TurboTune

Performance is Profit®

TurboTune

Comparison to IAM and Methodology

Critical Path Software Inc.
Phone 212-686-4350

CRITICAL PATH SOFTWARE

TurboTune
Methodology and Comparison to IAM

TurboTune compared to IAM

These two products are designed to improve On-line and Batch VSAM performance through the reduction the number of necessary I/O's.

- IAM provides an alternative access technique for some VSAM access files.
- IAM cannot be used for Catalogs, CICS, Page Data sets, RLS etc.
- IAM utilizes its own data buffering technique that implements data compression and de-compression, which requires additional CPU usage.
    - As not all files do not compress and de-compress well due to differences in their structure. Therefore, the additional amount of CPU required to use IAM will vary by file however there will be an increase.

- TurboTune tunes your VSAM datasets to their optimum performance levels.
- TurboTune can be used for all VSAM files.
- TurboTune only improves your VSAM file selections. As such it will always decrease CPU usage.
- TurboTune allows you to use all the most advanced CICS and VSAM data sharing options available under OS/390 or z/OS operation systems.
- TurboTune has never failed to obtain CPU savings and improve performance.

TurboTune proprietary software far exceeds the capabilities of any other software on the market today. Assisting clients in the understanding of what they are ***not doing*** to improve performance rather than simply analyzing what they are doing. TurboTune provides solutions. TurboTune has no direct equivalent competitor.

Test for yourself:

Suggested Evaluation Process:

1. Select a set of VSAM files requiring optimization and have not been converted to IAM.
2. Run a mainframe performance tool against SMF data to measure CPU utilization and I/O activity against these files.
3. Apply the TurboTune changes against the files. Then rerun the same mainframe performance tool against SMF data to measure CPU utilization and I/O activity against the files. Record data.
4. Convert the unaltered VSAM files to IAM. Then rerun the same mainframe performance tool against SMF data to measure CPU utilization and I/O activity against the files.

5. Compare the IAM results against the TurboTune results making adjustment for variances in insert, replace, delete and retrieve activity against the files. We believe that

This comparison will show the differential savings achieved by utilizing TurboTune rather than IAM.

TurboTune Methodology

TurboTune was developed to analyze mainframe VSAM Data Sets. This automated tool generates a series of reports which describe in detail those data sets which are not optimized. TurboTune performs an average of 1,500 calculations per file to generate the appropriate VSAM procedures to fix problems. It is based on a knowledge database which contains over 5.5 million system statistics gathered from hundreds of client systems over the past 40 years.

TurboTune is a suite of scientifically designed mathematical relationships which develop a unique performance model for each client based upon comparison to hundreds of successful tuning modules across the globe. TurboTune solutions are based on a client's unique individual hardware and software usage. This allows TurboTune to correctly evaluate the performance level of the unique VSAM based system. TurboTune solutions provide the correct current tuning procedures and identifies optimization techniques that may be missing from the application data flow.

Other tuning and performance products, including IAM, generally possess a more generic design in order that they can be quickly installed on all computers. This approach, however, will invariably lead to findings which will be less than optimal.

TurboTune is targeted to analyze application software which operates within VSAM, DB2 and IMS environments. It also includes a built-in DASD analyzer. It has approximately 75 internal analytical performance programs which look at specific statistical information from the following sources:

- **File usage**
- **Listcats**
- **Proc Lib information**
- **Job Lib information**
- **SMF/RMF information**
- **Sysout JCL**

The statistics will include such measurements as:

- **File sizes**
- **File placement**
- **All VSAM options**
- **CI and CA sizes**
- **BUFND**
- **BUFNI**
- **On-line update reqsts**
- **Add/Delete/Browse**
- **Waits-on-string**
- **Program fetches**

- **Program usage counts**
- **PPTs/FCTs/PCTs/TCTs**
- **Compile options**
- **Link options**
- **Hardware model**
- **Volumes of DASD**
- **Operating system**
- **Number of terminals**
- **Partition size**
- **Region size**
- **Blocking factors**
- **Core sort sizes**
- **Megs of real memory**
- **MIP ratings**
- **Number of channels**
- **Channel utilization**
- **DASD utilization**
- **Terminal response times**
- **Numerous other statistics**

TurboTune will mathematically form several hundred ratios from these statistics and insert this information into a spreadsheet where it processes millions of iterations. It then compares a client's performance statistics with statistics deemed to be ‘best in class’. The power of this approach is based upon the iterative processing of hundreds of "what-if" scenarios in order to customize a performance model which is based on an applications usage and the client's unique and individual mainframe configuration. TurboTune output is series of reports with detailed solutions concerning which files require improvement.

Automation: TurboTune then produces appropriate improved file definitions for all data sets requiring improvement. These definitions are “wrapped” in JCL allowing the client to easily improve one or one thousand files at a time in a test environment, test the improved file, then move it to production and begin savings while experiencing the performance improvements.

A typical client engagement will include the following:

- Dynamic adjustment of settings for file usage
- File-by-file comparison to an expert database
- Removal of obsolete options/parameters
- Calculation of better free space settings on a file-by-file basis
- Reduction of multiple extents
- Buffer usage improvement
- Perform 1,500 calculations per file
- Shorten batch windows
- Improve Cache Controllers
- Review of the following:
    - IMS/DB/DC
    - DB2 & CICS
    - System files
    - IDMS drivers
    - ADABAS “
    - VSAM
    - TMON
    - Omegamon
    - Cache Controllers
    - RMF
    - SMF
    - Catalogs
    - Buffering Tool
    - Etc.

TurboTune Success

The only function of TurboTune is to improve performance thereby reducing costs in a mainframe environment.

A TurboTune engagement has never been completed without either improving performance and/or reducing costs to operate application software.

TurboTune is particularly effective in a merger environment. An accurate analysis of the respective firm’s data can determine the:

- Degree of fit between different sets of data
- Compatibility between different applications
- Best-of-breed performance on a feature to feature basis
- Anticipated cost of combined operation

Various statistics have been developed to effectively measure performance improvements for pre vs. post optimization:

- The most obvious method of measuring savings is to refer to the existing charge back system. Savings are occurring when monthly charges to process an application cost less than before tuning. The volume of workload must always be considered/equivalent in any savings calculation.

- The unique KSDS Data to Index EXCP ratio measures the speed of processing in both Batch and Online VSAM files. Savings will accrue between pre-post measurements up to a maximum ratio of 7.5, which is the measure of an optimum system.

- The unique EXCP to ACTIVITY ratio is the number of EXCP's (physical I/O) it takes to perform an activity (INSERT, DELETE, UPDATE, or RETRIEVE) upon data within a VSAM file. Savings occur as this ratio approaches 0.003 (optimum system).

- Another common technique chosen to measure savings is through the use of SAS reports and SMF/RMF data to evaluate before/after statistics for: CPU Seconds, Physical I/O, DASD usage and Job Run Time.

- CICS shut down statistics will show the number of transactions processed online and the amount of CPU resource which is required to process these transactions. The comparative average milliseconds per transaction calculations from these statistics will illustrate CPU savings on a before/after basis.

- The average response time of transactions measured by TMON and/or OMEGAMON can be used to determine performance improvements.

- The before/after recording of CPU, I/O, EXCPs, DASD and RUN TIME can verify savings for significant key jobs.

- The extraction of Batch Job Run Time data from a MICS data base and measuring wall clock run time will allow before and after savings measurement. This will require a SAS report or similar tool.

- STROBE, in some instances, can be used to measure before/after improvement for a single application.

- On a more simplistic note, clock run time can be used as a measure of savings (ie: an overnight job is completed at 6 AM vs. 9 AM).

Additional Savings Opportunities
Significant savings can be realized by extending the time between hardware upgrades as well as gaining improved labor efficiencies within the client IT and user communities. Degradation in application performance over time leads many companies to conclude that their system is CPU bound and that an expensive upgrade remains as the only solution. This is not always the case. Critical Path Software client engagements over the past 36 years have led to the conclusion that companies are most often "I/O bound", and that an expensive upgrade is often NOT the optimal or most cost-effective solution. The challenge is knowing how to most effectively reduce I/O contention. This improves the throughput of all applications which run on the same computer and share the same storage resources.

Problems with VSAM vary however TurboTune algorithms identify and provide solutions for most, if not all, elements responsible for creating increased operational cost in the following areas:

- Slow online response time with increasing degradation
- Batch windows increasing
- CPU utilization increasing rapidly
- DASD costs escalating
- Costly reruns
- Downtime