Use Hash to reduce run time

Xiaotian Wu, PPD
Yu Zhu, PPD

PharmaSUG China 2016
Paper #79

Outline

- 1. Introduction
- 2. What's Hash?
- 3. An example
- 4. Conclusion

Introduction

Processing extremely large datasets can cost huge amount of time and effort, and therefore will cause an impact on delivery timeline. Even simple DATA steps and PROC SORT procedures can be problematic and time-consuming.

Introduction

Common techniques used to reduce run time:

1. Use compress=yes option
2. Use tagsort
3. Replace intermediate data steps with SAS views
4. Replace SQL joins with sort/merges
5. Use formats instead of merges where possible
6. Use “if – else” rather than “if – if”, “select – when” rather than “if – else” , when possible

Today we’ll focus on HASH object

Hash Object

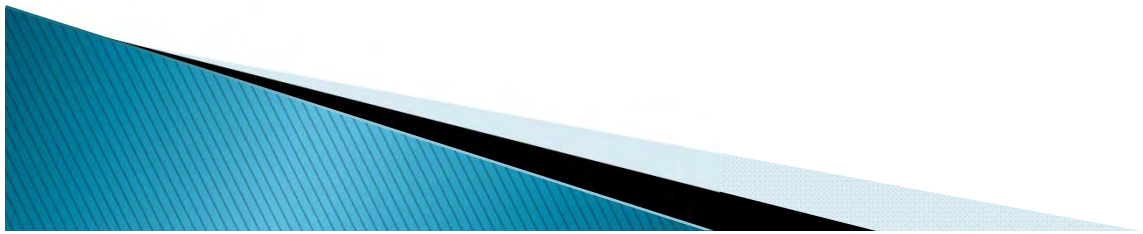
Accoring to *Secosky and Bloom (2007, p.1),*

A SAS hash object is: an in-memory lookup table accessible from the DATA step. A hash object is loaded with records and is only available from the DATA step that creates it. A hash record consists of two parts: a **key part** and a **data part.** The key part consists of one or more character and numeric values. The data part consists of zero or more character and numeric values.

Hash Object

- It can be thought of as a type of array that a program can access using keys. The programming language applies a hash function that maps the keys to positions in the array.

- SAS hash objects exist only in data step, not any other procedures. Therefore, when data step ends, SAS deletes the hash object.



Syntax

HASH is always performed in a data step and called by a certain method. Usually, we first declare a hash object by:

```
DECLARE HASH ALEX (DATASET: 'WORK.DIARY' )
```

where the dataset dairy is assigned to a hash object called Alex. Then a method is called by:

```
ALEX.METHOD ('Titles')
```

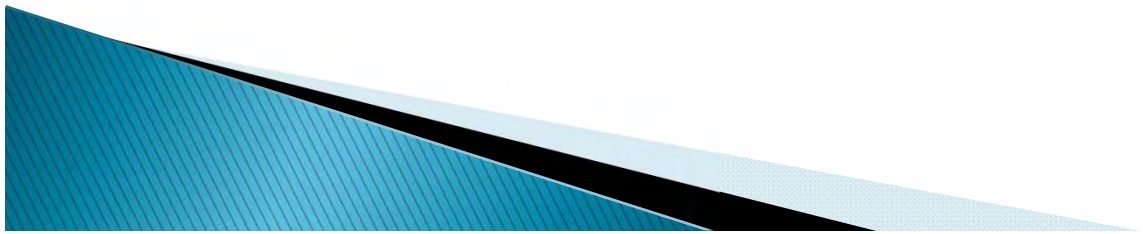
where method can be definekey, definedata, etc in the pool of 26 already identified methods. 'Titles' is the specification passed to the method.

An example

- In this section, a small merging example will be performed using

1. sort/merge;
2. tagsort/merge;
3. proc sql;
4. hash object.

- The example dataset we are using is an asthma questionnaire dataset with more than 3,000,000 observations and 1.6GB in size. The purpose is to merge a baseline flag to the original dataset by subject, parameter and date.



1. Proc sort/merge

```
proc sort data=qs1;by subject parameter date;run;
proc sort data=flag(keep=subject parameter date flag);by subject parameter date;run;
data qs2;
merge qs1(in=a) flag;
by subject parameter date;
if a;
run;
```

>5min

2. TAGsort/merge

```
proc sort data=qs1 tagsort;by subject parameter date;run;
proc sort data=flag(keep=subject parameter date flag) tagsort;by subject parameter date;run;
data qs2;
merge qs1(in=a) flag;
by subject parameter date;
if a;
run;
```

~3.5min

3. Proc SQL

```
PROC SQL;
CREATE TABLE qs2 AS
  SELECT a.*, b.flag
  FROM qs1 a LEFT JOIN flag1 b
  ON a.subject=b.subject and  a.parameter=b.parameter and a.date=b.date;
QUIT;
```

~1.5min

4. Hash object

```
data qs2;
if _n_=0 then set flag;
if _n_=1 then do;
   declare hash flag(dataset:'work.flag');
   flag.definekey('subject','parameter','date');
   flag.definedata('flag');
   flag.definedone();
end;
set qs1;
if flag.find() ne 0 then call missing(flag);
run;
```

Step 2

Step 3

Step1

Step 4

Step 1

```
data qs2;
if _n_=0 then set flag;
if _n_=1 then do;
  declare hash flag(dataset:'work.flag');
  flag.definekey('subject','parameter','date');
  flag.definedata('flag');
  flag.definedone();
end;
set qs1;
if flag.find() ne 0 then call missing(flag);
run;
```

Step 2

```
data qs2;
if _n_=0 then set flag;
if _n_=1 then do;
  declare hash flag(dataset:'work.flag');
  flag.definekey('subject','parameter','date');
  flag.definedata('flag');
  flag.definedone();
end;
set qs1;
if flag.find() ne 0 then call missing(flag);
run;
```

Step 3

```
data qs2;
if _n_=0 then set flag;
if _n_=1 then do;
  declare hash flag(dataset:'work.flag');
  flag.definekey('subject','parameter','date');
  flag.definedata('flag');
  flag.definedone();
end;
set qs1;
if flag.find() ne 0 then call missing(flag);
run;
```

Step 4

```
data qs2;
if _n_=0 then set flag;
if _n_=1 then do;
  declare hash flag(dataset:'work.flag');
  flag.definekey('subject','parameter','date');
  flag.definedata('flag');
  flag.definedone();
end;
set qs1;
if flag.find() ne 0 then call missing(flag);
run;
```

1min 28sec

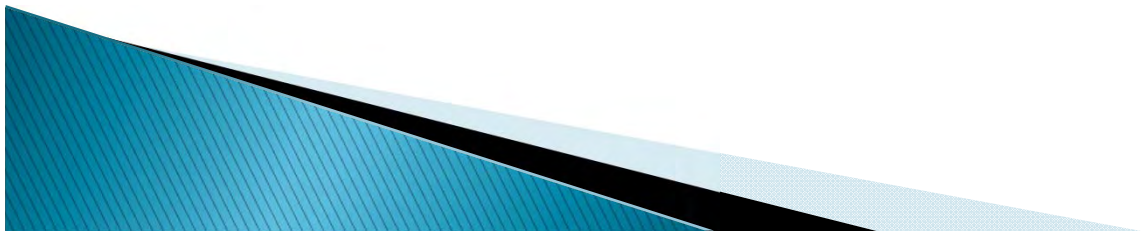
Conclusion

Hash table is time saving in two ways:

- There is no need to resort data;
- One can create the data step as a SAS view so no datasets will be output which saves time for intermediate datasets (so much more efficient than sorting where need to read in and write out data)

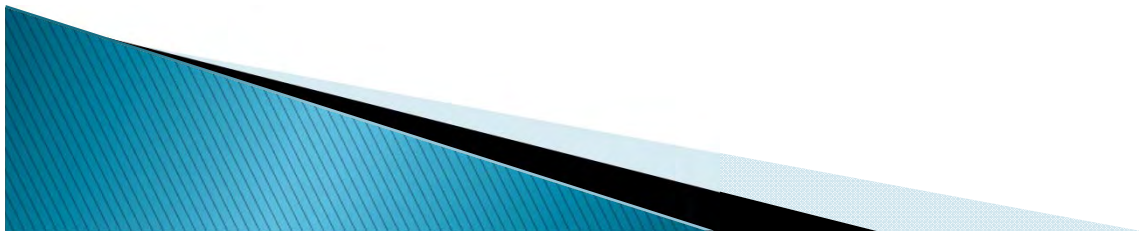
Conclusion

Hash programming has been a popular topic among SAS users since SAS 9 released. It is an efficient ands powerful way to do data look-ups, sorting, merging, etc. Plenty of resources and materials can be found out there if you would like to take a further step into hash programming.



References

- Getting Started with the DATA Step Hash Object. Secosky, J. & Bloom, J. Proceedings of SAS Global Forum 2007. Cary, NC: SAS Institute, Inc.
- Working with Big Data – Five Techniques To Reduce Run Times, David Mottershead, PPD
- SAS Hash Object Programming Made Easy, Michele M. Burlew, 2012, SAS Institute.
- Introduction to SAS® Hash Objects, Chris Schacherer, 2015, Clinical Data Management Systems, LLC



Contact infomation

Name: Xiaotian Wu
Organization: PPD
Address: 5 Corporate Avenue, 150 Hubin Rd
City, State ZIP: Shanghai
Work Phone: 021-80135116
E-mail: xiaotian.wu@ppdi.com
Web: www.ppdi.com

Questions?