

Ścieżkami Db2

POLISH DB2 USERS GROUP



Magazyn członków stowarzyszenia Polska Grupa Użytkowników Db2

**ŚWIĘTUJEMY 5 LAT
STOWARZYSZENIA
PDUG**



W tym numerze:

5 lat Polskiej Grupy Użytkowników Db2

*Algorytmy tworzenia indeksów dla złączeń
wewnętrznych tabel (JOIN) w Db2*

Testowanie Bazy danych Db2 w wersji 12

Kopiowanie Filesystemów dla Db2

ROS - Read on Standby

Techniques to move Db2 data from PROD to TEST



POLISH DB2 USERS GROUP

Polska Grupa Użytkowników Db2 jest stowarzyszeniem założonym w 2013 roku i jest jedną z najszybciej rozwijających się grup regionalnych w Europie. O dynamice rozwoju PDUG świadczy chociażby fakt, że w chwili obecnej stowarzyszenie liczy sobie ponad 100 członków i ciągle przybywają nowi. Aby Stowarzyszenie mogło się rozwijać i przetrwać następne kilkadziesiąt lat potrzebujemy pozytywnej energii.

Misja: wspieranie i wzmacnianie wspólnoty usług informatycznych, poprzez zapewnienie najwyższej jakości kształcenia i usług mających na celu promowanie i efektywne wykorzystanie Db2

PDUG podzielony jest na dwa panele z/OS i LUW, skupiając tym samym większość platform, na których jest dostępna Db2. Dzięki zaangażowaniu wolontariuszy podczas konferencji organizowanych dwa razy w roku członkowie obu tracków mają szansę wysłuchać prezentacji doświadczonych międzynarodowych specjalistów. Możemy się poszczycić, że niejednokrotnie gościli u nas sławy ze świata Db2, jak chociażby John Campbell, Steve Thomas czy Cristian Molaro. Warty podkreślenia jest fakt, że konferencje PDUG są bezpłatne dla członków, którzy zgodnie z własną deklaracją i statutem stowarzyszenia regularnie opłacają składki członkowskie, co dla organizacji non-profit jest jedną z kluczowych form finansowania swojej działalności. Istotne jest to, że na niektórych konferencjach nasi członkowie mogą skorzystać z darmowych certyfikacji dla Db2.

Nasze stowarzyszenie od samego początku współpracuje

z organizacją non-profit IDUG, która jest liderem na rynku konferencji Db2 w Europie. Nasi członkowie regularnie są prelegentami na corocznych eventach IDUG. Dzięki czemu jesteśmy rozpoznawalni nie tylko w Europie, ale także na świecie. Zapraszamy wszystkich do zapoznania się z planami jesiennego spotkania IDUG. Ponadto, w tym roku również zostanie rozlosowane zaproszenie uczestnictwa w Międzynarodowej Konferencji IDUG Db2 EMEA Malta 2018. Zachęcamy do wzięcia udziału w losowaniu.

Nasi koledzy są także posiadaczami tytułu IBM Champion. W roku 2017 wyróżnieni zostali Jacek Surma, Mateusz Książek, Mariusz Żyśk, Jacek Rafalak, Rafał Gulbas, Paweł Hryb, Tomasz Kraszucki, Mariusz Koczkar. Gratulujemy wszystkim nagrodzonym.

Zapraszamy wszystkich naszych członków do czynnego udziału w pracach stowarzyszenia, jak i współdecydowaniu o przyszłości naszej dynamicznie rozwijającej się grupy.

Bieżące informacje dotyczące prac stowarzyszenia są umieszczane w newsletterze na naszej stronie internetowej www.pdug.pl. Jesteśmy także na Facebooku oraz Twitterze. Cały czas pracujemy nad rozwojem i atrakcyjnością Polskiej Grupy Użytkowników Db2. Zachęcamy wszystkich do pomocy, bo jak wiadomo największa siła jest w grupie. Dziękujemy wszystkim, którzy angażują się w rozwój naszego stowarzyszenia.

Zespół PDUG.

5 LAT

POLSKIEJ GRUPY UŻYTKOWNIKÓW DB2

Mariusz Zyśk - Laurens Coster Sp. z o.o.



Stowarzyszenie Polska Grupa Użytkowników Db2 istnieje już od pięciu lat! Jako, że sami mamy kłopot z pamięcią, lubimy mówić, że to było po prostu bardzo dawno. Tak dawno, że w PDUG nie było jeszcze ścieżki LUW. No ale wróćmy do początku...

Jest 2012 rok i paru fanów bazy Db2, pośród nich Jacek, Michał, Roman, Tomek, Emil i Agata przy wsparciu jak się okazuje niezastąpionej Gosi wpadają na śmiały pomysł założenia w Polsce stowarzyszenia promującego tę bazę danych IBM Db2. Od



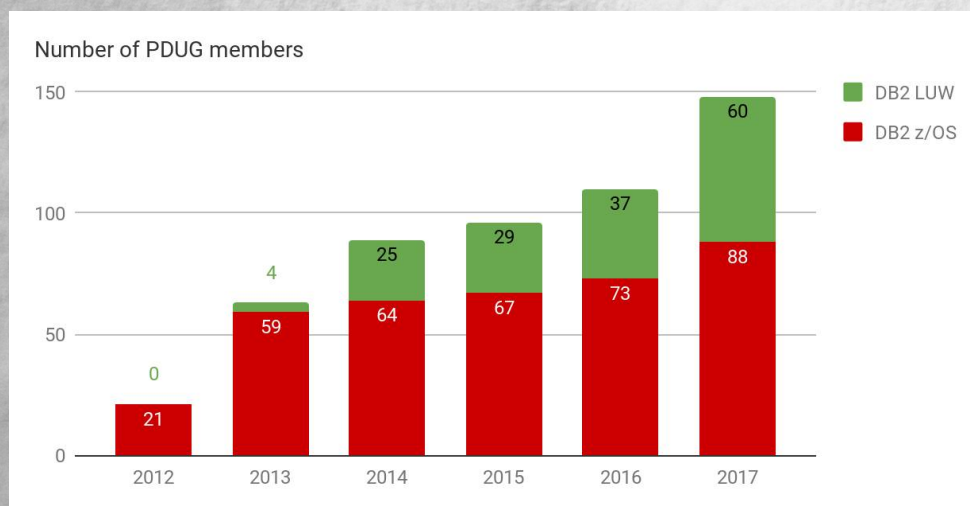
słów do czynów poszło szybko i jeszcze tego samego roku odbyło się spotkanie założycielskie.

Od tamtej pory minęło już 5 lat, a stowarzyszenie rozwija się wciąż bardzo dynamicznie między innymi dzięki zaangażowaniu wszystkich wolontariuszy, a także rosnącej świadomości potencjalnych użytkowników o istnieniu w ogóle bazy Db2. W paru akapitach przedstawimy co się wydarzyło przez ostatnie 5 lat w stowarzyszeniu.

Za sobą mamy już 11 konferencji, a przed nami 12, która odbędzie się 19-20 marca 2018 w Krakowie. Konferencje Polskiej Grupy Użytkowników Db2 odbywają się co pół roku i są podzielone na dwa równoległe panele: z/OS oraz LUW - choć nie zawsze tak było! Pierwsze dwie konferencje (w roku 2013) odbyły się tylko dla platformy z/OS, a dopiero o 2014 roku uruchomiona została ścieżka Linux Unix Windows.

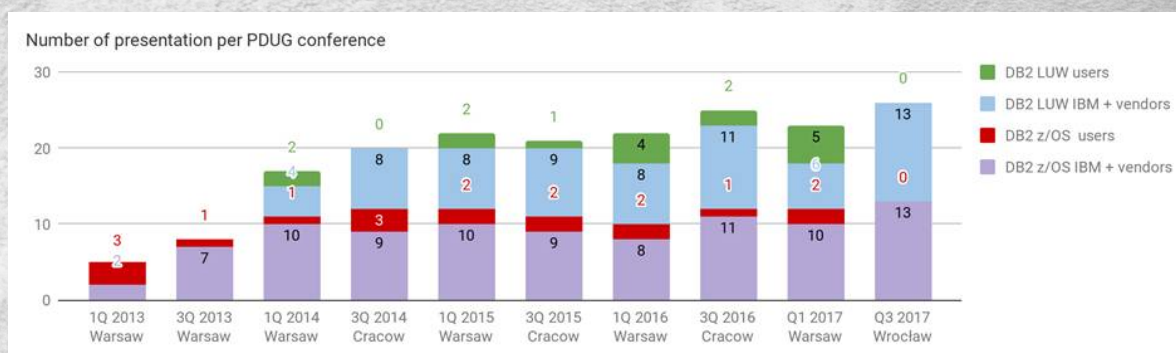
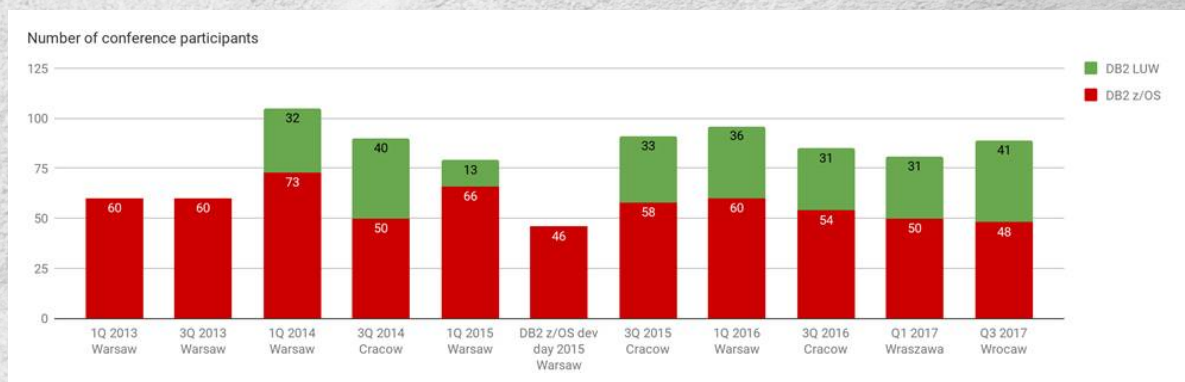
To co wyżej opisałem doskonale oddają diagramy, które znajdziesz poniżej drogi czytelniku. Pierwszy z nich opowiada historię w liczbach. Historię o tym w jaki sposób i jak szybko przekonał się i wciąż przekonujemy do bycia częścią tego

stowarzyszenia użytkowników dwóch platform!



Drugi z kolei diagram opisuje nam jak wyglądała sytuacja z uczestnikami poszczególnych konferencji. Dzięki tym liczbom można uświadomić sobie w jakim tempie PDUG urosł do rangi wielkiego stowarzyszenia, które zdobywa międzynarodowe nagrody (a o nich za chwilę)!

Jest jeszcze jedna super ważna rzecz, która nie może zostać pominięta! Czy wiesz, że w przeciągu pięciu lat trwania PDUG, na jedenastu konferencjach (i jednym dev day) zostało wygłoszonych **189 prezentacji przez ponad 100 speakerów z całego świata?** Te liczby powalają (przynajmniej nas!). Zerknijmy na diagram, żeby zobaczyć jak się rozkładała liczba prezentacji podczas wszystkich konferencji!



I jeszcze tylko, krótka lista wybranych speakerów, którzy w naszej opinii definiują światowe sławy DB2

Pavel Sustr	- IBM	Rafal Gulbas	- Accenture
Steen Rasmussen	- CA Technologies	Les King	- IBM
Philippe Dubost	- CA Technologies	John Campbell	- IBM
Chistiano Molaro	- Consultant	Kamil Kuduk	- IBM
Steve Thomas	- CA Technologies	Tomasz Kraszucki	- Asseco
Gareth Johnes	- IBM	Piotr Tkaczyk	- ASPartner
Ramon Menendez	- BMC	Jacek Surma	- PKO Bank Polski
Jacek Rafalak	- CA Technologies	Wen Jie Zhu	- IBM
Jan Nelken	- IBM	Gerard Kaduczek	- Asseco
Marcin Marczewski	- IBM	Zeljen Stanic	- CA Technologies
Nigel Slinger	- IBM	Michal Bialecki	- IBM
Phil Grainger	- IBM	Bart Steegmans	- IBM
Saghi Amirsoleymani	- RocketSoftware	Matthias Tschafler	- RocketSoftware
Paweł Hryb	- PKO Bank Polski	Michael Dewert	- IBM

Co do rozpoznawalności, to PDUG jako stowarzyszenie współpracuje ze stowarzyszeniem IDUG, które jest liderem na rynku konferencji Db2 na całym świecie. Nasi członkowie regularnie zostają prelegentami tych międzynarodowych konferencji - za co należą się im szczególne gratulacje i brawa. Oto niektórzy ze śmiałków, których nazwisko już na zawsze wyryło się historii Db2:

Tomasz Kraszucki

Pavel Hryb

Jacek Surma

Adam Pesta

Michał Białecki

Jacek Rafałak

Rafał Gulbas

Mateusz Książek

Między innymi dzięki nim jesteśmy rozpoznawalni jako rzetelna instytucja szkoląca świetnych specjalistów Db2 oraz zdobywamy międzynarodowe nagrody, a wśród nich

Best New User Group in 2013



Best New User Group in 2013



Recognition Award 2017



Wciąż to nie wszystko czy możemy się pochwalić. Rozpoznawalność budujemy nie tylko dzięki organizowanym eventom, prezentacjom naszych użytkowników na lokalnych i międzynarodowych konferencjach, ale również dzięki całokształtowi pracy niektórych z naszych członków. Takie szczególne zaangażowanie w rozwój społeczności i szerzenia wiedzy w tech-

nologiach IBM jest nagradzane właśnie przez tę firmę specjalnymi nagrodami. Nazwa takiej nagrody to IBM Champion i wśród naszej społeczności mamy co roku parę osób, które takie nagrody otrzymują. Rzućmy okiem na infografikę przedstawiającą naszych wspaniałych Championów.

Pięciolecie czas zacząć, czas się bawić, celebrować, świętować! Wspólnie jako ludzie, a nie tylko jako członkowie stowarzyszenia. Tak naprawdę PDUG to coś więcej niż grupa geeków Db2. PDUG to wspólnota wspaniałych ludzi, którzy się przyjaźnią, mogą na siebie liczyć, a także pomagają sobie i wspierają się nawzajem. Spotykamy się nie tylko na konferencjach i integrujemy się nie tylko na wieczorkach integracyjnych, choć te wydarzenia także uzupełniają fenomen PDUG'a - najlepszej grupy użytkowników Db2 na świecie...

Dodatkowo warto wspomnieć, że bycie członkiem naszego stowarzyszenia to nieskończone pasmo zalet, a pośród nich można wymienić:

- pierwszeństwo przy rejestracji na konferencje (ograniczona liczba miejsc)
- dostęp do prezentacji konferencyjnych (także z poprzednich konferencji)
- darmowy udział w wieczorku
- pakiet startowy i materiały konferencyjne
- udział w losowaniach sponsorów (vouchery na konferencje IDUG, certyfikacje, szkolenia) - ok 40 osób zdobyło certyfikaty z Db2 a ok 10 osób pojechało na konferencje IDUG w poprzednich latach dzięki voucherom ufundowanych dla członków PDUG
- stała zniżka na konferencje IDUG dla członków PDUG

Jakie mamy plany na przyszłość?

W dwóch słowach - rozwijać się. A przy użyciu większego zasobu słów to: będziemy brnąć do przodu, zamierzamy powiększać naszą społeczność, pozyskiwać nowych użytkowników, kontaktować się i nawiązywać współpracę z firmami, które pracują przy Db2. Utrzymamy również obecną formę stowarzyszenia, czyli 2 konferencje rocznie.

Do tego dokładamy kontynuację naszego (miejmy nadzieję, że już corocznego) periodyku, który w tym roku, który uka-

PDUG in the world

PDUG

- » PDUG is noticed worldwide!
- » our members share knowledge on international conferences like IDUG
- » our members gaining IBM Champion title
- » our members getting recognition awards



IBM Champion
2015 2016
2017 2018



IBM Champion
2016 2017 2018



IBM Champion
2017



IBM Champion
2013 2014 2015
2016 2017 2018

POLISH DB2 USERS GROUP

PDUG in the world

PDUG

- » and more Champions.. in PDUG



IBM Champion
2014 2017
Customer Recognition 2016



IBM Champion
2014



IBM Champion
2014



IBM Champion
2014 2017 2018



IBM Champion
2014 2017 2018



IBM Champion
2014 2017



IBM Champion
2018

POLISH DB2 USERS GROUP

zał się dzięki Jarosławowi Pawelcowi i Bartoszowi Faszewskiemu i autorom wspaniałych technicznych artykułów. Czwarte wydanie tego czasopisma właśnie trzymasz w rękach. Czy będziesz trzymał piąte za rok to może zależeć również od Ciebie! Jeśli masz pomysł na jakiś artykuł do periodyka

i chcesz by Twoje nazwisko zaczęło być bardziej rozpoznawalne w środowisku Db2 zgłoś się do nas! Jeśli masz pomysł i chciałbyś pomóc w innych aspektach PDUG - czekamy!

Bieżące informacje dotyczące stowarzyszenia są umieszczane na naszej stronie internetowej www.pdug.pl. Jesteśmy także obecni na facebooku, linkedin oraz twitterze. Zachęcamy do śledzenia nas w mediach społecznościowych.

Dziękujemy wszystkim, którzy angażują się w rozwój naszego stowarzyszenia.

Zarząd PDUG



Algorytmy tworzenia indeksów dla złączeń wewnętrznych tabel (JOIN) w DB2

Jacek Surma - PKO Bank Polski S.A.



Optymalizator DB2 analizując koszt zapytań typu JOIN, w którym łączymy dwie lub więcej tabel, ma do wyboru 3 rodzaje ścieżek dostępu do danych:

Nested loop join (METHOD=1)
Merge scan join (METHOD=2)
Hybrid join (METHOD=4)

W tabeli PLAN_TABLE w kolumnie METHOD możemy odnaleźć informację, która metoda dostępu do danych została wybrana. Na podstawie dokumentacji mamy:

METHOD - A number (0, 1, 2, 3, or 4) that indicates the join method used for the step:

- 0 First table accessed, continuation of previous table accessed, or not used.
- 1 *Nested loop* join. For each row of the present composite table, matching rows of a new table are found and joined.
- 2 *Merge scan* join. The present composite table and the new table are scanned in the order of the join columns, and matching rows are joined.
- 3 Sorts needed by ORDER BY, GROUP BY, SELECT DISTINCT, UNION, a quantified predicate, or an IN predicate. This step does not access a new table.
- 4 *Hybrid* join. The current composite table is scanned in the order of the join-column rows of the new table. The new table is accessed using list prefetch.

Rys: Tab0

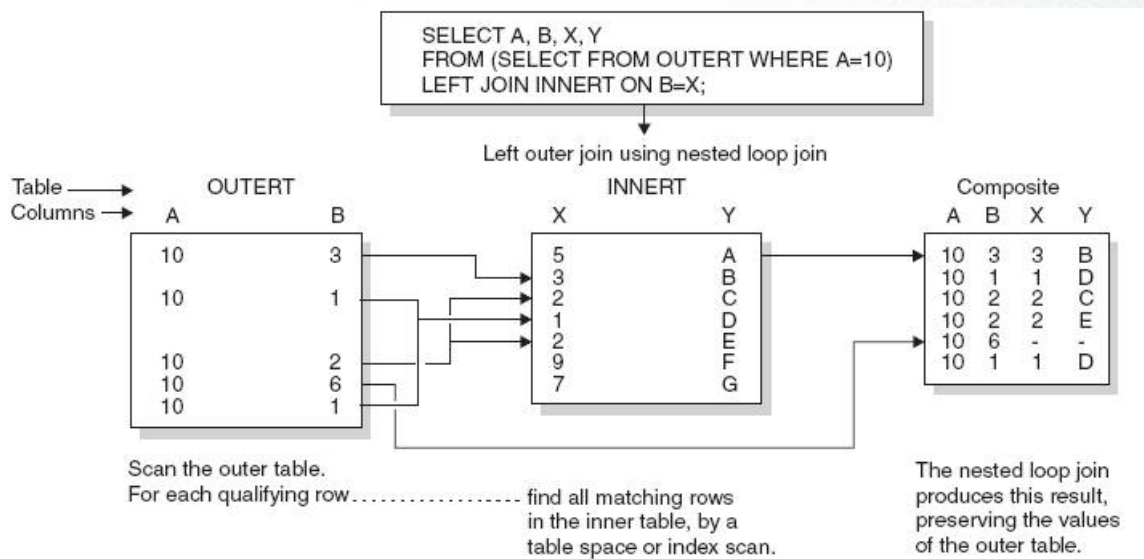
Przy tego typu złączeniach często obserwujemy problemy wydajnościowe objawiające się długim czasem wykonania oraz zwiększonym zużyciem CPU.

Najczęstszym powodem takiej sytuacji jest brak odpowiednio dopasowanych indeksów.

W artykule przedstawię zasady, którymi należy się kierować przy indeksowaniu każdej z tych trzech metod dostępu.

Zacznijmy od pierwszej najbardziej popularnej i wydajnej metody.

Nested loop join (METHOD=1)



Rys: METHOD1

W pierwszym kroku DB2 musi zdecydować, która tabela będzie zewnętrzna – **Outer Table** a która będzie wewnętrzna **Inner Table**.

Podczas operacji JOIN tabela zewnętrzna (**Outer**) jest zawsze czytana jako pierwsza a tabela wewnętrzna (**Inner**) jako druga.

Dla tabeli zewnętrznej (**Outer**) zakładamy index na wszystkich kolumnach z predykatów lokalnych tej tabeli. Tylko takie predykaty mogą być dla niej Matching, czyli definiować jeden kawałek indeksu do odczytu.

Nie zakładamy index-u na kolumnie ze złączenia (**join predicate**) dla tabeli **Outer** (chyba, że chcemy mieć dostęp typu Index Only).

Kolejność kolumn w indeksie dla predykatów połączonych operatorem AND (Boolean Term) ze zdania WHERE powinna być następująca:

Na pierwszym miejscu powinny znaleźć się w dowolnej kolejności wszystkie kolumny z predykatów typu 'equal' ('=', 'IS NULL', 'IS NOT DISTINCT FROM')

Na kolejnym miejscu wszystkie predykaty typu IN-List

Na kolejnym najbardziej selektywny predykat zakresu (>, <, >=, <=, BETWEEN, LIKE 'x%').

Na ostatnim miejscu pozostałe kolumny z predykatów zakresu (kolumny screening)

Rys: Tab1

Dla tabeli wewnętrznej (**Inner**) zakładamy index (najlepiej klastrowy) rozpoczynając od kolumny na predykanie ze złączenia (**join predicate**) a w dalszej kolejności dokładając kolumny z predykatów lokalnych tej tabeli zgodnie z kolejnością Rys: Tab1.

W tym wypadku zarówno predykaty join jaki i local mogą być Matching dla **Inner Table**.

W przypadku, gdy w naszym query występuje dodatkowo zdanie **ORDER BY**, to sortowanie jest zawsze wykonywane dla tabeli wewnętrznej (**Inner**) bez względu na to jakie mamy

indexy, dlatego też należy dążyć do tego, żeby tabelą zewnętrzną (**Outer**) była tabela której to kolumn tyczy się zdanie **ORDER BY**, ponieważ tylko w tym przypadku możemy uniknąć sortowania stosując odpowiednie index-y.

Generalna reguła mówi, że Db2 wybiera tabelę zewnętrzną (**Outer**) tą, która jest mniejsza, czyli zwraca mniej rekordów po rozwiązaniu lokalnych predykatów. Dla optymalizatora Db2 mała tabela to taka, która jest mniejsza niż 2% wielkości lokalnego Buffer Pool w którym jest zdefiniowana.

Ogólnie – dla tego typu złączeń najbardziej wydajne indeksy tworzymy według poniższych schematów, gdzie

IXouter – indeks na tabeli **Outer**

IXinner – indeks na tabeli **Inner**

IXouter = (Kolumny z lokalnych predykatów tabeli Outer zgodnie z kolejnością Rys: Tab1 a następnie kolumny z predykatów połączenia - Join)

IXinner = (Kolumny z predykatów połączenia – Join a następnie kolumny z lokalnych predykatów tabeli Inner zgodnie z kolejnością Rys: Tab1)

Rys: Tab2

W przypadku gdy chcemy mieć dostęp Index Only:

IXouter = (Kolumny z lokalnych predykatów tabeli Outer zgodnie z kolejnością Rys: Tab1, następnie kolumny z predykatów połączenia – Join a w ostatnim kroku pozostałe kolumny ze zdania SELECT)

IXinner = (Kolumny z predykatów połączenia – Join, następnie kolumny z lokalnych predykatów tabeli Inner zgodnie z kolejnością Rys: Tab1 a w ostatnim kroku pozostałe kolumny ze zdania SELECT)

Rys: Tab3

Jeśli mamy ORDER BY i chcemy mieć dostęp Index Only i jednocześnie chcemy uniknąć sortowania:

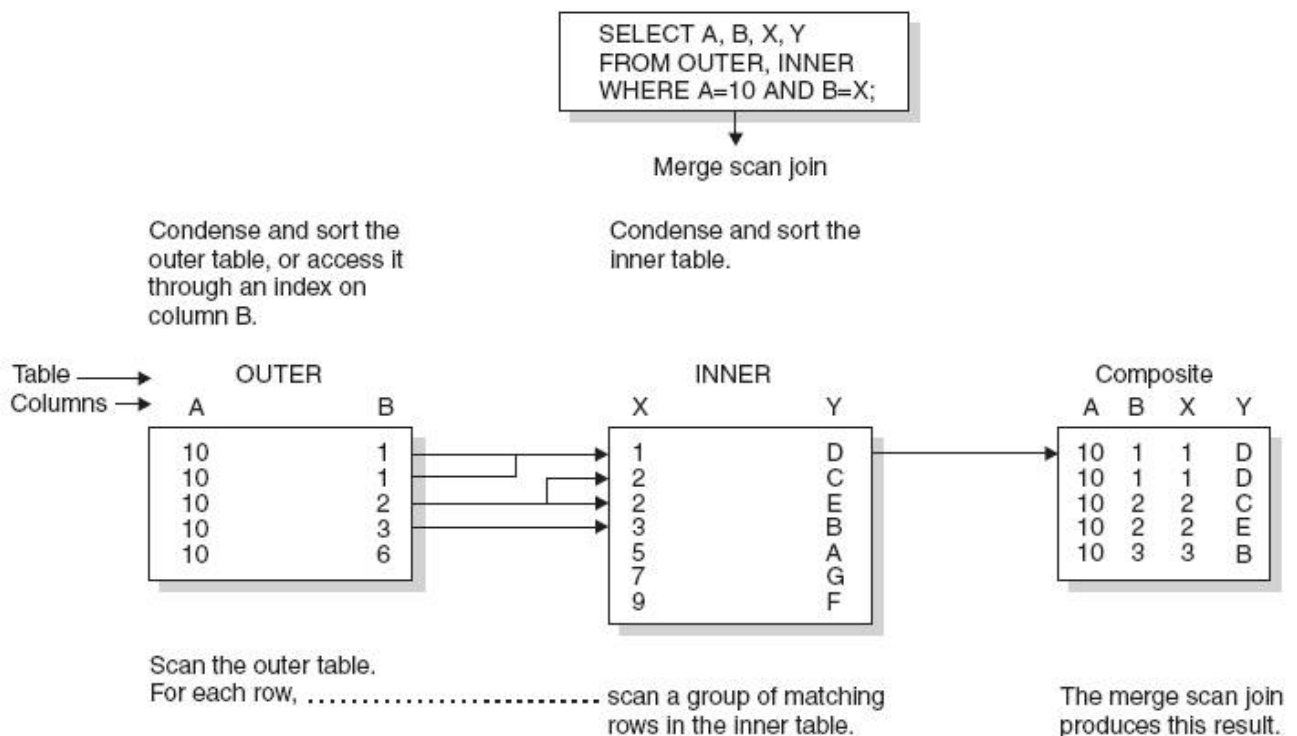
IXouter = (Kolumny typu 'equal' z lokalnych predykatów tabeli Outer, następnie kolumny z zdania ORDER BY w dokładnie takiej kolejności w jakiej występują w tym zdaniu, następnie pozostałe lokalne predykaty 'not equal' ze zdania WHERE, następnie kolumny z predykatów połączenia – Join a w ostatnim kroku pozostałe kolumny ze zdania SELECT)

IXinner = (Kolumny z predykatów połączenia – Join, następnie kolumny z lokalnych predykatów tabeli Inner zgodnie z kolejnością Rys: Tab1 a w ostatnim kroku pozostałe kolumny ze zdania SELECT)

Rys: Tab4

Najczęściej Nested Loop Join jest wybierany przez Optymalizator Db2, gdy lokalne predykaty są mocno filtrujące dla **Outer Table** i jest założony index klawiszujący na join column dla **Inner Table**.

Merge scan join (METHOD=2)



Rys: METHOD2

Przy takim dostępie do danych, Db2 skanuje (w celu rozwiązania lokalnych predykatów) w tym samym czasie zarówno tabelę zewnętrzną (**Outer table**) jak i tabelę wewnętrzną (**Inner table**). Wynik dla **Inner table** sortuje według kolumny z predykatu złączenia i umieszcza w Work File. Wynik dla **Outer table** jest sortowany tylko wtedy, gdy występują duplikaty. Optymalizator Db2 jako tabelę zewnętrzną (**Outer table**) wybiera tą, która ma mniej duplikatów a jako tabelę wewnętrzną (**Inner table**) tą, która w zdaniu SELECT zwraca mniej kolumn (mniejszy koszt sortowania).

Ogólnie – najbardziej wydajne indeksy (najlepiej klastrujące) tworzymy według schematów:

IXouter/IXinner = (Kolumny z predykatów połączenia – Join, następnie kolumny z lokalnych predykatów tabeli Inner zgodnie z kolejnością Rys: Tab1 a w ostatnim kroku pozostałe kolumny ze zdania SELECT)

Rys: Tab5

Najczęściej Merge Scan jest wybierany przez Optymalizator Db2, gdy tabele są duże (brakuje lokalnych, mocno selektywnych predykatów) i nie ma odpowiednich index-ów.

Hybrid join (METHOD=4)

Hybrid Join to zmodyfikowany Nested Loop Join, w którym mamy dodatkowo dostęp typu List Prefetch z powodu braku klastującego indeksu lub też mocno zdezorganizowanego istniejącego indeksu klastującego z predykatu złączenia na tabeli wewnętrznej (**inner table**).

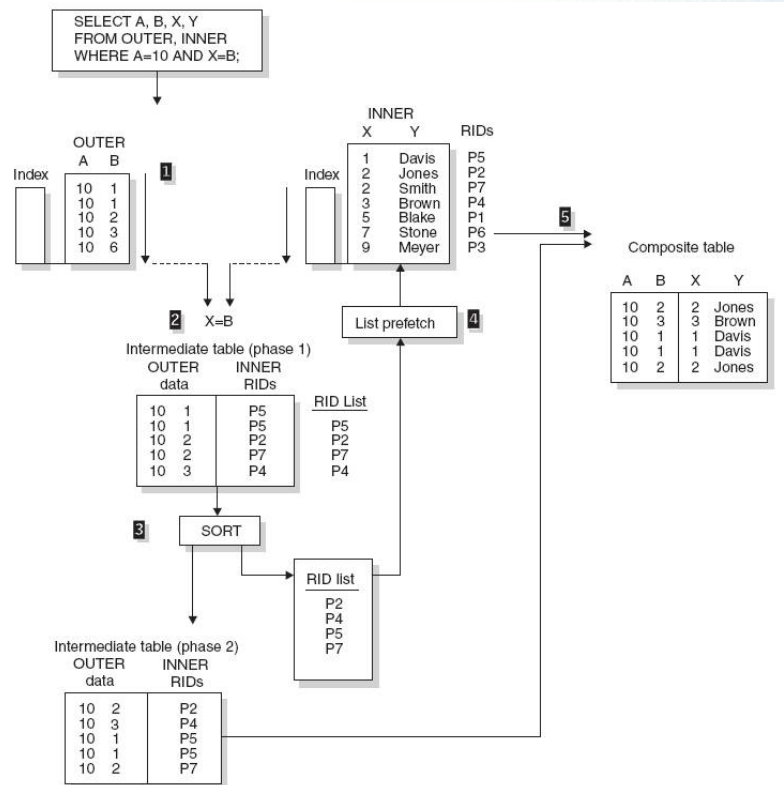
Najczęściej Hybrid Join jest wybierany przez Optymalizator Db2, gdy na **Inner Table** jest założony index nieklastujący na kolumnie z join oraz występuje dużo duplikatów w **Outer table**.

Index-y tworzymy analogicznie jak dla Nested Loop Join.

Podsumowując:

W celu zminimalizowania ilości GATEPAGE-y a co za tym idzie obniżenia zużycia CPU i skrócenia czasu wykonania operacji typu JOIN, powinniśmy dążyć do tego, aby Optymalizator Db2 wybierał (o ile to jest możliwe) najbardziej wydajny Nested Loop Join.

Osiągniemy to zakładając prawidłowe indeksy według podanych w artykule algorytmów.



Rys: METHOD4

PDUG TECH EVENT - 2018 POLAND WROCŁAW

April 09-10, 2018, SQLAdria Seminar - Ljubljana

April 12 SQLAdria, Belgrade, Serbia

April 16 IBM London South Bank Building - Latest News on Db2 for z/OS and Enterprise Analytics

April 17 IBM Client Center, Milan, Italy - European IDAA User Group meeting

May 29-31 DB2 Symposium, The Netherlands

June 10-14 SQLAdria, Sibenik, Croatia

June 18-22 IBM Hursley, UK, Db2 for z/OS Masterclass (John Campbell's education by invite)

September SQLAdria, Zagreb, Croatia

September PDUG, Wrocław(?), Poland

October 15-19, Db2 for z/OS Masterclass – IBM Silicon Valley Lab, San Jose
(John Campbell's education by invite)

November 4-8 IDUG EMEA 2018, Malta

November 13 SpDUG forum, Madrid

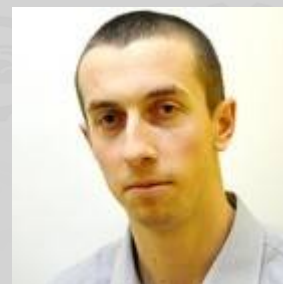
November 29-30 SQLAdria, Opatija, Croatia

**PDUG TECH EVENT
2018, POLAND, WROCŁAW**
www.pdug.pl

Testowanie Bazy danych Db2

w wersji 12

Mateusz Książek - PKO Bank Polski S.A.



1. Warunki testowe

Na potrzeby wykonania testów wydajnościowych, z czego każdy pierwszy test nie był brany pod uwagę w porównaniach, ponieważ miał za zadanie otwarcie zbiorów i załadowanie danych do pamięci. Zebrano dane z następujących rekordów SMF:

rekord 100 class (1, 3, 4, 5, 8) - statistics trace
rekord 101 class (1, 2, 3, 5, 7, 8) - accounting trace

V11 NFM,
V12 BNFM (V12R1M100)
V12 ANFM (V12R1M500).

Istotną zmianą w wersji V12 jest nowy sposób migracji silnika baza danych. IBM postanowił również zmienić konwencje nazewnictwa, która obowiązywała od lat tj. „Conversion Mode” w nowej wersji Db2 zmienił nazwę na „Before New Function Activation”, „New Function Mode” na „After New Function Activation”, natomiast ENFM został całkowicie usunięty. Zmiana podejścia do migracji związana jest z udostępnieniem możliwości szybszego aplikowania nowych funkcjonalności dla użytkowników Db2 (Continuous Delivery). Dzięki tym zmianom IBM będzie mógł w łatwy sposób dostarczyć nowe funkcjonalności w formie PTF-ów bez wymogu wykonywania kosztownego i pracochłonnego procesu migracji do nowej wersji.

Testy wydajnościowe wykonano na następującej konfiguracji:

- 2 z/OS LPARs: RAC1, RAC2, running z/OS 2.1
- 1 ICF LPAR: CCF0D, CFLEVEL 21
- 4 węzłowy data-sharing: DBP1, DBP3 à RAC1; DBP2, DBP4 à RAC2

Łącznie wykonano 9 testów (po 3 dla każdej wersji Db2) z czego każdy pierwszy test nie był brany pod uwagę w porównaniach, ponieważ miał za zadanie otwarcie zbiorów i załadowanie danych do pamięci. Zebrano dane z następujących rekordów SMF:

rekord 100 class (1, 3, 4, 5, 8) - statistics trace
rekord 101 class (1, 2, 3, 5, 7, 8) - accounting trace

na podstawie których wykonano raporty wydajnościowe. W czasie testów osiągnięto około 70% średniego produkcyjnego ruchu transakcyjnego. Tabela 1 przedstawia tryby/mody Db2 oraz interwały, w których wykonano testy. Niektóre dane wydajnościowe przedstawione w dalszej części artykułu będą się odnosić do kolumny TESTID.

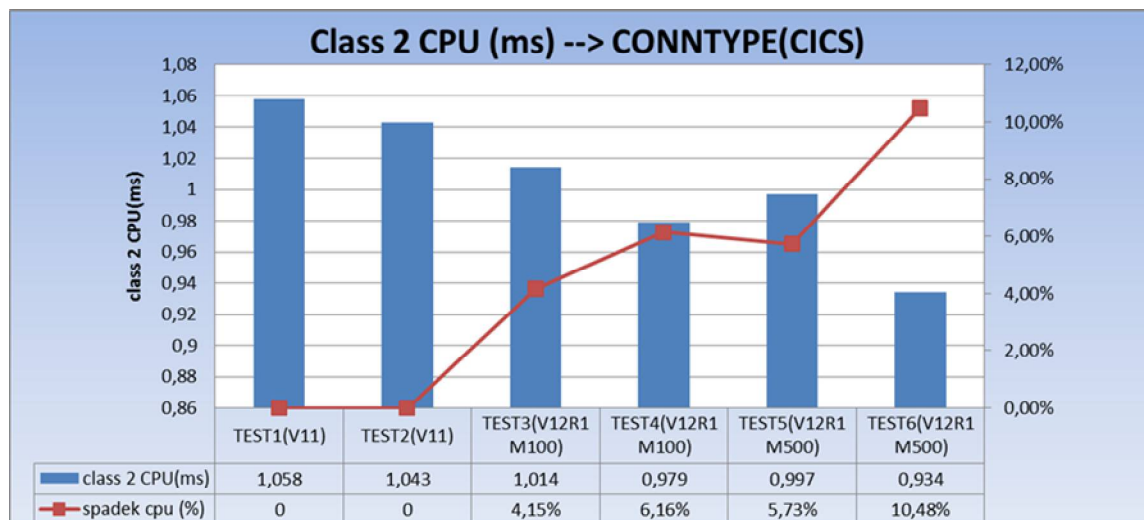
Tabela 1: Wersja DB2/CZAS/TESTID

DB2 VERSION	DATE	START Time	END Time	TESTID
V11	10/21/2017	12:12:00.00	12:19:00.00	TEST 1
V11	10/21/2017	12:27:00.00	12:35:00.00	TEST 2
V12R1M100	10/22/2017	08:41:00.00	08:47:00.00	TEST 3
V12R1M100	10/22/2017	08:57:00.00	09:06:00.00	TEST 4
V12R1M500	10/25/2017	18:39:00.00	18:44:00.00	TEST 5
V12R1M500	10/25/2017	18:53:00.00	18:59:00.00	TEST 6

2. Wyniki Performance-owe

Analizując wyniki należy porównywać TEST3, TEST5 do TEST1 oraz TEST4, TEST6 do TEST2 ze względu na podobieństwa symulowanego ruchu transakcyjnego. Testy pozwoliły zaobserwować znaczący spadek zużycia centralnego procesora (Db2 Class 2 CP CPU Time) po migracji do nowej wersji Db2.

Rysunek 1 Class 2 CPU Time



Największy spadek zużycia CP został osiągnięty po przejściu do ANFA i wykonaniu rebind-u całej aplikacji. We wszystkich poniższych wynikach przedstawiono rezultaty z testu piątego, jednak powinny być one odrzucone ze względu na zadania wsadowe, które spowodowały degradację CPU. Przypadek z TEST 5 pokazuje jak istotną kwestią jest zapewnienie odpowiednich warunków testowych, każda anomalia może spowodować przekłamanie końcowych wyników.

Tak, jak w przypadku CLASS 2 CPU zaobserwowano również znaczący spadek zużycia CPU w przypadku CLASS 7 (Package Times), przedstawiony na rysunku 2. Na dodatkowy uzysk w ANFA dołożyła się nowa funkcjonalność Fast Traverse Block (FTB).

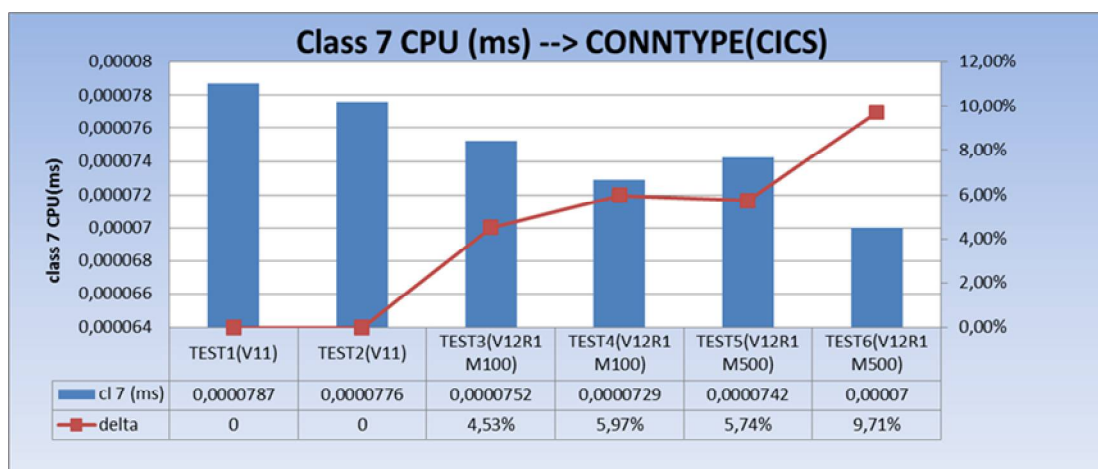
Jest to specjalny obszar pamięci zarezerwowany dla wybranych stron indeksu (root i non-leaf page). Pamięć (Real Memory) dla FTB jest alokowana automatycznie przez Db2 poza

bufferpoola-ami (ale alokowana proporcjonalnie, 20% wartości bufferpool-i). Nowa funkcja wspiera unikalne indeksy z kluczem krótszym niż 64 bajty. Db2 automatycznie wykrywa unikalne indeksy, które mogą korzystać z nowego obszaru pamięci. Wątek systemowy odpowiedzialny za selekcję indeksów obsługiwany jest przez procesory zIIP.

Rysunek 3 przedstawia zużycie pamięci oraz liczbę indeksów korzystających z FTB. Na wykresie widać, iż funkcja została

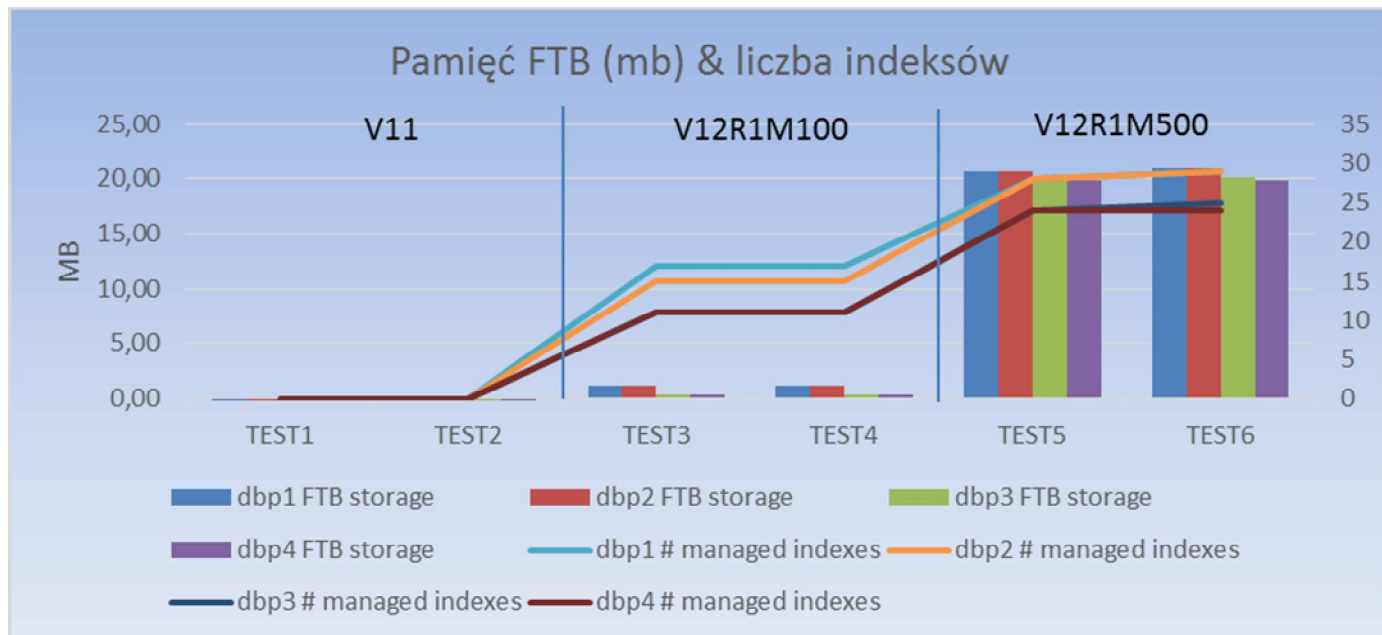
w pełni użyta dopiero w ANFA, ponieważ w trybie BNFA Db2 może wykorzystywać tylko indeksy które nie są „GBP-dependent”. Dopiero po przejściu do ANFA z FTB mogą korzystać wszystkie indeksy. Dane pokazują iż niewielkie

zużycie pamięci może się przełożyć na oszczędności na CP.



Rysunek 2 Class 7 Cpu Time

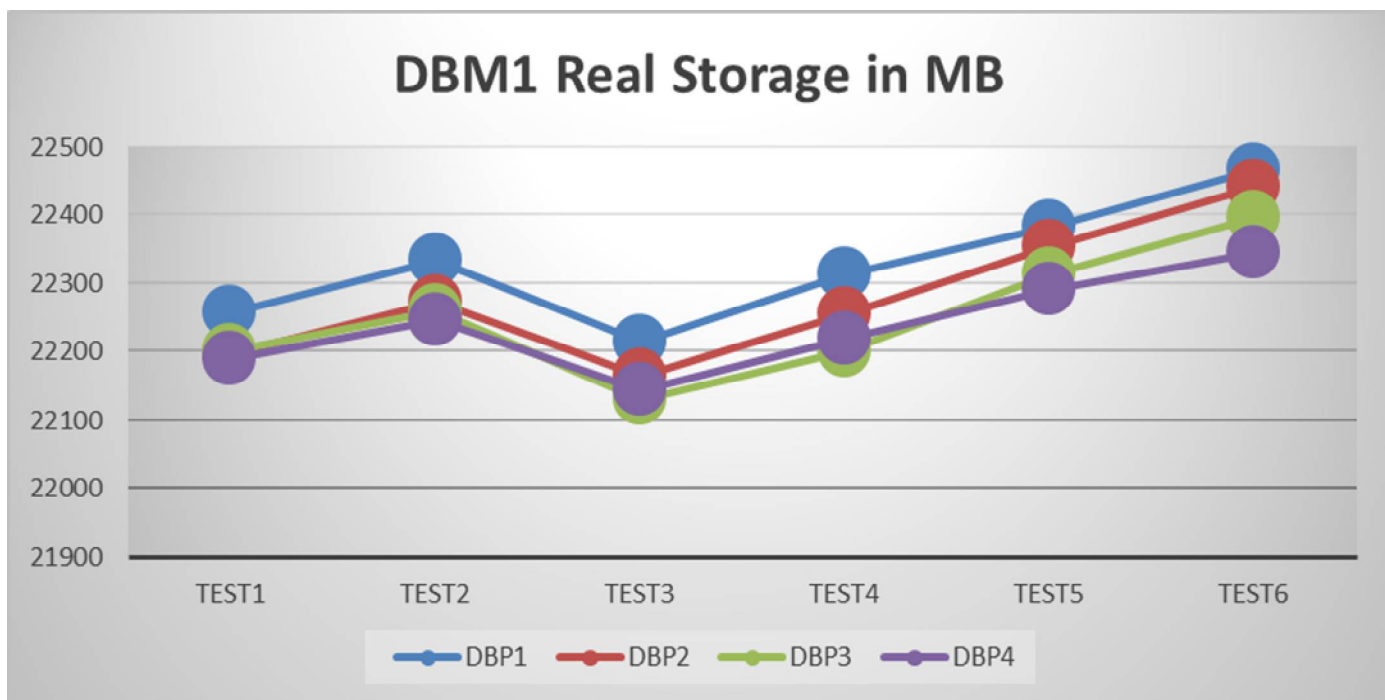
Według dokumentacji powinniśmy również zaobserwować spadek liczby GETPAGE na bufferpool-ach, w których są zaalokowane indeksy korzystające z FTB. Dane statystyczne z BP pokazały iż spadek



Rysunek 3 Wykorzystanie

Getpage jest niewielki i nie można jednoznacznie stwierdzić czy został spowodowany przez FTB. nie w tym obszarze powinno zachęcić użytkowników do większego wykorzystania RELEASE(DEALLOCATE).

Po migracji do Db2 V12 zaobserwowano dużą redukcję CLAIM Requests w trybie BNFM. Spadek (o około 40%) spowodowany jest usprawnieniem obsługi programów zbindowanych z opcją RELEASE(DEALLOCATE). W Db2 V12 zrezygnowano z wykonywania większości operacji DECLAIM, które były wykonywane w V11 w czasie COMMIT-u. Kolejne usprawnie-



Rysunek 4 Pamięć konsumowana

Interesująco wygląda również redukcja latch-a LC24, który jest odpowiedzialny za zarządzanie pamięcią w obszarze EDM pool. Z wersji na wersje EDM pool zużywa coraz więcej pamięci przez co koszt zarządzania nim również się zwiększał. W wersji V12 proces zarządzania EDM pool-em został wyraźnie usprawniony co przekłada się np. na wyraźny spadek zużycia pamięci przez DBD EMD pool(około 80%). Kolejny interesujący fakt to zniknięcie latch-y LC19(Log write latch contention) i LC23(buffer manager timer latch) w wersji V12, zaobserwowany już w trybie BNFM. Usunięcie tych dwóch latch-y wprost przekłada się na redukcję class 2 CPU time.

Zaobserwowano również znaczący spadek w operacjach Dynamic prefetch. W wersji V12 znacząco ograniczono niepotrzebne wywoływanie Dynamic Prefetch requests, dane wskazują na średnią redukcję na poziomie 50% per bufferpool. W przypadku obszaru LOCK-ów nie zaobserwowano większych zmian. Raporty Locking obejmujące: LOCK, UNLOCK, QUERY REQUEST, SUSPENSION nie wykazały znaczących zmian po migracji do ANFA.

Ze względu na nowe funkcje w silniku bazy danych Db2 z wersji na wersje zużywa coraz więcej pamięci. Sytuacja wygląda podobnie przy migracji do wersji V12, zaobserwowano wzrost zużycia pamięci o około 150 MB/per subsystem po aktywacji trybu ANFM.

Sprawdzono również zużycie CPU przez poszczególne ADDRESS SPACE Db2, jednak nie zaobserwowano znaczących zmian w konsumpcji CPU time (address space SRB time).

3. Utility

Wykonano podstawowe testy Utilit Db2, tj. sprawdzono LOAD, REORG, REBUILD i RUNSTAT. Testy polegały na uruchomieniu utility na wybranych obiektach aplikacji, sprawdzeniu nowych funkcjonalności oraz porównaniu CPU Time oraz Elapsed Time. Z wersji na wersje IBM „pozwała” coraz więcej procesów obsłużyć za pomocą procesorów zIIP, tak się również stało w tym przypadku utilit Db2 w wersji V12. W każdym z przebiegów odnotowano większy offload na procesory zIIP:

REORG – około 70% CP offloaded on zIIP(w V11 offload na poziomie 30%)

LOAD - około 70% CP offloaded on zIIP(w V11 offload na poziomie 5%)

REBUILD - około brak zmian w stosunku do wersji V11

RUNSTAT – około 40 % offloaded on zIIP

W przypadku Elapsed Time różnice były niewielkie . Największą zaletą jest oszczędność zużycia CP, na podstawie których IBM nalicza opłaty.

4. Podsumowanie

Przeprowadzone testy pozwalają stwierdzić, iż migracja do nowej wersji Db2 powinna istotnie zredukować zużycie CP. Nowe funkcjonalności i możliwości silnika bazy danych

powinny w znaczący sposób uprościć zarządzanie i aplikowanie zmian w strukturach Db2 naszej aplikacji.

Aplikowanie zmian i nowych funkcjonalności za pomocą PTF-ów wydaje się idealnym rozwiązaniem przy skomplikowanych instalacjach. Nowy proces pozwoli na uniknięcie kosztów migracyjnych.

W wersji V12 udało się również IBM-owi poprawić obsługę Flash Copy w Reorg-ach, pojawiła się też istotna funkcjonalność TRANSFER OWNERSHIP, która w łatwy sposób pozwoli na uprządkowanie tematu uprawnień w naszej bazie danych.

Kopiowanie Filesystemów dla Db2 w środowisku AIX

Bartosz Faszczewski - Bank Pekao S.A.



Podstawowym zadaniem każdego administratora zarówno baz danych jak i systemów operacyjnych jest zachowanie ciągłości procesów biznesowych i spójności magazynowanych danych. Takie uwarunkowania sprawiają, że powielamy nasze środowiska, budując klastry wysokiej dostępności (High Availability Disaster Recovery) zarówno na systemach operacyjnych, jak i bazach danych.

Zazwyczaj w naszych sieciach wykorzystujemy sieć SAN z macierzami dyskowymi, dlatego też jednym z podstawowych etapów przygotowania środowiska Unixowego w systemie AIX pod serwer baz danych Db2 jest utworzenie odpowiedniej struktury dzielącej przestrzeń dyskową. Przy odpowiedniej konfiguracji jesteśmy w stanie kontrolować to, co się dzieje z naszymi danymi na dysku, zwiększać je i przenosić bez konieczności zatrzymywania systemu operacyjnego. A o to właśnie między innymi chodzi w rozwiązaniach HADR.

System AIX posiada funkcję zarządzania przestrzenią dyskową o nazwie LVM (Logical Volume Manager), która pozwala utworzyć Volume Groups oraz Logical Volume dla naszych filesystemów. LVM zarządza zasobami dyskowymi przez odpowiednie mapowanie danych między fizyczną a logiczną warstwą. Dzięki

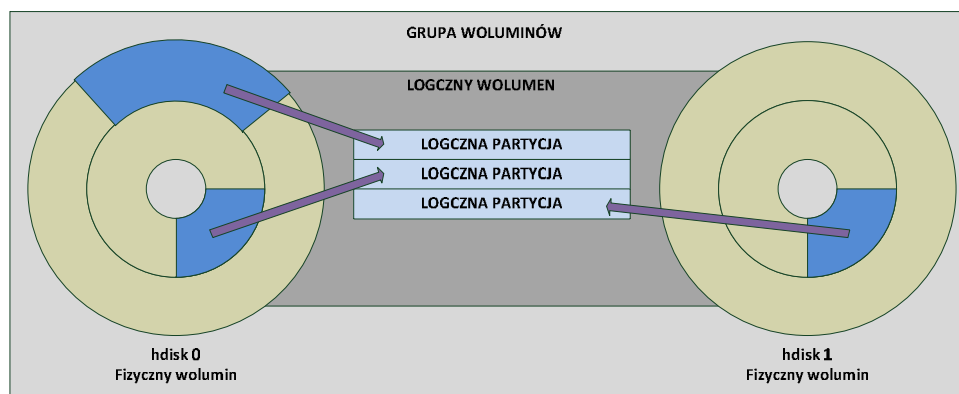
niemu podczas ciągłego działania systemu jesteśmy w stanie powiększać przestrzeń dyskową i bezpieczniejszymi sposobami zarządzać naszymi cennymi danymi.

W tym artykule chciałbym jednak skupić się na oszczędzeniu nakładu pracy przy tworzeniu filesystemów w środowisku klastrów z serwerem Db2.

Filesystemy to hierarchiczna struktura plików i katalogów systemu operacyjnego, dzięki odpowiedniej strukturze katalogów zachowujemy porządek i spójność między różnymi wersjami systemu operacyjnego. Tworząc filesystemy dla odpowiednich zasobów DB2 mamy możliwość kopiowania, przenoszenia zabezpieczania całych systemów plików. Ten układ pozwala nam stworzyć zasób w którym baza danych będzie mogła poruszać się w sposób automatyczny i dzięki temu bezpieczny z punktu widzenia całego systemu operacyjnego.

Załóżmy zatem, że posiadamy dwa lub więcej serwerów, na których będą założone identyczne filesystemy zawierające podstawowe kontenery na składowanie plików baz danych, binariów DB2, logów transakcyjnych, instalatorów itp. Tworzymy je jednorazowo na serwerze macierzystym i przenosimy ich

strukturę na inny serwer za pomocą narzędzia savevg. Dzięki niemu możemy kopiować i odtwarzać backupy wszystkich grup woluminów poza wolumenem systemowym rootvg. Co prawda to narzędzie pozwala na utworzenie backupu grupy wolumenu rootvg, jednak po odtworzeniu go na innym serwerze nie dałoby się zbootować systemu. Dlatego do tej wyjątkowej grupy służy inne narzędzie – mksysb.



Logiczna i fizyczna struktura przestrzeni dyskowej

Ta funkcjonalność działa również w przypadku, w którym chcemy dokonać migracji danych do nowszego serwera mającego zawierać identyczną strukturę co wcześniejszy serwer. Zatem gdy mamy już odpowiednio przygotowany serwer zawierający podział na Volume Groups oraz Logical Volume z Filesystemami wywołujemy narzędzie `savevg` przez wydanie komendy `savevg`, opcjonalnie z parametrem `-r`. Spowoduje to, że backup nie będzie zawierał danych, które znajdowały się w macierzystym serwerze.

Funkcja kopiowania `savevg` może zatem służyć także do zabezpieczania całych grup wolumenów DB2 na poziomie administracji systemu operacyjnego. Po wykonaniu polecenie:

`savevg -f /db2/db2vg_backup -i db2vg`

Zakładamy, że na naszym serwerze macierzystym mamy następującą strukturę filesystemów na grupie wolumetów o nazwie `db2vg`. W tym przykładzie wyseparowano dane wolumeny do przechowywania plików baz danych, logów archiwalnych, plików użytkownika instancyjnego oraz binariów DB2.

Filesystem	GB blocks	Free	%Used	lused %	lused	Mounted on
<code>/dev/dbdata1v</code>	60.00	7.42	88%	5190	1%	<code>/data</code>
<code>/dev/dbarch1v</code>	5.00	0.06	99%	54	1%	<code>/arch</code>
<code>/dev/dbbins1v</code>	10.00	0.49	96%	4681	4%	<code>/db2inst</code>
<code>/dev/dbhomel1v</code>	10.00	6.50	36%	10015	1%	<code>/db2_bin</code>

```
root@myhost: /> savevg -r -f /data/savevg.new' db2vg
Creating information file for volume group db2vg.
Backing up user Volume Group information files only.
Creating list of files to back up.
Backing up 6 files
6 of 6 files (100%)
0512-038 savevg: Backup Completed Successfully.
```

```
root@myhost: /> restvg -r -f /db2vg.savevg hdisk1 hdisk2

Will create the Volume Group: db2vg
Target Disks: hdisk1 hdisk2
Allocation Policy:
    Shrink Filesystems: no
    Preserve Physical Partitions for each Logical Volume: no
Enter y to continue: y
0516-1254 /usr/sbin/mkvg: Changing the PVID in the ODM.
db2vg
loglv00
db2vg01
```

Oczywiście to samo działanie możemy wykonać z poziomu smitty czyli System Management Interface Tool wydając polecenie smitty `savevg`.

W kolejnym etapie należy przekopiować nasz wygenerowany plik `savevg.new` na nowy serwer i tam odtworzyć za pomocą polecenia `restvg`. Składnia polecenia odtwarzającego nasz backup to:

```
root@myhost: /> restvg -r -f /data/savevg.new' hdisk1
```

Powyższe polecenie zrzuciło nam całą strukturę Volume Groups oraz Logical Volume z Filesystemami

Zwróćmy uwagę, że przy odtwarzaniu naszej struktury wskazujemy, na ilu docelowo dyskach ma się ona znajdować. W tym miejscu możemy określić jej rozmieszczenie i odtwo-

zenie nawet wtedy, gdy mamy mniejszą lub większą ilość dysków. Ilość dysków przypisanych do Volume Groups możemy sprawdzić poleceniem:

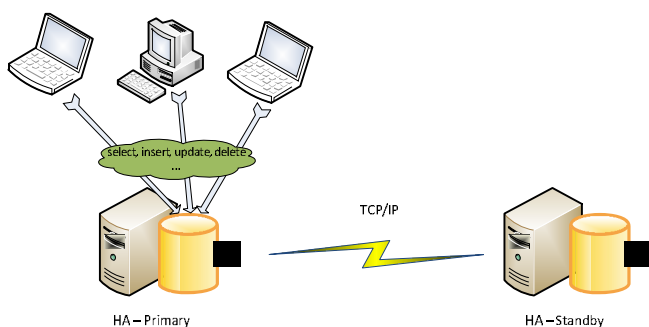
```
root@myhost: /data> lsvg -p db2vg
db2vg:
PV_NAME      PV STATE      TOTAL PPs  FREE PPs  FREE
DISTRIBUTION
hdisk2       active        239        0         00..00..00..00..00
hdisk3       active        239        15        04..00..00..00..11
hdisk4       active        247        0         00..00..00..00..00
```


Db2 HADR – ROS (Read on Standby)

Bartosz Faszczewski - Bank Pekao S.A.

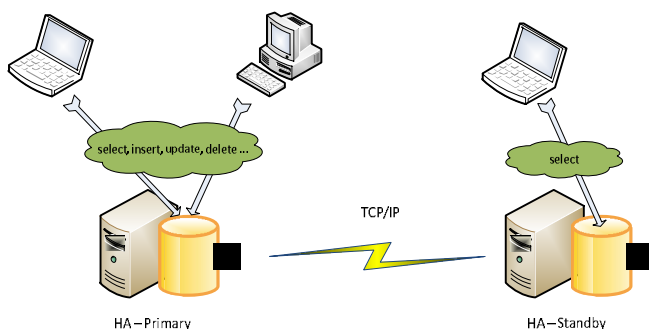


Bazy danych Db2 w dystrybucji LUW dostarczają nam mechanizmy wysokiej dostępności HADR (High Availability Disaster Recovery), które zapewniają synchronizację danych między dwoma bazami danych o nazwie Primary i Standby. Dzięki tej synchronizacji aplikacje łączące się z bazą danych pracują nieprzerwanie w przypadku awarii podstawowej bazy Primary ponieważ w momencie awarii w locie przełączane są do zawsze aktualnej bazy Standby. Taki mechanizm zapewnia wysoką dostępność oraz maksymalną ochronę danych przed ich utratą.



HADR - wszystkie aplikacje łączą się z Primary

W tym artykule chciałbym skupić się na opisie działania jednej z funkcjonalności HADR o nazwie HADR Read on Standby. Jest to funkcjonalność, która pozwala zredukować ogólny koszt transakcji na serwerze Db2.



HADR z ROS - część aplikacji (select) łączy się z Standby

Dzięki temu rozwiązaniu aplikacje w niektórych przypadkach mogą łączyć się z bazą Standby nawet wówczas, gdy całe środowisko działa bez awarii. Transakcje odczytu i modyfikacji danych na bazie Db2 wykonywane są na bazie podstawowej, czyli Standby i w tym zakresie nic się nie zmienia. Jednak wówczas, gdy aplikacja łącząca się z bazą danych nie ma obowiązku modyfikowania danych a jedynie ich odczytywania, to można skonfigurować ją w taki sposób, by łączyła się zawsze z bazą Standby. Należy jednak pamiętać, że w takim połączeniu zapytania do bazy będą wykonywane w trybie Uncommitted Read (UR), czyli nie będziemy blokować tablic dla transakcji pochodzących z serwera podstawowego. Warunkiem podstawowym takiego połączenia jest fakt, że baza Standby nie może się znajdować w trybie catch-up, czyli w stanie tzw. uzgadniania

zaległości. W tym trybie uruchamiana jest rezerwowa baza danych i następuje weryfikacja oraz synchronizacja operacji zapisanych w aktywnych i archiwalnych dziennikach transakcji. W przypadku, w którym baza znajduje się w każdym innym trybie czyli nearsync, sync, async lub supersync funkcjonalność ROS działa.

Dodatkowo w momencie pojawienia się awarii podczas łączenia się aplikacji z serwerem zapasowym Standby, połączenia typu Read on Standby zostaną zerwane, a wtedy serwer Standby przejmie obowiązki głównego serwera Primary. Do tego między innymi został powołany w środowisku wysokiej dostępności HADR.

Takie ograniczenie pozwala na odciążenie podstawowej bazy Primary. Należy jednak pamiętać, że wynik naszych zapytań może być nieaktualny. Dlatego to rozwiązanie jest zwykle stosowane dla aplikacji pozyskujących dane z bazy Db2 w celach raportowych. Pobrane dane bowiem mogą być w danym momencie nieaktualne z danymi w bazie Primary.

Jeśli nasze środowisko wysokiej dostępności działa poprawnie w HADR, to w celu uruchomienia ROS (Read on Standby) na naszym serwerze należy dokonać modyfikacji zmiennej DB2_HADR_ROS. Należy ją ustawić w tryb ON.

```
mydb1@:/db2inst/mydb1>db2pd -hadr -db MYDATABASEB

Database Member 0 -- Database MYDATABASE -- Active -- Up 146 days 19:33:41 -- Date 2018-03-07-15.14.31.075472

      HADR_ROLE = PRIMARY
      REPLAY_TYPE = PHYSICAL
      HADR_SYNCMODE = NEARSYNC
      STANDBY_ID = 1
      LOG_STREAM_ID = 0
      HADR_STATE = PEER
      PRIMARY_MEMBER_HOST = mydatabase1
      PRIMARY_INSTANCE = mydb1
      PRIMARY_MEMBER = 0
      STANDBY_MEMBER_HOST = mydatabase12
      STANDBY_INSTANCE = mydb1
      STANDBY_MEMBER = 0
      HADR_CONNECT_STATUS = CONNECTED
      HADR_CONNECT_STATUS_TIME = 11/27/2017 12:06:27.350618 (1511780787)
      HEARTBEAT_INTERVAL(seconds) = 7
      HADR_TIMEOUT(seconds) = 30
      TIME_SINCE_LAST_RECV(seconds) = 0
      PEER_WAIT_LIMIT(seconds) = 0
      LOG_HADR_WAIT_CUR(seconds) = 0.000
      LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.000758
      LOG_HADR_WAIT_ACCUMULATED(seconds) = 169426.398
      LOG_HADR_WAIT_COUNT = 215433749
      SOCK_SEND_BUF_REQUESTED,ACTUAL(bytes) = 0, 262088
      SOCK_RECV_BUF_REQUESTED,ACTUAL(bytes) = 0, 262088
      PRIMARY_LOG_FILE,PAGE,POS = S0387229.LOG, 14741, 6344163165992
      STANDBY_LOG_FILE,PAGE,POS = S0387229.LOG, 14619, 6344162670546
      HADR_LOG_GAP(bytes) = 20052
      STANDBY_REPLAY_LOG_FILE,PAGE,POS = S0387229.LOG, 14616, 6344162656300
      STANDBY_RECV_REPLAY_GAP(bytes) = 6771
      PRIMARY_LOG_TIME = 03/07/2018 15:14:30.000000 (1520432070)
      STANDBY_LOG_TIME = 03/07/2018 15:14:26.000000 (1520432066)
      STANDBY_REPLAY_LOG_TIME = 03/07/2018 15:14:26.000000 (1520432066)
      STANDBY_RECV_BUF_SIZE(pages) = 4300
      STANDBY_RECV_BUF_PERCENT = 0
      STANDBY_SPOOL_LIMIT(pages) = 0
      PEER_WINDOW(seconds) = 120
      PEER_WINDOW_END = 03/07/2018 15:16:30.000000 (1520432190)
      READS ON STANDBY ENABLED = Y
      STANDBY_REPLAY_ONLY_WINDOW_ACTIVE = N
```

Zwróćmy uwagę również na fakt, że wszelkie modyfikacje wykonywane na bazie Primary w HADR są zatwierdzane i przesyłane do bazy pomocniczej Standby w przypadku, gdy zostaną one zapisane na dysk. Zapis na dysk z kolei odbywa się po zatwierdzonej transakcji. Replikacja logów między bazami jest możliwa dla logów zapisanych na dysku twardym i dlatego problemem mogą okazać się długie transakcje, które na serwerze Primary pozostają w buforze dziennika podstawowego. Dlatego czytanie danych na bazie pomocniczej Standby z ROS może dawać nam niekompletne informacje. Analizując powyższe uwarunkowania możemy indywidualnie przeanalizować czy warto skrócić transakcje na serwerze podstawowym co sprawi, że na serwerze pomocniczym będziemy posiadać większe prawdopodobieństwo właściwych danych.

Techniques to move Db2 data from PROD to TEST (and related considerations)

Philippe Dubost - CA Technologies



A 12-year IT professional, Philippe Dubost (IBM Champion) is Product Manager at CA Technologies.

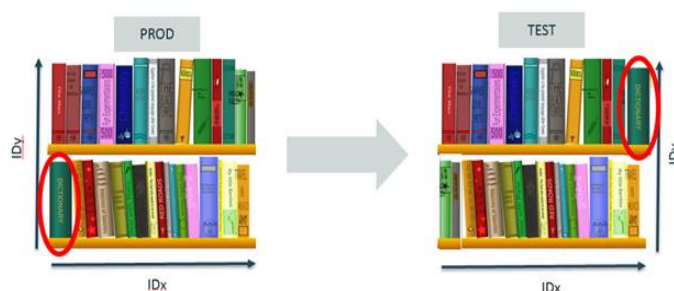
In this role, he is responsible for products planning and strategy, presenting and representing the products portfolio to customers and industry analysts, collecting customer requirements and transforming them into actionable Agile/Scrum stories in the engineering backlog.

Philippe is very active in the Db2 community, tweeting ([@Db2forZ](#)), blogging and given technical presentations in User Groups in Europe. Philippe is also the founder of [csDUG](#), the DB2 Regional User Group for Czech Republic and Slovakia.

In large scale organizations, it is often necessary to move data from Db2 Production subsystems to Test subsystems. Of the various technologies that are available to achieve this business objective, circumstances determine which is most appropriate. Additionally, it is sometimes necessary to consider extra technologies such as Encryption or Masking to meet data security expectations.

Techniques available to move DB2 data across subsystems

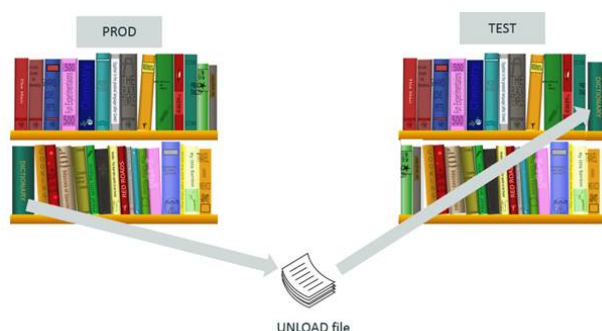
A picture says a thousand words. The following analogy may help you to digest some concepts that are often difficult for non-Db2 experts. A Db2 subsystem can be represented as a library that stores books (Db2 Tables), and each book contains data. In Db2, each object has an internal identifier, an ID. Those are called OBID "Object ID". For instance a Database has an ID, and a table has an ID. Think of a library where books have catalog numbers that are based on their location in the shelves. You can find a book in the shelves from its catalog number; similarly, you can locate a table in Db2 using a couple of IDs.



This concept is important to understand, because Db2 leverages those IDs to make sure that the

data is where it belongs. If the IDs do not match, Db2 fails to retrieve the corresponding data. Additionally, it is almost impossible for a Db2 administrator to ensure that a Db2 Object in the Test subsystem has the same object IDs as in the Db2 Production subsystem. As highlighted in the above illustration, the green dictionary (circled in red) is in both the PROD library and the TEST library, but not in the same place on the shelves. That's typically what PROD and TEST subsystems look like in DB2 environment.

UNLOAD / LOAD



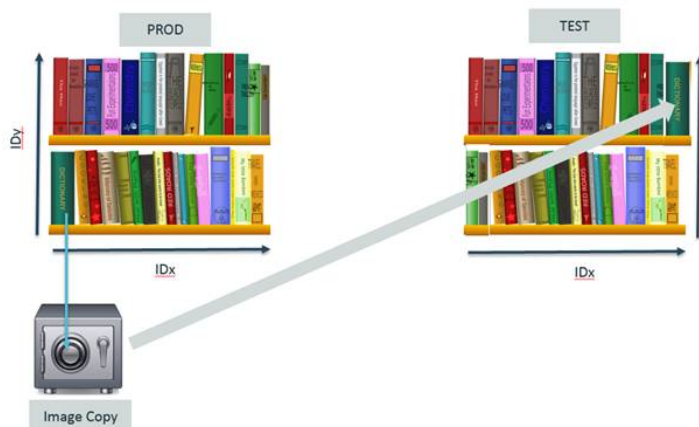
The most intuitive way of moving the data from a subsystem to another is by Unloading the data from the source subsystem (PROD), and Loading it back in the target subsystem (TEST).

However, in our library analogy, it's a bit as if you read the book in PROD, used a typewriter to extract the content, and gave this manuscript to a publisher. The publisher then created a new book that you stored in your TEST library. Similarly in DB2, this UNLOAD / LOAD technique is quite inefficient, slow, and CPU intensive.

That said, this technique has some good benefits. It's quite easy to put such a process in place (for instance, no need to care about the OBIDs), and it's quite adaptive; it can handle some differences in schema and attributes of the tables (tablespace). In our library analogy, the publisher may have created a book in different format (pocket size or large booklet) for you to store in the TEST library, but you have no trouble recognizing it as the same book.

To implement this technique, use [CA Fast Unload® for Db2 for z/OS](#) (to create the Unload file), and then [CA Fast Load® for Db2 for z/OS](#) (to load the Unload file into TEST table).

OBIDXLAT – leverage existing Image Copies



In your Db2 environment, you probably take Image Copies regularly to use in a Disaster Recovery scenario. In our library analogy, you may want to have a 2nd copy of your PROD book (an exact identical copy of the book) stored in a safe somewhere, so that if your PROD library burns down, you can replace it ASAP with the book in the safe. Rebuild index on TEST subsystem if required). Additionally, if required, [CA Fast Recover™ for Db2 for z/OS](#) is also able to perform log-apply to the target subsystem replicate the updates from the source since the last Image Copy.

RC/MERGER – “just” take a Copy



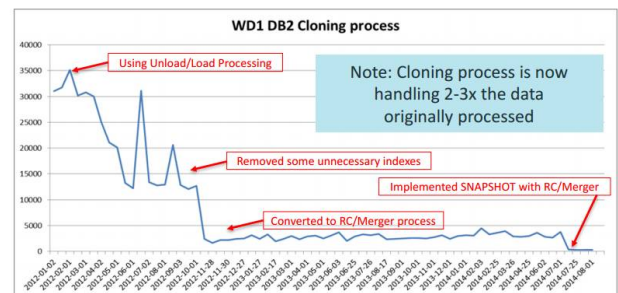
Using RC/Merger processing, you can move data from your DB2 Production subsystem to your DB2 Test subsystem just by running a Copy. This technique is protected by a patent in the USPTO. Simply said, RC/Merger coordinates IDs across PROD and TEST, and takes care of differences in the Db2 table schema and tablespace attributes.

In our library analogy, RC/Merger acts as your personal librarian, insuring that all the books in the TEST library are in the same position (IDs) and format/size (Schema, attributes) as the books in the PROD library.

Then moving data is a piece of cake. You can run a Copy program to create an Image Copy that you store right away in your TEST library. In Db2 terms, you simply need to make sure the copy is created using the VSAM naming convention for the underlying Db2 TEST tablespace.

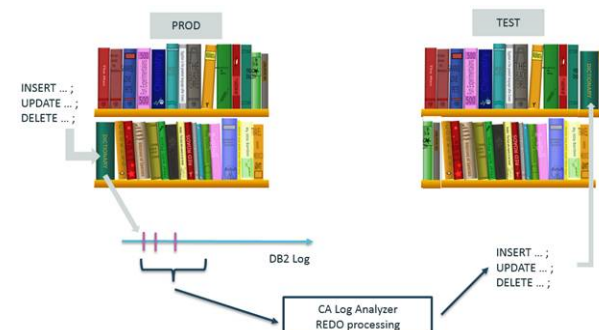
Saving Costs During Data Cloning

Customer Statistics Moving From Standard Unload/Load Method to RC/Merger



... technique can be implemented by using [CA RC/Migrator™ Db2 for z/OS](#) (that contains the RC/Merger feature), [CA ck Copy for Db2 for z/OS](#) (to create the Image Copy) | [CA Fast Recover™ for Db2 for z/OS](#) (to perform Rebuild ex on TEST subsystem if required).

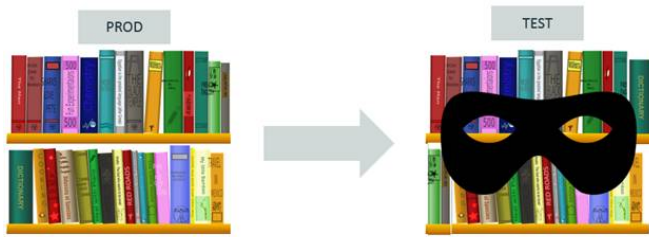
REDO processing to move DATA from PROD to TEST



When your data is “loaded” into the Db2 TEST subsystem, consider updating it on regular basis so the data is synchronized with PROD. You can perform this synchronization by using a Log Analysis tool. Such a tool reads the DB2 log and extracts the SQL DML activity (INSERT, UPDATE, DELETE, MERGE) that happened on the PROD table (in a given timeframe). The tool then plays the DML activity back on the TEST table (in other words, it executes those SQL statements against the TEST table to apply the same changes).

In our library analogy, if an editor came to annotate your green dictionary in PROD, to add a new word and its definition for instance, this REDO process would force him/her to go to the TEST library and make the same annotation there as well. This technique can be implemented by using [CA Log Analyzer™ for DB2 for z/OS](#) (that contains the REDO processing feature).

Masking considerations



Data Masking is an option if the business data is highly sensitive and must not be visible to employees who have access to the TEST environment. However, those employees typically include QA engineers (testers) who require the masked data to be good quality, in order to create their tests efficiently using QA Automation techniques.

Some considerations about masked data quality include:

Appropriate data format: For example, if a column, in a Db2 table, contains credit card number information, the Masking algorithm should produce masked data in a similar format (eg. respecting the Luhn algorithm) for the tests to work correctly.

Respect Db2 RI: In relational databases, such as DB2, tables are connected using RI techniques (Relational Integrity). The Masking algorithm has to respect those RIs, even when the RI keys require masking in case they contain sensitive information.

Persistent Masked data: QA engineers use this data to create automated tests in the TEST environment. If a column contains the customer's names, one row may correspond to "John Doe". If "John Doe" is masked by "Mickey Mouse", the "Mickey Mouse" mask may need to be persistent for QA Automation purpose. In other words, employees who have access to the TEST system shall not know who "Mickey Mouse" is in reality, but "Mickey Mouse" represents the same person every time the TEST environment is loaded (and masked).

Other business-related considerations: The masked data may need to follow certain rules, specific to a certain business. For example, financial institutions will have balance sheet tables that need to match across DB2 Tables given a certain timeframe (in other words, the balance shall be 0). The same applies to any data against which a checksum is performed.

Achieving the goal of good data masking, while maintaining the quality of the data for testing purposes is a challenge. CA Technologies offers several solutions to help with those challenges, such as:

[CA RC/Extract™ for Db2 for z/OS](#) can SUBSET data and "mask using scrambling".

[CA Test Data Manager](#), a recent powerful product offering to implement masking techniques.

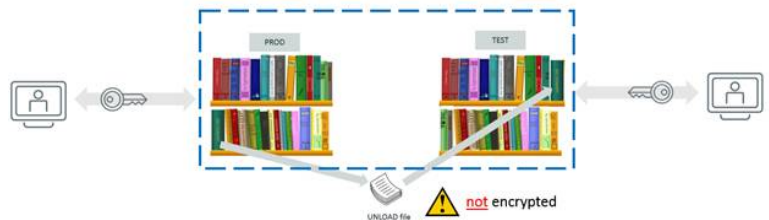
Encryption considerations

Large corporations may be required to encrypt their Db2 business data, as it typically contains very sensitive information (PI, PHI, ...). To access the data, users need the proper key:

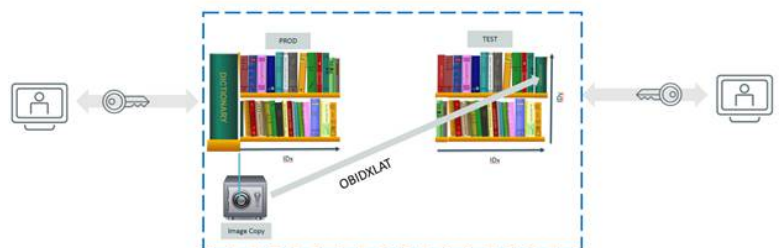
Several encryption techniques exist on the market, and users may implement their own. To take an example, IBM Guardium allows Encryption using EDITPROCs (containing the encryption algorithm) and an Encryption Key.

Consider the impact of your encryption methods, when trying to implement a mechanism to move data from one system to another.

If you are thinking about using the UNLOAD / LOAD technique, you should not have any technical issues. However, you need to be aware that the Unload File will contain unencrypted data (outside of the control of Db2), which may defeat the purpose of trying to encrypt your data in Db2!



If you are thinking about OBIDXLAT (or RC/Merger) technique, data security concerns are much reduced, but you need to keep in mind that the EDITPROCs and Encryption Keys must match between the DB2 Production subsystem and the DB2 Test subsystem (since you are using an exact copy of the (encrypted) book in PROD, you are just creating the same book in TEST – with the same encrypted data content, hence you need the same Encryption Key on the TEST subsystem to



read this book correctly).

A (small but important) note on Encryption: While DB2 encryption tools encrypt the data stored in Db2 tables, any key referenced by an index will be stored as unencrypted in the index (in other words your indexes will contain unencrypted data even if you activated the encryption on the related table). This helps SQL performance, as it avoids double-decryption for any SQL statement that uses an Index to access the data.

Although [CA Technologies](#) does not offer DB2 encryption product, we support customers using third party encryption software with our DB2 solutions for z/OS.

