

НАСТРОЙКА ВЫСОКОПРОИЗВОДИТЕЛЬНОГО ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА

Геворкян М.Н., Королькова А.В., Кулябов Д.С.

*Российский университет дружбы народов, кафедра систем телекоммуникаций
mngevorkyan@sci.pfu.edu.ru, akorolkova@sci.pfu.edu.ru, dharna@sci.pfu.edu.ru*

В работе представлен отчет о настройке и использовании высокопроизводительного вычислительного комплекса РУДН.

Ключевые слова: высокопроизводительные вычисления, параллельные вычисления, OpenMP, MPI, CUDA.

Введение

В связи с развитием многоядерной архитектуры центральных процессоров и технологии массового параллелизма на основе графических карт использование параллелизма в научных вычислениях стало возможно даже на небольших вычислительных кластерах с малым количеством узлов.

Аппаратная характеристика вычислительного комплекса

Описываемый высокопроизводительный вычислительный комплекс состоит из 3-х вычислительных узлов со следующими аппаратными характеристиками:

- один узел с двумя центральными процессорами Intel Xeon E5-2670 (2.60 GHz, 8 ядер, 16 потоков), 64 Гб ОЗУ;
- два узла с одним центральным процессором Xeon E5-2670, 64 Гб ОЗУ и одной видеокартой Nvidia Tesla M2090 (512 CUDA ядер, 6 Гб памяти) на каждом узле.

Система может быть расширена путем добавления новых вычислительных узлов.

Выбор такой конфигурации дает возможность проводить как параллельные вычисления с использованием центральных процессоров (в общей сумме 32 ядра или 64 потока) с применением технологий OpenMP, MPI и др., так и с помощью графических карт при использовании технологии CUDA.

Установленное программное обеспечение

На вычислительный комплекс был установлен GNU Compiler Collection (gcc) [1], в который включены компиляторы для языков C, C++ и Fortran (gfortran версии 4.4.7 с поддержкой OpenMP) [2] и интерпретатор языка Python версии 2.7. Также были установлены следующие библиотеки:

- OpenMPI и MPICH — для параллельных вычислений с использованием центральных процессоров [3-4];
- BLAS и LAPACK — библиотеки, реализующие численные методы решения задач линейной алгебры на языке Fortran, а также интерфейсы к ним для использования с языком C [5-6];
- библиотека GSL (GNU Scientific Library) [7];
- GAMESS (US) (General Atomic and Molecular Electronic Structure System) — пакет программ для вычислений в области квантовой химии [8];
- Python numpy и scipy — библиотеки, которые реализуют широкий спектр численных алгоритмов [9].

Основные этапы настройки системы управления заданиями

На первом этапе для всех узлов вычислительного комплекса была настроена единая система аутентификации пользователей. Это было необходимо для возможности запуска задачи на всех узлах от имени одного и того же пользователя. Следует также учесть, что домашние каталоги пользователей должны быть доступны на всех узлах вычислительной системы.

Далее был настроен беспарольный доступ по протоколу ssh с управляющего узла на вычислительные и обратно. Это необходимо для передачи заданий с центрального узла на вычислительные и возвращения результата.

Следующий этап — установка и конфигурация менеджера вычислительных ресурсов PBS/Torque [10]. В процессе настройки необходимо было прописать конфигурацию узлов и выделить один из них в качестве сервера. Далее было создано несколько очередей для различных задач. При создании очереди задаются такие параметры, как максимально доступное число процессоров, объем памяти и процессорное время.

Выбор реализации MPI

На вычислительный комплекс были установлены две различные реализации MPI: OpenMPI и MPICH. Обе реализации нельзя использовать одновременно, но есть возможность переключения между ними с помощью утилиты mpi-selector. После выбора конкретной реализации все команды MPI будут доступны пользователю.

Выводы

Вышеописанная конфигурация открывает большие возможности для самых различных научных вычислений с использованием параллелизма. В будущем возможна установка дополнительного программного обеспечения, а также расширение аппаратной части дополнительными вычислительными узлами.

Литература

1. GCC, the GNU Compiler Collection — <http://gcc.gnu.org/>.
2. OpenMP API specification for parallel programming — <http://openmp.org/wp/>.
3. Open MPI: Open Source High Performance Computing — <http://www.open-mpi.org/>.
4. MPICH: High-Performance Portable MPI — <http://www.mpich.org/>.
5. BLAS (Basic Linear Algebra Subprograms) — <http://www.netlib.org/blas/>.
6. LAPACK — Linear Algebra PACKage — <http://www.netlib.org/lapack/>.
7. GSL - GNU Scientific Library — <http://www.gnu.org/software/gsl/>.
8. The General Atomic and Molecular Electronic Structure System (GAMESS) — <http://www.msg.ameslab.gov/gamess/gamess.html>.
9. SciPy/NumPy — <http://scipy.org/>.
10. TORQUE Resource Manager - Adaptive Computing — <http://www.adaptivecomputing.com/products/open-source/torque/>.

HIGH PERFORMANCE COMPUTING SYSTEM CONFIGURATION

Gevorkyan M.N., Korolkova A.V., Kulyabov D.S.

*Peoples' Friendship University of Russia, Telecommunication Systems Department
mnggevorkyan@sci.pfu.edu.ru, akorolkova@sci.pfu.edu.ru, dharma@sci.pfu.edu.ru*

This paper presents a report about configuration and utilization of high-performance computing systems of Peoples' Friendship University of Russia.

Key words: high performance computing, parallel computing, OpenMP, MPI, CUDA.