

BEYOND OPTIMIZATION: HOW USING A PERFORMANCE MONITOR FOR ADABAS SAVES MONEY



TABLE OF CONTENTS

- 1 The value of a performance monitor
- 2 Selecting a performance monitor for Adabas
- 3 Conclusion

Performance monitors have been available for Adabas since the early 1980s and continue to be an essential tool. Aside from analyzing basic performance metrics, performance monitors are used by DBAs and programmers alike to drive down costs, recoup expenses and delay upgrades.

Performance monitors can be used in many different ways—for example, to:

- Collect chargeback information
- Debug programs
- Resolve error messages
- Identify programs using excessive resources
- Track historical resource utilization
- Tune the database to achieve optimal performance

This white paper explains why a performance monitor is a must-have. You'll learn how to compare monitors for Adabas. You'll also see why there is only one performance monitor you can count on to reliably optimize performance, deliver savings in support costs, increase revenue with chargebacks and delay machine upgrades. That's Adabas Review.

The value of a performance monitor

The traditional role of using performance monitors to optimize performance has diminished in recent years as CPUs, memory and I/O devices have gotten faster and cheaper. In contrast, the cost of DBAs and programmers to operate systems has risen while the number of employees with internal knowledge of the database and programs has declined. Consequently, the role of performance monitors now focuses on opportunities to recover costs and reduce the amount of labor required to debug problems.

Chargeback is a critical feature for many organizations as it translates into real money. Consequently, it can be one of the more visible uses of a performance monitor.

Here's what you can do with a performance monitor:

Recoup costs with chargebacks

Many organizations attempt to capture computer-resource consumption and bill departments accordingly. This process is referred to as chargeback. For non-database tasks, such as copying a dataset or printing a report, capturing this information and attributing it to a user is relatively straightforward using operating system functions. However, calls to the database can produce a significant amount of CPU usage and I/O that are hidden inside the database job and not attributed to individual users at the operating-system level.

Since databases are often one of the largest consumers of machine resources, organizations interested in chargebacks would naturally like to obtain information about which users are responsible for using these resources. Generally, this is not an exact science. The database itself does not provide this information in a simple form so it is derived by a performance monitor looking at the command and/or the number of I/Os attributed to the command.

Chargeback is a critical feature for many organizations as it translates into real money. Consequently, it can be one of the more visible uses of a performance monitor.

Reduce support costs with debugging and error resolution

While program debuggers allow you step through a program line by line or by stopping at various breakpoints, a performance monitor takes debugging and error resolution to the next level. Even when a program is executing as expected, problems can still occur. If the Data Definition Module (DDM) is pointing at the wrong database or file number, the call may execute when you expect it to but may point to the wrong location. This can result in unusual error messages or unexpected data results. Tracing database calls using a performance monitor filtered by user, program or file can help quickly determine what is going on and why.

Along with determining where a call is going, additional information about the call is available in the performance monitor. Users can analyze the contents of the Adabas Control Block (ACB) and the various buffers to determine what is actually being seen and sent by the database. For example, analyzing the sub-code provided on the Adabas Control Block can assist users in determining the sub-code associated with an error message that may not be provided in the error message received by the program. Using a performance monitor to debug programs and resolve errors can save a great deal of time and energy for a DBA or programmer trying to resolve an issue. Saving time and resources can reduce support costs.

Save with database tuning

Tuning the database involves analyzing the various pools, queues and tables along with overall throughput to determine where parameter changes can improve performance. However, improvements to infrastructure have reduced response times for database calls so radically that relatively few shops worry about tuning the database. Also, improvements to search algorithms in some of the larger pools and queues, along with dynamic sizing determined by the database, make it less critical for DBAs to perform this analysis. Due to the lack of tuning efforts over the years, many organizations may not realize they have some low-hanging fruit to gain easy savings. Depending on how you pay for your resources, tuning the database may either save real money or help put off a machine upgrade.

Monitor resource utilization

Despite all the performance improvements in the database and the infrastructure that supports it, poorly written programs can still create bottlenecks. Individual commands may run quickly. But an abundance of unnecessary calls increases run times. Consequently, there will always be a reason to monitor application program performance and the resources consumed by these programs. Excessive calls or I/Os are not the only culprits. Large numbers of held records or other resources can cause unexpected error messages or delays in processing.

Monitoring resource utilization is often necessary to "prove" the database is not running poorly and if an application problem really exists. DBAs often find monitoring resources is valuable if only as evidence the database is not at fault for performance problems.

Using a performance monitor to debug programs and resolve errors can

SAVE A GREAT DEAL OF TIME & ENERGY

for a DBA or programmer trying to resolve an issue. Saving time and resources can reduce support costs.

Plan for the future with historical analysis

Tracking historical data about detailed performance figures is an important function because many uses for the data only become apparent over time. For example, in order to tune the database, it is necessary to understand what is “normal” for the database by watching performance of queues, buffer efficiency and I/Os per table for a period of time. In addition, performance monitors can also record the number of commands executed against a particular file or database as a measure of CPU consumption and track the number of records in a file as a proxy for disk space used. This information can be used to extrapolate future growth for capacity planning. It is also possible to analyze the impact of a particular event, such as year-end processing or a hardware upgrade. Both of these examples can provide valuable insight that is simply not available without historical data. By the time you need the information, it is too late to start collecting it. So the sooner you start collecting data, the better.

Selecting a performance monitor for Adabas

There are currently three products on the market that provide performance monitoring for mainframe Adabas: Software AG's Adabas Review, TRIM by Treehouse Software and APAS® by Computer Associates. These are mature products that provide access to core Adabas internal information, full or selective logging to the command log, a history subsystem, basic tracing of commands and analysis of command log data, among other features. Let's take a closer look at the differences between these monitors.

FEATURE COMPARISON

Feature	Adabas Review	TRIM	APAS (Sunset 2012)
Can run in separate address space	Yes	No	No
Can run as a subtask	Yes	No	Yes
Supports Adabas Parallel Services & Adabas Cluster Services	Yes	No	No
Supports new Adabas features at release	Yes	No	No
Installation supported by System Maintenance Aid (SMA)	Yes	No	No
Offers chargeback support	Yes	No	Yes
Simultaneously traces multiple databases	Yes	No	No
Highly customizable online reports	Yes	No	Yes
Collects historical statistics	Yes	No	Yes
Provides sample reports	Yes	No	Yes
Provides user exits for custom reports	Yes	No	No
Online reporting for all available fields	Yes	No	Yes
Secure start, stop and display reports	Yes	No	Yes
Secure report modification	Yes	No	Yes
No new language needed to create reports	Yes	No	No

Stability and performance

TRIM and APAS were released in the early 1980s, and while both products have evolved, they do not appear to have undergone major rewrites. Adabas Review underwent a fairly substantial development in the mid-90s with a good portion of the rewrite specifically geared toward stability and performance along with developing a platform to support future enhancements. Adabas Review supports hub mode, which allows most processing to run in isolation in a separate address space. This improves database stability and reduces bottlenecks in the database. Hub mode is more complex to set up but runs as a completely separate job or hub apart from the database. A single hub can process calls from multiple databases, which allows programs that make calls to more than one database to be traced in a single place. Running a single trace for a program can greatly simplify the debugging process. TRIM and APAS do not offer this functionality.

TRIM runs in User Exit 4 as part of the Adabas Started Task. This implementation can be problematic on two fronts. From a stability perspective, if the user exit code has a problem, it can bring down the whole database. In addition, once control is turned over to the user exit, any overhead to perform monitoring functions causes Adabas to wait while TRIM does its work. This can become a performance bottleneck. A monitor running as a User Exit 4 is more prone to slowing down the database.

Processing options

Adabas Review offers hub mode, which runs processes from multiple databases in isolation. Local mode is also available and uses a subtask so that code problems do not affect the database.

Performance monitors are the must-have product for organizations using the mainframe.

Monitors practically pay for themselves in many ways, from recouping costs with chargebacks to reducing support costs by resolving errors faster.



About the author

Scott Ferrell

InTandem Technology, Inc.

Scott Ferrell is a consultant with InTandem Technology, Inc. He has worked with Software AG products for more than 30 years as a Database Administrator (DBA), a trainer/mentor and as a developer of systems software.

Guaranteed support

Adabas Review is guaranteed to work with the latest release of Adabas. In fact, new releases of Adabas are not made available until Adabas Review supports them. Adabas Review also provides monitoring support for Adabas Parallel Services and Adabas Cluster Services, which neither TRIM nor APAS provides.

Simplicity of use with complex capabilities

In many ways, the online system for Adabas Review feels like the bridge between APAS and TRIM. Adabas Review provides a menu-driven system that has much of the simplicity TRIM is known for. Adabas Review also allows for a great deal of customization. Fields and values are selected from easy-to-use input screens. APAS, on the other hand, requires a programming language to customize reports. Thus Adabas Review provides support for complex reports like APAS but with much of the simplicity of TRIM.

TRIM's largely menu-driven interface for monitoring is relatively straightforward to use. But it does not offer as much customization as the other monitors. For example, TRIM has some facilities for chargeback but requires coding to implement. APAS provides sample code to generate System Management Facilities (SMF) records from command log data. While a strength of APAS, new support for SMF in Adabas 8 is a strong replacement for this facility. In addition, the formulas used by APAS to calculate CPU for Adabas commands have not been changed since Adabas 4. Adabas Review provides sample code to do chargeback along with several alternative methods.

Report customization

Adabas Review provides numerous sample reports and allows for considerable customization of reports using fields picked from lists with selection criteria added by the user. TRIM has fewer customization options available but offers many canned reports via an easily navigable menu interface. APAS offers many customization options but requires that users learn a query language to build custom requests.

Adabas Review also provides numerous "starter" reports that can be used as is or modified. Most commonly used reports are provided so users will typically not need to create a lot of new reports from scratch.

When reports are created or modified, Adabas Review does not require users to learn a language to make changes. Selection criteria are modified by selecting fields from a list of provided fields and applying rules and assigning values (e.g., DURATION > 5). Consequently, there is almost no learning curve involved with creating Adabas Review reports. All detail and summary reports in Adabas Review have an option for a user exit, which allows greater customization. This is clearly an advanced feature as the user exits are written in Assembler. If you need highly specialized functionality, you have the ability to develop custom solutions. Numerous fields that are not provided on the command log from the user address space are also provided by Adabas Review. Items such as user type, job number, stepname and LPAR name along with the option for user-defined fields can be passed to Adabas Review and are available in reports.

Conclusion

Performance monitors are the must-have product for organizations using the mainframe. Monitors practically pay for themselves in many ways, from recouping costs with chargebacks to reducing support costs by resolving errors faster. Adabas Review is your best choice in performance monitoring as it will ALWAYS work with the latest release of Adabas and also provides many features that APAS and TRIM do not.

ABOUT SOFTWARE AG

The digital transformation is changing enterprise IT landscapes from inflexible application silos to modern software platform-driven IT architectures which deliver the openness, speed and agility needed to enable the digital real-time enterprise. Software AG offers the first end-to-end Digital Business Platform, based on open standards, with integration, process management, in-memory data, adaptive application development, real-time analytics and enterprise architecture management as core building blocks. The modular platform allows users to develop the next generation of application systems to build their digital future, today. With over 45 years of customer-centric innovation, Software AG is ranked as a leader in many innovative and digital technology categories. Learn more at www.SoftwareAG.com.

© 2017 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

SAG_Adabas-PerfMon_4PG_WP_Apr17

