DB2 UDB for UNIX, Linux, and Intel Platforms

Technology Preview

DB2 Worldwide Presales Support
IBM Toronto Laboratory

the power of ONE

Disclaimers and Trademarks

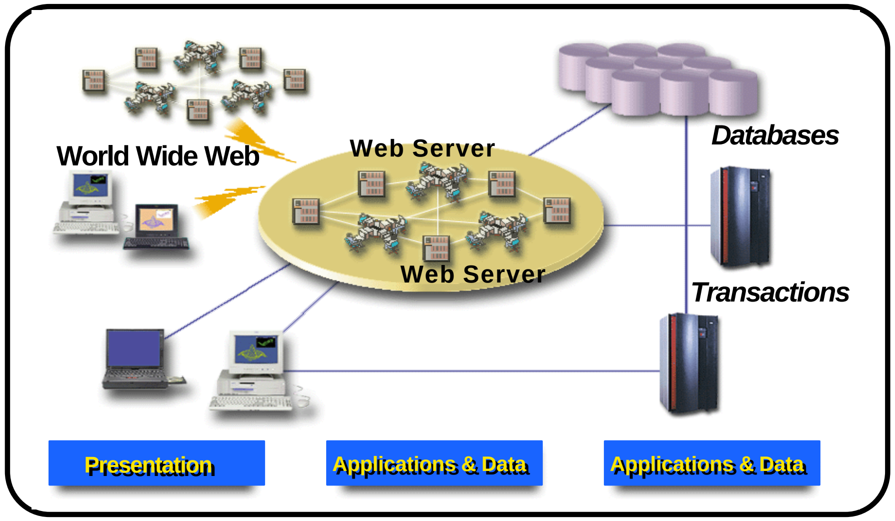
DB2 UDB for UNIX, Linux, and Intel Platforms

Product Investment Strategy

DB2 Worldwide Presales Support
IBM Toronto Laboratory

Investment Areas

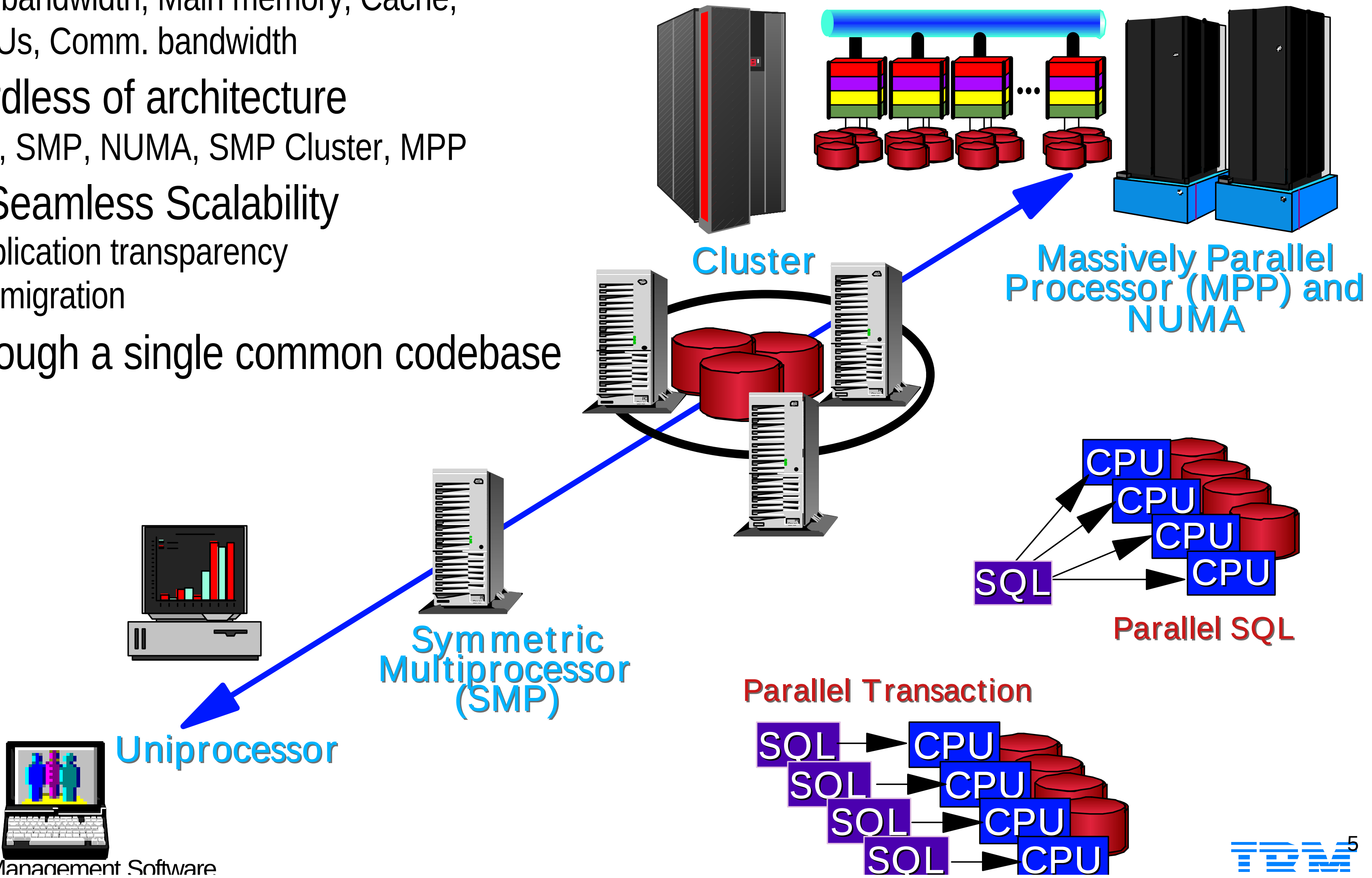
- Overall
  - Solution Oriented
    - key partners (SAP, Siebel, PeopleSoft, i2, Ariba)
    - within IBM (WebSphere, Tivoli, Lotus)
- Technology Oriented -> DB2 as an:
  - e-Commerce server
  - Business Intelligence server
  - Information Integration Server



Fully Exploit All Resources

- Full exploitation of ALL available resources
  - Within a single query
  - Across separate queries
- Regardless of resource class/amount
  - I/O bandwidth, Main memory, Cache, CPUs, Comm. bandwidth
- Regardless of architecture
  - Uni, SMP, NUMA, SMP Cluster, MPP
- With Seamless Scalability
  - Application transparency
  - No migration
- All through a single common codebase

Our Overriding Design Point



DB2 UDB for UNIX, Linux, and Intel Platforms

Platform and Client Support

DB2 Worldwide Presales Support
IBM Toronto Laboratory

the power of ONE

Packaging/Platforms Proposals

- Platform Support
  - Stabilizing OS/2 and PTX at Version 7.2
    - NUMA h/w strategy is to support Windows
  - Tier 1 Platforms
    - AIX
    - Solaris
    - HP-UX
    - Windows
    - Linux

```
x-Series (Intel)
z-Series (Linux/S390)
```

- Satellite Edition Merges with Personal Edition
- DB2 EE merges with DB2 EEE
    - Enterprise Server Edition (ESE)
    - Clustering/Partitioning is an Option

Database Migration

- DB2 V6 and V7 migration to future versions will be supported
  - All versions of product including WG, EE and EEE
- DB2 Datajoiner V2.1.1 migration will also supported
- Following database entities are migrated:
  - Database configuration file
  - Log file header
  - Catalog tables
  - table space files (DB2 DJ V2.1.1 only)
- Tools
  - DB2CKMIG - migration tools ensures that:
    - A database is not in backup pending state
    - A database is not in rollforward pending state
    - Tablespace ID is in normal state
    - A database is not in an inconsistent state
  - DB2RBIND - post migration utility to rebind all existing packages
  - DB2UIDDL (Migration of Unique Indexes for pre-V5 indexes)

Supporting Customers with backlevel Databases

64-bit Support

- full 64-bit database
  - Larger buffer pools, sort heap, other resources
  - AIX, Solaris, and HP-UX as of Version 7.1

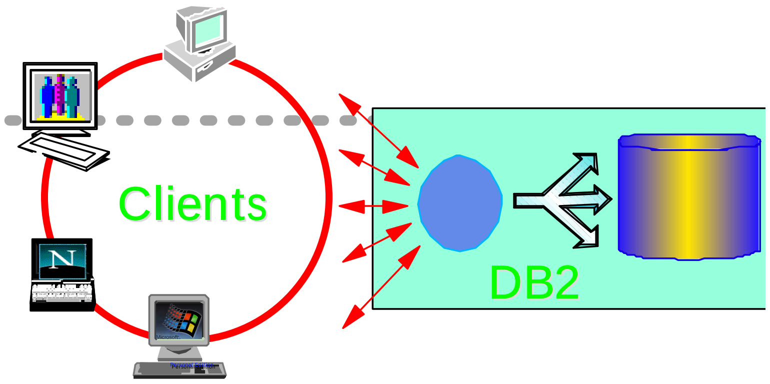
64 Bit Product	AIX PPC	Sun	HP	Linux IA64	Windows IA64
Personal Edition	N/A	N/A	N/A	yes	yes
Workgroup Edition	yes	yes	yes	yes	yes
EE / ESE	yes	yes	yes	yes	yes
Connect Enterprise	yes	yes	yes	yes	yes
App. Dev. Client	yes	yes	yes	yes	yes
Administration Client	yes	yes	yes	yes	yes
Runtime Client	yes	yes	yes	yes	yes
Relational Connect	yes	yes	yes	yes	yes
Datalinks Manager	no	no	no	no	no
Spatial Extender	yes	yes	yes	yes	yes
XML Extender	yes	yes	yes	yes	yes
Warehouse Manager	yes	yes	yes	yes	yes

Client Support

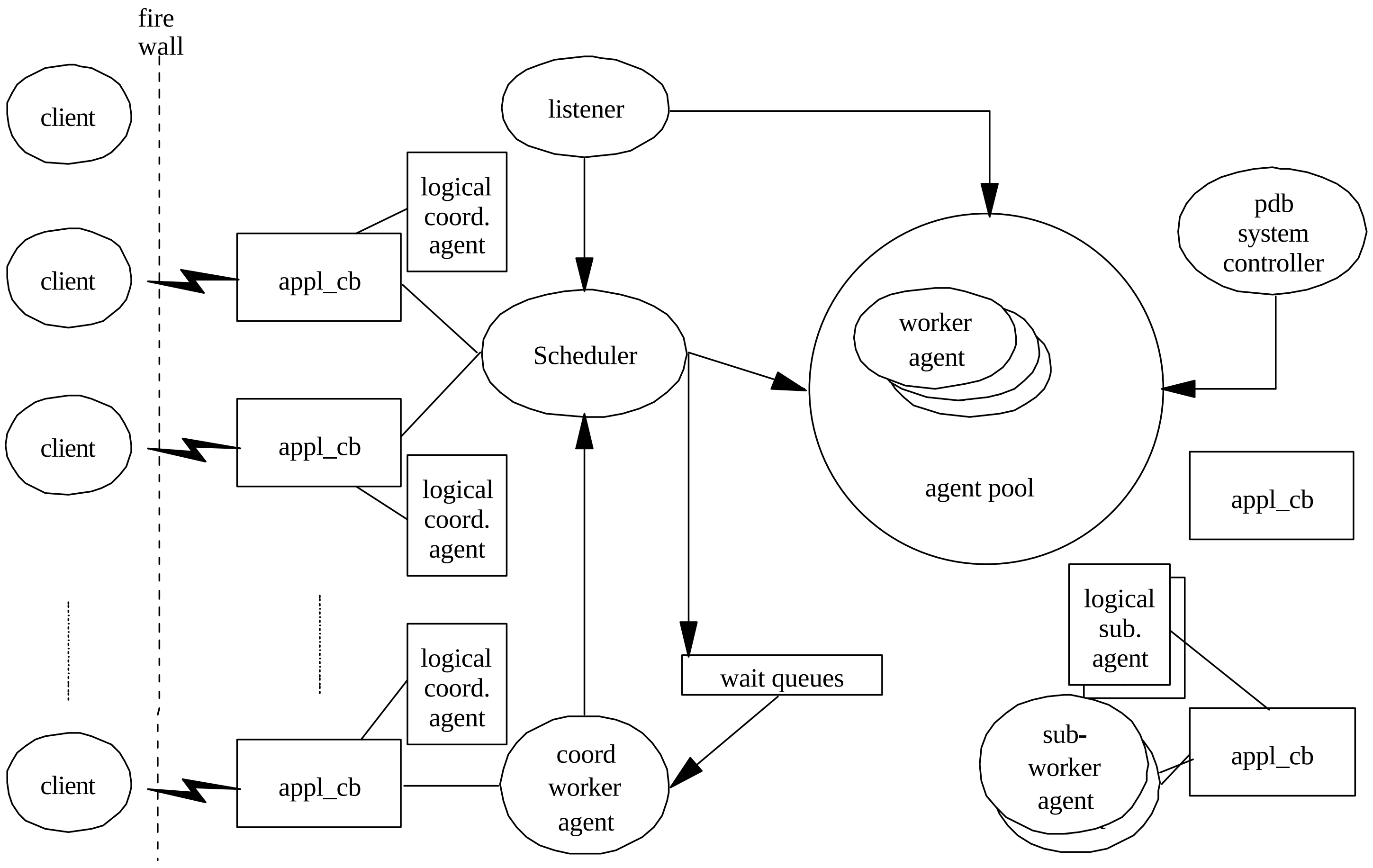
- New Client
  - can connect to all versions of new servers (32-bit/64-bit)
  - restrictions on downlevel feature support (to V7 servers)
  - can also connect to all other members of DB2 family without additional code

Client	UNIX 32-bit	UNIX 64-bit	Windows 32-bit	Windows 64-bit
V7 32-bit Client (Windows)	yes	no	yes	yes
V7 32-bit Client (UNIX)	yes	no	yes	no
V8 New Client	yes	yes	yes	yes

Connection Concentrator



- Connection Concentrator
  - Support large number of concurrently connected users
  - new concept of logical agents and database agents
  - n:m architecture



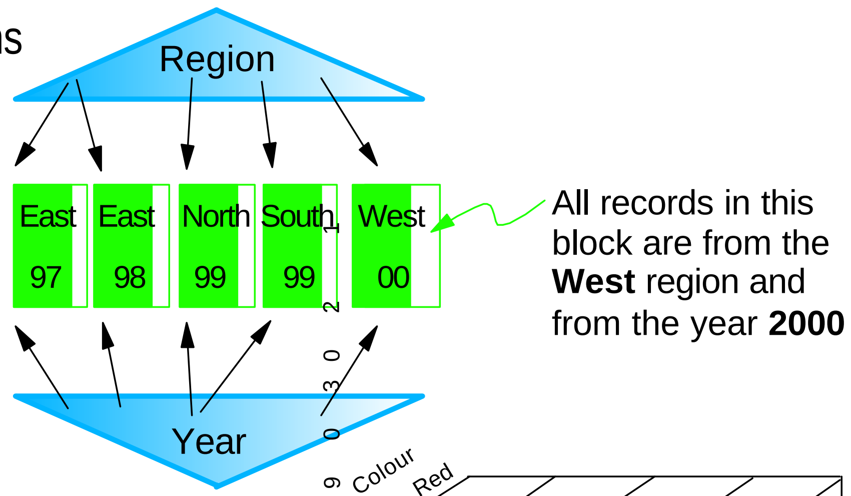
DB2 UDB for UNIX, Linux, and Intel Platforms

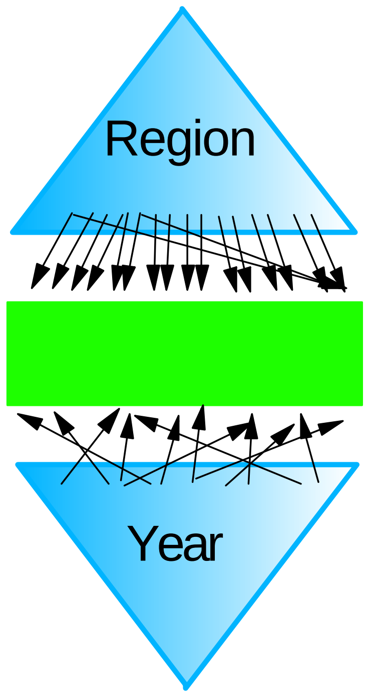
Business Intelligence Support

DB2 Worldwide Presales Support
IBM Toronto Laboratory

Multidimensional Clustering

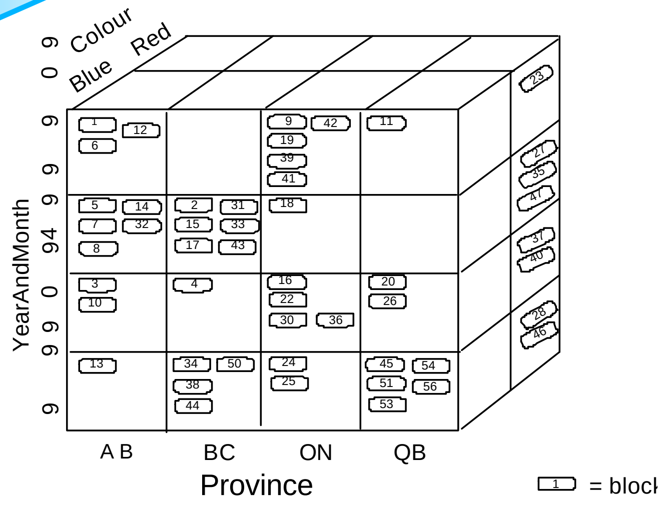
- Multidimensional Clustering
  - Provides range partitioning on multiple dimensions
  - Reduces need for indexing
  - Roll-in / roll-out improvements


Region
East 97
East 98
North 99
South 99
West 00
Year
All records in this block are from the **West** region and from the year **2000**



With MDC

- Clustering guaranteed !
- Smaller indexes
- Faster query response
- Simple definition syntax
- Fast roll-in & roll-out



Prior to MDC

- Clustering in one dimension only
- clustering NOT guaranteed (degrades once page free space is exhausted)

```
CREATE TABLE MDC1 (
      Date DATE,
      Province CHAR(2),
      Color VARCHAR(10),
      YearMonth generated as INTEGER(Date)/100, ... )
ORGANIZE BY DIMENSIONS (
      YearAndMonth, Province, Color
      )
```

Declare Global Temporary Tables

- Minimal undo logging
  - support the rollback of data changes to DGTT
  - NOT LOGGED clause mandatory in V7, but will become optional
- Index support
  - any standard index can be created on a temporary table
- Statistics support
  - RUNSTATS supported against the table

Catalog Caching

- This feature will extend the existing catalog cache to provide a cache on all nodes in an MPP system (i.e. become a distributed catalog cache) and cache:
  - systable information (including all its packed descriptors)
  - authorization information, including sysdbauth information and execute privileges of user-defined functions and stored procedures
- These caching enhancements will help to improve the overall performance of:
  - binding packages and compiling SQL statements, including usage of user-defined functions and stored procedures
  - operations that involve checking database-level privileges
  - operations that involve checking execute privilege for user-defined functions and stored procedures
- In particular, the performance of applications which are connected at non-catalog nodes will greatly benefit

User-Maintained Summary Tables

- Why support User Maintained Summary Tables?
  - Oracle supports User Maintained Summary Tables
    - easier to migrate existing Oracle users
  - Many warehouses have custom applications that maintain and load tables that are in reality user defined and maintained summary tables
    - It would be very beneficial to use them by the database routing mechanism.

```
>>-CREATE--+----------+---TABLE--table-name--------------------->
           '-SUMMARY--'

        .-MAINTAINED BY SYSTEM--.
>-----*-+-----------------------+------*-------------------|
        '-MAINTAINED BY USER----'
```

Null and Default Value Compression

- Reduce storage for typical data warehousing scenarios
  - Increase performance of large scans
- Available for all tables except global temporary tables
- Must use system default value, not user-defined values
- Eligible datatypes
  - Numeric
  - Char
  - Varchar
  - DBCS (fixed and variable)
  - BLOB

```
>>-CREATE--+----------+---TABLE--table-name--------------------->
           '-SUMMARY--'

       .-VALUE COMPRESSION-----.
>------+----------------------+-----------------------|
```

DB2 UDB for UNIX, Linux, and Intel Platforms

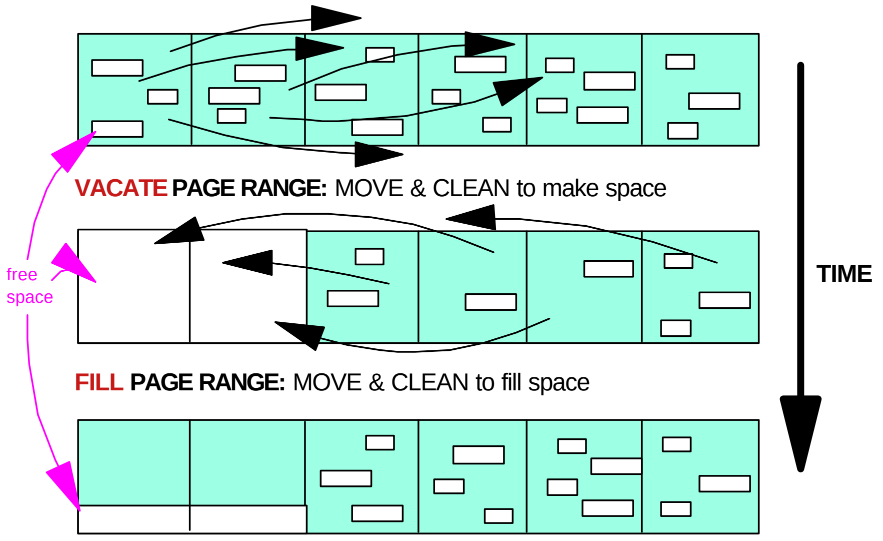
Online Maintenance Enhancements

DB2 Worldwide Presales Support

IBM Toronto Laboratory

Online Reorganization

- Feature
  - enhances database availability by allowing reorganization of tables while permitting read and write (online) table access
  - provides the capability for monitoring the status and progress of a table REORG
- Mechanism
  - table is reorganized in a piece-wise fashion and remains available to all users in either a readable (SHRLEVEL REFERENCE) or readable and writeable
  - incremental processing affords 'break points' such that the online REORG could be paused and then later resumed from the point of interruption
  - Node level reorganization for online table REORG can be resumed from the point where it was interrupted or paused instead of ALL nodes being restarted
  - REORG will reorganize table data objects only and not indexes, long fields or LOBs
  - Optionally, online REORG will allow a table to be truncated (i.e. free-up reclaimed table space) at the end of online processing
  - Done fully "in-place", works on regular tables
    - no requirement for large temp space



VACATE PAGE RANGE: MOVE & CLEAN to make space
free space
TIME
FILL PAGE RANGE: MOVE & CLEAN to fill space

Online Index Maintenance

- New REORG INDEX command
- Online index maintenance
  - Full R/W access to tables during index creation
  - Full R/W access to table & index during index reorg
- Mechanism
  - Ghost (or shadow) index created during normal transaction processing
  - When index creation is complete, old index is swapped out for the new, or the additional index is made available to applications and users
  - Sufficient temporary space needs to be available to build or reorganize the new index

Online Load

- Purpose of online load is to:
  - remove the concurrent tablespace access restrictions while still preventing full concurrent access to the table being loaded
  - allow read access (but no updates) to the existing portion of a table during a load insert (append) operation
- Two modes of operation.
  - Tablespace online, but table off-line
  - Tablespace online, table online for read
- Improved access to data for customers while the data is being loaded
- Additional Features
  - BCP file loader (IMPORT and LOAD)
    - Sybase export format
    - Migrate tables from Sybase directly into DB2
  - Autoloader Enhancements
    - Asynchronous processing of multiple files

Incremental AST Maintenance

- ASTs refreshed incrementally
  - Previously, each AST rebuilt entirely

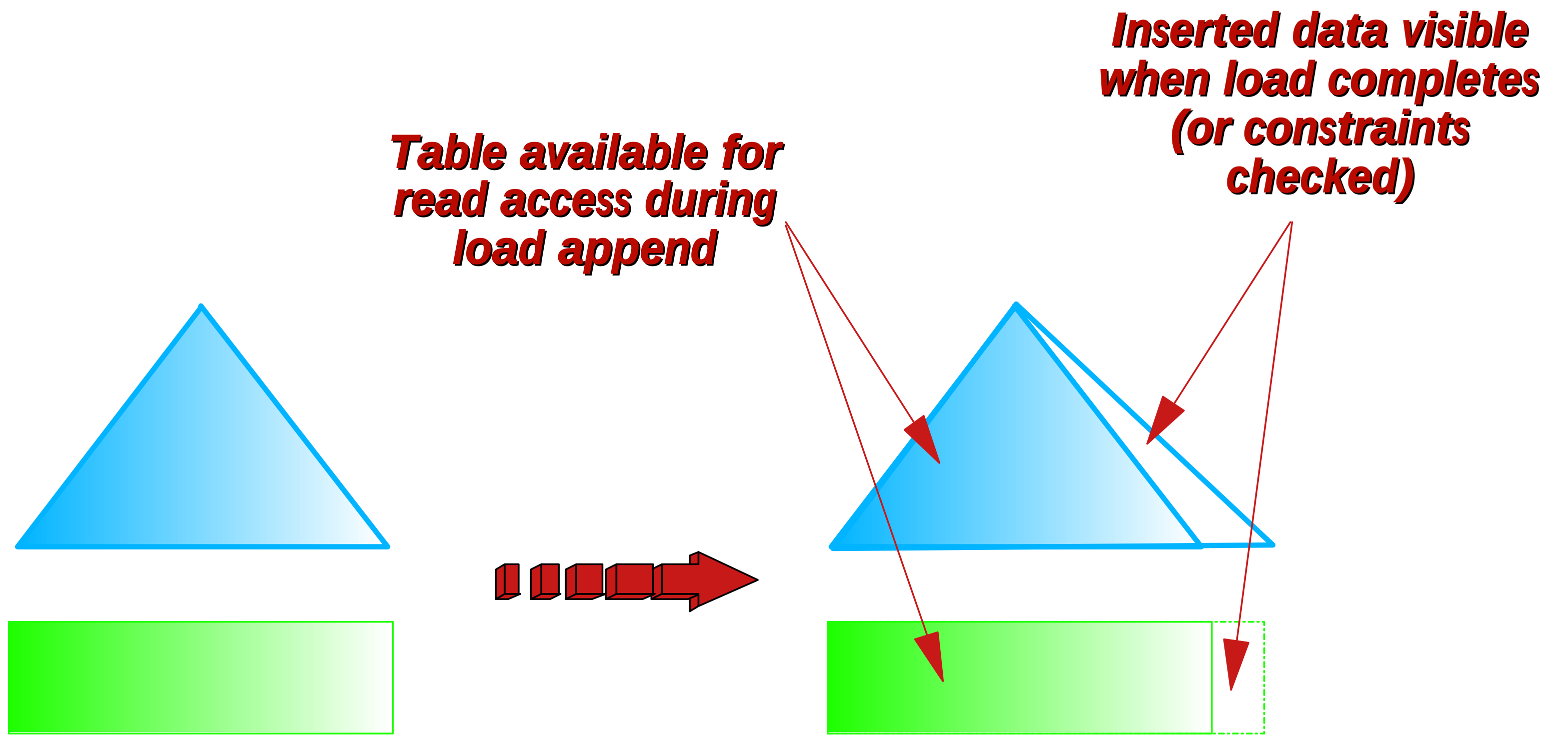


Table available for read access during load append
Inserted data visible when load completes (or constraints checked)

DB2 UDB for UNIX, Linux, and Intel Platforms

SQL Enhancements

DB2 Worldwide Presales Support
IBM Toronto Laboratory

MERGE SQL

- Common database design:
  - master table contains existing or current knowledge of a domain (for example a parts table)
  - transaction table containing a set of changes to be applied to the master table
  - transaction table can contain:
    - updates to objects existing in the master table a
    - new objects that should be inserted into the master table
- To apply the changes from the transaction table to the master table requires two separate operations:
  - an UPDATE operation for those rows already existing in the master table
  - an INSERT operation for those rows that do not exist in the master table
- New single SQL statement that makes processing of these separate operations more efficient as well as easier for the user to specify

```
MERGE INTO account AS a
USING (SELECT id, sum(balance) sum_balance FROM transaction
       GROUP BY id) AS t
ON a.id = t.id
WHEN MATCHED THEN
   UPDATE SET
      balance = a.balance + t.sum_balance
WHEN NOT MATCHED THEN
   INSERT (id, balance) =
          (t.id, t.sum_balance);
```

INSERT through UNION ALL View

- DB2 V7.1 supports UPDATE and DELETE on views containing UNION ALL in their bodies, but not INSERT
- Feature is needed for mainly two reasons:
  - ► Hierarchical Storage Management (HSM)
    - a view ranges over a partitioned table hierarchy of which parts (e.g. last years data) are stored on tertiary storage
  - ► Overcoming table size limits
    - use partitioned views is to overcome table size limits or facilitate data maintenance

```
CREATE TABLE Q1(order DATE, item VARCHAR(10), CHECK (MONTH(order) BETWEEN 1 AND 3);
CREATE TABLE Q2(order DATE, item VARCHAR(10), CHECK (MONTH(order) BETWEEN 4 AND 6);
CREATE TABLE Q3(order DATE, item VARCHAR(10), CHECK (MONTH(order) BETWEEN 7 AND 9);
CREATE TABLE Q4(order DATE, item VARCHAR(10), CHECK (MONTH(order) BETWEEN 10 AND 12);

CREATE VIEW V(order, item)
AS SELECT * FROM Q1
   UNION ALL
   SELECT * FROM Q2
   UNION ALL
   SELECT * FROM Q3
   UNION ALL
   SELECT * FROM Q4;

INSERT INTO V VALUES('2000-01-06', 'Shoes'), ('2000-06-17', 'Socks');
```

Indentity/Sequences in EEE

- Same support as is available today on DB2 V7 EE
  - CREATE SEQUENCE EMPNO AS ....
- In a partitioned environment
  - each node will maintain a local cache (of default size 20, unless otherwise specified by the user)
  - when a cache gets empty, it is refilled from a central location on the catalog node
  - unless the cache being refilled is on the catalog node (in which case a local request is made), the rest of the nodes will need to generate remote requests to refill their cache(s)
    - larger caches can reduce this overhead
  - Identity columns are allowed to form part of the partitioning key

Instead of Triggers

- Instead of triggers provide an extension to the updatability of views.
  - In DB2 only specific kinds of views are deletable, updatable, or insertable
  - Using an instead of trigger the requested update operation against the view gets replaced by the trigger logic, which performs the operation on behalf of the view
  - happens transparently to the application which believes all operations are performed against the view.

```
CREATE VIEW EMPV(EMPNO, FIRSTNME, MIDINIT, LASTNAME, PHONENO, HIREDATE, DEPTNAME)
AS SELECT EMPNO, FIRSTNME, MIDINIT, LASTNAME, PHONENO, HIREDATE, DEPTNAME
   FROM EMPLOYEE, DEPARTMENT WHERE EMPLOYEE.WORKDEPT = DEPARTMENT.DEPTNO

CREATE TRIGGER EMPV_INSERT INSTEAD OF INSERT ON EMPV
REFERENCING NEW AS NEWEMP DEFAULTS NULL FOR EACH ROW MODE DB2SQL
INSERT INTO EMPLOYEE (EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE)
        VALUES(EMPNO, FIRSTNME, MIDINIT, LASTNAME,
               COALESCE((SELECT DEPTNO FROM DEPARTMENT AS D WHERE D.DEPTNAME = NEWEMP.DEPTNAME),
                        RAISE_ERROR('70001', 'Unknown department name')),
               PHONENO, HIREDATE)

CREATE TRIGGER EMPV_UPDATE INSTEAD OF UPDATE ON EMPV
REFERENCING NEW AS NEWEMP OLD AS OLDEMP DEFAULTS NULL FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
 VALUES(CASE WHEN NEWEMP.EMPNO = OLDEMP.EMPNO THEN 0
             ELSE RAISE_ERROR('70002', 'Must not change EMPNO') END);
 UPDATE EMPLOYEE AS E
   SET (FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE)
     = (NEWEMP.FIRSTNME, NEWEMP.MIDINIT, NEWEMP.LASTNAME,
               COALESCE((SELECT DEPTNO FROM DEPARTMENT AS D WHERE D.DEPTNAME = NEWEMP.DEPTNAME),
                        RAISE_ERROR ('70001', 'Unknown department name')),
               NEWEMP.PHONENO, NEWEMP.HIREDATE)
 WHERE NEWEMP.EMPNO = E.EMPNO;
END

CREATE TRIGGER EMPV_DELETE INSTEAD OF DELETE ON EMPV
REFERENCING OLD AS OLDEMP FOR EACH ROW MODE DB2SQL
DELETE FROM EMPLOYEE AS E WHERE E.EMPNO = OLDEMP.EMPNO
```

Informational Constraints

- Informational constraints are rules that can be used in query rewrite but are not enforced by the database manager.
- In DB2 V7, constraints are not only used for enforcement, but also exploited for performance enhancement.
  - adding more constraints may result in the overhead in reverifying the constraints on Insert/Update/Delete operations
  - the application may choose to verify the constraint itself, since verification of the data can be costly
  - Informational constraints are a better alternative
- Constraint Options
  - ENFORCED
    - The constraint is enforced by the database manager during normal operations such as insert, update, or delete.
  - NOT ENFORCED
    - The constraint is not enforced by the database manager during normal operations such as insert, update, or delete. This should only be used when the data that is stored in the table is verified to conform to the constraint by some other method than relying on the database manager
    - When this option is used DB2 may return wrong results when any data in the table violates the constraint.
  - ENABLE QUERY OPTIMIZATION
    - The constraint can be used for query optimization under appropriate circumstances.
  - DISABLE QUERY OPTIMIZATION
    - The constraint can not be used for query optimization

Informational Constraints - Technical Description

- Syntax

```
ALTER TABLE T ADD CONSTRAINT CHK1 CHECK (C1 < C2)
  [[NOT] ENFORCED] [(ENABLE|DISABLE) QUERY OPTIMIZATION]
ALTER TABLE C ADD CONSTRAINT RI1 REFERENCES P
  [[NOT] ENFORCED] [(ENABLE|DISABLE) QUERY OPTIMIZATION]

ALTER TABLE T ALTER CHECK CHK1
  [[NOT] ENFORCED] [(ENABLE|DISABLE) QUERY OPTIMIZATION]
ALTER TABLE C ALTER FOREIGN KEY RI1
  [[NOT] ENFORCED] [(ENABLE|DISABLE) QUERY OPTIMIZATION]
```

- Scenarios
  - UNION ALL branch elimination
  - RI Join elimination
  - Rewrite outer to inner joins
  - Better cardinality estimation
- Example

```
CREATE TABLE H1(MONTH INT, DATA INT);
CREATE TABLE H2 LIKE H1;
ALTER TABLE H1
  ADD CONSTRAINT CHK1 CHECK (MONTH BETWEEN 1 AND 6) NOT ENFORCED;
ALTER TABLE H2
  ADD CONSTRAINT CHK2 CHECK (MONTH BETWEEN 7 AND 12) NOT ENFORCED;

CREATE VIEW FY AS SELECT * FROM H1 UNION ALL SELECT * FROM H2;

SELECT * FROM FY WHERE MONTH IN(1, 2, 3);
```

ORDER BY and FETCH FIRST in Subqueries

- Pushes ORDER BY and FETCH FIRST clauses from cursors into subqueries
  - Allowed in inline SQL PL FOR statement (SQL functions, triggers, ...)
  - Pull up ordering from nested query using ORDER BY ORDER OF
- Syntax

```
SELECT ... FROM ... WHERE ...
ORDER BY <EXPR1>, ..., <EXPRn> FETCH FIRST <N> ROWS ONLY

SELECT ... FROM (...) AS T ... ORDER BY ORDER OF T

SET V_NAME = (SELECT NAME FROM EMPLOYEE FETCH FIRST ROW ONLY);
```

- Restrictions
  - Cannot use ORDER BY and FETCH FIRST in outer fullselect of views and SQL table functions
  - Can not use ORDER OF based on unordered derived table i.e.
    - subquery without ORDER BY
    - table, nickname, view
    - common table expressions
    - table function

TO_DATE() and TO_CHAR functions

- TO_CHAR(<timestamp>, <format-string>)
  - Conversion of timestamp to char based on format string
  - Example:

```
        VALUES TO_CHAR(CURRENT TIMESTAMP, 'YYYY-MM-DD HH24:MI:SS');
         => '2002-02-28 23:30:23'
```

  - Restriction: Only supported format in V8 GA:

```
         'YYYY-MM-DD HH24:MI:SS'
```

- TO_DATE(<string-expression>, <format-string>)
  - Conversion of string expression into timestamp based on format string
  - Example:

```
        VALUES TO_DATE('2002-02-28 23:30:23', 'YYYY-MM-DD HH24:MI:SS');
```

  - Restriction: Only supported format in V8 GA:

```
         'YYYY-MM-DD HH24:MI:SS'
```

CALL Statement

- Replaces CALL api with an SQL statement
  - Can be dynamically prepared
    - "dynamic dispatch"
  - Can take expressions including scalar subqueries for arguments
  - Records dependencies
  - Robust support for procedure resolution, CLP
- Example

```
db2 -td%
CREATE SEQUENCE S1%
CREATE PROCEDURE P1(IN nv INT, IN arg1 VARCHAR(10), INOUT arg2 INT,
                    OUT result VARCHAR(40))
BEGIN
  SET result = CHAR(nv) || arg1 || CHAR(arg2);
  SET arg2 = arg2 + 1;
  RETURN 0;
END%
CALL P1(NEXTVAL FOR S1, 'John' || ' DOE', 37, ?)%
```

- Rules
  - IN arguments must be assignable, follow storage assignment rules
  - OUT arguments must be single hostvar or parameter marker, follow retrieval assignment rules
  - INOUT argument must comply with IN and OUT rules
  - CALL is NOT supported in triggers and SQL Functions in V8 GA
- Notes
  - 90 Parameter limit for SQL Procedures has been lifted

Snapshot table functions

- SNAPSHOT_AGENT(<dbname>, <partition>)
  - Returns agent information from application snapshot

```
        SELECT AGENT_ID FROM TABLE(SNAPSHOT_AGENT('MYDB', 0)) AS S;
```

- SNAPSHOT_APPL(<dbname>,<partition>)
  - Returns general information from application snapshot

```
        SELECT APPLNAME FROM TABLE(SNAPSHOT_APPL('MYDB', 0)) AS S;
```

- SNAPSHOT_BP(<dbname>, <partition>)
  - Returns bufferpool information

```
        SELECT DIRECT_READS FROM TABLE(SNAPSHOT_BP('MYDB', 0)) AS S;
```

- SNAPSHOT_CONTAINER(<dbname>, <partition>)
  - Returns container configuration information from a tablespace snapshot

```
        SELECT TABLESPACE_ID, CONTAINER_NAME
          FROM TABLE(SNAPSHOT_CONTAINER('MYDB', 1)) AS S;
```

- SNAPSHOT_DATABASE(<dbname>, <partition>)
  - Returns database snapshot

```
        SELECT DB_STATUS FROM TABLE('MYDB', 0)) AS S;
```

- SNAPSHOT_DBM(<partition>)
  - Returns DB Manager snapshot

```
        SELECT DB2START_TIME FROM TABLE('MYDB', 0)) AS S;
```

Snapshot table functions (cont'd)

- SNAPSHOT_DYN_SQL(<dbname>, <partition>)
  - Returns snapshot of dynamic statement cache (replaces SQLCACHE_SNAPSHOT)

```
SELECT STMT_TEXT FROM TABLE(SNAPSHOT_DYN_SQL('MYDB', -2)) AS S;
```

- SNAPSHOT_FCMNODE(<partition>)
  - Returns information on Fast Communication Manager

```
SELECT TOTAL_BUFFERS_SENT
  FROM TABLE(SNAPSHOT_FCMNODE('MYDB', -2)) AS S;
```

- SNAPSHOT_LOCK(<dbname>, <partition>)
  - Returns snapshot of locks

```
SELECT TABLE_NAME, LOCK_ESCALATION
  FROM TABLE(SNAPSHOT_LOCK('MYDB', -2)) AS S;
```

- SNAPSHOT_LOCKWAIT(<dbname>, <partition>)
  - Returns lock wait information from application snapshot

```
SELECT TABLE_NAME, LOCK_WAIT_START_TIME
  FROM TABLE(SNAPSHOT_LOCKWAIT('MYDB', -2)) AS S;
```

- SNAPSHOT_STATEMENT(<dbname>, <partition>)
  - Returns statement information from application snapshot

```
SELECT STMT_TEXT FROM TABLE(SNAPSHOT_STATEMENT('MYDB', -2)) AS S;
```

Snapshot table functions (cont'd)

- SNAPSHOT_SUBSECT(<dbname>, <partition>)
  - Returns information about subsections of access plans

```
SELECT STMT_TEXT, SS_EXEC_TIME FROM
 TABLE(SNAPSHOT_SUBSECT('MYDB',-2)) AS S;
```

- SNAPSHOT_TABLE(<dbname>,<partition>)
  - Returns activity information from a table snapshot

```
SELECT TABLE_NAME, ROWS_READ, ROWS_WRITTEN, PAGE_REORGS
 FROM TABLE(SNAPSHOT_TABLE('MYDB', -2)) AS S;
```

- SNAPSHOT_TBS(<dbname>, <partition>)
  - Returns activity information from a tablespace snapshot

```
SELECT TABLESPACE_NAME, DIRECT_READ_TIME
  FROM TABLE(SNAPSHOT_TBS('MYDB', -2)) AS S;
```

- SNAPSHOT_TBS_CFG(<dbname>, <partition>)
  - Returns configuration information from tablespace snapshot

```
SELECT TABLESPACE_ID, BUFFERPOOL_ID
  FROM TABLE(SNAPSHOT_TBS_CFG('MYDB', -2)) AS S;
```

Additional Functions

- ROUND, FLOOR, CEILING (CEIL), TRUNC
  - modified to return DECIMAL values instead of FLOAT
- ATANH, COSH, SINH and TANH
- ACOS, ASIN, ATAN, ATAN2, COS, COT, SIN and TAN
- RENAME INDEX
- SQL in User-defined Functions
  - allow inserts, updates and deletes in User-defined functions

DB2 UDB for UNIX, Linux, and Intel Platforms

Database Administration Enhancements

DB2 Worldwide Presales Support
IBM Toronto Laboratory

Database Maintenance Mode

- Allows administrators to force all users off the instance/database and put it into a quiesced mode.
  - In this mode only users with authority in this restricted mode are allowed to attach/connect to the instance/database.
  - SYSADM,SYSMAINT,SYSCTRL will always have access to instance while it is in quiesced
  - SYSADM will always have access to database while it is in quiesced.
- Using this command allows for exclusive access to the instance/database without having to force all users off the instance/database and then trying to stop users from attaching/connecting from outside the database engine
- When completed the administrator can then unquiesce the database and again allow other users to connect to the database
- Commands
  - db2start admin mode
  - db2 quiesce <defer | immediate > <force connections> <instance instance_name> for < user user_id | group group_id>
  - db2 unquiesce <instance instance_name>

Online Database Checking Tool

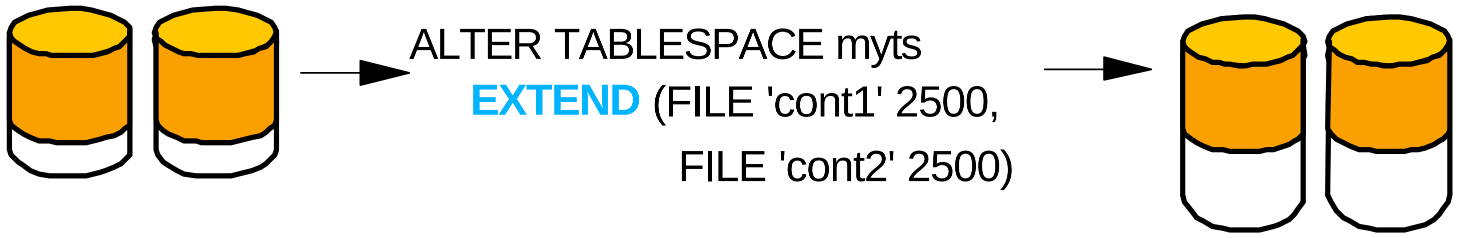
- DB2DART (Database Analysis and Repair Tool)
  - Inspect database for architectural integrity
    - The inspection checks the structures of table objects and structures of tablespaces are valid
  - Scope
    - in a single-node system, the scope is that single node only
    - In a multi-node system, it is the collection of all logical nodes defined in the node configuration file, db2nodes.cfg
- Objects
  - Databases, Tablespaces, Tables, Schemas can be specified
  - inspect check processing on whole databases will process all the objects of a table when the parent data object is found, including indexes, long fields, and LOB objects that could be located in other tablespaces
  - inspect processing will access database objects using isolation level uncommitted read
  - inspect check processing will write out unformatted inspection data results to the results file specified
    - After check processing completes, to see inspection details, the inspection result data will require to be formatted out with the utility db2inspf

Logging Enhancements

- Logging scalability
  - larger log size (256GB)
  - "Infinite" log space
    - In-flight transactions not limited by the size of active logs or even total disk space in the active log path
  - Enhanced log bandwidth
    - one log process used to write log records
    - additional process for undo and recovery work
- Logfile mirroring
  - NEWLOGPATH2 will be a db configuration parameter and not a registry variable
  - NEWLOGPATH2 should be set to the fully qualified pathname (similar to NEWLOGPATH)
    - another DB CFG parameter, LOGPATH2 will be added that can be used to query the secondary log path that is currently being used by the database (similar to LOGPATH).
  - BLOCK_ON_LOG_DISK_FULL new database configuration parameter will be use to replace the DB2_BLOCK_ON_LOG_DISK_FULL registry variable
    - allow user to specify that DB2 should not fail running applications on disk full condition from the active log path.
    - allows different databases under the same instance to have different behavior
    - allow user to change the behavior without having to do db2stop or even disconnect all application

Database Container Operations

- Three new container operations are being introduced which are considered extremely important to storage management within DB2
  - Dropping existing containers from a DMS tablespace
  - Reducing the size of existing containers in a DMS tablespace in addition to being able to extend them (V7.2 feature)
  - Adding new containers to a DMS tablespace such that a rebalance does not occur



```
ALTER TABLESPACE myts
    EXTEND (FILE 'cont1' 2500,
                  FILE 'cont2' 2500)
```

Multi-fixpak Install for UNIX

- Customer requirement that DB2 can support multiple Fixpaks coexistence on the same machine for a number of reasons:
  - They may have production running off a particular level of code, and do not want to switch to a Fixpak level until it can be thoroughly tested
  - Different teams from a customer may require different fixes
    - may not wish to switch to a different level once they have started using a level of code that is suited to their needs
- Current implementation requires multiple UNIX workstations to support running more than one level of DB2 (at the same version)
- Enhanced Fixpak structure
  - Regular Fixpak
    - will install on top of the existing code, and behave exactly as fixpaks current do
    - Customers who are satisfied with current behavior will just use this
  - Alternate Fixpak
    - Fixpak is similar to a fully installable image
    - same level of code as the regular Fixpak
    - utilities will be provided to install this as a "Fixpak" - but it is command line only
    - not a licensed version - though it can be independently installed without the GA version

Throttle Utilities

- Utility functions (such as Backup and Rebalance) may overconsume valuable system resources during production hours
  - potentially increasing response time and negatively impacting DB2's overall performance.
- Throttling is a technique which can be used to mediate the performance degradation caused by utility applications
  - selectively limits the resources that a utility may consume
  - degree to which the resources are limited is based on the current workload of the system
    - in a busy system, resources will be rationed
    - a utility will run “full out” on an unloaded system
    - a “throttle-enabled” utility will permit the DBA to specify whether or not a command should be run throttled
- Backup/Restore and Rebalance will have throttle capabilities
  - Reorg, Load and Runstats will be added in future fixpaks
- SET UTILITY_PRIORITY TO level x
  - session level command
  - x = 1 for minimal impact to production
  - x = 100 for full resources of the system
  - x = 0 reserved for future use

Rollforward to Localtime

- allows the user to rollforward to a PIT that is the user's local time rather than GMT time
  - makes it easier for users to rollforward to a specific point in time on their local machines, and eliminates potential user errors due to the translation of local to GMT time
- allows the user to control which log files to be rolled forward on the standby machine by allowing the user to disable the retrieval of archived logs
  - by controlling the logfiles to be rolled forward, one can ensure that the standby machine is X hours behind the production machine, to prevent the user affecting both systems.
  - if the standby system does not have access to archive (e.g. if TSM is the archive, it only allows the original machine to retrieve the files)
  - it might also be possible that while the production system is archiving a file, the standby system is retrieving the same file, and it might then get an incomplete log file.

```
-ROLLFORWARD----+-DATABASE-+---database-alias----------------->
                 '-DB-------'

>-----+--------------------------------------------------------+------->
      +-TO----isotime--+------------------+-+------------+-----+
                       '-USING LOCAL TIME-' '-NORETRIEVE-'
```

Kerberos Security for UNIX

- Feature provide Kerberos authentication support for the AIX and Solaris
- With the introduction of Kerberos support on AIX and Solaris, Kerberos authentication now becomes possible in the following client/server combinations:
  - AIX
    - to AIX, Solaris, Win2000/XP, DB2/390
  - Solaris
    - to AIX, Solaris, Win2000/XP, DB2/390
  - Win2000
    - to AIX, Solaris

DB2 UDB for UNIX, Linux, and Intel Platforms

Development Enhancements

DB2 Worldwide Presales Support
IBM Toronto Laboratory

Application Development

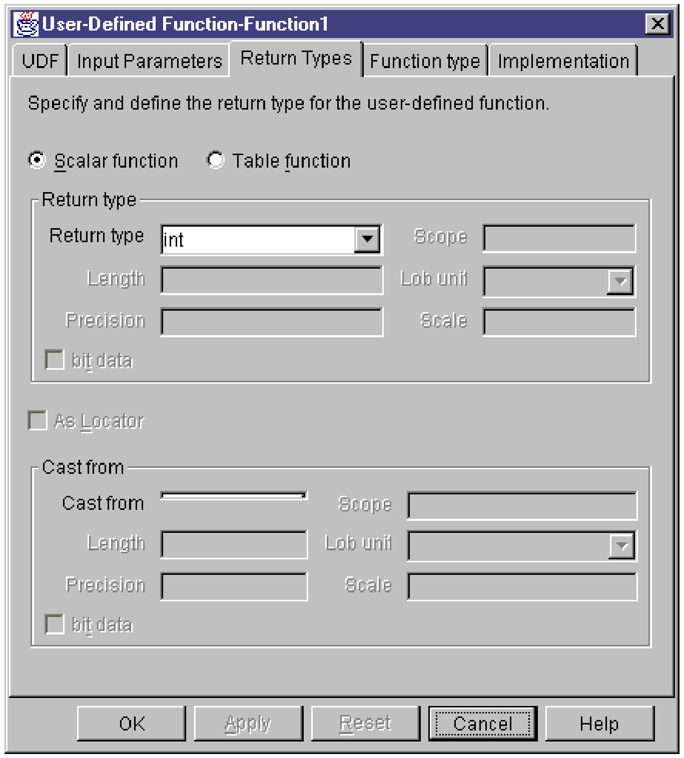
- DB2 Development Center
  - Provides tight integration between:
    - DB2 & WebSphere
    - DB2 and Visual Studio
  - SQL Assist - support for LOBs, XML, Datalinks
  - Stored Procedure builder development
    - SQL, Java2, Linux, feature exploitation
  - User Defined Functions development
    - Scalar and table SQL UDFs
    - MQSeries, XML, and OLEDB table UDFs
    - WebService table UDFs
  - Create structured data types for EJB session beans
- SQL in UDFs
- Java support
  - JDBC 3
  - Type 4 (native) JDBC driver
  - Java common client

Application Development Integration

- WebSphere integration
  - Integrated with native Project Explorer
  - Development Center dialogs and wizards
  - Dockable DB2 Server view
    - Tables, Views, and Triggers - Properties, sample content, and/or view source
    - SPs - Properties, source, test run, drop
    - UDFs - Properties, source, test run, drop
  - Development of SQL SPs
  - Development of SQL, MQ, and OLE DB UDFs
  - SQL Debugger integrated with WebSphere Debugger
- Microsoft Visual Studio add-ins:
  - Visual Basic 6.0, Visual InterDev 6.0, Visual C++ 6.0 and Visual Studio .Net
  - Launch point for various DB2 Centers / Tools
  - Dockable DB2 Development view
  - ADO client code generation
  - Integrated with Visual Source Safe
  - Development of Java and SQL SPs
  - Development of Java, SQL, MQ, and OLE DB UDFs
  - Tables, Views, and Triggers - Properties, sample content, and/or view source
  - Integrated SQL debugger

Development of User Defined Functions

- Support for entire DB2 family of servers (including AS/400 and OS/390)
- Access to UDFs defined in the database server catalog or the DC project
- Build, deploy, test, and debug the following UDFs:
  - SQL
    - Scalar UDFs, Row UDFs, Table UDFs
  - Java
    - External scalar UDFs, row UDFs, table UDFs
    - Multiple Java UDFs per class and jar
    - Helper Java class files
    - Helper Java jars
  - C
    - External scalar UDFs, row UDFs, table UDFs
    - Cross-platform C code generation and build
    - Multiple C UDFs per file and library
    - Helper C files
    - Helper libraries
  - OLE
    - External scalar UDFs
    - External row UDFs
    - External table UDFs
  - OLEDB
    - External table UDFs


User-Defined Function-Function1

Specify and define the return type for the user-defined function.

[✓ Scalar function] Table function

Return type

Return type: int

Length

Precision

Scope

Lob unit

Scale

bit data

As Locator

Cast from

Cast from

Length

Precision

Scope

Lob unit

Scale

bit data

Package Support

- Version Identifier for packages
  - allows different "versions" of the same package to exist in the database
  - production plan can be different from test plan
  - explain output will identify which version of a plan is being used
- New command to flush current active package cache
  - FLUSH PACKAGE CACHE

National Language Support

- DB2 client will be changed to conform to DRDA
  - From code page conversion viewpoint, conversion would be done by the receiver
  - Conversion tables will also be installed at the client
  - Support of Unicode V3.1
- New Code Pages
  - Arabic Code Page 425
  - Latin-1 HOST code page 1047
  - Unicode V3.1

Threading Architecture for Java SPs and UDF's

- Stored procedures are run in a fenced mode of operation by almost all of our ISVs and customers
  - this mode is fairly expensive in today's architecture due to the fact that each connection to the database which invokes a stored procedure allocates a process in which the STP and all subsequent STP invocations in the connection are run
  - The process based model is expensive not only from a startup perspective but also from a memory and operation system dispatch perspective.
  - Java stored procedures, and UDFs each get a copy of the JVM in their fenced process
- New threading support
  - Routines that are defined as threadsafe, will run in a single fenced mode process
    - one for Java routines, and one for non-Java routines
  - This will allow resource sharing of the JVM, and reduce the amount of context switching in general for users that run large numbers of fenced mode routines.

DB2 UDB for UNIX, Linux, and Intel Platforms

Control Center and Tool Enhancements

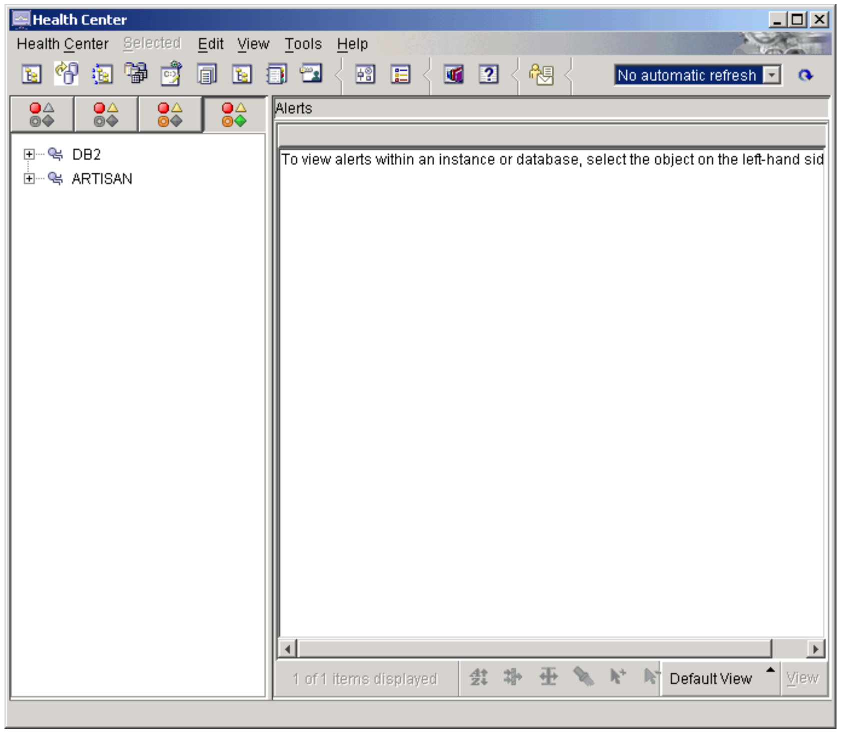
DB2 Worldwide Presales Support
IBM Toronto Laboratory

the power of ONE

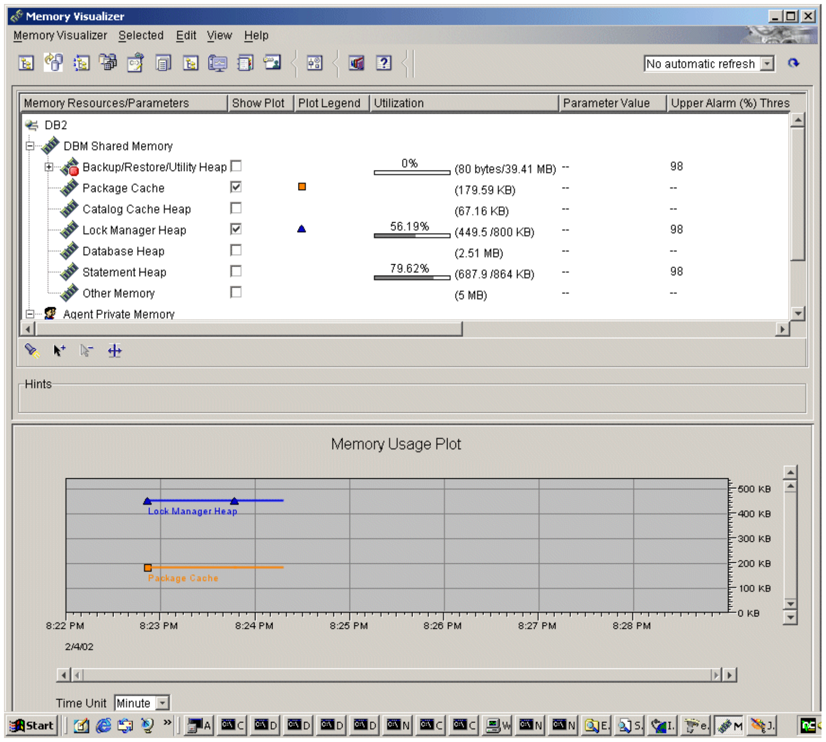
Control Center Enhancements

- Event Monitor GUI rewritten in Java
- Updated Panels
  - Script Center
  - Journal
- New Wizards
  - create tablespace wizard for ESE
  - performance configuration wizard for ESE
- New Command Support
  - Backup/Restore Enhancements
  - Multipartition control
  - Redistribute
  - Enhanced Details View
  - Add/Drop Node
  - Set Integrity
  - Privilege Support
  - Version Bind
  - DB2LOOK
  - Online Table Reorganization, Index Build, Rename Index, RUNSTATS
  - Dynamic Configuration Parameters
  - Container Operations, Log Mirroring
  - Multidimensional partitioning

Health Center/Memory Visualizer



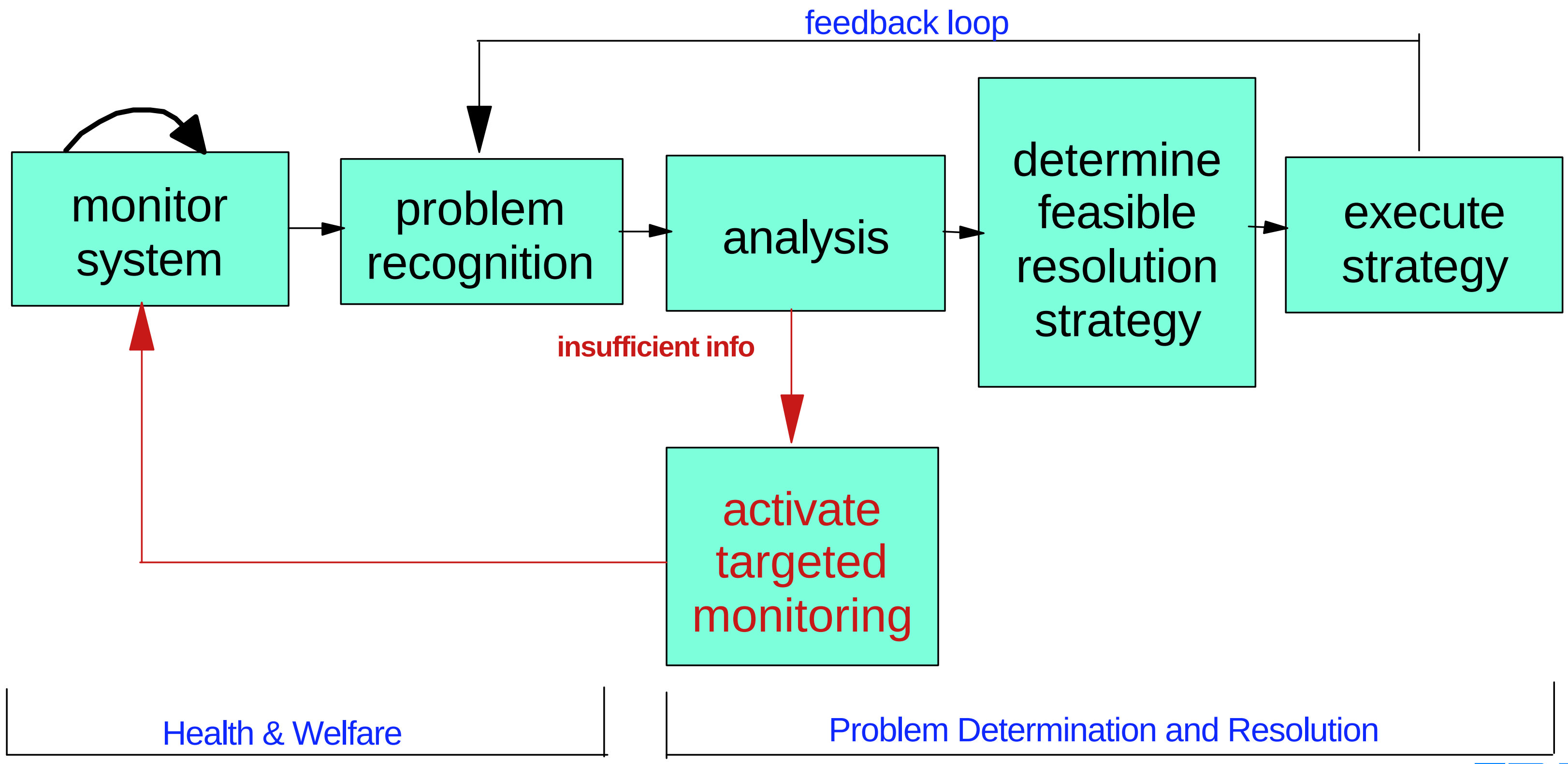
To view alerts within an instance or database, select the object on the left-hand sid



Memory Resources/Parameters	Show Plot	Plot Legend	Utilization	Parameter Value	Upper Alarm (%) Thres
DB2					
DBM Shared Memory					
Backup/Restore/Utility Heap	☐		0% (80 bytes/39.41 MB)	--	98
Package Cache	☑	🟧	(179.59 KB)	--	--
Catalog Cache Heap	☐		(67.16 KB)	--	--
Lock Manager Heap	☑	▲	56.19% (449.5 /800 KB)	--	98
Database Heap	☐		(2.51 MB)	--	--
Statement Heap	☐		79.62% (687.9 /864 KB)	--	98
Other Memory	☐		(5 MB)	--	--
Agent Private Memory					

Management by Exception

- Automated server-side health monitoring
  - Monitors key performance and resource allocation problems that DBAs must grapple with in their daily routine
  - All alarms or warnings based on trends or problem states can be surfaced through default or specialized notifications (e.g. pager, e-mail)
- Problem Determination and Resolution modules
  - Allows DBAs to focus on an object or related set of objects
  - **Disk storage management** and **concurrency (lock contention)** have been identified as being among top administration tasks that will benefit from automation.

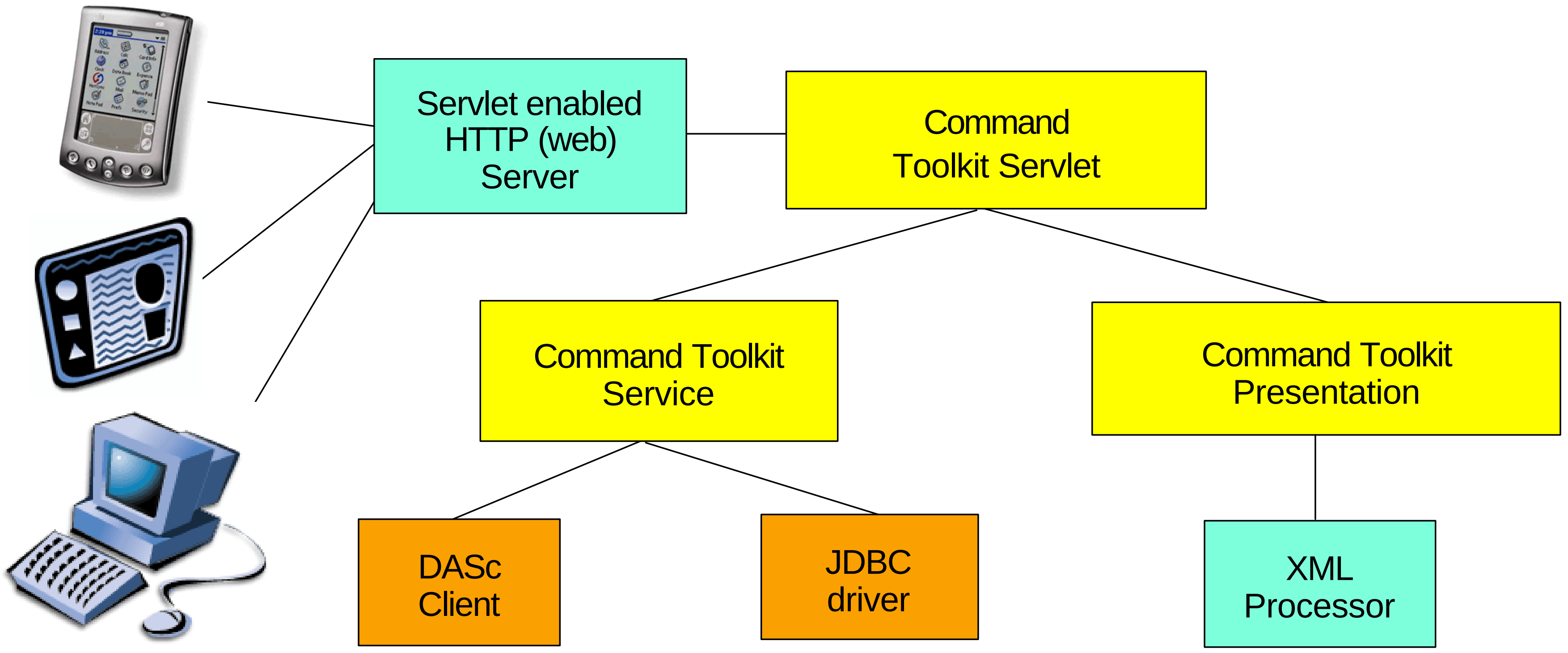


Memory Visualizer

- Part of the new DB2 Health Center
- Monitors memory usage in the data engine, including sort heaps, buffer pools, and caches
  - displays memory allocations for a particular instance, and all of its databases (manual and automatic refresh modes)
    - Instance - shared memory required for instance wide functionality (i.e. SQL compiler heap, monitor heap, EDU anchor heap, etc.)
    - Database(s) - shared memory required for database wide functionality (i.e. utility heap, database heap, sort heap, etc.)
    - Agent(s) - private memory used by an agent (i.e. sort heap, application heap, RDS compiler heap, etc.)
    - Current size, Maximum size (hard limit), Largest size (high water mark)
  - optionally permits modification to memory-related configuration parameters to adjust memory allocations
  - captures history of memory allocations on the client-side, valid for as long as tool is running
  - supports effective browsing of the hierarchical organization memory allocations
  - supports effective browsing of the history of memory allocations
  - load/save historical data for memory allocations and configuration parameter updates

Web-based Administration

- Remote execution from an HTTP client of:
  - SQL statements against a DB2 database
  - DB2 commands against a DB2 instance
  - operating system commands against a DB2 server
  - Stored Procedures through JDBC calls and DB2 administration server API



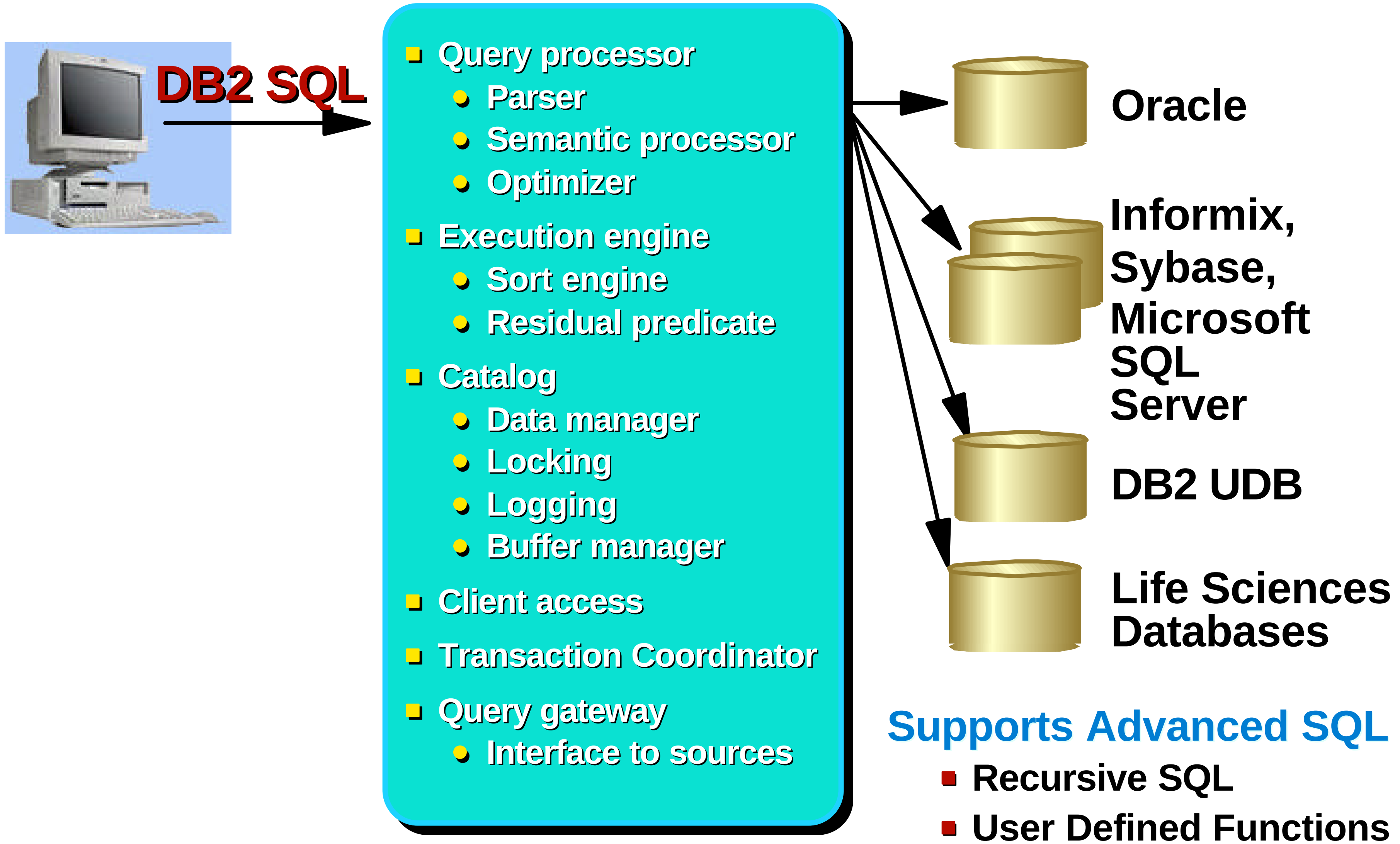
DB2 UDB for UNIX, Linux, and Intel Platforms

Additional Facilities

DB2 Worldwide Presales Support
IBM Toronto Laboratory

the power of ONE

DB2 Relational Connect



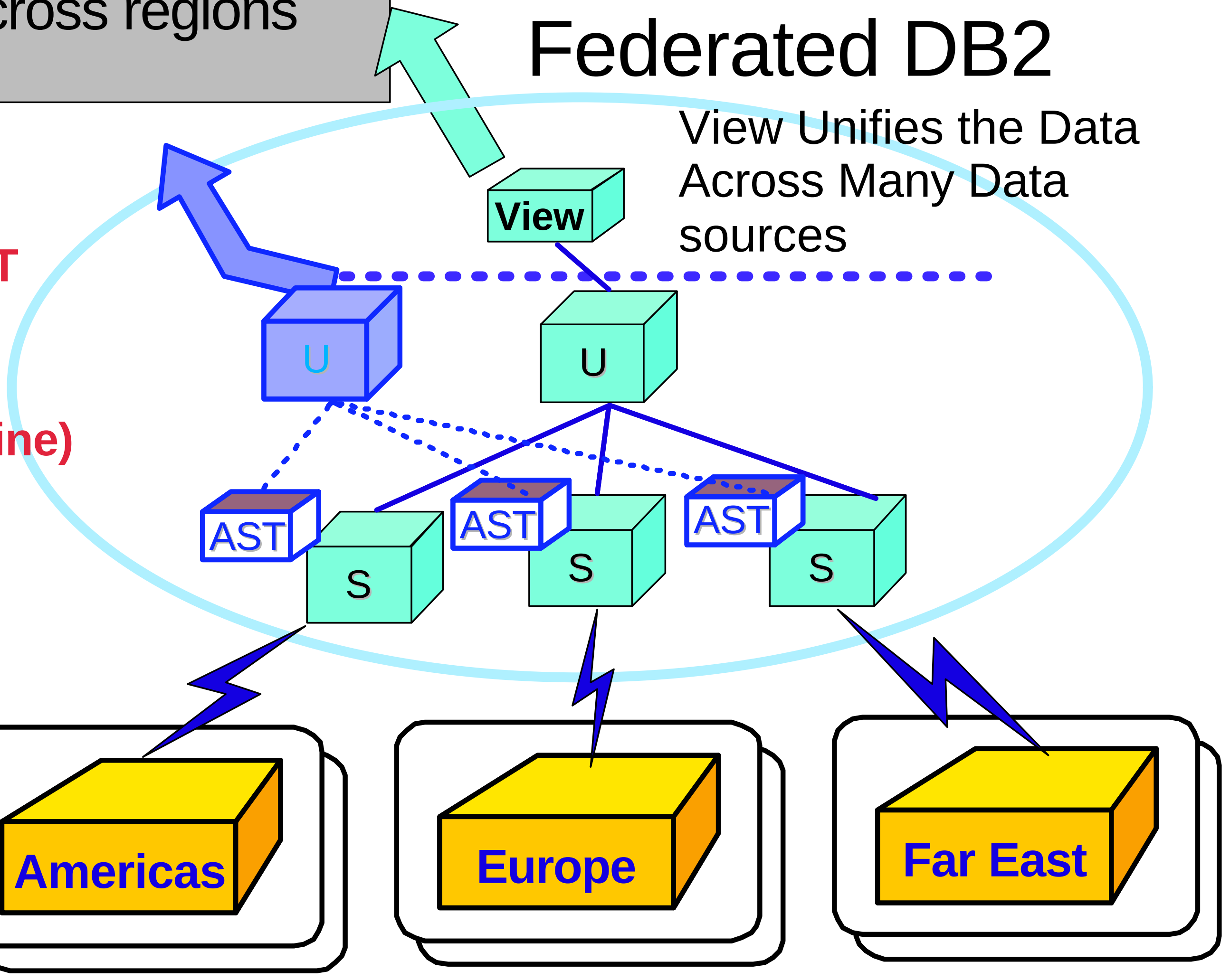
- Query processor
  - Parser
  - Semantic processor
  - Optimizer
- Execution engine
  - Sort engine
  - Residual predicate
- Catalog
  - Data manager
  - Locking
  - Logging
  - Buffer manager
- Client access
- Transaction Coordinator
- Query gateway
  - Interface to sources

Supports Advanced SQL

- Recursive SQL
- User Defined Functions
- Common Table Exp.

ASTs on Nicknames (Federated Data)

Query:
Daily Global Sales Across regions

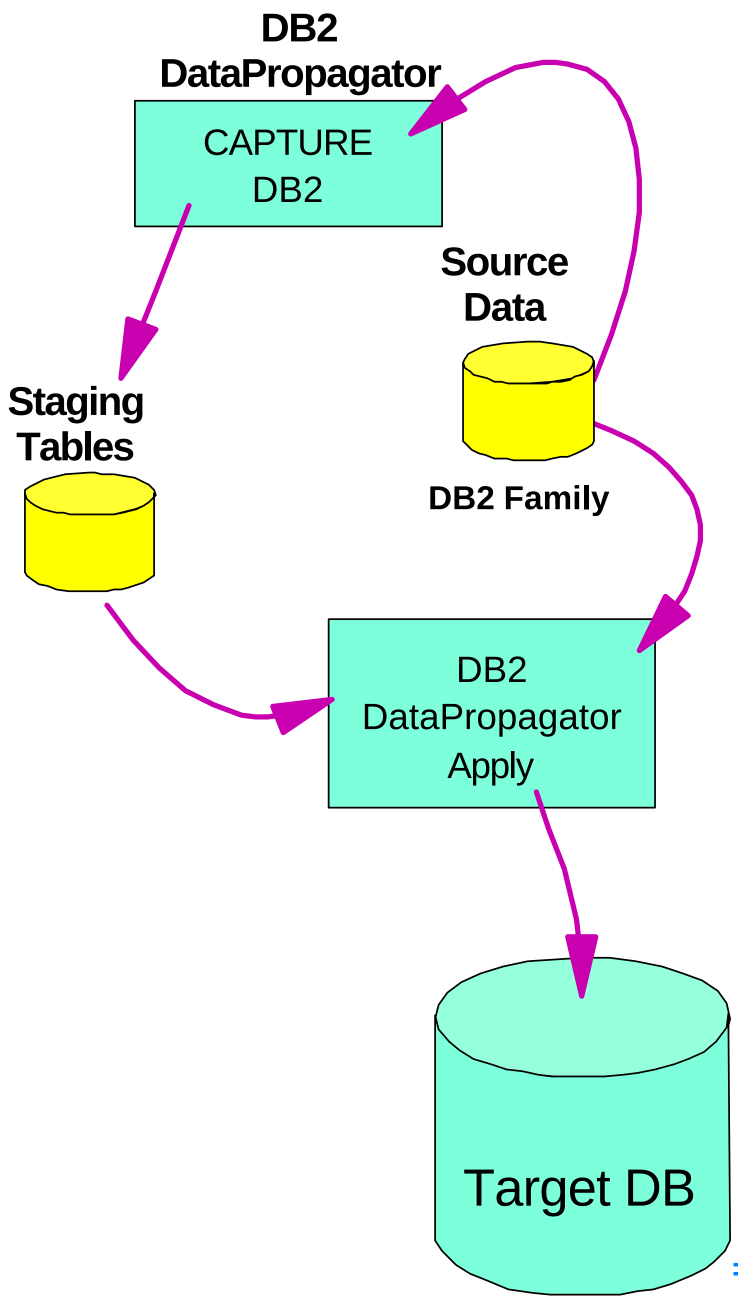


Query routed to local AST
- **Gain in performance**
- **Gain in availability**
  (Often backend is off-line)

Autonomous Source Data

DB2 Replication for Warehouse Change Data Update

- Performance
  - Multithreaded transaction based Capture
  - Multiple Capture Schemas
  - Reduced need for joins
- Usability
  - New Replication Admin features
  - Monitoring
  - Less rules, product more adaptive
  - Long name support
- Reliability
  - Emphasis on continuous operation
  - Improved restart and failure algorithms
- Serviceability
  - Dynamic Trace Facility
- Security
  - Improved password management



Spatial Extender Enhancements

- Catalog restructure
- Spatial index guidance tool
- Shape import/export improvement
- Geocoder API
- Spatial Extender migration tool
- Spatial Projection support
- Expanded spatial sample program
- Expanded spatial UDF's and methods

Information and Documentation

- Regular updates to documentation:
  - Use new documentation update facility to get latest HTML from the web
  - Entire HTML library updated between releases to reflect new and changed function and to add samples
- Enhanced documentation content:
  - New book on high availability, backup, and recovery
  - New Development Center tutorial
  - Focus throughout library on how to perform essential tasks
- Separately installable documentation CD:
  - Install complete DB2 library anywhere

DB2 UDB for UNIX, Linux, and Intel Platforms

Technology Preview

DB2 Worldwide Presales Support
IBM Toronto Laboratory