Migration-based Virtual Machine Placement in Cloud Systems

Kangkang Li, Huanyang Zheng, and Jie Wu

Temple University, Philadelphia, PA, USA

Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

Introduction

- Virtual Machine Placement (VMP)
- Motivational Example

Virtual Machine Placement (VMP)

- Heterogeneity of VMs

  --different resource demands

  --different running jobs

- Different VMs running on the same PM can have different job completion times.

Virtual Machine Placement (VMP)

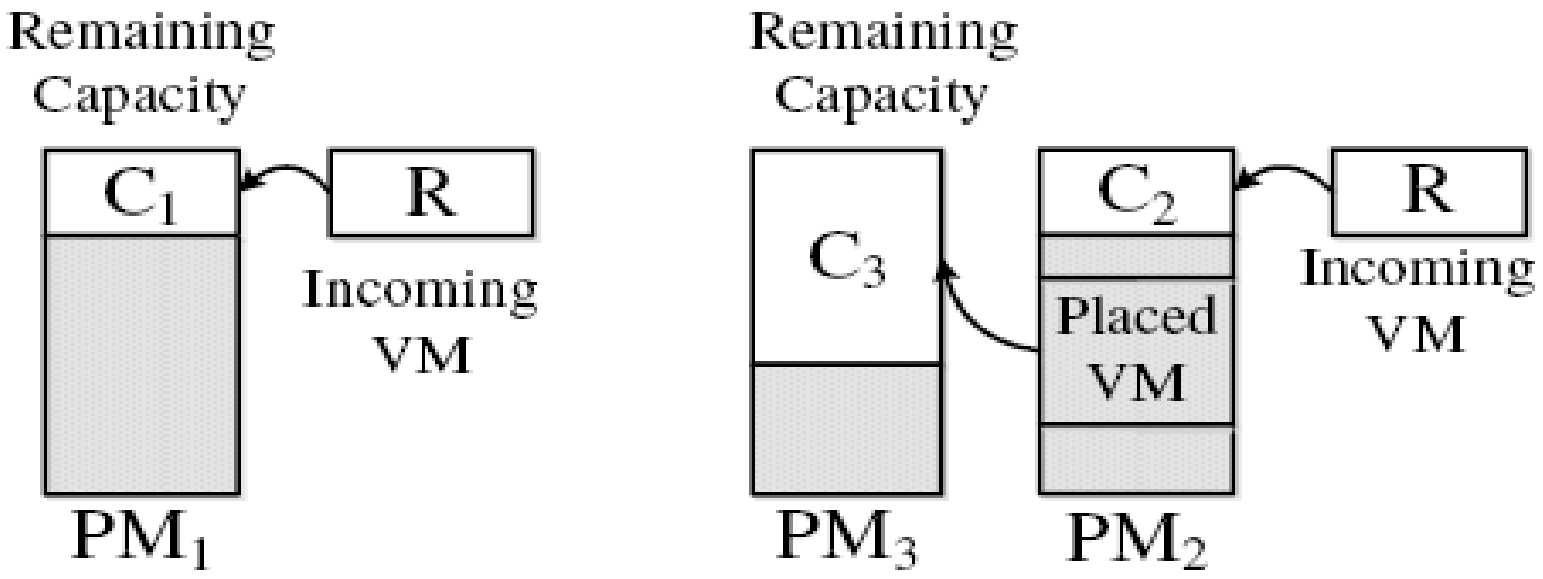
- Heterogeneity of PMs

  --different CPU architectures
  --different OSes
  --differing amounts of memory, etc.

- The same VM placed on different PMs will have different completion times.

- VM placement can be reduced to a knapsack problem, which is NP-hard.

Motivational Example



(a) Direct placement

(b) Migration-based placement

Off-line Migration-based Virtual Machine Placement Algorithm

- Completion time increase:

- Since the completion time of the same VM on a different PM host varies, for the migrated VM, there might be an increase in completion time increase, due to the host changing.

- Migration constraint:

- Even with the increased completion time of the migrated VM, the migration-based VM placement is still better than direct placement at minimizing the total completion time

Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

Problem Definition

- **Input**:

  Given a cloud system with N PMs, we have a set of M incoming VMs to be placed.

TABLE I
NOTATIONS

Notation	Description
VM_i	The i^{th} VM
PM_j	The j^{th} PM
R_i	The resource demand of VM_i
C_j	The remaining capacity of PM_j
x_{ij}	{0,1} variable indicates whether VM_i is placed into PM_j
t_{ij}	The job completion time of VM_i in PM_j

- **Objective**:
  Minimize the total completion time of the input VMs.

$$Minimize \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}t_{ij}$$

$$S.T. \sum_{j=1}^{N} x_{ij} \leq 1, \ \sum_{i=1}^{M} x_{ij}R_i \leq C_j, \ x_{ij} \in \{0,1\}$$

Off-line Migration-based Virtual Machine Placement Algorithm

- Direct Placement:

- Among those PMs which have enough resources to accept the incoming VM, we select the one with the minimal completion time.

- (emulated) Migration-based placement:

- We try to migrate one VM placed on the optimal PM to another host to make space to accept the incoming VM. In the off-line scenario, the migration process is emulated. Therefore, we call it (emulated) Migration-based Placement.

Off-line Migration-based Virtual Machine Placement Algorithm

- Determination of a qualified victim VM
- Can save enough resources for accepting a new VM
- Can find a new available host with enough current capacity to accept the new VM
- With the increased completion time of the migrated victim VM, the completion time by migrated-based placement is less than that of direct placement (migration constriaint is satisifed)

Off-line Migration-based Virtual Machine Placement Algorithm

Special Cases:

- We cannot find a qualified victim VM on the mincompletion time PM. We then try to place the incoming VM in the next-min-completion time PM. If the incoming VM still cannot be placed, we continue to try the next-next-min-completion time PM. The search process is minated when the incoming VM is placed, or encounters the best available PM found by Direct Placement.

- After trying all the PMs, and if the incoming VM is still not placed, we have no choice but to reject it.

Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

On-line Virtual Machine Placement

Major difference compared to the off-line scenario:

- The information of incoming VMs are unknown to us.
- Migration is actually implemented, which introduces delay in transferring the migrated VM image.
- An extra migration overhead is added to the migration constraint
- Migration overhead is proportional to the size of the migrated VM. Larger VMs need more time to transfer.

Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

A Hybrid Scheme

Motivation:

•On-line migration delay causes too much overhead in the total completion time.

•There is NO migration overhead in the off-line scenario.

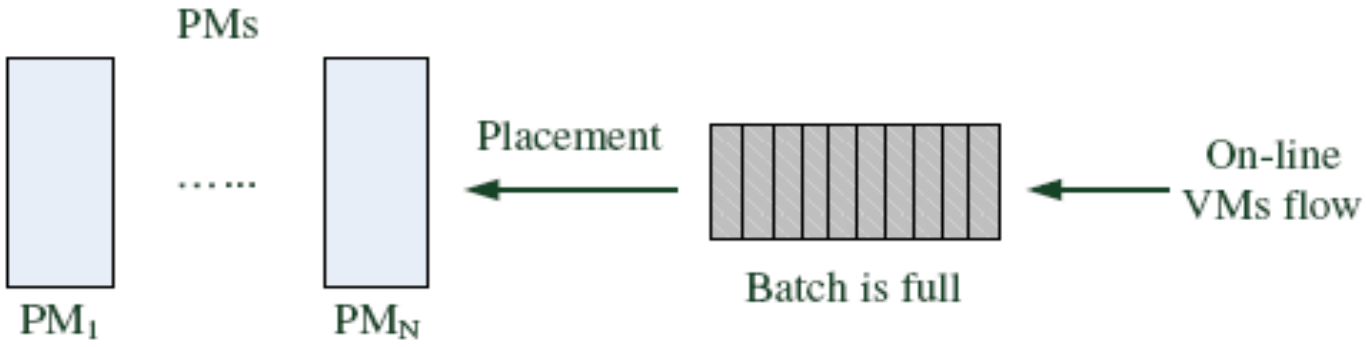
Solution:

•Reserve a batch to store the on-line VM requests

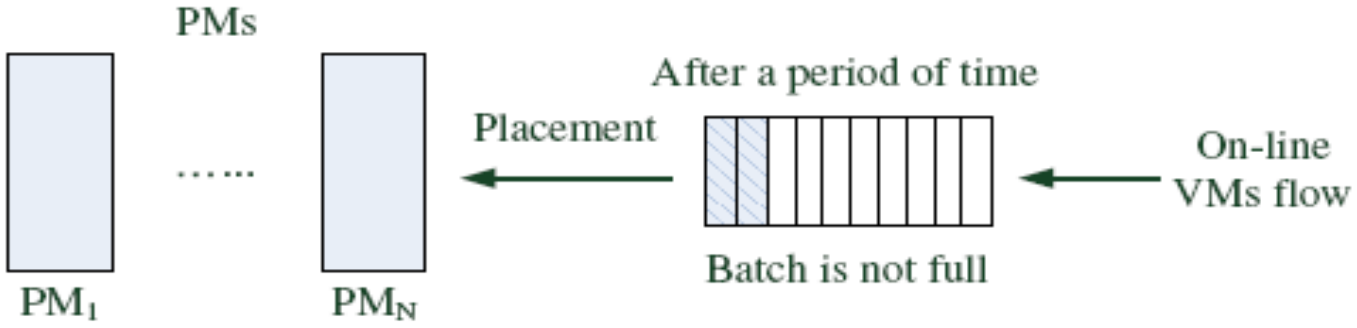
•The information of reserved VMs are known to us, enabling us to place these VMs as in the off-line scenario, and avoiding the migration overhead.

Two batch models

Provider-oriented:



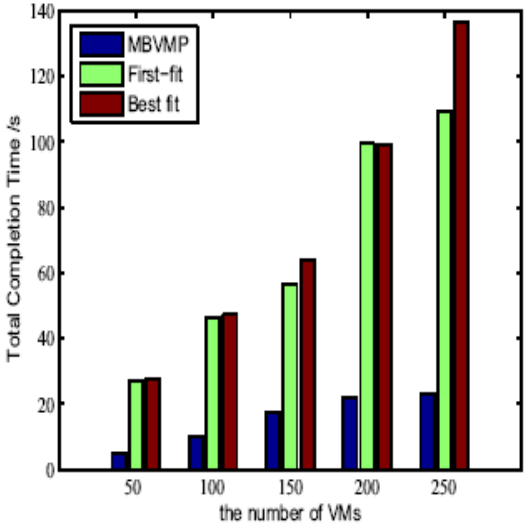
User-oriented:



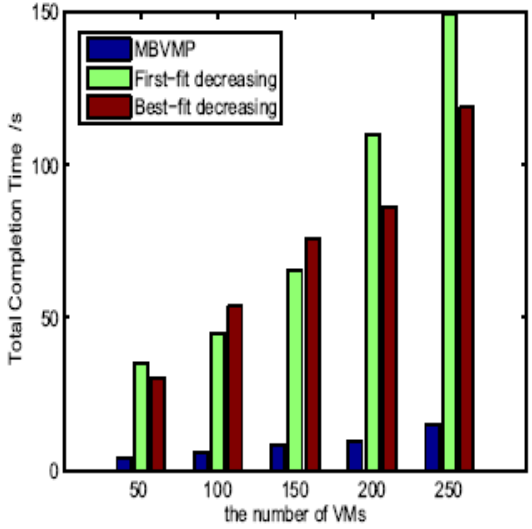
Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

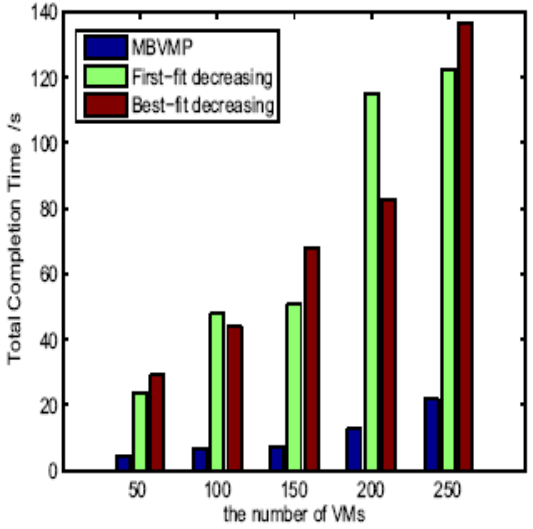
Simulations



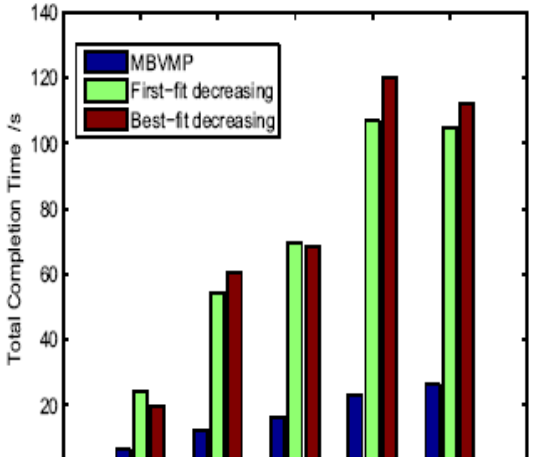
(a) Number of PMs: N=100

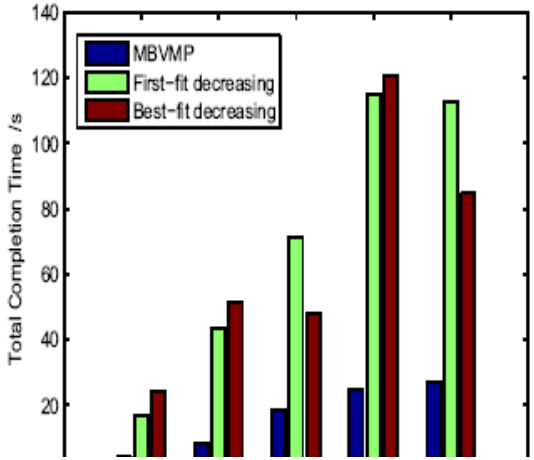


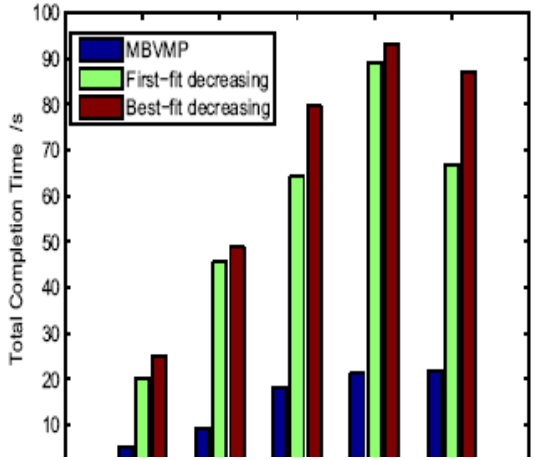
(b) Number of PMs: N=150



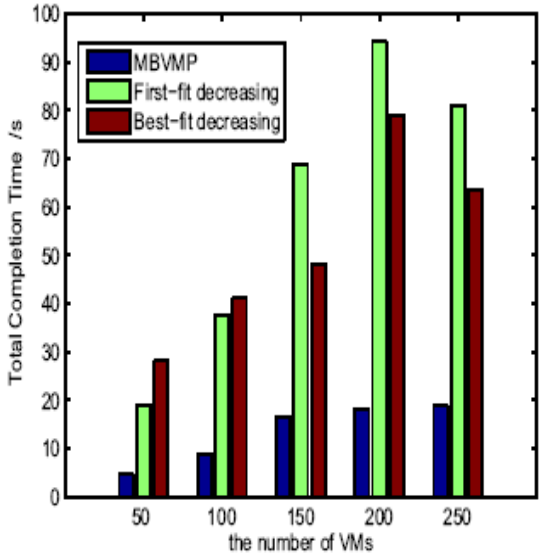
(c) Number of PMs: N=200



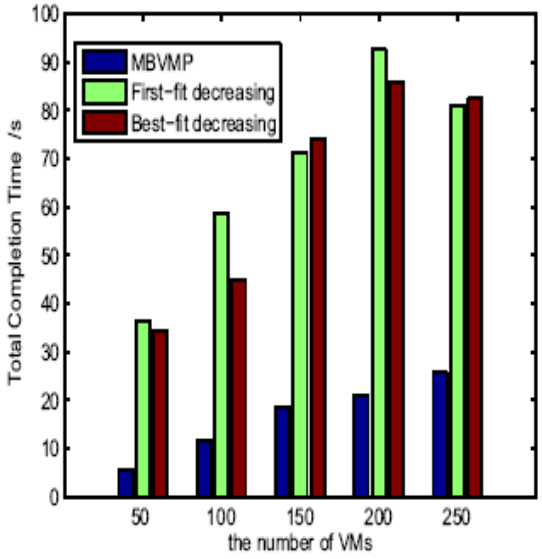




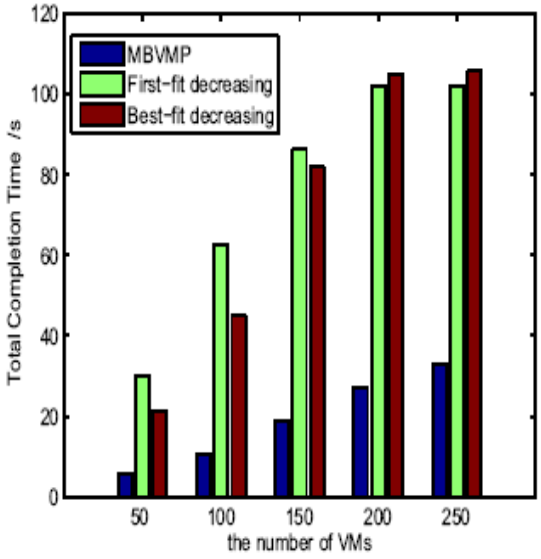
Simulations



(a) variance of PMs' capacity:[50,150]



(b) variance of PMs' capacity:[100,150]



(c) variance of PMs' capacity:[150,150]

PERFORMANCE COMPARISON FOR ON-LINE SCENARIO

Total Completion Time /s	Number of PMs			Coefficient of Migration Cost			Variance of PMs' Capacity		
	100	150	200	0.1	1	10	[10,50]	[10,100]	[10,150]
MBVMP	52.4475	55.2926	83.9606	56.1748	58.3576	53.1326	51.2983	53.9628	52.8962
First-fit	140.7237	214.7462	265.3581	150.1385	254.9847	284.9215	147.4251	221.2317	251.1223
Best-fit	130.2293	206.0019	233.3534	127.9724	215.9011	201.1946	125.9402	237.7812	263.9472

Outline

1. Introduction
2. Off-line VM placement
3. On-line VM placement
4. A Hybrid scheme
5. Simulation
6. Conclusion

Conclusions

- A Migration-based Virtual Machine Placement Algorithm is proposed for minimizing the total completion time of the input VMs.
- Both off-line and on-line scenarios are considered.
- We study the hybrid scheme of integrating off-line placement into an on-line scenario, and propose two batch models considering users and providers separately.
- Experiments and simulations verify the strength of our approach.

Thank you!

Questions?

kang.kang.li@temple.edu