An Introduction to
Windows-based Clusters
and the Computational Clustering Technical Preview Kit

This article provides an overview of the integrated Microsoft® Computational Clustering Technical Preview (CCTP) kit, 2003 Edition. The CCTP kit contains software packaged by Microsoft that assists in the design, implementation, configuration, and management of Microsoft Windows®–based high-performance computing (HPC) clusters.

BY MUNIRA HUSSAIN; JEFFREY MAYERSON; JENWEI HSIEH, PH.D.; AND KEVIN NOREEN

High-performance computing (HPC) clusters, particularly Beowulf-type clusters, have been widely adopted. Constructed from standards-based computing components, HPC clusters offer competitive price/performance. These clusters are typically used in universities and national laboratories for research as well as in various industries to solve technical engineering problems. As HPC cluster technology continues to mature and gain acceptance, more applications are being developed and written for parallel programming environments. For example, financial market applications include risk management, financial engineering, and stochastic analysis.

HPC clusters are operating system–agnostic and can be constructed from standards-based components determined by performance needs such as processor speed, I/O bandwidth, and memory. Decades ago, UNIX®-based systems dominated parallel computing. As a result, many organizations transitioning to HPC clusters today consider a migration from UNIX to Linux® or BSD operating systems the easiest path, involving the least amount of code porting and investment. However, HPC clusters running the Microsoft® Windows® operating system offer an alternative, innovative form of computing.

The Computational Clustering Technical Preview (CCTP) kit is a collection of software packaged by Microsoft that provides the tools and utilities needed for building Windows-based computational clusters. The CCTP kit, 2003 Edition, consists of an integrated bundle containing evaluation versions of operating systems, Microsoft programming tools, third-party software applications, and computational development libraries. In addition, the CCTP kit includes information sharing and best practices in the form of white papers. The CCTP package is periodically updated and revised to include the latest technologies.

Exploring CCTP kit software components
The CCTP kit includes the software building blocks needed to set up a Beowulf-type cluster on a Windows operating system. The package consists of cluster message-passing libraries, monitoring and management tools, compilers for code optimization, debugging tools, and parallel implementation software. Some of these components take advantage of the Microsoft .NET framework and architecture. The .NET framework is required on nodes to implement the Microsoft common language runtime (CLR).
Code can be compiled to Microsoft intermediate language (MSIL) and executed by any node running the CLR.

**Operating systems.** The CCTP kit comes with Windows 2000 Advanced Server and Windows XP Professional, and each operating system includes a 120-day free trial license. A Windows 2000 Advanced Server node can be used to create users and set up domain services. User and resource management is centralized to the domain controller to simplify the cluster configuration. Compute nodes also utilize Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) services running on the master node to help centralize network management; the DNS server provides secure dynamic updates to registered compute nodes within the cluster. Windows XP is a client version of the operating system, with stripped-down services and utilities.

**Message Passing Interface.** Both commercial and open source Message Passing Interface (MPI) libraries are included in the CCTP package. The MPI Software Technology MPI/Pro® library is a commercial implementation of the MPI standard; MPICH.NT is an open source, high-performance version of the standard ported by Argonne National Laboratories. MPI/Pro depends on the .NET framework runtime environment, through which it provides remote process startup and management services. These functions offer secure login to users on compute nodes.

**Compilers.** The CCTP package comes with evaluation versions of compilers from Intel and Microsoft. Microsoft Visual Studio® .NET development studio is included along with Intel® Fortran and C++ compilers for Windows. The Intel compilers can be used in the Visual Studio C++ environment to provide object and source code compatibility. Both Intel and Microsoft compilers are optimized for Intel architecture and Windows operating system platforms.

**Programming libraries.** Intel Math Kernel Libraries (MKL) are used by developers in scientific and research fields to perform linear algebra, vector math functions, and Fast Fourier Transform (FFT) calculations. MKL are multithreaded and employ OpenMP® threading technology to make use of all available processors. MKL also contain BLAS subroutines. The PLAPACK and LAPACK routines and libraries included in the CCTP kit are used in parallel implementations and in solving linear algebra algorithms and applications.

**Debuggers and programming tools.** Another added feature of the CCTP kit is the Intel VTune™ Performance Analyzer tool, which is used to probe a function, module, or application. VTune displays performance data and helps to identify bottlenecks and contingencies within software. This tool provides advanced debugging and can enhance the execution of software benchmarks and applications.

**Cluster monitoring tool.** The ability to manage a large number of nodes in an HPC cluster is critical. Therefore, CCTP includes the Computation Cluster Monitor (CCM) tool built by the Cornell Theory Center. It consists of a graphical user interface (GUI) and a daemon called cmon. The daemon must be installed on all compute nodes in the cluster. CCM lets administrators know the status of the nodes: free, down, or busy. It can display a graph of CPU and memory information for multiple nodes collectively, logging and storing this data as history. The tool also captures system-level information—such as BIOS, memory, hardware, and types of services—for each node (see Figure 1).

Along with the CCM application included in the CCTP kit, Microsoft Windows Advanced Server also includes a built-in tool called Perfmon (an abbreviation of “performance monitor”). This tool can monitor a large number of parameters—for instance, a complete range of counters for system usage and resources such as TCP/IP connections, CPU, and memory. The Perfmon tool also enables logging and sends alerts when certain thresholds are set. Because Perfmon is tied to Windows 2000 security, certain user groups and settings can be configured to limit the monitoring activity.

**Job scheduler.** Resource monitoring and workload management is important when multiple users share a cluster. This type of management may even be important for a single user who sends multiple computation requests to a cluster. The CCTP package contains the ClusterController™ job scheduler, a commercial application ported to Windows by MPI Software Technology. Jobs can be entered on ClusterController using either a batch file or an interactive format. Once jobs are queued, ClusterController monitors the state of the cluster, starts and stops jobs, and delivers the output.

In addition to the basic components described in this section, the CCTP kit includes various tools to enable application porting from UNIX to Windows. These tools are continually updated as more applications and benchmarks become available.

**Deploying compute nodes on a Windows cluster**

One of the challenges in setting up any cluster is the deployment and installation of compute nodes. Microsoft Windows has a built-in tool known as Remote Installation Services (RIS), which can be used to...
deploy the Windows operating system over a network to a large number of nodes simultaneously. RIS was originally available with Microsoft Windows 2000 server products for the deployment of client operating systems. The Windows Server 2003 server family includes an advanced version of RIS capable of deploying server operating systems. The distribution is copied to the RIS server and custom files can be created to taper the node installation and automate the process.

Configuring the RIS server
The RIS server can be any server running Windows 2003, but it must be connected to either a Fast Ethernet or Gigabit Ethernet\(^1\) switch. The RIS server should be configured as a DHCP server and have DNS with Microsoft Active Directory® directory service enabled. (Note: The DHCP and RIS functions can be performed by separate systems or combined onto the same server. The following example assumes that they are combined.)

Using the Windows Server 2003 CD, RIS can be installed on the server as a Windows component. Executing risetup.exe on the RIS server will copy the necessary distribution files to the server and then set up services. This executable program starts copying all the necessary files to a directory specified by the user. Once this activity is complete, the RIS service should be set to Enabled.

The RIS server must be authorized in Active Directory. This authorization enables the RIS server to interact with the compute nodes on the network; if authorization is not set, the compute nodes cannot contact the RIS server.

User settings and permissions can be set to verify that a user can create accounts in the domain before deploying a compute node operating system. This function also is handled in Active Directory.

Setting up compute nodes
Each compute node should possess the minimum hardware required for the operating system deployment. Compute nodes must support the Preboot Execution Environment (PXE) mechanism or be a supported network adapter with a RIS startup disk, and PXE must be enabled in the BIOS as the first booting sequence. All compute nodes should be connected to the same network as the RIS server.

When the compute node PXE boots, it sends a DHCP request to the RIS server. After this contact is complete, an IP address is assigned by the RIS server—which, in this example, is also the DNS and DHCP server—to the compute node through the following process:

1. Compute node broadcasts a request for an IP address across the subnet
2. RIS server responds with a DHCP offer message (DHCPOFFER)
3. Client receives offer and responds
4. DHCP server sends an acknowledgment (DHCPACK)
5. Compute node sends a request for a boot server
6. RIS sends a boot server request offer
7. Compute node and RIS server complete the packet transfer process

Looking toward the future of Windows-based clustering
Windows high-performance clustering has been evolving over the past few years and will continue to develop with other technologies. Many applications and tools are available only in the Windows environment, and these may become some of the first applications and tools used for Windows-based clustering. Early adopters believe that Windows-based HPC clustering holds great potential and may capture a broad range of market segments in the future. 

Acknowledgments
The authors would like to thank Greg Rankich from Microsoft for reviewing and providing feedback for this article.

Munira Hussain (munira_hussain@dell.com) is a systems engineer in the Scalable Systems Group at Dell. She has a B.S. in Electrical Engineering and a minor in Computer Science from the University of Illinois at Urbana-Champaign. Her current research interests include prototyping commercial and software solution stacks. She has also worked on productizing HPC clusters on IA-32 and IA-64 platforms.

Jeffrey Mayerson (jeffrey_mayerson@dell.com) is a systems engineer in the Scalable Systems Group at Dell. He completed his degree in Computer Information Systems and Management from the University of Wisconsin-Madison.

Jenwei Hsieh, Ph.D. (jenwei_hsieh@dell.com) is an engineering manager of the Scalable Systems Group at Dell. He has a Ph.D. in Computer Science from the University of Minnesota and a B.E. from Tamkang University in Taiwan.

Kevin Noreen (kevin_noreen@dell.com) is a product marketing strategist for clustering at Dell and has been with the Dell Enterprise Systems Group since 1998. Kevin obtained his B.A. in Management Information Systems from the University of Iowa.

\(^1\) This term indicates compliance with IEEE® standard 802.3ab for Gigabit Ethernet, and does not connote actual operating speed of 1 Gbps. For high-speed transmission, connection to a Gigabit Ethernet server and network infrastructure is required.