iTech-Ed

Virtual IMS user group: Newsletter 28

Welcome to the Virtual IMS user group newsletter. The Virtual IMS user group at www.fundi.com/virtualims is an independently-operated vendor-neutral site run by and for the IMS user community.

Virtual IMS user group presentation

The latest meeting of the Virtual IMS user group was meant to include a presentation entitled, “IMS performance – taming the beast”. Unfortunately, there was no presentation given, but a copy of the slides is available.

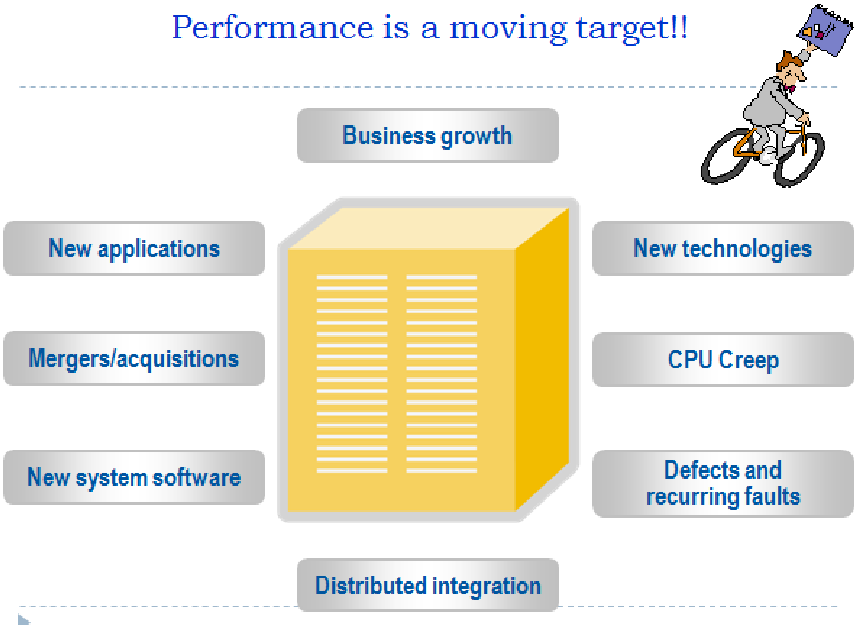


Figure 1: Performance is a moving target

Why is it necessary to perform ongoing monitoring of the IMS subsystem and applications performance? It’s a moving target (see Figure 1). The single biggest issue is migration to a new IMS version, which results in more work and testing that any other single activity. Then there are the environmental changes, such as switching to SMS (contention of DS placing), network changes, DR considerations (eg disk mirroring being implemented may increase response times and trigger a lot of timeouts), and hardware changes. Every one of these needs to be monitored – even if nobody touches the applications.

SMP maintenance includes looking at all the holddata

Contents:

for things that have to be rebound (usually every 3-6 months). Bufferpool settings are important. If you have enough memory to just make them big, check them every couple of months. With zparms, some people don't ever change them, but the number of log buffers and checkpoint frequency can have a major impact on performance. Make sure back-ups run and run correctly – especially for the catalog and directory.

In addition, you need to monitor the underlying VSAM datasets used for DIRECTORY/CATALOG tablespaces/indexspaces to ensure that they never run out of space. And regularly check and reorg the CATALOG/ DIRECTORY as needed to improve performance. On occasion, also run a health CHECK against the DB2 Catalog/Directory (eg run PGM=DSN1COPY, PARM=(CHECK) and PGM=DSN1CHKR against the CATALOG/DIRECTRY datasets, etc).

Ensure that DSNDB07 (SORT TEMPORARY WORKSPACE) datasets are big enough. Also monitor log usage and ensure that logs are large enough during periods of heavy updates. DB2 starts to scream when it begins running low on active logs.

Eventually, one can begin advanced monitoring of DB2 subsystems for possible areas of poor performance by watching for thresholds being hit. Some variables to watch are: QBSTDMC, QBSTSPD, QBSTREE, QBSTWEE, QBSTWFF, QBSTMAX, QBSTWFD, QBSTWKPD, QISEFAIL, QISTRPLM, QISTRSTG, QISTRMAX, QXNSMIAP, QJSTWTB, QJSTTVC, QWSDCKPT, Q3STCTHW.

And lastly, performance may deteriorate unless there is ongoing monitoring of the following:

- EDMPOOL usage (the smaller the EDM pool, the more critical monitoring becomes)
- Storage usage for DBM1
- Paging (you want low or zero paging for Bufferpools)
- How often datasets are physically being opened (and/or closed). It may be the zparm needs to be changed to keep more datasets open to prevent dataset thrashing (although keeping more files open may also have storage and checkpoint implications)
- Checkpoint frequency.

As businesses grow, new applications are required or the current vendor software needs new releases or enhancements are needed (eg front-end enhancements to back-end mainframes).

And for existing applications you can have performance creep that comes in the form of code/database changes, old code in a new environment, and

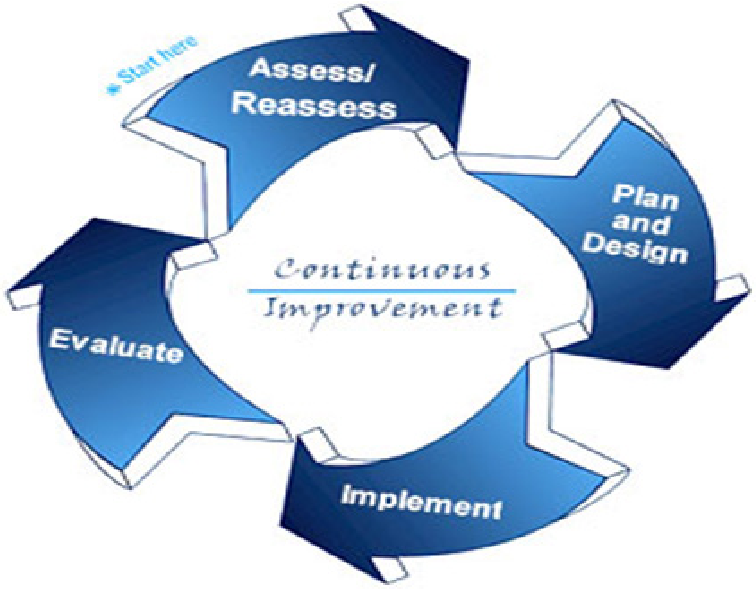
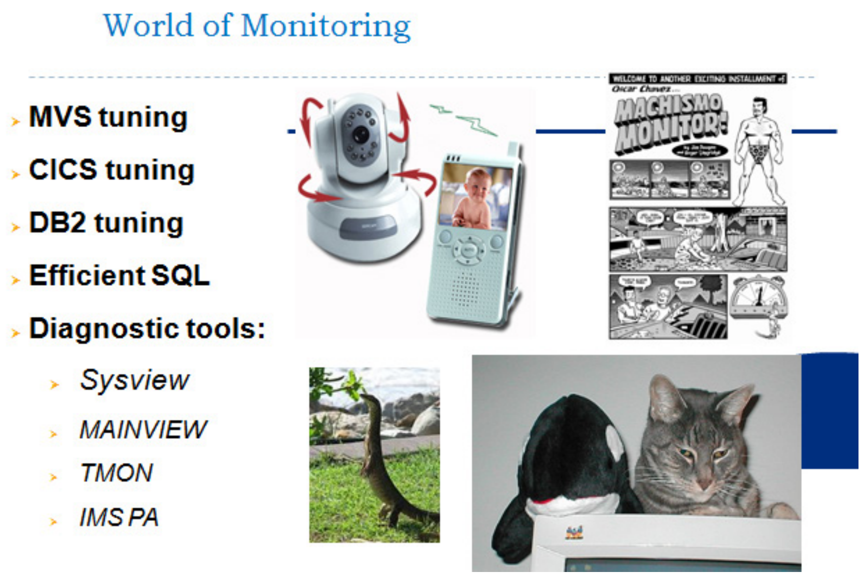


Figure 2: Performance Management

inefficiencies across a more complex environment. So even if you aren't making a conscious decision to grow the mainframe, it will happen.

Growth is inevitable. Despite the fact that organizations are not growing their mainframe, they probably are growing. The more dynamic the company, the larger the growth.


World of Monitoring

- MVS tuning
- CICS tuning
- DB2 tuning
- Efficient SQL
- Diagnostic tools:
  - *Sysview*
  - *MAINVIEW*
  - *TMON*
  - *IMS PA*

Figure 3: World of monitoring

The performance management capability maturity model is shown in Figure 2. Continuous improvement comes by considering your current ways of doing things as the worst ways of doing things, and making everyone responsible for improvements.

The unit of work by which we measure IMS workload is a transaction. An IMS transaction is a message from either a terminal or an application program that causes application program logic to be executed. A transaction profile typically covers the following:

- Host response times, including input queue time measurement and total elapsed time measurement.
- The CPU time required to process the transaction
- The number of database (DL/I and SQL) calls performed by the transaction
- The type of database calls performed (by database or table listing each database or table and the type of call).
- Number of I/Os required to perform this transaction.

A typical transaction generally performs many different types of functions. A transaction designated to process Internet banking can perform transfers and payments as well as balance inquiries. The profile by transaction is different for each of these functions. Analysis must be done to quantify these profiles to percentiles of the various functions of a transaction in order to make meaningful decisions.

This implies that making changes to a very low percentile-used function would not be beneficial, when compared to the total amount of CPU consumed and DL/I calls made. Cost as opposed to effort becomes a factor in most decisions.

When you have a transaction profiling repository with daily feeds into the repository from the IMS log records, other performance log records generated by vendor products, and SMF records, analysis and interpretation of the data become crucial issues.

The first issue in performance and tuning is to focus on good application design whenever possible. IMS subsystem performance issues are generally "fixed" issues, but you can influence areas within application design that affect performance. It’s always advisable to anticipate and correct problems before they occur (or before a new application is fully implemented).

The Systems Architect needs to be able to make broad performance assessments, to ensure that it falls well within acceptable limits. If these estimates suggest that achieving required performance will be a close-run thing, it is time to adjust the design or, if this is unacceptable, to call in a performance specialist. The world of monitoring is illustrated in Figure 3.

When it comes to IMS traces and monitors, generally, it is not necessary to turn on accounting classes 7 and 8, unless needed. In that case, you need to explicitly request the options to be turned on. Does your site use any packages?

Remember, use DB2 accounting data as the starting point for performance analysis. In particular, focus on Accounting class 3 (I/O and lock/latch wait times).

Accounting data also gives indications as to the type of I/O (random versus sequential), and the quantity of I/O.

Monitor the underlying VSAM datasets used for DIRECTORY/CATALOG tablespaces/indexspaces to ensure that they never run out of space, and regularly check and reorg the CATALOG/DIRECTORY as needed to improve performance.

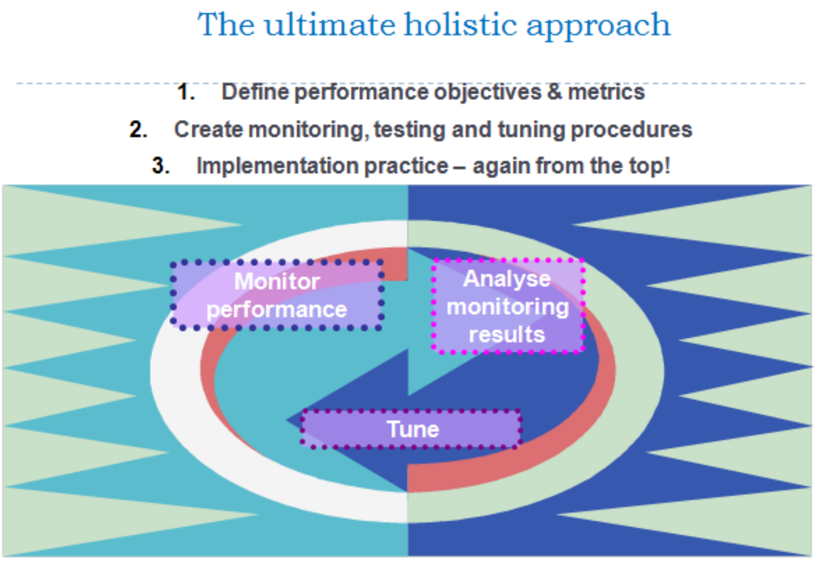
Version 4.1 of DB2 added a new address space to the DB2 subsystem. The new Stored Procedure Address Space runs as an allied address to DB2 itself. The Stored Procedure Address Space may be stopped without stopping the entire DB2 subsystem. The new address space insulates DB2 from application problems, for an increased measure of availability.

V5 of DB2 takes this architecture further, by adding support for multiple stored procedure Address Spaces. Multiple SPASs allow for additional flexibility, priority control, and scalability of business application growth. V5 allows you to use the MVS Workload Manager to establish the stored procedure address spaces. There can be many WLM address spaces running the stored procedures.

With the latest versions of DB2, typically no problems occur with the EDM and RID (more storage). But, there can be performance issues with Buffer manager and Buffer pools.

With all these enhancements, you need to be able to see what is going on behind the scenes within the DB2 subsystem, by using a DB2 performance monitor.

When thinking about database performance, it’s useful to think what type of processing will be done



The ultimate holistic approach

1. Define performance objectives & metrics
2. Create monitoring, testing and tuning procedures
3. Implementation practice – again from the top!

Figure 4: The ultimate holistic approach

against my database? Will it be direct or sequential? If the processing will be direct, then HISAM, SHISAM, (P) HIDAM, or (P)HDAM are good choices. If the access is going to be sequential, then HISAM, SHISAM, or (P) HIDAM are appropriate. If the applications that run against this database are doing both sequential and direct processing, then (P)HIDAM will perform well.

If there are a large number of updates performed; or the database records vary in length; or you need to take advantage of logical relationships; or there is a good chance that a secondary index is necessary; then it's probably best to use (P)HIDAM or (P) HDAM. For existing HIDAM and HDAM databases, the recommendation is to convert to PHIDAM or PHDAM, especially when greater capacity is needed.

Internally, IMS consists of two logging functions: the logical logger and the physical logger.

Naturally, both these log functions have their own TCBs. The responsibility of the logical logger is to ensure that all log requests are moved to the log buffers. The responsibility of the physical logger is to ensure that the log buffers are written to disk.

In order to log data, IMS makes use of two types of logs: the write ahead data set (WADS) and the online log data set (OLDS). The WADS is used to externalize log records that IMS considers critical. The WADS is used in a wraparound fashion, whilst the OLDS is used in a round-robin fashion. IMS ensures that data written to the WADS is transferred to the OLDS.

The write to the OLDS is done either when a track worth of buffers is filled or a timer pops every second, whichever occurs first. As the OLDS fills up, IMS schedules an archive utility to archive the OLDS to the system log data set (SLDS) and the recovery log data set (RLDS). Once archived, the OLDS can be reused.

The logging of updates is very critical for data integrity. This is necessary in case of abend, system failure, and/or disaster recovery. If a record is updated, it is often necessary to take both a before image and an after image (if task and update is successful) of the record of information. Therefore, it is feasible that every record updated requires two separate writings of information to the DB2 log. You can instantly see why tuning and performance analysis of the DB2 logging functions are a major consideration; especially in a high volume production environment with mission-critical business applications.

The ultimate holistic approach to taming the performance beast is illustrated in Figure 4. It involves:

- Defining performance objectives and metrics
- Creating monitoring, testing, and tuning procedures
- Implementation practice – again from the top!

A copy of this presentation is available for download from the Virtual IMS user group Web site at www.fundi.com/virtualims/presentations/IMSperformanceOct12.pdf.

You can see and hear the brief user group meeting by downloading the WMV file from www.fundi.com/virtualims/presentations/2012-08-07meeting.wmv.

Meeting dates

- 4 December 2012, Bill Chapin, Lead Mainframe Service Management Software Quality Assurance Engineer, BMC Software will be talking about: “Nearing 24x7 availability with structure changes too!”
- 5 February 2013.

IMS news

IBM has announced the IMS 13 Quality Partnership Program. IMS 13's Database Manager enhancements include: IMS full-function, high-availability database and fast-path data entry database alter support enhances availability; database versioning offers greater flexibility and speed in enabling new programs and databases; additional support for Java and SQL and tools ease IMS development and access to IMS data; and enhanced log, storage, and infrastructure support reduce total cost of ownership, improve performance, and enable growth.

IMS 13's Transaction Manager enhancements include: IMS Connect, the TCP/IP gateway to IMS transactions, operations, and data, offers improved IMS flexibility, availability, resilience, and security; the Intersystem Communication (ISC) connection TCP/IP link can help reduce maintenance costs and complexities; synchronous program switch and enhanced callout support ease IMS application development and connectivity to assist developers with business transformation; enhanced commands, security, repository, and open transaction manager access (OTMA) support simplify operations, improve availability, and can help reduce total cost of ownership.

General availability of IMS V13 has still to be announced. More information can be found at `http://www-01.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/3/897/ENUS212-373/index.html&lang=en&request_locale=en`.

GT Software has released Ivory Hub, a product suite that allows rapid application delivery for mobile, Web, and cloud. The Ivory Hub product suite enables users to tackle various business challenges from direct access to data and applications to transferring data between mainframe and non-mainframe systems. Ivory Hub Enterprise Access enables SQL access to distributed data on mainframe and non-mainframe (LAN-based) platforms from single SQL queries Supported access environments including: IMS and DL/I, CICS, DB2, VSAM, Adabas, Datacom/R or Datacom SQL, and sequential disk. More information can be found at `http://www.gtsoftware.com/news/gt-software-extends-enterprise-modernization-capabilities-ivory-hub`.

Recent IMS articles

Introducing IMS 13 Transaction and Database Servers by Barbara Klein, Information Management System z Strategy in IBM's *IMS newsletter.* You can find the article at `http://www-148.ibm.com/bin/newsletter/tool/landingPage.cgi?preview=1&lpId=5611`.

About the Virtual IMS user group

The Virtual IMS user group was established as a way for individuals using IBM's IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at *www.fundi.com/virtualims* provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join the Virtual IMS user group and share in the knowledge exchange.

To share ideas, and for further information, contact *trevor@itech-ed.com*.

The Virtual IMS user group is free to its members.