ЗАЩИТА ДАННЫХ В ПРОГРАММЕ КОМБИНАТОРНО-ЛОГИЧЕСКИХ ВЫЧИСЛЕНИЙ ДЛЯ КЛАСТЕРНОГО КОМПЬЮТЕРА

Д.И.Черемисинов

Объединенный институт проблем информатики НАН Б
ул. Сурганова 6, г. Минск, Беларусь
телефон(ы): + (375) 2842076; e-mail: cher@newman.bas-net.by
web: http://www.uiip.bas-net.by/~logic/

Обсуждается проблема предотвращения несанкционированного использования программы комбинаторно-логических вычислений для кластерного компьютера. Решением проблемы создания безопасного канала связи и управления между частями приложения предлагается использование протокола SSL/TLS вместо протокола TCP/IP.

Ключевые слова - Кластерный компьютер, защита данных, протокол SSL/TLS

1 ВВЕДЕНИЕ

В течение последних лет многопроцессорные системы типа MIMD все шире применяются, чтобы обеспечить вычислительные ресурсы для вычислений огромной трудоемкости. Наиболее популярным типом MIMD систем в настоящее время является кластерный компьютер. Вычислительный кластер – это набор компьютеров (вычислительных узлов), объединенных некоторой коммуникационной сетью. При этом общей физической оперативной памяти для узлов не существует. Каждый вычислительный узел имеет свою оперативную память и работает под управлением своей операционной системы. Наиболее простыми для программирования является однородные кластеры, у которых все узлы абсолютно одинаковы по своей архитектуре и производительности. Узлы очень часто представляет собой компьютер типа IBM PC, работающий под управлением ОС Linux, кластерные компьютеры такого типа называются кластерами Беовульф. Головной (управляющий) узел управляет всем кластером и является файл-сервером для вычислительных узлов. Он также является консолью кластера и шлюзом во внешнюю сеть. Большинство программ для кластерных компьютеров используют для обмена данными между процессами метод рассылки сообщений – MPI [1].

Потребность в решении комбинаторных задач и, следовательно, в разработке комбинаторно-логических алгоритмов возникает во многих областях, в том числе таких, как автоматизированное проектирование, создание систем искусственного интеллекта, разработка сетей связи и криптография. Под комбинаторно-логическими задачами подразумеваются *перечислительные* и *поисковые* задачи на конечных множествах, элементами которых служат объекты, представляющие собой комбинации элементов других конечных множеств - сочетания, перестановки, разбиения, покрытия, решения систем логических уравнений и т.п. Почти все комбинаторно-логические алгоритмы имеют экспоненциальную сложность. Вместе с тем на практике необходимо уметь решать комбинаторно-логические задачи достаточно больших размерностей. Одна из возможностей уменьшения времени решения комбинаторно-логические задач состоит в том, чтобы находить решение параллельно, т.е. одновременно на нескольких процессорах вычислительной системы.

Проблемы разработки параллельных алгоритмов решения комбинаторно-логических задач давно и интенсивно исследуются. Широкий спектр задач, относящихся к классу комбинаторно-логических, обеспечивает проектирование СБИС. Большинство этих задач для параметров сложности, встречающихся в практике современного проектирования, находятся на грани возможностей современной вычислительной техники.

2 АРХИТЕКТУРА РАСПРЕДЕЛЕННОЙ ПРОГРАММЫ

В ОИПИ НАН Беларуси в рамках союзной программы «Триада» разработан программный комплекс для решения ряда практически важных комбинаторно-логических задач [2]. Комплекс представляет собой распределенную программу, в которой комбинаторно-логические вычисления выполняет MPI-программа для кластерного компьютера «СКИФ-К1000», а программа, являющаяся агентом пользователя, работает на компьютере с ОС «Windows». Так как компоненты распределенной программы работают на вычислительных платформах разного типа, интерфейс между ними использует механизм сокетов для организации взаимодействия компонентов и специально разработанную программу связи, работающую на головном узле кластера (рис. 1).

Во время выполнения MPI-программа сразу запускается на фиксированном числе вычислительных узлов кластера. Множество вычислительных узлов, занятых MPI-программой меняется от запуска к запуску. Управляющая машина суперкомпьютера, через которую происходит запуск MPI-программ, может не входить в число вычислительных узлов (так происходит в СКИФ К-1000). В MPI-программе есть выделенный процесс (соответствующий узел кластера имеет виртуальный идентифика-

тор 0), который и обеспечивает прием параметров задачи и выдачу решения. Распределение процессов по вычислительным узлам суперкомпьютера до запуска программы не известно.

Термин «сокет» обозначает точку связи, через которую процесс может передавать или получать данные. Эта точка связи задается комбинацией *сетевого адреса* и номера *порта*. Сокеты позволяют организовать виртуальное соединение между процессами. При установлении соединения всегда есть вызывающая (клиент) и отвечающая (сервер) стороны. Клиент отличается от сервера тем, что перед установлением соединения он знает сетевой адрес сервера. При конфигурации сети кластерного компьютера оказывается, что сервером для соединения не могут служить ни MPI-программа, ни программа ввода данных, так перед выполнением они не могут знать адреса друг друга. Для установления соединения требуется использование специальной программы связи. Эта программа фиксировано расположена на управляющей машине суперкомпьютера, которая служит мостом между внутренней сетью суперкомпьютера и внешней сетью.

Пользователи и разработчики для управления вычислениями взаимодействуют с интерфейсным модулем распределенной программы – агентом пользователя. Этот модуль – программа ввода данных и визуализации результатов – играет роль клиента, обслуживание которого выполняет MPI-программа – сервер для этого клиента. Программа ввода данных и визуализации результатов использует платформу стандартного PC с ОС Windows. MPI-программа – вычислитель – работает на кластерном компьютере с ОС Linux.

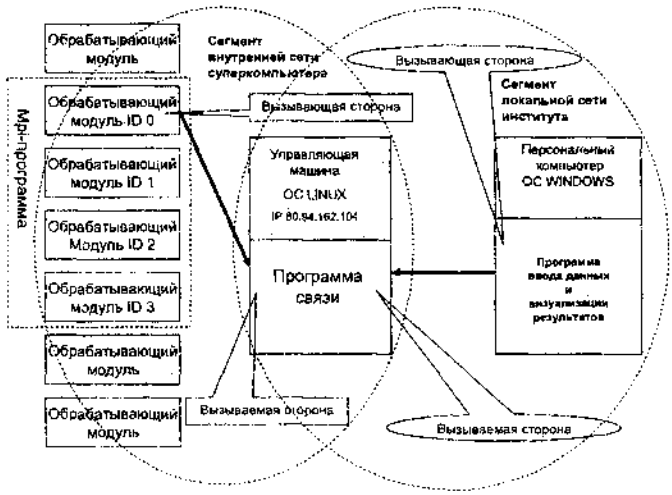

Рис.1. Положение компонентов распределенной программы в сети

Связь между агентом пользователя и вычислителем организуется через сеть (используется протокол интернет). Эта архитектура и конфигурация аппаратных средств ведет к реальной угрозе информационной безопасности. Процесс вычислителя, MPI-программа кластера – это дорогостоящий ресурс, и необходимо предотвратить несанкционированное использование, как программ, так и данных, используемых вычислителем.

3 ОРГАНИЗАЦИЯ ЗАЩИТЫ

Если предполагать, что удаленная сторона (кластер) является безопасным местом, то проблема защита распределенной программы в этом случае состоит в устранение возможности нападения «злоумышленника посреди пути передачи данных». В настоящее время имеются несколько широко используемых схем обеспечения безопасности данных при передаче по вычислительным сетям. Главные из них это SSH (Secure Shell) и SSL (Secure Socket Layer/Transport Level Security). Оба протокола работают на транспортном уровне сетей («выше» протокола TCP) и используют похожие схемы обеспечения безопасности. Для защиты данных распределенной программы выгодно использовать протокол SSL, поскольку он используется более широко, так как он принят основным протоколом безопасной передачи данных в WWW. Этот протокол обеспечивают прозрачную передачу данных, что позволяет использовать стандартный протокол Интернета ниже SSL. В SSL используются сертификаты, чтобы подтвердить подлинность сторон, а также зашифровать передаваемые данные. Существенно, что протокол SSL работает непосредственно на основе сокетов TCP/IP. В результате относительно легко обеспечить защиту данных в распределенной программе комбинаторно-логических вычислений, преобразуя интерфейс с сокетами TCP/IP в связь на основе сокетов SSL. Для целей защиты данных достаточно самоподписанных сертификатов безопасности.

В защищаемой программе использована библиотека OpenSSL, которая хорошо зарекомендовала себя как надежное средство преодоления угроз безопасности сети в ряде широко применяемых программных продуктов. OpenSSL – это открытая реализация протокола SSL/TLS и криптографических алгоритмов, разработанная Эриком Юнгом из Австралии. OpenSSL работает на всех распространенных платформах, включая все ОС типа Unix и все версии Microsoft Windows.

ЛИТЕРАТУРА

[1] Message Passing Interface Forum. MPI: A Message-Passing Interface standard, version 1.1. – [Электронный ресурс] *http://www.mpi-forum.org/docs/*. – 1995.

[2] Черемисинов, Д.И. Отладка и верификация MPI программ для решения логико-комбинаторных задач / Д.И.Черемисинов // Материалы III междун. конф. "Информационные системы и технологии (IST'2006)", 1–3 ноября 2006, ч. 1. – Мн.: Академия управления при президенте РБ. – С. 280–285.