Service Analytics: Concept and Applications

楼建光
Software Analytics Group, Microsoft Research
Dec 10, 2014
## Cloud Era

### Who Uses It | What Services are available | Why use it?
---|---|---
**Business Users** | EMail, Office Automation, CRM, Website Testing, Wiki, Blog, Virtual Desktop ... | To complete business tasks

### Developers and Deployers | Service and application test, development, integration and deployment | Create or deploy applications and services for users

### System Managers | Virtual machines, operating systems, message queues, networks, storage, CPU, memory, backup services | Create platforms for service and application test, development, integration and deployment

---

**SaaS**

**PaaS**

**IaaS**
Software is changing...

On-premise License
Small Scale

Online Services Subscription
Large Scale
How software is built & operated is changing

- Code centric
- In-lab testing
- Experience & gut-feeling
- Centralized development
- Long product cycle
- ...

- User centric
- Debugging in the large
- Data-driven decision making
- Distributed development
- Continuous release
- ...

Microsoft Research @ 20 Years
User-Centric Service

• Aspects of user satisfaction
  – Usability, reliability, availability, performance, security, privacy, power consumption, …

• User satisfaction as a key driving force for success
  – Prioritization guideline
  – Optimization target
  – Design goal

• Data-driven user satisfaction
  – Metrics
  – Measurement
  – Analysis
  – Action
Reality – 故障不可避免!?

- Google
  - Gmail 在2012/4，2012/6/07，2013/8/17，2014/1/24 等多次发生故障，影响超3300万人
  - 最近一次Google搜索故障，2014/8/27，搜啥都是车祸图片
- Microsoft
  - 2014/11/18 Windows Azure故障
- 微信
Service Quality Management

**Service-Quality Metrics**
- MTTR (mean time to repair)
- MTTF (mean time to failure)

**Managing Process**
- Incident management (ICM)
- Problem management
Incident Management: An Example

缺点:
1. 没有自动化
2. 故障只等着用户汇报
What is the Key?

Service engineering processes are moving to data-driven
Formulation: Service Analytics

Service analytics is to enable service practitioners to perform data exploration and analysis in order to quickly conduct service management tasks.
Logs Generated by An Online Service

Availability

Time

System Resource Measurements

System Events

Transaction Processing event logs

500+ metrics/5 minutes

100+ events/5 minutes

20,000+ log entries/5 minutes
Important Scenarios

Problem Detection
Detect potential issues based on system logs, events, counters, usage data, and customer support records

Problem Localization & Diagnosis
Identify the problem site for a service live site issue, or provide information to help pinpoint the potential causes

Problem Categorization & Prioritization
Categorize issues and failures to help understand the trend and prioritize management tasks
Example 1. mining invariants for service problem detection
Background

• Logs are the major source for telemetry and diagnosis
• Manually inspecting logs is not feasible
  – Large scale of system
  – High complexity of system
• Traditional rule/keyword based log analysis tools:
  – Heavily depend on the knowledge of operators
  – Difficult to keep rules updated when components are frequently revised or upgraded
Linear Program Invariant

• A predicate always holds the same value under different normal executions.
  – For example:

  \[ count(A) = count(B) = count(E) \]
  \[ count(B) = count(C) + count(D) \]
Invariant and Execution Path

$\text{count}(A) = \text{count}(B) = \text{count}(E)$  
$\text{count}(B) = \text{count}(C) + \text{count}(D)$

Sequential Execution: $\text{count}(A) = \text{count}(B) = \text{count}(E)$

Execution Branch: $\text{count}(B) = \text{count}(C) + \text{count}(D)$
Invariant Violation and Anomaly(1)

- A violation of invariant often indicates a system problem.

\[
\text{count(Enter)} \neq \text{count(Leave)}
\]

Problem on Critical Section Operations
Invariant Violation and Anomaly(2)

- Violated invariants often give diagnosis cues.

\[ count(A) > count(B) \]
\[ count(B) > count(C) + count(D) \]
Formulation of Invariant

- A linear invariant can be presented as a linear equation:

\[ a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_m x_m = 0 \]

where \( x_i \) is the message count of message \( i \).

- Given a set of logs, we have

\[
\begin{bmatrix}
1 & x_{11} & x_{12} & \cdots & x_{1m} \\
1 & x_{21} & x_{22} & \cdots & x_{2m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{n1} & x_{n2} & \cdots & x_{nm}
\end{bmatrix}
\theta = 0
\]

where

\[ \theta = [a_0, a_1, a_2, \ldots, a_m]^T \]
What Is A Meaningful Invariant?
-- Sparse Non-zero Coefficients

\[ c(B) = c(C) + c(D) \]

\[ c(A) = c(B) \]

are more meaningful than

\[ c(A) + 3c(B) - 2c(E) - 2c(C) - 2c(D) = 0 \]

Any vector in the Null Space of \( X \) is an invariant;
Only sparse invariants are interested.
What Is A Meaningful Invariant?

-- Integer Coefficients

Elementary workflow structures can be interpreted by integer invariants.

Sequential

Branch

Join

Integer invariants are easy to be understood by human operators.
Problem Statement

• Due to noise pollution, mining invariants is to find integer sparse solutions of regression.

\[
\begin{bmatrix}
1 & x_{11} & x_{12} & \ldots & x_{1m} \\
1 & x_{21} & x_{22} & \ldots & x_{2m} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{n1} & x_{n2} & \ldots & x_{nm}
\end{bmatrix}
\]

\[\theta = 0 \quad \text{arg min} \|X\theta\|_0\]

– Challenges:
  • A typical integer sparse regulation problem (NP-Hard)
  • Traditional method is to relax 0-norm to 1-norm. However, it cannot guarantee to find all invariants.
Learning Invariant Overview

Four Steps:
Auto log parsing, Message Grouping and Counting, Search Invariants, and Anomaly Detection
Example 2. Healing Online Service Systems via Mining Historical Issue Repositories
When a new issue occurred, how to leverage past diagnosis efforts, to identify proper healing action for the new issue?
## A Simple Example of An Issue

- **Symptoms**
  - Describing the particular sign and phenomena of the issue

- **Solution**
  - Recording diagnostic steps and resolution

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>Symptoms</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Title: Browse Homepage failed</td>
<td>• Diagnosis: SQL connection timeout, SQL-001 blue screen</td>
</tr>
<tr>
<td></td>
<td>• Datacenter: XXX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Type: Availability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Traces: Transaction logs</td>
<td></td>
</tr>
</tbody>
</table>

Simplified example of an issue
Characteristics of Logs

- Highly redundant events
  - E.g., 6 events \( x_1 \sim x_6 \) indicate the authentication failure
  - Bias issue representation

- Many irrelevant events to failure
  - E.g., event \( d \) indicates “SQL usage detection”
  - BUT Relevant to issues, e.g., appearing in only SQL-related issues
  - Downgrading discrimination of issue representation
    - e.g., one type of SQL issue needs to reboot SQL; another type needs to patch SQL

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>TX ID</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>A</td>
<td>A entering</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>A</td>
<td>created cookie</td>
</tr>
<tr>
<td>2</td>
<td>c</td>
<td>A</td>
<td>Site = *</td>
</tr>
<tr>
<td>3</td>
<td>d</td>
<td>A</td>
<td>Detected SQL usage</td>
</tr>
<tr>
<td>6</td>
<td>y_1</td>
<td>A</td>
<td>SQL-Exception</td>
</tr>
<tr>
<td>6</td>
<td>Z</td>
<td>A</td>
<td>A leaving</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>B</td>
<td>B entering</td>
</tr>
<tr>
<td>3</td>
<td>x_1</td>
<td>B</td>
<td>B is not sign</td>
</tr>
<tr>
<td>3</td>
<td>x_2</td>
<td>B</td>
<td>building authentication</td>
</tr>
<tr>
<td>4</td>
<td>x_3</td>
<td>B</td>
<td>create sign</td>
</tr>
<tr>
<td>4</td>
<td>x_4</td>
<td>B</td>
<td>create cookie</td>
</tr>
<tr>
<td>4</td>
<td>x_5</td>
<td>B</td>
<td>B does not valid</td>
</tr>
<tr>
<td>4</td>
<td>x_6</td>
<td>B</td>
<td>redirecting B</td>
</tr>
<tr>
<td>5</td>
<td>z</td>
<td>B</td>
<td>B leaving</td>
</tr>
</tbody>
</table>

Illustration of transaction logs
Our Approach

• Issue-signature extraction
  – Address the challenges posed by logs
• Similarity-metric definition
  – Cosine similarity based on Generalized Vector Space Model (GVSM)
• Healing-action adaptation
  – Structured healing action + fault localization
Signature Extraction

**An Issue**

- **Title:** Browse Homepage failed
- **Time:** 2012/06/25 13:04:33
- **Datacenter:** XXX
- **Type:** Availability
- **Traces:** Transaction logs

**Diagnosis:** SQL connection timeout, SQL-001 blue screen

**Rooting action:** Reboot SQL-001.

---

**Issue Signature**

<table>
<thead>
<tr>
<th>Index</th>
<th>Term (Event Set)</th>
<th>DMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X G T</td>
<td>0.34</td>
</tr>
<tr>
<td>2</td>
<td>N S O Y</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>B C</td>
<td>0.07</td>
</tr>
</tbody>
</table>

---

**Log Sequences**

- Parsing log messages

---

**Concept Lattice**

---

**Formal concept analysis**
- Reduce redundancy
- Extract trunk/branch relationship of execution paths

---

**Contrast analysis**
- Reduce weak-discrimination
- Measure correlation with Delta Mutual Information (DMI)

---

ASE 2012, Essen
Issue Comparison

Similarity definition: (Generalized Vector Space Model)
- Similarity between terms
- Similarity between signatures
  - Combine term similarity
  - Encode importance of term using DMI as weights
Healing-Action Adaptation

- Triple structure
  - $<\text{verb, target, location}>$
- Verb & Target
  - E.g., “recycle + AppPool”, “Reset + IIS”
  - Extracted from retrieved similar issues by analyzing their solution descriptions
- Location
  - Specific machine/server name, e.g., SQL23524-001
  - Obtained by applying fault-localization techniques
Evaluation

• 332 issues collected in time period: 11/01/2011~02/18/2012

• 146 issues with documented healing actions and recorded logs

69 issues on service upgrade

77 issues on service interruption or degradation
  – used in evaluation

---Internal production service: ServiceX

• Effect of our techniques on overall effectiveness
  Approach1: Ignore highly-correlated phenomenon (Mutual information + VSM)
  Approach2: Ignore weakly-discriminative phenomenon (FCA + TF-IDF + VSM)
  Our approach: FCA + contrast analysis

Overall ROC curves
Summary

• Mission of Service Analytics
  – Utilize data-driven approach to help create highly performing, user friendly, and efficiently built & operated online services

• Service Analytics is naturally tied with state of engineering practice of service

• Empowering future software practitioners with data analytics mindset & skills
Advertisement

• We are recruiting!
  – Software analytics researchers (Full-time employee, visiting researchers)
  – Software analytics interns