MigVisor: Accurate Prediction of VM Live Migration Behavior using a Working-Set Pattern Model

Cooperated by INTEL and SJTU

Plan

- Motivation & Problem
- System General Model and related work
- Design and Implementation
- Evaluation
- Conclusion

Plan

- Motivation & Problem
- System General Model and related work
- Design and Implementation
- Evaluation
- Conclusion

Motivation

- Object: Virtual machine live migration (VMLM)
  - a critical capability that is indispensable in managing data-center and cloud computing Infrastructure.
- Benefits
  - dynamic load balancing when a node is overloaded;
  - enhance server consolidation when a node is underutilized;
  - facilitating server maintenance
  - high availability of service when a node is at the end of its lifecycle

Problem

- However: VMLM failures occur and waste resources.
  - Only ~87% LM can succeed in practical environment.
  - Resource consumed for Live Migration are wasted for nothing.
  - Disturbance on other co-hosted VMs are wasted for nothing.
    - Overloaded server's performance are further degraded

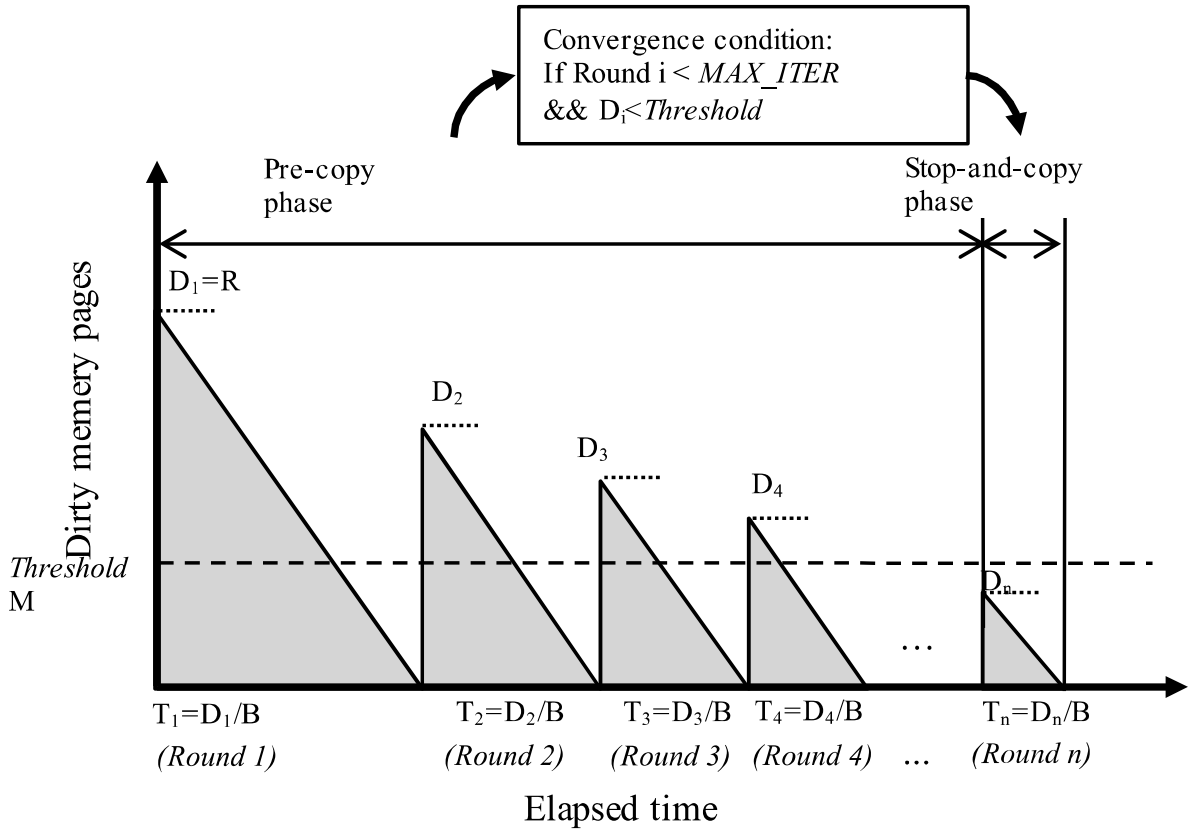
- **Challenge: Prevent VM Live Migration Failures**

Plan

- Motivation & Problem
- System General Model and Related Work
- Design and Implementation
- Evaluation
- Conclusion

Live Migration Process

- Pre-Migration Process



- Prediction on the LM behavior
  - **pre-copy phase**: number of iterations, and each iteration time
  - **Stop-and copy phase**: service downtime under SLA constraint
  - LM Fails: if pre-copy does not converge, downtime is too long

Pre-copy VMLM General Model

In the 1st iteration: all memory pages R are dirtied, so $D_1 = R$. The duration of the first time interval is therefore

$$T_1 = D_1/B. \quad (1)$$

In the 2nd iteration round: new dirty pages are denoted *f1(T1)*. The number of dirty pages for transmission in the second iteration is

$$D_2 = f_1(T_1). \quad (2)$$

The duration of the 2nd iteration is

$$T_2 = D_2/B \quad (3)$$

n_{th} iteration

$$D_n = f_{(n-1)}(T_{(n-1)}) = f_{(n-1)}(\frac{f_{(n-2)}(T_{(n-2)})}{B})... \quad (4)$$

$$T_n = D_n/B, \quad for \quad n > 2 \quad (5)$$

Background-Related Prediction Methods

- Methods of VMLM failure prevention
- A: void failures in advance by accurate prediction
  - LM behavior depiction with linear/polynomial functions
  - **Deficiency:** not accurate enough for dynamic workload in VM, cannot concern affinities between cohosted VMs.
- B: Slow down VM's running to increase success rate
  - decrease vCPU running frequency
  - impose delays to page writes
    - Stun During Page Send (SDPS) of VmWare VxMotion
  - **Deficiency:** not pure live migration, maybe suitable for long distance LM without restricted SLA.

Plan

- Motivation & Problem
- System General Model and Related Work
- Design and Implementation
- Evaluation
- Conclusion

Contribution-Migvisor

- *Goal:*
  - Allocate sufficient network bandwidth for ensuring the success
  - Choose a feasible candidate among cohosted VMs for migration
- Key： Accurate prediction for LV behaviour
- ***MigVisor*** predicts the behaviour of VM live migration by leveraging a program behavioural model:
  - **Using working-set pattern (WSP) model**
  - **Idea: predict near future with latest history with WSP**

Migvisor Predict with WSP

- Theoretical basis:
  - Can depict the behaviour of programs in general purpose computer systems, or computer utility [13]
  - Can provide an intrinsic measurement of a program's memory demands [15]
  - Can serve as a dynamic estimator of the segments currently needed by a program [14]
  - Can summarize the cache access behaviour [10]
- Leverage WPS model in VMLM
  - to improve the prediction accuracy of VM migration behaviour.
  - to overcome the difficulty in describing the memory dirtying page rate,

Prediction Information of Migvisor

- Information Set of Migration Behavior:
  - a VM live migration enters the stop-and-copy phase from the pre-copy phase after the n_{th} round iteration process
- a set of parameters such as the

$$T_{converge} = \sum_{i=1}^{n-1} T_i,$$

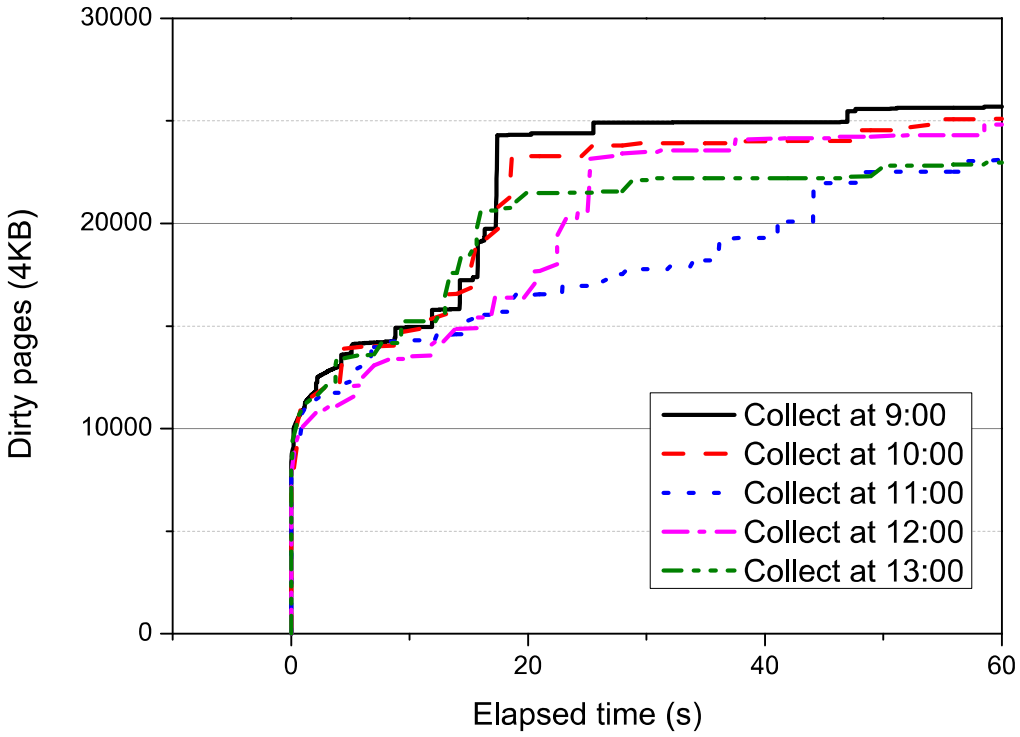
$$T_{downtime} = T_n,$$

$$T_{sum} = T_{converge} + T_{downtime},$$

$$P = \sum_{i=1}^{n} D_i.$$

- $T_{converge}$ **: convergence time**
- $T_{downtime}$**: downtime**
- $T_{converge}$ **: total migration time**
- P **:total number of dirtied pages**

Hard to Predict for VMLM



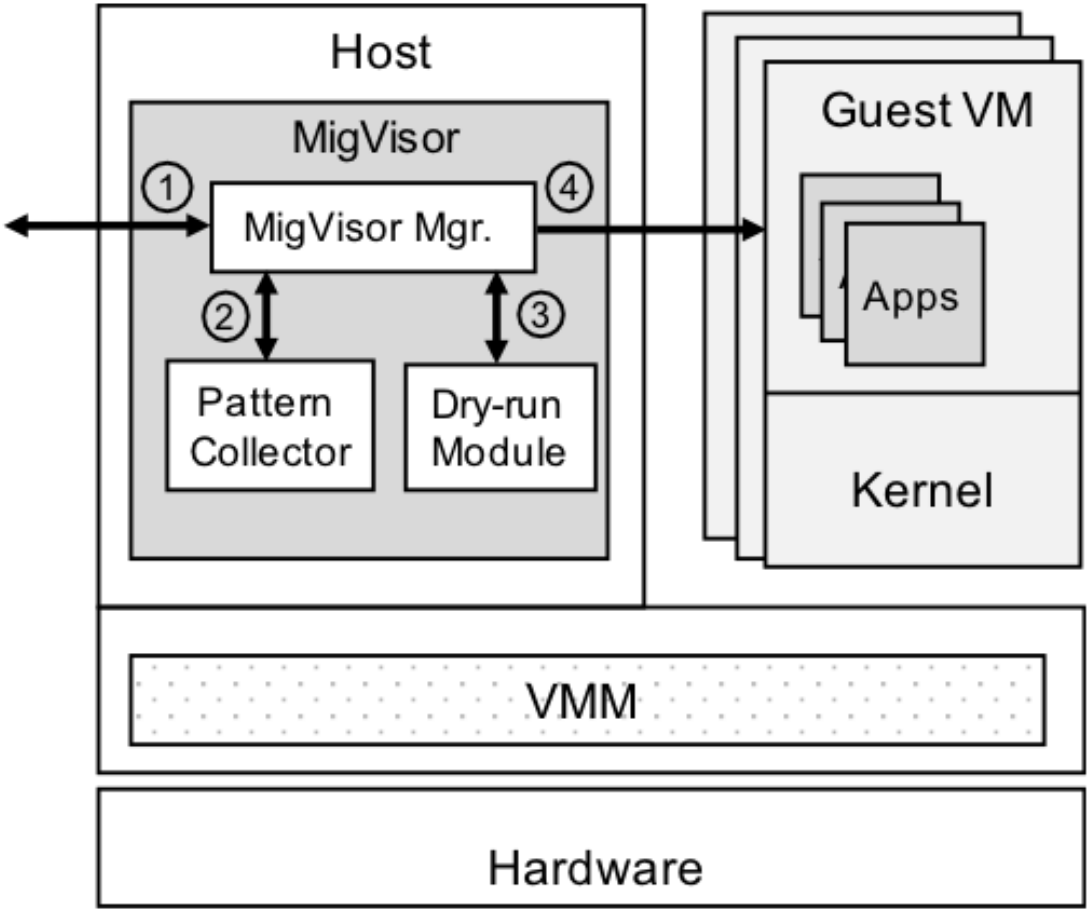
Dirty Page produced by Apache Benchmark

Ensure Live migration

- Least Convergent Bandwidth (LCB):
  - •use the VM migration behaviour prediction to determine the least convergent bandwidth,
  - •the minimum bandwidth required for a successful VMLM.
  - •allow VM managers to allocate a sufficient bandwidth to migrate a given VM.
- Key to Avoid VMLM failure!!!

Migvisor Architecture

- Three Modules:



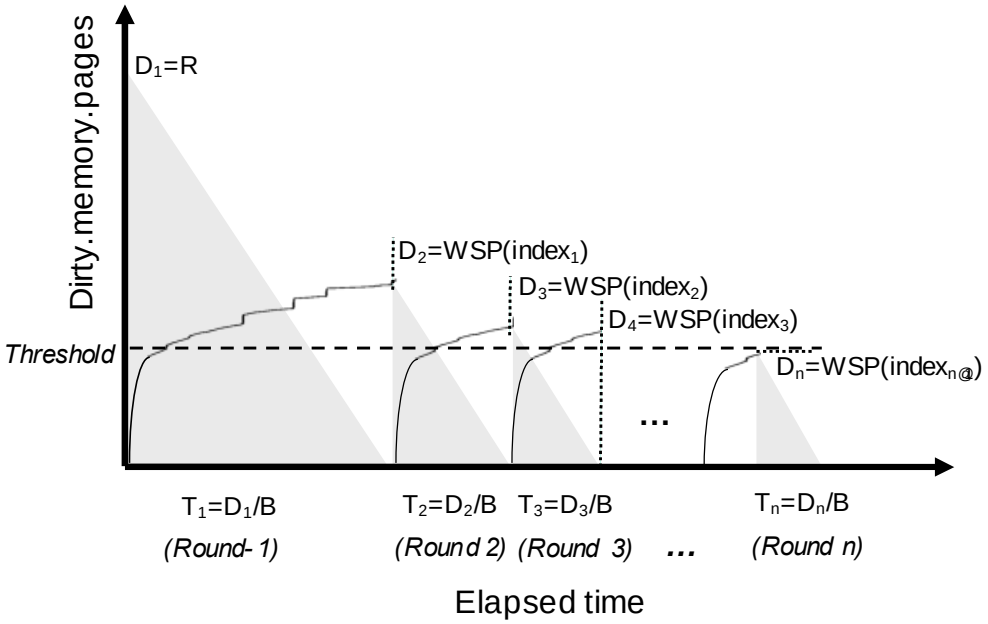
1. **MigVisor manager .**
2. **Pattern Collector (PC) module**
3. **Dry-Run (DR) module**

Migvisor Modules Functionality

- MigVisor manager
  - interacting with the user (VM Manager)
  - predicting the migration behaviour based on the pattern collector module and the dry run module.
- Pattern Collector (**PC**) module
  - for creating and maintaining the working-set pattern (WSP), which records the memory access footprint.
- Dry-Run (**DR**) module
  - execute an emulated migration,
  - Carry out the entire process of the VMLM with compression,
  - but the dirty memory pages are only counted without actually sending to the destination.

PC module

- Collected WSP is able to describe the memory dirtying behavior of a VM.
- a look-up into the WSP array of a VM, we can determine the amount of memory pages that will be dirtied in each iteration of the migration.

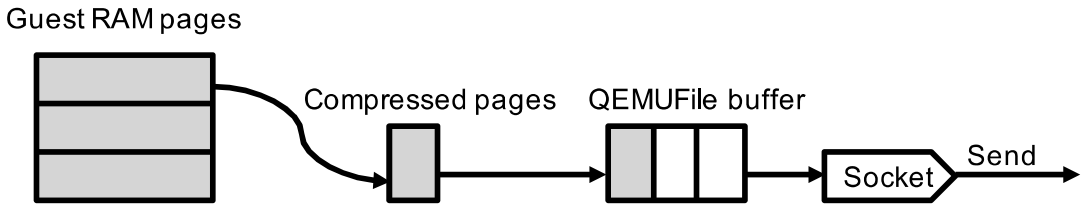


$$D_{i+1} = f_i(T_i) = WSP[index_i] \quad (12)$$

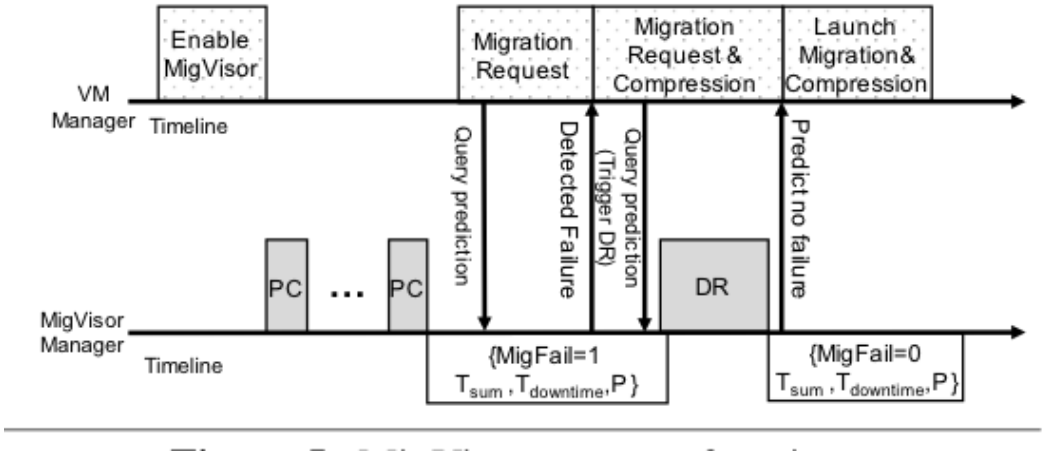
$$where \quad index_i = b\frac{T_i}{10ms}c. \quad (13)$$

Dry-run Module

- to execute an emulated migration,
  - carry out the entire process of the VMLM with compression,
  - the dirty memory pages are only counted and not actually send to the destination.
- Obtain VMLM information set of prediction
  - Including: failure flag, total migration time, downtime, and total number of dirtied pages is readily available.
- more adaptability when compression features are used
  - Owing to the full emulation of migration.



Migvisor Manager



MigVisor manager involves 4 stages of the prediction

1. When a migration is requested, query sent to MigVisor manager;
2. MigVisor manager employs PC prediction. If the prediction is positive, this VM migration is launched normally.
3. If negative by **the PC prediction, launch an analysis for compressed VM migration predicted by the DR module.**
4. **If both the PC and DR prediction methods set the MigFail flag, the VMLM process is abandoned or executed by force with the halt-migration analogical scheme such as SDPS when SLA permits.**

PC vs. DR

- The main difference
  - DR prediction is more accurate:
    - Execute the migration to obtain the actual size of the transferred pages with compression optimization.
    - More costly to complete the prediction to emulate the real migration.
  - PC prediction is more lightweight
    - Faster to get the necessary bandwidth for migration
- Migvisor manager need to choose between them
  - In general, invoke PC
  - DR is only invoked when PC gives a negative results
  - Both failed will trigger halt-migration.

Plan

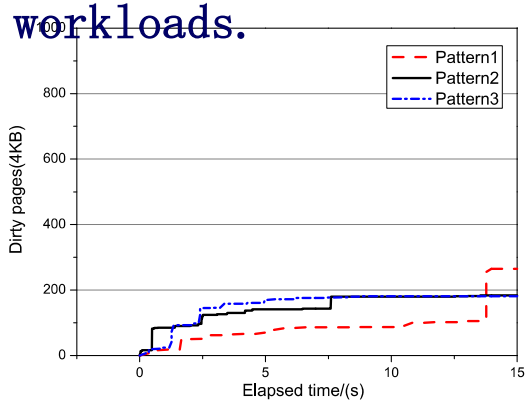
- Motivation & Problem
- System General Model and related work
- Design and Implementation
- Evaluation
- Conclusion

Evaluations

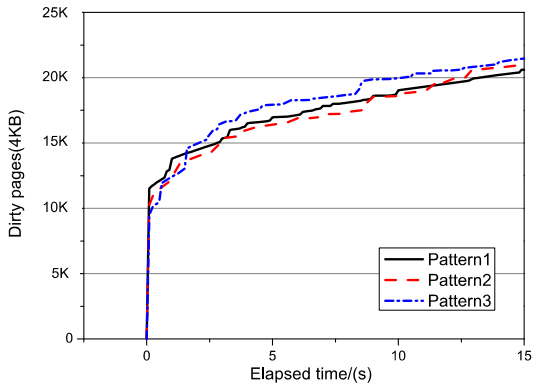
- Platform: Two servers for VM migration.
  - Each server is equipped with one Intel Core i5-4570 3.20GHz CPU, 8 GB of memory, and an Intel X540-T2 network interface card.
  - OS on the VM host and the guest VM is Red Hat Enterprise Linux Server release 6.4 with a Linux2.6.32 − 358.el6.x86 64 kernel,
  - each VM is configured with 1 VCPU and 1GByte of RAM.

Evaluation-Suitability of WSP

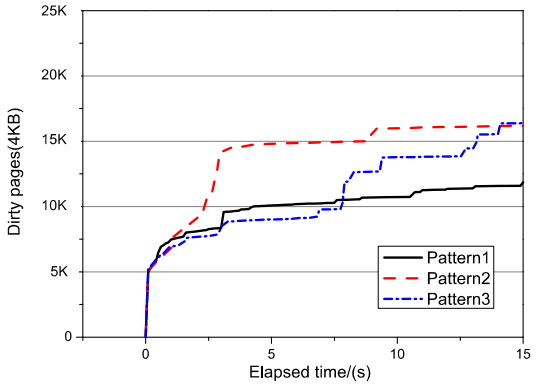
conduct a set of experiments to illustrate that the working-set model is suitable to describe a VM’ s memory access behaviour with a variety of workloads.



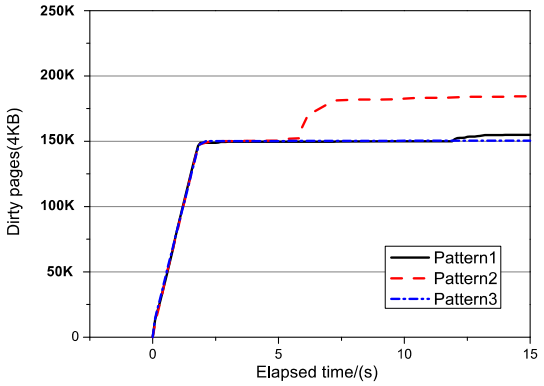
(a) VM with Linux idle Workload



(b) VM with AparchBench Workload



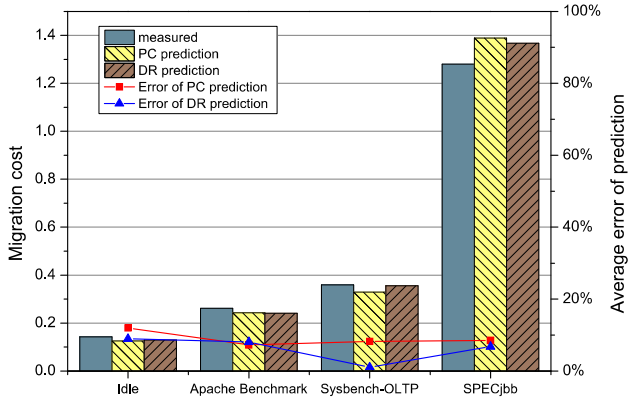
(c) VM with Sysbench-OLTP Workload



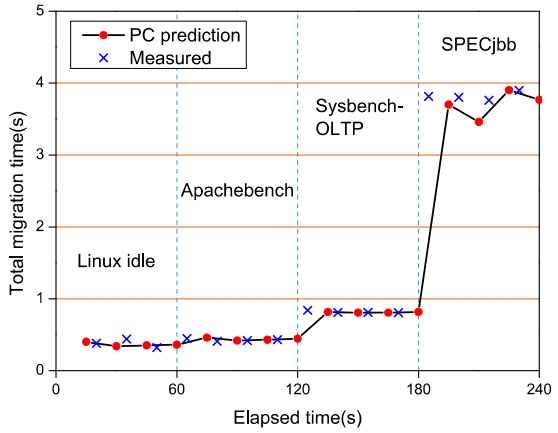
(d) VM with SPECjbb Workload

Evaluation-Prediction Error

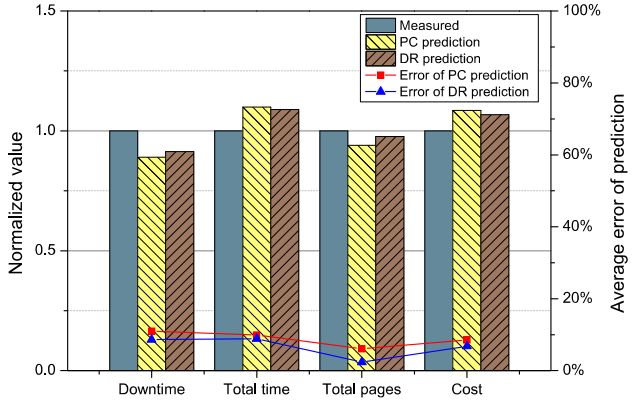
prediction error: the absolute value of the difference between the predicted value



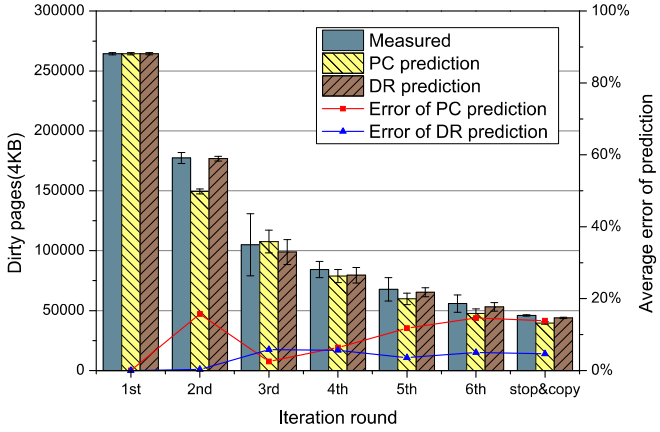
(a) Prediction error with static workload



(b) Prediction error with dynamic workload



(c) Detailed predicting information set for SPECjbb



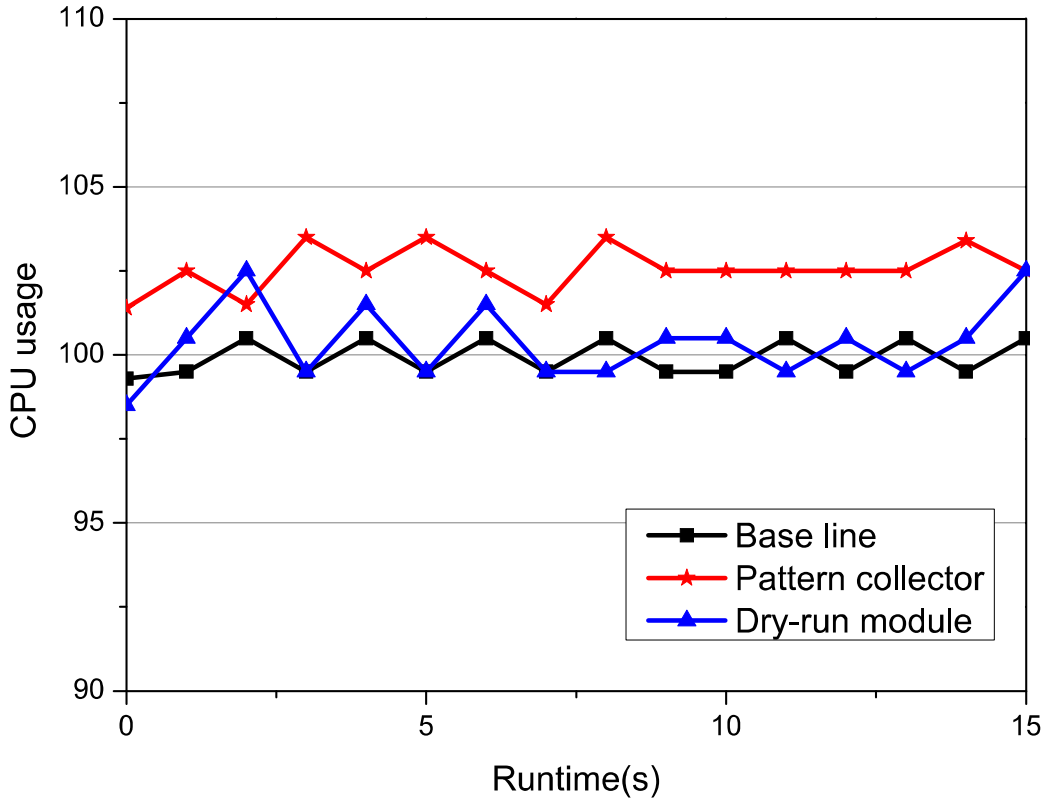
(d) Detailed predicting of dirty pages in pre-copy iteration for SPECjbb

Evaluation: Prediction Error

Name of prediction method	Accuracy
Base model [30]	81%
Simulation model [6]	90%
Refined model [30]	90.5%
Our PC prediction	91%
Our DR prediction	93.8%

Prediction methods' accuracy comparison

Evaluation: Overhead

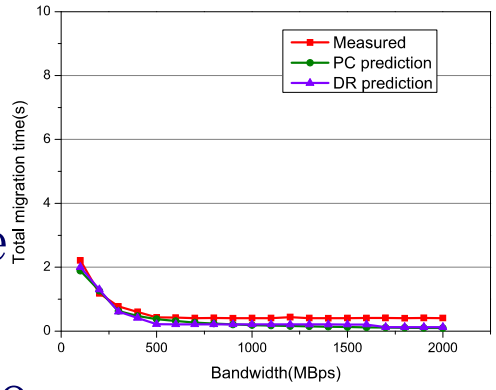


CPU overhead incurred by the PC and DR modules.

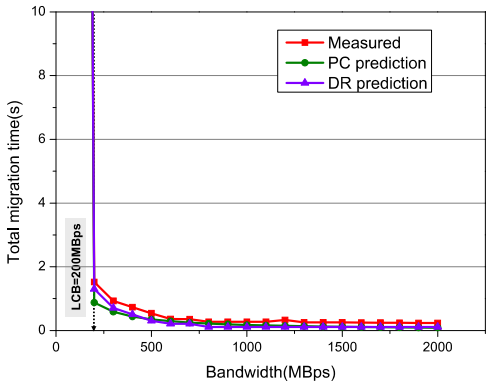
Discussion- Use Case

Case1: VM Migration Candidate Selection

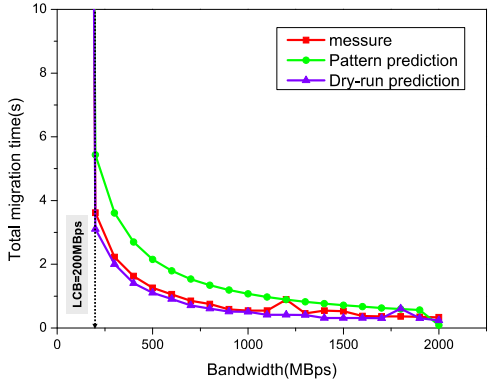
1. Feasible Candidate Selection:
2. Efficient Candidate Selection:



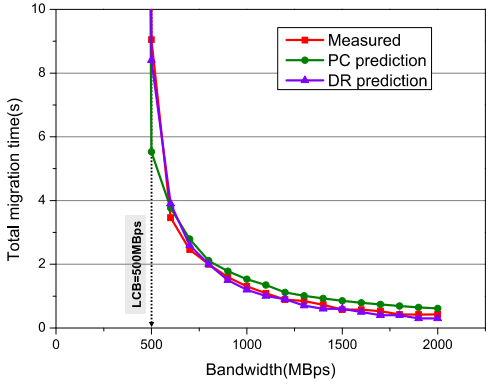
(a) Total Migration Time for Linux idle



(b) Total Migration Time for Apache Benchmark



(c) Total Migration Time for Sysbench-OLTP



(d) Total Migration Time for SPECjbb

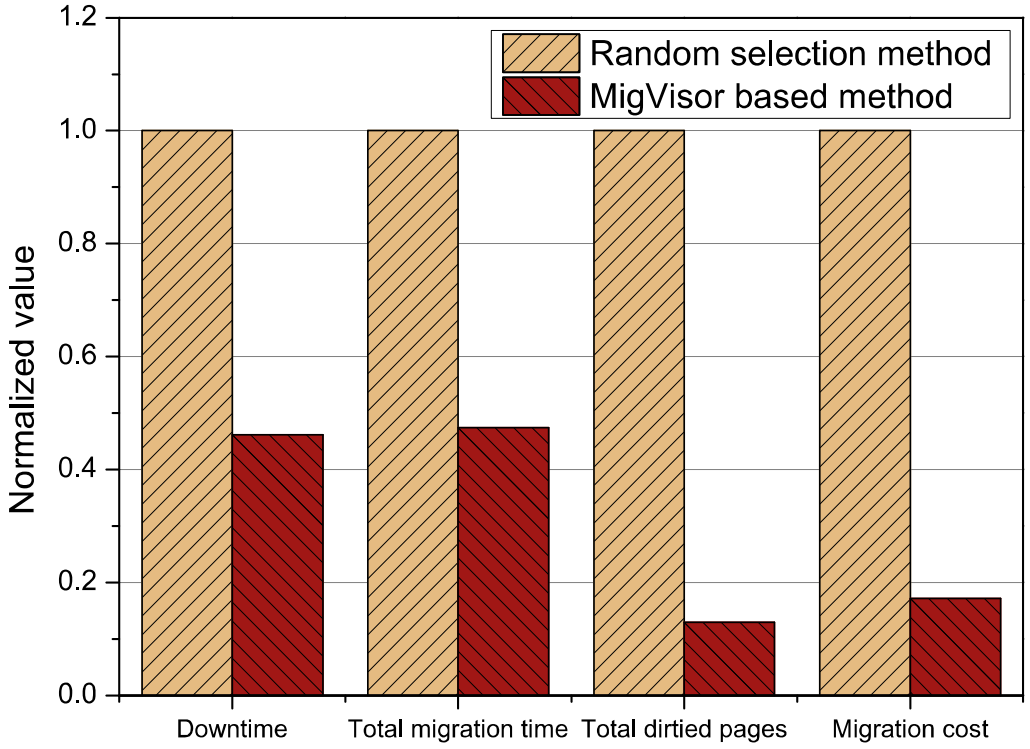
Predicted LCB for the Feasible Candidate Selection of Migvisor

(Least Convergent Bandwidth)

Cost Saving of Migvisor

VM Migration **Cost**: We adapt the migration cost model defined by Liu *et al* [30]:

$$C_i = a \cdot T_{sum} + b \cdot T_{downtime} + c \cdot P \qquad (11)$$



Cost saving by usage of the MigVisor based method for the efficient candidate selection

Plan

- Motivation & Problem
- System General Model and related work
- Design and Implementation
- Evaluation
- Conclusion

Conclusion

- Proposed MigVisor:
  - a VM migration behaviour prediction scheme *driven by a working-set pattern model*.
- Migvisor consists of two prediction methods:
  - PC and DR prediction
- MigVisor can accurately predict the *downtime, total migration time, and total number of dirtied pages*, as well as the *synthesized cost of the migration*.
- Migvisor can identify the minimum migration cost in a limited bandwidth scenario, and a *bandwidth larger than the LCB should be used for the migration*.