



Math-Net.Ru

Общероссийский математический портал

М. А. Антонов, В. Л. Шабанов, Автоматизированная информационная система проверки задач по программированию «Dudge», *Матем. моделирование и краев. задачи*, 2007, часть 4, 11–13

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.14.82

8 февраля 2025 г., 15:40:53



М. А. Антонов, В. Л. Шабанов

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПРОВЕРКИ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ «DUDGE»

Соревнования и олимпиады по программированию в настоящее время довольно популярны. Многие высшие учебные заведения России и мира проводят свои мероприятия и являются платформой для крупных международных соревнований. Проведение, анализ решений и выставление баллов участникам являются главной задачей организаторов.

С момента появления таких соревнований были реализованы различные методы решения данных проблем.

В начале программы участники проверялись вручную, посредством анализа алгоритма решения, после чего жюри принимало решение о начислении баллов. Этот способ проверки занимал много времени и был малоэффективен, поскольку не проводилась проверка алгоритма на конкретных входных данных. При такой проверке невозможно выявить оптимальность решения, т. к. алгоритм, реализованный участником, мог не совпадать с алгоритмом, разработанным жюри при составлении задачи.

Следующим способом проверки стала проверка решений с помощью ЭВМ: жюри вручную запускало алгоритм на наборе тестов, составленных с учётом возможных ошибок участников, а баллы выставлялись в зависимости от количества тестов, на которых решение участника дало правильный ответ. Такой способ проверки выявлял наиболее типичные ошибки, а также позволял абстрагироваться от алгоритма участника (т. к. более не было необходимости просматривать исходный код решения вручную), однако время работы и размер использованной оперативной памяти можно было оценить лишь приблизительно.

Далее для проверки стали применяться программные комплексы, призванные облегчить задачу многократного запуска решений на одном и том же наборе тестов. Такие комплексы могут самостоятельно проверять ответы, выводимые решениями участников, и ограничивать время работы этих решений. Одновременно с развитием локальных сетей стала возможным доставка решения до физического места проверки и отправка и обработка результатов без вмешательства жюри. Участнику необходимо лишь отправить исходный текст своего решения по сети на сервер и, спустя несколько секунд, получить ответ проверяющей системы.

Авторами настоящей работы разработана система Dudge (Distributed Judge System), предназначенная для автоматизированной

компиляции и проверки задач. Она может использоваться при проведении соревнований по программированию школьного и студенческого типов, а также для лабораторных работ.

В процессе разработки Dodge был проведён анализ недостатков аналогичных систем и выделены следующие основные требования к разработанной системе.

Многоплатформенность. Тестирующая система, созданная под конкретную ОС, будет ограничивать организаторов соревнования в выборе и, тем самым, распространит на себя недостатки той ОС, под которой работает. Клиентская и серверная части нашей системы написаны на Java и могут работать без перекомпиляции на любой платформе, где есть Java-машина. В тестирующей же части (т. н. «slave») платформозависимый код выделен в отдельную библиотеку, версии которой существуют как для Microsoft Windows 9x/2000, так и для UNIX-подобных ОС.

Распределённость. Разные части системы могут быть запущены на разных компьютерах, объединённых сетью, поддерживающей стек протоколов TCP/IP. В качестве хранилища информации используется СУБД, обращение к которой производится посредством SQL.

Отказоустойчивость. Зависимость между различными частями такова, что при сбое одной из них другие продолжают работать (или корректно завершают работу) без потери данных. Например, если выйдет из строя тестирующая часть, то это никак не отразится на возможности просмотра статистики соревнования или регистрации в нём.

Гибкость настройки. Система может использоваться как для проведения соревнований с различными правилами (АСМ и школьных), так и для лабораторных работ по программированию.

Мобильность. В отличие от распространённых web-ориентированных тестирующих систем, Dodge проста в настройке, а её развёртывание занимает небольшое время: никакая из частей системы (кроме DodgeSlave) не требует доступа на запись на жесткий диск компьютера для работы.

Система Dodge состоит из нескольких частей, которые могут работать на разных компьютерах.

1. DodgeClient — клиентская часть, реализующая пользовательский интерфейс участника соревнования. С помощью неё можно отсылать решения на соревнования, а также в реальном времени просматривать таблицу результатов.

2. DodgeServer — серверная часть системы, выполняющая роль сервера приложений и брокера запросов. Она занимается тем, что ставит задания по проверке решений, пришедших от DodgeClient, в очередь и распределяет их по запущенным экземплярам DodgeSlave

ve. Полученные результаты и всю сопутствующую информацию сервер сохраняет в базе данных, к которой обращается, используя SQL.

3. DodgeSlave получает задания от серверной части, выполняет их и посылает обратно результат проверки.

Проверка решения задачи состоит из следующих этапов:

а) компиляция исходного кода задачи с помощью выбранного участником компилятора;

б) установка ограничений на запускаемое решение; таким образом программа после запуска не сможет каким-либо образом повлиять на работу системы; ограничениями также являются максимальное время выполнения решения и наибольший допустимый объем памяти (они указываются в условии задачи);

в) многократный запуск откомпилированного решения на наборе входных тестов с наложенными ограничениями;

г) сравнение на эквивалентность результата выполнения решения на каждом из тестов с эталонным выводом.

Из-за ресурсоёмкости второго этапа и необходимости установки ограничений на используемую память и время выполнения для задачи, этот этап в DodgeSlave реализован в виде отдельной динамической библиотеки, написанной на C++ и откомпилированной в машинный код.

4. DodgeAdmin является интерфейсом администрирования системы. Он позволяет создавать новые соревнования, добавлять в них задачи и участников, а также просматривать статистику работы серверной и проверяющей частей системы.

Каждая команда-участник имеет учётную запись в системе, содержащую имя команды, пароль, e-mail, а также имена членов команды. Во время соревнования участник, зайдя под учётной записью своей команды, выбирает в DodgeClient задачу и язык программирования, отсылает исходный код своего решения данной задачи. После проверки системой его решения, он получает уведомление, содержащее результат проверки и дополнительную информацию (например, ошибки компиляции). Используя функцию просмотра статистики соревнования, пользователь может просмотреть распределение по местам участников соревнования и то, какие задачи и с каких попыток решил каждый из них.

Области применения системы. Система Dodge может использоваться как для проведения различных видов «текущих» соревнований по программированию, так и в круглосуточном режиме работы при установке на постоянно работающий сервер. Также она может применяться для автоматизированной проверки работы студентов на лабораторных работах по программированию.

Самарский государственный технический университет, г. Самара