Prof. dr hab. inż. Jacek Kitowski
Katedra Informatyki
Wydział Informatyki, Elektroniki i Telekomunikacji
Akademia Górniczo-Hutnicza
Al. Mickiewicza 30, 30-059 Kraków
kito@agh.edu.pl

Kraków, 14.9.2019

Recenzja rozprawy doktorskiej

Pani mgr inż. Joanny Kosińskiej
pt. „Autonomic management of cloud-native applications”

Promotor: Prof. dr hab. inż. Krzysztof Zieliński

Niniejsza recenzja przygotowana została na zlecenie Prodziekana Wydziału Informatyki, Elektroniki i Telekomunikacji Akademii Górniczo-Hutniczej w Krakowie, Pana Dra hab. inż. Krzysztofa Winczę (pismo IEiT.510-2/10/608/2019 z dnia 12.07.2019), działającego na podstawie uchwały Rady Wydziału z dnia 11.07.2019.

I. Ocena wyboru tematu i tez rozprawy

Kilkanaście ostatnich lat przyniosło znaczące uzależnienie praktycznie każdego aspektu życia od dostępności i efektywności systemów informatycznych. Jednym z najbardziej współcześnie interesujących kierunków badawczych są badania w zakresie systemów równoległych i rozproszonych, wykorzystywanych zarówno w obszarze badań naukowych, jak i szeroko rozumianej współpracy i wspólnych przedsięwzięć, także w obszarze aktywności gospodarczej. Bardzo silnie zaznacza się wpływ systemów informatycznych na podniesienie jakości życia codziennego. Niestety ilość oraz stopień różnorodności zasobów dostępnych w sieci, jak również strategii według których działają poszczególni członkowie współpracujących organizacji, powoduje coraz większe problemy z płynnością współpracy oraz dostępem do informacji i zasobów odpowiadających konkretnemu zadaniu. Wzrasta poziom komplikacji systemów, a wraz z nim trudność wykorzystania dostępnych zasobów. Próbuje się temu zaradzić ukrywając przed typowym użytkownikiem złożoność systemów informatycznych wprowadzając szereg rozwiązań typu chmurowego (obliczenia w chmurze – ‘cloud computing’), różnorodność w organizacji zasobów chmurowych (np. ‘edge’ czy ‘fog computing’), lecz każde rozwiązanie tego typu wiąże się ze wzrostem liczby i heterogeniczności zasobów wykorzystywanych w zróżnicowanych celach, w sposób ciągły przez wielu użytkowników jednocześnie. Stawia to przed systemami wysokie wymagania dotyczące niezawodności, dostępności i serwisowalności, które to elementy – przy uwzględnieniu skali – wymagają coraz częściej cech autonomiczności.

W nakreślonym powyżej zakresie problemowym korzystnie lokuje się recenzowana rozprawa Pani mgr inż. Joanny Kosińskiej. Jest ona poświęcona (ogólnie) zagadnieniom autonomicznego zarządzania obliczeniami w środowiskach chmurowych z wykorzystaniem informacji o stanie środowiska i aplikacji oraz reguł realizujących wybraną politykę zarządzania. W pracy rozwinięto i przedyskutowano projekt autonomicznego zarządzania aplikacjami chmurowymi wraz z realizacją w środowisku kontenerowym.

Zatem, kierunek badań, wybór problematyki rozprawy, tezy i celów pracy zaproponowany przez p. mgr inż. Joannę Kosińską oceniam zdecydowanie pozytywnie.

II. Ocena zawartości rozprawy

Przedstawiona do recenzji rozprawa w języku angielskim liczy 147 stron tekstu i składa się z 7 rozdziałów, spisu rysunków i tabel, skrótów używanych w tekście oraz 93 pozycji bibliograficznych uszeregowanych w kolejności cytowania. Do pracy załączono listingi konfiguracji i podstawowych komponentów systemu. W spisie treści występuje pomyłka w numeracji stron (bibliografia rozpoczyna się na stronie 141, a nie – jak wskazano w spisie – od strony 134), a także nieliczne niedoskonałości bibliograficzne, np. w [85]. Całość tekstu zredagowana jest starannie z wielką dbałością o przejrzystość zapisu i szatę graficzną.

W Rozdziale 1 uzasadniono znaczenie obliczeń z wykorzystaniem środowisk chmurowych, przedstawiono cechy natywnej aplikacji chmurowej oraz wskazano na potrzebę autonomiczności w zarządzaniu. Przedstawiono tezę i zakres rozprawy wraz z jej strukturą. Jako wprowadzenie do rozważań szczegółowych podsumowano podstawowe osiągnięcia pracy. Zwrócono uwagę na duże znaczenie wpływu stanu komponentów środowiska (tj. kontekstu) na autonomiczność jego zarządzania. Rozdział 1 stanowi dobre postawienie problemu, będąc wstępem do zagadnień rozwijanych w rozprawie.

W Rozdziale 2 podsumowano istniejący stan wiedzy w zakresie środowisk chmurowych i elementów zarządzania zasobami. W pierwszej części rozdziału wskazano na cechy natywnych aplikacji chmurowych i technologii kontenerowych, w kolejnej – na jeden ze sposobów reprezentacji wiedzy używanej w zarządzaniu (reprezentacja regułowa) wraz z krótkim przeglądem poszczególnych technologii zarządzania natywnym środowiskiem chmurowym. Rozdział kończą rozważania dotyczące autonomiczności w paradygmacie chmurowym.

Rozdziały kolejne są w pełni autorskie. Rozdział 3 stanowi podstawowy rozdział teoretyczny rozprawy. Zawarto w nim wymagania stawiane przed systemem zarządzania natywnymi aplikacjami chmurowymi i na tej podstawie przedstawiono model środowiska wykonawczego. W kolejnych częściach rozdziału pokazano warstwowy model zarządzania wraz z akcjami przypisanymi do każdej z warstw w zależności od przyjętego zestawu polityk i kontraktu SLA. Dokonano wyboru metryk ujmując dyskusję w formalizm stosu obserwowalności ze zwróceniem uwagi na konieczność uwzględnienia kontekstu, czyli stanu innych komponentów/kontenerów. Ostatnia część rozdziału dotyczy konceptualizacji proponowanych elementów pracy na podstawie znanego modelu self-CHOP autorstwa IBM, co – przy założeniu środowiska kontenerowego – doprowadza do wprowadzenia autonomiczności zwanej w pracy ‘cloud-native MRE-K loop’.

Rozważania teoretyczne doprowadziły – w kolejnym Rozdziale 4 – do projektu środowiska autonomicznego zarządzania dla natywnych systemów chmurowych, o nazwie AMoCNA. Stanowi ono podstawowe osiągnięcie technologiczne rozprawy. Wprowadzono pojęcie aktorów systemu (klientów i dostawców usług) oraz szczegółowo opracowano i przejrzyście przedstawiono graficznie współpracę między elementami poszczególnych warstw. Przedyskutowano dwie strategie rozmieszczenia kontenerów przez orkiestrator – z wykorzystaniem pojedynczego klastra i instalacji wieloklastrowej. Ciekawym pomysłem jest zwielokrotnienie poszczególnych warstw wraz z ich elementami, co może pozytywnie wpływać na efektywność pracy i/lub stanowić zasoby zastępcze w sytuacjach awaryjnych. Rozdział kończy przedstawienie w formie listingów algorytmów inicjacji systemu i kontenera.

Rozdział 5 jest poświęcony w całości zagadnieniom implementacji opracowanych elementów środowiska. Dotyczy wyboru technologii (środowiska chmurowego OpenStack oraz konteneryzacji z wykorzystaniem Kubernetesa), jak również odwzorowania elementów środowiska AMoCNA) na dostępny stos technologiczny. Opisano także wybraną natywną aplikację testową (udostępnioną jako 'open source' w ramach licencji Apache License 2.0). Wybór technologii Kubernetesa skutkował wykorzystaniem innych dostępnych narzędzi z tego obszaru, w tym kontenera z systemem wnioskującym w oparciu o zdefiniowane reguły zarządzania (Drools Rule Engine), kontenera zarządzającego (o nazwie Business Central, wykorzystującego Drools Workbench) oraz kontenera z systemem monitorującym (o nazwie Prometheus). Przedstawiono szczegółowo poszczególne elementy (w tym podstawową klasę 'Metric' i deklaratywne określenie polityk zarządzania) oraz wykonane kroki prowadzące do wykorzystania środowiska Kubernetes dla potrzeb implementacji opracowanej architekctury AMoCNA. Szczególną wagę przywiązano do mikroserwisów autonomicznych oraz mikroserwisu polityk zarządzania. Jak pisze Autorka, w implementacji wykorzystano w możliwie najwyższym stopniu gotowe moduły i rozwiązania w celu szybkiej implementacji i obniżenia poziomu błędów.

W Rozdziale 6 przedstawiono wykonane eksperymenty weryfikujące założone własności systemu. We wstępie opisano środowisko testowe oraz uzyskane operacyjne cechy aplikacji natywnej wykonywanej w środowisku o symulowanym obciążeniu wraz z założeniami SLA. W kolejnej części potwierdzono cechy opracowanego środowiska poprzez wykonane badania, takie jak zbadanie efektu dostrojenia środowiska chmurowego poprzez wertykalne i horyzontalne skalowanie środowiska oraz weryfikację testowej autonomiczności zarządzania będącej wynikiem wykorzystania regułowej bazy wiedzy. Ważnym elementem ewaluacji była ocena narzutu wnoszonego przez opracowany system AMoCNA w odniesieniu do CPU i wykorzystania pamięci – w pierwszym przypadku stwierdzono niewielkie dodatkowe obciążenie, a drugim – wskazano na znaczące zapotrzebowanie na ten zasób. We wnioskach z badań wskazano na ograniczenia systemu AMoCNA.

Rozdział 7 stanowi podsumowanie pracy wraz propozycją dalszych badań.

Na podstawie wykonanych eksperymentów można stwierdzić, że postawiona teza została udowodniona.

Zatem recenzowana rozprawa p. mgr inż. Joanny Kosińskiej stanowi duży wkład do rozwoju autonomicznego zarządzania natywnymi aplikacjami chmurowymi. Jest dobrze ulokowana w obszarze intensywnych prac prowadzonych w zakresie obliczeń i systemów chmurowych na świecie.

III. Zasadnicze osiągnięcia Autora rozprawy

Rozprawa doktorska p. mgr inż. Joanny Kosińskiej jest oryginalną pozycją naukową zawierającą wiele wartościowych koncepcji, z których część przedyskutowano powyżej, dokonano ich implementacji oraz uzyskano wartościowe wyniki, wzbogacające wiedzę w tym obszarze. Podkreślić należy zaawansowany technologiczny sposób przedstawienia problematyki i rozwiązań szczegółowych, świadczący o bardzo dobrym przygotowaniu Autorki w tym zakresie.

Do najważniejszych osiągnięć rozprawy zaliczyć należy następujące elementy:

- Monograficzna identyfikacja elementów składających się na natywną aplikację chmurową wraz z krytyczną analizą technologii stosowanych w zarządzaniu takim środowiskiem.
- Projekt teoretycznego modelu aplikacji chmurowych wraz z propozycją zbioru metryk i cech autonomiczności realizowanych przez regułową bazę wiedzy i silnik wnioskujący. W przypadku autonomiczności wykorzystano wprost koncepcję self-CHOP pochodzącą od IBM, odpowiednio modyfikując ją do potrzeb pracy.
- Opracowanie i szczegółowe omówienie architektury AMoCNA wykorzystującej teoretyczny model natywnych aplikacji chmurowych i technologie mikroserwisowe. W architekturze przewidziano zwielokrotnienie zasobów z potencjalną korzyścią dla wydajności i ciągłości pracy systemu.
- Implementacja elementów architektury AMoCNA z wykorzystaniem serwisów typu Docker, szerokie wykorzystanie gotowych modułów środowiska Kubernetes z zyskiem dla szybkości implementacji i redukcji możliwych błędów.
- Zaplanowanie i wykonanie szeregu eksperymentów z wykorzystaniem opracowanego rozwiązania, w tym eksperymentów związanych ze skalowaniem wertykalnym i horyzontalnym, jego krytyczna ocena na podstawie przeprowadzonych testów i wskazanie możliwości rozwoju.

Wymienione osiągnięcia są oryginalne i znaczące, na nich więc opieram *ogólnie pozytywną ocenę pracy*. Stanowią one odpowiedź na postawione problemy badawcze i wspierają rozwój autonomicznych środowisk chmurowych dla potrzeb obliczeń aplikacji natywnych. Ze względu na wygodę wykorzystania środowisk chmurowych przez użytkownika i złożoność samych środowisk zastosowanie autonomiczności dla zarządzania aplikacjami natywnymi jest perspektywicznym kierunkiem rozwoju.

IV. Uwagi dyskusyjne i krytyczne

Oprócz niewątpliwych walorów, rozprawa zawiera pewną liczbę elementów, które mogą być przedmiotem dyskusji. Wydaje się celowe podanie następujących spostrzeżeń:

1. Motywacja pracy i przegląd stanu wiedzy są opracowane na zbyt wysokim poziomie ogólności. W Rozdziale 1 stwierdza się (i słusznie!) długą historię obliczeń

autonomicznych. Stwierdzenie to należałoby wesprzeć stosownym komentarzem (dotyczącym np. systemów typu HAC lub fault tolerant bądź też procesorów, np. PowerPC BQC). W Rozdziale 2.3 słabo nakreślono tło paradygmatu Autonomic Computing. Autorka powołuje się na paradygmat self-CHOP i prace pochodzące z IBM bez podania odnośnika literaturowego. Korzystnie byłoby podać publikacje[1][2], zwłaszcza, że z jednego z tych lub zbliżonych artykułów zapożyczono Rysunek 2.8. Warto byłoby też podkreślić rolę innych centrów technologicznych, np. Motoroli[3] lub publikacji[4] szczególnie, że przedstawiona tam architektura, reprezentacja wiedzy i świadomość kontekstu ('context awareness') są zagadnieniami pokrewnymi do rozprawy.

2. Na stronie 22 przeprowadzono krótką dyskusję technologii kontenerowych, wymieniając kilka podstawowych, ze wskazaniem na technologię Dockera. Brakuje dyskusji porównującej dwie ważne technologie: Docker i Singularity (zaproponowane i rozwinięte przez Lawrence Berkeley National Laboratory) Ponieważ potencjalnie Singularity może być zintegrowane z systemem Kubernetes, zatem otwiera to możliwość wykorzystania osiągnięć rozprawy również w środowisku HPC, gdzie autonomiczne skalowanie horyzontalne mogłoby być istotne. Zagadnienia autonomicznego zarządzania w środowiskach HPC stają się coraz bardzo znaczące z uwagi na rozwój tego typu środowisk w kierunku rozwiązań Exaskalowych (por. publikacja[5] dotycząca autonomiczności choć abstrahująca od konteneryzacji).
3. W rozprawie adresuje się jedną z cech rozwiązania, jakim jest 'context awareness'. W przeglądzie literatury nie przedyskutowano tego problemu, mimo istniejących pozycji bibliograficznych. Jako przykład można podać publikację[6], w której zawarto pewien model referencyjny wiedzy ustrukturalizowanej, który mógłby być przedmiotem dyskusji.
4. W przyjętym rozwiązaniu założono regułową reprezentację wiedzy. Jakie inne reprezentacje można byłoby uwzględnić i jakie argumenty przemawiały za dokonanym wyborem?
5. W Rozdziale 4 pokazano dwie realizacje modelu AMoCNA – z wykorzystaniem pojedynczego klastra i struktury wieloklastrowej (Rys. 4.6). Realizacja dotyczyła tylko pojedynczego klastra. Z czego wynikło to ograniczenie ? Jaki jest wpływ przyjętej realizacji na wnioski wypływające z rozprawy ?

---

[1] Brent Miller, „The Autonomic Computing Edge: The Role of Knowledge in Autonomic Systems", developerWork, IBM, 13.9.2005

[2] Sam Siewert, „Apply RAS architecture lessons to the autonomic self-CHOP roadmap", IBM, 6.7.2005

[3] John Strassner, et al., „A Context Aware Policy Model to Support Autonomic Networking", 2008 Annual Int. Computer Software and Applications Conf., 1097-1102

[4] Rahmira Rufus, et al., "An Autonomic Computing System based on a Rule-based Policy Engine and Artificial Immune Systems, 27th Modern Artificial Intelligence and Cognitive Science Conf. 2016, 105-108

[5] Kevin A. Huck, et al., „An Autonomic Performance Environment for Exascale", Supercomputing Frontiers and Innovations, 2 (3) (2015) 49-66

[6] Emil Vassev and Mike Hinchey, „Knowledge Representation and Awareness in Autonomic Service-Component Ensembles – State of the Art.", 2011 14th IEEE Int. Symp. on Object/component/Service-Oriented Real-Time Distributed Computing Workshops, 110-119

6. Rys. 6.4 przedstawia efekt wynikający ze skalowania zbliżonego do wertykalnego. Nie przedyskutowano jasno powodu wzrostu 'Frontend latency' po reorganizacji struktury obliczeń wynikającej z działania orkiestratora Kubernetesa (Rys. 6.4a, przedział 8:46-8:48).
7. W eksperymencie skalowania horyzontalnego system AMoCNA doprowadził do zwiększenia liczby węzłów (poprzez dodanie węzła 7). Udział węzła 7 jest niewidoczny wdrugiej części Rys. 6.5, jednak, jak wynika z analizy Rys. 6.6, powinien się on był zaznaczyć począwszy od chwili 9:35. Niemniej – biorąc pod uwagę obniżenie średniego poziomu wykorzystania CPU od tego momentu – węzeł 7 uczestniczy w przetwarzaniu. Jakiego efektu wydajnościowego można byłoby spodziewać się po zastosowaniu wbudowanych do Kubernetesa systemów skalowania wertykalnego, abstrahując od trudności technicznych wykorzystania tej możliwości ?
8. W podsumowaniu pracy brakuje szerszej dyskusji ograniczeń proponowanego podejścia i jego komputerowej implementacji w zakresie skalowania, czyli jednej z podstawowych cech autonomiczności. Czy możliwa jest łatwa adaptacja do implementacji w strukturze wieloklastrowej? Jaki byłby wpływ wprowadzonych rozwiązań cząstkowych (np. implementacji pojedynczej instancji serwera KIE – str. 89) ?
9. W przedstawianych algorytmach lub listingach występują niekiedy stałe (np. w Algorytmach 2 i 3, str. 63-64, w listingach A1 i A15). Wydaje się, że ograniczają one wprost zastosowanie opracowanych rozwiązań programistycznych, mimo zawartej deklaracji, cyt. (str. 121):"(…) valuable performance improvements (…) allow to state that the described solution is production-ready." – przy takim stwierdzeniu brakuje podania wartości TLR. Np. wprowadzono wartość 20 ms jako graniczną przy której następuje uruchomienie procedury skalowania mikroserwisu CNApp, a przecież wartość ta zależy od nakładu obliczeniowego CNApp (Algorytm 3, str. 64). Nb. występuje pewna niekonsekwencja, gdyż na str. 122 znajduje się stwierdzenie, cyt.: „(…) it should be kept in mind that AMoCNA is just a prototype, not a production-ready software and not every case is being implemented".

Należy wyraźnie zaznaczyć, że podane uwagi nie kwestionują słuszności przyjętych koncepcji ani też nie wpływają w sposób istotny na poznawcze i utylitarne wartości zrealizowanych badań. Ich uwzględnienie może okazać się korzystne w dalszej działalności naukowej dotyczącej zbliżonych zagadnień.

V. Wniosek końcowy

Wspomniane w recenzji uwagi polemiczne nie umniejszają zasług Autorki ani nie kwestionują jej osiągnięć. W przedstawianej Rozprawie postawiono ważny współcześnie problem i zaproponowano jego rozwiązanie w sposób świadczący o dojrzałości naukowej Autorki. Rozprawa stanowi odpowiedź na postawione problemy badawcze i wnosi istotny wkład w rozwój obliczeń rozproszonych w szczególności w zakresie cech autonomiczności. Tematyka dobrze wpisuje się we współczesny nurt rozwoju środków i narzędzi informatyki.

Warto nadmienić, że Autorka Rozprawy jest współautorem 6 publikacji w czasopismach z listy JCR bądź indeksowanych przez Scopus. Na uwagę zasługuje współautorska praca z 2012r opublikowana w IEEE Trans. Service Computing, o wysokim wskaźniku wpływu. Analiza danych bibliograficznych również wskazuje na stosowny dorobek na obecnym etapie rozwoju naukowego – liczba cytowań wynosi 22 i 28, h_index=2 wg Web of Science i Scopus odpowiednio (dostęp wrzesień 2019).

Stwierdzam zatem z przekonaniem, że opiniowana rozprawa Pani mgr inż. Joanny Kosińskiej spełnia w stopniu zadawalającym wymagania przewidziane dla rozpraw doktorskich w odpowiedniej Ustawie i stawiam wniosek o dopuszczenie jej Autorki do dalszych etapów przewodu doktorskiego.