STUDIA INFORMATICA 2015
Volume 36 Number 1 (119)

Agnieszka NOWAK-BRZEZIŃSKA, Artur TUROS
Uniwersytet Śląski, Instytut Informatyki

BADANIE WYDAJNOŚCI SERWERÓW BAZODANOWYCH Z WYKORZYSTANIEM METOD EKSPLORACJI ODCHYLEŃ

Streszczenie. Artykuł przedstawia problematykę badania wydajności serwerów bazodanowych z zastosowaniem SAP Adaptive Server Enterprise (ASE). Autorzy proponują rozwiązanie monitorujące wydajność serwera ASE opierając się na wykrywaniu anomalii z mechanizmu Query Processing Metrics.

Słowa kluczowe: odchylenia w danych, dane nietypowe, wydajność baz danych, anomalie w danych, Adaptive Server Enterprise, Sybase

PERFORMANCE TESTING OF DATABASE SERVERS USING OUTLIER MINING METHODS

Summary. An article presents the issues of performance testing of database servers based on SAP Adaptive Server Enterprise (ASE). The authors propose a solution for performance testing of SAP ASE based on outlier detection from Query Processing Metrics mechanism.

Keywords: outliers in data, unusual data, database performance, anomalies in data, Adaptive Server Enterprise, Sybase

1. Wprowadzenie

Ogromne ilości danych, które przechowywane są w dzisiejszych czasach, w ciągle rozbudowujących się *centrach danych* stwarzają wiele problemów wydajnościowych współczesnych aplikacji. Połączenie funkcji OLTP z funkcją przetwarzania wsadowego wydaje się jeszcze bardziej komplikować tę sprawę. Przestoje w działaniach aplikacji, jej niedostępność oraz spowolnienia mogą być bardzo kosztowne dla przedsiębiorstwa. Dlatego obecnie gro-

madzi się wiele danych, opisujących dane oraz mierzących wykorzystanie zasobów systemu oraz jego stanu (są to tzw. metadane).

Samo gromadzenie i przechowywanie metadanych nie wydaje się mieć jednak większego sensu. Podczas awarii systemów ciężar analizy i interpretacji zgromadzonych informacji spoczywa i tak na pracownikach działów IT. Dlatego ważnym elementem wydaje się być opracowanie narzędzi służących automatyzacji czynności analitycznych.

Wśród metod i technik mogących służyć temu zadaniu ciekawym wyborem będą metody wykrywania odchyleń w danych. Już od jakiegoś czasu algorytmy z tej grupy metod są nie tylko wykorzystywane w przygotowaniu danych do analizy, ale coraz częściej są także końcowym etapem procesu odkrywania wiedzy w bazach danych. Wzrost popularności tej grupy metod spowodował powstanie nowej dziedziny wiedzy: eksploracji odchyleń (ang. *outlier mining*).

Nie tylko poszukanie w metadanych zależności i trendów może dać informacje o stanie systemu. W większości przypadków algorytmy eksploracji odchyleń świetnie sprawdzają się w badaniu krytycznych parametrów systemu. Gromadzenie historii wykorzystania zasobów w poszczególnych dniach może pozwolić w łatwy sposób na zidentyfikowanie anomalii na poszczególnych metrykach[1], co może być sygnałem dla pracowników działów IT o nietypowym działaniu systemów przez nich nadzorowanych.

W obecnych czasach ludzie na co dzień korzystają z ogromnej liczby aplikacji. Dostęp „online" do niektórych usług stał się już dla człowieka wręcz naturalny. Można wśród nich wymienić usługi e-bankowości, które w ostatnich latach dynamicznie się rozwinęły. Pomysł na rozszerzenie badań wcześniej realizowanych w odniesieniu do dziedziny finansów (o czym świadczą prace autorów [6, 7, 8]), na systemy monitoringu infrastruktury IT, powstał w czasie obserwowania problemów jednych z najbardziej krytycznych aplikacji w skali globalnej, w ING Services Polska, gdzie pracuje jeden z autorów niniejszej publikacji.

2. Konfiguracja Query Processing Metrics w SAP Adaptive Server Enterprise

Administratorom baz danych ASE[2] zostało udostępnione narzędzie *Query Processing Metrics* (*QPM*), gromadzące wiele statystyk, dotyczących wykorzystania zasobów przez uruchamiane na serwerze zapytania SQL. Wśród nich można odnaleźć czas wykonania zapytania na silniku serwera ASE (*exec_max, exec_min, exec_avg*), czas wykonania zapytania z uwzględnieniem czynników zewnętrznych, jak np. czas oczekiwania na zapis/odczyt z dys-

[1] Metryki – czyli mierzalne statystyki wykorzystania zasobów serwera.
[2] ASE – Adaptive Server Enterprise (SAP/Sybase)

ku czy sieć (*elap_max, elap_min, elap_avg*), liczbę fizycznych odczytów (*pio_max, pio_min, pio_avg*) oraz liczbę logicznych odczytów (*lio_max, lio_min, lio_avg*) zapytania. Narzędzie dostarcza także informacji o liczbie uruchomień na serwerze (*cnt*), ID użytkownika uruchamiającego zapytanie (*uid*) oraz unikalne identyfikatory (*hashkey, gid, id*) [1].

Powyższe dane są zwizualizowane dla każdej bazy z osobna w widokach *sysquerymetrics*. Aby uruchomić mechanizm, należy ustawić parametr konfiguracyjny serwera *enable metrics capture* na wartość 1 [1]:

```
sp_configure "enable metrics capture", 1
```

W następnym kroku należy ustalić progi dla poszczególnych metryk. Jest to kluczowy krok, ponieważ silnik bazodanowy nie gromadzi statystyk dla zapytań SQL, które nie przekroczą żadnego z zadanych progów [1]:

```
sp_configure " metrics elap max", <wartość>
sp_configure " metrics exec max", <wartość>
sp_configure " metrics lio max", <wartość>
sp_configure " metrics pio max", <wartość>
```

Do poprawnego ustalenia wartości progowych należy przeanalizować statystyki uruchamianych na serwerze zapytań SQL. Z tego powodu w pierwszej konfiguracji powinno się ustawić progi na minimalną zakładaną wartość (nawet 0). W następnym kroku dla każdego parametru określa się wartości minimalne, np. z 80% zapytań eksploatujących bazę danych.

W przypadku gdy z serwera ASE korzysta większa liczba aplikacji, istnieje możliwość zgrupowania zapytań SQL z podziałem na aplikacje (o ile korzystają one z różnych baz). Proces archiwizacji widoków *sysquerymetrics* daje taką możliwość przez dodanie identyfikatora aktualnie archiwizowanej bazy:

```
select db_id()
```

W dzisiejszych czasach przeważająca liczba aplikacji opiera swe działanie na wykonywaniu tych samych czynności każdego dnia. Determinuje to powtarzalność egzekwowanych zapytań SQL. Badając i porównując statystyki z wykonania danego zapytania, bądź bloku zapytań, możemy identyfikować nietypowe zachowania serwera, które mogą wpływać na jego wydajność.

Ze względu na brak znacznika czasu w widokach *QPM* warto także wzbogacić archiwizowane dane o datę i godzinę archiwizacji. Jeśli znacznik ten nie zostanie dodany, utracona zostanie możliwość porównywania statystyk SQL'i, a co za tym idzie analizowania wydajności serwera ASE, ponieważ będziemy posiadać jedynie informację o liczbie przebiegów zapytania (*SUM(cnt)*), bez umiejscowienia ich w czasie. Znacznikiem czasu może być, przykładowo, czas serwera:

```
select getdate()
```

3. Wykrywanie odchyleń

W dziedzinie administracji systemami informatycznymi zauważalny jest trend zmierzający do automatyzacji czynności wykonywanych przez pracowników działów IT. Do tej pory dotyczyło to czynności powtarzalnych, uciążliwych. W sytuacjach awaryjnych, wymagających czynności analitycznych, interakcja człowieka z systemem była wymagana. Czy automatyzacja tego typu działań jest również osiągalna?

Coraz bardziej popularnym zagadnieniem są tzw. *inteligentne monitoringi*. Obok przewidywania trendów czasu wysycenia zasobów (np. przestrzeni dyskowej) za pomocą metod regresyjnych, bardzo cenne mogą okazać się również metody wykrywania odchyleń w danych. Poszukiwanie anomalii, zdarzeń wyjątkowych oraz przewidywanie tego typu sytuacji jest jednym z kluczowych zadań administratorów systemów. Spowolnienia w działaniu aplikacji, niecodzienna utylizacja pamięci czy procesorów są jednymi z wielu problemów, które administratorzy mogą napotkać w swojej codziennej pracy, a przed którymi próbują się uchronić. Tematyka ta świetnie wpisuje się w założenia niedawno uformowanej dziedziny wiedzy eksploracji odchyleń (ang. *outlier mining*) [3, 4]. Traktuje ona odchylenia niekoniecznie jako błędy utrudniające, a nawet uniemożliwiające skuteczne odkrywanie wiedzy w bazach danych, a jako nową nieopisaną jeszcze wiedzę, nieraz bardziej wartościową od tej płynącej z pozostałych danych.

Wyróżnia się dwie podstawowe grupy metod: numeryczne oraz graficzne. Dla metod numerycznych w literaturze przytaczany jest jeszcze dodatkowy podział na: metody oparte na rozkładzie danych (ang. *distribution based*), metody oparte na odległości danych (ang. *distance-based*), metody oparte na gęstości danych (ang. *density-based*) oraz metody oparte na grupowaniu danych (ang. *clustering-based*) [3, 6].

Ze względu na charakterystykę systemów monitorujących najbardziej użytecznymi algorytmami eksploracji odchyleń wydają się być te z grupy metod opartych na rozkładzie danych. Można to umotywować faktem, że algorytmy z tej grupy posiadają najczęściej niewielką złożoność obliczeniową [2, 3], co jest priorytetowym założeniem projektowanych systemów tego typu. Ponadto, metadane monitoringów, pomimo swojej wielowymiarowości, są dosyć jednolite. Zauważalna jest znacząca przewaga cech ilościowych nad cechami jakościowymi (dane przeważnie przedstawiają mierzalne metryki), co również może być argumentem do wybrania algorytmu z tej grupy metod. Cechy jakościowe pojawiające się w takich zbiorach często mają jedynie funkcję opisową, co można traktować jako pewną charakterystykę monitorowanego systemu.

Na potrzeby poprzednich prac autorzy przeanalizowali algorytmy, należące do grupy metod opartych na rozkładzie danych. Zaimplementowane i dokładnie przebadane zostały dwa z nich: metoda oparta na średniej i odchyleniu standardowych oraz metoda oparta na rozstę-

pie międzykwartylnym. Eksperymenty dowiodły, że druga z wymienionych metod wykazuje się większą stabilnością, działając w dużych zbiorach danych, co świetnie wpasowuje się w potrzeby projektowanego systemu *inteligentnego monitoringu*. Jej niewielka złożoność obliczeniowa z możliwością różnych modyfikacji (np. wybór innej miary środka lub dobór odpowiedniego algorytmu sortującego) wydaje się być jej dodatkowym atutem [6, 7, 8].

Identyfikowanie anomalii za pomocą metody opartej na rozstępie międzykwartylnym opiera się na wyznaczeniu wszystkich obiektów, będących elementami odstającymi w badanym zbiorze. Oznacza to, że odchyleniem będzie każda obserwacja V_i niemieszcząca się w przedziale: $\langle Q_1 - p \cdot IQR, Q_3 + p \cdot IQR \rangle$, czyli każda wartość, która jest położona przynajmniej o p razy IQR poniżej Q_1 lub p razy IQR powyżej Q_3. We wzorze tym p oznacza parametr sterujący wrażliwością na odchylenia. Widać bowiem, że im większa jest wartość p, tym większy (szerszy) jest przedział wartości uznawanych za typowe dane. Im mniejsza wartość p, tym węższy przedział wartości uznawanych za typowe. Tak więc odpowiednio Q_1 – kwartyl pierwszy, Q_3 – kwartyl trzeci, IQR – rozstęp miedzykwartylny (różnica pomiędzy kwartylem trzecim a kwartylem pierwszym: $IQR=Q_3-Q_1$), zaś p – parametr skalujący wrażliwością testu na odchylenia [2, 3, 4, 6, 7, 8].

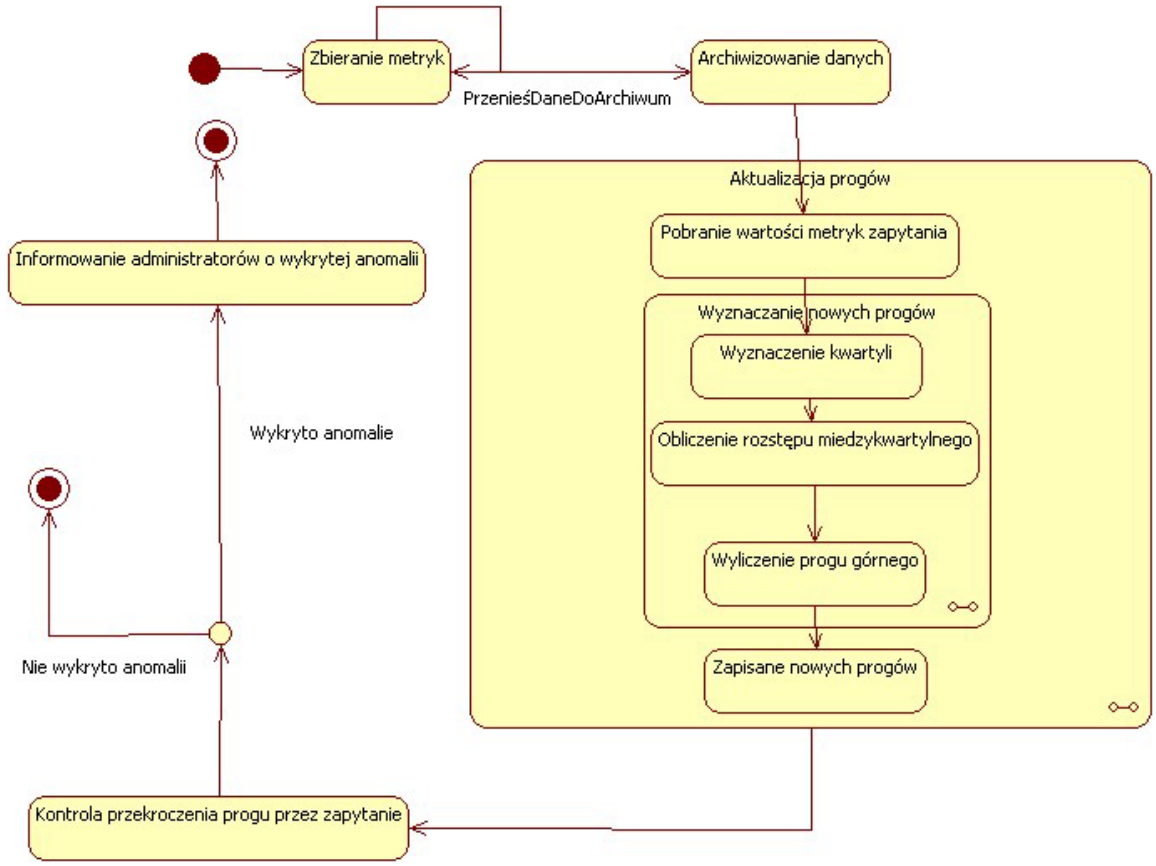
4. Wykrywanie anomalii w Query Processing Metrics

Kluczową ze względu na badanie wydajności serwera ASE jest metryka mechanizmu *QPM*, obrazująca sumaryczny czas wykonania zapytania (*elap_max*). Systemy bazodanowe są systemami deterministycznymi, w których zmiana sposobu działania (na lepsze lub na gorsze) jest zawsze spowodowana przez dokonane na systemie modyfikacje. Dlatego problemy wydajnościowe, działających już jakiś czas systemów, rzadko mają źródło w niepoprawnie pracującym silniku bazodanowym, a raczej należy upatrywać ich w zmianach konfiguracji bądź w czynnikach zewnętrznych.

Przedstawione w *QPM* metryki obrazują wykorzystanie zasobów czasu i pamięci [1]. Aby zredukować złożoność obliczeniową projektowanego systemu, można ograniczyć się do wyliczania jedynie górnego progu, od którego wartości metryk będą identyfikowane jako anomalie ($Q_3+p*IQR$). Ta informacja będzie kluczowa z punktu widzenia monitoringu wydajności lub monitoringu wykorzystania zasobów systemu, ponieważ niskie wartości tych metryk są z zasady pożądane (wyjątkiem może być tutaj sytuacja wypłukania pamięci podręcznej, co spowoduje obniżone odczyty logiczne przy jednoczesnym wzroście po stronie odczytów fizycznych).

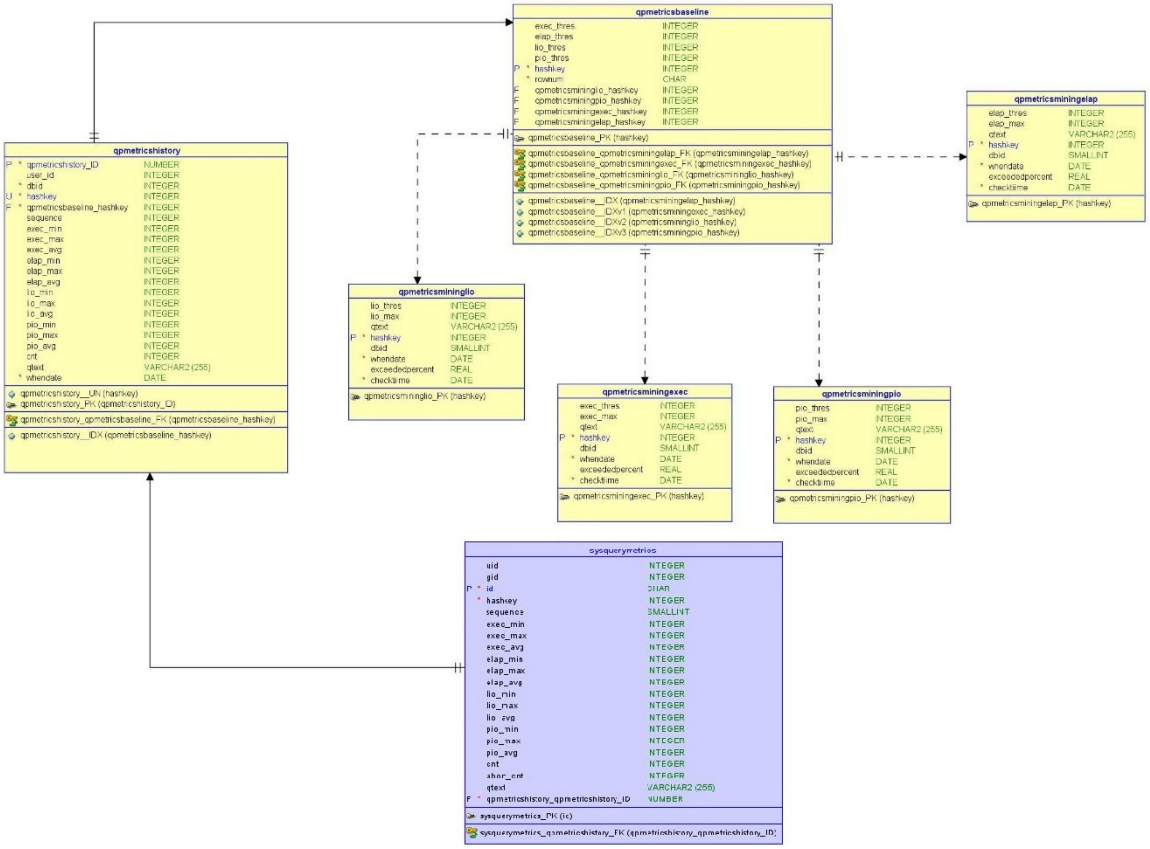
Identyfikowanie odchyleń z mechanizmu *QPM* powinno przebiegać w sposób trójetapowy: zarchiwizowanie widoków *sysquerymetrics* z dodaniem znacznika czasu, obliczenie pro-

gów dla każdej z metryk oraz zgromadzenie uzyskanych danych w tabeli przechowującej bazę zapytań (ang. *baseline*), kontrola przekroczenia progów przez nowo zarejestrowane wystąpienia zapytań oraz gromadzenie historii o wykrytych anomaliach. Przedstawiono to na rys. 1.



Rys. 1. Diagram maszyny stanowej opisujący proces wykrywania odchyleń z widoków sysquerymetrics
Fig. 1. State diagram which describes process of outlier detection from *sysquerymetrics* views

Powyższe czynności powinny być wykonywane cyklicznie, ze spersonalizowanym dla monitorowanego systemu harmonogramem pracy. System wymaga utworzenia kilku powiązanych ze sobą tabel, w których gromadzona będzie historia zapytań i wykrytych anomalii oraz baza progów. Wykrywanie anomalii będzie możliwe przez ciągłą kontrolę nowych rekordów pojawiających się w tabeli *qpmetricshistory* i porównywanie metryk badanego SQL'a z progami zawartymi w tabeli *qpmetricsbaseline*, które wyliczane będą za pomocą zaimplementowanego algorytmu wykrywania odchyleń. Wykryte anomalie powinny zostać archiwizowane z podziałem na kategorie w tabelach: *qpmetricsminingelap, qpmetricsminingexec, qpmetricsminingpio* i *qpmetricsmininglio.* Schemat bazy danych monitoringu wydajności serwera ASE ilustruje rys. 2.



Rys. 2. Model fizyczny bazy danych przechowującej dane o wykrytych odchyleniach w widokach *sysquerymetrics*

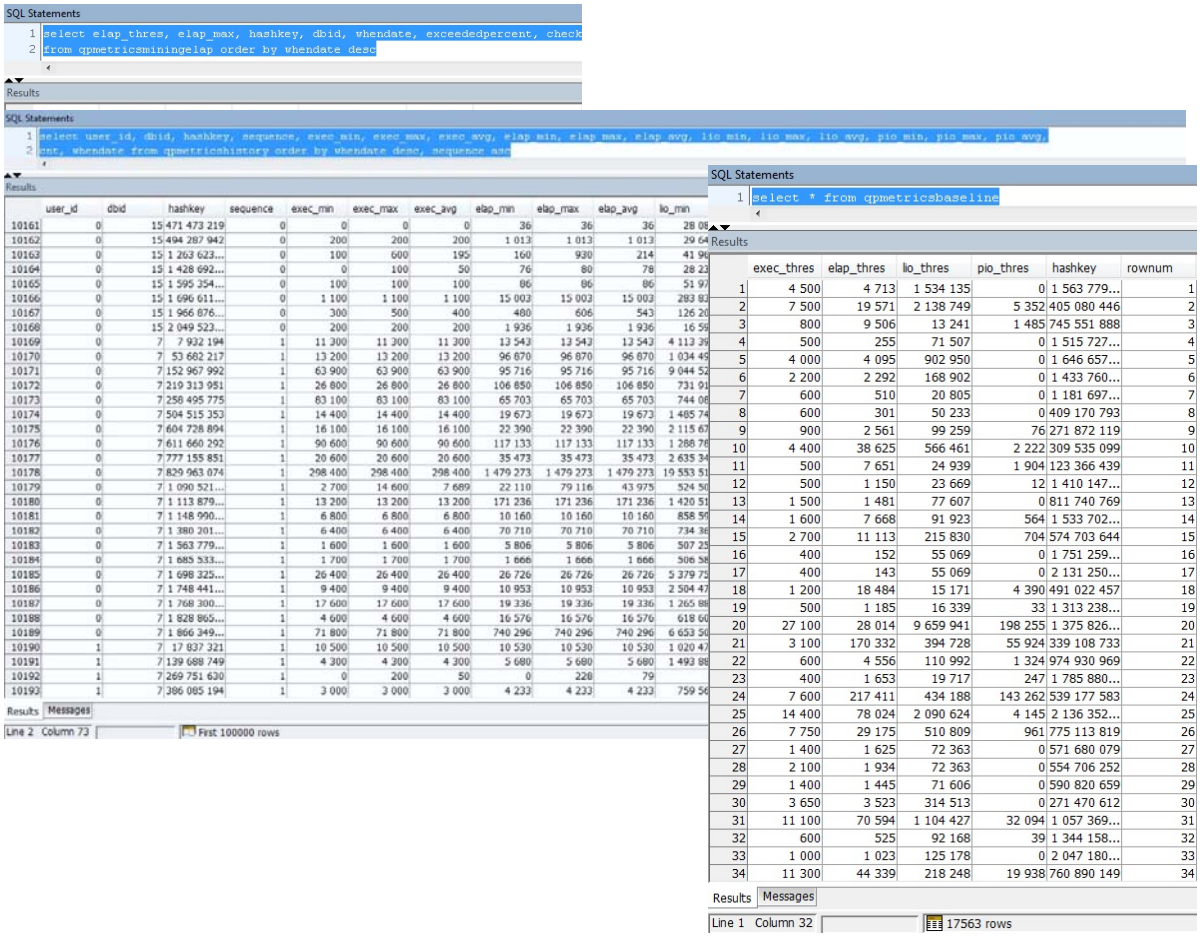
Fig. 2. Physical data model which stores data about detected outliers from *sysquerymetrics* views

Należy zauważyć, że wykryte odchylenia nie od razu świadczą o problemach wydajnościowych aplikacji. Naturalny jest fakt wystąpienia, wśród wielu zapytań zadawanych do bazy w ciągu dnia, kilku anomalii. Nakładanie się na siebie procesów, generowanie niestandardowego raportu obciążającego system, chwilowa zmniejszona wydajność sieci oraz inne zdarzenia mogą wpływać na wykonanie poleceń SQL. Dlatego tworząc automat analityczny do wykrywania spowolnień na serwerze ASE, należałoby zastosować spojrzenie bardziej globalne i zwracać uwagę na fakt, czy spowolnieniom ulega jedna, czy cała grupa uruchamianych w danej jednostce czasu zapytań.

Proste reguły wnioskowania, na metryce *elap_max*, włączone w procedury wykonywane na bazie danych mogłyby sprostać temu zadaniu. Aby skonfigurować działanie takiego kolektora, konieczne będzie dogłębne poznanie aplikacji i jej standardowych zachowań. Bez tej eksperckiej wiedzy system mógłby wykazać się pewną niestabilnością.

Kiedy reguły wnioskowania będą zwracały już oczekiwane rezultaty, dalsza analiza mogłaby polegać na szukaniu w *QPM* przyczyn spowolnień i włączeniu dla nich kolejnych reguł. Przykładowo, jeśli wraz z wysokim wskaźnikiem *elapsed time* szłyby zaniżone logiczne odczyty, z równocześnie zawyżonymi odczytami fizycznymi, oznaczałoby to, że na nasz pro-

ces nałożył się inny, który spowodował wypłukanie pamięci podręcznej bazy. Przykład dostępu do danych monitoringu zaprezentowano na rys. 3.



Rys. 3. Tabele monitoringu wydajności serwera ASE (zaimplementowanego w ING Services Polska Sp. z.o.o.) przechowujące dane o wykrytych odchyleniach

Fig. 3. Tables of ASE server's performance monitoring (implemented in ING Services Polska Sp. z.o.o.) which contain data about detected outliers

5. Wykrywanie anomalii w Query Processing Metrics

Rozdział ma na celu prezentację wyników przeprowadzonych eksperymentów wraz z ich analizą. W dobie coraz bardziej rozpowszechniającego się *pieniądza elektronicznego* instytucjom finansowym zostało postawione nowe wyzwanie. Dostęp do naszych rachunków bankowych, bez konieczności wizyty w oddziale banku, ogromnie zwiększył komfort korzystania z usług sektora bankowego. Należy jednak zwrócić uwagę na fakt, że informacje przechowywane przez banki należą do danych poufnych. Ich wyciek czy dostęp do nich przestępców może mieć bardzo negatywne oddziaływanie na społeczeństwo. Dlatego jednym z głównych zadań pracowników działów IT instytucji finansowych jest ciągłe dbanie o bezpieczeństwo zarządzanych systemów.

Każda niedostępność usług bankowości online lub spowolnienia ich działania mogą powodować pogorszony odbiór oferty instytucji finansowych. Dlatego wśród priorytetowych funkcji administratorów tego typu systemów leży utrzymanie aplikacji w ciągłym, stabilnym ruchu. Problemy wydajnościowe są jednak natury incydentalnej, co czyni je ciężkimi do przewidzenia. Monitoringi kluczowych parametrów systemu (np. wykorzystania procesorów) i chronienie systemu przed niedostępnością możliwe jest do zrealizowania za pomocą nadzorowania utylizacji poszczególnych parametrów. Jednak co z kontrolą i przewidywaniem problemów typu wydajnościowego, które mogą mieć przyczynę w szerokim wachlarzu sytuacji, z wpływem niestabilności działania praktycznie wszystkich warstw infrastruktury IT? Odpowiedzią na te potrzeby mogą być *inteligentne monitoringi*, oparte na algorytmach z dziedziny *data mining*.

W ING Services Polska, na kilku lustrzanych środowiskach (przy zastosowaniu na potrzeby testów różnych konfiguracji), został zaimplementowany i wdrożony system oparty na założeniach opisywanych wcześniej w niniejszej pracy. U podstawy jego zadań leży badanie i wynajdowanie anormalnych przebiegów zapytań SQL. System monitoruje metryki zawarte w historii *QPM* z retencją jednogodzinną, przy codziennym aktualizowaniu bazy zapytań. Po kilku miesiącach jego działania mechanizm zwrócił interesujące wyniki, które są mocno skorelowane z bieżącą kondycją systemu bazodanowego.

Praca monitoringu została poddana dogłębnej analizie i interpretacji. Otrzymane wyniki zostaną zaprezentowane w niniejszym rozdziale. Ze względu na bezpieczeństwo danych i wymagania compliance[3] z przedstawianych tabel usunięta została kolumna *qtext*, przechowująca treści zapytań SQL. Nie ograniczy to przejrzystości prezentowanych badań ze względu na obecność wcześniej opisywanego identyfikatora zapytań (*hashkey*), który można traktować jako unikalny klucz uruchamiany na danym serwerze ASE SQL'i.

W niniejszej pracy przedstawiono dane ze środowiska, na którym otrzymane zostały optymalne wyniki działania monitoringu (optymalne, tzn. najbardziej trafne przy zastosowanej konfiguracji). Wyniki przedstawiają tabele 1-6. W czasie pracy systemu zgromadzonych zostało 9045 unikalnych zapytań, których zanotowano sumarycznie 469132 wystąpienia (469132 zapytania zgromadzone w tabeli *qpmetricsbaseline*, przy jej kwartalnym oczyszczaniu). Wykrytych zostało: 3812 anomalii na metryce *elapsed time*, 2185 anomalii na metryce *execution time*, 8407 anomalii na metryce dotyczącej logicznych odczytów oraz 4892 anomalie na metryce odnoszącej się do odczytów fizycznych (historia wykrytych anomalii nie jest usuwana).

Tabela 1

Zapytanie 1 – anomalie (*elapsed time*)

[3] Compliance – zapewnienie zgodności działalności z regulacjami prawnymi, normami bądź zestawami zaleceń.

elap_thres	elap_max	whendate	exeededpercent	checktime	dbid	hashkey
39430	57103	2015-02-02 03:53:06.096	44.821198	2015-02-02 03:56:05.336	7	1366697728
37301	65943	2015-02-01 04:53:10.58	76.78615	2015-02-01 04:56:07.673	7	1366697728
38433	54063	2015-01-26 04:53:05.35	40.668182	2015-01-26 04:56:06.78	7	1366697728
38433	51050	2015-01-25 05:53:05.92	32.828552	2015-01-25 05:56:05.27	7	1366697728
31272	138756	2015-01-19 02:53:04.556	343.70685	2015-01-19 02:56:04.383	7	1366697728
51230	87936	2014-12-26 02:53:07.376	71.649414	2014-12-26 02:56:03.813	7	1366697728

Tabela 2

Zapytanie 1 – anomalie (*execution time*)

elap_thres	elap_max	whendate	exeededpercent	checktime	dbid	hashkey
11750	21700	2015-01-19 02:53:04.556	84.68086	2015-01-19 02:56:04.383	7	1366697728
12300	18900	2014-12-26 02:53:07.376	53.65854	2014-12-26 02:56:03.813	7	1366697728

Tabela 3

Zapytanie 1 – historia (*wycinek*)

dbid	hashkey	exec_max	elap_max	lio_max	pio_max	cnt	whendate
7	1366697728	9900	54063	1717569	1889	1	2015-01-26 04:53:05.35
7	1366697728	10200	51050	1729347	1683	1	2015-01-25 05:53:05.92
7	1366697728	21700	138756	1693075	1805	1	2015-01-19 02:53:04.556
7	1366697728	9300	27953	1683562	1713	1	2015-01-18 02:53:05.24
7	1366697728	7400	24710	1639213	1840	1	2015-01-12 03:53:03.873
7	1366697728	9000	23793	1644519	1736	1	2015-01-11 04:53:04.48
7	1366697728	4600	22160	1067466	1113	1	2015-01-07 02:53:05.806
7	1366697728	4000	16250	1048816	1165	1	2015-01-05 01:53:08.976
7	1366697728	6000	16606	1045998	1032	1	2015-01-04 01:53:08.156
7	1366697728	6600	21210	1427651	1492	1	2014-12-29 02:53:11.49
7	1366697728	18900	87936	1422878	1473	1	2014-12-26 02:53:07.376

Wśród wykrytych anomalii można doszukać się zapytań o bardzo nieregularnym charakterze działania, które często znajdują się wśród wykrytych anomalii (zapytanie 1), jak i zapytania, których nietypowe zachowanie można uznać za incydentalne (zapytanie 2).

Tabela 4

Zapytanie 2 – anomalie (*elapsed time*)

elap_thres	elap_max	whendate	exeededpercent	checktime	dbid	hashkey

10750	12346	2015-02-01 22:53:07.173	14.846512	2015-02-01 22:56:06.763	7	309535099
10150	23546	2015-01-31 22:53:06.346	131.9803	2015-01-31 22:56:04.803	7	309535099
8192	8960	2015-01-18 22:53:05.53	9.375	2015-01-18 22:56:04.323	7	309535099

Tabela 5

Zapytanie 2 – anomalie (*physical i/o*)

elap_thres	elap_max	whendate	exeededpercent	checktime	dbid	hashkey
146	738	2015-02-01 22:53:07.173	405.47943	2015-02-01 22:56:06.763	7	309535099
133	704	2015-01-31 22:53:06.346	429.3233	2015-01-31 22:56:04.803	7	309535099
166	769	2015-01-25 22:53:06.47	363.25302	2015-01-25 22:56:04.66	7	309535099
141	709	2015-01-24 22:53:06.966	402.83688	2015-01-24 22:56:04.823	7	309535099
142	707	2015-01-18 22:53:05.53	397.88733	2015-01-18 22:56:04.323	7	309535099
130	737	2015-01-17 22:53:05.166	466.9231	2015-01-17 22:56:03.77	7	309535099
130	718	2015-01-11 22:53:05.206	452.30768	2015-01-11 22:56:04.523	7	309535099
121	764	2015-01-10 22:53:04.486	531.40497	2015-01-10 22:56:04.556	7	309535099
105	335	2015-01-04 22:53:07.983	219.0476	2015-01-04 22:56:03.916	7	309535099
105	335	2015-01-03 22:53:08.24	219.0476	2015-01-03 22:56:03.9	7	309535099
105	312	2014-12-28 22:53:09.313	197.14285	2014-12-28 22:56:03.563	7	309535099
105	315	2014-12-27 22:53:10.146	200.0	2014-12-27 22:56:03.023	7	309535099
105	333	2014-12-20 22:53:06.62	217.14288	2014-12-20 22:56:03.41	7	309535099

Pierwszy z przytoczonych typów anomalii powinien być obserwowany i wykrywany przez administratorów. Częste, nietypowe zachowania mogą świadczyć o przewlekłych problemach systemu. W analizowanej sytuacji można zaobserwować, że zapytanie, oprócz wykrytej anomalii w metryce *elapsed time*, jest wychwytywane również w metryce dotyczącej spędzonego czasu na analizie silnika bazodanowego (*execution time*). Może to wskazywać na częste przeliczanie planu wykonania zapytania przez serwer ASE, co z kolei może mieć swoją źródłową przyczynę w niepoprawnie dobranych indeksach. W przypadku tabeli 3 odchyleniami są obiekty 1 i 2, oba jedynie na metryce *elapsed time* (tabela 1), oraz obiekty 3 i 11,

które mają nietypowe wartości zarówno w przypadku metryki *elapsed time* (tabela 1) oraz *execution time* (tabela 2).

Tabela 6

Zapytanie 2 – historia (*wycinek*)

dbid	hashkey	exec_max	elap_max	lio_max	pio_max	cnt	whendate
7	309535099	1300	12346	198778	738	1	2015-02-01 22:53:07.173
7	309535099	1600	23546	200859	704	1	2015-01-31 22:53:06.346
7	309535099	1400	5753	193336	61	1	2015-01-30 22:53:08.283
7	309535099	1700	9870	198672	64	1	2015-01-29 22:53:09.986
7	309535099	1600	5446	204446	60	1	2015-01-28 22:53:09.463
7	309535099	1800	4890	198313	64	1	2015-01-27 22:53:11.776
7	309535099	1100	7716	201155	62	1	2015-01-26 22:53:12.78
7	309535099	1500	8410	203376	769	1	2015-01-25 22:53:06.47
7	309535099	1200	8160	200109	709	1	2015-01-24 22:53:06.966
7	309535099	2100	5536	199110	56	1	2015-01-23 22:53:06.896
7	309535099	1100	4020	198511	63	1	2015-01-22 22:53:06.34
7	309535099	1600	3606	198949	56	1	2015-01-21 22:53:05.863
7	309535099	1500	4176	190527	65	1	2015-01-20 22:53:06.81
7	309535099	1900	4373	195503	57	1	2015-01-19 22:53:05.696
7	309535099	1700	8960	195385	707	1	2015-01-18 22:53:05.53
7	309535099	2600	8046	191005	737	1	2015-01-17 22:53:05.166
7	309535099	1600	4500	200198	64	1	2015-01-16 22:53:06.136
7	309535099	2300	4440	209068	57	1	2015-01-15 22:53:05.856

Drugi z przytoczonych przykładów ma charakter incydentalny. Zapytanie jedynie 3 razy miało wydłużony czas wykonania (tabela 4: obiekty 1, 2, 13 z tabeli 6), do tego nieznacznie przekraczając zakładany próg (z wyjątkiem sytuacji z 31 stycznia, gdzie zanotowane zostało przekroczenie progu o 130%: kolumna *exceededpercent*). Taki typ wykrytej anomalii może świadczyć o niewielkich, chwilowych problemach systemu, ale również poprzedzać jego dużą awarię. Z tego powodu powinien być wykrywany za pomocą automatów, które od razu przy jego wystąpieniu będą informowały administratorów. Szybka reakcja w takiej sytuacji może uchronić system przed niedostępnością. W analizowanej sytuacji (zapytanie 2) warto zauważyć, że wykryty SQL pojawia się zarówno w anomaliach metryki *elapsed time*, jak i *physical i/o*. Łącząc to z jego incydentalnym charakterem, można przypuszczać, że niespodziewanemu wypłukaniu uległa pamięć podręczna serwera (ang. *cache*), co wymusiło na silniku bazodanowym odczyt danych z dysku. Wskazuje na ten fakt również częsta bytność zapytania wśród odchyleń metryki *physical i/o* (tabela 5: obiekty 1, 2, 6, 7, 13, 14 z tabeli 6). Odczyt z dysku jest przeważnie o wiele bardziej czasochłonny niż odczyt z pamięci, stąd zaobserwowane wydłużone wykonanie.

6. Podsumowanie

Bazy danych należą do grupy systemów o znacznym skomplikowaniu. Najbardziej popularne w dzisiejszych czasach systemy relacyjnych baz danych były przez lata rozwijane i udoskonalane, co powoduje ich znaczną złożoność. Opisane przykłady prezentują jedynie skrawek możliwości automatycznej analizy i podejmowania decyzji, w których mogą nas wspomagać metody wykrywania odchyleń w danych.

Koncepcja *inteligentnego monitoringu* prezentowana w niniejszej pracy może pozwolić na identyfikację spowolnień bazy danych już w początkowej fazie. Konieczne będzie cykliczne badanie charakterystycznych okresów działania monitorowanej aplikacji i porównywanie wykrytych anomalii w poszczególnych dniach. Dalsza *automatyczna analiza* może być wzbogacona o podanie przypuszczalnej przyczyny źródłowej zaistniałego problemu.

Inteligentne monitoringi systemów komputerowych mogą być kolejnym polem do wykorzystania metod wykrywania odchyleń w danych. Dowodzi to twierdzeniu, że już nie wyszukiwanie błędów, a poszukiwanie anomalii w danych stało się głównym zadaniem tej dziedziny wiedzy.

BIBLIOGRAFIA

1. Performance and Tuning Series: Query Processing and Abstract Plans. Adaptive Server Enterpise, Sybase, Inc., 2009 http://infocenter.sybase.com/help/topic/com.sybase. infocenter.dc00743.1502/pdf/queryprocessing.pdf (styczeń 2015).
2. Larose D.: Odkrywanie wiedzy z danych, wprowadzenie do eksploracji danych. Wydawnictwo PWN, 2006.
3. Han J., Kamber M., Pei J.: Data Mining: Concepts and Techniques. Elsevier, 2012.
4. Nowak-Brzezińska A.: Eksploracja odchyleń w regułowych bazach wiedzy. Studia Informatica, Vol. 33, No. 2A(105), Gliwice 2012, s. 479÷492.
5. Hawkins D.: Identification of Outliers. Chapman and Hall, 1980.
6. Turos A.: Analiza metod wykrywania odchyleń w danych wielowymiarowych. Praca magisterska, Uniwersytet Śląski, 2013.
7. Nowak-Brzezińska A., Turos A.: Wykrywanie danych nietypowych – nowe podejście. Studia Informatica, Vol. 35, No. 2A (116), Gliwice 2014, s. 187÷198.
8. Nowak-Brzezińska A., Turos A.: Wykrywanie nietypowości w danych rzeczywistych. Systemy Inteligencji Obliczeniowej, Katowice 2014, Rozdział 11, s. 133÷143.
9. Kyriakides E., Polycarpou M. (red.): Intelligent Monitoring, Control, and Security of Critical Infrastructure Systems. Springer, 2015.

10. Garcia D. F.: Performance Modeling and Simulation of Database Servers. The Online Journal on Electronics and Electrical Engineering (OJEEE), Vol. 2, No. 1, Reference Number: W09-0034, s. 183÷188.
11. Lü K. J., Zhu Y.: Performance Metrics and Modelling of Web Database Systems. Proceedings of the 11th International Conference on Database and Expert Systems Applications (Lecture Notes in Computer Science 1875), Springer Verlag, London 2000, s. 805÷814.

Abstract

An article presents issues of performance testing of database servers based on SAP Adaptive Server Enterprise. Nowadays many of relational database systems share an OLTP and a batch processing functions. This model can cause a number of performance problems. Authors propose a solution for performance testing of SAP Adaptive Server Enterprise based on outlier detection from Query Processing Metrics mechanism. Significance of this concept is that slowdowns of the systems have an impact on the reception of the bank's offer and can cause many problems with batch processing (e.g.: delayed creation of the Financial Reports). For these reasons performance testing of database systems is very important, because early detection of an unusual behavior of IT infrastructure can be helpful in system slowdowns prevention.

Adresy

Agnieszka NOWAK-BRZEZIŃSKA: Uniwersytet Śląski, Instytut Informatyki, ul. Będzińska 39, 41-200 Sosnowiec, Polska, agnieszka.nowak@us.edu.pl.
Artur TUROS: ING Services Polska Sp. z o.o., Departament Usług IT, ul. Konduktorska 35, 41-155 Katowice, Polska, Artur.Turos@ingservicespolska.pl.