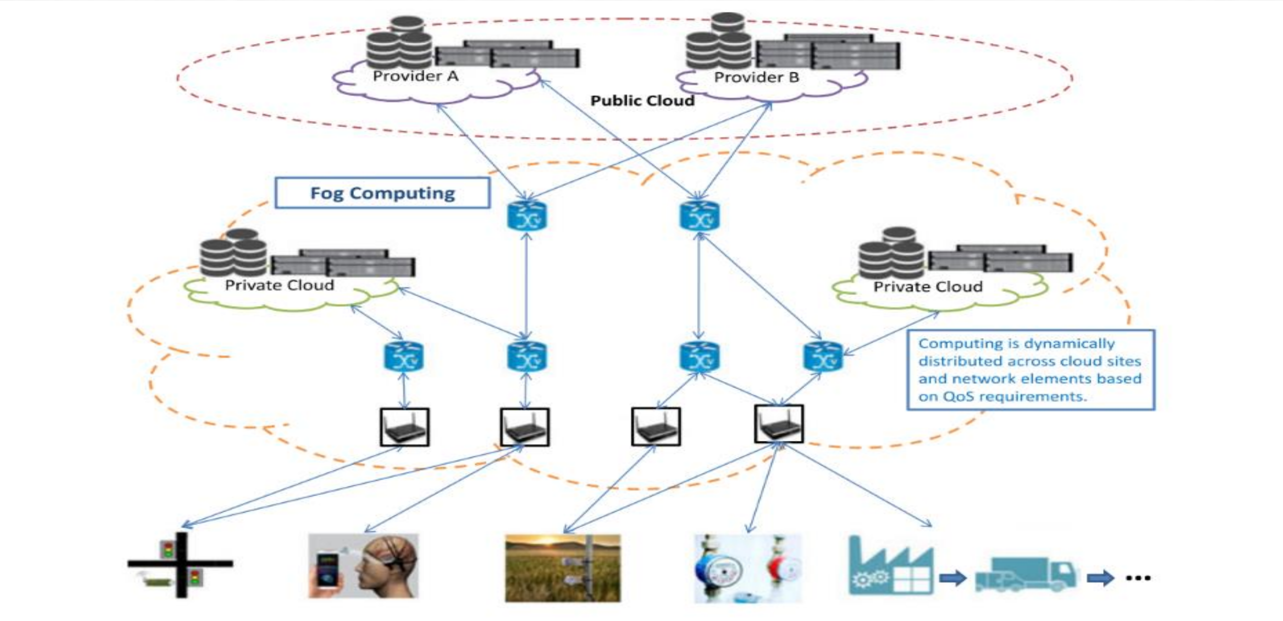
PLACEMENT POLICY IN FOG COMPUTING

Team Details
1. A. Sai Manideep Reddy (20EG105402)
2. B. Harshitha (20EG105405)
3. K. Adithya (20EG105417)

Project Supervisor
Dr. Pallam Ravi
Assistant Professor

Introduction

- Fog Computing is distributed computing paradigm that extents the services provided by the cloud to the edge of network.



Introduction

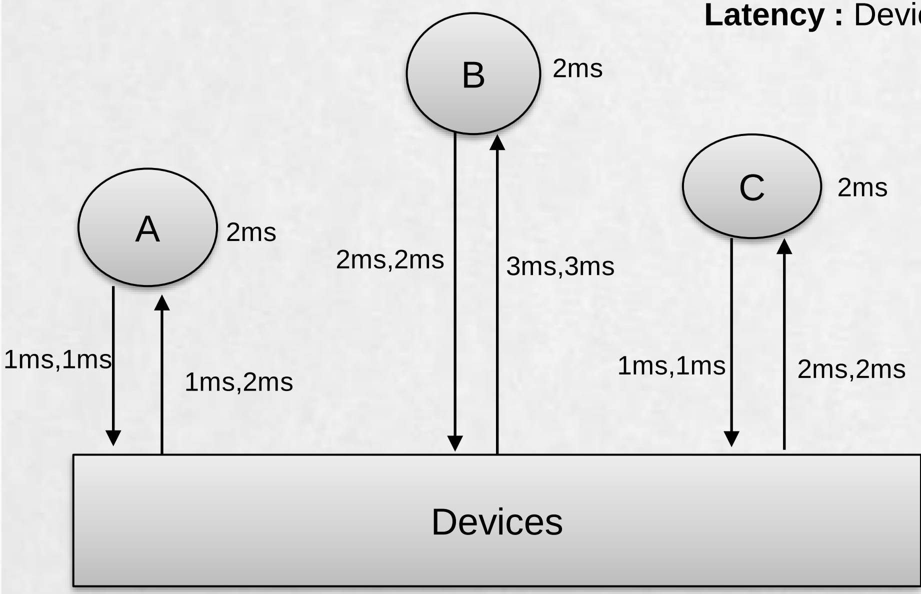
- A placement policy in fog computing refers to a set of rules, algorithms, and strategies used to determine where to deploy and allocate applications, services, and resources within a fog network.
- iFogSim is a popular simulation framework designed specifically for modeling and simulating fog computing environments.
- The requirements to simulate a placement policy using iFogSim:
  - Java Development Kit (JDK) installed (required for running Java-based simulations)
  - Eclipse IDE
  - iFogSim framework
- Applications of Fog Computing are **Smart Cities, Smart Grids, Industrial IoT ,Smart Appliances etc,.**

Problem Statement

Implementation of Placement Policy for minimizing the Energy Consumption in Fog Computing Environments.

Illustration of Problem:

Latency : Device to Node time+ Processing Time of node + Node to Device time



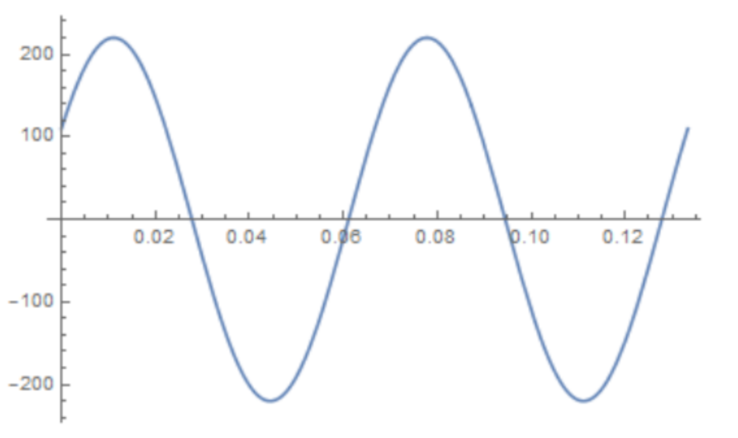
Strategy	Distribution	Placement	Latency Sum
1	$(1d_1, 3d_2)$	A: $\{d_1,d_2\}$, B: $\{d_2\}$, C: $\{d_2\}$	21
2	$(1d_1, 3d_2)$	A: $\{d_2\}$, B: $\{d_1,d_2\}$, C: $\{d_2\}$	26
3	$(2d_1, 2d_2)$	A: $\{d_1\}$, B: $\{d_1,d_2\}$, C: $\{d_2\}$	23
4	$(2d_1, 2d_2)$	A: $\{d_2\}$, B: $\{d_1,d_2\}$, C: $\{d_1\}$	25

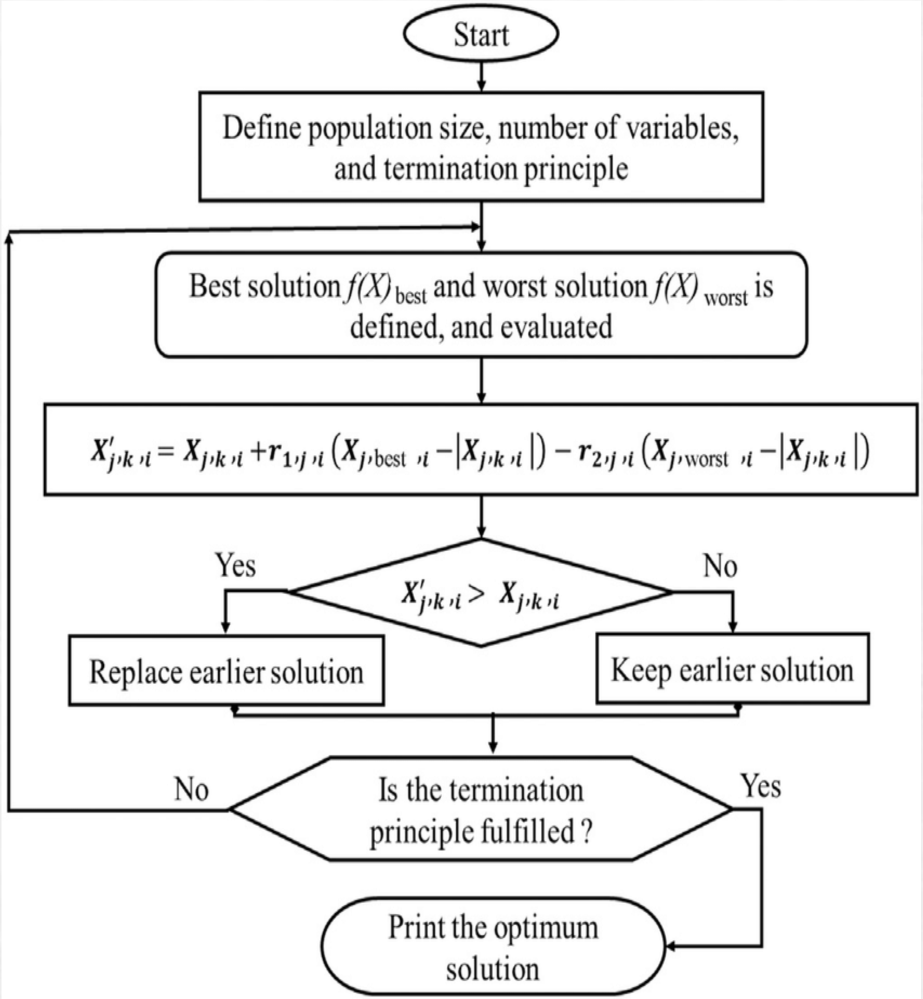
Proposed Method

The proposed method - **JAYA algorithm**

- JAYA algorithm is an optimization algorithm, which can employed to optimize the resource allocation, task scheduling , energy efficiency, and latency minimization.
- Jaya algorithm generally works in the following steps:

a. Problem formation
b. Fitness evaluation
c. Identify the best and worst solutions
d. Update the candidate solutions
e. Termination criteria
f. Repeat or Output




Start
Define population size, number of variables, and termination principle
Best solution $f(X)_{\text{best}}$ and worst solution $f(X)_{\text{worst}}$ is defined, and evaluated
$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,\text{best},i} - |X_{j,k,i}|\right) - r_{2,j,i}\left(X_{j,\text{worst},i} - |X_{j,k,i}|\right)$
Yes
$X'_{j,k,i} > X_{j,k,i}$
No
Replace earlier solution
Keep earlier solution
No
Is the termination principle fulfilled ?
Yes
Print the optimum solution

Proposed Method

Illustration :
Minimize(x_1+x_2)
Where -100<= x_1 <=100
-100<= x_2 <=100

Step 1: Initialize population

$$\begin{bmatrix} 20 & -10 \\ 58 & -28 \\ -22 & 29 \\ -1 & 2 \end{bmatrix}$$

Step 2: Evaluation of fitness values

Population	x_1	x_2	F(x)
1	20	-10	10
2	58	-28	30
3	-22	29	7
4	-1	2	1

Step 3: Identify the best and worst solutions

Population	x_1	x_2	F(x)	
1	20	-10	10	
2	58	-28	30	Worst
3	-22	29	7	
4	-1	2	1	Best

Step 4: Modify and update the candidate solution

$$\mathbf{X_{new}}=X_{j,k}+r_1(X_{best}-|X_{j,k}|)-r_2(X_{worst}-|X_{j,k}|)$$

58	-28	Worst
-1	2	Best

x_1	
r_1	r_2
0.0348	0.9307

x_2	
r_1	r_2
0.9045	0.5900

$$\begin{bmatrix} 20 & -10 \\ 58 & -28 \\ -22 & 29 \\ -1 & 2 \end{bmatrix}$$

$$\mathbf{X_{11}}=X_{1,1}+r_1(X_{best}-|X_{1,1}|)-r_2(X_{worst}-|X_{1,1}|)$$

$\mathbf{X_{11}}$=20+(0.0348(-1-|20|))-(0.9307(58-|20|))
= -16.0974

$\mathbf{X_{12}}$=-10+(0.9045(2-|-10|))-(0.5900(-28-|-10|))
= 5.184

$\mathbf{X_{21}}$=58+(0.0348(-1-|58|))-(0.9307(58-|58|))
= 55.9468

$\mathbf{X_{22}}$=-28+(0.9045(2-|-28|))-(0.5900(-28-|-28|))
= -18.477

$\mathbf{X_{31}}$=-22+(0.0348(-1-|-22|))-(0.9307(58-|-22|))
= -56.3056

$\mathbf{X_{32}}$=29+(0.9045(2-|29|))-(0.5900(-28-|29|))
= 38.2085

$\mathbf{X_{21}}$=-1+(0.0348(-1-|-1|))-(0.9307(58-|-1|))
= -54.1195

$\mathbf{X_{42}}$=2+(0.9045(2-|2|))-(0.5900(-28-|2|))
= 19.7

Population	x_1	x_2	F(x)
1	-16.0974	5.184	-10.9134
2	55.9468	-18.477	37.4698
3	-56.3096	38.2085	-18.1011
4	-54.1195	19.7	-34.4195

$F(X_{new})<F(X_{old})$ [Minimization]
Then update the solution otherwise preserve the old one .

Population	x_1	x_2	F(x)
1	20	-10	10
2	58	-28	30
3	-22	29	7
4	-1	2	1

Population	x_1	x_2	F(x)
1	-16.0974	5.184	-10.9134
2	58	-28	30
3	-56.3096	38.2085	-18.1011
4	-54.1195	19.7	-34.4195

Step 5: Termination criteria

Step 6: Output

Experiment Environment

- **Eclipse IDE:** Using this, we created a Java project with the iFogSim simulation framework.
- iFogSim is a well-known simulation framework designed specifically for modeling and simulating fog computing environments with a variety of applications.
- We used them to test the **VRgame.**

Experiment Screenshots

Cloudonly

```
============== RESULTS ==================
=========================================
EXECUTION TIME : 1020
=========================================
APPLICATION LOOP DELAYS
=========================================
[EEG, client, concentration_calculator, client, DISPLAY] ---> 226.43839296697556
=========================================
TUPLE CPU EXECUTION DELAY
=========================================
PLAYER_GAME_STATE ---> 0.3233442034056271
EEG ---> 3.7822568139261348
CONCENTRATION ---> 0.1359389632856112
_SENSOR ---> 0.6266152696653765
GLOBAL_GAME_STATE ---> 0.05600000000004002
=========================================
cloud : Energy Consumed = 3240139.906928527
proxy-server : Energy Consumed = 166866.59999999995
d-0 : Energy Consumed = 166866.59999999995
m-0-0 : Energy Consumed = 174789.72099999883
m-0-1 : Energy Consumed = 174780.11298874978
m-0-2 : Energy Consumed = 174774.84801999945
m-0-3 : Energy Consumed = 174566.72555499975
m-0-4 : Energy Consumed = 174646.03157249963
d-1 : Energy Consumed = 166866.59999999995
m-1-0 : Energy Consumed = 174524.02299999932
m-1-1 : Energy Consumed = 174789.72099999903
m-1-2 : Energy Consumed = 174661.82965999996
m-1-3 : Energy Consumed = 174596.54531999966
m-1-4 : Energy Consumed = 174789.72099999964
Cost of execution in cloud = 816805.9440000204
Total network usage = 196413.5
```

Experiment Screenshots

PSO

```
============== RESULTS ==================
=========================================
EXECUTION TIME : 818
=========================================
APPLICATION LOOP DELAYS
=========================================
[EEG, client, concentration_calculator, client, DISPLAY] ---> 226.44719232321253
=========================================
TUPLE CPU EXECUTION DELAY
=========================================
PLAYER_GAME_STATE ---> 0.45624987312607396
EEG ---> 3.896957507853503
CONCENTRATION ---> 0.16036522806585224
_SENSOR ---> 0.6202098891406491
GLOBAL_GAME_STATE ---> 0.05600000000004002
=========================================
cloud : Energy Consumed = 3238338.565857097
proxy-server : Energy Consumed = 166866.59999999995
d-0 : Energy Consumed = 166866.59999999995
m-0-0 : Energy Consumed = 174789.72099999883
m-0-1 : Energy Consumed = 174742.19821500024
m-0-2 : Energy Consumed = 174731.24707999948
m-0-3 : Energy Consumed = 174721.92474499985
m-0-4 : Energy Consumed = 174568.96897249983
d-1 : Energy Consumed = 166866.59999999995
m-1-0 : Energy Consumed = 174300.26659999983
m-1-1 : Energy Consumed = 174789.7209999993
m-1-2 : Energy Consumed = 174703.7508999998
m-1-3 : Energy Consumed = 174601.92036
m-1-4 : Energy Consumed = 174789.72099999967
Cost of execution in cloud = 814252.144000019
Total network usage = 197512.0
```

Experiment Screenshots

EPSO

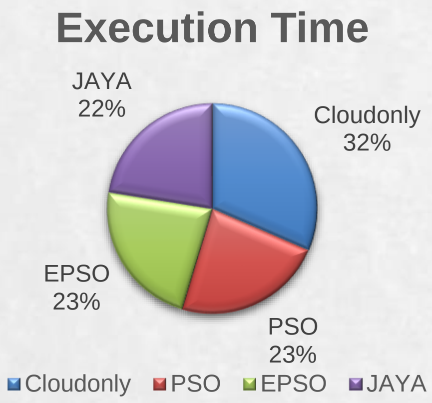
```
============== RESULTS ==================
=========================================
EXECUTION TIME : 830
=========================================
APPLICATION LOOP DELAYS
=========================================
[EEG, client, concentration_calculator, client, DISPLAY] ---> 226.45704059588414
=========================================
TUPLE CPU EXECUTION DELAY
=========================================
PLAYER_GAME_STATE ---> 0.4561200824329859
EEG ---> 3.7356353144139143
CONCENTRATION ---> 0.14426731207327162
_SENSOR ---> 0.5978524978212147
GLOBAL_GAME_STATE ---> 0.05600000000004002
=========================================
cloud : Energy Consumed = 3233950.5785713815
proxy-server : Energy Consumed = 166866.59999999995
d-0 : Energy Consumed = 166866.59999999995
m-0-0 : Energy Consumed = 174789.72099999888
m-0-1 : Energy Consumed = 174784.6310000001
m-0-2 : Energy Consumed = 174789.72099999952
m-0-3 : Energy Consumed = 174606.88438249956
m-0-4 : Energy Consumed = 174588.54383999977
d-1 : Energy Consumed = 166866.59999999995
m-1-0 : Energy Consumed = 174520.2360399997
m-1-1 : Energy Consumed = 174784.63099999918
m-1-2 : Energy Consumed = 174743.53433999993
m-1-3 : Energy Consumed = 174588.82887999978
m-1-4 : Energy Consumed = 174789.7209999996
Cost of execution in cloud = 808031.2000000183
Total network usage = 198166.0
```

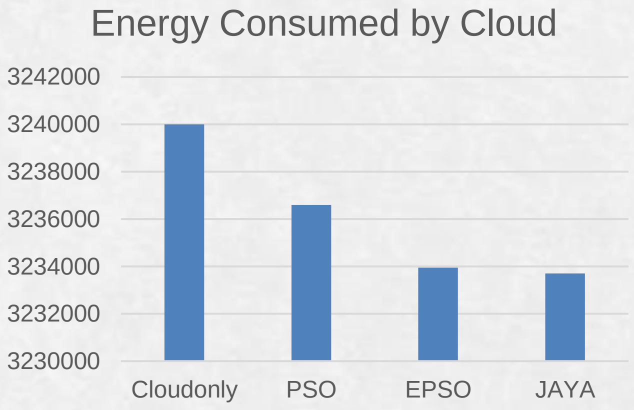
Experiment Screenshots

JAYA

```
============== RESULTS ==================
=========================================
EXECUTION TIME : 819
=========================================
APPLICATION LOOP DELAYS
=========================================
[EEG, client, concentration_calculator, client, DISPLAY] ---> 226.43568782320258
=========================================
TUPLE CPU EXECUTION DELAY
=========================================
PLAYER_GAME_STATE ---> 0.3232144601003516
EEG ---> 3.6861300678684223
CONCENTRATION ---> 0.146098503629733
_SENSOR ---> 0.5991628448447904
GLOBAL_GAME_STATE ---> 0.05600000000004002
=========================================
cloud : Energy Consumed = 3233713.3979999577
proxy-server : Energy Consumed = 166866.59999999995
d-0 : Energy Consumed = 166866.59999999995
m-0-0 : Energy Consumed = 174784.63099999877
m-0-1 : Energy Consumed = 174771.02224874997
m-0-2 : Energy Consumed = 174760.3720599992
m-0-3 : Energy Consumed = 174668.96202249944
m-0-4 : Energy Consumed = 174630.68903999997
d-1 : Energy Consumed = 166866.59999999995
m-1-0 : Energy Consumed = 174537.55221999952
m-1-1 : Energy Consumed = 174789.7209999992
m-1-2 : Energy Consumed = 174703.09937999968
m-1-3 : Energy Consumed = 174658.25647999992
m-1-4 : Energy Consumed = 174789.72099999932
Cost of execution in cloud = 807694.9440000197
Total network usage = 196404.5
```

Experiment Results





Findings

We have discovered that we can minimize latency and lower energy usage by implementing adjustments to the placement policies in the fog computing environment.

Justification

Parameters
Low latency and Energy consumption

Formula
Latency : Device to Node time+ Processing Time of node + Node to Device time
Total Energy Consumed = Energy consumed by Module $_1$+ Energy consumed by Module $_2$+….+ Energy consumed by Module $_N$

In what way the parameters are improved
PSO (parameters) : Inertia weight , Social and Cognitive components.
EPSO (parameters) : Local search mechanism and Inertia weight.
JAYA (parameters) : No specific parameters apart from common parameters like no. of iterations, Population size.