

Małgorzata BACH, Aleksandra WERNER, Adam DUSZEŃKO  
Politechnika Śląska, Instytut Informatyki

## DOBÓR STRUKTUR DANYCH POD KĄTEM OPTYMALIZACJI PRZETWARZANIA ANALITYCZNEGO<sup>1</sup>

**Streszczenie.** Od kilku lat analitycy przyznają, że zapotrzebowanie na moc obliczeniową ze strony aplikacji analitycznych rośnie w tempie wykładniczym. Potrzebne jest więc poszukiwanie nowych technologii wykorzystywania zasobów przez tego typu aplikacje. Wynikiem owych poszukiwań są, proponowane przez coraz więcej firm, rozwiązania bazujące na kolumnowym przetwarzaniu i przechowywaniu danych oraz technologii baz rezydujących w pamięci (ang. *in-memory database*). W celu zweryfikowania tezy głoszącej przewagę nowych rozwiązań nad podejściem klasycznym wykonano wiele badań, których wyniki zestawiono w niniejszym artykule.

**Słowa kluczowe:** przetwarzanie analityczne, bazy rezydujące w pamięci, PowerPivot, OLAP

## SELECTION OF DATA STRUCTURE FOR ANALYTICAL PROCESSING OPTIMIZATION

**Summary.** For several years, the analysts have admitted, that demand for the computing power of analytic applications grows exponentially. For this reason, the new technologies of resource usage are still looking for. The result of these quests are the column-based processing and the in-memory database solutions which are suggested through more and more companies. In order to verify the opinion, the new approach gives more efficient results than classic, a series of tests were executed and the results was summarized in this article.

**Keywords:** In-Memory databases, PowerPivot, column-based approach, OLAP

---

<sup>1</sup> Praca częściowo finansowana ze środków na naukę w latach 2010-2013 jako projekt rozwojowy O R00 0113 12.

## 1. Wstęp

W świecie gospodarki wolnorynkowej i związanej z nią konkurencji informacja jest jednym z największych atutów każdej organizacji. Jednakże wykorzystanie wszystkich cennych danych może być trudne. Aby uzyskać odpowiedź na pytanie biznesowe, często konieczne jest wykonanie tysięcy, a nawet milionów operacji (zależy to między innymi od ilości danych, które trzeba przeanalizować, oraz typów koniecznych agregacji). Z tego powodu realizacja niektórych zapytań może trwać tak długo, że zanim wygenerowana zostanie żądana odpowiedź, dane mogą być nieaktualne, a okazja na zyskowne wykorzystanie zawartej w niej wiedzy dawno minie. Konieczne jest więc poszukiwanie nowych rozwiązań, które przyspieszą ten proces i umożliwią pozyskiwanie informacji w akceptowalnym dla analizującego czasie.

W niniejszym artykule przedstawiono rozwiązanie PowerPivot firmy Microsoft, które wpisuje się we wspomniane poszukiwanie alternatyw dla – często za mało wydajnego – tradycyjnego podejścia, opartego na technologii OLAP. Zaprezentowano porównanie wydajności tego rozwiązania w stosunku do tradycyjnych kostek OLAP, stworzonych z wykorzystaniem Microsoft Analysis Services.

## 2. Wielowymiarowe kostki OLAP a przetwarzanie w pamięci i przetwarzanie kolumnowe

Wielowymiarowe analizy są konieczne w celu uzyskania szybkich odpowiedzi na złożone zapytania biznesowe typu:

*Jaki produkt sprzedawał się najlepiej w południowej Polsce, w 4 kwartale 2011 roku, z rozróżnieniem na sprzedaż bezpośrednią i internetową?*

Przez wiele lat to technologia OLAP (ang. *On-Line Analytical Processing*) – pozwalająca na budowanie wielowymiarowych kostek danych – była wykorzystywana dla uzyskiwania odpowiedzi na tego typu zapytania. Jednakże dynamicznie wzrastająca ilość danych do przeanalizowania powoduje, iż technologia ta staje się w wielu przypadkach za mało wydajna. Można wprawdzie próbować zmniejszyć problem rosnącego czasu dostępu do danych przez zastosowanie odpowiedniej agregacji, kompresji czy wprowadzenie partycjonowania wybranych tabel, jednakże i te zabiegi mogą okazać się niewystarczające.

Kolejną wadą tradycyjnego rozwiązania OLAP jest ograniczenie analizy do konkretnej liczby wymiarów. Konieczne jest definiowanie (przewidywanie) – już na etapie projektowania kostki – pytań, jakie końcowi użytkownicy będą zadawać w przyszłości.

Obecnie, gdy cykle życia produktów są coraz krótsze, wahania na rynkach finansowych częste, a zachowania konsumentów szybko ulegają zmianom, sztywność kostek OLAP jest

sporym problemem. Owszem, można zbudować nową lub przebudować starą kostkę, dodając (modyfikując) miary i wymiary, ale zwykle jest to proces czasochłonny i wymagający od firmy dodatkowych nakładów finansowych. Twórcy tego typu rozwiązań analitycznych zazwyczaj żądają dodatkowej (często niemałej) opłaty za stworzenie możliwości wykonywania nowych rodzajów zestawień.

Z badań przeprowadzonych przez firmę International Data Corporation (IDC) wynika, iż w przypadku rozwiązań opartych na technologii hurtowni i kostek wielowymiarowych aż 65-70% projektów wdrożeniowych kończy się zwykle dopiero po upływie półtora roku (25% firm potrzebuje na to 6-11 miesięcy, 23% 12-17 miesięcy, a 27% 18-48 miesięcy) i jedynie w 36% przypadków udaje się osiągnąć wszystkie zakładane cele. Wielomiesięczne analizy, konieczne do poprawnego skonstruowania hurtowni i modelu analitycznego, i tak samo długie prace wdrożeniowe powodują, iż często firma otrzymuje rozwiązanie, które już na starcie zdezaktualizowało się w znacznej mierze i może realizować tylko część bieżących potrzeb [6].

Wolne od wspomnianych wad wydają się, zyskujące w ostatnim czasie coraz większą popularność, rozwiązania bazujące na technologii przetwarzania danych w pamięci operacyjnej (ang. *In-Memory Computing*). Wzrost pojemności pamięci i ogromny spadek ich cen (średnia cena 1 GB pamięci DRAM w 2000 r. wynosiła ok. 1000 \$, w 2005 r. – poniżej 100 \$, a obecnie ok. 20 \$) spowodowały, że powrócono do znanej od dawna – ale wcześniej niemożliwej do zrealizowania – koncepcji, aby systemy zarządzające bazami danych opierały się na pamięci operacyjnej jako głównej i wręcz jedynej przestrzeni gromadzenia danych [8].

Coraz większa liczba dostawców rozwiązań analitycznych próbuje implementować lub imitować wykorzystanie technologii in-memory. Zdaje się to potwierdzać prognozy analityków Gartner Group, którzy przewidują, iż w najbliższych latach 70% największych i najszybciej rozwijających się firm na świecie (uwzględnionych w rankingu Global 1000) będzie wykorzystywać ładowanie szczegółowych danych do pamięci operacyjnej jako podstawową i najważniejszą metodę optymalizacji wydajności i elastyczności aplikacji analitycznych [7].

W ciągu ostatnich lat, między innymi w kontekście omawianych problemów, zaczęto się również poważnie zastanawiać nad przechowywaniem danych w sposób kolumnowy – tzn. na każdej stronie danych przechowywane są dane z jednej kolumny. W tradycyjnej relacyjnej bazie dane są przechowywane wierszami. Oznacza to, że na każdej stronie przechowywany jest wiersz lub wiersze tabeli. Taka zmiana wydaje się korzystna ze względu na fakt, iż w zapytaniach analitycznych rzadko pobierane są pojedyncze wiersze, a częściej wykonywane różnego rodzaju agregaty na poszczególnych kolumnach. W przypadku organizacji kolumnowej zmniejsza się liczba niezbędnych operacji I/O, ponieważ pobierane są jedynie te kolumny, które akurat potrzebne są do realizacji danego zapytania. Kolejną istotną kwestią jest

fakt, że dane z jednej kolumny o wiele łatwiej skompresować niż dane z kolejnych wierszy. Dzieje się tak dlatego, że wartości w jednej kolumnie są zwykle mało zróżnicowane – są tego samego typu.

W kolejnym punkcie zostanie przedstawione rozwiązanie PowerPivot, w którym zastosowano mechanizm zwany VertiPaq, łączący w sobie ideę podejścia kolumnowego i przetwarzania danych w pamięci.

### 3. PowerPivot

Rozwiązanie PowerPivot firmy Microsoft jest dostępne w dwóch wersjach – jako rozszerzenie lokalnie pracującego programu Excel oraz jako narzędzie integrujące się z SharePoint 2010.

To właśnie PowerPivot jest pierwszym produktem korzystającym z nowego mechanizmu VertiPaq przetwarzania danych w pamięci oraz przechowywania kolumnowego (obecnie jeszcze szerzej wykorzystywany jest on w najnowszej wersji SQL Server 2012). Dzięki użyciu aparatu działającego w pamięci i wydajnych algorytmów kompresji PowerPivot umożliwia przetwarzanie z wysoką wydajnością nawet bardzo dużych zestawów danych [4, 9]. Podczas ładowania dane poddawane są silnej kompresji, co redukuje ich rozmiar w stosunku do wielkości analogicznych danych przechowywanych w bazie relacyjnej. Podczas pracy z narzędziem cały plik z danymi ładowany jest do pamięci RAM i bezpośrednio w niej liczone są potrzebne agregaty.

Zintegrowanie nowoczesnego rozwiązania PowerPivot z arkuszem kalkulacyjnym może się początkowo wydać dziwne. Prawda jest jednak taka, iż w wielu przedsiębiorstwach, niezależnie od stopnia zaawansowania wykorzystywanych narzędzi informatycznych wspomagających bieżącą działalność firmy (systemów klasy ERP czy CRM) czy też rozwiązań z obszaru Business Intelligence, i tak w powszechnym użyciu wśród analityków, księgowych czy też pracowników działów analiz, planowania i kontroli jest arkusz kalkulacyjny Excel. Często kadra zarządzająca życzy sobie, aby końcowe wyniki analiz przedstawiane były właśnie w postaci tabel czy wykresów w Excelu.

Twórcy rozwiązania PowerPivot postanowili więc nie walczyć z przyzwyczajeniami, lecz podjęli próbę pogodzenia dwóch światów – świata IT i świata biznesu (stąd robocza nazwa projektu: Gemini, czyli zodiakalny bliźniak), które można wyróżnić w każdej organizacji.

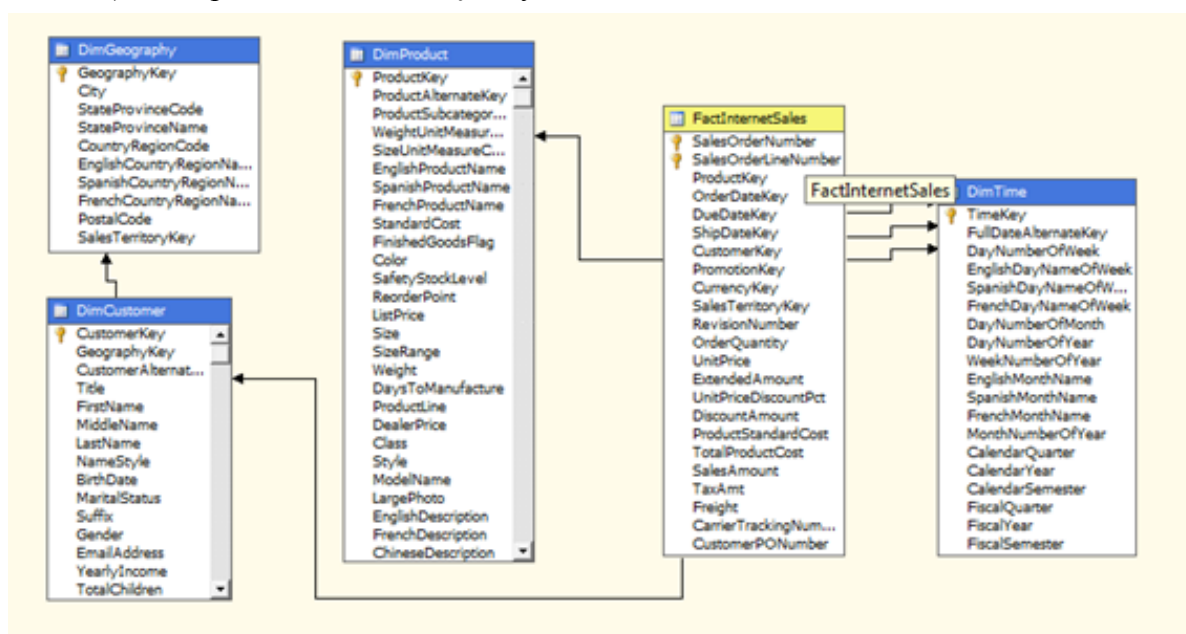
## 4. Opis przeprowadzonych testów

W celu weryfikacji coraz popularniejszej opinii o przewadze rozwiązań kolumnowych i in-memory postanowiono przeprowadzić testy sprawdzające, czy faktycznie, a jeżeli tak, to czy w każdym przypadku i o ile, te rozwiązania są lepsze od standardowych kostek OLAP.

Stanowisko badawcze stanowiły komputery z procesorem Intel® Core™ i5-2520M, 2GHz i 8GB pamięcią RAM.

Analizowana baza danych zawierała informacje o sprzedaży internetowej (m.in. dane o klientach, sprzedawanych produktach oraz czasie zakupu).

Tradycyjne rozwiązanie OLAP zostało stworzone z wykorzystaniem SQL Server Analysis Services (SSAS). Użyta do analiz wielowymiarowa kostka (wykorzystano architekturę MOLAP) miała postać zilustrowaną na rys. 1.



Rys. 1. Struktura kostki (ang. cube) wykorzystywanej w SSAS

Fig. 1 SSAS cube structure

Źródła danych dla poszczególnych tabel wymiarów i faktów w początkowej fazie testów zawierały odpowiednio:

- DimCustomer – 18 484 wierszy,
- DimGeography – 655 wierszy,
- DimProduct – 606 wierszy,
- DimTime – 1158 wierszy,
- FactInternetSales – 60 398 wierszy.

Dla tabel wymiarów stworzono indeksy zgrupowane (ang. *clustered*) na kolumnach kluczy głównych (odpowiednio: *CustomerKey*, *ProductKey*, *GeographyKey*, *TimeKey*). Nato-

miast dla tabeli faktów – indeksy niezgrupowane (ang. *nonclustered*) na kolumnach kluczy obcych<sup>2</sup>.

Do testów przygotowany został zbiór kilkunastu zapytań, zróżnicowanych pod względem selektywności oraz złożoności, różniących się liczbą analizowanych miar i wymiarów. Ostatecznie na potrzeby niniejszego artykułu wybrano 8 najbardziej – zdaniem autorów – reprezentatywnych.

Przykładowe zapytanie, realizujące zestawienie liczby zamówionych rowerów danego modelu w danym kolorze z podziałem na miasta, z których pochodzą klienci, sformułowane w języku MDX (ang. *Multidimensional Expressions*)<sup>3</sup>, wyglądało następująco:

```
SELECT crossjoin([Dim Product].[Color].members,
[Dim Product].[Model Name].members) on columns,
[Dim Customer].[City].members on rows
FROM [Cube_1]
WHERE ([Measures].[Order Quantity]);
```

Jako źródło danych w programie PowerPivot posłużyły te same tabele, na podstawie których zbudowano opisaną wcześniej kostkę. Jednak w tym przypadku nie budowano żadnej struktury wielowymiarowej, gdyż stosowna struktura była tworzona dynamicznie w pamięci RAM w trakcie realizacji zestawień. Dla zwizualizowania i analizy danych budowano tabele przestawne, zwracające analogiczne zestawienia jak w przypadku tradycyjnego rozwiązania OLAP, tworzonego z wykorzystaniem SSAS. Dzięki zainstalowaniu dodatku OLAP PivotTable Extensions możliwe było śledzenie odpowiednika zapytania MDX, który realizowany jest przez VertiPaq (informacje te można również uzyskać, korzystając z narzędzia SQL Server Profiler). Przykładowo odpowiednik przedstawionego wcześniej zapytania wyglądał następująco:

```
SELECT NON EMPTY
Hierarchize(DrilldownMember(CrossJoin({[DimProduct].[Color].[All]},
[DimProduct].[Color].[Color].AllMembers},
{([DimProduct].[ModelName].[All])}),
[DimProduct].[Color].[Color].AllMembers,
[DimProduct].[ModelName]))DIMENSION PROPERTIES PARENT_UNIQUE_NAME,
HIERARCHY_UNIQUE_NAME ON COLUMNS,
NON EMPTY Hierarchize({DrilldownLevel({[DimGeography].[City].[All]}),,,
INCLUDE_CALC_MEMBERS)) DIMENSION PROPERTIES PARENT_UNIQUE_NAME,
HIERARCHY_UNIQUE_NAME ON ROWS
FROM [Sandbox]
WHERE ([Measures].[Suma OrderQuantity]) CELL PROPERTIES VALUE,
FORMAT_STRING, LANGUAGE, BACK_COLOR, FORE_COLOR, FONT_FLAGS;4
```

<sup>2</sup> Nie testowano możliwości zastosowania innych typów indeksów czy mechanizmów mogących wpłynąć na efektywność. Zastosowanie wszelkich konstrukcji dodatkowych oczywiście może poprawić wydajność, ale celem badań było wskazanie zróżnicowania w punkcie wyjściowym badanych rozwiązań.

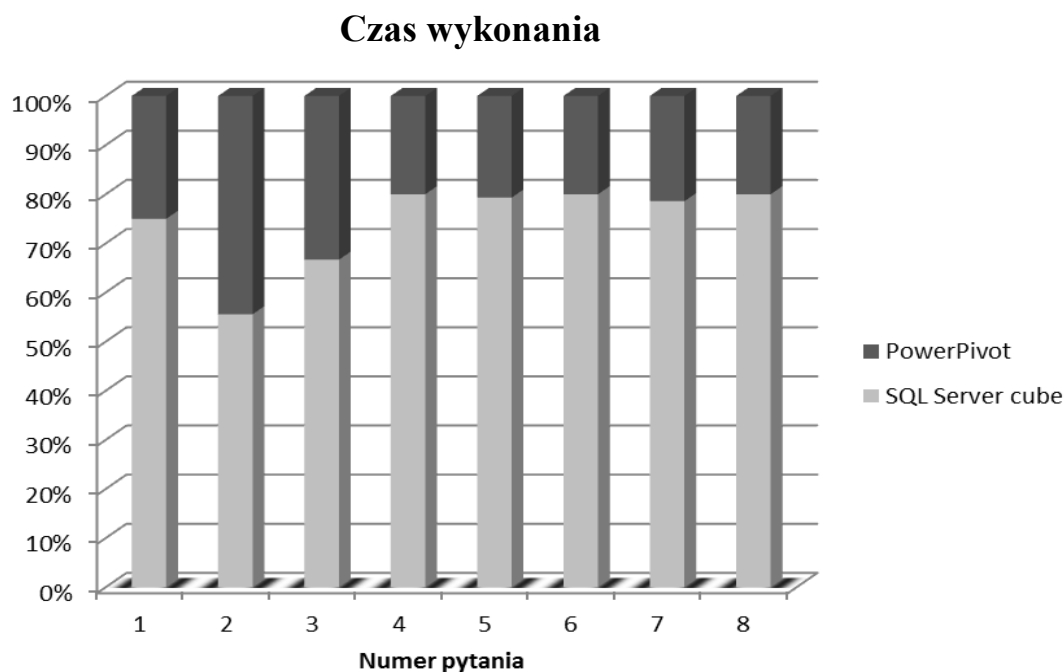
<sup>3</sup> Język MDX został stworzony przez inżynierów Microsoft – jest wielowymiarowym rozwinięciem języka T-SQL.

<sup>4</sup> Strukturze tworzonej w pamięci standardowo nadawana jest nazwa „SandBox”.

## 5. Analiza wyników

Jak już wspomniano w poprzednim punkcie, do testów przygotowano zestaw pytań różniących się stopniem złożoności. Na rysunku 2 przedstawiono wykres obrazujący stosunek czasu wykonania tych samych zapytań z wykorzystaniem tradycyjnego rozwiązania OLAP oraz rozwiązania PowerPivot. Jak widać, czas wykonania w przypadku PowerPivot jest od 1,25 (pytanie 2) do 4 (pytania 4, 6, 8) razy krótszy niż w przypadku standardowych rozwiązań SSAS.

W celu oceny skalowalności obu rozwiązań zwiększono liczbę wierszy w tabelach DimCustomer (z 18 484 do 110 904) oraz FactInternetSales (z 60 398 do 1 509 950).



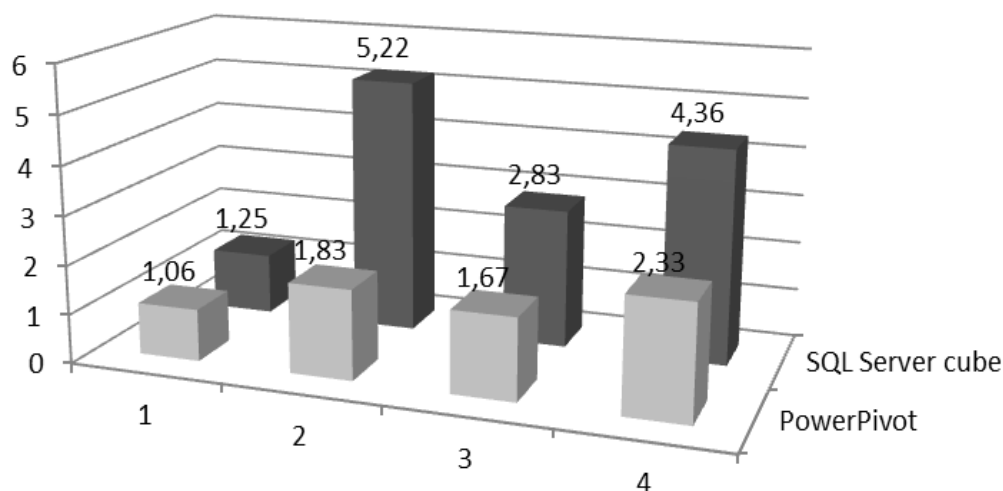
Rys. 2. Czasy wykonania pytań z wykorzystaniem SSAS i PowerPivot

Fig. 2. Execution time of SSAS and PowerPivot query

Ponieważ wspomniane zmiany miały największy wpływ na cztery spośród testowanych pytań (były to odpowiednio pytania: 3, 5, 6 i 7 – pozostałe nie uwzględniały wymiaru klienta), więc na rysunku 3 zaprezentowano wykres obrazujący wzrost czasu wykonania właśnie tych czterech zapytań.

W przypadku tradycyjnych kostek OLAP czas potrzebny na wykonanie zapytań wzrósł (w stosunku do czasu wykonania tych samych zapytań dla bazy o pierwotnej wielkości) od 1,25 (pytanie 1) do 5,2 raza (pytanie 2), podczas gdy w przypadku rozwiązania PowerPivot najmniejszy przyrost wyniósł 1,06 (dla pytania 1), natomiast największy – 2,3 raza (dla pytania 4).

### Wzrost czasów wykonania

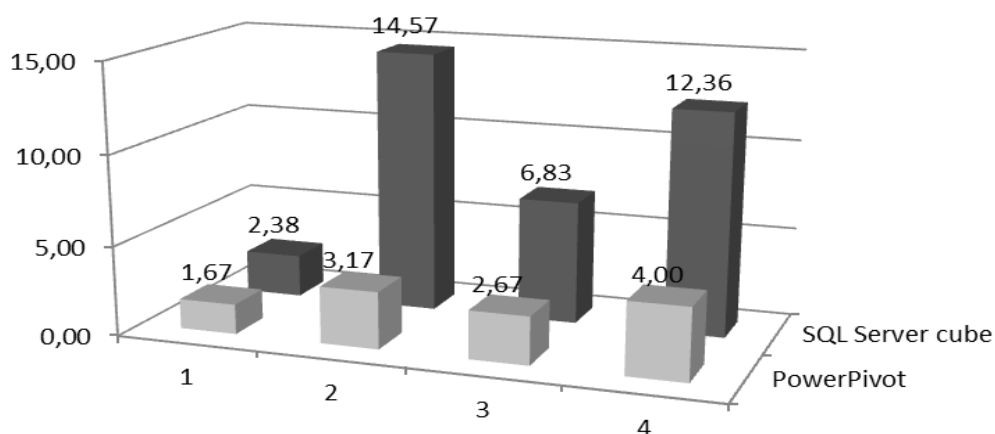


Rys. 3. Skalowalność rozwiązania (wzrost czasu wykonania przy wzroście liczby analizowanych danych – do 110 904 wierszy w tabeli DimCustomer oraz do 1 509 950 w tabeli FactInternetSales)

Fig. 3. The scalability test (110 904 rows in w DimCustomer table and 1 509 950 rows in FactInternetSales table)

Badania jeszcze raz powtórzono po kolejnym zwiększeniu liczby rekordów w tabelach DimCustomer i FactInternetSales odpowiednio do 332 712 oraz 4 529 850 wierszy. Na rysunku 4 przedstawiono przyrost czasu wykonania zapytań w stosunku do czasu odnotowanego dla początkowej liczby wierszy w tabelach.

### Wzrost czasów wykonania



Rys. 4. Skalowalność rozwiązania (wzrost czasu wykonania przy wzroście liczby analizowanych danych – do 332 712 wierszy w tabeli DimCustomer oraz 4 529 850 w tabeli FactInternetSales)

Fig. 4. The scalability test (332 712 rows in w DimCustomer table and 4 529 850 rows in FactInternetSales table)



Jak można zauważyć, PowerPivot znacznie lepiej radzi sobie przy zwiększaniu rozmiaru analizowanych danych. W przypadku rozwiązania PowerPivot czas wzrósł maksymalnie 4-krotnie (pytanie 4), podczas gdy dla tradycyjnej kostki OLAP (dla pytania 2) ponad 14-krotnie.

Niestety wraz ze wzrostem liczby analizowanych danych zaobserwowano, iż w rozwiązaniu PowerPivot ładowanie danych oraz przygotowanie odpowiednich struktur w pamięci RAM trwają niepokojąco długo. Dla tabel DimCustomer i FactInternetSales, zawierających odpowiednio 332 712 oraz 4 529 850 wierszy, zauważono, iż zajętość pamięci RAM w trakcie tworzenia niektórych zestawień przekraczała często 95%, co oznacza próbę wykorzystania pamięci wirtualnej. Oczywiście praca systemu o charakterze rezydującego w pamięci [8], jakim jest PowerPivot, w takich warunkach nie jest wskazana. Pierwsze spowolnienie pracy spowodowane jest próbą systemu operacyjnego zwolnienia części zajętej pamięci operacyjnej. Angażuje to bardzo mocno operacje dyskowe, składające najrzadziej używane obszary pamięci RAM na dysk twardy. Jednak w sytuacji, gdy większość pamięci jest zajęta przez system PowerPivot i system operacyjny, nie jest możliwe zwolnienie takiej ilości pamięci, jaka jest niezbędna. Generalnie jest to zachowanie typowe dla systemów rezydujących w pamięci, wymagających, aby przestrzeń dostępnej pamięci operacyjnej mogła objąć wszystkie struktury wykorzystywane do reprezentacji analizowanych danych.

Wobec powyższego ewentualne dalsze badania wydajności rozwiązania PowerPivot muszą być przeprowadzone z wykorzystaniem komputera wyposażonego w pamięć operacyjną o większej pojemności.

Przy dalszym zwiększaniu ilości analizowanych danych należy również uwzględnić możliwość partycjonowania danych wykorzystywanych w tradycyjnym rozwiązaniu OLAP. W opisanych badaniach nie stosowano partycji, ponieważ byłyby one zbyt małe i zamiast przyspieszyć, mogłyby jeszcze spowolnić wykonanie zapytań. Wskazaniem do zastosowania podziału na partycje (prócz liczby wierszy) jest czas poświęcony na operacje związane ze składowaniem danych. Jeżeli 30% (lub więcej) czasu wykonania zapytania MDX pochłania silnik składowania (procent ten można określić, śledząc zdarzenia „Query Subcube” w narzędziu SQL Server Profiler), wówczas należy zastosować podział danych na partycje. W zadnym z wykonywanych, w ramach opisanych testów, zapytań nie osiągnięto wspomnianego progu 30%. Przykładowe czasy realizacji zdarzeń „Query Subcube” oraz całkowity czas realizacji dla jednego z analizowanych zapytań MDX przedstawiono na rysunku 5.

EventClass	Duration	EventSubclass	TextData	ConnectionID	NTUserName	ApplicationName	IntegerData	Start Time
Progress Report End	62	14 - Query	Zakończono odczyt danych z partycji...	21	Gosia		0	2012-01-07 16
Query Subcube	62	2 - Non-ca...	000000000100000000000000000000,00...	21				2012-01-07 16
Query End	967	0 - MDXQuery	select [Dim Product].[color].chldr...	21	Gosia	Microsoft SQ...		2012-01-07 16
Query Begin	0	0 - MDXQuery	select crossjoin([Dim Product].[co...	21	Gosia	Microsoft SQ...		2012-01-07 16
Query Subcube	0	1 - cache ...	0000000000000000000000000000,00...	21				2012-01-07 16
Progress Report Begin	14	14 - Query	Rozpoczęto odczyt danych z partycji...	21	Gosia			2012-01-07 16
Progress Report End	94	14 - Query	Zakończono odczyt danych z partycji...	21	Gosia		0	2012-01-07 16
Query Subcube	94	2 - Non-ca...	0000000001000000000100000000,00...	21				2012-01-07 16
Query Subcube	62	1 - cache ...	000000000000000000000100000000,00...	21				2012-01-07 16
Query Subcube	0	1 - cache ...	00000000000000000000000100000000,00...	21				2012-01-07 16
Query Subcube	31	1 - cache ...	0000000001000000000000000000,00...	21				2012-01-07 16
Query Subcube	62	1 - cache ...	0000000001000000000100000000,00...	21				2012-01-07 16
Query Subcube	31	1 - cache ...	0000000000000000000000000000,00...	21				2012-01-07 16
Query Subcube	62	1 - cache ...	000000000000000000000100000000,00...	21				2012-01-07 16
Query Subcube	0	1 - cache ...	0000000001000000000000000000,00...	21				2012-01-07 16
Query Subcube	0	1 - cache ...	0000000001000000000000000000,00...	21				2012-01-07 16
Query End	331672	0 - MDXQuery	select crossjoin([Dim Product].[co...	21	Gosia	Microsoft SQ...		2012-01-07 16
Query Begin	0	0 - MDXQuery	select [Dim Product].[Model Name].m...	21	Gosia	Microsoft SQ...		2012-01-07 16
Progress Report Begin	14	14 - Query	Rozpoczęto odczyt danych z partycji...	21	Gosia			2012-01-07 16
Progress Report End	16	14 - Query	Zakończono odczyt danych z partycji...	21	Gosia		0	2012-01-07 16
Query Subcube	16	2 - Non-ca...	000000000000000000000100000000,00...	21				2012-01-07 16
Query Subcube	0	1 - cache ...	0000000000000000000000000000,00...	21				2012-01-07 16
Query Subcube	0	1 - cache ...	0000000000000000000000000000,00...	21				2012-01-07 16

Rys. 5. SQL Server Profiler – przykładowa analiza czasu wykonania  
 Fig. 5. SQL Server Profiler – execution time analysis

## 6. Zakończenie

Konieczność wydajnej analizy coraz większych ilości danych powoduje, iż dotychczas stosowane rozwiązania często stają się niewystarczające i konieczne jest poszukiwanie innych, bardziej efektywnych. Firma Microsoft, dostrzegając wspomniane problemy, opracowała mechanizm VertiPaq przetwarzania danych w pamięci oraz przechowywania kolumnowego. Zastosowanie tego nowoczesnego rozwiązania spowodowało znaczny wzrost pozycji firmy Microsoft w dziedzinie Business Intelligence, o czym można się przekonać, analizując raport firmy Gartner, zwany magicznym kwadratem BI (ang. *Magic Quadrant for Business Intelligence*), opublikowany w 2011 r. [3].

Przeprowadzone testy wykazały, iż PowerPivot pozwala na znacznie szybszą realizację zapytań biznesowych (analitycznych) niż tradycyjne rozwiązania, bazujące na wielowymiarowych kostkach (zrealizowanych z wykorzystaniem SSAS). Pokazały również, iż znacznie lepiej radzi sobie z przyrostem analizowanych danych, oczywiście pod warunkiem, iż dysponujemy maszynami o odpowiedniej ilości pamięci operacyjnej, pozwalającej na utworzenie wszystkich niezbędnych struktur wykorzystywanych do reprezentacji danych.

Uzyskane wyniki pozwalają potwierdzić, iż wykorzystanie pamięci RAM jako głównej i wręcz jedynej przestrzeni gromadzenia i przetwarzania analitycznego danych jest kierunkiem słusznym. Ta teza zostanie poddana dalszej weryfikacji w kolejnych badaniach, w których planuje się wykorzystanie najnowszej wersji Analysis Services 2012, oferującej oprócz

klasycznego trybu wielowymiarowego nowy tryb tabelaryczny, wywodzący się w prostej linii od silnika Vertipaq, wcześniej stosowanego w PowerPivot.

## BIBLIOGRAFIA

1. Mistry R., Misner S.: *Introducing Microsoft SQL Server 2008 R2*. Microsoft Press, 2010.
2. Vesset D., McDonough B., Wilhide K., Wardley M., McCullough R., Sonnen D.: *Worldwide Business Analytics Software 2007-2011 Forecast Update and 2006 Vendor Shares. Business Intelligence, Data Warehousing, and Analytics Applications Forecasts Point*, IDC.
3. Gartner-Magic Quadrant for Business Intelligence Platforms, <http://sqlresearch.com/2011/02/02/gartnermagic-quadrant-for-business-intelligence-platforms/> (stan na styczeń 2012).
4. 10 głównych powodów, dla których warto wypróbować dodatek PowerPivot dla programu Excel 2010, <http://office.microsoft.com/pl-pl/excel/10-gownych-powodow-dla-ktorych-warto-wyprobowac-dodatek-powerpivot-dla-programu-excel-2010-HA101810443.aspx> (stan na maj 2011).
5. Mendrala D., Szeliga M.: *Serwer SQL 2008. Usługi biznesowe. Analiza i eksploracja danych*. Helion, 2009.
6. Vesset D., McDonough B.: *Worldwide Business Intelligence Tools 2008 Vendor Shares*. IDC, Framingham 2009.
7. Mejsner B.: Turbo BI pod strzechy, <http://cio.cxo.pl/news/344542/Turbo.BI.pod.strzechy.html> (stan na grudzień 2011).
8. Bach M., Duszeńko A., Werner A.: *Koncepcja pamięciowych baz danych oraz weryfikacja podstawowych założeń tych struktur*. *Studia Informatica*, Vol. 31, No. 2B(90), Wydawnictwo Politechniki Śląskiej, Gliwice 2010.
9. Winston W.L.: *Microsoft Excel 2010. Analiza i modelowanie danych biznesowych*. PROMISE, 2011.

## Abstract

With the increasing need to analyze a large amount of data, the requirements concerning the analytical processing system reaction time increasing also. For this reason, the new data models and methods existing models extensions are still looking for. For example the con-

cepts of column-based systems and in-memory database are response to the growing demand for efficient processing of large amounts of data needed for multi-dimensional analysis.

During the research the free add-in to the 2010 version of the spreadsheet application Microsoft Excel – PowerPivot – were studied. PowerPivot has novel mechanism VertiPaq, combining the idea of the column-based approach and the in-memory processing.

Besides, specialized tools to support decision-making processes – the OLAP cubes, which are an integral part of MS SQL Server database management tools, were used.

As a critical parameter of performed tests, the data access time was taken. As a result of the research it was found that although the traditional OLAP cubes are now the primary source of data analysis supporting processes, the database architecture changing – especially architecture oriented to optimize search time of data stored in columns and in-memory processing – brings the more effective analysis of its contents than classic solution.

Wpłynęło do Redakcji 31 stycznia 2012 r.

### **Adresy**

Małgorzata BACH: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, malgorzata.bach@polsl.pl.

Aleksandra WERNER: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, aleksandra.werner@polsl.pl

Adam DUSZEŃKO: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, adam.duszenko@polsl.pl.