

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.415.2

СИСТЕМА АКТИВНОГО МОНИТОРИНГА ПАКЕТОВ И РЕСУРСОВ ПЛАТФОРМЫ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

© 2013 г.

В.А. Карбовский

НИИ наукоемких компьютерных технологий Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики

vladislav.k.work@gmail.com

Поступила в редакцию 30.11.2012

Предлагается подход к применению предметно-ориентированного тестирования корректности выполнения пакетов и работоспособности ресурсов, используемых облачными платформами, на примере системы активного мониторинга пакетов и ресурсов многопрофильной инструментально-технологической платформы CLAVIRE, а также описывается архитектура данной системы мониторинга.

Ключевые слова: предметно-ориентированное тестирование, система мониторинга, платформа облачных вычислений, CLAVIRE, Easyflow.

Введение

Непрерывно возрастающая сложность современных систем распределенных вычислений связана с увеличением числа ресурсов этих систем. Под ресурсами платформы стоит понимать как вычислительные ресурсы, к которым можно отнести персональные компьютеры, серверы, кластеры, распределенные вычислительные инфраструктуры и многое другое, а также ряд прикладных пакетов, которые и занимаются расчетом поставленной пользователем задачи.

Каждый вычислительный пакет имеет свой собственный уникальный формат как входных, так и выходных данных, а также особенности выполнения, например, часть пакетов может завершить свой расчет без оповещения об ошибке, однако результат, по тем или иным причинам может оказаться некорректным. В связи с этим традиционные методы тестирования не могут полностью решить задачу проверки правильности выполнения пакета или работоспособности ресурса, обслуживающего данный пакет. Для решения поставленной задачи следует применять подход предметно-ориентированного тестирования, т.е. тестирова-

ния, направленного на предметное содержание объекта.

Многопрофильная инструментально-технологическая платформа (МИП) CLAVIRE включает в себя оба типа ресурсов: вычислительные ресурсы и прикладные пакеты. Таким образом, тестирование направлено на две группы предметов, предполагая выполнение следующих категорий проверок:

1. Тестирование вычислительных ресурсов. Предполагает определение доступности подключенных ресурсов, корректной их настройки для работы в среде. Проверяется тестовым запуском тестового пакета.

2. Тестирование вычислительных пакетов. Позволяет определить, корректно ли развернуты вычислительные пакеты на ресурсах за счет тестового запуска специально сформированного задания.

Тестирование основано на исполнении заранее подготовленных сценариев, позволяющих проверять работоспособность среды. Сценарии подготавливаются экспертами пакета, обладающими необходимым опытом. Эксперт формирует входные данные, а также указывает ожидаемые выходные данные, основываясь на апри-

орном знании. Результатом работы сценария является выходной файл, который проверяется на соответствие ожидаемому. Таким образом, проверка каждого сценария формируется индивидуально.

Разрабатываемая система призвана обеспечить возможность покрытия ресурсов многопрофильной инструментально-технологической платформы предметно-ориентированными тестами, подготовленными экспертами.

1. МИТП CLAVIRE

МИТП создания и управления распределенной средой облачных вычислений CLAVIRE (CLOUD Applications VIRTUAL Environment) [1] предназначена для создания, исполнения и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим в облаке неоднородных вычислительных ресурсов корпоративного уровня, уровня центров компетенции, центров обработки данных, инфраструктур экстренных вычислений и распределенных хранилищ данных.

МИТП CLAVIRE разрабатывается в СПбГУ ИТМО по заказу ЗАО «АйТи» в составе проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологического производства».

Областью применения МИТП является создание инновационных производств на основе предоставления пользователям через сеть Интернет высокопроизводительных предметно-ориентированных сервисов в рамках модели облачных вычислений для различных нужд науки, промышленности, бизнеса и социальной сферы с возможностью использования ресурсов уже существующих распределенных вычислительных инфраструктур.

Особенности применения и конфигурирования МИТП в конкретных предметных областях определяются в каждом случае общими потребностями данной области в широкодоступных высокопроизводительных вычислениях и наличием прикладного программного обеспечения (ПО), допускающего предоставление в форме облачных сервисов.

В состав МИТП входят четыре высокоуровневые подсистемы. К ним относятся:

1. Подсистема человеко-компьютерного взаимодействия (ЧКВ), обеспечивающая диалог между пользователем и системой, работу с системой управления вычислениями и инструментарий для работы с данными.

2. Подсистема работы с данными и знаниями. Отвечает за хранение состояния платформы, характеризуемого: данными, накопленными в ходе эксплуатации платформы, данными о конфигурации платформы и рабочим набором формализованных знаний, используемых в процессе функционирования платформы.

3. Инфраструктурная подсистема. Обеспечивает внутреннюю целостность ядра платформы за счет предоставления сервисных механизмов взаимодействия компонентов; она является связующим звеном в работе других подсистем.

4. Подсистема исполнения. Включает в себя модули контроля вычислительной инфраструктуры и управления выполнением заданий на вычислительных сервисах, входящих в ее состав. Обеспечивает доступ к основным функциональным возможностям вычислительной облачной инфраструктуры.

Система активного мониторинга пакетов и ресурсов взаимодействует с подсистемой исполнения МИТП.

2. Предметно-ориентированное тестирование

Предметно-ориентированное тестирование – это метод создания тестов на основе анализа предметной области и ее моделирования [2, 3]. Процесс создания теста состоит из трех этапов:

1. Подготовка сценария. Заключается в выборе задач.

2. Подготовка входных данных. На этом этапе подбираются такие входные данные, которые позволяют выполнить сценарий с возможностью получения выходных данных, близких по значению к ожидаемым.

3. Подготовка данных, основанных на априорном знании, а также на результатах анализа и моделирования предметной области. Эти данные, как эталонные, будут использоваться при сравнении выходных данных, полученных при реальном расчете с использованием тестируемого ресурса или пакета.

Работу по подготовке теста выполняет эксперт указанной предметной области, т.к. именно эксперт обладает всеми необходимыми знаниями, позволяющими составить корректный сценарий и подобрать соответствующие данные [4].

В системе активного мониторинга пакетов и ресурсов принят стандарт формализации тестов

вого сценария на основе языка XML3. Формализованный сценарий состоит из трех элементов (рис. 1):

1. Элемент дерева XML `<data>`. Содержит список необходимых сценарию входных файлов, а также соответствие этих файлов параметрам сценария.

2. Элемент `<efscript>` – непосредственно сценарий выполнения тестового задания, представляет собой Workflow4 (WF) [5] который формализуется в виде скрипта на специальном предметно-ориентированном языке программирования – EasyFlow [6]. Такой подход является предпочтительным, так как обладает гибкостью, выразительностью, позволяет поддерживать тесты в понятном и читаемом виде.

3. Элемент `<checkblock>`. Содержит в себе список ожидаемых выходных файлов, а также значения (или интервалы значений). Соответствие указанных значений полученным значениям из выходных файлов реального расчета говорит об успешно пройденном тесте.

Таким образом, тестирование сводится к успешному или неуспешному (с выводом информации об ошибке) выполнению сценария.

```
<?xml version="1.0" encoding="UTF-8"?>
<taskscript>
  <data>
    <file name="in0" efname="input0"/>
    <file name="in1" efname="input1"/>
    <param name="in_format" value="short"/>
  </data>
  <efscript>
    require in0, in1;
    step a runs testp
    (
      inf0 = in0,
      inf1 = in1,
      operation = "plus"
    );
  </efscript>
  <checkblock>
    <file name="test.out" value="successful"/>
  </checkblock>
</taskscript>
```

Рис. 1. Формализованный тестовый сценарий

Характерной чертой языка EasyFlow, используемого в блоке `<efscript>`, является полное абстрагирование от особенностей распределенной вычислительной среды, в которой работает пользователь. Фактически EasyFlow — это высокоуровневый язык описания абстрактных workflow (AWF) [5]. Такой подход позволяет описывать саму решаемую задачу, а не способ ее исполнения на конкретной вычислительной архитектуре.

Таким образом, все WF, описанные на языке EasyFlow, полностью независимы от конкретной архитектуры вычислений и хранения данных, что позволяет пользователям распределенной среды беспрепятственно обмениваться ими и запускать их на различных вычислительных ресурсах

Рассмотренные подходы к описанию WF, реализуются многопрофильной инструментально-технологической платформой CLAVIRE [1].

3. Архитектура системы мониторинга

Концепция слоев, или уровней, одна из общеупотребительных моделей, используемых разработчиками программного обеспечения для разделения сложных систем на более простые части [7]. Слой более высокого уровня пользуется услугами, предоставляемыми нижележащим слоем, но тот не «осведомлен» о наличии соседнего верхнего слоя. Более того, обычно каждый промежуточный слой «скрывает» нижний слой от верхнего.

Расслоение системы на слои предоставляет целый ряд преимуществ:

1. Отдельный слой можно воспринимать как единое целое, не заботясь о наличии других слоев.
2. Возможность выбирать альтернативную реализацию базовых слоев.
3. Возможность сведения зависимости между слоями к минимуму.
4. Созданный слой может использоваться несколькими слоями более высокого уровня.

Для разработки системы активного мониторинга пакетов и ресурсов МИТП CLAVIRE была выбрана трехуровневая архитектура, включающая в себя три слоя (рис. 2):

- уровень представления;
- уровень бизнес-логики;
- уровень доступа к данным.

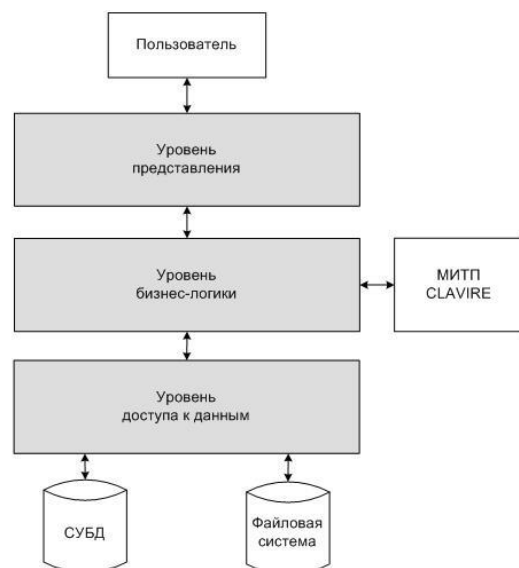


Рис. 2. Расслоение системы

Уровень представления включает все, что

имеет отношение к взаимодействию пользователя с системой. К основным функциям этого уровня относятся отображение информации и интерпретация вводимых пользователем данных с преобразованием их в соответствующие операции в рамках бизнес-логики и уровня доступа к данным.

Уровень доступа к данным обеспечивает доступ приложения к источникам хранения данных. Код этой категории несет ответственность за управление системами управления базами данных или файловыми системами и т.п.

Бизнес-логика описывает основные функции приложения, предназначенные для достижения поставленной перед ним цели. К таким функциям относятся вычисления на основе введенных пользователем и хранимых данных, проверка всех элементов данных и обработка команд, поступающих от уровня представления, а также передача информации уровню доступа к данным.

НИР выполнена в соответствии с проектом №14.В37.21.0176 Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. Выполнение работ по проекту соответствует выполняемым в последнее время научно-образовательным программам по развитию суперкомпьютерного образования в стране [8–11].

Заключение

На основании проведенных исследований можно сделать следующие выводы:

1. Разработанная система позволяет использовать перспективную технологию предметно-ориентированного тестирования в рамках ресурсов и пакетов многопрофильной инструментально-технологической платформы создания и управления распределенной средой облачных вычислений CLAVIRE.

2. Система позволяет обеспечить, при наличии соответствующих экспертов, полноту покрытия тестами любых необходимых пакетов и ресурсов.

3. Система использует современные технологии и методы разработки, отвечая требованиям реального применения.

Примечания

1. Грид-системы различного уровня, суперкомпьютеры с удаленным доступом.

2. Также распространен английский перевод термина – Urgent Computing.

3. Англ. eXtensible Markup Language (XML) – расширенный язык разметки.

4. Англ. Workflow (WF) – поток работ, заданий.

Список литературы

1. Бухановский А.В., Васильев В.Н., Виноградов В.В., Смирнов Д.Ю., Сухоруков С.С., Яппаров Т.Г. CLAVIRE: Перспективная технология облачных вычислений второго поколения // Изв. вузов. Приборостроение. Современные тенденции развития распределенных вычислений. 2011. Т. 54, № 10. С. 7–14.

2. Avik S. Domain specific test case generation using higher ordered languages for specification: Ph.D dissertation. 2005. P. 12–24.

3. Barnett M., Grieskamp W., Nachmanson L., Schulte W., Tillmann N., Veanes M. Model-Based Testing with AsmL.NET // Proc. 1st European Conference on Model-Driven Software Engineering, December 2003. P. 43.

4. Ковальчук С.В., Маслов В.Г. Интеллектуальная поддержка процесса конструирования композитных приложений в распределенных вычислительных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 32–34.

5. Марьин С.В., Ковальчук С.В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде. // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 21–23.

6. Князьков К.В., Ларченко А.В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 36–43.

7. Fowler M. Patterns of Enterprise Application Architecture // Addison Wesley. 2003. P. 45–67.

8. Гергель В.П., Стронгин Р.Г. Опыт Нижегородского университета по подготовке специалистов в области суперкомпьютерных технологий // Вестник Нижегородского университета им. Н.И. Лобачевского. 2010. № 3-1. С. 191–199.

9. Воеводин В.В., Гергель В.П. Суперкомпьютерное образование: третья составляющая суперкомпьютерных технологий // Вычислительные методы и программирование: новые вычислительные технологии. 2010. Т. 11, № 2. С. 117–122.

10. Гергель В.П., Линев А.В., Мееров И.Б., Сысоев А.В. Об опыте проведения программ повышения квалификации профессорско-преподавательского состава по направлению высокопроизводительные вычисления // Открытое и дистанционное образование. 2010. № 3. С. 15–20.

11. Воеводин В.В., Гегель В.П., Соколинский Л.Б., Демкин В.П., Попова Н.Н., Бухановский А.В. Развитие системы суперкомпьютерного образования в России: текущие результаты и перспективы // Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. № 4. С. 268–274.

**ACTIVE MONITORING SYSTEM OF PACKAGES AND RESOURCES
FOR CLOUD COMPUTING PLATFORM**

V.A. Karbovskii

We propose an approach of applying domain-oriented testing of the packages correctness and resources performance using by cloud platforms for the example of active monitoring system for packages and resources of technology platform CLAVIRE, and also describes the architecture of the monitoring system.

Keywords: domain-oriented testing, monitoring system, cloud computing platform, CLAVIRE, Easyflow.