WESTERN SYDNEY
UNIVERSITY

A Learning Automata based Dynamic Resource Provisioning in Cloud Computing Environments

Authors:
Hamid Reza Qavami
Shahram Jamali
Mohammad Kazem Akbari
Bahman Javadi

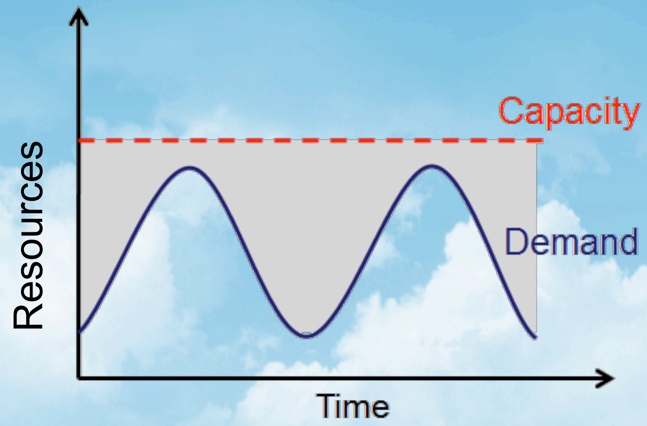
Presenting By:
Dr .Bahman Javadi
Western Sydney University, Australia

Resource Provisioning Necessity

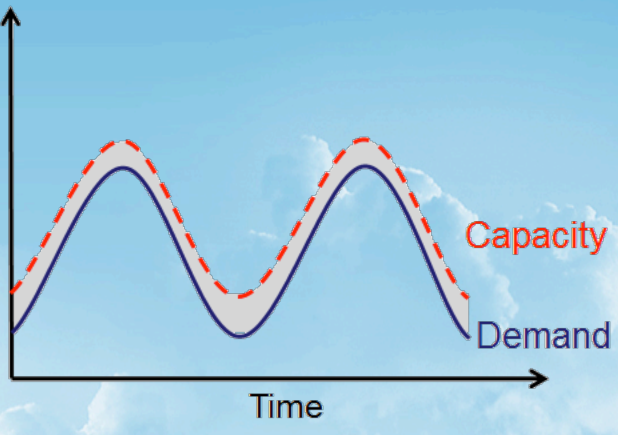
- Cloud Resources are:
  - When ever
  - Where ever
  - Resource Pool

- But, let's put on the pink glasses
  - There is not such infinite resource pool!
  - And also when and where ever available
  - Costs, Energy, Environment …

Dynamic Provisioning
The Efficient Provisioning



Static Solution



Cloud based solution

- Energy
- Utilization
- Cost

Problem Statement

- Minimizing used VMs for application
  - For
    - Cost optimization
    - Utilization

$$\min\left(\sum_{n=1}^{MaxVM} VMlist_n^W\right)$$

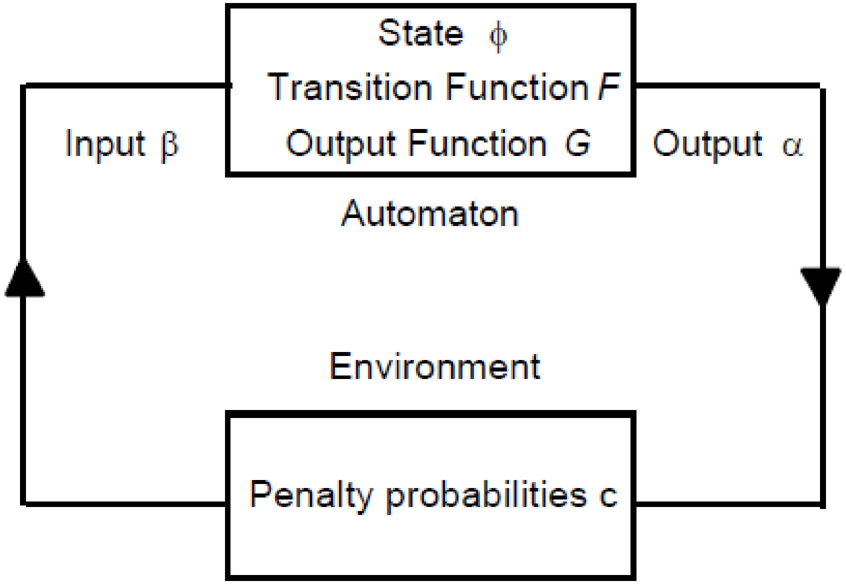
  - While
    - Keeping QoS and SLA parameters

$$\sum_{1}^{MaxOnlineVMs} MIPS_{VirtualMachines} > \sum_{1}^{CurrentCloudLetNumber} MI_{CloudLets}$$

Dynamic Resource Provisioning

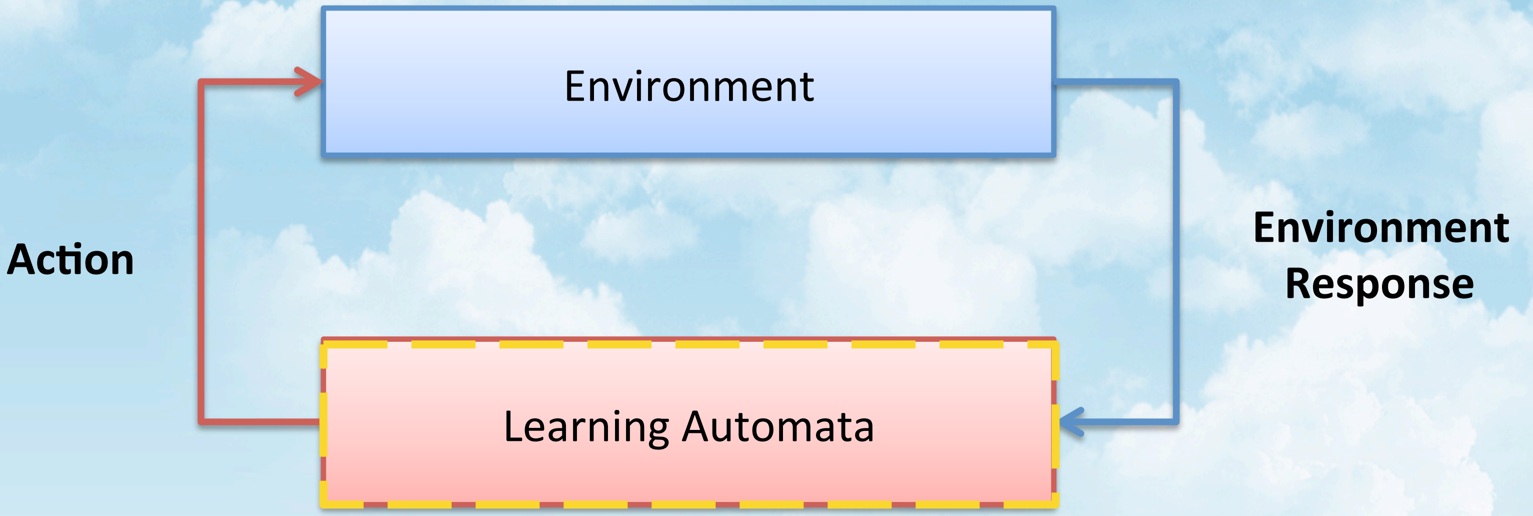
- ***Proposed machine learning approach***:
  - **Dynamic Resource Provisioning**

    Using Learning Automata



Learning Automata

$$LA \equiv \{\underline{\alpha}, \underline{\beta}, \underline{p, T}\}$$



Defining States

- 3 outputs (states) for L.A
  - Resource increase
    - More resources would be needed
  - Resource decrease
    - Less resources would be needed
  - No changes
    - Current resources would be just enough

$$\alpha \equiv \{\alpha_1, \alpha_2, \alpha_3\}$$

Feedback

- Average VMs utilization as the feedback
  - It is simply observable from VMM or the VM itself
  - Less monitoring overhead
  - Informative
    - Give us a good status about resources comparing to load

$$VMs\ Avg.\ Utilization^{w} = \frac{\sum_{i=1}^{\substack{MaxOnline \\ VMs}} VM_{i}^{w} Utilization}{MaxOnlineVMs}$$

Learning Algorithm Responding to Last (ith) Action

- If (c_i = 0) then

favorable response

  - Reward P_i & Punish the others

$$p_i(n+1) = p_i(n) + a \times (1 - p_i(n))$$

$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j, \quad j \neq i$$

- If (c_i = 1) then

unfavorable response

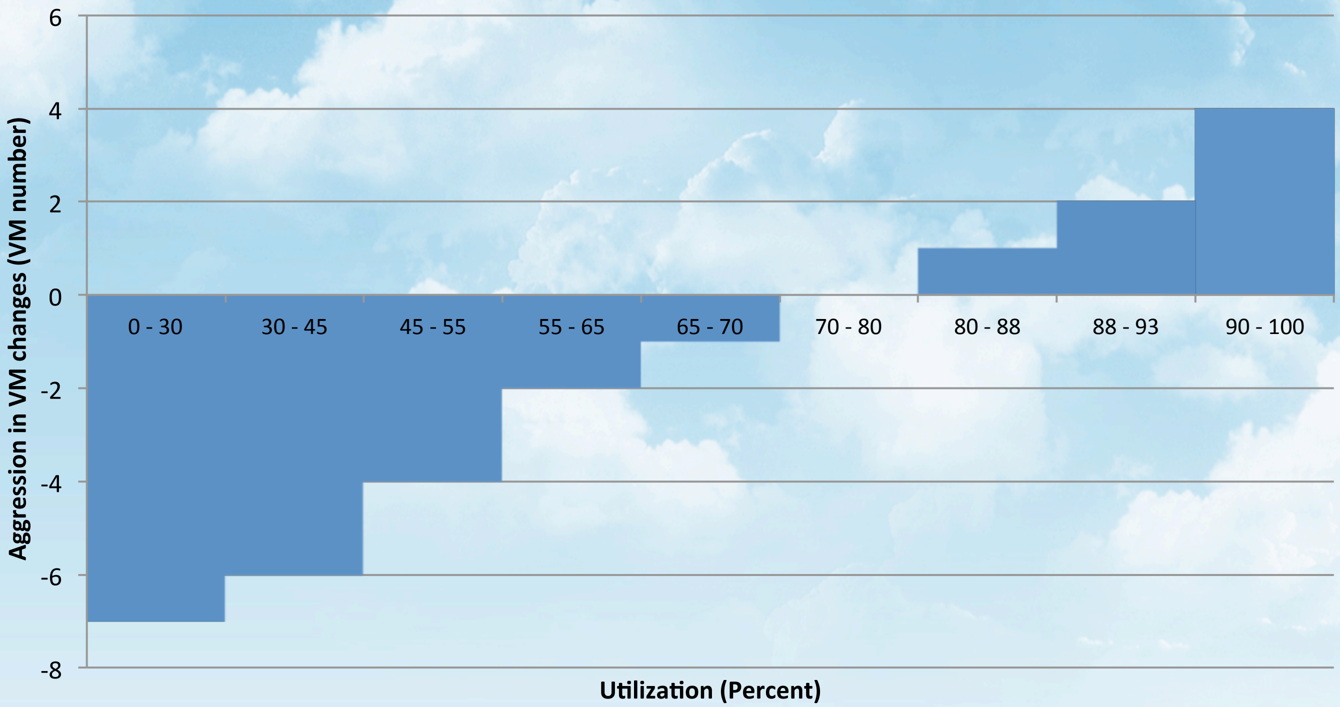
  - Punish P_i & Reward the others

$$p_i(n+1) = (1-a)p_i(n)$$

$$p_j(n+1) = \frac{a}{r-1} + (1-a)p_j(n) \qquad \forall j, \quad j \neq i$$

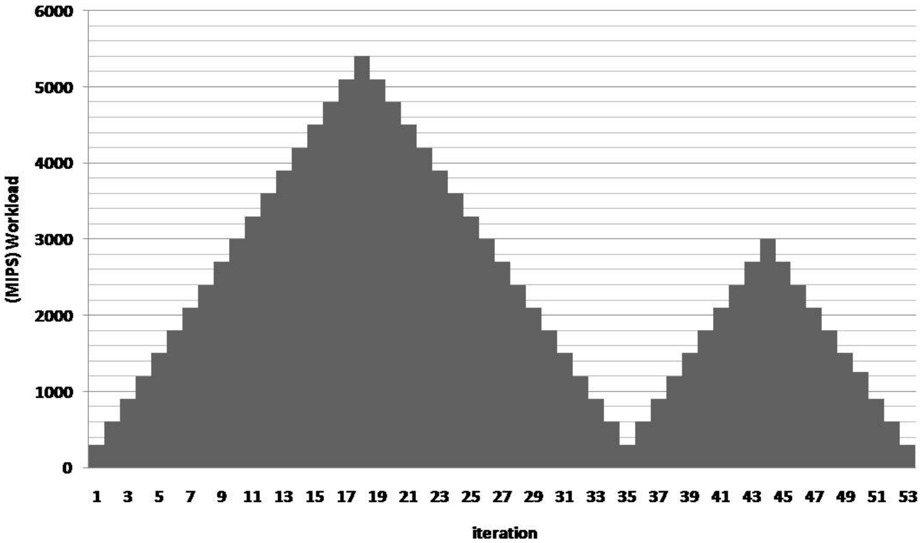
Novel Intensity Control System

- Slow convergence rate in approaches
  - Not enough states for decision
  - More states, make it slower

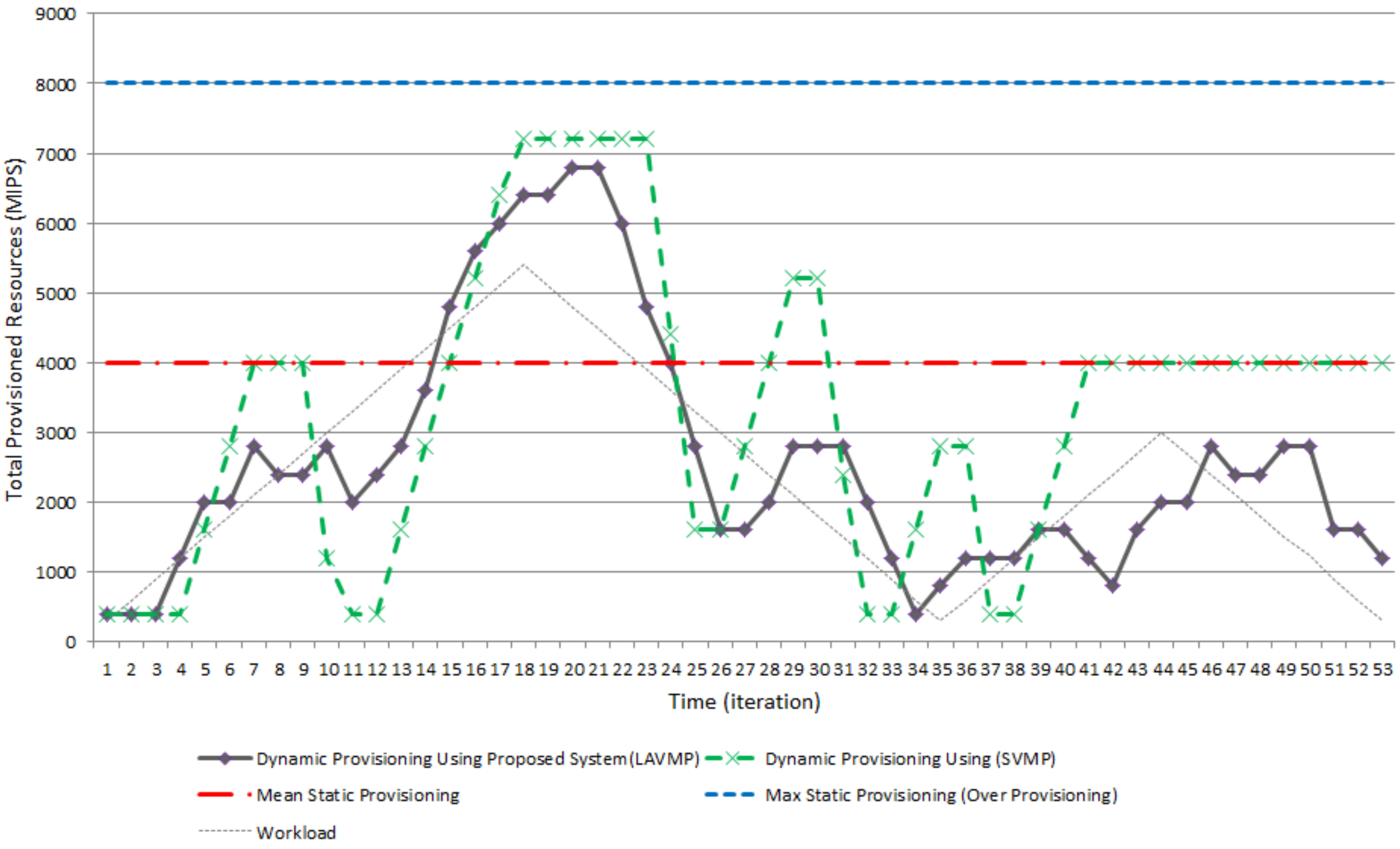


Experimental Setup

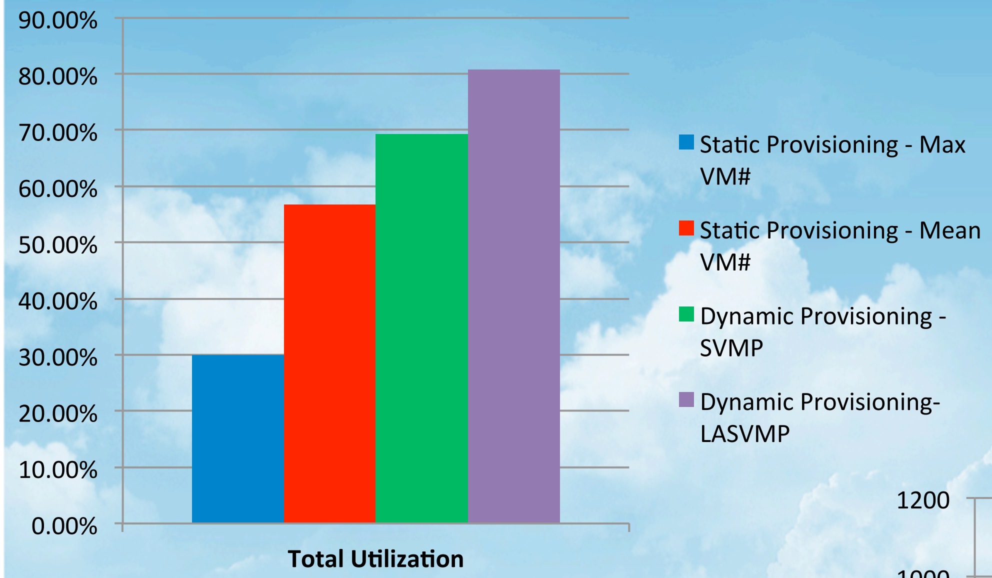
- A scenario in 4 different experiments (using CloudSim):
  - CPU intensive Workload
  - Maximum VM number
    - 20
  - VM processing Core(s)
    - 1 core
  - Core processing power
    - 400 MIPS
  - VM RAM
    - 512 MB

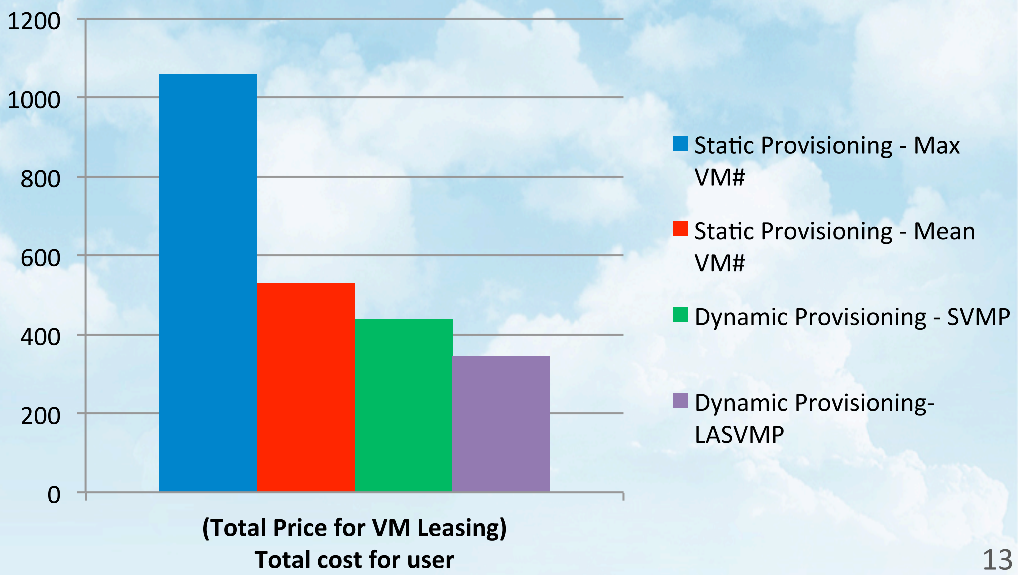


Experimental Results



Higher Utilization, Lower Cost

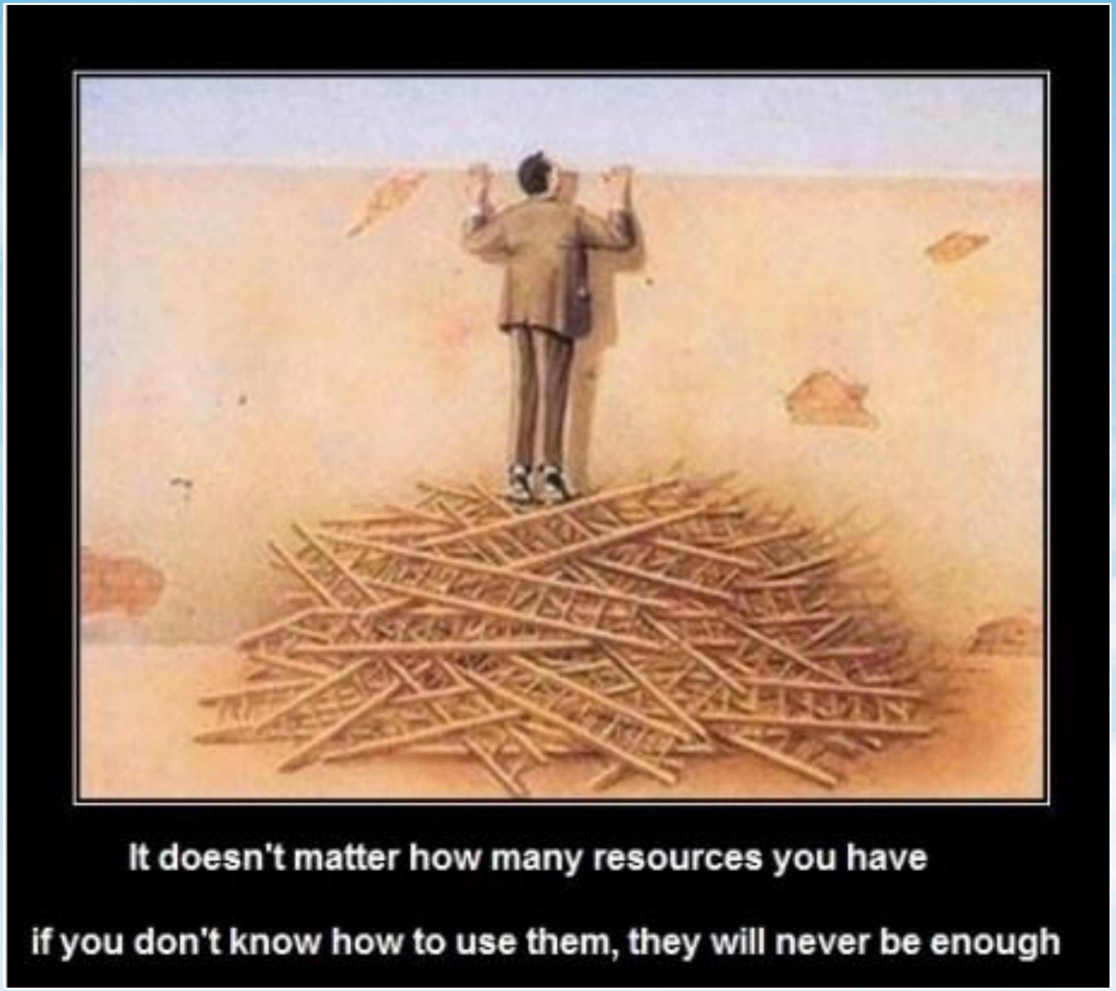




Conclusions

Defining Resource Provisioning problem	In Application layer
	Cost & QoS aware
Implementing & developing	Dynamic VM provisioning ability in Cloudsim
Our Solution for defined problem	Dynamic Resource Provisioning
	Using Learning Automata (LAVMP)
Covering the convergence lack with	Novel aggression control system
For utilizing VMs	A dispatcher (Job scheduler)
Achievements	Cost is reduced
	While QoS parameters are considered and improved

Thank You For Your Attention


It doesn't matter how many resources you have
if you don't know how to use them, they will never be enough