

# Облачная мультиагентная платформа на основе JADE и Google App Engine

Кузнецов Кирилл, 661

Научный руководитель:

к.ф.-м.н., доцент Бугайченко Д.Ю.

Рецензент:

к.ф.-м.н., Грачев М.К.

# Обоснование

- Множество мультиагентных систем, написанных на JADE
- Перенос в облако позволит:
  - Иметь доступ к ресурсам облака: хранилищу данных и вычислительным мощностям
  - Снизить издержки на поддержку и администрирование серверов

# Обоснование

Динамический доступ к ресурсам облака по необходимости важен для систем с нерегулярной нагрузкой:

- Моделирование
- Системы, требующие проводить периодический нагруженный расчет
- Системы с сезонной нагрузкой
- Приложения, для которых ожидается бурное развитие, например стартапы

# JADE

Java Agent Development Framework — одна из самых широко используемых платформ

- Библиотека для создания агентов
- Runtime для исполнения агентов
- Инструменты для отладки, мониторинга, создания агентов
- Реализована в соответствии с FIPA

# Постановка задачи

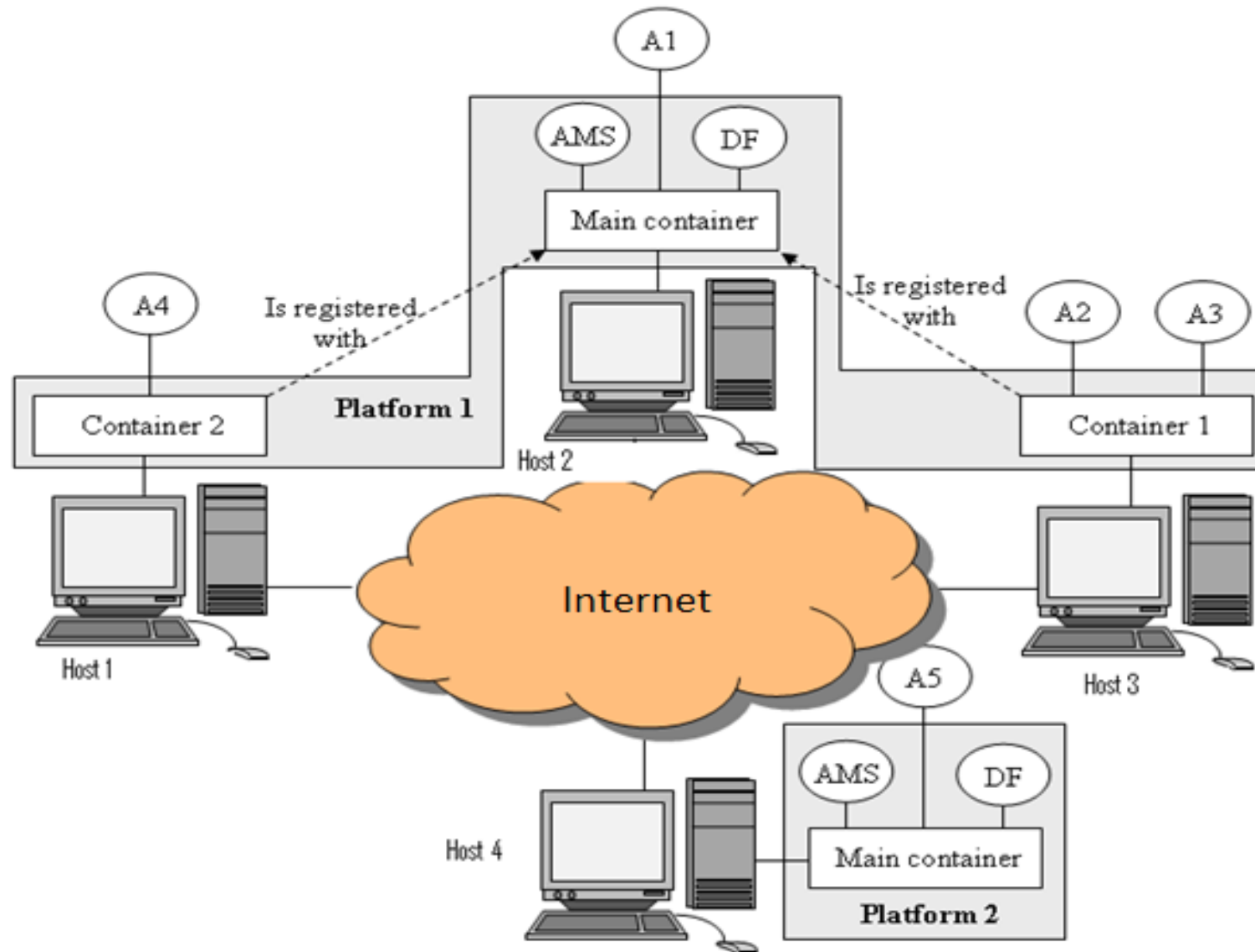
- Изучить внутреннюю архитектуру JADE
- Проанализировать существующие решения в сфере предоставления облачных сервисов, выбрать наиболее подходящий для переноса JADE
- Перенести платформу JADE в облако: мультиагентные системы созданные для выполнения в оригинальном JADE должны работать в облаке
- Предложить демонстрационную систему

# Google App Engine

Облачный хостинг для приложений на Python или Java

- Бесплатный базовый вариант (в отличие от всех конкурентов)
- Берет на себя значительную часть администрирования нагрузки
- Надежный
- Хорошая масштабируемость (лучше чем у остальных PaaS)
- Нет ограничение на использование ресурсов

# Архитектура JADE



# Ограничения

- Нет работы с TCP сокетами, общение по сети только через HTTP запросы
  - Проблема отсутствует, например, в Amazon ES2
- В стандартном варианте нет возможности создания потоков, однако она есть в Backend экземплярах – Backend threads
- Белый список классов

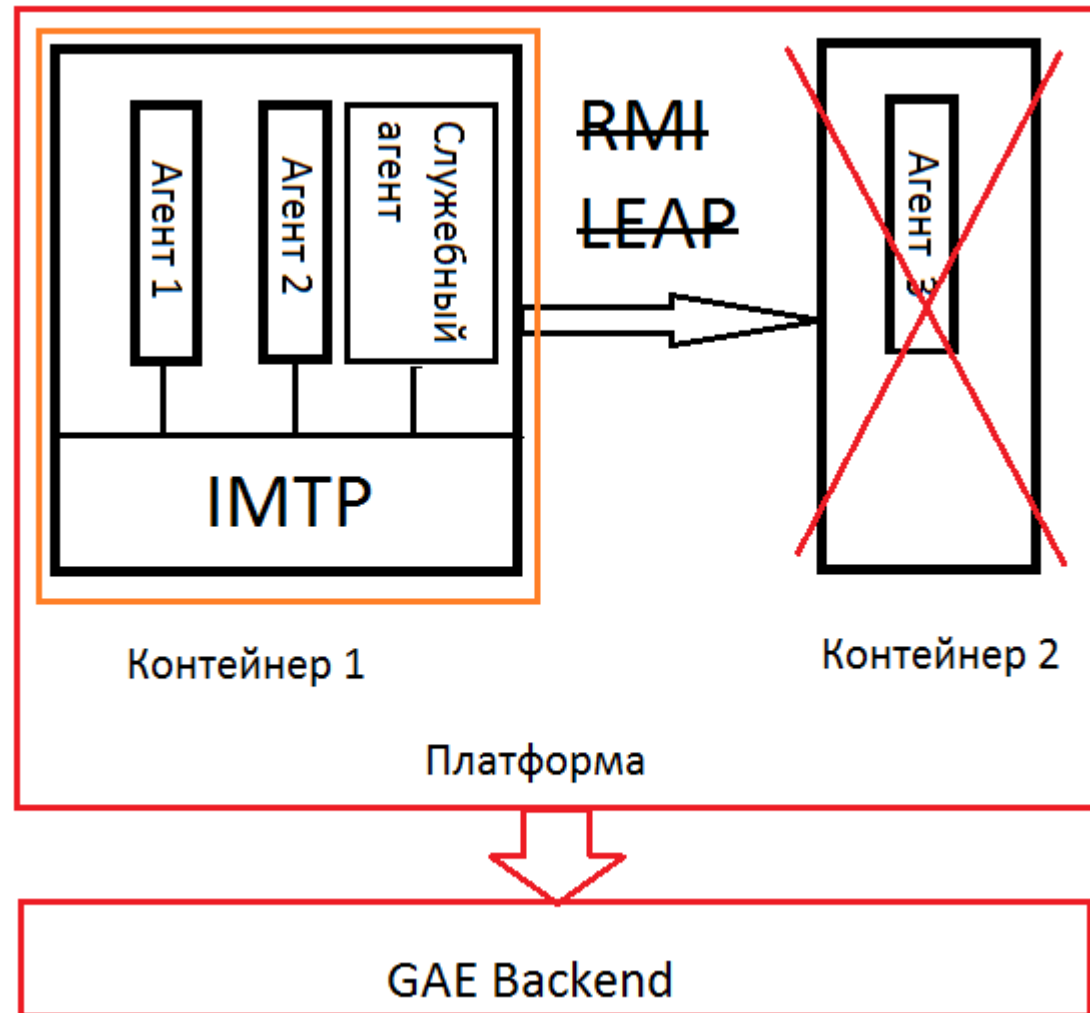


# Проблемы

- IMTP (Internal Message Transport Protocol) — сообщение внутри платформы, реализован через RMI
- Каждый агент выполняется в своем потоке
- В JADE используются классы, не поддерживаемые GAE

# Решение

- Отказ от использования нескольких контейнеров
- Переработана вертикальная цепочка обработки команд и сообщений
- Размещение в backend
- Классы только из “белого списка” GAE



# Сложности

- Отсутствие возможности отладки – тестовая среда Google App Engine не поддерживает запуск backend'ов
- Недостаточная документированность внутреннего устройства IMTP в JADE

# Демонстрационная МАС

- Адаптирована простейшая логистическая система
- 3 вида агентов:
  - агенты, выдающие задания (Operator)
  - агенты, соответствующие транспортным единицам (Vehicle) – по сравнению с оригинальной задачей усложнена работа
  - агент, создающий Operator'ов и агентов-перевозчиков (добавлен)
- Вместо использования Swing GUI – генерация задач и агентов
- Переписано пространственное позиционирование

# Результаты

- Исследована архитектура JADE
- Проанализированы наиболее популярные существующие облачные сервисы и обоснован выбор Google App Engine для осуществления интеграции с JADE
- Изучено API Google App Engine и выявлены потенциальные проблемы
- JADE перенесено в облако GAE (один контейнер на платформу)
- На базе существующей MAC реализована демонстрационная система и запущена на получившейся платформе