Introduction to Fully Automated Application Performance Analysis

GiAPA

by iPerformance

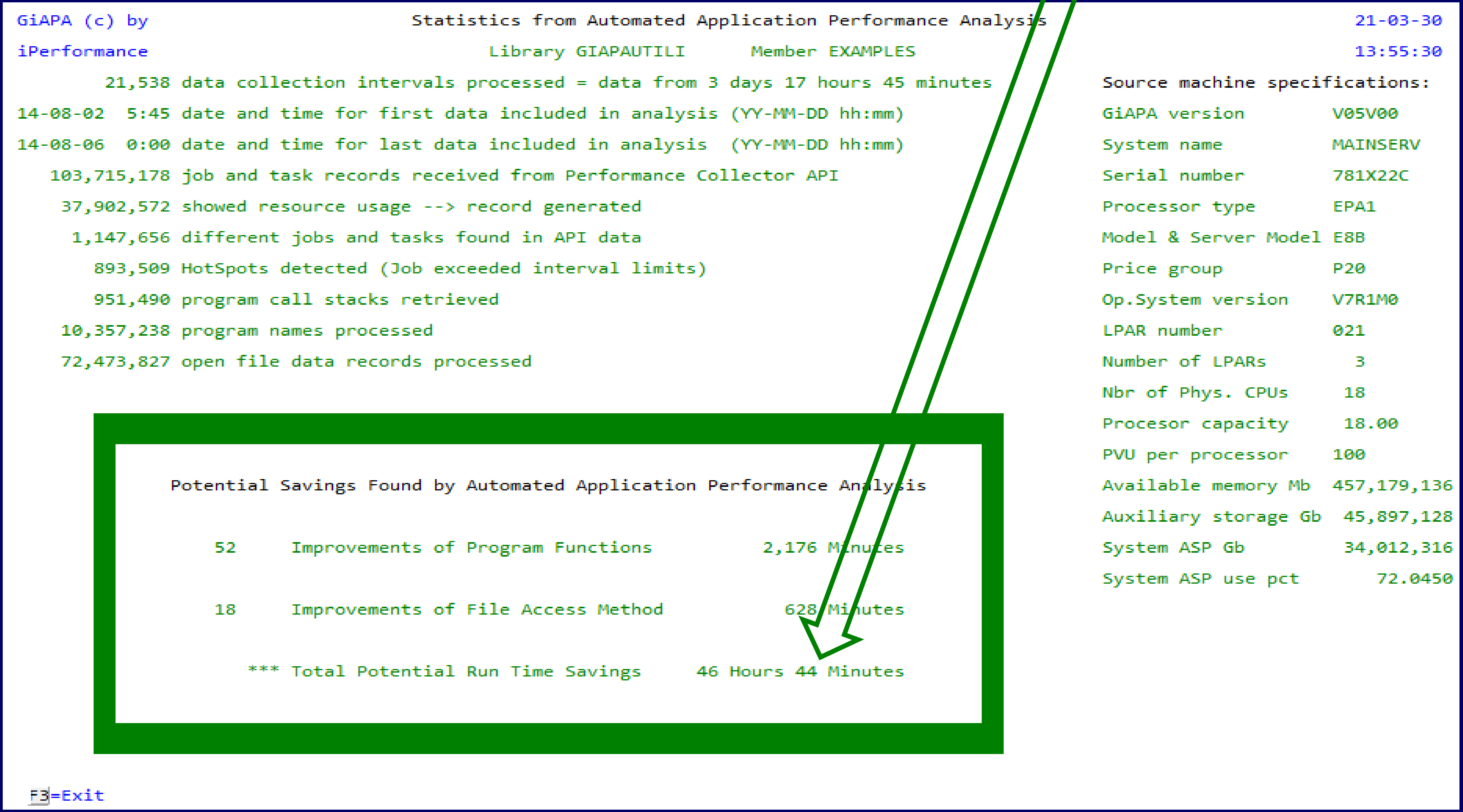
Business Partner IBM

www.giapa.com

Automatically generated optimization hints for:

- **Programs**
- **Access of data bases**

Output Generated : Overview of Potential Savings Found



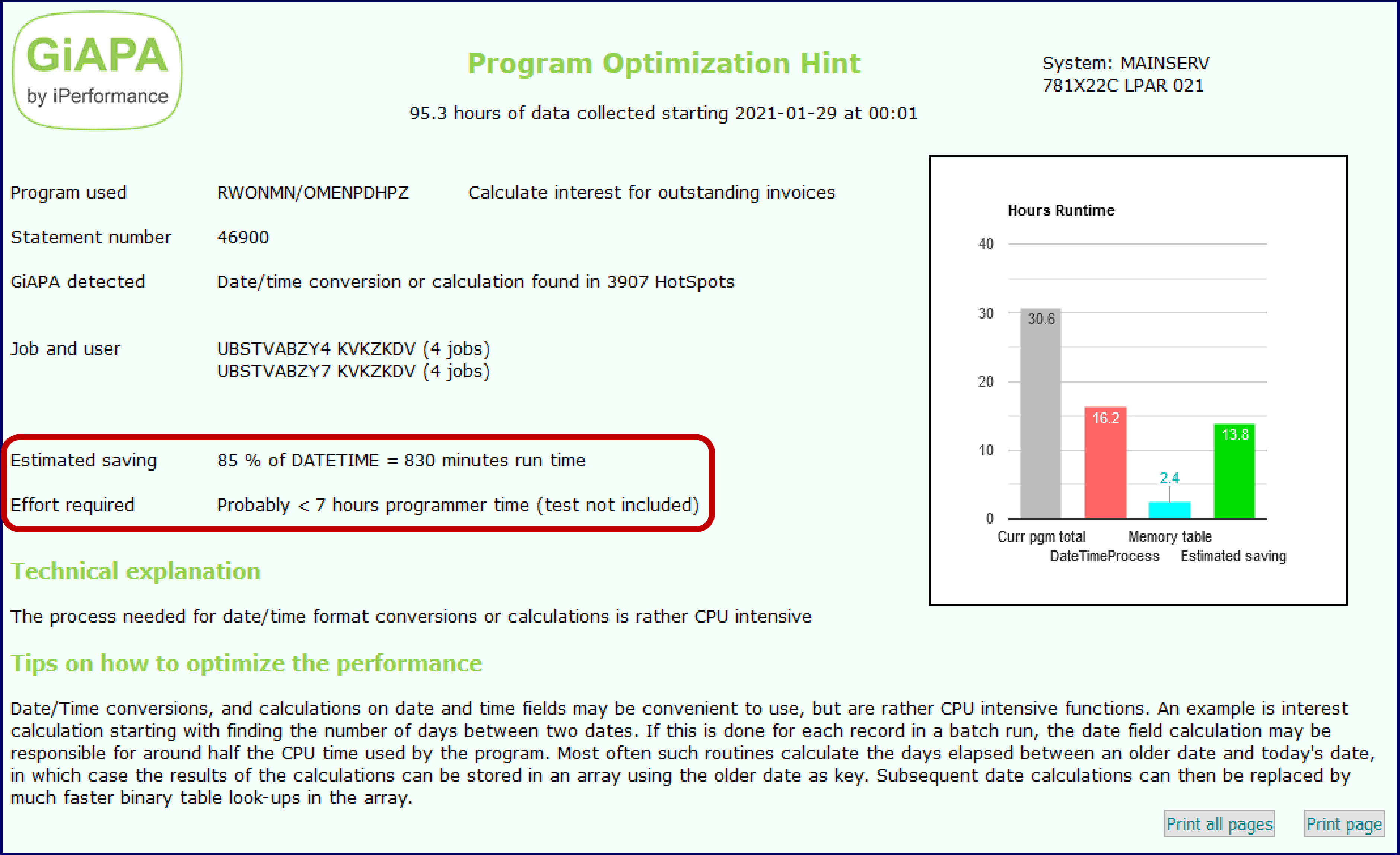
```
GiAPA (c) by                       Statistics from Automated Application Performance Analysis                  21-03-30
iPerformance                              Library GIAPAUTILI      Member EXAMPLES                               13:55:30
        21,538 data collection intervals processed = data from 3 days 17 hours 45 minutes     Source machine specifications:
14-08-02  5:45 date and time for first data included in analysis (YY-MM-DD hh:mm)             GiAPA version        V05V00
14-08-06  0:00 date and time for last data included in analysis  (YY-MM-DD hh:mm)             System name          MAINSERV
   103,715,178 job and task records received from Performance Collector API                   Serial number        781X22C
    37,902,572 showed resource usage --> record generated                                      Processor type       EPA1
     1,147,656 different jobs and tasks found in API data                                      Model & Server Model E8B
       893,509 HotSpots detected (Job exceeded interval limits)                                Price group          P20
       951,490 program call stacks retrieved                                                   Op.System version    V7R1M0
    10,357,238 program names processed                                                         LPAR number          021
    72,473,827 open file data records processed                                                Number of LPARs        3
                                                                                               Nbr of Phys. CPUs     18
                                                                                               Procesor capacity     18.00
                                                                                               PVU per processor     100
     Potential Savings Found by Automated Application Performance Analysis                     Available memory Mb  457,179,136
                                                                                               Auxiliary storage Gb  45,897,128
         52      Improvements of Program Functions          2,176 Minutes                      System ASP Gb         34,012,316
                                                                                               System ASP use pct       72.0450
         18      Improvements of File Access Method           628 Minutes

            *** Total Potential Run Time Savings     46 Hours 44 Minutes


 F3=Exit
```

Request trial installation and get this overview of savings possible plus
one free example (please see next slides) based data from your productions server.
For a detailed description of a Free Trial please use this link:
https://www.giapa.com/GiAPA_HowToRunFreeTrial.pdf

Example of Optimization Hint for a Program



Program Optimization Hint

System: MAINSERV
781X22C LPAR 021

95.3 hours of data collected starting 2021-01-29 at 00:01

Program used	RWONMN/OMENPDHPZ	Calculate interest for outstanding invoices
Statement number	46900	
GiAPA detected	Date/time conversion or calculation found in 3907 HotSpots	
Job and user	UBSTVABZY4 KVKZKDV (4 jobs) UBSTVABZY7 KVKZKDV (4 jobs)	
Estimated saving	85 % of DATETIME = 830 minutes run time	
Effort required	Probably < 7 hours programmer time (test not included)	

Technical explanation

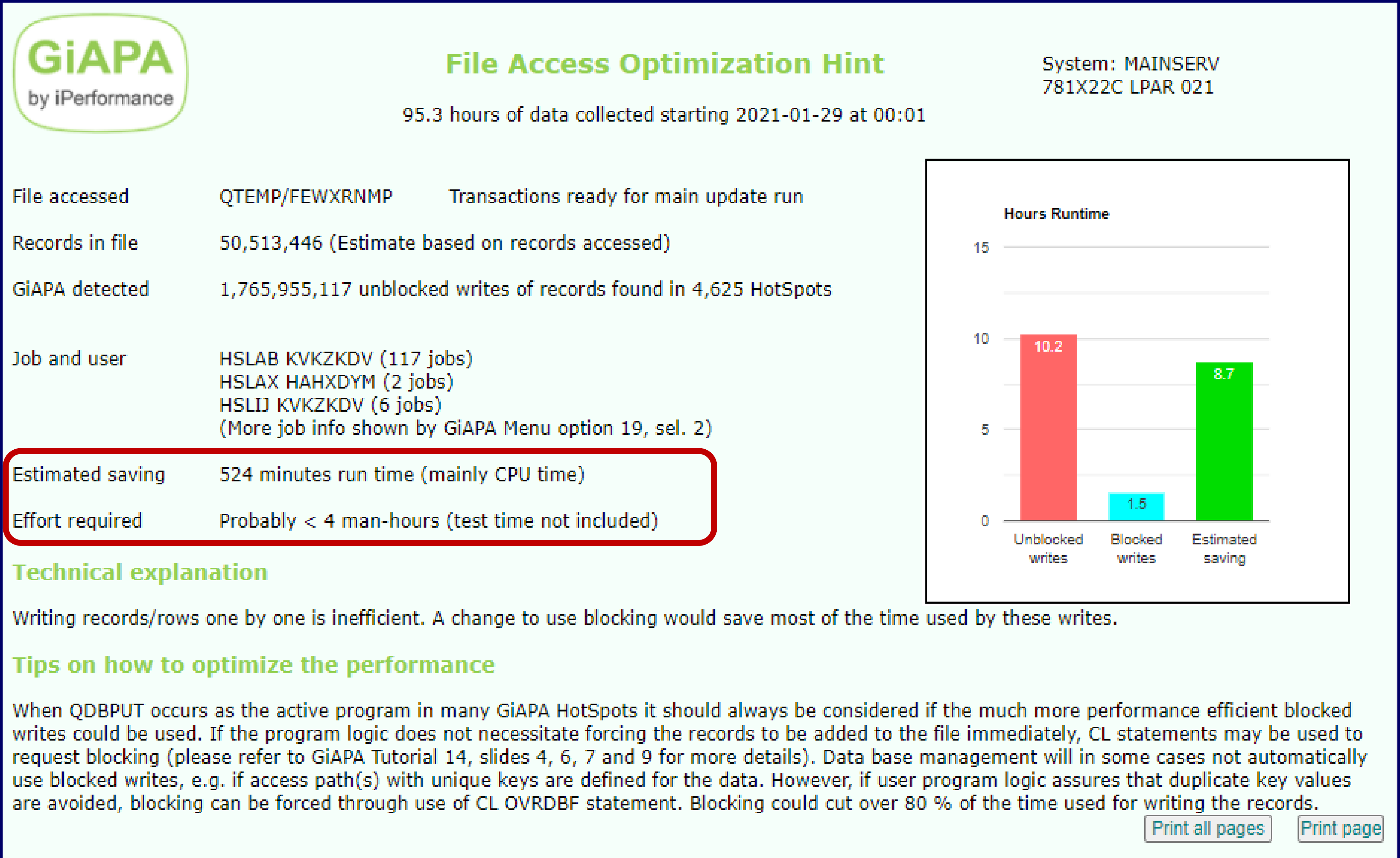
The process needed for date/time format conversions or calculations is rather CPU intensive

Tips on how to optimize the performance

Date/Time conversions, and calculations on date and time fields may be convenient to use, but are rather CPU intensive functions. An example is interest calculation starting with finding the number of days between two dates. If this is done for each record in a batch run, the date field calculation may be responsible for around half the CPU time used by the program. Most often such routines calculate the days elapsed between an older date and today's date, in which case the results of the calculations can be stored in an array using the older date as key. Subsequent date calculations can then be replaced by much faster binary table look-ups in the array.

Data collection uses less than 0.1 % CPU. The results are produced 100 % automatically.
All jobs are analyzed, and only programs with optimization potential are reported.

Example of Optimization Hint for Accessing a Data Base



File Access Optimization Hint

System: MAINSERV
781X22C LPAR 021

95.3 hours of data collected starting 2021-01-29 at 00:01

File accessed	QTEMP/FEWXRNMP Transactions ready for main update run
Records in file	50,513,446 (Estimate based on records accessed)
GiAPA detected	1,765,955,117 unblocked writes of records found in 4,625 HotSpots
Job and user	HSLAB KVKZKDV (117 jobs) HSLAX HAHXDYM (2 jobs) HSLIJ KVKZKDV (6 jobs) (More job info shown by GiAPA Menu option 19, sel. 2)
Estimated saving	524 minutes run time (mainly CPU time)
Effort required	Probably < 4 man-hours (test time not included)

Technical explanation

Writing records/rows one by one is inefficient. A change to use blocking would save most of the time used by these writes.

Tips on how to optimize the performance

When QDBPUT occurs as the active program in many GiAPA HotSpots it should always be considered if the much more performance efficient blocked writes could be used. If the program logic does not necessitate forcing the records to be added to the file immediately, CL statements may be used to request blocking (please refer to GiAPA Tutorial 14, slides 4, 6, 7 and 9 for more details). Data base management will in some cases not automatically use blocked writes, e.g. if access path(s) with unique keys are defined for the data. However, if user program logic assures that duplicate key values are avoided, blocking can be forced through use of CL OVRDBF statement. Blocking could cut over 80 % of the time used for writing the records.

For more details please check https://www.giapa.com/product-intro/giapa-video-4-minutes
or https://www.giapa.com/GiAPA2021Presentation%20(Published)/