Server Consolidation

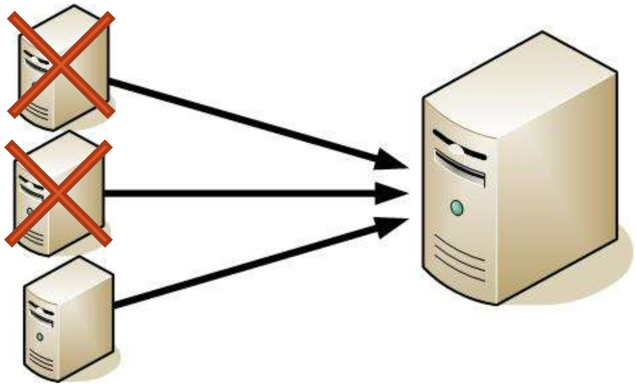
Heaviest First Algorithm

By : Peyman Navidi

General Definition

- Server consolidation lets one physical server **run multiple VMs**, which **reduces the amount of physical hardware** required. Other server consolidation benefits include **lower power and cooling costs**.

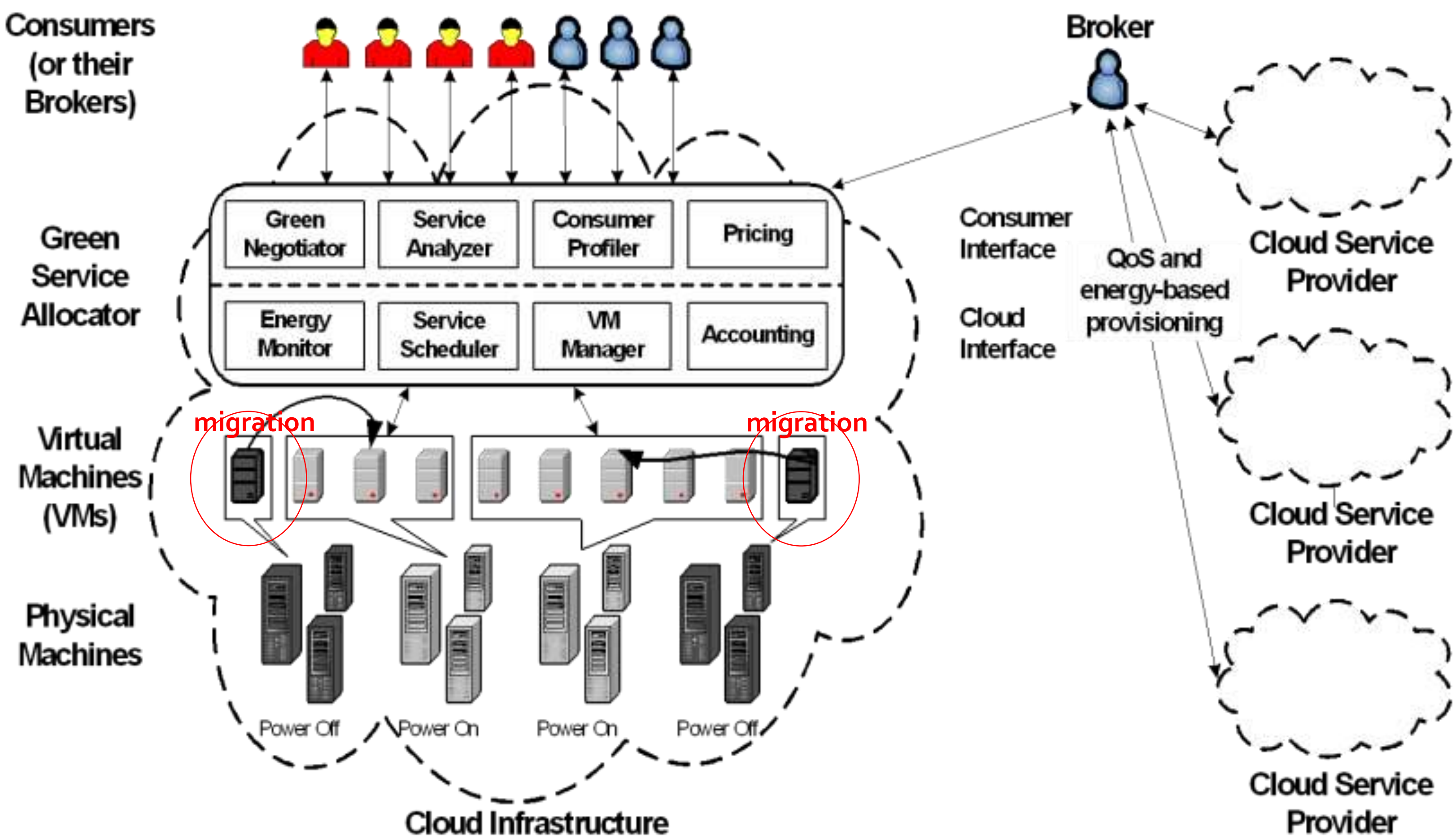
- Server consolidation is an approach to the **efficient usage of computer server resources** in order to reduce the **total number of servers** or server locations that an organization requires.



An Overview of The Main Research

- However, Cloud Data Centers consume excessive amount of energy.
- Reducing the operational costs of powering and cooling Data Centers.
- High carbon footprints are incurred due to the massive amount of electricity needed to power and cool the numerous servers hosted in these data centers.
- Reduce greenhouse gas emissions
- Improving reliability

Green Cloud Computing Architecture



Three Sub-Problems

- **When to migrate VMs?**
  - Host overload detection algorithms
  - Host underload detection algorithms

- **Which VMs to migrate?**
  - VM selection algorithms

- **Where to migrate VMs?**
  - VM placement algorithms

About Paper

Server Consolidation Algorithms with Bounded Migration Cost and Performance Guarantees in Cloud Computing

2011 Fourth IEEE International Conference on Utility and Cloud Computing

Yufan Ho, Pangfeng Liu, Jan-Jan Wu

Department of Computer Science and Information Engineering, National Taiwan University

Taipei, Taiwan

Abstract

Consolidation of virtual machines is essential to achieve energy optimization in cloud computing environments. As virtual machines dynamically enter and leave a cloud system, it becomes necessary to relocate virtual machines among servers. However, relocation of virtual machines introduces run-time overheads and consumes extra energy, thus an careful planning for relocation is necessary. We model the relocation problem as a modified bin packing problem and propose a new server consolidation algorithm that guarantees server consolidation with bounded relocation costs. We also conduct a detailed analysis on the complexity of the server consolidation problem, and give a upper bound on the cost of relocation. Finally, we conduct simulations and compare our server consolidation algorithm with other relocation methods, like First Fit and Best Fit method. The experiment results suggest an interesting trade-off between server consolidation quality and relocation cost. Our algorithm is able to trade about 1% in server consolidation quality for a reduction about 50% in relocation cost, when compared with other well known bin packing algorithms. We also note that the relocation cost incurred in our method is much less than the theoretical bound we provided. The reason is that we overestimate the amount of relocation from theoretical analysis, and the actual amount of relocation found from experiments is much less than the worst-case bound from theoretical analysis.

Description of The Model

- We model the **virtual machine relocation** problem in cloud computing as a modified **bin packing** problem.
- Virtual Machine -> **item**
- Phisycal Mashin -> **bin**
- State of Items and Bins:
  - A state describes the status of **items, bins,** and the **mapping function** from items to bins.
  - $S = \{T, B, F\}$
  - $T = \{t_1, \ldots, t_n\}$ $s(t) \in (0, 1]$ the weight of item t
  - $B = \{b_1, \ldots, b_m\}$ $H(b) = \sum_{F(t)=b} s(t)$
  - $F: T \rightarrow B$
  - $F(t) = b$ means that item t is in bin b
  - $H(b) \leq 1$, for all $b \in B$, since each bin has capacity 1
  - $D(S) = s(T)/|B|$ degree of saturation
  - $s(T) = \sum_{t \in T} s(t)$ $|B|$ number of bins in B

Describe Relocating Algorithm

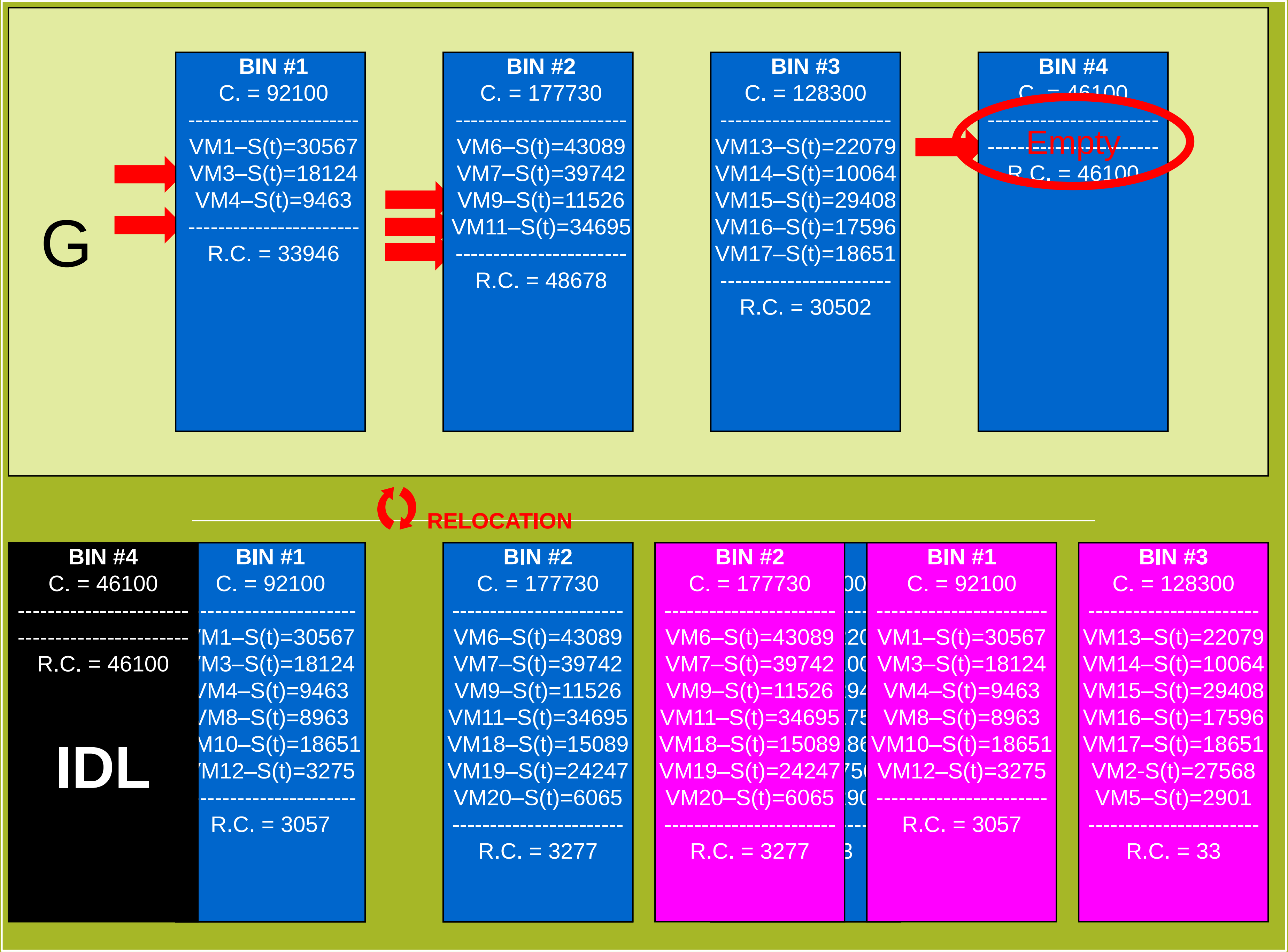
- We use a **set G** to denote those bins that have **not been packed**, and **initially every bin is in G**. In each phase we remove a bin from G and pack it. We repeat this process until every bin is packed or becomes empty.

- In each phase we remove and pack the bin b that has the **highest weight** in G.
- We do so by checking all items in other bins in G to see if they can fit into b.
- If we find any item t that can be added into b, we add t to b. migration
- We repeat this process until no such t can be found.
- During packing some bins might become empty. These empty bins are removed from G. Idle (Power Off)

- They call this algorithm as "**Heaviest First**" relocation method.

Heaviest First Relocation Method

- Place all bins into G
- **while** there are bins in G **do**
    - choose the bin b in G that has the highest weight
    - **for** every bin b' in G other than b **do**
        - **for** each item t in b' **do**
            - **if** t could be added into b **then**
                - relocate t from b' to b
            - **end if**
        - **end for**
        - **if** b' becomes empty **then**
            - remove b' from G
        - **end if**
    - **end for**
    - remove b from G
- **end while**



G
BIN #1
C. = 92100
VM1–S(t)=30567
VM3–S(t)=18124
VM4–S(t)=9463
R.C. = 33946
BIN #2
C. = 177730
VM6–S(t)=43089
VM7–S(t)=39742
VM9–S(t)=11526
VM11–S(t)=34695
R.C. = 48678
BIN #3
C. = 128300
VM13–S(t)=22079
VM14–S(t)=10064
VM15–S(t)=29408
VM16–S(t)=17596
VM17–S(t)=18651
R.C. = 30502
BIN #4
Empty
R.C. = 46100
RELOCATION
BIN #4
C. = 46100
R.C. = 46100
IDL
BIN #1
C. = 92100
R.C. = 3057
BIN #2
C. = 177730
VM6–S(t)=43089
VM7–S(t)=39742
VM9–S(t)=11526
VM11–S(t)=34695
VM18–S(t)=15089
VM19–S(t)=24247
VM20–S(t)=6065
R.C. = 3277
BIN #2
C. = 177730
VM6–S(t)=43089
VM7–S(t)=39742
VM9–S(t)=11526
VM11–S(t)=34695
VM18–S(t)=15089
VM19–S(t)=24247
VM20–S(t)=6065
R.C. = 3277
BIN #1
C. = 92100
VM1–S(t)=30567
VM3–S(t)=18124
VM4–S(t)=9463
VM8–S(t)=8963
VM10–S(t)=18651
VM12–S(t)=3275
R.C. = 3057
BIN #3
C. = 128300
VM13–S(t)=22079
VM14–S(t)=10064
VM15–S(t)=29408
VM16–S(t)=17596
VM17–S(t)=18651
VM2-S(t)=27568
VM5–S(t)=2901
R.C. = 33

Analysis

- **First Fit Possible**
  - When we pack an item set T into a bin set B using the First Fit method, we need to determine an orderings of T (denoted by τ) and an order of B (denoted by β).
  - Then we place the first item t in τ into the first bin in β that has sufficient remaining capacity to accommodate t, and remove t from τ . We repeat this process until τ becomes empty.

- **Relocation Problem**
  - A relocation is a transition from a state $S = \{T, B, F\}$ to a new state $S' = \{T, B, F'\}$, by changing the mapping function from F to F'. That is, we will relocate some items in T to their new destination bins.
  - Relocating items to new destination bins incurs **relocation costs**. We assume that the cost of relocating an item t is $\boldsymbol{s}(\boldsymbol{t})$, the weight of t.
  - The **more resources** a virtual machine requires, the **more cost** to relocate it.

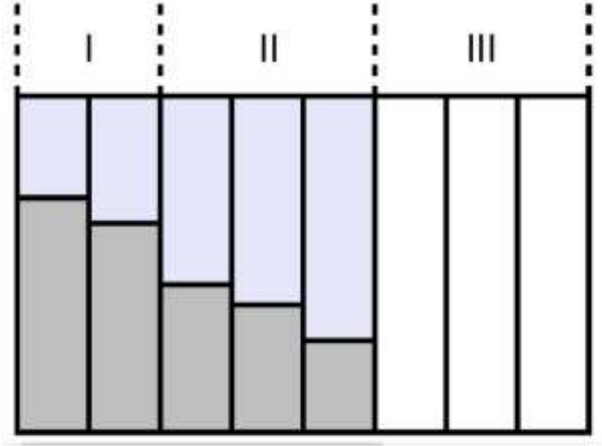
Analysis

- **Definition 1 :**
  - A state $S = \{T, B, F\}$ is First Fit possible if there exist an ordering τ of items in T , and an ordering β of bins in B, such that if we use the First Fit method to place items in T to bins in B, the resulting **mapping function F' will be as same as F**.

- **Witness of First Fit :**
  - To verify that a state $S = \{T, B, F\}$ is First Fit possible, it is necessary to find an ordering of bins (β), as in **Definition 1**.
  - We define such an ordering of bins B as a witness to the state S. It is now easy to verify whether a state S is First Fit possible – we just need to find a witness β for S, that every bin b does not have enough remaining capacity to accommodate any item in a bin that appears later than b in β.

Analysis

- Let S and S' be the input and output states of Heaviest First relocation method.
- We now classify the bins in S into three categories according to the number of steady items.
- We use N_1, N_2, and N_3 to denote the number of bins in Group 1, 2 and 3 respectively.
  - **Group 1** : Bins that contain at least ½ steady items.
  - **Group 2** : Bins that contain less than ½ steady items.
  - **Group 3** : Bins that become empty after Heaviest First relocation method.



Analysis of Relocation Cost

- **Lemma 1**:
  - For any resulting state of Heaviest First relocation method, the number of bins in Group 3 is at least the number of bins in Group 2 minus 1 :

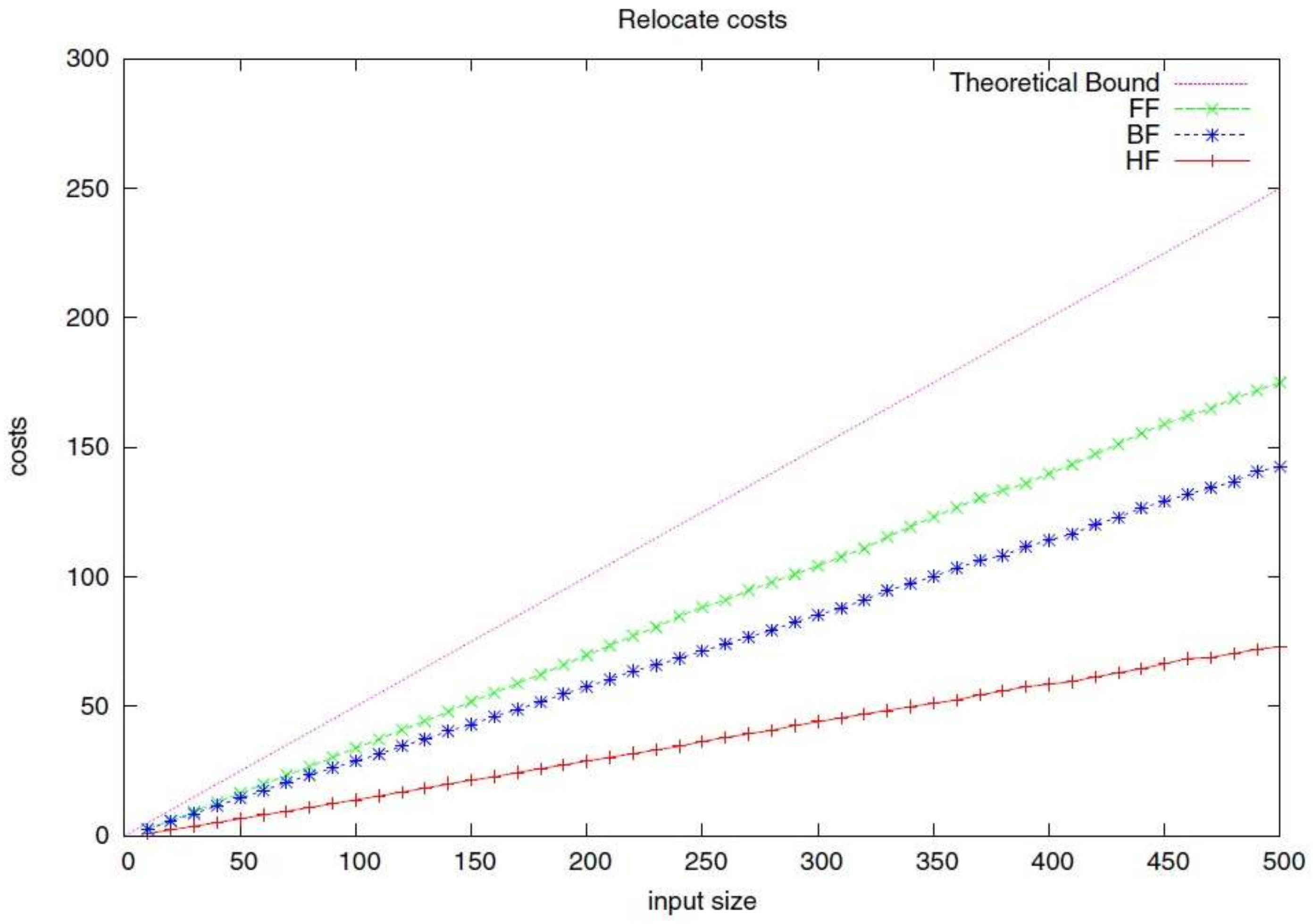
$$N_3 \geq N_2 - 1$$

- The total **relocation cost** for Heaviest First relocation method to relocate a state $S = \{T, B, F\}$ is bounded by $|B|/2$.

- **Corollary 1:**
  - When applying Heaviest First relocation method to relocate a state $S = \{T, B, F\}$, the **ratio between relocation cost** and **total weight of items** is less than or equal to ½ $D(S)$.

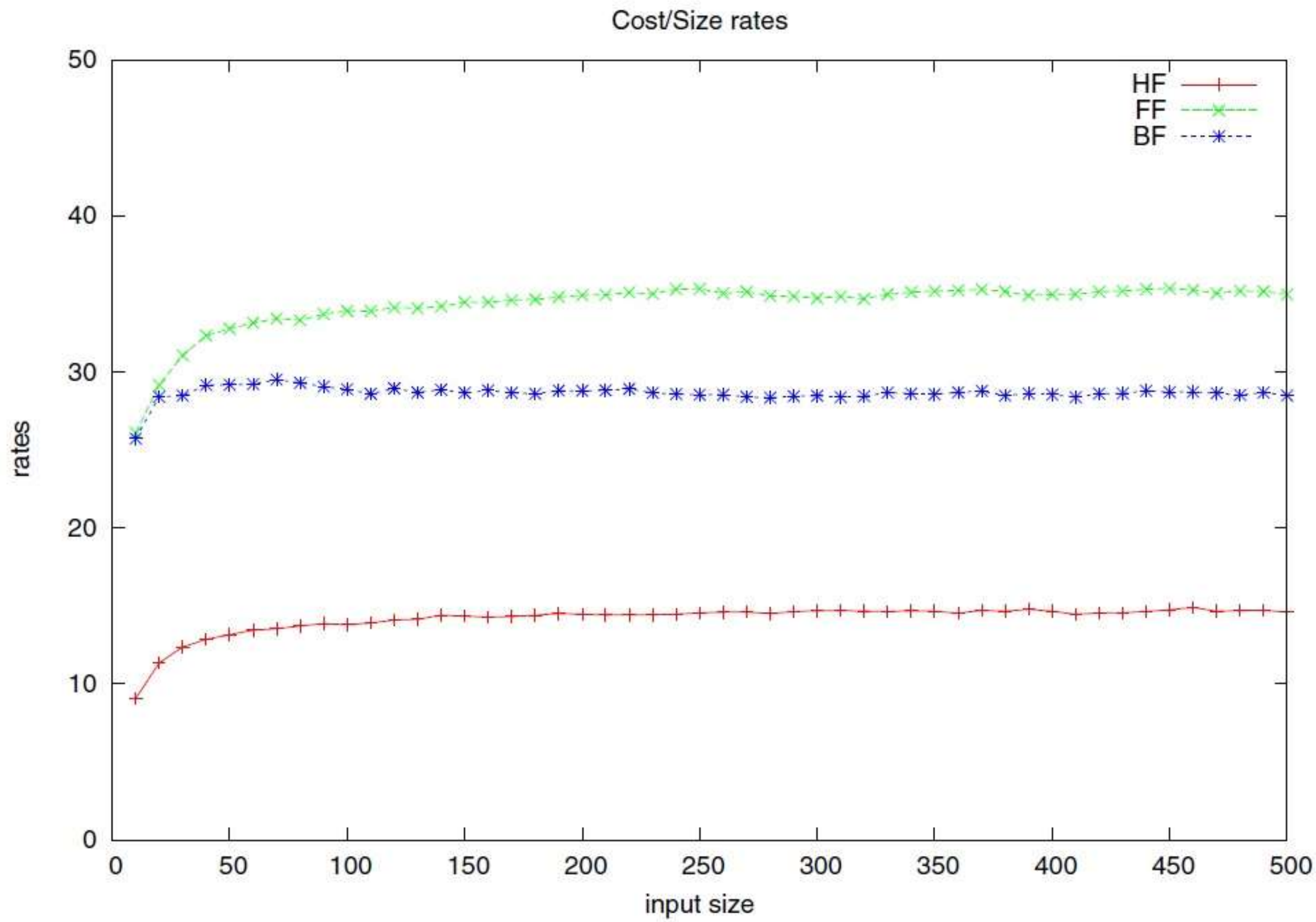
Experiments

- Heaviest First reduces the relocation costs by 58% when compared with Naive First Fit algorithm, and 48% with Best Fit algorithm.



Experiments

- The ratio is about 15% in Heaviest First relocation method, 29% in Best Fit algorithm, and 36% in Naive First Fit algorithms.



Related Work

- **Srikantaiah et al. :**
  - Investigate the inter-relationship between **energy consumption, resource utilization, and performance** of consolidated workloads.
  - The study reveals the energy performance tradeoffs for consolidation and shows that **optimal operating points exist**.
- **Li et al. :**
  - Propose **EnaCloud**, which places **applications dynamically** on servers, with consideration of energy efficiency in a cloud platform.
  - The application placement is abstracted as **a bin packing** problem, and an energyaware **heuristic algorithm** is proposed to get an appropriate solution.
- **Beloglazov and Buyya :**
  - Propose a **resource management policy** for virtualized cloud data centers.
  - They propose and evaluate **heuristics for dynamic reallocation** of virtual machines to minimize energy consumption, while providing reliable quality of services.
  - The heuristics are based on the idea of setting **upper and lower utilization thresholds** for hosts and keeping total utilization of CPU by all virtual machines between these thresholds.

Related Work

- **Liao et al. :**
  - Use **live migration** to transfer loads among servers on a **multi-layer ring-based** overlay to reduce power consumption.
  - Active servers are organized into **3-layer rings**, and each ring consists of servers that have **similar workloads** within a specific load interval, so that the inner ring has the servers with the **heavy loads**, the outer ring has the the servers with **light workloads**, and the middle ring has **the rest of the servers**.
- **Hanson et al. :**
  - Propose AMP, an **autonomic manager** in charge of the power states of servers.
  - AMP is designed to work in **conjunction with other** autonomic managers, and is responsible for **cutting excess power** use and preserving the **lifespan** of servers it manages.
  - AMP considers **broader metrics** in its decision, which may include **power consumption**, **server lifespan**, and **interactions with other application managers**.

CONCLUSIONS

- We propose a Heaviest First relocation method for scheduling the relocations of virtual machines as a solution of the relocation problem.

- As the simulating result shows, Heaviest First relocation method can trades a little quality of consolidation for a large amount of relocation cost.

- **How SLA violation be prevented?**

References

- Amazon ec2. http://aws.amazon.com/ec2/
- Yufan Ho and Pangfeng Liu and Jan-Jan Wu Server Consolidation Algorithms with Bounded Migration Cost and Performance Guarantees in Cloud Computing, 2011 IEEE DOI 10.1109/UCC.2011.30.
- A. Beloglazov and R. uyya. Energy efficient resource management in virtualized cloud data centers. Cluster Computing and the Grid, IEEE International Symposium on, 0:826–831, 2010.