

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.4:004.7

М.К. Буза, О.М. Кондратьева, В.О. Шукело

ГРАФИЧЕСКАЯ СИСТЕМА ПОДГОТОВКИ И УПРАВЛЕНИЯ ЗАДАНИЯМИ ПОЛЬЗОВАТЕЛЕЙ НА КЛАСТЕРЕ

Предлагается приложение, которое позволяет управлять прохождением задач на суперкомпьютере и имеет графический интерфейс. Приложение поддерживает реализации на языке C/C++ на основе технологий MPI, OpenMP, POSIX Threads.

Введение

Применение высокопроизводительных распределенных вычислений открывает широкие возможности для решения сложных задач. Это обуславливает возникновение различных проблем: от распараллеливания алгоритмов и оптимизации их работы на современных компьютерах до разработки средств поддержки доступа к вычислительным системам.

В настоящее время доступ к вычислительным ресурсам кластера предъявляет достаточно высокие требования к квалификации пользователей. Им необходимо владеть многими дополнительными навыками, такими как копирование файлов на суперкомпьютер и обратно, получение доступа к командной строке кластера, умение работать с командной строкой. Поэтому актуальной является задача разработки графического интерфейса для упрощения доступа пользователей к кластеру. Такой интерфейс позволит расширить круг пользователей суперкомпьютеров.

Задача интерфейса – обеспечить выполнение всех действий, необходимых пользователю, только средствами интерфейса. Правильно организованный интерфейс позволяет абстрагироваться от многих деталей, связанных с доступом к кластеру.

В данной работе рассматривается графическая система, которая автоматизирует основные операции, выполняемые пользователями кластера: управление файлами, компиляцию и запуск задач, а также мониторинг вычислительного процесса. Проект выполнен при частичной поддержке ГПНИ «Информатика и космос». Апробация всего функционала разработанного интерфейса осуществлялась на суперкомпьютере СКИФ БГУ.

1. Суперкомпьютер СКИФ БГУ

На суперкомпьютере СКИФ-1000-2 установлена операционная система (ОС) Linux Fedora 8.0, которая обеспечивает основу для выполнения всех остальных программ. Удаленный доступ к суперкомпьютеру СКИФ осуществляется по протоколу SSH (Secure Shell).

Для запуска заданий пользователей на кластере используется система управления заданиями Torque PBS (Portable Batch System). Основная функция PBS – запуск задач в вычислительной среде по расписанию. Torque PBS наиболее часто используется для управления вычислительным процессом в кластерах, построенных из компьютеров под управлением ОС GNU/Linux и других Unix-подобных ОС.

Доступ к суперкомпьютеру осуществляется с любого компьютера локальной сети БГУ. Для работы на суперкомпьютере необходимо установить на персональном компьютере программы, которые обеспечат удаленный доступ к командной строке сервера и возможность обмена файлами между компьютером и сервером.

К основным операциям, которые выполняют пользователи кластера, относятся: файловые, компиляция и запуск приложений, слежение за процессом выполнения задачи. Для обмена файлами между компьютером и сервером существуют программы с графическим интерфейсом. Например, пользователь, у которого на локальной машине установлена ОС Windows, может

использовать для файловых операций программы WinSCP или FileZilla, а для подключения к командному интерфейсу кластера – SSH-клиент PUTTY.

Подключившись к кластеру, пользователь осуществляет традиционные операции: компиляцию и запуск приложений, просмотр результатов, слежение за процессом выполнения. Управлять суперкомпьютером приходится посредством интерфейса командной строки, что для неподготовленного пользователя может оказаться неудобным и сложным.

2. Средства поддержки доступа к кластеру

В последнее время начинают разрабатываться средства для упрощения доступа к вычислительным ресурсам кластера. Существуют некоторые разработки для запуска заданий пользователей на кластере посредством графического интерфейса, например графический интерфейс управления задачами ОС Clustrx [1] и система управления суперкомпьютером компании Мелкон [2].

Суперкомпьютерная ОС Clustrx является разработкой российской компании T-Massive Computing. ОС имеет интерактивный графический веб-интерфейс пользователя, который реализует функции управления файлами, задания параметров и запуска задачи на выполнение, слежения за состоянием очереди задач. Возможности, предоставляемые интерфейсом, различаются для пользователей разного уровня привилегированности.

Система управления суперкомпьютером украинской компании Мелкон представляет собой веб-портал для управления кластером. Система объединяет интерфейс пользователя и средства администратора. Интерфейс пользователя предоставляет сервисы компиляции и запуска задач, работы с файлами пользователя, обмена сообщениями с другими пользователями и администраторами. Интерфейс администратора имеет возможности управления пользователями, мониторинга кластера, диагностики, просмотра статистики использования ресурсов, уведомления о неполадках.

Обе системы реализуют необходимый функционал и имеют графический интерфейс. Однако первая представляет собой специализированную ОС, а для управления второй требуется участие системного администратора. Эти проекты поддерживают доступ к кластеру через графический интерфейс, тем не менее сохраняя при этом стиль ОС Linux. Кроме того, они являются коммерческими программными продуктами. Поэтому задача разработки собственной графической системы, покрывающей функции разноуровневого пользователя, является актуальной и практически значимой.

В данной работе предлагается система управления кластером, которая обладает следующими свойствами:

- упрощает подготовку и управление заданиями пользователей;
- ориентирована на пользователей с неадминистративным уровнем полномочий, работа которых с кластером заключается в выполнении на нем вычислительных задач;
- имеет удобный графический интерфейс, через который осуществляется взаимодействие с системным программным обеспечением суперкомпьютера;
- не нуждается в дополнительной настройке со стороны сервера;
- требует для работы только установленную виртуальную машину Java (JVM версии 1.6 и выше).

3. Структура и функции графической системы

Система подготовки и управления заданиями состоит из двух основных частей: ядра и пользовательского интерфейса. Ядро системы отвечает за установление подключения к кластеру. Оно взаимодействует с ОС и PBS, выполняя всю работу по обслуживанию запросов от интерфейса пользователя. Основой визуальной части системы является настольное приложение.

Система поддерживает программные реализации на языке C/C++ на основе технологий MPI, OpenMP, POSIX Threads. Ее легко можно настроить на работу с прикладными пакетами, например с пакетом молекулярной динамики LAMMPS.

Основным объектом, с которым работает система, является задача. Она выполняется на сервере и представляет собой программу с ее входными данными и параметрами для обработки. Система создает и поддерживает на сервере базу данных, содержащую сведения о доступных задачах. Каждая задача характеризуется набором параметров. Значения одних параметров задаются через интерфейс (пользователем или по умолчанию), других – определяются системой. Параметры и их назначение приведены в таблице.

Параметры задачи

Название	Описание
Name	Имя задачи
Status	Состояние задачи
Task id	Идентификатор задачи, присваиваемый системой при создании задачи и используемый как имя директории на сервере
Source file	Путь к файлу или директории с исходным кодом задачи на локальном компьютере
Source mode	Тип задачи, определяемый используемой технологией (MPI, OpenMP, POSIX Threads)
Input file	Путь к файлу или директории с входными данными для задачи на локальном компьютере
Output file	Путь к файлу или директории для выходных данных на локальном компьютере
Node count/list	Количество или список узлов, запрашиваемых пользователем
Walltime	Время, которое отводится на выполнение задачи. В случае если задача не успеет выполниться за это время, она будет принудительно завершена
Add-al cmdpars	Параметры командной строки
Last run time	Время выполнения задачи
Completed date	Дата и время завершения задачи

Задача имеет различные представления: запись в базе данных, директория на сервере, запись в Torque PBS. Во время своего жизненного цикла задача может находиться в одном из следующих состояний:

- `created` – созданная;
- `compilation-failed` – с ошибками компиляции;
- `compiled` – успешно скомпилированная;
- `scheduled` – добавленная в очередь на выполнение;
- `running` – выполняемая в данный момент;
- `completed` – нормально завершившаяся;
- `aborted` – завершившаяся неудачно или отмененная;
- `purged` – удаленная с сервера.

4. Графический интерфейс пользователя

Интерфейс и возможности системы соответствуют пользователям двух типов:

– ученые-исследователи, которые работают на кластере с готовыми пакетами и выполняют подготовку файлов входных данных, запуск приложений и просмотр файлов выходных данных;

– прикладные программисты, которые используют кластер как инструмент настройки параллельных программ. Программистам необходимо компилировать и запускать задачи и следить за процессом их выполнения.

Рассмотрим действия, которые необходимо выполнять всем пользователям: подключение к кластеру и управление своими задачами.

Подключение к суперкомпьютеру. После запуска графического интерфейса появляется окно входа в систему. Здесь пользователь вводит имя и пароль. Каждому потенциальному пользователю логин и пароль выдаются администратором кластера по письменному заявлению. Уровень доступа определяется статусом пользователя. Единые идентификационные данные для группового доступа не предусмотрены.

После подключения к суперкомпьютеру открывается главное окно интерфейса (рис. 1). В нем отображается список собственных задач пользователя. Для каждой задачи в этом списке указываются: состояние; расположение исходного кода на локальном компьютере; имя, данное пользователем, и идентификатор, присвоенный системой. Список задач автоматически обновляется при изменении их состояния.

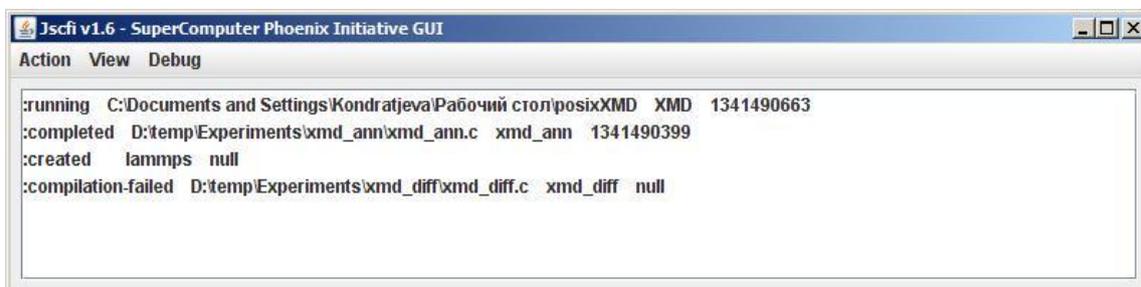


Рис. 1. Главное окно системы

Управление задачами включает следующие акции.

Создание новой задачи. При создании задачи пользователь определяет значения ее параметров в соответствующем окне (рис. 2). После создания задача переходит в состояние **created**. Значения параметров, связанных с местоположением файлов, задаются относительно локального компьютера. Система сама при необходимости выполняет файловые операции между локальным компьютером и суперкомпьютером, а пользователь работает с файлами, используя привычное для себя программное обеспечение.

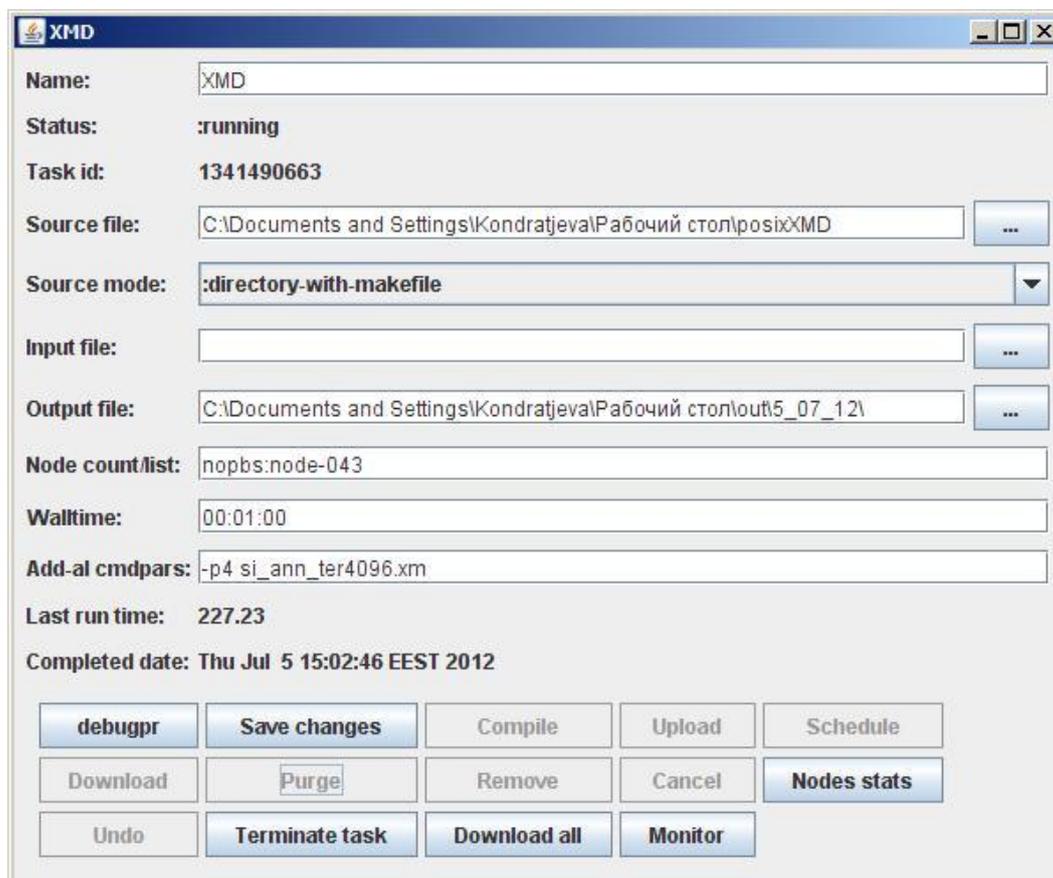


Рис. 2. Окно создания новой задачи

Компиляция и выполнение задачи. Созданную задачу нужно откомпилировать и в случае успешной компиляции запустить на выполнение.

Файлы с исходными текстами программ, расположенные на локальном компьютере, сначала копируются на кластер, а затем компилируются. После успешной компиляции задача переходит в состояние **compiled**.

При ошибках компиляции задача переходит в состояние **compilation-failed**. Если пользователь вносит изменения в файлы с исходными текстами программ, необходимо обновить эти файлы на кластере (команда **Upload**) и провести повторную компиляцию (**Compile**).

Для задачи в состоянии **compiled** пользователь может:

- скопировать входные данные на сервер (**Upload**);
- запустить задачу на выполнение (**Schedule**).

При выборе второго действия система создает скрипт для запуска и помещает задачу в очередь на выполнение. Задача переходит в состояние **scheduled**. После того как задача начинает выполняться, она переходит в состояние **running**. В случае успешного завершения задачи ее статус меняется на **completed** и в поле Last run time отображается время, которое прошло от старта до завершения задачи. В случае неуспешного или принудительного (**Terminate task**) завершения задача переходит в состояние **aborted** и в поле Last run time отображаются время выполнения и код завершения задачи.

Для задачи в состоянии **completed** пользователь может:

- скопировать выходные данные на локальный компьютер (**Download**);
- подготовить новые входные данные и запустить задачу;
- удалить файлы задачи с сервера (**Purge**), задача переходит в состояние **purged**;
- удалить задачу из списка задач (**Remove**).

Копирование результатов. Вычислительная задача пользователя может выводить данные в файлы, находящиеся на кластере. Чтобы скопировать эти файлы на локальную машину, пользователь должен выполнить команду **Download**.

Таким образом, пользователи, работающие с готовыми пакетами, могут придерживаться следующего сценария:

- создавать требуемые задачи;
- запускать созданные задачи на выполнение;
- наблюдать за своими задачами при необходимости;
- просматривать файлы выходных данных для завершившихся задач.

5. Мониторинг вычислительного процесса

Анализ эффективности параллельной программы представляет собой сложный процесс, для поддержки которого существуют различные вспомогательные программные средства [3], например мониторинг вычислительного процесса. Средства мониторинга, как правило, собирают и накапливают данные о функционировании и производительности приложений. Эти данные используются для анализа работы приложений и могут быть полезны как ученым-исследователям, так и прикладным программистам для оптимизации процесса вычислений.

Средства мониторинга зависят от того, кто и для чего их использует. Это может быть анализ работы вычислительной системы в целом [4]. Пользователям суперкомпьютера необходим мониторинг своих приложений. И здесь возможны варианты. Одним достаточно знать только время выполнения задачи, другим требуется более детальная информация для исследования параллельных алгоритмов и оптимизации их реализаций: сколько времени приложение находилось в каждом из состояний, на каких узлах оно выполнялось, как выполнялось приложение по сравнению с предыдущими запусками.

Мониторинг системы можно осуществлять, используя команды ОС Linux [5, 6]. Например, команда **top** выводит в консоль список процессов с сортировкой по процессорному времени. В этом списке отображается, в частности, состояние процесса (выполняется, ожидает и т. д.) и занимаемое им процессорное время. Система управления заданиями Torque PBS позволяет получить аналогичную информацию с помощью команды **qstat**. Периодически вызывая команду

`top` или `qstat`, можно контролировать состояние своих задач, что требует некоторых усилий со стороны пользователя. При этом пользователь видит не только свои задачи, но и задачи других пользователей. В предлагаемой системе в главном окне отображается обновляемый список собственных задач пользователя (см. рис. 1).

Команда `time` ОС Linux позволяет выполнить программу и получить информацию о времени, которое прошло от запуска до завершения задачи. В разработанном интерфейсе эта информация после завершения задачи выводится в поле Last run time (см. рис. 2).

По команде `pbsnodes` системы управления заданиями Torque PBS отображается текущее состояние вычислительных узлов кластера. Это действие также реализовано в разработанной системе (Nodes stats).

На эффективность параллельной программы влияют, в частности, конфигурация компьютера и системное программное обеспечение. Поэтому программисты исследуют свои реализации и адаптируют их к конкретным многопроцессорным вычислительным системам. Если время расчета задачи оказывается неудовлетворительным, программист нуждается в дополнительных данных о функционировании программы. Это могут быть уровень загрузки процессоров, объем занятой задачей оперативной памяти, интенсивность ввода/вывода. ОС Linux ведет системные журналы, в которых накапливает необходимые сведения. Доступ к этим журналам требует определенных навыков. Предлагаемая система позволяет запустить приложение в режиме мониторинга, в котором для вычислительных узлов задачи сохраняются данные для анализа производительности приложения.

Просмотреть результаты мониторинга можно в текстовом и графическом виде. В качестве примера на рис. 3 представлены данные по загрузке вычислительного узла, на котором работает двухпоточное приложение. Пользователи могут подключать к системе свои визуализаторы.

```
node-043 1341485637.527430 174.207
node-043 1341485641.529430 179.410
node-043 1341485645.531450 171.413
node-043 1341485649.533190 161.430
node-043 1341485653.534200 157.460
node-043 1341485657.537440 159.121
node-043 1341485661.538430 158.961
node-043 1341485665.540190 156.931
node-043 1341485669.542210 158.420
node-043 1341485673.543200 156.211
```

а)



б)

Рис. 3. Данные мониторинга «Загрузка вычислительного узла»: а) фрагмент текстового файла; б) график

Разработанная система предоставляет пользователю возможность запустить свое задание как под управлением PBS, так и без него. Во втором случае программист должен перечислить узлы, на которых он хочет запустить задачу. Используя различные варианты запусков, можно получить реальные данные о влиянии конфигурации компьютера на программу.

Заключение

Разработанная графическая система автоматизирует основные операции, выполняемые пользователями суперкомпьютера. Система предоставляет исследователям возможность работать на кластере со своими пакетами, используя привычное программное обеспечение, а программистам оказывает помощь в разработке действительно эффективных параллельных программ.

Полагаем, что разработанный графический интерфейс будет способствовать более широкому использованию отечественных многопроцессорных вычислительных систем за счет существенного упрощения доступа пользователей к кластеру.

Список литературы

1. Кластерный комплекс МГУ «Ломоносов». Руководство пользователя [Электронный ресурс]. – 2012. – Режим доступа : http://parallel.ru/sites/default/files/cluster/T500_user_guide-3.pdf. – Дата доступа : 29.06.2012.
2. Система управления кластером [Электронный ресурс]. – 2012. – Режим доступа : <http://melkon.com.ua/ru/cms/sistema-upravleniya-klasterom.html>. – Дата доступа : 29.06.2012.
3. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – СПб. : БХВ-Петербург, 2002. – 608 с.
4. Кореньков, В.В. Архитектура и пути реализации системы локального мониторинга ресурсного центра / В.В. Кореньков, П.В. Дмитриенко // Электронный журнал «Системный анализ в науке и образовании». – 2011. – № 3.
5. Колисниченко, Д. Linux. От новичка к профессионалу / Д. Колисниченко. – СПб. : БХВ-Петербург, 2011. – 656 с.
6. Фуско, Дж. Linux. Руководство программиста / Дж. Фуско. – СПб. : Питер, 2011. – 448 с.

Поступила 06.07.2012

*Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: bouza@bsu.by
kondratjeva@bsu.by
vi0oss@gmail.com*

М.К. Bouza, О.М. Kondratjeva, V.O. Shukelo

GRAPHICAL SYSTEM FOR PREPARE AND CONTROL OF USER TASKS ON A CLUSTER

An application for control of tasks on a supercomputer is developed. The application has a graphical interface and supports C/C++ realization on the basis of MPI, OpenMP and POSIX Threads technologies.