ABAP Code for Parallel Cursor - Loop Processing

Author: Karthik C Sunil
Submitted: 18-Aug-2007
Related Links:

Abstract

This page will explain how to code very efficiently for nested un-related tables. The article describes how to improve the performance of processing huge data amounts.

Problem Description

The most common performance problem that occurs in ABAP programs is because of huge number of records in the internal tables. The problem complexifies if a program has huge nested internal tables. How much ever efficient data select routines are, data processing routines would be contributing significantly for the bad performance. When analysed it would be revealed that the where condition that is used in inner loops expend a significant amount of processing time. The idea is to avoid where conditions in the inner loops by maintaining the loop indexes manually.

Conventional Code for nested loops

```
Error rendering macro 'code': Invalid value specified for parameter 'lang'

loop at lt_vbpa into wa_vbpa.
   loop at lt_kna1 into wa_kna1 where kunnr = wa_vbpa-kunnr.

****** Your Actual logic within inner loop *****

endloop.
endloop.
```

Code sample: Parallel Cursor method

```
Error rendering macro 'code': Invalid value specified for parameter 'lang'

sort: lt_vbpa by kunnr,  "Sorting by key is very important
   lt_kna1 by kunnr.  "Same key which is used for where condition is used here
loop at lt_vbpa into wa_vbpa.
   read lt_kna1 into wa_kna1     " This sets the sy-tabix
      with key kunnr = wa_vbpa-kunnr
      binary search.
   if sy-subrc = 0.              "Does not enter the inner loop
      v_kna1_index = sy-tabix.
      loop at lt_kna1 into wa_kna1 from v_kna1_index. "Avoiding Where clause
         if wa_kna1-kunnr <> wa_vbpa-kunnr.  "This checks whether to exit out of loop
            exit.
      endif.

****** Your Actual logic within inner loop *****

   endloop. "KNA1 Loop
   endif.
endloop.  " VBPA Loop
```

Statical Analysis

Nested loop for BSEG and BKPF internal tables were analysed for Conventional Method and Parallel Cursor methods. Following graph explains the observations.
Observation: One can observe that as the data increases, the time taken for the nested loop increases drastically, at the same time, the Parallel cursor method did not suffer any considerable time impact.

Verdict: Use the parallel cursor method whenever there is a need to process data in a nested loop.