LB-MAP: LOAD-BALANCED MIDDLEBOX ASSIGNMENT IN POLICY-DRIVEN DATA CENTERS

MANAR ALQARNI
DEPARTMENT OF COMPUTER SCIENCE
CALIFORNIA STATE UNIVERSITY DOMINGUEZ HILLS

INTRODUCTION

- Middleboxes “network appliances” or “network functions (NFs)” are intermediary computer networking Devices.

-NFV is a network virtualization technology that virtualizes middleboxes (or network functions) into building blocks that create communication services.

DATA CENTER TOPOLOGY

- Fat tree networks.

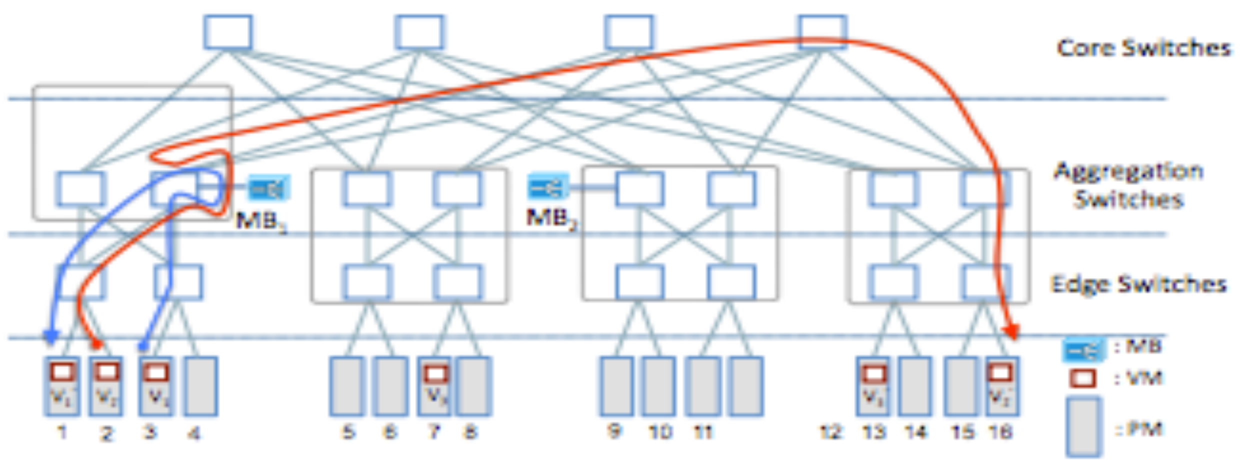

Fig. 1. A k-ary fat tree with $k = 4$ and 16 physical machines (PMs). There are two communicating VM pairs: (v_1, v_1') and (v_2, v_2'), and two middlebox instances MB_1 and MB_2. The capacity of each MB $\kappa = 2$. The minimum communication takes place as follows: (v_1, v_1') traverses MB_1 (colored blue, with cost of 3) while (v_2, v_2') traverses MB_1 too (colored red, with cost of 5), resulting in minimum total cost of 8 under uniform energy model.

DATA CENTER TOPOLOGY

- A k-ary fat-tree with k = 4, where k is the number of ports of each switch.

Core switches handles huge amount of traffic across the entire data center, therefore consuming lots of energy power.

Aggregate switches and edge switches transmit less amount of traffic therefore consume less power.

The lower two layers are separated into k pods.

each containing k/2 aggregation switches and k/2 edge switches

There are k^2/ 4 k-port core switches

LOAD BALANCED MIDDLEBOX ASSIGNMENT PROBLEM (LB-MAP)

Network Model: We model a data center as an undirected general graph G(V, E). $V = V_p \cup V_s$ includes the set of PMs

Middlebox Model: Among all the network devices in data center, load balancers have the highest failure probability.

-This is due to high number of software faults and hardware faults related to application-specific integrated circuit (ASIC) and memory.

- mb_j ($1 \leq j \leq m$) is located at switch $sw(j) \in V_s$, it must traverse one of the instances.

LOAD BALANCED MIDDLEBOX ASSIGNMENT PROBLEM (LB-MAP)

Energy Model:

-We use re, ra, and rc to denote the power consumption, when it transmits a VM communication.

***Uniform Energy Model*:**

the energy consumption of VM communication is measured as the minimum number of switches it traverses.

Skewed Energy Model: **The core switches handle more traffic therefore usually consume more energy power than aggregate switches, which consume more energy power than edge switches.**

EXAMPLE 1

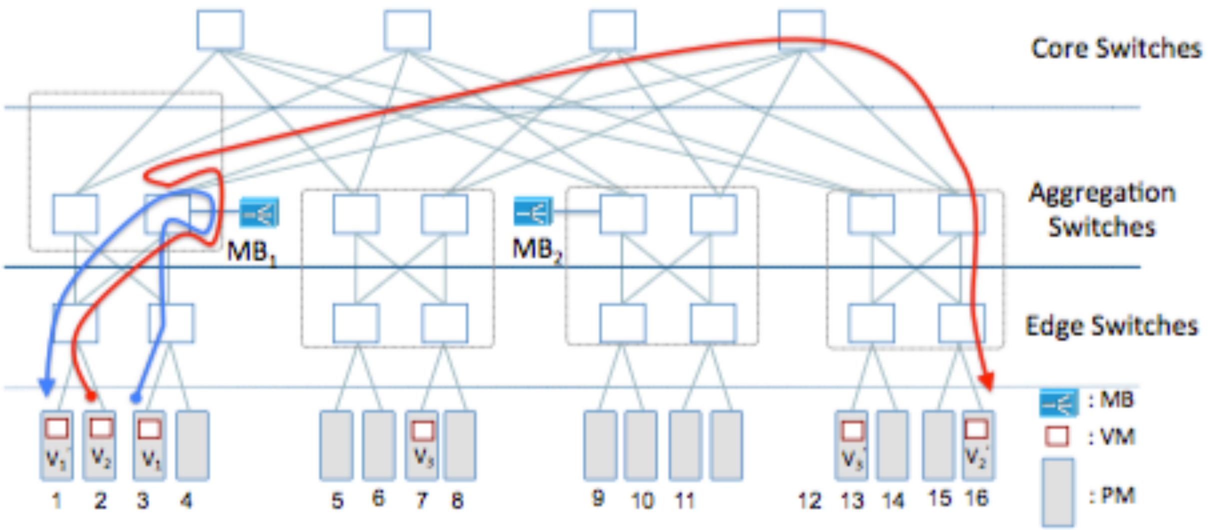

Fig. 1. A k-ary fat tree with $k = 4$ and 16 physical machines (PMs). There are two communicating VM pairs: $(v_1, v_1^{'})$ and $(v_2, v_2^{'})$, and two middlebox instances MB_1 and MB_2. The capacity of each MB $\kappa = 2$. The minimum communication takes place as follows: $(v_1, v_1^{'})$ traverses MB_1 (colored blue, with cost of 3) while $(v_2, v_2^{'})$ traverses MB_1 too (colored red, with cost of 5), resulting in minimum total communication cost of 8.

EXAMPLE 2

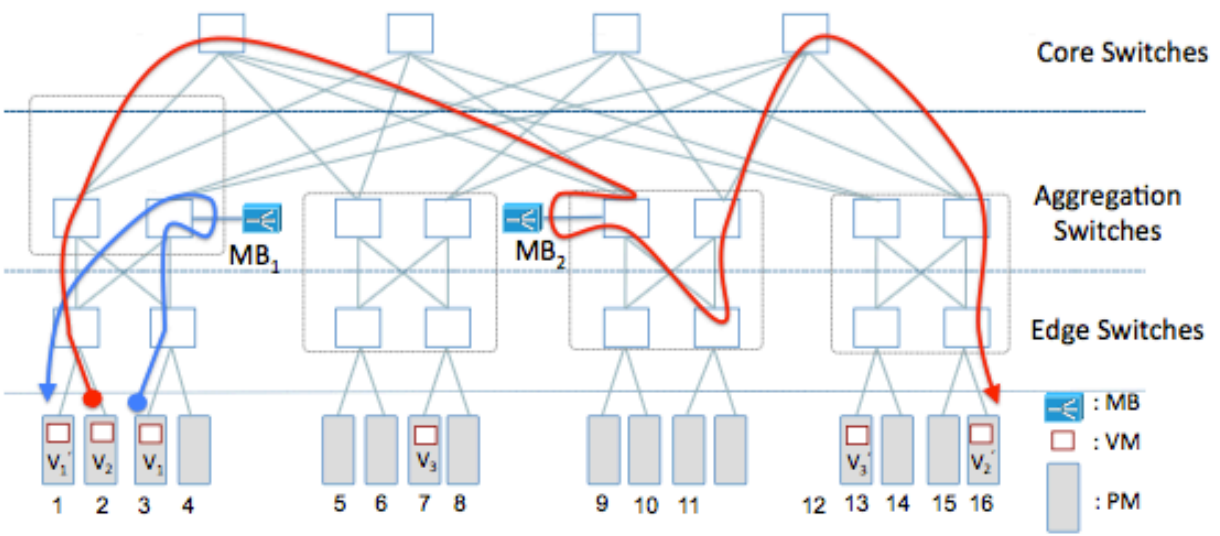

Fig. 2. Load-Balanced VM communication with each middlebox can accommodate one VM pair. The capacity of each MB $\kappa = 1$. The minimum energy communication is then: $(v_1, v_1^{'})$ traverses MB_1 (shown in blue color, with cost of 3) while $(v_2, v_2^{'})$ traversing MB_2 (shown in red color, with cost of 8), resulting in total communication cost of 11.

PROBLEM FORMULATION OF LB-MAP

-Let c(i, j) denote the minimum energy consumption between PM (or switch) i and j.

-Let $c_{i,j}$ be the minimum power consumption for VM pair (v_i, v_i') when it is assigned to middlebox instance mb_j

$$c_{i,j} = c(S(v_i), sw(j)) + c(sw(j), S(v_i')).$$

PROBLEM FORMULATION OF LB-MAP

-Now we define the load balanced middlebox assignment function as $p : P \rightarrow M$, signifying that VM pair (v_i, v_i') ∈ P is assigned to middlebox instance $p(i) \in M$. Given any middlebox assignment function p, the power consumption for VM pair (v_i, v_i') is then

$$c_{i,\mathrm{p(i)}} = c(S(v_i), sw(p(i))) + c(sw(p(i)), S(v_i')).$$

PROBLEM FORMULATION OF LB-MAP

- Denote the total energy consumption of all the l VM pairs with middlebox assignment p as C^p. Then

$$C^p = \sum_{i=1}^{l} c_{i,p(i)} = \sum_{i=1}^{l} \left(c\left(S(v_i), sw(p(i)) \right) + c\left(sw(p(i)), S(v_i') \right) \right)$$

MINIMUM COST FLOW PROBLEM (MCF)

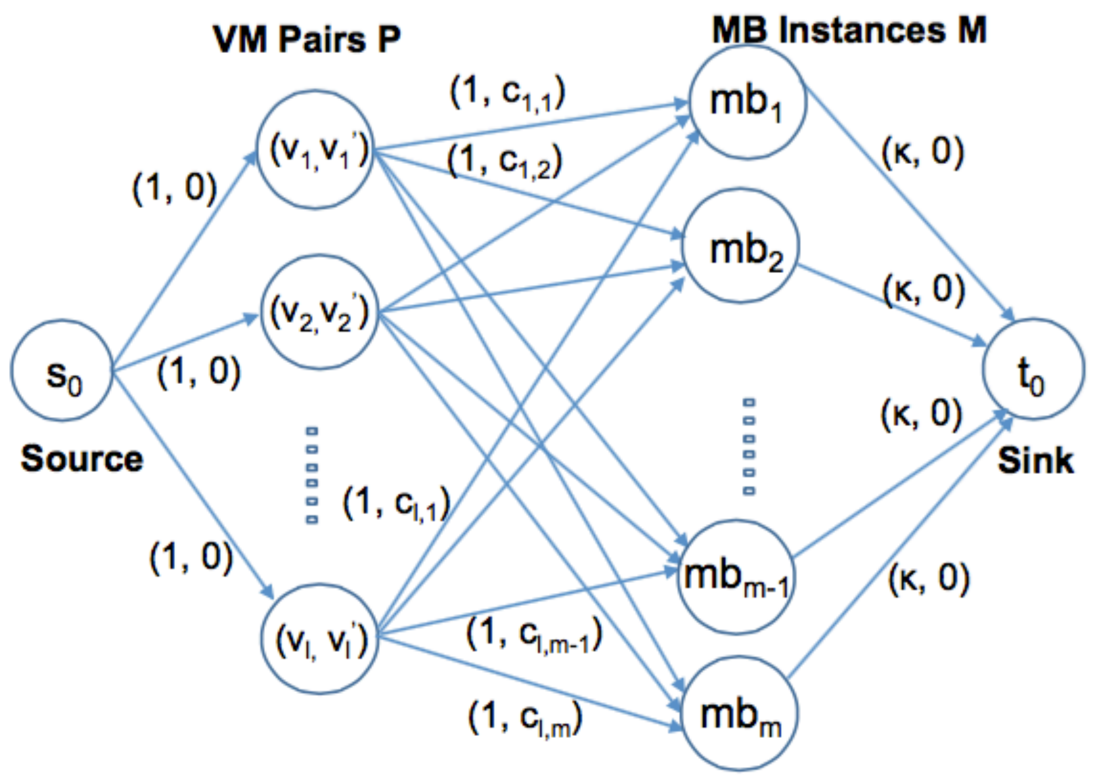

Fig. 3. LB-MAP is equivalent to minimum cost flow problem. In each parenthesis on the edge, the first value is the capacity of the edge and the second the cost of the edge.

MINIMUM COST FLOW PROBLEM (MCF)

- **It can be solved efficiently by many combinatorial algorithms.**

- **For any flow network, the algorithm has <u>the time complexity</u> of O(a^ 2· b · log(a · c)), where a, b, and c are the number of nodes, number of edges, and maximum edge capacity in the flow network.**

VM-BASED ALGORITHM

VM-Based Algorithm:

For each VM pair, it is assigned to an MB instance such that it gives the minimum energy consumption for this VM pair among all the MB instances, while satisfying this MB instance's capacity.

Algorithm 1: VM-Based Algorithm.

Input: A data center $G(V, E)$ with l VM pairs and m MB

Output: Total energy cost C for all the l VM pairs.

Notations:

i: the index for VM pairs

j: the index for middlebox instances

$load(j) = 0$: the current load of mb_j

c^i_{min}: the minimum energy cost for VM pair $(v_i, v_i^{'})$

j^*: middlebox mb_{j^*} is assigned to $(v_i, v_i^{'})$

```
C = 0;
for (i = 1 to l)
    c^i_min = infinite;
    for (j = 1 to m)
        if (c(i, j) ≤ c^i_min and load(j) < κ)
            c^i_min = c(i, j);
            j* = j;
        end if;
    end for;
    load(j*)++;
    C = C + c^i_min;
end for;
RETURN C.
```

MB-BASED ALGORITHM

MB-Based Algorithm:

For each MB instance,it is assigned κ VM pairs among all the VM pairs that give the minimum energy consumption when going through that MB instance.

Algorithm 2: MB-Based Algorithm.

Input: A data center $G(V, E)$ with l VM pairs and m MBs
Output: Total energy cost C for all the l VM pairs.
Notations:
i: the index for VM pairs
j: the index for middlebox instances
X_j: the set of VM pairs assigned to mb_j
$assigned[i]$: true if $(v_i, v_i^{'})$ is assigned, false if not

$C = 0$;
for $(i = 1$ to $l)$
  $assigned[i] = false$;
end for;
for $(j = 1$ to $m)$
  $X_j = \phi$;
  for $(i = 1$ to $l)$
    if $(assigned[i] == false)$
      $X_j = \{(i, c(i,j))\} \cup X_j$;
    end if;
  end for;
  Sort X_j in the non-descending order of $c(i,j)$;
  $X_j = \{(x_1, c(x_1, j)), (x_2, c(x_2, j)), (x_3, c(x_3, j)), ...\}$,
    where $c(x_1, j) \le c(x_2, j) \le c(x_3, j)...$;
  for $(k = 1$ to $\kappa)$
    $C = C + c(x_k, j)$;
  end for;
end for;
RETURN C.

VM-MB-BASED ALGORITHM

VM-MB-Based Algorithm:

In each round, it checks which VM pair is assigned to which MB instance, such that when that VM pair traverses that MB instance, it yields the minimum energy consumption among all the unassigned VM pairs and all the MB instances in that round.

Algorithm 3: VM-MB-Based Algorithm.

Input: A data center $G(V, E)$ with l VM pairs and m MBs

Output: Total energy cost C for all the l VM pairs.

Notations:

i: the index for VM pairs

j: the index for middlebox instances

$load(j) = 0$: the current load of mb_j

c_{min} and j^*: the minimum energy cost obtained in each round by assigning middlebox mb_{j^*} to $(v_i, v_i^{'})$

Total energy cost in the data center

```
C = 0;
for (i = 1 to l)
    assigned[i] = false;
end for;
while (there are still unassigned VM pairs)
    c_min = infinite;
    for (i = 1 to l)
        if (assigned[i] == false)
            for (j = 1 to m)
                if (load(j) < κ and c(i, j) ≤ c_min)
                    c_min = c(i, j);
                    j* = j;
                end if;
            end for;
        end if;
    end for;
    load(j*)++;
    C = C + c_min;
end while;
RETURN C.
```

PERFORMANCE EVALUATION

-The source and destination VMs of each VM pair are randomly placed on the PMs and the MB instances are randomly placed on the switches.

-In all the simulation plots, each data point is an average of 10 runs, and the error bars indicate 95% of confidence interval.

EFFECT OF NUMBER OF VM PAIRS *L*

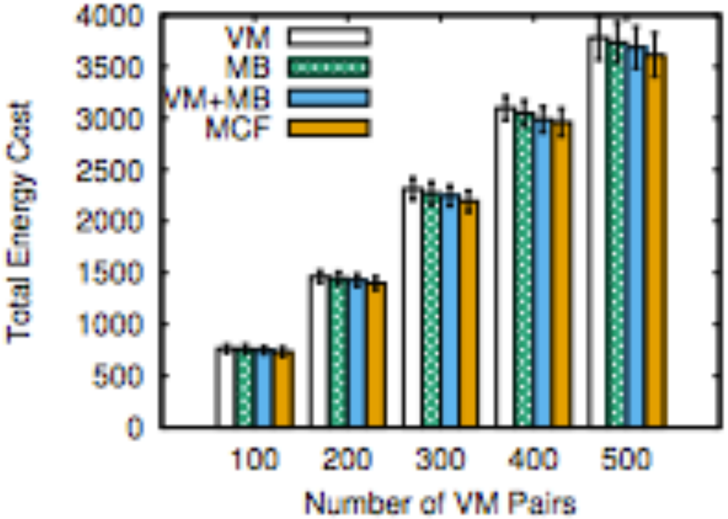

Fig. 4. Varying number of VM pairs under uniform energy model. Here, number of MBs $m = 3$, number of PMs in data center is 128.

EFFECT OF NUMBER OF MB INSTANCES *M*

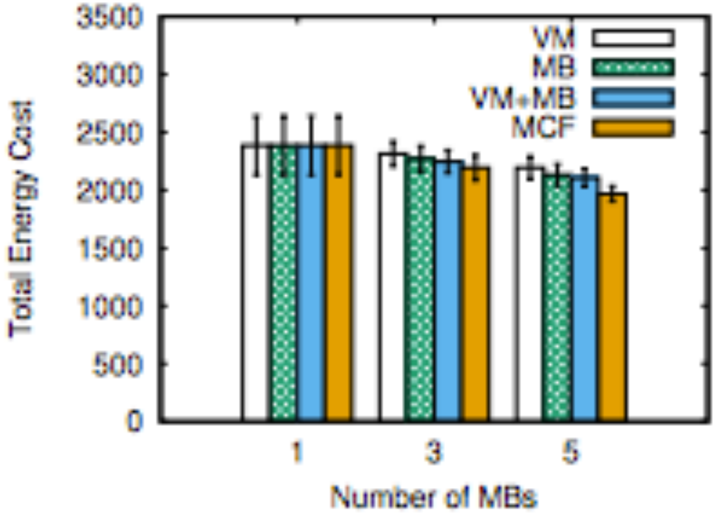

Fig. 5. Varying number of MB instances under uniform energy model. Here, number of VM pairs $l = 300$, number of PMs in data center is 128.

COMPARISON IN LARGE DATA CENTERS

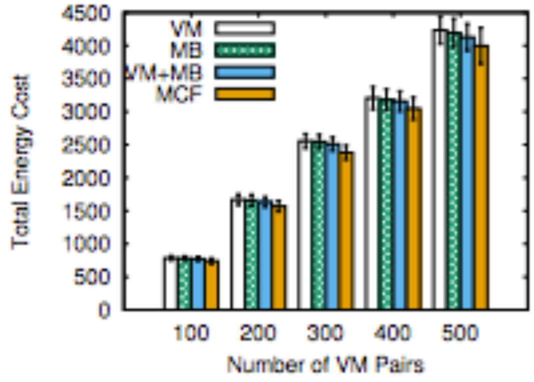

Fig. 6. Varying number of VM pairs under uniform energy model. Here, number of MBs $m = 3$, number of PMs in data center is 1024.

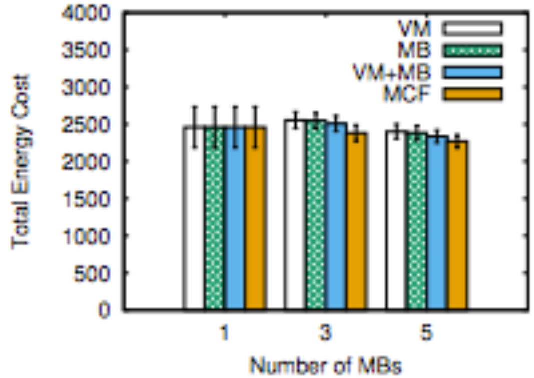

Fig. 7. Varying number of MB instances under uniform energy model. Here, number of VM pairs $l = 300$, number of PMs in data center is 1024.

COMPARISON UNDER SKEWED ENERGY MODEL

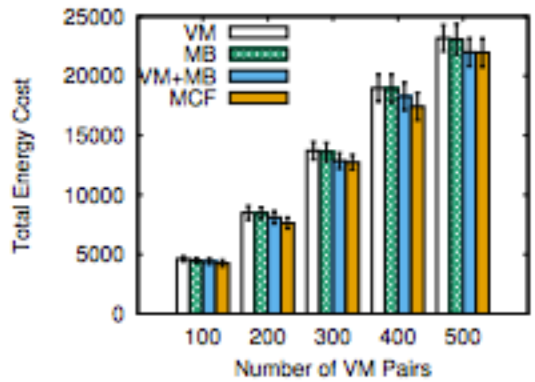

Fig. 8. Varying number of VM pairs under skewed energy model. Here, number of MBs $m = 3$, number of PMs in data center is 128.

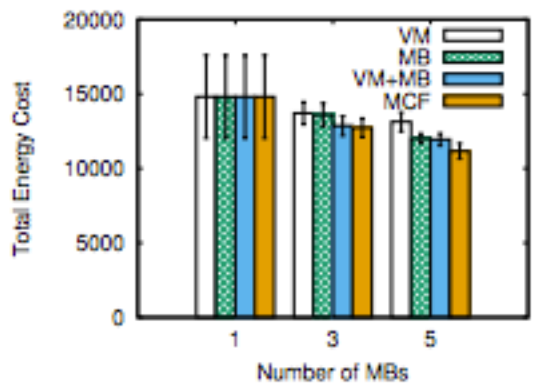

Fig. 9. Varying number of MB instances under skewed energy model. Here, number of VM pairs $l = 300$, number of PMs in data center is 128.

COMPARISON UNDER SKEWED ENERGY MODEL

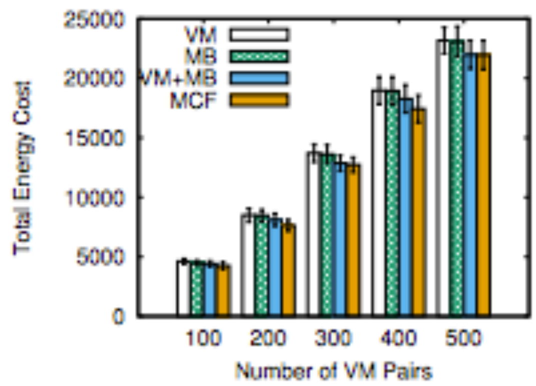

Fig. 10. Varying number of VM pairs under skewed energy model. Here, number of MBs $m = 3$, number of PMs in data center is 1024.

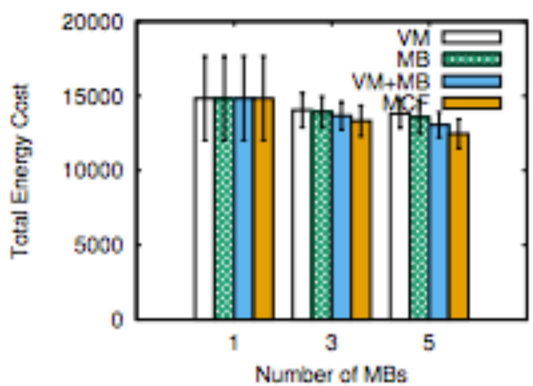

Fig. 11. Varying number of MB instances under skewed energy model. Here, number of VM pairs $l = 300$, number of PMs in data center is 1024.

CONCLUSION

-The goal of LBMAPis to minimize the energy cost of all the communicating virtual machine pairs who must traverse a middlebox for policy requirement, while taking into account of the limited capacity of the middlebox.

-We formulated LB-MAP formally and proved that LB-MAP is equivalent to the well-known minimum cost flow problem (MCF).

CONCLUSION

- We also designed a suite of efficient heuristic algorithms based on different criteria.

- Via extensive simulations, we showed that all the heuristic algorithms perform close to the optimal minimum cost flow algorithm, while VM+MB-Based performs best among all the heuristic algorithms.

FUTURE WORK

- We assume that there is only one middlebox type such as load balancers. In the future, we will consider a more general problem wherein multiple types of middleboxes exist, each having multiple instances.