

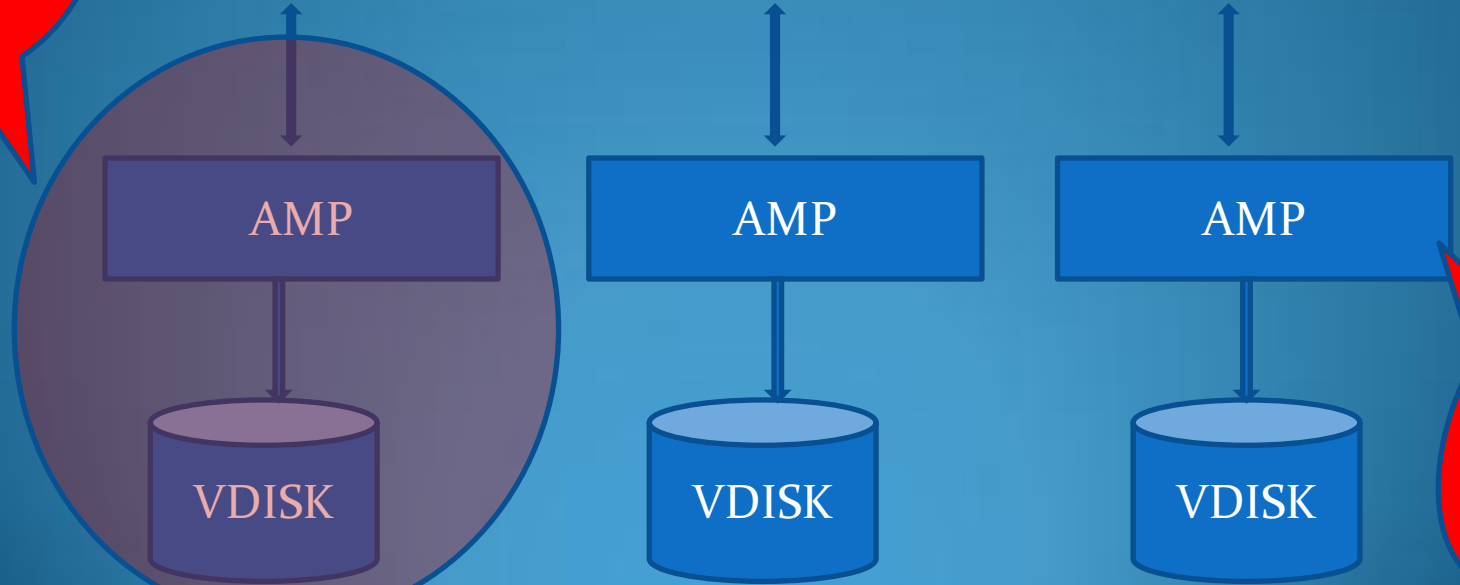
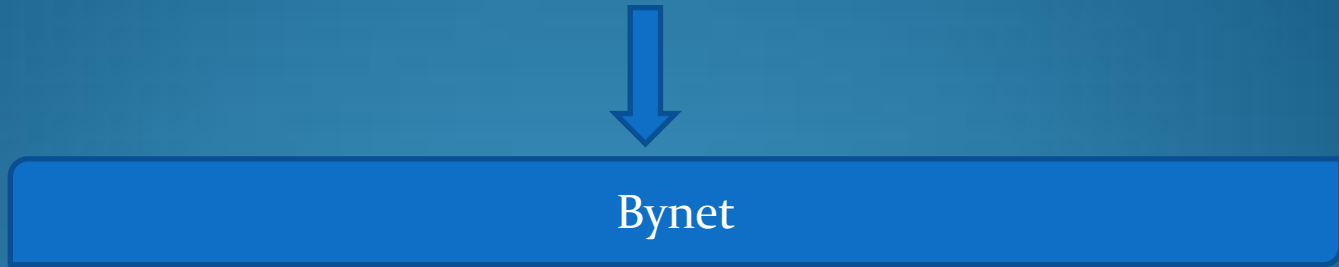
*How to
Efficiently Load/Query Teradata
Using SAS*

Suren Selvaratnam -CBA

Agenda

- A quick Intro on Teradata architecture
- Few techniques to get best out of both

A Minute Intro About Teradata Architecture



Share Nothing

Primary Keys

Few techniques to get best out of both

Selecting an appropriate primary index when loading data

- Know your data before you move data into Teradata
- Always define a Primary index when you load data into Teradata
- Never, ever assume Teradata will handle the primary key generation efficiently.
- If no primary key is defined, Teradata will pick the first column

Using Load Techniques

FAST LOAD

- Fastload is used to load empty tables and is “very fast”
- Ideal if creating a new table and cannot append to a table that already contains data.
- Attempting to use Fastload when appending to a table that contains rows, will end with error message.
- FastLoad does not load duplicate rows

Using Load Techniques

MULTILOAD

- MultiLoad loads both empty and existing Teradata tables
- Multi Load can load duplicate rows
- Drop the following items on the target tables before the load (unique secondary indexes, foreign key references, join indexes)
- SAS must be able to locate the Teradata MultiLoad utility on your system.

Using Load Techniques

Teradata Parallel Transporter (TPT)

- Provides a full-featured application programming interface (API) that enables you to access the Teradata loading feature
- Available only from SAS 9.2
- Requires CLIV2 library , TTU and Teradata Parallel Transporter API

Using Load Techniques

Teradata Parallel Transporter (TPT) – Advantages

- **Future enhancements**
- **Consistent interface**
- **TPT API libraries are loaded directly into SAS**
- **Simple configuration**
- **Control over utility slot usage**
- **No directory security issues**
- **Integration with Teradata Workload Manager**

Sample Code

```
%let gdwuser=selvasur@LDAP;
%let gdwpword=;
*libname PTEMP clear;
*assign Teradata library;
LIBNAME PTEMP teradata username="&gdwuser." password="&gdwpword." database = PTEMP
SERVER="Teradata.gdw.myserver";
* Create local temporary database;
data test_fastload;
array x (100);
do i=1 to 100;
    x(i)=ranuni(i);
end;
do i=1 to 1000000;
    output;
end;
Run;
```

Sample Code

```
/* Create Table layout in Terdata */  
data PTEMP. test_fastload (dbcreate_table_opts="PRIMARY INDEX(i)");  
    set test_fastload (obs=0) ;  
run;
```

```
/* Copy the data from SAS to Teradata using Fastload */  
options obs=max;  
proc append data= work. test_fastload  
    base= PTEMP. test_fastload (fastload=yes Sessions= 3);  
run;
```

```
/* Copy the data from SAS to Teradata using Multiload */  
proc append data= work. test_fastload  
    base= PTEMP. test_fastload (Multiload=yes Sessions= 3);  
run;
```

```
/* Copy the data from SAS to Teradata using TPT */  
proc append data= work. test_fastload  
    base= PTEMP. test_fastload (<Multiload> or <Fastload>=yes TPT=yes);  
Run;
```

Common SAS error message when using Fastload

NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.

ERROR: Fastload could not allocate data sessions: Too many load/unload tasks running: try again later.

NOTE: The data step has been abnormally terminated.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: There were 1 observations read from the data set
WORK.CBSDM_COMMSEE_DC_CLIENT.

WARNING: The data set PTEMP.my_TABLE may be incomplete. When this step was stopped there were 0 observations and 2
variables.

ERROR: ROLLBACK issued due to errors for data set PTEMP. my_TABLE.DATA.

NOTE: DATA statement used (Total process time):

real time	3.20 seconds
user cpu time	0.09 seconds
system cpu time	0.00 seconds
Memory	275k

Reference

Fast Load and Multiload

<http://support.sas.com/resources/papers/teradata.pdf>

Teradata Parallel Transporter

<http://support.sas.com/resources/papers/proceedings11/142-2011.pdf>

Thank You