

Recenzja rozprawy doktorskiej

mgr inż. Macieja Gawła

pt.

„Inteligentne zarządzanie usługami sieciowymi w wielowdrożeniowych środowiskach chmur obliczeniowych”.

Rozprawa została przekazana mi do recenzji w formie oprawionego maszynopisu (książki) liczącego 148 stron w języku angielskim, co dopuszczają odpowiednie zapisy w art. 13, ust. 2 i ust. 5 “Ustawy o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki” z dnia 14 marca 2003r. z późniejszymi zmianami (zwanej dalej w tej recenzji “Ustawą”). Moją szczegółową ocenę stopnia spełnienia przez Rozprawę trzech podstawowych wymogów stawianych rozprawom doktorskim w art. 13, ust.1 Ustawy i dotyczących kolejno oryginalności opisanego rozwiązania, wykazanej wiedzy teoretycznej Doktoranta oraz zademonstrowanej umiejętności samodzielnego prowadzenia przez niego pracy naukowej przedstawiam poniżej. W podsumowaniu wskazuję także na zauważone przeze mnie niedociągnięcia i słabe strony Rozprawy oraz formułuję wniosek dotyczący jej dalszego procedowania.

Problem rozprawy i ocena jego rozwiązania

1. Motywacja podjęcia badań i uzasadnienie tezy Rozprawy:

Zgodnie z informacją podaną na str. 16 Rozprawy badania Doktoranta były prowadzone w ramach programu “doktoratu wdrożeniowego” we współpracy z partnerem przemysłowym Samsung R&D Institute Poland i dotyczyły konkretnego problemu technologicznego. Jego rozwiązanie, zgodnie z brzmieniem art. 13 ust.1 Ustawy przedstawione w Rozprawie stanowi w mojej ocenie „oryginalne rozwiązanie problemu w oparciu o opracowanie projektowe, konstrukcyjne [lub] technologiczne” i całkowicie spełnia ten wymóg ustawowy. Wspomniany problem dotyczy jednego z najpoważniejszych wyzwań współczesnej informatyki, jakim jest zarządzanie funkcjonowaniem aplikacji masowego użytkownika uruchamianych na rozległej, niejednorodnej i dynamicznie zmiennej infrastrukturze obliczeniowej – od poszczególnych urządzeń Internetu Rzeczy po platformy wielochmurowe, tworzące swoiste kontinuum obliczeniowe od „brzegu do chmury”. Efektywne wykorzystanie przez taką aplikację dostępnych w danej chwili i w danych miejscu infrastruktury wykonawczej zasobów pamięciowych i obliczeniowych, jej prawidłowa reakcja na chwilowe awarie lub niedostępność poszczególnych elementów infrastruktury, jak też sprawne usuwanie błędów czy aktualizacje oprogramowania nie byłyby możliwe bez wykorzystania modelu architektury aplikacji opartej na mikrousługach i zastosowania techniki ich wirtualizacji opartej na kontenerach. Chociaż platforma Kubernetes (publicznie dostępna od kilku lat jako „open source” lub w postaci własnych dystrybucji innych dostawców) oferując mechanizmy do implementacji, eksploatacji i skalowania w oparciu o różne standardowe lub definiowane przez użytkownika metryki znacznie ułatwiła projektowanie aplikacji zbudowanych ze skonteneryzowanych usług i wdrażanie coraz bardziej złożonych rozwiązań, brakuje mechanizmów zarządzania dynamiczną migracją usług implementujących takie aplikacje podczas ich działania. Z tego względu Doktorant sformułował następującą tezę:

“Niezbędnym mechanizmem inteligentnego zarządzania systemem wieloklastrowym jest umożliwienie dynamicznej migracji usług podczas jego działania”.

Choć w kontekście treści całej Rozprawy takie sformułowanie tezy jest trafne i oddaje istotę opracowanego rozwiązania, to jej szczegółowa interpretacja przez czytelnika na wstępie Rozprawy (podrozdział 1.1.), tj. jeszcze przed zapoznaniem się z eksperymentalną oceną opracowanego rozwiązania przedstawioną w Rozdziale 7. pozostawia niepotrzebnie zbyt dużo swobody. Nie pomagają w tym zbytnio dość skąpe wyjaśnienia w komentarzu bezpośrednio pod tezą, że przez „inteligentne zarządzanie” Doktorant rozumie tylko proces adaptacyjny (bez odniesienia się do możliwości wykorzystania mechanizmów uczenia maszynowego) oraz że celem migracji jest zapewnienie odpowiedniej wartości metryki „Quality of Experience” QoE (zdefiniowanej dopiero na str. 103 Rozprawy). Ponadto w tezie jest mowa o migracji usług w systemie „wieloklastrowym”, a w

wyjaśnieniach umieszczonych poniżej niej wspomina się o migracji usług między „chmurami”. Dopiero lektura całej rozprawy pozwala rozjaśnić te wątpliwości interpretacyjne.

2. Istota problemu i jego aktualność

Ostatecznie rzecz ujmując, istota problemu podejmowanego w Rozprawie polega na opracowaniu platformy zarządzającej cyklem życia kontenerów podczas działania aplikacji chmurowej w środowisku wielochmurowym. Każda pojedyncza chmura w takim środowisku jest hostem dla kontenerów tworzących klastry, które w przypadku awarii, chwilowej niedostępności lub przeciążenia zasobów klastra musiałyby być alokowane na bieżąco do innych klastrów w sposób nie pogarszający działania aplikacji. Wykorzystanie kontenerów jest tu kluczowe, ze względu na możliwość przenoszenia usług implementujących daną aplikację między klastrami w trakcie trwania jej obliczeń. Z tego obrazu wyłania się nietrywialny problem automatycznej realokacji kontenerów między klastrami funkcjonującymi na różnych chmurach. Rozwiązanie tego problemu musi obejmować cały szereg rozwiązań szczegółowych, dotyczących komunikacji między klastrami, monitorowania wykorzystywanych przez nie elementów infrastruktury oraz zarządzania ich konfiguracją. Ich realizowalność w środowiskach wielochmurowych w oparciu o aktualnie dostępne platformy i narzędzia do wdrażania aplikacji chmurowych na pojedynczych chmurach nie jest z punktu widzenia aktualnego stanu wiedzy opisanego w literaturze przedmiotu oczywista i jako taka jest obecnie przedmiotem intensywnych badań wielu konsorcjów międzynarodowych. Z tym z kolei związany jest jeszcze jeden ważny aspekt rozwiązania przedstawionego przez Doktoranta w Rozprawie, jakim jest wykorzystanie wyłącznie platform oprogramowania typu open-source. Z wielu powodów, opis których przekracza zakres tej recenzji, zdominowały one współczesny rynek usług chmurowych, a bez open-source wdrażanie środowisk wielochmurowych, z wieloma różnymi dostawcami, byłoby praktycznie niemożliwe. Reasumując, podejmowany w Rozprawie problem wpisuje się w jak najbardziej aktualny trend budowania nowoczesnych infrastruktur wielochmurowych o zasięgu globalnym przy równoczesnym wykorzystaniu wirtualizacji, konteneryzacji, mikrousług i oprogramowania open-source.

3. Metody rozwiązania problemu

Rozwiązanie postawionego w Rozprawie problemu technologicznego zostało szczegółowo opisane przez Doktoranta w Rozdziale 6., a jego eksperymentalna ocena przedstawiona w Rozdziale 7. Na podstawie poprzedzających te rozdziały Rozdziałów od 2. do 5. można bez wątpliwości zakwalifikować opisane rozwiązanie jako oryginalne osiągnięcie projektowe i technologiczne, stanowiące istotną innowację produktową w pełni zdatną do wdrożenia na globalnym rynku usług chmurowych.

Bazę do opracowania tego rozwiązania stanowią dwa modele logiczne opisane kolejno w Rozdziałach 2. i 3. Każdy z tych modeli ujmuje w syntetyczny sposób aktualny stan rozwoju/zaawansowania technologii kluczowych dla opisanego w Rozprawie rozwiązania i porządkuje związane z nimi pojęcia. I tak, model logiczny aplikacji chmurowej opisany w Rozdziale 2. uzasadnia konieczność budowy aplikacji (usługi) chmurowej jako dynamicznej struktury powiązanych ze sobą mikrousług uruchamianych w kontenerach na wielu klastrach z punktu widzenia rewolucyjnej koncepcji technologii 5G, zakładającej możliwość budowania aplikacji do dowolnych zastosowań i o zróżnicowanych wymaganiach komunikacyjnych w oparciu o tylko jedną wspólną sieć fizyczną. Z kolei model logiczny wieloklastrowej infrastruktury wykonawczej opisany w Rozdziale 3 objaśnia strukturę środowiska wykonawczego dla wspomnianych aplikacji chmurowych, od ogólnej topologii infrastruktury wielochmurowej poprzez platformy orkiestracji kontenerów po architekturę Service Mesh. W tej ostatniej obiekty „sidecar” (serwery proxy, każdy sparowany z obsługiwaną przez niego mikrousługą) tworzą siatkę powiązań warstwy „data plane” znajdującą się pod kontrolą obiektów warstwy „control plane”. Te ostatnie mogą konfigurować usługi podległej jej warstwy „data plane” oraz zbierać i gromadzić w jednym miejscu metryki dotyczące ich działania. Oba wymienione wyżej modele logiczne stanowią dobre wprowadzenie do rozważań nt. przyjętej w Rozdziale 5. strategii rozwiązania problemu Rozprawy w odniesieniu do zagadnienia dynamicznej migracji usług w wieloklastrowym środowisku wykonawczym. Doktorant wyróżnia trzy kluczowe wyzwania z tym

związane, a mianowicie zapewnianie łączności pomiędzy klastrami (ang. network connectivity), monitorowanie ich działania (ang. observability) oraz zarządzanie i orkiestrację usług (ang. service management and orchestration). Dla każdego z nich przeprowadza kompleksowy przegląd wyników całego szeregu projektów typu „open source” (wymienionych odpowiednio w Tabelach 5.1, 5.2 i 5.3) oraz wykazuje ich niedostatki z punktu widzenia założeń wcześniej przedstawionych modeli logicznych. W oparciu o ten przegląd możliwe jest ostateczne sformułowanie wymagań dla opracowanej w Rozprawie platformy AMS (podrozdziały 5.3.3, 5.4.3 i 5.5.3).

Z kolei zagadnienie inteligentnego zarządzania systemem wieloklastrowym występujące w problemie Rozprawy zostało przeanalizowane przez Doktoranta w Rozdziale 4. Tutaj główną osią rozważań są dwa standardy: specyfikacja ETSI MANO trzech kluczowych komponentów referencyjnej architektury systemu zarządzania wirtualizacją usług oraz referencyjna architektura MAPE-K autonomicznego systemu sterowania adaptacyjnego (ang. self-adaptive system). Odwołując się do ETSI MANO Doktorant wykazał, że metryki i polityki stosowane aktualnie w mechanizmach platformy wirtualizacji K8 używanej przez wszystkich liczących się na świecie graczy rynku usług chmurowych nie nadają się do zarządzania wirtualizacją usług w przypadku wieloklastrowego środowiska wykonawczego. Z kolei w oparciu o MAPE-K sformułował kryteria jakie powinien spełniać autonomiczny system zarządzania wykonaniem usług. Posłużył się nimi następnie do kompleksowej oceny całej gamy znanych rozwiązań dotyczących dynamicznej realokacji usług chmurowych (od wcześniejszych wykorzystujących hipernadzorcę po najbardziej współczesne wykorzystujące kontenery) pod kątem ich wykorzystania w środowisku wieloklastrowym. Z bardzo wyczerpującego przeglądu zamieszczonego na zakończenie Rozdziału 4. wynika, że do chwili obecnej praktycznie żadne zaproponowane w literaturze rozwiązanie dynamicznej realokacji usług nie potrafi w określonych sytuacjach zagwarantować autonomii systemowi zarządzania wirtualizacją w wieloklastrowym środowisku wykonawczym, tj. skutecznie wyeliminować konieczność ingerencji operatora.

Zwieńczeniem tych rozważań jest prezentacja w Rozdziale 6. osiągnięcia projektowego Doktoranta w postaci systemu AM. Doktorant definiuje tam formalnie problem realokacji mikrousług, przedstawia koncepcję rozwiązania procesu ich migracji między klastrami silnie uwarunkowaną specyfiką platformy K8 (tj. koniecznością implementacji migracji usług poprzez ich odtwarzanie na klastrach docelowych) oraz opisuje konstrukcję swojego systemu AMS bazującego na ogólnym schemacie komponentów architektury MAPE-K systemu autonomicznego. Dla każdego wymaganego standardowo przez ten schemat komponentu (*Monitor*, *Analyzer*, *Planner* i *Executor*) uzasadnia szczegółowo swoje poszczególne decyzje projektowe, odwołując się do wyników analizy aktualnego stanu technologii chmurowych przeprowadzonej wcześniej w Rozdziałach 2. i 3., ujętych w postaci wspomnianego modelu logicznego aplikacji chmurowej i modelu wieloklastrowej infrastruktury wykonawczej. Należy podkreślić, że konstrukcja systemu AMS opracowana przez Doktoranta stanowi twórcze rozwinięcie schematu architektury MAPE-K. Polega ono na uzupełnieniu jej o dodatkowy komponent *Controller* odpowiedzialny za podejmowanie decyzji o uruchomieniu procesu adaptacji (realokacji usług). Oryginalnym rozwiązaniem zaimplementowanym w tym komponencie jest wykrywanie anomalii w trakcie obliczeń oparte na statystycznej analizie rejestrowanych periodycznie próbek stanu monitorowanego zasobu w pewnym okienku czasowym i charakteryzowanie jego zmian za pomocą standaryzacji Z (ang. z-score). Gdy wartość monitorowanej metryki wychodzi poza dopuszczalny zakres zmian znormalizowanej w ten sposób zmiennej losowej opisującej dany zasób ogłaszany jest alarm i uruchamiane są działania naprawcze w postaci realokacji usług. Mechanizm ten został zaimplementowany z wykorzystaniem ogólnie dostępnego narzędzia open-source Prometheus. Wreszcie komponent *Knowledge*, także wymagany przez schemat MAPE-K, został zaimplementowany w systemie AMS jako nierelacyjna baza danych przechowująca wszystkie kolejne grafowe reprezentacje modelu logicznego aplikacji i modelu wieloklastrowej infrastruktury wykonawczej budowane na bieżąco przy każdej wykonywanej w systemie realokacji oraz katalog opisów wszystkich aktualnie dostępnych obiektów (mikrousług) poszczególnych klastrów. Wiedza ta wykorzystywana jest przez moduł *Analyzer*, do uruchomienia automatycznej rekonfiguracji systemu i wznowienia obliczeń aplikacji chmurowej. Oryginalność tego rozwiązania polega na skupieniu całości informacji o stanie obliczeń aplikacji nadzorowanej przez AMS tylko w jednym węźle chmury, co powinno ułatwiać zapewnienie skalowalności aplikacji funkcjonującej na wielu chmurach. Nie mniej należałoby zadbać

nad zabezpieczeniem systemu przed upadkiem tego wrażliwego komponentu, np. poprzez jego zdublowanie.

W Rozdziale 7. Doktorant przeprowadził ocenę praktyczną swojego rozwiązania. Skonfigurował środowisko testowe z wykorzystaniem ogólnodostępnych platform typu open-source w postaci infrastruktury wykonawczej obiektów reprezentujących odpowiednio 11 klastrów, każdy z trzema wirtualnymi węzłami w konfiguracji 1 master i 2 slave'y, każdy dysponujący 5-cioma CPU, typowymi zasobami pamięci RAM i HDD, z dostępem do dobrego łącza sieciowego. Na dodatkowym 12. klastrze uruchomiony został testowany system AMS. Dla tak skonfigurowanego środowiska przygotował trzy syntetyczne aplikacje użytkowe wymagające różnej topologii mikrousług: wielopoziomowego drzewa, grafu acyklicznego o nieregularnej strukturze oraz klastrów połączonych szeregowo. Przeprowadził z nimi szereg eksperymentów dla łącznie pięciu realistycznych scenariuszy testowych: utraty poszczególnych zasobów infrastruktury (awaria pojedynczego węzła w klastrze, awaria całego klastra, degradacja przepustowości łącza sieciowego klastra), błędów w działaniu samego systemu monitorowania (anomalia zachodzących w procesie obliczania znormalizowanych wartości metryk) oraz klasycznej optymalizacji wykorzystania chmurowej infrastruktury wykonawczej (związanych ze zmianami obciążenia infrastruktury wykonawczej przez wykonywaną aplikację). Przeprowadzone eksperymenty udokumentował wykresami wizualizującymi uzyskane wyniki. Na każdym z nich uwidoczniiony jest przebieg w czasie wartości chwilowych danej metryki charakteryzującej monitorowany zasób oraz trendy zmian średniej i odchyłki standardowej tych wartości rejestrowanych w ruchomym okienku czasowym. Zestawienie ich na jednym wykresie pozwoliło Doktorantowi przekonująco zilustrować działanie całego opracowanego przez siebie mechanizmu – widać wyraźnie jak po przełamaniu linii trendu zmian odchyłki standardowej przez wartości metryki monitorowanej (tj. wykrycia anomalii) system reaguje wykonując działania naprawcze (realokację usług). Działania te są całkowicie autonomiczne, tj. nie wymagają wykonywania żadnych dodatkowych czynności przez operatora. Również czas ich wykonania jest względnie krótki, w porównaniu z czasem wykonywania aplikacji. Całość prezentacji wyników dopełniają wyczerpujące komentarze Doktoranta do każdego wykresu. Oceniając ten kluczowy fragment Rozprawy mógłbym dodać tylko, że dywagacje na początku Rozdziału 7. nt. weryfikacji i walidacji, w połączeniu z cytowaniem poz. [58] dot. inżynierii chaosu uważam za nietrafione. Wystarczyło sięgnąć do klasyków – Barry Boehm weryfikację określił prosto jako czynność odpowiadająca na pytanie, czy robimy system dobrze, a walidację jako czynność odpowiadająca na pytanie, czy zrobiliśmy dobry system. W mojej opinii Doktorant na oba te pytania odpowiedział w Rozprawie twierdząco – zrobił go dobrze, o czym świadczą Rozdziały 2-6 i jest to dobry system, o czym świadczy Rozdział 7. Nie mam wątpliwości, że rozmiar wykorzystanej do eksperymentów infrastruktury oraz całe spektrum zrealizowanych scenariuszy tworzą solidne podstawy do walidacji systemu AMS

4. Oryginalność i przydatność dorobku rozprawy

Do osiągnięć Doktoranta opisanych w Rozprawie zaliczam:

1. Oryginalny projekt architektury systemu AMS zarządzania wykonaniem aplikacji chmurowych w wieloklastrowej infrastrukturze wykonawczej, która:
 - a. umożliwia automatyczną dynamiczną realokację mikrousług składowych aplikacji podczas jej wykonywania, niezależnie od jej topologii i wykorzystywanych zasobów (system AMS eliminuje konieczność ingerencji operatora w przypadku wystąpienia anomalii),
 - b. może zostać zaimplementowana z wykorzystaniem standardowych i powszechnie używanego przez dostawców chmur obliczeniowych platform i oprogramowania typu open-source (system AMS jest przenośny),
 - c. rozszerza architekturę referencyjną MAPE-K systemu sterowania adaptacyjnego o moduł detekcji anomalii na podstawie analizy trendów zmian monitorowanych parametrów (system AMS zapewnia niezawodność działania monitorowanej aplikacji).
2. Walidację opracowanej architektury dla zbioru realistycznych i reprezentatywnych scenariuszy testowych (system AMS nadaje się do wdrożenia w chmurowych środowiskach produkcyjnych).

3. Krytyczny przegląd wyników najbardziej znanych i aktualnych projektów B+R oprogramowania chmurowego open-source dotyczących zapewniania łączności pomiędzy klastrami, monitorowania ich działania oraz zarządzania i orkiestracji usług pod kątem możliwości ich wykorzystania w wieloklastrowej infrastrukturze wykonawczej.

Wymienione wyżej osiągnięcia Doktoranta stanowią oryginalne i właściwe rozwiązanie problemu postawionego w Rozdziale 1. Rozprawy.

Zademonstrowana w rozprawie wiedza autora i znajomość literatury

Lektura wszystkich rozdziałów Rozprawy, w tym szczególnie Rozdziału 5. świadczy o głębokiej i aktualnej wiedzy Doktoranta w zakresie projektowania i wdrażania wysoce dostępnych rozwiązań w oparciu o wielochmurowe środowiska obliczeniowe, zarządzania ich zasobami oraz rozwoju ich oprogramowania w formule „open-source”. Doktorant bardzo trafnie zidentyfikował trzy podstawowe wyzwania stojące obecnie przed technologiami chmurowymi na świecie (*network connectivity, observability, service management and orchestration*), które wynikają zarówno z aktualnie dominującej na rynku tendencji do zapewniania firmom daleko idącej swobody w korzystaniu z usług chmur wielu różnych dostawców równocześnie, dopasowywanych dynamicznie do aktualnego zapotrzebowania wykonywanych aplikacji, jak i opisanych w Rozprawie nowych możliwości oferowanych przez technologię 5G. Dla każdego z tych wyzwań sformułował szczegółowe problemy jakie należy rozwiązać w celu wdrożenia aplikacji chmurowej w środowisku wieloklastrowym, przeprowadził krytyczną analizę dorobku najbardziej znanych projektów podejmujących te problemy, dokonał kompleksowej oceny ich rezultatów, wskazał na ich ewentualne braki oraz sformułował wnioski dotyczące przyjętej w Rozdziale 6. własnej strategii ich rozwiązania. Oprócz wspomnianego przeglądu światowego dorobku projektowego w obszarze zagadnień podejmowanych w Rozprawie Doktorant dokonał także w Rozdziale 4. przeglądu literatury światowej dotyczącej metod inteligentnego zarządzania zasobami chmur obliczeniowych. Cytowane tam publikacje są aktualne i znaczące, a ich wybór świadczy o dobrej znajomości literatury przedmiotu przez Doktoranta.

Umiejętność samodzielnego prowadzenia pracy naukowej

Redakcja Rozprawy i sposób prowadzenia wywodu świadczy o biegłości Doktoranta w planowaniu, realizacji i ocenie wyników złożonego przedsięwzięcia o charakterze B+R, którego celem jest znacząca innowacja produktowa. Oprócz wiedzy technicznej zademonstrowanej w Rozprawie wykazał się dużą dozą inżynierskiej intuicji, która w połączeniu z krytyczną oceną opracowanego rozwiązania przedstawioną w Rozdziale 8. świadczy o jego zdolności do samodzielnego formułowania i rozwiązywania problemów technologicznych w oparciu o oryginalne i własne opracowanie projektowe, nadające się do wdrożenia w skali całej branży dostawców technologii chmurowych. W szczególności lista 11 zagadnień „do dalszego opracowania” zaproponowanych na zakończenie Rozprawy przekonuje mnie o tym, że dokonania Doktoranta nie mają ani charakteru tylko „odtwórczego” (tj. nie ulepszają jakichś znanych wcześniej rozwiązań) ani „przyczynkowego” (nie stanowią li tylko pewnego rozwiązania cząstkowego mającego sens dopiero w jakimś większym przedsięwzięciu B+R). W tym kontekście wynik Rozprawy można zakwalifikować jako samodzielnie przeprowadzony „Proof of Concept”, którym Doktorant posłużył się do wykazania realnych możliwości rozwiązania postawionego przez siebie problemu przy użyciu aktualnie dostępnych technologii oraz do wskazania dalszych możliwości doskonalenia opracowanego przez siebie rozwiązania.

Niedociągnięcia i słabe strony Rozprawy

Doktorant posługuje się językiem angielskim sprawnie, stosowane formy gramatyczne, słownictwo, interpunkcja, itp. są bez zarzutu. Spośród innych drobnych uwag dotyczących redakcji Rozprawy jakie nasunęły mi się podczas jej lektury mógłbym wymienić:

- Określenie „*Single pane of glass*” użyte na str. 63 w czystej postaci (bez dodatkowej adnotacji w nawiasie, jak np. „*uniform display*”) brzmi nieco zbyt żargonowo jak na rozprawę naukową.
- Na wielu rysunkach w Rozdziale 7. osie poziome wyskalowane zostały w sekundach, podczas gdy umieszczone w tekście komentarze dotyczące przebiegu wykreślonych tam funkcji odwołują się do

minut. To sprawia, że interpretacja niektórych wykresów przez czytelnika wymaga niepotrzebnie dodatkowego wysiłku.

- Czytelność wykazu narzędzi w tabelach 5.1, 5.2 i 5.3 byłaby większa, gdyby dodano do nich także kolumny z datą wydania każdego z nich oraz odsyłaczem do strony macierzystej. W tym kontekście nie jest dla czytelnika jasne, dlaczego w tekście omówiono dokładniej tylko część z wymienionych tam narzędzi.
- Ze względu na znaczną liczbę wymienionych w tekście rozprawy nazw różnego rodzaju pakietów oprogramowania przydałby się ich słownik obok listy skrótów czy spisu rysunków.
- Wymagane komponenty głównej pętli standardu MAPE-K zostały omówione w p. 6.2 w innej kolejności niż w p. 6.3.2. Zachowanie kolejności zgodnej ze standardem, tj. tak jak w p. 6.3.2, oprócz zachowania porządku estetycznego pozwoliłoby mocniej uwypuklić rolę komponentu *Controller* w prezentowanym w Rozprawie rozwiązaniu.

Z merytorycznego punktu widzenia do treści rozprawy zgłaszam następujące uwagi:

- W Rozprawie została zacytowana tylko jedna publikacja Doktoranta z 2019 roku. Na tym etapie prac należałoby się spodziewać publikacji podsumowującej wyniki eksperymentów z Rozdziału 7., które są już zdecydowanie warte upublicznienia na forum międzynarodowym.
- O w pełni inteligentnym zarządzaniu realokacją usług można by mówić, gdyby wielkość okna czasowego do wyliczania parametrów statystycznych do standaryzacji Z nie była ustalana przez operatora (jak napisano na str. 97), a automatycznie przez odpowiednio wytrenowany predyktor. Jakie mechanizmy uczenia maszynowego mogłyby tu znaleźć realistycznie zastosowanie?
- W eksperymentach opisanych w Rozdziale 7. Doktorant wykorzystał aplikacje syntetyczne. O jakiej skali można by mówić w przypadku aplikacji rzeczywistych (szczególnie z obszaru Big Data) i jakie rozmiary mógłby wówczas osiągać komponent *Knowledge*?
- Czy i jakie ograniczenia dotyczące rodzaju zbieranych danych może napotkać implementacja komponentu *Monitor* na chmurach dostawców komercyjnych, takich jak Google, Amazon czy Microsoft?
- Rozważania nt. realokacji usług pomiędzy klastrami prowadzone są w Rozprawie z punktu widzenia działania tylko jednej aplikacji chmurowej. Jak w modelu przyjętym przez Doktoranta należałoby uwzględnić sytuację, gdy realokacji podlegałyby usługi niezbędne do realizacji wielu równocześnie wykonywanych aplikacji i które by konkurowały między sobą o dostęp do określonych zasobów?

Nadmieniam, że powyższe uwagi mają jedynie charakter polemiczny i mogłyby stanowić podstawę do ew. dyskusji podczas publicznej obrony.

Wniosek końcowy

Moje uwagi krytyczne wyliczone powyżej oraz mój wcześniejszy komentarz dotyczący (tylko chwilowych, jak wyjaśniłem na początku mojej recenzji) niejasności dotyczących tezy w żadnym stopniu nie wpływają na moją wysoką ocenę recenzowanej Rozprawy. Z przekonaniem stwierdzam, że spełnia ona całkowicie wymagania stawiane w Ustawie rozprawom doktorskim i wnoszę o jej dopuszczenie do publicznej obrony.



prof. dr hab. inż. Bogdan Wiszniewski