



ZALETY NOWSZYCH WERSJI I KIERUNKI ROZWOJU SPDS-A

SŁAWOMIR BOKINIEC



AGENDA

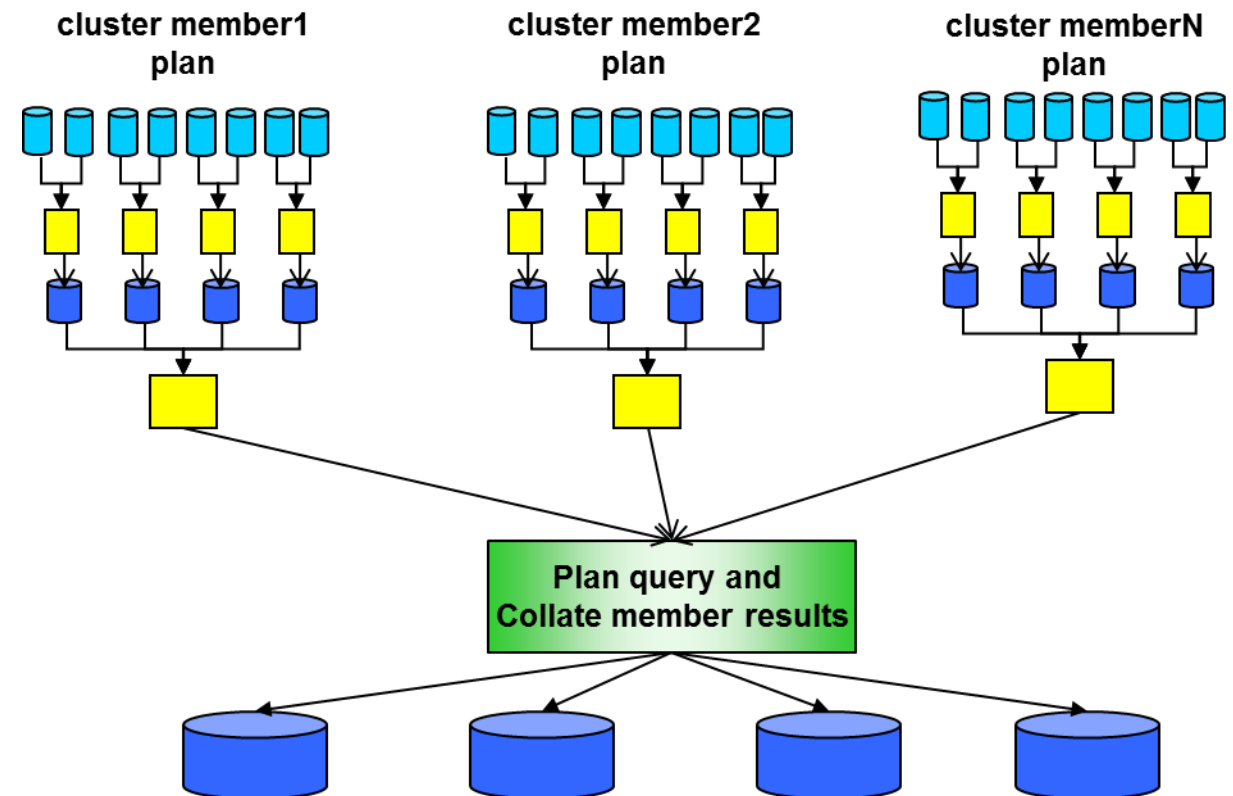
- Wybrane zalety wersji 5.1 i wcześniejszych
- SPDS z SAS Gridem
- Co kształtuje kierunki rozwoju?
- Nowsze wersje
- SPDS z Hadoopem



WYBRANE ZALETY SPDS-A

WCZEŚNIEJSZYCH WERSJI

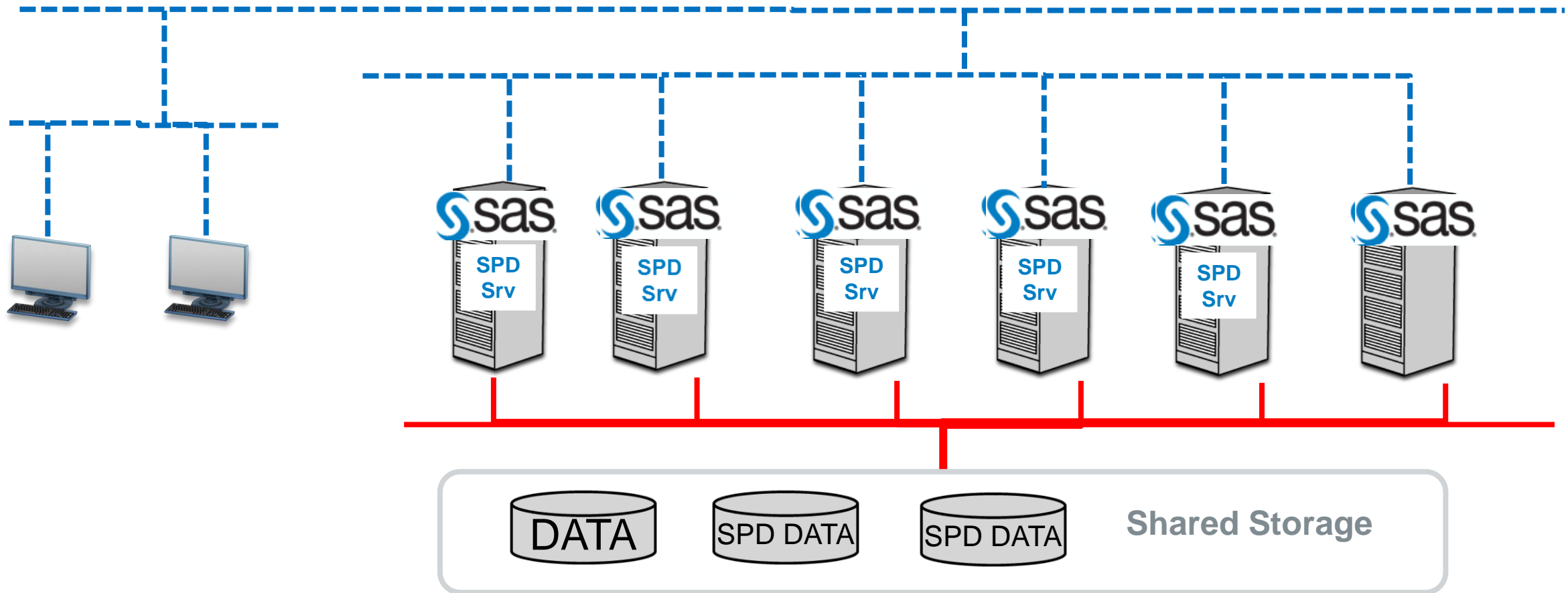
- Skalowalne, partycjonowane I/O
- Optymalizacje dla warunku where
- ACL-e
- Parallel Group by (PGB)
- Star join
- Parallel join (też z PGB)
- Uniwersalny indeks łączący zalety technologii B-Tree i bitmapy



- Uniwersalność klastrowania/partycjonowania
- Proc append uniquesave=rep
- Skuteczne Indeksy proste i nieraz wydajne przetwarzanie bez indeksów
- Mechanizm „dynamic locking” pozwalający na równoczesny zapis i modyfikację
- Podobnie jak Base może przechowywać dane posortowane
 - Dynamic Cluster BY Clause Optimization
- Dostęp spoza SAS-a
- Współpraca z SAS Grid

SAS SPDS Z SAS GRID'EM

- SAS na węzłach łączy się do lokalnego SPDS-a (localhost)



- Join Planner
 - jointech_pref=<index|hash|merge|seq>
- Ułatwienia w identyfikowaniu użytkownika proxy (spdsbase)
- Silny algorytm szyfrowania danych AES-256
- Wsparcie dla platformy Windows 64 bit
- Dynamiczne klastry zarządzane „online”

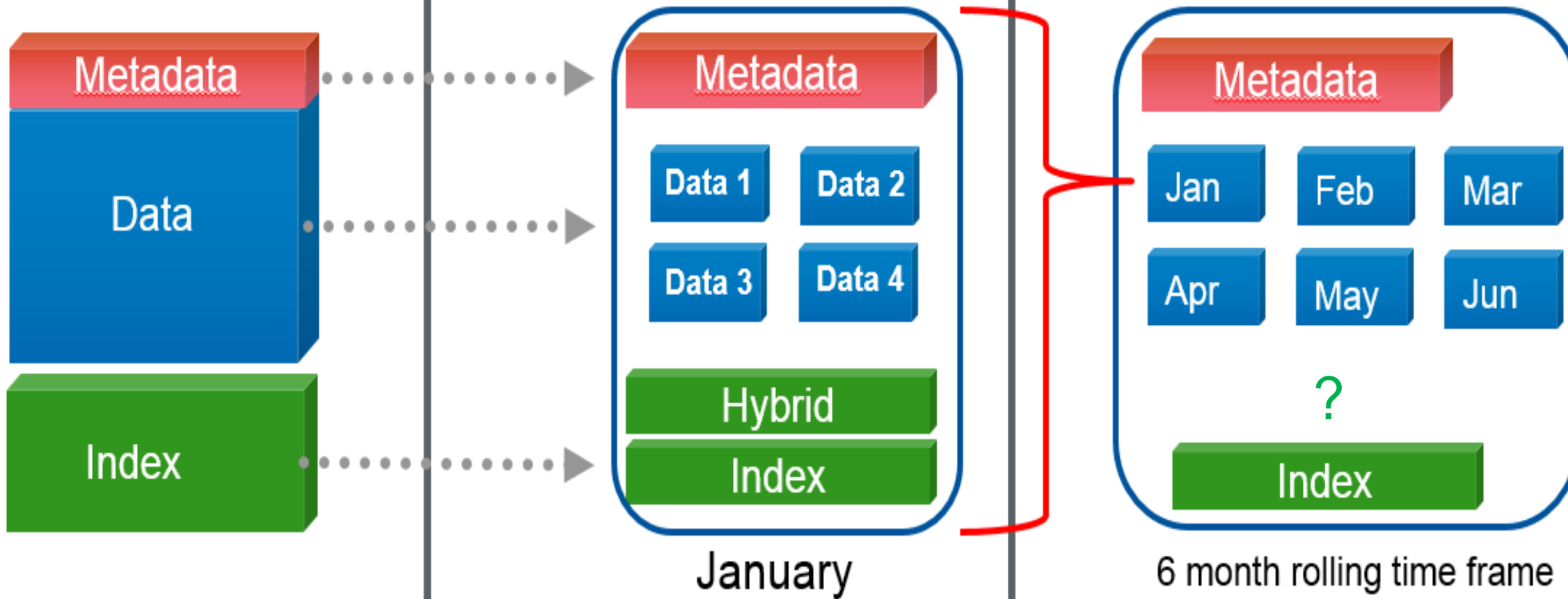
PRZYPOMNIENIE ZALET SPDS-A

DYNAMICZNY KLASTER

SAS data set

SPD Engine/Server Table

SPD Server Cluster Table



WERSJA 5.1 DYNAMICZNY KLASTER – UŁATWIENIA W ZARZĄDZANIU

- Zapewnienie dostępności klastra 24/7 – nie wymaga rozpięcia:
 - dodawanie membrów
 - zastępowanie membrów
 - usuwanie membrów
- Ilość slotów w klastrze może wzrastać dynamicznie wraz z dodawanymi membrami klastra
- Klaster może zostać w całości usunięty wraz ze wszystkimi membrami

WERSJA 5.1 MONITOROWANIE OPTYMALIZATORA JOIN'ÓW

- `%let spdssqlr=_method details='what_joinwhy_join';`
- Przykładowy fragment loga:

```
**WHY_JOIN( 1?): Plan a Inner Join
**WHY_JOIN( 1?): INDEX available on 2 tables
**WHY_JOIN( 1?): Index Join pass 1
**WHY_JOIN( 1?): Inner table [X0000001].FACT_CLAIM (alias = fct) Index agent_mk
**WHY_JOIN( 1?): Idx dup_ratio(0.00) < indexselectivity(0.70)
**WHY_NIDX( 1?): No 2nd pass, outer table too big
**WHY_NIDX( 1?): No useful index found
**WHY_MERG( 1?): Index join not selected, do merge join
**WHY_JOIN( 2?): Plan a Inner Join
**WHY_JOIN( 2?): INDEX available on 1 tables
**WHY_JOIN( 2?): Index Join pass 1
**WHY_JOIN( 2?): Inner table [X0000001].DIM_CUSTOMER (alias = c) Index
customer_mk
**WHY_JOIN( 2?): Idx dup_ratio(1.00) >= indexselectivity(0.70)
**WHY_INDX( 2?): magic = 104 (jtech_pref=index) and good dup_ratio)
SPDS_NOTE: PROC SQL planner chooses indexed join.
**WHY_JOIN( 3?): Plan a Inner Join
**WHY_JOIN( 3?): INDEX available on 1 tables
**WHY_JOIN( 3?): Index Join pass 1
**WHY_JOIN( 3?): Inner table [X0000001].DIM_INSURANCE_POLICY (alias = p) Index
insurance_policy_mk
**WHY_JOIN( 3?): Idx dup_ratio(1.00) >= indexselectivity(0.70)
**WHY_INDX( 3?): magic = 104 (jtech_pref=index) and good dup_ratio)
SPDS_NOTE: PROC SQL planner chooses indexed join.
**WHY_NHSH( 1?): Insufficient hash memory, increase buffersize
**WHY_NPLL( 1?): Join does not meet pllj requirements
**WHY_MERG( 1?): no transformation to hash or pllj
SPDS_NOTE: PROC SQL planner chooses merge join.
```

vs we wcześniejszych wersjach

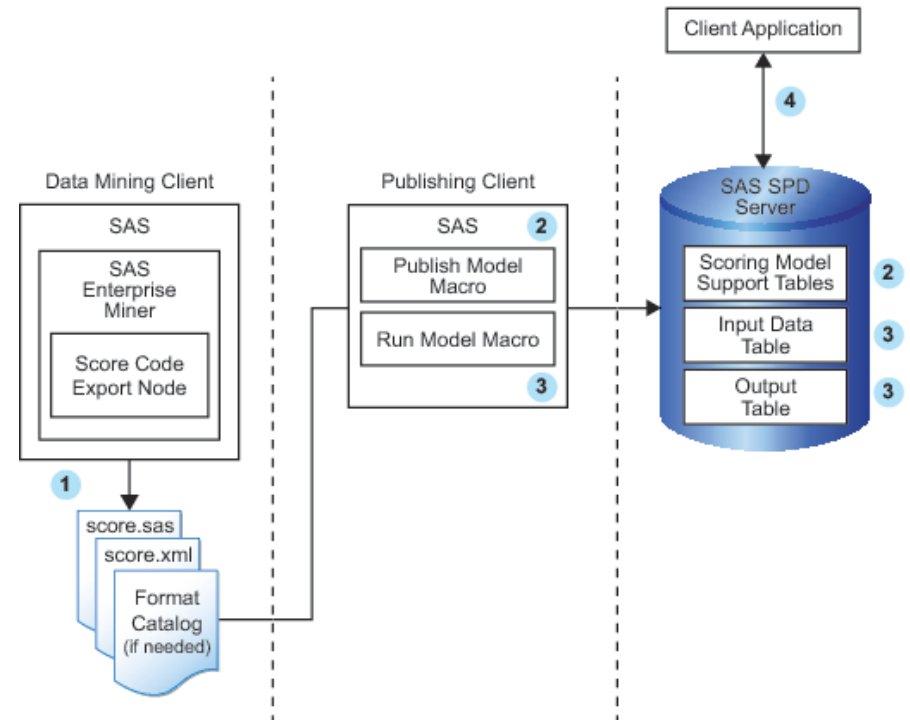
```
SPDS_NOTE: SQL execution methods chosen are:
  sqxslct
    sqxuniq
      sqxjndx(3)
        sqxjndx(2)
          sqxjm(1)
            sqxsrc ([X0000001].FACT_CLAIM (alias = fct) )
            sqxsort
              sqxfil
                sqxsrc ([X0000001].DIM_AGENT (alias = a) )
                sqxsrc ([X0000001].DIM_CUSTOMER (alias = c) )
          sqxfil
            sqxsrc ([X0000001].DIM_INSURANCE_POLICY (alias = p) )
```

KIERUNKI ROZWOJU KTO I CO JE KSZTAŁTUJE?

- Użytkownicy
- Ciśnienie” na nowe funkcjonalności
 - Wydajność i funkcjonalności lepsze niż BASE w prawie każdej sytuacji (ip=YES)
 - Wydajność ACL-i
 - NLS
- Łatwość użycia i redukowanie ograniczeń
 - Użytkownik nie zastanawia się nad typem indeksów
 - Administracja i monitorowanie
- Integracja z SAS-em
- Trendy w SAS-ie
 - Korzystanie z Hadoopa
- Bezpieczeństwo

KIERUNKI ROZWOJU INTEGRACJA Z SAS-EM

- Wsparcie na systemy operacyjne
- Proces instalacji
 - nie wymaga planu
- Metadane
- Scoring Accelerator
- I inne



SAS SPDS 5.2 PRZEDE WSZYSTKIM Z HADOOP'EM

- Możliwość odczytu, zapisu i modyfikacji tabel przechowywanych również na Hadoop'ie
- Dostęp do Hadoopa „przezroczysty” dla użytkowników
- Możliwość uwierzytelniania Kerberosem
- [Aktualne wsparcie dystrybucji](#)
- [Z ostatniego Global Forum: How to Leverage the Hadoop Distributed File System as a Storage Format for the SAS® Scalable Performance Data Server ...](#)
- Zrównoleglony odczyt też bez warunku where
- Optymalizacja WHERE możliwa po stronie Hadoopa (przez MapReduce)

SAS SPDS 5.2 PRZYKŁADOWY KOD

```
%let spdswdeb=YES;
```

```
105 +DATA MYSPDS.&source.3;
```

```
106 + SET MYSPDS.&source;
```

```
107 + WHERE joinvalue < 3;
```

```
whinit: WHERE (joinvalue<3)
```

```
whinit: wh-tree presented
```

```
/-NAME = [joinvalue]
```

```
--CLT----|
```

```
\-LITN = [3]
```

```
WHERE processing Hadoop cluster optimization candidate
```

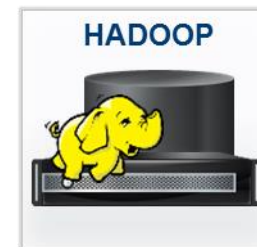
```
whinit returns: ALL EVAL6
```

```
108 +RUN;
```

WHERE processing is optimized on the Hadoop cluster

NOTE: There were 9375485 observations read from the data set SOURCE.TESTDATA.

```
WHERE joinvalue<3;
```



SAS SPDS 5.2 WYBRANE INNE FUNKCJONALNOŚCI

- CLUSTER LIST OUT=<data set>

The SAS System

The SPDO Procedure

Cluster List				
Cluster Name	Member Name	Variable Name	Minimum Value	Maximum Value
USSALES	NE_REGION	STORE_ID	1	20
USSALES	SE_REGION	STORE_ID	60	70
USSALES	CENTRAL_REGION	STORE_ID	60	70

- Dodatkowe zabezpieczenie przed włamaniami nieznanymi klientów
- Autentykacja przez SAS Metadata Server i LDAP
 - Jedno hasło
- Współdzielenie procesów proxy (spdsbase) użytkownika w sesji SAS
 - Zmniejszenie liczby procesów spdsbase w systemie operacyjnym

SAS SPDS 5.3 ZAPOWIEDZI

- Zabezpieczenie komunikacji sieciowej z klientami.
- Szybsze odczyty zawartości biblioteki
 - też przy proc datasets czy z dictionary.tables
 - optymalizacja dla ACL-i
- Dalszy rozwój współpracy z Hadoop'em (rozwój też w SPDE)
- Wsparcie dla nowszych wersji i dodatkowych dystrybucji Hadoop'ów
 - Cloudera: 5.4
 - Hortonworks: 2.3
 - dodanie MapR 5.0



5.3

- Wsparcie dla Log4Sas
- Przekazywanie warunku where do klastra na Hadoop'ie

Przyszłe:

- Dalsza integracja i wsparcie dla następnej wersji SAS-a
- Realizacja oczekiwań klientów



DZIĘKUJĘ