


GiAPA: Automatically Generated Optimization Hints for Programs Across All Jobs



Program Optimization Hint

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

System: MAINSERV
781X22C LPAR 021

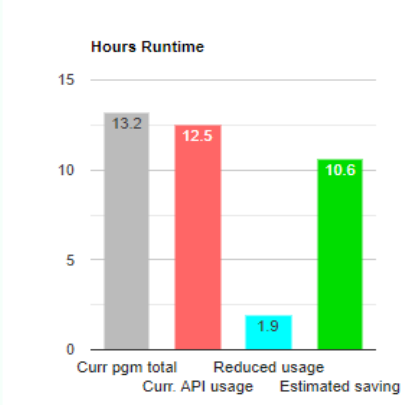
Program used	GOLCBZG/W9DOWYL	Validate transaction files received
Statement number	8	
GiAPA detected	API QUSROBJD was the active program in 3008 HotSpots (QUSROBJD = API: RTV OBJ DESCRIPTION)	
Job and user	NTRDCPN RDCPNDAXBV O66AKYRBH RDCPNDAXBV O20EBT RDCPNDAXBV	
Estimated saving	85 % of QUSROBJD = 639 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.



Category	Hours Runtime
Curr pgm total	13.2
Curr. API usage	12.5
Reduced usage	1.9
Estimated saving	10.6

GiAPA: File Access Analyzed Across All Jobs Reveals Optimization Potential



File Access Optimization Hint

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

System: MAINSERV
781X22C LPAR 021


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 KFBVVXBBS (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 alternative 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.



Category	Minutes Runtime
Repeated reads	182.2
Memory table	11.2
Estimated saving	171

Totals for estimated savings

Potential Savings Found by Automatic Analysis:	
27 Improvements of program functions	2,435 Minutes
8 Improvements of File Access Methods	277 Minutes
*** Total Potential Run Time Savings	45 Hours 12 Minutes

Data collection uses minimal disk space:
Data is compressed 92%.

GiAPA: SQL Plan Cache data for user selected jobs is collected and pre-analyzed

```

GiAPA (c) by iPerformance          Plan Cache Snapshots of SQL Access Plan Data          24-03-22
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

Dumps available, last 3 are shown
Text explaining Plan Cache "Access Plan Reason Code"
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.HashTabCr

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.HashTabCr

Please observe that the results shown here only are random examples of texts that may appear.
Enter=Go to top F2=Cmd Line F3=Exit F6=Show Current Users PageUp/PageDown
    
```

✔ 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

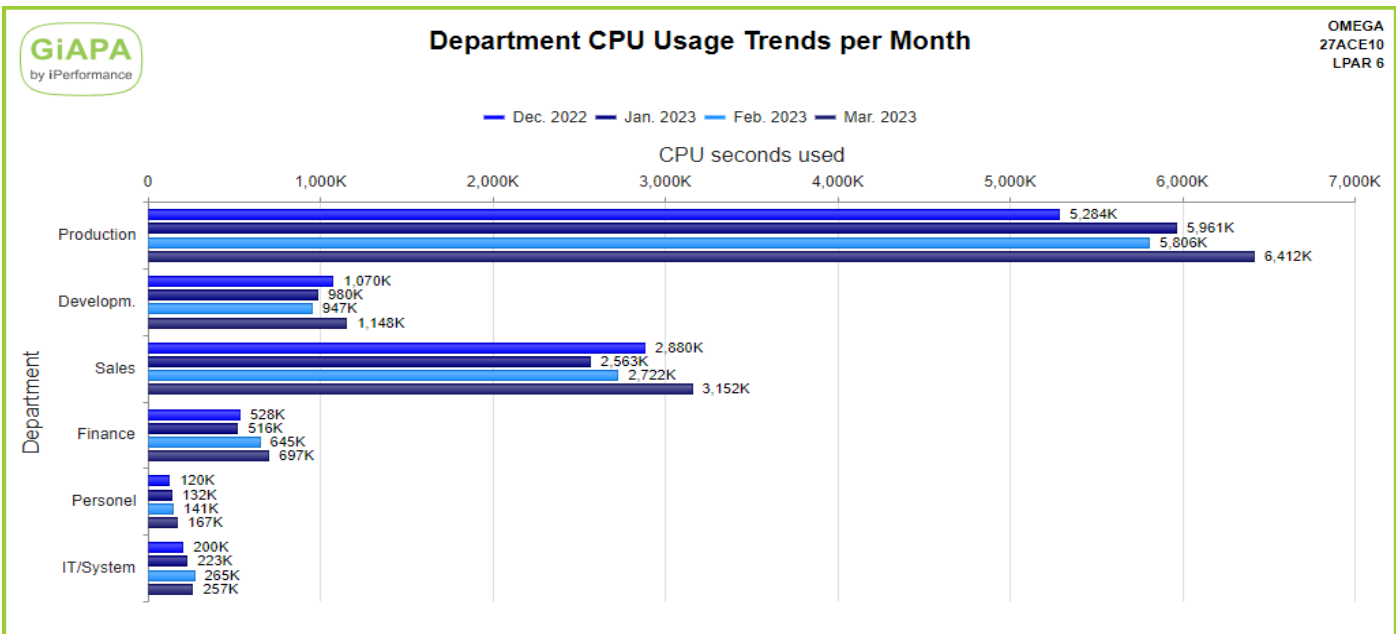
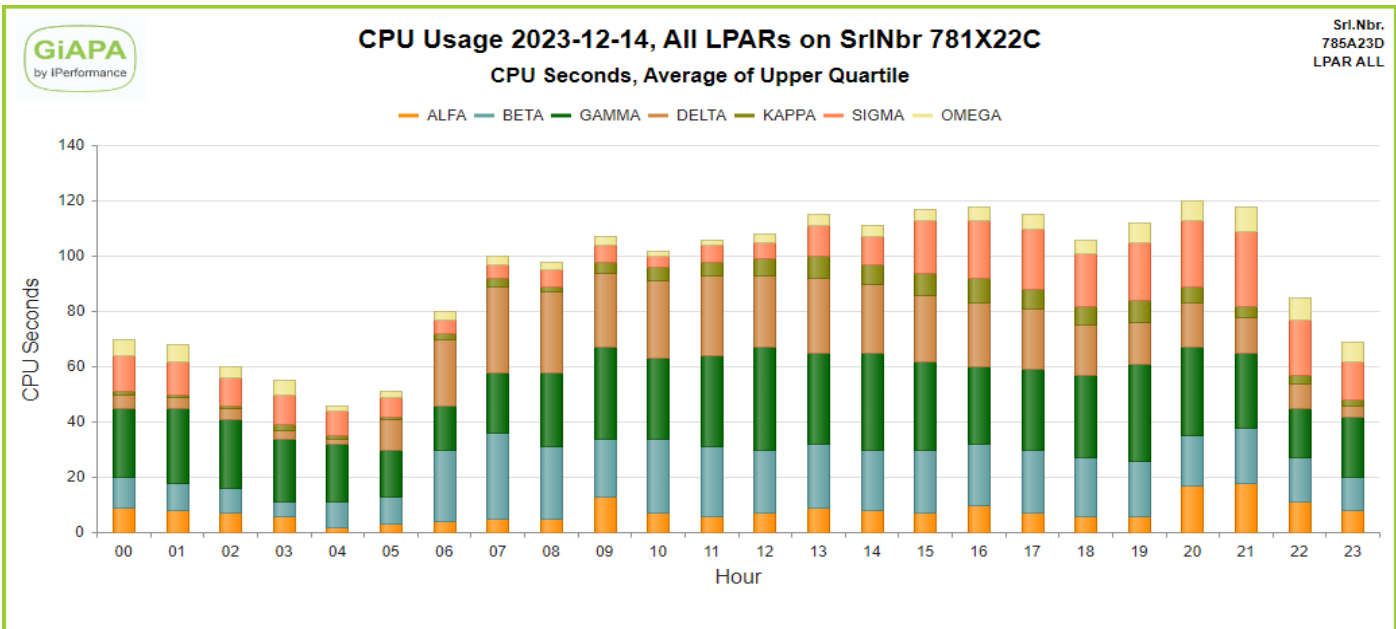
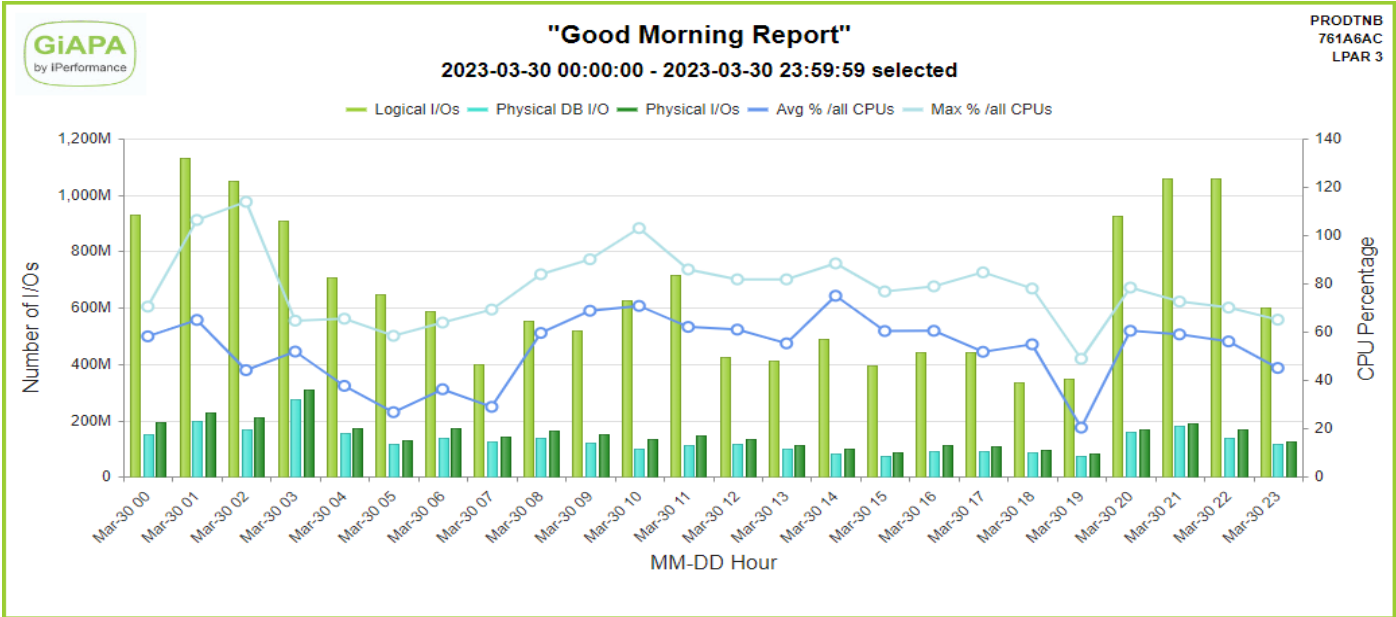
Date and Time Current User Date and Time Current User Date and Time Current User Date and Time Current User
23-11-28 12:52:10 CASASALEX 23-11-28 12:48:30 DCCCADMIN 23-11-28 12:44:49 CASASALEX 23-11-28 12:41:08 CASASALEX
23-11-28 12:52:00 ALSLOGJDBC 23-11-28 12:48:20 DCCCADMIN 23-11-28 12:44:39 DCCCADMIN 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
    
```

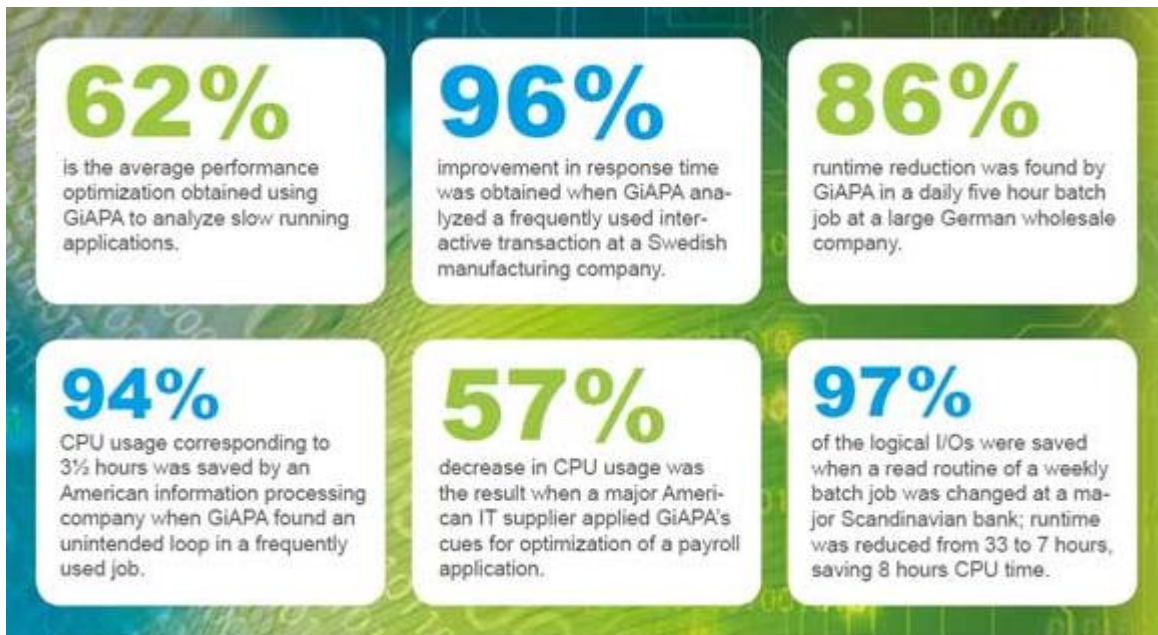
✔ Current user names are valuable information when analyzing data base host server jobs

GiAPA has much more to offer - please visit www.giapa.com for our five-minute video, technical presentation, references, and Free Trial !

Standard or user defined graph examples, generated and emailed by batch jobs



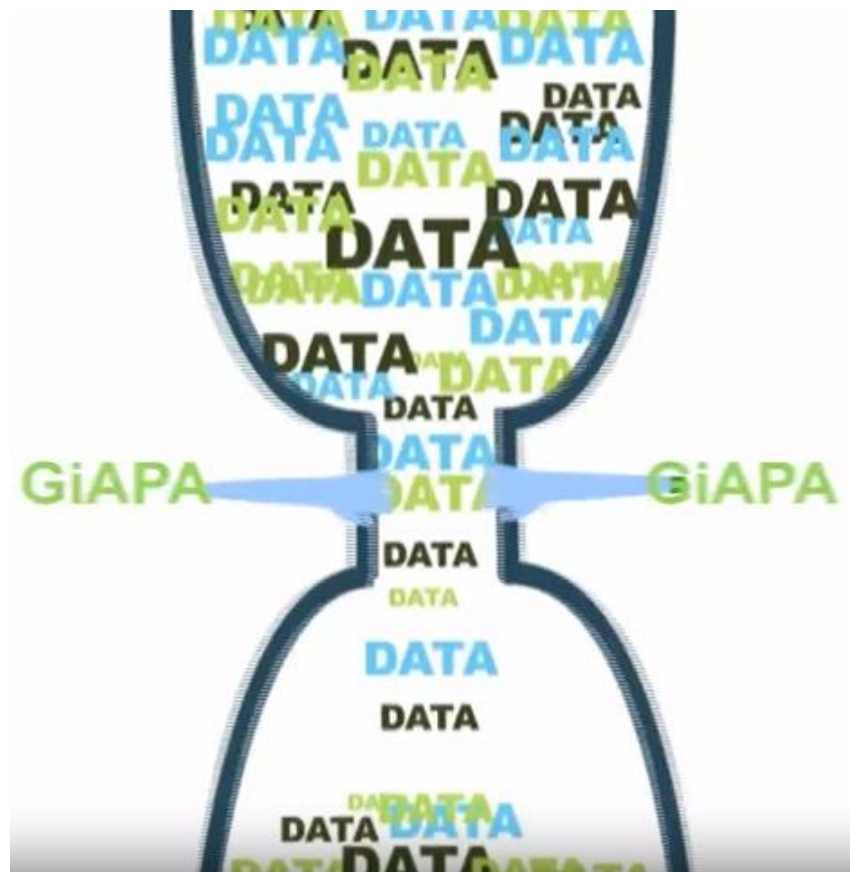
References: On <https://www.giapa.com> click on these fields - - -



- - - you will find a success story behind each of them!

GiAPA uses < 0.1 % CPU while collecting resource data for all jobs running – customers run GiAPA 24/7.

It will analyze all your applications, locate the bottlenecks, and show modifications needed to improve performance.



GiAPA typically detects substantial saving potentials in applications believed to run efficiently, because programs producing the correct results within a reasonable time never were performance analyzed.