

## **СРАВНЕНИЕ ЭФФЕКТИВНОСТИ FOG COMPUTING С CLOUD COMPUTING НА БАЗЕ БИБЛИОТЕКИ IFOGSIM**

---

**Мухамадазиз РАСУЛМУХАМЕДОВ,**

*кандидат физика-математических наук, доцент, Ташкентский  
государственный транспортный университет, Ташкент, Узбекистан*

*E-mail: mrasulmuxamedov@list.ru*

**Фаррух ШУКУРОВ,**

*ассистент, Ташкентский государственный транспортный университет,  
Ташкент, Узбекистан*

**Шохсанам ШУКУРОВА,**

*студент магистратуры, Ташкентский государственный транспортный  
университет, Ташкент, Узбекистан*

*DOI: <https://doi.org/10.47689/978-9943-7818-0-1-pp59-65>*

**Аннотация:** В данной статье описывается пользование библиотекой java iFogSim, которая помогает имитировать в виртуальной среде fog computing. Рассматривается оптимальное построение fog computing и проводится анализ пропускной способности и вычислительной скорости в совокупности.

**Ключевые слова:** Java, Eclipse, iFogSim, Raspberry Pi, виртуальная машина, Docker, туманная среда, облачная среда, cloud computing.

Чтобы тщательно оценить этот проект, необходимо определить настройку оценки с четко определенными границами, метриками и включенными устройствами [1].

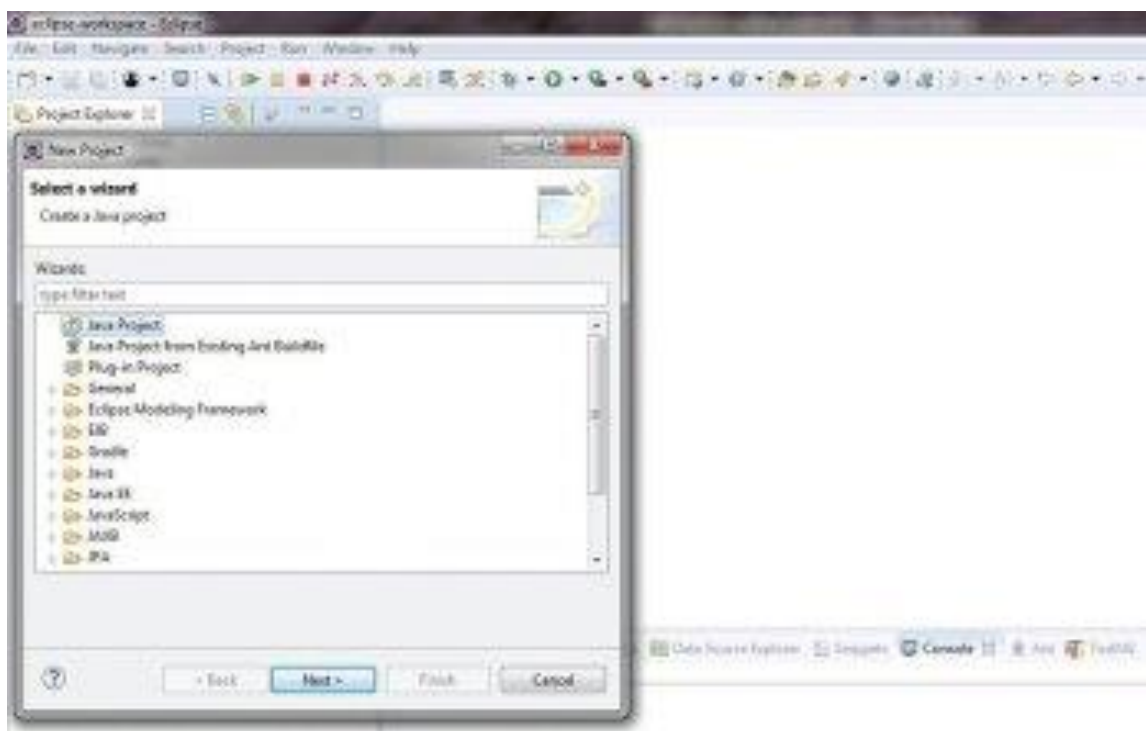
Установленная настройка оценки в этой работе представляет собой сетевую топологию, включающую программы и устройства, необходимые для адекватного анализа разработанной системы облачных вычислений. Топология состоит из четырех отдельных устройств: промежуточного программного обеспечения облако-туман, узлов управления туманом, ячеек тумана и датчиков.

Для имитации топологии будет использовано среда облачных и туманных вычислений iFogSim.

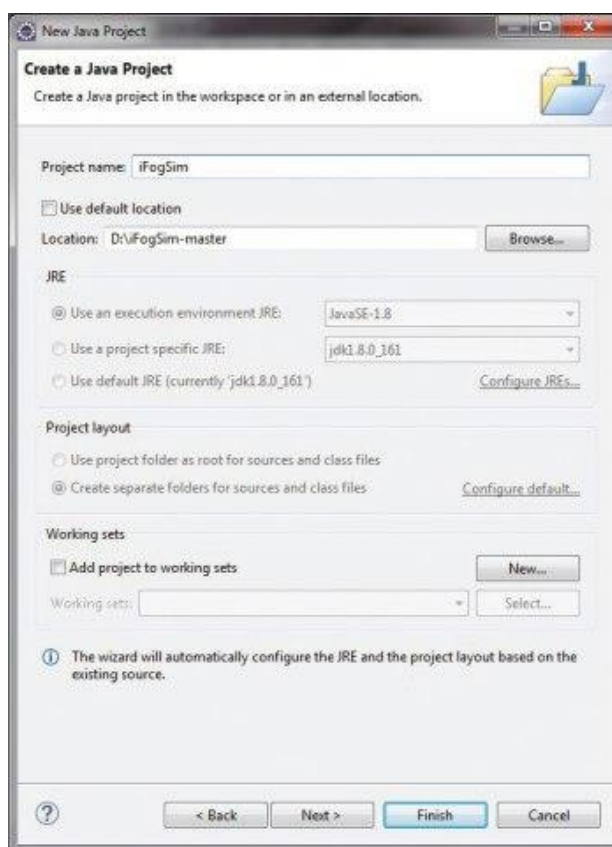
Библиотеку iFogSim можно загрузить с URL-адреса <https://github.com/Cloudslab/iFogSim>. Эта библиотека написана на Java, поэтому для настройки и работы с инструментарием потребуется Java Development Kit (JDK)[2].

После загрузки набора инструментов сжатия в формате Zip он извлекается и создается папка iFogSim-master. Библиотека iFogSim может быть выполнена в любой интегрированной среде разработки на основе Java (IDE), такой как Eclipse, Netbeans, JCreator, JDeveloper, jGRASP, BlueJ, IntelliJ IDEA или Jbuilder.

Чтобы интегрировать iFogSim в Eclipse IDE, нам нужно создать новый проект в IDE (как показано на рисунках 3.1 и 3.2).



**Рис. 1 – Создание нового проекта в Eclipse IDE**



**Рис. 2 – Интеграция iFogSim с проектом в Eclipse IDE**

Как только библиотека настроена, структуру каталогов iFogSim можно просмотреть в Eclipse IDE в Project Name -> src.

Существует множество пакетов с Java-кодом для различных реализаций туманных вычислений, IoT и граничных вычислений [3].

Для работы с iFogSim в режиме графического интерфейса пользователя существует файл с именем FogGUI.java в org.fog.gui.example. Этот файл может быть непосредственно выполнен в IDE, и в рабочую область моделирования могут быть импортированы различные компоненты облака и тумана.

В Fog Topology Creator есть меню Graph, в котором есть опция для импорта топологии.

После выполнения выходные данные можно просмотреть в консоли Eclipse IDE.

В iFogSim есть различные сценарии для нескольких приложений, которые можно моделировать; К ним относятся программно-определяемые сети (SDN) и их интеграция с облачными и туманными вычислениями [4].

В базовой установке iFogSim есть ряд примеров, которые были реализованы и запрограммированы. Например, в org.fog.test.perfeval есть пример интеллектуального наблюдения. При его выполнении можно оценить различные параметры, такие как энергия, стоимость, производительность камеры и т.д.

Starting DCNS...

Placement of operator object\_detector on device d-0 successful.

Placement of operator object\_tracker on device d-0 successful.

Creating user\_interface on device cloud

Creating object\_detector on device d-0

Creating object\_tracker on device d-0

Creating motion\_detector on device m-0-0

Creating motion\_detector on device m-0-1

Creating motion\_detector on device m-0-2

Creating motion\_detector on device m-0-3

0.0 Submitted application dcns

=====

===== RESULTS =====

=====

EXECUTION TIME : 2583

=====

APPLICATION LOOP DELAYS

=====

[motion\_detector, object\_detector, object\_tracker] ---> 5.3571428571438195

[object\_tracker, PTZ\_CONTROL] ---> 3.110000000000363

=====

TUPLE CPU EXECUTION DELAY

=====

MOTION\_VIDEO\_STREAM ---> 2.957142857143481

DETECTED\_OBJECT ---> 0.1111607142865978

OBJECT\_LOCATION ---> 1.5285714285710128

CAMERA ---> 2.1000000000000364

=====

cloud : Energy Consumed = 1.3338424452551037E7

proxy-server : Energy Consumed = 834332.9999999987

d-0 : Energy Consumed = 1048835.431000002

m-0-0 : Energy Consumed = 846301.761000042

m-0-1 : Energy Consumed = 846301.761000042

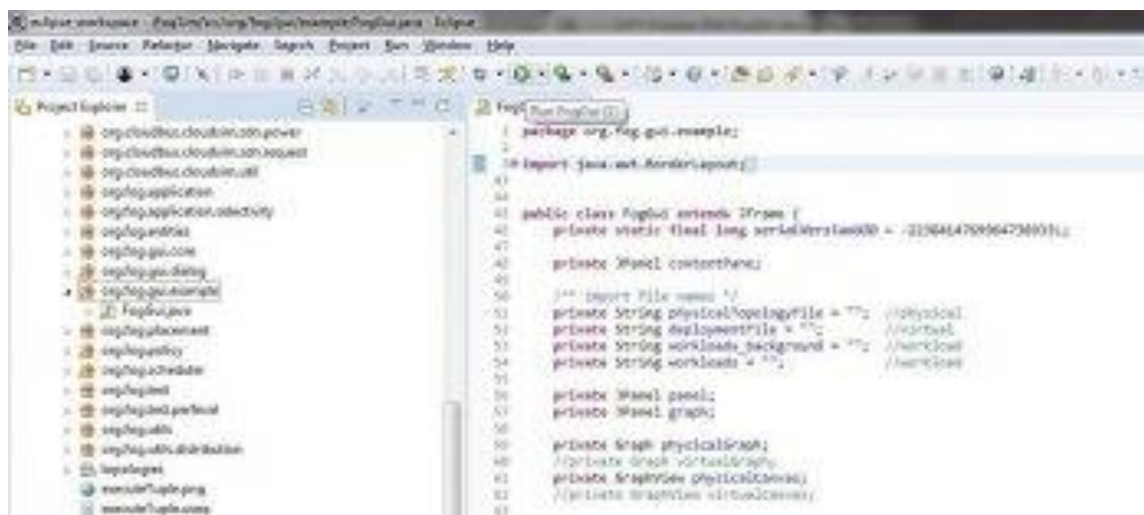
m-0-2 : Energy Consumed = 846301.761000042

m-0-3 : Energy Consumed = 846301.761000042

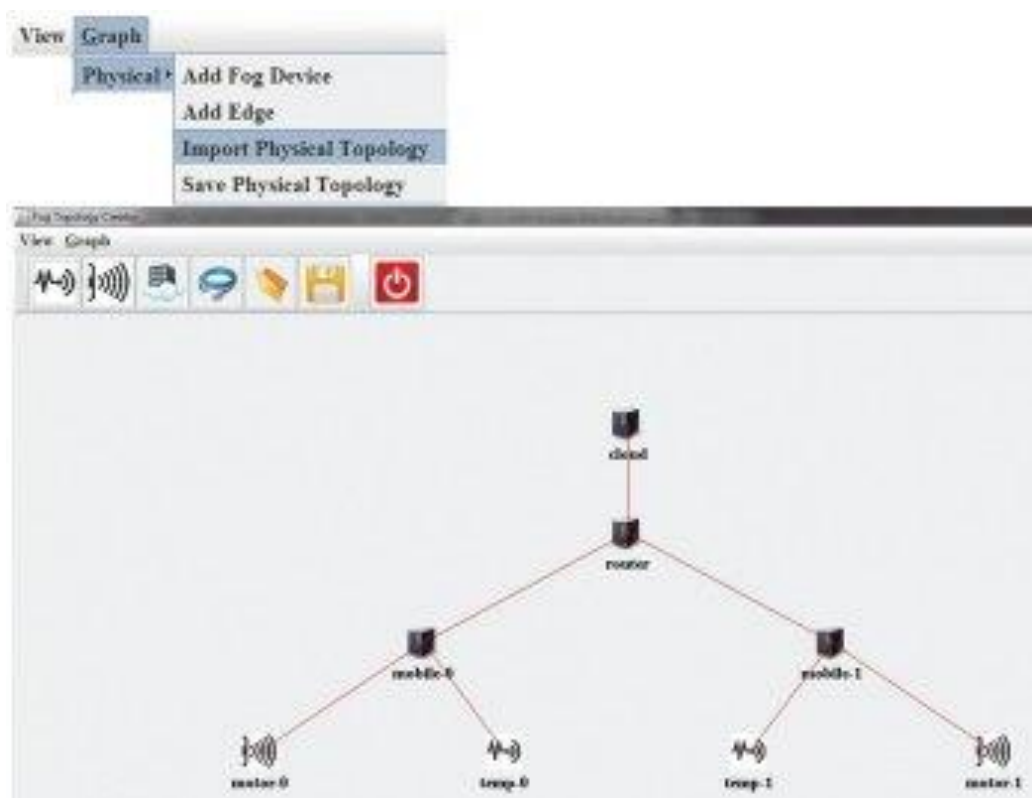
Cost of execution in cloud = 26120.742857167836

Total network usage = 11101.12

Это полностью настраиваемые библиотеки, которые могут быть улучшены с помощью новых алгоритмов. Эти новые алгоритмы могут быть запрограммированы в существующих библиотеках iFogSim, так что производительность предлагаемого или нового алгоритма можно анализировать в сетях на основе тумана с помощью iFogSim [5].

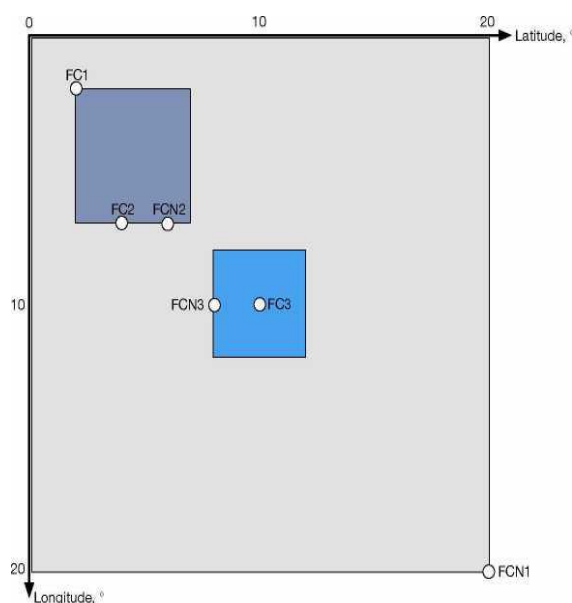


**Рис. 3 – Открытие FogGui.Java в Eclipse**



**Рис. 4 – Импорт топологии в графическом интерфейсе iFogSim**

В этой настройке подключенные устройства Интернета вещей – это сенсорные модули, состоящие из датчика температуры и влажности. Эти сенсорные модули подключаются к соответствующим Raspberry Pi3 b+ с помощью сенсорных модулей, называемых GrovePi.



**Рис. 5 – Сетка расположения устройства для оценки родительским элементом**

Статистически интерпретированные данные представлены в Таблице 1 и Таблице 2 отображающие результаты оценки. Данные вычисленной статистической оценки, используемые для анализа времени развертывания в контрастных средах, включают в себя минимум, первый квартиль, среднее значение, медиану (второй квартиль), третий квартиль, максимальное время развертывания и стандартное отклонение  $\sigma$ . Данные являются результатом пяти аналогичных исполнений приложения АЗ, выполненных в полной настройке оценки. Кроме того, данные уточняются путем расчета времени развертывания для каждого запроса, чтобы дать представление о времени относительно развернутых запросов. В Таблице 3.1 приведены те же статистические данные для времени запуска облачной виртуальной машины и времени загрузки образа Docker для дальнейшего анализа.

**Таблица 1**

**Время запуска дополнительной ВМ и загрузки образа**

Метрики	Мин.	Q1	Сред.	Медиана	Q3	Макс.	
Время запуска ВМ	36.29	37.52	39.69	38.22	40.36	46.41	3.35
Время загрузки Образа	64.32	65.68	67.05	66.61	67.96	71.00	1.95

**Таблица 2**

**Развертывание данных, оценки времени**

Метрики	Количество						
Развернутые Облачные Службы	15						
Развернутые Туманные Службы	15						
Итоговые Развернутые Службы				30			
Метрики	Мин.	Q1	Сред.	Медиана	Q3	Макс.	
Время развертывания тумана	27.56	28.09	28.87	28.52	28.99	31.20	1.26
Время развертывания Облака	257.22	258.01	262.09	262.50	265.68	267.05	3.95
Итоговое время развертывания	285.57	286.21	290.96	290.59	295.57	296.88	4.65
Время развертывания тумана к Запросам	1.84	1.87	1.92	1.90	1.93	2.08	0.08



Время развертывания облака к Запросам	17.15	17.20	17.47	17.50	17.71	17.80	0.26
Итоговое время развертывания к Запросам	9.52	9.54	9.70	9.69	9.85	9.90	0.15

Подводя итоги этого сценария, можно сделать вывод, что время развертывания в облаке намного выше, чем время развертывания в тумане. Конечно, необходимо вести учет времени запуска виртуальной машины и времени загрузки образа Docker, и этот факт также можно рассматривать как недостаток облачной среды. Результат этого сценария демонстрирует явные преимущества развертывания услуг Интернета вещей в туманной среде по сравнению с облачной.

#### **БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ:**

1. Guliamova M.K., & Aliev R.M. (2021). Database Concept, Relevance and Expert Systems. Scientific and Educational Areas Under Modern Challenges, 2021, – PP. 125–127. Чебоксары: SCC “Interaktiv plus”.
2. Tokhirov E., Aliev R. Improving the braking distance of the train before level crossing // InterConf. – 2020.
3. Gulamov J., Shukurov F., Shokhidayeva Sh. Automated information system for inventory accounting of a car depot warehouse // JournalNX – 2021.
4. Nurmukhamedov T., Shukurov F., Khakimov Sh. Information technologies for forecasting car depot inventory using correlation analysis // JournalNX-2021.
5. R. Aliev Analysis of controlling the state of track sections on lines with speed and high-speed train traffic German International Journal of Modern Science № 14, – 2021. – PP. 57–58.