Verification of Cloud Systems using BigMC

Under the guidance of Prof. Dharmendra
Kumar Yadav
(MNNIT Allahabad)

Presented By

Amit Jain	(20164056)
Ankit Chaudhary	(20164037)
Lokesh Singh	(20164068)
Arun Verma	(20164058)

Objective

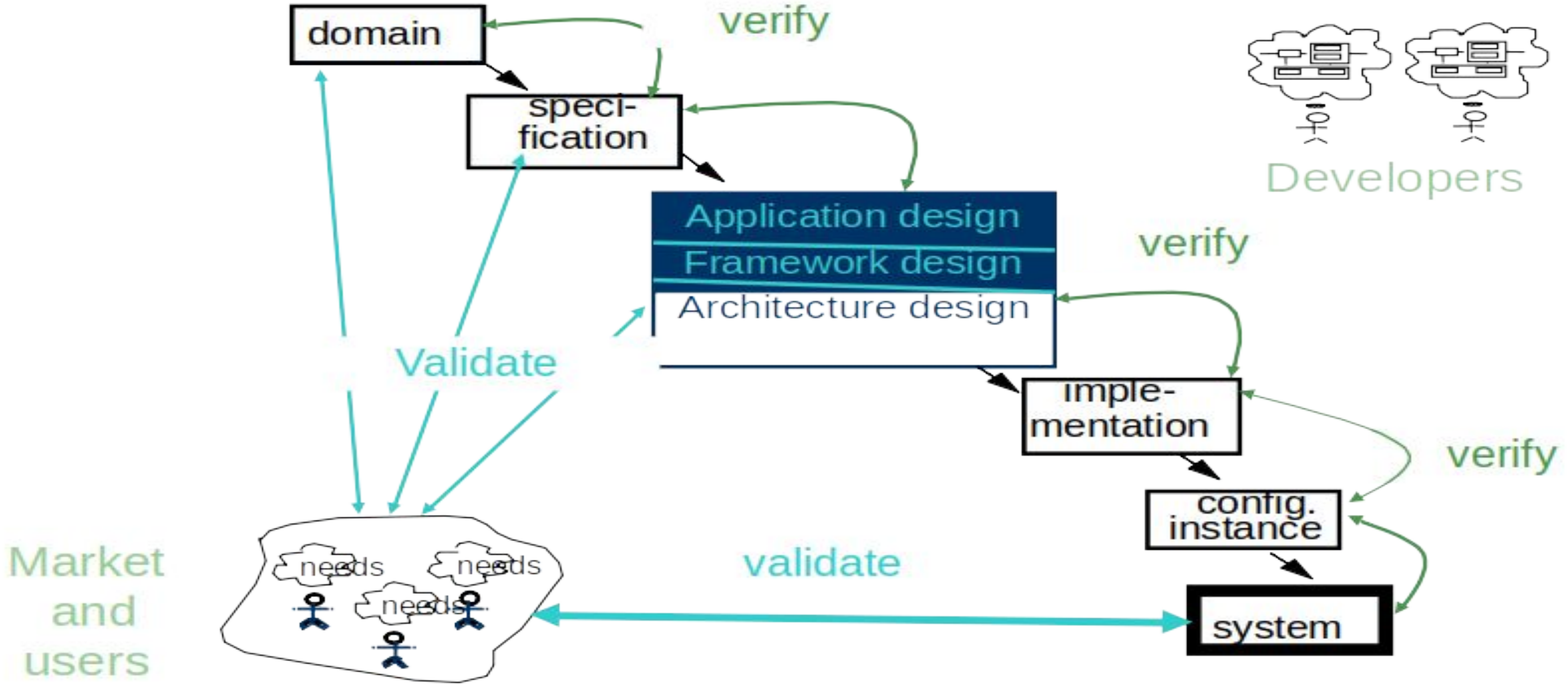
To model structural and behavioral aspects of cloud back-end part and elasticity controller using bigraphical approach.

And verify their correctness for horizontal and vertical scaling using BigMC model checker.

Formal Methods

- We use formal approach for verification.
- Formal verification has mainly 3 parts
    - Modeling of Systems and its dynamics.
    - Natural language properties -> Formal specification
    - Verification of Systems using model checking tool.
        - BigMC model checker

What are Formal Methods

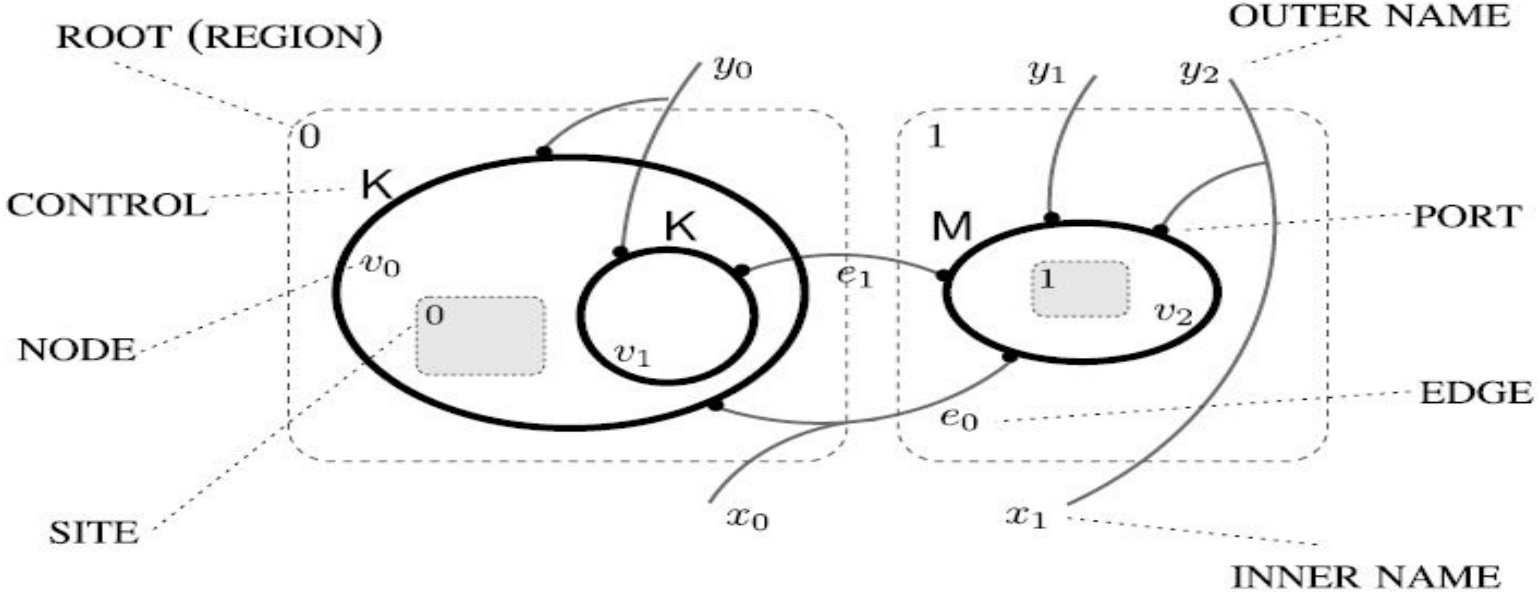


Elastic management ?

“The degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible.”

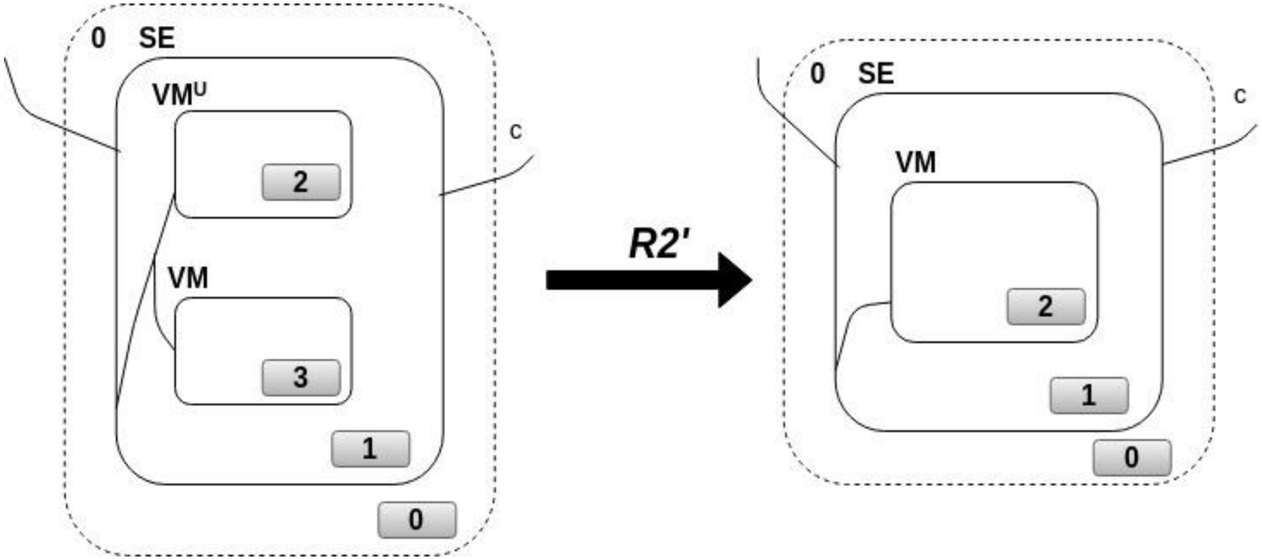
Bigraphical Reactive Systems(BRS)

- ❖ BRS is used to model the system formally in a graphical way.
- ❖ It’s easy to visualize “Motion” and “Arrangement” of elements using graphical approach.



Bigraphical Reactive Systems(BRS)

BRS provides reaction rules that define the dynamic evolution of the system by specifying how the set of bigraphs can be reconfigured.



Redex (R)

Reactum (R’)

R →R’
R : Redex
R’ : Reactum

Modeling of Cloud structure(CS)

$$CS = (V_{CS}, E_{CS}, ctrl_{CS}, CS^{P}, CS^{L}) : I_{CS} \rightarrow J_{CS}$$

V_{CS} *and* E_{CS} are sets of nodes and edges of the bigraph CS.

$ctrl_{CS} : V_{CS} \rightarrow K_{CS}$ a control map that assigns each node $v \in V_{CS}$ with a control $k \in K_{CS}$.

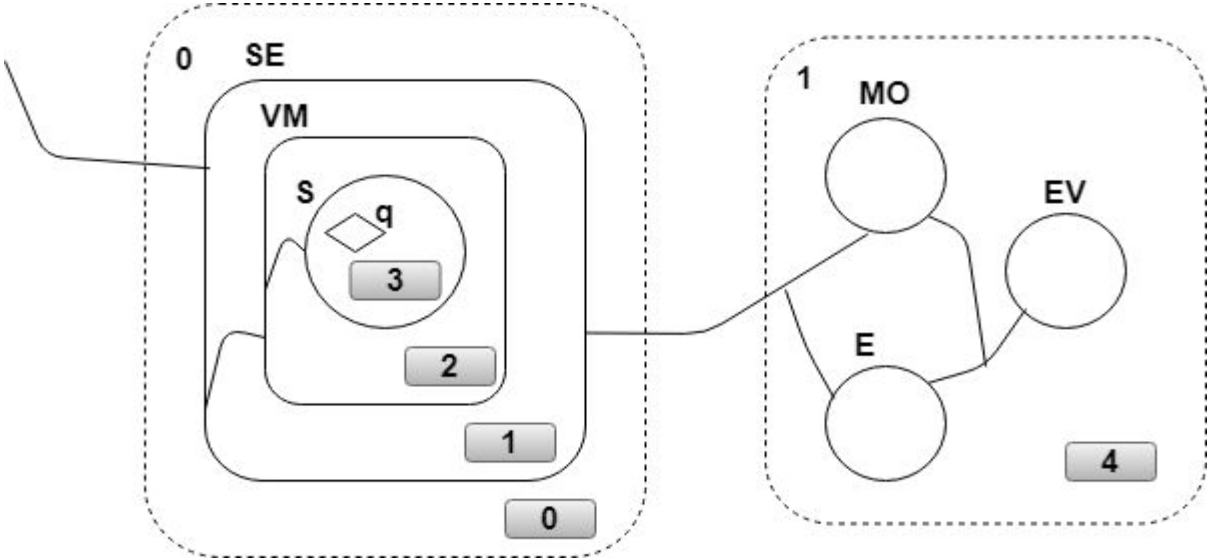
$CS^{P} = (V_{CS}, ctrl_{CS}, prnt_{CS}) : m_{CS} \rightarrow n_{CS}$ is the place graph of CS.

$CS^{L} = (V_{CS}, E_{CS}, ctrl_{CS}, link_{CS}) : X_{CS} \rightarrow Y_{CS}$ represents link graph of CS.

$I_{CS} = <m_{CS}, X_{CS}>$ *and* $J_{CS} = <n_{CS}, Y_{CS}>$ are the inner and outer interfaces of CS.

Modeling of Cloud structure(CS)

$$CS \overset{def}{=} ((SE.(VM.(S.q|d3)|d2)|d0)||(MO|EV|E|d4)$$



Cloud elements

Elastic Controller

Fig. Cloud Structure(Horizontal Scaling).

Modeling of Cloud structure(CS)

$$CS \overset{def}{=} (SE.(VM.((PC.P)|(RAMC.RAM)|(DISCC.DISC)|d_2)|d_1)|d_0)||(MO|EV|E|d_4)$$

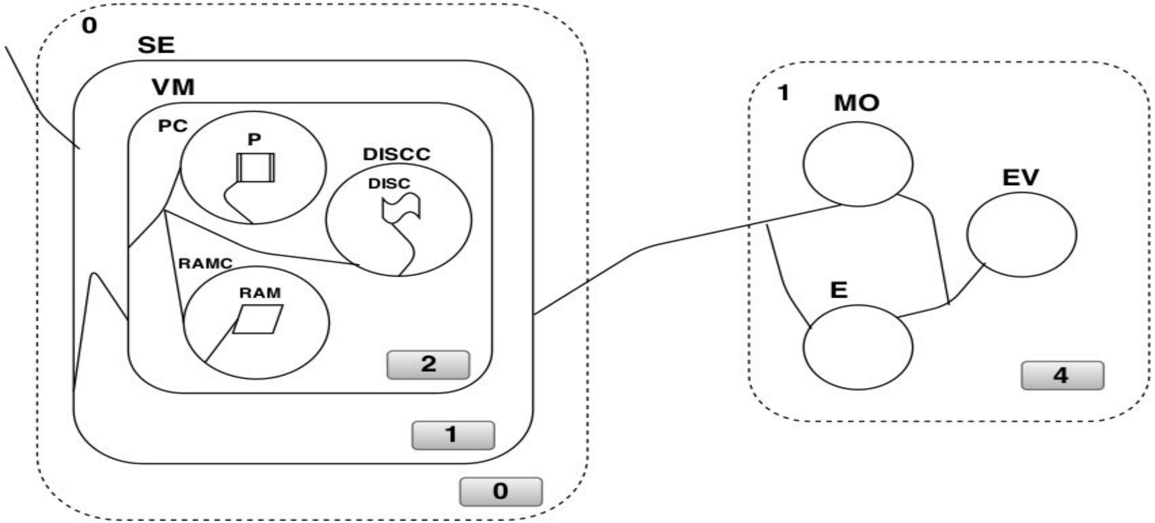


Fig. Cloud Structure (Vertical Scaling) .

Cloud Element	Control	Arity	Sort
Server	SE	3	e
Overloaded Server	SE^L	3	e
Unused Server	SE^U	3	e
Virtual Machine	VM	2	v
Overloaded VM	VM^L	2	v
Unused VM	VM^U	2	v
Service Instance	S	1	s
Overloaded SI	S^L	1	s
Unused SI	S^U	1	s
Request	q	0	q
Evaluator	EV	1	o
Monitor	MO	2	m
Effector	E	2	f

States and ports of elements(For Horizontal Scaling)

Elastic Strategies

- ❖ Set of actions that are triggered in case the specified triggering conditions are fulfilled.

Strategies are of the form of

strat: if CS ⊨ φ then Ri.

CS: Cloud Structure
φ : Condition
Ri : Reaction rule

Modeling of Elastic Strategies

Conditions

- All service instances are overloaded

  $\varphi 1 \equiv \forall s \in V_{CS}\ ctrl_{CS}(s) = S^{L}$

- A Service instance is unused

  $\varphi 4 \equiv \exists s \in V_{CS}\ ctrl_{CS}(s) = S^{U}$

- All VMs are overloaded

  $\varphi 2 \equiv \forall v \in V_{CS}\ ctrl_{CS}(v) = VM^{L}$

- A VM is unused

  $\varphi 5 \equiv \exists v \in V_{CS}\ ctrl_{CS}(v) = VM^{U}$

- All Servers are overloaded

  $\varphi 3 \equiv \forall e \in V_{CS}\ ctrl_{CS}(e) = SE^{L}$

Modeling of Elastic Strategies

Reaction rule	Algebraic form
Deploy a new service instance	$R1 \overset{\text{def}}{=} (\mathrm{SE.(VM.}d1)\vert d0) \mid \mathrm{id} \rightarrow (\mathrm{SE.(VM.}\boldsymbol{(\mathrm{S.}d2)}\vert d1)\vert d0) \mid \mathrm{id}$
Consolidate a new service instance	$R2 \overset{\text{def}}{=} (\mathrm{SE.(VM.}(\mathbf{S}^{\mathbf{U}}.d2)\vert d1)\vert d0) \mid \mathrm{id} \rightarrow (\mathrm{SE.(VM.}d1)\vert d0) \mid \mathrm{id}$
Deploy a new VM instance	$R3 \overset{\text{def}}{=} (\mathrm{SE.}d0) \mid \mathrm{id} \rightarrow (\mathrm{SE.}\boldsymbol{(\mathrm{VM.}d1)}\vert d0) \mid \mathrm{id}$
Consolidate a VM instance	$R4 \overset{\text{def}}{=} (\mathrm{SE.(}\mathbf{VM}^{\mathbf{U}}.d1)\vert d0) \mid \mathrm{id} \rightarrow (\mathrm{SE.}d0) \mid \mathrm{id}$
Turn on a new server	$R5 \overset{\text{def}}{=} \mathrm{id} \rightarrow (\mathrm{SE.}d0) \mid \mathrm{id}$

Modeling of Elastic Strategies(Horizontal Scaling examples)

$R1 \stackrel{def}{=} \quad (SE.(VM.(S^{L}.(d3)|d2)|d1)|d0)|id \ \rightarrow (SE.(VM.(S.(d3)|S.(d4)|d2)|d1)|\ d0)|id$

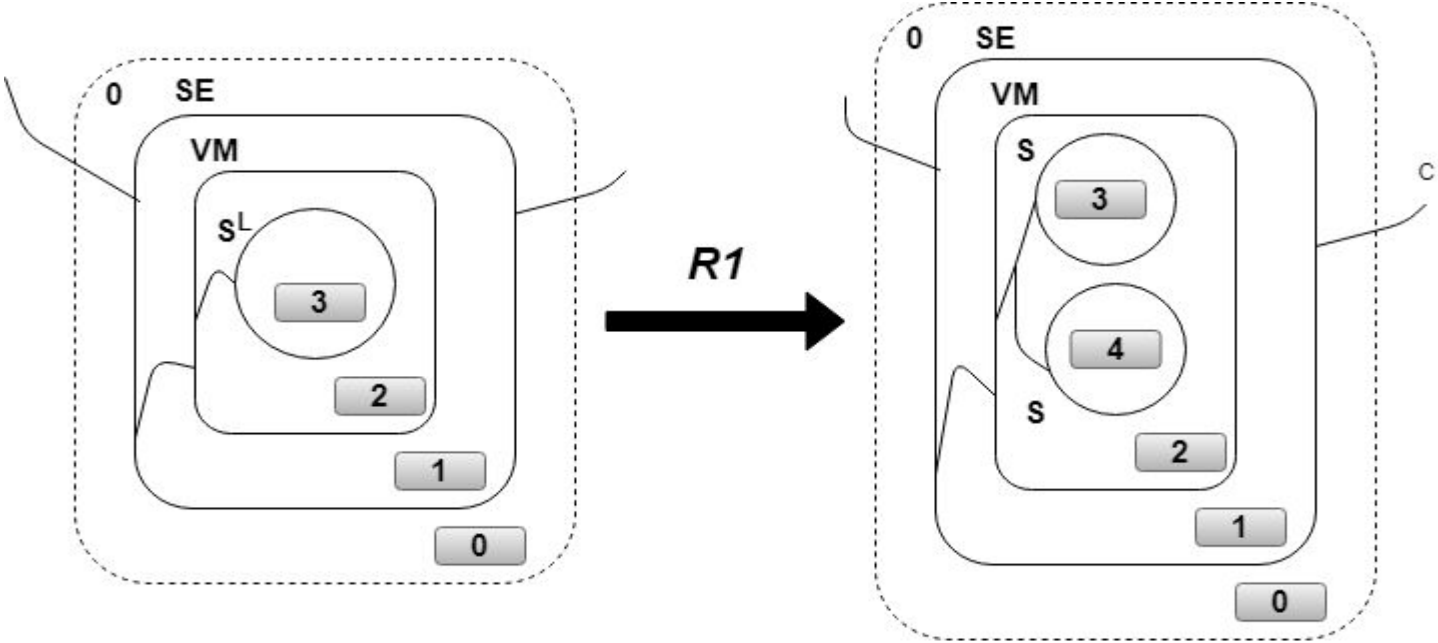


Fig. Deploy a new service instance

Modeling of Elastic Strategies(Horizontal Scaling examples)

$$R2 \stackrel{\text{def}}{=} \quad (SE.(VM.((S^{U}.d2)|(S.d4)|d2)|d1)|d0)|id \rightarrow (SE.(VM.(S.d2)|d1)|d0|id$$

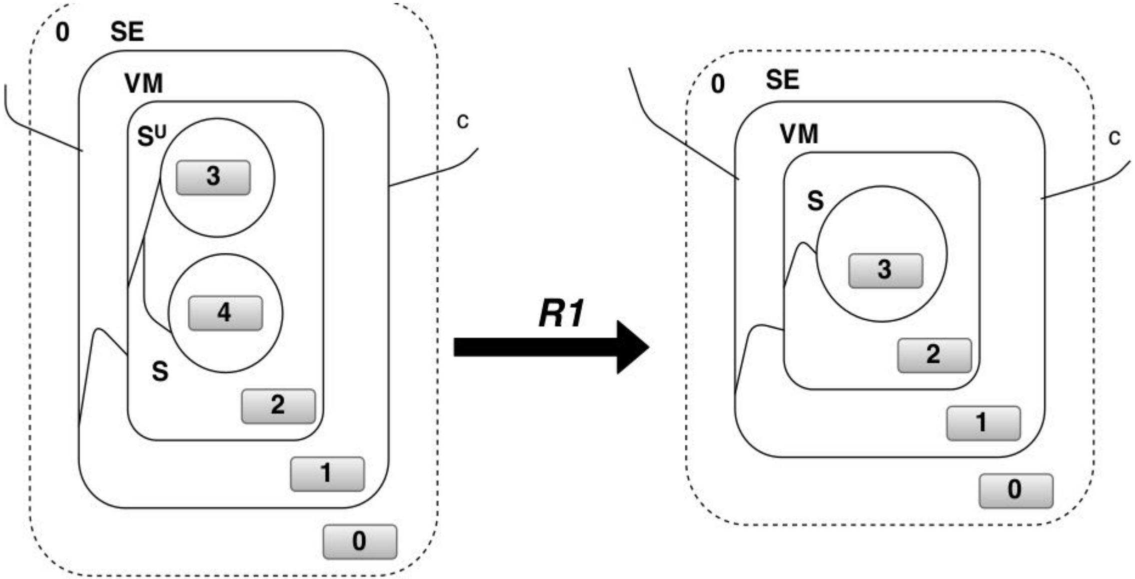


Fig. Consolidate a service instance

Modeling of Elastic Strategies (Vertical Scaling examples)

$$R4 \overset{\text{def}}{=} (SE.(VM.((PC.P^{L})|d2)|d1)|d0) \rightarrow (SE.(VM.((PC.(P|P))|d2)|d1)|d0)$$

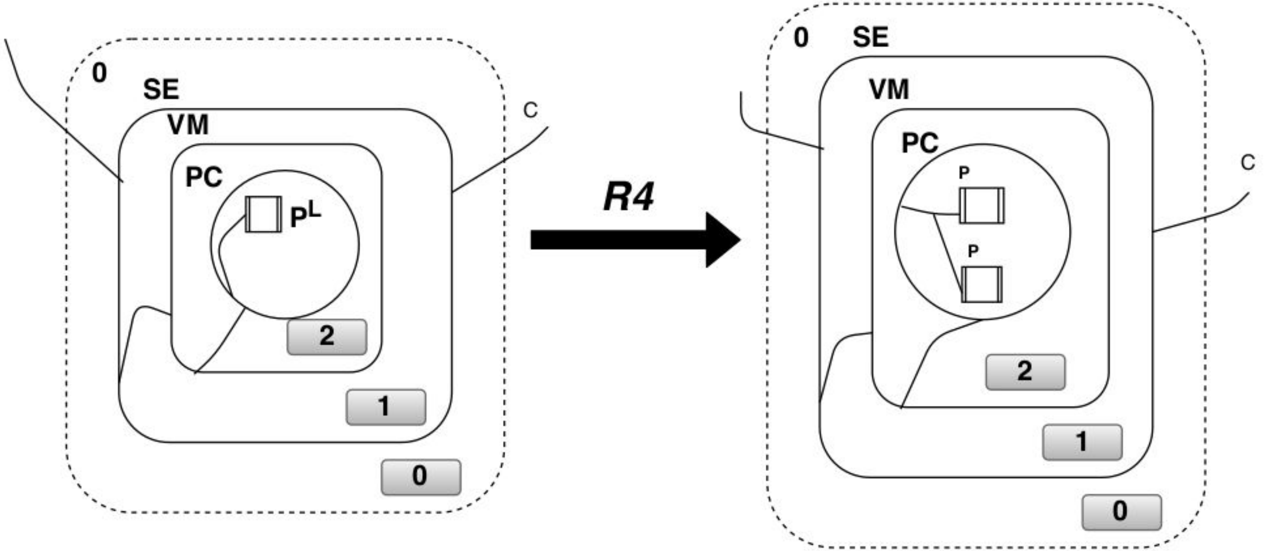


Fig. Turn on a new processor

Modeling of Elastic Strategies(Vertical Scaling examples)

$$R4' \stackrel{\text{def}}{=} (SE.(VM.(P^U|P^U|d2\ )|d1)|d0)|id\ \rightarrow (SE.(VM.(P|d2)|d1)|d0)|id$$

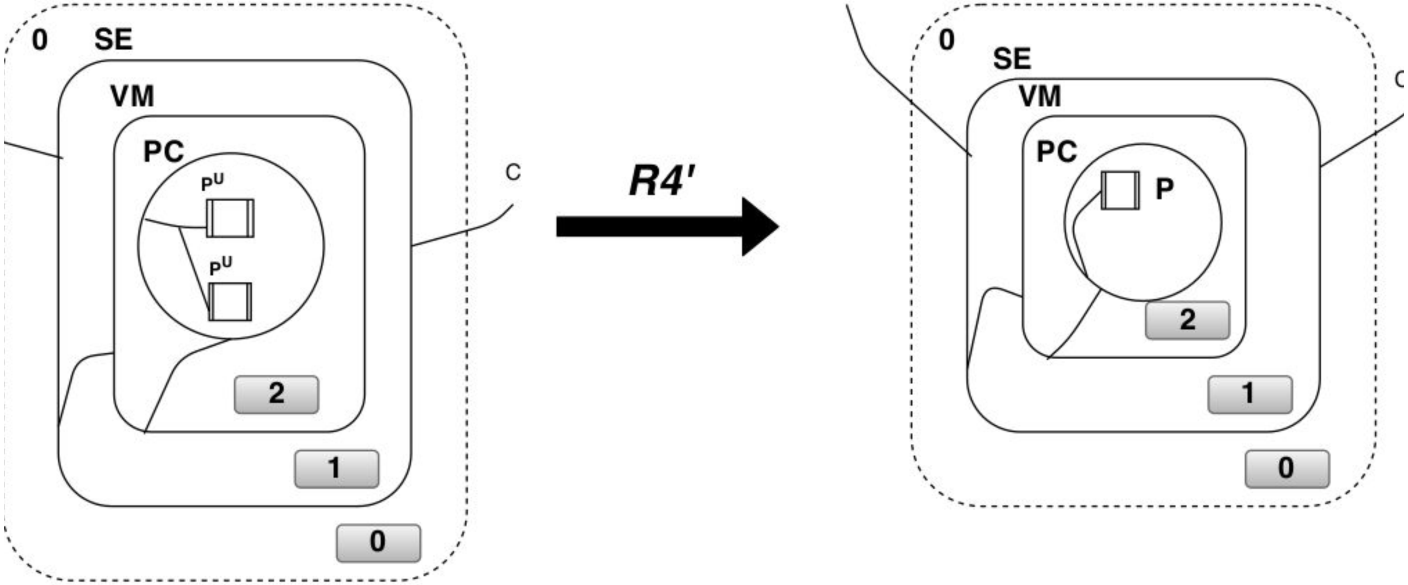


Fig. Turn off the underloaded processor

Properties(Formal specification)

- Express how the system should behave ideally.

- Should hold in every transiting-state for the model to be correct.

- If anything goes wrong, property violation reported by tool.

- Example
  - When the server gets overloaded, ideally new server should be deployed.
  - If virtual machine is unused , remove it.
  - If RAM is getting full, increase RAM allocated to the VM.

- Properties to be converted in **formal specification** to avoid ambiguity.

Changing workload

Workload in the system can be altered by changing the states of element with the help of reaction rules.

- SE -> SEL (Overload a server)
- SE -> SEU (Underload a server)
- VM -> VMU (Underload vm)
- VML -> VM (Stabilize the vm)

  .

  .

  .

  And so on.

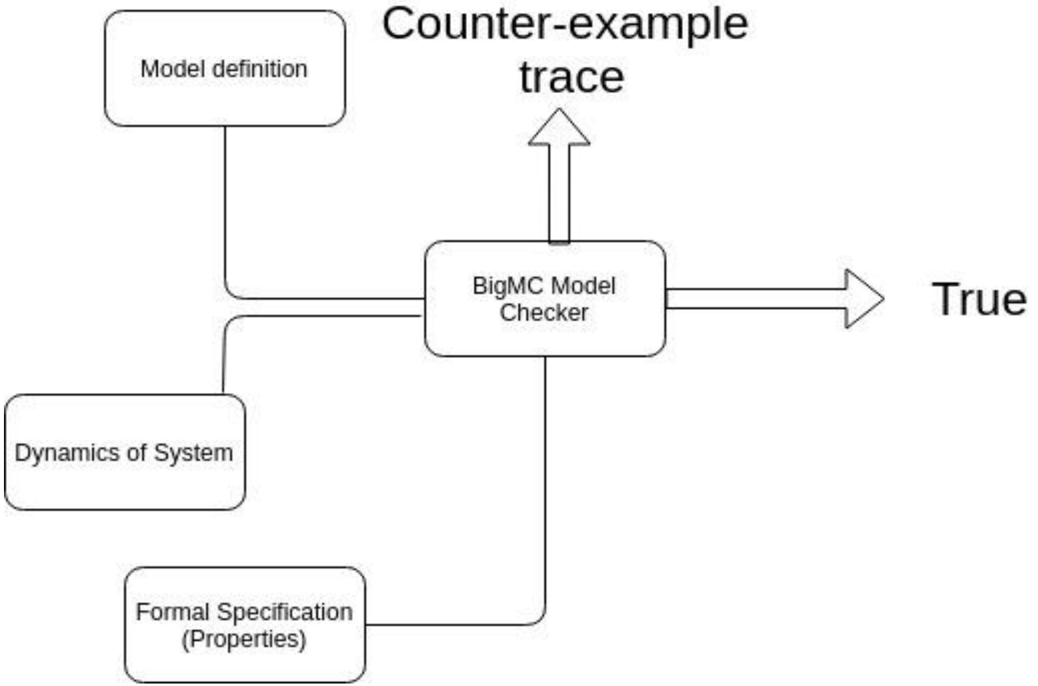
BigMC model checker

- Designed to operate on Bigraphical Reactive Systems.

- Model checking is accomplished through an exhaustive search of all possible states of the bigraphical model that satisfy the property to be verified.

- Provides a counter-example whenever the desired property violates.
  - Shows transition system path by which this configuration was reached.

BigMC model checker

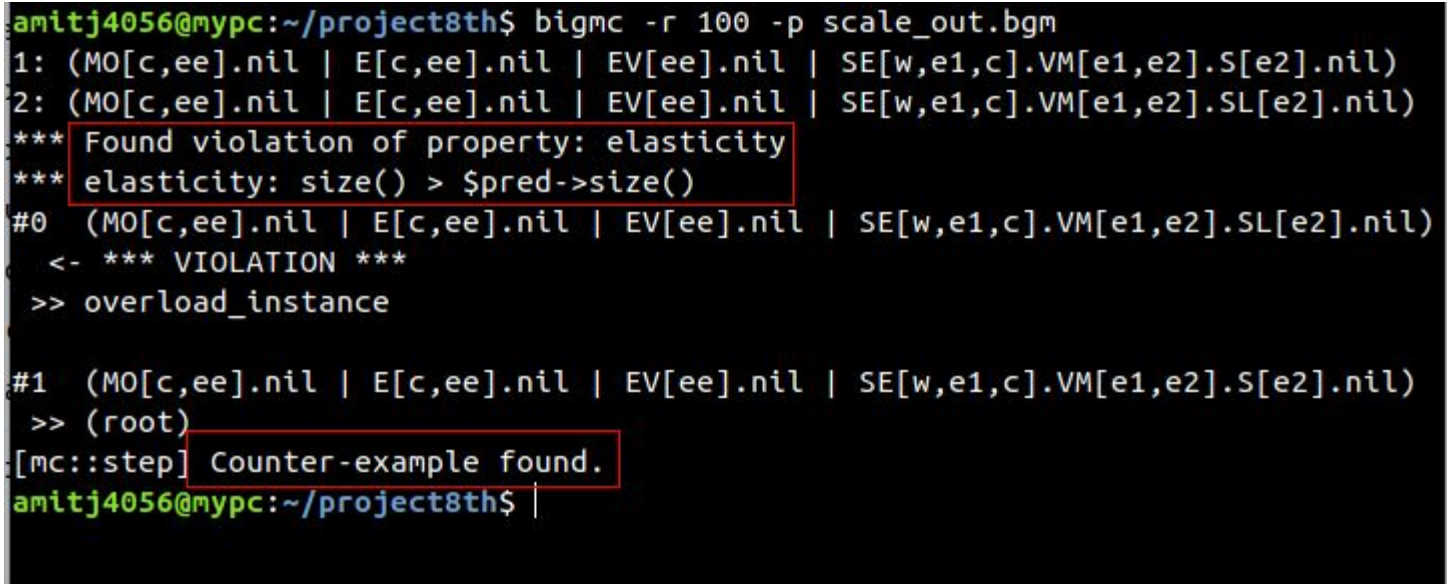


True : Satisfies property

False : report counter example trace

Fig. Model Checking Methodology

Property violation(counter-example)



```
amitj4056@mypc:~/project8th$ bigmc -r 100 -p scale_out.bgm
1: (MO[c,ee].nil | E[c,ee].nil | EV[ee].nil | SE[w,e1,c].VM[e1,e2].S[e2].nil)
2: (MO[c,ee].nil | E[c,ee].nil | EV[ee].nil | SE[w,e1,c].VM[e1,e2].SL[e2].nil)
*** Found violation of property: elasticity
*** elasticity: size() > $pred->size()
#0  (MO[c,ee].nil | E[c,ee].nil | EV[ee].nil | SE[w,e1,c].VM[e1,e2].SL[e2].nil)
  <- *** VIOLATION ***
 >> overload_instance

#1  (MO[c,ee].nil | E[c,ee].nil | EV[ee].nil | SE[w,e1,c].VM[e1,e2].S[e2].nil)
 >> (root)
[mc::step] Counter-example found.
amitj4056@mypc:~/project8th$
```

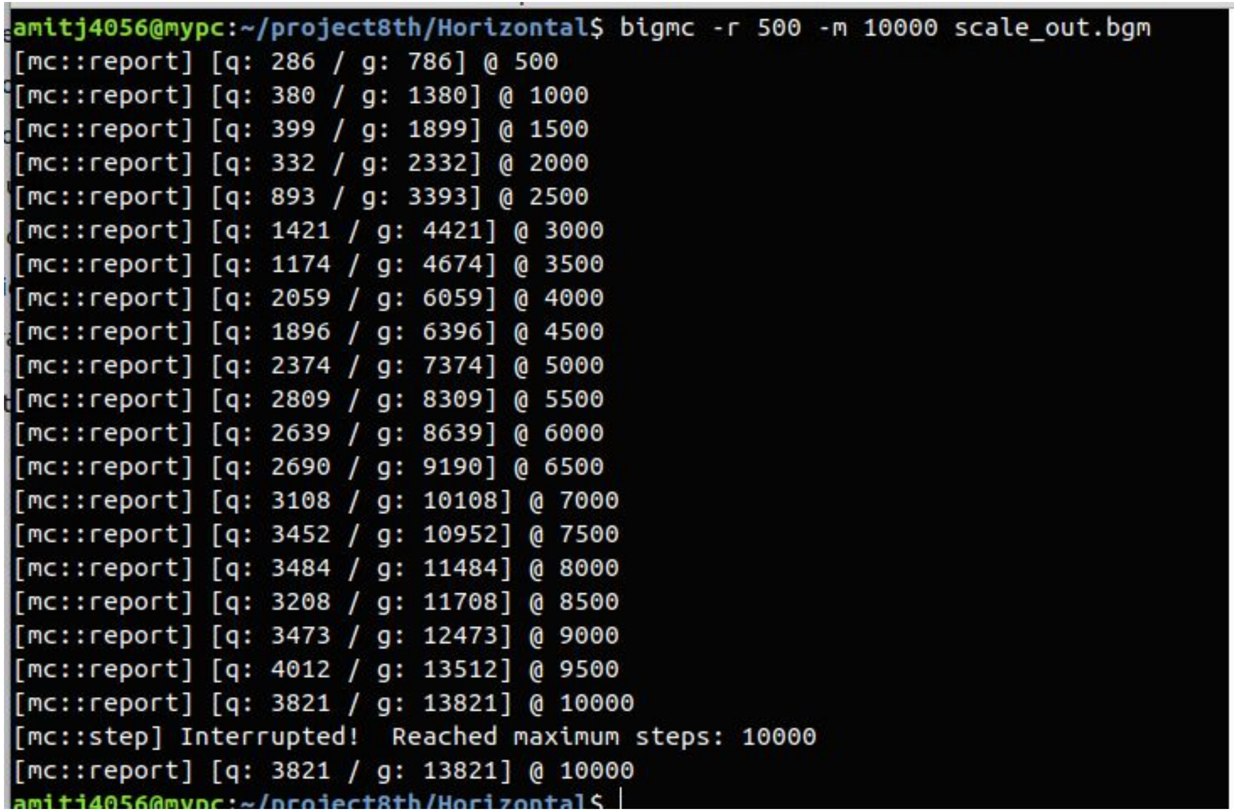
Fig. Property violation

Verification results

- Horizontal scaling
  - Service instance
  - Virtual Machine
  - Server

- Vertical Scaling
  - Processor
  - RAM
  - DISC

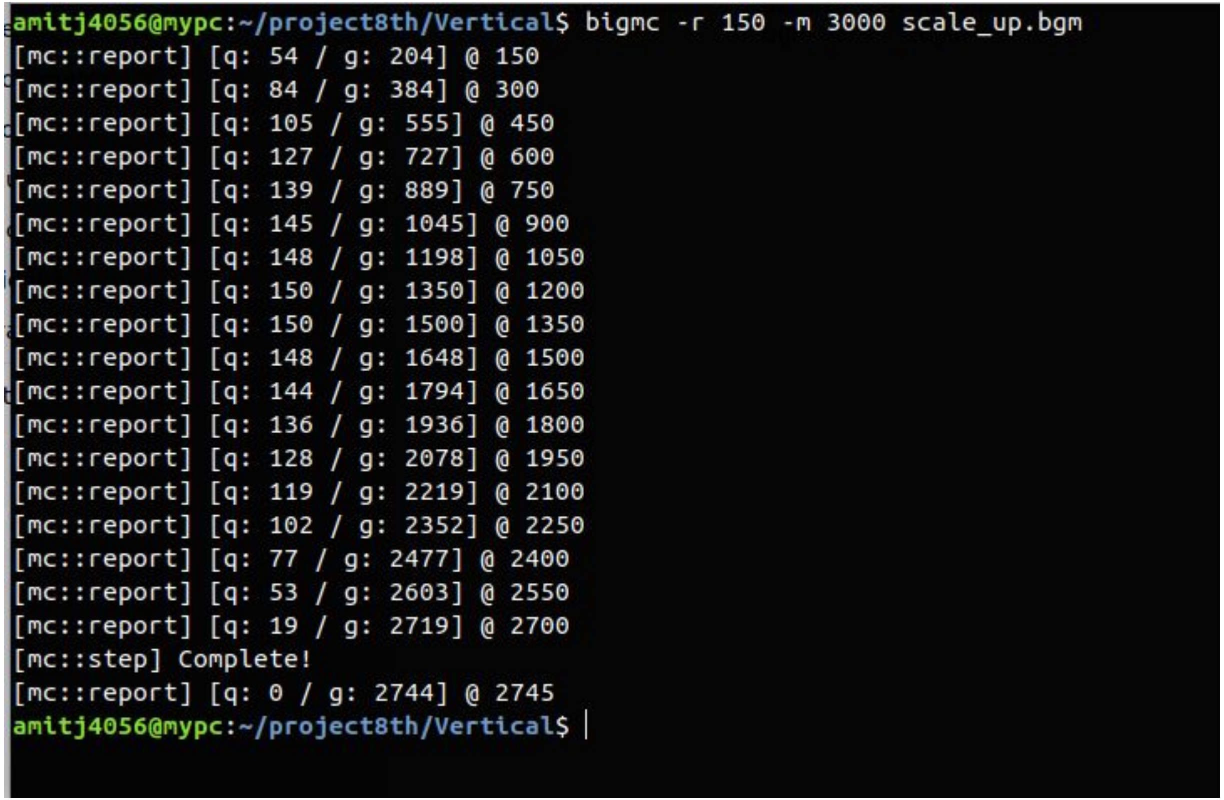
- BigMC model checker verified specification successfully without any violation.



```
amitj4056@mypc:~/project8th/Horizontal$ bigmc -r 500 -m 10000 scale_out.bgm
[mc::report] [q: 286 / g: 786] @ 500
[mc::report] [q: 380 / g: 1380] @ 1000
[mc::report] [q: 399 / g: 1899] @ 1500
[mc::report] [q: 332 / g: 2332] @ 2000
[mc::report] [q: 893 / g: 3393] @ 2500
[mc::report] [q: 1421 / g: 4421] @ 3000
[mc::report] [q: 1174 / g: 4674] @ 3500
[mc::report] [q: 2059 / g: 6059] @ 4000
[mc::report] [q: 1896 / g: 6396] @ 4500
[mc::report] [q: 2374 / g: 7374] @ 5000
[mc::report] [q: 2809 / g: 8309] @ 5500
[mc::report] [q: 2639 / g: 8639] @ 6000
[mc::report] [q: 2690 / g: 9190] @ 6500
[mc::report] [q: 3108 / g: 10108] @ 7000
[mc::report] [q: 3452 / g: 10952] @ 7500
[mc::report] [q: 3484 / g: 11484] @ 8000
[mc::report] [q: 3208 / g: 11708] @ 8500
[mc::report] [q: 3473 / g: 12473] @ 9000
[mc::report] [q: 4012 / g: 13512] @ 9500
[mc::report] [q: 3821 / g: 13821] @ 10000
[mc::step] Interrupted!  Reached maximum steps: 10000
[mc::report] [q: 3821 / g: 13821] @ 10000
amitj4056@mypc:~/project8th/Horizontal$
```

- ❖ Programs ends without reporting any property violation.
- ❖ Explored 10,000 different transition states.

Fig. Output of BigMC for horizontal scaling



```
amitj4056@mypc:~/project8th/Vertical$ bigmc -r 150 -m 3000 scale_up.bgm
[mc::report] [q: 54 / g: 204] @ 150
[mc::report] [q: 84 / g: 384] @ 300
[mc::report] [q: 105 / g: 555] @ 450
[mc::report] [q: 127 / g: 727] @ 600
[mc::report] [q: 139 / g: 889] @ 750
[mc::report] [q: 145 / g: 1045] @ 900
[mc::report] [q: 148 / g: 1198] @ 1050
[mc::report] [q: 150 / g: 1350] @ 1200
[mc::report] [q: 150 / g: 1500] @ 1350
[mc::report] [q: 148 / g: 1648] @ 1500
[mc::report] [q: 144 / g: 1794] @ 1650
[mc::report] [q: 136 / g: 1936] @ 1800
[mc::report] [q: 128 / g: 2078] @ 1950
[mc::report] [q: 119 / g: 2219] @ 2100
[mc::report] [q: 102 / g: 2352] @ 2250
[mc::report] [q: 77 / g: 2477] @ 2400
[mc::report] [q: 53 / g: 2603] @ 2550
[mc::report] [q: 19 / g: 2719] @ 2700
[mc::step] Complete!
[mc::report] [q: 0 / g: 2744] @ 2745
amitj4056@mypc:~/project8th/Vertical$ |
```

- ❖ Programs ends without reporting any property violation.
- ❖ Explored around 2700 different transition states.

Fig. Output of BigMC for vertical scaling

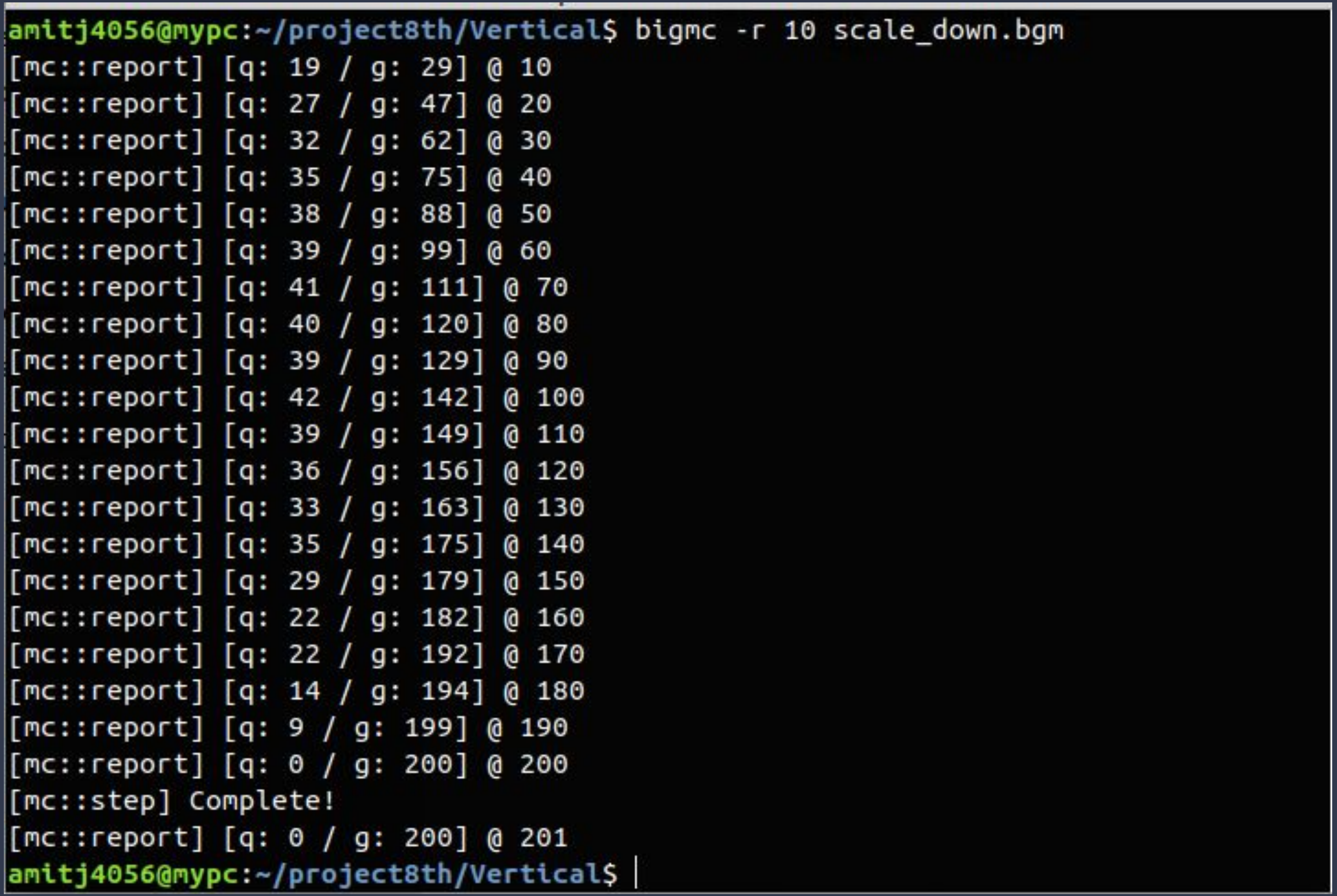
Conclusion

We provided structural and behavioral aspects of elastic cloud systems and they have been modeled using the BRS formalism.

Also we modeled horizontal and vertical scale strategies for (de)provisioning cloud system resources at service and infrastructure levels.

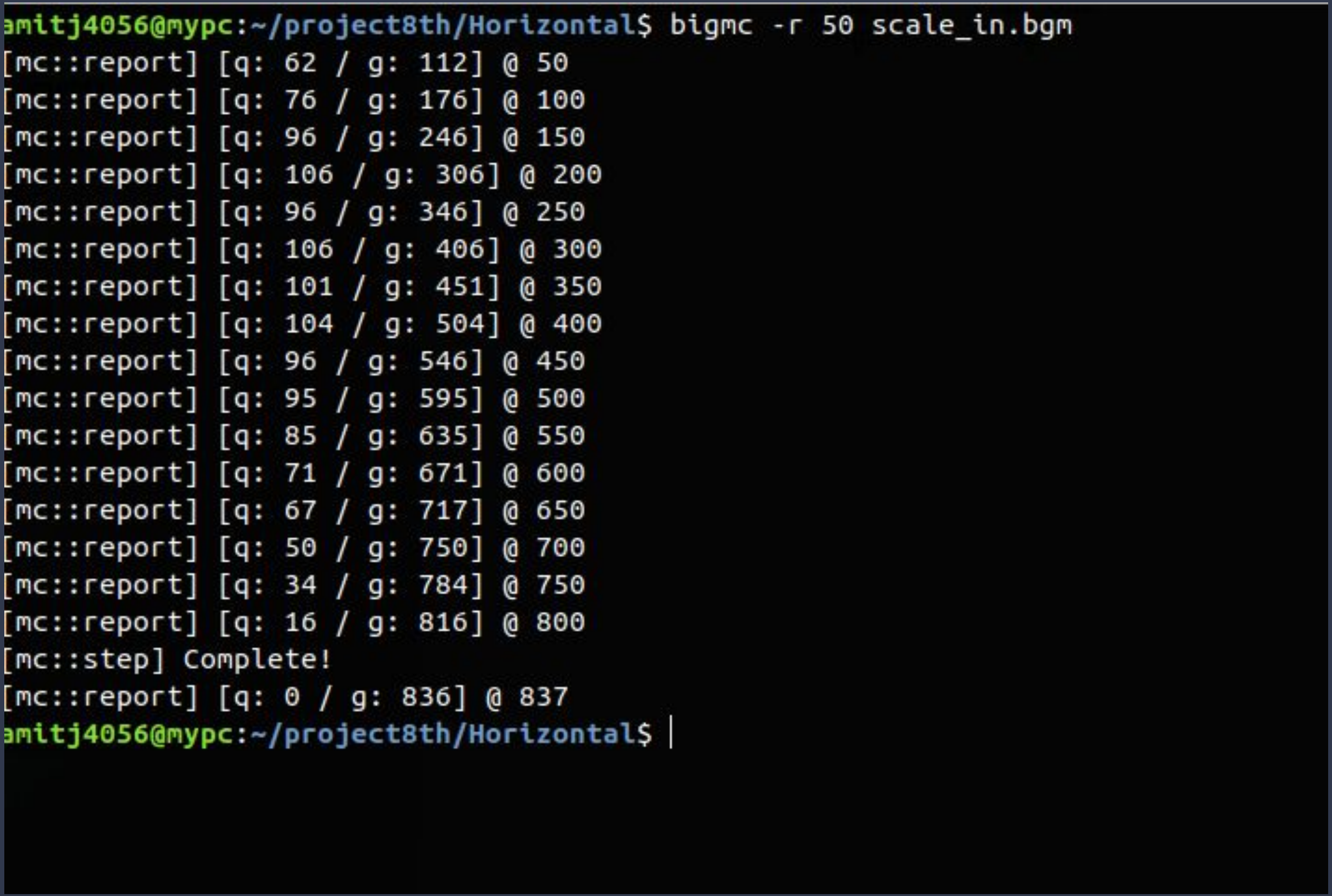
Thank you !!

Outputs



```
amitj4056@mypc:~/project8th/Vertical$ bigmc -r 10 scale_down.bgm
[mc::report] [q: 19 / g: 29] @ 10
[mc::report] [q: 27 / g: 47] @ 20
[mc::report] [q: 32 / g: 62] @ 30
[mc::report] [q: 35 / g: 75] @ 40
[mc::report] [q: 38 / g: 88] @ 50
[mc::report] [q: 39 / g: 99] @ 60
[mc::report] [q: 41 / g: 111] @ 70
[mc::report] [q: 40 / g: 120] @ 80
[mc::report] [q: 39 / g: 129] @ 90
[mc::report] [q: 42 / g: 142] @ 100
[mc::report] [q: 39 / g: 149] @ 110
[mc::report] [q: 36 / g: 156] @ 120
[mc::report] [q: 33 / g: 163] @ 130
[mc::report] [q: 35 / g: 175] @ 140
[mc::report] [q: 29 / g: 179] @ 150
[mc::report] [q: 22 / g: 182] @ 160
[mc::report] [q: 22 / g: 192] @ 170
[mc::report] [q: 14 / g: 194] @ 180
[mc::report] [q: 9 / g: 199] @ 190
[mc::report] [q: 0 / g: 200] @ 200
[mc::step] Complete!
[mc::report] [q: 0 / g: 200] @ 201
amitj4056@mypc:~/project8th/Vertical$
```

Outputs



```
amitj4056@mypc:~/project8th/Horizontal$ bigmc -r 50 scale_in.bgm
[mc::report] [q: 62 / g: 112] @ 50
[mc::report] [q: 76 / g: 176] @ 100
[mc::report] [q: 96 / g: 246] @ 150
[mc::report] [q: 106 / g: 306] @ 200
[mc::report] [q: 96 / g: 346] @ 250
[mc::report] [q: 106 / g: 406] @ 300
[mc::report] [q: 101 / g: 451] @ 350
[mc::report] [q: 104 / g: 504] @ 400
[mc::report] [q: 96 / g: 546] @ 450
[mc::report] [q: 95 / g: 595] @ 500
[mc::report] [q: 85 / g: 635] @ 550
[mc::report] [q: 71 / g: 671] @ 600
[mc::report] [q: 67 / g: 717] @ 650
[mc::report] [q: 50 / g: 750] @ 700
[mc::report] [q: 34 / g: 784] @ 750
[mc::report] [q: 16 / g: 816] @ 800
[mc::step] Complete!
[mc::report] [q: 0 / g: 836] @ 837
amitj4056@mypc:~/project8th/Horizontal$
```