

МЕТОДЫ МОДЕЛИРОВАНИЯ И ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ ОБЛАЧНЫХ СИСТЕМ

П.А. Михайлов, Г.И. Радченко

В процессе использования промышленных грид и облачных систем, возникают вопросы, связанные с внесением изменений в структуру и алгоритмы работы распределенных вычислительных сетей (РВС) и тем как эти изменения отразятся на работоспособности системы. В статье раскрываются основные подходы к методологиям экспериментального исследования РВС, и облачных систем в частности. Оцениваются достоинства и недостатки подходов натурального моделирования, эмуляции, эталонного тестирования и симуляции РВС. Дается краткий обзор систем симуляции РВС. Результатом проведенного анализа является архитектура и реализация прототипа собственной системы моделирования частных облачных PaaS-систем. Раскрываются особенности реализации системы, а также результаты испытания разработанных моделей на облачной платформе Mjolnirг.

Ключевые слова: распределенные вычислительные системы, моделирование, симуляция, облачные вычисления, облака, Mjolnirг.

Введение

В процессе разработки и эксплуатации распределенных вычислительных систем, в особенности облачных вычислительных систем, возникают вопросы, связанные изменением структуры данных систем, алгоритмов их работы и архитектуры. Особый интерес представляет исследование влияния этих изменений на ключевые параметры распределенной вычислительной системы: производительность, безопасность, отказоустойчивость, масштабируемость и др. В настоящее время проводится множество исследований, посвященных изучению поведения больших распределенных систем, а также создается программное обеспечение для осуществления этих исследований. Нами предлагается собственное решение для симуляции алгоритмов планирования в частных облачных платформах, на основе системы CloudSim.

Дальнейшая структура статьи выглядит следующим образом. В разделе 1 представлены моделирование распределенных вычислительных, включая натурное моделирование, эмуляцию, эталонное тестирование и симуляцию таких систем. Раздел 2 посвящен обзору систем симуляции распределенных вычислительных систем, включая такие системы как CloudSim, CDOSim, TeachCloud, SPECI, DCSim. В разделе 3 дается обзор наиболее распространенных подходов к планированию облачных ресурсов: централизованного, иерархического и распределенного планирования. В разделе 4 представлена архитектура и особенности реализации разработанной нами системы моделирования частных облачных PaaS-платформ. В последнем разделе кратко изложены результаты испытания системы моделирования частных облачных PaaS-платформ на примере платформы Mjolnirг. В заключении представлены результаты работы и направления дальнейших исследований.

1. Технологии моделирования распределенных вычислительных систем

В статье [1] предлагается следующая классификация методологий экспериментального исследования крупномасштабных и распределенных вычислительных сетей (рис. 1).

Натурное моделирование — это метод, при котором реальная задача (некоторое приложение, сервис и т.д.) выполняется в реальной вычислительной среде (операционной системе, сервере и др.). Процесс выполнения задачи при натурном моделировании отличается от штатного тем, что в процессе выполнения собирается статистическая информация о работе приложения и состоянии системы. Натурное моделирование применяется в тех случаях, когда сложное поведение и взаимодействие РВС не может быть адекватно проанализировано и смоделировано.

		Вычислительная среда	
		<i>Реальная</i>	<i>Модельная</i>
Задача	<i>Реальная</i>	Натурное моделирование	Эмуляция
	<i>Модельная</i>	Эталонное тестирование	Симуляция

Рис. 1. Классификация методологий экспериментального исследования РВС

Данная ситуация может возникать в силу различных программных (сложные алгоритмы планирования, резервирования ресурсов, различные стеки базового ПО) или аппаратных особенностей РВС (hyper-threading, управление кэшем, паразитный трафик внутри сети и др.). С другой стороны, для проведения достоверного эксперимента требуется получить полный контроль над РВС, что бывает крайне сложно в реальных условиях. Для решения данной проблемы разрабатываются широкомасштабные вычислительные среды, предназначенные исключительно для проведения экспериментов. Примерами таких систем могут служить Grid'5000 [2] или Planet-Lab [3] и др.

Эмуляция — это подход, при котором реальная задача выполняется в рамках модели вычислительной среды. Основой данного подхода является создание синтетических экспериментальных условий для исполнения реального приложения. Выделяют два типа эмуляторов. Первый тип эмуляторов основывается на технологии виртуальных машин, обеспечивая прослойку между реальным аппаратным обеспечением и исполняемой задачей (например, система Microgrid [4]). Второй тип эмуляторов исполняет задачи непосредственно на базе предоставляемого аппаратного обеспечения. В этом случае, контроль над ним обеспечивается посредством ограничения производительности (сетевых соединений или процессоров).

Эталонное тестирование состоит в том, что пакет моделирования автоматизирует процесс физического моделирования РВС путем формирования синтетических потоков заданий с целью оценки эффективности ее функционирования [5]. Яркими примерами таких методик являются системы HPL Linpack [6] и пакет NAS [6]. Модельное приложение должно охватывать и измерять такие характеристики тестируемой вычислительной среды, как процессорные мощности, пропускная способность сети, скорость ввода/вывода данных и т.д. Главным отличием модельного приложения от реального явля-

ется то, что при выполнении модельного приложения собираются данные о функционировании системы в процессе вычисления, а результаты выполнения самого приложения не имеют значения.

Симуляция — это подход, при котором модельное приложение выполняется в рамках модели вычислительной среды. Симуляция позволяет сфокусировать процесс на моделировании определенной части среды, абстрагируясь от остальной части системы. Такой подход позволяет достичь высокого уровня повторяемых результатов в рамках большого набора различных платформ и экспериментальных условий, позволяя оценить поведение РВС при изменяющихся условиях и, на основе этого, оптимизировать стратегии управления потоками задач. Главным достоинством данного подхода является гибкость системы, так как и приложение и вычислительная среда являются моделями, есть возможность легко изменять условия эксперимента. Минусом же является крайне высокая сложность разработки моделей приложения и вычислительной среды. Примерами средств симуляции могут служить такие проекты, как GridSim [8], SimGrid [9], CloudSim [10] и другие.

2. Обзор систем симуляции распределенных вычислительных систем

В настоящее время проводится множество исследований посвященных изучению поведения больших распределенных систем, а также создается программное обеспечения для осуществления этих исследований. Примерами подобного рода ПО могут выступать GridSim, SimGrid и CloudSim. В то время как первые два решения нацелены на моделирование грид-систем, CloudSim является одной из немногих платформ, ориентированных на моделирование облачных вычислительных систем.

Стоит отметить, что средства моделирования грид-систем являются подходящим решением для создания моделей очень больших облачных вычислительных систем.

Однако виртуализация позволяет разворачивать частные облачные системы на небольших испытательных стенда, где важную роль играет детальное моделирование политик резервирования облачных ресурсов, сервисов, загруженности приложений и др. Поэтому создания детальных моделей облачных вычислительных систем необходимо использовать ПО, разработанное непосредственно для моделирования облачных вычислительных систем.

2.1. Платформа CloudSim

Платформа CloudSim — это обобщенное и масштабируемое средство симуляции, которое позволяет осуществлять полноценное моделирование и симуляцию облачных вычислительных систем и инфраструктур [10, 11]. Она является расширением базовой функциональности платформы GridSim, обеспечивая возможность моделирования хранилища данных, веб-сервисы, распределение ресурсов между виртуальными машинами и др.

На рис. 2 представлена многоуровневая архитектура платформы CloudSim.

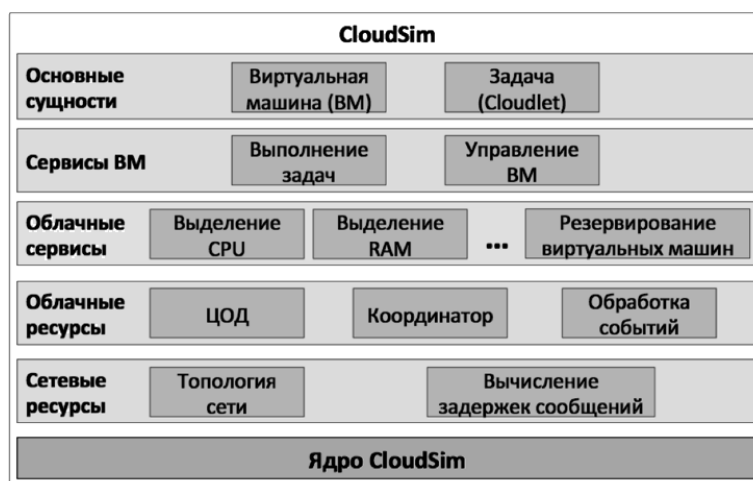


Рис. 2. Архитектура платформы CloudSim

Ядро CloudSim основывается на движке SimJava, который поддерживает такие базовые функции, как использование очередей при обработке событий, создание облачных сущностей (сервисов, узлов, центров обработки данных, брокеров ресурсов и виртуальных машин), взаимодействие между элементами системы и управление протеканием симуляции. При разработке модели облачной среды, пользователю необходимо произвести доработку ключевых для своей модели компонентов для достижения результатов, максимально приближенных к реальности.

Основными сущностями платформы при моделировании являются Виртуальная машина и Задача. Данные компоненты являются специфичными для различных групп облачных систем. Так, например, для облачных систем типа PaaS будет характерно развертывание нескольких приложений на одной виртуальной машине.

Наиболее важными компонентами при моделировании являются компоненты, отвечающие за политики управления ресурсами. К задачам, которые решают эти компоненты относятся следующие:

- выделение процессорных мощностей, оперативной памяти и других ресурсов для различных сущностей моделируемой системы;
- развертывание виртуальных машин на узлах моделируемой системы;
- распределение задач между виртуальными машинами моделируемой системы.

Принцип работы модели, основанной на платформе CloudSim, подразумевает доработку необходимых базовых компонентов платформу и предварительное описание моделируемой системы и сценария проведения симуляции в виде исходного кода. После запуска симуляции все данные о моделируемой системе передаются в ядро CloudSim, где производится симуляция.

Стоит отметить, что платформа CloudSim не подразумевает изменений в моделируемой системе или сценарии проведения симуляции непосредственно во время функционирования модели, что накладывает некоторые ограничения на возможности платформы.

2.2. CDOSim

Платформа CDOSim [12] (Cloud Deployment Options Simulator) — это симулятор, основной задачей которого является оценка производительности облачной вычислительной системы или инфраструктуры. Платформа CDOSim предназначена для сравнения

различных вариантов развертывания облачных систем и инфраструктур, что позволяет оптимизировать использование доступных ресурсов и повысить производительность облачной вычислительной системы. Платформа CDOSim позволяет гибко изменять основные параметры развертывания облачных систем и инфраструктур:

- стратегии резервирования ресурсов системы;
- конфигурации экземпляров виртуальных машин;
- аппаратные и программные средства, используемые для разработки вычислительной системы;
- параметры сети.

2.3. TeachCloud

Платформа TeachCloud [13] — это симулятор облачных вычислительных систем, который специально разрабатывался для обучения. Платформа TeachCloud предоставляет простой графический интерфейс, используя который ученики могут изменять конфигурацию и настройки облачной вычислительной системы, проводить эксперименты с возможными конфигурациями такой системы.

2.4. iCanCloud

Платформа iCanCloud — это программный симулятор больших сетевых хранилищ данных [14]. Платформа iCanCloud позволяет максимально оптимизировать использование ресурсов конкретным приложением в конкретной вычислительной среде. Платформа имеет полноценный графический интерфейс, который позволяет полностью спроектировать и выполнить симуляцию облачного хранилища данных. Кроме того, платформа iCanCloud позволяет выполнить распараллеливание симуляции облачной вычислительной системы.

2.5. SPECI

Программа SPECI [15] (Simulation Program for Elastic Cloud Infrastructures) — это симулятор который позволяет воссоздавать и исследовать поведение и масштабирование крупных центров обработки данных. Программа SPECI направлена на работу с центрами обработки данных, которые уже спроектированы, но еще не построены. Использование программы SPECI в подобных случаях позволяет выявить уязвимости и «узкие» места системы.

2.6. DCSim

Платформа DCSim (Data Center Simulator) — это симулятор, который в первую очередь предназначен для работы с облачными вычислительными IaaS-системами [16], где основной проблемой является выбор подходящей в каждом конкретном случае политики выделения и резервирования ресурсов. Платформа DCSim позволяет значительно ускорить процесс разработки и запуска IaaS-системы.

3. Подходы к планированию облачных ресурсов

Планирование ресурсов является критически важным аспектом управления ресурсами облачных вычислительных систем. Соответственно, возможность моделирования различных подходов и алгоритмов планирования ресурсов является необходимой возможностью любой системы моделирования распределенных вычислительных систем.

Планирование подразумевает совместное использование ресурсов на нескольких уровнях [17]:

- *аппаратный уровень*, т.е. совместное использование сервера между несколькими виртуальными машинами;
- *уровень виртуальных машин*, т.е. каждая виртуальная машина может использоваться одновременно несколькими приложениями;
- *прикладной уровень*, т.е. каждое приложение может выполнять несколько задач параллельно.

Основной задачей планировщика облачной вычислительной системы является увеличение пропускной способности системы, то есть увеличение количества задач, завершенных за единицу времени, и сокращение времени выполнения каждой задачи.

Существуют различные подходы к планированию ресурсов в облачных вычислительных системах. Наиболее распространенными являются:

- централизованное планирование;
- иерархическое планирование;
- распределенное планирование.

3.1. Централизованное планирование ресурсов

При централизованном подходе, планирование ресурсов осуществляется одним главным узлом, который обладает актуальной информацией о состоянии всех компонентов системы [18].

При использовании этого подхода все задачи отправляются на главный узел, где они попадают в очередь и ожидают подбора ресурсов для выполнения. Схема работы системы, основанной на данном подходе, представлена на рис. 3.

Основной проблемой этого подхода является малая масштабируемость при росте количества компонентов системы, так как главный узел является «узким местом» системы. Основным достоинством этого подхода является высокая эффективность планирования, так как главный узел обладает всей необходимой для планирования информацией. Кроме того, этот подход является относительно простым в реализации.

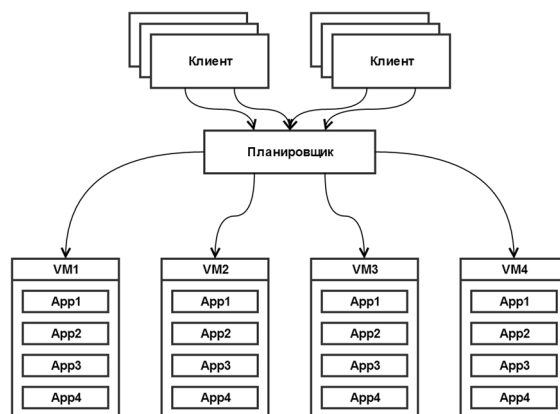


Рис. 3. Схема работы централизованного планировщика ресурсов облачной вычислительной системы.

3.2. Иерархическое планирование ресурсов

Данный подход подразумевает, что кроме главного планировщика, который получает все задачи у каждого узла или группы узлов есть свой собственный планировщик (рис. 4), на который происходит перенаправление заданий с главного узла [19].

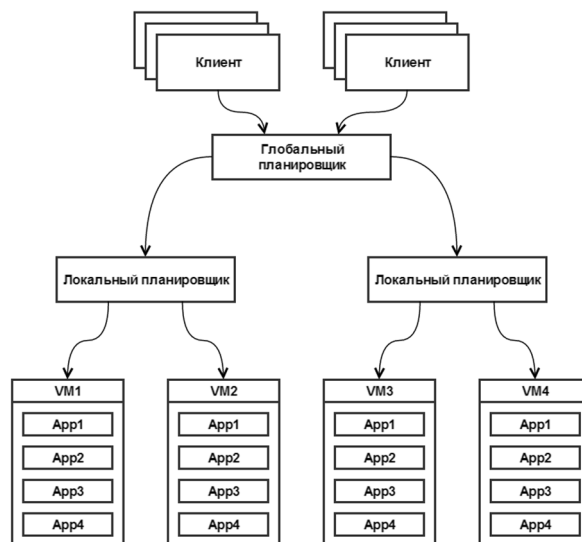


Рис. 4. Схема работы иерархического планировщика ресурсов облачной вычислительной системы

Таким образом, главный узел отвечает за глобальное планирование, а вспомогательные планировщики за локальное. Основным достоинством данного подхода является то, что он позволяет использовать разные алгоритмы для глобального и локального планирования выполнения задач.

3.3. Распределенное планирование ресурсов

При использовании данного подхода система разделяется на группы узлов, у каждой из которых есть свой собственный планировщик задач, способный принимать новые задачи из вне.

Существует два возможных способа реализации данного подхода.

Прямое взаимодействие планировщиков (рис. 5). У каждого планировщика есть список других доступных планировщиков, что позволяет каждому из них получать полную информацию о состоянии системы. Процесс планирования ресурсов для выполнения задачи строится следующим образом: если задача не может быть выполнена на доступных ресурсах, то производится подбор планировщика, которому доступны необходимые ресурсы, и задача передается ему.

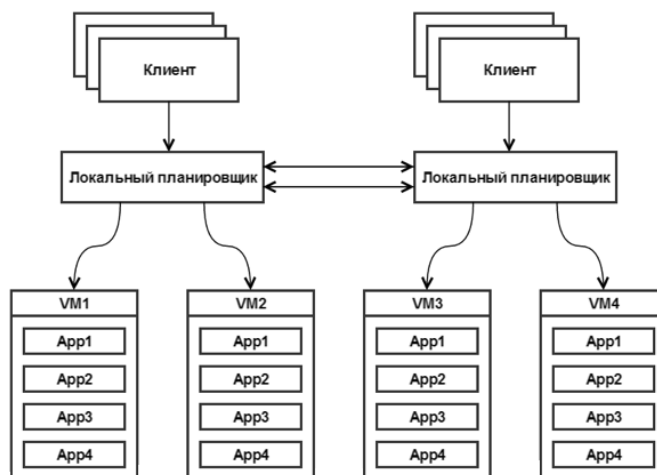


Рис. 5. Схема работы децентрализованного планировщика ресурсов облачной вычислительной системы с прямым взаимодействием

Взаимодействие планировщиков через общий пул задач (рис. 6). При использовании данного подхода планировщики не знают о состоянии друг друга. Взаимодействие происходит через общий пул задач. Процесс планирования ресурсов для выполнения задачи строится следующим образом: если задача не может быть выполнена, то она помещается в общий пул задач, откуда ее забирает для выполнения планировщик, который может ее выполнить. Данный подход может быть модифицирован так, что все задачи изначально будут поступать в общий пул задач, что в некоторых случаях может быть более выгодно.

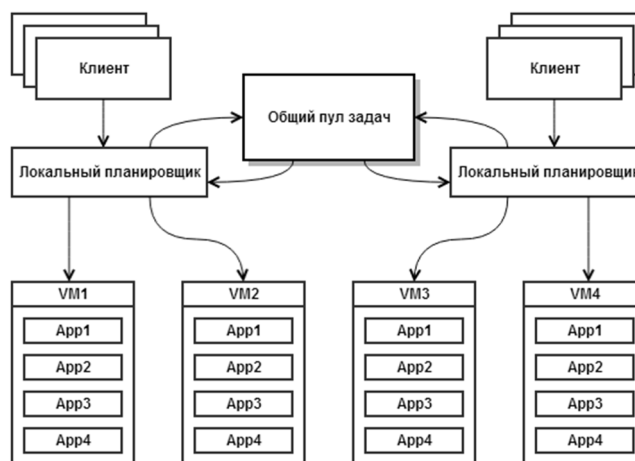


Рис. 6. Схема работы децентрализованного планировщика ресурсов облачной вычислительной системы с взаимодействием через общий пул

4. Система моделирования частных облачных PaaS-платформ

Для моделирования эффективности различных алгоритмов планирования ресурсов, нами была разработана система для моделирования и оценки производительности алгоритмов планирования частных облачных PaaS-систем. В качестве основы для создания собственной системы моделирования, нами была использована платформа

CloudSim — масштабируемая платформа симуляции, позволяющая осуществлять полноценное моделирование и симуляцию облачных вычислительных систем и инфраструктур.

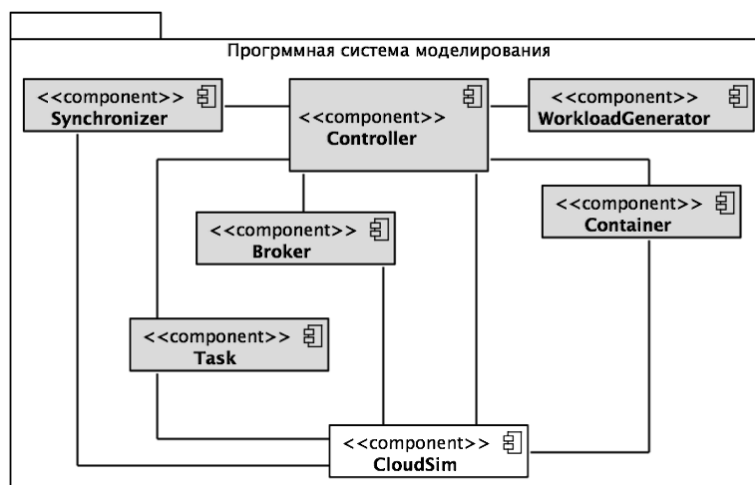


Рис. 7. Компоненты программной системы моделирования частных облачных PaaS-систем

Система моделирования частных облачных PaaS-систем состоит из шести компонентов (рис. 7):

1. *Task* — компонент, описывающий структуру и особенности выполнения задач, выполнение которых симулирует система. Данный компонент обеспечивает возможность симуляции выполнения вложенных подзадач, а также возможность указания приложения, которым должно быть выполнено данное задание.
2. *Container* — компонент, обеспечивающий функционал виртуальной машины в рамках симуляции. Обеспечивает возможность моделирования развертывания приложений, а также управления их жизненным циклом.
3. *WorkloadGenerator* — компонент, который осуществляет во время симуляции генерацию нагрузки в режиме реального времени. Данный компонент обеспечивает периодическое добавление задач в подсистему планирования моделируемой системы, что позволяет симулировать предсказуемую динамическую нагрузку на систему.
4. *Synchronizer* — компонент, обеспечивающий корректное внесение изменений в моделируемую систему во время выполнения симуляции. В ходе симуляции необходимость синхронизации возникает при симуляции спонтанной нагрузки, а также при использовании децентрализованного подхода к планированию, при котором за планирование ресурсов отвечают несколько планировщиков ресурсов.
5. *Broker* — компонент, реализующий управление ресурсами моделируемой системы во время симуляции. В задачи данного компонента входит распределение виртуальных машин по имеющимся в моделируемой системе узлам, распределение приложений по виртуальным машинам, распределение задач по приложениям.
6. *Controller* — компонент, обеспечивающий подготовку сущности моделируемой системы на основе содержимого конфигурационного файла, подготовленного пользователем.

Основной для моделирования облачной системы является конфигурационный файл модели — JSON документ, содержащий подробное описание аппаратной инфраструкту-

ры моделируемой системы, а также сценарий проведения симуляции. Конфигурационный файл состоит из двух блоков:

- *datacenter* — содержит описание инфраструктуры моделируемой системы;
- *workload* — описание сценария проведения симуляции.

В блоке *datacenter* описываются физические узлы (блок *hosts*), виртуальные машины (блок *vms*), алгоритмы планирования задач и распределения приложений (блок *brokers*). Каждый описываемый физический узел в блоке *hosts* обладает рядом атрибутов:

- *quantity* — количество узлов с описываемыми характеристиками;
- *ram* — оперативная память узла;
- *storage* — доступная дисковая память;
- *bandwidth* — пропускная способность сети;
- *CPUs* — список имеющихся процессоров с вычислительной мощности, указанной в миллионах процессорных инструкций в секунду (*mips*).

Описание виртуальных машин в блоке *vms* имеет структуру, схожую с описанием физических узлов:

- *quantity* — количество виртуальных машин с описываемыми характеристиками;
- *cpu* — количество требуемых вычислительных ядер;
- *size* — размер образа;
- *ram* — требуемая оперативная память;
- *bandwidth* — пропускная способность сети;
- *mips* — требуемая вычислительная мощность в миллионах процессорных инструкций в секунду.

В блоке *brokers* указываются пути к файлам, содержащим алгоритмы планирования задач и распределения приложений в моделируемой облачной среде. Разработанная система моделирования частных облачных PaaS-систем обеспечивает возможность внедрения собственных алгоритмов управления выполнением задач и распределения приложений. Для их описания необходимо реализовать программные интерфейсы *com.model.wrapper.broker.TasksScheduler* и *com.model.wrapper.broker.AppsScheduler* (рис. 8).

```

package com.model.wrapper.broker;

import java.util.List;
import com.model.wrapper.cloudlet.Task;
import com.model.wrapper.vm.Container;
import com.model.wrapper.App;

public interface TasksScheduler {
    public void scheduleTasks(List<Task> tasks,
                             List<Container> containers);
}

public interface AppsScheduler {
    public void distributeApps(List<Task> tasks,
                              List<App> apps, List<Container> containers);
    public void redistributeAppsFor(Task task,
                                     List<Container> containers);
}
    
```

Рис. 8. Интерфейсы TaskScheduler и AppsScheduler

Описание сценария проведения симуляции в блоке *workload* содержит список задач для моделируемой системы, количество повторного добавления задач в очередь и временную задержку перед повторным добавлением задач. Описание задачи имеет следующую структуру:

- *quantity* — количество задач с аналогичным описанием;
- *length* — количество инструкций процессора, необходимых для выполнения задачи;
- *fileSize* — размер файла задания в мегабайтах;
- *outputSize* — размер результата выполнения задачи в мегабайтах;
- *app* — приложение, которое должно произвести выполнение задачи;
- *subtasks* — необязательный блок, описывающий список задач, которые будут добавлены в очередь после завершения симуляции выполнения задачи-владельца.

5. Испытание системы моделирования частных облачных PaaS-платформ

Для испытания системы моделирования частных облачных PaaS-платформ, нами была разработана модель частной облачной платформы Mjolnir [20].

Тестирование производилось на основе задачи параллельной обработки текстового файла большой размера. Платформа Mjolnir была развернута и протестирована в виртуальной распределенной вычислительной среде, состоящей из 11 узлов, один из которых был оборудован 4-ядерным процессором и 2 гигабайтами оперативной памяти и играл роль сервера, а остальные имели по одному одноядерному процессору и по 512 мегабайт оперативной памяти и играли роль пользовательских машин. Виртуальные машины были развернуты на вычислительном сервере со следующими характеристиками: 2 процессора Intel Xeon X5680 (6 ядер, 3.33 GHz) 12 GB DDR3 RAM.

Файл размером в 1 гигабайт был разбит на 100 фрагментов по 10 мегабайт. Каждый фрагмент был отправлен в очередь для обработки. Запущенные экземпляры обработчика брали отправленные фрагменты и подсчитывали количество вхождений для каждого слова в тексте. Результаты эксперимента, а также сравнение экспериментальных результатов и результатов симуляции представлены на рис. 9.

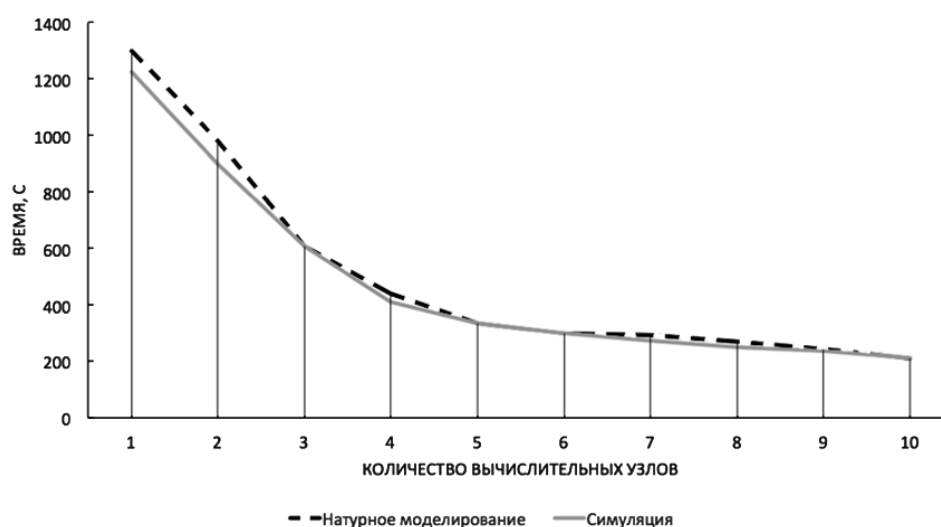


Рис. 9. Сравнение результатов симуляции и натурального моделирования

Сравнение результатов работы модели и данных вычислительного эксперимента показало, что относительная погрешность оценки времени выполнения задач платформы Mjolnir составляет 5.2 %, что показывает, что модель корректна и пригодна для разработки алгоритмов планирования созданной платформы.

Заключение

В данной работе, нами был выполнен обзор технологий моделирования распределенных вычислительных систем. Были проанализированы возможности наиболее распространенных на сегодняшний день платформ симуляции облачных систем. Также, были рассмотрены основные подходы к планированию облачных ресурсов. На основе обзора, нами разработана архитектура и проведены испытания системы моделирования частных облачных PaaS-платформ. Сравнение результатов работы модели и данных вычислительного эксперимента показало, что модель корректна и пригодна для разработки алгоритмов планирования созданной платформы.

В качестве развития разработанной системы, планируется создание веб-приложения, обеспечивающего удобный пользовательский интерфейс для решения задач симуляции частных облачных PaaS-систем.

Работа выполнена при поддержке Российского Фонда Фундаментальных исследований (грант № 14-07-00420).

Литература

1. Gustedt, J. Experimental methodologies for large-scale systems: a survey / J. Gustedt, E. Jeannot, M. Quinson // Parallel Process. Lett. — World Scientific, 2009. — Vol. 19. — P. 399–418.
2. Bolze, R. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed / R. Bolze, F. Cappello, E. Caron, M. Dayde, F. Desprez et al. // Int. J. High Perform. Comput. Appl. — USA: Sage Publications, 2006. — Vol. 20. — P. 481–494.
3. Chun, B. Planetlab: an overlay testbed for broad-coverage services / B. Chun, D. Culler, T. Roscoe // ACM SIGCOMM. — USA: ACM, 2003. — Vol. 33. — P. 3–12.
4. Song, H.J. The MicroGrid: a Scientific Tool for Modeling Computational Grids / H.J. Song // Proc. IEEE Supercomput. — USA: IEEE, 2000. — P. 4–10.
5. Корсуков, А.С. Инструментальные средства полунатурного моделирования распределенных вычислительных систем / А.С. Корсуков // Современные технологии. Системный анализ. Моделирование. — Россия: Иркутский государственный университет путей сообщения, 2011. — Т. 3. — С. 105–110.
6. Endo, T. Linpack evaluation on a supercomputer with heterogeneous accelerators / T. Endo // Parallel & Distrib. Process. (IPDPS), 2010 IEEE Int. Symp. — USA: IEEE, 2010. — P. 1–8.
7. Bailey, D.H. NAS parallel benchmark results / D.H. Bailey // Proc. Supercomput. '92. — USA: IEEE, 1992. — P. 1–13.
8. Buyya, R. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing / R. Buyya, M. Murshed // Concurr. Comput. Pract. Exp. — USA: Wiley, 2002. — Vol. 14. — № 13-15. — P. 1175–1220.

9. Quinson, M. SimGrid: a generic framework for large-scale distributed experiments / M. Quinson // 2009 IEEE Ninth Int. Conf. Peer-to-Peer Comput. — USA: IEEE, 2009. — P. 126–131.
10. Calheiros, R.N. CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. / R.N. Calheiros. Eprint: Australia, 2009. — 9 p.
11. Buyya, R. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities / R. Buyya, R. Ranjan, R.N. Calheiros // 2009 Int. Conf. High Perform. Comput. Simul. — USA: IEEE, 2009. — P. 1–11.
12. Fittkau, F. CDOSim: Simulating cloud deployment options for software migration support / F. Fittkau, S. Frey, W. Hasselbring // 2012 IEEE 6th Int. Work. Maint. Evol. Serv. Cloud-Based Syst. — USA: IEEE, 2012. — P. 37–46.
13. Jararweh, Y. TeachCloud: a cloud computing educational toolkit / Y. Jararweh et al. // Int. J. Cloud Comput. 2012. — InderScience Publ., 2012. — Vol. 2. — P. 237–257.
14. Núñez, A. iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator / A. Núñez // J. Grid Comput. 2012. — Germany: Springer, 2012. — Vol. 10. — P. 185–209.
15. Sriram, I. SPECI, a Simulation Tool Exploring Cloud-Scale Data Centres / I. Sriram // Lect. Notes Comput. Sci. 2009. — Germany: Springer, 2009. — Vol. 5931. — P. 381–392.
16. Keller, G. DCSim: A data centre simulation tool / G. Keller // Integr. Netw. Manag. 2012. — USA: IEEE, 2012. — P. 1090–1091.
17. Li, W. Modeling for Dynamic Cloud Scheduling Via Migration of Virtual Machines / W. Li, J. Tordsson, E. Elmroth // 2011 IEEE Third Int. Conf. Cloud Comput. Technol. Sci. 2011. — USA: IEEE, 2011. — P. 163–171.
18. Maguluri, S.T. Stochastic models of load balancing and scheduling in cloud computing clusters / S.T. Maguluri, R. Srikant, L. Ying // INFOCOM, 2012 Proc. IEEE. — USA: IEEE, 2012. — P. 702–710.
19. Nidhi, K. Cloud Load Balancing Techniques: A Step Towards Green Computing / K. Nidhi, I. Chana // IJCSI Int. J. Comput. Sci. Issues. 2012. — USA: Eprint, 2012. — P. 238–246.
20. Savchenko, D. Mjolnirr: private PaaS as distributed computing evolution / D. Savchenko, G. Radchenko // MIPRO 2014. Proceedings of the 37th International Convention, 2014. — USA: IEEE. — P. 386–391.

Михайлов Прохор Андреевич, магистр кафедры системного программирования Южно-Уральского государственного университета (Челябинск, Российская Федерация), prohormihailov@gmail.com

Радченко Глеб Игоревич, к.ф.-м.н., доцент кафедры системного программирования Южно-Уральского государственного университета (Челябинск, Российская Федерация), gleb.radchenko@susu.ru.

Поступила в редакцию 4 августа 2014 г.

MODELING AND PERFORMANCE EVALUATION OF CLOUD SYSTEMS

P.A. Mihailov, South Ural State University (Chelyabinsk, Russian Federation),
G.I. Radchenko, South Ural State University (Chelyabinsk, Russian Federation)

During usage of industrial grid and cloud systems, there are issues related to the changes in the structure and algorithms of distributed computing systems and how these changes will affect the system performance. The article describes the main approaches to the experimental study of methodologies for cloud systems. The strengths and weaknesses of approaches of natural modeling, simulation, benchmarking and simulation of cloud systems are evaluated. A brief review of systems simulation is provided. As the result of the analysis we present the design and implementation of a prototype of own system for simulation of private cloud PaaS-systems. We describe the implementation of the system, as well as the test results of the developed models on the example of Mjolnir cloud platform.

Ключевые слова: distributed computing systems, modeling, simulation, cloud computing, cloud, Mjolnir.

References

1. Gustedt J., Jeannot E., Quinson M. Experimental methodologies for large-scale systems: a survey // *Parallel Process. Lett.* World Scientific. 2009. Vol. 19. P. 399–418.
2. Bolze R., Cappello F., Caron E., Dayde M., Desprez, F. et al. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed / R. Bolze, F. Cappello, E. Caron, M. Dayde, F. Desprez et al. // *Int. J. High Perform. Comput. Appl.* USA: Sage Publications. 2006. Vol. 20. P. 481–494.
3. Chun B., Culler D., Roscoe T. Planetlab: an overlay testbed for broad-coverage services // *ACM SIGCOMM.* USA: ACM. 2003. Vol. 33. P. 3–12.
4. Song H.J. The MicroGrid: a Scientific Tool for Modeling Computational Grids *Proc. IEEE Supercomput.* USA: IEEE. 2000. P. 4–10.
5. Korsukov A.S. Instrumentalnyye sredstva polunaturalnogo modelirovaniya raspredelennyh vychislitelnyh system [Tools for seminatural simulation of distributed computing systems] // *Sovremennyye tekhnologii. Sistemnyy analiz. Modelirovaniye.* [Modern technology. System analysis. Modeling]. Russia: Irkutsk State University of Railway Transport. 2011. Vol. 3. P. 105–110.
6. Endo T. Linpack evaluation on a supercomputer with heterogeneous accelerators // *Parallel & Distrib. Process. (IPDPS), 2010 IEEE Int. Symp.* USA: IEEE, 2010. P. 1–8.
7. Bailey D.H. NAS parallel benchmark results // *Proc. Supercomput. '92.* USA: IEEE, 1992. P. 1–13.
8. Buyya R., Murshed M. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing // *Concurr. Comput. Pract. Exp.* — USA: Wiley. 2002. Vol. 14. № 13-15. P. 1175–1220.

9. Quinson M. SimGrid: a generic framework for large-scale distributed experiments // 2009 IEEE Ninth Int. Conf. Peer-to-Peer Comput. USA: IEEE. 2009. P. 126 — 131.
10. Calheiros R.N. CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. Eprint: Australia, 2009. 9 p.
11. Buyya R., Ranjan R., Calheiros R.N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities // 2009 Int. Conf. High Perform. Comput. Simul. USA: IEEE, 2009. P. 1–11.
12. Fittkau F., Frey S., Hasselbring W. CDOSim: Simulating cloud deployment options for software migration support // 2012 IEEE 6th Int. Work. Maint. Evol. Serv. Cloud-Based Syst. USA: IEEE. 2012. P. 37–46.
13. Jararweh Y. et al. TeachCloud: a cloud computing educational toolkit // Int. J. Cloud Comput. 2012. InderScience Publ. 2012. Vol. 2. P. 237–257.
14. Núñez A. iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator // J. Grid Comput. 2012. Germany: Springer. 2012. Vol. 10. P. 185–209.
15. Sriram I. SPECI, a Simulation Tool Exploring Cloud-Scale Data Centres // Lect. Notes Comput. Sci. 2009. Germany: Springer. 2009. Vol. 5931. P. 381–392.
16. Keller G. DCSim: A data centre simulation tool // Integr. Netw. Manag. 2012. USA: IEEE. 2012. P. 1090–1091.
17. Li W., Tordsson J., Elmroth E. Modeling for Dynamic Cloud Scheduling Via Migration of Virtual Machines // 2011 IEEE Third Int. Conf. Cloud Comput. Technol. Sci. 2011. USA: IEEE, 2011. P. 163–171.
18. Maguluri S.T., Srikant R., Ying L. Stochastic models of load balancing and scheduling in cloud computing clusters // INFOCOM, 2012 Proc. USA: IEEE. 2012. P. 702–710.
19. Nidhi K. Chana I. Cloud Load Balancing Techniques: A Step Towards Green Computing // IJCSI Int. J. Comput. Sci. Issues. 2012. USA: Eprint. 2012. P. 238–246.
20. Savchenko D., Radchenko G. Mjолnirr: private PaaS as distributed computing evolution // MIPRO 2014. Proceedings of the 37th International Convention, 2014. USA: IEEE. P. 386–391.

Received 4 August 2014.