Performance Analysis and Optimization in period end closing

- **Purpose**
  - **Introduction**
  - **First Analysis Steps**
    - General Performance Analysis
    - Job Logs, Spool Lists and Schedule Monitor
  - Performance Relevant Parameters
    - Object Selection
      - Internal orders
      - Logistical orders: Production orders, process orders, product cost collectors
      - Known Performance Issues with SQL statements for DB selection of logistical orders
    - Master data and status checks
    - Status checks
    - Schedule Manager and Work Lists
    - Transaction data
    - Parallelization
    - Locking problems
  - Analysis of long running jobs
    - Find the bottleneck
    - SAT, SE30 Runtime Analysis in parallel session
    - Important Transaction Codes
  - Related Content
    - Related SAP Notes/KBAs

**Purpose**

This document provides an overview on how to analyze performance issues with OBJ period end closing transactions. It will give a guideline for the processing of performance issues and the usage of the most common performance analysis tools. It also includes an overview of notes, known program errors, performance issues and bottlenecks.

**Introduction**

CO-PC-OBJ offers a number of period end closing transactions/reports, as for example surcharge/overhead calculation, WIP calculation, results analysis, variance calculation, settlement. These period end closing activities can be started for different kinds of objects: internal orders, logistical orders (production orders, process orders), projects/WBS elements, sales orders, maintenance orders, and more. The transactions and report names depend on these objects. There are transactions for single object processing and for collective/mass processing.

Customers often complain about performance issues and long run times. Long processing times can have different root causes:

- General performance problems on the customer system
- Big number of objects to be processed
- Organizational problems on customer side
- Complex business scenarios
- High document/data volume
• Program errors: problematic select statements, unnecessary loops, inappropriate data processing, unnecessary program parts, bad programming
• Database problems, very big DB tables, missing indexes, old DB statistics
• Customer Exits
• Handling Issues, inappropriate usage of transactions/report parameters

In general, it is recommended to do a performance analysis in the sequence

1. First steps: check job logs, Schedule Monitor, general information from end user
2. Runtime Analysis
3. Debugging

First Analysis Steps

If an end user reports a performance issue, the first steps should be to find out which of the above root causes may be involved. The end user should answer the following questions:

• Are there specific performance problems with one or more transactions/reports, e.g. collective settlement of production orders CO88? Or is there a general performance issue with many transactions which indicates problems with system settings, system resources, general system performance?
• Did the runtimes suddenly get longer, e.g. in comparison with the runtimes in the last period or before an upgrade or support package implementation? Or did the runtime gradually getting longer in the last periods/years? Can it be explained with a gradually growing number of objects and documents (transaction data volume, size of DB tables,…) on the system?
• Have new business processes, functionalities, changes in customizing or other changes been implemented which could have resulted in the longer runtimes?
• Have bottlenecks already been identified?
• Is there a performance problem already in test run, or only in update run?
• Is there already a performance problem in a single processing transaction?

General Performance Analysis

In case of general performance issues on the system, the incident should be handled by COE or basis experts, Early Watch, etc. They have much better knowledge about the system monitor and performance tools.

Job Logs, Spool Lists and Schedule Monitor

If a report has already been started and finished in background processing and the job name, user or date is known, the job log and the spool list should be checked (Transaction SM37).

Example: Settlement Job
The job log already shows a lot of useful information:

- Total runtime of the job
- How much time took the sender selection?
- Is the processing time getting longer for the same percentage of senders processed?

If e.g. a performance problem already occurs with the sender selection (object selection), a long time will have passed between the message ‘Sender selection…..’ and the message ‘x Sender selected’. Information about known problems with object selection can be found in the chapter ‘Object Selection’.

If the processing time is getting longer for the same percentage of senders processed, this could indicate a problem with growing internal tables, loops etc.

The Spool List should at least show a basic list with some important information.
Here the Processing category statistics is important. The statistics shows that most of the objects had inappropriate status or category ‘No change’. From 2804 selected objects, only 46 objects have been settled. In such a case, it should be checked if it is really necessary to select all these objects.

The **Schedule Manager Monitor**, transaction code SCMO, should be used to check older job jogs, compare runtimes of recent jobs and find general information about how the period end closing is scheduled at your end.

Tipp: The Schedule Manager Monitor stores information about all CO-PC-OBJ period end closing jobs (and other programs which have been connected) even if you don’t use the Schedule Manager or even know that it exist! The Schedule Manager Monitor also keeps much information of jobs for which an SM37 job log is no longer available because it has already been deleted!

Example: Schedule Manager Monitor  SCMO
### Schedule Manager Monitor: General Data Selection

**Execute**

**Application**

<table>
<thead>
<tr>
<th>Field</th>
<th>Fr.Value</th>
<th>To value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Fr.Value</th>
<th>To value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>End Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>End Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Name</td>
<td>RK070D00</td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Run</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worklist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Struct. tree, left
- Struct. tree, up
The Schedule Manager Monitor can be used to find out if jobs with comparable number of objects etc. have had shorter run times in the past. It gives information about how the run times changed, how the number of selected objects has changed etc.

**Performance Relevant Parameters**

Most CO-PC-OBJ reports have the program logic:

1. Selection of objects
2. Master data and status checks
3. Processing of the objects: selection of transaction data (totals, line items,…), program internal calculations (e.g. WIP calculation, variance calculation,…), comparison of old and new data (is there a difference between the value of the previous period and the value of the current period?), preparation of documents (if something has changed, the data will be updated).
4. Generation and posting of CO documents and follow-up documents

The runtime of a period end closing report depends on

- the input data volume (which objects are selected for processing, how much transaction data is selected for the objects),
- the processing complexity and
- the output data volume (how much data is generated by processing the object).
Object Selection

CO-PC-OBJ reports can process different types of objects. The object selection logic and the function modules for the object selection depend on the objects.

Long runtimes are often caused by the selection of too many old objects. Old orders which are not used anymore, but still exist on the system, are selected and processed. The period end closing programs check the status of the selected objects very early during processing. Processing for orders with status ‘Created’ (CRTD, system status I0001), ‘Closed’ (CLSD, system status I0046), or ‘Deletion Flag’ (DLFL, system status I0076) is stopped then. However, the status checks take time, and if e.g. 10000 objects are selected, and 9000 of these objects have inappropriate status, the program spends a lot of time with the status checks. This can be avoided when old objects are deleted and archived frequently. A deletion flag can be set and revoked any time for old objects, and depending on the object type, objects with deletion flag will not be selected at all from the DB tables. This will save the time for the status check then.

Different period end closing reports can have different logics regarding the system status.

Internal orders

Order category AUFK-AUTYP = 01. Master data is completely stored in DB table AUFK. A selection variant for KO8G, RKO7KO8G is necessary. Selection variants offer various possibilities to exclude orders from being selected. Status selection profiles can be used to exclude orders with defined system or user status from selection. (Known Issues with selection variants, status selection profiles?)

Orders with deletion indicator (AUFK-LOEKZ) will be selected from the database, but processing will stop after status check.

Internal orders can be closed or flagged for deletion in collective processing with transaction KOK4 or KOK2.

Logistical orders: Production orders, process orders, product cost collectors

Order categories AUFK-AUTYP 05, 10, 40. Master data is in AUFK, AFKO and AFPO.

The period end closing transactions for logistical orders are usually started for a plant. Some reports offer more detailed selection options. E.g. settlement report RKO7CO88 can be started for selected order types or order numbers/ranges.

System status and DB selection logic

- Variance calculation: SAPMKKS0 FORM VIEW_VKKS0_LESEN. Note 66612 orders with status CLSD (I0046), LOCK (status I0043), or which are flagged for deletion (field AUFK-LOEKZ set, status DLFL I0076), or have a deletion indicator set (status
DLT I0013) will NOT be selected from the database.
- WIP calculation, Settlement, Overhead Calculation: orders with deletion flag (field AUFK-LOEKZ, status DLFL I0076) are NOT selected from the database.

The deletion flag for production orders can be set with transaction CO78 (Archiving of production orders).

SAP Note 306576: COMPOSITE NOTE: performance overheads

SAP Note 393686: INFO: CO-PC-OBJ (Performance)

SAP Note 2420801: Status selection of period-end closing reports for production orders

**Known Performance Issues with SQL statements for DB selection of logistical orders**

There is a known performance problem with joined selection from the master data tables. If object selection takes hours or takes significantly longer after an upgrade or basis support package, the problem could be in LCOSEF01 FORM ORDER_OBJNR_GET. The following select statement can take very long:

```sql
select (lt_dbfields)
    FROM ( ( afko INNER JOIN aufk
        ON afko~aufnr = aufk~aufnr )
    LEFT OUTER JOIN afpo
        ON afko~aufnr = afpo~aufnr )
    INTO CORRESPONDING FIELDS OF TABLE lt_order_sel
    FOR ALL ENTRIES IN lt_maufnr
    WHERE afko~maufnr = lt_maufnr-aufnr
    AND loekz IN lr_loekz
    AND pkosa = space
```

This problem can be fixed with database specific hints (also release dependent) or by replacing the select statement above with separate selections. The following notes could be applied:

SAP Pilot Note 2073728: CO Period-End Tasks: Performance issue with Sybase ASE
SAP Pilot Note 2001405: CO period-end closing: Performance problems with Oracle DB
SAP Pilot Note 1664736: CO43: Overhead calculation terminates with short dump
SAP Pilot Note 1504067: Performance problems in overhead calculation and settlement
SAP Note 545932: Performance probl. in CO-PC-OBJ period-end closing: DB hints
SAP Note 587278: Performance problems in overhead calculation and settlement

Other known performance issues with select statements:

SAP note 1592568: Settlement: Selection with MAUFN takes a very long time
SAP note 740626: Cost Object Controlling: Performance Object selection
SAP note 2088295: COB Hierarchies: Performance in Period End Closing

**Master data and status checks**
When the object selection is finished, the objects are sorted and processed. Master data is checked: e.g. important indicators like ‘Statistical Order’, ‘Revenue Postings’, control keys for period end closing (Results Analysis key, Variance key, Costing Sheet). The result of these checks decides how the further processing of the object is done.

**Status checks**

The SAP general status management (function group BSVA) is a functionality which allows a transaction control. A business transaction (e.g. actual settlement KOAO, or actual overhead calculation KZPI,…) can only be carried out if no system status or user status forbids this transaction and if at least one system status allows the transaction. The program has therefore to check which system or user status is currently active for the processed object and if the business action is allowed.

A general overview about system status is provided with transaction BS23:

A double-click on a status displays the transaction control:
Status checks can be very time consuming if status buffer tables are not refreshed frequently in mass processing. If runtime analysis shows that much time is spent in function modules of function group BSVA (FM STATUS_CHECK, STATUS_CHECK_MULTI, STATUS_CHANGE_FOR_ACTIVITY…), performance notes should be searched with key words performance, status, status check, status buffer, STATUS_BUFFER_REFRESH together with the report and transaction code of the period end closing program.

If an object has a status which does not allow e.g. settlement or other transactions, there will be either an error message (status management message class BS or application message class) in the message list. Or, in some cases, processing for the object is stopped without error message, and there will only be a processing statistics on the result screen. In settlement, the processing category shows how many of the selected objects have processing category ‘Not relevant’ or ‘Inappropriate Status’ and have therefore not been settled. For these categories, there is no further information about the objects (order numbers etc.) and the status of these objects. The modification of note 1428700 ‘Enhancing the status information for the settlement’ offers more detailed information for the settlement programs.

If a long running job spends most of the time with object selection and status checks for objects which after all do not need to be processed, performance will be improved by better object
selection, deletion and archiving of old objects, and usage of status selection profiles.

**Schedule Manager and Work Lists**

**Transaction data**

The most important transaction data tables in CO are the CO document tables (COBK document header, COEP and COEPB document lines) and the CO totals tables COSS, COSP, and COBK. The totals tables summarize the many COEP line items which have identical keys. Almost all CO-PC-OBJ period end closing reports are based on the totals tables and select the costs from these tables. The number of total lines for an object therefore has several effects on performance:

- The cost selection from the totals tables gets slower when the overall size of the totals tables gets very big (millions of entries)
- The size of internal tables often depends on the number of selected totals. More data has to be processed, and loops at these internal tables take longer
- The follow up documents can get bigger (more document lines), and processing of the follow up documents takes longer

Performance issues can occur if the number of totals for an object (order, WBS Element,....) is very big. The number of totals depends on the contents of the key fields. The most relevant key fields which can cause a big number of totals are

- COSP-, COSS-, COSB-KSTAR: cost element
- COSP-, COSS-, COSB-HRKFT: origin (CO key subnumber) -> material number, funds, functional area,...
- COSS-PAROB: partner object

The number of different cost elements is usually not critical, in most customer scenarios the number of different cost elements which are involved for one object is about 10-30.

The number of different partner objects in COSS can get big if, for example, there are many different cost centers/activity types are involved in a production scenario, or if many different orders settle to single WBS elements.

The CO key sub-numbers (COKEY) is a key field which combines different original information. These are e.g. the fields in DB tables COKEY. These fields involve organizational units (functional area, fund, grant, segment, budget period, production month), or the material number. The organizational units are usually not critical because there is only a small number of different functional areas etc. .

In production scenarios, the material number can cause many COSP totals. In the material master data (transaction MM03), the view Costing 1 offers the indicator ‘Material Origin’. If this indicator is set, the material number will be updated in the CO subkey. Each combination of plant, material number and all of the other COKEY* fields will be converted into a number which will be updated in key field COSP-HRKFT. In production scenarios with many hundred different raw materials, the number of COSP totals can get very big.
This can especially cause big performance problems and huge documents in settlement. If the finished goods are delivered to different valuation segments or sales order segments, each of these segments is a different settlement receiver. The number of document lines in the settlement document is the product of the selected original total lines X number of settlement receivers. In extreme cases, this can result in a document line overflow or in a big number of splitted follow up documents.

If the number of total lines in the CO totals tables exceeds significantly several hundred lines per object number, performance problems for almost all CO period end closing reports can be expected.

If such big total line numbers are observed, it should always be checked with the customer if such a high level of detail is really necessary. In general, a customer should always take care that he does not work with critical settings which create a very high data volume.

SAP Note 612641: PCC settlement: Error F5727

SAP Note 718873: PCC settlement: Document overflow with error KD557

Parallelization

SAP generally recommends the usage of parallelization for the mass processing period end closing reports. This recommendation can be found in many CO performance notes. Parallelization will reduce the overall job run times in most business scenarios. However, in some scenarios, the usage of parallelization must be optimized, e.g. by

Parallel Processing is started by entering a server group when the report is started. Some reports have a parallel processing indicator; setting the indicator to ‘X’ will provide a field where the server group can be entered. Alternatively the server group can be entered in the background processing job parameters. If parallelization is used, the period end closing main program will start on the server where the user started the report. In the main program, all objects will be selected for processing. Then function module SPTA_PARA_PROCESS_START (function group SPTA) is called. The main program creates groups of objects with defined package size (usually ~200 objects), which are then processed in parallel on the servers of the server group. Server groups are maintained with transaction RZ12.

Depending on the system resources (number of servers, number of dialog processes on each server, hardware,…..), parallel processing can significantly reduce the overall processing times because many objects are processed in parallel instead of sequentially. However, there can be performance bottlenecks in the object selection, packaging logic or other program parts of the main program, or performance problems on the DB update server, or on the enqueue server. In such cases, parallelization will most likely not show the expected improvements.

Restriction of the number of work processes

In SAP standard, parallelization will use all available work processes of the servers. If there are too many work processes due to parallel processing, the system load can get too high. As a consequence, the number of processes for parallel processing should be restricted. This can be done in general as described in note 1812322. Application specific (Report specific) restrictions of the number of work processes are possible after implementation of note 1074098 (WIP).
calculation, Revaluation, Overhead Calculation, Settlement). With the solution of note 1074098, the customer can maintain the maximum number of work processes by creating entries in DB table T811FLAGS. This number will then be used as a maximum number of work processes for the relevant period end closing reports.

For variance calculation, note 1305475 implements the same solution. For Results Analysis, note 1444215 can be used.

For settlement, the modification of note 1173665 implements a more comfortable solution. The user can enter the number of dialog processes in the technical parameters (number of work processes) of the settlement report: Call transaction SE38, enter the settlement report RKO7CO88 (or RKO7KO8G, RKO7CJ8G, RKO7VA88), and choose "Technical Settings" on the selection screen of the relevant settlement report to display the technical settings. As of Release 6.04 Support Package 1, the modification of this SAP Note is part of the standard SAP system.

If note 1074098 is already implemented and note 1173665 should also be used, it may be necessary to implement note 1173671 before implementing 1173665 to make the two notes "compatible".

In newer releases, the solution of note 1173665 is available in SAP standard. If note 1074098 should also be used, it is possible to use both solutions for settlement. In that case, the settlement program will use the number of work processes for a settlement job from the technical settings of the settlement report with highest priority. If no number is maintained here, the number from table T811FLAGS is used.

**Locking problems**

The CO-PC-OBJ period end closing reports create lock entries when started in update run. Example: Settlement of orders creates a lock entry for the processed object number on DB table COBRA and on DB table AUFK. Locks may also be set for external settlement receivers (V price materials, assets).

**Lock Table overflows**

The number of objects which create lock entries at the same time depends on the number of parallel processes for each job, the number of objects which are processed by each work process (package size), and the number of jobs which are running at the same time. This can result in lock table overflows (note 746138). Transaction SM12 shows a list of the current lock entries. If the number of lock entries is too big, and the lock table configuration is already at a limit, the number of locks can be reduced by a smaller number of work processes (see above) and by reducing the package size for settlement with note 1512222. It should also be avoided to start too many independent parallel running jobs. A known issue which occurs with lock table overflows when too many orders settle to the same material is described in note 595823.

**WAIT/Deadlock problems**

If parallel running jobs/work processes lock the same objects (e.g. when several settlement senders which are processed in different work processes settle to the same receiver material or asset), the lock can be set only by the first enqueue request. The other objects will have to wait until the work process is finished and the lock is released again. This can result in a situation
where many processes can be seen in the SM50 process overview with semaphore ‘WAIT’. Such problems can also occur if there is a performance problem with the database updates. In extreme cases, a Deadlock situation occurs.

**Analysis of long running jobs**

As explained previously in this document, the runtime of a period end closing report depends on the input data volume (number of objects, transaction data volume), the processing complexity, and the output data volume (CO documents and follow up documents). In ‘normal scenarios’, the number of cost totals in the CO tables COSS, COSP, COSB is in a scale of 20-200 entries, and the follow up documents (FI, CO, SL, ...) have about 2 – 20 document lines each. In such a scenario, a period end closing program can process several hundred to thousand objects per minute. When parallelization is used, jobs for several hundred thousand objects can be finished in 1-3 hours.

The processing time of course depends also on the total size of the transaction data tables (if there are already millions of entries in COSP and COSS, cost selection takes longer, etc.), the hardware capacity, system load, etc…).

If certain CO-PC-OBJ period end closing jobs have very long run times which cannot be explained with the high number of objects or high data volume, and the general system performance is ok, there might be a bottleneck which has to be identified. In such a case, the first thing which should be checked is if object selection takes most of the time. If yes, the object selection chapter of this document should be checked. If object selection takes only a few minutes, but processing runs for hours and hours, a runtime analysis should be done for the running job. The questions in the chapter ‘First analysis steps’ should be answered first before starting a runtime analysis.

A SAT/SE30 Runtime Analysis for a single transaction is only necessary if there is already an example order which shows significant long processing time (> 10 seconds to minutes) which indicates a problem somewhere.

**Find the bottleneck**

In many cases, a simple runtime analysis is already sufficient to find problematic source code. This is shown in the following two Demo examples. If the problematic source code is identified, note search should be done with the name of the Include or the problematic form routine or function module. The note search may find the responsible note which has implemented this problematic source code, and – if it is already a known issue – a note with a correction to fix the performance problem. If the issue is not known yet (maybe the new source code is only problematic in special cases or scenarios which have not been tested yet), the information about the identified bottleneck is very important to route an incident directly to the responsible developer and get a fix within a short time.

Demo Example 1: SE30 for KO88, one order, with manipulated source code (artificial bottleneck in SAPLKO72)
Start SE30:
Press 'Execute Button', start transaction KO88 in test run, after test run return to SE30 and press 'Evaluate' Button:
Runtime Analysis shows that most time is spent in processing of ABAP coding. Press F5/Hit List Button:
If there is a simple program error or problematic source code, this should be found directly in the SE30 hit list if the hit list is sorted by the column Net (%). From here, menu ‘Goto -> Display source code’ directly displays the artificial bottleneck. In the demo example, function module MESSAGES_INITIALIZE is called 100,000 times.

Demo Example 2: SE30 for KO88, one order, with manipulated source code (artificial bottleneck in SAPLKO72, this time a problematic select statement which selects all existing COEP line items into an internal table):

![ABAP Editor: Modify Include LK072F30](image)

This will have the following effect for the runtime analysis:
The Net % time is the ABAP processing time only. Here the bottleneck is a database selection of a lot of data, which takes much Gross %.

Menu ‘Goto -> Display source code’ immediately displays the critical select statement.

If runtime analysis shows that most of the time is spent on the Database, as in demo example 2,
an SQL trace (transaction ST05) can be useful.

A ST05 SQL trace for the demo example 2 settlement shows the following list:

Shift+F8 (Summarize Trace by SQL statement) shows the problematic selection:
Menu Goto -> Display ABAP source will display the problematic select statement in the source code.

SAT, SE30 Runtime Analysis in parallel session

If the performance issue occurs only in mass processing, a runtime analysis can be started when the job is already running.
If SE30/SAT is started in Parallel Session, the SM50 Process overview will be displayed. Here the process of the running job must be found via user name and report name.
If the process is identified, the runtime measurement can be started by setting the cursor on the process line and press Shift + F7 (Start measurement button). Measurement should then be done for some time (about 30 s to 1 min). Then go back to the SE30 start screen and evaluate.

One single measurement may not be sufficient to get a representative runtime analysis. At the beginning of a mass processing run, object selection, pre-read of status, preparations etc. are done. It may take some time until the job is in the object processing loop. Nevertheless, if a runtime analysis shows a significant high Net % in the hit list which can be reproduced with a number of subsequent runtime analysis measurements while the job is running, the bottleneck is most likely identified.

**Important Transaction Codes**

SCMO  Schedule Manager Monitor  
SM37  Job Logs  
RZ12  RFC Server Group Maintenance  
SM12  Lock Entries  
SM50  Process Overview  
SE30  Runtime Analysis  
ST05  Performance Trace

**Related Content**
Related SAP Notes/KBAs

SAP KBA 1578574: Bad performance in Variance Calculation

SAP KBA 1580830: Bad performance in CO88 settlement transaction

SAP KBA 1991171: Status CLSD on orders does not improve performance