Data Community

SQL Day

PLATINUM SPONSOR

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

STRATEGIC PARTNER

Microsoft

GOLD SPONSORS

CLOUDS ON MARS

Future Processing

SPYROSOFT

SILVER SPONSOR

dbWatch DATABASE CONTROL

elitmind think bright

VEEAM

BRONZE SPONSOR

Największa księgarnia IT w Polsce

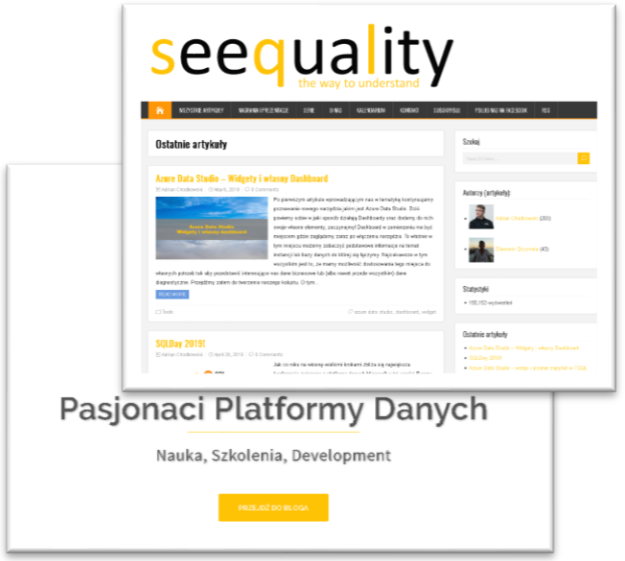
NOVATECH www.novatech.com.pl

Performance Tuning dla specjalistów Business Intelligence

Adrian Chodkowski

O mnie

- **Adrian Chodkowski**
- Konsultant Business Intelligence
- Specjalizacja: Platforma danych Microsoft
- Trener i wykładowca
- MCP, MCSA, MCSE:BI, MCSE: Data Management and Analytics
- **seequality.net**
- Adrian.Chodkowski@outlook.com
- @Twitter: Adrian_SQL
- LinkedIn: http://tinyurl.com/adrian-sql



Agenda

Parallelism

Parameter Sniffing
+
Memory Grant

Columnstore
+
partitioning

Minimal logging

Batchsize

PARALLELISM

Parallelism

DEMO

Parallelism

- Zapytania w SQL Server mogą być wykonywane wielowątkowo
- Dwa podstawowe ustawienia dotyczące wielowątkowści:
  - **Max Degree Of Parallelism**, **Cost Treshold of Parallelism**
- W systemach hurtowni danych kluczowe jest to aby „duże” zapytania były wielowątkowe
- Istnieje wiele przypadków powodujących, że plan jest jednowątkowy m.in:
  - Affinity Mask,
  - Zapytanie poniżej progu lub MAXDOP=1
  - Backward Scan
  - Użycie skalarnych funkcji użytkownika (patrz: **Scalar function inlining**)
  - Użycie obiektów systemowych
  - Operatory niewspierające równoległości

```
OPTION(QUERYTRACEON 8649)

OPTION(USE HINT(
'ENABLE_PARALLEL_PLAN_PREFERENCE'))
```

Szukaj na planie **NonParallelPlanReason**

- MaxDopSetToOne
- EstimatedDOPIsOne
- NoParallelFastForwardCursor
- NoParallelCursorFetchByBookmark
- ParallelismDisabledByTraceFlag
- NoParallelCreateIndexInNonEnterpriseEdition
- NoParallelPlansInDesktopOrExpressEdition
- CLRUserDefinedFunctionRequiresDataAccess
- TSQLUserDefinedFunctionsNotParallelizable
- DMLQueryReturnsOutputToClient
- MixedSerialAndParallelOnlineIndexBuildNotSupported
- CouldNotGenerateValidParallelPlan
- NoParallelForMemoryOptimizedTables

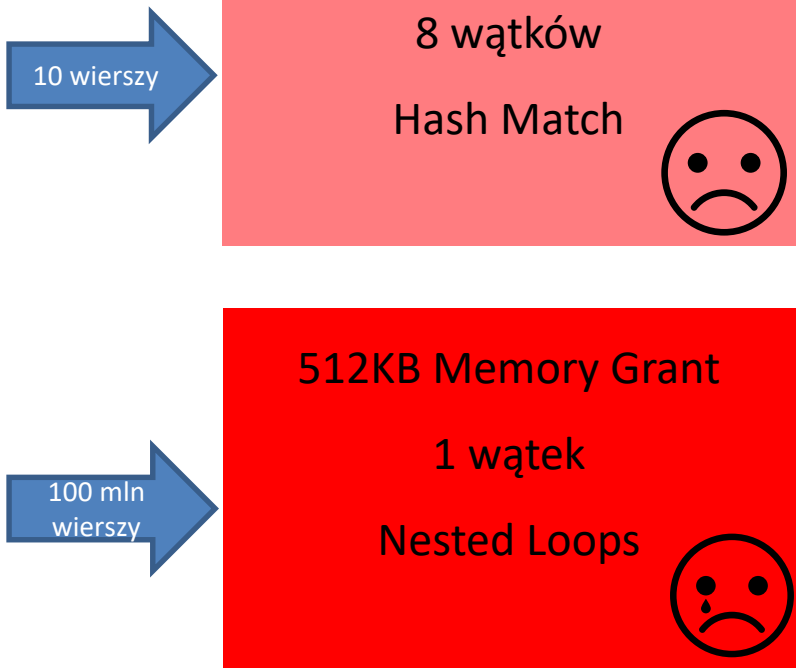
PARAMETER SNIFFING

Parameter Sniffing

DEMO

Parameter Sniffing

- Standardowo plan zapytania jest umieszczany w cache po to aby nie trzeba było go generować na nowo przy następnym uruchomieniu
  (oszczędzając przy tym czas potrzebny na rekompilacje)
- W cache zapisywana jest wersja z danego wykonania z operatorami odpowiednimi do statystyk z momentu wykonania
- W przypadku gdy np. procedura w zależności od parametru działa na zdecydowanie różnych zbiorach pod kątem liczby wierszy oznacza to, że jest ona **parameter sensitive**, a więc pobieranie planu z cache może być problematyczne
- Operatory mogą być dobrane w nieodpowiedni sposób szczególnie jeśli chodzi o **operatory złączenia** i operatory wymagające przydziału pamięci (**memory grant**)



Parameter Sniffing

- Wyłączenie Parameter Sniffing w Database Scoped configurations

```
ALTER DATABASE SCOPED CONFIGURATION SET PARAMETER_SNIFFING = Off;
```

- Rekompilacja planów:

```
OPTION (RECOMPILE)  WITH (RECOMPILE)  sp_recompile
```

- W niektórych przypadkach pomocne może być:
- Globalna lub lokalna zmiana minimalnego przydziału pamięci:

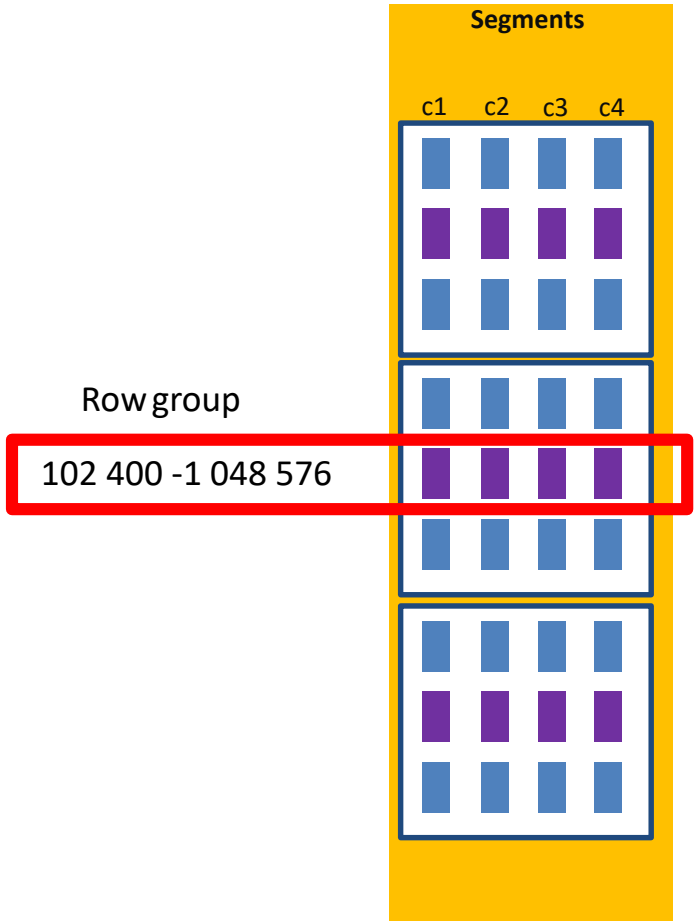
```
OPTION (min_grant_percent=15) sp_configure 'min_memory_per_query'  Resource Governor
```

- Adaptive Query Processing w tym:
  - Memory Grant Feedback
  - Adaptive Join

COLUMNSTORE + PARTITIONING

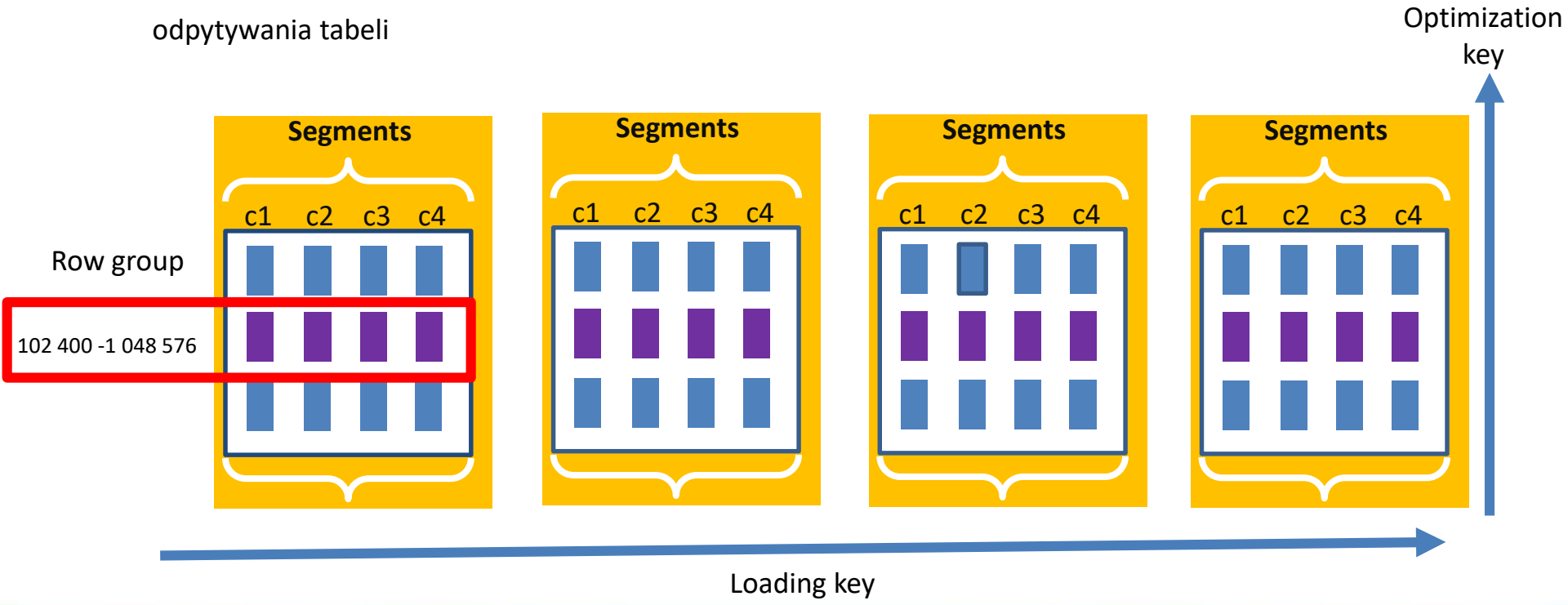
Columnstore index

- Columnstore składa się z grup wierszy
- Grupa wierszy może mieć **od 102 400 do 1 048 576** wierszy
- Im większa grupa tym lepiej
- Mniejsze ilości wierszy nie tworzą grupy
- Dane kompresowane są w ramach segmentu czyli kolumny wewnątrz grupy wierszy
- Przy odczytywaniu danych z indeksu kolumnowego może nastąpić **Segment Elemination** czyli wybranie na podstawie metadanych tych segmentów, które zawierają pożądane dane
- Ułożenie danych w indeksie według najczęściej odpytywanej kolumny znacznie usprawni powyższy proces



Partitioned Columnstore index

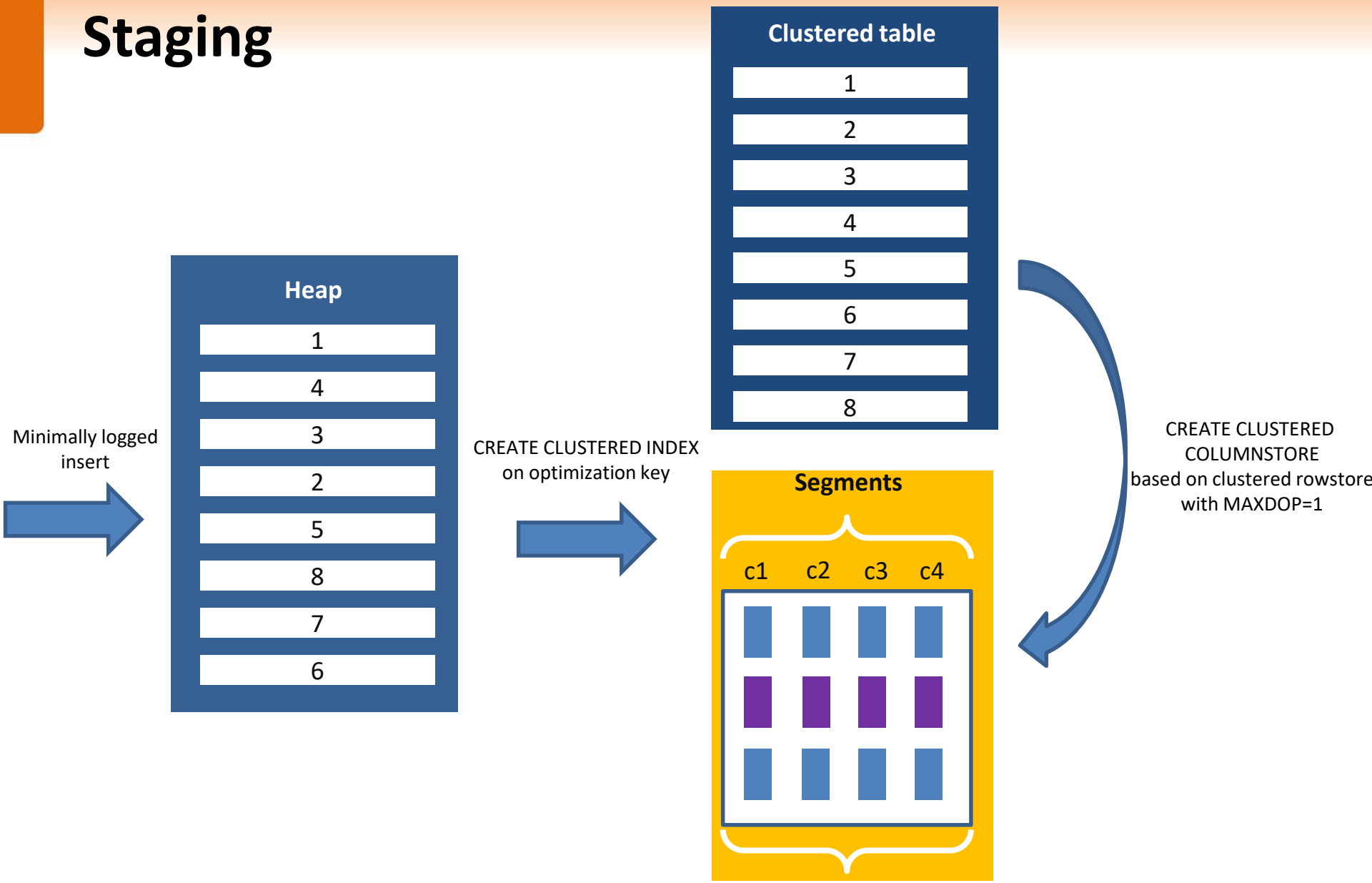
- Partycje na indeksie kolumnowym dodają dodatkowy podział
- Indeks kolumnowy per partycja może być postrzegany jako osobny indeks
- **Klucz partycji (Loading Key)** = klucz ładowania
- **Klucz sortowania tabeli (Optimization Key)** = najczęściej wybierany klucz odpytywania tabeli



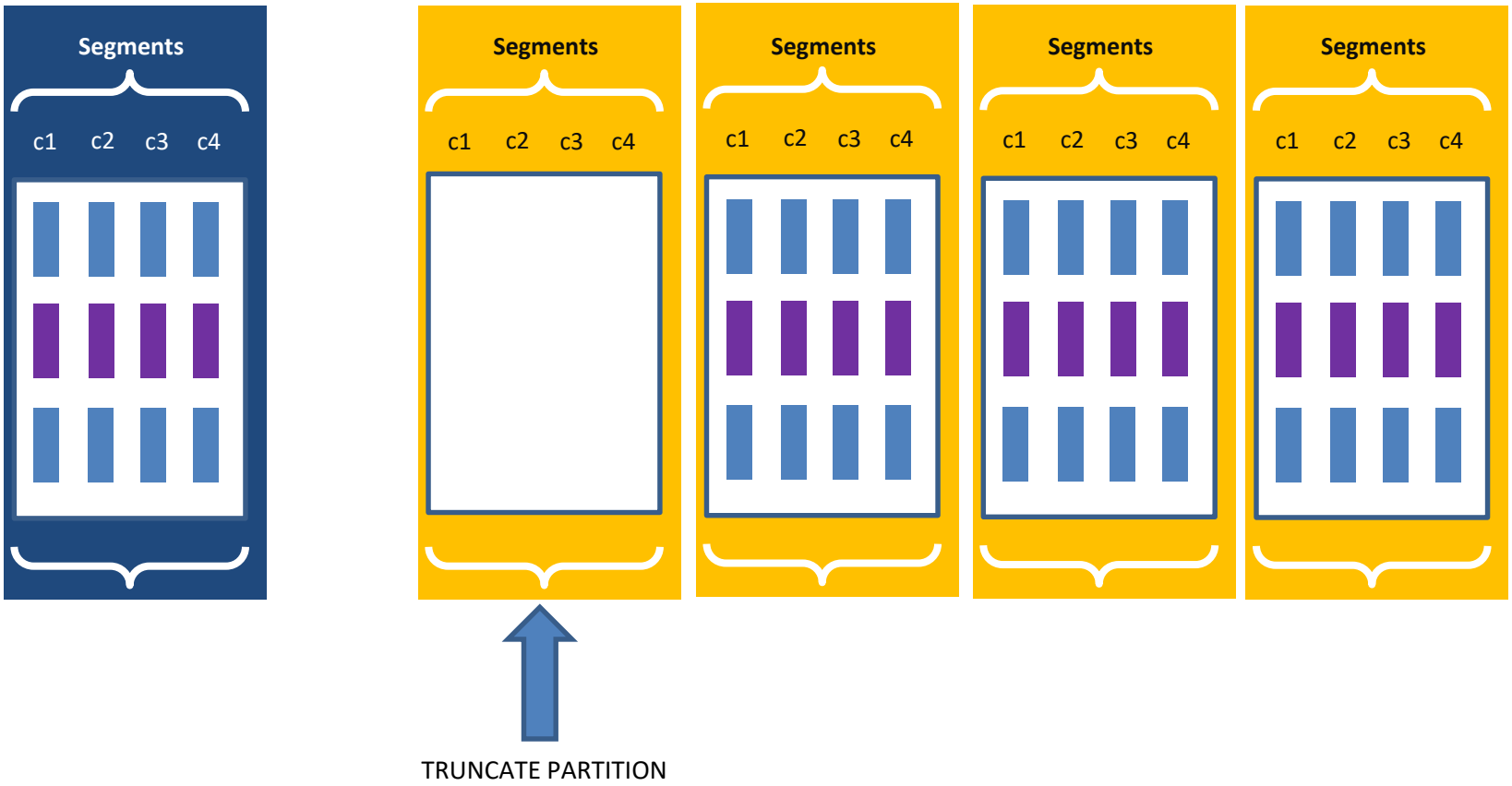
COLUMNSTORE + PARTITIONING

DEMO 1

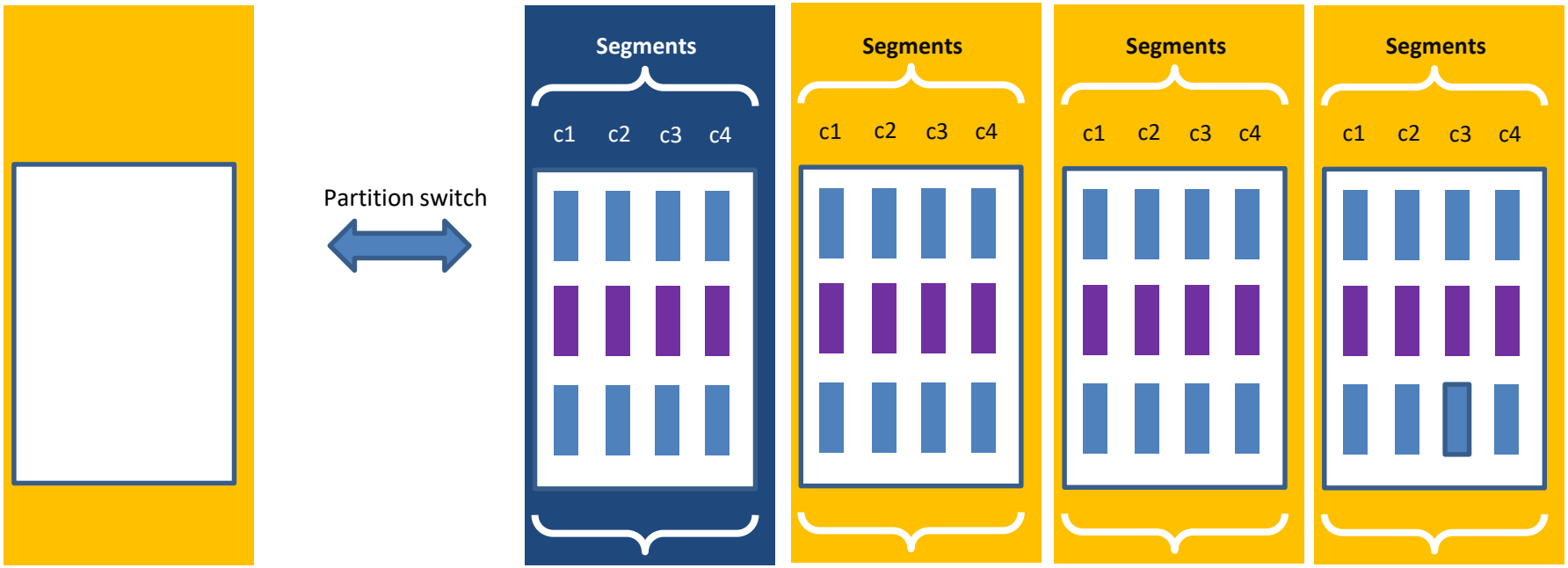
Staging


Clustered table
1
2
3
4
5
6
7
8
Heap
1
4
3
2
5
8
7
6
Minimally logged insert
CREATE CLUSTERED INDEX on optimization key
Segments
c1
c2
c3
c4
CREATE CLUSTERED COLUMNSTORE based on clustered rowstore with MAXDOP=1

TRUNCATE PARTITION



PARTITION SWITCH



COLUMNSTORE + PARTITIONING

DEMO 2

MINIMAL LOGGING

Minimal logging

- Minimalne logowanie nie oznacza braku logowania w ogóle.
- Gdy coś jest minimalnie logowane SQL Server nie loguje w dzienniku poszczególnych wierszy, a jedynie informacje o zaalokowaniu stron i ekstentów.
- Dla przykładu jeśli na stronie mieści się 100 wierszy to w logu znajdzie się informacja o tej jednej stronie zamiast 100 wierszach.
- W momencie gdy transakcja BULK IMPORT zostanie potwierdzona (np. gdy liczba wierszy osiągnie BATCHSIZE) SQL Server zrzuca wszystkie strony danych na dysk.
- Operacje minimalnie logowane nie wspierają przywracania do punktu w czasie.
- W przypadku implementacji operacji minimalnie logowanych testujmy, bo zachowanie SQL Servera jest zależne od bardzo wielu czynników!

Sposoby:
- BCP
- BULK INSERT
- INSERT … SELECT
- SELECT INTO
- ETL Tools
- …

https://blogs.msdn.microsoft.com/sqlserverstorageengine/2008/02/04/bulk-import-optimizations-minimal-logging/

Recovery model

- Niektóre operacje w SQL Server powodują logowanie informacji inaczej niż zazwyczaj
- Poziom logowania w pierwszej mierze zależy od ustawienia Recovery Model
  - FULL
    - Umożliwia odtworzenie bazy do punktu w czasie,
    - Dane nie są tracone w przypadku uszkodzenia pliku danych,
    - Wymaga kopii zapasowych dziennika transakcji.
  - BULK LOGGED
    - Jak tryb FULL umożliwiający minimalne logowanie operacji BULK,
    - Tymczasowe rozwiązanie dla baz z FULL gdy potrzeba załadować duże ilości danych,
  - SIMPLE
    - Dziennik jest nadpisywany po zatwierdzeniu transakcji,
    - Brak kopii dziennika, dane można odtworzyć do punktu ostatniego backupu plików danych.

Recovery model

- Niektóre operacje w SQL Server powodują logowanie informacji inaczej niż zazwyczaj
- Poziom logowania w pierwszej mierze zależy od ustawienia Recovery Model
  - FULL
    - Umożliwia odtworzenie bazy do punktu w czasie,
    - Dane nie są tracone w przypadku uszkodzenia pliku danych,
    - Wymaga kopii zapasowych dziennika transakcji.
  - BULK LOGGED
    - Jak tryb FULL umożliwiający minimalne logowanie operacji BULK,
    - Tymczasowe rozwiązanie dla baz z FULL gdy potrzeba załadować duże ilości danych,
  - SIMPLE
    - Dziennik jest nadpisywany po zatwierdzeniu transakcji,
    - Brak kopii dziennika, dane można odtworzyć do punktu ostatniego backupu plików danych.

Recovery model

TESTUJ!

Sp_tableoption ' table lock on bulk load'

Table Indexes	Rows in table	Hints	With or Without TF 610	Concurrent possible
Heap	Any	TABLOCK	Minimal	Yes
Heap	Any	None	Full	Yes
Heap + Index	Any	TABLOCK	Depends (3)	No
Cluster	Empty	TABLOCK, ORDER (1)	Minimal	No
Cluster	Empty	None	Minimal	Yes (2)
Cluster	Any	None	Minimal	Yes (2)
Cluster	Any	TABLOCK	Minimal	No
Cluster + Index	Any	None	Depends (3)	Yes (2)
Cluster + Index	Any	TABLOCK	Depends (3)	No

(1) Dla INSERT … SELECT wiersze muszą być dostarczone w porządku indeksu. Dla BULK INSERT hint ORDER musi być podany.
(2) Równoległe wstawianie możliwe w określonych warunkach."Bulk Loading with the Indexes in Place". Wiersze wstawiane do na nowo zaalokowanych stron są minimalnie logowane.
(3) Zależy od planu wybranego przez optymalizator, indeks niezgrupowany na tabeli może być w pełni lub minimalnie logowany.

https://blogs.msdn.microsoft.com/sql_server_team/sql-server-2016-minimal-logging-and-impact-of-the-batchsize-in-bulk-load-operations/

Minimal logging

DEMO

BATCHSIZE

BATCHSIZE – problemy

BATCH

BATCHSIZE – problemy

BATCH

Alokacja ekstentu

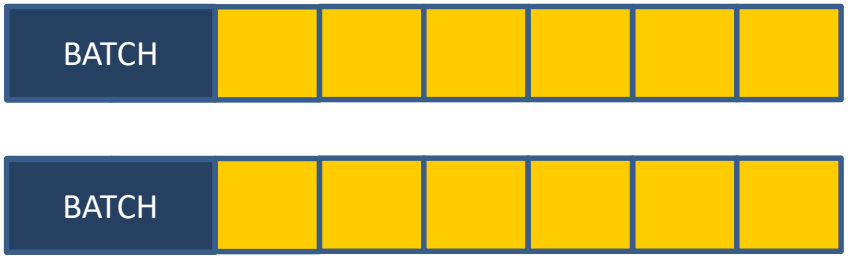
BATCHSIZE – problemy

Pojedynczy batch zajmuje część zaalokowanego ekstentu

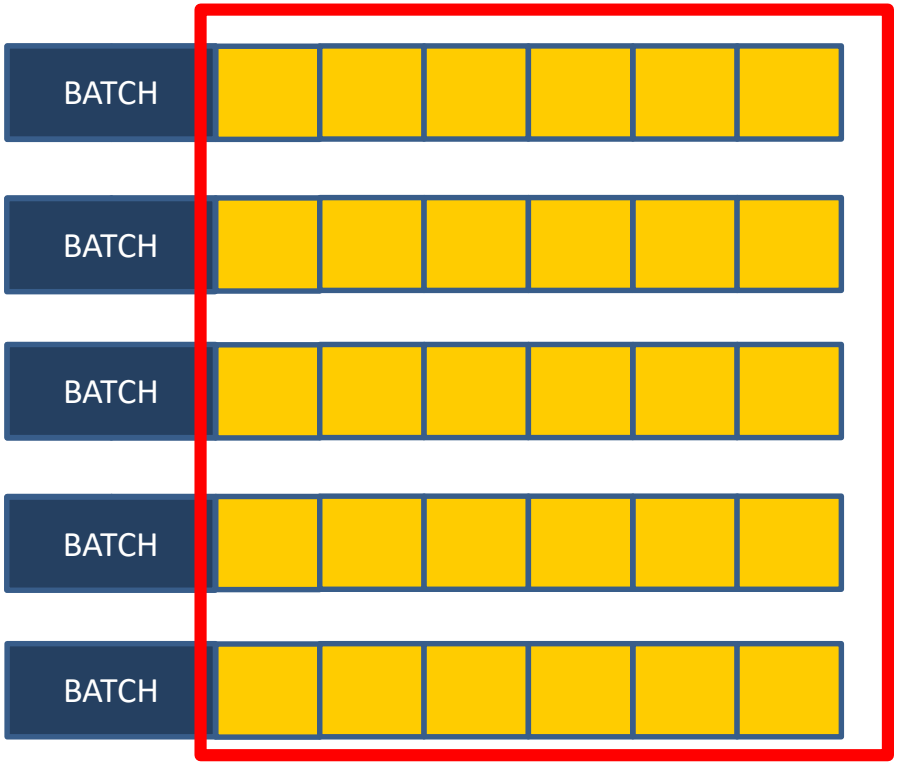
BATCHSIZE – problemy

Alokacja ekstentu

Recovery model

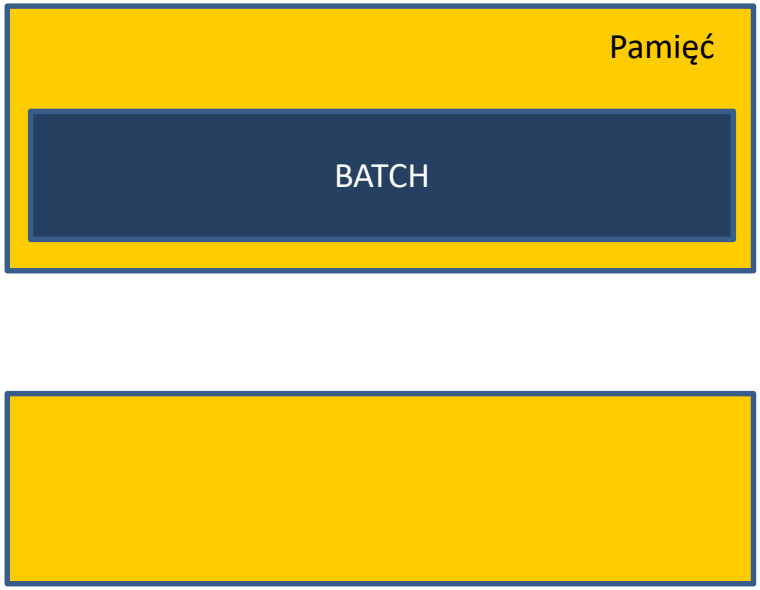


BATCHSIZE – problemy

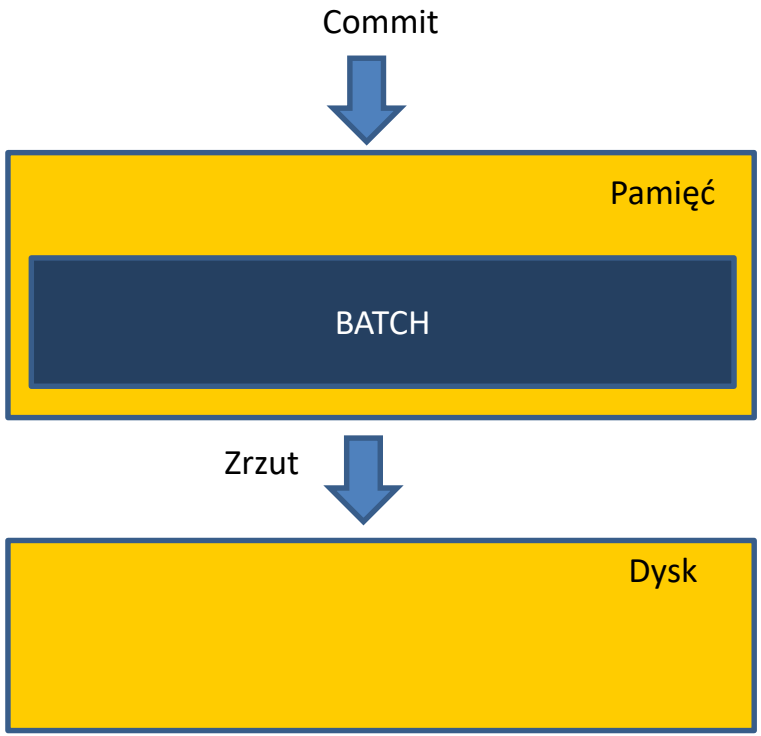


Co z tym miejscem?

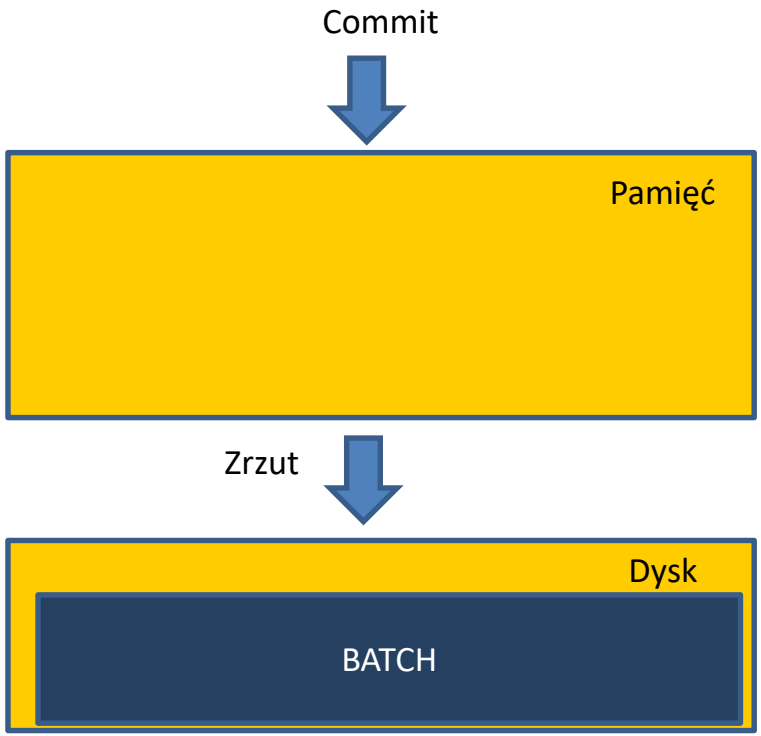
BATCHSIZE – problemy



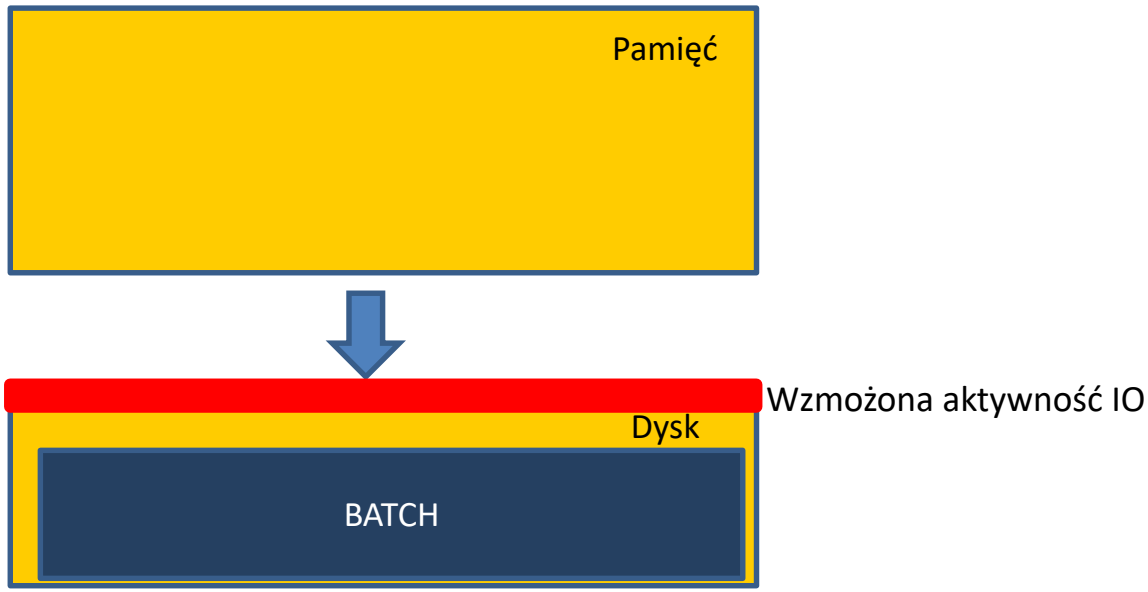
BATCHSIZE – problemy



BATCHSIZE – problemy



Recovery model



BATCHSIZE

- Dla sterty zacznijmy z BATCHSIZE=0, dla Columnstore z wartością od 102400 do BATCHSIZE=1048576
- Przy indeksie nieposortowanym źródle dla indeksu klastrowanego upewnijmy się, że nie mamy zrzutów na dysk i dostosujmy na tej podstawie BATCHSIZE (przy założeniu, że kolumny o zmiennej szerokości są zajęte w połowie)
- Determinuje w jakich porcjach mają być wrzucane dane podczas BULK IMPORT
- Przy każdym batchu alokowane są nowe extenty - nawet jak w starym jest jeszcze miejsce),
- Przy każdym batchu występuje commit, który zrzuca dane na dysk,
- W przypadku problemów z dyskiem i miejscem Tiger Team zaleca obliczenie BATCHSIZE jako

64KB/(średnia szerokość wiersza) * N gdzie N to liczba extentów od 1 do 64

https://blogs.msdn.microsoft.com/sql_server_team/sql-server-2016-minimal-logging-and-impact-of-the-batchsize-in-bulk-load-operations/

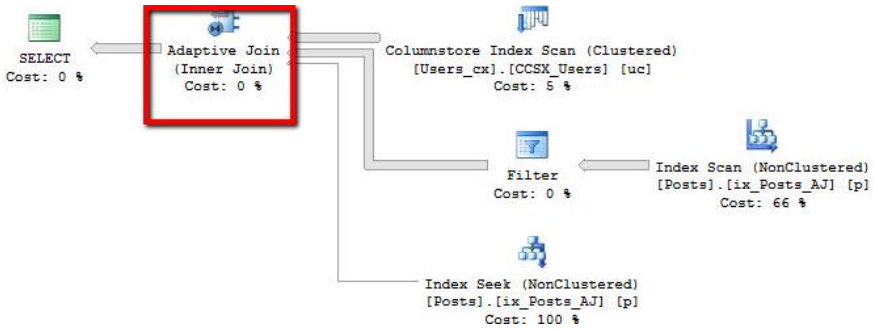
BATCHSIZE

DEMO

Inne przydatne mechanizmy

- Niemal wszystkie dobre praktyki związane z pisaniem SELECT mają uzasadnienie w zapytaniach analitycznych
- Warto pamiętać o następujących mechanizmach i obiektach:
  - Indeksy filtrowane,
  - Widoki zmaterializowane,
  - Tryb batch,
  - Funkcje okna,
  - APPLY,
  - Adaptive Join,
  - Incremental Statistics
  - Tabele prekalkulowane
  - Power BI Aggregations (i podobne)

```
CREATE NONCLUSTERED INDEX
[NCI_Filtered10]
ON [Sales].[InvoiceLines]
(
PackageTypeID ASC,
StockItemID ASC,
[InvoiceID] ASC
)
INCLUDE ( [Quantity],
[TaxRate],
[TaxAmount])
WHERE PackageTypeID=10
```




```
UPDATE STATISTICS
dbo.BigFactTable(idx_date)
WITH RESAMPLE ON PARTITIONS(8);
```

Dziękuję!

adrian.chodkowski@outlook.com

Performance Tuning dla specjalistów
Business Intelligence

Data Community

SQL Day

PLATINUM SPONSOR

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

STRATEGIC PARTNER

Microsoft

GOLD SPONSORS

CLOUDS ON MARS

Future Processing

SPYROSOFT

SILVER SPONSOR

dbWatch DATABASE CONTROL

elitmind think bright

VEEAM

BRONZE SPONSOR

Największa księgarnia IT w Polsce

NOVATECH www.novatech.com.pl