

# J90 and VPP300: A Comparison of the Fortran90 Environments

Frank Schmitz, Forschungszentrum Karlsruhe, Germany

**ABSTRACT:** *The Fortran Programming Environment is the environment most frequently used at the Karlsruhe Research Centre. This paper displays the differences a Fortran90 user may find on two different kinds of Parallel Vector Computer: J90 and VPP300. Performance evaluation, availability of tools, and different solutions for parallel applications are the main topics of this presentation.*

*This paper is available online at [http://www.fzk.de/hikwww/J90\\_VPP300\\_Vortrag.html](http://www.fzk.de/hikwww/J90_VPP300_Vortrag.html) and [http://www.fzk.de/hikwww/J90\\_VPP300\\_Vortrag.ps](http://www.fzk.de/hikwww/J90_VPP300_Vortrag.ps)*

## Introduction

Using computer hardware seems to be very simple when you see people working everywhere at their PCs or workstations. But if they wish to run the first large program on PCs or workstations it will run for days. In our days it is very easy to change to something like a supercomputer. The program is running, but you achieve only a speed-up factor of two or less.

If users have chosen compiler options and tools for optimizing programs at their PCs or workstations, they can use similar things also on a supercomputer. But if they have not used tools before, they should be able to optimize the program first at their PCs or workstations. Maybe that there is no need to use a supercomputer.

The problem is, that after a user has started to develop a program, changing to another different hard- and software is in many cases suitable to solve problems. If one CPU is not fast enough to get the solution after an acceptable time, are there tools available to run the problem on more than one CPU?

At the 'Forschungszentrum Karlsruhe' a J916 and a VPP300/8 are installed. The compute server VPP300/8 is part of the 'Virtual Computing Centre' a project of cooperation between 'University of Karlsruhe' and 'Forschungszentrum Karlsruhe'. Another compute server of the 'Virtual Computing Centre' is an SP2 with 100 processors.

## Hardware

Cray J90 and Fujitsu VPP300 differ in architecture, performance and price, but both machines are vector computers which can solve big problems in parallel.

Figure 1 gives an overview of both architectures.

The most significant difference is that J90 is a global memory machine and VPP300 is a local memory machine which can be seen in Figure 2.

The differences in the hardwares of the machines, installed at the Computing Centre of Forschungszentrum Karlsruhe' are shown in Table 1.

If the memory is slow you have the problem to get a speedy answer, and if you are in communication with a non IEEE-754 hardware, something should be done to convert the data.

Application performance is the most interesting thing, if someone wants to get a compute server.

WAVEPACK solves the time-dependent Schrödinger equation in 2 dimensions, including non-diagonal mass-parameters.

Bandsolver is optimized for long vectors.

## Software

Vector computers have their own class of applications and use the hardware in an optimal way. This means that we should use only applications which are working always in vector mode.

But in many cases the users need programs which do not work in vector mode, for instance to debug, to tune and also to pre- and postprocessing data.

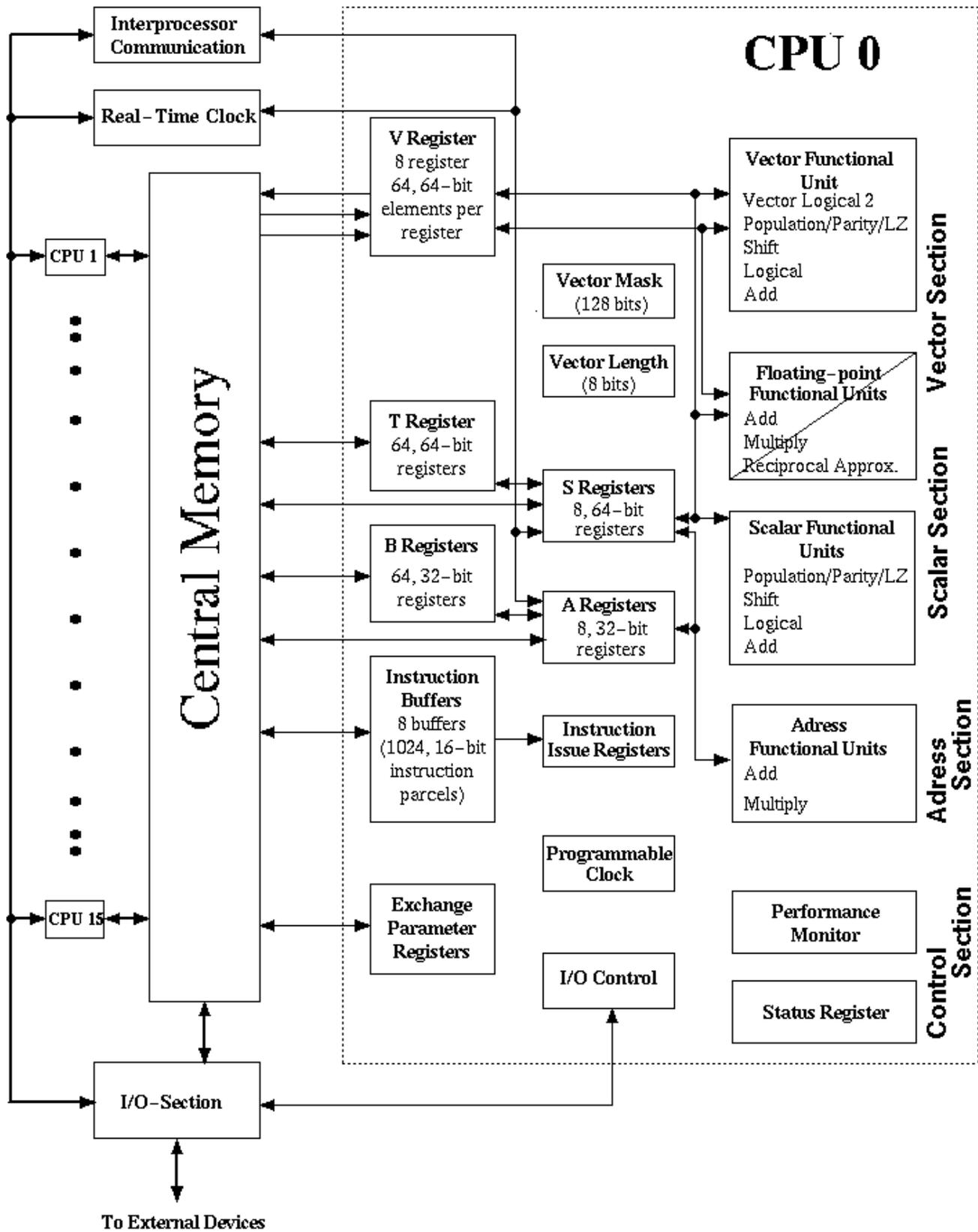


Figure 1: Description of J916

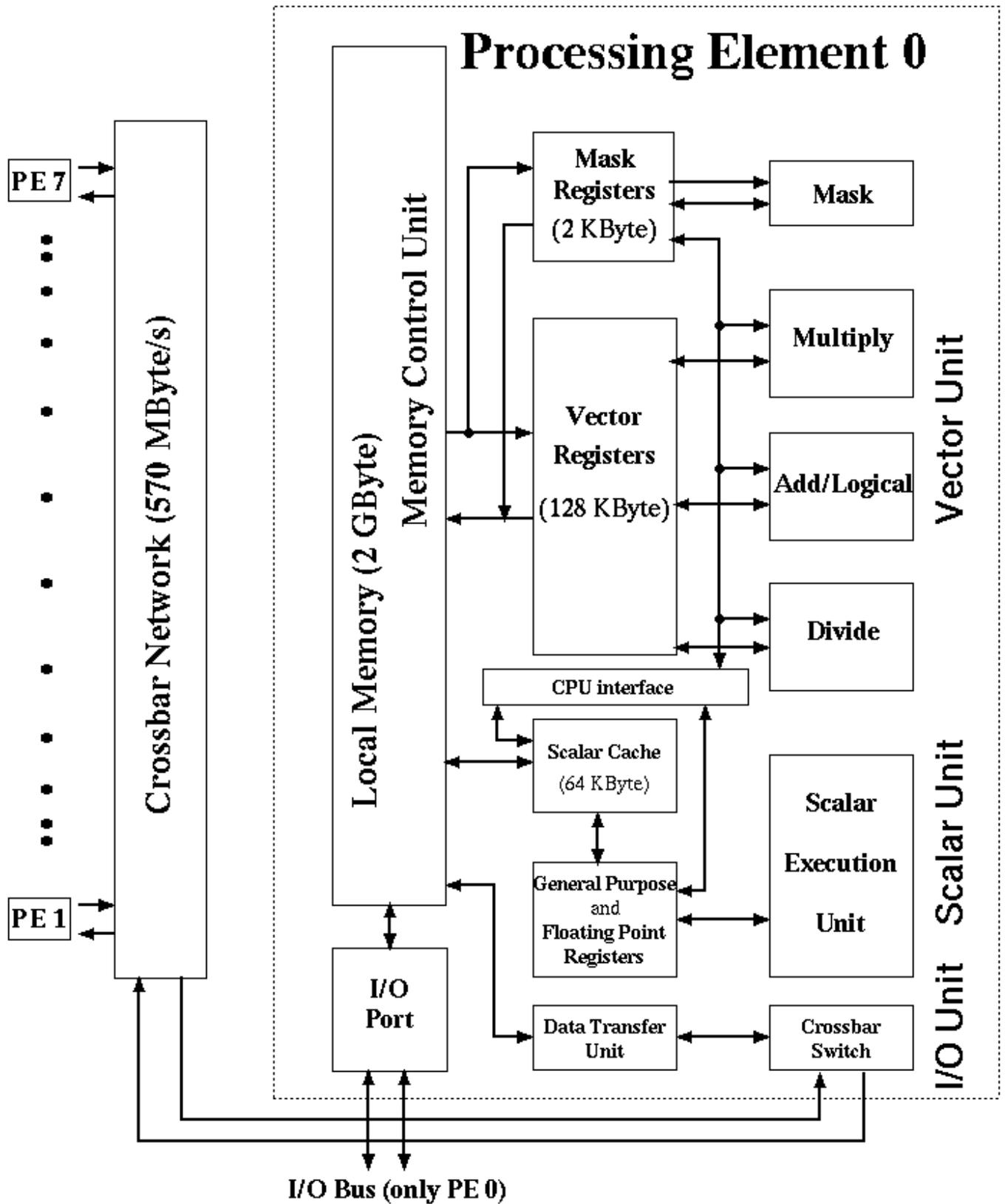


Figure 2. Description of VPP300/8

Table 1. Hardware comparison

CPU	VPP300	J916
Number of processors	8	16
Peak performance per processor	2.2 GFlops	200 MFlops
Cycle time	about 7 ns	10 ns
Vector register	128 KByte, reconfigurable	4 KByte
Vector length	64-2048	64
Arithmetical pipelines	addition, multiplication, division Two pipelines can work in parallel	addition, multiplication, reciprocal, the floating-point multiply functional unit is needed to find the reciprocal (Newton), i.e. at maximum two op- erations in parallel. Integer addition, integer multiplication

Table 2. Main memory

	VPP300	J916
Load/Store pipeline	1 Load, 1 Store 18.6 GByte/s	1 Load, 1 Load/Store 1.6 GByte/s
Main storage access	vector indirect, vector compress/expand, vector mask	stride, gather/scatter
Numeric representation	IEEE-754	Cray number representation
Main memory	2 GByte per prozessor Synchronous dynamic random access memory (SDRAM) at 60 ns access time	4 GByte shared memory' 70 ns CMOS DRAM

Table 3. Application performance comparison

	VPP300	J916
WAVEPACK (vector) RS6000-590: 5.28 s	0.70 s (527 MFlops)	3.54 s (100 MFlops)
Bandsolver (vector, scalar) RS6000-590: 6.52 s	0.61 s (817 MFlops), 32.04 s	5.64 s (86 MFlops), 62.41 s

\* Software available in the Fortran environment:

o Fortran90 compiler

+ J90: Cray CF90 Version 2.0.0.1

+ easy to use standard optimization compiler options:

-O {i, SCALARi, VECTORi, TASKi, INLINEi ; i=0,3}

+ extended optimization compiler options:

-O {AGRESS, BL, OVERINDEX, RECURRENCE, TASKINNER}

+ VPP300: Fujitsu Fortran90/VP Compiler Driver V10L10 L96031

+ standard optimization compiler options:

-O {n | b | e | f}

+ additional optimization compiler options:

- O {p, u, e, r, l, g}
- Ne
- K {auto, fast, VPP300}
- sc
- Wv,-{an | ae | ad},-s{c | v | l},-te,-Of,-q{s | m | v},
- G{e | f},-OP

o Tools

+ J90:

- + TotalView 2.0.0.4 is a special implementation of the BBN source-level debugger. For the user it provides an easy way to use the X Window System interface.
- + 'hpm' Version 80.7 monitors hardware performance during program execution.
- + Flowtrace generates timing and other information about procedure calls during execution.
- + 'flowview' 2.0.0.3 reports the data generated by the Flowtrace utility in line mode or by using the X Window System interface.
- + Perftrace collects hardware performance statistics by program unit.
- + 'perfview' 2.0.0.3 processes data generated by Perftrace. The utility can use the X Window System interface.
- + 'prof' indicates how much time was spent in various segments of the code within routines.
- + 'profview' 2.0.0.3 analyses data generated by the 'prof' command. The statistics can be shown in the X Window mode.

+ VPP300:

- + Analyzer V11L10 L96031 is a package that includes tools for program tuning, collecting hardware information, and debugging.
  - + 'xldb' (fdb V11.10) seems to be an X Window System interface to an old fashioned debugger.
  - + Sampler creates an information file, if the variable FJSAMP is exported before execution of the program is started. To analyse the information file, the program 'fjsamp' has to be started.
  - + PEPA (Processing Element Performance Analyser) collects events relating to a PE (not to a program). Only FJPEPA=ON should be exported
- + VPP-Workbench Version 1.0.1 and Workbench

o MPI

+ J90:

- + MPT 1.0.0.2, MPI model implementation 1.00.10., p4 layer (the implementation of MPI is based on MPICH, a public domain implementation developed by researchers at Argonne National Laboratory and Mississippi State University).
- + Starting environment for compile and run:
  - + . ENV\_START.ksh
  - + module load mpt
- + Compile a program: 'f90 -o matvec matvecmpi.f90'
- + Run a program: 'mpirun -np 4 -machinefile machine.CRAY'

```

matvec'
+ Results:
  Program started at: date: 19960918  time: 110910.409
  Result : 12502500. .... 12502500.
  Program ended at: date: 19960918  time: 110948.778
+ VPP300:
  + Beta-Version of the Fujitsu implementation of MPI.
  + Compiling and linking with many libraries: 'f90 -o
  matvec matvecmpi.f90 -L... -l...'
  + The program only works in the NQS-environment: 'matvec
  -np 4'
  + Results:
    Program started at: date: 19960924  time: 095659.580
    Result : 12502500.00000000 .... 12502500.00000000
    Program ended at: date: 19960924  time: 095738.950
  + If the Fortran90 intrinsic function 'cshift' is written
  in FORTRAN77 syntax, the result is:
    Program started at: date: 19960918  time: 112339.470
    Result : 12502500.00000000 .... 12502500.00000000
    Program ended at: date: 19960918  time: 112401.000
o PVM
+ J90:
  + MPT 1.0.0.2, PVM 3.3.7 (Cray Version 3.0.0.5)
  + Starting the environment for compile and run:
    + . ENV_START.ksh
    + module load mpt
  + Compile a program: 'f90 -o matvec matvecpvm.f90'
  + Run a program: 'pvmd3 hostfile &; matvec'
  + Results:
    Program started at: date: 19960918  time: 091619.584
    Result : 12502500. .... 12502500.
    Program ended at: date: 19960918  time: 091643.885
+ VPP300:
  + Optimized PVM Version 3.3
  + Compiling and linking with many libraries: 'f90 -o
  matvec matvecpvm.f90 -L... -l...'
  + The program only works in the NQS-environment: 'matvec'.
  + Results:
    Program started at: date: 19960930  time: 075114.000
    Result : 12502500.00000000 ..... 12502500.00000000
    Program ended at: date: 19960930  time: 075153.150
  + if the Fortran90 intrinsic function 'cshift' is written
  in FORTRAN77 syntax, the result is:
    Program started at: date: 19960930  time: 075456.260
    Result : 12502500.00000000 ..... 12502500.00000000
    Program ended at: date: 19960930  time: 075457.540
o Scientific libraries
+ J90:
  + Scientific libraries are grouped into the following
  sections:
    + Level 1,2,3 BLAS
    + LAPACK, LINPACK, EISPACK
    + FFT
+ VPP300:

```

- + SSL2 (Scientific Subroutine Library) V13L10 L95111 (linear algebra, eigenvalues and eigenvectors, nonlinear equations, extrema, interpolation and approximation, transforms; different routines are optimized).
- + Optimized level 1,2,3 BLAS written in FORTRAN77 (not Fujitsu).
- + FFTVPLIB (not Fujitsu)

## Performance

What do you understand by 'Performance'?

If anyone is testing or modifying large programs, the time of compilation is of significant interest. After testing, program performance is needed to get a short turn-around time.

The scalar performance and the optimization technique limit the speed of the compiler, but on the other side we need a highly optimizing compiler to speed up program execution.

The manufacturer has two possibilities: Compilation can be done on a special hardware (PC or workstation), or the scalar unit can be speeded up.

Compiling large programs on J90 takes a very long time and if you have changed 'include'-files (all routines should be compiled), the compilation has to be done in parallel to reduce the waiting time.

You have the same problem on VPP300, but only ad-hoc solutions may be conceivable at the moment to do compilation in parallel.

A compilation of 'matvec.f90' uses the following resources:

- J90: 0m11.40s (real), 0m6.91s (user), 0m0.79s (sys)
- VPP300: 0m3.86s (real), 0m2.83s (user), 0m0.23s (sys)

CPU-performance is of utmost interest, if testing is finished and you wish to run the production program. Programs with vector lengths >500 seem to be better installed on VPP300.

## Conclusion

Although VPP300 is poorly equipped with tools and not many third-party products are available, it is an interesting machine because the vector CPU performance, memory and efficiency of the compiler are commensurate with the costs of the machine.

- The problems which should be solved are:
- NFS performance between PE0 and the other PEs.
- State-of-the-art debugger.
- Reducing compilation time.

- Better scalar performance.
- HPF.
- MPI performance.
- DFS.
- Optimized intrinsic functions.

J90 is very well balanced in hardware and software, but the scalar applications should at least run twice as fast, because it may be very interesting to use J90 as a universal server as well and not only in vector computing. The following points are of special interest:

- IEEE data representation.
- Reducing compilation time.
- HPF.
- Improvement of MPI performance.
- Scalar performance

To get a user friendly hard- and software configuration a list of requirements of both hard- and software vendors, not only one written from a user's point of view, could look like as follows:

- Software development, testing, optimization and production runs should be possible just in one environment. The users should not be aware of the different systems in the background.
- The environment should be fast enough for programs tests and production runs.
- The users and the computing centre work with only one computing resource.
- Memory should be no problem for the user program.
- Reliable and usable tools are needed for serial and parallel applications.
- Costs ?

If you have questions or comments, please contact me (See Attendee List).