ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ ИНТЕГРАТОРА ЛОГОС-МИП С СУПЕР-ЭВМ

А. А. Тюндина, А. Г. Надуев, А. Д. Черевань, Д. А. Жуков, Д. А. Кожаев

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Введение

Модульная интеграционная платформа (МИП) предназначена для решения ряда задач в рамках пакета программ «ЛОГОС», в том числе для подготовки и проведения связанных расчетов комплексных мультидисциплинарных задач математического моделирования.

В состав МИП входит Интегратор, который обеспечивает:

– копирование данных задач с локального компьютера на супер-ЭВМ;

– постановка задач в очереди систем пакетной обработки (СПО)SLURM и JAM;

– получение статуса и управление запущенными задачами;

– передача результатов расчета задач с супер-ЭВМ на локальный компьютер пользователя.

Для решения поставленных задач в Интеграторе реализована подсистема взаимодействия с супер-ЭВМ, обеспечивающая:

– передачу файлов между локальным компьютером и супер-ЭВМ с использованием протокола SFTP;

– постановку задач в очередь планировщика;

– отслеживание статуса задачи и возможность отмены задачи на супер-ЭВМ с использованием протокола SSH;

– автоматическое формирование скриптов запуска расчетных математических методик под разные планировщики SLURM и JAM.

Подсистема взаимодействия с супер-ЭВМ

Разработанная подсистема взаимодействия с супер-ЭВМ позволяет использовать возможности протоколов SSH и SFTP для выполнения различных операций с файлами и каталогами между локальным компьютером пользователя и супер-ЭВМ, а также запуска задач в рамках проведения мультидисциплинарных вычислений на супер-ЭВМ.

SFTP – sshfiletransferprotocol (англ.) – протокол прикладного уровня, предназначенный для копирования и выполнения других операций с файлами поверх надежного и безопасного соединения [1].

SSH – secureshell (англ.) – сетевой протокол прикладного уровня, позволяющий производить удаленное управление операционной системой и туннелирование соединений [2].

Подсистема взаимодействия с супер-ЭВМ состоит из двух модулей:

– модуль удаленного доступа QtSsh;

– модуль удаленного запуска.

Модуль удаленного доступа QtSsh

Модуль QtSsh – программный модуль, созданный для работы с удаленной машиной (например, копирование файлов или запуск задач), используя возможности протоколов SSH и SFTP.

В процессе предоставления удаленного доступа к файловой системе супер-ЭВМ выполняются следующие действия:

– устанавливается ssh-соединение;

– осуществляется идентификация и аутентификация пользователя на удаленном компьютере;

– создается канал ssh-соединения;

– инициируется sftp-сессия;

– выполняется одна из следующих операций:

– копирование файлов / каталогов с локального компьютера на удаленный компьютер;

– копирование файлов / каталогов с удаленного компьютера на локальный компьютер;

– удаление файла / каталога;

– запуск расчетной задачи на удаленном компьютере.

– завершение sftp-сессии;

– закрытие канала ssh-соединения;

– завершение ssh-соединения.

Модуль QtSsh выполнен в виде отдельной динамической библиотеки и состоит из двух уровней:

– интерфейс взаимодействия динамической библиотеки с пользовательским приложением. Интерфейс разработан в виде C++ класса Ssh. В его задачи входит предоставить пользователю основные функции и скрыть детали реализации;

– основная часть, реализующая функциональность библиотеки QtSsh, которая состоит из нескольких классов, реализует взаимодействие с функциями libssh [3].

Модуль удаленного запуска

Взаимодействие с системой управления расчетами на супер-ЭВМ осуществляется при помощи модуля удаленного запуска, который разработан с учетом использования различных систем пакетной

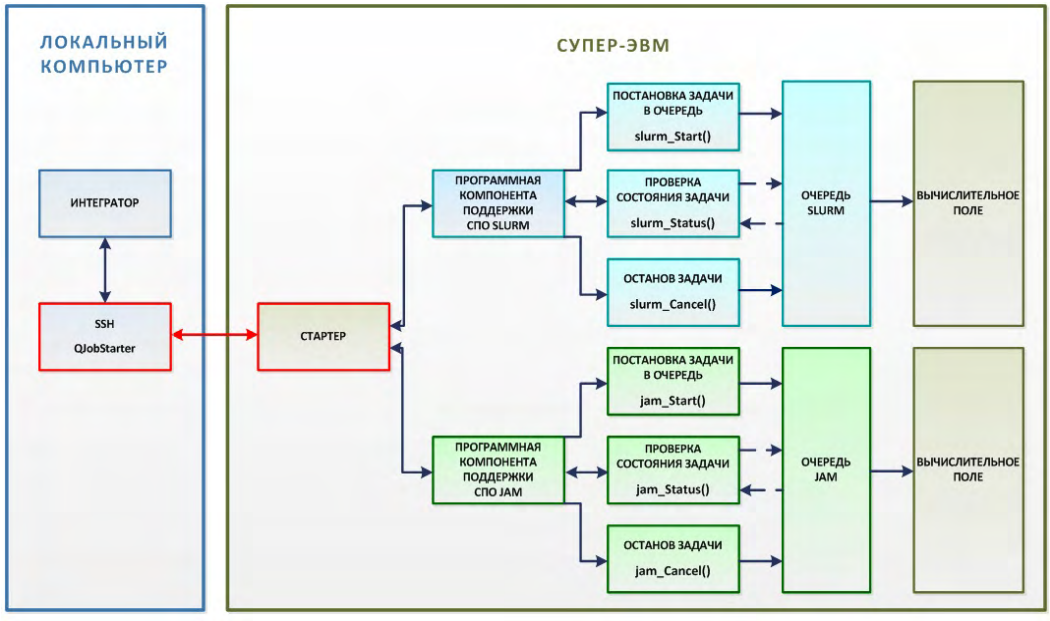

Рис. 1. Общая схема взаимодействия модуля удаленного запуска с использованием различных СПО

обработки заданий и обеспечивает следующее взаимодействие локального компьютера и супер-ЭВМ:

– автоматическое формирование скриптов пакетных заданий в зависимости от используемой системы управления заданиями и данных, предоставляемых пользователями;

– непосредственный запуск задачи на супер-ЭВМ (постановка задачи в очередь) с последующим формированием сигнальных файлов начала и завершения текущей задачи;

– отслеживание статуса запущенной задачи на всем протяжении ее выполнения с последующим возвратом кода состояния пользователю;

– отмена задачи по требованию пользователя.

На рис. 1 представлена общая схема взаимодействия модуля удаленного запуска с использованием различных планировщиков заданий.

Модуль удаленного запуска состоит из двух программных компонент, обеспечивающих взаимодействие локального компьютера с супер-ЭВМ, с использованием протокола SSH:

– первая программная компонента – *библиотека QJobStarter*;

– вторая программная компонента – *стартер запуска заданий* с использованием различных планировщиков (SLURM и JAM).

Библиотека QJobStarter

Библиотека QJobStarter функционирует на локальном компьютере и предоставляет интерфейс выполнения команд для удаленного запуска заданий на супер-ЭВМ с помощью стартера.

Задание – это командный файл интерпретатора shell, который содержит информацию управления и резервирования вычислительных ресурсов. В задании обычно указывается сценарий (программа) запуска исполняемого модуля программы и название содержащего его файла.

Стартер – это программа постановки задачи в очередь планировщика заданий, получения статуса задачи, удаления задачи из очереди заданий.

Программный интерфейс библиотеки включает в себя следующий набор функций:

– функция авторизации удаленного соединения в соответствии с используемыми паролем и логином пользователя;

– функция выполнения запуска заданий на супер-ЭВМ с локального компьютера при помощи стартера;

– функции копирования файлов из / в рабочую директорию локального компьютера в / из рабочей директории супер-ЭВМ.

Стартер запуска заданий

Стартер запуска заданий состоит из двух программных компонент, обеспечивающих взаимодействие локального компьютера с супер-ЭВМ, с использованием протокола SSH:

– программная компонента поддержки планировщика SLURM;

– программная компонента поддержки планировщика JAM.

Такая реализация стартера обусловлена несовместимостью команд планировщиков SLURM и JAM, а так же различными форматами данных выходной информации, предоставляемой этими планировщиками.

Данные программные компоненты обеспечивают взаимодействие пользователя с соответствующими планировщиками посредством удаленного выполнения их команд и предоставляют пользователю следующий набор функций:

– _Start() – осуществляет подготовку и запуск задания в пакетном режиме в соответствии с входной информацией, содержащейся в конфигурационном файле, предоставляемым пользователем. Обеспечивает запись информации о начале выполнения задания в соответствующий сигнальный файл;

– _Status() – обеспечивает отслеживание хода выполнения задания на всем протяжении времени счета с последующим возвратом статуса задания пользователю;

– _Cancel() – осуществляет принудительное завершение задания с последующей записью соответствующей информации в сигнальный файл.

Механизм реализации вызова данных функций осуществляется из командной строки стартера с помощью определенного набора команд и аргументов.

Формирование скрипта запуска и запуск задания

Подготовка скрипта запуска задания для соответствующего планировщика осуществляется автоматически на основе конфигурационного файла, предоставляемого пользователем и в соответствии с исходными параметрами задания.

На рис. 2 представлена структурная схема механизма формирования скрипта запуска для планировщиков SLURM и JAM на основе конфигурационного файла, предоставляемого пользователем.

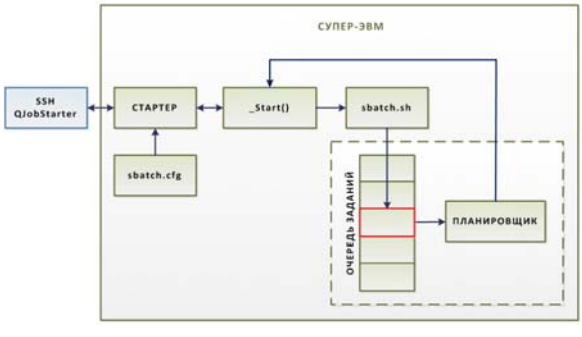

Рис. 2. Механизм формирования скрипта запуска и запуск задания

Данный механизм состоит из трех основных этапов:

– этап 1 – загрузка конфигурационного скрипта в стартер;

– этап 2 – формирование скрипта запуска на основе информации, содержащейся в конфигурационном файле;

– этап 3 – постановка задачи в очередь на счет, с последующим возвращением кода завершенной операции стартеру.

При этом в обоих случаях сигнальный файл о начале хода выполнения задания будет сформирован только в случае фактического старта планировщиком соответствующего задания. На момент ожидания задания в очереди, файл не формируется стартером, так как фактически задание еще не запущено, а только ожидает выделения соответствующих ресурсов.

Отслеживание статуса исполняемой задачи

Отслеживание статуса исполняемой задачи может осуществляться пользователем на всем протяжении времени счета. При этом каждый раз после выполнения соответствующей команды пользователю будет возвращен код текущего состояния задания на основе информации, полученной от планировщика и в соответствие с его уникальным идентификатором.

На рис. 3 представлена структурная схема механизма мониторинга текущего состояния задания в процессе его работы в соответствие с его идентификатором для планировщиков SLURM и JAM.

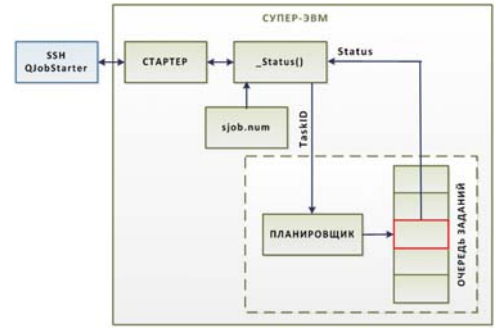

Рис. 3. Механизм мониторинга текущего состояния задачи по ее идентификатору

В процессе запуска задания стартером создается файл, в котором хранится его уникальный идентификатор, присвоенный планировщиком. В данном случае стартер берет на себя функцию хранения идентификатора запущенной задачи в соответствующем файле для дальнейшего мониторинга состояния этой задачи в любой момент времени.

Удаление задачи из очереди

В процессе выполнения задания, помимо функции мониторинга пользователю предоставляется возможность принудительного завершения задания с последующей записью соответствующей информации в сигнальный файл. На рис. 4 представлена структурная схема механизма принудительного завершения задания в соответствии с его идентификатором.

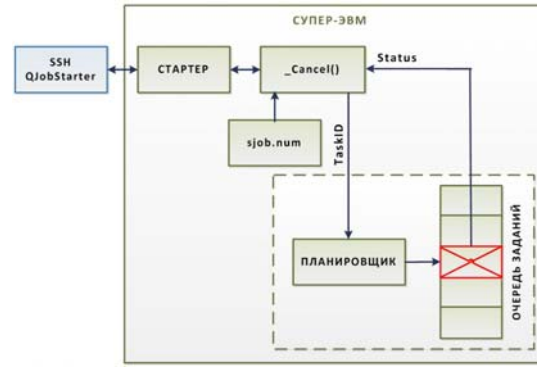

Рис. 4. Механизм принудительного завершения задания по ее идентификатору

Как и в случае команды получения статуса исполняемого задания, данный механизм основывается на исполнении планировщиком соответствующей команды, в результате чего из очереди будет выбрано задание с идентификатором, хранящимся в файле (sjob.num или jamjob.num) и произведено удаление. В процессе удаления задания из очереди, стартером будет сформирован сигнальный файл, в котором будет записана соответствующая информация, а пользователю будет возвращен код соответствующей операции.

Заключение

В докладе описана организация взаимодействия Интегратора ЛОГОС-МИП с Супер-ЭВМ, которая обеспечивает Интегратор возможностью взаимодействия с Супер-ЭВМ для подготовки и расчета задач оптимизации и параметрических исследований распределено, с учетом особенностей инфраструктуры Супер-ЭВМ.

Литература

1. Протокол SFTP [Электронный ресурс]: Википедия – Режим доступа: https://ru.wikipedia.org/wiki/SFTP.
2. Протокол SSH [Электронный ресурс]: Википедия – Режим доступа: https://ru.wikipedia.org/wiki/SSH.
3. LibSSH – The SSH Library [Electronical resource]: www.libssh.org / Thelibssh team – Lausanne, Switzerland, [2017] – Mode of access: https://api.libssh.org/master/index.html.