

Exp No: 7

Date :

CLOUD SIMULATION

MODEL CLOUD ENVIRONMENT USING CLOUD SIM

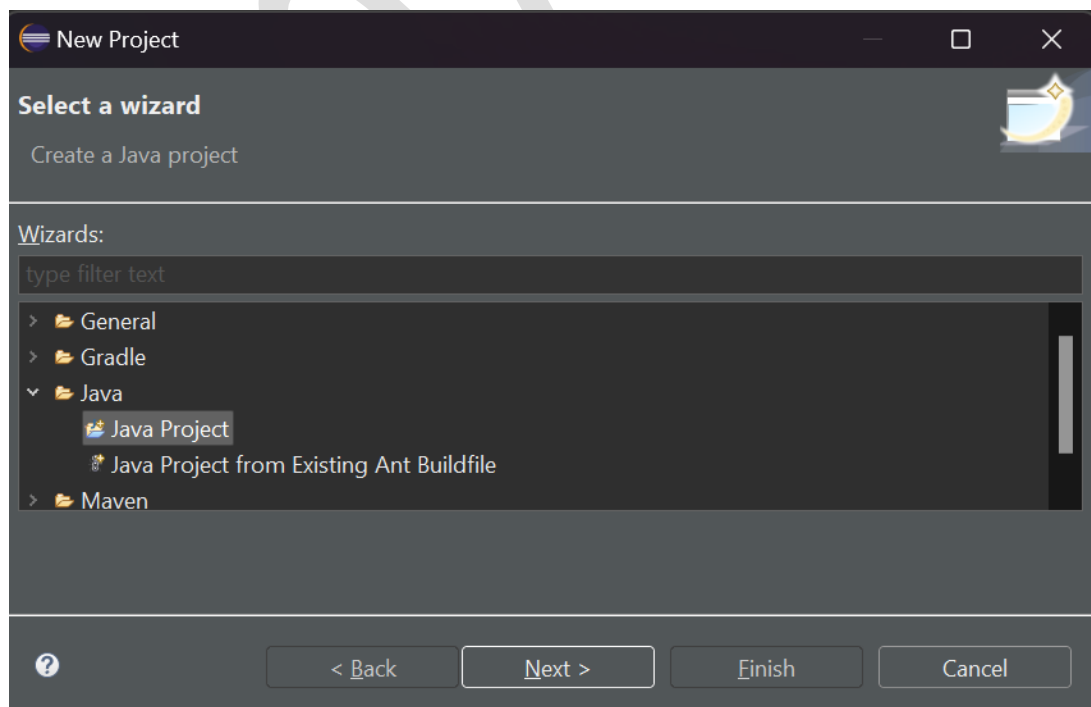
AIM:

To model the cloud environment using cloud sim tools.

PROCEDURE:

1. Now within the Eclipse window navigate the menu: File -> New -> Project, to open the new project wizard.
2. Select the 'Java Project' from the window popup and click Next.
3. Unselect the 'Use default location' option and then click on '*Browse*' to open the path where you have unzipped the Cloudsim project and finally click Next to set project settings.
4. Now select the example program from folder examples from the unzipped folder.
5. Run the sample program

OUTPUT:



```

1 package org.cloudbus.cloudsim.examples;
2
34 * Title:      CloudSim Toolkit
35
36
37 /**
38  * A simple example showing how to create a data center with one host and run one cloudlet on it.
39  */
40 public class CloudSimExample1 {
41     /** The cloudlet list. */
42     private static List<Cloudlet> cloudletList;
43     /** The vm list. */
44     private static List<Vm> vmList;
45
46     /**
47      * Creates main() to run this example.
48      *
49      * @param args the args
50      */
51     public static void main(String[] args) {
52         Log.println("Starting CloudSimExample1...");
53
54         try {
55             // First step: Initialize the CloudSim package. It should be called before creating any entities
56             int num_user = 1; // number of cloud users
57             Calendar calendar = Calendar.getInstance(); // Calendar whose fields have been initialized with
58             boolean trace_flag = false; // trace events
59
60             /* Comment Start - Dinesh Bhagwat
61              * Initialize the CloudSim library.
62              * init() invokes initCommonVariable() which in turn calls initialize() (all these 3 methods are
63              * initialize() creates two collections - an ArrayList of SimEntity Objects (named entities wh
64              * a LinkedHashMap (named entitiesByName which denote the LinkedHashMap of the same simulation

```

```

1 package org.cloudbus.cloudsim.examples;
2
34 * Title:      CloudSim Toolkit
35
36
37 /**
38  * A simple example showing how to create a data center with one host and run one cloudlet on it.
39  */
40 public class CloudSimExample1 {
41     /** The cloudlet list. */
42     private static List<Cloudlet> cloudletList;
43     /** The vm list. */
44     private static List<Vm> vmList;
45
46     /**
47      * Creates main() to run this example.
48      *
49      * @param args the args
50      */
51     public static void main(String[] args) {
52         Log.println("Starting CloudSimExample1...");
53
54         try {
55             // First step: Initialize the CloudSim package. It should be called before creating any entities
56             int num_user = 1; // number of cloud users
57             Calendar calendar = Calendar.getInstance(); // Calendar whose fields have been initialized with
58             boolean trace_flag = false; // trace events
59
60             /* Comment Start - Dinesh Bhagwat
61              * Initialize the CloudSim library.
62              * init() invokes initCommonVariable() which in turn calls initialize() (all these 3 methods are
63              * initialize() creates two collections - an ArrayList of SimEntity Objects (named entities wh
64              * a LinkedHashMap (named entitiesByName which denote the LinkedHashMap of the same simulation

```

terminated: CloudSimExample1 [Java Application] C:\Users\keerthana\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0...

```

Starting CloudSim version 7.0
0.0: Datacenter_0 is starting...
0.0: Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 datacenter(s)
0.0: Broker: Trying to Create Vm #0 in Datacenter_0
0.0: Datacenter_0.VmAllocator: .VmAllocator]: Vm #0 has been allocated to Host #0
0.01: Broker: Vm #0 has been created in Datacenter #2, Host #0
0.01: Broker: Sending Cloudlet #0 to Vm #0
400.01: Broker: Cloudlet #0 return received
400.01: Broker: The number of finished Cloudlets is:1
400.01: Broker: All Cloudlets executed. Finishing...
400.01: Broker: Destroying Vm #0
400.01: Broker is shutting down...
400.01: Simulation: No more future events
400.01: CloudInformationService is shutting down...
400.01: CloudInformationService: Notify all CloudSim entities for shutting down.
400.01: Datacenter_0 is shutting down...
400.01: Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0             SUCCESS   2               0       400    0.01         400.01
CloudSimExample1 finished!

```

```

29 * Title: CloudSim Toolkit
30
31 package org.cloudbus.cloudsim.examples;
32
33 import java.text.DecimalFormat;
34
35 /**
36  * A simple example showing how to create
37  * a datacenter with one host and run two
38  * cloudlets on it. The cloudlets run in
39  * VMs with the same MIPS requirements.
40  * The cloudlets will take the same time to
41  * complete the execution.
42  */
43 public class CloudSimExample2 {
44
45     /** The cloudlet list. */
46     private static List<Cloudlet> cloudletList;
47
48     /** The vmlist. */
49     private static List<Vm> vmList;
50
51     /**
52      * Creates main() to run this example
53      */
54     public static void main(String[] args) {
55
56         Log.println("Starting CloudSimExample2...");
57
58         try {
59             // First step: Initialize the CloudSim package. It should be called
60             // before creating any entities.
61

```

```

0.01: Broker: Vm #0 has been created in Datacenter #2, Host #0
0.01: Broker: Vm #1 has been created in Datacenter #2, Host #0
0.01: Broker: Sending Cloudlet #0 to Vm #0
0.01: Broker: Sending Cloudlet #1 to Vm #1
1000.01: Broker: Cloudlet #0 return received
1000.01: Broker: The number of finished Cloudlets is:1
1000.01: Broker: Cloudlet #1 return received
1000.01: Broker: The number of finished Cloudlets is:2
1000.01: Broker: All Cloudlets executed. Finishing...
1000.01: Broker: Destroying Vm #0
1000.01: Broker: Destroying Vm #1
1000.01: Broker is shutting down...
1000.01: Simulation: No more future events
1000.01: CloudInformationService is shutting down...
1000.01: CloudInformationService: Notify all CloudSim entities for shutting down.
1000.01: Datacenter_0 is shutting down...
1000.01: Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0            SUCCESS    2               0       1000    0.01        1000.01
1            SUCCESS    2               1       1000    0.01        1000.01
CloudSimExample2 finished!

Log.println("Starting CloudSimExample2...");

try {
    // First step: Initialize the CloudSim package. It should be called
    // before creating any entities.

```

```

1 package org.cloudbus.cloudsim.examples;
2
3 import org.cloudbus.cloudsim.*;
4
19 public class DiamondAppMultiExtensionExample {
20
21     /** The guestList. */
22     private static List<GuestEntity> guestList;
23
24     private static NetworkDatacenter datacenter;
25
26     private static DatacenterBroker broker;
27
28
29     private static int numberOfHosts = 4;
30     private static int numberOfVms = 4;
31     private static int numberOfContainers = 0;
32
33     /**
34      * Example of diamond-shaped DAG.
35      *
36      *      B
37      *     /\
38      *    A  D
39      *   /\
40      *  C
41      *
42      * @param args
43      *      the args
44      */
45     public static void main(String[] args) {
46
47         Log.println("Starting DiamondAppExample...");
48
49         try {
50             int num user = 1; // number of cloud users

```

```

1 package org.cloudbus
2
3 import org.cloudbus
4
19 public class Diamo
20
21 /** The guest
22 private stati
23 private stati
24 private stati
25 private stati
26 private stati
27 private stati
28 private stati
29 private stati
30 private stati
31 private stati
32
33 /**
34      * Example of
35      *
36      *      B
37      *     /\
38      *    A  D
39      *   /\
40      *  C
41      *
42      * @param arg
43      *
44      */
45     public stati
46
47         Log.println("Starting DiamondAppExample...");
48
49         try {
50             int num user = 1; // number of cloud users

```

0.0: Broker: Trying to Create Vm #0 in Datacenter_0
0.0: Broker: Trying to Create Vm #1 in Datacenter_0
0.0: Broker: Trying to Create Vm #2 in Datacenter_0
0.0: Broker: Trying to Create Vm #3 in Datacenter_0
0.0: Datacenter_0.VmAllocator: .VmAllocator]: Vm #0 has been allocated to NetworkHost #0
0.0: Datacenter_0.VmAllocator: .VmAllocator]: Vm #1 has been allocated to NetworkHost #1
0.0: Datacenter_0.VmAllocator: .VmAllocator]: Vm #2 has been allocated to NetworkHost #2
0.0: Datacenter_0.VmAllocator: .VmAllocator]: Vm #3 has been allocated to NetworkHost #3
0.01: Broker: Vm #0 has been created in Datacenter #2, NetworkHost #0
0.01: Broker: Vm #1 has been created in Datacenter #2, NetworkHost #1
0.01: Broker: Vm #2 has been created in Datacenter #2, NetworkHost #2
0.01: Broker: Vm #3 has been created in Datacenter #2, NetworkHost #3
0.01: Broker: Sending NetworkCloudlet #0 to Vm #0
0.01: Broker: Sending NetworkCloudlet #1 to Vm #1
0.01: Broker: Sending NetworkCloudlet #2 to Vm #2
0.01: Broker: Sending NetworkCloudlet #3 to Vm #3
Datacenter_0: Warning - NetworkCloudlet #3 owned by Broker is already completed/finished.
Therefore, it is not being executed again
0.01: Broker: NetworkCloudlet #3 return received
0.01: Broker: The number of finished Cloudlets is:1
1000.02999999992355: Broker: NetworkCloudlet #0 return received
1000.02999999992355: Broker: The number of finished Cloudlets is:2
2016.0499999999221: Broker: NetworkCloudlet #1 return received
2016.0499999999221: Broker: The number of finished Cloudlets is:3


```

1 package org.cloudbus.cloudsim.examples;
2
3 import org.cloudbus.cloudsim.core.*;
4
5
6 public class TestScheduledResource {
7     static ScheduledResource res;
8     static SrcEntity src;
9     static DstEntity dst;
10    static class SrcEntity extends SimEntity {
11        public SrcEntity(String name) {
12            super(name);
13        }
14    }
15    @Override
16    public void processEvent(SimEvent ev) {
17        Log.printlnConcat(CloudSim.clock(), ":", getName(), " processEvent()");
18        res.enqueue(SimEvent.SEND, dst.getId(), 100.0, CloudActionTags.BLANK, null);
19        res.enqueueDelay(1000.0, SimEvent.SEND, dst.getId(), 100.0, CloudActionTags.BLANK, null);
20    }
21 }
22 static class DstEntity extends SimEntity {
23    public DstEntity(String name) {
24        super(name);
25    }
26    @Override
27    public void processEvent(SimEvent ev) {
28        Log.printlnConcat(CloudSim.clock(), ":", getName(), " processEvent()");
29    }
30 }
31 public static void main(String[] args) {
32     int num_user = 1;
33     Calendar calendar = Calendar.getInstance();
34     boolean trace_flag = true;
35     CloudSim.init(num_user, calendar, trace_flag);
36     res = new ScheduledResource("res", 1.0);

```

```

1 package org.cloudbus
2
3 import org.cloudbus
4
5
6 public class TestS
7     static Schedul
8     static SrcEntit
9     static DstEntit
10    static class Si
11        public SrcE
12            super(
13        }
14    }
15    @Override
16    public void
17        Log.pr
18        res.enc
19        res.enc
20    }
21 }
22 static class Ds
23    public DstE
24        super(
25    }
26    @Override
27    public void
28        Log.pr
29    }
30 }
31 public static
32     int num_user = 1;
33     Calendar calendar = Calendar.getInstance();
34     boolean trace_flag = true;
35     CloudSim.init(num_user, calendar, trace_flag);
36     res = new ScheduledResource("res", 1.0);

```

```

275.0: dst processEvent()
275.0: res processEvent(), tag=NETWORK_PKT_FORWARD
295.0: dst processEvent()
295.0: res processEvent(), tag=NETWORK_PKT_FORWARD
300.0: res processEvent(), tag=NETWORK_PKT_FORWARD
300.0: dst processEvent()
1000.0: res processEvent(), tag=BLANK
1000.0: res processEvent(), ru.amountLeft=100.0
1010.0: res processEvent(), tag=BLANK
1010.0: res processEvent(), ru.amountLeft=100.0
1020.0: res processEvent(), tag=BLANK
1020.0: res processEvent(), ru.amountLeft=100.0
1275.0: res processEvent(), tag=NETWORK_PKT_FORWARD
1275.0: dst processEvent()
1295.0: res processEvent(), tag=NETWORK_PKT_FORWARD
1295.0: dst processEvent()
1300.0: res processEvent(), tag=NETWORK_PKT_FORWARD
1300.0: dst processEvent()
1300.0: Simulation: No more future events
1300.0: CloudInformationService is shutting down...
1300.0: CloudInformationService: Notify all CloudSim entities for shutting down.
1300.0: res is shutting down...
1300.0: src is shutting down...
1300.0: dst is shutting down...
Simulation completed.
Simulation completed.

```

RESULT:

Thus, the cloud environment using cloud sim tools has been modelled.