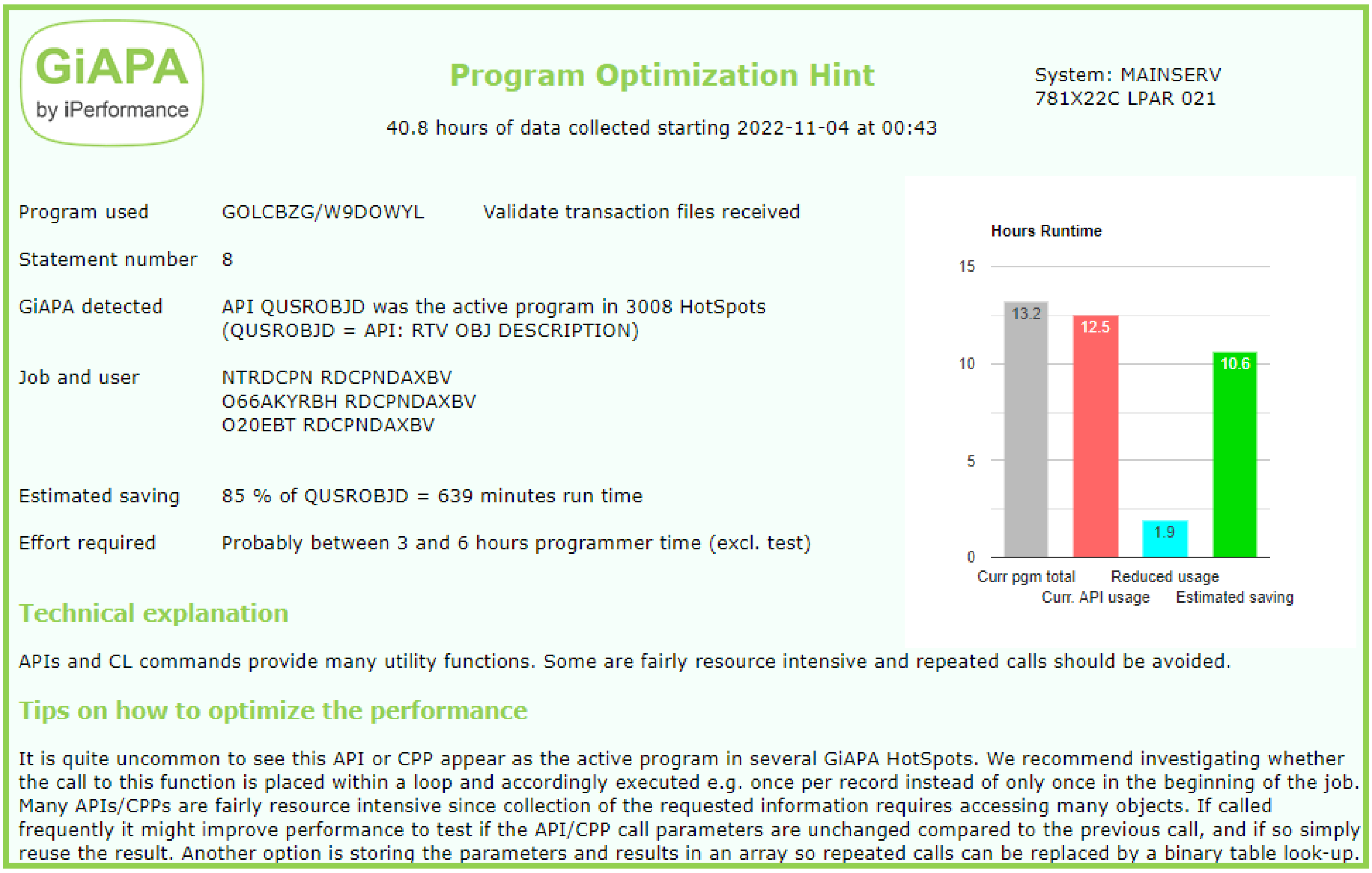
GiAPA's performance analysis locates +/- 80 % of all commonly seen inefficiencies, normally also many in old applications believed to run OK, because no-one ever before analyzed all active jobs

Automatically generated optimization tips based on analysis of all active programs



Program Optimization Hint

System: MAINSERV
781X22C LPAR 021

40.8 hours of data collected starting 2022-11-04 at 00:43

Program used GOLCBZG/W9DOWYL Validate transaction files received

Statement number 8

GiAPA detected API QUSROBJD was the active program in 3008 HotSpots
(QUSROBJD = API: RTV OBJ DESCRIPTION)

Job and user NTRDCPN RDCPNDAXBV
O66AKYRBH RDCPNDAXBV
O20EBT RDCPNDAXBV

Estimated saving 85 % of QUSROBJD = 639 minutes run time

Effort required Probably between 3 and 6 hours programmer time (excl. test)

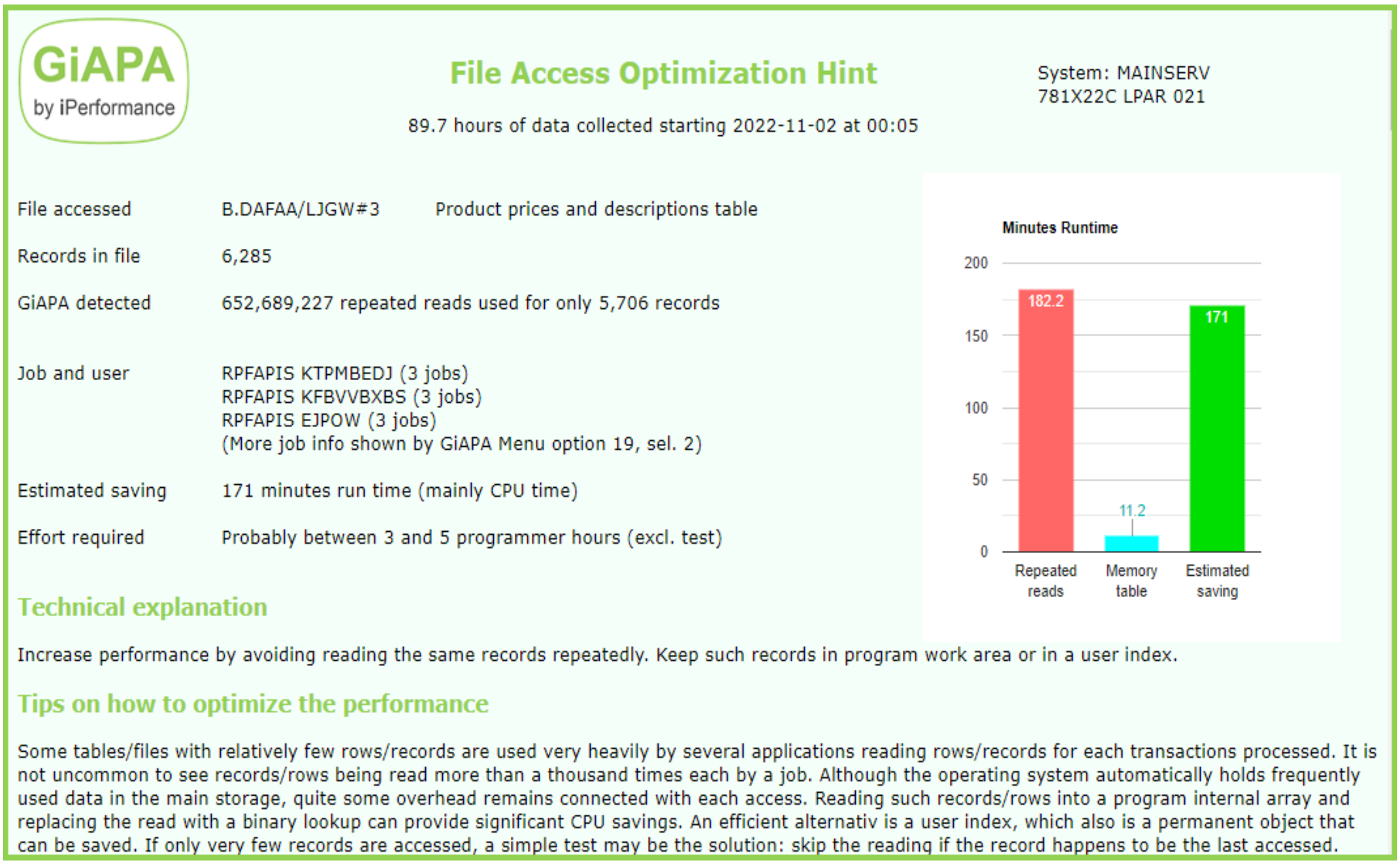
Technical explanation

APIs and CL commands provide many utility functions. Some are fairly resource intensive and repeated calls should be avoided.

Tips on how to optimize the performance

It is quite uncommon to see this API or CPP appear as the active program in several GiAPA HotSpots. We recommend investigating whether the call to this function is placed within a loop and accordingly executed e.g. once per record instead of only once in the beginning of the job. Many APIs/CPPs are fairly resource intensive since collection of the requested information requires accessing many objects. If called frequently it might improve performance to test if the API/CPP call parameters are unchanged compared to the previous call, and if so simply reuse the result. Another option is storing the parameters and results in an array so repeated calls can be replaced by a binary table look-up.

Analyzing file accesses across all jobs reveals significant savings, easy to implement



File Access Optimization Hint

System: MAINSERV
781X22C LPAR 021

89.7 hours of data collected starting 2022-11-02 at 00:05

File accessed B.DAFAA/LJGW#3 Product prices and descriptions table

Records in file 6,285

GiAPA detected 652,689,227 repeated reads used for only 5,706 records

Job and user RPFAPIS KTPMBEDJ (3 jobs)
RPFAPIS KFBVVBXBS (3 jobs)
RPFAPIS EJPOW (3 jobs)
(More job info shown by GiAPA Menu option 19, sel. 2)

Estimated saving 171 minutes run time (mainly CPU time)

Effort required Probably between 3 and 5 programmer hours (excl. test)

Technical explanation

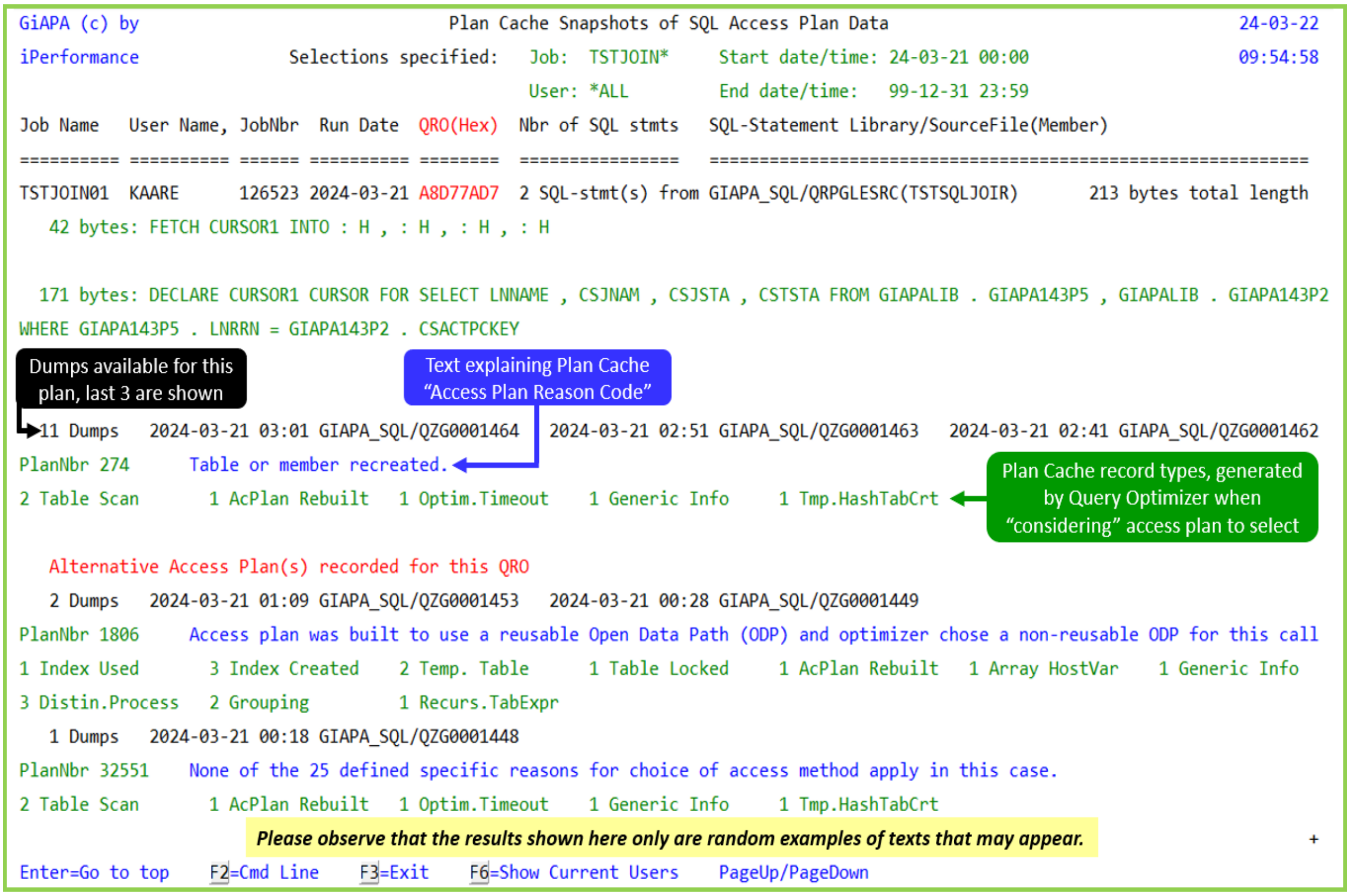
Increase performance by avoiding reading the same records repeatedly. Keep such records in program work area or in a user index.

Tips on how to optimize the performance

Some tables/files with relatively few rows/records are used very heavily by several applications reading rows/records for each transactions processed. It is not uncommon to see records/rows being read more than a thousand times each by a job. Although the operating system automatically holds frequently used data in the main storage, quite some overhead remains connected with each access. Reading such records/rows into a program internal array and replacing the read with a binary lookup can provide significant CPU savings. An efficient alternativ is a user index, which also is a permanent object that can be saved. If only very few records are accessed, a simple test may be the solution: skip the reading if the record happens to be the last accessed.

Data collection uses < 0.1% CPU. I/Os and disc space minimized by compressing data 92%

SQL Plan Cache data for jobs selected by user is automatically collected and pre-analyzed



```
GiAPA (c) by                          Plan Cache Snapshots of SQL Access Plan Data                            24-03-22
iPerformance                Selections specified:   Job:  TSTJOIN*     Start date/time: 24-03-21 00:00        09:54:58
                                                    User: *ALL         End date/time:   99-12-31 23:59
Job Name   User Name, JobNbr  Run Date  QRO(Hex)  Nbr of SQL stmts   SQL-Statement Library/SourceFile(Member)
========== ========== ====== ========== ======== ================   ============================================================
TSTJOIN01  KAARE      126523 2024-03-21 A8D77AD7  2 SQL-stmt(s) from GIAPA_SQL/QRPGLESRC(TSTSQLJOIR)       213 bytes total length
   42 bytes: FETCH CURSOR1 INTO : H , : H , : H , : H


  171 bytes: DECLARE CURSOR1 CURSOR FOR SELECT LNNAME , CSJNAM , CSJSTA , CSTSTA FROM GIAPALIB . GIAPA143P5 , GIAPALIB . GIAPA143P2
WHERE GIAPA143P5 . LNRRN = GIAPA143P2 . CSACTPCKEY

  11 Dumps   2024-03-21 03:01 GIAPA_SQL/QZG0001464   2024-03-21 02:51 GIAPA_SQL/QZG0001463   2024-03-21 02:41 GIAPA_SQL/QZG0001462
PlanNbr 274      Table or member recreated.
2 Table Scan       1 AcPlan Rebuilt   1 Optim.Timeout    1 Generic Info     1 Tmp.HashTabCrt

   Alternative Access Plan(s) recorded for this QRO
   2 Dumps   2024-03-21 01:09 GIAPA_SQL/QZG0001453   2024-03-21 00:28 GIAPA_SQL/QZG0001449
PlanNbr 1806     Access plan was built to use a reusable Open Data Path (ODP) and optimizer chose a non-reusable ODP for this call
1 Index Used       3 Index Created    2 Temp. Table      1 Table Locked     1 AcPlan Rebuilt   1 Array HostVar    1 Generic Info
3 Distin.Process   2 Grouping         1 Recurs.TabExpr
   1 Dumps   2024-03-21 00:18 GIAPA_SQL/QZG0001448
PlanNbr 32551    None of the 25 defined specific reasons for choice of access method apply in this case.
2 Table Scan       1 AcPlan Rebuilt   1 Optim.Timeout    1 Generic Info     1 Tmp.HashTabCrt
                                                                                                                       +
Enter=Go to top    F2=Cmd Line    F3=Exit    F6=Show Current Users    PageUp/PageDown
```

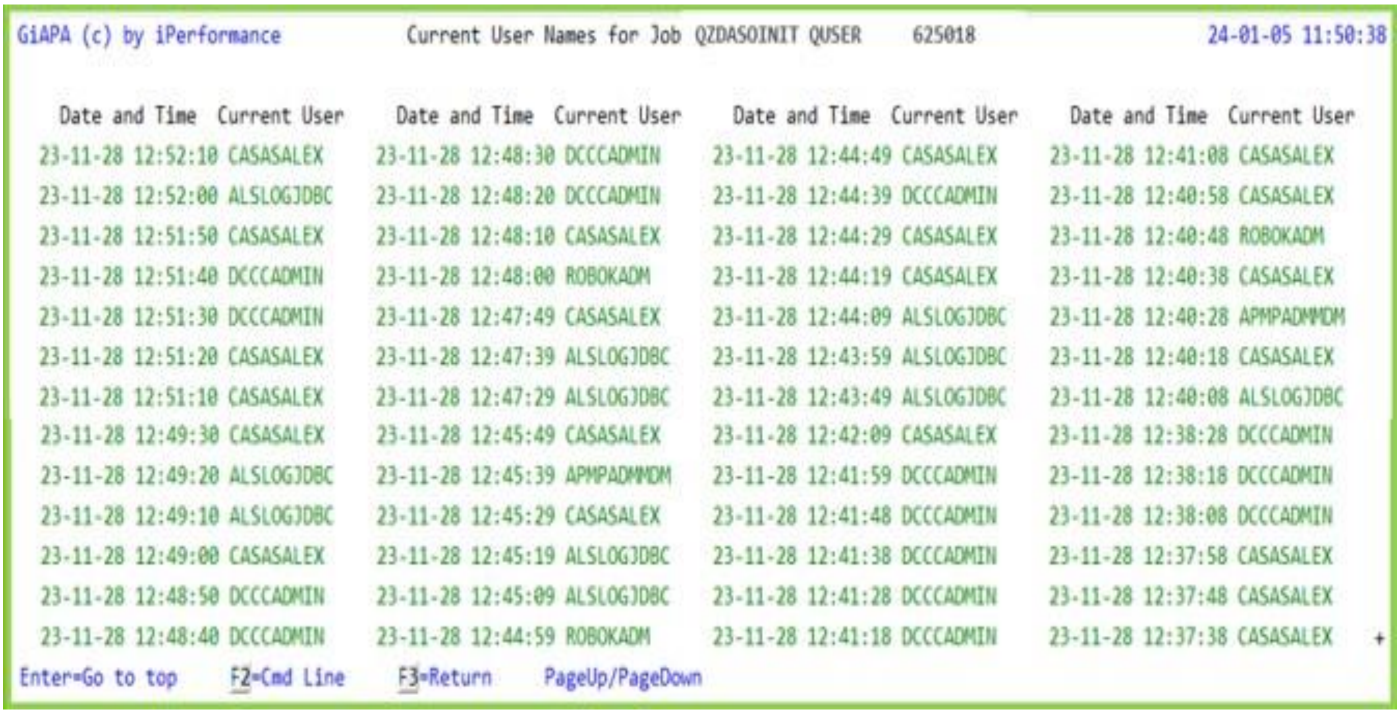
Dumps available for this plan, last 3 are shown

Text explaining Plan Cache "Access Plan Reason Code"

Plan Cache record types, generated by Query Optimizer when "considering" access plan to select

Please observe that the results shown here only are random examples of texts that may appear.

✓ **Shows location of snapshots needed for analysis using IBM's SQL Performance Center**

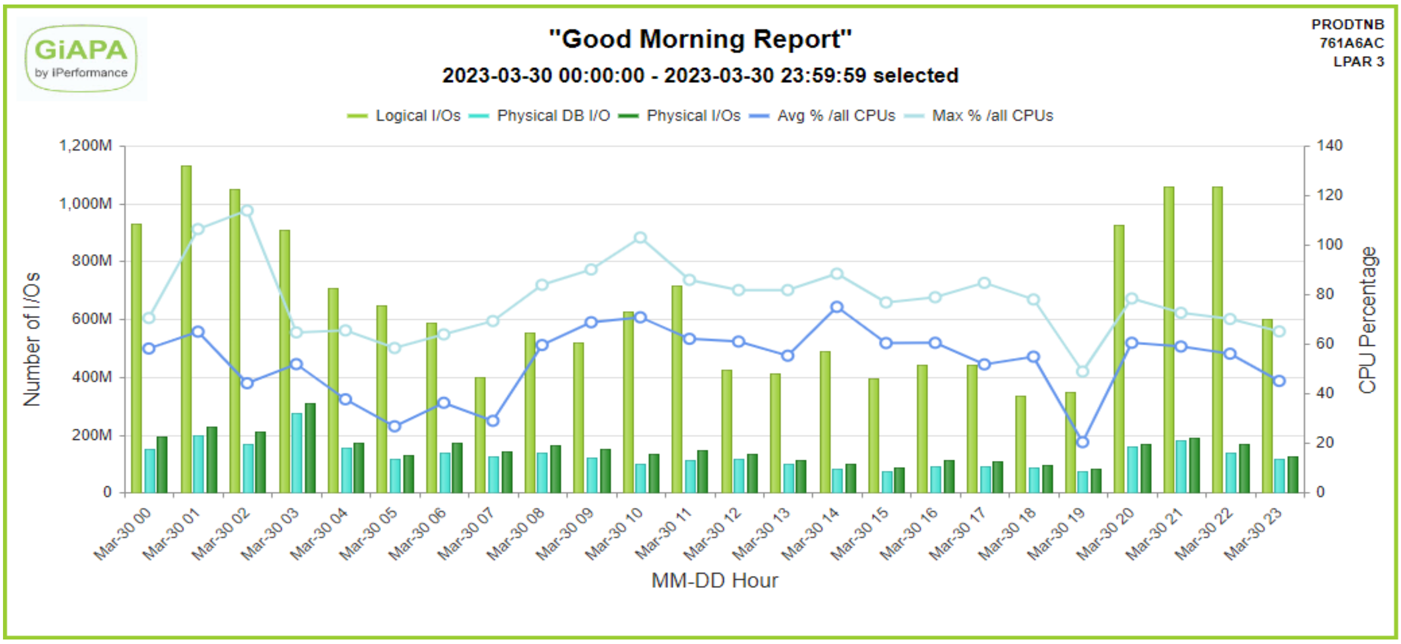


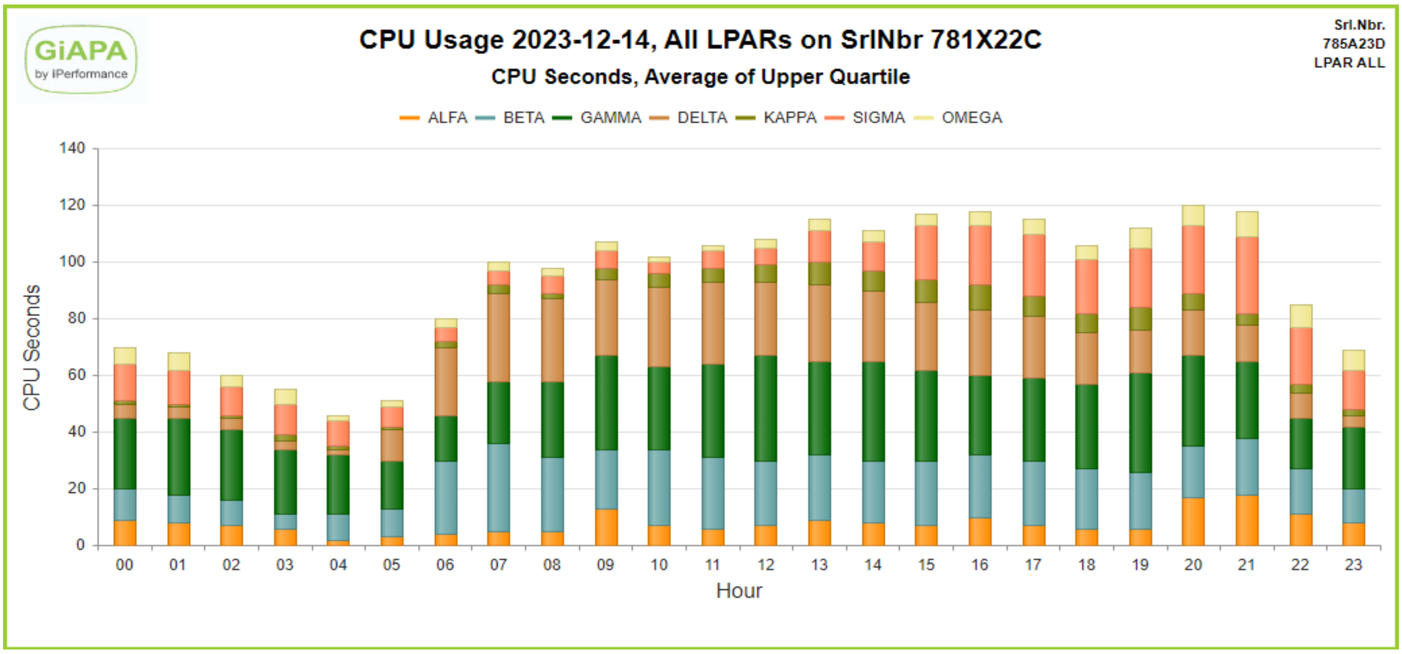
```
GiAPA (c) by iPerformance          Current User Names for Job  QZDASOINIT QUSER     625018              24-01-05 11:50:38

   Date and Time  Current User    Date and Time  Current User    Date and Time  Current User    Date and Time  Current User
  23-11-28 12:52:10 CASASALEX    23-11-28 12:48:30 DCCCADMIN    23-11-28 12:44:49 CASASALEX    23-11-28 12:41:08 CASASALEX
  23-11-28 12:52:00 ALSLOGJDBC   23-11-28 12:48:20 DCCCADMIN    23-11-28 12:44:39 DCCCADMIN    23-11-28 12:40:58 CASASALEX
  23-11-28 12:51:50 CASASALEX    23-11-28 12:48:10 CASASALEX    23-11-28 12:44:29 CASASALEX    23-11-28 12:40:48 ROBOKADM
  23-11-28 12:51:40 DCCCADMIN    23-11-28 12:48:00 ROBOKADM     23-11-28 12:44:19 CASASALEX    23-11-28 12:40:38 CASASALEX
  23-11-28 12:51:30 DCCCADMIN    23-11-28 12:47:49 CASASALEX    23-11-28 12:44:09 ALSLOGJDBC   23-11-28 12:40:28 APMPADMMDM
  23-11-28 12:51:20 CASASALEX    23-11-28 12:47:39 ALSLOGJDBC   23-11-28 12:43:59 ALSLOGJDBC   23-11-28 12:40:18 CASASALEX
  23-11-28 12:51:10 CASASALEX    23-11-28 12:47:29 ALSLOGJDBC   23-11-28 12:43:49 ALSLOGJDBC   23-11-28 12:40:08 ALSLOGJDBC
  23-11-28 12:49:30 CASASALEX    23-11-28 12:45:49 CASASALEX    23-11-28 12:42:09 CASASALEX    23-11-28 12:38:28 DCCCADMIN
  23-11-28 12:49:20 ALSLOGJDBC   23-11-28 12:45:39 APMPADMMDM   23-11-28 12:41:59 DCCCADMIN    23-11-28 12:38:18 DCCCADMIN
  23-11-28 12:49:10 ALSLOGJDBC   23-11-28 12:45:29 CASASALEX    23-11-28 12:41:48 DCCCADMIN    23-11-28 12:38:08 DCCCADMIN
  23-11-28 12:49:00 CASASALEX    23-11-28 12:45:19 ALSLOGJDBC   23-11-28 12:41:38 DCCCADMIN    23-11-28 12:37:58 CASASALEX
  23-11-28 12:48:50 DCCCADMIN    23-11-28 12:45:09 ALSLOGJDBC   23-11-28 12:41:28 DCCCADMIN    23-11-28 12:37:48 CASASALEX
  23-11-28 12:48:40 DCCCADMIN    23-11-28 12:44:59 ROBOKADM     23-11-28 12:41:18 DCCCADMIN    23-11-28 12:37:38 CASASALEX   +
Enter=Go to top    F2=Cmd Line    F3=Return    PageUp/PageDown
```

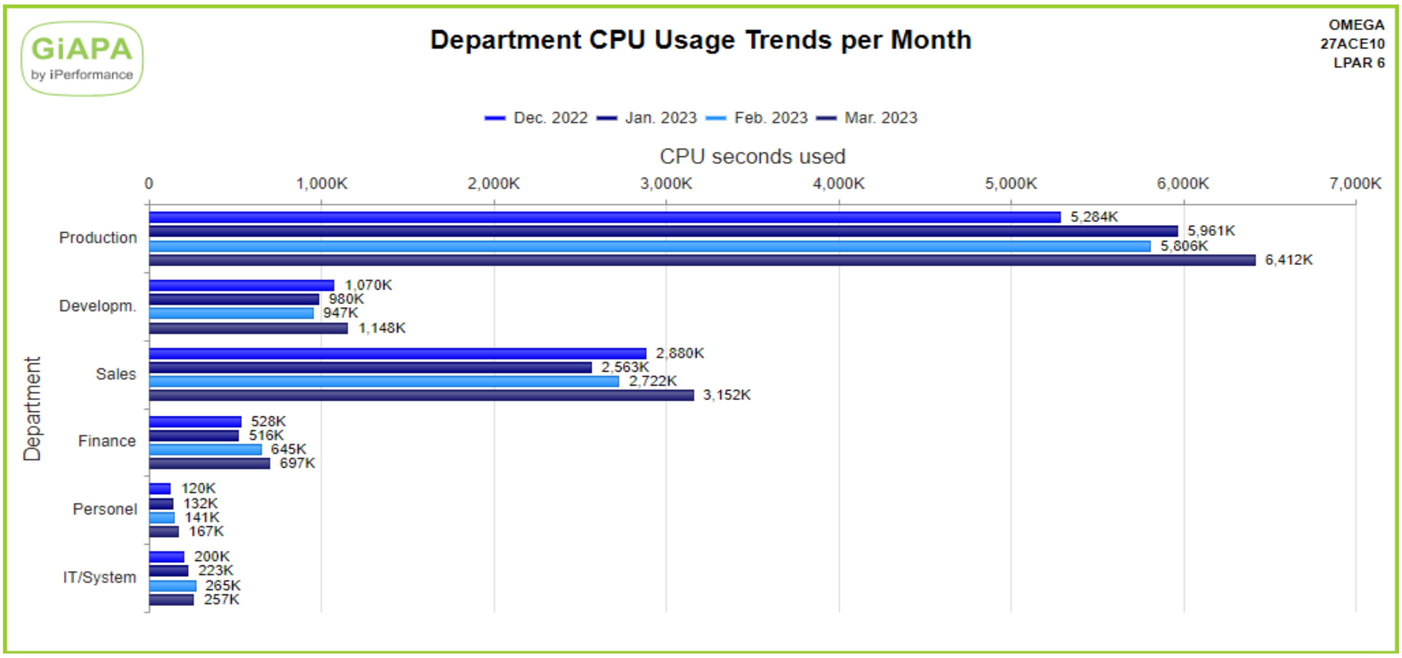
Current user names are valuable information when analyzing data base host server jobs

GiAPA has much more to offer - please watch our five minute video, technical presentation, references, and examples at www.giapa.com .

Standard or user defined graphs can be generated and emailed in night batch jobs


GiAPA
by iPerformance
"Good Morning Report"
2023-03-30 00:00:00 - 2023-03-30 23:59:59 selected
PRODTNB
761A6AC
LPAR 3
Logical I/Os
Physical DB I/O
Physical I/Os
Avg % /all CPUs
Max % /all CPUs
Number of I/Os
1,200M
1,000M
800M
600M
400M
200M
CPU Percentage
Mar-30 00
Mar-30 01
Mar-30 02
Mar-30 03
Mar-30 04
Mar-30 05
Mar-30 06
Mar-30 07
Mar-30 08
Mar-30 09
Mar-30 10
Mar-30 11
Mar-30 12
Mar-30 13
Mar-30 14
Mar-30 15
Mar-30 16
Mar-30 17
Mar-30 18
Mar-30 19
Mar-30 20
Mar-30 21
Mar-30 22
Mar-30 23
MM-DD Hour


GiAPA
by iPerformance
CPU Usage 2023-12-14, All LPARs on SrlNbr 781X22C
CPU Seconds, Average of Upper Quartile
Srl.Nbr.
785A23D
LPAR ALL
ALFA
BETA
GAMMA
DELTA
KAPPA
SIGMA
OMEGA
CPU Seconds
Hour


GiAPA
by iPerformance
Department CPU Usage Trends per Month
OMEGA
27ACE10
LPAR 6
Dec. 2022
Jan. 2023
Feb. 2023
Mar. 2023
CPU seconds used
1,000K
2,000K
3,000K
4,000K
5,000K
6,000K
7,000K
Department
Production
5,284K
5,961K
5,806K
6,412K
Developm.
1,070K
980K
947K
1,148K
Sales
2,880K
2,563K
2,722K
3,152K
Finance
528K
516K
645K
697K
Personel
120K
132K
141K
167K
IT/System
200K
223K
265K
257K