# The usage of MaxDB at SAP Hosting

## Table of Contents

# Hosting MaxDB database servers in the Enterprise

MaxDB has a unique, top position in the list of open source database. The SAP-certified database management system combines ease of use, high availability and exceptional performance with low costs. MaxDB underlies a permanent, ongoing development to hold up with todays enterprise requirements. We have visited SAP Hosting to puzzle out if MaxDB can face todays requirements in the enterprise hosting area.

## Building blocks to lower TCO

SAP Hosting is hosting many different SAP solutions on behalf of their customers. They have aquited great experience with SAP certified databases with MaxDB being one of them. Based on an anlaysis of several hundred servers and more than 1.400 customer installations they have found cost benefits of MaxDB compared to other system. For example, the number of database administrator resources needed to maintain MaxDB servers is considerably lower than number of administrators needed for other systems.

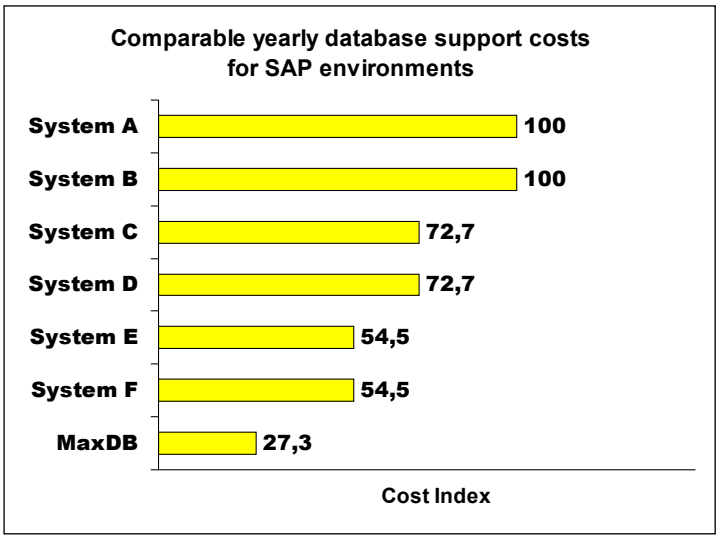
| DB Size / Instance | MaxDB | System A | System B | System C |
|---|---|---|---|---|
| **0 – 30 GB** | 0,1 | 0,2 | 0,2 | 0,2 |
| **30 – 100 GB** | 0,1 | 0,2 | 0,5 | 0,5 |
| **100 – 500 GB** | 0,2 | 0,4 | 0,5 | 0,5 |
| **500 GB – 1 TB** | 0,2 | 0,5 | 1,0 | 1,0 |
| **> 1 TB** | 0,3 | 1,0 | 1,5 | 1,5 |

*Required database administrator resources planned for MaxDB in comparison to other systems*

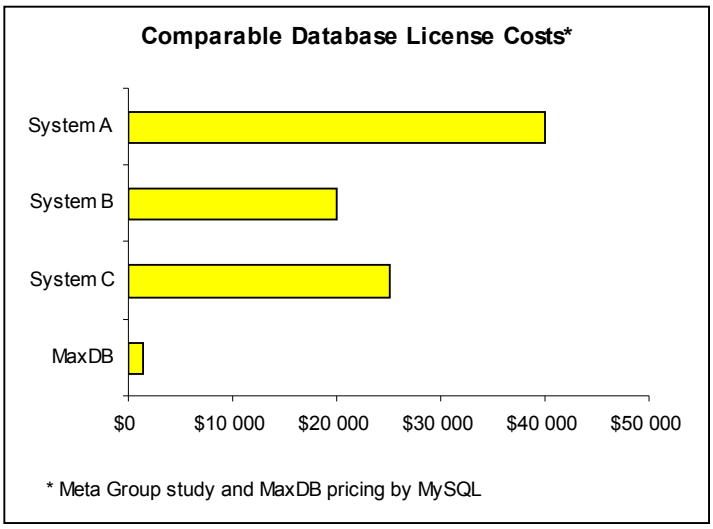
Sophisticated technical solutions are implemented in MaxDB to achive minimum resource requirement. MaxDB offers interesting solutions to ease typical hosting tasks: installation, backup and recovery, reorganisation, performance monitoring and high availability.

IDC[1] worked out that the TCO of a database can be described by analyzing five different areas of costs: downtime costs, training and education costs, staffing costs, software costs (e.g. licence and support fees) and hardware costs. How does MaxDB try to lower each of these individual cost factors? A visit to SAP Hosting gives insights into the daily work with the MaxDB database management system.

1 IDC, Maximizing the Business Value of Enterprise Database Applications on a Unix Platform. 2002




*TCO – low support costs of MaxDB*



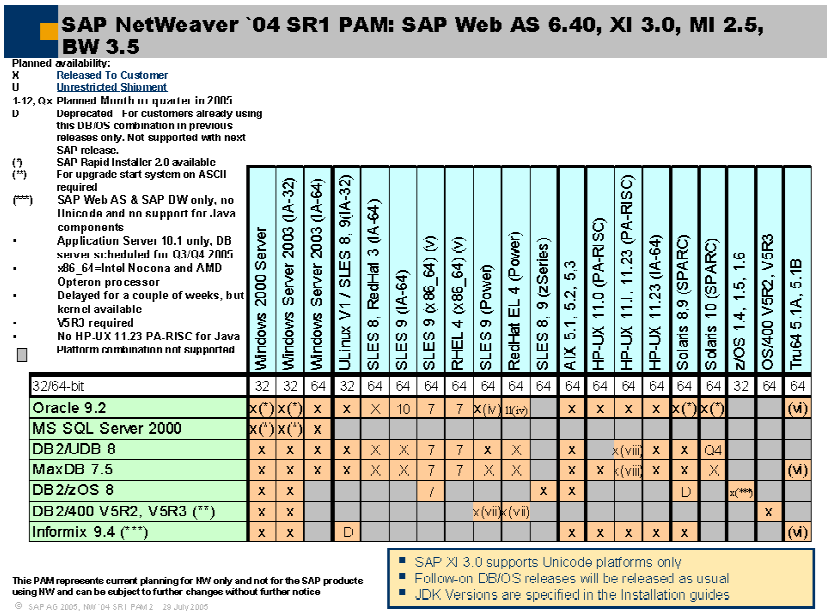

*TCO – low licence costs of MaxDB*

## Supported Plattforms and Flexible Installation

MaxDB supports a variety of operating systems and hardware platforms. That means MaxDB can be run on many different platforms and the user can choose among a wide varity of possible systems. The user can chose the combination of operating system and hardware platform which is best for him based on his criterias. For example the user can choose to optimize for costs and run the low cost Linux operating system.

32- and 64-bit hardware platforms are supported by MaxDB. 64-bit hardware becomes more and more a standard in the database business, because only this recent hardware generation can handle large amounts of main memory efficiently. Fast main memory is a crucial factor of a database. Main memory gets used to buffer frequently used data records. Programs can allocate only 3-4 GB main memory on most 32-bit systems. This is not enough to run todays enterprise databases. However, 64-bit systems lift this limitation. The amount of fast main memory that can be allocated by MaxDB on a 64-bit system is only limited by the total amount of main memory installed. MaxDB supports many 64-bit systems to take

advantage of this fact. The following 64-bit platforms are supported: x86_64 (Opteron, Athlon 64), x86_ia64/IA 64 (Itanium), Power[PC], PA-RISC, SPARC und True64.



**SAP NetWeaver `04 SR1 PAM: SAP Web AS 6.40, XI 3.0, MI 2.5, BW 3.5**

Planned availability:
X Released To Customer
U Unrestricted Shipment
1-12, Qx Planned Month or quarter in 2005
D Deprecated For customers already using this DB/OS combination in previous releases only. Not supported with next SAP release.
(*) SAP Rapid Installer 2.0 available
(**) For upgrade start system on ASCII required
(***) SAP Web AS & SAP DW only, no Unicode and no support for Java components
- Application Server 10.1 only, DB server scheduled for Q3/Q4 2005
- x86_64=Intel Nocona and AMD Opteron processor
- Delayed for a couple of weeks, but kernel available
- V5R3 required
- No HP-UX 11.23 PA-RISC for Java Platform combination not supported

| 32/64-bit | Windows 2000 Server | Windows Server 2003 (IA-32) | Windows Server 2003 (IA-64) | UnLinux V1 / SLES 8, 9(IA-32) | SLES 8, RedHat 3 (IA-64) | SLES 9 (IA-64) | SLES 9 (x86_64) (v) | RHEL 4 (x86_64) (v) | SLES 9 (Power) | RedHat EL 4 (Power) | SLES 8, 9 (zSeries) | AIX 5.1, 5.2, 5.3 | HP-UX 11.0 (PA-RISC) | HP-UX 11.i, 11.23 (PA-RISC) | HP-UX 11.23 (IA-64) | Solaris 8,9 (SPARC) | Solaris 10 (SPARC) | z/OS 1.4, 1.5, 1.6 | OS/400 V5R2, V5R3 | True64 5.1A, 5.1B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 32 | 64 | 32 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 64 | 32 | 64 | 64 | |
| Oracle 9.2 | x(*) | x(*) | x | x | X | 10 | 7 | 7 | x(v) | 11(iv) | | x | x | x | x | x(*) | x(*) | | | (vi) |
| MS SQL Server 2000 | x(*) | x(*) | x | | | | | | | | | | | | | | | | | |
| DB2/UDB 8 | x | x | x | x | X | X | 7 | 7 | x | X | | x | | x(viii) | x | x | Q4 | | | |
| MaxDB 7.5 | x | x | x | x | X | X | 7 | 7 | X | X | | x | x | x(viii) | x | x | X | | | (vi) |
| DB2/zOS 8 | x | x | | | | | / | | | | x | x | | | | D | | x(***) | | |
| DB2/400 V5R2, V5R3 (**) | x | x | | | | | | x(vii) | x(vii) | | | | | | | | | | x | |
| Informix 9.4 (***) | x | x | | D | | | | | | | | x | x | x | x | x | | | | (vi) |

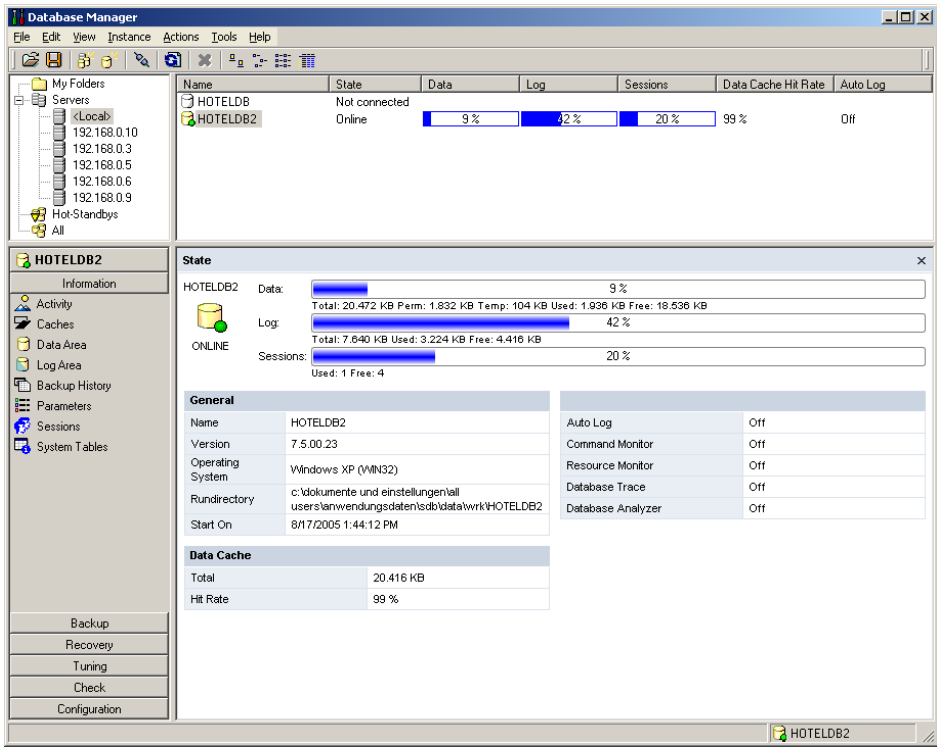
This PAM represents current planning for NW only and not for the SAP products using NW and can be subject to further changes without further notice

© SAP AG 2005, NW '04 SR1 PAM 2 29 July 2005

- SAP XI 3.0 supports Unicode platforms only
- Follow-on DB/OS releases will be released as usual
- JDK Versions are specified in the Installation guides

*Platform availability at the example of SAP NetWeaver*

The installation of MaxDB can be performed by a command line based tool and by a graphical tool. The GUI based Installation Manager is delivered for all supported plattforms that offer a graphical user interface. The Installation Manager is easy to use. It looks like a wizard and guides you through the installation process. Beginners will profit from this user interface and the possibility to create an example database instance. Advanced users won't be disappointed, because the wizard still allows accessing advanced settings.




*All administrative tasks can be performed through the Database Manager GUI*

The time required to install MaxDB is less than 15 minutes. 15 minutes is a limit that MySQL AB has also defined for their second database management system, the MySQL database server. Even for the first time user it takes less than 15 minutes to install the software, to do a basic setup and configuration, to create a database instance and to execute the first SQL statement on the newly installed enterprise server.

### Hands on: the practical examples

Throughout this article you will find boxes with additional information titled with „Hands on“. The boxes try to give practical examples for certain key features and functions of MaxDB mentioned in the main article. The goal of the hands on boxes is to give you a first, practical insight into some of MaxDB's outstanding featueres.

You don't need to read the hands on examples to understand the main article. You can leave them out or just read some of them which a are of most interest to you. However, note that in some cases the practical examples refer to each other. Therefore, it is recommded that you read all of them in the order in which they appear in the main article if you're a first time MaxDB user.

All examples given are based on MaxDB 7.6.00.12 and the example instance created in the course of „Hands on: the 15 minutes installation“. At the time of writing 7.6.00.12 is the most recent version that is recommended for use in production environments outside of SAP application. In order to run the given examples on other versions you might need to modify them. This may especially become true for older versions. We won't give any hints about version differences in this article.

However, if you have questions and comments feel free to discuss them on the MaxDB community mailinglist on http://lists.mysql.com/maxdb, the MySQL MaxDB webforum on http://forums.mysql.com/ or contact us using the Contact Form on http://www.mysql.com/company/contact/

## Hands on: the 15 minutes installation

MaxDB can be downloaded from http://dev.mysql.com/downloads/maxdb/ . At the time of writing 7.6.00.12 is the current release. All examples given in this article are base on this version. In order to rework the examples you should install MaxDB 7.6.00.12.

Download the version of „Complete Software Package“/“MaxDB Installer (tgz/zip packages)“ that you need for your operating system. Check if the GUI-tools are included, if not download the Database Manager and the SQL Studio seperately.

Extract the software archive on your server. All steps from now on require administrator (root) permissions. Start the graphical Installation Manager by calling SDBSETUP.

```
# su root
# tar -xvzf maxdb-all-linux-32bit-i386-7_6_00_12.tgz
# cd maxdb-all-linux-32bit-i386-7_6_00_12
# ./SDBSETUP
```

Select „Start installation/upgrade“ on the first page of the Installation Manager GUI. During the next step you can choose which software components get installed. Install all components by selecting the option „Server + Client“. Please make the program not only install the software but also an example database instance to work with by selecting the predefined installation procedure „Install software and create database instance“. For your first steps with MaxDB it's best to choose a standard layout of data and logfiles, therefore select „Desktop PC / Laptop“ to load the settings of a predefined standard configuration.

Confirm your current selections in the overview screen presented afterwards. In the next step you need to make textual input for the first time in the installation process. You are requested to specify a password for the (first) database administrator. Enter a password and make sure that you remember it, but do not write it down on a note and put that note on your computer screen ;-). Do not change any of the other settings on that screen. Confirm the default installation locations for software, data and logfiles in the next step. Please record all path settings, especially the path of the „Database Software“. We will refer to these paths later. Make sure that you have enough free space for the database software (about 230MB), the data files (256MB) and the log files (128MB) on the locations you've configured right now. Before the installation procedure starts you will get a summary of all options on the next screen. Hit „Install“ to start the actual installation.

After a few minutes, hopefully, you will be informed about the success of the installation with the words: „Your MaxDB installation was finished successfully!“. Once you read this, the installation of the database software and the creation of an example database instance called „MAXDB1“ has been finished. The instance has been started. You don't need administrator (root) permissions any more. Switch back to an unprivileged operating system user.

You can now execute your first SQL statement. Use the command line tool sqlcli for this first test. The tool is located in [Database Software]/bin. You did follow our advice to record the path settings you've choosed, didn't you?

```
# /opt/sdb/programs/bin/sqlcli -u DBADMIN,[password] -d MAXDB1
sqlcli=> SELECT CONCAT('Hi, this is MaxDB ', CONCAT(KERNEL, ' talking to you!'))
FROM DOMAIN.VERSIONS
```

Note that after a restart of the database server you have to restart the database instance manually. MaxDB did not install automatic startup scipts for you. In order to start the database instance make sure that the X-Server communication componet is running. Start the X-Server by calling [Database Software]/bin/x_server start. After that, you can use the Database Manager GUI to administer the database instance and start it.

The experts among you will know that a graphical installation tool is helpful and handy, but there's no question that also a command line tool can be very powerful. For example, the command line based MaxDB installation tool can perform an unattended installation. Using SDBINST you can specify all configuration settings using parameters. You don't need to make any inputs during the installation process because all the information required to perform the installation have been specified already. This way you can integrate the unattended installation into your own scripts.

```
# ./SDBINST --help

usage: SDBINST [-h | --help] [-l | --list] [-v | --version] [-b | --batch] [-package
<package 1,package 2, ... >] [-archive_dir <directory>] [-profile <installation
profile>] [-o <owner>] [-g <group>] [-indep_data <Independent Data Path>]
[-indep_prog <Independent Program Path>] [-pcr_7600 <precompiler runtime 7600 path>]
[-odbc_path <instpath>] [-cpc <precompiler path>] [-jdbc_path <java driver path>]
[-loader_path <instpath>] [-rdpython_path <instpath>] [-msgpath <instpath>]
[-syncmanpath <instpath>] [-dbana_path <instpath>] [-webpath <instpath>] [-pcr_7300
<precompiler runtime 7300 path>] [-pcr_7301 <precompiler runtime 7301 path>]
[-pcr_7403 <precompiler runtime 7403 path>] [-pcr_7500 <precompiler runtime 7500
path>] [-dbc_path <instpath>] [-depend <Dependent Program Path>]

[...]
```

*Help output shows options for the unattended installation*

In the hosting area scripts get used to automate daily tasks. Scripts make it easier and faster to handle certain tasks and they ensure the use of reproducible, verified and quality assured standard procedures. MaxDB supports the usage of scripts. This is an efficient way of working. The unattended installation is only one example out of many how MaxDB supports scripts. All maintenance tasks can be perfomed through the Database Manager GUI and on the command line.

Many systems deserve a lengthy, time consuming adjustment of database parameters. Only long-term experts know how to set the database parameters to get the best performance out of their system. MaxDB is different. MaxDB does not bother you with a long, unsorted and confusing list of database parameters. The database parameters are devided into three groups: general, advanced and support parameters. For a basic installation the database administrator needs to know only about 20 parameters from the group of the general parameters. You can safely use the preconfigured database parameter settings for a wide range of applications.

## File space management

At its heart hosting is about maintaining uninterrupted, fast operations and preparing for outages of all kinds. For maintaining uninterrupted and fast operations it's necessary to take care of efficient file organization and to improve the performance by removing bottlenecks if required. Classical backup and recovery technology and hot standby solutions are used to prepare for outages.

Automatic file management is a strength of MaxDB. MaxDB does not need any classical reorganisations. You do not need to schedule any special reorganisation operations to maintain optimal access patterns and minimal space requirements. The experts at SAP Hosting welcome this very much. There is no need to scedule any kind of outages to maintain file structures, no administrator resources are taken by such a task and no additional experts knowledge is required to run the database efficiently. This is a major difference between MaxDB and it's competitiors. All reorganisations are performed while the database is running. Update in place, sort by insertion and delete in place ensure that all MaxDB file structures are in optimal conditions even after years of uninterrupted work.

## Hands on: No need for reorganisation

Actually, this box should be titled with „hands off“. MaxDB does not offer any commands for reorganisation of data files (data volumes). The administrator does not get bohered with such tasks. The reason that there is no command for reorgnisation in MaxDB is that MaxDB does not need to be manually reorganised even after a vast amount of modifications and deletions. MaxDB does all reorganisations while the database runs. This way the storage is always in optimal conditions and MaxDB works with maximum performance all the time.

Internally MaxDB stores data in in B*-trees. B*-trees are a variantion of B-trees. One of their properties compared to other B-tree variations is that they tend to have a better fill level than other variantions of B-trees, e.g. B+-trees. All data of a table is saved using B*-trees. The tree is sorted in primary key order. Secondard keys are also organized in B*-trees. The principle of clustered indices is used for secondary trees, that means entries in a secondary tree do not point to data pages directly but contain the primary key value of the referenced records.

Three operations, which are applied to all B*-trees, ensure that no reorganisations are required. The three operations are called: update in place, sort by insertion and delete in place. Let us check for the basic ideas of all three operations.

To explain the operations one needs to know how MaxDB stores records on a data page. Each data page has a size of 8kb by default. A page is devided into three areas. Every data page starts with a list of unsorted data entries, followed by free space. At the end of the data page a sorted list of data entry pointers gets stored. The list of data entry pointers, the position list, is sorted in key order. That means the database can perform a fast sequential scan when it fetches records in ascending or descending key order.

Whenever the fill level of a page falls below a certain threshold value or if a page is full, MaxDB will perform all required B*-tree balancing operations implicitly. We will not mention this any more when we describe the properties of the the three operations: update in place, sort by insertion and delete in place.

When a data entry gets inserted into a page it gets appended to the end of the list of data entries on the page. The pointer to the newly added data entry gets inserted into the pointer list in a way that the pointer list remains sorted. Sorting is performed on the smaller of the two possible sort lists. MaxDB will not sort the data entries itself but it will sort the much smaller pointer list. Of course, sorting the smaller pointer list is faster than sorting the data entries. This is what MaxDB means by sort by insertion.

However, if a page overflows or underflows during an insert and a B*-tree balancing operation has to be performed, MaxDB will sort the data entries. Why? Having sorted data entries is of advantage when merging pages.

Whenever possible MaxDB tries to handle updates of data entries on their current page. This is called update in place. If a data entry gets updated and its overall length and its key value do not change, then MaxDB does not need to do anything but update the data entry itself. No changes to the position list are required because the position list is sorted by the key values and the key values have not changed. A simple update of the data entry in it's current place is possible, it still fits into the existing slot.

More work needs to be done if the lenght of a data entry changes but the key value remains the same. If the size changes, then the slot that is occupied by the data entry is either too small and needs to be increased or it is too large and should be shrunk in order to save space and avoid gaps. Resizing a slot means shifting forth or back all data entries that follow the resized entry. And, whenever the position of a data entry changes, MaxDB has to update the pointers in the position list.

If a data entry remains the same size but the key value gets changed, then there is little chance to handle this operation on page level. Each page, each leaf node in the tree, stores the records of a certain key value range. If the modified key value no longer matches the key value range of the original page, then the data entry has to be moved to another page which holds the matching key value range.

It is unlikely that the new key value can be kept on the original page. Therefore MaxDB does not try to handle an update with a key value change on a page, but transforms the update operation into a delete followed by an insert.

Delete in place means that MaxDB will not juggle data entries on a page if an entry gets deleted. MaxDB will only update the position list, that is all. Really? No, not always. MaxDB will of course handle the case that the fill level of a page falls below a threshold value and remove the page itself which might cause B*-tree balancing operations.

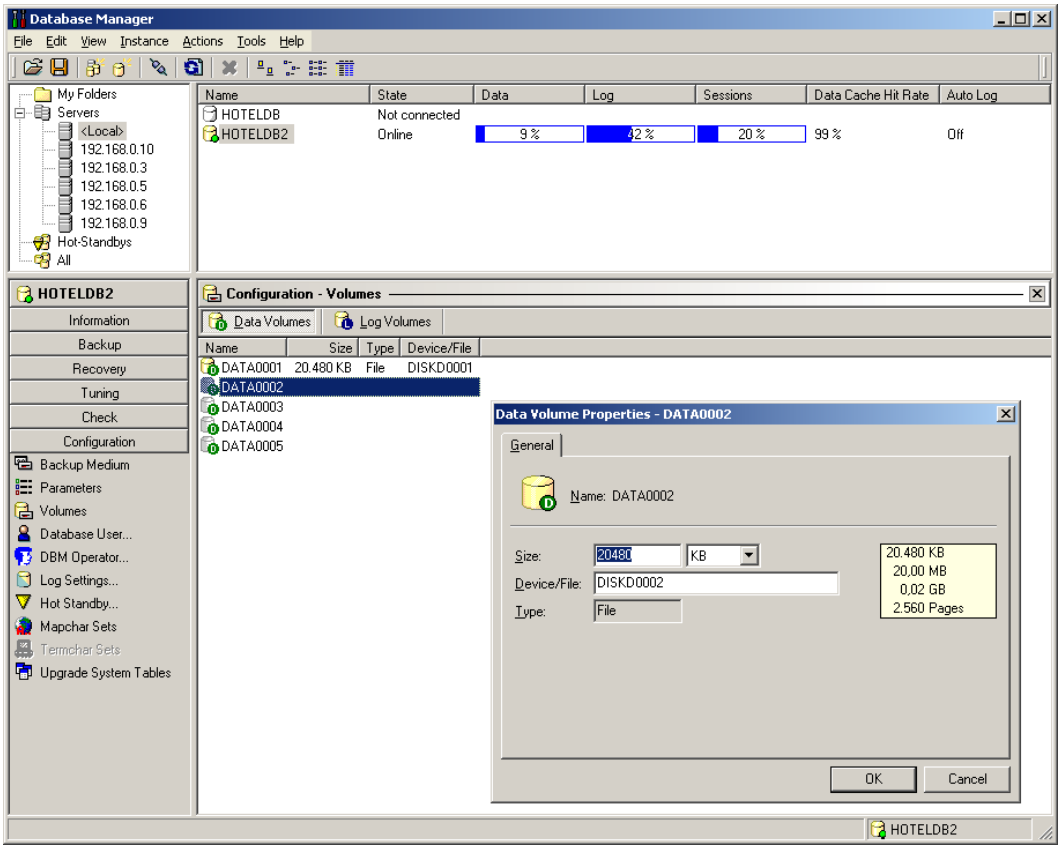
If you are worried that permanent balancing has a hugh impact on the transaction performance, then you might want to try the dynamic tables. Whenever you create a table with the attribute DYNAMIC, MaxDB will perform less reorganisations/balancing but in turn consuming a bit more space.

```
CREATE TABLE tdynamic(col1 INTEGER) DYNAMIC
```

Dynamic causes a waste of space but it might be interesting for tables with random access patterns and high fluctuation of their size.

No reorganisation is one key benefit of MaxDB from a database administrators perspective. Eventing and flexible file space management including online expansion and reduction of the database file space is another benefit. As explained earlier, all these tasks can be performed in two ways. You can either use a self-explaning GUI interface or command line based tools which can be called from inside you management scripts.

Whenever MaxDB hits the capacity limits of a data or log file, it can continue to operate. There is no need to shutdown MaxDB in such a case. You can add new data files (MaxDB speak: data volumes and log volumes) without interrupting the operation of the database and sceduling a downtime for maintenance. If you use the eventing features of MaxDB, you can ask MaxDB to send you an email or a SMS whenever you the capacity of a data file hits a userdefined threshold value. The eventing features are based on MaxDB internal functionality. No external monitoring tools are required. Of course, this does not mean that MaxDB can't be integrated with such tools if required.




*Wizard showing the online expansion of the database space*

## Monitoring the Performance of MaxDB

Like most databases MaxDB uses an optimizer to further improve the performance of the system. The task of an optimizer is to find the best execution plan for a certain SQL query. Depending on the situation the best plan can be the fastest plan or the plan with the lowest I/O overhead.

MaxDB is using a cost based optimizer. Cost based optimizers use information about actual data distribution to find the best execution plan. Prior to the development of cost based optimizers, rule-based optimizers have been used. Rule based optimizers do not take data distribution information into account and therefore they cannot make such detailed and balanced decisions as a cost based optimizer.

Cost based optimizers can collect data statistics during the execution of SQL queries or they can collect them during extra operations. Collecting information about data statistics during runtime can require more time than the time benefit gained from finding and executing the fastest execution plan. If previously recorded data statistics can be used then, the time loss is not that big. However, the problem with previously recorded data statistics is that they can get outdated due to changes in the underlaying tables. Also, for newly created tables, statistics might not be available at all. Outdated statistics lead to wrong decisions, that means they can make the optimizer choose another but not the fastest and most efficient execution plan.

## Hands on: Using Eventing for pro-active actions, for example: updating statistics

As of version 7.5.00.08 MaxDB offers database events and monitoring facilities. Starting with 7.6 this feature is stable and has been documented. Currently MaxDB offers about 20 events that can be monitored using an Event Dispatcher. Among the most interesting events are: DBFILLINGABOVELIMIT, LOGABOVELIMIT, ERROR, SYSTEMERROR, OUTOFSESSIONS and UPDSTATWANTED. Whenever an event gets triggered, the Event Dispatcher invokes a freely configurable handler program.

UPDSTATWANTED gets triggered whenever the database kernel recognizes that statistics should be updated. The update of statistics has been considered as such a common task that a „shortcut“ DBM command has been introduced. The DBM command auto_update_statistics is using the Event Dispatcher internally to update statistics automatically when needed. However, not in all cases it might be necessary to rebuild statistics immediately when statistics become outdated or are missing at all. Collecting statistics is a task that puts additional load and locks on the database. Therefore an administrator or a more sophisticated, and „intelligent“ software might want to check if the update task can be stalled for a while in order to be executed in times of low user load.

When you configure the Event Dispatcher manually, you'll start a new process that recieves the event notifications from the database kernel. That means that the Event Dispatcher operates independently from the database kernel. An Event Dispatcher that has been started on the database server machine locally will continue to run no matter if you shutdown the observed database instance or the X-Server communication component. But this also means, that you have to start the Event Dispatcher manually after rebooting the database server machine.

The Event Dispatcher dbmevtdisp programm resides in [Database Software]/bin. You can start as many Event Dispatchers as you have configured in the database parameter _MAXEVENTTASKS. The default is two: one for you and one that's implicitly started when you use the DBM command auto_update_statistics or auto_extend. Each Event Dispatcher has its own config and log file. We recommend to put the config and log file in the [Rundirectory] of your database instance.

```
# /opt/sdb/programs/bin/dbmcli -u DBADMIN,[password] -d MAXDB1 param_directget
RUNDIRECTORY
OK
RUNDIRECTORY      /var/opt/sdb/data/wrk/MAXDB1
```

But before you can start to configure the Event Dispatcher you have to tell MaxDB which events shall be monitored and send to the Event Dispatcher for dispatching. Use the DBM command event_list to get a list of all events that are active. The database will inform the Event Dispatcher only about events that are active. This in turn means, that your handler will only be triggered if the event has been activated with the DBM command event_set. However, in this short introduction we will use only events that are active by default. If you want to know which events are active by default, run:

```
# /opt/sdb/programs/bin/dbmcli -u DBADMIN,[password] -d MAXDB1 event_list
```

Before you can start the Event Dispatcher you have to create a config file. A config file gets created automatically, if needed, whenever you add a new event handler to it. Let us define a simple handler for ONLINE. The reason that we use ONLINE here instead of UPDSTATWANTED is that we can easily trigger it and verify our very first event handler configuration.

```
# /opt/sdb/programs/bin/dbmevtdisp add /var/opt/sdb/data/wrk/MAXDB1/myevt.cfg Name
== ONLINE Command == "/bin/bash -c \"echo '\$EVTDATE$, \$EVTTIME$ Event:
\$EVTNAME$, value1: \$EVTVAL1$, value2: \$EVTVAL2$, text: \$EVTTEXT$ ' >>
/tmp/myevt_events.log\""
```

For the sake of simplicity the handler does nothing but write a message to a logfile. The message contains some variables like $EVTDATE$ that wil be replaced with actual values when the handler gets invoked by the Event Dispatcher. A list of all variables provided by MaxDB is available in the documentation. Start the event dispatcher and write down the Event Dispatcher instance number. You need to specify the instance number when you stop the Event Dispatcher.

```
# /opt/sdb/programs/bin/dbmevtdisp start /var/opt/sdb/data/wrk/MAXDB1/myevt.cfg -l
/var/opt/sdb/data/wrk/MAXDB1/myevt.log -d MAXDB1-u DBADMIN,[password]

Event Dispatcher instance [n] running
[...]
```

Start and stop the database instance several times and check your logfile /tmp/myevt_events.log.

```
# /opt/sdb/programs/bin/dbmcli -u DBADMIN,[password] -d MAXDB1 db_offline
# /opt/sdb/programs/bin/dbmcli -u DBADMIN,[password] -d MAXDB1 db_online
# cat /tmp/myevt_events.log
```

You can stop the Event Dispatcher using dbmevtdisp stop [n]. [n] is this instance number of the Event Dispatcher that is shown when starting the Event Dispatcher.

```
# /opt/sdb/programs/bin/dbmevtdisp stop [n] -d MAXDB1 -u DBADMIN,[password]
```

The MaxDB documentation gives an example of a more useful handler. It shows how to send an email to the database administrator.

```
# /opt/sdb/programs/bin/dbmevtdisp add /var/opt/sdb/data/wrk/MAXDB1/myevt.cfg Name
== UPDSTATWANTED Command == "/bin/bash -c \"mail -s 'MaxDB instance \$DBNAME$ on
server \$SERVERNAME$ has triggered \$EVTNAME$' <
/var/opt/sdb/data/wrk/MAXDB1/evt_updstatwanted.msg \""
```

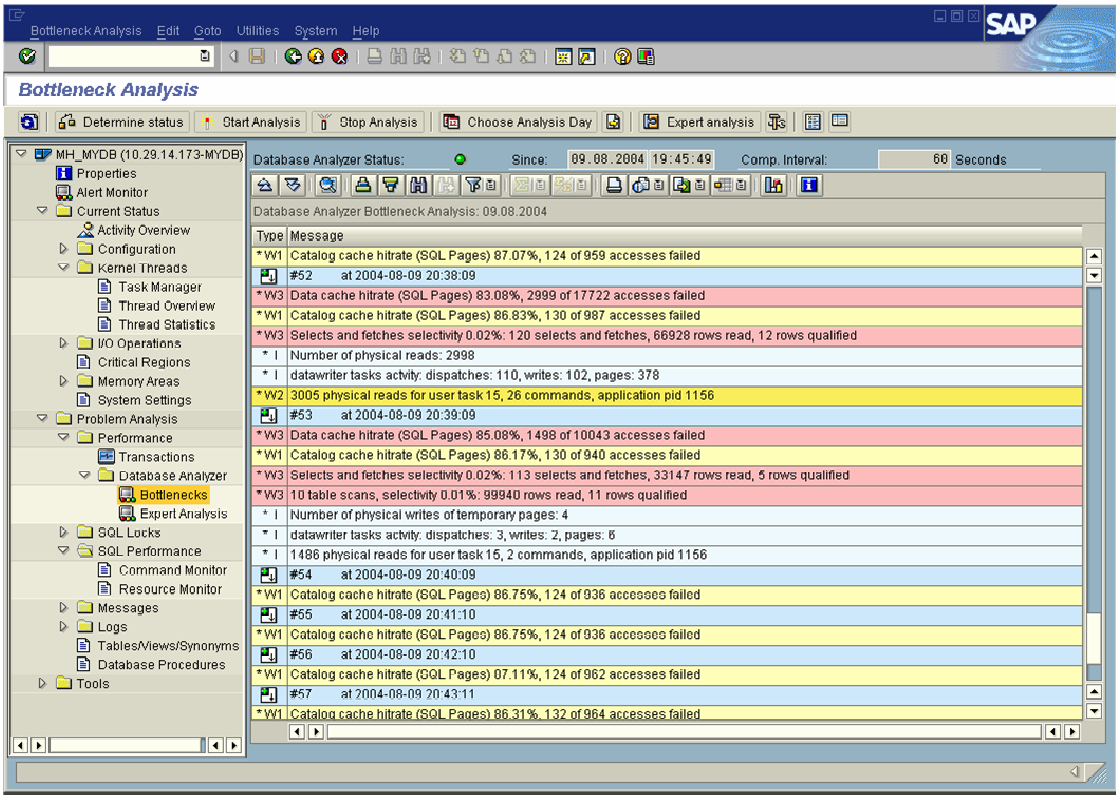
Note that the mail program gets the mail body from the file /var/opt/sdb/data/wrk/MAXDB1/evt_updstatwanted.msg which you need to create manually. Once the administrator has recieved the email he can investigate the system table SYSUPDSTATWANTED to find out which tables and columns have outdated or no statistics at all.

MaxDB tries to combine the best out of both technologies, the best of online evaluations and statistics. The optimizer will use data statistics collected during runtime if only one table is accessed by a SQL statement. If the SQL statement affects more than one table, e.g. during a join operation, then MaxDB will use previously recorded data statistics. MaxDB has a clever solution on the problem of outdated statistics. The database has mechanisms to detect outdated statistics and can send a note to the database administrator requesting the update of the recorded data statistics or even update the statistics automatically. In other words: the database administrator does not need to worry about outdated statistics. MaxDB will tell him if an update is required. This is another step to lower the amount of maintenance work required when hosting and running MaxDB databases. The database administrator does not need to monitor data changes and try to figure out if changes have invalidated the recorded statistics and therefore if they could have a negative impact on the optimizer. MaxDB frees the user from this task and takes care that the system always can work with maximum performance.

Three tools are delivered with MaxDB to fine tune the database in it's environment with the operating system and the database application. The tools are: Diagnose Monitor, Resource Monitor and Database Analyzer. The Diagnose Monitor and the Resource Monitor are used to find and record slow and resource consuming SQL statements. Quite often, few slow and not optimized queries have a severe impact on the performance of the entire system. Those slow SQL statements can be spotted using the Diagnose Monitor and the Resource Monitor.

For long-term performance monitoring of the database and some relevant operating system parameters the Database Analyzer gets used. The tool collects figures about all components of the MaxDB

database system. The figures give insights to the database administrator which database component is a bottleneck already or could develop to a bottleneck. This way the database administrator gets information if, for example, the system is CPU bound, I/O bound or too many locks are requested. MaxDB can assign a relevance to every figure, every event that gets recorded by the Database Analyzer. This helps to get a quick and easy overview. Inside SAP R/3 there is a graphical tool to analyze the Database Analyzer protocol. As you can see in the image it gives you a quick overview of current bottlenecks. It shows which problems have occured and when the problem occured.

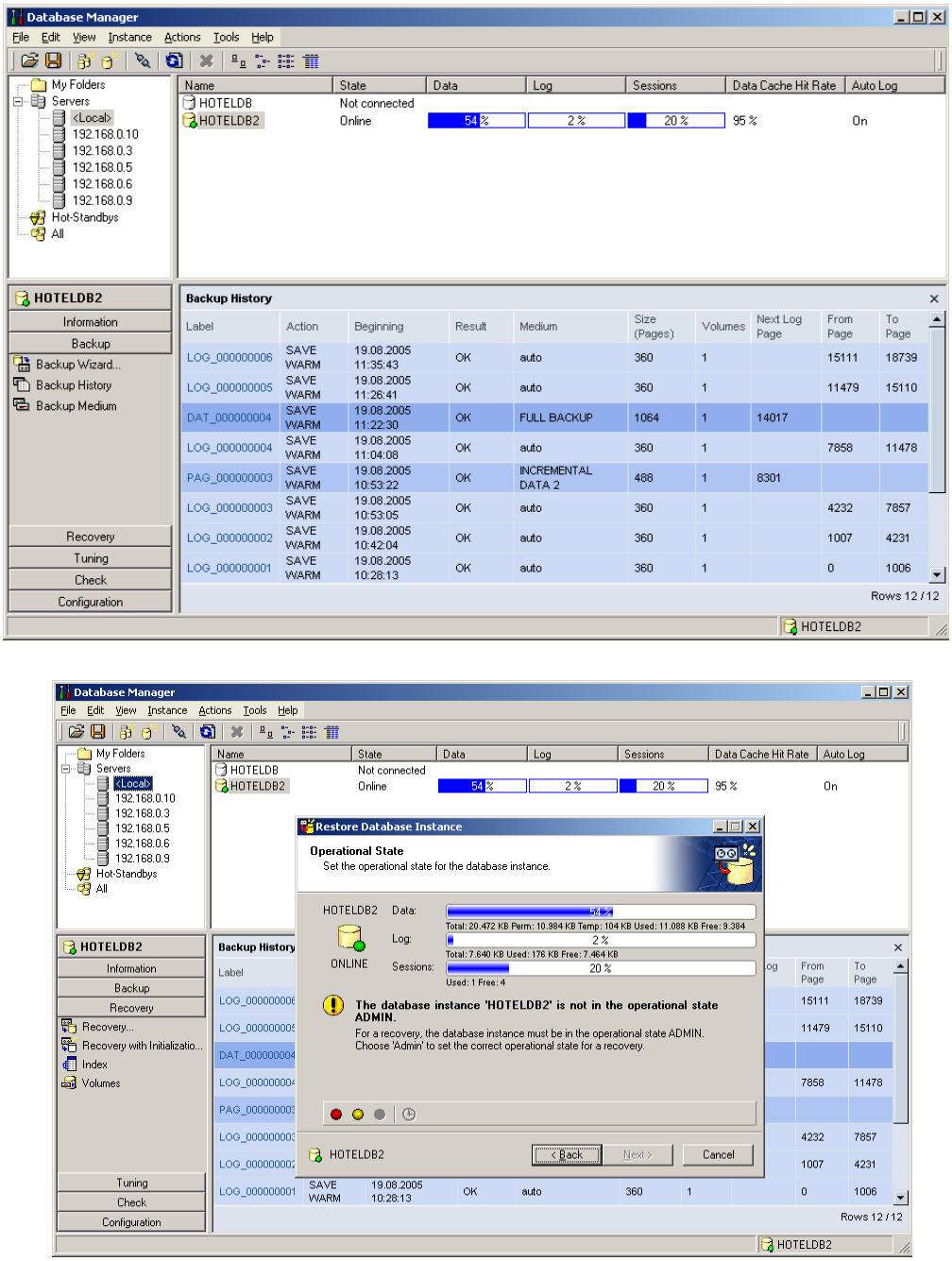



*Long-term bottleneck analysis using the Database Analyzer shown at the example of SAP R/3*

## Backup and Recovery

The technicians at SAP Hosting distinguish between two different hosting scenarios in their landscape: hosting a production system and hosting a development system. Every hosting scenario has it's own demands on backup and recovery. On production systems classical backup and MaxDB's fast snapshot technology get used. Surprisingly development systems have even higher demands on the performance of backup and recovery. That's why file system snapshots get used. File system snapshots are taken using – for example – Logical Volume Manager (LVM) capabilities.

MaxDB has a build-in transaction consistent online backup. It features automatic log backups as well as incremental and full data backups. Logfiles do not require any maintenance work if automatic log backups are activated. If automatic log backups are activated, MaxDB will perform a backup automatically whenever it's required. Once MaxDB has backed up log entries it releases the space that the log entries have occupied inside the log file and makes it possible to overwrite the already backed up log entries. This way it's unlikely that a log file can overflow, because MaxDB releases space inside the log file frequently for reuse.







*Database Manager GUI: Backup History and Backup Wizard*

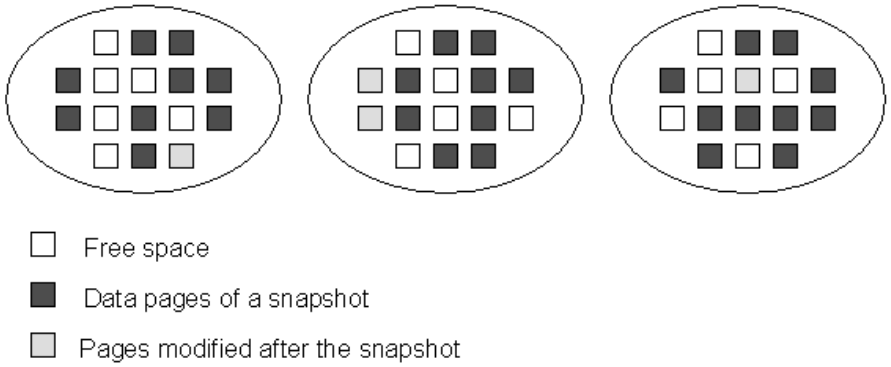
The possibility to combine incremental and full data backups with log backups during recovery leads to a great variety of possible recovery scenarios. This increases the data security. For example, if an incremental data backup is corrupted and cannot be recovered, then you could recover and redo all log backups made during the time covered by the corrupted incremental data backup. Redoing the log backup requires more time than recovering an incremental data backup, but it provides an alternative recovery where one might otherwise not exist. The varity of possible recovery scenarios gives you additional security in case of backup media failures.

Based on log backups you can also do a Point-in-time recovery with MaxDB. All backup and recovery tasks can be performed using self-explaining wizards of the Database Manager GUI. But of course, you can also perform all actions on the command line and integrate the command line calls into your scripts if required.

Parallel processing gets used to speed up backup and recovery. All backup and recovery tasks are performed by extra tasks. This ensures that the regular database operations are not negatively influenced performancewise. Several tasks can write to and read from backup media in parallel.

It's possible to integrate MaxDB's backup and recovery facilities into third-party solutions such as TSM (IBM/Tivoli) and NetWorker (Legato). Many other backup tools can be integrated using the Backint for MaxDB and Backint for Oracle interface.

SAP Hosting does make use of MaxDB's build-in database snapshots in a creative way whenever the application software on a system gets updated. The shadow paging and converter technology of MaxDB allows it to freeze in the status of the database. Database pages that have been frozen in by a snapshot cannot be overwritten or modified, but the database can continue to perform read operations on these pages. Write operations do not modify frozen pages. Instead of modifiying the existing pages MaxDB will redirect write operations to new, unused data pages.




*Principle of MaxDB database snapshots shows using three data files (data volumes)*

Before any application software gets updated at SAP Hosting a database snapshot gets created. If problems occur during the update of the application software, the update process gets canceled and the database administrator restores the snapshot that was created before the update. All changes that have been made to the database during the application software update process are reverted. Restoring a database snapshot is a very fast operation. All data pages that belong to the snapshot are still contained in the data files of the database, they just need to be thawed. It's not necessary to read data pages from an external storage medium and to import then into the database. The time consuming step of copying and importing database pages is skipped. Another application of snapshots is the use in the SAP training environment. At the end of the class all modifications to sample databases can be undone within short time. It's not required to recover the originial state of the sample databases from a recovery.

On development systems another type of snapshots get used: file system snapshots. Do not mix up file system snapshots with MaxDBs database snapshots. File system snapshots include entire harddisks and are taken using LVM for example. One of the main drawbacks of snapshots on the storage level is that they are not incremental in the way the build-in MaxDB backup and recovery features are.

However, on development systems it's common that the entire database software gets updated. Though this is not a risky operation in general a file systems snapshot is taken before the database software gets updated. In case of an error the file systems is put back into place. Again, this is faster than recovering a database backup and the file system snapshot includes the database software itself. We recommend that before you take a file system snapshot you should issue a database snapshot to keep a consistent stage of your data inside the database. If you skip that step, then you might have to rely on the automatic emergency recovery capabilities of MaxDB based on logfiles and checkpoints.

## Hands on: snapshots using dbmcli

MaxDB is able to freeze in the current status of the database and save it as a snapshot. You can keep only one snapshot in the database. A snapshot can be recovered or thawed as often as you want.

Snapshots can only be created, removed or recovered in admin mode. This is not much of a problem. All operations with snapshots are pretty fast. In simple words a MaxDB snapshot is created by copying the contents of the converter into a protected area on the data volumes. The converter is a MaxDB component that maps logical data page addresses into physical ones. The space requirements of the converter are rather low. That's why all operations on snapshots are pretty fast. The size of the converter is about 28MB for a database with a capacity of 50GB.

You can use the Database Manager GUI to work with a snaphot. Use the menu entry Instance - Snapshot for your work. Note that the menu entry can only be selected if the database is in admin mode. The usage of the GUI is self explaining. Therefore we will demonstrate how to administer snapshots using dbmcli. As in our previous hand-on sections we expect that you have created an example instance as described on „Hands on: the 15 minutes installation“. Let us start!

Create a database table and insert one record. We need this to be able to demonstrate the effects of snapshots. Use the following SQL statement. You can execute it using SQL Studio, for example.

```
CREATE TABLE t1 (col1 CHAR(1))
INSERT INTO  t1 (col1) VALUES ('a')
SELECT * FROM t1
```

Start the tool dbmcli on the shell to create a snaphot. As said before, a database snapshot can only be created, removed or recovered in admin mode. Therefore the database has to be switched in admin mode before creating a snapshot and the database has to be switched back to online mode once this is done.

```
# /opt/sdb/programs/bin/dbmcli -u DBADMIN,[password] -d MAXDB1
dbmcli on MAXDB1> db_admin
dbmcli on MAXDB1> db_execute CREATE SNAPSHOT
dbmcli on MAXDB1> db_online
```

The snapshot that has just been created contains all contents of the database. That means it holds also table t1 and the one record we have inserted into it. Add another records to the table and check if t1 shows two records afterwards. The modified data pages of the table t1 will be saved by MaxDB on another place than the data pages that belongs to the records in the snapshot.

```
INSERT INTO  t1 (col1) VALUES ('a')
SELECT * FROM t1
```

Let's go back to the snapshot using db_execute RESTORE SNAPSHOT.

```
dbmcli on MAXDB1> db_admin
dbmcli on MAXDB1> db_execute RESTORE SNAPSHOT
dbmcli on MAXDB1> db_online
```

Check the contents of t1 again. As you can see, there is only one record. That means, that you can see only the contents that have been in the table when you have created the snapshot.

Snapshots can be removed using the following commands.

```
dbmcli on MAXDB1> db_admin
dbmcli on MAXDB1> db_execute DROP SNAPSHOT
dbmcli on MAXDB1> db_online
```

# High Availability

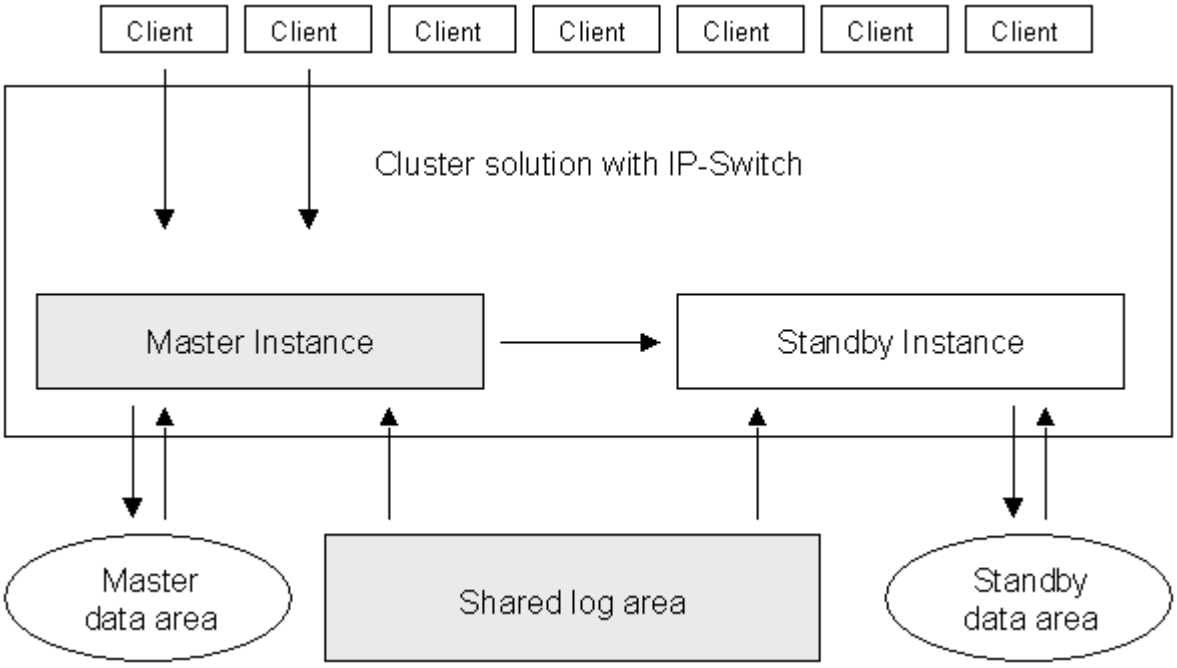
Very high demands on the availability of the database server are fulfilled by a MaxDB Hot-Standby setup. Standby systems shorten the time for outages in case of a total failure of the database server. MaxDB distinguishes between standby and hot-standby setups.

Simple standby systems are updated in regular intervals using log backups from the production system. This approach is also known as log shipping. In such a setup, the contents of the standby system differ from the production system by the amount of time that is between the creation of a log backup on the production system and its recovery on the standby system. The time delay between the standby and the production system can be an advantage. For example unwanted and accidental deletions and modifications that have been caused by a faulty application might fall into the time delay. In such a case one would interrupt the log shipping and redirect all clients to a standby system which has still the data that have been accidently modifed.

MaxDB hot standby systems are based on a shared log. The log is shared between the production system and one standby system. The log is the only shared component. The standby system is using its own data area which is independent from the data area of the production system. Only the production system is allowed to write new entries to the shared log. The standby system has only read access to the log.

When a standby system is set up an initial copy of the data of the production system is done. Once the copy has been done, the standby system starts to monitor the shared log continiously and redoes all new log entries it finds. This way it's ensured that the standby system and the production system have the same data sets. It is possible to configure a time delay for redoing log entries. As explained above this can be of advantage if a logical error occurs and records get accidently deleted or modified.

A cluster solution handles the switch over from a failed master system to a standby system In case of an outage of the master system. IP-Switching will be used to redirect application clients to the standby system. Meanwhile the standby system will redo all remaining log entries and switch from admin to online mode to be able to handle application clients.




*Overview on the MaxDB hot standby setup*

According to internal test, the time required to switch a hot standby into full online mode is less than the time required for the application client fail over. Usually only a few seconds are needed by the MaxDB hot standby system before it can serve application clients. IBM and SAP have created a whitepaper on hot standby, that might be of interest for those who are interested in more details. The whitepaper is available on http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100442 .

## Conclusion

Hosting a database includes the following tasks: installation, update, backup and recovery, maintaining performance and high availability. MaxDB offers practical and cost efficient solutions for all of these tasks. Some technologies and solutions are unique when compared to competitors. The development of MaxDB is driven by the idea of uninterrupted services and lowest maintenance work. This philosophy and the already implemented solutions result in cost advantages for MaxDB.

The MySQL AB Team is glad to answer your questions. Please use http://www.mysql.com/company/contact/ to contact us.

*Ulf Wendel, Support Manager MaxDB*