Efficient Resource Provisioning for Smart Buildings Utilizing Fog and Cloud Based Environment

Anila Yasmeen, Nadeem Javaid, Obaid Ur Rehman, Hina Iftikhar, Muhammad Faizan Malik, and Fatima J. Muhammad

2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)
Smart Grid

IoT is used for collecting information from consumer and producer

- Energy Saving
- Sustainability
- Reliability
Integration of Smart Grid with Cloud Computing
Contribution

• A three layered model based on cloud and fog framework is proposed. These layers consist of core cloud layer, fog layer and end user layer.

• A new Hybrid Particle Swarm Optimization with Simulated Annealing (PSO-SA) is proposed to find an optimum allocation of tasks to the available VMs.
System Model

Power generation unit
Energy storage unit
System Workflow

Calculate the optimal scheduling of power consumption

Information of generation and consumption

Load of each device

Scheduling decision
Lack of Power Generation From MG
Problem:

- **Goal:** Minimizing the response time, cost
- VMs are allocated on basis of storage and memory requirement to host for balancing on fog network
- Servers are responsible to manage all hosts according to policy and may be more than one VMs are assigned to one host.
- Load balancing is the distribution of workload for multiple links to avoid overloading and to achieve minimum response time.
Algorithm 1 PSO-SA based resource allocation

1: Input: List of tasks, List of the VMs
2: Initialization: PopSize, Pvelocity, maxIter, Particle=tasks;
3: Calculate the load, capacity of VM
4: for $i=1$ to $PopSize$ do
5: \hspace{1em} Particle[i].best=current position
6: \hspace{1em} Particle[i].bestfitness=current fitness
7: end for
8: Calculate $Pbest$ and $Gbest$ for each machine
9: $Gbest= Particle$.best with lowest fitness
10: for $j=1$ to $maxIter$ do
11: \hspace{1em} for $t=1$ to $Particle$ do
12: \hspace{2em} InertiaValue ();
13: \hspace{2em} Update $Pvelocity$ ();
14: \hspace{2em} Update Position ();
15: \hspace{2em} if current fitness $\neq$ Particle[t].bestfitness then
16: \hspace{3em} Particle[t].best=current position
17: \hspace{3em} Particle[t].bestfitness=current fitness
18: \hspace{2em} end if
19: end for
20: $Gbest= Particle$.best with lowest fitness
21: end for
22: Return $Gbest$
Particle Swarm Optimization
Simulated Annealing
Simulation Setup

- Distributed fog framework and centralized cloud
- Exist 3 region. Each region includes 100 house
- Run 24 hours
- Loading balancing algorithm: PSO-SA, RR, and throttled

Metric

- Response time
- Process time
- Cost
Baseline Algorithm

- **RR**: Allocate resources to each host by equal time slicing for utilization of resources.
- **Throttled**: Allocate the first VM that is available in the table.
Request per hour of the algorithms
Response time of the algorithms

(a) Building 1

(b) Building 2

(c) Building 3
Processing time of the algorithm

(a) Fog 1
(b) Fog 2
(c) Fog 3
Overall cost of the algorithms
Conclusion