IBM DFSORT/VSE Performance Considerations

Wolfgang Kraemer

VSE Product Mgmnt
Dept 3221
71032-14 Boeblingen
WKRAEMER at DEVM
wkraemer at de.ibm.com

Update 2001-07-15

Contents

Notes 1
Further Info on DFSORT/VSE 3

A. DFSORT/VSE Performance and Hints

Types of Sorting A.2
DFSORT/VSE Sort Technique Hints A.4
DFSORT/VSE Virtual Storage Areas A.5
DFSORT/VSE Virtual Storage Hints A.6
More DFSORT/VSE Performance Hints A.8
DFSORT/VSE I/O Recommendations A.9
DFSORT/VSE SORTWK Hints A.10
STXIT Performance Impacts A.12

B. DFSORT/VSE 3.1

DFSORT/VSE 3.1 Performance Highlights B.2
DFSORT/VSE 3.1 Performance B.3

C. DFSORT/VSE 3.2

DFSORT/VSE 3.2 Enhancements C.2
DFSORT/VSE 3.2 Performance C.3

D. DFSORT/VSE 3.3

DFSORT/VSE 3.3 Enhancements D.2
DFSORT/VSE 3.3 Performance D.3

E. DFSORT/VSE 3.4

DFSORT/VSE 3.4 Enhancements E.2
DFSORT/VSE 3.4 Performance E.4

Notes

Note

All information contained in this document has been collected and is presented based on the current status.

It is intended to update the performance information in this document.

It is the responsibility of any user of this document

- to use the latest update of this document
- to use this performance data appropriately

This document is unclassified and primarily intended for VSE customers.

It is also available from the INTERNET via the VSE/ESA home page

http://www.ibm.com/servers/eserver/zseries/os/vse

(http://www.ibm.com/s390/vse/ former URL)

Starting with VSE/ESA 2.4, these documents are also available on the VSE/ESA CD-ROM kit SK2T-0060.

The following documents are available via Internet or CD-ROM in Adobe Reader format (.PDF):

'IBM VSE/ESA 1.3/1.4 Performance Considerations'
'IBM VSE/ESA V2 Performance Considerations'
'IBM VSE/ESA Turbo Dispatcher Performance'
'IBM VSE/ESA I/O Subsystem Performance Considerations'
'IBM VSE/ESA VM Guest Performance Considerations'
'IBM VSE/ESA Hints for Performance Activities'
'IBM VSE/ESA TCP/IP Performance Considerations'
'IBM DFSORT/VSE Performance Considerations' (this document)
'IBM VSE/ESA CICS Transaction Server Performance'
'IBM VSE/ESA V2.5 Performance Considerations'
'IBM VSE/ESA Performance on xSeries (NUMA-Q) Enabled for S/390'

The files are
VE13PERF.PDF, VE21PERF.PDF, VE21TDP.PDF, VEIOPERF.PDF, VEVMPERF.PDF, VEPERACT.PDF, VETCPPER.PDF, VESORTP.PDF, VECICSTS.PDF, VE25PERF.PDF, VEXEFSP.PDF

Notes ...

Disclaimer

This document has not been subjected to any formal review or testing procedures and has not been checked in all details for technical accuracy. Results must be individually evaluated for applicability to a particular installation.

Any performance data contained in this publication was obtained in a controlled environment with specific data and is presented only to illustrate techniques and procedures to assist to understand IBM products better.

The results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicability of this data in their specific environment.

Acknowledgements

Thanks to all who contributed directly or indirectly, be it by measurements, suggestions or in other ways.

Very special thanks is due to

Holly Yamamoto-Smith, IBM Sort Products Dept, San Jose

and the entire DFSORT/VSE development team.

All mistakes and inaccuracies in this document are my own.

Please, as in the past, contact me if you have

- suggestions or questions regarding this document
- questions on VSE/ESA performance, not covered in any of the VSE/ESA performance documents

Wolfgang Kraemer, IBM VSE Development, Boeblingen Lab, Germany

Trademarks

The following terms included in this paper are trademarks of IBM:

ES/9000	ESA/390	System/390	CICS	DFSORT
VM/ESA	VSE/ESA	ESCON	ECKD	RAMAC

Further Info on DFSORT/VSE

Further Info on DFSORT/VSE

For - further info on DFSORT/VSE performance,
- description of new functions

refer to the original DFSORT/VSE documentation, especially to

- DFSORT/VSE Installation and Tuning Guide,
  Version 3.3 (02/97) SC26-7041-02
  Version 3.4 (05/98) SC26-7041-03

- DFSORT/VSE Application Programming Guide,
  Version 3.3 (02/97) SC26-7040-02
  Version 3.4 (05/98) SC26-7040-03
  (Refer to Chapter 7: 'Improving Efficiency')

- DFSORT/VSE 3.3 Presentation Guide (02/97)
  DFSORT/VSE 3.4 Presentation Guide (05/98)
  (available from MKTTOOLS, contact your IBM representative)

- What's Happening with DFSORT/VSE?, by John Burt, IBM San Jose
  05/97, VM & VSE Tech Conf Kansas City, MO, Session 30H
  06/97, VM & VSE Tech Conf Mainz, Germany , Session 50E
  11/97, WAVV 97 Conf Chattanooga, TN

- DFSORT/VSE - Latest and Greatest'
  05/98, VM/VSE Tech Conf Reno, Session 31F, by Holly Yamamoto-Smith

- What's New with DFSORT/VSE
  09/98, WAVV Conf Albany, by Holly Yamamoto-Smith and John Burt,
  IBM SSD, San Jose

- DFSORT/VSE - The Complete Sorting and Reporting Solution,
  by John Burt,
  WAVV 2000, Colorado Springs, 10/2000

For more info, you can also surf the INTERNET:

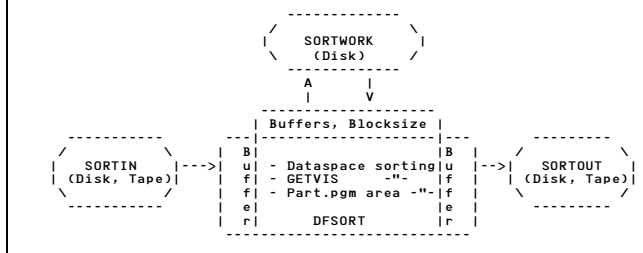
http://www.ibm.com/storage/dfsortvse/

DFSORT/VSE Performance and Hints

PART A.

DFSORT/VSE Performance and Hints

Types of Sorting



Types of Sorting

„ **Dataspace sorting ...**

„ **GETVIS sorting ...**

„ **Partition program area sorting**

í **Incore Sorting (w/o workfiles)**
Enough virtual storage available compared to file size
(performance heavily depends on enough real storage)

í **Non-Incore Sorting (using work files)**
Not enough virtual storage available compared to file size

í **'Work file sorting' on virtual disk**
If e.g. SORTWK1 is put on virtual disk

• Usage sequence if all sorting types are enabled:

1. Dataspace sorting
2. GETVIS sorting
3. Partition program area sorting (least efficient)

Types of Sorting ...

Type of Sorting	DOS/VS S/M 2	DFSORT/VSE	Performance
Dataspace (Before 3.3 only FLR)	-	ESA 1.3 ESA ESA 2.x	High a)
GETVIS (priv. space)	-	ESA 1.1/1.2 ESA 1.3 /370 ESA 1.3 /ESA ESA 2.x	High a)
Partition program area (old fashioned)	VSE/SP VSE/ESA	ESA 1.1/1.2 ESA 1.3 /370 ESA 1.3 /ESA ESA 2.x	Low a)
'Work file' on Virtual Disk	VSE/ESA ESA	ESA 1.3 ESA ESA 2.x	b)

- GETVIS sorting w/o 31-bit (i.e. before VSE/ESA 1.3) and partition program area sorting are limited by the partition size below the 16M line

- GETVIS and Dataspace Sorting are about identical in Elapsed and CPU-time

- Dataspace sorting and Virtual Disk require ESA H/W

- ESA 1.3 here includes VSE/ESA 1.4

a) No work file I/Os if sorting space big enough:
'Incore Sorting'.
Workfile I/Os increase CPU and Elapsed time:
'Non-Incore Sorting'

b) No physical I/Os to workfiles, small CPU-time overhead.
Virtual Disk may be applied e.g. to SORTWK1 extent,
others may stay on real disk

í **Full DFSORT benefits available with VSE/ESA 1.3 ESA and up**

í **DFSORT performance improves with DFSORT releases**

DFSORT/VSE Sort Technique Hints

- **Select/Enable type of sorting which would allow incore sorting**

  (w/o workfiles, or just with the biggest size possible)

  This is either GETVIS Sorting or Dataspace Sorting

  GETVIS and Dataspace Sorting have very similar performance

- **Avoid 'old fashioned' Partition Sorting even for smaller sorts**

  For WORK=0 no workfiles are used, BUT then only 1 buffer is used for non-VSAM SYSIN and SYSOUT.

  For WORK>0 at least these workfiles must be OPENed.

- **Avoid the need for using workfiles**

  Even when you put them on VSE Virtual Disk.

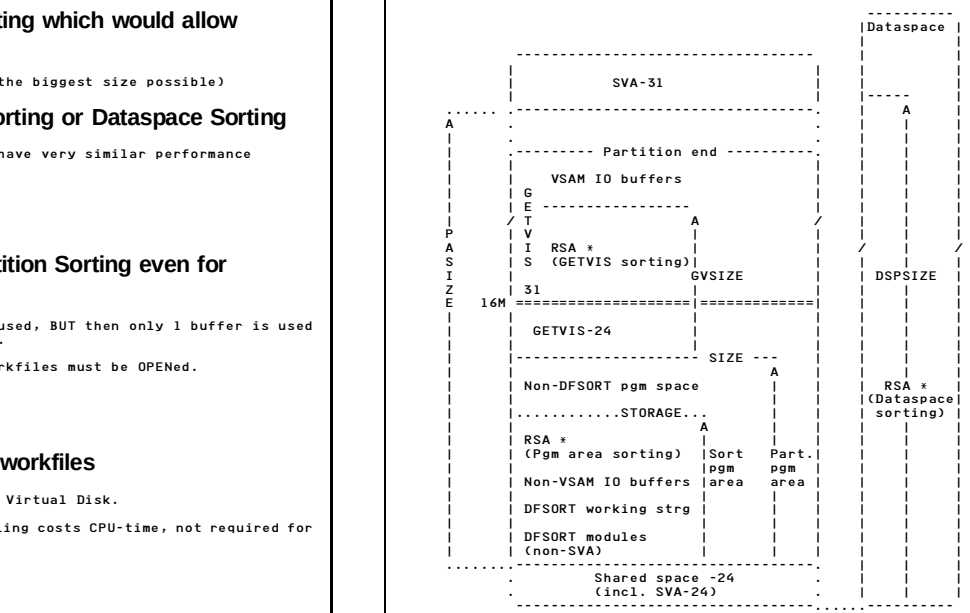
  Workfile and Virtual I/O handling costs CPU-time, not required for incore sorts.

  'No I/O is the fastest I/O'.

More DFSORT/VSE hints on next foils.

DFSORT/VSE Virtual Storage Areas

DFSORT/VSE Virtual Storage Areas



- STORAGE option designates the amount of the program area to be used by DFSORT. Default is 0 and means total SIZE value.

* RSA = Record Storage Area

DFSORT/VSE Virtual Storage Hints

Virtual Storage Performance Hints

- **General**
  - **More DFSORT/VSE code above the line**
  - **All GETVIS and Dataspace sorting areas may reside above the line**
- **DFSORT specifications in OPTION statement**
  - **GVSIZE=n, nK, nM, or MAX**

    Amount of partition GETVIS (-24 and -31) for GETVIS sorting. 'Enables' GETVIS sorting.

    Amount of GETVIS area used when GVSIZE=MAX (3.2 and up) is based upon

    - available partition GETVIS
    - amount of reserved partition GETVIS for operating system and user application (GVSRLOW and GVSRANY)
    - amount of real storage
    - current paging activity in system
  - **DSPSIZE= n or MAX**

    Amount of space (measured in MB) useable in the dataspace for Dataspace sorting.

    Amount of Dataspace used for DSPSIZE=MAX (3.2 and up) is decided similar to GVSIZE=MAX.

- **Specify enough GVSIZE or DSPSIZE**

  Use MAX only when need is required or impact on concurrent workload is under control

DFSORT/VSE Virtual Storage Hints ...

Virtual Storage Performance Hints (cont'd)

- **Specify enough EXEC SIZE for all Sorts**

  even w/o Partition Program Area Sorting, even w/o work files.

  The reason for this is that non-VSAM I/O buffers reside in the Partition Program area and more buffers are required to use more efficient channel programs (higher blocking).

  Use EXEC DFSORT,SIZE=768K or more, to run DFSORT efficiently

  Increase the SIZE value if
  - User exit routines are used
  - DFSORT modules are NOT loaded into SVA.

  A too high SIZE value can only reduce the GETVIS area, but only the GETVIS-24 part. GVSIZE not directly reduced.

  For program invoked DFSORT ('EXEC pgm, SIZE=nnnnK'), nnnn should also include the size for the calling pgm.

- **Place all eligible DFSORT modules into the SVA**

  All SVA-31 eligible ones, plus 16K into SVA-24

  Benefits:
  - No I/Os for loading of these phases required
  - Virtual and real storage for these phases only once required (beneficial for concurrent SORTs)
  - Better protection from 'rogue' applications (side effect)

  Continuously improved DFSORT/VSE SVA-24 requirements:

	DOS/VS S/M 2	D/V 3.1	D/V 3.2	D/V 3.3	D/V 3.4
SVA-24 only	240K	205K	65K	14K	16K
SVA-31 eligible	-	63K	210K	314K	354K

- D/V = DFSORT/VSE

More DFSORT/VSE Performance Hints

More DFSORT/VSE Performance Hints (cont'd)

- **Use the DIAG operand on the OPTION control statement to get addt'l performance related info**

```
regarding    - Virtual storage usage
             - Work space usage
             - Optimization parameters
             - Data movement

Refer to the DFSORT/VSE Installation and Tuning Guide
```

- **Avoid DFSORT specifications that degrade performance**

```
 e.g.  - BYPASS          - LOCALE
       - CKPT            - VERIFY
       - ERASE
       - EQUALS

For more info refer to the DFSORT/VSE Application Program Guide
```

- **Use the FASTSRT COBOL/VSE compiler option**

```
Applicable also for COBOL II, not for DOS/VS COBOL.
PL/I's PLISRTA is equivalent.

 - For COBOL/VSE invoked sorts, causes DFSORT to do the
   DASD I/Os for input and output files.

   Without FASTSRT, data is read/written record by record.

   DFSORT has usually better channel programs,
   resulting in
                - less I/Os / faster I/Os
                - less CPU-time

   so, in total, results in faster sorts.
```


DFSORT/VSE I/O Recommendations

DFSORT/VSE I/O Recommendations

- **ADD all devices as ECKD**

```
This is a general recommendation for (E)CKD and also a functional
need for all devices (3380/3390) on RAMAC or Internal Disk
```

- **Use DFSORT/VSE 3.4**

```
Gives most optimal channel programs.
Refer to 'DFSORT/VSE 3.4 Enhancements'.
```

- **For optimal I/Os to SORTINx and SORTOUT ...**

  use SAM ESDS (ACB) or (native) VSAM ESDS

```
ACB access (regarding data access) means:

    - optimal VSAM ECKD channel programs with cache bit settings
    - maximum I/O blocking (# bytes per I/O),
      resulting in lowest # I/Os


In DFSORT/VSE 3.4 channel programs by DFSORT have been improved
(DTFPH).
```

- **For SORTOUT as SAM or SAM ESDS (DTF) ... use a high SORTOUT BLKSIZE value**

```
Applies to blocked SORTOUT.
No need for SAM ESDA (ACB) or native VSAM.
```

- **For optimal I/Os to SORTWK**

  - if workfiles cannot be avoided - ...

  refer to the hints on next foils

DFSORT/VSE SORTWK Hints

DFSORT/VSE SORTWK Hints

```
Again, try to avoid usage of workfiles
by using large GETVIS or Dataspace sorting areas.

Also refer to DFSORT/VSE Application Programming Guide, p.276:
        'Using Work Space Efficiently'.
```

- **Adequate total size of workfiles**

```
When a SORT can't be performed in virtual storage, DFSORT/VSE must
use external work space.  How much external work space is needed
will depend on:

 - The size of your input files (ideally, the work files should
   each be as large as the input files plus about 25%)

 - The amount of virtual storage available to DFSORT/VSE for
   sorting (data space or GETVIS area).
```

- **Specify // EXEC SORT,SIZE=768K for optimal SORTWK I/Os**

```
All non-VSAM I/O buffers reside in the Partition Program area.
Thus use at least this SIZE value.
```

- **Use VSE Virtual Disk for SORTWK, if you have enough real storage**

```
Do not pass the point where you provoke VSE paging instead
```

- **Use 2 or more work files (instead of 1) on separate devices to potentially reduce elapsed time**

```
Benefit is from overlapping I/Os.

Using more than 3 will not reduce elapsed time further.
```


DFSORT/VSE SORTWK Hints ...

DFSORT/VSE SORTWK Hints (cont'd)

- **Control the number of SORTWK files to avoid unnecessary OPENs for program invoked sorts**

```
 - DFSORT/VSE will attempt to open all the workfiles
   as specified in the SORT WORK=n control statement

 - COBOL II and COBOL/VSE, when calling DFSORT, dynamically
   generate WORK=n (on the SORT statement) at runtime,
   based on the DLBLs in the JCL

 - DOS/VS COBOL programs may directly contain SORT WORK=n

 - DLBLs for SORTWK1 .. n  is selected in this sequence:
              - JCL of the job step
              - partition labels
              - standard labels

 - You can also specify SORTWK DLBLs and SORT WORK=n in the
   JCL of the job step.
   LE/VSE 1.4 APAR PQ04468 (PTF UQ05847) must be applied,
   to allow overriding of the COBOL generated SORT statement
```

  Use WORK=n values as small as possible

```
E.g. If there is enough GETVIS or Dataspace available
       to do incore sorts
 or
     If you know that certain sorts are always small

-> Specify WORK=1 (or even 0)
```

  Do not specify too many SORTWK DLBLs

```
If WORK=n is too high, unnecessary OPENs occur
```

- **Use SAM ESDS work files (in VSAM space)**

```
DFSORT/VSE initially only allocates/opens the first extent.
To allow secondary extents, make sure WRKSEC is specified,
(the installation default) together with a good size.

No delta in SORTWK I/Os vs SORTWK in BAM space exists,
since DFSORT sets up the same channel programs
```

STXIT Performance Impacts

STXIT Performance Impacts with Pgm Invoked Sorts

High performance impacts (CPU-time) may occur

if all of the following criteria are met:

- o DFSORT/VSE is called from a HLL
- o E15 and/or E35 user exit routines are used
- o STXIT=YES installation or STXIT runtime option is set

Overview on STXIT ('setexit') related installation/runtime options:

Ù **STXIT=YES option (or STXIT)**

Tells DFSORT/VSE to issue the STXIT SVC every time control is returned from the user exit routine (once per record).

So STXIT=YES costs CPU-time, especially for Turbo Dispatcher. The STXIT overhead is even higher, if TRAP(ON) is used for LE, instead of TRAP(OFF)

Ù **STXIT=MIN option (or MINSTXIT)**

New in DFSORT/VSE 3.4.

Tells DFSORT to NOT issue the STXIT SVC every time control is returned from the user exit routine (once for every record).

It simply saves the caller's STXIT, establishes its STXIT once, and restores the caller's STXIT before returning control to the caller (as it is also done for STXIT=YES).

Ù **STXIT=NO option (or NOSTXIT)**

Turns off DFSORT abend recovery. Also disables in DFSORT 3.2/3.3 the DFSORT recovery feature for SAM ESDS multivolume work files.

STXIT settings have an even higher impact e.g. in case of LE with TRAP(ON), which causes additional SVCs after user program STXIT calls.

Again:

STXIT options affect performance only when user exit routines are used.

Í **To avoid unnecessary CPU-time, see the STXIT recommendations on the next chart.**

STXIT Performance Impacts ...

Recommended STXIT abend recovery settings

STXIT definitions in ILUINST or equivalent runtime options

E15/E35 user exit routines	DFSORT/VSE 3.4	DFSORT/VSE <3.4
yes	STXIT=MIN *1	STXIT=NO *2
no	STXIT=MIN|YES *3	STXIT=YES *3
DFSORT default	STXIT=MIN	STXIT=YES

- None of the recommendations shown here (even the cases with E15/E35 user exit routines) result in any STXIT performance degradation (no STXIT calls on a per record basis)

*1 If an abend occurs in the user exit routine, DFSORT will not perform abend recovery. The performance overhead of STXIT=YES (CPU-time) is avoided (DFSORT does not restore its STXIT)

*2 This turns off DFSORT abend recovery. In DFSORT 3.2/3.3, it also disables the DFSORT recovery feature for SAM ESDS multivolume work files.

*3 Allows DFSORT to do abend recovery and clean up if an abend occurs. In this case STXIT=MIN and YES would have same function and good performance, since no user exits exist.

More info

Refer to - Information APAR II11222

- 'Taking Advantage of LE/VSE', SG24-4798-00 ITSO Boeblingen Red Book, 11/96, 75 pages Appendix A: Performance Considerations

DFSORT/VSE 3.1

PART B.

DFSORT/VSE 3.1

DFSORT/VSE 3.1 Performance Highlights

DFSORT/VSE 3.1 available since 09/94,
DFSORT/VSE 3.2 available since 10/95,
DFSORT/VSE 3.3 available since 02/97,
DFSORT/VSE 3.4 available since 05/98

DFSORT/VSE 3.1 Performance Highligths

Ù **ESA exploitation for improved performance and VSCR**

Ù **Usage of**

„ **31-bit Addressing for DFSORT modules Sorting Areas (RSAs)**

„ **ESA dataspaces for sorting of fixed-length records (FLR)**

„ **Partition GETVIS for sorting of variable-length records (VLR) + FLR**

Ù **Allocation of work files possible via VSAM managed space**

- File type then is SAM ESDS
- Dynamic secondary allocations of work files via VSE/VSAM space management
- Multivolume work files via VSE/VSAM space management

DFSORT/VSE 3.1 Performance

Performance Results for DFSORT/VSE 3.1

Possible Improvements

		DFSORT/VSE 3.1 vs DOS/VS S/M 2.5	
		FLR (fixed)	VLR (variable)
Partition program area sorting (real disk)	ET CPUT #IO	= = =	= +4% =
GETVIS sorting	ET CPUT #IO	-72% -27% -57%	-74% -36% -69%
Dataspace sorting	ET CPUT #IO	-72% -26% -57%	- - -

= Means 'equivalent within 2%'

Run parameters for controlled lab environment:
- 9121-742, VSE/ESA 1.3.4, 3390-3 DASDs
- Single partition runs
- 80 byte records
- Results from VSE Job Accounting

- **Partition program area sorting is the same for D/V 3.1 and S/M 2.5**
- **Benefit of new sorting options: Reducing elapsed times up to about 1/3rd by saving of up to about 1/2 of total DASD I/Os**
- **Dataspace sorting as fast as GETVIS sorting**

NOTE to performance figures for 'Possible Improvements':

The percentages shown are observed maximum values and thus cannot be added up across releases

DFSORT/VSE 3.2

PART C.

DFSORT/VSE 3.2

DFSORT/VSE 3.2 Enhancements

DFSORT/VSE 3.2 Enhancements vs 3.1

... besides additional productivity features

- **Moved additional DFSORT modules from SVA-24 to SVA-31**

  140K relief for VSE system

- **Reduced work file requirements**

- **Dynamic Control of storage for sorting via DSPSIZE/GVSIZE=MAX**

  Improved tuning of D/V to minimize paging activity.

  Especially useful for critical jobs.

- **Reduced number of I/Os to files on DASD**

  Improved I/O buffering and blocksize for

	SORTIN	SORTWORK	SORTOUT
Dataspace sorting	-	X	-
GETVIS sorting	X	X	X
Pgm area sorting	-	-	-

DFSORT/VSE 3.2 Performance

DFSORT/VSE 3.2 Performance

Possible Improvements

			DFSORT/VSE 3.2 vs 3.1		
Sorting	Work Files	Record Format	Elapsed Time	#I/O	CPU-Time
GETVIS	N	FLR	-77%	-87%	-16%
		VLR	-75%	-84%	-25%
GETVIS	Y	FLR	-64%	-75%	-18%
		VLR	-60%	-70%	-14%
Dataspace	Y	FLR	-35%	-45%	-13%

- Small, medium, and large fixed-length records (FLR) and variable-length records (VLR) files (from 4.6MB to 70.4MB)
- Number of records varied from 14,500 to 220,000
- Block sizes varied from 6,400 to 56,000
- Record lengths for FLR and avg record lengths for VLR were 320 bytes
- Input keys were in random order and 20 bytes long

- DFSORT/VSE 3.2 and 3.1 were resident in SVA
- VSE/ESA 2.1

- Stand-alone on 9021-742 with 512M real storage
- 3390-2 DASDs at a 3990-3 cached control unit (128M cache, 4M NVS)

- Results from VSE Job Accounting

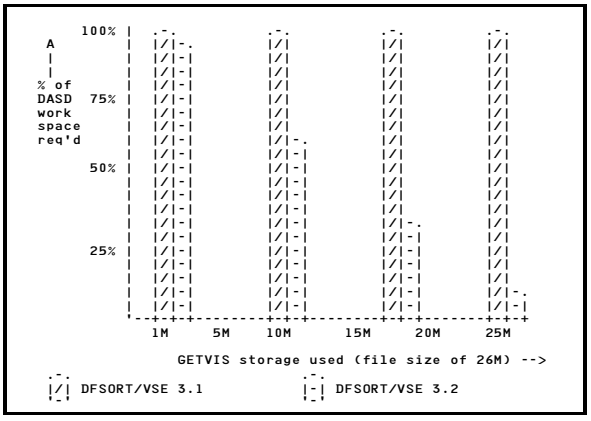
- Figures from Announcement Letter

- **Significant performance enhancements possible vs DFSORT/VSE 3.1**

  e.g. Elapsed Time can be improved by up to 77%

DFSORT/VSE 3.2 Performance ...

Required Work File Sizes



> As the amount of GETVIS area storage increases, the amount of DASD work space used in DFSORT/VSE 3.2 decreases vs DFSORT/VSE 3.1. When the GETVIS area is larger than the file size, the amount of GETVIS storage will be sufficient for DFSORT/VSE 3.2 to perform GETVIS sorting without DASD work space.

> Similar DASD work space reductions are obtained when using Dataspace sorting with DFSORT/VSE 3.2.

DFSORT/VSE 3.3

PART D.

DFSORT/VSE 3.3

DFSORT/VSE 3.3 Enhancements

DFSORT/VSE 3.3 Enhancements vs 3.2

... besides additional productivity features

- Ù **Dataspace Sorting for Variable Length Records (VLR)**
- Ù **SVA-24 and PrivateSpace-24 modules moved to SVA-31**
  - 50K relief (VSCR) for VSE system
  - Reduce FETCH I/Os

  Refer to chart on DFSORT Virtual Storage Areas for a survey across all DFSORT releases
- Ù **Improved I/O handling**
  - „ **Improved channel programs for ECKD defined devices**

    Such channel programs now optimally suited for VSE/ESA CKD/ECKD Conversion routine (SORTIN, SORTWORK, SORTOUT)
  - „ **Double Buffering**

	SORTIN	SORTWORK	SORTOUT
COPY & MERGE	X	n/a	X
Dataspace sorting	X	X	X
GETVIS sorting	X	X	X
Pgm area sorting	-	-	-

  - „ **Much higher blocking (KB/IO) for DASD and Tape I/O (via CCW chaining)**

    Applies to DASD I/O (SORTIN, SORTWORK, SORTOUT)
  - „ **Larger blocksize internally used (SORTWORK)**

DFSORT/VSE 3.3 Performance

DFSORT/VSE 3.3 Performance

Possible Improvements

			DFSORT/VSE 3.3 vs 3.2		
Activity	Work Files	Record Format	Elapsed Time	#I/O	CPU-Time
COPY	-	FLR	-19%	-44%	-30%
		VLR	- 7%	-42%	-48%
MERGE	-	FLR	-20%	-43%	-27%
		VLR	-11%	-42%	-40%
Dataspace SORT	Y	FLR	- 3%	-41%	-13%
	Y	VLR *1	-59%	-67%	-44%
Dataspace SORT	N	FLR	-26%	-87%	-33%
	N	VLR *2	-17%	-66%	-34%
GETVIS SORT	N	FLR	- 6%	-24%	+-0%
	N	VLR	-14%	-34%	-34%

- Small, medium, and large fixed-length records (FLR) and variable-length records (VLR) files (from 3MB to 67MB)
- Number of records varied from 10,000 to 220,000
- Block sizes varied from 6,400 to 12,800 byte
- Record lengths for FLR and avg record lengths for VLR were 320 bytes
- Input keys were in random order and 8 bytes long

- DFSORT/VSE 3.3 and 3.2 were resident in SVA
- VSE/ESA 2.1

- Stand-alone on 9672-R73 with 512M real storage
- 3390-2 DASDs at a 3990-3 cached control unit (128M cache, 4M NVS)

*1 DFSORT 3.2 used pgm area sorting for VLR, since dataspace sorting for VLR not available

*2 DFSORT 3.2 used GETVIS sorting for VLR, since dataspace sorting for VLR not available

- Results from VSE Job Accounting

- Figures from Announcement Letter

DFSORT/VSE 3.3 Performance ...

DFSORT/VSE 3.3 Performance (cont'd)

- **Significant performance enhancements possible (vs DFSORT/VSE 3.2)**

  e.g. Elapsed Time can be reduced by up to about 20%

Disclaimer

The actual performance characteristics that may be experienced by any specific user or for any specific file depend on many factors, including record length, file size, and DFSORT/VSE storage options.

Noticeably, in a multitasking environment, elapsed time results are application-profile and workload dependent. So, the results may differ from user to user.

IBM does not represent nor warrant that users will experience the same changes in performance characteristics observed in these examples.

DFSORT/VSE 3.4

PART E.

DFSORT/VSE 3.4

DFSORT/VSE 3.4 Enhancements

DFSORT/VSE 3.4 Enhancements

... besides additional functional enhancements

Most performance benefits from the I/O area of DFSORT:

- **Better exploitation of tracks**

  Applies to VLR on SAM SORTOUT files

  The type of benefits depends on the type of I/O subsystem:

Better exploitation of tracks	Real DASD	RAMAC/ID	RVA
- for less total I/O time - for less 'real DASD storage'	X X	X X	X -
ID stands for Internal Disk. RVA is RAMAC Virtual Array, which already compresses logical tracks			

- **Setup of even more optimal channel programs**

	SORTIN	SORTWORK	SORTOUT
SORT (non-VSAM)	Multitrack, ECKD	- ECKD	Multitrack, ECKD
SORT (VSAM ACB)	VSAMBSP	-	VSAMBSP
COPY/MERGE	Cmd chain., ECKD	n/a	(3.3 Cmd ch.) ECKD
- Multitrack: Multiple tracks per I/O - ECKD: Channel programs are now native ECKD, using SEQ cache indication in DX. CKD-ECKD conversion no more required - VSAMBSP: User controls #VSAM I/O buffers - Cmd chain.: Multiple blocks per I/O to non-VSAM DASD or Tape (standard for VSAM)			

- **Reduced #I/Os for better Elapsed Times**

  Reduces also total CPU-times, and the non-parallel shares for the Turbo Dispatcher

- **Reduced msec/IO for better Elapsed Times**

DFSORT/VSE 3.4 Enhancements ...

DFSORT/VSE 3.4 Enhancements vs 3.3 (cont'd)

- **Reduced DASD workfile requirements for VLR Dataspace and GETVIS Sorting**

  through more efficient use of RSA.

  The decrease in required external workspace comes along with

  - reduced total #I/Os
  - thus reduced total I/O time

- **Larger RSA for Dataspace Sorts results in more Incore Sorts**

  This stems from better space allocation and applies to the intermediate and the final sorting phase (RSA = Record Storage Area).

- **Reduced STXIT overhead via new DFSORT option**

  STXIT=MIN (default) or MINSTXIT will avoid CPU overhead of COBOL and PL/I programs using E15/E35 user exits.

  Was already discussed in a previous chart 'STXIT Performance Impacts'

- **Recent (06/2001) Enhancement for DFSORT 3.4: More than 2G records can be sorted**

  So far, the number of records to be sorted was stored as a FIXED(31) field (single word counter), i.e. limited to 2147483647 records.

  APAR PQ44102 describes how this was transparently changed to a double word counter, enhancing the sorting capacity.

  This is a response to the DFSORT requirement WAVV1997042.

DFSORT/VSE 3.4 Performance

Possible 3.4 Sorting Improvements

Sorting Activity	Work Files	Record Format	DFSORT/VSE 3.4 vs 3.3 Elapsed Time	#I/O	CPU-Time	Workfile Space
Dataspace Sorting	N	FLR	-23%	-78%	- 7%	-
	N	VLR	-21%	-69%	- 5%	-
	Y	FLR	-13%	-50%	-23%	+-0%
	Y	VLR	-29%	-59%	- 5%	-29%
GETVIS Sorting	N	FLR	-23%	-78%	- 6%	-
	N	VLR	-22%	-69%	- 4%	-
	Y	FLR	-11%	-50%	- 8%	+-0%
	Y	VLR	-30%	-59%	- 7%	-29%
Partition Sorting	N	FLR	+ 4% *)	-57%	+-0%	-
	N	VLR	- 6%	-67%	- 3%	-
	Y	FLR	-10%	-18%	- 6%	+-0%
	Y	VLR	- 7%	-31%	+-0%	+-0%

- VSE/ESA 2.2.1, standalone on 9672-R13 with 512M
- 3390-2 DASDs at 3990-3 control unit (256M cache, 4M NVS)
- All eligible DFSORT modules in SVA
- Results from VSE Job Accounting

- Records/files are described in next table

- Dataspace Sorting: 5 to 26M of data space was used,
  GETVIS -"- : 5 to 26M of partition GETVIS,
  Partition -"- : 64 to 5376K of program area.

*) For old-fashioned Partition Sorting, only small deltas seen due to the limited size of the partition pgm area.

- Figures from Announcement Letter, 'your mileage may vary'

í **Significant reductions in #I/Os for all types of sorting**

give better CPU-Times and Elapsed Times

DFSORT/VSE 3.4 Performance ...

Possible 3.4 COPY/MERGE Improvements

Activity	Work Files	Record Format	DFSORT/VSE 3.4 vs 3.3 Elapsed Time	#I/O	CPU-Time	Workfile Space
COPY (Disk)	-	FLR	-23%	-95%	-58%	-
	-	VLR	-22%	-96%	-54%	-
COPY (Tape)	-	FLR	-32%	-96%	-53%	-
	-	VLR	-38%	-96%	-43%	-
MERGE (Disk)	-	FLR	-20%	-95%	-54%	-
	-	VLR	-12%	-95%	-57%	-
MERGE (Tape)	-	FLR	-39%	-93%	-49%	-
	-	VLR	-33%	-94%	-42%	-

- Environment described in previous table

- Workfiles not applicable for COPY/MERGE

- Small, medium, and large fixed-length (FLR) and variable-length record (VLR) files (from 3M to 67M)
- Number of records varied from 10,000 to 220,000
- Block sizes varied from 6,400 to 56664 byte
- Record lengths were 320 bytes (FLR and avg VLR)
- Input keys were in random order and 8 bytes long

í **Significant reductions in #I/Os for COPY/MERGE**

give much better Elapsed Times plus better CPU-Times

DFSORT/VSE 3.4 Performance ...

DFSORT/VSE 3.4 vs Sort/Merge V2R5

Activity	Work Files	Record Format	DFSORT/VSE 3.4 vs S/M 2.5 Elapsed Time	#I/O	CPU-Time	Workfile Space
Dataspace Sorting	N	FLR	-35%	-71%	-10%	-
	N	VLR	-36%	-72%	-14%	-
	Y	FLR	-56%	-64%	-38%	-38%
	Y	VLR	-58%	-75%	-31%	-35%
GETVIS Sorting	N	FLR	-36%	-71%	-10%	-
	N	VLR	-32%	-72%	-11%	-
	Y	FLR	-56%	-64%	-40%	-38%
	Y	VLR	-58%	-75%	-31%	-35%
Partition Sorting *)	N	FLR	- 4%	-57%	+ 5%	-
	N	VLR	- 5%	-67%	- 7%	-
	Y	FLR	- 9%	-18%	- 8%	+-0%
	Y	VLR	- 7%	-23%	- 6%	+-0%
COPY (Disk)	-	FLR	-39%	-97%	-71%	-
	-	VLR	-35%	-97%	-75%	-
MERGE (Disk)	-	FLR	-36%	-97%	-66%	-
	-	VLR	-31%	-97%	-74%	-

- Environment and records/files described in previous two tables

- Note that Sort/Merge V2R5 only has Partition Sorting

- Dataspace Sorting: 5 to 26M of data space was used,
  GETVIS -"- : 5 to 26M of partition GETVIS,
  Partition -"- : 64 to 5376K of program area.

*) For old-fashioned Partition Sorting only small deltas due to the limited size of the partition pgm area.

- Figures from Announcement Letter, 'your mileage may vary'

í **Significant improvements vs Sort/Merge**

EOD End Of Document
HAND Have a nice day