Online services are prone to failures.
Non-crashing failures often go unnoticed.
Error messages do not tell why failures occur.
Reproducing production site failures is difficult.
Record and replay imposes high overhead and has privacy concerns.
Performance bugs are notoriously difficult to diagnose.
No runtime feedback (e.g., stack trace, error msg).
No environment data.
Offline bug reproduction & debugging is HARD.

**Motivation and Problems**

- **Production site**: Infrastructure-as-a-service
- **Platform as a service**: Amazon
- **User**:

**Components**

- **Runtime UBL & PerfCompass & PerfScope**
  - UBL learns the normal behavior of the system using unsupervised learning.
  - Anomalies have pre-failure symptoms. Neighborhood area size is used to raise an alarm when pre-failure symptoms are seen.
  - Segment large raw traces into smaller units.
  - Process all execution units to extract fault features.
  - Analyze extracted features to differentiate internal vs. external faults.
  - Identify execution units affected by the bug.
  - Rank and manage bug related functions.

**Approach**

- **UBL**: Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems.
- **PerfCompass**: Toward Runtime Performance Anomaly Fault Localization for Infrastructure-as-a-Service Clouds.
- **PerfScope**: Practical Online Server Performance Inference in Production Cloud Computing Infrastructures.

**Insight**:

- **In situ Online Service Failure Path Inference in Production Computing Infrastructures**
- **RDE**: Replay Debugging for Diagnosing Production Site Failures.

**Online**

- **PerfCompass**
  - Dynamic shadow server creation.
  - Guided binary execution exploration.
  - Input synthesis with symbolic execution.

**Offline profiling**

- **UML**
  - Anomaly Internal/external fault?
  - Affected functions

- **Insight**: Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems.

**Just-in-time Insight & RDE**

- **Use live VM cloning to decouple analysis from the production run**.
- **Leverage the production environment data and runtime outputs as guidance to search the failure paths**.
- **Performs symbolic execution along the inferred path to compute and solve path constraints and synthesize the production site data.**

**Results Summary**

- UBL is able to consistently rank the anomaly cause in the top three metrics.

**PerfCompass**

<table>
<thead>
<tr>
<th>System name</th>
<th>Fault description</th>
<th>Fault impact factor</th>
<th>Fault onset time dispersion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache</td>
<td>Flag setting bug</td>
<td>50 +/- 0.5%</td>
<td>374 +/- 63ms</td>
</tr>
<tr>
<td>MySQL</td>
<td>Deadlock bug</td>
<td>40 +/- 0.9%</td>
<td>38 +/- 3ms</td>
</tr>
<tr>
<td>Squid</td>
<td>File access bug</td>
<td>83 +/- 1%</td>
<td>0.35 +/- 0.09ms</td>
</tr>
<tr>
<td>Cassandra</td>
<td>Endless loop</td>
<td>51 +/- 5.7%</td>
<td>25 +/- 0.08ms</td>
</tr>
<tr>
<td>Hadoop</td>
<td>Endless read</td>
<td>81 +/- 0.5%</td>
<td>23 +/- 6ms</td>
</tr>
<tr>
<td>Hadoop</td>
<td>Thread shutdown</td>
<td>85 +/- 0.5%</td>
<td>110 +/- 20ms</td>
</tr>
</tbody>
</table>

**PerfCompass**

- UBL learns the normal behavior of the system using unsupervised learning.
- Anomalies have pre-failure symptoms. Neighborhood size is used to raise an alarm when pre-failure symptoms are seen.
- Segment large raw traces into smaller units.
- Process all execution units to extract fault features.
- Analyze extracted features to differentiate internal vs. external faults.
- Identify execution units affected by the bug.
- Rank and manage bug related functions.

**References**


Supported in part by NSF CNS1513942 grant and NSF CNS1149445 grant.