электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель Общероссийская общественная организация "Академия инженерных наук им. А.М. Прохорова"
ISSN 2307-0595

Кроссплатформенная система тестового контроля для мульти-компьютерного комплекса с гетерогенной архитектурой

12, декабрь 2017

Терехов А. А.[1], Сальман Д. А.[1,2,*], Воробьев А. Н.[1,**]

УДК: 004.054

[1]АО «НИИ «Аргон», Москва, Россия
[2]МГТУ им. Н.Э. Баумана, Москва, Россия
javaos@mail.ru
*molecada@yandex.ru
**worlok4yax@yandex.ru

Введение

Проектирование бортовых вычислительных машин (БЦВМ) различного назначения традиционно связано с созданием системного программного обеспечения, в состав которого включается, в том числе, тестовое программное обеспечение (ТПО) для обнаружения неисправностей аппаратных средств на различных этапах жизненного цикла изделия - от наладки до штатного использования в реальном времени в составе объекта. В указанных режимах для контроля исправности БЦВМ используются различные последовательности тестовых модулей, определяющих исправность отдельных устройств БЦВМ и, как следствие, отражающих их конкретную архитектуру. Запуск необходимой последовательности тестовых модулей на исполнение в соответствии с определенным режимом работы, формирование данных о результатах проверки, связь с оператором в определенных режимах работы осуществляет программа диспетчера тестов, работающая под управлением установленной в БЦВМ операционной системы.

Выбор операционной системы для конкретной БЦВМ определяет пользователь исходя из особенностей построения циклограммы системы управления объектом и комплекса пользовательских программ. Так за последние годы в БЦВМ с архитектурами intel, mips, elbrus, baikal устанавливались операционные системы Linux, Эльбрус, QNX4, QNX Neutrino, Windows, ОС МСВС 3.0, ОС МСВС 5.0. и т.д. [1]. Подобный подход к выбору ОС привел к многообразию программ диспетчера тестов с идентичными функциями, но реализованных в среде разных операционных систем [4].

Развитие информационных технологий, сетевых методов обработки информации, расширение круга задач, решаемых в мобильных объектах спецназначения с одной стороны, и новые возможности, предоставляемые современной компонентной базой – с другой,

обеспечили в ряде применений возможность перехода к созданию мультикомпьютерных информационных управляющих бортовых комплексов (БВК) [3].

В связи с необходимостью разработки БВК на базе отдельных специализированных БЦВМ с различной архитектурой, поиск подхода к построению унифицированного диспетчера тестов, не зависящего от типа установленной в локальных БЦВМ ОС, становится чрезвычайно актуальным.

1. Организация тестового программного обеспечения

Тестирование – это проверка исправности всех или отдельных компонентов вычислительной системы с целью выявления соответствия их характеристик заявленным производителем, а также определения реальной производительности системы. Применять тестирование можно как к корректно функционирующему оборудованию с целью подтверждения исправности, так и к оборудованию, работающему со сбоями с целью локализации неисправностей.

Варианты запуска тестовых программ представлены на рис.1.

Тестирование может быть комплексным (проверка всей системы в целом) или выполняться над отдельными комплектующими частями. В последнем случае для тестирования каждого компонента может быть необходим свой диспетчер тестов, который обычно поставляется вместе с соответствующими аппаратными средствами.



Рис. 1. Классификация тестового ПО

Программы для тестирования могут запускаться без предварительной установки, благодаря перемещению папки с соответствующей программой на компьютер оператора, либо с предварительной установкой.

Важным условием является наличие у диспетчера тестов понятного интерфейса. В частности, желательно, чтобы при выполнении тестов предоставлялась возможность

управления тестами, а также предоставление оператору исчерпывающей статистики о результатах их выполнения. Таким образом, можно отследить, происходит ли тестирование в нормальном режиме. Однако существуют и диспетчеры тестов с упрощенным интерфейсом, где оператор имеет возможность отследить только запуск и результат выполнения теста.

Программы тестирования делятся и по количеству устройств, которые можно одновременно тестировать. Если диспетчер тестов предоставляет оператору возможность тестировать одновременно более одного устройства, то для каждого из них организуется свой канал обмена данными, а оператору предоставляется несколько тестовых окон (по одному для каждого устройства).

Диспетчеры тестов отличаются также способом запуска тестов: тесты могут исполняться отдельно или запускаться группами. При этом в группе тесты могут запускаться как последовательно, так и параллельно. Второй вариант запуска значительно уменьшает время тестирования для многопоточных процессоров или многопроцессорных систем, поскольку для каждого теста создается свой поток для выполнения. Кроме того, групповое тестирование предоставляет возможность оператору более гибкой настройки сценария тестирования.

2. Специфика построения ТПО для БВК

К важнейшей характеристике БВК подобного типа следует отнести скрытность от пользовательских приложений различия в архитектуре отдельных компьютеров системы и способов связи между ними, а также различие в установленных в них операционных системах.

В БВК, в общем случае гетерогенных, при отсутствии общей памяти между вычислительными узлами передача командной информации от диспетчера тестов последовательностям тестовых модулей возможна исключительно в виде сетевых сообщений. При этом один из узлов, определенный на данном этапе работы, как сервер, передает сетевые сообщения о тестовой проверке прочим узлам, функционирующим в роли клиентов. В этом случае диспетчер тестов содержит серверную и клиентскую части. Причем обе части диспетчера тестов хранятся в памяти каждого из вычислительных узлов БВК с тем, чтобы в случае необходимости, функции сервера мог исполнять любой из исправных клиентских вычислительных узлов.

Таким образом, для поддержания представления совокупности различных компьютеров в виде единой системы, в БВК предусматривается дополнительный промежуточный уровень ПО (middleware), находящийся между верхним уровнем (frontend), на котором работают пользователи и распределенные приложения, и нижним уровнем (backend), на котором размещены локальные операционные системы.

На рис.2 представлены уровни ПО.

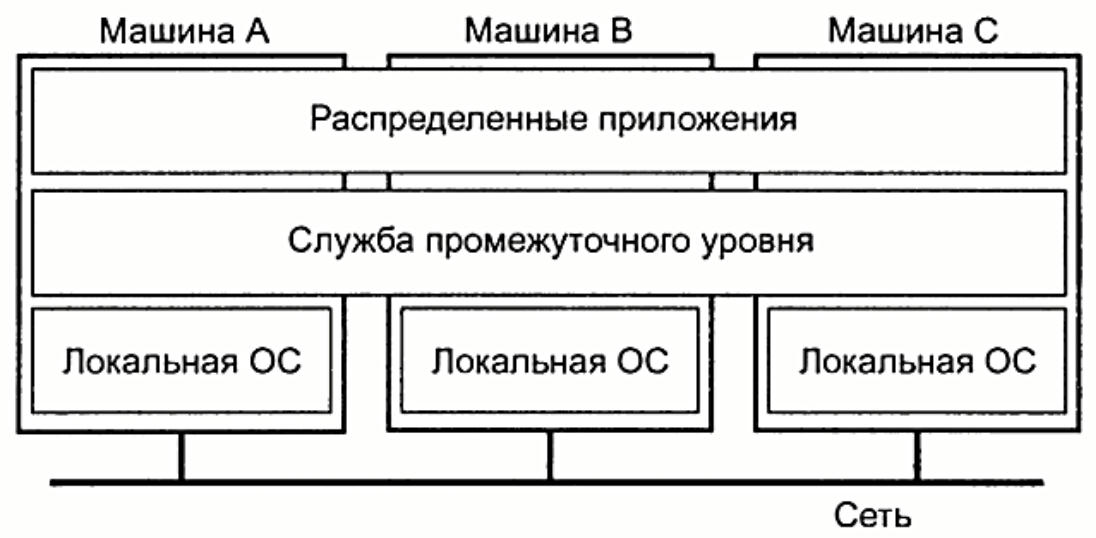


Рис. 2. Уровни ПО

Тестовые модули, отображающие архитектуру конкретных вычислительных узлов, реализуется на нижнем уровне и функционируют под управлением локальной ОС, установленной на этом узле.

В отличие от тестовых модулей, диспетчер тестов реализуется на промежуточном уровне, обеспечивая передачу сообщений (в том числе и от оператора) о режимах запуска тестов и реализацию управления этими режимами.

Единственным условием для успешного функционирования такой организации диспетчера тестов является поддержка всеми локальными ОС сетевого стандарта TCP.

Описанная организация построения диспетчера тестов позволяет решить проблему унификации, так как диспетчер тестов, входящий в состав службы промежуточного уровня, не привязан к конкретным операционным системам, установленным в отдельных вычислительных узлах гетерогенного БВК.

Следует отметить, что такая организация диспетчера тестов может быть применена с равным успехом в однопроцессорной БЦВМ, при условии размещения в памяти серверной и клиентской частей диспетчера.

3. Специфика организации диспетчера тестов БВК

Наличие в БВК отдельных вычислительных узлов, не связанных общей памятью, а поддерживающих взаимодействие лишь посредством информационных сетевых посылок, потребовало в процессе разработки ТПО поиска оригинального подхода в сравнении с традиционными системами тестового контроля в однопроцессорных БЦВМ.

Диспетчер тестов должен обеспечивать управление процессом тестирования аппаратных средств БВК под управлением ОС в различных режимах, ориентированных на различные периоды жизненного цикла изделия, сбор и анализ информации о результатах тестирования; осуществление взаимодействия сервера со множеством клиентов.

Кроме того, диспетчер тестов должен предоставлять пользователю БВК доступный графический интерфейс для выполнения оператором необходимого режима тестирования

аппаратных средств и последующего анализа его результатов на всех этапах жизненного цикла изделия.

При разработке ПО БВК ставится условие максимальной переносимости и функционирования ПО на разнородных программно-аппаратных платформах, то есть ПО должно быть гетерогенным. Это достигается следующими методами при проектировании и разработке:

- модульность (уровни backend, middleware, frontend);

- спецификация протоколов обмена между программными модулями, независящая от интерфейсов обмена данными;

- расширяемость спецификации протоколов под требования заказчика.

Чтобы обеспечить потенциальную кроссплатформенность программного обеспечения диспетчера тестов, для разработки кода программ выбран язык высокого уровня C++, а, в случае программ с графическим интерфейсом, инструментарий Qt - кроссплатформенная библиотека разработки графических интерфейсов на C++.

Для обеспечения обмена данными между клиентской и серверной частями диспетчера тестов, а также взаимодействия сервера с модулем графического интерфейса используется технология Ethernet, а также набор сетевых протоколов TCP/UDP.

Для задания тестирования составляются специальные файлы со сценариями для каждого клиента, которые хранятся на сервере. При подключении клиента происходит отправка соответствующего файла.

Помимо этого, необходим определенный протокол общения сервера с клиентами и сервера с графикой для передачи команд и актуальных состояний ВМ и всего БВК в целом, файлов статистики и файлов сценариев.

4. Общая характеристика ТПО БВК

Функциональные тесты вместе с Диспетчером тестов (ДТ) представляют собой Тестовое программное обеспечение (ТПО), которое предназначено для проверки корректного функционирования аппаратуры БВК под управлением ОС, функционирующей в каждой БЦВМ БВК, в том числе режиме реального времени.

Единственными требованиями к локальным ОС, функционирующим в отдельных вычислительных модулях (ВМ) БВК, является поддержка конкретной ОС транспортного протокола TCP/UDP модели OSI и поддержка библиотеки потоков pthread.h.

ТПО выполняет следующие функции:

- проверку функционирования каждого ВМ, входящего в состав комплекса, под управлением ОС;

- проверка взаимосвязи всех ВМ, входящих в состав комплекса, по сетевому интерфейсу;

- проверка комплексного функционирования всех ВМ, входящих в состав комплекса;

- задание оператором любой последовательности проверки каждого ВМ, в том числе однократное или многократное выполнение любых или заранее подготовленных последовательностей тестов, однократное или многократное выполнение любого теста;

- сбор и анализ информации о результатах тестирования;

- регистрация информации о процессе тестирования каждого ВМ;

- регистрация итоговой информации о процессе тестирования всех ВМ, входящих в комплекс;

- регистрация текущей информации о ходе тестирования ВМ, входящих в состав комплекса, и динамики процесса тестирования на дисплее выбранного ВМ;

- определение готовности посредством выполнения набора тестов начального запуска и проверки целостности информации на жестком диске (флэш-памяти) путем контрольного суммирования заранее заданного набора файлов;

- задание новой последовательности проверок оператором;

- выполнение процесса тестирования в режимах со связью или без связи с оператором, с отображением или без отображения результатов тестирования на экране монитора;

- осуществление предварительной настройки для выполнения последующего тестирования в любом заданном режиме;

- запуск процесса тестирования как по команде оператора, так и из прикладной программы с заданием параметров тестирования в командной строке.

Регистрация информации о процессе тестирования производится в файлах на жестком диске или флэш-памяти, которые в совокупности представляют собой базу данных, используемую для анализа состояния аппаратуры БВК, а также для определения надежности функционирования.

Реализуются активный, автоматический и пассивный режимы работы клиентской части ПО.

В активном режиме тестирование происходит только один раз с использованием сценариев соответствующего конфигурационного файла, хранящегося на сервере. Далее на стороне клиента ожидается ввод команды. При этом есть возможность повторного тестирования.

Автоматический режим реализуется аналогично активному, но при этом запрос инструкций происходит до тех пор, пока клиент не будет остановлен. При запуске клиентской части ПО выбирается время в секундах, через которое тестирование будет выполняться повторно.

Работа клиента в пассивном режиме реализуется в результате передачи оператором команды для начала тестирования клиента. Графический интерфейс позволяет редактировать сценарий, который будет отправлен клиенту.

Каждый из трех режимов предусматривает для клиента сценарий в определенном формате.

5. Организация диспетчера тестов БВК

Диспетчер тестов - специальная программа, помогающая оператору корректно провести тестирование, и последующий анализ его результатов. В его задачи входит получение

команд на запуск тестов от оператора, их анализ, исполнение и выдача результатов тестирования.

ТПО состоит из следующих структурных единиц – тестов, групп тестов, режимов. Управление указанными структурными единицами осуществляет Диспетчер тестов (ДТ).

Структурная схема ДТ представлена на рис.3.

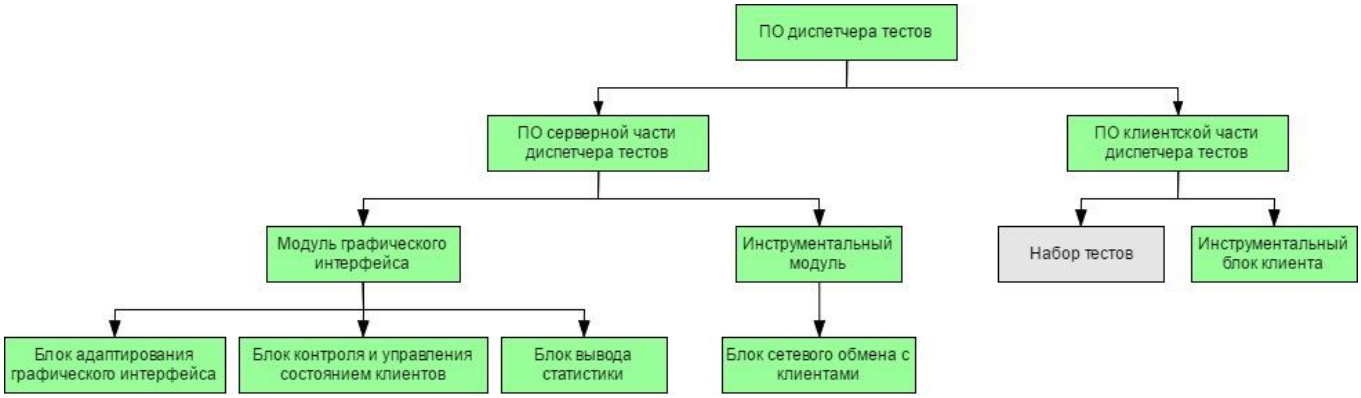


Рис. 3. Структурная схема ПО диспетчера тестов

Все программное обеспечение ДТ делится на серверную и клиентскую части.

Клиентская часть ПО содержит набор тестов и инструментальный модуль клиента. Серверная часть ПО содержит инструментальный модуль сервера, модуль обмена данными, блок графического интерфейса и модуль просмотра статистики.

Клиентская часть ПО диспетчера тестов принимает от сервера сценарии для дальнейшего тестирования. Проанализировав полученные команды, клиент вызывает соответствующие тесты.

После окончания тестирования вся информация будет передана на сервер и сохранена в файле, а, в последствии, всю статистику можно просмотреть с помощью модуля просмотра статистики. Блок графического интерфейса на сервере выведет результат, показав оператору, что тестирование данного клиента прошло успешно.

ПО клиентской части ДТ состоит из основного процесса, управляющего своим основным окном, и процесса, управляющего запуском тестов. ДТ запускает каждый тест, как отдельный процесс, а, в случае группы, как параллельно выполняемые процессы. Каждый запущенный тест формирует статистику своего исполнения, а по окончании теста статистика объединяется в едином файле. Этот подход делает независимыми друг от друга тесты одной группы и позволяет выполнять их параллельно. После запуска тестов, процесс, запускающий тесты, переходит в режим ожидания завершения теста/тестов. Основной процесс поддерживает связь с процессами, созданными для клиентов посредством общей области, в которой указанные процессы отображают текущие результаты комплексного тестирования. Основной процесс периодически просматривает эту область и отправляет информацию о состоянии клиента на сервер.

Серверная часть ПО диспетчера тестов является центральной частью, которая обеспечивает подключение клиентов с помощью сокетов, отправку инструкций, сбор статистики их тестирования и осуществляет общение с блоком графического интерфейса, передавая

актуальную информацию о состоянии клиентов и их тестировании в модуль обмена данными между сервером и блоком графического интерфейса.

ПО серверной части ДТ запускает процесс, который отслеживает функционирование клиентов, и для каждого клиента создается связь по протоколу TCP для сбора статистики выполнения тестирования клиентом. Основной процесс также поддерживает связь с графическим приложением в случае его наличия. При изменении состояния клиентов и информации о работе БВК сервер передает актуальную информацию в модуль графического интерфейса. При этом осуществляется получение и анализ команд оператора из графического ПО, а при необходимости команды отправляются в адрес соответствующего клиента.

Блок адаптирования графического интерфейса является динамической библиотекой, которая компилируется вместе с блоком контроля и управления состоянием клиентов. В ней содержатся все данные по каждому клиенту, необходимые для приложений, предоставляющих информацию оператору:

- ip-адрес;
- имя и описание;
- текущее и предыдущее состояния;
- количество клиентов;
- путь к файлам со статистикой;

Данный модуль выступает посредником между сервером и графической частью. Он получает по сокету данные от сервера и всегда хранит актуальную информацию, востребованную для графики. Благодаря подобной структуре появляется возможность создания специализированных программ, необходимых для конкретных мобильных объектов или особой задачи.

Тест представляет собой минимальную единицу управления и называется тестовым модулем. Тест представлен исполняемым файлом теста и информационным файлом, в котором описываются индивидуальные параметры тестирования. Каждый параметр может представлять собой либо строку символов, либо набор альтернативных параметров, из которых должен быть выбран один параметр, либо переключатель.

Тест может состоять из нескольких внутренних тестов. Управление внутренними тестами выполняется самим тестом.

ДТ может запускать тест однократно или многократно.

Связь ДТ с тестами производится посредством передачи параметров тестирования в параметрах командной строки при запуске теста. Тест, в свою очередь, возвращает обратно в ДТ код завершения теста, представляющий собой информацию о количестве сбоев, обнаруженных тестом.

Параметры формируются ДТ и состоят из обязательных системных параметров, индивидуальных для каждого теста. Индивидуальные параметры теста описываются в информационном файле теста, там же задается значение каждого параметра по умолчанию.

Тест считывает параметры и управляет своим выполнением в зависимости от заданных параметров.

Результаты выполнения теста записываются в файлы статистики. Существует один файл статистики, в котором регистрируется протокол выполнения тестов для данного клиента, и общий файл статистики для текущей сессии. В процессе выполнения тест может записывать в файл статистики информацию об обнаруженных сбоях или информацию о процессе тестирования. После завершения теста ДТ записывает информацию в файл статистики клиента и в общий файл статистики, который хранится только на сервере. В файлы статистики ДТ записывает информацию о времени запуска каждого теста, о времени завершения теста, если задана полная регистрация, или итоговую информацию о количестве запусков конкретного теста, а также о количестве обнаруженных сбоев. Если тест не обнаруживает ошибок, он может не формировать соответствующую запись в своем файле статистики. В этом случае в общем файле статистики будет присутствовать только информация, формируемая ТПО.

Заключение

За последние десятилетия повышение требований к объемам обрабатываемой на борту информации, с одной стороны, и новые возможности, появившиеся благодаря прогрессу в области элементной базы и сетевых технологий - с другой, позволили создать единую бортовую сеть, которую ранее невозможно было реализовать. Предложенная в статье технология объединяет установленные в подсистемах абсолютно разные модули управления объектом БЦВМ: от различных производителей, со специфической архитектурой, с различными операционными системами (в том числе реального времени).

В статье описан подход к построению унифицированной системы тестового контроля мультикомпьютерного гетерогенного БВК. Этот БВК представляет собой бортовую вычислительную сеть, управление тестированием отдельных вычислительных узлов, в котором на отдельных этапах жизненного цикла и корректности их взаимодействия в реальном времени осуществляется под управлением унифицированного диспетчера тестов, созданного в среде ОСРВ «Эльбрус».

Реализованная структура диспетчера тестов БВК предоставляет возможность гибкой компоновки наборов тестовых модулей, созданных в соответствии со структурой аппаратного обеспечения, для различных режимов проверки.

Один и тот же тестовый модуль может запускаться с разными параметрами в составе одной группы тестов последовательно или параллельно, для чего создаются копии этого теста.

В соответствии с предъявляемыми требованиями:

- тестовые модули проверяют соответствующие устройства в различных режимах функционирования, которые задаются входными параметрами;

- при разработке тестовых модулей должны соблюдаться требования сопряжения с диспетчером тестов;

- при разработке тестовых модулей учитывается возможность создания копии теста. Тест и его копия должны выполняться независимо и не влиять друг на друга.

Еще одной особенность созданного. является то, что предварительно сформированные режимы позволяют выполнять тестирование аппаратуры одновременно на всех вычислительных узлах комплекса по инициативе сервера.

Пользовательский интерфейс позволяет просматривать структуру тестовой системы, изменять эту структуру, добавляя и/или удаляя тесты, группы, режимы, запускать тесты и группы тестов, наблюдать за процессом тестирования.

Структура диспетчера тестов предусматривает его функционирование в среде любой операционной системы, поддерживающей интерфейс POSIX.

Предложенная организация диспетчера тестов является унифицированной, так как может быть использована с равным успехом в мультипроцессорных комплексах при наличии для вычислительных узлов общей памяти, а также для одиночных БЦВМ.

Список литературы

[1]. Михайлов В.А., Штейнберг В.И. Гонка без финиша // Радиопромышленность. 2009. № 2. С.19-35.

[2]. Михайлов В.А., Попов С.О., Штейнберг В.И. Опыт разработки вычислительных средств для авиационных терминалов связи с использованием СБИС СНК 1867ВЦ8Ф // Радиопромышленность. 2013. № 4. С.11-17.

[3]. Шпиев В.А. Вопросы методологии построения унифицированных базовых вычислительных средств для БВК // Радиопромышленность. 2009. № 2. С.87-94.

[4]. Аркусская С.В., Терехов А.А. Технология тестирования бортовых вычислительных комплексов // Наукоемкие технологии. 2014. Т.15. № 1. С.070-082.