

Jacek FRĄCZEK
Politechnika Śląska, Instytut Informatyki
Łukasz ZAJĄC
LGBS Polska Sp. z o.o.

ANALIZA WYDAJNOŚCI PRZETWARZANIA DANYCH W CHMURZE OBLICZENIOWEJ WINDOWS AZURE

Streszczenie. W artykule przedstawiono wyniki badań wydajnościowych dla rozwiązania problemu komiwojażera z wykorzystaniem równoległej wersji algorytmu genetycznego w środowisku wielordzeniowego serwera lokalnego oraz chmury obliczeniowej Windows Azure. Przeprowadzona analiza wyników umożliwia również dobranie odpowiedniej do określonych zastosowań konfiguracji serwerów w chmurze.

Słowa kluczowe: chmura obliczeniowa, Windows Azure, problem komiwojażera, TSP, równoległy algorytm genetyczny

DATA PROCESSING PERFORMANCE ANALYSIS IN WINDOWS AZURE CLOUD

Summary. The paper presents the results of performance tests for solving the traveling salesman problem with the use of a parallel version of a genetic algorithm on a local multi-core server, and in the Windows Azure cloud. The analysis of the obtained results allows the selection of a suitable configuration of the servers in the cloud.

Keywords: cloud computing, Windows Azure, traveling salesman problem, TSP, parallel genetic algorithm

1. Wstęp

Obecnie na rynku dostępna jest bardzo szeroka oferta usług przetwarzania danych w środowisku chmury obliczeniowej (ang. *Cloud Computing*), świadczonych m.in. przez takie firmy, jak: Amazon (Amazon Web Services), Rackspace, CenturyLink/Savvis, Salesforce.com,

Verizon/Terremark, Joyent, Citrix, Bluelock, Microsoft, VMware. Powszechność i dostępność usług przetwarzania danych w chmurze powoduje, że coraz więcej przedsiębiorstw oraz osób prywatnych z nich korzysta, a jak wskazuje artykuł [1] – wartość rynku tego typu usług będzie się w najbliższych latach znacząco zwiększała.

Zastosowanie chmury obliczeniowej może być szczególnie korzystne dla przedsiębiorstw, które potrzebują w krótkim czasie uzyskać dostęp do systemu komputerowego dysponującego ponadprzeciętną mocą obliczeniową. W takim przypadku wykorzystanie usług przetwarzania w chmurze może być bardziej opłacalne i szybsze w realizacji niż wykorzystanie posiadanych lub zakup nowych zasobów własnych (serwerów lub klastrów obliczeniowych). Jednak przetwarzanie w chmurze niesie z sobą ryzyko związane przede wszystkim z bezpieczeństwem danych oraz zapewnieniem odpowiedniej dostępności usług w czasie (w tym i nieprzerwanej, szybkiej transmisji danych), na które dodatkowo nakładają się często niejasne uwarunkowania prawne związane z możliwościami przetwarzania danych podmiotu gospodarczego przez inny podmiot, często zlokalizowany w innym kraju.

W artykule przedstawiono analizę wydajności przetwarzania zadań wymagających dużej mocy obliczeniowej na wielordzeniowym serwerze lokalnym oraz w chmurze obliczeniowej. W ramach przeprowadzonych testów badano czasy rozwiązania problemu komiwojażera z wykorzystaniem zrównoleglonej wersji algorytmu genetycznego dla:

- pojedynczych zadań różnej wielkości (zróżnicowanie liczby miast),
- zestawów zadań różnej wielkości

uruchamianych dla różnej liczby wątków:

- na serwerze lokalnym,
- w różnych konfiguracjach serwerów chmury obliczeniowej Microsoft Windows Azure (zróżnicowanie typów i liczby serwerów przetwarzających).

Dodatkowo w artykule zbadano zmiany w zachowaniu się serwisu zlokalizowanego w chmurze przy zmiennej liczbie użytkowników, którzy chcą z niego skorzystać. Uzyskane wyniki można wykorzystać jako wsparcie w podjęciu odpowiedniej decyzji dotyczącej sposobu zapewnienia przedsiębiorstwu wymaganego poziomu usług IT.

Część opublikowanych tu wyników pochodzi z pracy dyplomowej [2] napisanej przez Ł. Zając pod kierunkiem J. Frączka.

2. Przegląd literatury

Pierwsze koncepcje odpowiadające dzisiejszym rozwiązaniom przetwarzania w chmurze pojawiły wraz z wprowadzeniem na rynek komputerów typu *mainframe* już w latach 50.

XX wieku. Mimo tak długiej historii problem wydajności pracy nowoczesnych chmur obliczeniowych jest rozpatrywany w literaturze stosunkowo od niedawna, gdyż pierwsze usługi komercyjne tego typu (Amazon Web Service) pojawiły się na rynku w 2006 roku.

W pracy [3] przeprowadzono analizę zachowania się klastra zbudowanego z wieloprocesorowych maszyn zwirtualizowanych z użyciem *open source*'owego monitora maszyn wirtualnych Xen. Uzyskane rezultaty wskazywały na możliwość współdzielenia zwirtualizowanych wieloprocesorowych węzłów przez wiele aplikacji, bez istotnej utraty wydajności związanej z koniecznością komunikacji pomiędzy węzłami.

W pracy [4] przeprowadzono badania zachowania się, bezpieczeństwa oraz wydajności dostępu do danych przechowywanych w chmurze Simple Storage Service (S3) firmy Amazon. Podobne badania dla platformy Windows Azure Storage opisano w pracy [5].

W publikacji [6] przedstawiono stosunkowo szerokie badania wydajnościowe chmury Microsoft Windows Azure obejmujące: pracę z usługami przechowywania danych (obiekty binarne, kolejki, tabele), szybkość tworzenia maszyn wirtualnych w chmurze, wydajność transmisji TCP pomiędzy węzłami zlokalizowanymi w chmurze oraz wydajność przetwarzania poleceń SQL przez serwer bazy danych SQL Azure.

Badania przeprowadzone w wymienionych wyżej pracach nie obejmowały zagadnień wydajności przetwarzania informacji przez aplikacje zlokalizowane w chmurze obliczeniowej.

W pozycji [7] przeprowadzono analizę wykorzystania chmury publicznej Windows Azure oraz chmury prywatnej Nimbus do zastosowań naukowych. Wykonane kompleksowe badania obejmowały weryfikację oferowanej mocy przetwarzania, możliwości przechowywania i przesyłu danych oraz analizę kosztów wykorzystania tych usług. Interesująca – z punktu widzenia prezentowanej pracy – analiza wydajności przetwarzania w chmurze dotyczyła rzeczywistej aplikacji A-Brain służącej do analizy wyników neuroobrazowania mózgu i mapowania genetycznego. Jak piszą autorzy: *aplikacja ta jest reprezentatywna dla szerokiej klasy problemów naukowych, które dzielą wstępne zagadnienie na podzagadnienia, wykonują obliczenia (np. operacje na macierzach) i łączą wyniki końcowe (np. przetwarzanie obrazów, symulacje pogodowe, itd.)*. Autorzy – komentując rezultaty swoich prac – stwierdzili, że lepsze wyniki chmury prywatnej Nimbus są pochodną bardziej wydajnego, specjalistycznego sprzętu użytego do jej implementacji, a chmury publiczne *zazwyczaj bazują na ogólnodostępnym sprzęcie, który jest mniej wydajny w zastosowaniach naukowych*.

Wartością dodaną prezentowanej pracy są uzyskane nowe wyniki badań różnych aspektów wykorzystania technologii chmury obliczeniowej:

- porównanie wydajności realizacji czasochłonnych algorytmów w stosunku do fizycznego serwera lokalnego,

- zbadanie sposobu zachowania się chmury obliczeniowej w przypadku algorytmów wykorzystujących różne liczby wątków,
- zbadanie sposobu zachowania się chmury obliczeniowej zależnie od liczby użytkowników, którzy chcą skorzystać z oferowanych usług.

3. Chmura obliczeniowa Windows Azure

Windows Azure jest nazwą usługi publicznej chmury obliczeniowej oferowanej przez firmę Microsoft. Windows Azure udostępnia usługi w modelu *Platform as a Service* (PaaS) oraz *Infrastructure as a Service* (IaaS), podczas gdy usługi *Software as a Service* (SaaS) oferowane są przez platformę Microsoft Online Services.

Platforma Windows Azure składa się z trzech części [8]:

- Windows Azure – podstawowy element platformy umożliwiający publikowanie usług oraz aplikacji w sieci, przechowywanie oraz przetwarzanie danych przy zapewnieniu możliwości skalowania rozwiązania w zależności od potrzeb,
- Windows Azure SQL Database – baza danych korzystająca z Microsoft SQL Server,
- AppFabric – moduł wspierający integrację aplikacji z innymi rozwiązaniami, odpowiedzialny za komunikację oraz kontrolę dostępu.

W ramach prezentowanego artykułu wykorzystana została jedynie podstawowa część Windows Azure, dzięki której możliwe było wykonanie wszystkich wymaganych testów wydajnościowych.

Tabela 1

Specyfikacja serwera testowego i serwerów dostępnych Windows Azure (wg [9])

Serwer	Liczba rdzeni	Częstotliwość taktowania [GHz]	Ilość pamięci [GB]
Lokalny	8	2,5	16,00
WA bardzo mały (<i>Extra Small</i>)	CPU współdzielony	1,0	0,768
WA mały (<i>Small</i>)	1	1,6	1,75
WA średni (<i>Medium</i>)	2	1,6	3,50
WA duży (<i>Large</i>)	4	1,6	7,00
WA bardzo duży (<i>Extra Large</i>)	8	1,6	14,00

Platforma Windows Azure umożliwia utworzenie 5 klas maszyn wirtualnych o zróżnicowanych parametrach (liczba rdzeni, częstotliwość taktowania rdzenia, ilość pamięci, wydajność kanału we/wy – zestawienie cech pokazano w tabeli 1). W przeprowadzonych testach wydajnościowych nie badano zachowania się najsłabszego serwera klasy *Extra Small*.

4. Lokalny serwer testowy

Do badań w charakterze serwera lokalnego wykorzystano fizyczny serwer wyposażony w dwa 4-rdzeniowe procesory Intel Xeon E5420 2,5 GHz oraz 16 GB pamięci RAM. Liczba rdzeni serwera (8) odpowiadała maksymalnej liczbie rdzeni, którą oferuje Microsoft Azure (serwer klasy *Extra Large*). Komputer działał pod kontrolą systemu operacyjnego Microsoft Windows Server 2008 R2. Dostęp do serwera zapewniało stabilne łącze internetowe.

5. Algorytm i dane testowe

W ramach przeprowadzonych testów badano czas rozwiązania problemu komiwojażera z wykorzystaniem zrównoleglonej wersji algorytmu genetycznego. W schemacie działania algorytmu genetycznego czasochłonną operacją jest wytworzenie nowej populacji, która tworzy podlegający ewolucji zbiór potencjalnych rozwiązań [10]. Operacja przygotowania nowego pokolenia obejmuje takie czynności, jak selekcje, krzyżowanie, mutacje oraz wyliczanie wartości funkcji oceny. Przygotowana implementacja algorytmu rozwiązującego problem komiwojażera ma możliwość równoległego wykonywania części obliczeń związanych z procesem tworzenia nowej populacji. Proces ten można rozdzielić na zadaną liczbę wątków. Wyniki uzyskane z poszczególnych wątków są następnie scalane. Innymi parametrami wpływającymi na sposób pracy przygotowanego algorytmu są: liczba generacji, rozmiar populacji i prawdopodobieństwo mutacji.

Dane wykorzystane w testach pochodzą z tzw. biblioteki TSPLib [11] – referencyjnego zbioru przykładowych zadań dotyczących problemu komiwojażera (oraz pokrewnych) utrzymanego na Uniwersytecie w Heidelbergu (Niemcy). Dane przechowywane w TSPLib są danymi rzeczywistymi i obejmują listy punktów adresowych wybranych państw oraz ich współrzędnych geograficznych. Dla przykładowych zbiorów danych biblioteka TSPLib zawiera również odpowiadające im najlepsze znane do tej pory rozwiązania problemów wraz z ilustracjami w postaci mapy.

6. Specyfikacja przeprowadzonych testów wraz z wynikami

W ramach przeprowadzonych testów badano czas rozwiązania problemu komiwojażera z wykorzystaniem zrównoleglonej wersji algorytmu genetycznego w środowiskach:

- 1) serwera lokalnego dysponującego 8 rdzeniami procesora (2 procesory 4-rdzeniowe),

2) chmury obliczeniowej Windows Azure, przy zastosowaniu zmiennej liczby serwerów (1-4 maszyn) różnych typów (liczba rdzeni procesora 1, 2, 4, 8).

Opublikowane w rozdziale wyniki czasu trwania testów podano w milisekundach. Wszystkie badania zostały powtórzone trzykrotnie, a przedstawione w artykule rezultaty są średnią z uzyskanych wyników.

6.1. Badanie wydajności realizacji pojedynczych zadań z użyciem jednego serwera

Ten rodzaj badań polegał na uruchomieniu pojedynczych zadań różniących się parametrami konfiguracyjnymi algorytmu genetycznego oraz rozmiarem danych wejściowych. W przypadku chmury obliczeniowej Windows Azure wykorzystywano pojedynczy serwer, natomiast zmianie ulegała jego klasa (co wpływało m.in. na liczbę rdzeni procesora).

6.1.1. Badanie wydajności różnych klas serwerów w zależności od rozmiaru problemu

W tej grupie testów wykorzystano następujące zbiory danych z biblioteki TSPLib:

- zbiór danych Sahara Zachodnia: 29 punktów adresowych,
- zbiór danych Katar: 194 punkty adresowe,
- zbiór danych Luksemburg: 980 punktów adresowych,
- zbiór danych Oman: 1979 punktów adresowych,
- zbiór danych Egipt: 7 146 punktów adresowych,
- zbiór danych Maroko: 14 185 punktów adresowych,

We wszystkich testach przyjęto identyczne parametry konfiguracyjne algorytmu genetycznego:

- liczba generacji: 1 000,
- rozmiar populacji: 4 000,
- prawdopodobieństwo mutacji: 10%.

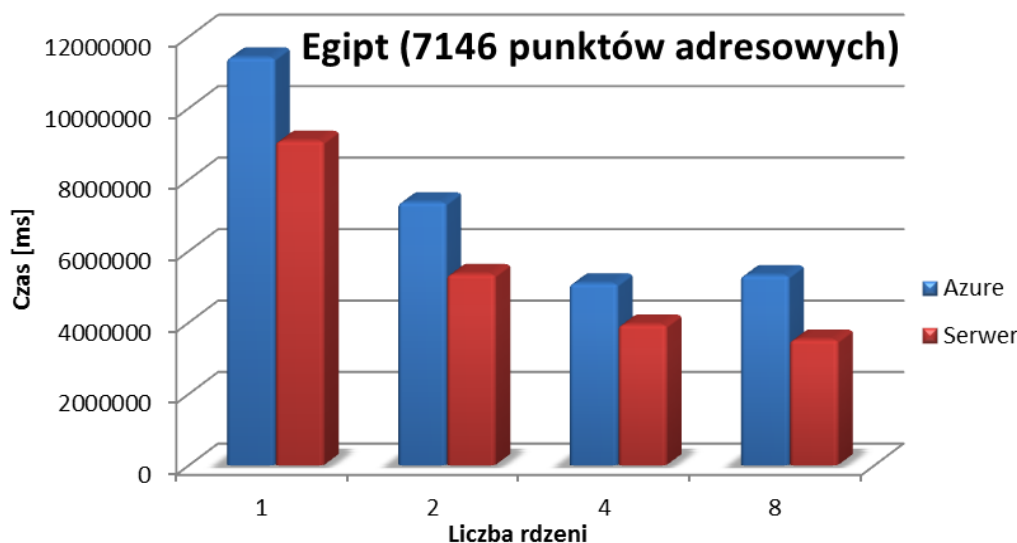
Każdy z testów został powtórzony zarówno na serwerze lokalnym, jak i w chmurze Windows Azure dla różnej liczby wątków, przy czym w przypadku chmury obliczeniowej wykorzystywano zawsze serwer o liczbie rdzeni odpowiadającej liczbie wątków użytych do rozwiązania zadań w ramach badania, tzn. dla zadania rozwiązywanego z użyciem czterech wątków wykorzystano serwer 4-rdzeniowy (maszyna klasy: *Large*).

Uzyskane wyniki dla zbioru danych Egipt przedstawiono w tabeli 2 i na rys. 1, a dla zbioru danych Maroko – w tabeli 3 i na rys. 2. Dla pozostałych, mniejszych zbiorów danych uzyskano wyniki analogiczne do wyników zebranych dla zbiorów Egipt i Maroko.

Tabela 2

Wyniki testu dla zbioru danych Egipt (7 146 punktów adresowych)

Liczba rdzeni	Czas serwer [ms]	Czas Azure [ms]	Względna różnica czasów w stosunku do czasu serwera
1	9080098	11412207	25,68%
2	5350444	7361770	37,59%
4	3938741	5092711	29,30%
8	3517274	5319960	51,25%



Rys. 1. Wyniki testu dla zbioru danych Egipt
 Fig. 1. Test results for Egypt dataset

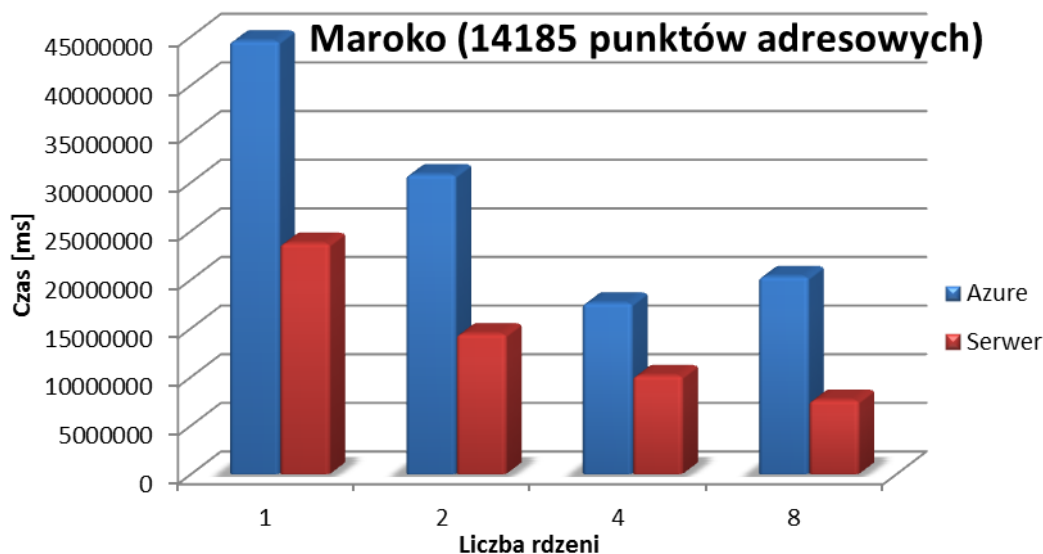
Tabela 3

Wyniki testu dla zbioru danych Maroko (14 185 punktów adresowych)

Liczba rdzeni	Czas serwer [ms]	Czas Azure [ms]	Względna różnica czasów w stosunku do czasu serwera
1	23746593	44554808	87,63%
2	14436680	30810981	113,42%
4	10144304	17606233	73,56%
8	7614888	20323790	166,90%

Analizując otrzymane wyniki, można jednoznacznie stwierdzić, że lokalny serwer fizyczny okazał się bardziej wydajny niż serwer wirtualny zlokalizowany w chmurze obliczeniowej. Fakt ten można wytłumaczyć posiadaniem przez serwer fizyczny procesorów o wyższej częstotliwości taktowania. Analizując uzyskane rezultaty, można również zauważyć, że wraz ze wzrostem złożoności problemu (liczby danych wejściowych) rosła przewaga serwera lokalnego nad serwerem zlokalizowanym w chmurze. Interesującym zjawiskiem jest fakt, że dla środowiska Windows Azure we wszystkich przeprowadzonych testach czasy rozwiązywania zadań przez serwer dysponujący 8 rdzeniami są gorsze niż czasy uzyskiwane przez serwer dysponujący 4 rdzeniami. Fakt ten wskazuje na prawdopodobne zastosowanie procesorów

4-rdzeniowych w fizycznych komputerach, na bazie których uruchomiono środowisko chmury Windows Azure w centrach przetwarzania danych Microsoft.



Rys. 2. Wyniki testu dla zbioru danych Maroko
Fig. 2. Test results for Morocco dataset

6.1.2. Badanie wydajności serwera klasy *Extra Large* w zależności od liczby wątków użytych w algorytmie

Tabela 4

Czasy wykonania testu na serwerze Windows Azure klasy *Extra Large* dla różnej liczby wątków

Liczba wątków	Czas [ms]
1	9 152 293
2	5 979 022
3	4 701 465
4	4 269 489
5	4 273 373
6	4 388 441
7	4 700 784
8	5 181 434

Ta grupa testów objęła badania czasów realizacji algorytmu testowego przez pojedynczy serwer Windows Azure klasy *Extra Large* (8-rdzeni) przy zmianach parametru określającego liczbę realizowanych równolegle wątków (1-8 wykorzystywanych rdzeni procesora serwera). Zastosowanie serwera określonej klasy umożliwiło zbadanie zachowania się środowiska Windows Azure w przypadku pracy wielowątkowej, przy wykluczeniu wpływu czynników związanych ze zróżnicowaniem konfiguracji różnych typów serwerów Windows Azure (np. rozmiar pamięci RAM, przetwarzanie operacji we/wy). Do testów użyto zbioru danych Egipt z biblioteki TSPLib (7146 punktów adresowych), a wszystkie testy przeprowadzono z wykorzystaniem identycznych parametrów konfiguracyjnych algorytmu genetycznego:

- liczba generacji: 1000,
- rozmiar populacji: 3360 (podzielny przez 2, 3, 4, 5, 6, 7, 8 – wszystkie rdzenie serwera będą obciążane równomiernie),
- prawdopodobieństwo mutacji: 10%.

Wyniki przeprowadzonych testów przedstawiono w tabeli 4 oraz na rys. 3.



Rys. 3. Czasy wykonania testu na serwerze Windows Azure klasy *Extra Large* dla różnej liczby wątków

Fig. 3. The time of the test, on a Windows Azure *Extra Large* server for a various number of threads

Analizując wykres przedstawiony na rys. 3, można zauważyć, że przy użyciu większej liczby wątków niż 4 (5-8) podczas wykonywania tego samego testu wraz ze wzrostem liczby wątków wzrasta również czas realizacji obliczeń. Dla 8 wątków uzyskany czas jest dłuższy niż czas uzyskany dla 3 wątków. Można stąd wnioskować, że Windows Azure gorzej radzi sobie z wykonywaniem zadań przy wykorzystaniu więcej niż 4 wątków. Fakt ten potwierdza wcześniejszy wniosek dotyczący rodzaju procesorów zastosowanych w serwerach używanych w centrach przetwarzania danych Microsoft.

6.2. Badanie wydajności pracy wielu serwerów w chmurze

Ta grupa testów objęła badania czasu realizacji zadań przez zespół serwerów (klaster) zlokalizowanych w chmurze obliczeniowej. W tym celu w środowisku Windows Azure uruchomiono wiele serwerów określonej klasy, które były w stanie jednocześnie przetwarzać dostarczane zadania.

Aby uzyskać wyniki, które będzie można z sobą porównać, zdecydowano się na realizację testów w następujących konfiguracjach dysponujących sumarycznie 8 rdzeniami procesora(-ów):

- lokalny serwer fizyczny: 8 rdzeni,
- 2 serwery Windows Azure klasy *Large*: 4-rdzeniowe (w sumie 8 rdzeni),
- 4 serwery Windows Azure klasy *Medium*: 2-rdzeniowe (w sumie 8 rdzeni),
- 8 serwerów Windows Azure klasy *Small*: 1-rdzeniowych (w sumie 8 rdzeni).

Zadania przekazywane do rozwiązania w Windows Azure były umieszczane w kolejce, a następnie były one z niej kolejno pobierane przez serwery, które już zakończyły przetwarzanie poprzednich zadań. Ta metoda pracy w naturalny sposób wprowadziła do architektury systemu mechanizm równoważenia obciążenia poszczególnych serwerów.

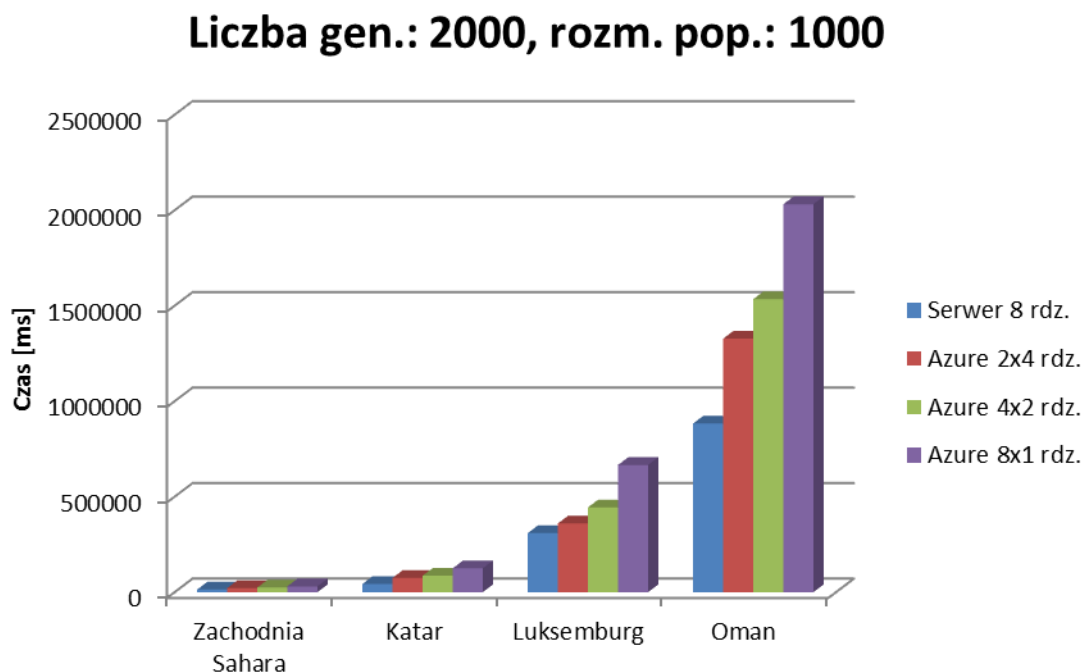
W ramach badań poszczególnych konfiguracji przeprowadzono wiele testów obejmujących różne zestawy danych wejściowych oraz różne parametry algorytmu testowego (przy stałym prawdopodobieństwie mutacji: 10%). Warunki przeprowadzenia testów oraz uzyskane wyniki przedstawiono w tabeli 5.

Tabela 5

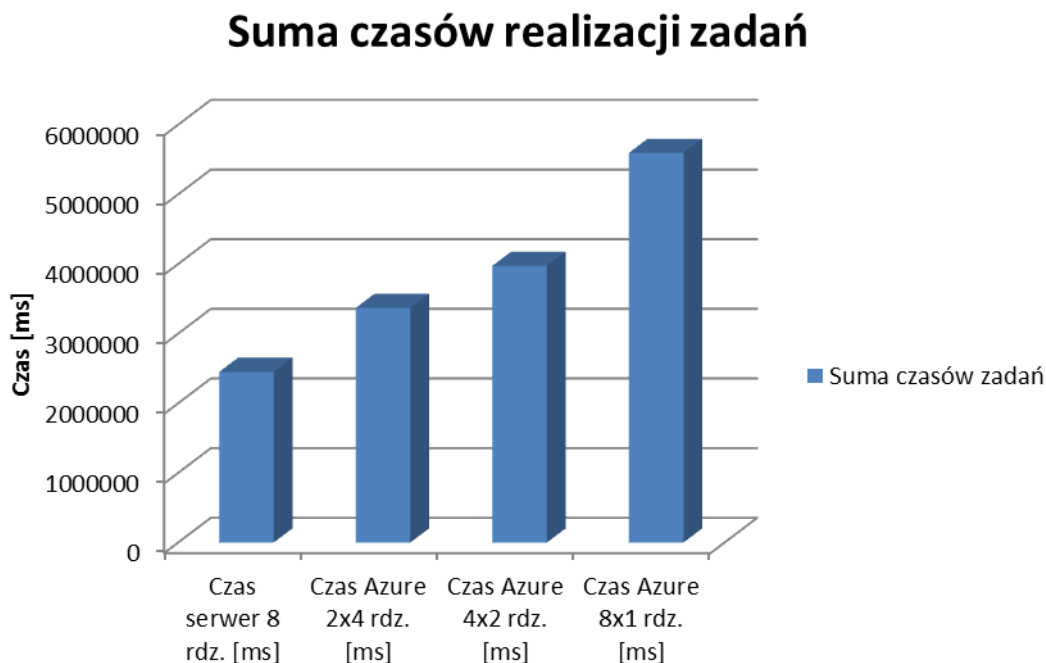
Czasy wykonania testów w środowiskach o identycznej liczbie rdzeni (8)

Państwo	Liczba generacji	Rozmiar populacji	Czas serwer 2x4R [ms]	Czas Azure 2x4R [ms]	Czas Azure 4x2R [ms]	Czas Azure 8x1R [ms]
Zachodnia Sahara	100	200	392	1 195	1 606	1 599
Zachodnia Sahara	200	100	570	1 247	905	1 470
Zachodnia Sahara	1 000	2 000	12 436	22 219	24 904	31 748
Zachodnia Sahara	2 000	1 000	14 018	21 790	24 495	30 650
Katar	100	200	516	1 440	2 392	2 305
Katar	200	100	729	1 566	2 267	3 361
Katar	1 000	2 000	49 028	75 518	91 827	126 514
Katar	2 000	1 000	43 342	74 933	86 729	124 910
Luksemburg	100	200	3 067	4 370	6 365	7 342
Luksemburg	200	100	2 860	4 656	6 201	6 826
Luksemburg	1 000	2 000	303 279	361 895	454 961	683 939
Luksemburg	2 000	1 000	309 615	360 070	444 068	666 353
Oman	100	200	12 560	19 982	19 523	22 099
Oman	200	100	10 816	16 909	16 402	18 982
Oman	1 000	2 000	803 807	1 073 349	1 254 892	1 836 326
Oman	2 000	1 000	883 466	1 327 650	1 534 382	2 031 543
		Suma czasów działania rdzeni procesora	2 450 502	3 368 789	3 971 921	5 595 968
		Bezwzględny czas realizacji wszystkich eksperymentów	2 450 530	1 862 222	1 755 195	2 270 585

Na rys. 4 przedstawiono wykres czasów wykonania testów dla wybranego przypadku parametrów algorytmu testowego (liczba generacji: 2 000, rozmiar populacji: 1 000).



Rys. 4. Czasy wykonania testów w środowiskach o identycznej liczbie rdzeni (8)
 Fig. 4. The time of the tests in environments with the same number of cores (8)



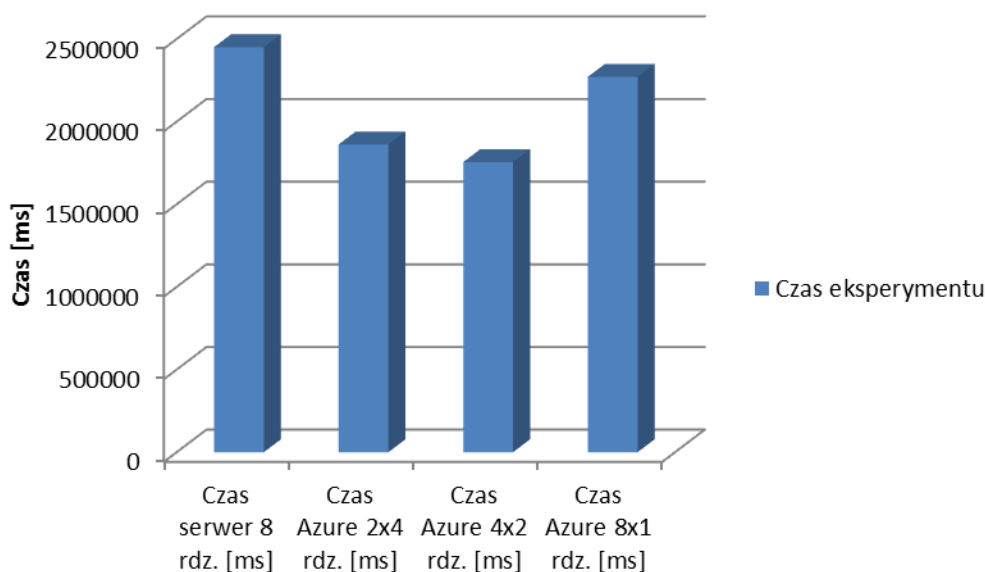
Rys. 5. Suma czasów realizacji wszystkich zadań dla poszczególnych platform
 Fig. 5. Total time of all executed tasks for each platform

Na rys. 5 oraz rys. 6 przedstawiono całkowite czasy realizacji testów dla poszczególnych platform, które są przedstawione w dolnej części tabeli 5. Pierwsze z wyliczonych wartości reprezentują sumaryczne czasy wykorzystane przez rdzenie/serwery na rozwiązanie wszyst-

kich zadań (w tabeli sumy te opisano jako „Suma czasów działania rdzeni procesora”). Sumaryczny czas przeznaczony na wykonywanie testu okazał się najmniejszy w przypadku lokalnego serwera fizycznego, a w środowisku chmury wzrastał on wraz ze wzrostem liczby serwerów tworzących klaster.

Na rys. 6 przedstawiono bezwzględny czas realizacji wszystkich zadań eksperymentu. W tym przypadku najszybszą konfiguracją okazał się Windows Azure z 4 serwerami klasy Medium dysponującymi 2 rdzeniami procesora. Pomimo uzyskania dopiero 3. wyniku pod względem długości sumarycznego czasu zużytego przez procesory na wykonanie wszystkich zadań, bezwzględny czas realizacji eksperymentu był najkrótszy. Uzyskany rezultat jest wynikiem specyfiki zestawu przetwarzanych w teście zadań, która okazała się najbardziej korzystna dla tej właśnie konfiguracji, pomimo że występowanie 2 ostatnich, czasochłonnych zadań faworyzowało konfigurację zawierającą 2 serwery klasy *Large* dysponujące 4 rdzeniami procesora (w czasie realizacji tych zadań 2 serwery z 4 klasy *Medium* i 6 serwerów z 8 klasy *Small* oczekiwało beczynnie na zakończenie wszystkich obliczeń).

Czas eksperymentu



Rys. 6. Bezwzględny czas realizacji wszystkich zadań testu

Fig. 6. The absolute duration of all the test tasks

Z przeprowadzonych badań można wysnuć jeszcze jeden praktyczny wniosek, mówiący, że w przypadku zadań o podobnej czasochłonności, które mogą być rozwiązywane równoległe na kilku serwerach, warto wykorzystać konfigurację o jak największej liczbie serwerów. Zrównoleglenie obliczeń na poziomie serwerów wykonujących niezależne zadania (których wykonanie nie jest synchronizowane) daje lepsze rezultaty niż zrównoleglenie na poziomie wątków pojedynczego serwera, których działanie musi być synchronizowane.

6.3. Badanie czasu realizacji żądania przez serwis przy zmiennej liczbie użytkowników

Ta grupa testów objęła badanie sposobu zachowania się serwisu zlokalizowanego w chmurze obliczeniowej przy zmieniającej się liczbie użytkowników chcących skorzystać z usługi. W przypadku tego testu usługa nie rozwiązywała problemu komiwojażera, tylko przez ustalony czas (200 ms) symulowała działanie, po którym kończyła pracę.

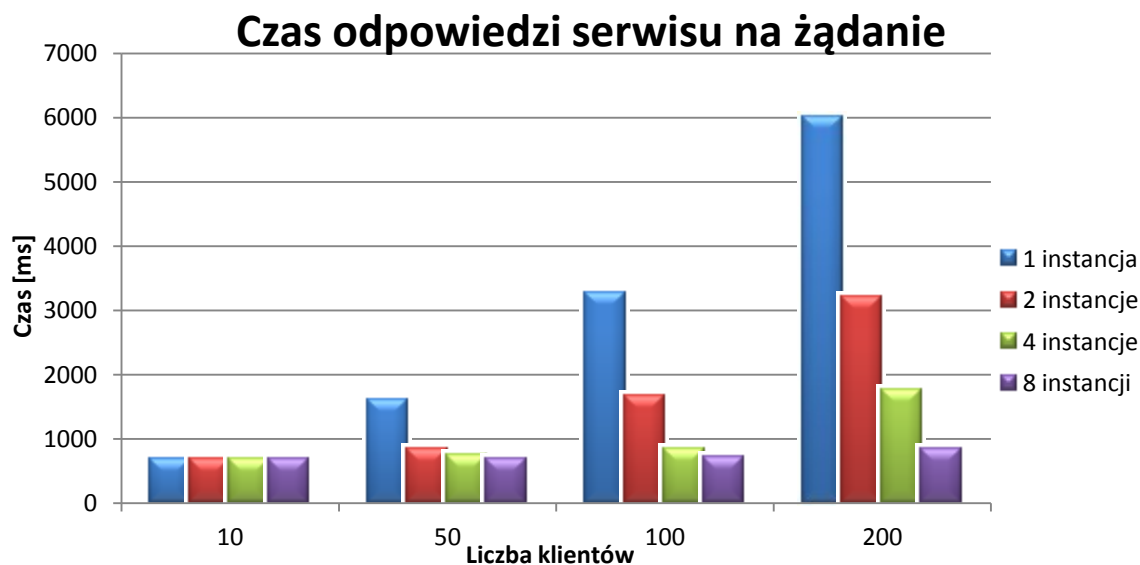
Uzyskane wyniki dla różnej liczby (1, 2, 4, 8) serwerów jednordzeniowych (serwery klasy *Small*) zlokalizowanych w chmurze Windows Azure przedstawiono w tabeli 6 oraz na rys. 7. Średni czas realizacji żądania oznacza czas, który upłynął od chwili wysłania żądania do serwisu do chwili otrzymania odpowiedzi.

Analiza uzyskanych wyników wskazuje, że przy wzroście liczby użytkowników chcących skorzystać z usługi wzrasta czas obsługi żądań (są one kolejgowane). Stąd też w przypadku gdy przewiduje się dużą liczbę klientów korzystających z serwisu zlokalizowanego w chmurze, korzystne jest zastosowanie konfiguracji obejmującej większą liczbę serwerów. Przeprowadzone testy wskazują również na poprawne zachowanie się modułu Windows Azure odpowiedzialnego za równoważenie obciążenia poszczególnych serwerów klastra oraz wskazują na dobrą skalowalność systemu przy wzroście liczby serwerów.

Tabela 6

Średnie czasy realizacji żądań przez serwis zlokalizowany w chmurze w zależności od liczby użytkowników i liczby serwerów wirtualnych

Liczba użytkowników	Liczba serwerów 1-rdzeniowych	Średni czas realizacji żądania [ms]
10	1	688
50	1	1 617
100	1	3 266
200	1	6 019
10	2	682
50	2	833
100	2	1 653
200	2	3 196
10	4	682
50	4	727
100	4	848
200	4	1 754
10	8	675
50	8	686
100	8	700
200	8	837



Rys. 7. Czas realizacji żądania przez serwis zlokalizowany w chmurze przy zmiennej liczbie użytkowników

Fig. 7. Cloud service request execution time by a variable number of users

7. Zakończenie

W ramach przeprowadzonych prac zbadano różne aspekty wykorzystania technologii chmury obliczeniowej:

- porównanie wydajności realizacji czasochłonnych algorytmów w stosunku do fizycznego serwera lokalnego,
- zbadanie sposobu zachowania się chmury obliczeniowej w przypadku algorytmów wykorzystujących różne liczby wątków,
- zbadanie sposobu zachowania się chmury obliczeniowej zależnie od liczby użytkowników, którzy chcą skorzystać z oferowanych usług.

W przypadku realizacji pojedynczych zadań serwer fizyczny zawsze wykazywał się większą wydajnością niż rozwiązania korzystające z Windows Azure. Zastosowanie większej liczby serwerów zlokalizowanych w chmurze dawało z kolei lepsze wyniki w przypadku realizacji zestawów zadań. Rodzaj optymalnej – dla rozwiązania danego zestawu zadań – konfiguracji serwerów Windows Azure jest zależny od liczby i rozkładu czasochłonności wykonania zadań wchodzących w skład zestawu. Dla dużej liczby zadań o małej lub podobnej czasochłonności wykonania bardziej odpowiednie będzie zastosowanie większej liczby serwerów klasy niższej, a dla mniejszej liczby zadań o dużej lub zróżnicowanej czasochłonności wykonania bardziej odpowiednie będzie zastosowanie większej liczby serwerów klasy wyższej.

BIBLIOGRAFIA

1. McKendrick J.: Cloud Computing Market Hot, But How Hot? Estimates are All Over the Map. Forbes, 2012: <http://www.forbes.com/sites/joemckendrick/2012/02/13/cloud-computing-market-hot-but-how-hot-estimates-are-all-over-the-map/>, dostęp 20.01.2013.
2. Zając Ł.: Analiza wydajności przetwarzania danych w środowisku chmury obliczeniowej MS Windows Azure. Praca dyplomowa magisterska, Politechnika Śląska, Wydział Automatyki, Elektroniki I Informatyki, Gliwice 2012.
3. Ranadive A., Kesavan M., Gavrilovska A., Schwan K.. Performance implications of virtualizing multicore cluster machines. Proceedings of the 2nd Workshop on System-Level Virtualization For High Performance Computing (Glasgow, Scotland, 2008), HPCVirt '08. ACM, Nowy Jork 2008.
4. Palankar M., Iamnitchi A., Ripeanu M., Garfinkel S.: Amazon S3 for science grids: a viable solution? Proceedings of DADC'08 – International workshop on Data-aware distributed computing, ACM, Nowy Jork 2008, s. 55÷64.
5. Calder B. et al.: Windows Azure Storage: a highly available cloud storage service with strong consistency. Proceedings of SOSP'11 – The Twenty-Third ACM Symposium on Operating Systems Principles, ACM, Nowy Jork 2011, s. 143÷157.
6. Zach H., Jie L., Ming M., Ruiz-Alvarez A., Marty H.: Early observations on the performance of Windows Azure. Proceedings of HPDC'10 – the 19th ACM International Symposium on High Performance Distributed Computing, ACM, Nowy Jork 2010, s. 367÷376.
7. Tudoran R., Costan A., Antoniu G., Bougé L.: A performance evaluation of Azure and Nimbus clouds for scientific applications. Proceedings of CloudCP'12 – the 2nd International Workshop on Cloud Computing Platforms, No. 4, ACM, Nowy Jork 2012.
8. Witryna Introducing Windows Azure: <http://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/>, dostęp: 20.01.2013.
9. Szczegóły cennika usługi Windows Azure: <https://www.windowsazure.com/pl-pl/pricing/details/>, dostęp: 20.01.2013.
10. Lazarova M., Borovska P.: Comparison of Parallel Metaheuristics for Solving the TSP. Proceedings of CompSysTech'08. Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, No. 17, ACM, Nowy Jork 2008.
11. Witryna biblioteki TSPLib: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>, dostęp 20.01.2013.

Wpłynęło do Redakcji 16 stycznia 2013 r.

Abstract

The paper presents the results of performance tests for solving the traveling salesman problem with the use of a parallel version of a genetic algorithm on a local multi-core server, and in the Windows Azure cloud (Table 1). The research work examined various aspects of the use of the cloud computing technology:

- Performance of the execution of time-consuming algorithms compared to a local physical server (Tables 2 and 3, Fig. 1 and 2)
- Analysis of the behavior of cloud computing servers in the case of algorithms that use a various number of threads (Tables 4 and 5, Fig. 3-6)
- Examination of the behavior of cloud computing services, according to the number of users who want to use the offered services (Table 6, Fig. 7)

The analysis of the obtained results allows the selection of appropriate configuration of the servers in the cloud. In the case of execution of single tasks, the physical server always showed higher performance than Windows Azure configurations. The use of a larger number of servers in the cloud, in turn, gave better results in the case of the sets of tasks. The optimal – for solving a given set of tasks - Windows Azure server configuration depends on the number and execution times distribution of the tasks included in the set. For a large number of tasks with short or very similar execution times, it is more appropriate to use a larger number of lower-end servers, and for fewer tasks with high or varying execution times, it is more appropriate to use more high-end servers.

Adresy

Jacek FRĄCZEK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, jacek.fraczek@polsl.pl.

Łukasz ZAJĄC: LGBS Polska Sp. z o.o., ul. Toszecka 101, 44-100 Gliwice, Polska, l.zajac@lgbs.pl.