IBM

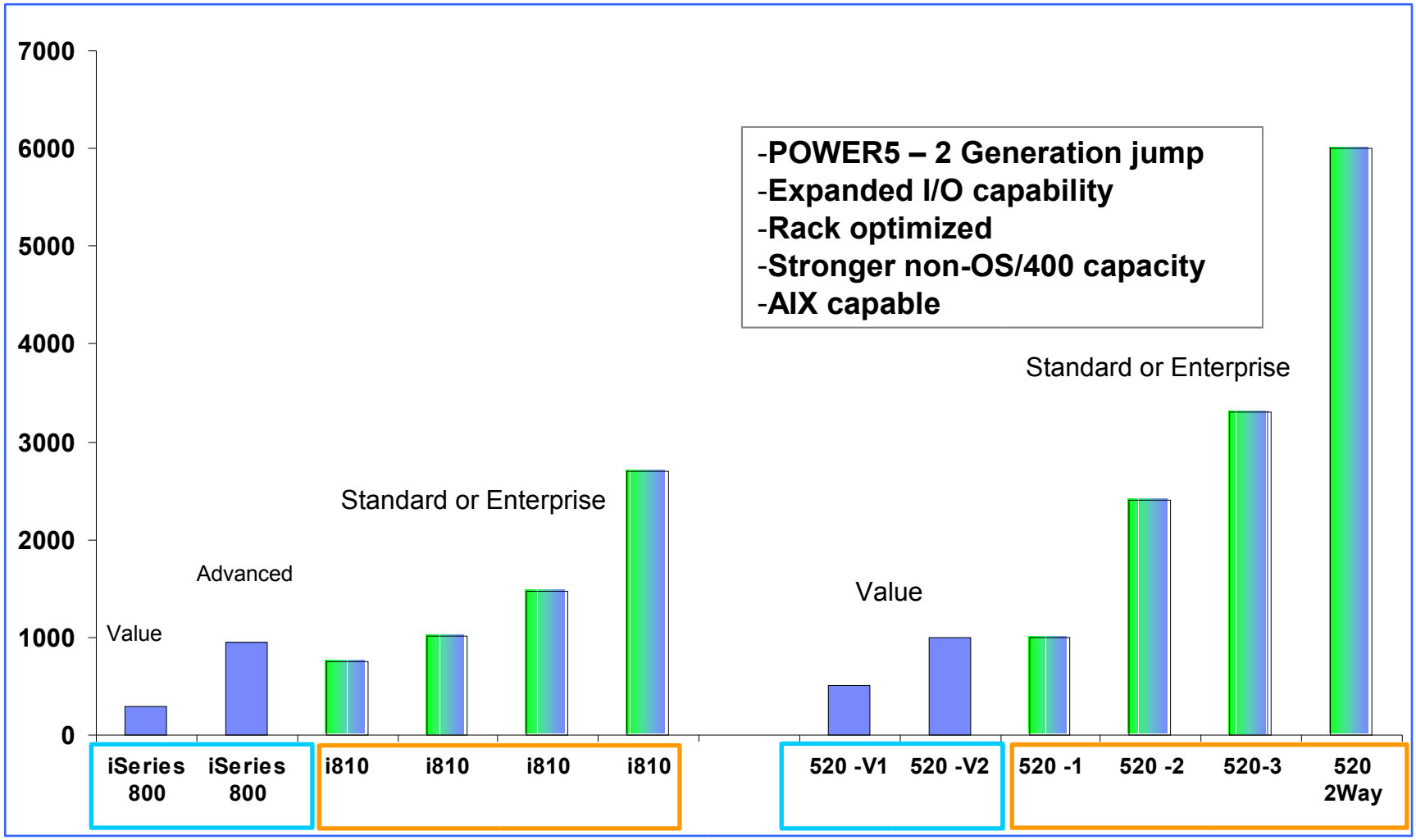
@server

iSeries. mySeries.

V5R3 Performance Updates

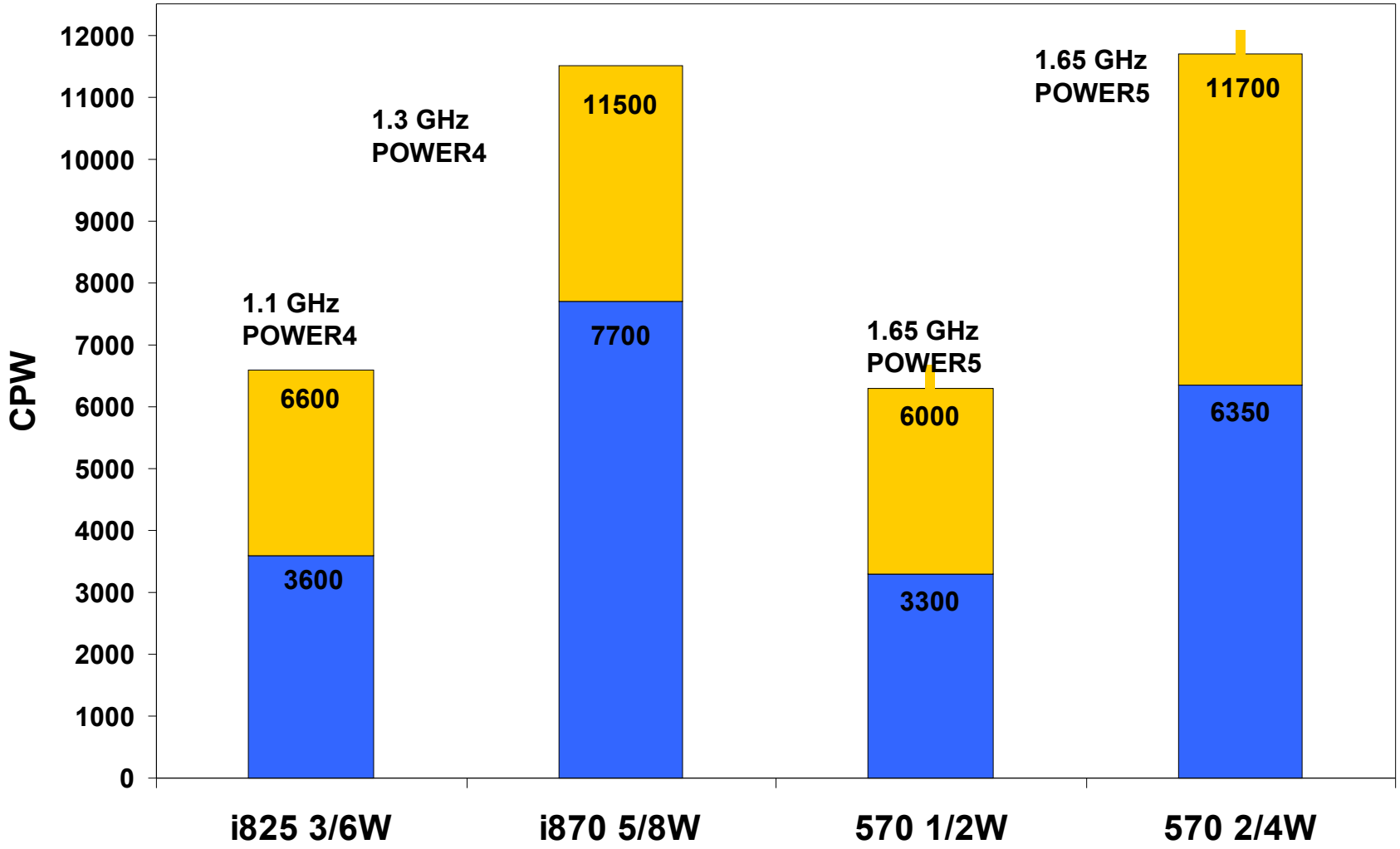
Common Belgium
June 2, 2004

Comparing 800, 820, 520 models



iSeries. mySeries.

Comparing 825, 870, 570 models (May 2004)



Performance Metrics Summary

Model-Feature	Processors	GHz	CPW Rating*	MCU* Rating	Max Main Stg GB/Disk TB	Max Disk Arms/LUNs
570-0920	2-4	1.65	6350-11700	14100-25900	64 / 38.5	546/545
570-0919	1-2	1.65	3300-6000	7300-13300	32 / 19	276/275
520-0905	2	1.65	6000	13300	32 / 19	278/277
520-0904	1	1.65	3300	7300	32 / 19	278/277
520-0903	1	1.5	2400	5500	32 / 19	278/277
520-0902	1	1.5	1000	2300	32 /19	278/277
520-0901	1	1.5	1000	2300	32 / 19	278/277
520-0900	1	1.5	500	Not recommended	32 / 19	278/277

Model-Features	Processors	GHz	CPW Rating*	MCU Rating*	Max Main Stg GB/Disk TB	Max Disk Arms/LUNs
890-0898	24-32	1.3	29300-37400	84100-108900	256 / 144.4	2047/2046
890-0897	16-24	1.3	20000-29300	57600-84100	256 / 144.4	2047/2046
890-0892	4-32	1.3	5600-37400	na-108900	192 / 144.4	2047/2046
870-0886	8-16	1.3	11500-20000	29600-57600	128 / 144.4	2047/2046
870-0891	2-16	1.3	3200-20000	na-29600	128 / 144.4	2047/2046
870-0869	5-8	1.3	7700-11500	20200-29600	64 / 144.4	2047/2046
825-0873	3-6	1.1	3600-6600	8700-17400	48 / 58	825/824
825-0890	1-6	1.1	1250-6600	na-17400	48 /58	825/824
810-0869	2	.75	2700	7900	16 / 13.8	198/197
810-0867	1	.75	1470	4200	16 / 13.8	198/197
810-0866	1	.54	1020	3100	16 / 13.8	198/197
810-0868	1	.54	750	1900	16 / 13.8	198/197
800-0865	1	.54	950	2900	8 / 4	63/62
800-0864	1	.54	300	Not recommended	8 / 4	63/62
800-0863	1	.54	300	Not recommended	8 / 4	63/62

*CPW and MCU rating values are internal IBM estimates for comparison among iSeries systems

Notes: Performance Metrics summary

This shows two tables, one for the IBM eServer i5 systems and the second table for the 800, 810, 825, 870, and 890 systems, summarizing several performance-related metrics. This includes:

- Number of processors
- GHz of each processor
- CPW ratings among iSeries systems
- MCU ratings among iSeries systems
- Maximum main storage size
- Maximum number of supported disk arms

The Domino Mail/Calendar number of users (MCU) estimates are not formally validated NotesBench benchmark numbers.

The Commercial Processing Workload (CPW) rating is based upon an internal iSeries benchmark used as a relative measure of performance among iSeries processors. The CIW (Compute Intensive Workload) available in earlier iSeries releases as useful for rating iSeries models using CPU intensive applications is no longer available.

The values shown are achieved using maximum configurations. Remember performance in customer environments may vary .

See the May 2004 Performance Capabilities Reference manual for more information – http://www.ibm.com/eserver/iseries/perfmgmt for latest information.

SAP Ratings

Model	CPW	SAPs	€ per SAP
520	1000	410	110
520	2400	986	68
810	2700	1170	79
520	3300	1311	65
825	3600-6600	1530-2840	109-138
520	6000	2451	71
870	7700*-11500	3100-4950	149-163
570	11700	4856	77
870	11700-20000	4950-8640	
890	20000-29300	8640-12690	
890	29300-37400	12690-16200	

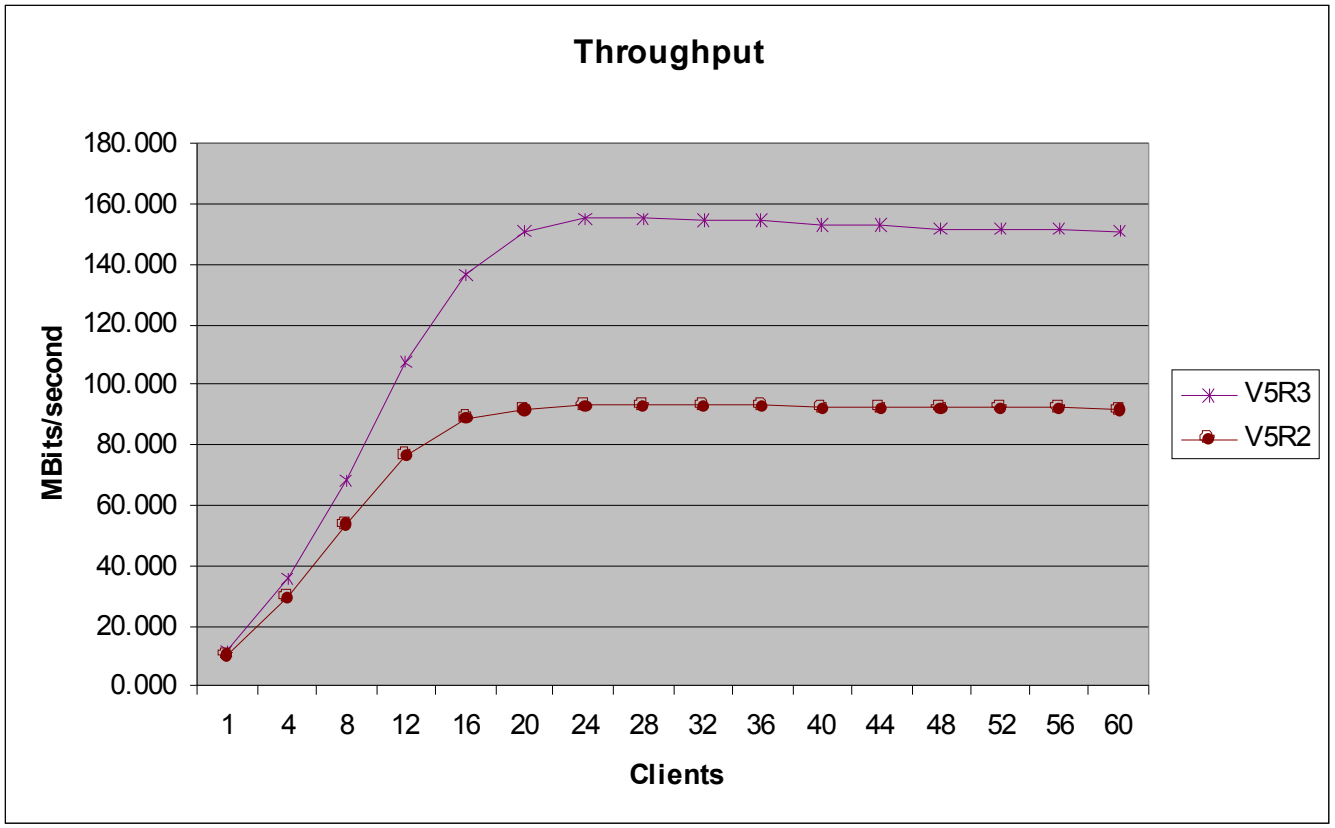
SAP R/3 Release 4.7 BASIS Release 6.40

Base system, 8 (*6) GB per processor, 3*35GB disk units, 2 LAN interfaces

iSeries NetServer File Serving

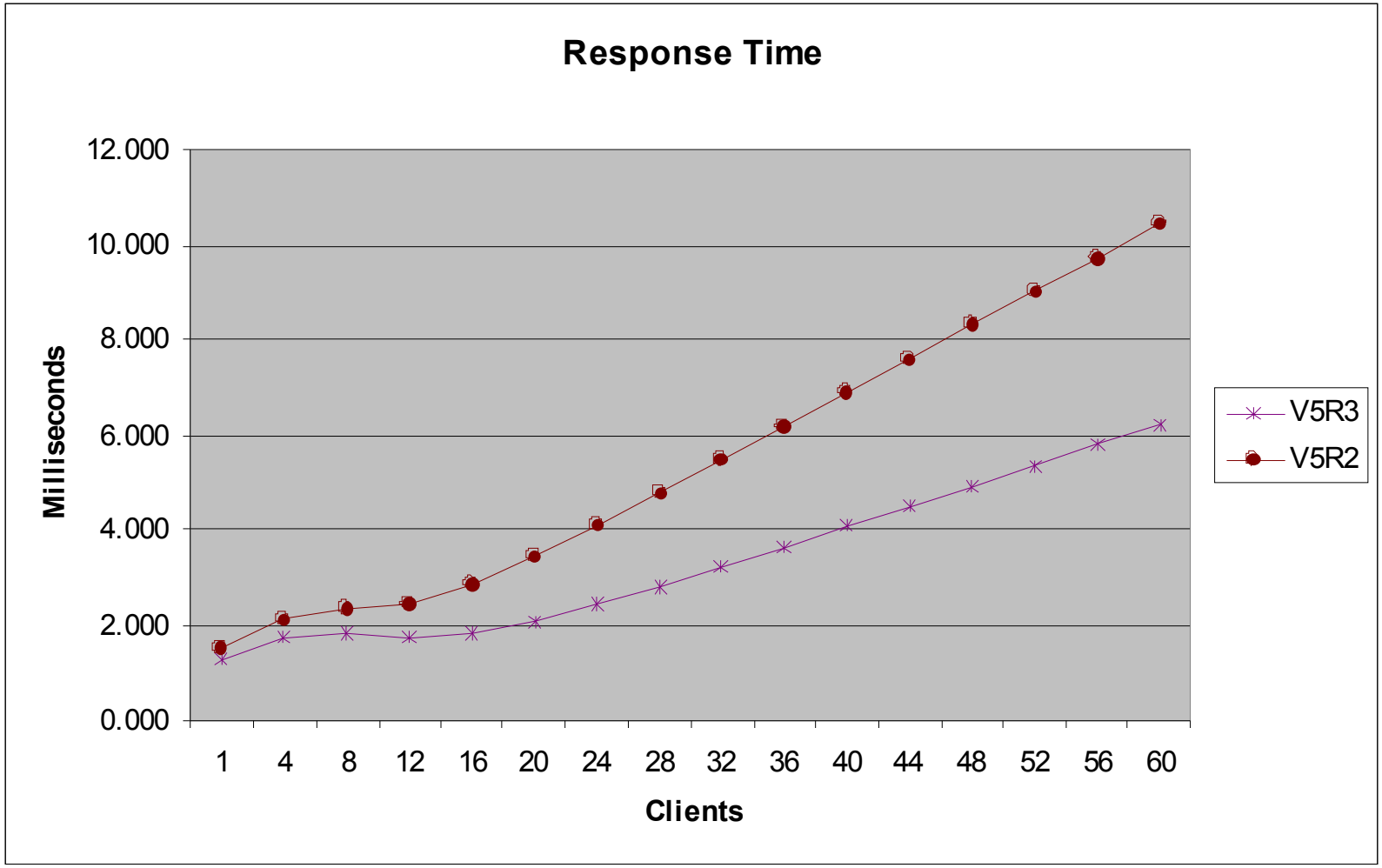
- Improvements to optimize the performance of binary file reads and writes when using the “root” (/), QOpenSys, and user-defined file systems (UDFS)

NetServer IFS throughput: V5R2, V5R3 comparisons



Megabits unit of measure used as connection is via LAN

NetServer IFS response time: V5R2, V5R3 comparisons



Notes:

iSeries Support for Windows Network Neighborhood (iSeries NetServer) supports the Server Message Block (SMB) protocol through the use of Transmission Control Protocol/Internet Protocol (TCP/IP) on iSeries. This communication allows clients to access iSeries shared directory paths and shared output queues. PC clients on the network utilize the file and print-sharing functions that are included in their operating systems. iSeries NetServer properties and the properties of iSeries NetServer file shares and print shares are configured with iSeries Navigator.

Clients can use iSeries NetServer support to install Client Access from the iSeries since the clients use function that is included in their operating system

Database Performance

- 4Q 2003 SQL Query Engine results
  - http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm
- SQL V5R3:
  - Result set caching for re-run of same queries
  - SQL Query Engine
    - Star Join enhancements
    - Constraint awareness
    - OnDemand statistics generation
  - Faster SQL Deletes
  - Faster Stored Procedure Call
  - Parallel Reorganize

Notes: Database Performance

The October, 2003 version of the Performance Capabilities Reference manual document V5R2-based performance results of a large suite of queries processed by the V5R2 SQL Query Engine.

For V5R3:

Star Join - Look Ahead Predicate Generation (SQE only)

A special type of process called look ahead predicate generation (LPG) may be used for joins. In this case, the optimizer attempts to minimize the random I/O costs of a join by pre-applying the results of the query to a large fact table. LPG will typically be used with a class of queries referred to as star join queries, however it can possibly be used with any join query.

A *star schema* is a specialized design that consists of multiple dimension tables, which describe aspects of a business, and one fact table, which contains the facts about the business. For example, if you have a mail-order business selling books, some dimension tables are customers, books, catalogs, and fiscal years. The fact table contains information about the books that are ordered from each catalog by each customer during the fiscal year.

The optimizer may modify a user query over two joined tables, for example, to firstly produce a temporary hash table based on the criteria for the first table and then probe this hash table when selecting rows from the second table. This can reduce the number of I/Os needed to complete the query.

SQE is aware of both **Referential and Check Constraints** and can rewrite queries to avoid unnecessary processing and to eliminate joins. The slide shows an example of DB2 UDB rewriting a query so that the range of values for a given column is clear to the optimizer. This means that the query itself acts as a source of table statistics which should assist the optimizer to decide on the best access method for the query.

There are some queries that the optimizer thinks will benefit from collecting the statistics inline immediately instead of having them collected in the background as these would not be available until the next time this query is run.

QAQQINI entry CACHE_RESULTS is used to specify if SQE queries can used cached result sets from previous queries.

Visual Explain in V5R3 supports the new SQE access methods.

Notes: Database Performance Updates

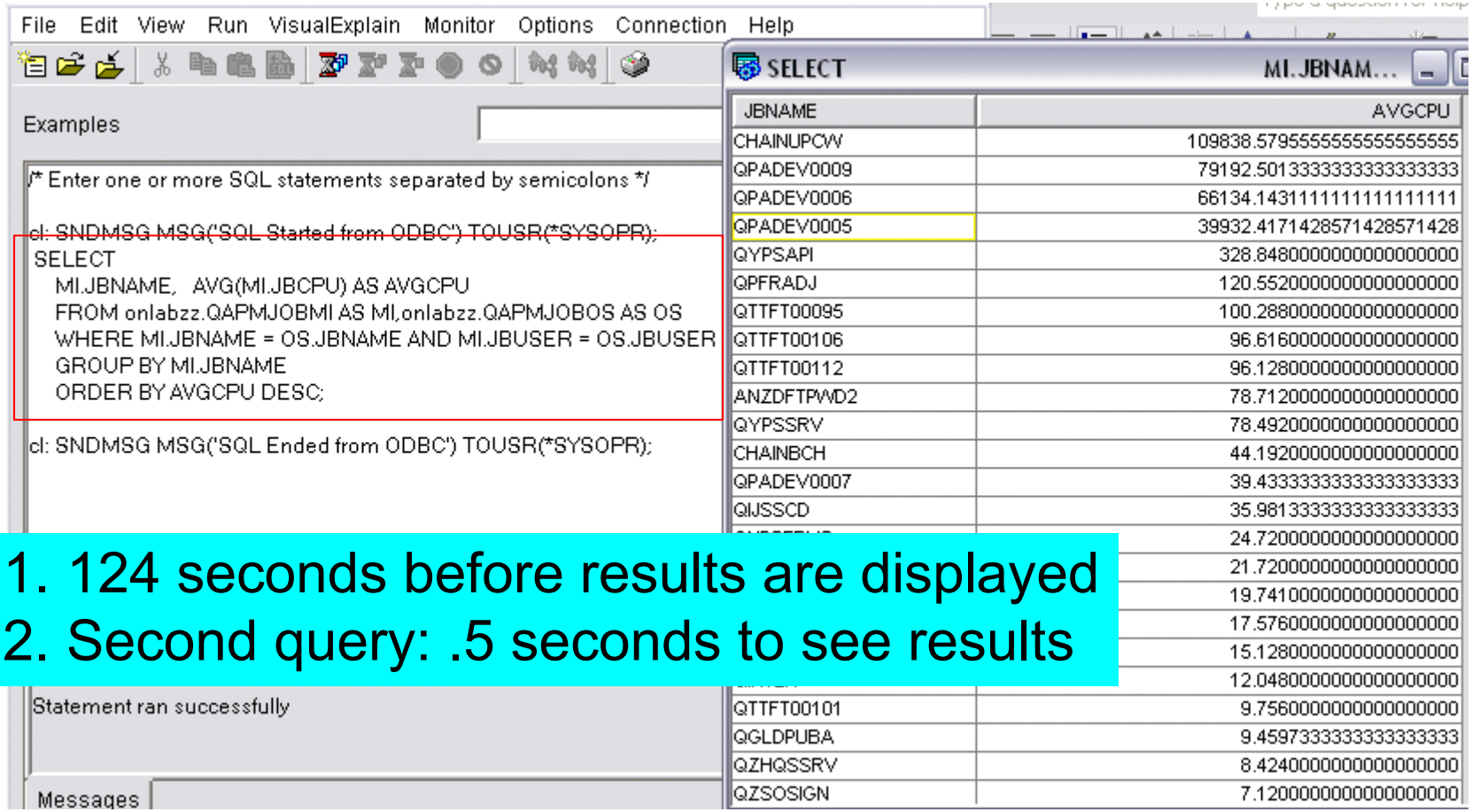
Calls to stored procedures in V5R3 will be cached. Some simple SQL statements produce C code immediately which gives better performance rather than running SQL and the number of cases where this is possible has increased in V5R3. It is necessary to recreate your procedures to take advantage of this. Both of these items should improve SQL performance. The V5R3 SQL Programming manual contains a number of hints for improving the performance of your stored procedures and functions.

An SQL DELETE statement that does not contain a WHERE clause will delete all rows of a table. In this case, the rows may be deleted using either a clear operation (if not running under commitment control) or a change file operation (if running under commitment control). If running under commitment control, the deletes can still be committed or rolled back. This implementation will be much faster than individually deleting each row, but individual journal entries for each row will not be recorded in the journal. This technique will only be used if all the following are true:

- The target table must not be a view
- A significant number (determined by the system) of rows are being deleted
- The job issuing the DELETE statement does not have an open cursor on the file
- No other job has a lock on the table
- The table does not have an active delete trigger
- The table is not the parent in a referential constraint with a CASCADE, SET NULL or SET DEFAULT delete rule
- The user issuing the DELETE statement has *OBJMGT or *OBJALTER system authority on the table in addition to the DELETE privilege
- SQL_FAST_DELETE_COUNT in QAQQINI can be used to prevent fast DELETE

You can speed up a file reorganization if you can use the parallel reorganize function. The reorganize can run in parallel if the DB2 UDB Symmetric Multiprocessing option is installed. To control the amount of resources used by the reorganize operation, you might want to change the query attributes using the CHGQRYA CL command or Change Query Attributes from iSeries Navigator.

Reusing cached results example



1. 124 seconds before results are displayed
2. Second query: .5 seconds to see results

iSeries. mySeries.

Notes: **Reusing cached results example**

In V5R3, as described later in the new QAQQINI Options slide, there is more wide-spread reuse of cached query results that can make the second through “nth” query almost instant, when the results would be exactly the same records being retrieved.

This is an iSeries Navigator Run SQL Scripts example.

Performance - New QAQQINI Options

Option	Description	Possible values
DATABASE_MONITOR_THRESHOLD	Allows only SQL statements with estimated runtime exceeding the threshold to be captured by the monitor	Integer, **2147483647 secs**
SQL_DBMON_OUTPUT	Controls the types of SQL statements collected by the monitor based on the requestor	***USER,** *ALL, *SYSTEM
SQL_STMT_COMPRESS_MAX**	Allows the user to adjust background access plan compression when using SQL packages	Integer(1-255, **2**)
IGNORE_DERIVED_INDEX	Allows SQE to process SQL statement even when an unsupported index type exists over the table(s)	***NO**, *YES
SQL_FAST_DELETE_COUNT **	Allows user to control when & how V5R3 SQL Fast Delete support is used	*NONE, *OPTIMIZE, Integer
CACHE_RESULTS **	Allows SQE queries to use cached results sets from previously run queries	***SYSTEM**, *JOB,*NONE

** Only available on V5R3, no PTFs for prior releases

Notes: New QAQQINI options

The QAQQINI options on this slide do not appear in the General Availability version of the V5R3 Database Performance and Query Optimization manual as they were late additions to the V5R3 code. They will be documented on the DB2 UDB for iSeries website in the future: http://www.iseries.ibm.com/db2.

CACHE_RESULTS

For queries involving temporary results (for example sorts and hashes), the database often saves the results across query pseudo-close/pseudo-open as long as the result set is not large, hoping to reuse the results for the next run of the query. In V5R3, the database also begins saving these temporary results even when a job is finished with them, assuming another job can later reuse the results. The caching is controlled by the database, removing results as storage usage becomes large. However this can mean an increase in the use of temporary storage in V5R3 when compared with previous releases. So the CACHE_RESULTS option provides a means to control this situation allowing cache results to be reused across the system or only within one job or not at all.

There are three other QAQQINI options that are new with V5R3 (that are documented in the manual):

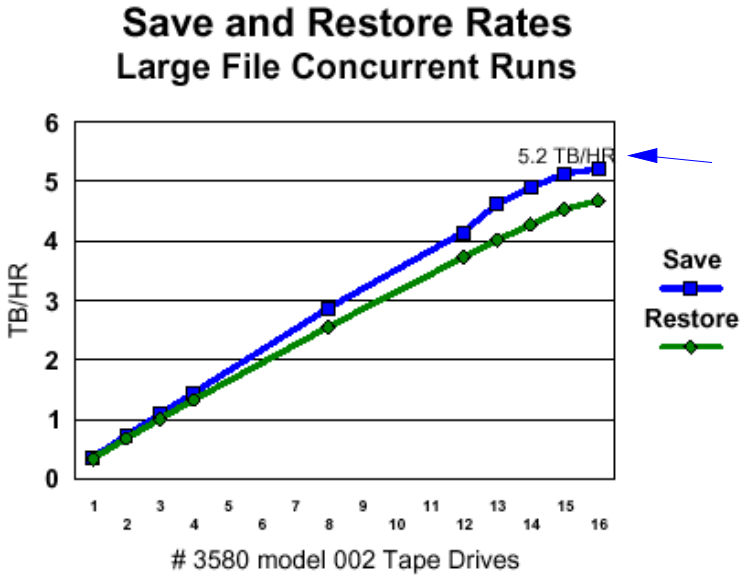
LOB_LOCATOR_THRESHOLD

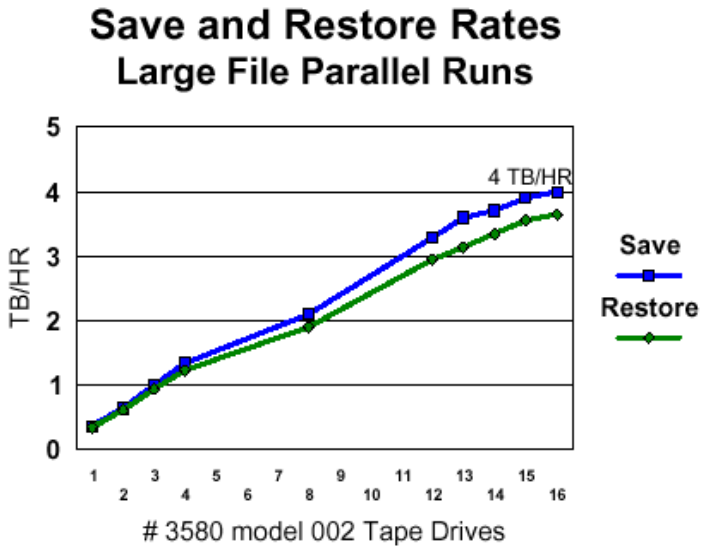
NORMALIZE_DATA

VARAIABLE_LENGTH_OPTIZATION

4Q 2003 358x Ultrium 2 Performance Results

- 3580-002 (Ultrium2) via 5704 FibreChannel adapter:
  - Recent measurements on iSeries using IBM's latest Ultrium LTO-2 tape drives combined with new PCI-X I/O and fibre channel tape controller running OS/400 V5R2 showed up to 5.2 TB/hour save/restore rates
  - Surpasses previous iSeries record of 2.7 TB/hour set in 2001





Save or restore different objects from a single library to multiple backup devices or different libraries to multiple backup devices at the same time from different jobs.

Notes: 4Q 2003 358x Ultrium 2 Performance Results

- The following workloads were designed to help evaluate the performance of single, concurrent and parallel save and restore operations for selected devices. Familiarization with these workloads can help in understanding differences in the save and restore rates .

- Database File related Workloads:
- The following workloads are designed to show some possible customer environments using database files.
- User Mix: NUMX3GB NUMX6GB NUMX12GB - The User Mix data is contained in a single library and made up of a combination of source files, database files, programs, command objects, data areas, menus, query definitions, etc. NUMX12GB contains 52900 objects.
- Source File: NSRC1GB - 96 source files with approximately 30,000 members.
- Large Database File: SR4GB, SR8GB, SR16GB, SR32GB SR64GB - The Large Database File workload is a single database file with members 4GB in size to create the database file being tested.

Notes: 4Q 2003 358x Ultrium 2 Performance Results

- The following workloads were designed to help evaluate the performance of single, concurrent and parallel Integrated File System related Workloads:
- Analysis of customer systems indicates about 1.5 to 1 compaction on the tape drives with integrated file system data. This is partly due to the fact that the OS/400 programs on the iSeries that store data in the integrated files system, do some disk management functions where they keep the IFS space cleaned up and compressed. And the fact that the objects tend to be smaller by nature, or are mail documents,
- HTML files or graphic objects that don't compact. The following workloads (1 Dir / M Obj, M Dir / M Obj, Domino, NW STG) show some possible customer integrated file system environments.
- 1 Directory Many objects 1 Dir / M Obj - This integrated file system workload consists of 111,111 stream files in a single directory where the stream files have 32K of allocated space, 24K of which is data.. Approximately 4 GB total sampling size.
- Many Directories Many objects M Dir / M Obj - This integrated file system workload consists 6 levels deep, 10 directories wide where each directory level contains 10 directories resulting in a total of 111,111 *DIRs and 111,111 stream files, where the stream files have 32K of allocated space, 24K of which is data.. Approximately 5 GB total sampling size.
- Domino Domino - This integrated file system workload consists of a single directory containing 90 mail files. Each mail file is 152 MB in size. The mail files contain mail documents with attachments. Approximately 13 GB total sampling size.
- Network Storage Space NW STG - This integrated file system workload consists of a Linux storage space of approximately 6 GB total sampling size.

Notes: 4Q 2003 358x Ultrium 2 Performance Results

The testing workloads for concurrent and parallel save and restore operations were performed on a dedicated system.

- Concurrent Saves and Restores - The ability to save or restore different objects from a single library to multiple backup devices or different libraries to multiple backup devices at the same time from different jobs. The workloads that were used for the testing were Large Database File and User Mix. For the tests multiple identical libraries were created, a library for each backup device being used.
- Parallel Saves and Restores - The ability to save or restore a single object or library across multiple backup devices from the same job. (Note: Integrated File System doesn't support parallel at this time). Understand that the function was designed to help those customers, with very large database files which are dominating the backup window. The goal is to provide them with options to help reduce that window. Large objects, using multiple backup devices, using the parallel function, can greatly reduce the time needed for the object operation to complete as compared to a serial operation on the same object.

- Concurrent operations to multiple backup devices will probably be the preferred solution for most customers. Note that the concurrent large files save/restore is the scenario that achieves the fastest "up to 5.2 terabytes per hour" statement. For the concurrent testing 16 libraries were built, each containing a single with 80 4 GB members. The file size was chosen to sustain a flow across the HSL, system bus, processors, memory and tapes drives for about an hour. We were not interested in peak performance here but sustained performance. Measurements were done to show scaling from 1 to 16 tape drives, knowing that near the top number of tape drives that the system would become the limiting factor and not the tape drives. This could be used by customers to give them an estimate at what might be a reasonable number of tape drives their situation.

Notes: 4Q 2003 358x Ultrium 2 Performance Results

- This testing consisted of an 840 24 way system 128 GB of memory. The 840 doesn't support the 15K RPM DASD in the main tower so only 4, 18 GB 10K RPM RAID protected DASD units were used in the main tower. 15 PCI-X towers (9094 towers), were attached and filled with 45, 35 GB 15K RPM RAID protected DASD units. 2757 IOAs in all 15 towers and 2844 IOPs. All of the towers attached to the system were configured into 8 High Speed Link (HSL) with two towers in each link. One 5704 fiber channel connector in each tower, or two per HSL. A total of 679 DASD, 675 of which were 35 GB 15K RPM DASD units all in the system ASP. We used the new high speed LTO -2 tape drives, model 3580 002 fiber channel attached.

- The workloads for concurrent and parallel save and restore operations were performed on a dedicated system.
- If you are running large file save and restore operations we would recommend only 2 high speed tape drives per HSL. If your data leans more toward user mix you could probably make use of more drives in a single HSL. How many will depend upon your data.
- Remember there are other factors that affect save and restore operations, like memory, number of processors available, and number of DASD available to feed those tape drives.
- The table below is the tabular version of the "large file concurrent save and restore" graphic shown earlier.

	1	2	3	4	8	12	13	14	15	16
Save	365 GB/HR	730 GB/HR	1.09 TB/HR	1.45 TB/HR	2.88 TB/HR	4.15 TB/HR	4.63 TB/HR	4.90 TB/HR	5.14 TB/HR	5.21 TB/HR
Restore	340 GB/HR	680 GB/HR	1.01 TB/HR	1.33 TB/HR	2.56 TB/HR	3.73 TB/HR	4.02 TB/HR	4.28 TB/HR	4.54 GB/HR	4.68 GB/HR

Fastest iSeries to iSeries connection guidelines

- Hardware capacity guidelines (software used to exchange data could reduce these values):
  - 1 Gbps Ethernet runs at around 75-80 MB/sec
  - 2 Fiber Optic HSL runs at roughly 200 MB/sec
  - Copper HSL-2 runs at roughly 400 MB/sec
- OptiConnect runs in two modes:
  - Native
  - IP over OptiConnect
    - IP over OptiConnect flows through the communications software stack
- Customer feedback to Rochester: Native mode (bypasses communications software stack) runs three times faster than IP over the LAN.

Number of Disk Arms sizing

- If application running on current system:
  - Run Collection Services, generate performance database files, use Performance Tools for iSeries System Report
  - Identify disk requests/second that are happening on your current system.
  - Use BMC Patrol-Predict for iSeries or Workload Manager (WLE), or ...
  - Guideline: Knowing average operations/second, look for average service time to be in the 3.5 to 10 millisecond range (lower for newer, higher for old 7200 rpm disks). Numbers higher than these indicate possible system bottleneck.....
  - If upgrading to a faster processor system, new system should be able to perform more disk requests each second. Expect number of disk requests/ second to increase and estimate that for any further sizing exercises.
  - These rules assume an average disk access size of between 6k and 10k in size (shown in the performance report). If you are in the 25k+ range, lower the values by 20%. This also assumes that ALL disks are the same size and are using the same type of controller.

Number of Disk Arms sizing -2

Guideline considerations

- 10k rpm disk drives on 2748/2778/4727/4778 controllers (with 15 drives attached to the controller) run about 20 disk ops/second with adequate to good performance
- Using 10k rpm drives and replacing the above controller with a 2757 means you run up to about 50 disk ops/second/disk with a possibility of still handling peaks at 60 ops/second
- Using 15k drives and a 2757 (15 drives/controller) you should base your sizing on 60 disk ops/second/drive. Most environments do not suffer with peaks of 80 disk ops/second/drive, but this is application dependent.

Notes: **Number of Disk Arms sizing**

To ensure you have sufficient disk arms to meet the needed workload, it is best to have performance runs from your current system run at a time when the disk workload is heavy. These can then be used as input to various tools including the Workload Estimator and Patrol/Predict.

You can also use the reports to determine the number of disk requests/second that are happening on your current system. It's reported in the System Performance Report among others. If this will be a new placement, your application software vendor(s) should be able to provide information from similar sized customers.

Once you know the disk workload in terms of operations/second, the first thing is to determine if a bottleneck currently exists. Depending on the speed/vintage of the disk drives and controllers, average service time sould be somewhere in the 3.5 to 10 millisecond range (lower for newer, higher for old 7200 rpm disks). The numbers are higher than these, there may be a disk bottleneck currently occuring, and therefore "pent-up demand". To determine this pent-up demand, you will need to use tools like Patrol/Predict and someone who is skilled in it's usage -- you'll also need to so some "guessing" because the tools assume a properly tuned system -- and if a bottleneck exists, it cannot be properly tuned.

Next, if upgrading to a larger/newer system, the new system should be able to perform more disk requests each second. However, the new system may also have more memory which acts like a read cache and reduces the number of physical disk accesses that must occur. The net is, there are increases to the number of disk requests/ second that must be estimated.

Once you have determined the number of disk accesses that must be performed each second, it is a relatively simple process to determine the number of disk arms needed.

Notes: **Number of Disk Arms sizing - 2**

Note, there is no substitute for proper modeling, but the following general rules apply in most cases. **These rules assume an average disk access size of between 6k and 10k in size (shown in the performance report). If you are in the 25k+ range, lower the values by 20%.** This also assumes that ALL disks are the same size and are using the same type of controller.

Historically, 10k rpm disk drives on 2748/2778/4727/4778 controllers (with 15 drives attached to the controller) could do about 20 disk ops/second with adequate/good performance - by the time you got to 25 disk ops/second, you were "probably starting to feel it". Any more than that and the controller started ot bottleneck.

Using 10k rpm drives and replacing one of controllers described above with a 2757 controller meant you could go up to about 50 disk ops/second/disk with a possibility of still handling peaks at 60 ops/second (after which the disk drive started to bottleneck).

With 15k drives and a 2757 (15 drives/controller) you should base your sizing on 60 disk ops/second/drive. Most environments do not suffer with peaks of 80 disk ops/second/drive, but this is application dependent.