POLITECHNIKA POZNAŃSKA

Wydział Informatyki

Monitoring Zdarzeń Systemowych i Bezpieczeństwa w Środowisku Systemów Rozproszonych

Łukasz Kufel

Streszczenie rozprawy doktorskiej

Promotor

Prof. dr hab. inż. Jan Węglarz

Poznań, 2017

Wprowadzenie

W dzisiejszych czasach wydajność i zaufanie do przedsiębiorstw jest bardziej zależne od systemów IT niż w przeszłości. Wszystkie usługi elektroniczne takie jak wewnętrzny portal z wiadomościami, poczta elektroniczna, program kadrowy czy internetowy serwis komercyjny firmy [17], działają na „niewidzialnych" systemach IT. To właśnie one umożliwiają rozwój i funkcjonalność przedsiębiorstwa. Systemy te muszą być coraz bardziej bezpieczne ze względu na poufność i wzrastającą ilość przetwarzanych danych. Nieupoważniony dostęp do ważnych danych firmowych może znacząco wpłynąć na wiarygodność i opinię rynkową danego przedsiębiorstwa.

Aby zapewnić wysoką dostępność i bezpieczeństwo tych systemów, przedsiębiorstwo musi znaleźć i wdrożyć wiele systemów [6], wśród nich między innymi system do monitoringu [20]. System do monitoringu będzie nieustannie analizował i weryfikował dostępność usług biznesowych, ich wydajność oraz zgodność ze standardami zabezpieczeń danych. System ten będzie dodatkowo informował odpowiednie zespoły wsparcia w przypadku wykrycia awarii lub nieautoryzowanej próby dostępu do poufnych danych. Zaprojektowanie odpowiedniego systemu do monitoringu rozproszonych systemów informatycznych przedsiębiorstwa może być czasochłonne i trudne, ze względu na wielkość infrastruktury IT jak i skomplikowane usługi biznesowe dostępne w firmie. Ostatecznie, system monitoringu musi również posiadać wiele protokołów integracyjnych (API), aby bezproblemowo zintegrować się z obecnym w firmie środowiskiem IT [31].

Aktualnie dostępne na rynku systemy do monitoringu dzieli się na dwie kategorie: narzędzia do monitoringu zdarzeń bezpieczeństwa (security information and event management - SIEM) oraz narzędzia do monitoringu infrastruktury IT [12, 13]. Narzędzia te gromadzą oraz przetwarzają zdarzenia systemowe i bezpieczeństwa korzystając z dostępnych czterech sposobów kolekcjonowania: monitoring agentowy, monitoring bezagentowy, monitoring hybrydowy oraz ostatnio zaprezentowany

monitoring przez strumienie danych. Monitoring systemów w środowiskach rozproszonych składa się z czterech etapów, tj. kolekcji zdarzeń [15], segregacji lub filtrowania [27, 30], analizy [24, 28, 29], oraz wizualnej prezentacji zgromadzonych zdarzeń i danych [19]. Zanim jednak system zacznie działać i zbierać zdarzenia, musi zostać wybrane i wdrożone narzędzie do monitoringu oraz systemy objęte monitoringiem muszą zostać dodane i skonfigurowane w narzędziu do monitoringu.

Badania przedstawione w rozprawie doktorskiej ograniczają się do pierwszego etapu – do kolekcji zdarzeń – i obejmują procesy wyboru, projektowania i wdrażania systemu do monitoringu w różnych środowiskach systemów rozproszonych. W rozprawie zaprezentowano również autorski sposób monitoringu na zamówienie oraz wpływ opóźnienia sieciowego (ang. network latency) [3, 25] na całościowy proces monitoringu. Aby zrealizować powyższy zakres prac, zdefiniowano następujące cele badawcze:

- Przegląd aktualnie dostępnych narzędzi do monitoringu oraz porównanie sposobów kolekcji danych,
- Wskazanie kluczowych kryteriów przy wyborze narzędzia do monitoringu systemów rozproszonych,
- Wdrożenie i przetestowanie kilku różnych narzędzi do monitoringu w środowiskach systemów rozproszonych znajdujących się w jednym centrum danych, w wielu centrach i w chmurze,
- Zaprojektowanie, wdrożenie i zweryfikowanie monitoringu na zamówienie, bazującego na hybrydowym sposobie gromadzenia zdarzeń,
- Przeprowadzenie eksperymentu i analizy w obszarze opóźnienia sieciowego podczas monitoringu systemów znajdujących się w wielu lokalizacjach geograficznych,
- Zaproponowanie idei rozwiązania, które zredukuje wpływ opóźnienia sieciowego na monitorowanie systemów znajdujących się w odległych lokalizacjach.

Podstawy monitoringu

W zarządzaniu systemami działającymi w środowiskach rozproszonych wyróżnia się narzędzia do monitoringu i diagnostyki. Narzędzia diagnostyczne są używane tylko w przypadku wnikliwej analizy, zazwyczaj na żądanie, na przykład w momencie wystąpienia awarii systemowej lub w sytuacji, gdy wymagane są pojedyncze pakiety sieciowe. Narzędzia do monitoringu są natomiast w ciągłym użyciu, tj. gromadzą dane i zdarzenia [26] w regularnych odstępach czasowych. Rozdział ten przedstawia założenia koncepcyjne projektowania systemów do monitoringu, definiuje zdarzenia systemowe i bezpieczeństwa oraz obszary, gdzie monitoring może zostać zastosowany. Pod koniec rozdziału omówiona została klasyfikacja systemów ze względu na ich krytyczność i potrzeby biznesowe. Zaprezentowano także przykładowe wartości czasowe, w jakich system monitoringu może kolekcjonować zdarzenia i jak długo je przechowywać.

Podczas projektowania i wdrażania systemu do monitoringu należy uwzględnić trzy główne warstwy, takie jak a) kolekcja, prezentacja i powiadomienia, b) sieć, c) środowisko systemów rozproszonych. Każda z tych warstw odgrywa ważną rolę w procesie monitoringu i ma znaczący wpływ na funkcjonowanie całego systemu monitorującego, w szczególności, gdy monitorowane systemy znajdują się w odległych lokalizacjach. Jednym z powodów, dla których wdrażane są systemy monitoringu jest kolekcja zdarzeń systemowych i bezpieczeństwa. W rozprawie zaprezentowano przykładowe zdarzenia każdego z wymienionych typów, jak również opisano możliwe poziomy ich krytyczności (platformy Windows: error, warning, information, failure audit, success audit oraz platformy Unix: emergency, alert, critical, error, warning, notice, informational i debug). W zależności od poziomu krytyczności danego zdarzenia, w systemie monitorującym możliwe jest skonfigurowanie powiadomienia, np. w postaci wiadomości email. Powiadomienie to może mieć na celu zaangażowanie zespołów wsparcia technicznego, aby usunąć wykrytą nieprawidłowość w działaniu

systemów rozproszonych. Monitoring systemów rozproszonych znajduje swoje zastosowanie nie tylko w gromadzeniu zdarzeń systemowych i bezpieczeństwa. Stanowi on również podstawę w dostarczaniu metryk związanych z dostępnością (Tabela 1) i wydajnością monitorowanych systemów biznesowych. Metryki te mogą wspomagać procesy biznesowe związane z planowaniem budżetu na kolejne lata, na rozwój i utrzymanie infrastruktury IT.

Tabela 1. Czas niedostępności / przestoju w działaniu i jego wpływ na raporty dostępności.

Niedostępność w miesiącu	Niedostępność w roku	Współczynnik dostępności
72 godziny	36.5 dni	90% "jedna dziewiątka "
7.20 godzin	3.65 dni	99% "dwie dziewiątki"
43.8 minuty	8.76 godzin	99.9% "trzy dziewiątki"
4.38 minuty	52.56 minuty	99.99% "cztery dziewiątki"
25.9 sekund	5.26 minut	99.999% "pięć dziewiątek"

Wśród systemów rozproszonych można wyróżnić trzy grupy serwerów ze względu na ich zastosowanie biznesowe: serwery krytyczne (ang. mission critical), ważne (ang. critical) oraz standardowe (ang. standard). Kategoryzacja serwerów, ze względu na zastosowanie biznesowe, umożliwia lepszą ich administrację oraz przystosowanie systemu monitorującego, np. serwery krytyczne mogę być monitorowane co jedną minutę zamiast pięciu jak w przypadku standardowych serwerów [10]. W przypadku wystąpienia awarii lub ważnego zdarzenia systemowego, powiadomienie zespołów wsparcia technicznego może odbyć się przez automatyczne połączenie telefoniczne lub wiadomość SMS, zamiast zwykłego powiadomienia przez widomość email.

Sposoby kolekcjonowania zdarzeń

W monitoringu systemów rozproszonych występują dwa popularne sposoby kolekcjonowania zdarzeń: monitoring agentowy i monitoring bezagentowy. Ostatnio dołączyły do nich dwa nowe, tj. monitoring hybrydowy oraz monitoring przez strumienie danych. Monitoring agentowy (ang. agent-based) [7] charakteryzuje się dedykowanym oprogramowaniem, które musi zostać zainstalowane na każdym monitorowanym zasobie lub systemie. Dodatkowe oprogramowanie, inaczej zwane oprogramowaniem agenta, ma za zadanie dokładne kolekcjonowanie aktualnych wartości poszczególnych metryk oraz przesłanie ich do centralnego systemu monitorującego. Choć rozwiązanie to dostarcza precyzyjnych danych, jest trudne w utrzymaniu a jego wdrożenie jest czasochłonne.

Alternatywą jest monitoring bezagentowy (ang. agentless), gdzie nie jest wymagana instalacja specjalistycznego oprogramowania, a wykorzystywane są systemowe, wbudowane protokoły i technologie kolekcjonujące kluczowe metryki i zdarzenia. Zalicza się do nich między innymi protokół SNMP (Simple Network Management Protocol) [23] oraz technologia WMI (Windows Management Instrumentation). Ten sposób kolekcjonowania danych ograniczony jest jednak do podstawowych metryk i nie jest zalecany w monitoringu krytycznych systemów biznesowych.

Monitoring hybrydowy to połączenie zalet dostępnych w monitoringu agentowym z łatwością wdrażania reprezentowaną przez sposób bezagentowych. Poszczególny sposób kolekcjonowania zdarzeń wybierany jest w zależności od potrzeb biznesowych i ustanowionej przez przedsiębiorstwo polityki monitoringu. Dla przykładu, monitoring agentowy jest wdrażany na krytycznych systemach biznesowych, zaś pozostałe systemy są monitorowane przez monitoring bezagentowy. W monitoringu hybrydowym możliwe jest wykorzystanie wbudowanych, systemowych harmonogramów zadań [18], które o

określonym czasie uruchomią skrypt lub program diagnostyczny w celu zgromadzenia aktualnych wartości monitorowanych metryk. Wynik działania harmonogramu zadań jest następnie analizowany przez centralny system monitorujący.

Najnowszym sposobem monitorowania systemów rozproszonych jest monitoring przez strumienie danych (ang. data streams). Polega on na zintegrowaniu agenta monitorującego z kodem źródłowym aplikacji lub transakcji biznesowej. Taka kooperacja pozwala na natychmiastowe przekazywanie ważnych danych z aplikacji do centralnego systemu monitorującego, jak na przykład czas realizacji zamówienia, ilość transakcji na minutę lub poziom błędów. W monitoringu przez strumienie danych oprogramowanie agenta ma za zadanie wyłącznie przekazywać zgromadzone dane z systemu, na którym jest on zainstalowany do centralnej konsoli. Proces ten może odbywać się przez protokół HTTPS lub przez magistrale przekazywania komunikatów, np. RabbitMQ.

Narzędzia do monitoringu

Na rynku dostępnych jest wiele programów i narzędzi służących do monitoringu systemów w środowiskach rozproszonych [12, 13]. Programy te dostępne są odpłatnie lub niektóre w ramach licencji open source i dzielą się na dwie główne kategorie: rozwiązania do zarządzania zdarzeniami bezpieczeństwa (SIEM - security information and event management) [9] oraz rozwiązania do monitoringu dostępności, zasobności i wydajności infrastruktury IT. Tabela 2 przedstawia narzędzia do monitoringu i zarządzania zdarzeniami bezpieczeństwa, natomiast Tabela 3 reprezentuje narzędzia do monitoringu infrastruktury IT. W przeglądzie narzędzi monitorujących wzięto pod uwagę następujące kryteria:

- Licencja - sprawdzenie dostępności wersji open source lub bezpłatnej.
- Sposób kolekcji - w jaki sposób gromadzone są zdarzenia systemowe i bezpieczeństwa. Dostępność kilku sposobów umożliwia efektywniejszą integrację z innymi systemami dostępnymi w przedsiębiorstwie.
- Obsługa systemów działających w chmurze - aktualnie wiele przedsiębiorstw wdraża lub planuje w niedalekiej przyszłości rozszerzenie infrastruktury IT o systemy działające w chmurze [1, 5].
- Powiadomienia - w przypadku narzędzi do monitoringu infrastruktury IT dodatkowo sprawdzone zostały sposoby powiadomień zespołów wsparcia.
- Wielkość wspieranego przedsiębiorstwa, firmy - recenzowane narzędzia dostępne były w dwóch kategoriach, dla małych i średnich firm (posiadające mniej niż 500 systemów, z których zbierane będą zdarzenia) oraz dla średnich, dużych i korporacyjnych środowisk (więcej niż 500 systemów).
- Cechy unikalne - unikalne własności każdego z przeglądanych narzędzi zostały umieszczone w tej kolumnie.

Tabela 2. Narzędzia do zarządzania zdarzeniami bezpieczeństwa (SIEM).

Narzędzie	Licencja	Sposób kolekcji	Obsługa chmury	Wielkość firmy	Cechy unikalne
AlienVault USM	Open source, komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Mała i średnia	Dostępność wersji open source, forum użytkowników związanych z bezpieczeństwem
BlackStratus	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Mała i średnia	Dostępność w modelu Software as a Service (SaaS)
EMC (RSA)	Komercyjna, bezpłatna	Agentowy	Tak	Średnia, duża, korporacja	Dostępność bezpłatnej wersji z możliwością przechwytywania pakietów sieciowych
EventTracker	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Mała i średnia	Dostępność w modelu Software as a Service (SaaS)
Fortinet (AccelOps)	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Mała i średnia	Monitoring dostępności i wydajności aplikacji, dostępność w modelu SaaS
HP ArcSight	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Możliwość integracji z wieloma dodatkami, dostępność w modelu SaaS
IBM QRadar SIP	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Dostępność w modelu SaaS oraz Infrastructure as a Service (IaaS)
Intel Security ESM	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Możliwość zintegrowania z innymi technologiami Intel Security
LogRhythm	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Wyszukiwanie kontekstowe i niestrukturalne
ManageEngine Log360	Komercyjna	Bezagentowy	Tak	Mała i średnia	Wyłącznie monitoring bezagentowy, dokładny audyt usług Active Directory

Micro Focus (NetIQ)	Komercyjna, bezpłatna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Obsługa systemów typu mainframe, bezpłatna wersja narzędzia Sentinel Log Manager
SolarWinds LEM	Komercyjna	Agentowy	Nie	Mała i średnia	Łatwa instalacja, współpraca z innymi produktami SolarWinds
Splunk SIP	Komercyjna	Agentowy	Tak	Średnia, duża, korporacja	Wydajna wyszukiwarka, dostępność w modelu SaaS, wiele opcji wdrożenia systemu
Trustwave	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Tak	Średnia, duża, korporacja	Wiele opcji wdrożenia systemu włącznie z jego obsługą przez producenta

Tabela 3. Narzędzia do monitoringu infrastruktury IT w środowiskach systemów rozproszonych.

Narzędzie	Licencja	Sposób kolekcji	Powia-domienia	Obsługa chmury	Wielkość firmy	Cechy unikalne
AppDynamics	Komercyjna	Strumienie danych	Email, SMS, API	Tak	Średnia, duża, korporacja	Dostępność w modelu Software as a Service (SaaS)
Datadog	Komercyjna	Strumienie danych	Email, SMS, API	Tak	Mała, średnia i duża	Obsługa wielu dostawców platfo-rm w chmurze i zespołów DevOps
Ganglia	Open source	Agentowy	Opcja przez Nagios	Tak	Średnia, duża, korporacja	Obsługa klastrów i systemów typu grid
Graphite	Open source	Strumienie danych	Brak	Tak	Średnia, duża, korporacja	Obsługa do 160000 metryk na minutę
HP Operations Manager	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Email, SMS, własny	Nie	Korporacja	Integracja z pozostałymi produktami HP

Hyperic	Open source, komercyjna	Hybrydowy (agentowy i bezagentowy)	Email, SMS	Tak	Mała i średnia	Łatwa instalacja i konfiguracja
IBM SmartCloud Monitoring	Komercyjna	Hybrydowy (agentowy i bezagentowy)	Email, SMS	Tak	Korporacja	Analizy przewidywań i raporty
ManageEngine AppManager	Komercyjna	Bezagentowy	Email, SMS, własny	Tak	Mała i średnia	Szybka i łatwa instalacja, monitoring wydajności aplikacji
Nagios	Open source, komercyjna	Hybrydowy (agentowy i bezagentowy)	Email, SMS, własny	Tak	Mała, średnia i duża	Wiele dodatków i forum wsparcia użytkowników
New Relic	Komercyjna	Strumienie danych	Email, SMS, API	Tak	Mała, średnia i duża	Dostępność w modelu Software as a Service (SaaS)
Prometheus	Open source	Strumienie danych	Email, SMS, API	Tak	Mała, średnia i duża	Aktywne forum użytkowników i twórców programu
Riemann	Open source	Strumienie danych	Email, SMS, API	Tak	Mała i średnia	Wsparcie przez forum użytkowników
Sensu	Open source, komercyjna	Agentowy	Email, SMS, API	Tak	Mała, średnia i duża	Automatyzacja plików konfiguracyjnych przez Chef i Puppet
TICK by InfluxData	Open source, komercyjna	Strumienie danych	Email, SMS, API	Tak	Mała, średnia i duża	Dostępność w modelu Software as a Service (SaaS)
Zabbix	Open source	Hybrydowy (agentowy i bezagentowy)	Email, SMS, własny	Tak	Średnia, duża, korporacja	Funkcja auto-detekcji zasobów, wiele dodatków

Wdrożenie systemu do monitoringu

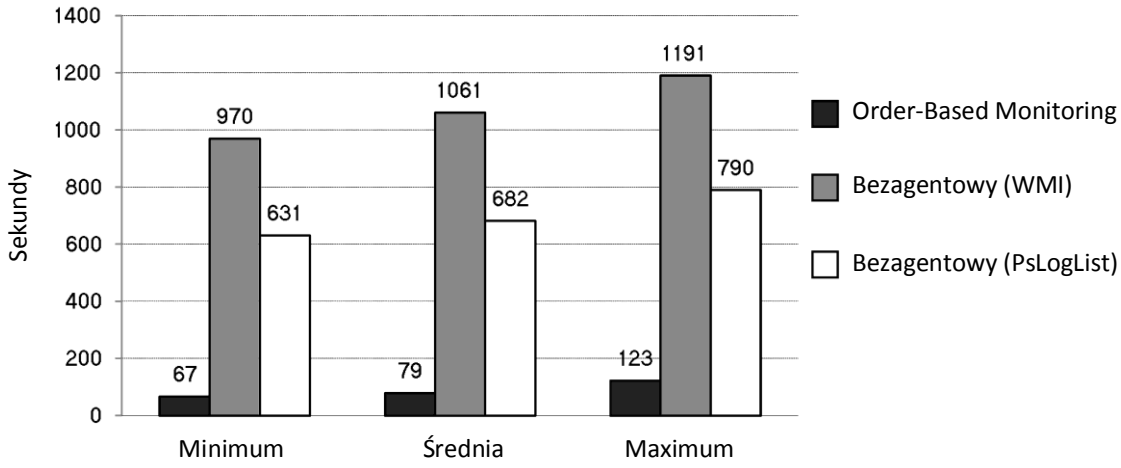
Proces wdrożenia systemu do monitoringu rozpoczyna się od planowania i wyboru narzędzia. Jak już wspomniano, na ryku dostępne jest wiele narzędzi i sposobów kolekcjonowania zdarzeń. Podczas wyboru narzędzia, osoba decyzyjna powinna zwrócić uwagę na następujące kryteria: funkcjonalność i skalowalność rozwiązania, dostępne protokoły powiadomień i integracji z rozwiązaniami obecnie dostępnymi w przedsiębiorstwie, łatwość wdrożenia i utrzymania całego rozwiązania oraz cena. Warto również zwrócić uwagę na dostępność wersji testowych, zazwyczaj ograniczone są czasem działania, aby przeprowadzić wdrożenie wstępne (ang. proof of concept).

W rozdziale tym przedstawiono przykładowe wdrożenie systemu monitorującego 130 serwerów znajdujących się w jednym centrum danych. Przeanalizowano wpływ opóźnienia sieciowego na monitoring 10 i 100 systemów zlokalizowanych w wielu centrach oraz przeprowadzono przykładowe wdrożenia systemów monitorujących serwery w chmurze. W każdym z przykładowych wdrożeń skupiono się na monitoringu zdarzeń systemowych i bezpieczeństwa, a także na weryfikacji autorskiego monitoringu hybrydowego bazującego na harmonogramie zadań i idei zaplanowanego zamówienia – order-based monitoring (OBM).

OBM działa podobnie do rozwiązania agentowego, gdyż musi być skonfigurowany na każdym z monitorowanych zasobów, jednak nie potrzebuje dedykowanej usługi systemowej. Wywołanie procesu monitorującego, który gromadzi dane o zgłoszonych metrykach, następuje poprzez systemowy harmonogram zadań - task scheduler w systemach Windows i cron w systemach Unix. Tym samym, podobnie jak w monitoringu bezagentowym, wykorzystywane są wbudowane technologie. OBM umożliwia szczegółowy dostęp do wszystkich metryk monitorowanego systemu, nie zajmuje zbyt wielu zasobów systemowych oraz nie jest zależny od języka programowania skryptowego lub od platformy systemowej, na której jest uruchamiany.

OBM gromadzi dane o monitorowanych metrykach poprzez wywołanie skryptów systemowych lub/oraz uruchomienie ogólnie dostępnych, bezpłatnych narzędzi diagnostycznych.

Wdrożenie monitoringu, dla przykładowego zbioru 130 serwerów, z kolekcją zdarzeń bazującą na OBM było ośmiokrotnie szybsze niż rozwiązanie bazujące na kolekcji bezagentowej z narzędziem PsLogList lub ponad 13-to krotnie efektywniejsze niż kolekcja bezagentowa z wykorzystaniem technologii WMI. Porównanie zaprezentowane na Rysunku 1 to całkowity czas kolekcji zdarzeń w sekwencyjnym monitoringu serwerów i bazuje na uśrednionych danych zgromadzonych przez trzy miesiące.

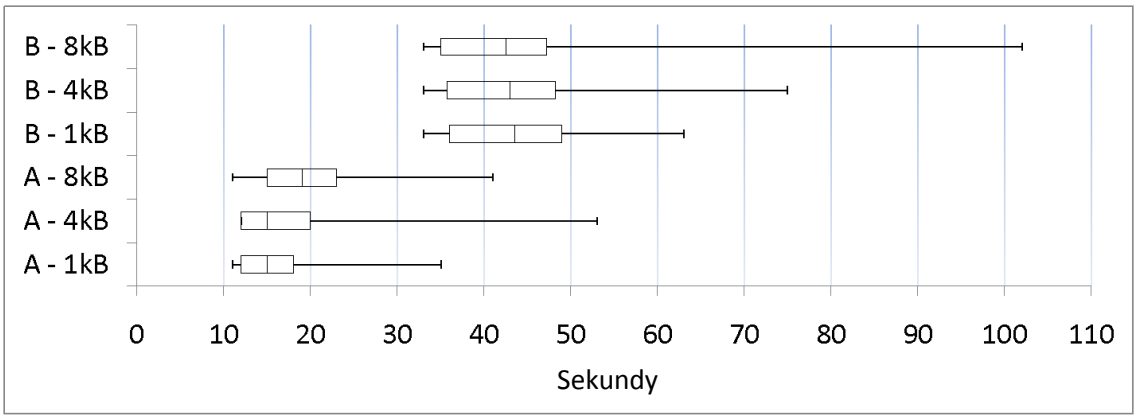


Rysunek 1. Porównanie całkowitego czasu kolekcji zdarzeń w sekwencyjnym monitoringu serwerów poprzez różne sposoby kolekcji.

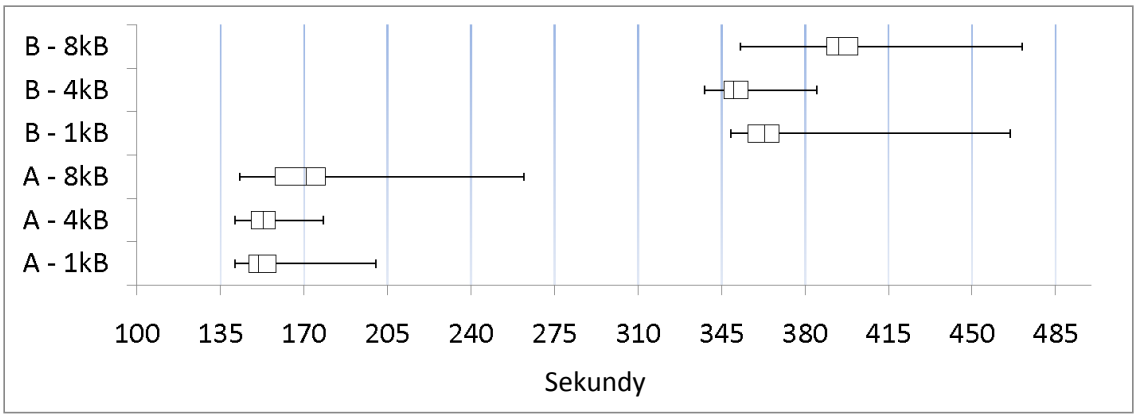
Następne wdrożenie z wykorzystaniem monitoringu hybrydowego z OBM przeprowadzone zostało dla trzech centrów danych, jednym znajdującym się w Irlandii oraz dwoma w USA (jednym w stanie Waszyngton i drugim w stanie Arizona) [11]. W centrum danych w Arizonie znajdowało się przykładowe 100 serwerów, natomiast narzędzia monitorujące były zainstalowane w pozostałych dwóch lokalizacjach. Eksperyment został przeprowadzony z wykorzystaniem trzech plików ze zdarzeniami systemowymi i bezpieczeństwa o różnych rozmiarach: 1 kB, 4 kB oraz 8 kB. Głównym celem badawczym było przeanalizowanie opóźnienia sieciowego na czas potrzebny do sekwencyjnego przesłania plików ze zdarzeniami z serwerów w Arizonie na serwery z narzędziami monitorującymi. Wyniki zawarte na Rysunku 2 wyraźnie wskazują, że

odległość pomiędzy serwerem z narzędziem monitorującym a monitorowanymi systemami ma znaczący wpływ na całkowity czas kolekcji zdarzeń, natomiast rozmiar pliku od 1 kB do 8 kB nie ma większego znaczenia.

2A) Wyniki dla 10 serwerów



2B) Wyniki dla 100 serwerów

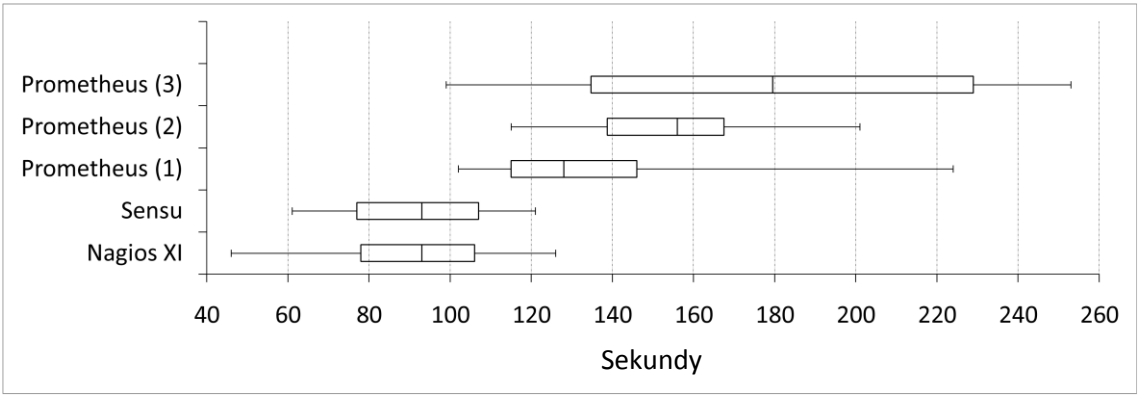


Rysunek 2. Porównanie przesyłania plików ze zdarzeniami w monitoringu OBM dla wielu lokalizacji. Rysunek 2A) przedstawia minimalny czas, medianę oraz maksymalny czas dla przesyłania plików pomiędzy lokalizacjami w USA (np. A - 1kB) oraz pomiędzy Irlandią a USA stan Arizona (np. B - 8kB) dla 10 serwerów. Rysunek 2B) przedstawia wyniki dla 100 serwerów.

Opóźnienie sieciowe nie może być pominięte lub zredukowane, jednak możliwe jest zredukowanie ilości połączeń między systemem monitorującym a monitorowanymi serwerami przez zaproponowany w rozprawie koncept lokalnego dystrybutora. Dystrybutor jest pośrednikiem i zawsze znajduje się w tej samej lokalizacji, co monitorowane serwery i transakcje biznesowe [21]. Jego rolą jest gromadzeniu plików ze zdarzeniami a następnie przesyłanie całego zbioru w pojedynczym połączeniu z

serwerem, gdzie znajduje się system monitorujący. Pomimo wprowadzenia dodatkowego elementu w całym procesie monitorującym, korzyści są znaczące i zauważalnie usprawniają monitoring zdalnych zasobów. Dodatkowo, dystrybutor może pełnić rolę weryfikującą i podejmować autonomiczne decyzje naprawcze, tj. zrestartowanie usługi lub przełączenie pakietów sieciowych na system zapasowy.

Kolejne wdrożenia przeprowadzone zostało z systemami działającymi w środowiskach chmur oferowanych przez Amazon AWS oraz Google Cloud [13]. W eksperymencie wykorzystano trzy popularne narzędzia monitorujące: Nagios XI, Prometheus oraz Sensu. Narzędzia te zainstalowane zostały w tym samym centrum danych, każde na osobnym serwerze z systemem operacyjnym CentOS. Po skonfigurowaniu narzędzi monitorujących oraz serwerów działających w chmurze, przystąpiono do właściwej części eksperymentu – do zmierzenia czasu, w którym każde z narzędzi poinformuje zespoły wsparcia o wykryciu awarii na monitorowanych systemach. Jako przykładową awarię wybrano obciążenie procesora większe niż 95%. Awaria symulowana była przy pomocy pakietu cpuburn. Wyniki zaprezentowane na Rysunku 3 bazują na ponad 400 próbach badawczych i wskazują, że sprawdzając obciążenie procesora co jedną minutę narzędziami Nagios XI oraz Sensu, zespoły wsparcia zostaną poinformowane w przeciągu 93 sekund, w 50% przypadków. Czas ten był dokładnie mierzony od momentu, gdy obciążenie przekroczyło próg 95%, poprzez powtórne wykrycie przez narzędzie monitorujące (wartość ponad 95% musiała wystąpić w dwóch kolejnych odczytach) do dostarczenia powiadomienia na konto email.



Rysunek 3. Wyniki eksperymentu z wdrożeniem narzędzi monitorujących systemy w chmurze. Czas mierzono od momentu awarii do dostarczenia powiadomienia na konto email. Prometheus był testowany w trzech wariantach: (1) parametr group wait 10s oraz group interval 2m, (2) group wait 30s oraz group interval 5m, (3) grupowanie wyłączone.

W przypadku narzędzia Prometheus, czas powiadomienia był znacznie dłuższy, ze względu na mechanizm obliczania obciążenia procesora bazujący na osi czasu (ang. time series) i w 50% przypadków był następujący:

- 128 sekund w pierwszym ustawieniu grupowania,
- 156 sekund w drugim ustawieniu grupowania,
- 179.5 sekundy, gdy grupowanie zdarzeń było wyłączone.

Grupowanie zdarzeń ma na celu zredukowanie ilości powiadomień w przypadku wystąpienia krytycznej awarii mającej wpływ na wiele systemów równocześnie, np. awaria kluczowego węzła sieciowego lub usługi DNS [2, 14].

Podsumowując, monitoring systemów działających w chmurze jest zbliżony do monitoringu systemów w jednym lub wielu centrach danych. Najważniejszym punktem wyróżniającym monitoring systemów w chmurze jest zapewnienie bezpieczeństwa pakietom sieciowym przesyłanym pomiędzy narzędziem monitorującym a monitorowanymi systemami [4].

Podsumowanie

W rozprawie doktorskiej przedstawiony został problem wyboru, zaprojektowania oraz wdrożenia rozwiązania do monitoringu zdarzeń systemowych i bezpieczeństwa w środowiskach systemów rozproszonych. Badania naukowe objęły systemy Windows i Unix działające w jednym centrum danych, w wielu rozproszonych centrach oraz w chmurze. Po omówieniu podstaw monitoringu i sposobów kolekcjonowania zdarzeń, zaprezentowana została wnikliwa analiza aktualnie dostępnych narzędzi do monitoringu. W dalszej części pracy przedstawiono kluczowe kryteria przy wyborze narzędzia do monitoringu oraz omówiono wdrożenie autorskiego monitoringu hybrydowego bazującego na harmonogramach zadań i idei zaplanowanego zamówienia – order-based monitoring (OBM). Zaprezentowano wpływ opóźnienia sieciowego na całkowity proces monitoringu oraz wprowadzono koncept lokalnego dystrybutora w celu jego ograniczenia. W eksperymencie przeprowadzonym na systemach działających w chmurze pokazano, iż w przypadku zaistnienia awarii, monitoring z wykorzystaniem strumieniowego sposobu kolekcji zdarzeń będzie potrzebował więcej czasu na powiadomienie zespołów wsparcia niż inne sposoby. Tym samym osiągnięto cele badawcze postawione na początku rozprawy.

W trakcie prac nad doktoratem systemy rozproszone ewoluowały i dziś są coraz częściej wdrażane w środowiskach chmur [8, 16, 22]. Nowy model dostarcza wiele udogodnień w zarządzaniu infrastrukturą IT, chociażby mechanizm szybkiego tworzenia lub usuwania systemów (ang. auto scaling). Z drugiej jednak strony stanowi wyzwanie dla tradycyjnych rozwiązań monitorujących, gdzie czas działania systemów liczony jest w miesiącach lub latach. Tematyka ta jest interesującym obszarem dalszych prac badawczych nad monitoringiem zdarzeń.

Bibliografia

[1] Aceto G., Botta A., De Donato W., Pescape A., Cloud monitoring: A survey, Computer Networks, vol. 57, pp. 2093-2115, 2013.

[2] Casalicchio E., Caselli M., Coletta A., Measuring the global Domain Name System, IEEE Network, vol. 27, no. 1, pp. 25-31, 2013.

[3] De Vito L., Rapuano S., Tomaciello L., One-way delay measurement: State of the art, IEEE Trans. Instrumentation and Measurement, vol. 57, no. 12, pp. 2742-2750, 2008.

[4] Dev Mishra A., Beer Singh Y., Big data analytics for security and privacy challenges, International Conference on Computing, Communication and Automation (ICCCA), India, pp. 50-53, 2016.

[5] Fatema K., Emeakaroha V. C., Healy P. D., Morrison J. P., Lynn T., A survey of cloud monitoring tools: Taxonomy, capabilities and objectives, Journal of Parallel and Distributed Computing, vol. 74, no. 10, pp. 2918-2933, 2014.

[6] Ishibashi K., Maintaining quality of service based on ITIL-based IT service management, Fujitsu Scientific & Technical Journal, vol. 43, no. 3, pp. 334-344, 2007.

[7] Jennings N. R., On agent-based software engineering, Artificial Intelligence, vol. 117, no. 2, pp. 277-296, 2000.

[8] Katsaros G., Kousiouris G., V. Gogouvitis S., Kyriazis D., Menychtas A., Varvarigou T., A self-adaptive hierarchical monitoring mechanism for clouds, Journal of Systems and Software, vol. 85, no. 5, pp. 1029-1041, 2012.

[9] Kavanagh K. M., Rochford O., Bussa T., Magic quadrant for security information and event management, Gartner, August 2016.

[10] Kent K., Souppaya M., Guide to computer security log management, US Nat'l Inst. Standards and Technology, http://dx.doi.org/10.6028/NIST.SP.800-92, 2006.

[11] Kufel L., Network latency in systems event monitoring for multiple locations, Scientific Programming, vol. 2015, article ID 371620, 2015.

[12] Kufel L., Security event monitoring in a distributed systems environment, IEEE Security & Privacy, vol. 11, no. 1, pp. 36-43, 2013.

[13] Kufel L., Tools for distributed systems monitoring, Foundations of Computing and Decision Sciences, vol. 41, no. 4, pp. 237-260, 2016.

[14] Lee S., Levanti K., Kim H., Network monitoring: Present and future, Computer Networks, vol. 65, pp. 84-98, 2014.

[15] Massie M., Li B., Nicholes B., Vuksan V., Monitoring with Ganglia, O'Reilly Media, 2013.

[16] Montesa J., Sánchez A., Memishi B., S. Pérez M., Antoniu G., GMonE: A complete approach to cloud monitoring, Future Generation Computer Systems, vol. 29, no. 8, pp. 2026-2040, 2013.

[17] Poggi N., Carrera D., Gavaldà R., Ayguadé E., Torres J., A methodology for the evaluation of high response time on e-commerce users and sales, Information Systems Frontiers, vol. 16, no. 5, pp. 867-885, 2014.

[18] Qin Z., Rojas-Cessa R., Ansari N., Task-execution scheduling schemes for network measurement and monitoring, Computer Communications, vol. 33, no. 2, pp. 124-135, 2010.

[19] Rastogi R., S A., G S., G P., D P., Singh A., Design and development of generic web based framework for log analysis, IEEE Region 10 Conference (TENCON), Singapore, pp. 232-236, 2016.

[20] Robinson W.N., A requirements monitoring framework for enterprise systems, Requirements Eng., vol. 11, no. 1, pp. 17-41, 2006.

[21] Sedlar U., Volk M., Sterle J., Kos A., Sernec R., Contextualized monitoring and root cause discovery in IPTV systems using data visualization, IEEE Network, vol. 26, no. 6, pp. 40-46, 2012.

[22] Smit M., Simmons B., Litoiu M., Distributed, application-level monitoring for heterogeneous clouds using stream processing, Future Generation Computer Systems, vol. 29, pp. 2103-2114, 2013.

[23] Subramanyan R., Miguel-Alonso J., Fortes J., Design and evaluation of a SNMP-based monitoring system for heterogeneous, distributed computing, tech. report TRECE 00-11, School of Electrical and Computer Eng., Purdue Univ., 2000.

[24] Suh-Lee C., Jo J.-Y., Kim Y., Text mining for security threat detection discovering hidden information in unstructured log messages, IEEE Conference on Communications and Network Security (CNS), USA, pp. 252-260, 2016.

[25] Svoboda P., Laner M., Fabini J., Rupp M., Ricciato F., Packet delay measurements in reactive IP networks, IEEE Instrumentation & Measurement Magazine, vol. 15, no. 6, pp. 36-44, 2012.

[26] Terenziani P., Coping with events in temporal relational databases, IEEE Trans. Knowledge and Data Eng., vol. 25, no. 5, pp. 1181-1185, 2013.

[27] Vaarandi R., A data clustering algorithm for mining patterns from event logs, Proceedings of the 3rd IEEE Workshop on IP Operations & Management, pp. 119-126, 2003.

[28] Vaarandi R., Mining event logs with SLCT and LogHound, Proceedings of the IEEE/IFIP Network Operations and Management Symposium, pp. 1071-1074, 2008.

[29] Vaarandi R., Pihelgas M., LogCluster - A data clustering and pattern mining algorithm for event logs, 11th International Conference on Network and Service Management (CNSM), IEEE Conference Publications, pp. 1-7, 2015.

[30] Vaarandi R., Platform independent event correlation tool for network management, Network Operations and Management Symposium, IEEE Conference Publications, pp. 907-909, 2002.

[31] Yonatany M., Platforms, ecosystems, and the internationalization of highly digitized organizations, Journal of Organization Design, 6: 2, doi:10.1186/s41469-017-0012-3, 2017.