

TECHNOLOGIA CHMUROWA USŁUG TYPU PAAS W REALIZACJI PROJEKTÓW INFORMATYCZNYCH

GRZEGORZ SZYJEWSKI¹, PIOTR NIEMCEWICZ²

¹ Uniwersytet Szczeciński
Wydział Nauk Ekonomicznych i Zarządzania
e-mail: grzegorz.szyjewski@usz.edu.pl

² Asseco Data Systems SA
e-mail: piotr.niemcewicz@assecods.pl

SŁOWA KLUCZOWE

internet, przedsięwzięcia informatyczne, chmura, PaaS, systemy internetowe, usługi internetowe

STRESZCZENIE

W artykule przedstawiono możliwość wykorzystania nowoczesnych technologii chmurowych w projektach informatycznych, o nieznanym parametrze wymaganej wydajności produkcyjnej. Badania przeprowadzono na rzeczywistym przedsięwzięciu, realizowanym w maju 2016. Był nim pierwszy Ogólnopolski Test Informatyczny online. Projekt ten charakteryzował się dużą niepewnością, związaną z wymogami wydajnościowymi produktu końcowego. Przedstawione dane pokazują zalety oraz ogromne możliwości, jakie daje wykorzystanie usług typu PaaS w technologii chmurowej, przy realizacji projektów informatycznych.

Wprowadzenie

W każdym przedsięwzięciu niezwykle istotnym elementem jest przestrzeń, w jakiej jest ono realizowane. Ten element może istotnie wpływać na harmonogram prac i wykorzystanie zasobów. W praktyce ma to bezpośrednie przełożenie na najistotniejszy parametr, jakim jest koszt realizacji projektu. W przypadku przedsięwzięć np. budowlanych, teren, na którym będą prowadzone prace, może istotnie ułatwić lub utrudnić ich realizację. Jednak nie zawsze takie pro-

blemy kojarzone są z projektami informatycznymi, które realizowane są głównie przy biurkach. Niemniej jednak przestrzeń, w której prowadzone są prace, jest często bardzo istotna również w przypadku projektów informatycznych. Chodzi tutaj o środowisko (infrastrukturę), w którym jest tworzony, a następnie eksploatowany produkt przedsięwzięcia. Sytuacja staje się bardziej skomplikowana, jeżeli tworzony jest produkt na tyle specyficzny, że nie można oszacować jego końcowej wydajności produkcyjnej. Wskazanie odpowiednich parametrów środowiska, w którym będzie funkcjonowała aplikacja, staje się wówczas dużym problemem.

1. Chmurowe platformy typu PaaS

Jednym z pojęć informatycznych, które zadomowiło się w języku codziennym, jest chmura obliczeniowa, funkcjonująca równolegle i zamiennie w formie „cloud”. Chmurę obliczeniową zdefiniować można jako: model dostarczania usług informatycznych, w którym zasoby znajdują się w internecie, a dostęp do nich odbywa się za pomocą narzędzi i aplikacji typu web-based (Gopal, Kaushik, 2016). Dane i aplikacje znajdują się na serwerach dostawcy usługi, jednak mechanizmy przetwarzania w chmurze pozwalają na dostęp do nich jedynie przez czas dostępu urządzenia do sieci. Z punktu widzenia użytkownika, dane znajdują się w bliżej nieokreślonym miejscu, „w chmurze”.

Dokument „The NIST Definition of Cloud Computing”, przygotowany przez National Institute of Standards and Technology, definiuje chmurę PaaS jako model umożliwiający użytkownikowi platformy (konsumentowi) dostarczanie do infrastruktury chmury aplikacji własnych lub pozyskanych z zewnątrz albo od dostawcy chmury (Mell, Grance, 2011, s. 6). Użytkownicy nie zarządzają infrastrukturą chmury i nie kontrolują tej infrastruktury, składającej się z takich komponentów jak: sieć, serwery, systemy operacyjne, systemy składowania danych. W ich rękach pozostaje kontrola nad wdrażaniem aplikacji i ustawieniami konfiguracji aplikacji, w środowiskach hostowanych na platformie.

Jedną z komercyjnie działających platform typu PaaS jest technologia opracowana przez firmę Jelastic. Łączy ona w sobie cechy charakterystyczne dwóch modeli chmur obliczeniowych – elastyczność IaaS z łatwością użycia PaaS (unicloud.pl, 2014). W przedstawionym rozwiązaniu zasoby skalowane są elastycznie (skalowanie pionowe i poziome), co jest cechą unikalną dla tej technologii. Chmura automatycznie kontroluje i optymalizuje użycie zasobów. Raportowanie i rozliczanie odbywa się w sposób transparentny dla konsumenta i operatora platformy. Zasoby rozliczane są w cyklach godzinowych, w zależności od zużycia.



Rysunek 1. Panel konfiguracji środowiska na platformie Jelatic

Źródło: <https://app.unicloud.pl>.

Spośród ważniejszych funkcjonalności dostępnych dla użytkownika wymienić można: rezerwację zasobów przy pomocy suwaków, ustawienie minimalnych i maksymalnych parametrów, określanie warunków brzegowych dla uruchomienia skalowania oraz proste klonowanie środowisk. Wykorzystana w badaniach infrastruktura UniCloud to jedyna publiczna chmura Jelatic dostępna w Polsce. Polska wersja wyróżnia się spośród wszystkich hostowanych instancji wykorzystaniem jako nośnika danych jedynie dysków SSD.

2. Wydajność jako składowa użyteczności systemów

W nowoczesnych systemach informatycznych nacisk kładziony jest głównie na ich użyteczność produkcyjną, znaną również pod pojęciem jakości użytkowej. Ten element określa stopień dopasowania produktu do potrzeb użytkownika końcowego. SI tworzone są w celu realizacji określonych funkcji oraz dostarczania określonych możliwości użytkowych, które definiują jego funkcjonalność (Śmiałkowska, 2015, s. 99). System informatyczny nie będzie stanowił wartości produkcyjnej, jeżeli nie spełnia oczekiwań użytkownika końcowego. Powodem takiej sytuacji mogą być błędy popełnione na etapie wczesnej analizy potrzeb, nieudanych prób reengineeringu¹ obsługiwanych procesów lub nawet podczas późniejszego wytwarzania produktu.

Spełnienie oczekiwań związanych z użytecznością teoretycznie mogłoby oznaczać sukces projektu wdrożeniowego. Pozostaje jednak kwestia dostępności systemu, bez której nawet naj-

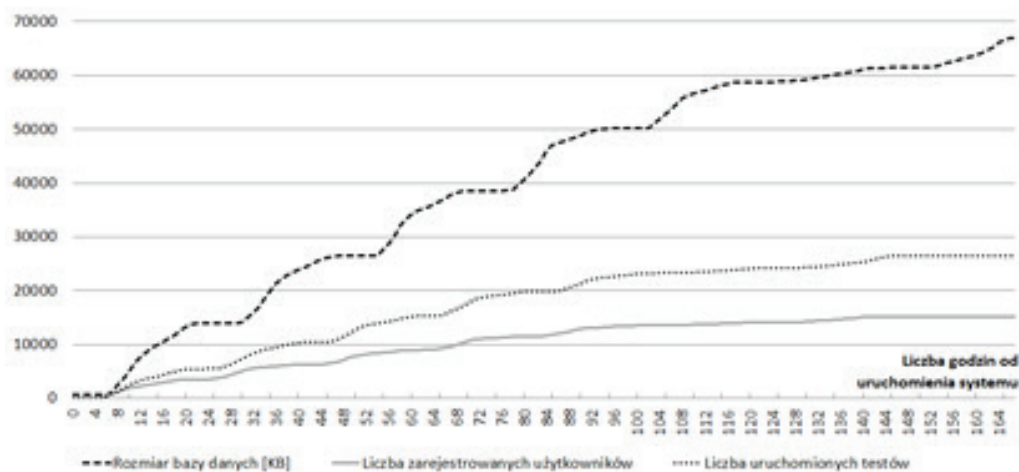
¹ Reengineering to działanie polegające na radykalnym przeprojektowaniu procesów biznesowych oraz powiązanych z nimi systemów, procedur czy nawet struktury organizacyjnej. Celem takiego działania jest optymalizacja toku pracy i produktywności organizacji (Manganelli, Klein, 1998).

lepsze dopasowanie produktu będzie bez znaczenia w końcowym bilansie, stanowiącym o powodzeniu projektu informatycznego. Na dostępność systemu wpływają dwa czynniki: organizacyjno-proceduralny oraz sprzętowy. Pierwszy element polega na kontroli dostępu do systemu, związanej z ochroną danych. W tym przypadku chodzi o tworzenie oraz realizację procedur weryfikacji i autoryzacji użytkowników, związanych z kontrolą przydzielonych im uprawnień (Szyjewski, 2014). Drugi element to dostępność fizyczna systemu, czyli zapewnienie odpowiedniej infrastruktury, na której system będzie mógł prawidłowo funkcjonować. W tym elemencie kluczowa staje się wydajność systemu. Zależy ona od stopnia skomplikowania realizowanych obliczeń oraz liczby użytkowników, którzy będą jednocześnie te obliczenia realizować. To przekłada się na ilość i jakość zasobów sprzętowych, które są niezbędne do prawidłowego funkcjonowania produktu. W praktyce oznacza to, że najlepiej przygotowany pod względem jakościowym system informatyczny będzie bezużyteczny w przypadku pojawienia się problemów z jego wydajnością.

Jednym z elementów procesu wytwarzania systemów informatycznych jest testowanie przygotowanego produktu. Z uwagi na fakt, iż znane są założenia związane z jego funkcjonalnością, wykonanie testów jakościowych można stosunkowo łatwo przeprowadzić w warunkach symulacji w środowisku testowym. Problematiczne natomiast staje się przeprowadzenie w takich warunkach testów wydajności systemu. Składa się on bowiem z kombinacji takich elementów jak przepustowość czy czas odpowiedzi, które z kolei ściśle zależą od liczby użytkowników końcowych oraz rozmiaru bazy danych (Zakrzewicz, Wojciechowski, 2002, s. 243–260). Części z tych elementów nie da się jednoznacznie oszacować na etapie analizy czy wytwarzania systemu. Chodzi tutaj głównie o liczbę użytkowników, która w niektórych typach przedsięwzięć staje się trudna do przewidzenia. Również wielkość bazy danych staje się niewiadomą, ponieważ zależy od popularności systemu, który jest wytwarzany w ramach projektu informatycznego. W rezultacie realizacja projektu polega na opracowaniu systemu, którego wydajność musi być proporcjonalna do obciążenia, będącego parametrem trudnym do oszacowania przed wdrożeniem rozwiązania.

Prawidłowym podejściem w takich przypadkach będzie optymalizacja kodu systemu, polegająca na minimalizacji wykorzystania zasobów sprzętowych podczas realizacji operacji obliczeniowych. Nie może to jednak przekładać się na spadek użyteczności funkcjonalnej, która jest parametrem stałym, zdefiniowanym w ramach zakresu prowadzonego projektu. Problemem pozostaje jednak dobór sprzętowego środowiska produkcyjnego, w którym funkcjonować ma produkt. Przykładem przedsięwzięcia, w którym trudna do oszacowania była wymagana końcowa wydajność, jest pierwszy Ogólnopolski Test Informatyczny (OTI), który odbył się w maju 2016 roku. Organizatorem było Polskie Towarzystwo Informatyczne, które w ramach obchodów 35-lecia swojego istnienia postanowiło udostępnić platformę w trybie online, pozwalającą na sprawdzenie kompetencji gracza z zakresu wiedzy i umiejętności informatycznych. W ramach OTI każda chętna osoba mogła wykonać test, wybierając jeden z trzech poziomów trudności: podstawowy, rozszerzony lub zaawansowany.

Projekt polegający na przygotowaniu systemu, umożliwiającego realizację tego wydarzenia, cechował się brakiem możliwości oszacowania wymaganej końcowej wydajności systemu. Wynikało to z niewiadomej, jaką była liczba użytkowników, którzy ostatecznie wezmą udział w wydarzeniu. Dodatkowym elementem wpływającym na prawidłowe funkcjonowanie systemu był czas. Łatwy do przewidzenia był fakt, iż wraz z upływem czasu funkcjonowania systemu liczba zrealizowanych testów będzie rosła. To przełoży się na rozmiar bazy, której wydajność będzie spadać wraz z przyrostem kolejnych danych. Wszystkie te elementy powodowały dużą niepewność w projekcie, wynikającą nie tylko z braku możliwości przewidzenia produkcyjnego obciążenia systemu, ale również dynamiki trendu związanej ze wzrostem zapotrzebowania na zasoby sprzętowe.



Rysunek 2. Rozmiar bazy danych oraz liczba użytkowników w kolejnych godzinach dostępności systemu OTI
Źródło: opracowanie własne.

Na rysunku 2 przedstawiony został wykres, ilustrujący przyrost: rozmiaru bazy danych systemu, liczby użytkowników oraz liczby przeprowadzonych testów w trakcie trwania wydarzenia. Wykres został przygotowany na podstawie rzeczywistych danych, które były dostępne już po zakończeniu OTI. Pierwszym parametrem trudnym do oszacowania była liczba osób, które zdecydują się wziąć udział w wydarzeniu. Kolejnym, jaka będzie średnia liczba testów uruchomionych przez pojedynczą osobę. Każdy z zarejestrowanych mógł bowiem uruchomić jeden, dwa lub trzy testy, co w praktyce oznacza podwójne lub potrójne zwiększenie obciążenia systemu informatycznego. Opisane dwa parametry wpływają następnie na rozmiar bazy danych, która była kluczowym elementem związanym z wydajnością systemu. Im większy rozmiar danych, tym dłuższy czas zapytań (mniejsza wydajność bazy) oraz większe obciążenie zasobów sprzętowych (Niemann, Korfiatis, Zicari, Göbel, 2013). Znając przybliżoną liczbę uczestników

oraz średnią liczbę wykonanych testów, można by wykonać obliczenia w celu oszacowania rozmiaru bazy w kolejnych dniach. Niestety, brak tych informacji powoduje zarówno dużą niepewność w projekcie, jak i brak możliwości przewidzenia, jakie zasoby sprzętowe będą potrzebne do utrzymania dobrej wydajności systemu w całym okresie trwania wydarzenia.

3. Wykorzystanie technologii Jelastic w przedsięwzięciach informatycznych

Cykl życia każdego systemu informatycznego można podzielić na kolejno następujące po sobie etapy. W zależności od podejścia podział cyklu może być bardziej lub mniej szczegółowy. Bardzo ogólny schemat przedstawia ten proces jako cztery kolejne fazy: planowanie, analiza, wytwarzanie oraz implementacja i eksploatacja (Valacich, George, Hoffer, 2015, s. 20). Cykl życia systemu należy bowiem rozumieć jako jego ewolucję w czasie. Zaczynając od koncepcji, aż do wycofania z użycia (Swacha, 2015, s. 29), czyli momentu, w którym ponownie rozpoczyna się faza planowania związana z rozwojem lub całkowite zamknięcie systemu. W projektach informatycznych zakresem staje się produkt, który musi być zrealizowany (wytworzony i wdrożony) przy zachowaniu założonego budżetu oraz w określonym czasie (Szyjewski, 2001). Te trzy parametry powinny być w miarę dokładnie znane po zakończeniu fazy analizy. Wówczas doprecyzowany jest już końcowy kształt produktu (zakres), dzięki czemu można ostatecznie oszacować czas realizacji oraz koszt jego wytworzenia.

W przypadku produktów stanowiących systemy informatyczne, których działanie opiera się na technologiach serwerowych, istotnym elementem jest określenie wymogów sprzętowych. Jest to niezbędne do zachowania wyznaczonej wydajności systemu, w trakcie fazy eksploatacji. Czynność ta staje się jednak problematyczna w przypadku systemów o nieznanym obciążeniu produkcyjnym, takich jak opisany w poprzedniej części artykułu (OTI). W trakcie wytwarzania systemu informatycznego, gdzie najistotniejszym elementem staje się jego wydajność w kontekście obciążenia przez użytkowników, ważne jest końcowe środowisko produkcyjne. Optymalizacja aplikacji na poziomie kodu źródłowego jest czynnością oczywistą, jednak bardzo wiele elementów zależy również od konfiguracji składowych środowiska, w którym funkcjonuje aplikacja. Wytwarzanie produktu w środowisku deweloperskim, następnie przenoszenie do produkcyjnego, które znajduje się np. na innym serwerze, zwiększa czasochłonność oraz ryzyko w projekcie.

Technologia Jelastic pozwala uniknąć opisanych problemów, ponieważ jest platformą elastyczną, którą bez problemu można dostosować do potrzeb realizacji projektu informatycznego. Najważniejszym elementem staje się kontrola kosztów. Topologia środowiska posiada jasno ustalone zależności pomiędzy wykorzystywanymi elementami a obciążeniem konta klienta. Zarządzanie środowiskiem jest możliwe w czasie rzeczywistym, co oznacza, że działa ono (i generuje koszty) tylko w takiej skali, w jakiej jest wykorzystywane. Drugim ważnym czynnikiem jest możliwość pracy w środowisku deweloperskim, które następnie może stać się produkcyjnym, a wszystko za sprawą narzędzi umożliwiających klonowanie oraz skalowanie zasobów. Środowisko deweloperskie może być bowiem utrzymywane przy minimalnej niezbędnej rezerwacji

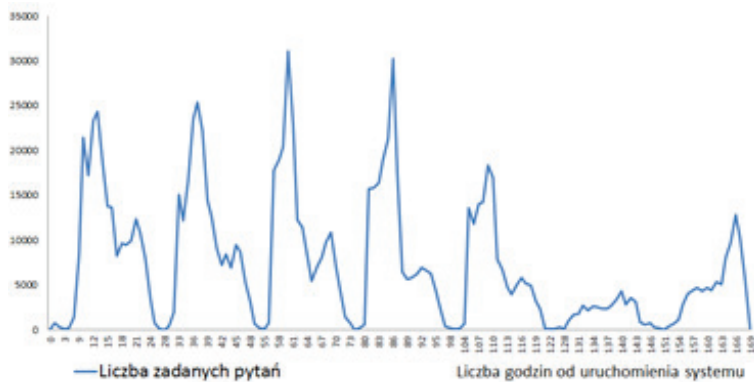
zasobów, nie generując kosztów produkcyjnych w takcie fazy wytwarzania. Okresowo może być jednak przeskalowane do parametrów produkcyjnych, np. w celu realizacji testów wydajności lub innych prac wymagających użycia serwera o parametrach produkcyjnych. Co więcej, środowisko może zostać całkowicie zatrzymane na czas, w którym w projekcie nie są realizowane prace wymagające użycia serwera. Taka operacja oznacza z kolei całkowite odejście od kosztów utrzymania infrastruktury w czasie, w którym nie jest ona wykorzystywana.

Użycie technologii Elastic znakomicie rozwiązuje również opisany wcześniej problem nieznanego obciążenia produkcyjnego aplikacji. Przejście z fazy wytwarzania do fazy implementacji i późniejszej eksploatacji systemu informatycznego wymaga określenia parametrów środowiska produkcyjnego. Wykorzystanie technologii Elastic sprawia, że środowisko deweloperskie może stać się produkcyjnym poprzez użycie opcji klonowania. Zasoby środowiska mogą być szybko dostosowywane do aktualnych wymagań, zarówno w sposób ręczny, jak i automatyczny, a wszystkie zmiany rozliczane są w interwale czasowym jednej godziny. Dzięki temu wydajność aplikacji jest na bieżąco dostosowywana do wymagań użytkowników końcowych, a ponoszone przez właściciela produktu koszty są adekwatne do wykorzystanych zasobów. Nie występują również problemy związane z relokacją systemu, ponieważ aplikacja funkcjonuje cały czas w ramach tego samego środowiska.

4. Technologia Elastic w praktyce

Opisany system do obsługi przedsięwzięcia o nazwie Ogólnopolski Test Informatyczny jest przykładem aplikacji, w której najważniejszym elementem była wydajność. Wynika to z faktu, iż faza eksploatacji systemu była bardzo krótka i wynosiła jedynie 168 godzin (7 dni). Dodatkowo był to system przeznaczony dla użytkownika typu masowego, w którym ostateczna liczba korzystających była nieznana. Realizacja pojedynczej procedury przeprowadzenia testu była stosunkowo krótka (ok. 10 minut), jednak bardzo dynamiczna, ponieważ o końcowym wyniku decydował czas, w jakim udzielone były odpowiedzi. Nawet minimalne opóźnienia w funkcjonowaniu systemu, wynikające z niewystarczającej wydajności produktu, stanowiłyby o porażce przedsięwzięcia. Użytkownik podczas realizacji testu miał zmagać się z przygotowanymi przez organizatorów pytaniami, a nie z niewydolnością systemu informatycznego.

Możliwymi rozwiązaniami były środowiska na maszynach fizycznych lub wirtualnych, o architekturze spełniającej przeszacowane wymogi zapobiegające powstawaniu wąskich gardeł. Ze względu na krótki czas trwania testów nie było możliwości dostrojenia klasycznych rozwiązań do optymalnego zapotrzebowania. Rozkład testów w czasie poznany został dopiero po zakończeniu przedsięwzięcia. Doskonałą alternatywą dla maszyny o przeszacowanych parametrach okazało się wykorzystanie innowacyjnych rozwiązań, dopasowujących wydajność środowiska do wymaganego zapotrzebowania.



Rysunek 3. Obciążenie systemu zapytaniami w kolejnych godzinach

Źródło: opracowanie własne.

Na rysunku 3 przedstawiony został wykres obciążenia systemu, wynikającego z aktywności i liczby użytkowników. Wykres przedstawia liczbę pytań zadawanych przez system w ramach uruchomionych testów. Ten element bardzo dobrze odzwierciedla rzeczywiste obciążenie aplikacji, biorąc pod uwagę fakt, iż nie występuje ustalona zależność pomiędzy pojedynczą grą a wykorzystaniem systemu. W ramach jednego testu mogło być wylosowanych, wyświetlonych i sprawdzonych od kilku do kilkudziesięciu pytań, w zależności od tego, jak szybko udzielane były odpowiedzi. Na wykresie bardzo dobrze widać również okresowość obciążenia. Wzrastało ono w trakcie dnia oraz zmniejszało się w czasie godzin nocnych. Dodatkowo wyraźny spadek zaobserwować można w trakcie dwóch ostatnich dni trwania wydarzenia, które przypadły na weekend.



Rysunek 4. Obciążenie zasobów przez system

Źródło: opracowanie własne z wykorzystaniem statystyk Jelastic.

Na rysunku 4 przedstawione zostało obciążenie zasobów, generowane przez działanie aplikacji. Z wykresów godzinowego obciążenia sieci wynika, że rozkład obciążenia pokrywa się z rytmem pracy aplikacji – zanika w godzinach nocnych oraz wzrasta w trakcie dnia. Z uwagi na przyjętą architekturę systemu obciążenie podstawowych zasobów platformy (CPU, RAM) było znikome, niemal niezauważalne. W związku z tym dane te zostały celowo pominięte na wykresach. Większość obciążenia skupiła się na dyskach (zużycie IOPS). Można wnioskować, że zastosowanie dysków SSD spowodowało przejście obsługi większości zapytań, dzięki czemu zarówno procesor, jak i pamięć nie brały większego udziału w procesie obsługi testów wykonywanych przez użytkowników.

W pierwszym dniu funkcjonowania, środowisko używało w sumie 83 MHz CPU i 1,77 GB RAM, w tym baza MySQL zagospodarowała 24 MHz CPU i 1,6 GB RAM. W ciągu dwóch pierwszych dni parametry ustabilizowały się na poziomie: CPU około 55–80 MHz oraz RAM 1,89 GB. Dobór optymalnych ustawień środowisk, przy zastosowaniu rozwiązań klasycznych, byłby zadaniem bardzo obciążającym zespół projektowy. Przed implementacją należałoby wówczas przeprowadzić testy obciążeniowe (beta), natomiast zastosowanie platformy chmurowej oraz dynamicznie dopasowywanie parametrów „w locie” pozwoliło na pominięcie tych czasochłonnych działań. Ciekawym zjawiskiem, jakie można zaobserwować na przedstawionych wykresach, jest również brak znaczącej zmiany w użyciu dysków (IOPS) w ostatnich dniach funkcjonowania systemu. Pomimo znaczącego spadku wykorzystania aplikacji przez użytkowników w dwóch ostatnich dniach użycie tego elementu nie zmieniło się w porównaniu do dni poprzednich. Najprawdopodobniej wynika to z faktu, iż w czasie funkcjonowania systemu przyrastała ilość zapisanych danych. W ostatnich dniach była tak duża, że zbilansowała mniejsze wykorzystanie systemu przez użytkowników.

Podsumowanie

Realizacja projektów informatycznych niesie ze sobą cały szereg zagrożeń. W celu wyeliminowania tych, które mogą wystąpić po uruchomieniu systemu, przeprowadzane są testy wytworzonego narzędzia. Kontrola powinna dotyczyć dwóch aspektów: jakościowego oraz ilościowego. Ten pierwszy ma na celu sprawdzenie, czy wszystkie przyjęte założenia zostały zrealizowane oraz czy narzędzia funkcjonują prawidłowo. Drugi powinien skontrolować poprawność działania systemu w trakcie korzystania z niego przez określoną liczbę użytkowników. Jest to szczególnie trudne do przeprowadzenia w przypadku systemów dedykowanych dla użytkowników masowych. Wynika to z faktu, iż trudno jest przewidzieć ich liczbę oraz czas, w którym będą jednocześnie korzystać z aplikacji. Dodatkowo przeprowadzenie takich testów w warunkach symulowanych nie zawsze odzwierciedla rzeczywiste warunki produkcyjne, w jakich będzie funkcjonować aplikacja. Wszystkie te elementy składają się na brak możliwości trafnego oszacowania parametrów zasobów, jakie niezbędne będą do prawidłowego działania przygotowywanego produktu.

Opisane w publikacji przedsięwzięcie było przykładem projektu charakteryzującego się takimi właśnie cechami: dużą niepewnością co do ostatecznej liczby użytkowników oraz warunkiem wydajności systemu jako priorytetem związanym z jego użytecznością. Jako rozwiązanie tych problemów wskazana została technologia chmurowa typu PaaS w postaci aplikacji UniCloud/Jelastic. Realizacja przedsięwzięcia ujawniła nieznaną wcześniej liczbę użytkowników oraz potrzebne parametry zasobów. Potwierdziła również przyjęte założenia, iż wykorzystanie technologii chmurowych jest rozwiązaniem dla opisanych wcześniej problemów. W trakcie stosunkowo krótkiej fazy eksploatacji systemu zasoby dynamicznie dopasowywały się do obciążenia aplikacji poprzez skalowanie odpowiednich parametrów środowiska produkcyjnego. Przygotowany produkt działał płynnie, a koszty związane z obsługą zasobów były adekwatne do ich wykorzystania przez aplikację. Na podstawie przeprowadzonych badań można stwierdzić, iż opisane rozwiązanie jest dobrym podejściem przy realizacji projektów informatycznych, wykazujących niepewność w zakresie produkcyjnego obciążenia tworzonego produktu.

Literatura

- Gopal, D.G., Kaushik, S. (2016). Emerging Technologies and Applications for Cloud-Based Gaming: Review on Cloud Gaming Architectures. W: P.V. Krishna, *Emerging Technologies and Applications for Cloud-Based Gaming*. VIT University.
- Manganelli, R.L., Klein, M.M. (1998). *Reengineering*. Warszawa: Polskie Wydawnictwo Ekonomiczne.
- Mell, P., Grance, T. (2011). *The NIST Definition of Cloud Computing*, Special Publication 800–145. Gaithersburg: National Institute of Standards and Technology.
- Niemann, R., Korfiatis, N., Zicari, R., Göbel, R. (2013). *Does query performance optimization lead to energy efficiency? A comparative analysis of energy efficiency of database operations under different workload scenarios*, arXiv preprint arXiv:1303.4869.
- Niemcewicz, P. (2014). *UniCloud, czyli PaaS + IaaS*. Pobrane z: <https://pomoc.unicloud.pl/unicloud/podstawy/czym-jest-paas/> (17.12.2014).
- Swacha, J. (2015). Metodyka kształtowania ryzyka w cyklu rozwojowym systemu informatycznego. W: P. Kosiuczenko, M. Śmiałek, J. Swacha, *Od procesów do oprogramowania: badania i praktyka*. Warszawa: Polskie Towarzystwo Informatyczne.
- Szyjewski, G. (2014). *Organizacja uwierzytelniania internetowych transakcji biznesowych* (praca doktorska). Szczecin: Uniwersytet Szczeciński.
- Szyjewski, Z. (2001). *Zarządzanie projektami informatycznymi. Metodyka tworzenia systemów informatycznych*. Agencja Wydawnicza Placet.
- Śmiałkowska, B. (2015). Metoda oceny użyteczności i funkcjonalności hurtowni danych. W: P. Kosiuczenko, M. Śmiałek, J. Swacha, *Od procesów do oprogramowania: badania i praktyka*. Warszawa: Polskie Towarzystwo Informatyczne.
- Valacich, J.S., George, J.F., Hoffer, J.A. (2015). *Essentials of systems analysis and design*. Pearson Education.
- Zakrzewicz, M., Wojciechowski, M. (2002). *TPC Benchmarking*, materiały konf. PLOUG 2002. Zakopane.

PAAS SERVICES IN CLOUD TECHNOLOGY, FOR IT PROJECT EXECUTION

KEYWORDS | Internet, IT project, cloud, PaaS, internet systems, internet services

ABSTRACT | The following article presents the possible use of cloud technologies within IT projects, where the level of expected efficiency cannot be established prior. The survey has been conducted on an actual project, the first on-line Polish IT Test, realized in May 2016. The project had very high degree of uncertainty, regarding the end-product efficiency requirements. Presented data show advantages and great possibilities of using PaaS services in cloud technology, for IT projects execution.

