iCSI:

A Cloud Garbage VM Collector for Addressing Inactive VM with Machine Learning

In Kee Kim[+], Sai Zeng[*], Christopher Young[*], Jinho Hwang[*],

and Marty Humphrey[+]

[+]University of Virginia

[*]IBM T.J. Watson Research Center

Motivation

- **1** in **3** data center servers is a **zombie** (not producing any useful work)
  - Recent study from Stanford University (2015).

- That is translated into:

- **10** million comatose servers world wide
- **30** billion dollars in data center capital investment
- **40** percent electrical energy waste
- **∞** maintenance, software license, cooling cost..

Motivation (Cont'd)

- Why Zombie (Inactive) VMs are living in Data Centers?
  - VMs are cheaper to create, and easier to forget.
    - More common/critical in Private/Hybrid Clouds.
  - Financial owners may not be the actual user.
  - Many zombie VMs keep legacy installations and data for future use.
  - **Identifying active/inactive VMs with certainty is difficult with conventional tools.**

Challenges – Detecting Active/Inactive VMs)

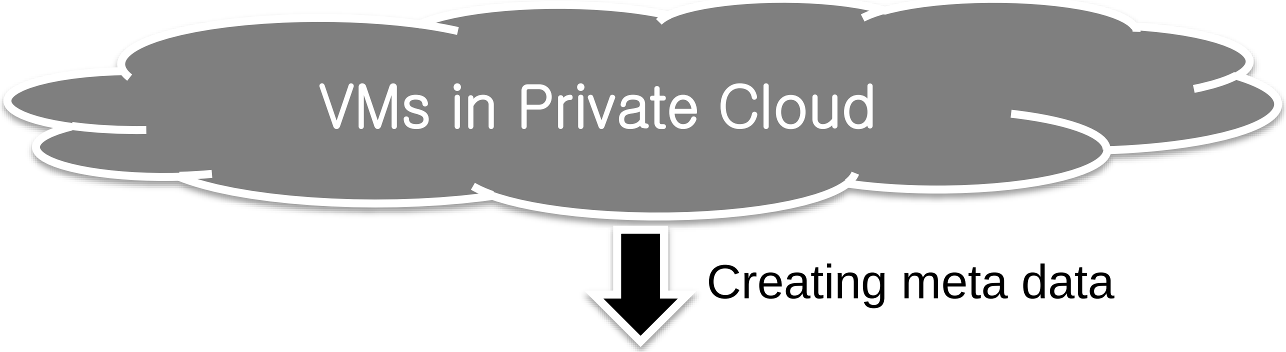
- Correlation between "**Resource Idleness**" and "**Requirement Idleness**" may exist, but not very reliable.
  - Inactive VMs can look "**active**"
    - Virus scan; Disk defrag; System update; Other background services.
    - Even worse: running applications that are not actually needed by users.
  - Active VMs can look "**inactive**"
    - Users are doing lightweight text editing.
    - Failover VMs that are idle most of the time, but required to be available at any time.

Approach:

iCSI – Inactive Cloud Server Identification System

Feature Selection for VM Identification

- 70 (Linux) VMs with Random Sampling.
- Ground-truths were provided by the actual users.
- Linux Primitive Commands are used:
  - `ps`, `netstat`, `last`, `ifconfig`, etc.
- Extract Information with Five Categories:



VMs in Private Cloud

Creating meta data

Process	Utilization	Login	Network	Others
…	…	…	…	…

Creating VM Metadata

Metadata	Description
Process	- Defined 25 classes of *significant* processes. - Ignoring kernel and management processes (e.g., patch update).
Utilization	- CPU/MEM usage of the significant processes.
Login	- Login frequency and duration. - Differentiate daytime/nighttime login.
Network	- Port # / State of TCP connections.
Others	- IP and Host information.

Correlation Analysis

- Tried to find strong features from metadata:

$$-r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

  - Failed to find (global) correlation with active / inactive VMs.

- However, there are strong correlated features based upon the purpose of VMs:

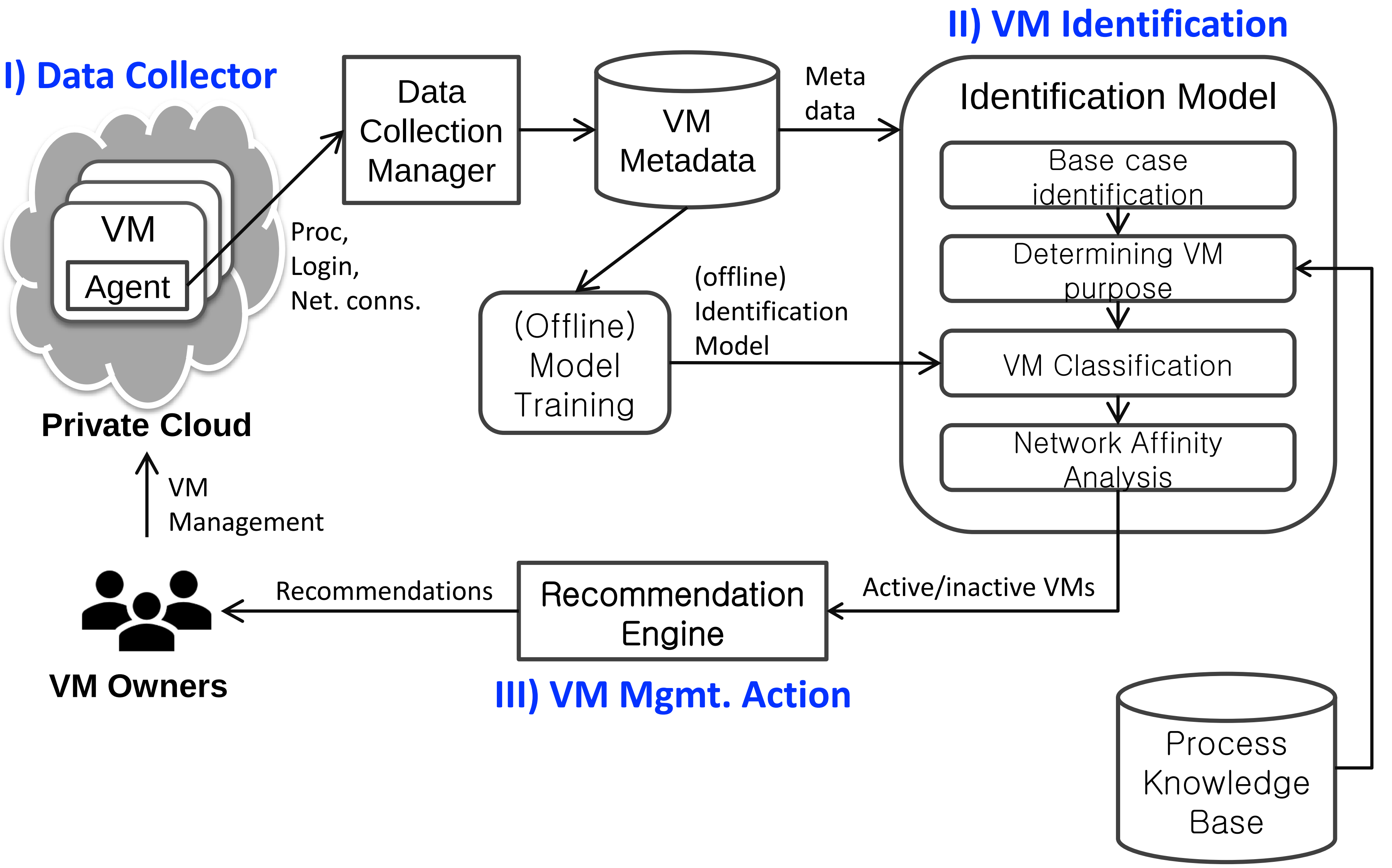
<Analytics>

Features	Correlation
%CPU of Significant Procs	0.95
%MEM of VMs	0.95
# of Important Open Ports	0.90
# of Established Conn.	0.97
Etc.	

<Development>

Features	Correlation
%CPU of Imp. Procs > 5%	0.72
%MEM of Imp Procs > 5%	0.73
# of Logins > 15	0.85
Daytime Login > 24 hrs	0.91
Etc.	

iCSI System Design (Overview)



I) Data Collector

VM

Agent

Private Cloud

Proc,
Login,
Net. conns.

VM
Management

VM Owners

Data Collection Manager

VM Metadata

Meta
data

(Offline) Model Training

(offline)
Identification
Model

II) VM Identification

Identification Model

Base case identification

Determining VM purpose

VM Classification

Network Affinity Analysis

Process Knowledge Base

Active/inactive VMs

Recommendation Engine

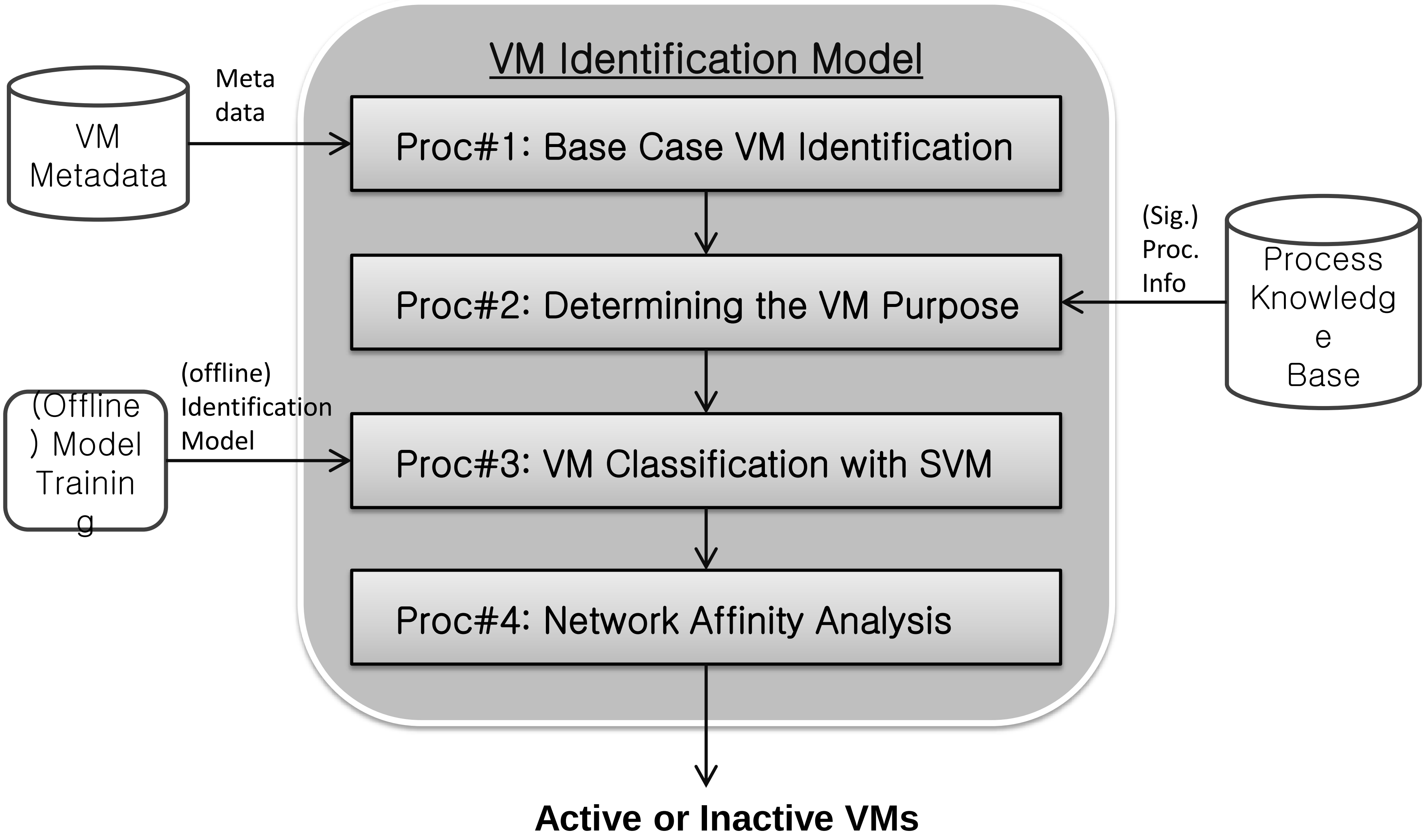
Recommendations

III) VM Mgmt. Action

Lightweight Data Collector

- A **bash script** is deployed to VMs.
  - This script should not mess up production services.
    - Gradually deployed it from a small-scale data center to large-scale data centers.
  - Executed in every 4 hours.
  - Only collects 50KB data and sends it to the manager via cURL.
  - Deployed via an IBM Data Center Management tool.
    - Can be replaced with chef, puppet, and others.

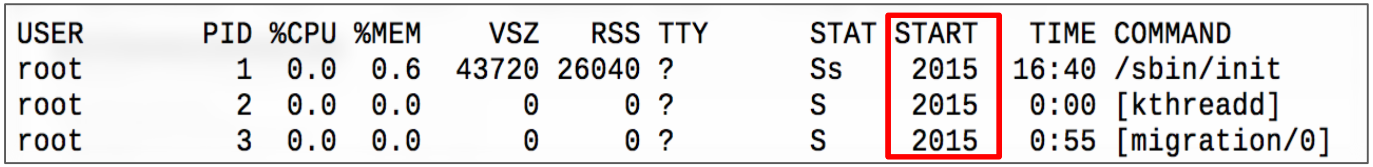
VM Identification


VM Identification Model
VM Metadata
Meta data
Proc#1: Base Case VM Identification
Proc#2: Determining the VM Purpose
(Sig.) Proc. Info
Process Knowledge Base
(Offline) Model Training
(offline) Identification Model
Proc#3: VM Classification with SVM
Proc#4: Network Affinity Analysis
Active or Inactive VMs

Proc#1: Base Case Identification

- Four Rules based on “**explicit**” usage pattern.

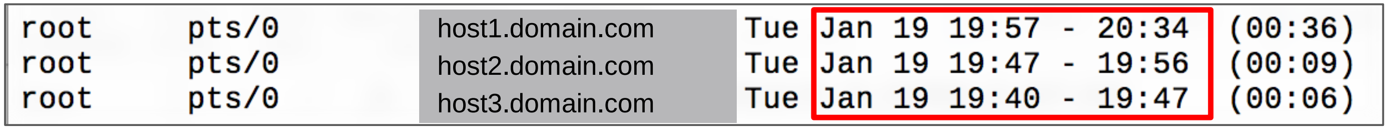
1. Long Running VM Instance:



```
USER       PID %CPU %MEM   VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.6 43720 26040 ?        Ss    2015  16:40 /sbin/init
root         2  0.0  0.0     0     0 ?        S     2015   0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S     2015   0:55 [migration/0]
```

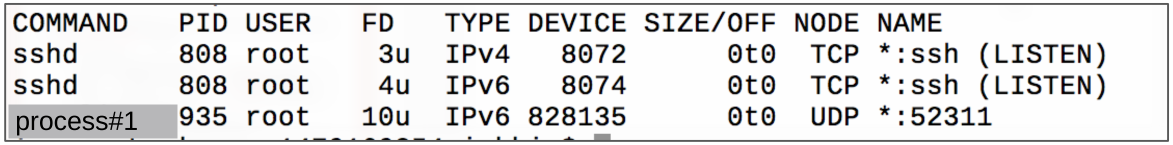
2. No Significant Processes:
   - Based on 25 classes for significant user processes.

3. No Login Activity over last 3 months:



```
root      pts/0      host1.domain.com   Tue Jan 19 19:57 - 20:34  (00:36)
root      pts/0      host2.domain.com   Tue Jan 19 19:47 - 19:56  (00:09)
root      pts/0      host3.domain.com   Tue Jan 19 19:40 - 19:47  (00:06)
```

4. No Established Connection with other VMs during data collection period.



```
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd       808 root    3u  IPv4   8072      0t0  TCP *:ssh (LISTEN)
sshd       808 root    4u  IPv6   8074      0t0  TCP *:ssh (LISTEN)
process#1  935 root   10u  IPv6 828135      0t0  UDP *:52311
```

Listen ports and Mgmt ports are not considered.

Proc#2: Determining the Purpose of VMs

- A key to find strong correlated factors for Active/Inactive VM Identification.
- Idea: the purpose can be determined by “running process”
  - A VM with MySQL can be used for Storage, Development, Test,…

$$\begin{aligned} \boldsymbol{Input} &: process_i \\ \boldsymbol{Output} &: \{purpose_1 : w_1, purpose_2 : w_2, ..., purpose_n : w_n\} \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} \boldsymbol{Input} &: MySQL \\ \boldsymbol{Output} &: \{\text{“}Storage\text{”} : 0.7, \text{“}Development\text{”} : 0.2, \text{“}Test\text{”} : 0.1\} \end{aligned}$$

Determined with user feedback

Proc#3: Active/Inactive VM Classification

- Idea: Using Linear SVM (Support Vector Machine) with different (specified) correlated features.
- **Linear SVM**:
  - An optimal margin-based classifier with linear kernel.
  - Linear SVM tries to find a small number data points that separate all data points of two classes with a hyperplane.
  - **Use specific correlated features according to the purpose of VMs.**

Server Purpose	Correlated Features
Analytics	%CPU, %MEM, #OpenPorts
DevOps	#SigProcs, %CPU_SigProcs, %MEM_SigProcs, #EstConns
Development	#LoginFreq (Daytime), AvgLoginHr, #SSH/VNCs, #UserActivityProcs
. . .	

Proc#3: Active/Inactive VM Classification

- Addressing the multiple purposes for VMs.
  - Run SVM classifier multiple times with different weight.
  - Ensemble of all classification results.
    - Classification Result: $\psi \in \{0, 1\}$
    - Weight for a Purpose: $\omega \in \{0, 1\}$

$$\boxed{Classification\ Result = \frac{\sum_{i=1}^{n} \omega_i \times \psi_i}{\sum_{i=1}^{n} \omega_i}}$$

Proc#4: Network Affinity Analysis

- Idea: If an **active** VM-(A) depends on / or is connected with VM-(B), VM-(B) must be **active**.
- This rule works very well for cluster configurations:
  - Linear SVM classifier can successfully classify Hadoop/Mesos master as “**active**” but, not for slave nodes.

Recommendation Policies

- 0 ≤ VM Identification Result ≤ 1 (0: Inactive, 1: Active)

Recommendation	Trigger Conditions
No Action	• Active VMs (Classification Result > 0.5)
Terminating VM	• Classification Result == 0
Suspending VM	• 0 < Classification Result ≤ 0.5
Resizing VM	• 0 < Classification Result ≤ 0.5 • Significant Processes are running on the VM

- More sophisticated policies can be designed with data center infrastructure.

Performance Evaluation of iCSI

Evaluation Setup

- Evaluation Pool:
  - 750 VMs on IBM Research Cloud Infrastructure. (3 data centers)
  - Ground Truth: User Feedbacks

- Evaluation Criteria:
  1. Classification Accuracy.
     - Goal: Minimizing **False Negative** Errors
       - Active VMs are **incorrectly** identified as Inactive.
     - Validated with k-fold CV.
  2. VM Cost Saving
  3. VM Utilization Improvement.

- Baselines:
  - Pleco (CNSM 2016) and Garbo (SoCC 2015)

iCSI Identification Accuracy

# Testset	# Identified as Active VM	Recall
750	**460 (63%)**	**0.90**

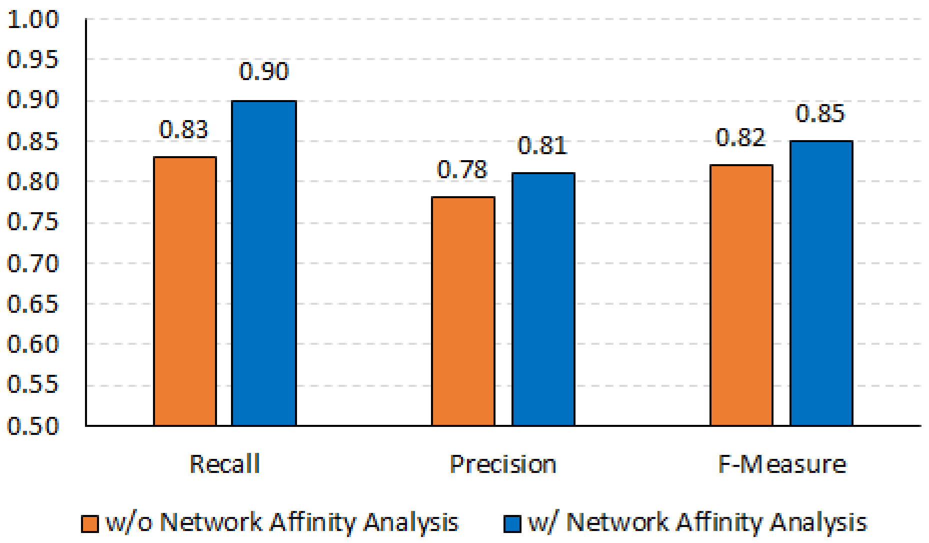
$$\textbf{Recall} = \frac{\overbrace{\textbf{True Positive}}^{\text{Classified Active as Active}}}{\textbf{True Positive} + \underbrace{\textbf{False Negative}}_{\text{Classified Active as “Inactive”}}}$$

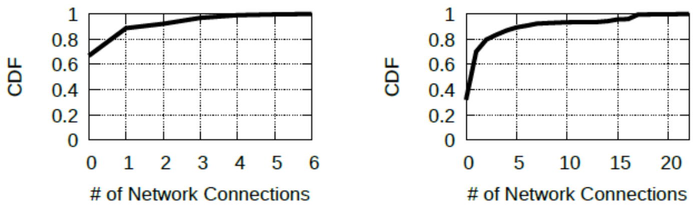
iCSI Classification Accuracy

- Accuracy Comparison with Baselines:

	Recall	Precision	F−Measure
Pleco	0.75	0.69	0.72
Garbo	0.70	0.67	0.68
iCSI	0.90	0.81	0.85

Improve with Network Affinity Analysis





(a) Inactive VMs.

(b) Active VMs.

Fig. 5. CDF of External Network Connections of VMs.

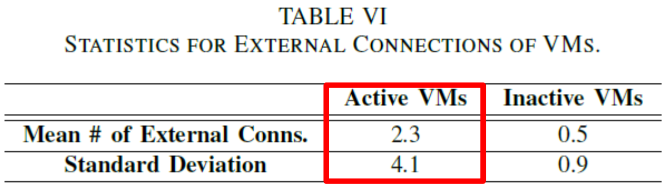
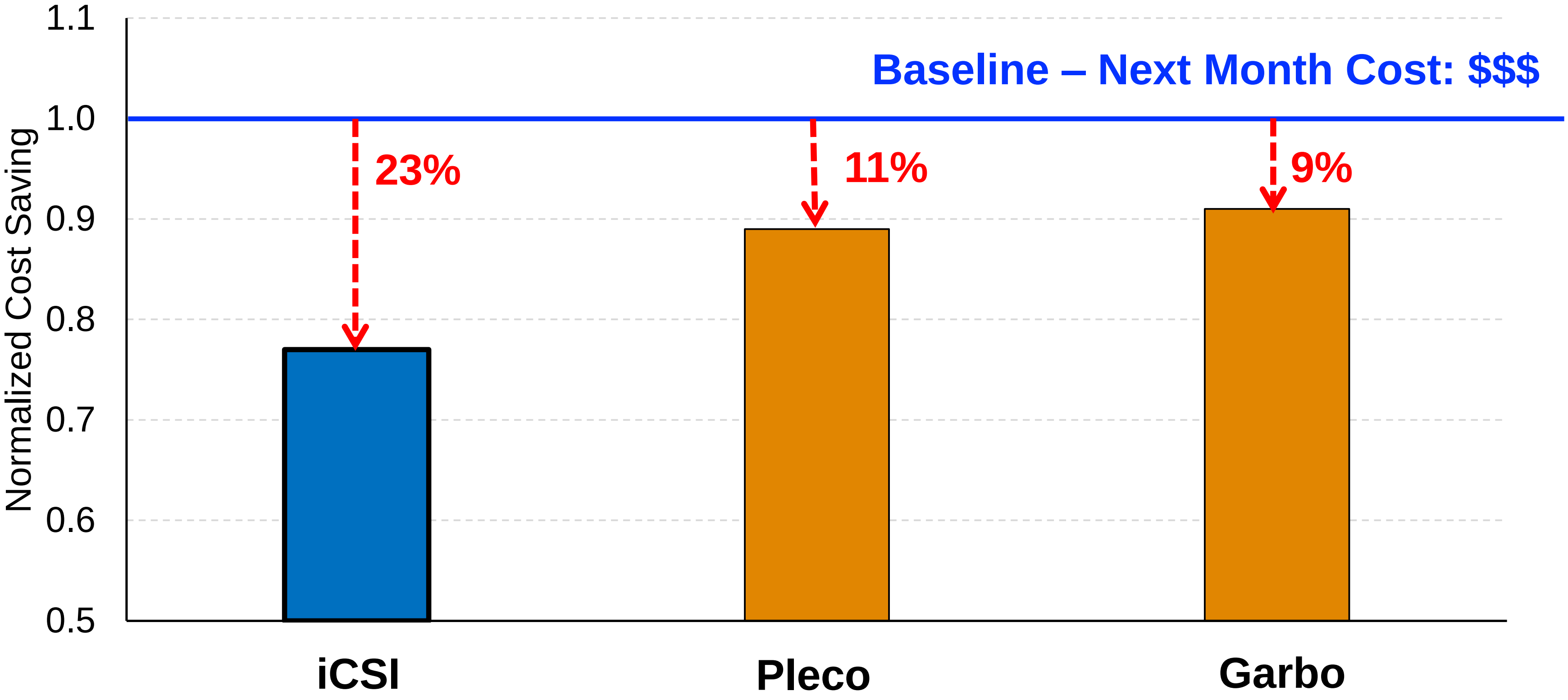


TABLE VI
STATISTICS FOR EXTERNAL CONNECTIONS OF VMS.

	Active VMs	Inactive VMs
Mean # of External Conns.	2.3	0.5
Standard Deviation	4.1	0.9

Cloud Cost Saving

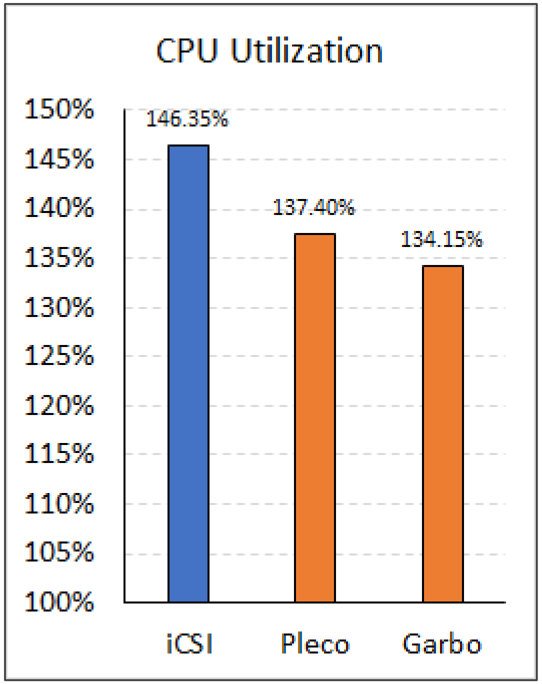
- $Penalty\ Cost\ =\ \sum_{i=1}^{n}(\omega_i \sum_{j=1}^{m} cost_{vm_j})$
- $Total\ Cost\ =\ Cost_{active_{vm}} + PenaltyCost$

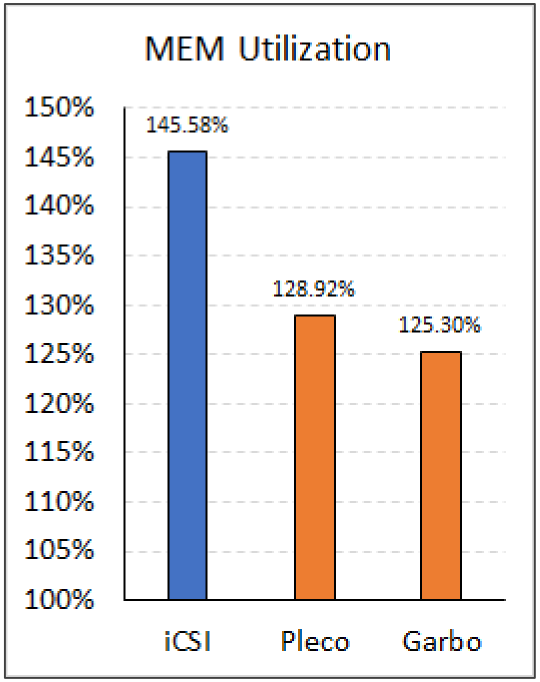


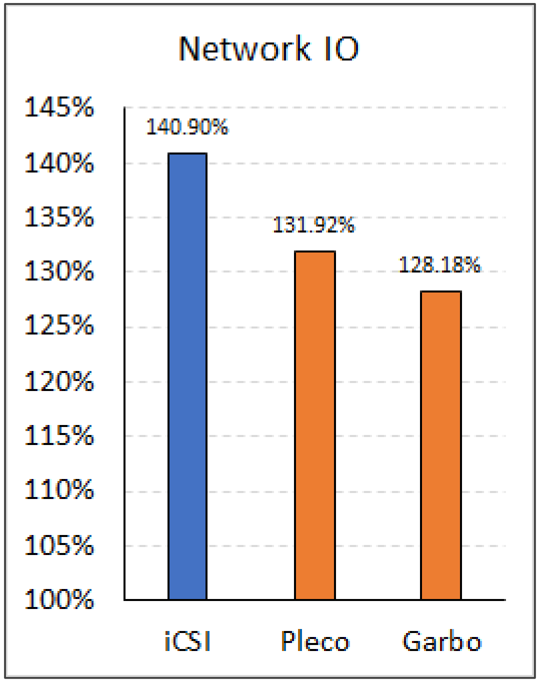
VM Utilization Improvement

- Average Utilization Improvement

	iCSI	Pleco	Garbo
Average Improvement of VM Utilization	46%	31%	29%







Conclusion

- We have created iCSI:
  - A lightweight approach – only collects few kbytes data from each VM.
  - We have found specific correlated features according to the purpose of VMs on the production clouds.
    - Linear SVM classifier directly uses the specific correlation features.
  - VM identification mechanism is composed of heuristics (rule-based) and machine learning (Linear SVM)
  - iCSI has over 90% of recall to identify active/inactive VMs.
  - For the future work, ***dealing with privacy regulations*** will be an critical issue.

Questions?

Thank you!

Support – Accuracy Metrics

- False Negative and False Positive:

		Identification Result	
		Active	Inactive
Truth	Active	TP: Active VMs are correctly identified as active.	FN: Active VMs are incorrectly identified as inactive.
	Inactive	FP: Inactive VMs are incorrectly identified as active.	TN: Inactive VMs are correctly identified as inactive.

- Accuracy Metrics

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}$$

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Future Works

- Improving iCSI System:
  - Current version is focused on managing Linux VMs:
    - Need to be expanded to Windows VMs.
    - Windows VMs covers large portion of VMs in private clouds (e.g. legacy applications)
  - Need a better approach for determining the purpose of VMs.
  - Need to be verified with larger scale data centers or real production clouds.
- Dealing with Regulations and Privacy Issues.
  - We could only collect **U.S. Owned VMs** for this work!

State-of-the-art

	Pleco (CNSM 2016)	Garbo (SoCC 2015)	Janitor Monkey (Netflix 2013)
Desc.	**Reference Model (ALDM) + Decision Tree**	**Graph Theory + “mark and swap”**	**Aging of VM + User Feedback**
Target Platform	Private Clouds	Amazon Web Services	Amazon Web Services
Cons	Expensive Data Collection. App. Dependent. Static Connection.	Only Considering Network Connectivity.	Depending on user feedback. Not fully automated system.