Prof. dr hab. inż. Stanisław Kozielski
Instytut Informatyki
Politechnika Śląska

Gliwice, 10.07.2014 r.

Recenzja

rozprawy doktorskiej mgr inż. Piotra Pęgla pt.: „Self-healing, highly available monitoring system for distributed environments"

Promotor rozprawy: prof. dr hab. inż. Jacek Kitowski
Promotor pomocniczy: dr inż. Włodzimierz Funika

1. Aktualność i znaczenie tematyki rozprawy

Budowane współcześnie systemy informatyczne to w wielu przypadkach systemy bardzo złożone, wykorzystujące wiele różnych technologii, gotowych komponentów, tworzone często dla środowisk rozproszonych. Nadzorowanie i prowadzenie takich systemów jest trudnym zadaniem, dlatego bardzo cenne są wszelkie rozwiązania wspierające proces ich monitorowania. Naturalnym jest przy tym oczekiwanie, że narzędzia monitorujące będą niezawodne, inaczej mówiąc będą należały do klasy systemów wysokiej dostępności, minimalizujących ryzyko błędnego działania. Gdyby jednak taka sytuacja wystąpiła, to można również oczekiwać, że system monitorujący samodzielnie wykryje ten stan i poprawnie rozwiąże problem, czyli będzie miał zdolność samonaprawy. Opracowano już szereg systemów monitorujących, zazwyczaj mocno ukierunkowanych na potrzeby monitorowanych aplikacji, jednak systemy te w minimalnym tylko stopniu posiadają cechy wysokiej dostępności i praktycznie, poza wyjątkami (np. systemy PANACEA oraz Nagios) nie mają zdolności samonaprawy.
W opiniowanej rozprawie podjęto taka właśnie tematykę. Autor analizuje metody i techniki, które mogą zapewnić systemowi monitorującemu możliwości samonaprawy i własności wysokiej dostępności. Co więcej, proponowane rozwiązania wspierają przeprowadzenie samonaprawy również w systemie monitorowanym. Dla sprawdzenia opracowanych koncepcji autor zaprojektował i zaimplementował system monitorujący AgeMon, o oryginalnej architekturze i złożonej funkcjonalności. Możliwości tego systemu zostały sprawdzone w serii eksperymentów.

Podjęcie przez autora opiniowanej rozprawy takiej tematyki badań oceniam jako wybór zadań bardzo aktualnych, a przy tym trudnych. Uważam, że rozprawa dotyczy ważnej i aktualnej problematyki badawczej.

2. Zakres pracy

Praca składa się z 9 rozdziałów. W rozdziale pierwszym autor przedstawił motywacje podjęcia badań i zdefiniował podstawowe dla tematyki rozprawy pojęcia takie jak: wysoka dostępność, samonaprawa, system agentowy, niezawodność. Pojęcia te zostały omówione w bardzo szerokim kontekście, z wieloma odniesieniami do źródłowych publikacji, co potwierdza dobrą znajomość przez autora literatury w podejmowanej dziedzinie. Została również przedstawiona następująca teza pracy:

System monitorujący samonaprawiający (ang. self-healing) umożliwiający automatyczną naprawę systemu monitorowanego pozwala na efektywne zarządzanie złożonym systemem rozproszonym oraz podniesienie jego dostępności.

W rozdziale określono też oczekiwane cechy samonaprawiającego się systemu monitorującego AgeMon, który autor postanowił zrealizować dla weryfikacji postawionej tezy. Rozdział kończy krótkie omówienie organizacji całej rozprawy.

Drugi rozdział pracy jest poświęcony różnym technologiom, metodom i projektom, które mogłyby być użyteczne dla projektowanego w pracy systemu monitorującego. Na wstępie przedstawiono przegląd algorytmów i technologii lokalizacji usług oraz agentów w środowiskach rozproszonych. Kolejno rozważono algorytmy wyboru lidera w systemach komputerowych, np. wyboru po awarii w środowisku rozproszonym nowego agenta obsługującego bazę danych. Następnie dokonano przeglądu istniejących systemów monitorujących, podkreślając m.in. brak mechanizmów dających możliwość samonaprawy tych systemów. Natomiast w ostatniej części tego rozdziału autor przedstawił trzy systemy, z których jeden, PANACEA, wyróżnia się możliwościami samonaprawy, a dwa inne posiadają możliwości samo-adaptacji w zmieniającym się otoczeniu.

W trzecim rozdziale pracy autor rozważa przypadki użycia i definiuje wymagania dla projektowanego samonaprawiającego się systemu monitorującego AgeMon. Analizując przypadki użycia autor wyróżnił trzy kategorie użytkowników takich systemów: typowy użytkownik odpowiedzialny za monitorowanie aplikacji, administrator systemu wykonujący złożone zadania monitorowania i programista aplikacji, które będą monitorowane. Następnie zdefiniowano kilkanaście wymagań funkcjonalnych (na czele tej listy znalazło się wymaganie zdolności systemu do samonaprawy) oraz wymagań niefunkcjonalnych (z oczekiwaniem architektury wysokiej dostępności na pierwszym miejscu).

W rozdziale czwartym autor przedstawił, bazując na wymaganiach zdefiniowanych w rozdziale poprzednim, architekturę, a następnie szczegóły projektu samonaprawiającego się systemu monitorującego. Kluczową dla architektury systemu AgeMon jest koncepcja ról, użytych do grupowania wymaganych w systemie funkcjonalności. Zdefiniowano pięć typów ról: *Monitoring Role* – używaną do wyszukiwania monitorowanych danych oraz interakcji z monitorowanym systemem , *GUI Role* – zapewniającą interaktywną komunikację z administratorem systemu oraz wyświetlanie monitorowanych danych, *CLI Role* – udostępniającą zaawansowane funkcje używane przez administratora systemu, *Persistence Role* – zapamiętującą na trwałe wyniki monitorowania, *Rule Role* – określającą reguły, według których podejmowane są działania stosownie do wykrytych zmian w stanie monitorowanego systemu. Wszyscy agenci są zdolni do wykonywania jednej lub kilku ról. Role mogą być dynamicznie uruchamiane bądź zatrzymywane, co daje podstawy do stworzenia cech samonaprawy systemu. Jako drugą podstawową część systemu, obok ról, wskazano warstwę komunikacji agentów. Realizuje ona dwie główne funkcje: odszukiwania agentów tej samej grupy oraz komunikację między nimi. Pierwsza funkcja umożliwia najpierw zarejestrowanie agenta w określonej grupie agentów, a następnie udostępnia informacje o wszystkich agentach grupy i w ten sposób zapewnia automatyczny dostęp do całej grupy. Druga funkcja pozwala agentom na wymianę komunikatów z innymi agentami. Możliwe jest wysłanie komunikatu do jednego agenta oraz do wszystkich agentów danej grupy.

Treść rozdziału czwartego pozwala bardzo wysoko ocenić znajomość tematyki podejmowanej przez autora oraz przedstawiony proces projektowania systemu monitorowania. Autor wykazał się głęboką znajomością metod, technologii, wzorców projektowych i narzędzi przydatnych do zrealizowania systemu. W procesie projektowania autor przedstawił i ocenił szereg wariantów alternatywnych, rozważył niezawodność proponowanych rozwiązań,

zwracając m.in. uwagę na eliminację tzw. pojedynczych punktów podatności na awarię (ang. *single point of failure*).

W rozdziale piątym została przedstawiona pierwsza, główna, część implementacji sytemu monitorującego. W pierwszej kolejności autor przedstawił realizację agenta i grupy agentów. Główną częścią obiektowo zorientowanej implementacji agenta jest interfejs, obejmujący dwa rodzaje metod: służących do opisu samego agenta oraz komunikacji z innymi agentami. Implementacja agenta ma dwie postaci: agenta lokalnego, reprezentującego stan aktualnego agenta oraz agenta zdalnego, reprezentującego dowolnego agenta z grupy agentów.

Główną część rozdziału piątego stanowi szczegółowy opis implementacji każdej z pięciu ról tworzących trzon systemu AgeMon. Dla każdej z ról omówiono najpierw technologie wykorzystane do ich implementacji. W opisie implementacji wykorzystano różne formy graficznej ilustracji: diagramy klas, grafy agentów wykonujących dane role, komponenty graficznego interfejsu dla komunikacji z użytkownikiem, W procesie tworzenia kodu programów wykorzystano znane praktyki najlepszego kodowania, np. SOLID,YAGNI, KISS. Do implementacji wykorzystano dobrze znane, nowe technologie, takie jak m.in. Java Swing i biblioteka JGraph (rola GUI), JMX (rola monitorująca), Java Scripting Engine (rola regułowa), Guice DI framework (wszystkie role).
Poza szczegółami implementacji poszczególnych ról w ostatniej części rozdziału piątego przedstawiono kilka mechanizmów wykorzystywanych w całym systemie AgeMon. Należą do nich interfejs usług realizowanych na potrzeby agentów wraz z procedurą uruchamiania i zamykania usług w trakcie uruchamiania bądź kończenia pracy agentów; budowa usług wykorzystywanych w wielu złożonych komponentach systemu, a także wzorzec projektowy „wstrzykiwania zależności" (*Dependency Injection*) upraszczający projektowanie powiązanych komponentów programów.

Rozdział szósty prezentuje szczegóły koncepcji oraz implementację drugiej części systemu monitorującego AgeMon, obejmującej warstwę komunikacji między agentami tego systemu. Składa się ona z trzech części: warstwy transportowej, warstwy abstrakcyjnego agenta oraz warstwy abstrakcyjnego monitorowania. Warstwa transportowa, stanowiąca najniższy poziom komunikacji, umożliwia przesłanie komunikatu pod określony fizyczny adres: pojedynczy lub wiele adresów uczestników grupy. Warstwa abstrakcyjnego agenta ukrywa niskopoziomowe adresowanie, wprowadza koncepcję agentów i definiuje interfejsy agenta i grupy agentów, z metodami ułatwiającymi przesyłanie komunikatów. Najwyższa warstwa abstrakcyjnego monitorowania definiuje pojęcie komunikatu i procesora komunikatów.

W drugiej części rozdziału szóstego omówiono kilkadziesiąt typów komunikatów używanych w systemie AgeMon. Dla każdego komunikatu przedstawiono diagram prezentujący role, które mogą wysyłać dany komunikat oraz role, które mogą przetwarzać dany komunikat.

Pierwsza część rozdziału siódmego przedstawia mechanizmy zaimplementowane w systemie AgeMon w celu zapewnienia jego wysokiej dostępności oraz samonaprawy. Omówiono m.in. następujące mechanizmy lub cechy systemu: automatyczne odszukiwanie agentów, niezawodny protokół transportowy, tolerancja błędów sieci, brak pojedynczych punktów podatności na awarie, redundancja ról, procedura zastępowania niedziałającego agenta, kooperacyjny tryb analizy i wykonania reguł, wykorzystanie złożonych reguł do samonaprawy. W podsumowaniu przedstawiono strategie wykorzystania poszczególnych, zdefiniowanych w systemie ról do uzyskania cech wysokiej dostępności i samonaprawy systemu AgeMon.

W drugiej części rozdziału siódmego rozważono możliwości wykorzystania systemu monitorującego AgeMon do naprawy systemu monitorowanego (w szczególności aplikacji monitorowanej). Wśród rozwiązań transparentnych (kiedy monitorowany system „nie zdaje sobie sprawy" z działań systemu monitorującego) przeanalizowano m.in.: restartowanie

aplikacji po zatrzymaniu w wyniku błędu, uruchamianie aplikacji w określonych okolicznościach, wsparcie w rozwiązywanie problemu braku miejsca na dysku przy archiwizacji danych, wsparcie systemu operacyjnego przy problemach z dyskiem lub restartowanie karty sieciowej po rekonfiguracji sieci, a wreszcie wysłanie e-maila do administratora, jeśli żadna automatycznie wykonywana akcja nie wystarczy do naprawy systemu. Rozpatrzono też przykłady możliwości przekazywania z systemu AgeMon informacji do monitorowanej aplikacji, czyli integracji, w pewnym zakresie, systemu AgeMon z monitorowaną aplikacją. Taka aplikacja, świadoma funkcjonowania systemu monitorującego, może wtedy wykorzystać te informacje do naprawy swego działania. Rozważono przykłady zarządzania buforem *cache*, równoważenia obciążenia dla aplikacji obejmującej wiele serwerów JBoss, a także optymalizacji niektórych parametrów *run-time'u* aplikacji, np. rozmiar puli wątków, rozmiar puli połączeń bazy danych, rozmiaru buforów *cache* itp.

Rozdział ósmy pracy został poświęcony badaniom eksperymentalnym, które miały na celu ocenę wybranych cech systemu AgeMon. Autor skupił uwagę na testowaniu wymagań niefunkcjonalnych. Pierwsze badania dotyczyły wydajności systemu, mierzonej liczbą komunikatów z danymi wysyłanych przez kilkunastu lub kilkudziesięciu agentów monitorujących i czasem ich zapisu w bazie danych przez jednego agenta zapisującego na trwałe wyniki monitorowania. Drugi eksperyment umożliwił ocenę wydajności nisko poziomowej warstwy komunikacji. Trzecia grupa eksperymentów pozwoliła na ocenę czasów opóźnień lub, inaczej mówiąc, czasów działania różnych części systemu AgeMon. Wyznaczono czas podejmowania decyzji w oparciu o stan monitorowanego systemu, oceniono czas zapisu na trwałe wyników monitorowania, oceniono też czas tracony w systemie AgeMon na komunikację w sieci komputerowej.

Druga część badań służyła sprawdzeniu tezy, że zastosowane techniki samonaprawy mogą być użyte do zbudowania systemu o wysokiej dostępności. W przeprowadzonych eksperymentach potwierdzono poprawne działanie systemu AgeMon przy symulowanej awarii agenta zapisującego na trwałe wyniki monitorowania oraz agenta monitorującego stan systemu. Przedyskutowano też wybrane dwie możliwości wykorzystania mechanizmów AgeMon do naprawy systemu monitorowanego. W oddzielnym eksperymencie sprawdzono poprawność działania systemu przy bardzo dużym obciążeniu trwającym przez 14 dni – nie zanotowano utraty żadnego komunikatu, co pozwala wysoko ocenić niezawodność systemu.

W ostatniej części tego rozdziału przedstawiono i przedyskutowano wyniki oceny jakości kodu systemu AgeMon. Ocenę taką przeprowadzono przy użyciu analizatora kodu Sonar i uzyskano wyniki lepsze od kilku znanych projektów typu *open source*.

W ostatnim, dziewiątym, rozdziale pracy autor podsumował uzyskane rezultaty, podkreślił, że zweryfikowane zostały pozytywnie wszystkie wymagania postawione przed projektowanym systemem, wymienił rozwiązania stanowiące nowe koncepcje. Na zakończenie autor przedstawił i przedyskutował listę nowych cech, które mogłyby zostać uwzględnione przy ewentualnej rozbudowie systemu.

3. Ocena pracy

Autor wykazał się głęboką wiedzą w zakresie projektowania systemów monitorujących środowiska rozproszone, a w szczególności w zakresie metod i algorytmów zapewniania tym systemom własności wysokiej dostępności i możliwości samonaprawy. Wszechstronna analiza tych metod, algorytmów i technik pozwoliła autorowi wybrać takie spośród nich, które w połączeniu z oryginalną architekturą i złożoną funkcjonalnością całego projektu dały rozwiązanie o unikalnych cechach.

System AgeMon należy do nielicznych systemów monitorujących o możliwościach samonaprawy i wysokiej dostępności, co m.in. osiągnięto poprzez eliminację pojedynczych punktów podatności na awarię oraz redundancję szeregu komponentów.

Do najważniejszych rezultatów badawczych uzyskanych przez autora zaliczam:

- opracowanie modelu systemu monitorującego AgeMon, opartego na rolach grupujących niezbędne funkcjonalności, i agentach wykonujących role i współpracujących z sobą; opracowanie trójwarstwowego podsystemu komunikacji agentów,
- analizę i wybór najlepszych technik samonaprawy, które mogą być użyte w systemach monitorujących, włączenie tych technik do systemu AgeMon, a także dobór rozwiązań tworzących wysoką dostępność tego systemu,
- stworzenie prototypu systemu monitorującego i przeprowadzenie eksperymentalnej weryfikacji cech systemu AgeMon.

Na wysoką ocenę zasługują też inne rozwiązania zastosowane w ramach tego projektu, m.in. wskazanie i analiza możliwości wykorzystania mechanizmów systemu monitorującego AgeMon do naprawy aplikacji monitorowanych, a także zastosowanie przy implementacji systemu AgeMon takich technik budowy komponentów tego systemu, które zapewniają możliwość ich wielokrotnego użycia w innych systemach monitorujących.

Wysoko oceniam eksperymentalną część pracy, która pozwoliła potwierdzić spodziewane cechy systemu AgeMon i sprawdzić jego funkcjonalność, a także ocenić wartości liczbowe szeregu jego parametrów. Implementacja prototypu oraz sposób przeprowadzenia eksperymentów pozwalają niezwykle wysoko ocenić wiedzę i doświadczenie autora w zakresie zaawansowanych technologii programistycznych oraz projektowania złożonych systemów rozproszonych.

Warto w tym miejscu podkreślić, że rezultaty badań przedstawione w pracy były wcześniej publikowane w kilkunastu publikacjach, w tym w czterech pracach indeksowanych przez *Web of Science*. Prace te były też prezentowane na kilku konferencjach międzynarodowych. Wyniki te zostały więc zweryfikowane w międzynarodowym środowisku naukowym.

4. Uwagi do pracy

Lektura pracy nasunęła mi m.in. następujące uwagi i pytania:

1) W podrozdziale 2.4.1 został przedstawiony system PANACEA stanowiący ramę (ang. *framework*) ułatwiającą budowę aplikacji o możliwościach samonaprawy. Wśród prezentowanych w rozdziale 2 systemów jest to właściwie jedyny system udostępniający własności samonaprawy. Autor przyjął inną koncepcję budowy systemu AgeMon, przy czym warto byłoby bardziej wyraźnie podkreślić, przyjęcie których założeń uniemożliwiło autorowi wykorzystanie takiej ramy.

2) Czasy zapisu wyników monitorowania, przedstawione w rozdziale 8, wyznaczone w eksperymentach dotyczących wydajności, będą prawdopodobnie istotnie zależały od konkretnej wykorzystywanej bazy danych. Warto byłoby wyraźnie określić, za jaką część wyznaczonych czasów odpowiada zastosowana baza danych HSQL.

3) Str. 60, rys. 4.2: w niebieskich prostokątach zamiast "*Agent Monitoring Rule*" winien chyba być podpis „*Agent Monitoring Role*".

4) Str. 165, wiersz 8: jest: „...950 messages..." – winno chyba być: „...950 messages per second ...".

5) Str. 167, tabela 8.5: w pierwszej kolumnie winny chyba być następujące dane: 1K, 2.5K, 5K.

Przedstawione uwagi nie zmieniają mojej pozytywnej i wysokiej oceny rozprawy.

5. Wniosek końcowy

Podsumowując stwierdzam, że w recenzowanej rozprawie został sformułowany, a następnie poprawnie rozwiązany problem naukowy obejmujący wybór i wprowadzenie do projektu systemu monitorującego środowiska rozproszone takich metod i technik, które zapewniają temu systemowi możliwości samonaprawy i własności wysokiej dostępności.

Autor wykazał się głęboką wiedzą w zakresie metod budowy systemów monitorowania o wysokiej dostępności i możliwościach samonaprawy, a także umiejętnościami i doświadczeniem praktycznym, udokumentowanymi zaprojektowaniem i implementacją systemu monitorowania o powyższych cechach i udaną eksperymentalną weryfikacją opracowanej koncepcji.

Stwierdzam, że oceniana praca spełnia z nadmiarem wszystkie wymagania stawiane rozprawom doktorskim przez obowiązujące przepisy. Wnoszę o dopuszczenie Pana mgr Piotra Pęgla do dalszych etapów przewodu doktorskiego.

A. Jarielski