

ParaStation Version: 4.1.1
Date: 04/04/27
Document number: PS4.1.1-01en

Overview

ParaStation4 provides communication and management functions for Linux-based compute clusters. This Software Product Description documents the functionality available with ParaStation Version 4.1.1 as well as the system prerequisites required for installation and operation.

Technical features

ParaStation4 is divided into the modules

- Process Management (psmgmt-4),
- Communication (pscom-4),
- MPI (mpich-ps4),
- Documentation (ps-doc).

The 'Process Management' and 'Communication' modules are mandatory for operation.

Process management

The Process Management module is responsible for starting, monitoring, and terminating processes and entire applications in the cluster.

For this purpose, a Daemon process (*psid*) is executing on each compute node.

Load distribution, process placement: When starting a parallelized application, ParaStation evaluates the current node list, e.g. the list of currently available nodes and cpus on this nodes, the utilization of these nodes, the number of processes executing on the respective nodes, and possible user restrictions. Based on this current information a temporary node list is generated, in which the individual processes of a parallel application are started. The distribution of processes is based on adjustable criterias. If less cpus than processes are available, the processes are distributed to the temporary node list in round robin fashion.

The criteria listed for distribution can be influenced by the following environment variables:

- **PSI_NODES:** includes a list of node numbers to be taken into consideration.
- **PSI_HOSTS:** includes a list of node names to be taken into consideration.
- **PSI_HOSTFILE:** includes the name of a file with node names to be taken into consideration.
- **PSI_LOOP_NODES_FIRST:** if set, processes will be placed on different nodes first. Otherwise, processes will be placed within nodes first.
- **PSI_EXCLUSIVE:** if set, the processes will be placed on up to now unused nodes. It is also ensured, that further jobs will not be placed on this nodes.

The sort criteria for the temporary node list can be influenced by environment variable **PSI_NODES_SORT**. Currently available criteria: Round robin, load (1 min average), load (5 min average) load (15 min average), number of already executing processes started via ParaStation, and the combination of number of ParaStation processes and load (1 min average).

Nodes can be explicitly defines as starter nodes in ParaStation. On this node, which is typically a front-end computer, parallel applications can be started, but no compute processes are placed on it. In the same manner nodes can be configured as pure compute nodes, starting of applications is not possible on such nodes. By default, both types are permitted.

The maximum number of processes per node can be configured.

Parts of the compute cluster can be exclusively reserved for a user or user group. Only this user or user group member is allowed to start processes on the node reserved for him. All other users are mapped to the remaining nodes.

Parallel applications: Applications that were parallelized by calls to the ParaStation MPI Library are distributed to the available nodes according to the pattern described earlier. The start can be executed by calling the corresponding program with appropriate parameters (-np), or by means of calling the supplied *mpirun* program. The application data is transferred via the networks and protocols supported by ParaStation4, refer to section '*Supported networks and communication protocols*'.

Applications that were parallelized by means of alternative MPI libraries can be started via ParaStation, too. The distribution of processes is performed as described above. Communication is carried out via the corresponding mechanism implemented in the MPI library, e. g. TCP/IP. At present, the following MPI environments are supported:

- MPIch (with ch_p4),
- MPIch-GM.

The applications are started by means of corresponding *mpirun* commands, e. g. *mpirun_chp4* for applications which were linked by MPIch with ch_p4, or *mpirun_chgm* for applications linked with MPIch-GM.

Serial applications: Applications that were not parallelized by means MPI can be started via ParaStation, too. By calling *psmstart* an application is started on a suitable node according to the criteria mentioned above. The application itself can create new processes and threads on this node.

Process monitoring, process termination: Each process created by ParaStation on a compute node is permanently monitored. If one of these processes is terminated, e. g. because of a program error, or if a node is no longer available or accessible, all processes associated with this application on the respective nodes are terminated. All previously allocated resources will be freed. A corresponding value is passed to the calling process.

Executing processes can be listed by means of the *psadmin* command at any time. The individual processes are identified by a cluster-wide unique process ID. The system administrator or owner (starter) of an application can terminate the application by means of the *kill* function of the *psadmin* command at any time.

I/O forwarding: The default input (file descriptor 0) and default output (1) and default error output (2) of each process started are linked back to the calling process via the management network. ParaStation4 ensures that all outputs of all processes are collected centrally by a so-called logger process on the start node, independent of the node on which the processes are executing. Subsequently, the logger process forwards the outputs.

By default, the inputs on file descriptor 0 are sent to the first process, within MPI this process is assigned rank 0. The inputs can be assigned to another process by means of an environment variable.

If the starting process reads its default input from a (pseudo) terminal (pty), the I/O is routed via pseudo terminals for the processes started by ParaStation.

Signals: With the exception of SIGKILL and SIFSTOP, all signals to the logger process are forwarded to all processes executing in a distributed manner. When terminating an interactive application by using SIGTERM (^C), all processes belonging to this application are terminated on all nodes.

User administration: For application starts within the cluster ParaStation uses only user IDs, the names are only resolved on the start node. Therefore, users must be typically know on the front-end node, (complex) user management on the compute nodes is not necessary.

Batch systems: ParaStation4 supports the batch queuing systems LSF, PBS-Pro and OpenPBS. Corresponding environment variables are analyzed. Other systems could be typically integrated using prologue scripts.

Supported networks and communication protocols

At present, ParaStation4 supports communication using the pSPORT library via the following networks¹ or communication paths:

- Fast or Gigabit Ethernet (TCP/IP, also locally to the same node),
- Fast or Gigabit Ethernet (optimized ParaStation protocol, also locally to the same node),
- Shared memory (local communication on a node, specially for SMP systems),
- Local process (locally within a process).

The appropriate network is automatically selected for communication: shared memory for communication within SMP nodes, optimized ParaStation protocol for communication using Ethernet. If a protocol is not available on jobs startup, an other protocol will be selected automatically.

The communication networks and protocols to use can be defined upon job start. If several Ethernet-based networks are available, an environment variable can be used to select the interface for transmitting the data.

Via the underlying protocol the pSPORT library ensures secure and reliable communication between all participating processes. Errors are automatically detected and eliminated, e. g. through repeated transmission of data. The fragmentation and flow control implemented allows the transmission of data buffers of any size.

Communication libraries

libmpich.a: For the implementation of parallelized applications, ParaStation4 provides an MPI library based on MPIch², version 1.2.5. This library communicates with the PSPORT interface via an abstract device interface specially developed for ParaStation.

Small data buffers are directly transmitted to the receiving process, for a larger data volume a rendezvous procedure is used. The buffer size at which the rendezvous starts can be influenced by an environment variable.

The entire functionality implemented in MPIch version 1.2.5, including trace and debug options as well as asynchronous transmit and receive functions (*isend()*, *irecv()*, *iwait()*) is also available for ParaStation4. Libraries with wrappers functions (*libfmpich.a*, *libmichfarg.a*, *libpmpich++.a*, etc.), for example for Fortran, are contained.

libpsport.a: This library implements the PSPORT interface.

libpse.a: Implements process management functions.

libpsi.a: Implements further communication functions for process management.

All libraries are both installed as static (*.a*) and dynamic (*.so*) version. The following table explains the compilers and versions supported by ParaStation4 MPI:

¹ In future, support for Infiniband-based and Myrinet-based networks will be available.

² For further informationen about MPIch see <http://www-unix.mcs.anl.gov/mpi/mpich>.

Architecture:	IA32			
	Language:	Compiler		
		GNU	Intel (V7.0, V7.1)	Portland Group (V4.0)
	C	gcc 2.95	icc	pgcc
	C++	g++	icc	pgcc
	Fortran 77	f77 (gcc 2.95)	ifc	pgf77
	Fortran 90	-	ifc	pgf90
Architecture:	IA64			
	Language:	Compiler		
		GNU		
	C	gcc		
	C++	gcc		
	Fortran77	f77		
Architecture:	x86_64			
	Language:	Compiler		
		GNU		
	C	gcc		
	C++	gcc		
	Fortran77	f77		

ParaStation4 supports application threads, which have been parallelized by means of the Linux *pthread* libraries. It must be noted, however, that the MPI operations themselves are not thread-safe, e.g. that several threads cannot call MPI functions simultaneously.

Within ParaStation4 there are no limitations with respect to the number of processes per application and the number of simultaneously executing processes per compute node. The decisive factors in this case are the available system resources such as, for example, the size of the available memory and configured limits, like maximum number of processes per node.

Kernel module

The optimized ParaStation4 protocol for Ethernet is loaded as module into the Linux kernel. For the kernel versions listed in section 'Installation prerequisites' appropriate precompiled *p4sock.o* modules are contained in the *pscom-4* package.

In addition, a modified version of the e1000 driver (*e1000.o*, version 5.0.43) is part of ParaStation4. In conjunction with the module *e1000_glue.o*, even better results are achieved using Gigabit Ethernet. The source code of this modified driver will come with the *pscom-4* package.

ParaStation 4.1.1

User programs

ParaStation4 comprises the following commands and programs:

<i>Program:</i>	<i>Description:</i>
psid	ParaStation daemon
psld	ParaStation license daemon
psiadmin	central administration command, command-line-based
mlisten	diagnostic tool for multicasts
test_nodes	diagnostic tool for communication layer
mpirun	start program for parallel applications
mpirun_chp4	start program for applications parallelized by ch_p4
mpirun_chgm	start program for applications parallelized by MPIch-GM
psmstart	start program for serial applications
PSM_INSTALL	(de)installation program for ParaStation

Moreover, all tools available for MPIch such as, for example, *mpicc*, *mpif77*, *mpif90*, as well as the debugging and analysis tools are contained in a format adapted for the respective compiler.

License daemon

The llicense daemon is no longer available for this version of ParaStation. The *psid* will handle all licensing issues.

Documentation

Documentation is available as separate, installable package³. Documentation contained:

- ParaStation Users Guide (English, PDF and HTML format)
- ParaStation Administrator's Guide (English, PDF and HTML format)
- Manual pages with descriptions of ParaStation and MPIch commands, functions, and environment variables: (English, groff format)
- Description of library functions for ParaStation-specific libraries: libpsport, libpse, libmpe, libmpi (HTML and groff format, English).

Installation prerequisites

To install and operate ParaStation4, the following prerequisites must be met:

Hardware: ParaStation 4.1.1 supports the listed architectures, both as single und multiprocessor environment:

- Intel IA32 or AMD Athlon,
- Intel IA64 (Itanium),
- AMD x86_64 (Opteron); Note: currently as beta version.

Operating system: ParaStation 4.1.1 supports on IA32 all distributions based on gcc 2.95 and glibc 2.2.5, e. g. RedHat Linux 7.3 or SuSE 7.3. Packages for gcc3.2 based distributions, like RedHat 8.0,

³ The documentation is also available online in HTML format on the ParTec web server (<http://www.par-tec.com/documentation/ParaStation/html/index.html>).

are also available. On IA64, RedHat Advanced Server 2.1 will be supported, and on x86_64, UnitedLinux 1.0 is supported.

Network: For management tasks, a TCP/IP connection must be established between all compute nodes. ParaStation4 uses multicast messages. Therefore, all nodes must be configured in the same IP subnet. The (multicast) routing entries in the nodes and network switches must be made accordingly. The operating system must support the installed network hardware, corresponding low level drivers are also used by ParaStation4.

Management data and application data are principally separated and can be transmitted via different networks. For a list of supported data networks refer to section '*Supported networks and communication protocols*'.

Free disk space: The installation of all ParaStation4 packages requires approximately 50 MB free disk space per node. Administrator privileges are required for installation. The packages can be exported from a file sever via NFS to all compute clusters.

Kernel versions: The following Linux kernel versions are supported for IA32 at present:

<i>Version:</i>	<i>Description:</i>	<i>Kernel configuration:</i>
Linux_2.4.7.SuSE	SuSE	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.7-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.7-4GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.7-64GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=1
Linux_2.4.10.SuSE	SuSE	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.10-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.16.SuSE	SuSE	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.16-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.16-4GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-4GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0

<i>Version:</i>	<i>Description:</i>	<i>Kernel configuration:</i>
Linux_2.4.18-64GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=1
Linux_2.4.19.SuSE	SuSE	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.19-4GB.SuSE	SuSE	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.19-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.19-4GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.19-64GB-SMP.SuSE	SuSE	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=1
Linux_2.4.20-4GB-SMP	Vanilla	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.7-10smp.redhat	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-3smp.redhat	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-3-4GB-smp.redhat	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-14-64GB-smp.redhat	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=1
Linux_2.4.18-14-4GB.redhat	RedHat	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-27.8.0smp	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-27.7.xsmp.redhat	RedHat	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.9	Vanilla	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0

<i>Version:</i>	<i>Description:</i>	<i>Kernel configuration:</i>
Linux_2.4.9-SMP	Vanilla	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18	Vanilla	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-SMP	Vanilla	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=0 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-4GB-SMP	Vanilla	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-64GB-SMP	Vanilla	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=0 MODULE_CONFIG_HIGHMEM64G=1
Linux_2.4.18-5-4GB-SMP.deb	Debian	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.19-24-4GB-SMP.mdk	Mandrake	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-9-4GB-SMP.msc	MSCLinux	MODULE_CONFIG_SMP=1 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0
Linux_2.4.18-9-4GB.msc	MSCLinux	MODULE_CONFIG_SMP=0 MODULE_CONFIG_HIGHMEM=1 MODULE_CONFIG_HIGHMEM4G=1 MODULE_CONFIG_HIGHMEM64G=0

Media

All ParaStation modules are available in binary format as rpm or tar packages. The installation can take place either using the RedHat Packet Manager (rpm) or by unpacking the corresponding tar archives.

The installable packets are available in the download area of ParTec's web server www.par-tec.com and can be downloaded from this location.

License

A valid license is required for the operation of ParaStation4. This license must comprise at least all configured compute clusters and the sum of the physical processors installed in them. Virtual processors (hyper threading) need not be considered.

The license entitles to install and operate all ParaStation4 management components as well as all current and future networks supported by ParaStation4 and listed in the respective license.

In addition to that, the license conditions as defined in ParTec's standard terms and conditions shall apply.

Support

Free and unlimited support for all packages is granted for the period of one year after acceptance. The maximum response time is one working day. Support is performed by telephone, email, and/or remote login. Onsite support at the installation site is not included.

The support comprises all ParaStation components as well as the open source software utilized (if applicable). Other open source software tools that have been provided free of charge are only supported if resources are available, a general claim cannot be advanced on the basis of this support agreement.

The installation of updates when new ParaStation versions become available is not subject of the support agreement.

Update, new versions

A valid ParaStation4 license entitles to installation and operation of all available subversions, e.g. all ParaStation4 software packages with identical top-level version number. This applies especially to new functionality and extensions. These are identified by a new subversion number, e. g. version 4.1.0.

Scope of delivery

ParaStation4 comprises the following components:

- License document,
- License in electronic format (for installation in the system),
- Software packages in the download area of ParTec's web site,
- Documentation (ParaStation Users' Guide and ParaStation Administrator's Guide) in electronic format,
- One-year support.

Copyright

ParTec and ParaStation are registered trademarks of ParTec AG. All other product and brand names are trademarks or registered trademarks of their respective owners.

The information in this version of the software product detailed description is valid as from the time of publishing. Errors & omissions excluded.

Further information

For further information about ParaStation4 take a look at <http://www.par-tec.com> or send an email to sales@par-tec.com.