



Projeto de Otimização de Performance e Redução de Custos

Sistema On-line

Empresa de Serviços



Objetivo do Projeto

Comprovar a viabilidade de ganhos de performance e redução de custos em ambiente Mainframe de um novo sistema de Call Center.



Cenário

Robô simulando carga de 100 usuários com alguns cenários similares ao ambiente de Produção.

Premissas

- Focar sub-rotina XXX0024
- Definir e utilizar processo não-intrusivo
- Aplicar processo de otimização
- Escolher pontos a otimizar
- Efetuar as alterações propostas
- Efetuar medições prévias e posteriores
- Apurar os ganhos

Desafios

- Redução do time to market
- Volume crescente de implantações
- Manter desempenho adequado
- Manter disponibilidade
- Otimização rápida



Cenário em Produção

- Ambiente Mainframe Z/OS
- Aplicações CICS, Cobol, DB2
- Ambiente de Produção de 5000 MIPs

Tópicos do Projeto

- Escopo e processo utilizado
- Descrição Macro
- Dados Obtidos
- Resultados
- Premissas para cálculo da economia de MIPS
- ROI

Escopo e processo utilizado

- Focar a sub-rotina XXX0024: a mais utilizada no on-line
- 62% do uso de CPU nos testes de stress do novo sistema
- Mais de 80 comandos SQL de difícil avaliação manual
- Provar que o processo é viável para todo workload em DB2

Processo Escolhido

Geração de traces de performance do DB2 tendo como metas:

- evitar impacto de desempenho (amostragem)
- automação na análise dos traces
- detectar comandos de maior consumo
- extensa gama de informações
- propor alterações de índices (processo automático)
- propor otimizações de SQL de alto consumo

Descrição Macro

- Coletar amostras de traces de performance do DB2
- Aplicar processo automático de análise dos traces
- Recomendar automaticamente índices e alterações
- Teste de stress da situação anterior
- Efetuar alterações, teste de stress e medições na situação nova



Dados Obtidos

Tabelas mais acessadas e analisadas

XXXXXRMT
XXXXXCMT
XXXXXAMT
XXXXXOMT

Comandos de maior consumo (99%)

Stmt:	5638	XXXXXRMT	CUR_XXXXXRDV_SAD
Stmt:	2835	XXXXXOMT	CUR_CLI_AMBOS
Stmt:	6869	XXXXXCMT	CUR_IDPC_SAD
Stmt:	7642	XXXXXCMT	(select sem cursor)
Stmt:	2776	XXXXXADV	(select sem cursor)

Dados Obtidos

Solução – Tabelas (recomendação automática)

XXXXXRMT - ALTERAÇÃO DE 1 (UM) ÍNDICE

XXXXXCMT - CRIAÇÃO DE 1 (UM) ÍNDICE

XXXXXAMT - CRIAÇÃO DE 1 (UM) ÍNDICE

XXXXXOMT - CRIAÇÃO / DELEÇÃO DE ÍNDICES

Solução para otimizar comandos

Ganho Previsto

Stmt:	5638	(índice)	< 80%
Stmt:	2835	(índice/alterar SQL)	< 10 a 50%
Stmt:	6869	(índice)	< 75%
Stmt:	7642	(alterar SQL)	< 60%
Stmt:	2776	(índice / alterar SQL)	< 80%

Resultados

<u>OBJETO MENSURADO</u>	<u>ANTES</u>	<u>DEPOIS</u>	<u>%REDUÇÃO</u>
TEMPO DE RESPOSTA MÉDIO	0,231	0,103	55%
GRUPO DE TRN PE**	0,383	0,09	76%
GRUPO DE TRN VI**	0,187	0,072	62%
CPU TOTAL PQD0024	1224	3	99%

Resultados permitiram revisão do capacity previsto para o sistema em 30%

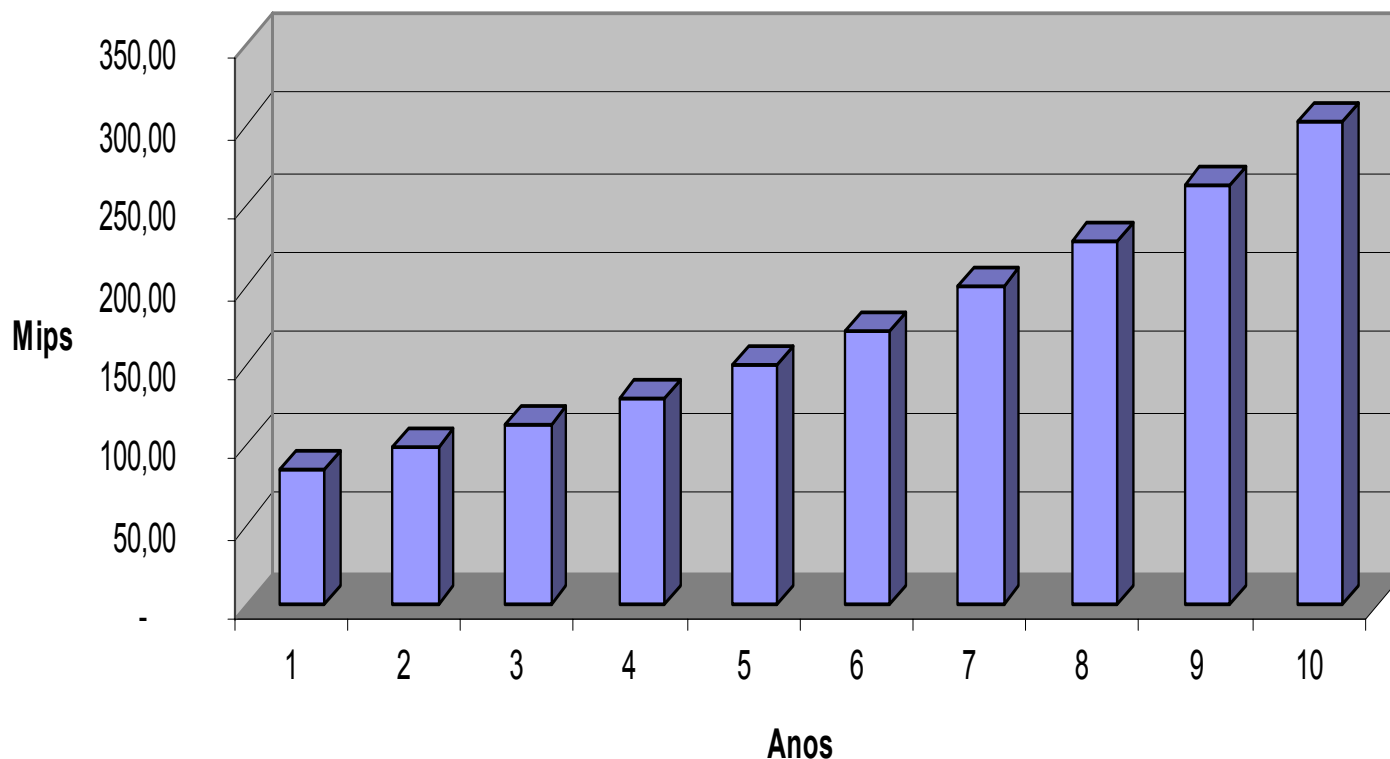
Premissas para cálculo da economia de MIPS

- O processador do ambiente é de 215 MIPS
- O perfil da carga de transações do stress é similar ao da Produção
- O teste de stress foi executado em 51 minutos. (13h53 às 14h44)
- A CPU disponível para um processador seria de 51 minutos, ou 3.060 segundos
- CPU economizada após alterações: 1.221 segundos
- Economia em percentual: 39,9% (1221 segundos de 3.060 segundos)
- Economia em MIPS: 39,9% de 215 MIPS, ou 85,78 MIPS
- Usuários simultâneos: 100

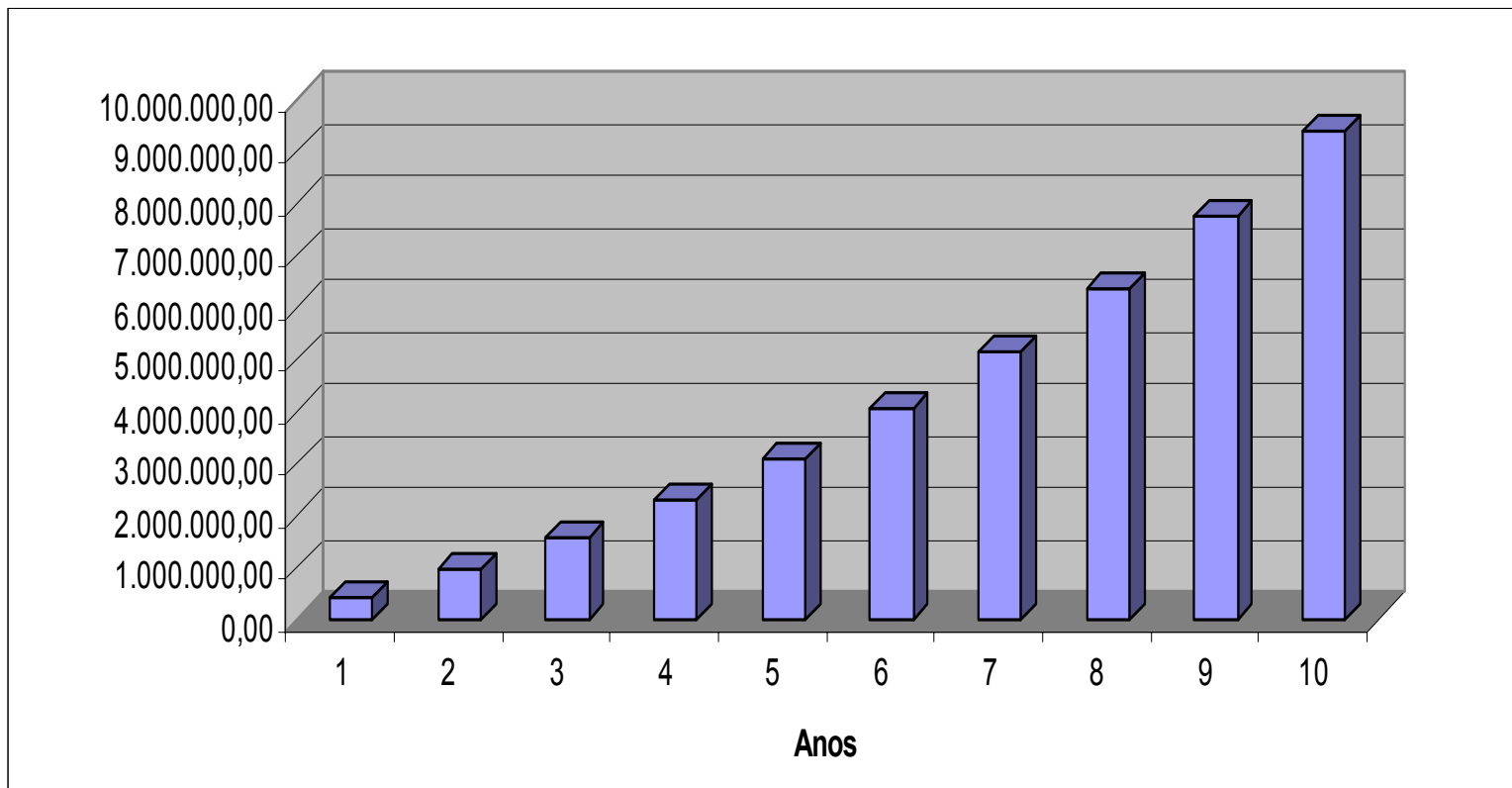
**Milhares de usuários simultâneos em
Produção, maior economia**

ATRIBUTO	VALOR
Qtde de processadores	1
MIPS por processador	215
Custo por MIPS (US\$) em cinco anos (suposição)	27.000,00
Período (em anos) apuração do custo	5
Economia em CPU (minutos) (em 51 minutos do teste)	20,35
Economia em tempo de resposta	55%
Máquina economizada (%)	39,9%
MIPS poupados (39,9% de 1 processador de 215 MIPS)	85,78
US\$ economizado anualmente (sem considerar nenhum crescimento anual de CPU)	463.212,00
Crescimento anual da CPU (%) - estimado	15%

ECONOMIA EM 10 ANOS – MIPS Acumulados (Crescimento anual da CPU de 15%)



ECONOMIA EM 10 ANOS - VALOR (US\$) **(Crescimento anual da CPU de 15%)**



ECONOMIA EM 10 ANOS

VALOR TOTAL

US\$ 9.404.925,93

Outros Ganhos

Aplicar o processo no workload DB2

- SQLs idênticos e similares consolidados
- Relacionamentos múltiplos
 - Programas x Sub-Programas x Transações
- SQLs com scans detectados
- Indicar predicados candidatos a Índices
- Recomendações de Índices
 - Problema Fatorial (Combinações)
 - Previsão de ganhos
 - Automação

Metodologias para otimização

- Tabelas DB2 mais acessadas
- Programas DB2 de maior consumo
- Transações DB2 com maior consumo
- Comandos SQL que mais consomem
- SLA - foco de estudos: On-line / Batch / horários específicos
- Maiores consumos de LOG do DB2
- Sugestões de índice em geral (por tabela, programas etc.)

Outras Considerações

- Método preventivo e pró-ativo
 - Processo em Pré-Produção (Homologação)
- Detectar desvios de consumo
- Processo de análise de impacto
 - Comparar histórico de planos de acesso
 - Evitar as “segundas-feiras de terror”



Outros Ganhos

Melhor atendimento

Menor tempo de resposta proporciona melhor atendimento aos clientes da organização

Estabilidade e Qualidade

Ambiente otimizado proporciona maior estabilidade e qualidade tanto interna quanto externamente



Dúvidas?

José Tito

jose.tito@eccox.com.br

Obrigado!



BACKUP TÉCNICO



RESUMO PROGRAMA

PROGRAM NAME: **xxxx0024** (%CPU-WKLD= 4.019946)

ACCOUNTING DATA

DB2_ELAP: 2,180.366448
DB2_CPU : 630.375678
AVG_CPU : 0.596946

PERFORMANCE DATA

DB2_ELAP: 19.559378
DB2_CPU : 1.181888
AVG_CPU : 0.000393

TOTAL EXECUTIONS : 3003
TOTAL_REOPT : 0
TOTAL_NOREOPT : 0

	INDEX	DATA	SORT
ROWS_SCAN_TBL_1	1,760	2,295	512
ROWS_SCAN_TBE_2	23	1,887	512
ROWS_QUAL_STAG_1	1,611	303	512
ROWS_QUAL_STAG_2	1,609	303	228
TOT_ROWS_INSERT	0	0	1,038
TOT_ROWS_UPDATE	0	0	0
TOT_ROWS_DELETE	0	0	0
TOT_PAGE_SCAN	4,345	2,191	2,956
TOT_PAGE_SCAN_RI	0	0	0
TOT_PAGE_ROWS_DEL_RI	0	0	0



RECOMENDAÇÃO DE ÍNDICES (TABELA XXXXXOMT)

VOLUME: 343.279 REGISTROS

COMBINAÇÕES TESTADAS: 2104

ESTIMATIVA MELHORA: 21,87 %

Índices atuais:

- XXXXXO11 - recomendacao: manter inalterado
- XXXXXO11 - recomendacao: deletar
- XXXXXO12 - recomendacao: manter inalterado
- XXXXXO13 - recomendacao: manter inalterado
- XXXXXO14 - recomendacao: deletar
- XXXXXO15 - recomendacao: deletar
- XXXXXO16 - recomendacao: deletar

Índices novos:

Criar 3 novos índices



RECOMENDAÇÃO DE ÍNDICES (TABELA XXXXXCMT)

VOLUME: 391.064 REGISTROS

COMBINAÇÕES TESTADAS: 1564

ESTIMATIVA MELHORA: 11,58 %

Índices atuais:

XXXXXXOI1 - recomendacao: alterar

XXXXXCI1 - recomendacao: manter inalterado

Índices novos:

Criar 5 novos índices



Statements do programa analisados

Stmt: 5638 tabela XXXXXRMT cursor: CUR_XXXXXRDV_SAD (cont)

Texto SQL Original

```
DECLARE CUR_PQPCDRDV_SAD CURSOR FOR
SELECT PCD_COD_PRO_CMR_CD
FROM ATISOWN.PQPCDRDV
WHERE
PCD_TIP_DIR_PRO_CD = ? AND
PCD_COD_DIR_CD = ? AND
PCD_COD_PRO_CMR_CD >= ? AND
PCD_FEC_INI_VIG_FF <= CURRENT_DATE AND
PCD_FEC_FIN_VIG_FF >= CURRENT_DATE AND
PCD_ITE_BAJ_SIS_TS IS NULL
ORDER BY PCD_COD_PRO_CMR_CD OPTIMIZE FOR 1 ROW
```



Statements do programa analisados

Stmt: 2835 tabela XXXXOMT cursor: CUR_CLI_AMBOS

DB2 elapsed medido: 40% tempo total
(ou 1.7% - teste stress)

Custo Original (com Union): 7 service units
8 ms (CPU)

Solução:

Diversas soluções foram encontradas...

Statements do programa analisados

Stmt: 2835 tabela XXXXXOMT cursor: CUR_CLI_AMBOS

a) Soluções que **não dependem** da análise do código fonte:

a.1) Continuar com **Union** e **criar** 2 novos índices (*)

su: 6 CPU: 7

b) Soluções que **dependem** da análise do código fonte:

b.1) Mudar para **OR** ao invés de Union, mas é necessário saber se PRC_COD_PRO_CM \bar{R} _CD ≥ 0 é utilizado... se for, esta solução não é válida, pode causar oscilações tempo resposta

su: 3 CPU: 3

(detalhe: usou técnica de interseção de índices e fez sort)

b.2) idem a.1, mas modificando o SQL (ORDER e UNION) pode-se chegar a:

su: 3 CPU: 3

Statements do programa analisados

Stmt: 2835 tabela XXXXXOMT cursor: CUR_CLI_AMBOS

Análise dos índices da tabela XXXXXOMT (342.279 regs)

1) Novo índice: ATISOWN. PQPRCOIA (único)
PRC_COD_CLI_CD ASC (132.003)
PRC_ITE_BAJ_SIS_TS ASC (1)
PRC_COD_PRO_CMR_CD ASC (342.279)

2) Novo índice: ATISOWN. PQPRCOIB (único)
PRC_COD_CLI_TIT_CD ASC (132.003)
PRC_ITE_BAJ_SIS_TS ASC (1)
PRC_COD_PRO_CMR_CD ASC (342.279)



Statements do programa analisados

Stmt: 2835 **tabela XXXXXOMT cursor: CUR_CLI_AMBOS**

Análise dos índices da tabela XXXXXOMT (343.279 regs)

CREATE INDEX ATISOWN.XXXXXOI1

(PRC_TIP_PRO_CMR_CD ASC,	(2)
PRC_SBT_PRO_CMR_CD ASC)	(16)

CREATE UNIQUE INDEX ATISOWN.XXXXXOI4

(PRC_COD_CMR_PRN_CD ASC,	(48)
PRC_IND_CMR_GCO_IN ASC,	(2)
PRC_COD_PRO_CMR_CD ASC,	(343.279)
PRC_TIP_PRO_CMR_CD ASC,	(2)
PRC_SBT_PRO_CMR_CD ASC)	(16)

CREATE INDEX ATISOWN.XXXXXOI5

(PRC_COD_PET_CD ASC,	(96)
PRC_COD_AGR_SUB_NU ASC)	3

CREATE INDEX ATISOWN.XXXXXOI6

(PRC_CET_COS_CLI_CD ASC)	(1)
--------------------------	-----



Statements do programa analisados

Stmt: 7642 tabela XXXXXCMT cursor:(select sem cursor)

DB2 elapsed medido: 50% tempo total
(não localizado - teste stress)

Custo Original: 5 service units
6 ms (CPU)

Solução:

Alterar a sintaxe no comando SQL

Novo Custo Esperado: 1 service units (> 80%)
2 ms (CPU) (> 70%)



Statements do programa analisados

Stmt: 7642 **tabela XXXXXCMT cursor:(select sem cursor)**

Texto SQL original

```
SELECT IDP_COD_PRO_CMR_CD, IDP_FEC_INI_VIG_FF, IDP_FEC_FIN_VIG_FF
FROM ATISOWN.PQIDPCDV
WHERE IDP_NUM_IDE_NU = ? AND
IDP_IDE_PRO_CMR_CD = ? AND
IDP_FEC_INI_VIG_FF <= CURRENT_DATE AND
IDP_FEC_FIN_VIG_FF >= CURRENT_DATE AND
IDP_ITE_BAJ_SIS_TS IS NULL AND
(IDP_TIM_ULT_MOD_TS IN
( SELECT MAX ( IDP_TIM_ULT_MOD_TS )
FROM ATISOWN.PQIDPCDV
WHERE IDP_NUM_IDE_NU = ? AND
      IDP_IDE_PRO_CMR_CD = ? AND
      IDP_FEC_INI_VIG_FF <= CURRENT_DATE AND
      IDP_FEC_FIN_VIG_FF >= CURRENT_DATE AND
      IDP_ITE_BAJ_SIS_TS IS NULL
```



Statements do programa analisados

Stmt: 7642 tabela XXXXXCMT cursor:(select sem cursor)

Texto SQL Alterado

```
SELECT A.IDP_COD_PRO_CMR_CD , A.IDP_FEC_INI_VIG_FF
      ,A.IDP_FEC_FIN_VIG_FF
FROM ATISOWN.PQIDPCDV A WHERE A.IDP_NUM_IDE_NU = ? AND
      A.IDP_IDE_PRO_CMR_CD = ? AND A.IDP_FEC_INI_VIG_FF <= CURRENT_DATE
AND
      A.IDP_FEC_FIN_VIG_FF >= CURRENT_DATE AND A.IDP_ITE_BAJ_SIS_TS IS NULL
AND
(A.IDP_TIM_ULT_MOD_TS IN ( SELECT MAX ( B.IDP_TIM_ULT_MOD_TS )
FROM ATISOWN.PQIDPCDV B WHERE B.IDP_NUM_IDE_NU =
      A.IDP_NUM_IDE_NU AND
      B.IDP_IDE_PRO_CMR_CD = A.IDP_IDE_PRO_CMR_CD AND
      B.IDP_FEC_INI_VIG_FF = A.IDP_FEC_INI_VIG_FF AND
      B.IDP_FEC_FIN_VIG_FF = A.IDP_FEC_FIN_VIG_FF AND
      B.IDP_ITE_BAJ_SIS_TS = A.IDP_ITE_BAJ_SIS_TS))
```



Statements do programa analisados

Stmt: 2776 **tabela XXXXXADV cursor: (select sem cursor)**

DB2 elapsed medido: 1% tempo total
(ou 0.76% - teste stress)

Custo Original: 6 service units
7 ms (CPU)

Solução:

1)Alterar comandos SQL

Novo Custo esperado: 1 service units (> 90%)
2 ms (CPU) (> 80%)

2)Criar novo índice e alterar comando SQL (ainda não testado)



Statements do programa analisados

Stmt: 2776 **tabela XXXXXADV cursor: (select sem cursor)**

CREATE UNIQUE INDEX ATISOWN.XXXXXAI0

(PSP_COD_PRO_CMR_CD ASC, (171.614)
PSP_PRO_SER_CTO_CD ASC, (1.009.000)
PSP_TIP_PRO_SER_CD ASC) (9)

CREATE UNIQUE INDEX ATISOWN.XXXXXAI1

(PSP_PRO_SER_CTO_CD ASC) (1.009.000)

CREATE UNIQUE INDEX ATISOWN.XXXXXAI2

(PSP_NUM_AVA_NU ASC, (1)
PSP_PRO_SER_CTO_CD ASC) (1.009.000)

CREATE UNIQUE INDEX ATISOWN.XXXXXAI3 (não utilizado)

(PSP_CET_COS_CLI_CD ASC, (1)
PSP_PRO_SER_CTO_CD ASC) (1.009.000)

CREATE INDEX ATISOWN.XXXXXAI4

(PSP_COD_PRO_SER_CD ASC) (455)



Statements do programa analisados

Stmt: 2776 **tabela XXXXXADV cursor: (select sem cursor)**

Texto SQL Original

```
SELECT PSP_PRO_SER_CTO_CD
FROM PQPSPADV
WHERE PSP_COD_PRO_CMR_CD = ? AND
PSP_FEC_INI_VIG_FF <= CURRENT_DATE AND
PSP_FEC_FIN_VIG_FF >= CURRENT_DATE AND
PSP_ITE_BAJ_SIS_TS IS NULL AND
PSP_TIM_ULT_MOD_TS IN
(( SELECT MAX ( PSP_TIM_ULT_MOD_TS )
FROM PQPSPADV
WHERE PSP_COD_PRO_CMR_CD = ? AND
PSP_FEC_INI_VIG_FF <= CURRENT_DATE AND
PSP_FEC_FIN_VIG_FF >= CURRENT_DATE AND
PSP_ITE_BAJ_SIS_TS IS NULL))
```



Statements do programa analisados

Stmt: 2776 tabela XXXXXADV cursor: (select sem cursor)

Texto SQL Alterado

```
SELECT A.PSP_PRO_SER_CTO_CD FROM ATISOWN.PQPSPADV A
WHERE A.PSP_COD_PRO_CMR_CD = ? AND
A.PSP_FEC_INI_VIG_FF <= CURRENT_DATE AND
A.PSP_FEC_FIN_VIG_FF >= CURRENT_DATE AND
A.PSP_ITE_BAJ_SIS_TS IS NULL AND
A.PSP_TIM_ULT_MOD_TS IN ( SELECT MAX ( B.PSP_TIM_ULT_MOD_TS )
FROM ATISOWN.PQPSPADV B WHERE
B.PSP_COD_PRO_CMR_CD = A.PSP_COD_PRO_CMR_CD AND
B.PSP_FEC_INI_VIG_FF = A.PSP_FEC_INI_VIG_FF AND
B.PSP_FEC_FIN_VIG_FF = A.PSP_FEC_FIN_VIG_FF AND
B.PSP_ITE_BAJ_SIS_TS = A.PSP_ITE_BAJ_SIS_TS)
```