ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems

Min Du, Feifei Li

School of Computing, University of Utah
A Simplified Cloud

Cloud Controller (CLC) →

Cluster Controller (CC) →

Node Controller (NC)

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
A Simplified Cloud

Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)
A Simplified Cloud

Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)
A Simplified Cloud

Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)
A Simplified Cloud

Monitor the Cloud

- To provide system-wide visibility
- CloudWatch (AWS/Eucalyptus)
A Simplified Cloud

Questions

1. Monitor more efficiently?
2. Utilize the statistics for security purpose?
ATOM Architecture
ATOM Architecture
ATOM Architecture
ATOM Architecture
ATOM Architecture

Motivation Design Evaluation Discussion

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
ATOM Architecture

ATOMIC Architecture

- Detect Anomaly from Reported Data
- Selectively Report Data

CLC
- Monitoring (Anomaly Detection)

Cluster Controller (CC)

NC
- Tracking
- Introspection & Orchestration

VM

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
ATOM Architecture

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
ATOM Architecture

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
ATOM Architecture

MONITORING (ANOMALY DETECTION)

ANOMALY FOUND.

INTROSPECT INTERESTING MEMORY SPACE.

VM

SELECTIVELY REPORT DATA

DETECT ANOMALY FROM REPORTED DATA

CLC

Cluster Controller (CC)

NC

TRACKING

INTROSPECTION & ORCHESTRATION

VM

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
ATOM Architecture

CLC

MONITORING
(ANOMALY DETECTION)

Cluster Controller (CC)

ANOMALY FOUND.

NC

TRACKING

INTROSPECTION
& ORCHESTRATION

VM

INTROSPECT INTERESTING MEMORY SPACE.

DETECT ANOMALY FROM REPORTED DATA.

SELECTIVELY REPORT DATA.
Tracking Component

Motivation Design Evaluation Discussion

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Tracking Component

Motivation

Design

Evaluation

Discussion

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$
  - $\Delta = 4$

**ATOM:** Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Tracking Component

What if a small error $\Delta$ is allowed?
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}; \Delta = 4$
Tracking Component

Node Controller \{CPUUtilization, NetworkIn, DiskReadBytes, ...\} Cloud Controller

What if a small error $\Delta$ is allowed?

- **Sequence**: \{0, 6, 0, 6, 0, 6, ...\}; $\Delta = 4$
- A naive way:
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \quad \Delta = 4$
- **A naive way:**

![Graph showing data value over time]
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- A naive way:
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** \{0, 6, 0, 6, 0, 6, ...\}; $\Delta = 4$
- **A naive way:**

![Diagram showing data points over time]

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$

- A naive way:

![Graph showing data values over time](image)
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- A naive way:
Motivation Design Evaluation Discussion

Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \; \Delta = 4$
- **A naive way:**

![Data Value Graph](image_url)
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence**: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- **A naive way**:

![Diagram showing data points at time 0 and 4 with a small error $\Delta$.]
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- A naive way:

\[\begin{array}{c}
0 \\
2 \\
4 \\
6
\end{array}\]

\[\begin{array}{c}
\text{Data Value}\end{array}\]

\[\begin{array}{c}
\text{Time}
\end{array}\]
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- A naive way:
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- A naive way:

* Values sent: $\{0, 6, 0, 6, 0, 6, \ldots\}$
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence**: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$

- **A naive way**:
  * Values sent: $\{0, 6, 0, 6, 0, 6, \ldots\}$
  * Optimal offline algorithm could only send one value: 3
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- A naive way:

* Values sent: $\{0, 6, 0, 6, 0, 6, \ldots\}$
* Optimal offline algorithm could only send one value: 3
* Competitive ratio: Unbounded
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence**: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:

![Graph showing data value over time](image)
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:

```
Node Controller ➔ {CPUUtilization, NetworkIn, DiskReadBytes, ...} ➔ Cloud Controller
```

Data Value

0 2 4 6

Time
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:
Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:

\begin{itemize}
  \item Node Controller \[\{CPU\text{Utilization}, \text{NetworkIn}, \text{DiskReadBytes}, \ldots\}\]
  \item Cloud Controller
\end{itemize}
Motivation Design Evaluation Discussion

Tracking Component

What if a small error $\Delta$ is allowed?

- Sequence: $\{0, 6, 0, 6, 0, 6, ...\}; \Delta = 4$
- The optimal one dimension online tracking algorithm:

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence**: $\{0, 6, 0, 6, 0, 6, \ldots\}$; $\Delta = 4$
- The optimal one dimension online tracking algorithm:

![Diagram showing data points and error correction]
What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \quad \Delta = 4$
- The optimal one dimension online tracking algorithm:
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- **The optimal one dimension online tracking algorithm:**

* Values sent: $\{0, 3\}$
Tracking Component

What if a small error $\Delta$ is allowed?

- **Sequence:** $\{0, 6, 0, 6, 0, 6, \ldots\}; \Delta = 4$
- **The optimal one dimension online tracking algorithm:**
  
  * Values sent: $\{0, 3\}$
  * Competitive ratio: $\log \Delta$
Tracking Component

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Monitoring Component

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Monitoring Component

Data matrix reported from each node:

\[
\begin{bmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
\end{bmatrix}
\]

\(n\) time instances

- Anomaly detection using this matrix;
- Use Principal Component Analysis (PCA);
- Sliding window;
- Metrics identification after anomalies are detected.

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Monitoring Component

Data matrix reported from each node:

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
\]

\(d\) metrics

\(n\) time instances
Monitoring Component

Data matrix reported from each node:

\[
\begin{bmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{bmatrix}
\]

\(d\) metrics

\(n\) time instances

- Anomaly detection using this matrix;
## Monitoring Component

Data matrix reported from each node:

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(\text{now})0} & V_{(\text{now})1} & V_{(\text{now})2} & \cdots & V_{(\text{now})d}
\end{pmatrix}
\]

\(d\) metrics across \(n\) time instances

- Anomaly detection using this matrix;
- Use Principal Component Analysis (PCA);
## Monitoring Component

**Data matrix reported from each node:**

$$
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & & \ddots & \cdots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
$$

$n$ time instances

$d$ metrics

- Anomaly detection using this matrix;
- Use Principal Component Analysis (PCA);
- Sliding window;
Monitoring Component

Data matrix reported from each node:

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
\]

\(n\) time instances

\(d\) metrics

- Anomaly detection using this matrix;
- Use Principal Component Analysis (PCA);
- Sliding window;
- Metrics identification after anomalies are detected.
Monitoring Component - Anomaly Detection
Monitoring Component - Anomaly Detection

PCA:
Monitoring Component - Anomaly Detection

PCA:
Monitoring Component - Anomaly Detection

PCA:
Monitoring Component - Anomaly Detection

PCA:
Monitoring Component - Anomaly Detection

PCA:

- threshold \( Q \) is computed according to a given false alarm rate \( \alpha \).
- Tracking component introduces error \( \Delta \) to data matrix.
- Given \( \mu \), dynamically adjust \( \Delta \) according to PCA results, to ensure false alarm rate \( \alpha' \in (\alpha - \mu, \alpha + \mu) \).
PCA:

Threshold $Q_\alpha$ is computed according to a given false alarm rate $\alpha$. 

The $Q_\alpha$ is computed according to a given false alarm rate $\alpha$. This threshold is computed to identify anomalous data points.

Projection Length > Threshold

Motivation Design Evaluation Discussion

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Monitoring Component - Anomaly Detection

PCA:

+ Threshold $Q_\alpha$ is computed according to a given false alarm rate $\alpha$.

+ Tracking component introduces error $\Delta$ to data matrix.
PCA:

- Threshold $Q_\alpha$ is computed according to a given false alarm rate $\alpha$.
- Tracking component introduces error $\Delta$ to data matrix.
- Given $\mu$, dynamically adjust $\Delta$ according to PCA results, to ensure false alarm rate $\in (\alpha - \mu, \alpha + \mu)$.
Motivation Design Evaluation Discussion

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

\[
\begin{bmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{\text{now}0} & V_{\text{now}1} & V_{\text{now}2} & \cdots & V_{\text{now}d}
\end{bmatrix}
\]
Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.
Motivation Design Evaluation Discussion

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
\]
Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & & & \ddots & \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(\text{now})0} & V_{(\text{now})1} & V_{(\text{now})2} & \cdots & V_{(\text{now})d}
\end{pmatrix}
\]
Motivation Design Evaluation Discussion

Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
\]

Main idea: Compare each dimension of the abnormal data points and normal ones.
Monitoring Component - Metrics Identification

Goal: Pinpoint the abnormal dimensions of suspicious data points to assist Orchestration component.

\[
\begin{pmatrix}
V_{00} & V_{01} & V_{02} & \cdots & V_{0d} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
V_{(n-2)0} & V_{(n-2)1} & V_{(n-2)2} & \cdots & V_{(n-2)d} \\
V_{(n-1)0} & V_{(n-1)1} & V_{(n-1)2} & \cdots & V_{(n-1)d} \\
V_{(now)0} & V_{(now)1} & V_{(now)2} & \cdots & V_{(now)d}
\end{pmatrix}
\]

Main idea: Compare each dimension of the abnormal data points and normal ones.
Monitoring Component

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Orchestration Component

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Orchestration Component
Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;
Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;
Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;

RELEVANT MEMORY SPACE
Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;
- Raise alarm;
- Optionally, kill process.

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Orchestration Component

Virtual Machine Introspection (VMI)

- Introspect VM memory using existing VMI tools;
- Raise alarm;
- Optionally, kill process.
Evaluation

+ Implemented on the Eucalyptus Cloud platform;
Evaluation

+ Implemented on the Eucalyptus Cloud platform;

+ Modified Node Controller and Cloud Controller source code.
Recall the two questions:
Recall the two questions:

1. Monitor more efficiently?
Recall the two questions:

1. Monitor more efficiently?

2. Utilize the statistics for security purpose?
Evaluation

Recall the two questions:

1. Monitor more efficiently?
   ▶ Tracking Component

2. Utilize the statistics for security purpose?
Recall the two questions:

1. Monitor more efficiently?
   ▶ Tracking Component

2. Utilize the statistics for security purpose?
   ▶ Monitoring and Orchestration Component
Motivation Design Evaluation Discussion

Evaluation

Recall the two questions:

1. Monitor more efficiently?
   ▶ Tracking Component

2. Utilize the statistics for security purpose?
   ▶ Monitoring and Orchestration Component

Metrics monitored for each VM:

- The default 7 metrics monitored by Eucalyptus CloudWatch.
Evaluation - Tracking

A comparison on number of values sent by NC for each metric.

- **VM workload**: TPC-C benchmark on MySQL database;
- **Δ**: The average for each metric when VM is **idle**.
Evaluation - Tracking

A comparison on number of values sent by NC for each metric.

- VM workload: TPC-C benchmark on MySQL database;
- Δ: The average for each metric when VM is idle.
Evaluation - Monitoring

Experiment setting:

- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.
Evaluation - Monitoring

Experiment setting:

- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.
Evaluation - Monitoring

Experiment setting:

- 3 VMs being monitored: VM 1 idle, VM 2 and 3 run Apache web server;
- VM 2 and 3 are compromised as DDoS bots later.

<table>
<thead>
<tr>
<th>Dim (j)</th>
<th>vm1-d1</th>
<th>vm1-d2</th>
<th>vm1-d3</th>
<th>vm1-d4</th>
<th>vm1-d5</th>
<th>vm1-d6</th>
<th>vm1-d7</th>
<th>vm2-d1</th>
<th>vm2-d2</th>
<th>vm2-d3</th>
<th>vm2-d4</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd_j</td>
<td>23.70</td>
<td>-0.98</td>
<td>-0.98</td>
<td>-0.55</td>
<td>-0.57</td>
<td>4.27</td>
<td>3.76</td>
<td>9.14</td>
<td>64.18</td>
<td>65.05</td>
<td>3.50</td>
</tr>
<tr>
<td>stddev_j</td>
<td>0.78</td>
<td>0.42</td>
<td>0.58</td>
<td>0.00</td>
<td>0.67</td>
<td>0.00</td>
<td>0.71</td>
<td>3.17</td>
<td>8.01</td>
<td>8.30</td>
<td>0.00</td>
</tr>
<tr>
<td>meandiff_j</td>
<td>0.16</td>
<td>-0.26</td>
<td>-0.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dim (j)</th>
<th>vm2-d5</th>
<th>vm2-d6</th>
<th>vm2-d7</th>
<th>vm3-d1</th>
<th>vm3-d2</th>
<th>vm3-d3</th>
<th>vm3-d4</th>
<th>vm3-d5</th>
<th>vm3-d6</th>
<th>vm3-d7</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd_j</td>
<td>-0.51</td>
<td>-0.82</td>
<td>4.23</td>
<td>9.04</td>
<td>60.56</td>
<td>61.16</td>
<td>1.45</td>
<td>-0.56</td>
<td>1.89</td>
<td>-0.51</td>
</tr>
<tr>
<td>stddev_j</td>
<td>0.31</td>
<td>0.00</td>
<td>0.35</td>
<td>7.23</td>
<td>6.06</td>
<td>6.98</td>
<td>0.17</td>
<td>3.39</td>
<td>0.12</td>
<td>3.65</td>
</tr>
<tr>
<td>meandiff_j</td>
<td>0.39</td>
<td>-0.23</td>
<td>-0.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Metrics Identification Result

ATOM: Automated Tracking, Orchestration and Monitoring of Resource Usage in Infrastructure as a Service Systems
Evaluation - Orchestration

- Received a VMI request with information:
  - A possible network problem;
  - Similar patterns for VM 2 and 3.
Evaluation - Orchestration

- Received a VMI request with information:
  - A possible network problem;
  - Similar patterns for VM 2 and 3.

- Node Controller call existing VMI tools to introspect:
  - VM 2: Volatility found suspicious DDoS process;
  - VM 3: Same with VM 2?
  - Raise alarm to user;
  - Kill the processes automatically using StackDB if confirmed.
Discussion - Overhead

Overhead introduced:
▶ On NC:
  O (1) to apply tracking algorithm and call VMI tools;
▶ On CLC: Depending on the PCA algorithm used, polynomial to sliding window size and number of dimensions monitored.

Overhead saved:
▶ Significant amount of network traffic sending from NC to CC to CLC;
▶ Significant amount of memory space to be introspected by VMI.
Discussion - Overhead

Overhead introduced:

▶ On NC: $O(1)$ to apply tracking algorithm and call VMI tools;
▶ On CLC: Depending on the PCA algorithm used, polynomial to sliding window size and number of dimensions monitored.
Discussion - Overhead

Overhead introduced:

▶ On NC: $O(1)$ to apply tracking algorithm and call VMI tools;
▶ On CLC: Depending on the PCA algorithm used, polynomial to sliding window size and number of dimensions monitored.

Overhead saved:

▶ Significant amount of network traffic sending from NC to CC to CLC;
▶ Significant amount of memory space to be introspected by VMI.
Discussion - Possible Extension

Monitor more metrics;

Extend VMI tools to find more complicated attacks.
Discussion - Possible Extension

- Monitor more metrics;
- Extend VMI tools to find more complicated attacks.
Thank you.
Thank you.

Questions?