

With **IBM Power Systems** running IBM i you have
a lot of resources available ...



... are they used **optimally?**

VERIFY THAT RESOURCES ARE USED EFFICIENTLY:
LET GIAPA “X-RAY” YOUR SERVER



GiAPA
by iPerformance



Performance data is **analyzed** automatically
– no external experts needed!

OPERATIONS

... identify reasons for peaks experienced:

- What happened
- Which job is the culprit
- Responsible user
- Which program and source statement

In addition, GiAPA also

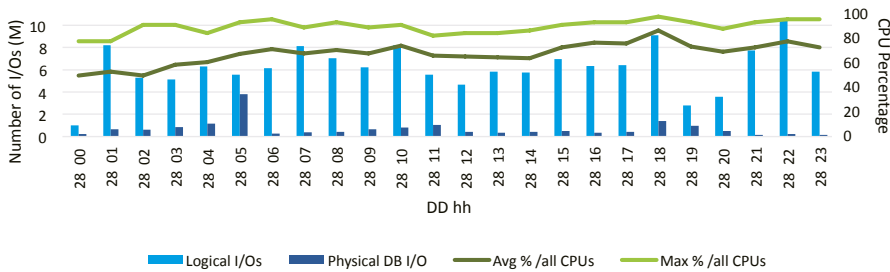
- ✓ warns QSYSOPR if a job is looping
- ✓ reports who used which Query when
- ✓ lists files not used the last xx months
- ✓ shows temporary index generations
- ✓ ... and much more!



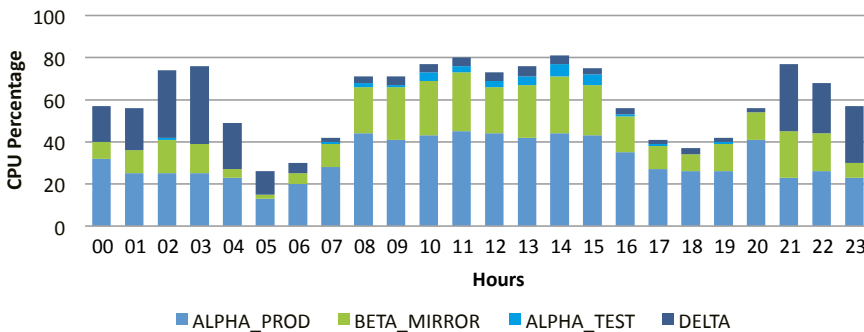
IT MANAGEMENT

... easily get the full overview with
a “Good Morning Report”

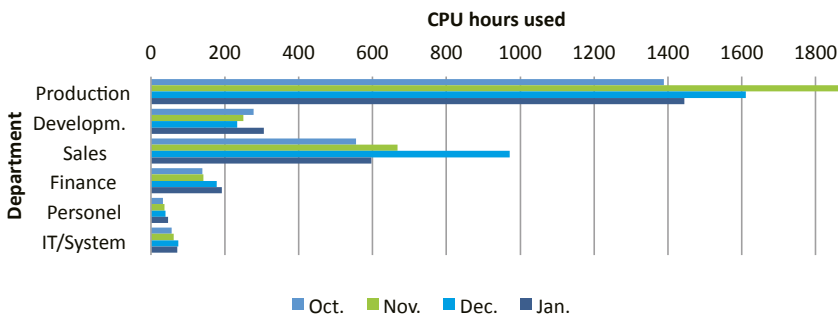
24 Hour Resource Usage - CPU% and I/Os per Hour



CPU % for all LPARs on Server



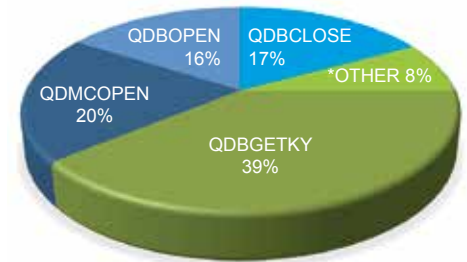
Department CPU Usage Trends per Month



DEVELOPMENT

...get a powerful quality
control tool that pinpoints
optimization potential
down to source code line.

For a job assumed to run OK,
GiAPA returned the following cues
on how to improve performance:

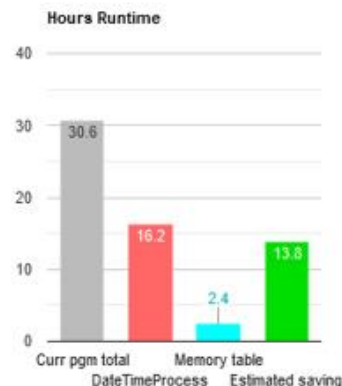


1. 53% runtime may be gained by keeping files open
2. QDBGETKY (Read by key) used 39% runtime. GiAPA reported that 176 million reads (= 135 + 41) are used to access 1359 records (= 731 + 628) in two files; a table within the program could save 85% of the reads.

GiAPA “File Analysis Summary” report for Job XYZ:

File Name	I/O	Writes	Reads	Other I/Os	# of records in file	Superfluous I/Os
A10DQA04	I	0	135.580.207	0	731	135.579.476
R1CBDI01	I	0	41.387.642	0	628	41.387.014
(Other files with fewer I/Os not shown here)						
*** Total		62.031	203.212.888	17.079		176.966.490

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

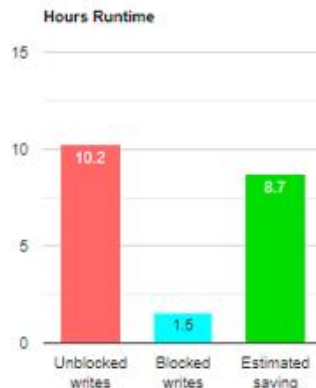
Date/Time conversions, and calculations on date and time fields may be convenient to use, but are rather CPU intensive functions. An example is interest calculation starting with finding the number of days between two dates. If this is done for each record in a batch run, the date field calculation may 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.

[Print all pages](#)

[Print page](#)

Examples of GiAPA's fully automated Performance Analysis

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.

[Print all pages](#)

[Print page](#)

62%

is the average performance optimization obtained using GiAPA to analyze slow running applications.

96%

improvement in response time was obtained when GiAPA analyzed a frequently used inter-active transaction at a Swedish manufacturing company.

86%

runtime reduction was found by GiAPA in a daily five hour batch job at a large German wholesale company.

94%

CPU usage corresponding to 3½ hours was saved by an American information processing company when GiAPA found an unintended loop in a frequently used job.

57%

decrease in CPU usage was the result when a major American IT supplier applied GiAPA's cues for optimization of a payroll application.

97%

of the logical I/Os were saved when a read routine of a weekly batch job was changed at a major Scandinavian bank; runtime was reduced from 33 to 7 hours, saving 8 hours CPU time.

0.1%

CPU is the average overhead used by GiAPA to collect detailed performance data for all jobs and tasks every 15 seconds.

A leading global provider of supply chain solutions reported that five years use of GiAPA had saved them

€1,000,000!

Want to know more?
www.giapa.com contains

- Four minutes introductory video
- Product presentation video
- Complete GiAPA course as online tutorials
- Download and installation instructions
- References and success stories

GiAPA reseller:

SOSY

Contact: Helle Mering
hm@sosy.dk
+45 2030 3109



iPerformance ApS, Denmark
sales@giapa.com
Tel. +45 5835 3305



Power Systems and IBM i are trade marks of International Business Machines Corporation