Cloud AppProfiler: Telco Cloud Applications Tracing and Monitoring

CTPD Project

By:
Sarra KHAZRI / Pr Mohamed CHERIET

Montreal, December 11, 2013
Outline

- Issue
- Objectives
- Review of Literature
- Proposed Solution
- Cloud Applications Tracing Challenges
- Results
- Future work
- Demo
Poor performance can be caused by the lack of proper resources:

- limited bandwidth
- limited disk space
- limited memory
- limited CPU
- limited network connections
- limited latency

Performance issues in the system can end a service delivery.
Issue

Poor performance causes companies to:

- Lose customers
- Deal with the service outage
- Reduce bottom line revenues
- Reduce employee productivity
- Deal with general lost productivity.
Cloud Applications Tracing Challenges

- Virtualization
- Performance metrics
- Migration
- Scalability
Cloud Applications Tracing Challenges

- **Virtualization**: monitoring the hypervisor layer isn't something traditional systems managements were easy to manage.
- **End User Response Profiling**: End user response time is difficult to monitor for cloud application for two reasons:
  - cloud applications operate across the open public network
  - the end users are often distributed across the globe.
- **Performance metrics**: Various metrics needed to be calculated
- **Cloud Scalability**: Scalability is very large and it isn't predictable and measurable
Cloud Applications Tracing Challenges

- **Scalability:** to ensure that the monitoring can cope with a large number of probes.
- **Elasticity:** So that the virtual resources created and destroyed by expanding and contracting networks are monitored correctly.
- **Migration:** So that any virtual resource which moves from one physical host to another is monitored correctly.
- **Adaptability:** So that monitoring framework can adapt to varying computational and network loads in order to not be invasive.
- **Automatic:** So that the monitoring framework can keep running without intervention and configuration.
The main objective is to design and develop a new model to trace and monitor applications in the cloud.

We seek through this solution to achieve the following objectives:

- Collecting data from applications running on the cloud using a monitoring agent.
- Storing data and calculating applications performance metrics.
- Visualizing metrics in graphs and charts.
- Analyzing applications performance metrics and displaying warning and alerts in case of problems.
State of Art

- **Paid Solution:**
  - AppDynamics
  - Manage Engine Applications Manager.

- **Free Solution:**
  - The Lattice Monitoring Framework[2010]
Proposed solution

Architecture
Proposed solution

Cloud App Profiler Architecture

- Compute Node Module
- Availability Module
- Resource Utilization Module
- Throughput Module
- Network Module
Methodology

Performance Analysis of cloud-based streaming Applications

Calculation of the Performance metrics

- End-to-end delay
- Jitter
- Packet Loss
- Throughput
- Number of requests accepted and refused
- Application uptime
- Application CPU, memory utilization
Methodology

Performance Analysis of cloud-based streaming Applications

Performance Metrics

- End-to-end delay
- Jitter
- Packet loss
- Throughput
- Application Throughput
- Application Availability
- Application Resources utilization
Methodology

Performance Analysis of cloud-based streaming Applications

Calculation of the Performance metrics

- **Data Collection**
  - *wireshark*
  - *tshark*

- **Analyse**
  - *Software Management*

- **Visualisation**
  - *Graphs & charts (jquery library &*

54%
# Methodology

## Performance Analysis of cloud-based streaming Applications

### Calculation of the Performance metrics

<table>
<thead>
<tr>
<th>timestamp</th>
<th>Source node</th>
<th>destination node</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:13:43.880866</td>
<td>host1.49609</td>
<td>host2.49609</td>
</tr>
<tr>
<td>08:13:46.627595</td>
<td>host2.49609 &gt; host1.49609</td>
<td>rr 10l 61209s 227j @4984.5664062500 + 2.7451171875</td>
</tr>
<tr>
<td>08:13:49.145088</td>
<td>host1.49609</td>
<td>host2.49609: sr @4989.4814453125 44894326 315p 80640b</td>
</tr>
<tr>
<td>08:13:53.194153</td>
<td>host2.49609 &gt; host1.49609</td>
<td>rr 15l 61413s 353j @4989.4814453125 + 4.0458984375</td>
</tr>
</tbody>
</table>

- **UDP src port**
- **UDP dst port**
- **NTP timestamp**
- **Sender’s octet count**
- **Sender’s packet count**
- **Interarrival jitter**
- **Cumulative number of packet lost**
- **Highest sequence number**
- **Last RTCP-SR Timestamp preceived**
Methodology

Calculation of the Performance metrics

- Delay (second) = t2 - (t1 + DLSR)
- Jitter (second) = Interarrival Jitter / sampling rate of media codec
- Packet Loss(%) = [(highest sequence number i – highest sequence number i-1)/(cumulative number of lost packet i - cumulative number of lost packet i -1)] * 100
- Throughput (kbps) = X * Y * 8 / Z

- X = RTPpayload + Rtpheader(12) + UDP(8) + IP(20) + Frame Relay(6) (bytes/packet)
- Y = timestamp i – timestamp i-1 (seconds)
- Z = [cumulative number of lost packet i - cumulative number of lost packet i -1] – (cumulative number of lost packet i - cumulative number of lost packet i -1)
Results
Results

Export Graph
Future Works

Integration of **Application Profiler** in **Smart Cloud Profiler**:

- Contribute to the tracing of telecommunications applications in the ecolotic project: ims apps
- Have a automatic cloud app tracing system.
Demo
References

- Haibo Mi; Huaimin Wang; Gang Yin; HuaCai; Qi Zhou; Tingtao Sun, "Performance problems diagnosis in cloud computing systems by mining request trace logs," Network Operations and Management Symposium (NOMS), 2012 IEEE, vol., no., pp.893,899, 16-20 April 2012
- [http://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf](http://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf)
- [http://www.unc.edu/courses/2010spring/law/357c/001/cloudcomputing/examples.html](http://www.unc.edu/courses/2010spring/law/357c/001/cloudcomputing/examples.html)
Thank You