

## СРЕДСТВА ОТЛАДКИ ЭФФЕКТИВНОСТИ MPI-ПРОГРАММ

А.Б. Бугеря, П.И. Колударов, М.И. Кулешова

Данная работа посвящена разработке диалоговой системы, позволяющей отлаживать эффективность MPI-программ, используя один из известных подходов к отладке эффективности параллельных программ - контроль эффективности посредством анализа трасс, собранных при выполнении параллельной программы.

Высокая сложность разработки параллельных программ, часто не позволяющая эффективно использовать возможности высокопроизводительных компьютеров, является общепризнанным фактом. В настоящее время ведутся интенсивные исследования в области автоматизации разработки параллельных программ, в частности, в области создания инструментов для отладки и исследования параллельных программ. Цели применения этих инструментов могут быть различными. Это в первую очередь поиск ошибок в программе, в том числе и таких специфических для параллельных программ как ошибки при передаче сообщений, ошибки синхронизации, ошибки доступа к общим ресурсам и т.д. Но и затем, после того как программа начала работать, и работать правильно, отладка программы на этом обычно не заканчивается. Сразу возникает вопрос: а насколько эффективно она работает? Равномерно ли распределена вычислительная нагрузка? Для того, чтобы ответить на эти и другие подобные вопросы применяют специальные инструменты исследования эффективности параллельных программ.

При отладке эффективности по трассе, в процессе выполнения программы сохраняется информация о том, что, как, в какое время и на каких данных каждая ветвь программы выполняла. Затем полученные трассы могут быть проанализированы. Однако использовать визуальный просмотр трасс в силу большого объема и неструктурированности данных на больших задачах представляется достаточно утомительной работой, не говоря уже о том, что процесс перекачивания всего объема полученных трасс с вычислительного комплекса, где производился запуск программы, на компьютер пользователя займет достаточно много времени. Более того, основной объем передаваемых данных, скорее всего, никогда просмотрен не будет, так как не будет интересовать пользователя. Поэтому возникла идея разработать диалоговую систему, позволяющую производить предварительную обработку трасс на вычислительном комплексе, выделяя и группируя интересующие пользователя данные, осуществлять автоматический анализ трасс с целью предоставить пользователю информацию о характеристиках эффективности программы, прогнозировать характеристики эффективности параллельной программы на моделируемой параллельной вычислительной машине без проведения реальных вычислений, а также, если пользователя интересует подробная информация по каким-то аспектам выполнения программы, показывать ему не всю трассу, а только те события, которые удовлетворяют выбранным пользователем критериям, и на компьютер пользователя передавать только те части трасс и протоколов анализаторов трасс, которые хочет в данный момент просматривать пользователь.

Разработанная диалоговая система состоит из ряда взаимодействующих между собой компонент:

- Пользовательский Интерфейс

Пользовательский Интерфейс выполнен в стиле Microsoft Visual Studio под Windows. Это многооконное и многодокументное (MDI) приложение. Имеется пять стандартных типов окон. Окно проекта (Project Bar), в котором в виде многоуровневого дерева с папками указаны файлы с исходным текстом программы. Также есть возможность работать и без создания проекта. Окно вывода информации (Output) о результатах каких-либо действий пользователя в системе (Например, подключение к Монитору и т.п.). Окна с исходным текстом программы. Окна с событиями трасс. Окна с протоколом работы анализатора трасс. Пользовательский Интерфейс выполняется на компьютере пользователя, демонстрирует в структурированном виде всю запрашиваемую пользователем информацию и предоставляет удобный интерфейс ко всем функциям системы. Всю информацию для визуализации процесса Пользовательский Интерфейс получает от Монитора, ему же отправляет все команды и запросы пользователя.

- Монитор

Программа, управляющая процессом установки соединения, обработки запросов пользователя и завершением сеанса работы. Монитор запускает Анализатор трасс, затем передает ему команды от Пользовательского Интерфейса, и обратно - результаты выполнения команд. Можно сказать, что Монитор является логическим и управляющим центром разработанной системы. Может выполняться как на удаленном вычислительном комплексе, так и на компьютере пользователя.

- Анализатор трасс

Программа, непосредственно анализирующая трассы выполнения MPI-программ на удаленном вычислительном комплексе и передает запрашиваемую пользователем информацию о трассах, результатах их обработки (протокол работы анализатора эффективности) и о файлах с исходными текстами программы. Определяет все возможные связи между различными видами отображаемой информации и передает информацию об этих связях Пользовательскому Интерфейсу для осуществления переходов между различными видами окон.

- Анализатор эффективности

Анализатор эффективности является отдельной программой и может использоваться и как часть описываемой системы (в этом случае он вызывается Анализатором трасс), и как самостоятельный инструмент отладки эффективности MPI-программ. Анализатор эффективности обрабатывает трассы, накопленные во время выполнения программы на всех процессорах, и на основании этой обработки решает две различные задачи. Первая задача - вычисление реальных временных характеристик выполнения программы. Вторая задача – вычисление прогнозируемых временных характеристик. Прогнозирование временных характеристик параллельных программ является важнейшим инструментом, позволяющим разработчику моделировать выполнение параллельной программы на связанных коммуникационной сетью и объединенных в узлы различных типах процессоров. При этом становится достижимой оценка технических решений еще на стадии проектирования параллельной вычислительной машины. Помимо этого, становится возможным получение стабильных характеристик выполнения программ.

При отладке эффективности программы пользователь не обязательно должен запускать ее с тем большим объемом вычислений, который будет характерен для использования программы при решении реальных задач. Он может ограничить количество регулярно повторяющихся внешних итераций. Но при этом время работы программы на подготовительном и завершающем этапах (до начала первой итерации и после окончания последней итерации) может быть сравнимо со временем работы основного цикла. Для того чтобы получить представление о реальной эффективности распараллеливания при малом числе итераций, необходимо оформить основной цикл в виде отдельного интервала. Если в программе не выделены интервалы, то Анализатор эффективности вычисляет характеристики для всей программы целиком.

Результатом работы анализатора производительности является файл, содержащую следующую информацию:

- информация об интервале;
- основные характеристики выполнения и их компоненты (**Main characteristics**);
- минимальные, максимальные и средние значения характеристик выполнения программы на разных процессорах (**Comparative characteristics**);
- характеристики выполнения программы на каждом процессоре (**Execution characteristics**);
- информация о точках в исходном тексте программы (**SRC Characteristics**).

Описанные компоненты отладчика представляют собой распределенную систему, которая может выполняться (в общем случае) на 3-х отдельных компьютерах, возможно, разной архитектуры с разными операционными системами. Все компоненты представляют собой отдельные исполняемые модули. Все взаимодействия между ними реализованы с помощью семейства протоколов TCP/IP, используя любой свободный TCP/IP порт. Такая схема взаимодействия достаточно гибка и позволяет успешно производить отладку эффективности программ на вычислительных комплексах, где применение отладчиков предусмотрено не было, в том числе и на вычислительных комплексах с системой очередей, закрытыми TCP/IP портами, с медленным каналом доступа к удаленной машине. Пользовательский интерфейс выполнен в виде прототипа программы для Windows, Монитор - и для Windows, и для UNIX, а Анализатор трасс - для UNIX. Диалоговый режим работы предоставляет возможность указать, какие именно события, какие именно участки выполнения программы в трассах, какие соответствующие им пункты протокола анализатора эффективности необходимо инспектировать и осуществлять переходы от событий в трассе к пунктам протокола анализатора и к строкам исходного кода, им соответствующего и обратно. А также существует возможность в тексте программы интерактивно выделять интервалы и запускать анализатор эффективности с указанными интервалами.

Работа выполнялась в рамках научно-технических программ Союзного государства «ТРИАДА» и «СКИФ-ГРИД», Программы №14 Президиума РАН, а также была поддержана грантом Президента РФ для ведущих научных школ НШ-2139.2008.9.

#### ЛИТЕРАТУРА:

1. А.Б. Бугеря, П.И. Колударов, М.И. Кулешова Диалоговый интерфейс для отладки параллельных программ. Научный сервис в сети Интернет: технологии параллельного программирования: Труды Всероссийской научной конференции (18-23 сентября 2006 г., г.Новороссийск). -М.: Изд-во МГУ, 2006, стр. 86-88.
2. В.Ф. Алексахин, К.Н. Ефимкин, В.Н. Ильяков, В.А. Крюков, М.И. Кулешова, Ю.Л. Сазанов. Средства отладки MPI-программ в DVM-системе. Научный сервис в сети Интернет: Труды Всероссийской научной конференции (19-24 сентября 2005 г., г.Новороссийск). -М.: Изд-во МГУ, 2005, стр. 113-115.
3. Отладка эффективности MPI-программ. [http://www.keldysh.ru/dvm/dvmhtml107/rus/deb\\_mpi/paUG.html](http://www.keldysh.ru/dvm/dvmhtml107/rus/deb_mpi/paUG.html)