An Availability-aware Approach to Resource Placement of Dynamic Scaling in Clouds

Wenting Wang, Haopeng Chen, Xi Chen
Shanghai Jiaotong University, China

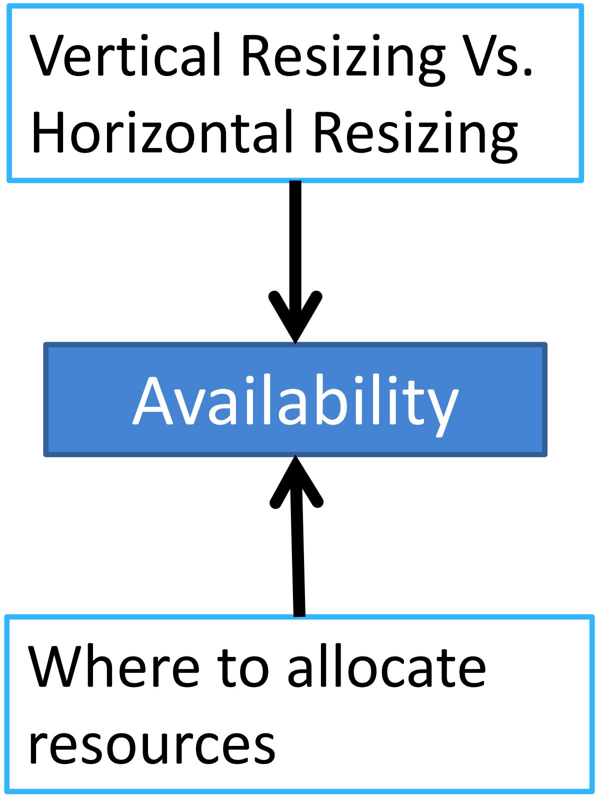
2012-06-24

Agenda

- Problem Statement
- Modeling and Approach
- Evaluation

Problem Statement

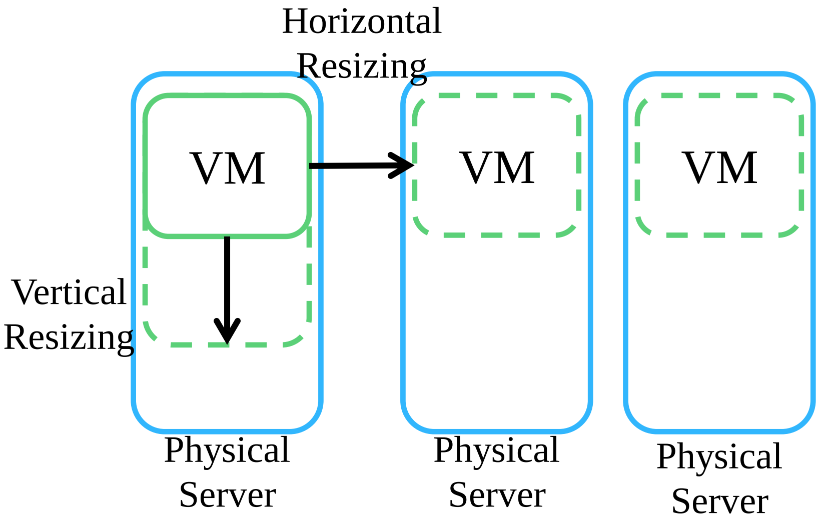
- Two Problems in scaling resources

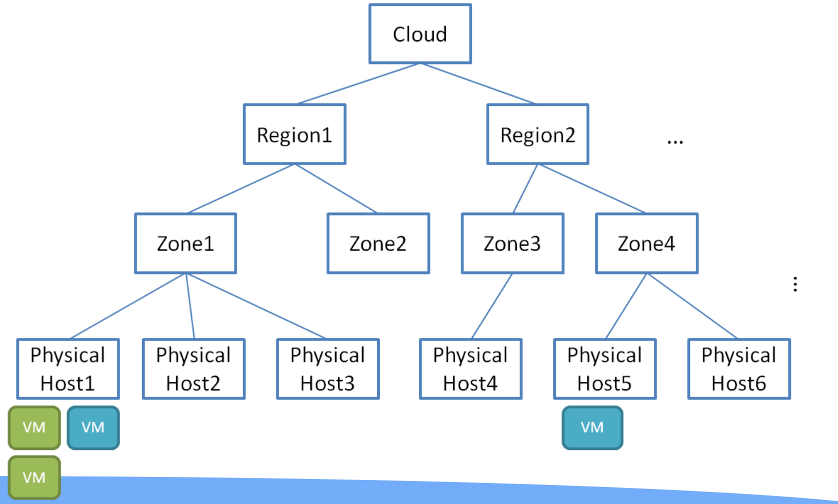


Vertical Resizing Vs. Horizontal Resizing

Availability

Where to allocate resources



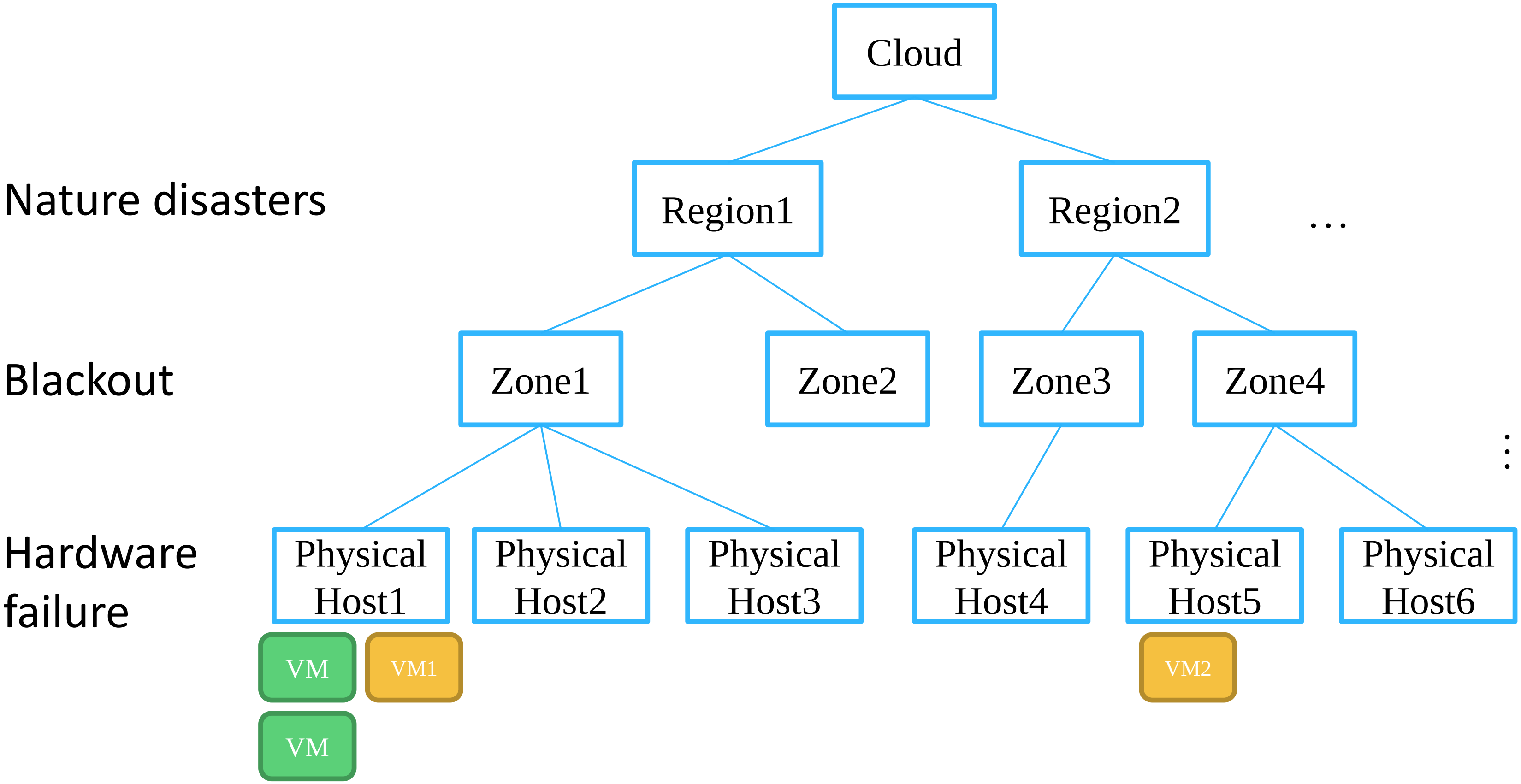


Agenda

- Problem Statement
- Modeling and Approach
- Evaluation

Modeling and Approach

- Availability Modeling



Nature disasters

Blackout

Hardware failure

Cloud

Region1

Region2

…

Zone1

Zone2

Zone3

Zone4

⋮

Physical Host1

Physical Host2

Physical Host3

Physical Host4

Physical Host5

Physical Host6

VM

VM1

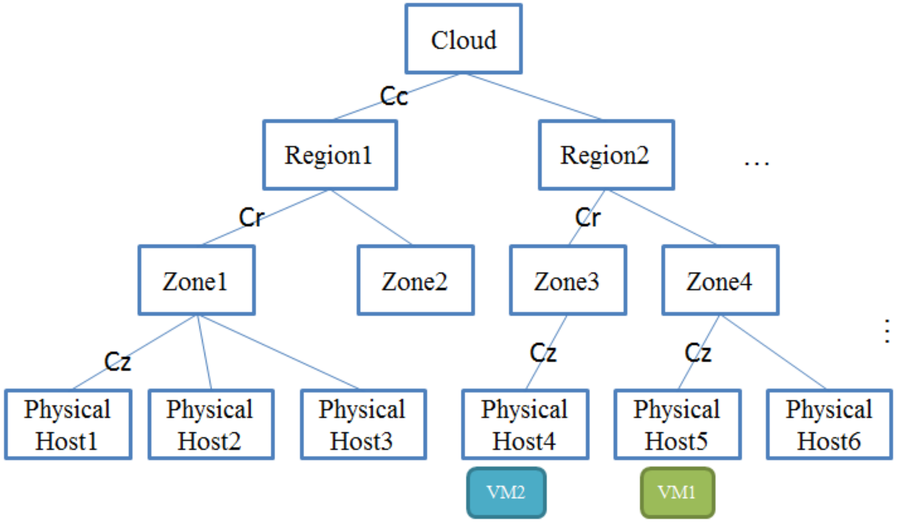
VM

VM2

Modeling and Approach

- The Availability of One VM

$$P_i(VM) = P_i \prod_{j \in PP(i)} P_j$$



- The Failure Probability of Two VMs

$$\left(\overline{P_u(VM)}\right) \bigwedge \left(\overline{P_v(VM)}\right) = \overline{\prod_{n \in C(u,v)} P_n}$$

$$+ \prod_{n \in C(u,v)} P_n * \left(\overline{\prod_{x \in N(u), x \notin C(u,v)} P_x}\right)\left(\overline{\prod_{y \in N(v), y \notin C(u,v)} P_y}\right)$$

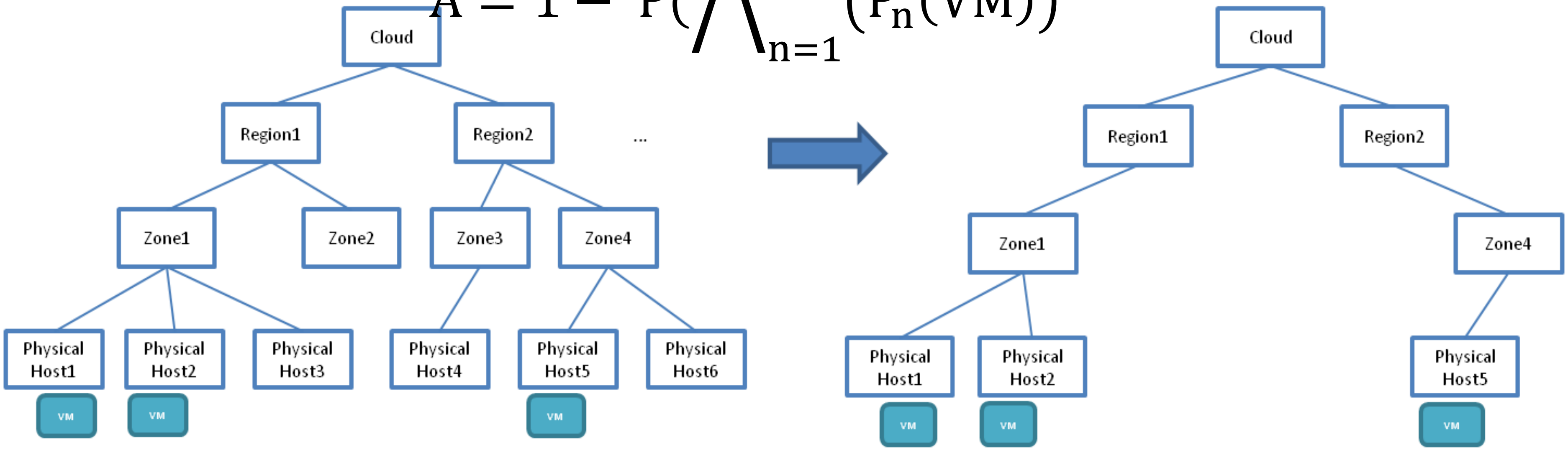
Modeling and Approach

- The Failure Probability of multiple VMs

$$\overline{P'_m} = \overline{P_m} + P_m \prod_{n\in \mathrm{children}(m)} \overline{P'_n}$$

- ➢ The application availability(denoted as A) is based on the sub-tree generated by multiple VMs:

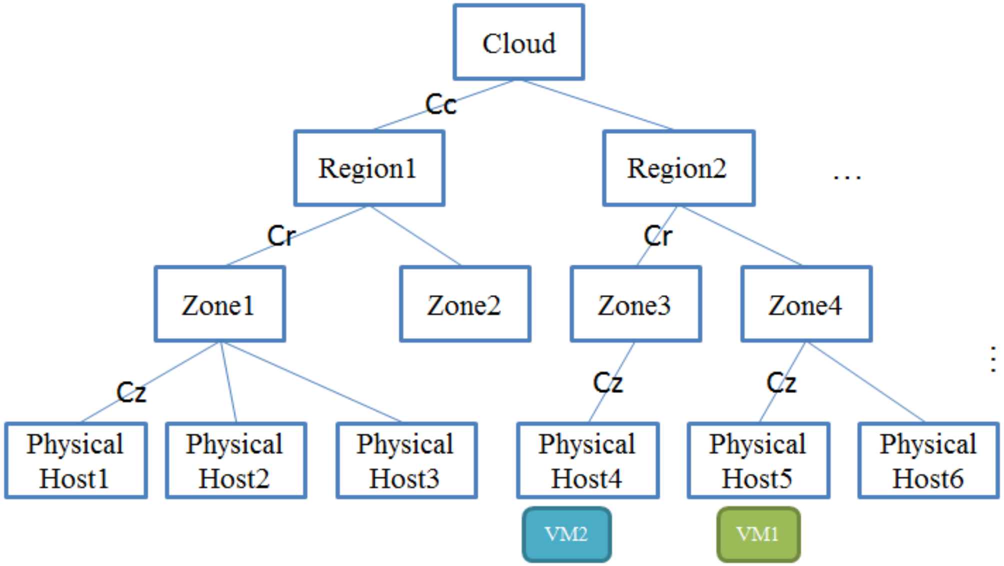
$$A = 1 - P(\bigwedge_{n=1}^{k}(\overline{P_n(VM)}))$$



Modeling and Approach

REliable, INtelligent & Scalable Systems

- Communication Cost Modeling
  - Let cc(v_1,v_2) donates the communication cost between VM_1 and VM_2.
  - Then the communication cost from one VM v to the other VMs in an application (where S is the set of all VMs composing the application) is $cc(v, S - \{v\}) = \sum_{x!=v} cc(v, x)$



Modeling and Approach

Input: **Quantity** denotes demanded unitized resource quantity need to be resized up or down

Input: Scale=up/down shows scaling flag

Input: relocatedTimes is the max times of relocation

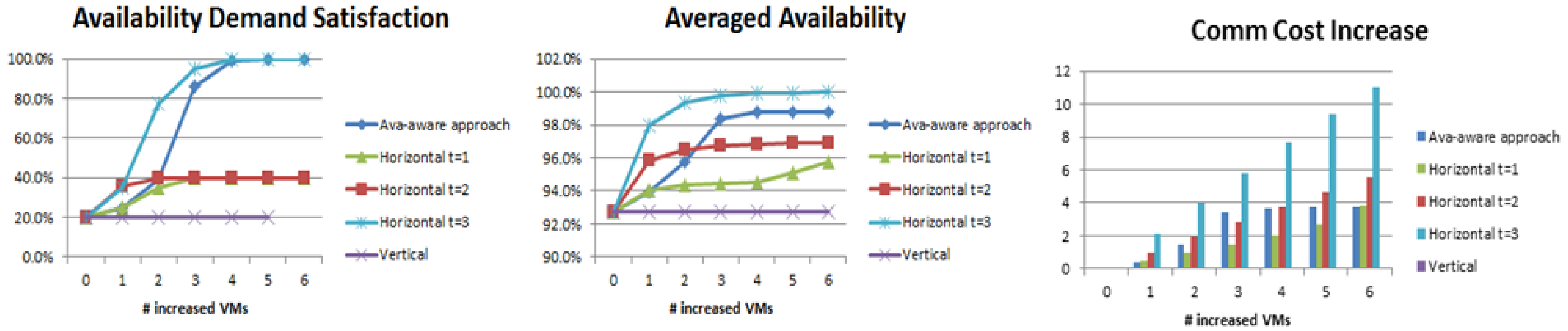
```
Ac=calculateAvailability();
t =1;
For(k=1;k<=Quantity;k++){
 If(scale == up){
    //the current availability is met
    If(Ac>=Ar){
      VerticalResizeUp(S, 1);
    }
    Else{// Ac<Ar
      HorizontalResizeUp(S, 1, t);
      t++;
    }
 }
 Else{ //scale down
    VerticalResizeDown(S, 1);
 }
 Ac=calculateAvailability();
} //end for
while(Ac<Ar && relocatedTimes >0){
  //rebalance overall application
  Relocate(S);
  Ac=calculateAvailability();
  relocatedTimes --;
}
```

Agenda

- Problem Statement
- Modeling and Approach
- Evaluation

Simulation

- Evaluation of Availability Model

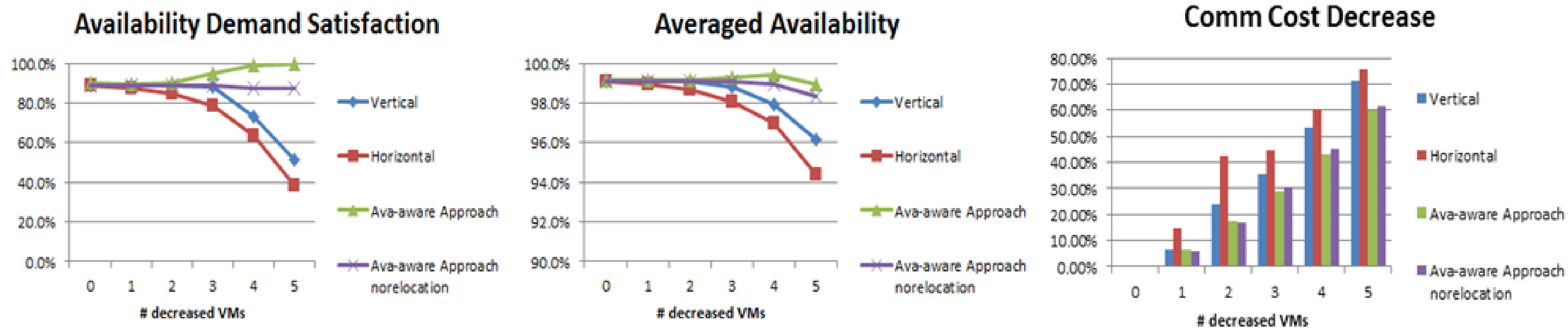


(a)Demand satisfaction

(b) Averaged availability

(c)Communication cost Increase ratio

Figure3 availability enhancement and performance change when scaling up



(a)Demand satisfaction

(b) Averaged availability

(c)Communication cost Decrease ratio

Figure4 availability declination and performance change when scaling down

- Q & A
- Thank you~