

**TDB—ОТКРЫТАЯ РАСПРЕДЕЛЕННАЯ ПРОГРАММНАЯ СИСТЕМА  
ИНТЕРАКТИВНОЙ ОТЛАДКИ МРІ-ПРОГРАММ:  
АРХИТЕКТУРНЫЕ РЕШЕНИЯ И  
ОБЩИЕ ПРИНЦИПЫ РЕАЛИЗАЦИИ\***

А.И. Адамович, М.Р. Коваленко  
ИЦМС ИПС РАН, Переславль-Залесский, Россия

Работа посвящена изложению основных архитектурных решений, принятых при проектировании распределенной программной системы интерактивной отладки МРІ-программ TDB и принципов ее реализации. Приводится описание всех компонентов системы, их роль, базовые механизмы и принципы взаимодействия.

## **1. Введение**

Разработка отладчика МРІ-программ—часть работ по созданию инструментального программного обеспечения (ПО) параллельных вычислительных систем (ПВС), проводимых с начала 90-х годов в Институте программных систем РАН. Центральным стержнем этих работ является Т-система—программная среда, обеспечивающая поддержку разработки и выполнения параллельных программ на основе идеологии автоматического динамического распараллеливания [1,2]. Успешная реализация нескольких различных версий Т-системы, дала возможность говорить о создании «промышленного» варианта данной среды программирования.

Существенной частью нового варианта Т-системы, разработанного в рамках совместной программы СКИФ Беларуси и России, стало ее развитое окружение—набор различных программных средств, позволяющих облегчить путь от начала написания программы до получения высококачественного программного продукта. В частности, созданная в рамках данной разработки система интерактивной отладки МРІ-программ ПС TDB обеспечила возможность снизить трудозатраты на одном из важнейших и наиболее трудоемких этапов реализации параллельных приложений—этапе отладки. При этом следует отметить тот факт, что, хотя ПС TDB и создавалась специально с целью отладки приложений, разработанных с использованием Т-системы, она все же является универсальной, и ничто не препятствует ее использованию при создании самых различных МРІ-программ.

Данная статья посвящена важнейшим решениям, использованным при проектировании и реализации ПС TDB.

Заметим, что на этапе проектирования системы мы старались учесть опыт других подобных разработок [3-5]. Важной особенностью, отличающей ПС TDB от других интерактивных параллельных отладчиков [4,5], является ее открытость. Это существенное свойство системы, помимо многих прочих преимуществ, которые будут рассмотрены далее по тексту статьи, позволяет также—при использовании ПС TDB—обеспечить существенную экономию финансовых средств в сравнение с коммерческими реализациями параллельных отладчиков.

---

\* Работа выполнена в рамках программы «СКИФ» Союзного государства

Далее будут описаны использовавшиеся архитектурные принципы, включая основные особенности их реализации в текущей версии системы.

## 2. Основные архитектурные принципы

При проектировании ПС TDB на выбор архитектурных принципов, использованных для построения всего программного комплекса, оказывали влияние следующие основные факторы:

- крайне важно было поддержать собственные—созданные в ИПС РАН—инструментальные средства разработки параллельных программ [1,2];
- требовалось обеспечить совместимость ПС TDB как с открытыми версиями MPI, свободно распространяемыми в виде исходных текстов, так и с коммерческими (например, SCALI MPI), тексты и детали реализации которых недоступны;
- наличие на базовой аппаратно-программной платформе широкого выбора свободного программного обеспечения с открытыми текстами позволяло надеяться на возможность использования в ПС TDB значительного числа готовых компонент из состава свободного ПО;
- было необходимо обеспечить возможность одновременной отладки различных программ несколькими пользователями в рамках единого многопользовательского аппаратно-программного комплекса.

С учетом приведенных факторов и опыта других исследований [3] удалось сформулировать основные архитектурные принципы, легшие в основу реализации ПС TDB. В соответствие с ними, архитектура ПС TDB является:

- распределенной и многокомпонентной. ПС TDB состоит из набора компонентов, функционирующих на различных частях ПВС (как на фронтальной ЭВМ, так и на вычислительных узлах). Схема взаимодействия между компонентами соответствует расширенной модели «клиент-сервер».
- открытой и переносимой. Одним из важнейших элементов архитектуры системы является «TDB-протокол»—набор соглашений, декларирующий интерфейс взаимодействия между собой различных компонентов ПС TDB. Благодаря этому, существует возможность использовать разнообразные варианты реализации компонентов системы, при условии, что они поддерживают данный набор соглашений. Например, возможно реализовать несколько различных клиентских компонентов (текстовых, графических и др.), набор серверов отладки, поддерживающих различные базовые отладчики (gdb, idb, pgdbg и другие) или обеспечить поддержку нескольких систем распараллеливания программ. Как следствие, можно добиться переносимости системы даже в том случае, когда невозможно или нежелательно использование основных, стандартных реализаций одного или нескольких из компонентов. В этом случае их можно заменить специализированными реализациями, способными функционировать в специфических условиях (нестандартные аппаратные платформы, программные среды и т.д.).
- гибкой, с акцентом на максимальное использование имеющихся в наличии свободно-доступных программных компонентов. В то же время сохраняется возможность использования коммерческих продуктов, например, в части серверов отладки.
- многопользовательской, обеспечивающей бесконфликтность и безопасность процесса отладки при одновременном осуществлении отладки несколькими пользователями.

### 3. Компоненты ПС TDB и их назначение

Компоненты ПС TDB (см. рис. 1) естественным образом делятся на две группы: компоненты, функционирующие на фронтальной ЭВМ и компоненты, функционирующие на вычислительных узлах. К одной группе относятся первичный демон, центральный сервер и клиентский компонент, к другой—вторичный демон, сервер отладки и библиотечный компонент.

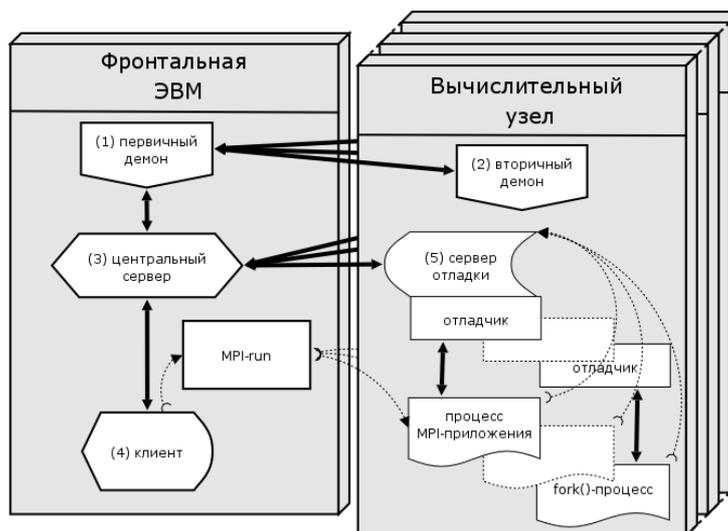


Рис. 1. Архитектура ПС TDB.

**Первичный и вторичный демоны** являются мониторными, постоянно функционирующими процессами. Основная задача, решаемая вторичным демоном, и одна из задач первичного демона—отслеживать состояние вычислительных узлов ПВС. Мониторинг состояния ПВС позволяет обеспечить клиентский компонент информацией о наборе вычислительных узлов, на которых возможен запуск рассматриваемого MPI-приложения в режиме отладки.

Все остальные компоненты ПС TDB не являются постоянно функционирующими процессами и запускаются клиентским компонентом системы или непосредственно пользователем в процессе проведения сеанса отладки конкретного MPI-приложения.

**Центральный сервер** ПС TDB обеспечивает взаимодействие клиентского компонента с остальными компонентами ПС TDB—играя, в частности, роль программного мультиплексора пакетов TDB-протокола. Использование центрального сервера в качестве отдельного компонента системы позволило в значительной степени повысить степени открытости всей системы, облегчить создание различных реализаций клиента.

В контексте расширенной модели «клиент-сервер», реализуемой в ПС TDB, центральный сервер выступает в роли сервера для клиентского компонента, и в роли клиента для других компонентов (первичного демона и серверов отладки).

**Библиотечный компонент** предназначен для инициализации подключения процессов отлаживаемого MPI-приложения к ПС TDB на начальной стадии сессии отладки. С этой целью библиотечный компонент подменяет ряд функций, стандартных для параллельных приложений такого типа, на специальные, которые:

- предварительно запускают на исполнение сервер отладки и механизм присоединения к нему текущего процесса;
- затем вызывают оригинальную функцию.

ПС TDB поддерживает различные реализации MPI и имеет в своем составе набор соответствующих библиотек. Подходящий экземпляр библиотечного компонента включается в состав исполняемого файла MPI-приложения на этапе редактирования связей.

**Сервер отладки**—компонент ПС TDB, непосредственно управляющий отладкой процессов приложения на вычислительных узлах, а также отвечающий за присоединение процессов отлаживаемого приложения к сессии отладки.

Запуск сервера отладки производится из пользовательского MPI-приложения в процессе выполнении им специальных функций библиотечного компонента.

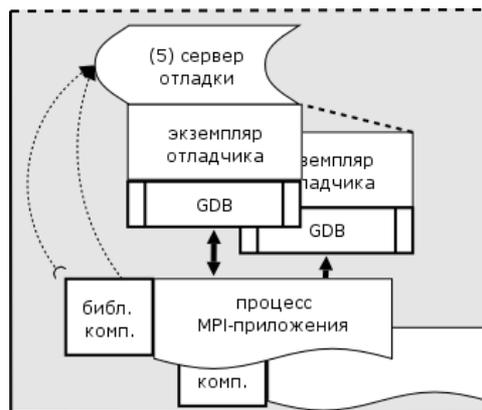


Рис. 2. Подробная иллюстрация архитектуры компонентов сервера отладки и механизма присоединения процесса к серверу отладки.

Для управления отлаживаемыми процессами сервер отладки (см. рис. 2) имеет набор специальных программных объектов—экземпляров отладчиков, управляющих работой базовых системных отладчиков (например, GNU gdb). Таким образом, процессу отлаживаемого приложения в сервере отладки соответствует пара «экземпляр отладчика сервера отладки—системный отладчик», которая занимается непосредственным управлением отладкой данного процесса, интерпретируя и выполняя команды, поступающие от клиента через центральный сервер.

Набор процессов MPI-приложения, присоединенных к одному серверу отладки, образует MPI-узел. Наряду с персональными командами над конкретными процессами сервер отладки способен производить и групповые операции над предопределенными группами процессов MPI-узла, поддерживая, таким образом, парадигму групп и групповых команд ПС TDB.

**Клиентский компонент** (клиент)—программная составляющая ПС TDB, обеспечивающая взаимодействие пользователя с ПС TDB в процессе интерактивной отладки. Таким образом, данный компонент оказывается критически важным для решения основной задачи ПС TDB—минимизации усилий и затрат на этапе отладки разрабатываемого параллельного MPI-приложения.

На основных этапах сеанса отладки клиент выполняет следующие функции: подготавливает среду выполнения отлаживаемого MPI-приложения, запускает задачу на исполнение в режиме отладки; управляет процессом интерактивной отладки MPI-приложения в целом и, наконец, производит завершение процесса отладки.

Специфика ПС TDB заключается в том, что, в отличие от обычных, непараллельных отладчиков, ПС TDB работает сразу с многими процессами отлаживаемого MPI-приложения на нескольких вычислительных узлах. В случае отладки таких сложных многопроцессовых MPI-приложений каждая из подзадач работы отладчика усложняется многократно, и пользователь сталкивается с необходимостью оперировать намного большим объемом поступающей информации различного рода.

Очевидно, что клиентский компонент, способный обеспечить эффективность процесса интерактивной отладки параллельного приложения, должен

- представлять информацию в легко считываемом, удобном виде—для того, чтобы пользователь мог охватить картину состояния отлаживаемого приложения целиком и в то же время не был перегружен излишне исчерпывающей информацией;
- иметь интуитивно понятный интерфейс для упрощения задачи управления процессом отладки.

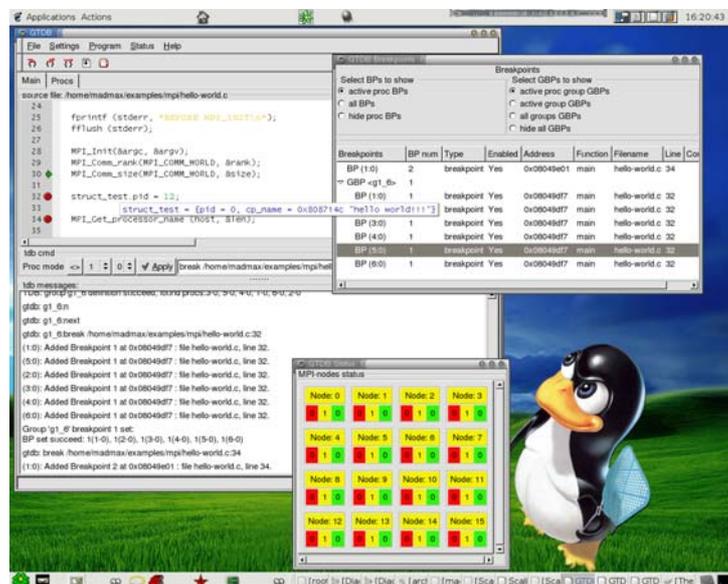


Рис. 3. Интерфейс GTDB. Иллюстрация сеанса отладки.

Текущая реализация ПС TDB включает в себя полноценный графический клиентский компонент—GTDB (см. рис. 3), удовлетворяющий всем этим требованиям.

## Л и т е р а т у р а

1. Abramov S.M. et al. Autotransformation of evaluation network as a basis for automatic dynamic parallelizing // Proceedings of NATUG Meeting “Transputer: Research and Application”, 1993.

2. С. М. Абрамов, А. И. Адамович, М. Р. Коваленко. Т-система—среда программирования с поддержкой автоматического динамического распараллеливания программ. Пример реализации алгоритма построения изображений методом трассировки лучей // Программирование, 1999, 25 (2), с. 100-107.

3. Doreen Cheng and Robert Hood. A Portable Debugger for Parallel and Distributed Programs // Proceedings of the 1994 conference on Supercomputing.

4. Etnus TotalView delivers the features, flexibility, and power you need to debug today's code // URL: <http://www.etnus.com/Support/papers/TotalViewBrochure.pdf>
5. The Distributed Debugging Tool. An Overview of Features // URL: <http://www.streamline-computing.com/ddt-overview.pdf>