Distributed ANSYS Guide
Copyright and Trademark Information

© 2009 SAS IP, Inc. All rights reserved. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, Ansoft, AUTODYN, EKM, Engineering Knowledge Manager, CFX, FLUENT, HFSS and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.
ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.
Table of Contents

1. Overview of Distributed ANSYS ................................................................. 1
2. Configuring and Starting Distributed ANSYS ................................................. 5
   2.1. Prerequisites for Running Distributed ANSYS ....................................... 5
   2.1.1. MPI Software .................................................. 5
   2.1.2. ANSYS Software .............................................. 7
   2.2. Setting Up the Environment for Distributed ANSYS .............................. 8
   2.2.1. Optional Setup Tasks ........................................... 10
   2.2.2. Using the mpitest Program ........................................... 11
   2.2.3. Interconnect Configuration .......................................... 12
   2.2.4. Other Considerations .............................................. 13
   2.3. Starting Distributed ANSYS ....................................................... 13
   2.3.1. Starting Distributed ANSYS via the Launcher .................................. 14
   2.3.2. Starting Distributed ANSYS via Command Line ............................... 15
   2.3.3. Starting Distributed ANSYS via the HPC Job Manager ....................... 16
   2.3.4. Starting Distributed ANSYS in ANSYS Workbench ............................ 17
   2.3.5. Using MPI appfiles ................................................ 17
   2.3.6. Files that Distributed ANSYS Writes ........................................... 18
3. Running Distributed ANSYS ................................................................. 19
   3.1. Supported Analysis Types ....................................................... 19
   3.2. Supported Features ............................................................ 20
   3.3. Running a Distributed Analysis .............................................. 21
   3.4. Restarts in Distributed ANSYS ................................................ 22
   3.5. Understanding the Working Principles and Behavior of Distributed ANSYS ........ 23
   3.6. Example Problems ........................................................... 25
   3.6.1. Example: Running Distributed ANSYS on Linux .............................. 26
   3.6.2. Example: Running Distributed ANSYS on Windows .......................... 28
   3.7. Troubleshooting .............................................................. 29
Index ......................................................................................... 35

List of Tables

1.1. Solver Availability in Shared-Memory and Distributed ANSYS .................. 2
2.1. Platforms and MPI Software .......................................................... 6
2.2. LS-DYNA MPP MPI Support on Windows and Linux .............................. 6
3.1. Required Files for Singleframe Restarts .............................................. 23
3.2. Required Files for Multiframe Restarts .............................................. 23
# Chapter 1: Overview of Distributed ANSYS

Solving a large model with millions of DOFs or a medium-sized model with many iterations can require many CPU hours. To decrease processing time, ANSYS offers different options for distributing model-solving power using multiple processors (also known as cores).

It is important to fully understand the terms we use, both relating to our software and to the physical hardware. The terms Distributed ANSYS and shared-memory ANSYS refer to our software offerings, which run on shared memory or distributed hardware configurations (described below).

Both Distributed ANSYS and shared-memory ANSYS require HPC licenses. Distributed ANSYS and shared-memory ANSYS allow you to use two processors without using any HPC licenses. Additional licenses will be needed to run with more than two processors. Several HPC license options are available. See *HPC Licensing for Distributed ANSYS* (p. 3) for more information.

<table>
<thead>
<tr>
<th>Shared-memory ANSYS</th>
<th>Distributed ANSYS</th>
<th>Shared Memory Environment</th>
<th>Distributed Memory Environment</th>
<th>Mixed Memory Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared-memory ANSYS refers to running ANSYS across multiple cores on a single machine. Most, but not all, of the solution phase runs in parallel when using the shared-memory architecture. Many solvers in ANSYS can use the shared-memory architecture. In addition, pre- and postprocessing can make use of the multiple cores, including graphics operations and other data and compute intensive operations. For more information on running shared-memory ANSYS, see Using Shared-Memory ANSYS of the Advanced Analysis Techniques Guide.</td>
<td>With Distributed ANSYS, the entire /SOLUTION phase runs in parallel, including the stiffness matrix generation, linear equation solving, and results calculations. With multiple cores, you can see significant speedup in the time it takes to run your analysis. This document describes using Distributed ANSYS.</td>
<td>Shared memory environment refers to the physical hardware configuration in which a single shared memory address space is accessible by all cores. Therefore each CPU &quot;shares&quot; the memory with the others. A common example of a shared memory system would be a Windows machine with dual cores. Both cores share the same main memory space through a common bus architecture.</td>
<td>In a distributed memory environment, each CPU or computing node has its own memory address space which is not shared by other CPUs or computing nodes. Communication between machines is by MPI (Message Passing Interface) software on the network. A common example of a distributed memory system would be any collection of desktop workstations linked by a network. When the collection of linked machines is a dedicated computing engine for a given period of time and is not used during that time for everyday tasks such as email or browsing, it is called a cluster.</td>
<td>Mixed memory indicates that the cluster is using a combination of both shared and distributed memory. A common example of a mixed memory system would be a cluster of CPUs with two cores in each physical box sharing that...</td>
</tr>
</tbody>
</table>
memory, but with a number of these units connected to each other by a network. The Distributed ANSYS solvers support the mixed memory scenario by treating the shared memory as if it were distributed memory. The AMG solver does not support the mixed memory scenario as it is a shared-memory only solver.

The following table shows how the various solvers, element formulation, and results calculation behave under Distributed ANSYS and shared-memory ANSYS. You can run Distributed ANSYS in either distributed parallel mode (across multiple machines) or in shared-memory parallel mode (using multiple cores on a single machine). However, while both modes will see a speedup in CPU time, only distributed parallel mode will allow you to take advantage of increased memory availability across the multiple machines.

Table 1.1 Solver Availability in Shared-Memory and Distributed ANSYS

<table>
<thead>
<tr>
<th>Solvers/Feature</th>
<th>Shared-Memory</th>
<th>Distributed ANSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCG</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>JCG</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Distributed sparse</td>
<td>--</td>
<td>Y</td>
</tr>
<tr>
<td>AMG</td>
<td>Y</td>
<td>--</td>
</tr>
<tr>
<td>Sparse</td>
<td>Y</td>
<td>--</td>
</tr>
<tr>
<td>ICCG</td>
<td>Y</td>
<td>--</td>
</tr>
<tr>
<td>Element formulation, results</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>calculation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphics and other pre- and</td>
<td>Y</td>
<td>Y[1]</td>
</tr>
<tr>
<td>postprocessing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. If running Distributed ANSYS on a shared-memory machine with multiple cores as the master or host machine.

In ANSYS, the solution time is typically dominated by three parts: the time spent to create the element matrices and form the global matrices or global systems of equations, the time to solve the linear system of equations, and the time spent post-processing the solution (i.e., calculating derived quantities such as stress and strain).

Shared-memory architecture runs a solution over multiple cores on a single, multicore machine. When using shared-memory ANSYS, you can reduce each of the three main parts of the overall solution time by using multiple cores. However, this architecture is limited by the memory bandwidth; you typically see very little reduction in solution time beyond two to four cores.

The distributed-memory architecture of Distributed ANSYS runs a solution over multiple cores on a single machine or on multiple machines. It decomposes large problems into smaller domains, transfers the domains to each core, solves each domain, and creates a complete solution to the model. Because each of the three main parts of the overall solution time are running in parallel, the whole model solution time is significantly reduced. The memory required is also distributed over multiple systems. This memory-distribution method allows you to solve very large problems on a cluster of machines with limited memory.

Distributed ANSYS works by launching ANSYS on multiple machines. The machine that ANSYS is launched on is called the master or host machine and the other machines are called the slave machines. (If you launch Distributed ANSYS using multiple cores on the same machine, you would use the same terminology when referring to each process, e.g., the master and slave processes). All pre-processing and post-processing
commands are executed only on the master machine. Only the SOLVE command and any necessary supporting commands (e.g., \texttt{SOLU}, \texttt{FINISH}, \texttt{EOF}, \texttt{EXIT}, etc.) are sent to the slave machines to be processed.

The maximum number of cores allowed in a Distributed ANSYS analysis is currently set at 1024. Therefore, you can run Distributed ANSYS using anywhere from 2 to 1024 cores (assuming the appropriate HPC licenses are available) for each individual job.

Files generated by Distributed ANSYS are named \texttt{JobnameN.ext}, where \(N\) is the process number. The master process is always 0, and the slave processes are 1, 2, etc. When the solution is complete and you issue the \texttt{FINISH} command in \texttt{SOLU}, Distributed ANSYS combines all \texttt{JobnameN.rst} files into a single \texttt{Jobname.rst} file, located on the master machine.

The remaining chapters explain how to configure your environment to run Distributed ANSYS, how to run a Distributed ANSYS analysis, and what features and analysis types are supported in Distributed ANSYS. You should read these chapters carefully and fully understand the process before attempting to run a distributed analysis. The proper configuration of your environment and the installation and configuration of the appropriate MPI software are critical to successfully running a distributed analysis.

\textbf{ANSYS LS-DYNA} If you are running ANSYS LS-DYNA, you can use LS-DYNA’s parallel processing (MPP or SMP) capabilities. Use the launcher method or command line method as described in Chapter 2, Configuring and Starting Distributed ANSYS (p. 5) to run LS-DYNA MPP. Also see LS-DYNA Parallel Processing Capabilities in the \textit{ANSYS LS-DYNA User’s Guide} for more information on both the SMP and MPP capabilities. You will need an ANSYS LS-DYNA Parallel license for every processor beyond the first one.

\textbf{HPC Licensing for Distributed ANSYS}

ANSYS, Inc. offers the following high performance computing license options:

- \textbf{ANSYS HPC} - These physics-neutral licenses can be used to run a single analysis across multiple processors and work with numerous applications, including Distributed ANSYS.
- \textbf{ANSYS HPC Packs} - These physics-neutral licenses share the same characteristics of the ANSYS HPC licenses, but are combined into predefined packs to give you greater value and scalability.
- \textbf{Physics-Specific Licenses} – Legacy physics-specific licenses are available for various applications. The physics-specific license for Distributed ANSYS is the ANSYS Mechanical HPC license.

For detailed information on these HPC license options, see \textit{HPC Licensing} in the \textit{ANSYS, Inc. Licensing Guide}.

The HPC license options cannot be combined with each other in a single solution; for example, you cannot use both ANSYS HPC and ANSYS HPC Packs in the same analysis solution.

The order in which HPC licenses are used is specified by your user license preferences setting. See \textit{Specifying HPC License Order} in the \textit{ANSYS, Inc. Licensing Guide} for more information on setting user license preferences.

Distributed ANSYS allows you to use two processors without using any HPC licenses. ANSYS HPC licenses and ANSYS Mechanical HPC licenses add processors to this base functionality, while the ANSYS HPC Pack licenses function independently of the two included processors.

The HPC license options described here do not apply to ANSYS LS-DYNA; see the \textit{ANSYS LS-DYNA User’s Guide} for details on parallel processing options with ANSYS ANSYS LS-DYNA.
Chapter 2: Configuring and Starting Distributed ANSYS

Before running a Distributed ANSYS analysis, you must configure your system properly. Your system must meet specified criteria. You must have supported MPI software correctly installed, and the machines running Distributed ANSYS must be properly configured.

The following topics are available for configuring and starting Distributed ANSYS:

- 2.1. Prerequisites for Running Distributed ANSYS
- 2.2. Setting Up the Environment for Distributed ANSYS
- 2.3. Starting Distributed ANSYS

2.1. Prerequisites for Running Distributed ANSYS

Your system must meet the following requirements to run Distributed ANSYS.

- Homogeneous network: All machines must be the same type, OS level, chip set, and interconnects.
- You must be able to remotely log in to all machines, and all machines in the cluster must have identical directory structures (including the ANSYS installation, MPI installation, and on some systems, working directories). Do not change or rename directories after you've launched ANSYS. For more information on files that are written and their location, see Files that Distributed ANSYS Writes in the Distributed ANSYS Guide.
- All machines in the cluster must have ANSYS installed, or must have an NFS mount to the ANSYS installation. If not installed on a shared file system, ANSYS must be installed in the same directory path on all systems.
- Distributed ANSYS allows you to use two processors without using any HPC licenses. Additional licenses will be needed to run Distributed ANSYS with more than two processors. Several HPC license options are available. See HPC Licensing for Distributed ANSYS for more information.
- All machines must have the same version of MPI software installed and running. The table below shows the MPI software and version level supported for each platform.

If you plan to use only the AMG solver in shared-memory ANSYS, MPI software is not required. If you are using only the AMG solver, skip the rest of this document and continue with Activating Parallel Processing in a Shared-Memory Architecture in the Advanced Analysis Techniques Guide.

2.1.1. MPI Software

The MPI software you use depends on the platform. The following table lists the type of MPI software required for each platform. HP-MPI and Intel MPI are included on the Linux installation media and are installed automatically when you install ANSYS. Similarly, HP-MPI is included on the UNIX installation media and is installed automatically when you install ANSYS on an HP machine. For questions regarding the installation and configuration of the native MPI software versions, please contact your MPI or hardware vendor.

Distributed ANSYS runs on the following platforms:

- HP PA8000 / HP IA-64 (native MPI)
- Intel IA-32 Linux (HP-MPI, Intel MPI)
Intel IA-64 Linux (HP-MPI, Intel MPI)
SGI Altix 64-bit Linux (MPI MPT)
AMD Opteron 64-bit Linux (HP-MPI, Intel MPI)
Intel Xeon EM64T 64-bit Linux (HP-MPI, Intel MPI)
Sun - single box, multiple processor only (native MPI only)
IBM - single box, multiple processor only (native MPI only)
Windows 32-bit (HP-MPI, Intel MPI)
Windows 64-bit (HP-MPI, MS MPI, Intel MPI)

Table 2.1 Platforms and MPI Software

<table>
<thead>
<tr>
<th>Platform</th>
<th>MPI Software</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP PA8000 64-bit / HP-UX 11.11 (64-bit)</td>
<td>HP-MPI 2.2.5</td>
<td><a href="http://docs.hp.com/en/B6060-96023/index.html?jumpid=reg_R1002_USEN">http://docs.hp.com/en/B6060-96023/index.html?jumpid=reg_R1002_USEN</a></td>
</tr>
<tr>
<td>HP IA-64 / HP-UX 11.23</td>
<td>HP-MPI 2.2.5</td>
<td><a href="http://docs.hp.com/en/B6060-96023/index.html?jumpid=reg_R1002_USEN">http://docs.hp.com/en/B6060-96023/index.html?jumpid=reg_R1002_USEN</a></td>
</tr>
<tr>
<td>IBM AIX64 64-bit / AIX 5.3</td>
<td>POE 4.3.2.6</td>
<td>Contact your IBM vendor. Earlier or later versions of POE may not work correctly with Distributed ANSYS.</td>
</tr>
</tbody>
</table>
| Linux Intel (including EM64T) and AMD (32- and 64-bit) | HP-MPI 2.3.1 Intel MPI 3.2.1 | HP-MPI: http://docs.hp.com/en/T1919-90018/T1919-90018.pdf
| Windows 64-bit / Windows XP x64 / Windows Vista x64 | Microsoft HPC Pack (MS MPI) | http://www.microsoft.com/hpc/ |

ANSYS LS-DYNA  If you are running ANSYS LS-DYNA, you can use LS-DYNA's parallel processing (MPP or SMP) capabilities. Use the launcher or the command line method as described in Starting Distributed ANSYS in the Distributed ANSYS Guide to run LS-DYNA MPP. To run LS-DYNA's MPP version with ANSYS, you need to use the MPI versions listed above for HP and IBM systems. Note that MPP is not available for Sun Solaris systems. For Windows and Linux systems, please see the following table for LS-DYNA MPP MPI support. For more information on using ANSYS LS-DYNA in general, and its parallel processing capabilities specifically, see the ANSYS LS-DYNA User's Guide.

Table 2.2 LS-DYNA MPP MPI Support on Windows and Linux

<table>
<thead>
<tr>
<th>MPI version for DYNA MPP</th>
<th>32-bit Windows</th>
<th>64-bit Windows</th>
<th>32-bit Linux</th>
<th>64-bit Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP-MPI</td>
<td>n/a</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
2.1.2. ANSYS Software

To run Distributed ANSYS, you must install ANSYS on all machines in the cluster, or have an NFS mount to the ANSYS installation. Install ANSYS following the instructions in the ANSYS, Inc. Installation Guide for your platform. Be sure to complete the installation, including all required post-installation procedures. On Windows systems, you must use the Universal Naming Convention (UNC) for all file and path names for Distributed ANSYS to work correctly.

To run Distributed ANSYS on more than two processors, you must have a sufficient number of HPC licenses to accommodate the desired number of processors. See HPC Licensing for Distributed ANSYS for more information.

Installing HP-MPI on Windows

You can install HP-MPI from the Installation launcher by choosing Install MPI for Distributed Mechanical APDL (ANSYS). On the next screen, select Install HP-MPI for Distributed Mechanical APDL (ANSYS). The HP-MPI installation program will start. An HP-MPI installation README file will open simultaneously. Follow the instructions in the README file as you complete the HP-MPI installation.

The instructions for installing HP-MPI are also found on the ANSYS distribution media in the following README files:

Program Files\Ansys Inc\V121\ANSYS\HP-MPI\INSTALL_HP-MPI_README.mht
or
Program Files\Ansys Inc\V121\ANSYS\HP-MPI\INSTALL_HP-MPI_README.doc

Installing Intel MPI on Windows

You can install Intel MPI from the Installation launcher by choosing Install MPI for Distributed Mechanical APDL (ANSYS). On the next screen, select Install Intel MPI for Distributed Mechanical APDL (ANSYS). The Intel MPI installation program will start. An Intel MPI installation README file will open simultaneously. Follow the instructions in the README file as you complete the Intel MPI installation.

The instructions for installing Intel MPI are also found on the ANSYS distribution media in the following README files:

Program Files\Ansys Inc\V121\ANSYS\Intel-MPI\INSTALL_Intel-MPI_README.mht
or
Program Files\Ansys Inc\V121\ANSYS\Intel-MPI\INSTALL_Intel-MPI_README.doc

Microsoft HPC Pack (Windows HPC Server 2008)

You must complete certain post-installation steps before running Distributed ANSYS on a Microsoft HPC Server 2008 system. The post-installation instructions provided below assume that Microsoft HPC Server 2008 and Microsoft HPC Pack (which includes MS MPI) are already installed on your system. The post-installation instructions can be found in the following README files:
Microsoft HPC Pack examples are also located in Program Files\Ansys Inc\V121\ANSYS\WinHPC. Jobs are submitted to the Microsoft HPC Job Manager either from the command line or the Job Manager GUI.

To submit a job via the GUI, go to Start> All Programs> Microsoft HPC Pack> HPC Job Manager. Then click on Create New Job from Description File.

### 2.2. Setting Up the Environment for Distributed ANSYS

After you've ensured that your cluster meets the prerequisites and you have ANSYS and the correct version of MPI installed, you need to configure your distributed environment using the following procedure.

1. Obtain the machine name for each machine on the cluster.

   **Windows:**
   - Right-click on My Computer, left-click on Properties, and select the Network Identification or Computer Name tab. The full computer name will be listed. Note the name of each machine (not including the domain).

   **UNIX/Linux:**
   - Type hostname on each machine in the cluster. Note the name of each machine. You will need this name to set up the .rhosts file, as well as for the ANS_ADMIN utility.

2. (UNIX/Linux only) Set up the .rhosts file on each machine. The .rhosts file lists all machines in the cluster. The machines should be listed using their complete system name, as taken from hostname. For example, an .rhosts file for a two-machine cluster might look like this:

   ```
golinux1.ansys.com jqd
golinux2 jqd
   ```

   Change/verify .rhosts file permissions on all machines by issuing:

   ```
   chmod 600 .rhosts
   ```

   Verify communication between machines via rsh or ssh (e.g., rsh golinux2 ls). You should not be prompted for a password. If you are, check the .rhosts permissions and machine names for correctness. For more information on using remote shells, see the man pages for rsh or ssh.

3. If you want the list of machines to be populated in the Mechanical APDL Product Launcher, you need to configure the hosts121.ans file. You can use the ANS_ADMIN utility to configure this file. You can manually modify the file later, but we strongly recommend that you use ANS_ADMIN to create this file initially to ensure that you establish the correct format.

   **Windows:**
   - Start>Programs >ANSYS 12.1 >Utilities >ANS_ADMIN

     Choose Configuration options, and then Configure Cluster to configure the hosts121.ans file.
1. Specify the directory in which the hosts121.ans will be configured: Select the **Configure a hosts121.ans file in a directory you specify** option and click **OK**. Enter a working directory. Click **OK**.

2. Enter the system name (from Step 1) in the **Machine hostname** field and click **Add**. On the next dialog box, enter the system type in the **Machine type** drop-down, and the number of processors in the **Max number of jobs/processors** field and click **OK** for each machine in the cluster. When you are finished adding machines, click **Close**, then **Exit**.

An example hosts121.ans file where machine1 has 2 processors and machine2 has 4 processors would look like this:

```
machine1 intel 0 2 0 0 MPI 1 1
machine2 intel 0 4 0 0 MPI 1 1
```

**UNIX/Linux:**

```
/ansys_inc/v121/ansys/bin/ans_admin121
```

Choose **ANSYS/Workbench Configuration**, and then click **Configure Cluster**. Under **Select file to configure**, choose the hosts121.ans file to be configured and choose **Configure for Distributed ANSYS**. Click **OK**. Then enter the system name (from Step 1) in the **Machine hostname** field and click **Add**. On the next dialog box, enter the system type in the **Machine type** drop-down, and the number of processors in the **Max number of jobs/processors** field for each machine in the cluster. Click **Add**. When you are finished adding machines, click **Close**.

The hosts121.ans should be located in your current working directory, your home directory, or the apdl directory.

4. (Windows only) Add `%MPI_ROOT%\bin` to the **PATH** environmental variable on all Windows machines. This line must be in your path for distributed processing to work correctly.

5. (Windows only) Setting up environment variables.

On the head node, where ANSYS and ANSYS Licensing is installed, set the following:

```java
ANSYS121_DIR=C:\Program Files\ANSYS Inc\v121\ansys
ANSYSLIC_DIR=C:\Program Files\ANSYS Inc\Shared Files\Licensing
```

where `C:\Program Files\ANSYS Inc` is the location of the product install. If your installation location is different than this, specify it instead.

On Windows systems, you must use the Universal Naming Convention (UNC) for all ANSYS environment variables on the compute nodes for Distributed ANSYS to work correctly.

On the compute nodes, set the following:

```java
ANSYS121_DIR=\head_node_machine_name\ANSYS Inc\v121\ansys
ANSYSLIC_DIR=\head_node_machine_name\ANSYS Inc\Shared Files\Licensing
```

For Distributed LS-DYNA:

On the head node and the compute nodes, set **LSTC_LICENSE** to ANSYS. This tells the LS-DYNA executable to use ANSYS licensing.
Since the LS-DYNA run will use ANSYS licensing for LS-DYNA, you do not need to set LSTC_LICENSE_SERVER.

6. (Windows only) Share out the ANSYS Inc directory on the head node with full permissions so that the compute nodes can access it.

### 2.2.1. Optional Setup Tasks

The tasks explained in this section are optional. They are not required to get Distributed ANSYS to run correctly, but they may be useful for achieving the most usability and efficiency, depending on your system configuration.

- On UNIX/Linux systems, you can also set the following environment variables:
  - **ANSYS_NETWORK_START** - This is the time, in seconds, to wait before timing out on the start-up of the client (default is 15 seconds).
  - **ANSYS_NETWORK_COMM** - This is the time to wait, in seconds, before timing out while communicating with the client machine (default is 5 seconds).
  - **ANSSEE_RUN_COMMAND** - Set this ANSYS environment variable to 1 to display the actual mpirun command issued from ANSYS.

On HP or Linux systems running HP-MPI:

- **MPI_REMSH** - This is the path to the remote shell (ssh or rsh). Set this environment variable to specify a full path to a remote shell. For example, setting `MPI_REMSH = /usr/bin/ssh` will use ssh instead of the default remote shell (rsh). Note that selecting the Use Secure Shell instead of Remote Shell option on the launcher will override `MPI_REMSH`, if `MPI_REMSH` is not set or is set to a different location. You can also issue the `– usessh` command line option to use ssh instead of rsh. The command line option will override the environment variable setting as well.
  - **MPI_WORKDIR** - Set this environment variable to specify a working directory on either the master and all nodes, or on specific nodes individually. For more information, see Files that Distributed ANSYS Writes.
  - **MPI_IC_ORDER** - Set this environment variable to specify the order in which the interconnects on the system are to be used. The interconnects will be tried in the order listed from left to right. If an interconnect is listed in uppercase, no interconnects listed after that one will be tried. If `MPI_IC_ORDER` is not set, the fastest interconnect available on the system is used. See the HP-MPI documentation for more details.
  - **MPI_ICLIB_<interconnect>** - Set this environment variable to the interconnect location if the interconnect is not installed in the default location, especially on AMD Opteron systems using the GM (Myrinet) interconnect:

```bash
setenv MPI_ICLIB_GM <path>/lib64/libgm.so
```

See the HP-MPI documentation for the specific interconnect names (e.g., `MPI_ICLIB_GM`).

- **MPIRUN_OPTIONS** - Set this environment variable to -prot to display a grid of interconnects among the systems being used for distributed processing.

- **ANS_HPMPI_SRUN_OPTIONS** - Use this environment variable to define the -n (specifies the number of processes and assigns ranks in a block allocation) and -N (specifies the number of nodes allocated to the job and assigns ranks in a cyclic allocation) values for -srun. For example, to specify two machines with two processes on each node, you would set this environment variable as follows:

```bash
setenv ANS_HPMPI_SRUN_OPTIONS "-n4 -N2"
```
You cannot specify -sr un or its options directly on the Distributed ANSYS command line. Valid only for HP XC clusters. See the HP-MPI User's Guide for details on -sr un.

- **ANS_HPMPI_PRUN_OPTIONS** - Use this environment variable to define the -n (specifies the number of processes and assigns ranks in a block allocation) and -N (specifies the number of nodes allocated to the job and assigns ranks in a cyclic allocation) values for -pr un. For example, to specify two machines with two processes on each node, you would set this environment variable as follows:

  ```bash
  setenv ANS_HPMPI_PRUN_OPTIONS "-n4 -N2"
  ```

You cannot specify -pr un or its options directly on the Distributed ANSYS command line. Valid only for MPI on an Elan (Quadrics) interconnect. See the HP-MPI User's Guide for details on -sr un.

If you set the **ANS_HPMPI_PRUN_OPTIONS** environment variable and also specify -np on the Distributed ANSYS command line, the environment variable will override the -np setting. If you have -pr un on your system and you use the -np command line option but do not set this environment variable, Distributed ANSYS will use the specified number of processors on a single machine (i.e., shared-memory parallel mode). If the number of processors specified with -np exceeds the number of processors available on a single machine, Distributed ANSYS will then use multiple machines connected via the Elan interconnect until the specified number of processors is reached.

When running jobs on HP XC clusters (-sr un) or clusters that use the Quadrics interconnect (-pr un), ANSYS highly recommends that you run from jobs in a writable directory that is visible from all hosts to ensure proper location and execution of files for your job.

- **ANS_HPMPI_ELAN_OFF** - Set this environment variable to ignore the Elan (Quadrics) interconnect.

On Linux systems running Intel MPI:

- Issue the command line option -usessh to use ssh instead of rsh.
- See the Intel MPI reference manual (for Linux) for further information and additional environment variables and their settings: http://software.intel.com/en-us/articles/intel-mpi-library-documentation/.

On IBM systems:

- **LIBPATH** - on IBM, if POE is installed in a directory other than the default (/usr/lpp/ppe.poe), you must supply the installed directory path via the LIBPATH environment variable:

  ```bash
  export LIBPATH=nondefault-directory-path/lib
  ```

To verify that these environment variables are set correctly on each machine, run:

```bash
rsh machine1 env
```

### 2.2.2. Using the mpitest Program

The `mpitest` program is a ping program to verify that MPI is set up correctly. The `mpitest` program should start without errors. If it does not, check your paths, `.rhosts` file, and permissions; correct any errors, and rerun. Additional communication tests are performed.

When running the mpitest programs, you must use an even number of nodes.

**On UNIX/Linux:**

For HP-MPI (default), issue the following command:

```
mpitest121 -machines machine1:2
```
For Intel MPI, issue the following command:

```bash
mpitest121 -mpi intelmpi -machines machine1:2
```

You can use any of the same command line arguments (such as `-machines`) with the `mpitest` program as you can with full Distributed ANSYS.

**On Windows:**

Issue the following command to run a local test on Windows using HP-MPI:

```bash
ansys121 -np 2 -mpitest
```

Use the following procedure to run a distributed test on Windows using HP-MPI:

1. Create a file named `machines` in your local/home directory. Open the `machines` file in an editor.
2. Add your master and slave machines in your cluster. For example, in this cluster of two machines, the master machine is `gowindows1`. List the machine name separately for each core on that machine. For example, if `gowindows1` has four processors and `gowindows2` has two, the `machines` file would look like this:
   ```
gowindows1
gowindows1
gowindows1
gowindows1
gowindows2
gowindows2
   ```
3. From a command prompt, navigate to your working directory. Run the following:

```bash
ansys121 -mpifile machines -mpitest
```

### 2.2.3. Interconnect Configuration

Low-end hardware, such as interconnects and cables, can reduce the speed improvements you see in a distributed analysis. We typically recommend that you use an interconnect with a communication speed of 200 megabits/second (20 megabytes/second) or higher.

Distributed ANSYS supports the following interconnects. Not all interconnects are available on all platforms; see [http://www.ansys.com/services/ss-hardware-support.asp](http://www.ansys.com/services/ss-hardware-support.asp) for a current list of supported interconnects. Other interconnects may work but have not been tested.

- Elan (Quadrics) (recommended)
- InfiniBand (recommended)
- Myrinet (recommended)
- GigE (minimum recommended)
- Ethernet (not recommended)

Interconnects plug into a PCI (Peripheral Component Interconnect), PCI-X (extended), or PCIe (PCI Express) slot on the system. You will need a PCI-X or a PCIe slot for the faster interconnects to accommodate the higher speeds.

Hardware for specific types of interconnects is generally incompatible with other proprietary interconnect types (except Ethernet and GIGE).
Systems can have a network of several different types of interconnects. Each interconnect must be assigned a unique hostname and IP address.

On Windows x64 systems, use the Network Wizard in the Compute Cluster Administrator to configure your interconnects. See the Compute Cluster Pack documentation for specific details on setting up the interconnects. You may need to ensure that Windows Firewall is disabled for Distributed ANSYS to work correctly.

See Optional Setup Tasks (p. 10) for environment variables specific to the Elan (Quadrics) interconnect or refer to the HP-MPI documentation at http://docs.hp.com.

### 2.2.4. Other Considerations

Other factors can also affect your distributed analysis.

- A single 64-bit machine with multiple processors running under shared memory typically works as well as a cluster (multiple machines), and can often be faster. This type of comparison requires evaluating many parameters, such as the network speed, the model being solved, the distributed solver being used, and the hardware involved. The PCG solver performs very little I/O; therefore, running multiple processors on one machine works very well because the multiple processors can communicate rapidly with each other. However, the DSPARSE solver runs in out-of-core mode by default, with high I/O. Running multiple processors on a single machine with the DSPARSE solver may not work well if the hard drive cannot quickly handle the multiple I/O requests generated by the DSPARSE processes. For better performance in this case, run the DSPARSE solver using a single processor on multiple machines, assuming that you are using a high-speed network to quickly handle the communication between machines. See the PCG and DSPARSE solver descriptions in the Basic Analysis Guide for more information.

- PCG considerations: Review the following guidelines if you will be using the PCG solver in distributed mode. Note that these are not steadfast rules, but rather recommendations to help you get started.
  - The master machine needs more memory than the slave machines.
  - Deploy 64-bit platforms such as Linux on Itanium or AMD chips.
  - Use the /3GB switch on 32-bit platforms on Windows systems.
  - As a broad guideline, use the following formula to get a general idea of memory usage for the DPCG solver. In this formula, \( n \) is the total number of CPU processors used, and MDOF is million degrees of freedom.

Master machine (Machine 0): 
\[
\text{MDOF}_{\text{maximum}} = \frac{\text{Machine(0) Memory (GB)}}{0.1 + 1.0/\text{No. of machines}}
\]

Slave machines (Machine 1 - \( n \)): 
\[
\text{MDOF}_{\text{maximum}} = \frac{\text{Machines (1...n) Memory (GB)}}{n}
\]

For example, if you have a master machine that is a 32-bit Windows machine with 2.2 GB available RAM, using the /3GB switch, and a total of four machines in the cluster, you could solve a problem up to 6.3 MDOF:

\[
\text{MDOF} = \frac{2.2 \text{ GB}}{0.1 + 1/4} = 6.3 \text{ MDOF}
\]

In this scenario, the slave machines must have 6.3 / 4 or about 1.5 GB of available memory.

### 2.3. Starting Distributed ANSYS

After you've completed the configuration steps, you can use several methods to start Distributed ANSYS. We recommend that you use the Mechanical APDL Product Launcher to ensure the correct settings. All methods are explained here.
• Use the launcher
• Use the command line
• Use the HPC Job Manager on Windows x64 systems to run across multiple machines
• Use Remote Solve in ANSYS Workbench.

**Important Points** You can use an NFS mount to the ANSYS installation; however, we do not recommend NFS-mounting the working directories. Doing so can result in significant declines in performance.

On all systems, Distributed ANSYS will not delete any files from the slave nodes when the analysis is completed. Distributed ANSYS creates the same basic files as ANSYS (see Files that ANSYS Writes in the Basic Analysis Guide for more information), except that Jobname becomes JobnameN. You can delete these files from the slave nodes when your analysis is complete (including any restarts that you may wish to perform). If you do not wish to have the files necessary for a restart saved, you can issue RESCONTROL, NORESTART.

On Windows systems, there is no ANSYS output console window when running the Distributed ANSYS GUI (interactive mode). All standard output from the master process will be written to a file named file0.out. (Note that the jobname is not used.)

### 2.3.1. Starting Distributed ANSYS via the Launcher

Use the following procedure to start Distributed ANSYS via the launcher.

This procedure is also used for running LS-DYNA’s MPP version of ANSYS. See LS-DYNA Parallel Processing Capabilities in the ANSYS LS-DYNA User's Guide for a detailed explanation on using ANSYS LS-DYNA’s parallel processing capabilities.

1. Open the Mechanical APDL Product Launcher:
   - **Windows:**
     - Start > Programs > ANSYS 12.1 > Mechanical APDL Product Launcher
   - **UNIX/Linux:**
     - launcher121

2. Select the correct environment and license.


   Specify the MPI type to be used for this distributed run (Linux and Windows only). MPI types include:
   - MS MPI (Windows 64-bit only)
   - HP-MPI
   - Intel MPI

   If you choose MS MPI, you cannot specify multiple hosts or an MPI file.

   All other platforms allow only one type of MPI. See Table 2.1: Platforms and MPI Software (p. 6) for the specific MPI version for each platform.

   Choose whether you want to run on a local machine, specify multiple hosts, or specify an existing MPI file (such as a host.list or an HP-MPI appfile):
   - If local machine, specify the number of processors on that machine.
• If multiple hosts, select the machines you want to use from the list of available hosts. The list of available hosts is populated from the hosts121.ans file. Click on the machines you want to use and click Add to move them to the Selected Hosts list to use them for this run. If you click Add more than once for a machine, the number of processors for that machine will increment each time, up to the maximum allowed in the hosts121.ans file.

You can also add or remove a host, but be aware that adding or removing a host from here will modify only this run; the hosts121.ans file will not be updated with any new information from this dialog box.

• If specifying an MPI file, type in the full path to the file, or browse to the file. If typing in the path, you must use the absolute path.

Additional Options for HP or Linux systems using HP-MPI On these systems, you can choose to use secure shell (SSH) instead of remote shell (RSH). This option will override MPI_REMSH, if the path to SSH is different. See Optional Setup Tasks (p. 10) for more information on MPI_REMSH. ANSYS uses RSH as the default, whereas HP-MPI uses SSH as the default.

If you are using the launcher, you can select the Use launcher-specified working directory on all nodes option on the High Performance Computing Setup tab. This option uses the working directory as specified on the File Management tab as the directory structure on the master and all nodes. If you select this option, all machines will require the identical directory structure matching the working directory specified on the launcher. This option will override any existing MPI_WORKDIR settings on the master or the nodes.

4. Click Run to launch ANSYS.

2.3.2. Starting Distributed ANSYS via Command Line

You can also start Distributed ANSYS via the command line using the following procedures.

These procedures are also used for running LS-DYNA’s MPP version with ANSYS. To run the LS-DYNA MPP version, substitute lsdyna121 for ansys121 in the examples below. Only the command line options specific to Distributed ANSYS apply to LS-DYNA MPP (i.e., -dis, -np, -machines, -mpifile). See LS-DYNA Parallel Processing Capabilities in the ANSYS LS-DYNA User’s Guide for a detailed explanation on using ANSYS LS-DYNA’s parallel processing capabilities.

Running on a Local Host If you are running distributed ANSYS locally (i.e., running across multiple processors on a single local machine), you need to specify the number of processors:

ansys121 -dis -np n

You may also need to specify the MPI software. The default is HP-MPI. To specify Intel MPI, use the -mpi command line option as shown below:

ansys121 -dis -mpi intelmpi -np n

For example, if you run a job in batch mode on a local host using four processors with an input file named input1 and an output file named output1, the launch commands for UNIX/Linux and Windows would be as shown below.

On UNIX/Linux:

ansys121 -dis -np 4 -b < input1 > output1 (for default HP-MPI)

or

ansys121 -dis -mpi intelmpi -np 4 -b < input1 > output1 (for Intel MPI)
On Windows Vista and XP, 32-bit and 64-bit systems:

```bash
ansys121 -dis -np 4 -b -i input1 -o output1  (for default HP-MPI)
```

or

```bash
ansys121 -dis -mpi intelmpi -np 4 -b -i input1 -o output1  (for Intel MPI)
```

**Running on Multiple Hosts**  If you are running distributed ANSYS across multiple hosts, you need to specify the number of processors on each machine:

```bash
```

You may also need to specify the MPI software. The default is HP-MPI. To specify Intel MPI, use the -mpi command line option as shown below:

```bash
```

For example, if you run a job in batch mode using two machines (one with four processors and one with two processors), with an input file named `input1` and an output file named `output1`, the launch commands for UNIX/Linux and Windows would be as shown below.

**On UNIX/Linux:**

```bash
ansys121 -dis -b -machines machine1:4:machine2:2 < input1 > output1  (for default HP-MPI)
```

or

```bash
ansys121 -dis -mpi intelmpi -b -machines machine1:4:machine2:2 < input1 > output1 (for Intel MPI)
```

**On Windows Vista and XP, 32-bit and 64-bit systems:**

```bash
ansys121 -dis -b -machines machine1:4:machine2:2 -i input1 -o output1  (for default HP-MPI)
```

or

```bash
ansys121 -dis -mpi intelmpi -b -machines machine1:4:machine2:2 -i input1 -o output1  (for Intel MPI)
```

The first machine specified with `-machines` in a Distributed ANSYS run must be the host machine and must contain any files necessary for the initiation of your job (i.e., input file, database file, etc.).

If both the `-np` and `-machines` options are used on a single command line, the `-np` will be ignored.

---

**Note**

Do not run Distributed ANSYS in the background via the command line (i.e., do not append an ampersand (&) to the command line).

### 2.3.3. Starting Distributed ANSYS via the HPC Job Manager

If you are running on Windows x64 systems using Microsoft HPC Pack (MS MPI), you need to use the HPC Job Manager to start Distributed ANSYS. For more information, refer to the following README files:

Program Files\Ansys Inc\V121\ANSYS\WinHPC\README.mht

or

Program Files\Ansys Inc\V121\ANSYS\WinHPC\README.docx
2.3.4. Starting Distributed ANSYS in ANSYS Workbench

If you are running ANSYS Workbench, you can start a Distributed ANSYS job in the Mechanical application; go to Tools > Solve Process Settings. Select the remote solve process you want to use and click the Advanced button. You will need to select Distribute ANSYS Solution (if possible) and enter the command line arguments to be submitted. Use the options as described in Starting Distributed ANSYS via Command Line (p. 15). If Distribute ANSYS Solution (if possible) is selected, you will not need to specify the -dis flag on the command line.

If you are running a remote solution on multiple machines, use the -machines option to specify the machines and processors on which to run the job. If you are running a remote solution on one machine with multiple processors, specify the number of processors in the Max Number of Utilized Processors field. You will not need to add any command line arguments.

For more information on running a distributed solution in ANSYS Workbench, see Using Solve Process Settings.

2.3.5. Using MPI appfiles

You can specify an existing MPI file (such as an HP-MPI appfile) on the command line, rather than typing out multiple hosts or a complicated command line:

```bash
ansys121 -dis -mpifile appfile_name
```

For an Intel MPI appfile, include the -mpi command line option:

```bash
ansys121 -dis -mpi intelmpi -mpifile appfile_name
```

The format of the appfile is system-dependent.

If the file is not in the current working directory, you will need to include the full path to the file. The file must reside on the local machine.

You cannot use the -mpifile option in conjunction with the -np (local host) or -machines (multiple hosts) options.

If the Specify Multiple Hosts launcher option or the -machines command line option was used, ANSYS generates a default appfile named host.list. You can rename this file, move it, or modify it for future runs if necessary. See the documentation for your vendor/MPI type for details on working with the appfile.

On IBM AIX systems, verify that the host.list file is consistent with the number of processors specified. For more information, see the IBM Parallel Environment for AIX documentation, Operation and Use, Volume 1, Using the Parallel Operating Environment.

Using the HP-MPI appfile  HP-MPI uses an appfile to define the machines in the array (or the local host). A typical HP-MPI appfile might look like this:

```bash
-h mach1 -np 2 /ansys_inc/v121/ansys/bin/ansysdis121 -dis
-h mach2 -np 2 /ansys_inc/v121/ansys/bin/ansysdis121 -dis
```

See the HP-MPI user documentation for details on working with the HP-MPI appfile.

Using the Intel MPI appfile  Intel MPI uses an appfile to define the machines in the array (or the local host). A typical Intel MPI appfile might look like this:

```bash
-host mach1 -env env_vars env_var_settings -np 2 /ansys_inc/v121/ansys/bin/ansysdis121 -dis -mpi INTELMPI
-host mach2 -env env_vars env_var_settings -np 2 /ansys_inc/v121/ansys/bin/ansysdis121 -dis -mpi INTELMPI
```
Using Other appfiles  A typical appfile for other UNIX/Linux machines might look like this:

mach1
mach1
mach2
mach2

2.3.6. Files that Distributed ANSYS Writes

Distributed ANSYS writes files to the master and slave nodes as the analysis progresses. Depending on the system you're running on, you will have varying levels of control over the location of those files.

If you are running on Sun, IBM, or SGI Linux Altix systems, you must have identical working directory structures set up on the master and all slaves. Distributed ANSYS will always use the current working directory on the master node and will expect identical directory structures to exist on all slave nodes. If you are using the launcher, the working directory specified on the File Management tab is the directory that Distributed ANSYS will expect.

If you are running on HP or Linux systems running HP-MPI, you have additional controls over the location of the files:

- By default, Distributed ANSYS uses the current working directory on the master node and the login ($HOME) directory on the slave nodes.
- You can set the MPI_WORKDIR environment variable on the master node to any directory. Distributed ANSYS will then expect that exact directory structure to exist on all of the slave nodes.
- You can set the MPI_WORKDIR environment variable on any or all of the slave nodes but NOT the master node. In this scenario, Distributed ANSYS will then use the directory specified by MPI_WORKDIR on the nodes where it is set, and will use the current working directory on the master node and the login ($HOME) directory on any slave nodes that do not have MPI_WORKDIR set.
- If you are using the launcher, you can select the Use launcher-specified working directory on all nodes option on the High Performance Computing Setup tab. This option uses the working directory as specified on the File Management tab as the directory structure on the master and all nodes. If you select this option, all machines will require the identical directory structure matching the working directory specified on the launcher. This option will override any existing MPI_WORKDIR settings on the master or the nodes.
- All working directories that are specified must exist before running a job.

If you do not want the files necessary for a restart to be saved, you can issue the RESCONTROL,NORESTART command. If you use this command, the remote processors will not have files such as .ESAV, .OSAV, .RST, or .x000, in the directory at the end of the run. The host processor will not have files such as .ESAV, .OSAV, .x000, .RDB, or .LDHI at the end of the run. Distributed ANSYS will remove all the above scratch files at the end of the solution phase (FINISH or /EXIT). This option is applicable to both singleframe and multiframe restart control and is useful for file clean-up on remote or host processors.
Chapter 3: Running Distributed ANSYS

In Distributed ANSYS, the entire solution phase runs in parallel, including the stiffness matrix generation, linear equations solving, and results calculations. As a result, Distributed ANSYS usually scales much better than a shared memory parallel run, and can often achieve over 8X speedup when using a sufficient number of processors.

You can run one job in Distributed ANSYS on a single machine with multiple processors or, on certain platforms, on multiple machines with one or more processors in each machine.

The processor you use to launch the run will do preprocessing in Distributed ANSYS, as well as any postprocessing. During /SOLUTION, each processor creates its own JOBNAME.RST or .RTH file. These files are then automatically merged for postprocessing.

The following topics on running Distributed ANSYS are available:

3.1. Supported Analysis Types
3.2. Supported Features
3.3. Running a Distributed Analysis
3.4. Restarts in Distributed ANSYS
3.5. Understanding the Working Principles and Behavior of Distributed ANSYS
3.6. Example Problems
3.7. Troubleshooting

3.1. Supported Analysis Types

The following analysis types are supported by Distributed ANSYS:

- Static linear or nonlinear analyses for single field structural problem (DOFs: UX, UY, UZ, ROTX, ROTY, ROTZ, WARP).
- Single field thermal analyses (DOF: TEMP).
- Full transient analyses for single field structural and single field thermal analysis, including HHT transient analyses.
- Modal analyses using the Block Lanczos, PCG Lanczos, and Supernode solutions (MODOPT, LANB; LANPCG; and SNODE). Note that the Block Lanczos eigensolver and Supernode eigensolver are not distributed eigensolvers. Therefore, you will not see the full performance improvements that you would with a fully-distributed solution. When possible, you should use the PCG Lanczos option.
- Buckling analyses using the Block Lanczos eigensolver (MODOPT, LANB). Note that the Block Lanczos eigensolver is not a distributed eigensolver. Therefore, you will not see the full performance improvements that you would with a fully distributed solution.
- Structural harmonic analyses.
- Low-frequency electromagnetic analysis using only the following elements: SOLID96, SOLID97, SOLID122, SOLID123, SOLID231, SOLID232, SOLID236, SOLID237, and SOURC36 (when used in a model with the above elements only).
- High-frequency electromagnetic analysis using elements HF119 and HF120.
Coupled-field analyses using only the following elements: PLANE223, SOLID226, SOLID227.

Superelements in the use pass of a substructuring analysis.

Modal cyclic symmetry analyses, including prestressed modal cyclic symmetry.

Spectrum and FLOTRAN analyses are not supported.

3.2. Supported Features

This section lists features that are supported in Distributed ANSYS and features that are not supported. These are not comprehensive lists, but represent major features and capabilities found in the ANSYS program.

Supported Features:

The following features are supported by Distributed ANSYS:

- Large deformations (NLGEOM,ON).
- Line search (LNSRCH,ON).
- Auto time stepping (AUTOTS,ON).
- Solution controls.
- Initial conditions (IC).
- Initial state (INISTATE).
- Nonlinear material properties specified by the TB command.
- Gasket elements and pre-tension elements.
- u-P formulations introduced by the 18x elements and TARGE169 - CONTA178.
- Contact nonlinearity (TARGE169 - CONTA178, CONTAC52), with the following restrictions for the CONTA17x elements:
  - KEYOPT(1) = 0, 2 only
  - KEYOPT(10) = 0, 1, 2 only
  All other KEYOPTS are supported as documented in the element descriptions.
- The radiosity solver, except for surface elements SURF251 and SURF252.
- Single- and multi-frame restarts.
- Partial solution (PSOLVE); only the EIGLANB and EIGLANPCG options are supported.
- Prestress effects (PSTRES).
- Mass summary option on the IRLF command (IRLF,-1).
- User programmable features, including the user-defined element (USER300).

Non-supported Features:

The following ANSYS features are not supported by Distributed ANSYS:

- p-Elements.
- Substructure generation pass and expansion pass.
- Automatic substructuring.
- MPC184 elements with KEYOPT(2) = 0.
• SURF151 and SURF153 elements with KEYOPT(3) = 10.
• Element morphing.
• Arc-length method (ARCLEN).
• Inertia relief (IRLF,1).
• Partial solution (PSOLVE); all options except EIGLANB and EIGLANPCG are unsupported.
• Fast thermal solution option (THOPT).
• MAGOPT with VALUE = 3 (use the previously stored H field solution and calculate the complete H field) used with the magnetic element SOLID117.
• Variational Technology options on the STAOPT, MODOPT, and TRNOPT commands.
• Expansion pass (EXPASS).
• Transient mode superposition analyses (TRNOPT,MSUP).
• Harmonic mode superposition analyses (HROPT,MSUP).
• Spectrum analyses (ANTYPE,SPECTRUM).

Note
You may, however, use Distributed ANSYS for the modal analysis phase and use ANSYS for the subsequent mode superposition, spectrum, and expansion analyses. You will need the files Jobname.DB, .FULL, .ESAV, and .RST (not Jobname0) generated by the master process.

• Optimization and probabilistic design.
• The PGR file.
• Multiple load vectors in a modal analysis (MODCONT).
• ANSYS Multi-field solver - multiple code coupling (MFX)
• ANSYS Multi-field solver- single code (MFS)

3.3. Running a Distributed Analysis

The following steps assume that you have set up your distributed environment and launched ANSYS following the steps in Chapter 2, Configuring and Starting Distributed ANSYS (p. 5). After you have your distributed environment correctly configured, you run a distributed analysis in Distributed ANSYS the same way you do in shared-memory ANSYS. You can also perform a single- or multi-frame restart as described in the Basic Analysis Guide.

1. Set up your analysis (geometry, loads, etc.) as you normally would.
2. Specify your solution output using OUTRES. We recommend that you not use OUTPR.
3. Set your analysis options by running the Solution Controls dialog: Main Menu> Solution> Analysis Type> Sol'n Controls and click on the Sol'n Options tab. Choose the solver you want to use and click OK. You can use one of the following solvers. See the EQSLV command description for more information on each of the solvers, default settings, etc.

   • Distributed sparse (EQSLV,DSPARSE): This solver performs factorization of the matrix and back/forward substitution in distributed parallel mode, and has demonstrated the best performance, with speed improvements exceeding 8X when using 16 or more processors.
• PCG (EQSLV, PCG)
• JCG (EQSLV, JCG) (not available from the Solution Controls gui.)

Choosing the SPARSE, PCG, or JCG solver when you're running Distributed ANSYS will automatically run the distributed version of these solvers.

Other solvers (ICCG, QRM, etc.) will not work in a distributed environment.

4. Solve the analysis.

   Command(s): SOLVE
   GUI: Main Menu> Solution> Solve> Current LS

5. After the solution completes, specify the set of results to be read from the results file. Note that a SET command is required as not all solution data is in the database.

   Command(s): SET
   GUI: Main Menu> General Postproc> Read Results

6. Postprocess your results as you would for any analysis.

**Notes on Running Distributed ANSYS:**

Only the master machine reads the config.ans file.

Distributed ANSYS ignores the /CONFIG, noeldb command.

### 3.4. Restarts in Distributed ANSYS

Distributed ANSYS supports both singleframe and multiframe restarts for nonlinear and full transient analyses. The procedures and command controls are the same as described in the Basic Analysis Guide; however, restarts in Distributed ANSYS have additional limitations:

• The job launch procedure (or script) used when restarting Distributed ANSYS must not be altered following the first loadstep and first substep, i.e., you must use the same number of machines (or processors) and the same host and slave relationships among these machines (or processors) in the restart job that follows.

For example, if you use the following launch script for loadstep one and substep one:

```bash
ansys121 -dis -machines mach3:1:mach2:1:mach4:1 -i jobone -o output1
```

where the slave machines are `mach2` and `mach4`, the host machine is `mach3` (the host machine always appears first in the machine list), the input file is `jobone` and output file `output1`, then for the restart (both singleframe and multiframe), you must use the following job launch script:

```bash
```

which uses the same `-machines` description to ensure consistency. Any alterations in the `-machines` field will result in restart failure.
• The files needed for a restart must not be removed from the host or slave machines when performing a restart. Each machine (or processor) has its own restart files that are written from the previous run. The restart process needs to use these local files to perform the correct match.

### Table 3.1 Required Files for Singleframe Restarts

<table>
<thead>
<tr>
<th>Host Machine</th>
<th>Slave Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname.DB</td>
<td></td>
</tr>
<tr>
<td>Jobname0.ESAV</td>
<td>JobnameN.ESAV</td>
</tr>
<tr>
<td>Jobname0.OSAV (required when Jobname0.ESAV is not there)</td>
<td>JobnameN.OSAV (required when JobnameN.ESAV is not there)</td>
</tr>
<tr>
<td>Jobname0.EMAT</td>
<td>JobnameN.EMAT</td>
</tr>
<tr>
<td>Jobname0.RST (this is the local .RST file for this domain)</td>
<td>JobnameN.RST (this is the local .RST file for this domain)</td>
</tr>
</tbody>
</table>

### Table 3.2 Required Files for Multiframe Restarts

<table>
<thead>
<tr>
<th>Host Machine</th>
<th>Slave Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname.LDHI</td>
<td></td>
</tr>
<tr>
<td>Jobname.RDB</td>
<td></td>
</tr>
<tr>
<td>Jobname0.xnnn</td>
<td>JobnameN.xnnn</td>
</tr>
<tr>
<td>Jobname0.RST (this is the local .RST file for this domain)</td>
<td>JobnameN.RST (this is the local .RST file for this domain)</td>
</tr>
</tbody>
</table>

In all restarts, the result file Jobname.RST (or Jobname.RTH or Jobname.RMG) on the host machine is re-created in each restart by merging the JobnameN.RST again. The input deck (or interactive session) for a restart should reside on the host machine.

If you do not require a restart, issue RESCONTROL,NORESTART in the run to remove or to not write the necessary restart files on the host and slave machines. This option is useful for file clean-up and control.

### 3.5. Understanding the Working Principles and Behavior of Distributed ANSYS

The fundamental difference between Distributed ANSYS and shared-memory ANSYS is that \( n \) number of ANSYS jobs will be running at the same time (where \( n \) is the total number of CPU processor cores used) for one model. These \( n \) jobs are not aware of each other’s existence unless the individual CPU cores are communicating (sending messages). Distributed ANSYS is the method by which the CPU cores communicate with each other in the right location and at the appropriate time.

Upon starting a solution, Distributed ANSYS automatically decomposes the problem into \( n \) CPU domains so that each CPU core works on only a portion of the model. When the existing /SOLUTION phase ends (e.g., FINISH is issued), Distributed ANSYS works on the entire model again (i.e., it behaves like shared-memory ANSYS).

Following is a summary of behavioral differences between Distributed ANSYS and shared-memory ANSYS.

**Database Jobname Conventions**

The master processor will create database files with the prefix Jobname and slaves Jobname1 through Jobnamen. When you RESUME from the default
file, ANSYS reads first from Jobname.DB. If Jobname.DB is not found, it will then try to read Jobname0.DB. All actions in PREP7 or POST while in Distributed ANSYS will work on Jobname.DB by default. If a non-default file is specified, then Distributed ANSYS behaves the same as shared-memory ANSYS.

General Jobname Conventions

The master processor will create files with the prefix Jobname0 and slaves Jobname1 through Jobnmen. When you issue commands such as PASAVE, CDWRITE, or LSWRITE to write to the default file name (by leaving the file name field blank), Distributed ANSYS will save all of these to Jobname0 with the appropriate extension. When you PARESU, CDREAD, or LSREAD from the default file, ANSYS reads first from Jobname0.EXT. If Jobname0.EXT is not found, it will then try to read Jobname.EXT. All actions in PREP7 or POST while in Distributed ANSYS will work on Jobname0.EXT by default. If a non-default file is specified, then Distributed ANSYS behaves the same as shared-memory ANSYS.

Post-Processing Files and Filenames

During the /SOLUTION phase, Distributed ANSYS creates Jobnamen.RST or Jobnamen.RTH files, as well as Jobnamen.MODE and Jobnamen.IST files if appropriate, for each running process (you may see these working files in the directory). At the successful completion of the run, Distributed ANSYS automatically merges the Jobnamen files into a single file called Jobname.RST (or .RTH, .MODE, or .IST). From that point, postprocessing is the same as in shared-memory ANSYS.

Use of APDL

In pre- and post-processing, APDL works the same in Distributed ANSYS as in shared-memory ANSYS. However, in /SOLUTION, Distributed ANSYS does not support certain *GET items. In general, Distributed ANSYS supports global solution *GET results such as total displacements and reaction forces. It does not support element level results specified by ESEL, ESOL, and ETABLE labels. Unsupported items will return a *GET value of zero.

Batch and Interactive Mode

You can launch Distributed ANSYS in either interactive or batch mode on the master processor. However, the slave processor is always in batch mode. The slave process cannot read the START121.ANS or STOP121.ANS files. The master process sends all /CONFIG, LABEL commands to the slave processors as needed.

On Windows systems, there is no ANSYS output console window when running the Distributed ANSYS GUI (interactive mode). All standard output from the master process will be written to a file named file0.out. (Note that the jobname is not used.)

Condensed Data Input

Multiple commands entered in an ANSYS input file can be condensed into a single line if the commands are separated by the $ character (see Condensed Data Input in the Command Reference). Distributed ANSYS cannot properly handle condensed data input. Each command must be placed on its own line in the input file for a Distributed ANSYS run.

Postprocessing with Database File and SET Commands

Shared-memory ANSYS can postprocess using the Jobname.DB file (if the solution results were saved), as well as using the Jobname.RST file. Distributed ANSYS, however, can only postprocess using the Jobname.RST file
and cannot use the Jobname.DB file as no solution results are written to the database. You will need to issue a \texttt{SET} command before postprocessing.

Print Output (\texttt{OUTPR} Command)

In Distributed ANSYS, the \texttt{OUTPR} command prints NSOL and RSOL in the same manner as in shared-memory ANSYS. However, for other items such as ESOL, Distributed ANSYS prints only the element solution on the CPU domain of the master processor. Therefore, \texttt{OUTPR}, ESOL has incomplete information and is not recommended. Also, the order of elements is different from that of shared-memory ANSYS due to domain decomposition. A direct one-to-one element comparison with shared-memory ANSYS will be different if using \texttt{OUTPR}.

ASCII Job Output Files

When a Distributed ANSYS job is executed, the output for the master processor is written to the screen by default. If you specified an output file via the launcher or the \texttt{-o} command line option, the output is written to that file. Distributed ANSYS automatically outputs the ASCII files from each slave processor to Jobnamen.OUT. Normally these slave output files have little value because all of the job information is on the master processor (Jobname0.OUT). The same principle also applies to the other ANSYS ASCII files such as Jobname0.ERR, Jobname0.MNTR, etc. However, if the job hangs, you may be able to determine the cause by studying the contents of the slave output and/or error files.

Error Handling and Job Hanging

When an error occurs on one of the CPU processes during the Distributed ANSYS execution, the process sends an error message to all other Distributed ANSYS processes so that the entire run exits gracefully. However, if an error message fails to send, the job may hang, and you will need to manually kill all the processes. You can view each Jobnamen.ERR and/or Jobnamen.OUT file if you suspect that one of the jobs hung. This action should rarely be required.

Large Number of CE/CP and Contact Elements

In all HPC products (both shared-memory ANSYS and Distributed ANSYS), the program can handle a large number of coupling and constraint equations (CE/CP) and contact elements. However, specifying too many of these items can force Distributed ANSYS to communicate more among each CPU core, resulting in longer elapsed time to complete a distributed parallel job. You should reduce the number of CE/CP if possible and make potential contact pairs in a smaller region to achieve non-deteriorated performance. In addition, for assembly contact pairs or small sliding contact pairs, you can use the command \texttt{CNCHECK}, TRIM to remove contact and target elements that are initially in far-field (open and not near contact). This trimming option will help to achieve better performance in Distributed ANSYS runs.

3.6. Example Problems

This section contains tutorials for running Distributed ANSYS on Linux and Windows platforms.

This tutorial is divided into two parts.

The first part walks you through setting up your distributed environment and running a test to verify that communication between systems is occurring correctly. Running this part is optional (though strongly recommended), and need only be done when you initially configure your environment, or when you change your environment by adding, removing, or changing machines.
The second part of the tutorial walks you through setting up your distributed environment and then directly running a sample problem. The tutorial is designed so that you can modify settings (such as the number of machines in the cluster, number of processors on each machine, etc.), but we strongly recommend that the first time you run this tutorial, you follow the steps exactly. Once you’re familiar with the process, you can then modify the tutorial to more closely match your particular environment. You can then use the files generated in future GUI or batch runs.

Sample inputs to use for each example problem are included on the media under Program Files\Ansys Inc\V121\ansys\Data\models (Windows) or /ansys_inc/v121/ansys/data/models (UNIX/Linux). Static and modal examples are provided. The problem setup for each platform is the same whether you run the static or the modal example.

3.6.1. Example: Running Distributed ANSYS on Linux

The following tutorial walks you through the setup of your distributed solver environment, and is applicable only to systems running ANSYS 12.1 on a 64-bit Linux cluster (RH 3.0) under HP-MPI 2.3.1. ANSYS 12.1 includes HP-MPI 2.3.1.

One of the sample problems, tutor1_carrier_linux.inp (static) or tutor2_carrier_modal.inp (modal), is required to complete the tutorial. Save this file to your working directory before beginning the tutorial. You can run either sample problem using the problem setup described here.

**Part A: Setup and Run mpitest**

1. Set up identical installation and working directory structures on all machines (master and slaves) in the cluster.
2. Type `hostname` on each machine in the cluster. Note the name of each machine. You will need this name to set up both the `.rhosts` file and the Configure Cluster option of the ANS_ADMIN121 utility.

   Set up the `.rhosts` file on each machine. The `.rhosts` file lists each machine in the cluster, followed by your username. The machines should be listed using their complete system name, as taken from `uname`. For example, each `.rhosts` file for our two-machine cluster looks like this (where golinux1 and golinux2 are example machine names, and jqd is an example username):

   ```
   golinux1  jqd
   golinux2  jqd
   ```

   Change/verify `.rhosts` file permissions on all machines by issuing:

   ```
   chmod 600 .rhosts
   ```

   Navigate to your working directory. Run the following:

   ```
   /ansys_inc/v121/ansys/bin/mpitest121
   ```

   The mpitest program should start without errors. If it does not, check your paths, `.rhosts` file, and permissions; correct any errors; and rerun.

**Part B: Setup and Run a Distributed Solution**

1. Set up identical installation and working directory structures on all machines (master and slaves) in the cluster.
2. Install ANSYS 12.1 on the master machine, following the typical installation process.
3. Install ANSYS 12.1 on the slave machines.

Steps 2 and 3 above will install all necessary components on your machines, including HP-MPI 2.3.1.

4. Type **hostname** on each machine in the cluster. Note the name of each machine. You will need this name to set up both the `.rhosts` file and the **Configure Cluster** option of the **ANS_ADMIN121** utility.

5. Set up the `.rhosts` file on each machine. The `.rhosts` file lists each machine in the cluster, followed by your username. The machines should be listed using their complete system name, as taken from `uname`. For example, each `.rhosts` file for our two-machine cluster looks like this (where `golinux1` and `golinux2` are example machine names, and `jqd` is an example username):

   ```
golinux1  jqd
golinux2  jqd
```

6. Change/verify `.rhosts` file permissions on all machines by issuing:

   ```
   chmod 600 .rhosts
   ```

7. Verify communication between machines via rsh. If the communication between machines is happening correctly, you will not need a password.

8. Run the **ANS_ADMIN121** utility:

   ```
   /ansys_inc/v121/ansys/bin/ans_admin121
   ```

9. Choose **ANSYS / Workbench Configuration**, and then click **Configure Cluster**. Under **Select file to configure**, choose the `hosts121.ans` file to be configured and choose **Configure for Distributed ANSYS**. Click **OK**. Then enter the system name (from Step 4) in the **Machine hostname** field and click **Add**.

   On the next dialog box, enter the system type in the **Machine type** drop-down, and the number of processors in the **Max number of jobs/processors** field for each machine in the cluster. Click **OK**.

   When you are finished adding machines, click **Close** and then click **Exit** to leave the **ANS_ADMIN121** utility. The resulting `hosts121.ans` file using our example machines would look like this:

   ```
golinux1  linem64t 0 1 0 0  /home/jqd  MPI 1 1

golinux2  linem64t 0 1 0 0  /home/jqd  MPI 1 1
```

10. Start ANSYS using the launcher:

    ```
    launcher121
    ```

11. Select the correct environment and license.

12. Go to the **High Performance Computing Setup** tab. Select **Use Distributed Computing (MPP)**. You must also specify either local machine or multiple hosts. For multiple hosts, select the machines you want to use from the list of available hosts. The list of available hosts is populated from the `hosts121.ans` file. Click on the machines you want to use and click **Add** to move them to the Selected Hosts list to use them for this run. You can also add or remove a host, but be aware that adding or removing a host from here will modify only this run; the `hosts121.ans` file will not be updated with any new information from this dialog box.

    If necessary, you can also run secure shell (ssh) by selecting **Use Secure Shell instead of Remote Shell (ssh instead of rsh)**.

13. Click **Run** to launch ANSYS.
14. In ANSYS, select File>Read Input From and navigate to tutor1_carrier_linux.inp or tutor2_carrier_modal.inp.

15. The example will progress through the building, loading, and meshing of the model. When it stops, select Main Menu>Solution>Analysis Type>Sol'n Controls.

16. On the Solution Controls dialog box, click on the Sol'n Options tab.

17. Select the Pre-Condition CG solver.

18. Click OK on the Solution Controls dialog box.

19. Solve the analysis. Choose Main Menu>Solution>Solve>Current LS. Click OK.

20. When the solution is complete, you can postprocess your results as you would with any analysis. For example, you could select Main Menu>General Postproc>Read Results>First Set and select the desired result item to display.

3.6.2. Example: Running Distributed ANSYS on Windows

The following tutorial walks you through the setup of your distributed solver environment, and is applicable only to systems running ANSYS 12.1 on a Windows cluster under HP-MPI 2.0.

One of the sample problems, tutor1_carrier_win.inp (static) or tutor2_carrier_modal.inp (modal), is required to complete the tutorial. These files are found in Program Files\Ansys Inc\V121\ansys\Data\models. Copy the file you want to use to your working directory before beginning the tutorial. You can run either sample problem using the problem setup described here.

1. Set up identical installation and working directory structures on all machines (master and slaves) in the cluster.

2. Install ANSYS 12.1 on the master machine, following the typical installation process.

3. Configure ANSYS 12.1 on the slave machines.

4. Install and register HP-MPI 2.0 on both machines following the instructions in Prerequisites for Running Distributed ANSYS (p. 5).

5. Add %MPI_ROOT%\bin to the PATH environmental variable on both machines (assuming HP-MPI 2.0 was installed on the C:\ drive). This line must be in your path for the mpirun command to be recognized.

6. On each machine, right-click on My Computer, left-click on Properties, and select the Network Identification or Computer Name tab. The full computer name will be listed. Note the name of each machine (not including the domain). You will need this name to set up the Configure Cluster option of the ANS_ADMIN utility.

7. Run the ANS_ADMIN utility on the master machine: Start >Programs >ANSYS 12.1 >Utilities >ANS_ADMIN.

8. Choose ANSYS / Workbench Configure Cluster to configure the hosts121.ans file.

   • Specify the directory in which the hosts121.ans will be configured (two possible locations are presented). Also select Configure for ... Distributed ANSYS. Click OK.

   • Enter the system name (from Step 6) in the Machine hostname field and click Add. On the next dialog box, enter the system type in the Machine type drop-down, and the number of processors in the Max number of jobs/processors field and click OK. Repeat this step for each machine in the cluster. When you are finished adding machines, click Close and then File >Exit. The resulting hosts121.ans file using our example machines where machine1 has 2 processors and machine2 has 4 processors would look like this:
9. Start ANSYS using the launcher: **Start >Programs >ANSYS 12.1 > Mechanical APDL Product Launcher.**

10. Select ANSYS Batch as the Simulation Environment, and choose a license. Specify *tutor1_carrier_win.inp* or *tutor2_carrier_modal.inp* as your input file. Both of these examples use the PCG solver. You must specify your working directory to be the location where this file is located.

11. Go to the **High Performance Computing Setup** tab. Select **Use Distributed Computing (MPP)**. You must specify either local machine or multiple hosts. For multiple hosts, select the machines you want to use from the list of available hosts. The list of available hosts is populated from the *hosts121.ans* file. Click on the machines you want to use and click **Add** to move them to the Selected Hosts list to use them for this run. Click on a machine in Selected Hosts and click **Edit** if you wish to add multiple processors for that host. You can also add or remove a host, but be aware that adding or removing a host from here will modify only this run; the *hosts121.ans* file will not be updated with any new information from this dialog box.

12. Click **Run**.

13. When the solution is complete, you can postprocess your results as you would with any analysis.

### 3.7. Troubleshooting

This section describes problems which you may encounter while using Distributed ANSYS, as well as methods for overcoming these problems. Some of these problems are specific to a particular system, as noted.

To aid in troubleshooting, you may need to view the actual MPI run command line. On UNIX/Linux the command is *mpirun*, and you can view the command line by setting the **ANS_SEE_RUN_COMMAND** environment variable to 1. On Windows the command is *mpiexec*, and you can view the command line by setting the **ANS_SEE_RUN** and **ANS_CMD_NODIAG** environment variables to TRUE.

On Windows x64 systems, use the Performance Monitor to view the individual processor performance.

**Recovering from a computer or network crash**

Be sure to kill any lingering processes (UNIX: type `kill -9` from command level) on all processors and start the job again.

**No permission to system**

This error can occur if you do not have login access to a remote system where the distributed solver is supposed to run. If you use UNIX, you can experience this problem if you have not set permissions on the .rhosts file properly. Before starting a distributed solver run, be sure you have access to all the remote systems you have specified (i.e., you should be able to rsh to those systems) and that the control node is included in its own .rhosts file. If you run on UNIX, be sure you have run the following command:

```
chmod 600 .rhosts
```

**Distributed ANSYS hangs on IBM systems**

On IBM systems, if Distributed ANSYS hangs when using < for input, verify that the input file contains a complete last line. An incomplete last line can be generated by the emacs editor or by a bad file transfer. Open the file in a vi editor and re-save it. You may also see the last line of the input file repeated...
Distributed ANSYS hangs on Linux platforms when using OFED 1.3 Infiniband with HP-MPI 2.2.7

This problem may occur on systems using OpenFabric Infiniband (OFED 1.3) with HP-MPI 2.2.7 running on Linux platforms with a kernel older than 2.6.12. This occurs because the HP-MPI library enabling the OFED fork safety feature is not supported by Linux kernels older than 2.6.12. To resolve this problem, you should disable the HP-MPI fork safety. See this HP webpage for more information: http://docs.hp.com/en/T1919-90015/ch02.html?jumpid=reg_R1002_USEN.

libibverbs: Fatal: couldn't open sysfs class 'infiniband_verbs'

This error message happens on RedHat 4 systems that do not have InfiniBand hardware installed. RedHat 4 pre-installed some InfiniBand libraries. HP-MPI attempts to auto detect the available interconnects. If InfiniBand libraries are installed, HP-MPI will open these libraries and attempt to detect if InfiniBand is in a usable state. The detection process will fail if no InfiniBand hardware is installed on the host. This causes the error message to be displayed.

The workaround is to remove IBV from the default interconnect search order. This should only be done on RedHat 4 systems that do not have InfiniBand hardware installed. The default MPI_IC_ORDER is:

```
```

Create or Edit the /etc/hpmpi.conf file to remove IBV from the MPI_IC_ORDER environment variable. The $MPI_ROOT/etc/hpmpi.conf file can be used as a template.

```
MPI_IC_ORDER="vapi:udapl:itapi:psm:mx:gm:elan:TCP"
```

Alternatively, an interconnect selection flag can be given on the mpirun command line. For example, to select TCP as the interconnect:

```
$MPI_ROOT/bin/mpirun -TCP ...
```

See this HP webpage for more information: http://h21007.www2.hp.com/portal/site/dspp/menu-item.863c3e4cbecd3f3515b49c108973a801/?ciid=41d6d59a1fac4110VgnVCM100000275d6e10RCRD

libibverbs: Warning: fork()-safety requested but init failed

This is a benign warning message that is issued when using HP-MPI v2.2.5.1 or later on Linux kernels older than v2.6.16, and OFED (OpenFabrics Enterprise Distribution) version 1.2 or later. The application will run successfully. The message is repeated for each rank used by the application, so multiple instances of this message may be found in the output file. The message can be avoided by any one of the following:

- Run on a kernel of version 2.6.16 or newer.
- Set HP-MPI environment variable MPI_IBV_NO_FORK_SAFE to 1. For Example:

```
/opt/hpmpi/bin/mpirun -np 8 -e MPI_IBV_NO_FORK_SAFE=1 -srun
```

See this HP webpage for more information: http://h21007.www2.hp.com/portal/site/dspp/menu-item.863c3e4cbecd3f3515b49c108973a801/?ciid=41d6d59a1fac4110VgnVCM100000275d6e10RCRD

Seg fault abort on Linux Itanium (linia64) systems

A random error may cause a seg fault abort on Linux Itanium (linia64) systems. This abort appears to be limited to Linux Itanium systems using Infiniban. Since the abort is random, you might not see it when using Linux Itanium with Infiniban. The abort happens when trying to invoke ANSYS, not in the middle
of any runs. If you encounter this abort the suggested fix is to manually edit the "/ansys_inc/v121/ansys/bin/anssh.ini" file. At line 1799 the following command needs to be changed: Current_hard_stack=`ulimit -Hs`; change to: Current_hard_stack=`ulimit -s`. That is, remove the capital "H" from the "-Hs" ulimit argument.

**Distributed ANSYS fails to launch when running from a fully-qualified pathname.**

Distributed ANSYS will fail if the ANSYS installation path contains a space followed by a dash if `%ANSYS121_DIR%\Bin\Platform` is not in the system PATH. Add `%ANSYS121_DIR%\Bin\Platform` to the system PATH and invoke `ansys121` (without the fully qualified pathname). For example, if your installation path is:

C:\Program Files\Ansys -Inc\v121\Bin\Platform

The following command to launch Distributed ANSYS will fail:

"C:\Program Files\Ansys -Inc\v121\Bin\Platform\ansys121.exe" -g

However, if you add `C:\Program Files\Ansys -Inc\v121\Bin\Platform` to the system PATH, you can successfully launch Distributed ANSYS by using the following command:

`ansys121 -g`

**Slave Machine Name:_desktop failed.**

User name unable to CreatProcess for \pathname\<solver executable>

The directory name is invalid.

**Local Machine Name: Process exited with code 1**

Check that the directory structures on subsequent machines are identical to those on the host machine. (Windows only)

**mpid: MPI BUG: requested interconnect not available**

The default locations for GM (Myrinet) and its libraries are /opt/gm and /opt/gm/lib, respectively. If the libraries are not found, you may see this message. To specify a different location for the GM libraries, set the `MPI_ICLIB_GM` environment variable:

```
setenv MPI_ICLIB_GM <path>/lib/libgm.so
```

AMD Opteron and other 64-bit systems may have a specific 64-bit library subdirectory, /lib64. On these systems, you need to point to this location:

```
setenv MPI_ICLIB_GM <path>/lib64/libgm.so
```

**Note**

The environment variable needs to be set on each system (such as in the .cshrc or .login files).

The required licmsgs.dat file, which contains licensing-related messages, was not found or could not be opened. The following path was determined using environment variable ANSYS121_DIR. This is a fatal error -- exiting.
Check the ANSYS121_DIR environment variable to make sure it is set properly. Note that for Windows HPC clusters, the ANSYS121_DIR environment variable should be set to \HEADNODE\Ansys Inc\v121\ansys, and the ANSYSLIC_DIR environment variable should be set to \HEADNODE\Ansys Inc\Shared Files\Licensing on all nodes.

*** FATAL *** Invalid working directory value= documents\temp (specified working directory does not exist)

When running Distributed ANSYS on a Windows platform that uses HP-MPI, you may get this fatal error if you use path names that contain spaces. There are two easy workarounds:

1. Do not use spaces in path names for the -dir, -i, and -o options when running from the ANSYS Launcher.
2. Run Distributed ANSYS from a Command Prompt window and escape all strings containing spaces with "\" at the beginning and "\" at the end. For example:

   ansys120 -b -dis -mpi hmpii -dir "\"%userprofile%\my documents\temp\"" -i "\"%userprofile%\my documents\temp\pcg.inp\"" -o "\"%userprofile%\my documents\temp\out.txt\""

Solver aborts

If you try to access more nodes than available, the solver will hang. (IBM only)

MPI: asgetnetinfo_array((null)) failed : array services not available

When running Distributed ANSYS on an SGI, the job may fail with the above error message in the log file. To correct this problem, change the authentication method in the file /usr/lib/array/ arrayd.auth file from

   AUTHENTICATION NOREMOTE

to

   AUTHENTICATION NONE

Ensure that you start the array services with /usr/etc/arrayd start as root. Also, make sure that the you can do an rsh and not get a permission denied error.

no_mp_jobs: Not enough resources available

This can happen when you request more processors than are available. This happens if -np is greater than the number of available processors. Use

   mpinfo -N

to check the number of available processors. (Sun only)

MPI: could not run executable

If you encounter this message, verify that you have the correct version of MPI installed correctly, and verify that you have a .rhosts file on each machine. If not, create a .rhosts file on all machines where you will run Distributed ANSYS, make sure the permissions on the file are 600, and include an entry for each hostname where you will run Distributed ANSYS.

mpid: Cannot set work directory: No such file or directory mpid: MPI_WORKDIR=<dir>
You will see this message if you set the `MPI_WORKDIR` environment variable but the specified directory doesn't exist. If you set `MPI_WORKDIR` on the master, this message could also appear if the directory doesn't exist on one of the slaves. In Distributed ANSYS, if you set the `MPI_WORKDIR` environment variable on the master node, Distributed ANSYS will expect all slave nodes to have the same directory.

**pfloader: Error: failed to open file '<directory>/host.list': No such file or directory**

If you are using a Quadrics interconnect with the `-machines` option, you may see this or a similar message if the `host.list` file does not exist in the working directory on each node on the cluster. The `host.list` file needs to exist on the master as well as on each of the slave nodes. For the `-machines` option to function as expected using Quadrics, a shared working directory should be used. The `-np` option is the preferred method when running Quadrics.

**Distributed ANSYS using Intel MPI hangs on Linux systems when using large “stdin” input files**

When running Distributed ANSYS (-dis command line option) with the Intel MPI software (-mpi intelmpi), a hang will occur when using input decks approximately 150KB and larger. To avoid this problem, use `-i` and `-o` in place of `< >` (redirection) to specify the input and output files when using Distributed ANSYS with Intel MPI.

**WARNING: Unable to read mpd.hosts or list of hosts isn’t provided. MPI job will be run on the current machine only. (Intel MPI)**

When using Intel MPI, you must have an `mpd.hosts` file in your working directory when going across boxes that contain a line for each box. Otherwise, you will encounter the following error.

`mpixec`: unable to start all procs; may have invalid machine names
remaining specified hosts:
xx.x.xx.xx (hostname)

**Error executing ANSYS. Refer to System-related Error Messages in the ANSYS online help. If this was a Distributed ANSYS job, verify that your MPI software is installed correctly and is in your PATH, check your environment settings, or check for an invalid command line option.**

You may encounter the above message when setting up HP-MPI or running Distributed ANSYS using HP-MPI on a Windows platform. This may occur if you did not correctly run the set password bat file or add `%MPI_ROOT%\bin` to your PATH. Verify that you completed both of those items according to the HP-MPI installation readme instructions.

You may also see this error if `Ansys Inc\v121\ansys\bin\<platform>` (where `<platform>` is intel or winx64) is not in your PATH.

If you need more detailed debugging information, use the following:

1. Open a Command Prompt window and set the following:

   ```
   SET ANSSEE_RUN=TRUE
   SET ANS_CMD_NODIAG=TRUE
   ```

2. Run the following command line: `ansys121 -b -dis -i myinput.inp -o myoutput.out`. 
Index

Symbols
.rhosts file, 8

A
analysis type
   supported by Distributed ANSYS, 19

D
Distributed ANSYS
   advantages, 19
   installing, 7
   Microsoft HPC Pack, 7
   overview, 1
   prerequisites for, 5
   required files, 8
   running, 21
   setting up the environment, 8
   solvers, 21
   starting, 13
   supported analysis types, 19
   supported features, 20
   testing, 11
   troubleshooting, 29

H
High Performance Computing
   setting up the environment, 8
hosts121.ans file, 8
HPC licensing, 1

I
Installing Distributed ANSYS, 7

M
Microsoft HPC Pack
   installing for Distributed ANSYS, 7
MPI software, 5
mpitest program, 11

P
parallel processing
   troubleshooting, 29
Prerequisites for Distributed ANSYS, 5

S
Solvers
   Distributed ANSYS, 21

starting Distributed ANSYS, 13

T
Troubleshooting
   Distributed ANSYS, 29