzy hostami, w postaci odchylenia standardowego wartości średniej, czyli średniego błędu kwadratowego S_A [36], wykonano serię $n=15$ pomiarów czasu restartu T_r maszyny wirtualnej. Średni błąd kwadratowy S_A obliczano za pomocą wzoru $S_A = \sqrt{\sum \varepsilon_i^2 / n(n-1)}$, gdzie:

n – jest liczbą pomiarów, na podstawie których wyznaczono średnią arytmetyczną t_{sr}

$\varepsilon_i = t_i - t_{sr}$ - są odstępstwami pomiaru czasu restartu T_r od średniej arytmetycznej $t_{sr} = (\sum T_{r_i})/n$

3. Instalacja i konfiguracja środowiska badawczego

W związku z nakreślonym celem, zbudowano środowisko badawcze w postaci klastra laboratoryjnego [37,38,39], w zakresie podstawowej zdolności przyłączeniowej, odporności na uszkodzenia *FT* oraz pełnej redundancji połączeń światłowodowych pomiędzy klastrem a pamięcią masową typu *Fiber Channel SAN Storage*, pracującej pod kontrolą systemu operacyjnego *Linux Openfiler ESA*.

Klaster składał się z dwóch serwerów *IBM System X3500*, wraz ze współpracującą z nim współdzieloną pamięcią masową pracującą na maszynie *DELL PowerEdge 2900* pod kontrolą systemu operacyjnego *Openfiler ESA ver. 2.99.1*, na których zainstalowano oprogramowanie serwerowe *VMWARE ESXi ver. 5.5*.

Zarządzanie klastrem sprawowano za pomocą pakietu oprogramowania *VMware vCenter Operations Management Suite 5.8*, tj. *VMware vCenter Server Appliance ver. 5.5*, które zainstalowano na oddzielnym serwerze zarządzającym z oprogramowaniem serwerowym *Vmware ESXi ver. 5.5*.

Oprogramowanie administracyjne, w postaci pakietu *VMware vSphere Client ver.5.5*, zarządzające całym klastrem zainstalowano na kolejnym komputerze. W celu instalacji oprogramowania zarządzającego *VMware vCenter Server Appliance*, korzystano z kreatora tego oprogramowania z poziomu klienckiego programu administracyjnego *vSphere Client*, in-

staującego plik *VMwarevCenter-Server-Appliance-5.5.0.10000-1624811_OVF10.ovf*, zawierającego oprogramowanie zarządzające.

W celu uzyskania dostępu do konsoli administracyjnej *VMware vCenter Server Appliance*, na serwerze zarządzającym uruchomiono maszynę wirtualną dostępną pod nazwą *VMware vCenter Server Appliance*. Dostęp do uruchomionej konsoli maszyny wirtualnej *VMware vCenter Server Appliance* uzyskiwano za pomocą dowolnej przeglądarki internetowej. Dla zapewnienia funkcjonalności związanej z przenoszeniem wirtualnych maszyn pomiędzy hostami i klastrami (*VMotion*) oraz odporności na awarie *FT*, utworzono dla nich wewnętrzną sieć komunikacyjną pod nazwą *VMotion* oraz *Fault Tolerance*.

3.1. Funkcjonalność *Fault Tolerance* w klastrze *ESXi HA*

W związku z celem badawczym, w zbudowanym klastrze laboratoryjnym korzystano z funkcjonalności odporności na uszkodzenia *FT*. Zanim funkcjonalność *Fault Tolerance* może być wykorzystana, należy ją uprzednio włączyć. Tę funkcjonalność, dostępną pod nazwą *Turn on Fault Tolerance*, włączano dla każdej badanej maszyny wirtualnej. Przykładowo, po włączeniu tej funkcjonalności dla uruchomionej testowej wirtualnej maszyny działającej na serwerze *ESXi* (192.168.0.20), status tej maszyny został zmieniony i maszyna ta została wprowadzona w stan *Protected*. Równocześnie klaster utworzył dla niej kopię we wtórnej lokalizacji *Secondary Location* na serwerze *ESXi* (192.168.0.71). Jeśli teraz serwer *ESXi* (IP=192.168.0.20) lub też uruchomiona na nim testowa maszyna wirtualna nagle ulegnie awarii, wówczas włączona funkcjonalność *Fault Tolerance* zapewnia, że jej kopia zlokalizowana we wtórnej lokalizacji *Secondary Location* na serwerze *ESXi* (IP=192.168.0.71) przechodzi w stan aktywności (*on line*). Teraz ta wirtualna maszyna pracuje na serwerze 192.168.0.71 w trybie *Protected* wraz z swoją kopią w wtórnej lokalizacji umiejscowionej na serwerze 192.168.0.20. Należy zauważyć, że wirtualny dysk twardy z tej wirtualnej maszyny pozostał niezmienny i nadal rezyduje w pamięci masowej.

4. Analiza wpływu sytuacji krytycznych na pracę klastra *ESXi HA*

4.1. Redundancja kanału *FC (Fibre Channel)*

Jak pokazano wcześniej na Rys. 2., w związku z budową klastra *VMware ESXi* wysokiej dostępności, zastosowano redundantne fizyczne połączenia światłowodowe *FC*, od „*FC conn. 1*” do „*FC conn. 6*”. Połączenia redundantne „*FC conn. 1*” i „*FC conn. 2*” łączą „Serwer A”

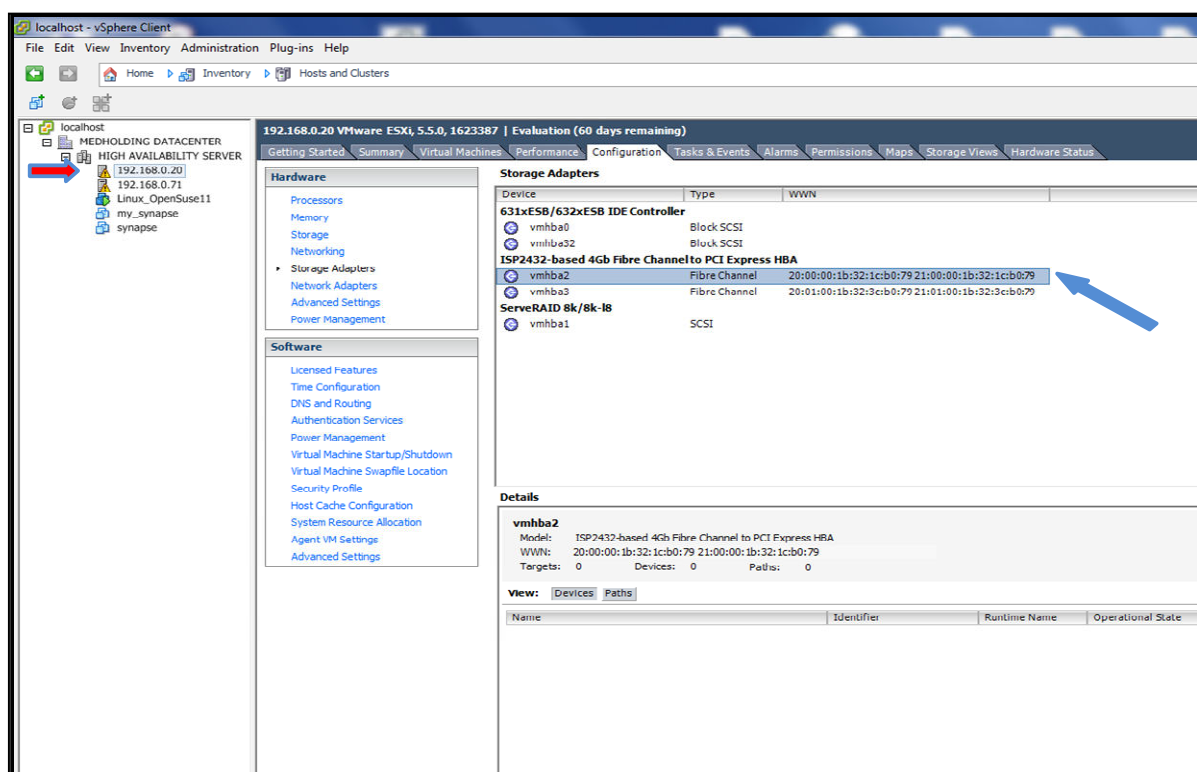
z przełącznikiem *SAN FC 200*, natomiast połączenia „*FC conn. 2*” i „*FC conn. 3*” łączą z nim „*Serwer B*”. Z kolei połączenia „*FC conn. 5*” i „*FC conn. 6*” łączą przełącznik z współdzieloną pamięcią masową *FC Fiber Channel San Storage*.

Na kolejnych rysunkach przedstawiono wystąpienie awaryjnych sytuacji, ilustrujących zachowanie klastra w przypadku uszkodzenia któregoś z połączeń redundantnych.

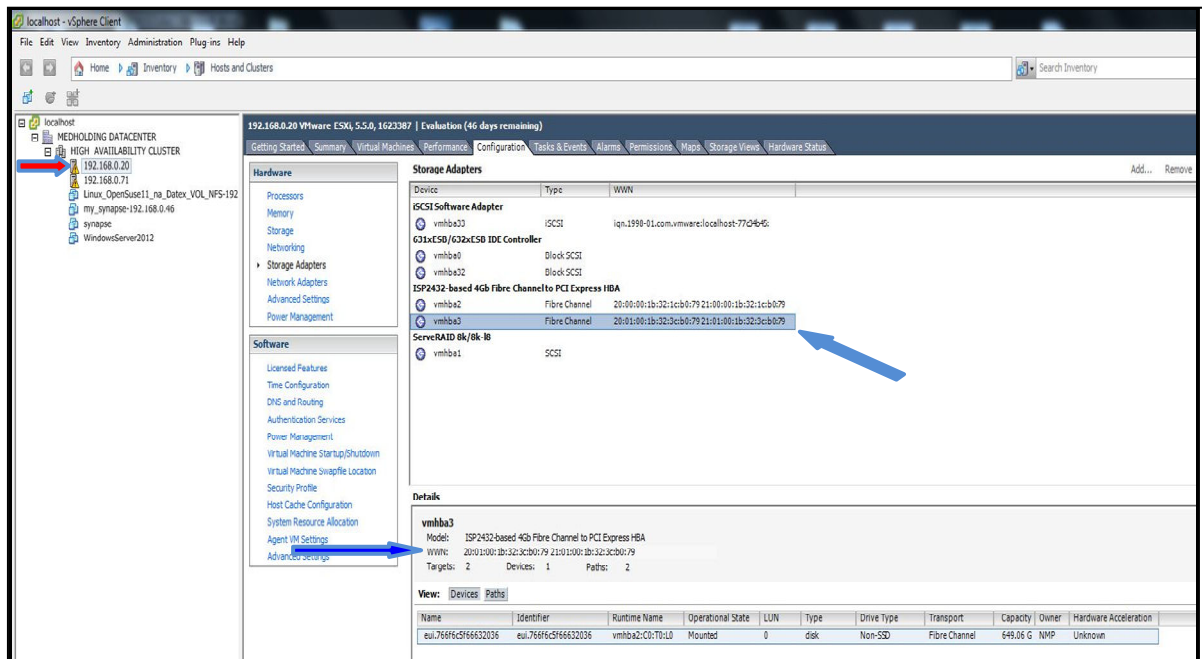
1. Przerwa w połączeniu „*FC Conn. 3*”

W konsekwencji przerwy w połączeniu „*FC Conn. 3*”, spowodowanej np. przerwaniem włókna światłowodowego, uszkodzeniem złączy światłowodowych, awarią karty światłowodowej itp., dla serwera *ESXi* (192.168.0.20) i adaptera pamięci masowej *ISP2432-based 4GB Fibre Channel to PCI Express HBA*, urządzenie kanału światłowodowego *vmhba2* nie pracuje.

Jak pokazano na Rys. 6, urządzenie *vmhba2* nie wskazuje już na żadną pamięć masową *FC*, jednak urządzenie *vmhba3*, co pokazano na Rys. 7, ciągle wskazuje na tę samą pamięć masową pod nazwą *eui.766f6c5f66632036*, o całkowitej pojemności 649.06 GB. W rezultacie, każda wirtualna maszyna zainstalowana na tym serwerze ciągle jest połączona z pamięcią masową *FC*.



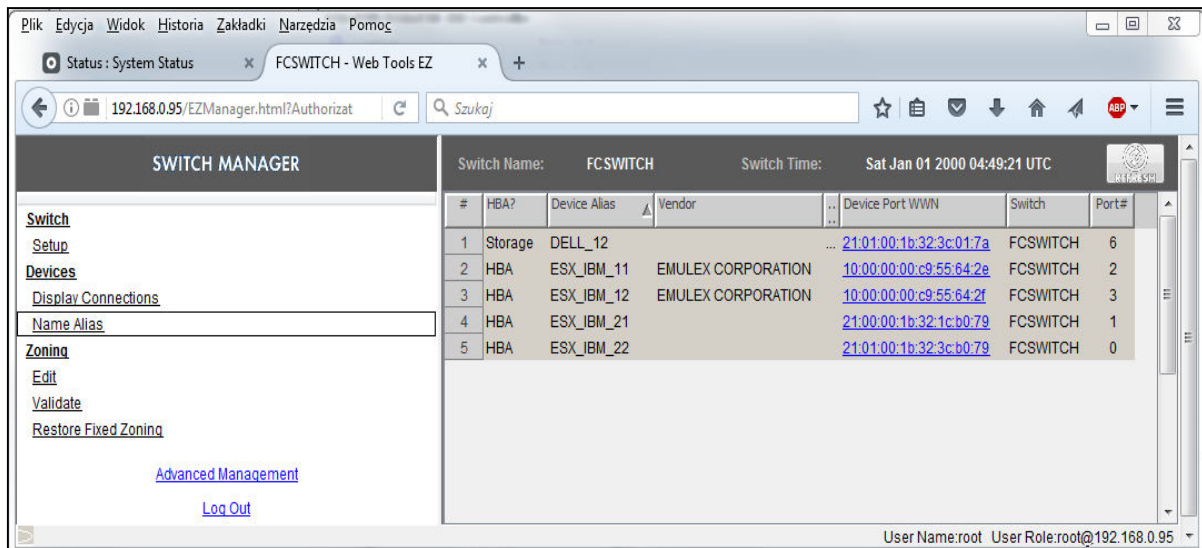
Rys. 6. Brak wskazań urządzenia *vmhba2* na pamięć masową *FC*
Fig.6. There is no indication the device *vmhba2* on the *FC* storage



Rys. 7. Połączenie maszyn wirtualnych, poprzez urządzenie *vmhba3*, do pamięci masowej *eui.766f6c5f66632036*
 Fig.7. Connection of the virtual machines, through the *vmhba3* device, to the *eui.766f6c5f66632036* mass storage

2. Przerwa w połączeniu "FC Conn. 5"

Jeśli połączenie "FC Conn. 5" uległo uszkodzeniu, wówczas w konsekwencji urządzenie o aliasie *DELL_11* nie pracuje. Jak pokazano na Rys. 8, nie ma go na liście urządzeń.



Rys. 8. Urządzenie *DELL_11* nie pracuje; nie ma go na liście dostępnych urządzeń
 Fig. 8. The device of *DELL_11* does not work; It is not in the list of available devices

Dzięki pełnej redundancji kanału światłowodowego, wszystkie serwery *ESXi* w klastrze, jako całość, nadal działają. Dla serwera *ESXi* 192.168.0.20, urządzenia kanału światłowodowego

vmhba2 oraz *vmhba3* są w stanie ciągłej pracy. Każde urządzenie kanału światłowodowego ciągle wskazuje na tę samą pamięć masową FC pod nazwą *eui.766f6c5f66632036*, o pojemności 649.06 GB. W konsekwencji, każda z włączonych wirtualnych maszyn w klastrze nadal pracuje. Po przywróceniu połączenia “*FC Conn. 5*”, a następnie przerwaniu połączenia “*FC Conn. 6*”, klaster nadal połączony jest z pamięcią masową. Nie zaobserwowano przerwy w komunikacji pomiędzy maszyną wirtualną, a pamięcią masową.

5. Ocena dostępności Systemu Radiografii Pośredniej *Synapse*

Jak wspomniano, celem pracy była ocena dostępności, pod kątem ciągłości działania, medycznego Systemu Radiografii Pośredniej *Synapse* klasy *enterprise* firmy *FUJIFILM Medical Systems USA, Inc.*, działającego w postaci maszyny wirtualnej, używanego w Szpitalu Specjalistycznym im. Prof. E. Michałowskiego MEDHOLDING S.A., służącego do gromadzenia oraz prezentacji rentgenogramów (RTG) i tomogramów (TK). System ten został zainstalowany w postaci maszyny wirtualnej działającej na platformie *VMWARE ESXi*.

Ocenę jego dostępności dokonano poprzez pomiar czasu potrzebnego na restart tej maszyny wirtualnej w badawczym środowisku klastrowym na innym hoście, podczas wystąpienia zdarzeń krytycznych, powodującej awarię tej maszyny wirtualnej lub hosta *ESXi*, na której została uruchomiona.

5.1. Instalacja systemu *Synapse* w środowisku badawczym

Na potrzeby badań, wykonano instalację kopii fragmentu tego systemu w sieci laboratoryjnej, składającej się z pamięci masowej typu *Fiber Channel SAN Storage*, pracującej pod kontrolą systemu operacyjnego *Linux Openfiler ESA* i współpracującej z klastrem *VMWARE ESXi*. Instalację przeprowadzono zgodnie z opisem zawartym w pracy [37].

5.2. Pomiary czasu restartu maszyny wirtualnej z systemem *Synapse*

W celu oceny dostępności systemu *Synapse* do diagnostyki medycznej wykonano, za pomocą zaproponowanej metody pomiarowej, serię 15 pomiarów czasu restartu T_r tej maszyny wirtualnej na innym hoście, w sytuacjach wystąpienia zdarzeń krytycznych, powodującej awarię tej maszyny wirtualnej lub hosta *ESXi* na której została uruchomiona.

Do pomiarów czasu restartu maszyny wirtualnej z systemem *Synapse* wykorzystano polecenie *ping* z systemów linuksowych. Dokładność „podziałki przyrzędu pomiarowego” dla tej me-

tody pomiarowej ustalono dla parametru $-i = 0.01s$, czyli $T_0 = 10$ ms. Końcowa postać polecenia *ping* miała składnię:

```
# ping 192.168.0.44 -i 0.01 -w 5000 > czas_restartu.txt
```

W wyniku dokonanych pomiarów otrzymano 15 plików tekstowych z zarejestrowanymi czasami odpowiedzi z maszyny wirtualnej podczas jej restartu podobnymi do tych, które wcześniej pokazano na Rys. 6. Dla każdego z tych plików, z zapisanych w nim czasów, odczytywano czas odpowiedzi T_r . Wyniki obliczeń wartości średniej czasu restartu T_R oraz dokładności jego oszacowania w postaci średniego błędu kwadratowego S_A (odchylenie standardowe) podano w Tab. 1.

Ilość pomiarów: 15		Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu restartu maszyny wirtualnej	
Nr pomiaru	Czas restartu t_i [ms]	Błąd $\varepsilon_i = t_i - t_{sr}$ [ms]	ε_i^2 [ms] ²
1	4005	-17,00	289,00
2	4020	-2,00	4,00
3	4019	-3,00	9,00
4	4021	-1,00	1,00
5	4045	23,00	529,00
6	4011	-11,00	121,00
7	3992	-30,00	900,00
8	4039	17,00	289,00
9	4040	18,00	324,00
10	4001	-21,00	441,00
11	4038	16,00	256,00
12	4027	5,00	25,00
13	4007	-15,00	225,00
14	4052	30,00	900,00
15	4013	-9,00	81,00
$t_{sr} =$	4022		$\sum \varepsilon_i^2 =$ 4394,00
$S_A = [\sum \varepsilon_i^2 / n(n-1)]^{1/2} =$		5	ms
Czas restartu $T_R =$		4022	± 5 ms
Błąd względny $\delta =$		0,1	%

Tab. 1. Obliczenie czasu T_R podczas restartu maszyny wirtualnej
Tab. 1. Calculating the T_R time during the restart of the virtual machine

Wyniki obliczeń przedstawiają się następująco:

1. Czas restartu maszyny wirtualnej na innym hoście klastra: $T_R = 4022 \pm 5$ ms,
2. Średni błąd kwadratowy: $S_A = 5$ ms, jest mniejszy od dokładności „podziałki przyrządu pomiarowego” tej metody wynoszącej $T_0 = 10$ ms,
3. Błąd maksymalny pomiaru: $\varepsilon_{max} = 30$ ms.

5.3. Restart maszyny wirtualnej systemu „Synapse”

Przy zastosowaniu zaproponowanej metody pomiarowej nie zarejestrowano, podczas restartu maszyny wirtualnej systemu *Synapse* w sytuacjach wystąpień zdarzeń krytycznych, spadku do zera prędkości odczytu dysku systemowego tej maszyny wirtualnej, zarejestrowanej przez wizualizacyjną jego część w postaci programu *HD Tune Pro*, co wskazywałoby, że albo:

1. całkowity czas fizycznego restartu maszyny wirtualnej z jednego hosta na drugi jest krótszy niż odstęp czasu wynikający z częstości pomiarów tym programem prędkości odczytu dysku wirtualnego, skutkującego tym, że program ten nie jest "świadomy" tego, że mierzy prędkość odczytu dysku wirtualnego z maszyny wirtualnej, która w międzyczasie została uruchomiona na drugim hoście,
2. klastr generuje odpowiedź na polecenie *ping* w każdej chwili czasowej, tzn. odpowiada na niego maszyna wirtualna znajdująca się albo Hoście 1 albo na Hoście 2.

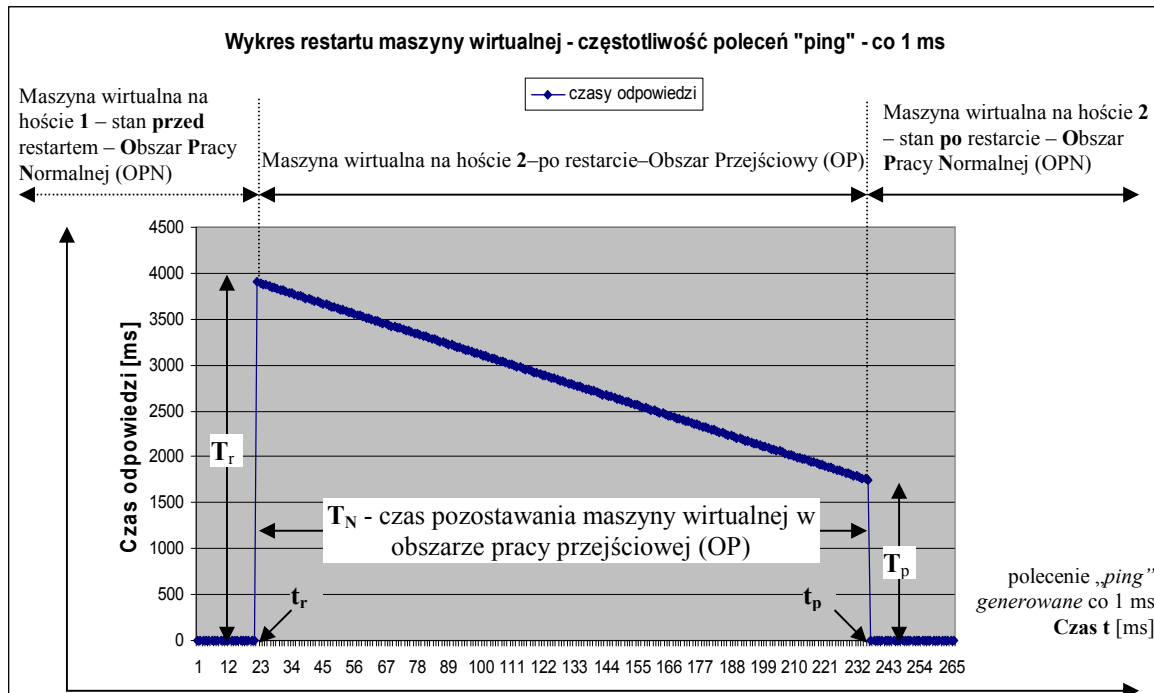
W celu wyjaśnienia tego problemu, przeprowadzono zatem badania czasu restartu maszyny wirtualnej dla jeszcze większej dokładności „podziałki przyrządu pomiarowego”, tj. $T_0=1$ ms, wykonując poniższe polecenie *ping* o składni:

```
# ping 192.168.0.44 -i 0.001 -w 5000 > czas_restartu.txt
```

Czasy odpowiedzi z maszyny wirtualnej podczas jej restartu rejestrowano w pliku tekstowych *czas_restartu.txt*. Przy tej dokładności, plik ten zawierał kilkanaście tysięcy odczytów czasów odpowiedzi z maszyny wirtualnej na poziomie średnich wartości czasów T_s odpowiedzi z obszaru pracy normalnej (OPN), w tym ok. 230 odczytów ulokowanych w obszarze przejściowym (OP) pomiędzy chwilami t_r i t_p , w których następowały zdarzenia polegające na nagłych zmianach wartości czasów odczytów.

Odczyty pomiędzy chwilami t_r i t_p rozpoczynały się nagłym skokiem czasu odpowiedzi maszyny wirtualnej na hoście 1 od wartości średniej z obszaru pracy normalnej (OPN) sprzed restartu do wartości maksymalnej T_r , następnie stopniowo malały do pewnej wartości minimalnej T_p , i po nagłym skoku powracały do wartości średnich czasów T_s odpowiedzi z obszaru pracy normalnej maszyny wirtualnej na drugim hoście.

Po zaimportowaniu do arkusza kalkulacyjnego fragmentu jego zawartości, tj. zawierającego wszystkie odczyty z obszaru przejściowego (OP) pomiędzy chwilami t_r i t_p oraz kilkadziesiąt odczytów z obszaru pracy normalnej (OPN) sprzed obszaru przejściowego i po nim, otrzymano wykres czasów odpowiedzi maszyny wirtualnej na polecenia *ping*, przedstawiony na Rys. 9.

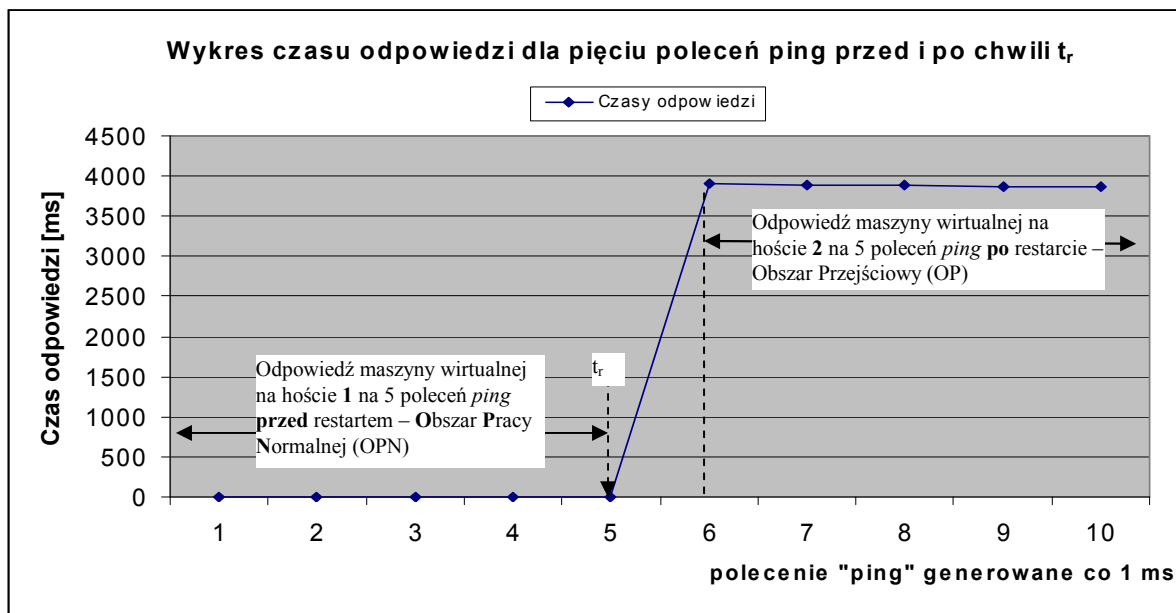


Rys. 9. Wykres restartu maszyny wirtualnej w funkcji czasu
Fig. 9. Virtual machine restart graph as a time function

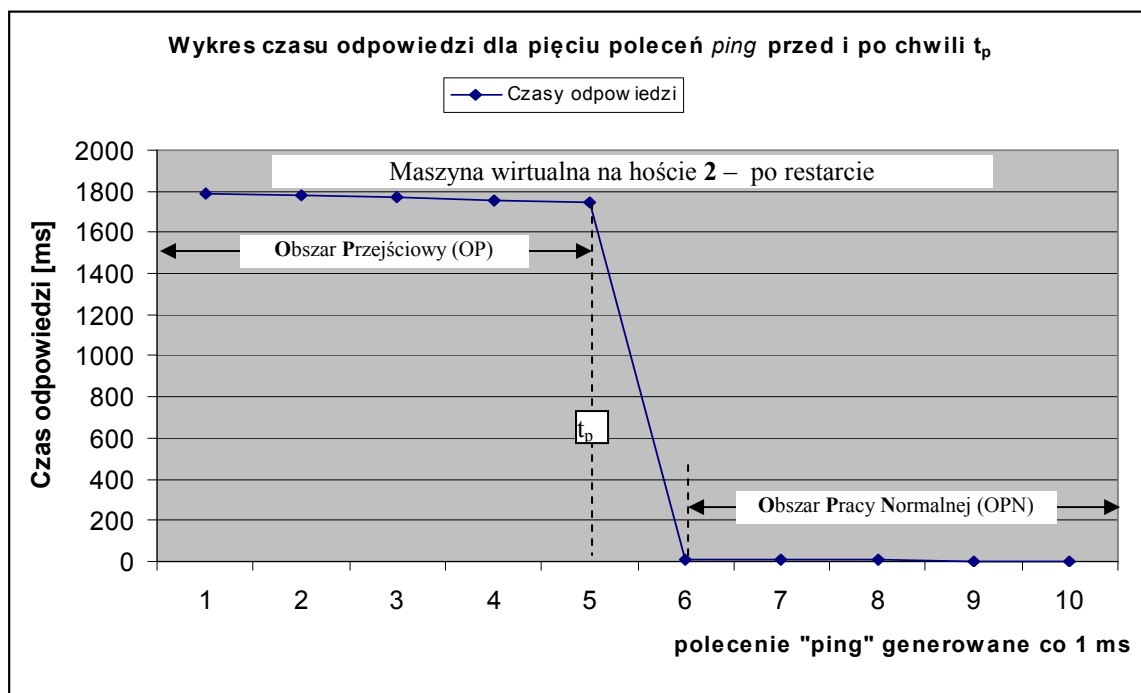
Z przedstawionego Rys. 9 widać, że odczyty czasów odpowiedzi z maszyny wirtualnej w obszarze przejściowym układały się pomiędzy chwilami czasowymi t_r oraz t_p wzdłuż prostej o ujemnym współczynniku kierunkowym.

Na tym rysunku, w chwili czasowej $t_r = 21$ ms (w 21 poleceniu *ping*) rozpoczynał się nagły skok czasu odpowiedzi od wartości średniej z obszaru pracy normalnej (OPN) do wartości maksymalnej T_r , natomiast chwili czasowej $t_p = 235$ ms (w 235 poleceniu *ping*) następował nagły skok czasu odpowiedzi od wartości T_p do średnich wartości czasów T_s odpowiedzi z obszaru pracy normalnej (OPN).

W celu lepszego zobrazowania czasów odpowiedzi maszyny wirtualnej w bezpośredniej bliskości chwil czasowych t_r oraz t_p , na Rys. 10 pokazano wykres czasu odpowiedzi dla pięciu poleceń *ping* przed i po chwili t_r , natomiast wykres czasu odpowiedzi dla pięciu poleceń *ping* przed i po chwili t_p pokazano na Rys. 11. W chwilach czasowych t_r i t_p następował nagły skok czasu odczytu odpowiedzi, który odbywał się w czasie wygenerowania kolejnego polecenia *ping*.



Rys. 10. Wykres czasu odpowiedzi dla pięciu poleceń ping przed i po chwili t_r
 Fig. 10. Graph of the response time for five orders ping before and after a moment t_r



Rys. 11. Wykres czasu odpowiedzi dla pięciu poleceń ping przed i po chwili t_p
 Fig. 11. Graph of the response time for five orders ping before and after a moment t_p

Z Rys. 10 i Rys. 11 widać, że klastr generuje odpowiedź na każde polecenie ping, tzn. odpowiada na nie maszyna wirtualna znajdująca się na Hoście 1 albo na Hoście 2.

5.3.1. *Badanie związku pomiędzy parametrem "VM Restart Priority", a czasem restartu maszyny wirtualnej systemu „Synapse”*

Domyślne ustawienia dla maszyn wirtualnych, kontrolujące porządek w którym restartowane są wirtualne maszyny oraz sposób w jaki usługa *VMware HA* reaguje w przypadku utraty połączeń sieciowych z innymi hostami, ustalane są za pomocą parametru *VM Restart Priority*.

Ustawienia te stosują się do wszystkich maszyn wirtualnych w klastrze w przypadku wystąpienia zdarzeń krytycznych, tj. jeśli: (i) host jest izolowany od sieci; tzn. kiedy host ciągle działa, ale nie może komunikować się z innymi hostami w klastrze, (ii) host się zawiesił, (iii) host jest całkowicie wyłączony z powodu awarii sprzętu. Możliwa jest również konfiguracja wyjątków dla określonych maszyn wirtualnych.

Parametr *VM Restart Priority* determinuje relatywny i sekwencyjny porządek restartu maszyn wirtualnych, po wystąpieniu zdarzenia krytycznego, na nowych hostach; najpierw następuje restart maszyn wirtualnych o wyższym priorytecie, a następnie o priorytecie niższym. Parametr *VM Restart Priority* przyjmuje jedną z czterech wartości: (i) *Disabled*, (ii) *Low*, (iii) *Medium*, (iiii) *High*. Wartość *Disabled* jest równoznaczna z tym, że funkcjonalność *VMware HA* jest wyłączona dla tej maszyny wirtualnej, co oznacza, że ta maszyna nie jest restartowana na innych hostach *ESXi*, jeśli host, na którym ona jest uruchomiona, ulegnie awarii. Jest ona nadal monitorowana, co oznacza, że jeśli przestanie funkcjonować, wówczas restartowana jest na tym samym hoście.

VMware rekomenduje przypisanie wyższego priorytetu restartu dla ważniejszych maszyn wirtualnych. Priorytet *High* zaleca się dla serwerów bazodanowych, *Medium* dla serwerów aplikacji, *Low* dla pozostałych serwerów.

5.3.1.1. *Badanie związku pomiędzy VM Restart Priority, a czasem restartu T_R*

W niniejszej pracy postanowiono zbadać, czy istnieje jakiś związek pomiędzy parametrem *VM Restart Priority* z czasem restartu T_R maszyny wirtualnej na innym hoście, w przypadku awarii hosta, na którym była uruchomiona. W tym celu, dla wcześniej ustalonej dokładności „podziałki przyrządu pomiarowego”, tj. 1 ms, wykonano serię $n=15$ pomiarów czasu restartu T_R maszyny wirtualnej dla trzech wartości parametru *VM Restart Priority*, tj.: *High*, *Medium*, *Low*. Wyniki pokazano w Tab. 2.

1. Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu T_R restartu maszyny wirtualnej												
VM Restart Priority - High				VM Restart Priority - Medium				VM Restart Priority - Low				
Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu restartu maszyny wirtualnej			Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu restartu maszyny wirtualnej			Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu restartu maszyny wirtualnej			
Nr pomiaru	Czas restartu t_i [ms]	Błąd $\delta_i = t_i - t_{sr}$ [ms]	δ_i^2 [ms] ²	Nr pomiaru	Czas restartu t_i [ms]	Błąd $\delta_i = t_i - t_{sr}$ [ms]	δ_i^2 [ms] ²	Nr pomiaru	Czas restartu t_i [ms]	Błąd $\delta_i = t_i - t_{sr}$ [ms]	δ_i^2 [ms] ²	
1	4022,00	-11,27	126,94	1	4014,00	-8,60	73,96	1	4016,00	-9,73	94,74	
2	4044,00	10,73	115,20	2	4031,00	8,40	70,56	2	3946,00	-79,73	6357,40	
3	4054,00	20,73	429,87	3	4047,00	24,40	595,36	3	4030,00	4,27	18,20	
4	4031,00	-2,27	5,14	4	4038,00	15,40	237,16	4	4046,00	20,27	410,74	
5	4051,00	17,73	314,47	5	3995,00	-27,60	761,76	5	4034,00	8,27	68,34	
6	3983,00	-50,27	2526,74	6	3986,00	-36,60	1339,56	6	4049,00	23,27	541,34	
7	4035,00	1,73	3,00	7	4053,00	30,40	924,16	7	3986,00	-39,73	1578,74	
8	4027,00	-6,27	39,27	8	4039,00	16,40	268,96	8	4052,00	26,27	689,94	
9	4007,00	-26,27	689,94	9	4036,00	13,40	179,56	9	4062,00	36,27	1315,27	
10	4052,00	18,73	350,94	10	4033,00	10,40	108,16	10	4009,00	-16,73	280,00	
11	4013,00	-20,27	410,74	11	4040,00	17,40	302,76	11	4035,00	9,27	85,87	
12	4049,00	15,73	247,54	12	4042,00	19,40	376,36	12	3998,00	-27,73	769,14	
13	4030,00	-3,27	10,67	13	3991,00	-31,60	998,56	13	4048,00	22,27	495,80	
14	4052,00	18,73	350,94	14	3995,00	-27,60	761,76	14	4057,00	31,27	977,60	
15	4049,00	15,73	247,54	15	3999,00	-23,60	556,96	15	4018,00	-7,73	59,80	
$t_{sr} = 4033,27$			$\sum \delta_i^2 = 5868,93$	$t_{sr} = 4022,60$			$\sum \delta_i^2 = 7555,60$	$t_{sr} = 4025,73$			$\sum \delta_i^2 = 13742,93$	
$S_A = \left(\frac{\sum \delta_i^2}{n(n-1)} \right)^{1/2} =$			5 ms	$S_A = \left(\frac{\sum \delta_i^2}{n(n-1)} \right)^{1/2} =$			6 ms	$S_A = \left(\frac{\sum \delta_i^2}{n(n-1)} \right)^{1/2} =$			8 ms	
Czas restartu $T_R =$			4033 ± 5 ms	Czas restartu $T_R =$			4023 ± 6 ms	Czas restartu $T_R =$			4026 ± 8 ms	
Błąd względny $\delta =$			0,13 %	Błąd względny $\delta =$			0,15 %	Błąd względny $\delta =$			0,20 %	
				VM Restart Priority								
				High	Medium	Low						
				Czas restartu T_R [ms]	4033 ± 5	4023 ± 6	4026 ±	8				
				Błąd względny δ [%]	0,13	0,15	0,20					
$T_{R\ \acute{s}r} = (T_{R\ h} + T_{R\ m} + T_{R\ l})/3$ [ms]				4027 ± 6								

Tab. 2. Czasy restartu maszyny wirtualnej
 Tab. 2. The virtual machine restart times

Obserwacja 1

Jak widać z Tab. 2., na podstawie oszacowanego średniego błędu kwadratowego S_A dla mierzonego czasu restartu T_R maszyny wirtualnej, dla różnych wartości parametru *VM Restart Priority*, w granicach oszacowanych dokładności pomiarów czasy restartu są porównywalne. Przyjęto zatem, że czas restartu maszyny wirtualnej jest niezależny od wartości *VM Restart Priority* i wyznaczono jego średnią wartość $T_{R\ \acute{s}r} = 4027 \pm 6$ ms.

5.3.1.2. Badanie związku pomiędzy VM Restart Priority, a czasem powrotu T_P maszyny wirtualnej do obszaru pracy przejściowej

W podobny sposób postanowiono zbadać, czy istnieje jakiś związek pomiędzy parametrem *VM Restart Priority* z czasem powrotu T_P maszyny wirtualnej z obszaru pracy przejściowej do obszaru pracy normalnej, tj. do średnich wartości czasów T_S odpowiedzi z obszaru pracy normalnej (OPN), podczas restartu na innym hoście, w przypadku awarii hosta na którym była uruchomiona. Wykonano serię $n=15$ pomiarów czasu powrotu T_P maszyny wirtualnej do obszaru pracy normalnej, dla trzech wartości parametru *VM Restart Priority*, tj.: *High, Medium, Low*. Wyniki pokazano w Tab. 3.

2. Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu powrotu T_P maszyny wirtualnej z obszaru przejściowego do obszaru pracy normalnej												
VM Restart Priority - High				VM Restart Priority - Medium				VM Restart Priority - Low				
Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu powrotu maszyny wirtualnej z obszaru przejściowego do obszaru pracy normalnej			Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu powrotu maszyny wirtualnej z obszaru przejściowego do obszaru pracy normalnej			Ilość pomiarów: 15	Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu powrotu maszyny wirtualnej z obszaru przejściowego do obszaru pracy normalnej			
Nr pomiaru	Czas powrotu t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	Nr pomiaru	Czas powrotu t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	Nr pomiaru	Czas powrotu t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	
1	1745,00	73,13	5348,48	1	1536,00	-167,27	27978,14	1	1708,00	-15,67	245,44	
2	1725,00	53,13	2823,15	2	1614,00	-89,27	7968,54	2	1702,00	-21,67	469,44	
3	1711,00	39,13	1531,42	3	1738,00	34,73	1206,40	3	1944,00	220,33	48546,78	
4	1691,00	19,13	366,08	4	1710,00	6,73	45,34	4	1779,00	55,33	3061,78	
5	1765,00	93,13	8673,82	5	1679,00	-24,27	588,87	5	1729,00	5,33	28,44	
6	1988,00	316,13	99940,28	6	1671,00	-32,27	1041,14	6	1679,00	-44,67	1995,11	
7	1812,00	140,13	19637,35	7	1768,00	64,73	4190,40	7	1619,00	-104,67	10955,11	
8	1702,00	30,13	908,02	8	1660,00	-43,27	1872,00	8	1719,00	-4,67	21,78	
9	1275,00	-398,87	157503,15	9	1844,00	140,73	19805,87	9	1713,00	-10,67	113,78	
10	1369,00	-302,87	91728,22	10	1726,00	57,73	3333,14	10	1630,00	-93,67	8773,44	
11	1343,00	-328,87	108153,28	11	1762,00	58,73	3449,60	11	1730,00	6,33	40,11	
12	1802,00	130,13	16934,68	12	1776,00	72,73	5290,14	12	1667,00	-56,67	3211,11	
13	1675,00	3,13	9,82	13	1606,00	-97,27	9460,80	13	1772,00	48,33	2336,11	
14	1725,00	53,13	2823,15	14	1770,00	66,73	4453,34	14	1763,00	39,33	1547,11	
15	1750,00	78,13	6104,82	15	1654,00	-49,27	2427,20	15	1701,00	-22,67	513,78	
$t_{sr} = 1671,87$			$\sum e_i^2 = 522485,73$	$t_{sr} = 1703,27$			$\sum e_i^2 = 93110,93$	$t_{sr} = 1723,67$			$\sum e_i^2 = 81859,33$	
$S_A = \sqrt{\sum e_i^2 / n(n-1)} = 50 \text{ ms}$			$S_A = \sqrt{\sum e_i^2 / n(n-1)} = 21 \text{ ms}$			$S_A = \sqrt{\sum e_i^2 / n(n-1)} = 20 \text{ ms}$						
Czas powrotu $T_P = 1672 \pm 50 \text{ ms}$			Czas powrotu $T_P = 1703 \pm 21 \text{ ms}$			Czas powrotu $T_P = 1724 \pm 20 \text{ ms}$						
Błąd względny $\delta = 2,98 \%$			Błąd względny $\delta = 1,24 \%$			Błąd względny $\delta = 1,15 \%$						
				VM Restart Priority								
				High	Medium	Low						
				Czas powrotu T_P [ms]	1672 ± 50	1703 ± 21	1724 ± 20					
				Błąd względny δ [%]	2,98	1,24	1,15					
$T_{P_{\text{sr}}} = (T_{P_{\text{h}}} + T_{P_{\text{m}}} + T_{P_{\text{l}}})/3$ [ms]				1700 ± 30								

Tab. 3. Czasy powrotu maszyny wirtualnej do obszaru pracy normalnej

Tab. 3. The virtual machine return time to normal work area

Obserwacja 2

Jak widać z Tab 3., na podstawie oszacowanego średniego błędu kwadratowego S_A dla mierzonego czasu powrotu T_P maszyny wirtualnej do obszaru pracy normalnej, dla różnych wartości parametru *VM Restart Priority*, w granicach oszacowanych dokładności pomiarów czasy powrotu są porównywalne. Przyjęto zatem, że czas powrotu maszyny wirtualnej do obszaru pracy normalnej jest niezależny od wartości *VM Restart Priority* i wyznaczono jego średnią wartość $T_{P_{\text{sr}}} = 1700 \pm 30 \text{ ms}$.

5.3.1.3. Badanie związku pomiędzy *VM Restart Priority*, a czasem T_N pozostawiania maszyny wirtualnej w obszarze pracy przejściowej

W końcu zbadano, czy istnieje jakiś związek pomiędzy parametrem *VM Restart Priority* z czasem T_N pozostawiania maszyny wirtualnej w obszarze pracy przejściowej, podczas restartu na innym hoście, w przypadku awarii hosta na którym była uruchomiona. Wykonano serię $n=15$ pomiarów czasu T_N pozostawiania maszyny wirtualnej w obszarze pracy przejściowej (OP), dla trzech wartości parametru *VM Restart Priority*, tj.: *High*, *Medium*, *Low*. Wyniki pokazano w Tab. 4.

3. Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu T_N pozostawania maszyny wirtualnej w obszarze pracy przejściowej												
VM Restart Priority - High				VM Restart Priority - Medium				VM Restart Priority - Low				
Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu pozostawania maszyny wirtualnej w obszarze pracy przejściowej				Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu pozostawania maszyny wirtualnej w obszarze pracy przejściowej				Obliczenie średniego błędu kwadratowego S_A (odchylenie standardowe) pomiaru czasu pozostawania maszyny wirtualnej w obszarze pracy przejściowej				
Ilość pomiarów: 15	Czas trwania stanu przejściowego t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	Ilość pomiarów: 15	Czas trwania stanu przejściowego t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	Ilość pomiarów: 15	Czas trwania stanu przejściowego t_i [ms]	Błąd $e_i = t_i - t_{sr}$ [ms]	e_i^2 [ms] ²	
Nr pomiaru				Nr pomiaru				Nr pomiaru				
1	227,00	-0,07	0,00	1	247,00	16,07	258,14	1	230,00	0,33	0,11	
2	231,00	3,93	15,47	2	241,00	10,07	101,34	2	224,00	-5,67	32,11	
3	234,00	6,93	48,07	3	230,00	-0,93	0,87	3	208,00	-21,67	469,44	
4	233,00	5,93	35,20	4	232,00	1,07	1,14	4	226,00	-3,67	13,44	
5	228,00	0,93	0,87	5	231,00	0,07	0,00	5	230,00	0,33	0,11	
6	199,00	-28,07	787,74	6	231,00	0,07	0,00	6	236,00	6,33	40,11	
7	221,00	-6,07	36,80	7	228,00	-2,93	8,60	7	236,00	6,33	40,11	
8	235,00	7,93	62,94	8	237,00	6,07	36,80	8	233,00	3,33	11,11	
9	229,00	1,93	3,74	9	219,00	-11,93	142,40	9	234,00	4,33	18,78	
10	225,00	-2,07	4,27	10	227,00	-3,93	15,47	10	237,00	7,33	53,78	
11	224,00	-3,07	9,40	11	227,00	-3,93	15,47	11	230,00	0,33	0,11	
12	224,00	-3,07	9,40	12	228,00	-4,93	24,34	12	233,00	3,33	11,11	
13	235,00	7,93	62,94	13	238,00	7,07	49,94	13	228,00	-1,67	2,78	
14	232,00	4,93	24,34	14	222,00	-8,93	79,80	14	229,00	-0,67	0,44	
15	229,00	1,93	3,74	15	228,00	-2,93	8,60	15	231,00	1,33	1,78	
$t_{sr} = 227,07$			$\sum e_i^2 = 1104,93$	$t_{sr} = 230,93$			$\sum e_i^2 = 742,93$	$t_{sr} = 229,67$			$\sum e_i^2 = 695,33$	
$S_A = \left[\sum e_i^2 / n(n-1) \right]^{1/2} = 2 \text{ ms}$				$S_A = \left[\sum e_i^2 / n(n-1) \right]^{1/2} = 2 \text{ ms}$				$S_A = \left[\sum e_i^2 / n(n-1) \right]^{1/2} = 2 \text{ ms}$				
Czas trwania stanu przejściowego $T_N = 227 \pm 2 \text{ ms}$				Czas trwania stanu przejściowego $T_N = 231 \pm 2 \text{ ms}$				Czas trwania stanu przejściowego $T_N = 230 \pm 2 \text{ ms}$				
Błąd względny $\delta = 1,01 \%$				Błąd względny $\delta = 0,81 \%$				Błąd względny $\delta = 0,79 \%$				

	VM Restart Priority		
	High	Medium	Low
Czas pozostawania w obszarze pracy przejściowej - T_N [ms]	227 ± 2	231 ± 2	230 ± 2
Błąd względny δ [%]	1,01	0,81	0,79
$T_{N \text{ sr}} = (T_{N \text{ h}} + T_{N \text{ m}} + T_{N \text{ l}}) / 3$ [ms]	229 ± 2		

Tab. 4. Czasy pozostawania maszyny wirtualnej w obszarze pracy przejściowej
 Tab. 4. Times remain a virtual machine in the transition area

Obserwacja 3

Jak widać z Tab 4., na podstawie oszacowanego średniego błędu kwadratowego S_A dla mierzonego czasu T_N pozostawania maszyny wirtualnej w obszarze pracy przejściowej, dla różnych wartości parametru *VM Restart Priority*, w granicach oszacowanych dokładności pomiarów czasu T_N są porównywalne. Przyjęto zatem, że czas pozostawania maszyny wirtualnej w obszarze pracy przejściowej jest niezależny od wartości *VM Restart Priority* i wynosi $T_{N \text{ sr}} = 229 \pm 2 \text{ ms}$.

5.3.1.4. Funkcja restartu maszyny wirtualnej

Na podstawie przeprowadzonych pomiarów: (i) czasu restartu T_R maszyny wirtualnej, (ii) czasu powrotu T_P maszyny wirtualnej z obszaru pracy przejściowej do obszaru pracy normalnej, (iii) czasu pozostawania T_N maszyny wirtualnej w obszarze pracy przejściowej oraz dokonanego oszacowania błędów pomiarów tych wielkości obliczono współczynnik kierunkowy wykresu restartu maszyny wirtualnej w obszarze pracy przejściowej, jak również dokonano oszacowania maksymalnego błędu jego wyznaczenia, co pokazano w Tab. 5.

Współczynnik kierunkowy wykresu restartu maszyny wirtualnej obliczono ze wzoru:

$$A = f(T_R, T_P, T_N) = ((T_R - T_S) - (T_P - T_S)) / T_N = (T_R - T_P) / T_N$$

Maksymalny błąd bezwzględny jego oszacowania [36] obliczono ze wzoru:

$$\Delta A = \pm (|\partial f / \partial T_R| \Delta T_R + |\partial f / \partial T_P| \Delta T_P + |\partial f / \partial T_N| \Delta T_N)$$

Po wyznaczeniu pochodnych cząstkowych:

$$\partial f / \partial T_R = 1 / T_N$$

$$\partial f / \partial T_P = -1 / T_N$$

$$\partial f / \partial T_N = T_P / T_N^2$$

otrzymano wzór na błąd bezwzględny $\Delta A = \pm (|\Delta T_R / T_N| + |\Delta T_P / T_N| + T_P |\Delta T_N / T_N^2|)$,

gdzie ΔT_R , ΔT_P , ΔT_N , oznaczają wcześniej obliczone błędy bezwzględne pomiarów tych wielkości. Błąd względny obliczono ze wzoru $\delta = (\Delta A / A) \cdot 100\%$.

	VM Restart Priority		
	High	Medium	Low
Współczynnik kierunkowy A	-10,40 ± 0,32	-10,04 ± 0,18	-10,02 ± 0,18
Błąd względny δ [%]	3,05	1,76	1,80
Współczynnik kierunkowy A_{sr} = (A _{high} + A _{medium} + A _{low}) / 3 = -10,16 ± 0,23			

Tab. 5. Współczynnik kierunkowy wykresu restartu maszyny wirtualnej w obszarze pracy przejściowej

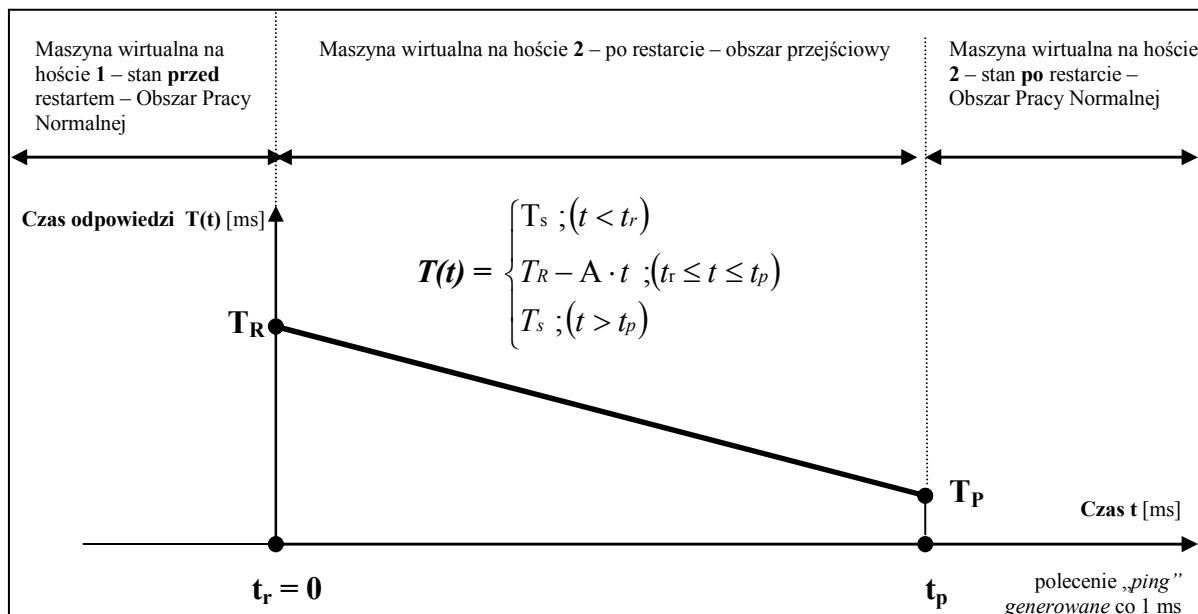
Tab. 5. Directional coefficient of the graph of the restart of the virtual machine in the transition area

Obserwacja 4

Jak widać z Tab. 5., na podstawie bezwzględnego maksymalnego błędu obliczenia współczynnika kierunkowego wykresu restartu maszyny wirtualnej w obszarze pozostawiania maszyny wirtualnej w obszarze pracy przejściowej, dla różnych wartości parametru *VM Restart Priority*, w granicach oszacowanych dokładności pomiarów współczynniki kierunkowe są porównywalne.

Przyjęto zatem, że współczynnik kierunkowy wykresu czasów odpowiedzi podczas restartu maszyny wirtualnej w obszarze pracy przejściowej jest niezależny od wartości *VM Restart Priority*, a jego średnią wartość wynosi $A = A_{sr} = (A_{high} + A_{medium} + A_{low}) / 3 = -10,16 \pm 0,23$.

Podsumowując przeprowadzone obliczenia, podczas restartu maszyny wirtualnej, funkcja czasu odpowiedzi $T(t)$ maszyny wirtualnej na polecenia *ping* w funkcji czasu, przyjmując początek układu współrzędnych w punkcie $t_r = 0$, ma postać pokazaną na Rys. 10.



Rys. 10. Wykres funkcji $T(t)$ podczas restartu maszyny wirtualnej w funkcji czasu

Fig. 10. Virtual machine restart graph as a function $T(t)$ of time

gdzie:

T_s -średni czas odpowiedzi maszyny wirtualnej na polecenie *ping* w obszarze pracy normalnej

t_r - chwila czasowa w której rozpoczyna się proces restartu maszyny wirtualnej na hoście 1

t_p - chwila czasowa określająca powrót maszyny wirtualnej do obszaru pracy normalnej na hoście 2

6. Podsumowanie i wnioski

Przedstawiona w pracy analiza wpływu zdarzeń krytycznych na pracę klastra *VMWARE ESXi HA*, z pełną redundancją połączeń światłowodowych, współpracującego z systemem pamięci masowej typu *Fiber Channel SAN Storage* pracującej pod kontrolą systemu operacyjnego *Linux Openfiler ESA*, który wykorzystano do przetestowania, pod kątem ciągłości działania w sytuacjach wystąpienia zdarzeń krytycznych, systemu *Synapse*, prowadzi do następujących wniosków:

1. Redundantne połączenia światłowodowe w klastrze *Wmware ESXi HA FT* zapewniają ciągłość pracy systemu *Synapse* w każdej chwili czasowej w przypadku uszkodzenia któregośkolwiek z nich,

2. Uszkodzenia któregokolwiek z połączeń redundantnych nie generują żadnych spadków prędkości odczytu, przerw czy zaników w dostępie do dysków wirtualnych rezydujących w pamięci masowej.

Na podstawie wykonanych pomiarów oraz obserwacji, ocena dostępności systemu *Synapse* pod kątem ciągłości jego działania, dokonana poprzez pomiar czasu potrzebnego na restart tej maszyny wirtualnej w badawczym środowisku klastrowym na innym hoście, w sytuacji wystąpienia zdarzeń krytycznych, powodujących awarię tej maszyny wirtualnej lub hosta *ESXi* na której została uruchomiona, przedstawia się następująco:

1. System *Synapse*, daje odpowiedzi na zapytania w każdej chwili czasowej przez maszynę wirtualną znajdującą się albo na Hoście 1 albo na Hoście 2,
2. Spadki prędkości odczytu jego dysków systemowych, w czasie restartu maszyny wirtualnej na innym serwerze klastra, nie osiągają zera i są niezauważalne dla stacji roboczych wyświetlających rentgenogramy i tomogramy,
3. Czas restartu maszyny wirtualnej, pomijalny w porównaniu z ogólnym czasem pracy systemu działający w systemie ciągłym, praktycznie nie ma wpływu na jego dostępność i jest niezależny od wartości parametru *VM Restart Priority*,
4. Funkcja $T(t)$ restartu maszyny wirtualnej w obszarze przejściowym, określającej czas odpowiedzi maszyny wirtualnej na polecenia *ping* w funkcji czasu, jest funkcją liniową.

LITERATURA

1. VMware Infrastructure Architecture Overview, White paper, VMware Inc., 2006.
2. Bożyk K.: Technologia SAN (Storage Area Networks)-Praca dyplomowa magisterska, Samodzielny Zakład Sieci Komputerowych Politechnika, Łódź 2005.
3. Mon-Chau S.: Storage Area Network, Infotrend Technology Inc., 2007.
4. vSphere ESXi Bare-Metal Hypervisor | VMware Polska, www.vmware.com [dostęp 2015-11-29], <http://www.vmware.com/products/esxi-and-esx.html>.
5. Long J.: Install and Configure an ESXi 5.1 Host, MOREnet 221 N. Stadium Blvd., Suite 201, Columbia, Mo. 65203, 2012.
6. Fibre Channel SAN Configuration Guide, VMware, Inc., EN-000109-05, 2010.
7. Deploying VMware High Availability & Fault Tolerance cluster on HA3969U (NFS), Storageflex Inc, 2014.
8. Host Bus Adapter, <http://searchstorage.techtarget.com/definition/host-bus-adapter>.
9. VMware vSphere 5.1 Documentation Center, <https://pubs.vmware.com/vsphere-51/index.jsp#com.vmware.vsphere.doc/GUID-1B959D6B-41CA-4E23-A7DB-E9165D5A0E80.html>.

10. Fiber Channel Storage Area Network (FC SAN),
<https://www.techopedia.com/definition/1081/fiber-channel-storage-area-network-fc-san>.
11. Smith E., Hultman R.: Fibre Channel SAN Topologies, EMC, P/N H8074.7, EMC Corporation, 2015.
12. Pico J.: An Analysis From A Security Perspective, Fibre Channel Storage Area Networks - SANS Institute, 2006.
13. Norman D.: Fibre Channel Technology for Storage Area Networks, 2001.
14. Black D.L.: A Storage Menagerie: SANs, Fibre Channel, Replication and Networks, Miami 2011.
15. Dale D.: iSCSI & Fibre Channel SANs, 2007.
16. Fibre Channel Storage area Network,
<https://www.google.pl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=73&ved=0ahUKEwiQ6me4trSAhVFVBQKHdj8AIE4RhAWCCgwAg&url=http%3A%2F%2Fjbiet.edu.in%2Fcoursefiles%2Fcourse%2FHO%2Fcourse4%2FSAN4.pdf&usq=AFQjCNGuJ-qgJaBf9f1Kt-hAfMhE-HxFyw&sig2=5iJgB9TpakoLLkGcM11-Sw&cad=rja>.
17. VMware vSphere 5.1: 16Gb Fibre Channel SANs with HP ProLiant DL380 Gen8 servers and HP 3PAR Storage, 2017.
18. Connecting Sans Over Metropolitan and Wide Area Networks, White Paper, 2001.
19. Openfiler, <https://www.openfiler.com/products>.
20. Bastiaansen R.: Install and configure Openfiler for ESXi shared storage with NFS and iSCSI, 2013.
21. Seget V.: How to configure Openfiler iSCSI Storage for use with VMware ESX, 2015.
22. Openfiler Administration Guide, Openfiler Ltd. UK, 2009.
23. Tivari R.: How to install openfiler 2.99, 2013.
24. Intel: Delivering Low Cost High IOPS VM Datastores Using Openfiler, Document Number: 329238-002US, 2014.
25. Gegere J.: How to configure Openfiler v2.3 iSCSI Storage with VMware ESXi v4, 2017.
26. 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.
27. vSphere 5 Documentation Center - vSphere Availability, EN-000916-00, 2012.
28. How To Install and Update A Redhat Linux Kernel RPM, 2004.
29. Ball B.: Red Hat Linux 7.3, "Księga Experta", Tłumaczenie: Maciej Pasternacki, ISBN: 83-7197-787-5, 2002.
30. vSphere Monitoring and Performance, VMware, Inc., EN-001102-03, 2013.
31. Monitoring ESXi Performance Using Server & Application Monitor, SolarWinds Worldwide, LLC, 2013.
32. ServerView Event Manager, FUJITSU Software ServerView Suite, 2017.
33. Seagate Nytro vCenter Plug-In Solution, Seagate Technology LLC, 100769979, Rev.C, 2015.
34. Automating High Availability (HA) Services with VMware HA, Technical Note, VMware, Inc., 2006.

35. VMware vSphere 4 - ESX and vCenter Server - How VMware HA Works, https://pubs.vmware.com/vsphere-4-esx-vcenter/index.jsp?topic=/com.vmware.vsphere.availability.doc_41/c_useha_works.html.
36. Dryński T.: Ćwiczenia laboratoryjne z fizyki, PWN, Warszawa, 1976.
37. artykuł zgłoszony do publikacji - Bok Z.: Heterogeniczne dyski wirtualne w środowisku VMWare ESXi, Studia Informatica, Politechnika Śląska, 2017.
38. Novell Cluster Services Implementation Guide for VMware Open Enterprise Server, Novell Inc., 2016.
39. Avaya Aura Virtualized Environment, Avaya Inc., 2013.

Recenzent:

Wpłynęło do Redakcji 30 października 2017 r.

Abstract

In the article they made analysis of the impact of critical events in the work of the virtual machines, for business continuity and the assessment of the availability, in VMware ESXi cluster high availability with full redundancy fiber connectivity on the example of the portion of the specialized hospital System of indirect Radiography "Synapse" FUJIFILM Medical Systems USA, Inc., used in Specialized Hospital named Prof. E. Michalowski MEDHOLDING S.A. In order to assess the availability of the system that is running as a virtual machine on one of the host in VMWARE ESXi laboratory cluster infrastructure, a method for measuring the time needed to restart this virtual machine on another host in this cluster, in situations of critical events that caused the failure of the host at which it was running, has been proposed.

Presented in the work of the analysis of the influence of critical events to work VMWARE ESXi cluster HA leads to the following conclusions:

1. Redundant fiber optic connections in the cluster, ensure the continuity of the system "Synapse" at any time in the event of a failure of either of them.
2. Damage to any of the redundant connections do not generate any inheritance reading speed, interruptions or blackouts in the access to the virtual disks residing in the mass storage.

On the basis of measurements and observations, assessment of the availability of the system "Synapse" for the continuity of its activities, by measuring the time needed to restart the virtual machine in a clustered research environment on another host, in a situation of critical events that caused it to crash the virtual machine or the ESXi host on which it is running, is as follows:

1. "Synapse" system, installed in the form of a virtual machine in the lab cluster, will give you the answers to queries at any time by the virtual machine is either on Host 1 or on Host 2,
2. Falls in the reading speed of his system disks, during the restart of the virtual machine on other server of the cluster, they aren't achieving the zero and are unnoticeable for workstations showing roentgenograms and tomograms,
3. Time to reboot the virtual machine, negligible compared with the general working time system operating in a continuous system, there is virtually no impact on its availability and is independent of the values of the parameter the VM Restart Priority
4. Virtual machine restart function $T(t)$ in transition area, determining the response time of the virtual machine on the ping command as a function of time, is a linear function.