## **Proposed Model**

The proposed algorithm of efficiently scheduling of tasks and the virtual machines in the virtualized cloud environment can be broadly divided into two phases, which has been described below.

1. **First Phase Of Allocation**

In this phase, the task are scheduled to VM using **Least Loss Energy Density Algorithm.** This algorithm attempts to allocate task to their favourite core to optimise the energy consumption of the system.

The **energy density is calculated by** of a task Ti on a VM . The energy density of a task gives its average energy consumption per unit time on the respective VM type. This value does not provide any global perspective on how the power consumption of the system changes when a certain task is not allocated to its preferred VM type. The global prospective can be achieved through a metric termed as **density difference** (DD). The density difference can be determined by subtracting the energy density of a task on the current core type from the next higher energy density value of the same task on another core i.e. **DD =**  **-** . It defines how much extra energy will be consumed, if the task is allocated to the next higher energy consumption core instead of its current preferred VM. The DD values are stored in a matrix called MT ( value in a matrix MT corresponds to the DD value of on a VM ). The intuition behind such a mechanism is to reduce the losses by allocating the tasks with higher energy density difference first. The process can be started from any VM type. If any of the task in the order cannot be allocated to, the algorithm moves to the next VM instead of checking the next tasks in the order. This action is performed to avoid allocation of any unfavourable task to the current VM type.

**FLOWCHART OF LLED:**

Task Arrival

Creation of VMs

Calculate Energy Density (ED) of every task on very VM (**ED=Average energy consumption of a VM / Deadline of each task)**

Calculate Density Difference for each task (**DD =**  **-**)

Check if it consumes more energy in that VM

Task with the highest DD values are allocated in the VM

Allocate the task in that VM

Yes

No

Deallocate the task in that VM

Check if any other VM available with Min DD value

No

Yes

Allocate the task in the next VM

1. **Second Phase Of Allocation**

In this phase, VM are scheduled to the best Host using **Find Host Algorithm.** This algorithm attempts to allocate VM to their favourite Host based on certain parameters.

The purpose of this algorithm is to reduce the energy consumption of the host. FHA has been designed to work well with heterogeneous networks, containing groups of hosts working with different transmission rates. Before presenting the FHA, lets describe variables and functions used by it. HostUtilization returns the host’s CPU usage percentage. It can be calculated by **HostUtilization=**.HostEnergyEfficiency returns the energy consumption of the Host which is calculated by **HostEnergyEfficiency (EH) =Emin** +, where is constant. At the beginning, the **HostUtilization** is calculated and then we find the **HostEnergyEfficiency**. VM is allocated to host with minimum **EH** value. If HostEnergyEfficiency is same for two or more host, then the utilization level of the host is considered; the host with maximum utilization is considered the best host for VM allocation. Hence, we get an energy-aware VM scheduling method through this algorithm.

**Flowchart of FHA:**

Calculate HostEnergyEfficiency (EH) of every host.

**HostEnergyEfficiency (EH) =Emin** +

Create Host List

Calculate HostUtilization**=**

Check if **EH** is same for two or more hosts

No Yes

Check the Utilisation level of the hosts; the host with the highest utilisation level is considered as the best host

Consider the Host with minimum **EH value**

The best host is selected for allocation of the VM