SAP® MaxDB™

Expert Session 8:
New Features in SAP MaxDB Version 7.8

MaxDB/liveCache Development
Melanie Handreck
Heike Gursch
April 2010

THE BEST-RUN BUSINESSES RUN SAP™ **SAP**

SAP MaxDB version 7.8 is currently released with SAP Netweaver 7.2 (which is currently in Ramp-up phase).

It might be available for older SAP releases later on. Please always check in the *Product Availability Matrix* on the Service Marketplace if a new database version is released for your specific SAP release, **before** installing it.

General Information

This document describes *major* changes between MaxDB 7.8 and older versions.
However, it does not contain *all* changes of MaxDB version 7.8.
For details please refer to the (online) documentation.
Some features have been implemented in lower versions as well.

Even if some of the features described below are not directly perceptible, they might
have a strong impact, e.g. on performance or scalability, and thus are briefly described.

Agenda

1. **Administration**
   - Isolated Software Installation
   - Database Installation Wizard
   - Longer Passwords
   - Backup Enhancements
   - New Events
   - Automatic Index Recreation
2. Performance
3. Monitoring

This session describes new features in the areas of Administration, Performance and Monitoring.

It starts with information about the isolated software installation, followed by the advanced database installation wizard, enhancements regarding system security, backup enhancements, new database events and the automatic recreation of indexes.

Two short demos complete this first part.

Isolated Software Installation

SAP

Software Installations are fully isolated

- Parallel MaxDB installations with the same version
- Parallel MaxDB installations with different versions
- Integrated Software Installation Management

- More than one client installation on the same host
  - E.g. each application server has its own MaxDB client runtime
- Isolated maintenance of MaxDB/liveCache instances
  - Upgrade of database „A" does not influence database „B" and its current connections
- No collection of files belonging to different installations in one directory
- Downward compatibility to legacy versions

In SAP DB / SAP MaxDB versions 7.2 – 7.7 all databases running on one server are sharing the *independent data* and the *independent programs directory*. The *independent data directory* (e.g. /sapdb/data) contains the configuration data and *rundirectories* of SAP MaxDB instances.

The *independent programs directory* (e.g. /sapdb/programs) contains programs and libraries shared by the SAP MaxDB instances and SAP MaxDB applications. These programs are downwards compatible. They are installed with the same version as the highest MaxDB software version installed on that server.

As these programs and libraries are shared by all databases running on that server, all databases are influenced, when this software is updated. For example the tcp/ip listener (X-Server), which handles remote connections to the database, is part of the *independent programs directory*. When the independent software is updated, all remote connections to a server need to be stopped (on Windows and in versions < 7.4.03 on UNIX).

To solve such problems, the database software is fully isolated as of version 7.8. The upgrade of one database instance does not influence any other database on the same server anymore. Furthermore it is now possible to install more than one client software on one server (in older versions the client software was part of the independent programs directory).

This new software installation is downward compatible to older MaxDB versions.

Glossary of Installation Terms

Old Terms

- Independent Program Path
  - Once per machine
  - Components shared by each MaxDB installation
    - Client programs, e.g. dbmcli, dbmrfc
    - Client runtimes, e.g. SQLDBC, Precompiler Runtimes, JDBC, ODBC
    - Network communication server (x_server)
- Independent Data Path
  - Once per machine
  - Installation registry
  - Data files and log files for each MaxDB installation – e.g. rundirectories
- Dependent Path
  - More than once per machine allowed
  - Database server software

New Terms

- Global Program Path
  - Once per machine
  - Components shared by each MaxDB installation
    - Installer tools, e.g. sdbuninst, sdbverify, sdbconfig
    - Global listener (sdbgloballistener)
- Global Data Path
  - Once per machine
  - Installation registries
  - No private data path: Data files and log files for each MaxDB installation – e.g. rundirectories
- Installation Path
  - More than once per machine allowed
  - Database server software
    - Listener (x_server), Database Kernel
  - Database client software
    - dbmcli, dbmrfc, SQLDBC, …

The *global program path* still contains components shared by all MaxDB installations. But these are only installer tools and the global listener, now.

Each installation now has its own X-Server, which is part of the *installation path*. The global listener only routes the remote connections to the X-Server of the specific database.

For each database or client installation, an own *installation path* needs to be specified. This directory contains all relevant software.

The *global data path* contains the installation registries and log files, data files needed by the global listener and - if the *private data path* option is not used - the rundirectories of the database instances.

DBMCLI commands to get the relevant directories:

dbmcli inst_enum <installation path>
Shows e.g. the data path and the port number used by the X-Server of this installation.
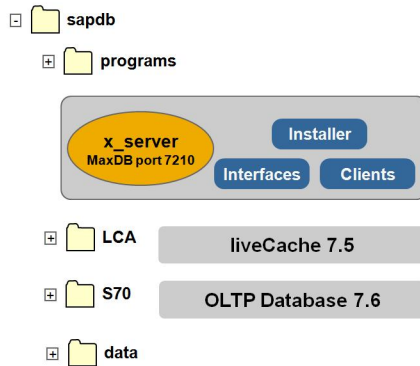
dbmcli –d <database name> dbm_getpath
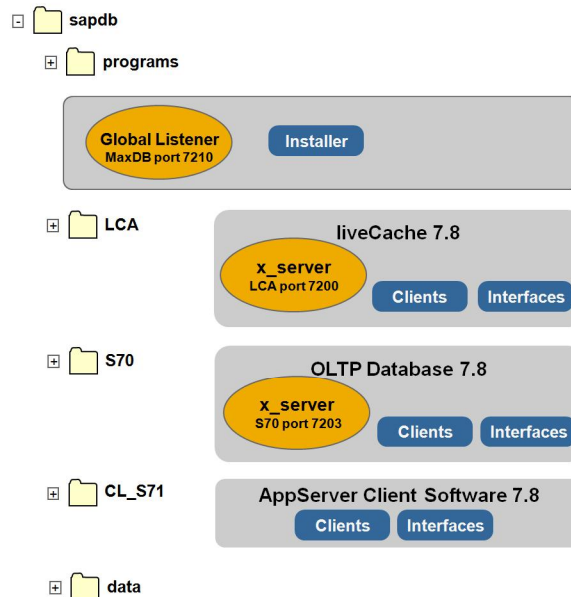Shows e.g. the installation path, the private data path, the global program and global data path.

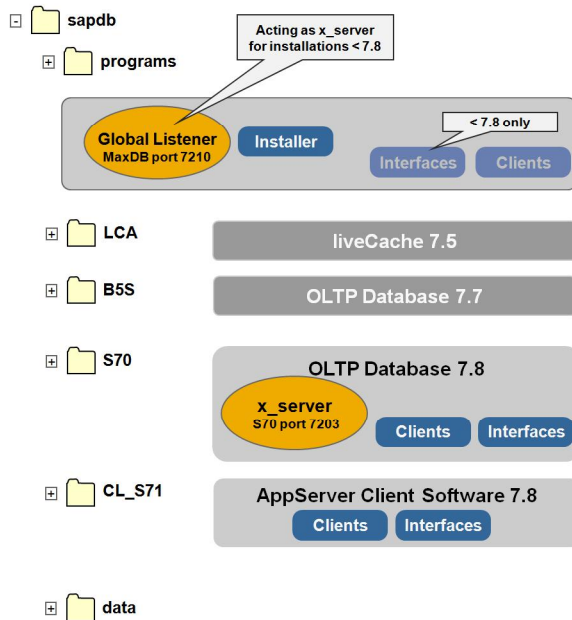## MaxDB 7.8 Installation

MaxDB/liveCache < 7.8

- □ 📁 sapdb
  - ⊞ 📁 programs

    x_server
    MaxDB port 7210
    Installer
    Interfaces   Clients

  - ⊞ 📁 LCA   liveCache 7.5
  - ⊞ 📁 S70   OLTP Database 7.6
  - ⊞ 📁 data

MaxDB/liveCache 7.8

- □ 📁 sapdb
  - ⊞ 📁 programs

    Global Listener
    MaxDB port 7210
    Installer

  - ⊞ 📁 LCA   liveCache 7.8

    x_server
    LCA port 7200
    Clients   Interfaces

  - ⊞ 📁 S70   OLTP Database 7.8

    x_server
    S70 port 7203
    Clients   Interfaces

  - ⊞ 📁 CL_S71   AppServer Client Software 7.8
    Clients   Interfaces

  - ⊞ 📁 data

The interfaces and the client software, which were part of the *independent global path* in earlier versions, are now part of each MaxDB server installation. The client software can furthermore be installed as a stand-alone installation side-by-side to the MaxDB server installations. Several client installations – of the same or of different versions – can be installed side-by-side as well. This way different application servers running on the same host can use their own client installation. Just make sure that the PATH variable used by each application server points to the correct *installation path*.

MaxDB Installations Side by Side

sapdb
- programs

Acting as x_server for installations < 7.8

Global Listener MaxDB port 7210 | Installer | < 7.8 only | Interfaces | Clients

- LCA — liveCache 7.5
- B5S — OLTP Database 7.7
- S70 — OLTP Database 7.8 — x_server S70 port 7203 | Clients | Interfaces
- CL_S71 — AppServer Client Software 7.8 | Clients | Interfaces
- data

If older MaxDB versions are installed on the same server, the global listener acts as X-Server for these databases.

The *global programs directory* then contains the interfaces and clients of these older software installations.

The clients and interfaces of version 7.8 are stored in the *installation path* of that installation.

MaxDB 7.8 Data Directories

■ All data files are stored in subdirectories of the *Global Data Path*.
■ There are installation specific directories.

sapdb
  data
    config — < 7.8 only — Install Registry | Database Parameter Files
    installations
      LCA — Install Registry | Installation Log Files
      S71Runtime — Install Registry | Installation Log Files
    wrk — Installation Log Files | Database Run Directories
    globals — Install Registry
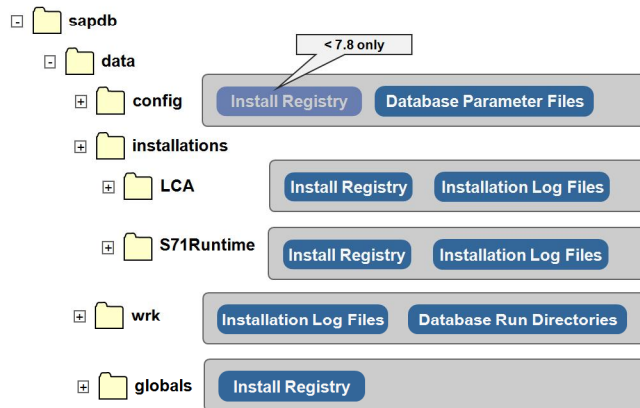
No *private data path* activated:

The install registry and installation log files are now stored installation-dependent in different subdirectories of the *global data path*. The install registry of the global components is stored in directory *<global data path>/globals*.

The database rundirectories are still stored in *<global data path>/wrk* and the parameter files are still stored in *<global data path>/config*.

If older MaxDB versions are installed on the same server, their installation log files are stored in *<global data path>/wrk* and an additional install registry is stored in *<global data path>/config*.

- New installation parameter *Private Data Path* (Option `-private_data <path>`)
- Used in standard SAP installations.

⊟ 📁 **sapdb**
   ⊟ 📁 **data**

< 7.8 only   Installations w/o private data path

     ⊞ 📁 **config**   [Install Registry] [Database Parameter Files]

     ⊞ 📁 **installations**

Installations w/o private data path

        ⊞ 📁 **LCA**   [Install Registry] [Installation Log Files]

     ⊞ 📁 **wrk**   [Installation Log Files] [Database Run Directories]

Installations w/o private data path

     ⊞ 📁 **globals**   [Install Registry]

  ⊟ 📁 **S70**

Installation with private data path

     ⊟ 📁 **data**

       ⊞ 📁 **config**   [Database Parameter Files]

       ⊞ 📁 **installations**

          ⊞ 📁 **S70**   [Install Registry] [Installation Log Files]

       ⊞ 📁 **wrk**   [Database Run Directories]

For SAP installations the *private data path* is activated.

This causes

- the database *rundirectory* to be stored in *<private data path>/wrk*

- the database parameter file to be stored in *<private data path>/config*

- the install registry and installation log files to be stored in *<private data path>/installations*.

The data of older installations is still stored in the old locations.

Installations with *private data path* enabled can exist side-by-side with installations without *private data path* enabled.

DEMO

All Expert Sessions use a database called EXPERTDB. This short demo shows the directory structure of this database, which was updated to version 7.8 in the last Expert Session.

Installation Wizard

SAP

Integrated *Installation Wizard* in Database Studio (also available in version 7.7.07)

- New installation of instances
- Simplified and advanced installation mode
- Database copy via backup/restore
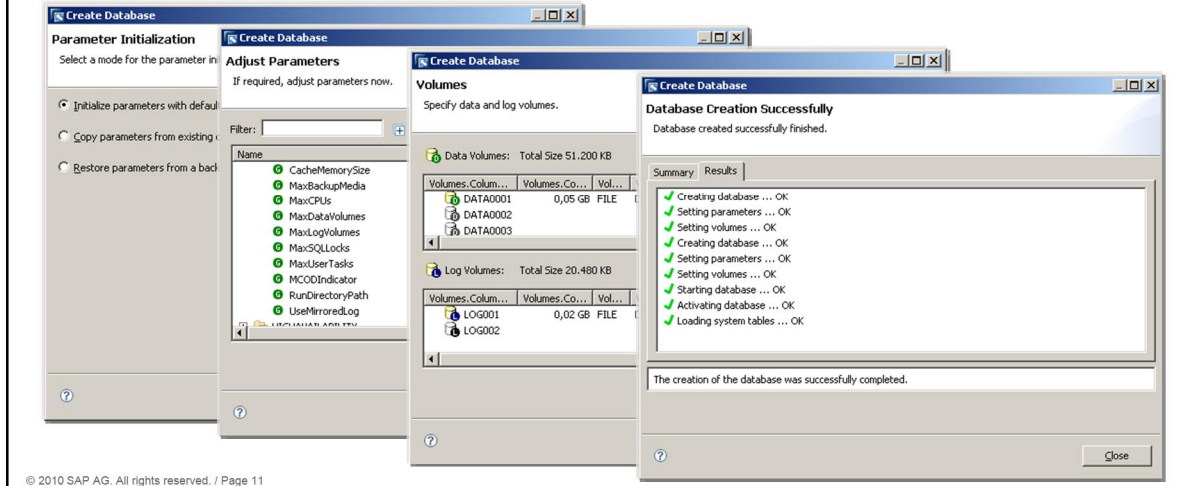
In the *Create Database Wizard* of Database Studio you can choose to use either a simple or an advanced mode for the database installation.

In the *simple installation mode* only a few database parameters have to be set manually, most parameters are pre-defined. The database is installed with one data and one log volume.

In the *advanced mode* all parameters can be set and any number of volumes can be configured. Furthermore in the advanced mode it is possible to restore a backup during the installation process. This is e.g. used for a system copy.

Security Enhancements

MaxDB 7.8 improves the system security

- New password identification algorithm, which now allows passwords to be longer than 9 characters
- The following general restrictions apply to user names and passwords:
    - User names must not have more than 18 characters.
    - Passwords must not have more than 256 characters.

SAP MaxDB clients with software version < 7.8 can only handle users with a maximum password length of 9 characters. So if an old client software version is used in combination with a MaxDB version >= 7.8 the used passwords still can only have a length of 9 characters. Otherwise the old client won't be able to connect to the database.

Backup Templates

> **New *Backup Template* definition supporting all relevant backup/restore scenarios**

- New syntax for *Backup Template* (former *Medium*) creation
- Supporting external backups via snapshot clone/split mirror, encrypted backups and compressed backups
- Templates are stored in XML format in file *BackupTemplates.xml*.

```
backup_template_create MyZipBackup TO BACKUPTOOL NETWORKER
    PIPE "/sapdb/SID/archive/mypipe"
    COMPRESSED CONTENT COMPLETE DATA BLOCKSIZE 8
    ENCRYPT AES256-CBC OWNER myself
```

With MaxDB version 7.8 new backup features are supported. To be able to do that, the backup templates had to be extended. Therefore they are now stored in an XML format and new dbmcli commands were implemented to maintain the backup templates:

backup_template_create <template_name> TO (<ext_snapshot_rule> | (
            <backup_destination> <backup_content>
            [<template_options>]))

backup_template_delete <backup template name>

backup_templates_file_date_show

backup_template_show   ([BY] NAME <backup template>) | ALL

backup_template_show_next

SAP

Integration of external backups via snapshot clones or split mirror into the MaxDB backup history

- External backups are supported for I/O consistent snapshots and mirrors.
- MaxDB provides an integration into the backup history for those backups.

```
backup_template_create ExternalBackup TO EXTERNAL SNAPSHOT
backup_start ExternalBackup
    <Create the I/O consistent Snapshot Clone>
backup_finish ExternalBackup ExternalBackupID <Returned ID>
```

A backup of the data area exists only when the data area is copied physically. This applies to the consistent split of a mirrored disk or the cloning of a consistent snapshot, for example. A snapshot in a disk system is not a data backup if there is no physical copy of the snapshot data.

Snapshot clones and split mirrors are not controlled by the MaxDB backup mechanism. For a system restore, you must be able to perform the backup without MaxDB tools.

A backup of the data volumes at file system level is not a data backup because this does not guarantee the input/output (I/O) consistency.

DBMCLI commands for external backups:

backup_template_create <template_name> TO EXTERNAL SNAPSHOT

backup_start <template_name>

          <create the IO consistent Snapshot clone>

backup_finish <template_name> ExternalBackupID <external_backup_ID>

          | <template_name> [AS] FAILED [WITH [ERROR] <tool_rc>]

Compressed Backups

Integrated support of compressed backups

- Automatic compression and decompression of backups
- Useful for full/incremental data backups and log backups
- Database I/O threads execute the compression and decompression using a gzip algorithm (Level 6)
- Compression of log backup especially makes sense when using multiple log queues or multiple log partitions
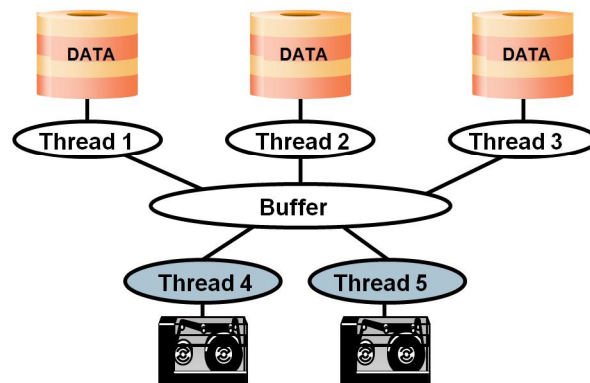
The backup compression can only be used for backup templates of the FILE and PIPE types.

In tests a compression to 25% of the original backup size has been reached for standard SAP systems.

DBMCLI commands:

backup_template_create <template_name> TO <backup_destination> COMPRESSED <backup_content> [<template_options>]

backup_start <template name>

Encrypted Backups

SAP

Integrated backup encryption and decryption

- Automatic encryption and decryption of backups
- Needs a PSE file (See SAP Note 662340 - SSF Encryption Using the SAPCryptolib)
- Encryption algorithm:
    - AES128-CBC
    - AES192-CBC
    - AES256-CBC
    - DES-CBC
    - DES3-CBC
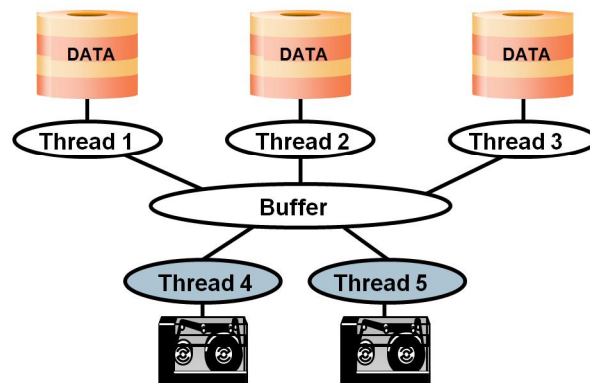
Prerequisite: You have created a private-public key pair using sapgenpse.

For more information about sapgenpse and SAP Cryptolib, see the online documentation, chapter *Using the SAP Cryptographic Library for SNC*.

When creating the backup template for your backup, specify the encryption algorithm and the private key owner in the backup template properties.

DBMCLI commands:

backup_template_create <template_name> TO <backup_destination> <backup_content> ENCRYPT [USING] <AES128-CBC | AES192-CBC | AES256-CBC | DES-CBC | DES3-CBC> [FOR] {OWNER <owner>}

backup_start <template name>

New Events

New events allow enhanced monitoring

- New events in MaxDB 7.8
    - AutoLogBackup
      0: The automatic log backup task has been started
      1: The automatic log backup task has been stopped
      2: Automatic log backup has been enabled
      3: Automatic log backup has been disabled
      4: A log backup was successfully finished
    - BackupResult
      Sends the return code of a finished backup
    - CheckData
      Sends 0 or an error number when a
      check data finishes
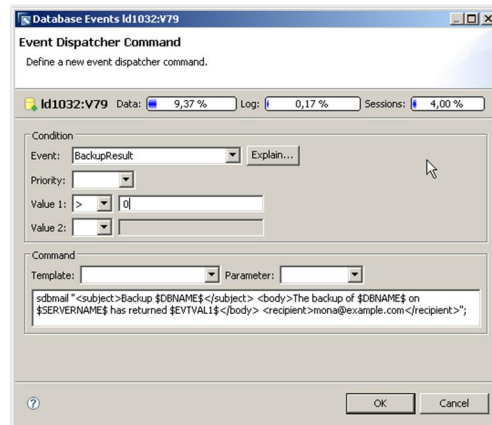
The database manager event dispatcher was already explained in detail in the Expert Session 'New Features in SAP MAxDB version 7.7'. With database version 7.8 a couple of new events has been introduced. They can be configured using dbmcli or Database Studio.

DBMCLI commands:

event_dispatcher ADD NAME == <event_name> [PRIORITY == <priority>]

        [VALUE1 (==|>=|<=|>|<) <value1>] [VALUE2

        (==|>=|<=|>|<) <value2>] COMMAND == <command>

event_dispatcher SHOW

event_dispatcher DELETE <entry_ID>

Enhanced Automatic Recreation of Bad Indexes

Optional automatic recreation of bad indexes

- Redoing log is sped up by omitting index creation for CREATE INDEX statements
- Automatic recreation of bad indexes during online operation or after restart can be activated
- Different rules for unique and non unique indexes can be applied
- Implementation uses the event dispatcher

Automatic Recreation of Bad Indexes for lddbqtx:QTX

Automatic Recreation of Bad Indexes

lddbqtx:QTX Data: 90,54 %   Log: 24,20 %   Sessions: 17,20 %

Automatic recreation of bad indexes is currently deactivated.

○ Activate automatic recreation of bad indexes
  Options for automatic recreation of bad indexes
  ● UNIQUE - indexes are automatically repaired if they are broken
  ○ ALL - indexes that are detected to be bad are automatically repaired

Activate   Cancel

You can instruct the database to recreate bad indexes automatically. You can also specify whether you want all bad indexes to be recreated automatically or only unique indexes that prove to be bad.

If the function is ON, the system recreates bad indexes after having received a respective event from the database kernel.

DBMCLI-Command:
 auto_recreate_bad_index <mode>

Options:

    <mode>   New mode for the database. Possible values are:

    ALL   All indices that are detected to be bad are automatically repaired.

    UNIQUE Unique indices are automatically repaired if they are broken.

    OFF   The database system does not perform any action if bad indices are detected.

    SHOW   Show current status of automatic bad index recreation feature.

DEMO

In this short demo it is shown how a normal and a compressed data backup can be created. The size difference is shown.

Furthermore it is explained how to configure a backup template for encryption and for external snapshots with Database Studio.

Additionally an event is configured, which sends an email in case the autmatic log backup gets disabled.

Agenda

1. Administration
2. **Performance**
   - Dynamic CPU Adjustment
   - Load Balancing
   - Prefetching
   - Volume Balancing
   - Cache Pinning
   - Variable Output
3. Monitoring

In this chapter about performance improvements, the kernel features dynamic CPU adjustment, load balancing, prefetching, volume balancing, cache pinning and variable output are explained.

Dynamic CPU Utilization

Adjust number of usable CPUs during online operation

- Modify the number of usable CPUs between 1 and MaxCPUs
- Parameter:
  UseableCPUs : 1 - MaxCPUs

User Kernel Thread	UKT 2	UKT 3
user user user user	user user user	user user ser
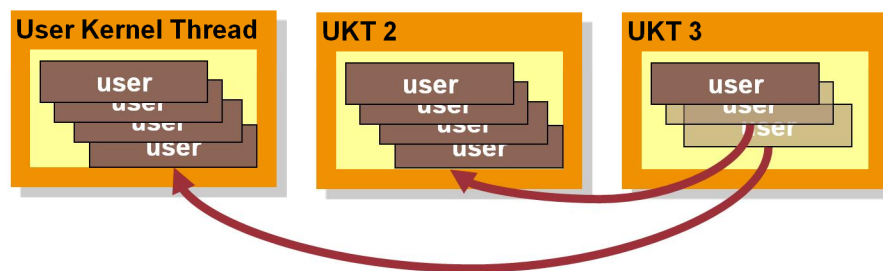
MaxDB can dynamically adjust the number of CPU cores to be used. The dispatcher moves user tasks out of the inactive user kernel threads when the tasks become active.

The parameter can be changed in ONLINE mode.

This is especially useful in systems were hardware resources can be dynamically added or removed.

Load Balancing

CPU load balancing now is default for MaxDB and liveCache

- Load balancing can significantly improve the scalability of the database
- Runnable tasks are moved to idle threads when the current thread is busy
- The communication interface detects reasonable moves
- Parameter:
  LoadBalancingCheckInterval : > 0

User Kernel Thread
user
user
user
Tasks

UKT 2
user
user
user
Tasks

UKT 3
user
user
user
Tasks
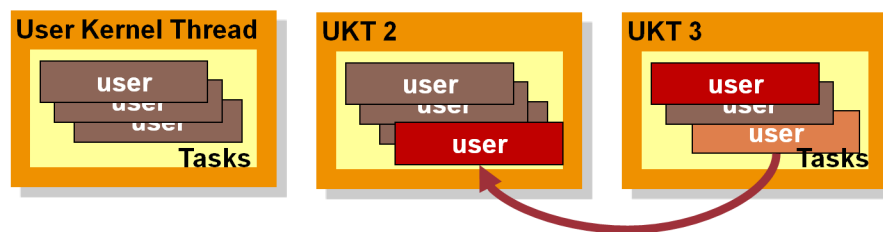
Load balancing enables an optimal utilization of all threads and thus of all the CPUs allocated to the database.

After the time interval of **LoadBalancingCheckInterval** seconds, the database kernel searches for a task to be moved to another UKT. This is helpful when one UKT has a particularly heavy load and another UKT has a smaller load.

Between the checks after **LoadBalancingCheckInterval** seconds, statistics are collected. The greater the time for gathering the data, the more meaningful the UKT load statistics that result. With smaller values, the shifting that occurs may not be optimal.

Moving tasks is time-consuming. It should only be done when it is expedient. Two more parameters **LoadBalancingWorkloadThreshold** and **LoadBalancingWorkloadDeviation** influence the behavior to make sure that not too many tasks are moved. In OLTP operation, unbalanced load distribution among the UKTs is often due to poorly-optimized statements with high CPU loads for a single job. Such statements should be identified and optimized. But nevertheless there are situations where load balancing makes sense and improves scalability.

In MaxDB versions < 7.8 load balancing was particularly used for liveCache instances. These often run very CPU-intensive LCA routines over long periods. Multiple LCA routines should not work sequentially on one CPU if another CPU is free.

The database console (x_cons) provides information about tasks that have been moved.

The parameter **LoadBalancingCheckInterval** can be changed in ONLINE mode.

I/O Prefetch for Scans

> **Full and range table scans benefit from the new I/O thread implementation in version 7.7. User tasks send asynchronous I/O orders to the I/O systems and continue gathering more pages of the table.**

- Prefetch-I/O use the priority medium whereby single block I/O use the priority high.
- Create index uses the prefetching if the result index tree doesn't fit into the cache.
- Parameter:
  The read ahead becomes active if a table access affects more than **ReadAheadTableThreshold** pages.
- Monitoring
  - The view SYSINFO.IOJOBS shows current I/O in the system
    Read Ahead I/O orders have the name "DataCache: …".
  - The view SYSDBA.MONITOR_PAGES shows a summary of read ahead requests and physical reads via read ahead

MaxDB uses parallel I/O requests to speed up table scans and table range scans. User tasks read index level 1 pages into the cache, determine the volume block positions of the pages stored in the separators by reading the converter and send asynchronous I/O requests to the I/O system. The user task doesn't wait for every single I/O before sending the next I/O request.

User tasks use asynchronous I/O requests if the size of the scan exceeds the number of pages specified as value of the parameter **ReadAheadTableThreshold.** The query optimizer evaluates the range size before the statement execution starts.

The database uses asynchronous I/O for scans only, if the number of current running I/O requests is below the value of **MaxDataAreaReadAheadRequests**. The determination of the current running I/O requests happens during the operation on the index level 1 page. This operation prevents the system from I/O overload situations. I/O orders for short SQL commands should not be blocked by the asynchronous parallel I/O.

Asynchronous I/O read requests have the priority low.

Data Volume Balancing

Data on volumes is redistributed after adding a volume.

Triggers for balancing are:

- Add data volume
- Uneven data volume filling levels during restart

Data-Volumes
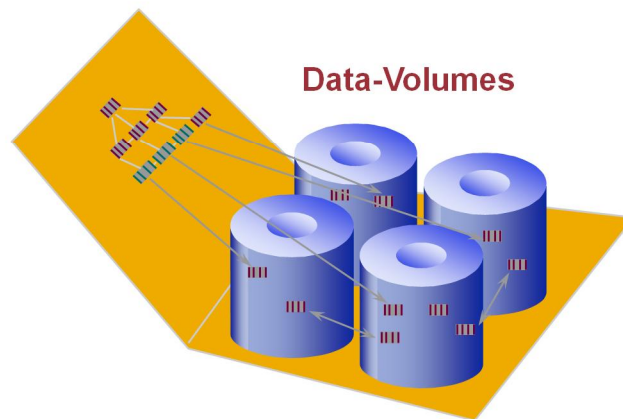
An important role in the access performance of the database is played by the MaxDB striping mechanism, which distributes the data pages evenly on the disks. Additional striping can be performed by the hardware.

Striping guarantees even distribution of the I/O load on the available disks. Even load balancing of all the data volumes in the database also prevents individual data volumes from overflowing. A table can be larger than a single data volume without the need for maintenance tasks to be carried out.

If new data volumes are added the database must ensure that some data is copied automatically to the new disk to get an even distribution again. This additional I/O must be done without affecting the normal operation.

In versions 7.4 – 7.7.05 volume balancing is not available. As of version 7.7.06.08 the feature has been introduced again.

Table Cache Pinning

The data cache management can use different LRU (least recently used) strategies for indicated tables.

- Keep data of a table in cache
    CREATE TABLE … CACHE
    ALTER   TABLE … CACHE

- The database parameter **DataCachePinAreaThreshold** specifies the maximum percentage of the data cache used for tables having the CACHE attribute.

- Remove data of a table from cache earlier
    CREATE TABLE … NOCACHE
    ALTER   TABLE … NOCACHE

As of version 7.8 MaxDB can hold pages of tables and indexes having the CACHE attribute in a LRU pin area of the data cache. The parameter **DataCachePinAreaThreshold** defines the maximum size of the data cache in percent used as pin area.

Pages that have to be swapped out of the pin area will be inserted into the global LRU list at the beginning of the last 10%. The pages will be linked back into the beginning of the PIN LRU list when they are reused.

Additionally the SQL Manager can occupy pages in the cache. The pages are marked as "in use" for a longer timeframe. Those pages will not be swapped out of the cache because they are in use. This allows a direct memory access by the SQL Manager to the cache pages. The creation of result sets speeds up significantly.

(Affected Parameter is **TempResultsetPinAreaThreshold**.)

All liveCache data is considered to have the CACHE attribute and thus will use the part of the data cache reserved via **DataCachePinAreaThreshold** if the parameter **EnableLivecachePinAreaUsage** has the value YES.

Variable Output

Output packages transport variable length field values to the database client

- Variable input has been implemented with version 7.7
- Parameter:
  EnableVariableOutput: YES/NO

```
CREATE TABLE mytab
  (k1 varchar(3),
  (f1 varchar(255),
   primary key(k1)

INSERT INTO mytab VALUES ('001','abcd')

SELECT f1 from mytab
➔ transport only 4 byte for f1 plus length information
```

The parameters **EnableVariableInput** and (as of 7.8 additionally) **EnableVariableOutput** determine how the system transfers the content of variables between client and kernel.

The system either transfers the actual length of the content or the maximum possible length of the content. In the latter case, the system fills up the content up to the maximum length (depending on the data type) which is space consuming. Setting the parameter to YES reduces communication costs.

Changes to these parameters only apply after a restart of the database.

Agenda
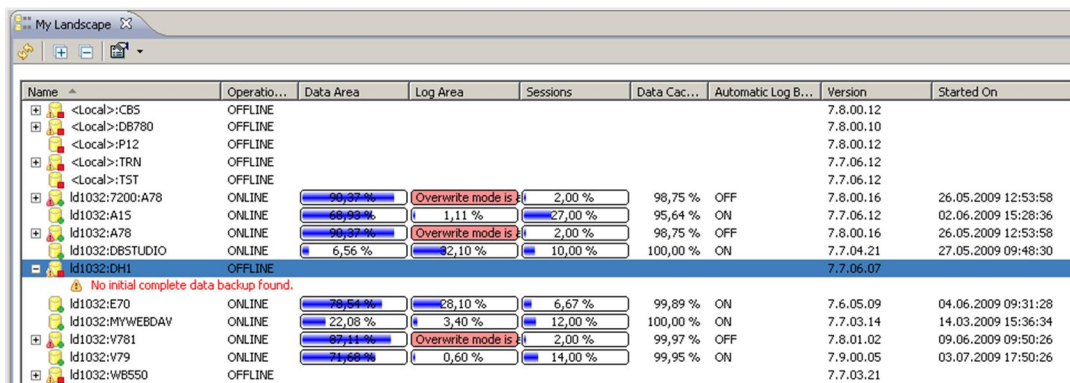
1. Administration
2. Performance
3. **Monitoring**
   - Databases Monitor
   - SQL Performance Analysis

In this last part of this session two monitoring enhancements of Database Studio are presented: the Databases Monitor and the SQL Performance Analysis.

Landscape monitoring in a nutshell

- Monitoring window per landscape with most important status information like
  - Database filling rate
  - Log filling rate
  - Number of connected sessions
  - Other configurable columns

My Landscape ⊠

Name ▲	Operatio...	Data Area	Log Area	Sessions	Data Cac...	Automatic Log B...	Version	Started On
⊞ <Local>:CBS	OFFLINE						7.8.00.12	
⊞ <Local>:DB780	OFFLINE						7.8.00.10	
<Local>:P12	OFFLINE						7.8.00.12	
⊞ <Local>:TRN	OFFLINE						7.7.06.12	
<Local>:TST	OFFLINE						7.7.06.12	
⊞ ld1032:7200:A78	ONLINE	90,37 %	Overwrite mode is	2,00 %	98,75 %	OFF	7.8.00.16	26.05.2009 12:53:58
ld1032:A1S	ONLINE	68,93 %	1,11 %	27,00 %	95,64 %	ON	7.7.06.12	02.06.2009 15:28:36
⊞ ld1032:A78	ONLINE	90,37 %	Overwrite mode is	2,00 %	98,75 %	OFF	7.8.00.16	26.05.2009 12:53:58
ld1032:DBSTUDIO	ONLINE	6,56 %	32,10 %	10,00 %	100,00 %	ON	7.7.04.21	27.05.2009 09:48:30
⊟ ld1032:DH1	OFFLINE						7.7.06.07	
⚠ No initial complete data backup found.								
ld1032:E70	ONLINE	78,54 %	28,10 %	6,67 %	99,89 %	ON	7.6.05.09	04.06.2009 09:31:28
ld1032:MYWEBDAV	ONLINE	22,08 %	3,40 %	12,00 %	100,00 %	ON	7.7.03.14	14.03.2009 15:36:34
⊞ ld1032:V781	ONLINE	87,11 %	Overwrite mode is	2,00 %	99,97 %	OFF	7.8.01.02	09.06.2009 09:50:26
ld1032:V79	ONLINE	71,68 %	0,60 %	14,00 %	99,95 %	ON	7.9.00.05	03.07.2009 17:50:26
⊞ ld1032:WB550	OFFLINE						7.7.03.21	

The *Databases Monitor* provides an overview of the state of all databases integrated in one landscape in Database Studio. On the first glance you'll get information about critical database states like db_full, log_full or database down.

Opening the *Databases Monitor*:

In the explorer tree, go to the top level using the (*Up One Level*) button.

Select the database landscape that you want to monitor.

In the context menu of the database landscape, choose *Databases Monitor*.

It is possible to add or remove columns from the displayed view. Furthermore the refresh interval can be specified and the *Databases Monitor* can be configured to be opened automatically, when Database Studio is started.

Statement Monitoring

SAP

Better monitoring with default settings

- Statement monitoring is ON as default
- Shared SQL cache provides monitoring data
- Reduced impact on system load and scalability

© 2010 SAP AG. All rights reserved. / Page 29

SQL monitoring measures runtimes, I/O accesses, and the selectivity of SQL statements, and calculates the total costs for repeatedly executed SQL statements. This lets you identify SQL statements that have short runtimes individually, but whose frequent execution results in a heavy load.

This monitoring is based on the Shared SQL cache and is activated as a default. The data is collected in system table COMMANDSTATISTICS.

SAP

SQL command monitoring in Database Studio

SQL Performance Analysis (lddbqtx:QTX)

lddbqtx:QTX ONLINE Data: 89,51 %   Log: 6,76 %   Sessions: 13,20 %

Monitoring and Analysis | Bottleneck Candidates | Bottleneck Candidate Detail

Bottleneck Candidates
This page lists all statements that match at least one of the filter pro

Bottleneck Candidates

Cum Runtime ▼	Statement
946.631.339,000	SELECT "LDATA" FROM "REPOLOAD" WHE
363.126.853,000	SELECT TABNAME, BLOCKNR, FIELDSLG, I
114.433.036,000	SELECT NUM_TABS_FEW_SAMPLE_ROWS
53.109.089,000	SELECT "UNAM", "UDAT", "UTIME", "L_D
50.286.549,000	SELECT T_00 . "INTF_NAME" , T_00 . "SO
47.385.158,000	SELECT "SQLX" , "EDTX" , "DBNA" , "CLAS
32.241.024,000	SELECT "DATA" FROM "REPOSRC" WHERE
29.340.423,000	SELECT * INTO ?,?,?,?,? FROM "PKRT_OB
26.770.059,000	SELECT "DATA" FROM "REPOSRC" WHERE
26.760.808,000	SELECT COUNT(*) INTO ? FROM DDNTT V
22.050.802,000	SELECT /*+ KEYACCESS */ * FROM "TCP
21.229.298,000	SELECT * INTO ?,?,?,?,?,?,?,?,?,?,?,?,?,
20.894.295,000	SELECT T_01 . "META_KEY" , T_01 . "FIEL
20.797.259,000	SELECT "TABNAME" FROM "D010TAB" WH
19.974.864,000	SELECT "TABNAME" FROM "D010TAB" WH
18.386.462,000	INSERT INTO "D010TAB" ("MASTER" , "TA
17.125.151,000	SELECT "OBJECT_ID_USED" , "INDEX_USE
16.113.857,000	SELECT "SQLX" , "EDTX" , "DBNA" , "CLAS
15.203.541,000	INSERT INTO "D010TAB" ("MASTER" , "TA
14.119.005,000	SELECT /*+ KEYACCESS */ * FROM "TCPS
13.174.899,000	SELECT TABNAME, BLOCKNR, FIELDSLG, I
11.498.729,000	DELETE FROM "D010TAB" WHERE "MASTE
10.741.853,000	SELECT /*+ ORDERED */ T_02 . "INTF_N/
10.293.108,000	UPDATE "PKRT_LOAD" SET IS_VALID = ? ,
9.768.715,000	INSERT INTO "D010INC" ("MASTER" , "IN
8.591.219,000	SELECT "INCLUDE" FROM "D010INC" WHE
8.351.312,000	SELECT * INTO ?,?,?,? FROM "PKRT_OB
7.824.319,000	SELECT "INCLUDE" FROM "D010INC" WHE
7.151.279,000	SELECT * INTO ?,?,?,?,?,?,?,?,? FROM "
7.111.967,000	SELECT "DATA" FROM "REPOTEXT" WHER

SQL Performance Analysis (lddbqtx:QTX)

lddbqtx:QTX ONLINE Data: 89,52 %   Log: 6,71 %   Sessions: 12,80 %

Monitoring and Analysis | Bottleneck Candidates | Bottleneck Candidate Detail

Bottleneck Candidate Detail
This page lists all single executions of the statement along with the parameters used if any. In case of SELECT statements an explain is provided as well.

Single Executions (no single executions available)

Statement	Time	Runtime	Selectivity	Rows Read	Rows Qual	Rows Fetched	Fetch Calls	Cache IO	Physical IO	Page Access	Page Access/Rows Qual	Strategy
SELECT COUNT(*) INTO ? FROM DDNTT WHERE TABFORM <> ...												

Statement

```
SELECT COUNT(*)
INTO ?
FROM DDNTT
WHERE TABFORM <> 'J'
AND TABFORM <> 'V'
```

Parameter

No	Data Type	Value

Explain

	Owner\Schemaname	Tablename	Column or Index	Strategy	Pagecount|Costvalue
SELECT COUNT(*) INTO ? FROM DDNTT WHERE TABF					
Table Access	SAPQTX	DDNTT		TABLE SCAN	10665
Result				RESULT IS COPIED , COSTVALUE IS	10667

SQL performance analysis in Database Studio lets you investigate which SQL statements that the application triggers are particularly cost-intensive for the database, and why this might be the case.

As already explained, SQL monitoring is activated as a default.

If you activate *Detail Collection* as well, then the system records the executions of SQL statements that meet the criteria you define, as well as the parameters used. If you do not activate the *Detail Collection*, the system only records aggregated values for the execution of SQL statements.

Filters let you restrict the list of SQL statements that the system records in SQL monitoring. All the SQL statements that fulfill at least one of the filter conditions are displayed (OR operator). The following types of filters are available:

Cumulative Filters: These filters use the information that the system collects during simple SQL monitoring. These filters help you find SQL statements that the system runs frequently. Minor improvements in the performance of such SQL statements can result in large performance gains in the overall system.

Single Filters: These filters use information that the system collected in the *Detail Collection*. They help you find SQL statements for which single executions are particularly slow. When you change the filter condition of a single filter, the system modifies the corresponding *Detail Collection* condition in the SQL monitoring settings automatically – i.e. the recording criteria of the *Detail Collection* is changed.

Identification of clients running commands

- Sessions know the Shared SQL command ID of commands they are running

System table SESSIONS contains the CURRENTCOMMANDID. Using this column a JOIN to table COMMANDSTATISTICS can be executed, which provides the information, which SQL statement is currently executed by a specific database session. Further useful information is provided in column APPLICATIONPROCESS. With this information one can check in transaction SM50/SM66 which application process and which application program is executing that SQL statement.

Running Commands in DBACOCKPIT

SAP

Running commands are displayed in the Task Manager

As of SAP release 7.20 Basis SP 03 the *Task Manager* in transaction DBACOCKPIT shows the SQL statements which are executed by active user tasks. A double-click on the statement shows the complete SQL statement.

Questions & Answers

SAP

Thank You!
Bye, Bye – And Remember Next Session

May 04, 2010	**Session 9:** **SAP MaxDB Optimized for SAP Business Warehouse**

Thanks a lot for your interst in this session.

The recording and the PDF will be available at http://maxdb.sap.com/training soon.

Please also mark your calendar for our next session. Topic *SAP MaxDB Optimized for SAP Business Warehouse* will be presented on May 04, 2010

Thank you!