The WRF project is an ongoing collaboration to develop a next-generation regional forecast model and data assimilation system for operational numerical weather prediction and atmospheric research (www.wrf-model.org). WRF consists of about 360,000 lines of Fortran 90 and C source code, and is parallelized using the MPI programming model. WRF has been ported to a wide variety of platforms including Linux Clusters and, more recently, AMD Opteron processor-based clusters running Microsoft Windows Compute Cluster Server 2003 and PGI® Windows x64 compilers.

WRF can be compiled using the PGF95™ Fortran compiler from the version 7.0 PGI CDK™ Cluster Development Kit™ on a Linux Cluster. The graphic to the left shows WRF running the hurricane Katrina forecast, which projects the path of Katrina over a 72 hour period based on an input dataset acquired August 27, 2005. In this case, the forecast was run on a cluster of four HP DL 145 G2 servers, each of which incorporates two dual-core AMD Opteron processors running Red Hat Enterprise Linux 4.0 on a Gigabit Ethernet interconnect. This enables WRF runs on up to 16 cores across the four servers. In this case, WRF is run as a pure MPI application across all of the available cores.

The PGI CDK package is available and supported on Linux Clusters running a wide variety of Linux distributions, and includes the PGF95 Fortran, PGCC® ANSI C99, and PGC++™ ISO/ANSI-compliant C++ compilers. It also includes the OpenMP and MPI-capable PGDBG® debugger and PGPROF® performance profiler, which greatly simplify the processing of porting, debugging and tuning HPC applications like WRF for multi-core AMD Opteron processor-based Linux Clusters.
PGDBG is a graphical MPI-parallel and OpenMP thread-parallel debugger for Linux clusters. PGDBG can debug programs on SMP servers, distributed-memory clusters, and hybrid clusters where each node contains multiple multi-core processors. Debugging an HPC cluster application is extremely challenging. PGDBG provides a comprehensive set of graphical elements to simplify the task. Using a single window, you have precise control over each MPI process in a cluster or thread in an SMP system. The main window displays F95, C99 or C++ source code, optionally interleaved with assembly, and one-touch commands to Run, Break, Quit, Print or print a Stack traceback. Additional buttons on the main window allow selecting and controlling individual or collective threads and processes, and the commands to examine their state. I/O can be displayed in the main window or in a separate window. PGDBG contains several MPI and OpenMP-specific features, including the ability to examine MPI message queues, auto-detect and attach to running MPI processes or OpenMP threads, distinguish between OpenMP shared and private data, and set up sub-groups of processes or threads for one-touch control. The process/thread GUI grid is color-coded to quickly display whether a process is running (green), stopped (red), hung on a signal (blue) or killed/exited (black).

The PGPROF profiler is an interactive, powerful but easy-to-use postmortem statistical analyzer for MPI-parallel and OpenMP thread-parallel F95, C or C++ applications. Like PGDBG, PGPROF can profile programs on SMP servers, distributed-memory clusters, and hybrid clusters where each node contains multiple multi-core processors. PGPROF allows profiling using instrumentation (inserting timer calls) at the function or line level, using gprof-style sampling, or using x64 hardware counters on systems with the Performance Application Programming Interface (PAPI). New in release 7.0 of the PGI compilers and tools, PGPROF includes the capability to analyze traces generated by the widely-used oprofile package for system-wide profiling using x64 hardware counters. PGPROF displays information in intuitive easy-to-use formats such as bar charts, percentages, counts or seconds. It allows you to quickly determine where execution time is spent and see which functions were called and how often. PGPROF displays counts of MPI sends, receives and other MPI communications. It includes a feature that allows the user to read in multiple PGPROF trace files from runs of the same application on different numbers of nodes/processors, and generates a scalability analysis that allows you to quickly determine which parts of your application are scaling well and which might be bottlenecks.

In addition to the PGI compilers and tools, the PGI CDK Cluster Development kit includes the following capabilities and features:

- Network floating multi-user seats for all PGI compilers and tools
- Outstanding single and multi-core performance on both AMD and Intel x64 processors
- Pre-compiled/pre-configured MPICH1 and MPICH2 message-passing libraries and utilities
- Pre-compiled/pre-configured TORQUE resource management and batch queuing system
- Pre-compiled ScaLAPACK math library
- AMD’s ACML library with hand-optimized BLAS and FFT routines for AMD processors
- Installation utilities to simplify the setup and management of your Linux Cluster

For detailed information on other research community cluster applications like WRF that can be built and maintained using the PGI CDK Cluster Development Kit, see the PGI tips and techniques pages online at: