

Monitor wydajności systemu AMODIT

Last updated by | Adrian Kotowski | 18 lip 2022 at 13:59 CEST

Contents

- [1. Wprowadzenie](#)
- [2. Pojęcie operacji i jej atrybuty, idea monitora systemowe...](#)
- [3. Zasady działania monitora](#)
- [4. Metryki rejestrowane przez monitor](#)
- [5. Konfiguracja monitora](#)
- [6. Zastosowanie monitora](#)
- [Zadania na backlog](#)

1. Wprowadzenie

Monitor wydajności systemu AMODIT to narzędzie wbudowane w aplikację webową AMODIT oraz usługę asynchroniczną, którego celem jest zapisywanie w bazie danych informacji o wykonaniu poszczególnych operacji wraz z dodatkowymi informacjami (metrykami) dotyczącymi tychże operacji. Wspomniane metryki dotyczą głównie czasowych parametrów wykonywania operacji.

Monitor powstał w celu ujednoczenia istniejących wcześniej mechanizmów, które działały niezależnie i składowały dane wynikowe w różnych miejscach. Uniwersalny charakter monitora ma pomoc w analizie wydajności systemu poprzez rejestrowanie szerokiej gamy typów operacji oraz rejestrowanie większej liczby przydatnych metryk, które pozwolą wykryć najbardziej problematyczne operacje, takie jak np.: reguły z bardzo niewydajnym kodem albo joby, które spowalniają system z powodu przetwarzania zbyt dużej liczby plików.

2. Pojęcie *operacji* i jej atrybuty, idea monitora systemowego oraz wykaz dostępnych wariantów typów operacji

Operacją w kontekście monitora nazywamy pojedynczą wyróżnialną z punktu widzenia użytkownika systemu oraz powtarzalną aktywność. Przykładem operacji jest np. wykonanie konkretnej reguły o znanej nazwie z wskazanego procesu. Wszystkie wykonania tej samej operacji w przeciągu zdefiniowanego w konfiguracji monitora interwału czasu są zliczane oraz rejestrowane wspólnie jako jeden wiersz w bazie.

Wiersz reprezentujący zarejestrowaną operację w bazie składa się z następujących atrybutów (kolumn w tabeli):

- atrybuty identyfikujące operacje globalnie** - atrybuty grupujące operacje na najwyższym poziomie (np. podział na organizacje),
- typ operacji** - atrybut dzielący operacje na biznesowe aktywności (np. wykonanie joba, wykonanie reguły),
- lokalne atrybuty identyfikujące operacje (zasoby operacji)** - atrybuty wyróżniające operacje w ramach jednej organizacji,
- metryki operacji** - atrybuty opisujące stan operacji.

Atrybuty identyfikujące globalnie operacje:

- nazwa maszyny serwera aplikacyjnego** - nazwa maszyny na której jest uruchomiony komponent systemu AMODIT, gdzie monitor zarejestrował operację.
- typ źródła** - typ komponentu systemu, który zarejestrował operację, dostępne typy: aplikacja webowa AMODIT (WEB) lub usługa asynchroniczna (AsyncServ).
- nazwa (instancji) komponentu** - nazwa wyróżniająca komponent wśród innych komponentów systemu tego samego typu, w przypadku aplikacji webowej jest to nazwa witryny ustawiana w IIS, natomiast dla usługi asynchronicznej jest to nazwa usługi pod jaką została ona zainstalowana w systemie operacyjnym.
- nazwa organizacji** - nazwa specjalnie wyodrębnionej jednostki w ramach systemu AMODIT dla której rejestrowana jest operacja, organizacja posiada swoją własną bazę danych na potrzeby systemu AMODIT przy jednoczesnym współdzieleniu aplikacji webowej AMODIT (witryny) oraz usługi asynchronicznej z innymi organizacjami. Opisany przypadek istnienia wielu organizacji występuje na środowisku chmurowym zarządzanym przez firmę Astrafox. W przypadków instalacji systemu AMODIT na własnych serwerach klientów zazwyczaj istnieje jedna organizacja.
- nazwa maszyny serwera z bazą danych** - nazwa maszyny z serwerem bazodanowym z którego korzysta komponent systemu AMODIT, gdzie monitor zarejestrował operację.

Każda operacja ma ustalony **typ operacji**, który jest szablonem określający w jaki sposób dana operacja jest rejestrowana oraz jaką aktywność systemu operacja odwzorowuje. Przykładem typu operacji można nazwać np. *wykonanie reguły* (bez specyfikacji o którą regułę chodzi). Wszystkie dostępne typy operacji są przedstawione w tabeli nr 1.

Typ operacji określa jakie **zasoby operacji (atrybuty identyfikujące operacje lokalnie)** oraz **metryki operacji** są wykorzystywane w rejestrowaniu operacji.

Zasób operacji to klucz identyfikujący albo część jego część, która pozwala wyróżnić oraz uszczegółwić operację. Każdy typ operacji może mieć od 1 do 3 zasobów (zatem klucz może składać się z maksymalnie 3 części), które reprezentowane są w formie hierarchii. Taki format danych pozwala później na łatwiejsze grupowanie operacji w raportach na podstawie istniejących już poziomów, które tworzą zasoby ustawione kolejno w hierarchii. Przykładami zasobów, które tworzą klucz identyfikujący operacje typu *wykonanie reguły* są kolejno: *nazwa procesu*, *typ reguły* oraz *nazwa reguły*.

Metryka operacji określa z kolei dane jakie opisują zarejestrowaną operację. Wyróżnić możemy **standardowe metryki**, czyli takie, które są zapisywane dla każdej operacji oraz **metryki niestandardowe**, czyli metryki które są rejestrowane dla operacji w ramach tylko określonego wariantu jakiegoś typu operacji. Metryki zostały opisane dokładniej w dalszej części artykułu w oddzielnej sekcji. Przykładową metryką jest np. liczba wszystkich wywołań danej operacji w ustalonym interwale czasu albo łączny czas ich wykonania.

W ramach jednego typu operacji może istnieć kilka **wariantów**, czyli kilka różnych konfiguracji wybranych zasobów i metryk. Tworzenie kilku wariantów dla jednego typu operacji ma służyć rejestrowaniu także składowych części operacji z bardziej szczegółowymi danymi.

Bardzo dobrym przykładem na zilustrowanie działania wariantów jest przypadek typu operacji rejestrujący *wykonanie joba*. Typ ten posiada kilka wariantów: jeden główny wariant oraz warianty dodatkowe. Wariant główny rejestruje pełne wykonanie wszystkich jobów z osobna, natomiast warianty pomocnicze dodatkowo zapisują informacje na temat wykonania jedynie wybranych części składowych poszczególnych jobów, które z racji wykonywanej funkcjonalności (np. wykorzystanie zewnętrznej biblioteki do dzielenia załączników przychodzących w emailach) mogą mieć kluczowy wpływ na wydajność systemu.

Tabela nr 1. Wykaz wszystkich dostępnych wariantów typów operacji z opisem oraz prezentacją wykorzystywanych zasobów oraz metryk.

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	<Nazwa joba>	#TOTAL#				Wariant główny typu operacji JobExecution, opisuje pełne wykonanie joba, rejestrowane dla wszystkich jobów
JobExecution	RulePeriodicHourly	<Nazwa reguły>				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba RulePeriodicHourly, wariant służy rejestrowaniu wykonania poszczególnych cyklicznych reguł godzinowych
JobExecution	RulePeriodicDaily	<Nazwa reguły>				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba RulePeriodicDaily, wariant służy rejestrowaniu wykonania poszczególnych cyklicznych reguł dobowych

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	RulePeriodicProcess	<Nazwa reguły>				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba RulePeriodicProcess, wariant służy rejestrowaniu wykonania poszczególnych cyklicznych reguł procesu
JobExecution	SynchronizeDictionaries	<Nazwa słownika>				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba SynchronizeDictionaries, wariant rejestruje synchronizację dla każdego słownika oddzielnie

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	LoadMails	FindCaseFromBarcodes	<Nazwa procesu>			Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba LoadMails, wariant rejestruje czynność wyszukiwania spraw po kodzie kreskowym (zapytanie do bazy) w trakcie odczytywania załączników z emaili, czynność może wynikać z ustawień globalnych albo procesu zatem nazwa procesu jest także rejestrowana

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	LoadMails	SplitMessage	<Nazwa procesu>	Liczba załączników po podziale	Łączny rozmiar załączników (w kB)	Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba LoadMails, wariant rejestruje czynność dzielenia załączników z emaili na podstawie barcode albo pustej strony (wykorzystanie zewn. biblioteki GdPicture), czynność może wynikać z ustawień globalnych albo procesu zatem nazwa procesu jest także rejestrowana

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	LoadScans	FindCaseFromBarcodes	<Nazwa procesu>			Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba LoadScans, wariant rejestruje czynność wyszukiwania spraw po kodzie kreskowym (zapytanie do bazy) w trakcie odczytywania skanów z katalogu na dysku/zasobie, czynność może wynikać z ustawień globalnych albo procesu zatem nazwa procesu jest także rejestrowana

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	LoadScans	SplitFile	<Nazwa procesu>	Liczba plików po podziale	Łączny rozmiar plików (w kB)	Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba LoadScans, wariant rejestruje czynność dzielenia skanów pobieranych z katalogu na dysku/zasobie na podstawie kodu kresowego albo pustej strony (wykorzystanie zewn. biblioteki GdPicture), czynność może wynikać z ustawień globalnych albo procesu zatem nazwa procesu jest także rejestrowana

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	LoadScans	ExtractScans		Liczba plików po podziale	Łączny rozmiar plików (w kB)	Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba LoadScans, wariant rejestruje czynność dzielenia skanów pobieranych z wspólnego dla wszystkich procesów wskazanego katalogu na dysku/zasobie na podstawie kodu kresowego albo pustej strony (wykorzystanie zewn. biblioteki GdPicture)
JobExecution	FullTextSearch	GetTextFromPdfFile				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba FullTextSearch, wariant rejestruje czynność odczytywania tekstu z plików o rozszerzeniu PDF (OCR) za pomocą zewn. biblioteki GDPicture

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
JobExecution	FullTextSearch	GetTextFromMsOfficeFile				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba FullTextSearch, wariant rejestruje czynność odczytywania tekstu z plików o rozszerzeniach: .docx, .dotx, .xlsx i .xlsm za pomocą dedykowanej biblioteki
JobExecution	FullTextSearch	GetTextFromFile				Wariant pomocniczy typu operacji JobExecution, rejestrowany tylko dla joba FullTextSearch, wariant rejestruje czynność odczytywania tekstu z plików o wszystkich pozostałych rozszerzeniach (inne niż: pdf, .docx, .dotx, .xlsx i .xlsm) za pomocą dedykowanej biblioteki

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
RuleExecution	<Nazwa procesu>	<Typ reguły>	<Nazwa reguły>			Wariant główny typu operacji RuleExecution, wariant rejestruje wykonanie wszystkich reguł w systemie, niezależnie od ich typu (zarówno reguły ręczne i automatyczne wywoływane przez aplikację webową jak i reguły cykliczne uruchamiane przez usługę synchroniczną)
DictionarySync	<Nazwa słownika>	#TOTAL#				Wariant główny typu operacji DictionarySync, wariant rejestruje wszystkie synchronizacje słowników, zarówno te wykonywane w ramach joba SynchronizeDictionaries, jak i synchronizacje wykonywane przez aplikację webową

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
ReportLoading	<Nazwa raportu>	#DataTable#				Pierwszy wariant typu operacji ReportLoading, wariant rejestruje każdorazowe załadowanie tabeli z danymi wybranego raportu w momencie jak użytkownik wchodzi w aplikację webową w widok raportu
ReportLoading	<Nazwa raportu>	#GroupingTree#				Drugi wariant typu operacji ReportLoading, wariant rejestruje każdorazowe załadowanie drzewka folderów służących (panel po lewej stronie) do grupowania danych w raporcie w momencie jak użytkownik wchodzi w aplikację webową w widok raportu

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
CaseListLoading	<Nazwa widoku systemowego>	Typ użytkownika (admin, regular, temporary, external)				Wariant główny typu operacji CaseListLoading, wariant rejestruje załadowanie listy spraw z wybranego systemowego widoku (Wszystkie, Do wykonania, Przekazane, Usunięte, Zamknięte, Moje, Mój zespół), wariant uwzględnia także typ użytkownika systemu inicjującego czynność (ze względu na różny poziom złożoności sprawdzania uprawnień dla spraw zależny od typu użytkownika)

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
AdvancedSearchLoading	<Nazwa zapisanego widoku użytkownika> lub #SearchViewWithoutName#	Typ widoku (table, folder, list, view)	Typ użytkownika (admin, regular, temporary, external)			Wariant główny typu operacji AdvancedSearchLoading, wariant rejestruje załadowanie listy spraw z wybranego customowego widoku (widok zaprojektowany przez użytkownika i wyświetlany jako zakładka), wariant uwzględnia typ widoku (sposób prezentacji danych) oraz typ użytkownika inicjującego czynność
SkanujToInfo	SendAttachment	<Nazwa procesu>				Wariant typu operacji SkanujToInfo, wariant rejestruje czynność wysłania załącznika do usługi Skanuj.to, wariant uwzględnia proces do którego należy sprawa w której znajduje się wysyłany załącznik

Typ operacji (operationType)	Zasób 1 (resourceLevel1)	Zasób 2 (resourceLevel2)	Zasób 3 (resourceLevel3)	Metryka niestandardowa nr 1 (FileCounter)	Metryka niestandardowa nr 2 (FileSizeCounter)	Opis słowny wariantu typu operacji
RuleFunctionExecution	<Typ reguły>	<Nazwa reguły>	<nr linii w kodzie reguły>_<Nazwa_funkcji>			Wariant typu operacji RuleFunctionExecution, wariant rejestruje fakt wykonania wybranej funkcji reguł, wariant uwzględnia typ reguły, nazwę reguły w której kodzie jest wykonywana funkcja oraz numer linii kodu reguły gdzie znajduje się wywoływana funkcja, aktualnie rejestrowanie operacji jest wspierane tylko dla dwóch funkcji: CallRest oraz CallWebserviceMethod

3. Zasady działania monitora

Istota działania monitora

Monitor systemu aktualnie umożliwia rejestrowanie operacji pochodzących zarówno z aplikacji webowej jak i usługi asynchronicznej. Monitor ten nie jest w żadnym razie zcentralizowanym programem do rejestrowania operacji z wszystkich komponentów systemu. Główną koncepcją działania monitora opiera się na tym, że jest on mechanizmem, który można warunkowo włączyć w ramach istniejących komponentów (usługi i aplikacji webowej). Zatem w przypadku gdy infrastruktura systemu AMODIT składa się z N komponentów to maksymalnie funkcjonować może też i N monitorów. Oczywiście monitory te powinny być tak skonfigurowane, aby zapisywać dane (rejestrowane przez komponenty działające nawet na różnych serwerach i w różnych organizacjach) do tej samej bazy.

Monitor zapisuje dane operacji w ramach organizacji do bazy danych wspomnianej organizacji. Ponadto monitor zapewnia możliwość dodatkowego (opcjonalnego) zapisywania danych z wszystkich organizacji do jednej wskazanej bazy danych (co jest bardzo przydatną opcją na środowisku chmurowym).

Sposób rejestrowania danych

Jeśli uruchomiony w ramach komponentu systemu monitor wykryje wykonanie operacji to zapisuje ten fakt tymczasowo w pamięci programu (jeszcze nie w bazie danych) wraz z odpowiednim zestawem metryk opisującym operację. Przy kolejnym wykonaniu tej samej operacji rekord operacji zapisany w pamięci programu jest aktualizowany (modyfikacja liczby wystąpień operacji oraz pozostałych metryk). Monitor nie rejestruje operacji, które nie zostały wykonane. Zliczanie operacji ma miejsce w **interwałach czasowych**, których czas trwania jest ustawiany w konfiguracji monitora. Po zakończeniu każdego interwału wszystkie zarejestrowane w pamięci programu operacje są zapisywane do bazy danych do tabeli **systemmonitorlog**. Następnie pamięć programu jest czyszczona z zarejestrowanych operacji, by monitor mógł w niej rejestrować nowe operacje z następnego interwału czasu. Cykle ten powtarza się dla każdego kolejnego interwału.

Obrazek nr 1. Struktura tabeli systemmonitorlog. Wykaz kolumn tabeli oraz przypisane im role.

Table: systemmonitorlog

Columns:

smId	int AI PK	
smServerName	varchar(256)	Atrybuty identyfikujące operacje globalnie
smDatabaseServerName	varchar(256)	
smAppInstanceName	varchar(60)	
smOrganizationName	varchar(60)	
smSourceType	varchar(10)	
smOperationType	varchar(60)	Typ operacji
smResourceLevel1	varchar(256)	Atrybuty identyfikujące operacje lokalnie (zasoby operacji)
smResourceLevel2	varchar(256)	
smResourceLevel3	varchar(256)	
smDateFrom	datetime	Metryki operacji
smDateTo	datetime	
smFileCounter	int	
smFileSizeCounter	int	
smExecutionNum	int	
smTimeoutNum	int	
smExceptionNum	int	
smTotalDuration	int	
smAverageDuration	int	
smMaxDuration	int	
smMeasurementDuration	int	
smExecutionShare	decimal(8,2)	

Mechanizm synchronizacji interwałów

Interwały czasowe w ramach monitora są zawsze liczone od godziny 24:00 (północy) poprzedniego dnia. Dzięki takiemu podejściu wszystkie monitory będą zliczały operacje w takich samych

pokrywających się z dokładnością do minuty interwałach czasowych niezależnie od tego kiedy poszczególny monitor został uruchomiony. Warunkiem koniecznym do synchronizacji interwałów jest oczywiście taka sama długość interwału skonfigurowana dla wszystkich monitorów.

4. Metryki rejestrowane przez monitor

Standardowe metryki:

- ExecutionNum** - liczba zliczonych przypadków wywołania danej operacji (w ramach interwału czasu)
- TotalDuration** - łączny czas wykonywania wszystkich zliczonych przypadków wywołania operacji, czas podany w milisekundach
- AverageDuration** - średni czas wykonania operacji wyliczony na podstawie zliczonych przypadków wywołania operacji, czas wyrażony w milisekundach
- MaxDuration** - najdłuższy zarejestrowany czas wykonania operacji spośród zliczonych przypadków wywołania operacji, czas wyrażony w milisekundach
- DateFrom** - początek interwału czasowego w którym zliczane są przypadki wykonania operacji
- DateTo** - koniec interwału czasowego w którym zliczane są przypadki wykonania operacji
- MeasurementDuration** - czas trwania interwału, czas wyrażony w milisekundach
- ExecutionShare** - procentowy udział łącznego czasu wykonania wszystkich zliczonych przypadków operacji w ramach interwału w stosunku do całkowitego czasu trwania interwału, metryka wyrażana w procentach z dokładnością do dwóch miejsc po przecinku
- TimeoutNum** - liczba timeoutów (przerwań wykonywania kodu wynikających z przekroczenia limitu czasu) wyrzuconych w kodzie programu w trakcie wszystkich zliczonych wykonań operacji w ramach jednego interwału czasu
- ExceptionNum** - liczba wyjątków wyrzuconych w kodzie programu w trakcie wszystkich zliczonych wykonań operacji w ramach jednego interwału czasu

Niestandardowe metryki:

- FileCounter** - metryka zliczająca liczbę plików jakie zostały wykorzystane w ramach operacji (informacji o szczegółowym sposobie wykorzystania metryki należy szukać w opisie wariantu typu operacji)
 - FileSizeCounter** - metryka zliczająca rozmiar wszystkich plików jakie zostały wykorzystane w ramach operacji (informacji o szczegółowym sposobie wykorzystania metryki należy szukać w opisie wariantu typu operacji), rozmiar wyrażony w kilobajtach (kB)
-

5. Konfiguracja monitora

Monitor systemu można skonfigurować na trzy sposoby:

- za pomocą konfiguracji w pliku amodit.config
- poprzez konfigurację w pliku web.config/app.config
- korzystając z rejestru systemowego Windows

Jeśli chodzi o konfigurację monitora za pomocą kluczy w rejestrze to wszystkie wymagane do konfiguracji klucze należy dodać w ścieżce: *HKEY_LOCAL_MACHINE\SOFTWARE\Astrafox\AMODIT*.

W przypadku ustawienia konfiguracji na więcej niż jeden sposób decyduje następująca kolejność wedle priorytetu: 1. amodit.config, 2. app.config/web.config, 3. rejestr systemowy.

Parametry konfiguracji monitora (klucze w konfiguracji/klucze w rejestrze systemowym):

- EnableOperationMonitor** - parametr określa czy monitor jest aktywny, akceptowalne wartości: *true* i *false*, domyślną wartością jest *false*,
- OperationMonitorConnectionString** - connection string określający bazę danych, do której mogą być dodatkowo zapisywane dane o operacjach zarejestrowane przez monitor, w przypadku podania pustej wartości dla parametru monitor nie przeprowadza dodatkowego zapisu danych, domyślną wartością jest pusty string, wypełnienie parametru jest zalecane na środowisku chmurowym, gdzie funkcjonuje wiele organizacji i warto aby monitor oprócz zapisywania danych zarejestrowanych dla organizacji do baz bazy tych operacji oddzielnie zapisywał dane z wszystkich organizacji na chmurze do jednej bazy
- OperationMonitorDatabaseType** - parametr określający silnik bazodanowy bazy, do której dodatkowo mają być zapisywane dane z monitora, dostępne wartości: *MSSQL* oraz *MySQL*, domyślną wartością jest silnik bazodanowy bazy skonfigurowany dla aplikacji webowej i usługi ustawiany w ramach głównej konfiguracji bazy danych, silnik bazodanowy w przypadku głównej konfiguracji aplikacji webowej/usługi oraz silnik bazodanowych dodatkowej bazy wskazanej w konfiguracji monitora powinny być takie same (przykład: jeśli system AMODIT korzysta z bazy MySQL to również monitor powinien zapisywać dane zarejestrowanych operacji do bazy MySQL, niekoniecznie tej samej instancji bazy danych i niekoniecznie na tej samej maszynie)
- OperationMonitorTimerInterval** - czas trwania interwału czasowego monitora, wartości parametru są wyrażane w minutach, domyślną interwału wartością jest 60 minut (1 godzina)

Obrazek nr 2. Przykład konfiguracji monitora za pomocą pliku amodit.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="DatabaseType1" value="MSSQL"></add>
    <add key="DatabaseType" value="MySQL"></add>
    <add key="Organizations" value="amod"></add>
    <add key="EnableOperationMonitor" value="true"></add>
    <add key="OperationMonitorConnectionString" value="Server=127.0.0.1;Database=amod;Uid=amod_user;Pwd=[REDACTED];charset=utf8;port=3306"></add>
    <add key="OperationMonitorDatabaseType" value="MySQL"></add>
    <add key="OperationMonitorTimerInterval" value="1"></add>
  </appSettings>
  <connectionStrings>
    <add name="AMODDB" connectionString="Server=127.0.0.1;Database=amodit_quicktest;Uid=amod_user;Pwd=[REDACTED];charset=utf8;port=3306"/>
  </connectionStrings>
</configuration>
```

6. Zastosowanie monitora

Dane z monitora zapisywane do bazy danych mogą być użyte jako zewnętrzne źródło danych, które może zasilić raport zdefiniowany w AMODIT. Rekomendowanym typem raportu jest w tym przypadku raport typu *Pivot*, który wydatnie pozwala pogrupować operacje według określonej kolejności oraz zaprezentować dane z wskazanej metryki z uwzględnieniem przedziałów czasu.

Obrazek nr 3. Przykładowy raport typu Pivot zbudowany na danych z monitora. Raport prezentuje jako metryki łączny czas wykonania wszystkich zliczonych przypadków wywołania operacji oraz liczbę wywołań danej operacji z podziałem na dni miesiąca.



Monitor test

smlServerName	smlOperationType	smlResourceLevel1	smlResourceLevel2	smlResourceLevel3	smlTotalDuration	smlDateTo (Miesiąc i dzień)		
						04-28	04-29	Totals
		AllCases	Admin		619		3.00	3.00
		ForwardedCases	Admin		28		1.00	1.00

DESKTOP-JPPVFI2	CaseListLoading	MyCases	Admin		93		1.00	1.00	
		ReceivedCases	Admin		289		3.00	3.00	
					315		2.00	2.00	
	ReportLoading	Dłuuuuuuuuuuuga nazwa raportu	#DataTable#		147		2.00	2.00	
			#GroupingTree#		65		1.00	1.00	
		Monitor test	#DataTable#		26		1.00	1.00	
					30		2.00	2.00	
					43		1.00	1.00	
				77		2.00	2.00		
	RuleExecution	barcode	Manual	1!@#\$\$%^&*()_+	12		1.00	1.00	
				CallWebservice	192		1.00	1.00	
				dfgdsgsdgdg	8	1.00		1.00	
				test'	51		1.00	1.00	
	RuleFunctionExecution	Manual	CallWebservice	0_CallWebService	134		1.00	1.00	
	Totals							1.00	23.00

Oczywiście istnieje dużo więcej możliwości jeśli chodzi o budowanie raportów na podstawie danych z monitora. Przy definiowaniu raportu można się skupić także na innych metrykach np. procentowy czas wykonywania operacji.

Jedną z przydatnych technik analizy danych z monitora jest tworzenie oddzielnych raportów dla każdego typu operacji. Pomysłem wartym uwagi jest też sortowanie operacji na podstawie danych z metryk dotyczących różnych czasów wykonania. W ten sposób łatwo wskazać, które operacje są wykonywane najdłuższej.

Obecnie, aby utworzyć raport na podstawie danych z monitora należy utworzyć źródło zewnętrzne. W planach jest jednak utworzenie serii predefiniowanych raportów wbudowanych w AMODIT, które będą prezentowały dane z monitora w takiej formie, aby użytkownik samodzielnie mógł ocenić efektywność poszczególnych operacji bez konieczności własnoręcznego tworzenia dodatkowych raportów.

Zadania na backlog

Showing 1 result. [View query](#)

ID	Title	State	Assigned To
6679	> 🏆 Monitor wydajności wersja 3.0	<input type="radio"/> New	