

УДК 04981.211

ОТЛАДЧИК ПАРАЛЛЕЛЬНЫХ ПРИЛОЖЕНИЙ (PDB)

В. К. Федоров, С. Н. Киселев
(РФЯЦ-ВНИИЭФ, г. Саров)

Описываются возможности диалогового отладчика *параллельных* программ. Дается схема взаимодействия программных компонентов, которые составляют отладчик. Представлен пользовательский интерфейс.

Отладчик ориентирован на платформы Unix/Linux. Он обеспечивает отладку программ, написанных на языках программирования C, C++ и Fortran с использованием MPI- и OpenMP-стандартов.

Ключевые слова: многопроцессорная вычислительная система, отладчик, параллельное приложение.

Введение

Стадия отладки является неизменным этапом в жизненном цикле программных систем. Поэтому проблема отладки *параллельных* приложений на кластере или суперЭВМ всегда будет актуальной для разработчиков программного обеспечения. Решение этого вопроса — весьма сложная, нетривиальная задача.

Свободно распространяемые отладчики, как правило, не ориентированы на вычислительную систему с распределенной памятью. В их концепцию не входит управление множеством процессов, которые размещены по узлам кластера. Таковы отладчики, входящие в состав интегрированных систем разработки программного обеспечения или в операционную систему (ОС) Linux, например, GDB [1].

С другой стороны, существуют отладчики TotalView [2], DDT (Distributed Debugging Tool) [3], PGDBG Graphical Symbolic Debugger [4], которые по функциональности содержат все необходимые средства для отладки программ в вычислительной системе с распределенной памятью на базе ОС Linux, однако являются коммерческими продуктами. Приобретаемая лицензия на использование отладчика накладывает ограничение на число одновременно отлаживаемых процессов для всех пользователей. Стоимость таких отладчиков пропорциональна максимальному числу процессов, указанных в лицензии. Приобретенный продукт ориентирован

на определенную версию ОС; при смене ОС необходимо обновлять версию отладчика и приобретать новую лицензию.

Существует также расширение Intel® Parallel Debugger Extension [5] для Microsoft Visual Studio. Однако оно позволяет отлаживать программы только под ОС Windows и только на одном компьютере.

По указанным причинам в РФЯЦ-ВНИИЭФ в 2009 г. был разработан новый графический отладчик параллельных приложений (PDB) [6], который достаточно эффективно обеспечивает решение возложенных на него задач.

В арсенале отладчика PDB есть все стандартные возможности:

- просмотр стека вызовов потока;
- управление точками останова и наблюдения;
- обзор, изменение переменных и ячеек памяти;
- контроль потоков.

В представляемом программном инструменте также реализованы средства отладки подмножества процессов. Наряду с этим осуществляется поддержка групповых точек останова.

Реализованная концепция отладки не требует каких-либо специальных изменений в программной библиотеке коммуникационной среды и пользовательских программах. PDB поддерживает системы пакетной обработки заданий JAM [7] и SLURM [8].

Отладчик создан на языке программирования Java. В нем используются базовые компоненты Eclipse-платформы [9, 10].

Программные компоненты отладчика

Большинство современных средств отладки параллельных программ основано на представлении программы как совокупности выполняющихся процессов [2–4]. Исходя из этого построен и отладчик параллельных приложений PDB.

В состав отладчика PDB входят следующие компоненты:

- центральный программный компонент;
- программный агент;
- базовый отладчик, обеспечивающий отладку процесса.

Указанные компоненты и схема их взаимодействия изображены на рис. 1.

Данная конфигурация программных компонентов представляет собой распределенную архитектурную модель. Центральный программный компонент выполняется на инструментальном компьютере. Программный агент, базовый отладчик и процесс параллельной программы выполняются на вычислительных ресурсах, выделенных для отладки.

Центральный программный компонент. Основные средства, предназначенные для отлад-

ки параллельной программы, располагаются в центральном компоненте. Он представляет собой отдельное программное приложение. На рис. 2 показана логическая структура центрального компонента отладчика.

Его основными элементами являются модуль управления отладчиком и контроллер сессии отладки. Они обеспечивают управление элементами сессии отладки, такими как:

- выполнение операций с множеством процессов;
- контроль потоков каждого из процессов;
- работа с программными переменными;
- поддержка точек останова и наблюдения.

Взаимодействие с пользователем обеспечивается при помощи графического интерфейса. В оконном интерфейсе выводится состояние компонентов отладчика, соответствующее состоянию отлаживаемой программы. Также предоставляется набор функций для отладки параллельного приложения, доступных через различные меню.

Обмен данными между контроллером сессии отладки и программными агентами выполняет сетевой командный интерфейс.

Программный агент. Этот компонент является промежуточным звеном между центральным программным компонентом и базовым от-

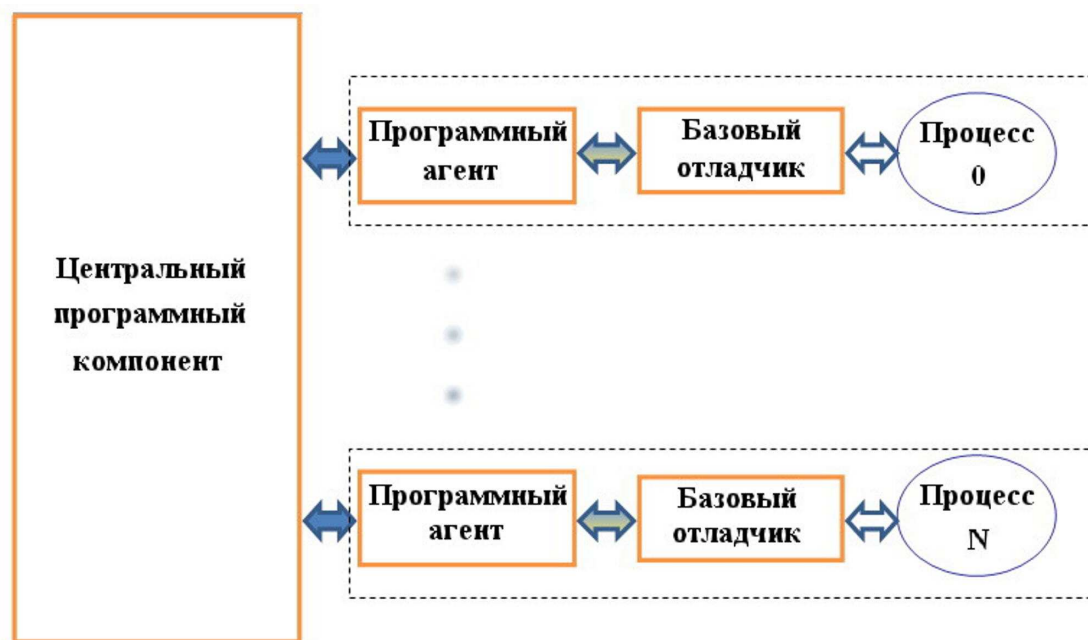


Рис. 1. Компонентная схема отладчика

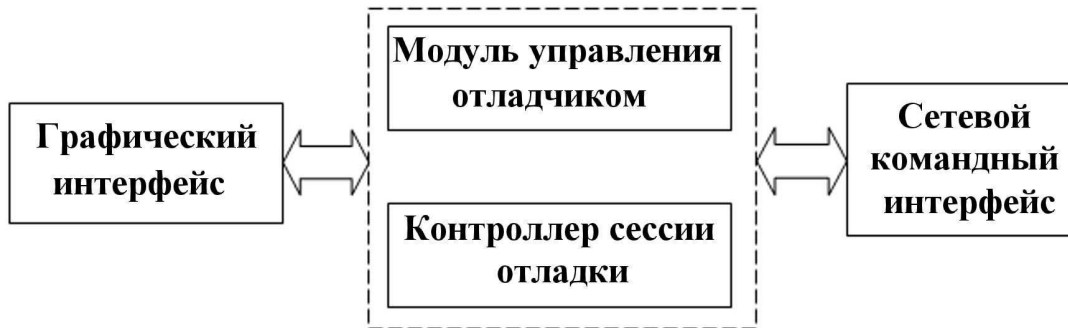


Рис. 2. Схема центрального компонента отладчика

ладчиком. Он выступает в качестве сервера и обслуживает поступающие к нему запросы. В процессе выполнения отлаживаемого процесса программный агент генерирует асинхронные сообщения. Эти сообщения могут представлять собой извещения об останове на контрольной точке, данные, выдаваемые отлаживаемым процессом на экран, диагностические сообщения базового отладчика и др. Обмен с базовым отладчиком осуществляется при помощи протокола MI (Machine Interface) [1]. Это строчный протокол, посредством которого осуществляется машинно-ориентированный командный интерфейс с базовым отладчиком. Протокол предназначен специально для поддержки систем, которые используют отладчик как свой компонент.

Базовый отладчик. Данный компонент представляет собой системную программу для отладки процесса.

В качестве базового отладчика был выбран GDB. Он является свободно распространяемым, входит в состав всех дистрибутивов Linux, обладает необходимыми средствами для отладки процесса (запуск и останов процесса, работа с точками останова и точками наблюдения, просмотр и изменение значений переменных), предоставляет протокол MI для интеграции с внешними программами.

В качестве базового может выступать не только GDB, но и любой отладчик, совместимый с ним, например, входящий в состав компилятора Intel.

Принцип работы базового отладчика следующий. Он запускает отлаживаемый процесс, предоставляет агенту информацию о нем и вы-

полняет команды отладки процесса, передаваемые ему от агента.

Процессу пользователя предоставляется виртуальный терминал для ввода/вывода информации. Терминальные данные процесса (запись/чтение стандартных потоков ввода/вывода) получает и передает программный агент.

Архитектура и реализация отладчика

Применение в PDB стороннего базового отладчика не внесло ограничений в процесс разработки. В то же время это позволило сосредоточить ресурсы разработчиков на центральном компоненте отладчика, реализовать недостающие в GDB функции графической оболочки и возможность обобщенного управления процессами параллельного приложения.

Центральный программный компонент реализован как модульное приложение с поддержкой графического интерфейса. Взаимодействие модулей выполнено при помощи событийной модели и реестра сервисов. Такой подход позволяет не запутаться в обилии взаимодействующих частей и практически безгранично расширять центральный программный компонент. В основе центрального компонента лежит OSGi (Open Services Gateway Initiative) — спецификация динамической модульной шины для создания Java-приложений, разрабатываемая консорциумом OSGi Alliance [11]. Круг применений данной спецификации довольно широк: изначально она разрабатывалась для создания встроенных систем (в частности автомобилей BMW), но сейчас на базе OSGi строят многофункциональные *desktop*-приложения [9, 12].

В описываемой версии отладчика центральный компонент архитектурно разделен на десять модулей. Каждый модуль реализует определенный набор функций или содержит концептуальное описание реализации. Главные модули центрального компонента отладчика и их взаимодействие представлены на рис. 3. В эту схему не вошли утилитарные модули, например модуль поддержки конфигурации.

Ключевыми модулями являются модель отладчика и графический интерфейс.

Модель отладчика устанавливает правила согласованной работы модулей отладчика и определяет прототипы элементов этих модулей. Этот модуль описывает протоколы взаимодействия модулей, предоставляет интерфейсы элементов программы и реализует алгоритмы работы составных частей.

Модуль графического интерфейса осуществляет вывод на экран состояния объектов отладчика и обеспечивает взаимодействие с пользователем.

Модуль реализации отладчика содержит код, реализующий элементы отладки, такие как сессия отладки, точки останова и др.

Модуль сетевого командного интерфейса осуществляет обмен с программными агентами. Доступ к среде пересылки данных производится посредством Java-библиотеки NIO (New Input Output) [13]. Библиотека предназначена для реализации высокопроизводительного ввода/вывода в Java-приложениях. Для обработки соединений используется служба *select*, предоставляемая библиотекой, что дает возможность обрабатывать огромное количество данных, по-

ступающих от множества соединений. Обмен реализован с использованием неблокирующих соединений. При этом для достижения максимальной пропускной способности библиотека использует службы уровня ОС, когда это возможно.

Взаимодействие между модулями отладчика обеспечивается при помощи внутренних команд и асинхронных сообщений. Асинхронные сообщения (события), с одной стороны, генерируются базовым отладчиком или отлаживаемым процессом. К таким событиям относятся срабатывание точки останова, вывод данных на печать отлаживаемым процессом и др. Полученное событие передается диспетчером в соответствующий модуль для выполнения требуемых операций. С другой стороны, действия пользователя также порождают события, которые запускают механизмы отладчика для их обработки.

В процессе отладки программы модули центрального компонента отладчика осуществляют манипуляции с отлаживаемыми процессами — это изменение их состояния (пуск/стоп), чтение/изменение значений переменных и др. Для выполнения таких операций используется собственный программный интерфейс. Командами данного интерфейса являются Java-объекты, которые передаются по сети посредством TCP-протокола. Для передачи команды по сети она предварительно преобразуется в поток байтов и *заключается в конверт*, заголовком которого является битовая строка. Номер разряда строки соответствует номеру процесса параллельной программы. Таким образом, установкой необходимых разрядов строки осуществляется обращение к определенному множеству отлаживаемых процессов.



Рис. 3. Архитектура центрального компонента отладчика: \Rightarrow — внутренние команды; \leftarrow — асинхронные сообщения

Описание работы параллельного отладчика

Центральный программный компонент реализован с использованием RCP-платформы [10]. Применяемая методика определяет наличие рабочего пространства — отдельного каталога, который автоматически создается в домашнем каталоге пользователя. Эта рабочая область служит для сохранения информации между сессиями отладки. Сохраняются следующие элементы сессии: состояние окон графического интерфейса, параметры, связанные непосредственно с сессией: точки останова, параметры задания и т. д. Это позволяет избежать некоторой рутины при повторном запуске отладчика после корректировки отлаживаемой программы, например, расстановки точек останова и задания групп процессов.

Рассмотрим взаимодействие модулей отладчика в целом, а также некоторые детали их работы.

Отладчик запускается пользователем. Центральный программный модуль стартует как обычное графическое приложение. Запуск параллельного приложения выполняется посредством систем пакетной обработки заданий JAM или SLURM [7, 8]. Центральный программный модуль отладчика взаимодействует с системой пакетной обработки, запуская соответствующие утилиты.

Далее пользователь в диалоговой форме вводит параметры задания. Для передачи параметров задания в систему пакетной обработки создается скрипт задания.

Процесс создания сессии отладки состоит из следующих шагов:

- 1) на вычислительном поле системы пакетной обработки заданий стартует набор агентов, соответствующих процессам программы;
- 2) агент подключается к центральному модулю;
- 3) каждый агент запускает базовый отладчик;
- 4) базовый отладчик порождает соответствующий процесс задания.

Центральный модуль ожидает подключения всех агентов. При подключении всех агентов начинается сеанс отладки параллельной программы.

Обмен данными между агентами и центральным модулем осуществляется по сетевому каналу при помощи протокола TCP. Обмен агента с базовым отладчиком производится посредством

неименованных каналов с использованием машинного командно-строчного интерфейса MI.

Связь с каналами ввода/вывода процесса реализована с использованием виртуального терминала. Терминальный обмен с процессом осуществляет агент, транслирующий данные, передаваемые от виртуального терминала центральному модулю. Поступившие данные от каждого отлаживаемого процесса выводятся в соответствующем окне графического интерфейса, а также записываются в файл. Для хранения выводимых данных каждого процесса задания создается отдельный файл, размещаемый в рабочей области отладчика.

Графическая оболочка отладчика

Управление отладкой параллельной программы осуществляется при помощи графического интерфейса. Пользователю предоставляется интерактивный контроль как над отдельным процессом, так и над их группой.

На рис. 4 (см. также цветную вкладку) представлен графический интерфейс отладчика. При создании сессии отладки, т. е. после старта задания на кластере, становятся активными графические компоненты интерфейса отладчика, которые помечены на рисунке указателями.

Указатель 1 обозначает окно управления группами процессов. Окно позволяет запускать и останавливать одновременно все процессы выбранной группы, а также манипулировать списками групп — создавать/удалять группы, добавлять/исключать процессы из группы. В этом окне отображается состояние процессов. Каждому состоянию процесса соответствует цвет пиктограммы: зеленый обозначает выполнение процесса, желтый — приостановку, красный — завершение процесса.

Окно, помеченное указателем 2, предназначено для управления отдельным процессом или потоком (запуска или приостановки его выполнения). В нем осуществляется навигация по потокам и фреймам стека процесса.

Локальные переменные выбранного фрейма стека (соответствующего функции отлаживаемой программы) выводятся в окне, отмеченном указателем 3. Окно предназначено для контроля и изменения значений переменных программы.

Точки останова выводятся в окне просмотра исходного текста, которое отмечено указателем 4. На левой вертикальной линейке окна

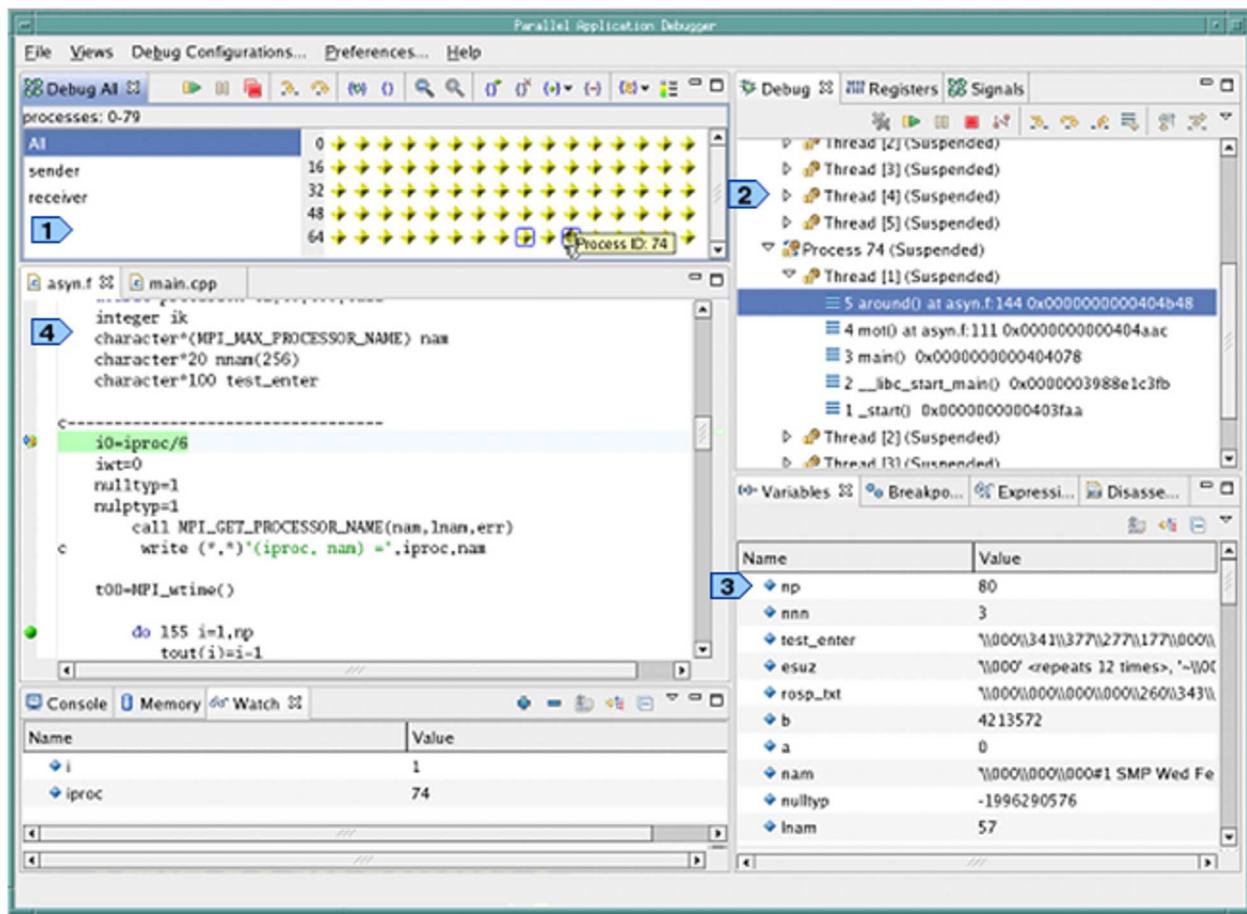


Рис. 4. Графический интерфейс отладчика

отображаются цветные пиктограммы, обозначающие типы точек останова. Устанавливаемая пользователем точка останова будет принадлежать только процессам выбранной группы. Такой подход позволяет гибко управлять точками останова в параллельном приложении.

Окна отладчика логически связаны. Переход от одного процесса к другому в окнах 1 или 2 вызывает автоматическое изменение отображаемой информации в остальных окнах.

Заключение

Параллельный отладчик PDB прошел длительный этап тестирования, модернизации и использования на разных аппаратно-программных платформах. При его разработке учтен опыт эксплуатации таких программных продуктов, как GDB и отладчика, входящего в состав компилятора Intel.

Отладчик PDB имеет следующие отличительные особенности, которые делают его достаточно удобным для пользователя:

- переносимость на разные платформы;
- графический интерфейс;
- быстрое время отклика (менее 2 секунд при отладке программы на 120 процессах);
- отладка нескольких процессов программы, запущенных на одном узле.

По функциональным возможностям PDB сопоставим с отладчиками TotalView [2] и DDT [3].

С 2009 г. отладчик PDB функционирует на неоднородном вычислительном комплексе РФЯЦ-ВНИИЭФ. Также PDB поставляется в составе базового программного обеспечения компактных суперЭВМ разработки ВНИИЭФ.

В настоящее время разрабатывается новая версия отладчика. Базовый отладчик подлежит исключению из программных компонентов PDB,

его функции будет выполнять центральный компонент. Это позволит расширить функциональность отладчика, сократит трафик между центральным компонентом и программными агентами, что в целом улучшит качество обслуживания пользователей.

Список литературы

1. *Stallman R. M., Pesch R. H.* Debugging with GDB. Free Software Foundation, 1992.
2. TotalView. <http://www.totalviewtech.com>.
3. The Distributed Debugging Tool. <http://www.allinea.com>.
4. PGDBG Graphical Symbolic Debugger. <http://www.pgroup.com>.
5. Intel® Parallel Debugger Extension. <http://software.intel.com>.
6. *Киселев А. Б., Киселев С. Н., Колпаков С. И., Федоров В. К.* Свидетельство о государственной регистрации программы для ЭВМ № 2010615637 Программа Отладчик параллельных приложений (PDB) // Электронный бюллетень. 2010. № 4. С. 332. <http://www.fips.ru>.
7. *Киселев А. Б., Киселев С. Н.* Система пакетной обработки заданий JAM // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2009. Вып. 4. С. 60–66.
8. SLURM. <http://www.llnl.gov/linux/slurm>.
9. *McAffer J., Vanderlei P., Archer S.* Equinox and OSGi: The Power Behind Eclipse. Addison-Wesley Professional, 2009.
10. *McAffer J., Lemieux J.-M.* Eclipse Rich Client Platform. Addison-Wesley Professional, 2005.
11. OSGi Alliance. <http://osgi.org>.
12. Введение в OSGi. <http://samolisov.blogspot.com/2009/03/osgi.html>.
13. *Hitchens R.* Java NIO. O'Reilly, 2002. P. 91–145.

Статья поступила в редакцию 14.01.13.
