

**EXPERIENCES WITH  
SUPERCOMPUTING IN A BS2000  
ENVIRONMENT**

**SAVE, Madrid**

**October 1998**

Joachim Backes  
Dr. Martin Bürkle

Regionales Hochschulrechenzentrum

Postfach 3049  
D-6750 Kaiserslautern

## Table of Contents

Abstract.....	3
1 Environment: Coexistence between BS2000 and the VP world.....	4
2 The USER200 interface.....	6
3 The FORTRAN System .....	8
4 User training.....	9
5 Performance of File Transfer.....	10
6 Performance of job transfer.....	11
7 Overhead of VP2000.....	13
8 Conclusions.....	14

## **Abstract**

A common project of SIEMENS München and the Universität Kaiserslautern called VP2000 integrates the VP supercomputer series into the BS2000 environment. A complete FORTRAN system which is compatible to the VP FORTRAN system is available for BS2000 users. It is not necessary to learn the foreign job control language of the VSP because all functions have been embedded into the comfortable system dialog facility SDF of BS2000.

The implementation of the new facility was done in a very efficient way to keep overhead at a minimum. A comparison of local /COPY command in BS2000 to the /TRANSFER command between BS2000 and VSP shows the high performance of the connection.

An overview of the necessary user training and the achieved vectorization degrees show the scope of the new coexistence.

## 1. Environment: Coexistence between BS2000 and the VP world

The use of computers of the VP series hitherto was only possible in a JES2 or a JES/MAS environment. The cooperation project VP2000 however, shared between the SIEMENS AG and the RHRK (COMPUTER CENTER OF THE UNIVERSITY OF KAISERSLAUTERN), realized the attachment of the BS2000 world to the VP world in a comfortable way. Attachment - in this environment - does not mean access via slow lines, using telecommunication protocols like HDLC, MSV OR BSC (slogans: RJE, NJE, TRANSIT). The VP2000 solution allows a performant access, using the so called "SHARED DASD", that is common disk drives between the two systems BS2000 and VSP. These DASD's have full IBM DMS format (DATA MANAGEMENT SYSTEM).

The functions of the connection can be roughly divided into four parts:

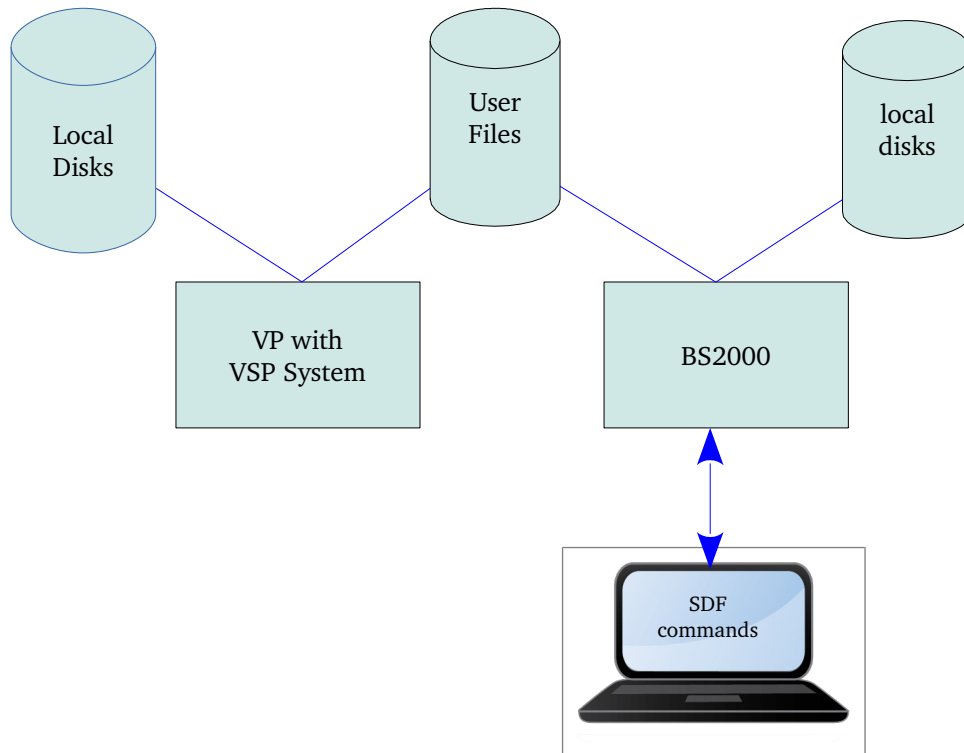
- JEF2000            Enables job input from BS2000 to VSP, the receipt of VP job output, Job/Data Set information and management services;
- COPY2000        Is used for the transfer of mass data between BS2000 and the VSP (both directions);
- F77                Compiler-, vectorizer-, tuning- and debugging system for FORTRAN77 objects, running in BS2000 as well as in VSP;
- USER2000       comfortable SDF interface for the call of services which are hidden behind JEF2000, COPY2000 and F77.

VP2000 is the short name for these four parts.

A very important aspect during the implementation has been the following maxim: relief of the VSP system, as far as possible, of pure scalar activities, that means:

- Avoid scalar VSP utilities;
- Information meant for the VP should be prepared in BS2000, for example
- preparation of job submission,
- postprocessing of listings,
- BS2000 direct access to the SHARED DASD, both for user data and VTOC entries;
- Compiling, linking, testing and optimizing of FORTRAN77 programs is accomplished in BS2000, the VP is provided only for production steps.

The following picture gives a short overview of a typical configuration supported by VP2000:



Critical actions, for example the access coordination of VP data sets during the COPY2000 processing, are realized by very small VSP "mini" jobs. Such jobs are running below the VP line (that means fully virtual), and they don't consume a lot of CPU and memory.

Functions, needing more elapsed time, like complete VP jobs, or management services which only can be processed by the VSP system - creation of a data set for example - are packed into VP batch jobs. This means: no transaction processing by the BS2000 part of VP2000, no use of session protocols.

Information which has been produced during the "mini" jobs, will be transferred back in the original binary format to BS2000. This avoids additional scalar load of the VP. The final processing of such data is done by BS2000. Example: VTOC entries of data sets.

As mentioned above, the access to the VP data sets is done by a complete file copy. Catalog services are performed by the VSP system, supported by the use of "agent" mini jobs for open, close a.s.o. Especially no new access method has been developed in the BS2000 for VP data sets.

## 2. The USER200 interface

The USER2000 interface allows the users to access the VP services, similar to the local BS2000 processing. That means: all activities concerning the VP access operate on command level. The definition of these commands is completely done on the basis of the command language SDF (System dialog facility). Explanations and help functions are integrated in a comfortable manner. There are no restrictions for users with a lot of MVS or MSP experience, who want to use the whole environment of the VSP Job control language. They can submit original VP jobs. But: the MVS or MSP dialog interface is not available, because the VSP itself does not support such an access.

As far as possible, the command names have been derived from the correspondent local BS2000 commands by inserting the string "VECTOR". Therefore an experienced BS2000 user has no problem to find out the related VP2000 command.

Normally, a BS2000 user sees no difference between BS2000 and VSP processing, internally created jobs are hidden. Additionally, this allows to transfer functions from the VSP to BS2000 without changing the user interface, only the message layout may change during such a migration.

The meaning of system codes, which are generated during VP activities or jobs, for example at program abortion, are available in BS2000 too, using the standard message system, including detailed explanation and proposals for a user reaction.

The following command groups are available:

<b>Group 1: Compile/link/go</b>	
COMPILE-FORTRAN-SOURCE	For the compilation of a FORTRAN77 source with the FORTRAN77 or the FORTRAN77/VP compiler on the BS2000 system.
LINK-VECTOR-PROGRAM	Linking of a scalar or vector program
ENTER-VECTOR-PROGRAM CALL-VECTOR-PROGRAM	For the execution of a linked program in the VP. CALL means: wait until the VP program has ended and until the results arrived in BS2000. ENTER means: the user can go on directly after the program start has been submitted.
ASSIGN-VECTOR-SYSDTA	Generates a description of the standard input data for the FORTRAN77 program, which can be found in BS2000 files or library members
SET-VECTOR-FILE-LINK GENERATE-DD-STATEMENT	Describe more generally than ASSIGN-VECTOR-SYSDTA the VSP data sets for the FORTRAN program execution.
OPTIMIZE-VECTOR-FORTRAN-SOURCE	Allows the (vectorized) optimization of FORTRAN77 programs, using the tools FORTUNE and VECTUNE.
<b>Group 2: Job management</b>	
ENTER-VECTOR-JOB CALL-VECTOR-JOB	For the execution of VP activities using the original VSP JCL. The difference between ENTER and CALL is the same as in the VECTOR-PROGRAM command.
CANCEL-VECTOR-JOB	Allows the cancellation of VP jobs
SHOW-VECTOR-JOB-STATUS	Delivers actual status information about VP jobs
MODIFY-VECTOR-PROTECTION	Changes the password for the VP access
ENTER-VECTOR-COMMAND-PROCEDURE CALL-VECTOR-COMMAND-PROCEDURE	Allow the call of VP JCL command procedures (with layout similar to BS2000 command procedures) for execution in the VP
PERFORM-VECTOR-SYSTEM-COMMAND	Allows privileged users to submit from BS2000 system commands or JES commands for the VSP.
<b>Group 3: Data set management</b>	
TRANSFER-VECTOR-FILE	Copies complete BS2000 files or library members into VP data sets or library members (and vice versa)
CREATE-VECTOR-FILE	Creates a VP data set or VP library
DELETE-VECTOR-FILE	Erases VP data set(s) or library(ies)
COMPRESS-VECTOR-LIBRARY	Compresses a VP library directly by B2000 (similar to TRANSFER-VECTOR-FILE)
SHOW-VECTOR-FILE-ATTRIBUTES	Gives information about the status of VP data sets, libraries and library members.

### 3. The FORTRAN System

All VP user programs are written in FORTRAN. Unfortunately there are some minor incompatibilities between FORTRAN77/VP and FOR1 source language concerning the length of names etc. Major differences arise in the routines handling error and exception situations.

In the scope of VP2000 project, the VP compiler was migrated to run in a BS2000 environment and to produce either

- scalar BS2000 code,
- scalar VP code,
- vectorized VP code.

The appropriate runtime system was also migrated to BS2000. And the same has been done for the two optimization tools FORTUNE and VECTUNE.

With an interface to AID, the new interactive debugger of BS2000, it is now possible to test programs in the comfortable BS2000 environment and transfer then to a VP system by simple recompilation (with different options).

All work related to the FORTRAN system was accomplished by employees of SIEMENS München.

A short example which is not representative can show some effects of vectorization against scalar execution depending on vector length.

Length of loop	FOR1 7.590	F77 7.590	F77/VP VP100
10	3	5.0	17
20	4.2	5.5	34
64	5.7	7.0	87
128	6.3	7.2	129
512	6.9	7.5	219
1024	6.9	7.6	230

All values in MFLOPS (Million Floating Point Operations per Second).

- Notes:**
1. FOR1 Version 1.6A was used, as the faster Version 2 was not available.
  2. The runtime system of FOR1 is faster than the migrated runtime V system of F77/VP. Results are too extensive to be presented here.



## 4. User training

The major part of our users who want to access the VP system are familiar with the BS2000 system.

We offer a User's Guide consisting of some 150 pages. This manual was written by members of the VP group at the University of Kaiserslautern and gives

- general information on vector computers,
- special information about VP hardware architecture,
- vectorization and optimization on FORTRAN level,
- introduction and examples to the USER2000 command interface.

The SDF syntax of USER 2000 commands allows the integration of user assistance that can be easily used in timesharing mode. By electing the appropriate level of guidance the users get most of the necessary assistance.

Users who do large projects get additional assistance by members of the VP group. This is necessary in error situations, for very high vectorization and special problems of resource shortage.

For external users, who have no knowledge on BS2000, we arrange introductions consisting of two lessons, each covering 90 minutes.

We observed that the vectorization information coming from the compiler and VECTUNE, together with the good vectorization technique of the compiler, lead to a very high vectorization rate of a user's program.

The average usage of the Vector Unit in the VP100 was 84 percent of total user CPU time during the first 6 months of this year.

We reached up to 92 percent of wall clock time as user CPU time over a period of one month.

Therefore we conclude that:

- our user support is sufficient and
- the VP system together with our access from BS2000 needs very little additional training compared to other supercomputers.

It is important to remark that in spite of the fact that VP's use a MVS compatible job control language, less than 10 percent of our users understand this cloudy and clumsy language.

## 5. Performance of File Transfer

For testing purposes an unformatted FORTRAN file was generated on a BS2000 public volume set, copied to a temporary file on BS2000 and then transferred to a VSP disk drive. The total amount of user data was 128 Mbytes, contained in 8192 records with increasing length from 1 to 8192 words. Disk type is 3480 and the CPU is a 7.590 model. With standard block length it size takes 17 seconds of CPU time and some 21 minutes of wall clock time to generate the file. It is evident that the huge elapsed time is related to the large number of 70 000 PAM pages to be written. The standard local /COPY in BS2000 takes some 5 CPU seconds and 430 seconds elapsed time. Almost all of the savings as well in CPU as in elapsed time are due to the usage of chained I/O. In this way 32 KBytes of information are transferred at a time. If we transfer the original file to a VSP disk, the format of record data has to be changed on the fly. In spite of this, CPU time is not twice the amount of generating the file as one would suggest, but rather 1.5 times of the generating time. The value of elapsed time is roughly the same as with generation of the file due to the fact that on the VP side all transports are performed on a track basis with 46 Kbytes at a time.

If we change the test and proceed to larger blocksize in BS2000, CPU time and elapsed time will decrease. The same is true for transfer to the VSP. The /COPY command will always take roughly the same CPU time and elapsed time, independent of the block size. The elapsed time is about two times the file generation time for 16\*STD as block size.

The elapsed time for transfer to VSP is better than a local /COPY when we use a large block size ( $\geq 8*STD$ ).

All measurements were done in parallel to normal work with better priority and not in a dedicated system.

BLKSIZE n*STD	CPU sec Generate	Elapsed sec Generate	CPU sec /COPY	Elapsed sec /COPY	CPU sec /T-V-F	Elapsed sec /T-V-F
1	16.9	1259	4.8	425	24.7	1264
2	12.2	672	5.4	431	17.2	688
4	8.9	412	5.5	416	13.4	427
8	7.3	280	5.5	412	11.6	288
12	6.8	224	5.4	424	10.8	248
16	6.5	203	5.4	423	10.1	215

Comparison of Write, COPY, TRANSFER-VECTOR-FILE

## 6. Performance of job transfer

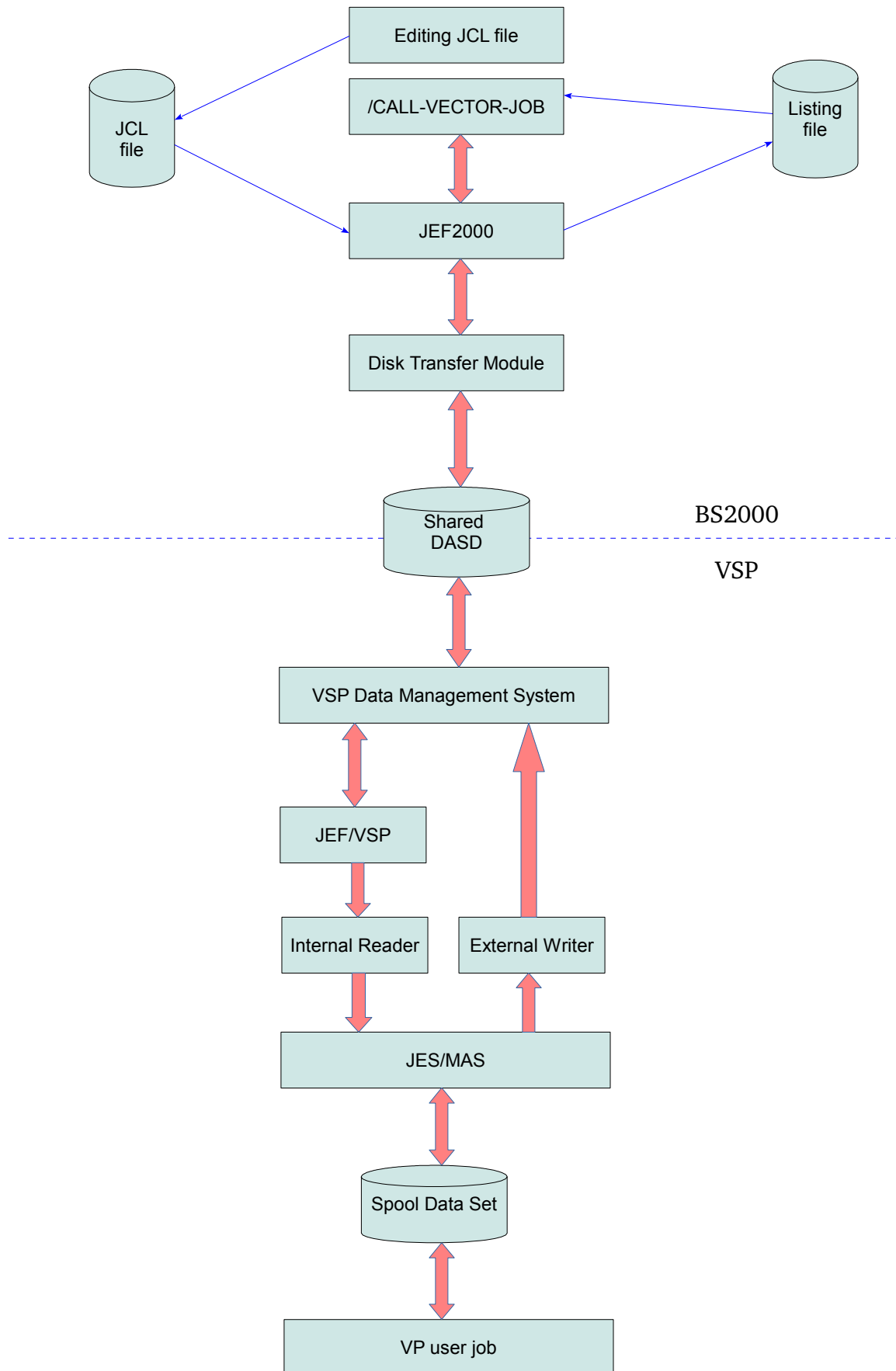
Job transfer, performed by the BS2000 and VSP component JEF, is not a CPU time consuming matter. Most of the elapsed time results from the different steps in BS2000 and VSP which are passed through.

An additional problem is the missing communication line between BS2000 and the VSP, that means: From the sending of a message to the receipt by the partner (both BS2000 and VSP), some idle time is spent. This idle state is interrupted by polling the shared DASD by the receiver, and up to this point the message cannot be processed.

Such a problem occurs twice: When sending the job to the VSP (polling by JEF/VSP) and when receiving the listing in BS2000 (polling by the corresponding JEF/BS2000).

Regarding a mini dummy job, we have measured a minimum of 11 seconds between entering the SDF command /CALL-VECTOR-JOB and the receipt of the listing in a BS2000 file. Nevertheless, these 11 seconds can be regarded as a good value especially under the aspect of the two problems mentioned above.

The following picture gives an overview of the data flow during the /CALL-VECTOR-JOB processing.



## 7. Overhead of VP2000

As we pointed out in the beginning, the connection between BS2000 and the VSP is realized by some jobs acting as agents. These main jobs like JEF, central file pager and disk transfer module don't take much CPU time. The total amount is less than 4 minutes per day.

In VSP the main job is the partner of JEF, and it takes less than 1 minute per day.

## **8. Conclusions**

The components of VP2000 build an environment for the BS2000 user that allows an easy to learn and easy to use access to VP systems.

They offer good tuning and debugging tools. They are well accepted by the users and they perform very well with a minimal overhead in resources and a small overhead in reaction from the users point of view.