

Tomasz SALAMON, Katarzyna HAREŹLAK  
Politechnika Śląska, Instytut Informatyki

## ANALIZA MODELI I WYDAJNOŚCI WYBRANYCH MECHANIZMÓW CHMURY OBLICZENIOWEJ

**Streszczenie.** W artykule skupiono uwagę na zagadnieniach dotyczących chmury obliczeniowej. Szczególną uwagę poświęcono platformie Windows Azure, na podstawie której zbudowano środowisko testowe. Środowisko to wykorzystano do zbadania wydajności pracy procesora, operacji dostępu do dysków, efektywności przesyłu danych w sieci oraz do przeanalizowania pracy menadżera ruchu.

**Słowa kluczowe:** chmura obliczeniowa, Windows Azure, wydajność

## THE ANALYSIS OF MODELS AND PERFORMANCE OF THE CHOSEN CLOUD COMPUTING MECHANISMS

**Summary.** Issues concerning cloud computing were analysed in the paper. Special focus was given to on one of the popular platforms – Windows Azure. An environment based on it was designed and configured and several experiments were conducted using it. They included tests regarding processor performance, the quality of file transfer and the speed of file operations and Traffic Manager work.

**Keywords:** cloud computing, Windows Azure, performance

### 1. Wstęp

Ideę chmury obliczeniowej łatwo zrozumieć osobie, której informatyczna wiedza jest bardzo skromna, ponieważ trudno jest ujednoczyć jej definicję. Taka osoba ograniczy się tylko do swojego celu i słusznie zauważy, że nie potrzebuje niczego więcej poza dokładnie tym, czego chce. Jest to całkowicie nowatorskie podejście, w którym nie ma mowy o tym, że użytkownik interesuje się czymś więcej niż zakupionym produktem. Znaczy to tyle, że jeśli firma chce zakupić instancję systemu operacyjnego na maszynie, nie musi interesować się

konfiguracją sprzętową, liczbą procesorów, zabezpieczeniem prądowym, wydajnością i stabilnością połączenia z Internetem. Również nie musi się przejmować personelem, a nawet licencjami oprogramowania. Tym wszystkim zajmuje się dostawca, a jedyny towar to użytkowany system. Podobnie gdy zakupiona będzie sama aplikacja, np. serwis multimedialny udostępniający wideo strumieniowo, portal społecznościowy czy aukcyjny – poprawne funkcjonowanie jest zapewnione przez dostawcę.

Użycie terminu „Cloud Computing” po raz pierwszy w znaczeniu dziś znanym przypisuje się dyrektorowi generalnemu Google – Ericowi Schmidtowi, który na konferencji dotyczącej strategii mechanizmów wyszukiwania 9 sierpnia 2006 roku powiedział (stenogram – tłumaczenie własne):

*To, co jest ciekawe, to istnienie nowego wschodzącego modelu, a wy wszyscy jesteście tu, bo jesteście częścią tego modelu. Nie sądzę, aby ludzie na prawdę zrozumieli, jak duża jest to okazja. Wychodzi się z założenia, że usługi przetwarzania danych oraz cała infrastruktura powinny znajdować się na serwerach. My nazywamy to chmurą obliczeniową (Cloud Computing) – te wszystkie usługi powinny znajdować się gdzieś w „chmurze”. A fakt, czy posiadasz odpowiednią przeglądarkę, sposób dostępu, czy posiadasz PC czy MAC, czy telefon komórkowy, a może BlackBerry lub jakiegokolwiek urządzenie, które jeszcze nie zostało wynalezione – nie ma wpływu na to, że masz dostęp do chmury [2].*

Fakt, iż chmura obliczeniowa jest nowym tematem, łatwo można pokazać na przykładzie zapytań wyszukiwarki Google oraz wbudowanego w nią narzędzia Trends. Badając przy jego użyciu występowanie takich zwrotów, jak Cloud Computing, Windows Azure, Google App Engine, Amazon EC2, Chmura Obliczeniowa – możemy zauważyć, iż przed wystąpieniem Pana Erica Schmidta nikt nie odpytywał wyszukiwarki w tej tematyce. Pierwsze próby zgłębiania zagadnienia pojawiły się w Google w lipcu 2006 roku oraz dotyczyły najstarszego na rynku rozwiązania firmy Amazon. Sam termin „Cloud Computing” po raz pierwszy został użyty niespełna rok później, tj. w kwietniu 2007 roku. Chociaż zainteresowanie było małe, to już wtedy była używana właściwa terminologia. Polski rynek zaczął szukać informacji na temat chmury obliczeniowej, konkretnie mechanizmu Google App, w kwietniu 2008 roku, natomiast tłumaczenie nazwy istnieje w bazie wyszukiwarki od kwietnia 2011 roku. Na przestrzeni tych lat technologie związane z przetwarzaniem w chmurze podlegały ciągłemu rozwojowi [3].

## 2. Modele chmury obliczeniowej

Przy pisaniu o typach chmury obliczeniowej potrzebna jest zmiana sposobu myślenia z dotychczasowego, tzn. z: co musi zostać zrobione, aby osiągnąć pożądaną cel?, na myśle-

nie: ile będzie kosztowało dzierżawienie już gotowego efektu? W prostych słowach chmura jest wszystkim, bo tak naprawdę całe dostępne oprogramowanie da się w niej zamknąć. Wszystko można zatem traktować jako usługę, nie martwiąc się sposobem, w jaki ta usługa zostanie pozyskana. Końcowy użytkownik otrzymuje gotowy efekt i zajmuje się tylko i wyłącznie nim. Cała chmura może zostać nazwana „wszystko jako usługa”. Różnica pomiędzy technologiami wewnętrznymi (firmowymi) a zastosowaniem chmury obliczeniowej polega na przenoszeniu odpowiedzialności. Dokładnie tym samym różnią się od siebie poszczególne modele chmury obliczeniowej. Przykładowa infrastruktura została umieszczona w tabeli 1.

Tabela 1  
Podział hierarchiczny infrastruktury [4]

Aplikacje
Dane
Środowisko uruchomieniowe
Oprogramowanie pośredniczące
System operacyjny
Wirtualizacja
Serwery
Nośniki
Sieć

Jak nie trudno się domyślić, w przypadku nieużywania chmury za całą tę infrastrukturę bierze odpowiedzialność firma. W tabeli 1 przedstawiono tylko informatyczną stronę projektu, którą da się rozdzielić na kilka warstw. Pozwala to dokonać sensownego podziału odpowiedzialności i stworzyć kilka modeli chmury obliczeniowej.

Mark Klems na swoim blogu wpisuje się świetnie w ten typ analizy problemu, porównując różne modele chmury obliczeniowej do cebuli [5], w której środku znajduje się infrastruktura obejmująca sprzęt, system operacyjny i środowisko uruchomieniowe. Drugą warstwą jest platforma, w której umieszczane są aplikacje. Końcowa, zewnętrzna powłoka to uruchomione aplikacje, gotowe, by można było z nich zdalnie korzystać.

Model **Oprogramowanie jako Usługa** (Software as a Service, SaaS) można umieścić na samej „górze” usług dostępnych w chmurze obliczeniowej. Znaczy to, że stanowi on ostatni produkt, który może dostać usługobiorca. Dostawca jest odpowiedzialny za wszystkie poziomy widoczne w tabeli 1.

**Platforma jako Usługa** (Platform as a Service, PaaS), podobnie jak Oprogramowanie jako Usługa, znajduje się na samej górze usług chmury obliczeniowej, natomiast zakres odpowiedzialności rozkłada się nieco inaczej. W przypadku hierarchii chmury użytkownik zajmuje się już dwoma znajdującymi się najwyżej elementami.

W przypadku **PaaS** dostawca odpowiada za kompletną infrastrukturę, wliczając w to maszyny, systemy operacyjne, wszystkie potrzebne biblioteki i środowiska uruchomieniowe.

Klient natomiast dostarcza aplikację i jest właścicielem końcowego produktu. Dzierżawione są wszystkie potrzebne do jego funkcjonowania elementy.

**Infrastruktura jako Usługa** zarówno według Microsoftu, jak i przedstawionej koncepcji warstwowej znajduje się w ścisłym centrum chmury. To, co kryje się pod samą nazwą, jest przez poszczególnych dostawców różnie interpretowane [1, 6]. Do niewątpliwych plusów rozwiązania należy zaliczyć możliwość bardzo prostej i szybkiej migracji firmowej infrastruktury do chmury – zakupu gotowej, zainstalowanej wersji systemu (np. Microsoft Windows, Linux itp.) lub wykonanie swojego obrazu i umieszczenie go w sieci.

#### **Pozostałe modele**

Warstwowy podział na modele nie wyczerpuje w pełni tego, co kryje termin *cloud computing*. Wynika to m.in. z tego, iż termin ten nie ma „sztywnej” definicji. Wieloletni pracownik firmy Microsoft, Roger Jennings, w książce „Cloud Computing with the Windows Azure Platform” [6], poza tymi już opisanymi w pracy, definiuje takie pozycje, jak: *Pliki (magazyn) jako Usługa*, *Baza danych jako Usługa*, *Oprogramowanie plus Usługa*, *Komunikacja jako Usługa*, *Monitoring jako Usługa*, *Narzędzia jako Usługa*, *Wszystko jako Usługa*.

### **3. Chmura obliczeniowa kontra własne rozwiązanie**

Chmura obliczeniowa udostępniana przez zewnętrznych dostawców, jak Amazon, Google czy Microsoft, jest logicznym tworem wykorzystującym już od dawna istniejący sprzęt i istniejące oprogramowanie. Teoretycznie możliwe jest zainstalowanie podobnego rozwiązania bezpośrednio w siedzibie firmy. Wynika z tego, że pod względem technologicznym rozwiązania się nie różnią. Wszystko dotyczy tylko problemu odpowiedzialności, tego, kto stworzy infrastrukturę, kto będzie usuwał powstające problemy i przede wszystkim kto i ile za to zapłaci. Taki punkt widzenia doprowadza do rozwinięcia tabeli 1 w zestawienie podziału odpowiedzialności pomiędzy właściciela firmy i dostawcę w poszczególnych wariantach rozwiązań serwerowych (tabela 2).

Kolor jasnoszary określa odpowiedzialność klienta, natomiast ciemnoszary to, za co odpowiada dostawca. W rozwiązaniach wewnętrznych to właściciel odpowiada za wszystkie warstwy infrastruktury. To on zatrudnia specjalistów, podpisuje umowy z dostawcami prądu czy usług telekomunikacyjnych, kupuje urządzenia i oprogramowania. Wszystkie te elementy nie są regulowane żadnymi przepisami, to podmiot właściciela jest w pełni odpowiedzialny za jakość posiadanych rozwiązań.

Tabela 2

## Podział hierarchiczny różnych typów infrastruktury [4]

Rozwiązania wewnętrzne	Infrastruktura jako Usługa	Platforma jako Usługa	Aplikacja jako Usługa
Aplikacje	Aplikacje	Aplikacje	Aplikacje
Dane	Dane	Dane	Dane
Środowisko uruchomieniowe	Środowisko uruchomieniowe	Środowisko uruchomieniowe	Środowisko uruchomieniowe
Oprogramowanie pośredniczące	Oprogramowanie pośredniczące	Oprogramowanie pośredniczące	Oprogramowanie pośredniczące
System operacyjny	System operacyjny	System operacyjny	System operacyjny
Wirtualizacja	Wirtualizacja	Wirtualizacja	Wirtualizacja
Serwery	Serwery	Serwery	Serwery
Nośniki	Nośniki	Nośniki	Nośniki
Sieć	Sieć	Sieć	Sieć

W tabeli 3 zebrano dane na temat uśrednionych kosztów rozwiązania serwerowego oraz kosztów korzystania z usług Windows Azure. W pierwszej grupie wydatków uwzględniono ceny serwerów PowerEdge R515, ProLiant DL165 G7, System x3755 M3 (zgodnie z danymi producenta [11, 12, 13, 15]), dysków, systemu operacyjnego oraz opłat za Internet [14] i energię elektryczną (za wartość ciągle pobieraną przez urządzenie uznano moc zasilacza).

Tabela 3

## Zestawienie kosztów Windows Azure oraz rozwiązania serwerowego

Architektura Windows Azure	Roczny abonament Windows Azure	Jednorazowy koszt zakupu serwera	Roczny koszt energii elektrycznej	Roczny koszt łączny symetrycznych 100M/100M	Sumaryczny roczny koszt
Extra Small	\$133,08	\$6081,42	3155,76 zł = \$1011,46	\$9615,38	\$15696,80
Small	\$598,44	\$7555,01		\$9615,38	\$17170,39
Medium	\$1196,88	\$11613,49		\$19230,76	\$30844,25
Large	\$2393,64	\$19280,26		\$38461,52	\$57741,78
Extra Large	\$4787,16	\$35014		\$76923,04	\$111937,04
A6	\$6781,88	\$19649,56		\$38461,52	\$19649,56
A7	\$13563,36	\$35754,04		\$76923,04	\$112677,08

W celu ujednoczenia miar wartości PLN zostały przeliczone po kursie dolara z dnia 27.09.2013: 1 USD = 3,12 PLN.

Otrzymane wyniki całkowicie dyskredytują własne rozwiązanie nawet w dłuższej perspektywie – szczególnie w przypadku najdroższym. Analizując głębiej, można zadać sobie pytanie: ile razy użytkownik-firma wykorzysta tak kosztowną konfigurację, jak A7?

## 4. Testy wydajności

Analizę wydajności środowiska „Chmury” przeprowadzono na podstawie jednej z popularnych platform – Windows Azure. Przygotowanie tej platformy do testów obejmowało utworzenie kilku komponentów:

1. **maszyn wirtualnych** – aby zapewnić działanie dodatkowych programów, należało dodać odpowiednie reguły w ścianie ogniowej (ang. *firewall*) maszyn wirtualnych (tabela 4) oraz dodać punkty końcowe (ang. *end points*) w konfiguracji platformy. W ten sposób możliwy był dostęp do serwera WWW oraz programu iperf z zewnątrz maszyny (porty 80 oraz 1500);

Tabela 4

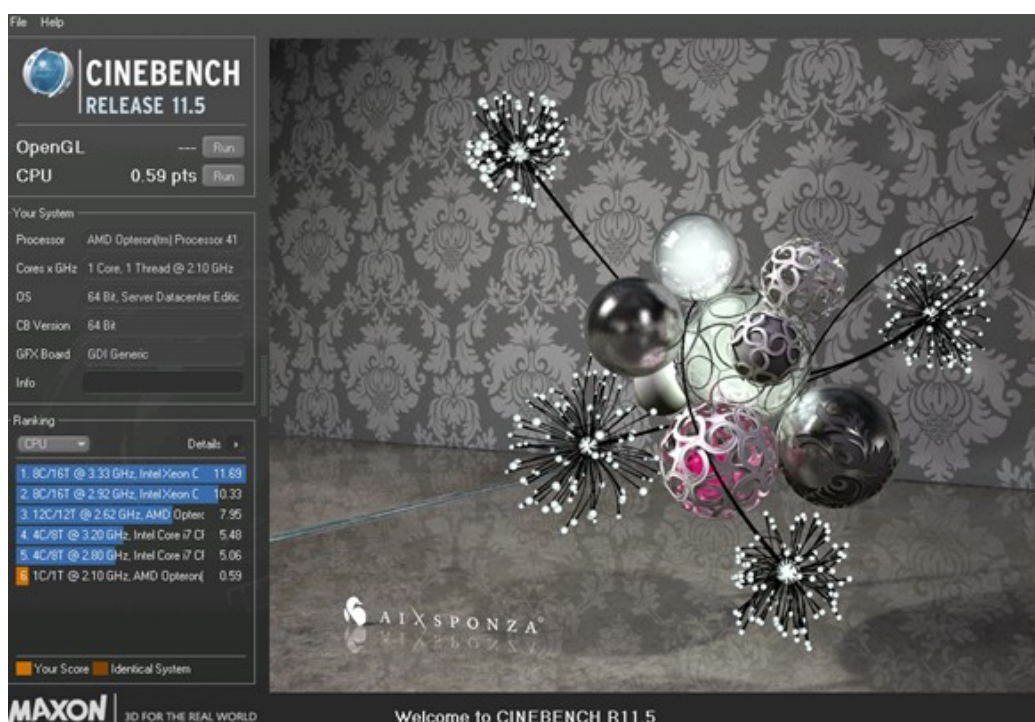
Maszyny wirtualne użyte do testów platformy Windows Azure

Nazwa maszyny	Adres DNS	Lokalizacja
salamontest1	salamontest1.cloudapp.net	Europa Północna
salamontest2	salamontest2.cloudapp.net	Europa Zachodnia
salamontest3	salamontest3.cloudapp.net	USA Wschód
salamontest4	salamontest4.cloudapp.net	Azja Wschód

2. **instancji sieci Web** – na platformie została utworzona jedna instancja sieci Web o nazwie salamontest i adresie <http://salamontest.azurewebsites.net>. Jej konfiguracja pozwalała na umieszczanie plików stron oraz skryptów za pomocą narzędzia Dropbox. W porównaniu z automatycznym wdrażaniem Visual Studio czy Git pozwala ono na umieszczanie własnych rozwiązań na lokalnej maszynie, skąd zainstalowana aplikacja automatycznie umieści je w serwisie Dropbox. Synchronizacja tego ostatniego z platformą Windows Azure odbywa się przez naciśnięcie jednego przycisku w panelu zarządzania stroną;
3. **menedżera ruchu** – programu Load Balancer, służącego do wyrównywania ruchu na poszczególnych instancjach udostępniających tę samą usługę.

### 4.1. Test procesora

Pierwszą częścią testu było uruchomienie na serwerze programu Cinebench 11.5, który wykonuje serię testów polegającą na wyrenderowaniu obrazu. Działanie kończy się wystawieniem oceny, którą na stronie producenta można porównać z innymi konfiguracjami sprzętowymi. Przez cały okres działania benchmarka użycie procesora utrzymywało się na poziomie 100%. Wykorzystywane zasoby w przypadku *salamontest1* zostały określone przez narzędzie *System Information* jako: procesor AMD Opteron(tm) Processor 4171 HE, 2095 Mhz, 1 Core(s), 1 Logical Processor(s), Total Physical Memory – 768 MB, Available Physical Memory – 231 MB, Total Virtual Memory – 1,75 GB, Available Virtual Memory – 1,05 GB.



Rys. 1. Zrzut ekranu programu Cinebench  
Fig. 1. The screen of the Cinebench program

Praca programu zakończyła się wyrenderowaniem grafiki widocznej na rysunku 1 oraz wydaniem niskiej noty na poziomie 0,59 punktu.

Drugim testem sprawdzającym wydajność procesora było obliczenie liczby Pi za pomocą wzoru:

$$\pi = 4 \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right) \quad (1)$$

Tabela 5

Kod programu obliczającego liczbę Pi: a) pseudokod, b) skrypt Windows PowerShell

a) Pseudokod	b) Windows PowerShell
<pre>double ObliczPi(int max) {     double wynik=1.0;     for(int i=3; i&lt;max; i+=2) {         wynik-=1/i;         i+=2;         wynik+=1/i;     }     return 4*wynik; }</pre>	<pre>\$parametr=\$args[0] if(\$parametr -gt 0) {     \$parametr*=[math]::Pow(10, 8)     Write-Host \$parametr     \$wynik=1     for (\$i=3; \$i -le \$parametr; \$i+=2) {         \$wynik-=1/\$i         \$i+=2         \$wynik+=1/\$i     }     Write-Host (4*(\$wynik)) } else {     Write-Host Podaj parametr! }</pre>

Program został napisany w postaci skryptu Windows PowerShell. Jego pierwszym elementem jest sprawdzenie, czy parametr przekazywany do programu jest większy od zera. Ponieważ ma on obciążać procesor, parametr jest mnożony przez  $10^8$ .

Poza wyliczoną liczbą Pi program zwraca przemnożony parametr. W celu oszacowania czasu trwania obliczeń wywołuje się go w postaci parametru komendy *Measure-Command*. Uzyskane wyniki zaprezentowano w tabeli 6.

Tabela 6

Zestawienie czasów wyliczania liczby Pi o określonej dokładności

Parametr	Czas obliczeń
$1 \cdot 10^8$	26,58 s
$2 \cdot 10^8$	52,77 s
$3 \cdot 10^8$	79,35 s
$4 \cdot 10^8$	105,14 s

#### 4.2. Przesył danych w sieci

Analizując jakość komunikacji w zbudowanym środowisku, przeprowadzono kilka prób kopiowania danych o różnej charakterystyce. Do testów użyto serwera *salamontest1*, skonfigurowanego jako maszyna wirtualna wyposażona w jeden rdzeń fizyczny oraz jedno łącze 100 Mbit. Komputer, który nawiązywał z nim połączenie, znajdował się na terenie Niemiec, oparty był na procesorze Intel Core i5 oraz pracował pod kontrolą systemu Microsoft Windows XP. Komunikacja ze światem odbywała się za pomocą łącza światłowodowego o parametrach 25 Mbit/25 Mbit.

Pierwszym przeprowadzonym testem było przekopiowanie z lokalnej maszyny folderu zawierającego bardzo dużą liczbę małych plików. W tym celu połączenie zdalnego pulpitu zostało skonfigurowane w taki sposób, aby lokalne zasoby dyskowe były widziane przez serwer. Kopiowanie danych odbyło się przy użyciu standardowego narzędzia systemu Windows. Przesyłane dane to 317 MB (333 123 103 bajtów) – 6802 plików zawartych w 402 folderach. Średni rozmiar pojedynczego pliku wynosi 48 974,28 B, czyli około 47 KB. Obserwowana szybkość transferu wahała się pomiędzy wartościami 0,5 KB/s a 1,2 MB/s. Przesył trwał 1 godzinę 29 minut i 11 sekund, średnia prędkość transmisji wynosiła 60,79 KB/s. Jest to niewielka prędkość. Tłumaczy to natomiast rozmiar przesyłanych informacji, będący podobnego rzędu. Spadek ten jest spowodowany głównie przez każdorazową operację zapisu/odczytu na obydwu dyskach – źródłowym i docelowym.

Następną częścią testu było przesłanie pojedynczych plików o większym rozmiarze: 389 745 712 bajtów (371 MB) oraz 1 268 927 bajtów (1,2 MB). Podsumowanie testu transferu plików zawiera tabela 7. Do opisu zostały użyte skróty „ks” oraz „sk”, oznaczające odpowiednio przesył z klienta na serwer oraz na odwrot.



Tabela 7

## Podsumowanie testu transferu plików

Wielkość danych	Liczba plików/katalogów	Czas przesyłu/transfer
333 123 103 bajtów (317 MB)	6802/402	ks: 1 h 29 m 11 s / 60,79 KB/s
389 745 712 bajtów (371 MB)	1/0	ks: 15 m 25 s / 411 KB/s
389 745 712 bajtów (371 MB)	1/0	sk: 11 m 56 s / 531,58 KB/s
1 268 927 bajtów (1,2 MB)	1/0	ks: 4,1 s / 302,24 KB/s
1 268 927 bajtów (1,2 MB)	1/0	sk: 3,04 s / 407,63 KB/s

Wyniki pomiarów pokazują, iż chwilowa szybkość transferu badana za pomocą narzędzi takich jak *speedtest.pl* [10] nie jest do końca miarodajna. Maksymalna średnia prędkość przesyłu to 531,58 KB/s, co przekłada się na szybkość 4,15 Mbps. Niewielka szybkość może być spowodowana sposobem przesyłania danych – usługą zdalny pulpit. Skupia się ona na bezpieczeństwie, a zatem połączeniu szyfrowanym, a nie na osiągnięciach.

Kolejna część badania transferu zakłada użycie narzędzia *iperf*. Jest to program służący do pomiaru wydajności łącza, pracujący w trybie klient-serwer. Sposób dokonywania pomiarów zakładał zbadanie szybkości przesyłu:

- 1) pomiędzy klientem opisanym na początku podrozdziału a każdym serwerem z tabeli 4,
- 2) wewnątrz Azure pomiędzy wszystkimi serwerami z tabeli 4.

Co do punktu pierwszego, na każdym z urządzeń został uruchomiony program *iperf* w trybie serwera. Klient natomiast odpytywał serwery po kolei, rejestrując szybkość odpowiedzi. Sytuacja została odwrócona, aby zbadać prędkość odwrotnego przesyłu. Punkt drugi wyglądał analogicznie, lecz w danym momencie jedna z maszyn w chmurze stawała się serwerem, reszta klientami. Użyte komendy to:

- 1) uruchomienie *iperf* w trybie serwera:

```
PS C:\iperf> .\iperf.exe -s -p 1500
-----
Server listening on TCP port 1500
TCP window size: 64.0 KByte (default)
-----
```

- 2) uruchomienie *iperf* w trybie klienta:

```
PS C:\iperf> .\iperf.exe -c -p 1500 salamontest1.cloudapp.net
```

Program można wywołać w taki sposób, aby symulował jedno połączenie transferujące dużą ilość danych lub wiele połączeń przesyłających małe informacje. Dane w tabeli 8 obejmują średnią z trzech kolejnych badań wysyłających/odbierających kolejno 1 MB, 6 MB, 25 MB.

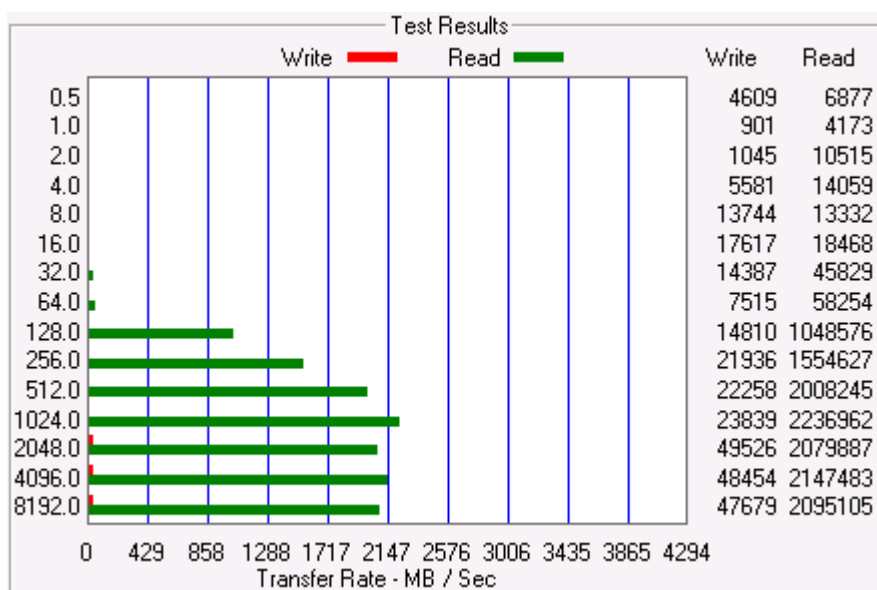
Tabela 8

Uśrednione wyniki pomiaru szybkości transferu danych pomiędzy serwerami wyrażone w megabitach na sekundę [Mbps]

z/do	Klient	salamontest1	salamontest2	salamontest3	salamontest4
Klient		4,12	4,14	1,82	0,69
salamontest1	3,396		4,84	3,76	1,23
salamontest2	3,65	4,86		3,56	1,15
salamontest3	2,32	3,97	3,63		1,63
salamontest4	0,829	1,24	1,19	1,62	

### 4.3. Operacje odczytu oraz zapisu

Sprawdzenie fizycznej szybkości dysku twardego opierało się na użyciu programu *ATTO Disk Benchmark*. Zapisuje oraz odczytuje on z dysku dane o różnym rozmiarze, zaczynając od 0,5 KB, a kończąc na 8 MB. Wykres przedstawiający rezultaty testu jest widoczny na rysunku 2. Zawiera on dwa rodzaje słupków: położony wyżej, w kolorze czerwonym (zapis na dysk), oraz położony niżej, w kolorze zielonym (odczyt). Dane poziome (oś X) przedstawiają szybkość transferu – jednostką są MB/s, oś pionowa, ułożona w formie listy, przyporządkowuje kolejno wielkości zapisywanych/odczytywanych na dysku paczek danych. Interpretując otrzymane wyniki, możemy zwrócić uwagę na różnicę w rzędzie wielkości zapisu oraz odczytu na korzyść tego ostatniego. Szybkość zapisu odpowiada osiągom przeciętnego dysku „talerzowego” SATA. Odczyt natomiast sięga bardzo wygórowanego poziomu najwyższej klasy dysków SSD (w praktyce są to macierze dysków tradycyjnych).



Rys. 2. Wyniki testu odczytu oraz zapisu dla architektury bardzo małej  
Fig. 2. The results of read and write tests in very small architecture

#### 4.4. Menedżer ruchu

Sprawdzanie Menedżera ruchu platformy Windows Azure należy rozpocząć od jej odpowiedniego skonfigurowania. Na platformie zostały utworzone cztery maszyny wirtualne wykorzystujące system Microsoft Windows Server 2012 Datacenter. Na każdej z nich został zainstalowany moduł IIS wraz z serwerem sieci Web obsługującym PHP. Wszystkie instancje zawierały plik `index.php` o następującej treści:

```
<?php
$domena_name="salamontestX.cloudapp.net";

// ODCZYT WARTOSCI
$plikbazy="baza.txt";
$baza = fopen($plikbazy, "r");
$wartosc = @fgets($baza);
fclose($baza);

// ZAPIS
$wartosc+=1;
$baza = fopen($plikbazy, "w");
fputs($baza, $wartosc);
fclose($baza);

// HTML
echo "<html>";
echo "<head> <meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>
</head><body>";
echo "<p align='center'><font size='5px'> Witaj na ". $domena_name . "
</font></p>";
echo "<p align='center'><font size='5px'> odwiedzone nas juz " . $wartosc . " ra-
zy</font></p>";
echo "</body></html>";
?>
```

Prezentowany skrypt pełni funkcję bardzo prostego licznika odwiedzin. Każde wyświetlenie strony powoduje inkrementację wartości znajdującej się w pliku bazy. Skrypt jednocześnie wyświetla w oknie przeglądarki nazwę serwera.

Kolejnym krokiem było stworzenie Menedżera oraz dodanie do niego wszystkich serwerów. Ważnym parametrem była wartość DNS TTL wynosząca standardowo 5 minut. Została ona ustawiona na minimalną dozwoloną wartość, tj. 30 sekund. Przekłada się ona na częstsze aktualizacje bazy nazw po stronie klienta. Obciążenie było generowane z kilku źródeł:

- komputer podłączony do łącza światłowodowego o parametrach 100 Mbps/25 Mbps (Polska),
- komputer podłączony do łącza o parametrach 25 Mbps/25 Mbps (Francja),
- konto wygenerowane w portalu *loader.io*, znajdujące się w sieci firmy Amazon (Stany Zjednoczone). W związku z licencją ograniczającą możliwości stosowania narzędzia nie było ono używane we wszystkich testach.

Komputery klienckie pracowały pod kontrolą systemu operacyjnego *Debian* z rodziny *Linux*. Generacja obciążenia odbywała się przez wydanie komendy:

```
for i in `seq 1 X`; do wget http://adresSerwera/ -O index.php; rm index.php;
done
```

Jej działanie można opisać w trzech krokach:

- 1) pobierz główny plik (index.php) z serwera *adresSerwera*,
- 2) usuń pobrany plik index.php,
- 3) wykonaj punkty 1 i 2 w pętli od 1 do X razy.

Po każdej serii prób pliki *baza.txt* były weryfikowane, a ich zawartość była zapisywana. Pojedynczy test trwał około 5 minut.

#### 4.4.1. Praca awaryjna

Praca awaryjna jest rodzajem Menedżera ruchu, który jako jedyny zawiera dodatkową opcję konfiguracyjną. Określa się w nim kolejność, w jakiej zapytania mają być kierowane do poszczególnych maszyn. Oznacza to sytuację, w której maszyna z największym priorytetem obsługuje cały ruch tak długo, jak pozostaje ona online. Zaraz po ustawieniu trybu pracy awaryjnej serwer *salamontest1*, mający największy priorytet, przejął obsługę nad całym ruchem. Wyniki są zatem zgodne z oczekiwaniami.

Dodatkowym zadaniem było zbadanie zachowania klastra serwerów przez ponowne uruchomienie pierwszej maszyny w trakcie trwania testu (około 10 minut). Tym razem było to dwa tysiące zapytań. Menedżer zauważył, że instancja została wyłączona po średnim czasie 2 minut i 19 sekund. Kolejne półtorej minuty trwało, zanim komputer uaktualnił swoją bazę nazw – razem, do momentu przełączenia domeny z serwera pierwszego na drugi, upłynęły średnio 3 minuty i 51 sekund. W tym czasie serwis pozostawał niedostępny. Po ponownym uruchomieniu maszyny jej przejście w tryb serwera głównego trwało średnio 1 minutę i 11 sekund.

W przypadku trybu „praca awaryjna” nie udało się znaleźć żadnych zachowań niepotwierdzających zasady, według której działa ten tryb. Podczas próby z poziomu serwerów w chmurze każda z nich zawsze wskazuje na tę z najwyższym priorytetem. Jedynym zauważonym mankamentem jest czas przełączania. Standardowe narzędzie wbudowane w platformę nie jest w stanie wykonać tego szybciej. Według dokumentacji stan maszyn jest sprawdzany co pięć minut. Czas przełączania może być w najlepszym przypadku natychmiastowy – czas życia pakietu DNS wygaśł dokładnie w momencie sprawdzenia stanu maszyn oraz awarii tej pierwszej. Jest to jednak przypadek skrajnie nieprawdopodobny. Maksymalny czas 5 minut i 30 sekund osiągniemy, gdy maszyna ulegnie awarii dokładnie sekundę po przeskanowaniu stanu urządzeń. Dla Menedżera urządzenie pierwsze wciąż pozostanie online, zapytania DNS ciągle będą na nią wskazywać. Dopiero po upływie 5 minut oraz odczekaniu maksymalnego czasu życia pakietu DNS komputer klienta będzie mógł dalej korzystać z usługi.

#### 4.4.2. Równoważenie obciążenia

Ten tryb pracy Menedżera w idealnym przypadku powinien każdego z trzech klientów skierować na inną maszynę. Z innej strony w przypadku jednego klienta ruch powinien być rozłożony równomiernie. Przeprowadzono dwa testy:

- 1) tylko jeden klient wysyłał zapytania w danym momencie,
- 2) wszyscy klienci wysyłali zapytania w tym samym czasie.

W obydwu przypadkach reakcja Menedżera jest podobna: dzieli on ruch niemalże po równo (tabela 9 i tabela 10). Trzeba jednak zauważyć, iż w przypadku testów europejskich (Polska i Francja) najbardziej obciążone są maszyny znajdujące się najbliżej pod względem geograficznym. Podobną tendencję można dostrzec w przypadku zapytań ze Stanów Zjednoczonych, gdzie serwer trzeci obsługuje ich największą część. Na potwierdzenie tej tezy można przytoczyć wyniki serwera azjatyckiego – w każdym przypadku bierze on najmniejszy udział w obsłudze klientów.

Tabela 9

Wyniki testu Menedżera ruchu w trybie równoważenie obciążenia cz. I

Klient	salamontest1 (Europa Północna)	salamontest2 (Europa Zachodnia)	salamontest3 (USA Wchód)	salamontest4 (Azja Wschód)
Polska	406	358	220	16
Francja	332	336	255	77
USA	254	209	411	126

Tabela 10

Wyniki testu Menedżera ruchu w trybie równoważenie obciążenia cz. II

Wszyscy klienci	salamontest1 (Europa Północna)	salamontest2 (Europa Zachodnia)	salamontest3 (USA Wchód)	salamontest4 (Azja Wschód)
	816	944	899	341

#### 4.4.3. Największa wydajność

Tabela 11

Wyniki testu Menedżera ruchu w trybie największa wydajność

Klient	salamontest1 (Europa Północna)	salamontest2 (Europa Zachodnia)	salamontest3 (USA Wchód)	salamontest4 (Azja Wschód)
Polska	0	1000	0	0
Francja	0	0	1000	0
USA	0	0	1000	0

Największa wydajność oznacza wybór przez program Menedżer ruchu serwera będącego najszybszym z punktu widzenia klienta. Próba polegała na kolejnym wysłaniu przez każdego z nich tysiąca zapytań. W przypadku Windows Azure ta opcja Menedżera ruchu pokazała, że na początku połączenia zostaje ustalona lokalizacja najbardziej optymalna i tylko z nią nawiązywany jest kontakt (tabela 11). Potwierdzeniem może być to, że każda maszyna przywoływania adresu Menedżera ruchu zawsze była kierowana do samej siebie. Dziwnym

zachowaniem jest natomiast wybranie przez francuskiego klienta serwera znajdującego się w Stanach Zjednoczonych.

## 5. Podsumowanie

W artykule skupiono uwagę na zagadnieniach dotyczących chmury obliczeniowej, która za pomocą istniejącej technologii sprzętowej i już gotowego oprogramowania stwarza nowatorski twór wpasowujący się idealnie w potrzeby współczesnych użytkowników. Na podstawie przeprowadzonej analizy można powiedzieć, że *Cloud computing* sprawdzi się świetnie we wszystkich nieprzewidywalnych sytuacjach, w których w krótkim odstępie czasu potrzeba sporych zasobów różnego typu. Używając odwróconej logiki, można stwierdzić, że rozwiązanie nie jest odpowiednie, przynajmniej nie zawsze, w sytuacjach przewidywalnych. Nieopłacalne jest przenoszenie lub tworzenie w chmurze infrastruktury, która jest niezmienna w czasie, a korzysta z niej stale taka sama liczba użytkowników. Natomiast biorąc pod uwagę skomplikowanie, położenie geograficzne potencjalnych użytkowników, indywidualność każdego przypadku oraz liczbę czynników, które mają wpływ na pożądaną efekt, nie da się jednoznacznie odpowiedzieć na pytanie, czy przekazanie odpowiedzialności na inne podmioty za pomocą chmury obliczeniowej jest rozwiązaniem lepszym czy gorszym od zastosowań wewnątrzfirmowych.

Studując wiedzę na temat *Cloud computingu*, szczególną uwagę skupiono na platformie Windows Azure firmy Microsoft. Na jej przykładzie można powiedzieć, że pomimo przejścia przez technologie tak dalekiej drogi, rozwój i selekcja nadal trwają. Firmy wciąż ulepszają istniejące rozwiązania, tworząc jednocześnie nowe. Chociaż Windows Azure istnieje na rynku już kilka lat, to od jego producenta wciąż słyszy się, iż jest on dziełem nieskończonym. Każda konferencja przynosi dodatkowe cechy istniejącym produktom oraz zupełnie nowe rozwiązania. Biorą się one z niczego innego jak właśnie z badań rynku oraz dostosowywania się do potrzeb klienta.

Przeprowadzone w ramach tej pracy testy, jak również rozważania prowadzone w [7, 8] mogą stanowić element tego procesu. Omawiane w artykule eksperymenty są tylko przykładem tego, w jaki sposób można analizować nowoczesne rozwiązania chmury. Zbudowane w ramach badań środowisko będzie modyfikowane i dalej rozwijane pod kątem kolejnych testów. Planuje się stworzenie kompleksowego modelu serwerów i usług w Windows Azure, gotowego do użytku wewnątrz konkretnej infrastruktury firmowej. Początkowo będzie to rozwiązanie równoległe do istniejącego, z czasem przejmie coraz większą liczbę zadań. Na podstawie obserwacji sposobu ich obsługi będzie można wybrać konfigurację najbardziej optymalną.

**BIBLIOGRAFIA**

1. Pęczak A.: PCWord. Nowy wymiar hostingu. [Online] 20 sierpnia 2013: <http://www.pcworld.pl/news/391664/Nowy.wymiar.hostingu.html>, 2013.
2. Schmidt E.: Search Engine Strategies Conference. Conversation with Eric Schmidt hosted by Danny Sullivan. [Online] 9 sierpnia 2006: <http://www.google.com/press/podium/ses2006.html>, 2013.
3. Google: Google Trends. Wyszukiwarka Google – zainteresowania: cloud computing, windows azure, google app engine, amazon ec2. Cały świat, 2004 – dziś. [Online] 4 września 1998: <https://www.google.com/trends>, 2013.
4. Microsoft Corporation: Windows Azure documentation. Windows Azure. [Online] 2009: <http://www.windowsazure.com/en-us/documentation>, 2013.
5. Klems M.: Cloud Times Random Thoughts of Mark Klems. [Online] 2008: <https://markusklems.wordpress.com/>, 2013.
6. Jennings R.: Cloud Computing with the Windows Azure Platform. Wiley Publishing Inc., Indianapolis 2009.
7. Bajerski P., Augustyn D. R., Bach M., Brzeski R., Duszeńko A., Werner A.: Bazy danych a chmury obliczeniowe. *Studia Informatica*, Vol. 33, No. 2A, Gliwice 2012, s. 9÷25.
8. Frączek J., Zając Ł.: Analiza wydajności przetwarzania danych w chmurze obliczeniowej Windows Azure. *Studia Informatica*, Vol. 34, No. 2A, Gliwice 2013, s. 97÷112.
9. Urząd Regulacji Energetyki: Średnia cena energii elektrycznej dla gospodarstw domowych. [Online] 1998: <http://www.ure.gov.pl/pl/urząd/informacje-ogolne/aktualnosci/5437,Srednia-cena-energii-elektrycznej-dla-gospodarstw-domowych.html>, 2013.
10. Speedtest.pl: Speed Test – test prędkości łącza. [Online] 2006: <http://www.speedtest.pl>, 2013.
11. Hewlett-Packard Development Company: HP ProLiant DL165 G7 Server. [Online] <http://www8.hp.com/us/en/products/proliant-servers/product-detail.html?oid=4091396#!tab=features>, 2013.
12. Dell: PowerEdge R515 2U Rack Server. [Online] 2013: <http://www.dell.com/us/business/p/poweredge-r515/pd?refid=poweredge-r515&isredir=true>, 2013.
13. Dell: PowerEdge R415 1U Rack Server Details. [Online] 2013: <http://www.dell.com/us/business/p/poweredge-r415/pd?refid=poweredge-r415&isredir=true>, 2013.
14. 3S Śląskie Sieci Światłowodowe: Kalkulator Usług – Internet. [Online] 2012: <http://www.3s.pl/services/c,46,kalkulator-uslug/internet.html>, 2013.
15. CBS Interactive Inc.: AMD Opteron 4171 HE / 2.1 GHz processor. Shopper.com – Compare prices, reviews, user opinions. [Online] 2011: [http://shopper.cnet.com/processors/amd-opteron-4171-he/4014-3086\\_9-34912913.html](http://shopper.cnet.com/processors/amd-opteron-4171-he/4014-3086_9-34912913.html), 2013.

Wpłynęło do Redakcji 31 stycznia 2014 r.

## **Abstract**

Issues concerning cloud computing, which uses existing hardware and software technology and creates a new useful tool for ordinary computer users were analysed in the paper. The analysis regarded cloud computing models and areas of their usage. The various models used in cloud computing in terms of responsibility for creating and managing an infrastructure were discussed (table 1 and 2).

Special focus was given to one of the popular platforms – Windows Azure. The costs of its usage were compared to expenses needed to be paid in case of developing own infrastructure (table 3). The findings completely discredit own solution even in the long run, especially in the most expensive case (A7 architecture).

An environment based on Windows Azure platform was designed and configured (table 4) and several experiments were conducted using it. They included processor performance tests consisting of two tasks: graphics rendering and calculating number known as Pi (chapter 4.1). Additionally the quality of file transfer and the speed of writing and reading a file were taken into account (chapters 4.2 and 4.3). There were several tests considering different transfer configurations analyzed. Finally, it was studied how Traffic Manager organizes load balance of incoming traffic across multiple hosted services. (chapter 4.4).

## **Adresy**

Tomasz SALAMON: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-100 Gliwice, Polska, salamon.tomek@gmail.com.

Katarzyna HAREŹLAK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-100 Gliwice, Polska, katarzyna.harezak@polsl.pl.