

Основной недостаток системы - высокая стоимость содержания, так как требуются N-ое количество серверов для реализации трехуровневой архитектуры. С одной стороны проблема решается арендой сервера, но с другой стороны становится сложнее организовать синхронизацию всех серверов, чтобы пользователь видел только актуальные данные.

Приложение WalkingDog - начало новой социальной сети, которая поможет пользователям завести новые знакомства как для себя, так и для своих питомцев.

Список использованных источников:

1. Учебно-методические материалы [Электронный ресурс] - <http://www.4stud.info>
2. Хабрахабр [Электронный ресурс] - <http://habrahabr.ru>

## МОДЕЛЬ АКТОРОВ КАК СОВРЕМЕННЫЙ ПОДХОД К ОРГАНИЗАЦИИ МНОГОПОТОЧНОСТИ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Верховцов П.А.

Хмелева А.В. – канд. техн. наук, доцент

В современном мире требование к пропускной способности информационных систем растут с каждым днем. С другой стороны на фоне усложнения архитектуры систем увеличиваются требования к тестируемости, устойчивости к отказам и горизонтальной масштабируемости систем.

Модель акторов (Actorsmodel) – теоретическая модель организации параллельных вычислений и взаимодействия в информационных системах. Впервые была предложена в 1973 году Карлом Хьюиттом, Питером Бишопом и Ричардом Штайгером [1].

Пример иерархии акторов представлен на рисунке 1:

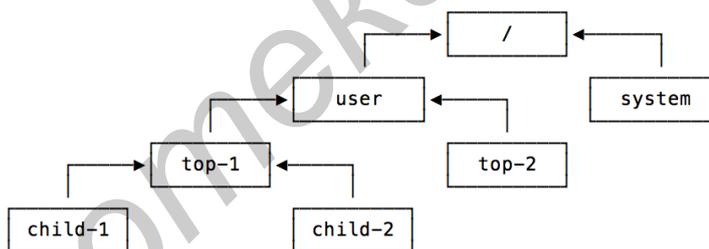


Рис. 1 – Иерархия акторов

На схеме видно, что каждый актор, за исключением корневого, имеет «родителя»

Основными особенностями модели акторов является использование параллелизма на основе передачи сообщений в совокупности с отсутствием общего изменяемого состояния что приводит к тому что огромное количество могут обрабатывать информацию, например, запросы пользователей параллельно.

При получении сообщения актор может:

- отправить конечное число сообщений другим акторам;
- создать конечное число новых акторов;
- выбрать поведение, которое будет использоваться при обработке следующего полученного сообщения.

Основные преимущества модели акторов:

- возможность создания большого количества акторов, порядка миллиона;
- полная асинхронность обработки, отсутствие блокирующих операции;
- устойчивость к отказам, благодаря механизму надзора со стороны родительских акторов, существует возможность перезапустить актор в случае ошибки;
- масштабируемость, создание нового актора не требует больших затрат, а также возможно разнесение узлов иерархии по разным физическим машинам.

Основные недостатками модели акторов:

- дополнительные расходы оперативной памяти на хранение сообщений и неизменяемых структур данных;
- большое количество «мусора» в памяти;
- сложно организовать логику, основанную на последовательном взаимодействии акторов.

Модель акторов хорошая альтернатива потокам исполнения так как позволяет декомпозировать решаемые задачи на мелкие части не пересекающиеся друг с другом по данным, что позволяет избежать блокировок и, следовательно, использование дорогостоящих операций по установке и проверке блокировки с помощью примитивов операционной системы.

Уже существует примеры успешного применения модели акторов в коммерческих разработках, такие компании как eBay, Amazon, Cisco, LinkedIn и другие используют данную модель в своих продуктах.

Список использованных источников:

1. Карл Хьюитт, Питер Бишоп, Ричард Штайгер: Универсальный модульный формализм акторов для искусственного интеллекта. IJCAI, 1973

## ОРГАНИЗАЦИЯ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

Видничук В.Н.

Данилова Г.В., м.т.н.

Сложности организации учебного процесса не дают возможности в достаточной степени освободиться от рутинной деятельности. Целью разрабатываемого дипломного проекта является создание модели для организации процессов, хранения, обработки и представления данных, а также разработка программного средства, которое может быть использовано в учебном процессе для того, чтобы эффективно организовать занятия учебного заведения и предоставить преподавателю актуальные данные о студенте.

Сегодня вопросы автоматизации и организации играют очень важную роль в жизни людей и компаний, поскольку на координирование тратится много времени. С появлением вычислительной техники появилась огромная возможность облегчить жизнь человека в вопросах правильной подачи информации. Истинные проблемы компаний почти всегда заключаются в элементах плохой организации. Стиль руководства и традиции, унаследованные от прошлого, не дают возможности в достаточной степени освободиться от рутинной деятельности.

Такая же ситуация существует и в учебных заведениях, в том числе и в высших, что затрудняет качество обучения студентов. Для того чтобы изменить такое положение, предлагается рассмотреть разрабатываемый дипломный проект по управлению организацией учебного процесса.

Цель данного дипломного проекта – разработать программное средство для организации процессов при сдаче практических работ.

Программное средство будет представлять собой клиент-серверное приложение. Сервер будет разработан на языке программирования JAVA с учетом всех современных тенденций: hibernate 4 [1], SpringMVC 4 [2], RESTfull, layout architecture project (многоуровневая архитектура проекта).

Клиент разрабатывается на платформе JAVA FX (платформа для создания десктопных приложений с насыщенным графическим интерфейсом). Данная технология выбрана в связи с её кроссплатформенностью и легким переносом на мобильные устройства.

В данном программном средстве будут задействованы роли преподавателя и студента:

Роль студента предполагает следующие действия:

- получить задание на следующую работу;
- просмотреть:
  - свои оценки за сданные работы;
  - свои несданные работы;
  - комментарии к своим работам от преподавателя.

Роль преподавателя предполагает следующие действия:

- создать работу для студентов;
- принять работу или оставив комментарий на работу, что надо доделать;
- перенаправить работу студента своему помощнику;
- просмотреть:
  - список студентов, не сдавших работы;