Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

André van Hoorn
Abteilung Software Engineering
Fakultät II - Department für Informatik
November 8, 2007

Motivation

- Availability of Enterprise Information Systems (e.g., banking & online shopping systems) is critical QoS requirement
- Anomaly detection is means for failure detection and diagnosis to improve availability
- Existing anomaly detection approaches based on timing behavior do not explicitly consider varying workload

Structure

1. Motivation
2. Foundations
3. Hypothesis & Goals
4. Results
5. Conclusions
6. Related Work

Foundations

1. Performance Metrics and Scalability
   - Response Time
     - Time interval elapsed between issued request and respective response
   - Execution Time
   - Throughput
     - Rate at which a system (resource) handles tasks
   - (Client-/Server-side)
   - Think Time, ...
   - Resource Utilization

2. Workload and Scalability
   - Workload
     - Amount of work currently requested from or processed by a system
     - Characteristics
       - Workload intensity
       - Service demand characteristics
   - Scalability
     - "Ability of a system to continue to meet its response time or throughput objectives as the [workload] increases" [SW01]
### Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

#### Hypothesis & Goals

**Assuming that varying workload implies varying response times:**

**Hypothesis**

Novel workload-sensitive anomaly detection based on response times realizable if varying workload intensity has characteristic impact on response time distributions.

#### Related Work

- **Motivation:** Availability important QoS attribute
  - Availability: \[ \text{MTTF} = \frac{\text{MTTF}}{\text{MTTR}} \]
- **Goal:** Improve availability by reduction of repair times
- **Strategy:** Use unusual behavior as indicator for failures

**Common approach for software systems:**
- Build model of “normal behavior”
- Monitor current behavior
- Detect deviations

### Anomaly Detection

- **Statistics**
  - Minimum, maximum
  - Sample mean, sample variance
  - \( p \)-Quantile \( x_p = \min(x | F(x) \geq p) \)
  - 1–3, quartiles: \( x_{0.25}, x_{0.5}, x_{0.75} \) (Median), \( x_{0.75} \)
  - Mode, skewness, ...

**Other distribution characteristics:**
- uni-/bi-/multimodal
- (a)symmetric
- left-right-skewed

### Descriptive Statistics

#### Parametric Distribution Families (Examples)

- **Normal Distribution**
  - 2-parameter: \( N(\mu, \sigma^2) \)

- **Log-normal Distribution**
  - 2-parameter: \( \Lambda(\tau, \mu, \sigma^2) \)
  - 3-parameter: \( \Lambda(\tau, \mu, \sigma^2) \)
Hypothesis & Goals

Project Goals I

1. Probabilistic Workload Driver
   - Develop application-generic methodology for generating realistic user behavior (e.g., based on probabilistic model)

2. Case Study with Response Time Analysis
   - Apply & evaluate workload generation technique
   - Obtain workload-dependent response times from sample application
   - Statistically analyze impact of workload on response times

3. Workload-Sensitive Anomaly Detection Prototype
   - Compute degree of anomaly for operation executions
   - Implementation of workload-sensitive AD prototype

Probabilistic Workload Driver – Approach

- **Challenge:** Generate valid sessions
- **Constraint:** Realistic behavior (not: “capture & replay”)
- **Approach:**
  - Workload configuration data model separated into
    - Application Model
    - User Behavior Model
    - User Behavior Mix
    - Workload Intensity
  - High-level design
  - Iterative execution model
  - Session model composition semantics
  - Implementation: Markov4JMeter (JMeter extension)

Application Model

- **Session layer** models allowed sequences of service calls in a session
- **Protocol layer** contains all protocol-specific (e.g., HTTP) request details

Figure: Sample application model illustrating separation into session and protocol layer.
User behavior model corresponds to specific application model
- Markov chain models probabilistic behavior within a session
- States correspond to states of session layer
- Includes definition of (client-side) think time

Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

Markov4JMeter

- Implemented workload driver as extension for existing workload tool Apache JMeter
- Markov4JMeter [vH07] released under GPL
- Feedback:
  
  "Markov4JMeter has worked very well for us. We have used it in several scripts for the last two months. There have been no bugs. The add-in should be made a part of the JMeter distribution."

Markov McWhinney, Portata, Inc., Mountain View, CA (Sep 9, 2007)

Markov chain models probabilistic behavior within a session

Results

Probabilistic Workload Driver

User Behavior Model

Markov chain

Figure: User behavior model $\pi_{10}$

Figure: User behavior model $\pi_{11}$

Markov4JMeter profile for JPetStore

User Behavior Mix, Workload Intensity

User Behavior Mix
- Assignment of user behavior models $\pi_j$ to application model $A$ with relative frequencies $p_i$
- Formally, $BM(T) A = \{ (\pi_j, p_i) \ldots (\pi_n, p_{n-1}) \}$

Workload Intensity
- Duration
- Function $R_{\geq 0} \rightarrow N$ defining number of concurrent users

Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

Results

Probabilistic Workload Driver

Case Study with Response Time Analysis

Structure

Motivation
- Performance Metrics and Scalability
- Workload Characterization
- Probability and Statistics

Hypothesis & Goals

Results
- Probabilistic Workload Driver
- Case Study with Response Time Analysis
- Workload-sensitive Anomaly Detection Prototype

Conclusions

Related Work

Sample Application

iBatis JPetStore
- Online shopping store
- 3-layer architecture
  - Presentation layer
  - Service layer
  - Persistence layer
- Deployment
  - Application Server
  - Database Server

Markov4JMeter profile for JPetStore

- Identified 29 request types
- Grouped request types into 15 services

<table>
<thead>
<tr>
<th>Service</th>
<th>Request Type</th>
<th>Service</th>
<th>Request Type</th>
<th>Service</th>
<th>Request Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkout</td>
<td>http/submit</td>
<td>Login</td>
<td>http/login</td>
<td>Checkout</td>
<td>http/submit</td>
</tr>
<tr>
<td>Signon</td>
<td>http/signup</td>
<td>Purchase</td>
<td>http/purchase</td>
<td>SignIn</td>
<td>http/login</td>
</tr>
<tr>
<td>Add to Cart</td>
<td>http/addCart</td>
<td>New Order</td>
<td>http/newOrder</td>
<td>AddAccount</td>
<td>http/addAccount</td>
</tr>
<tr>
<td>Remove Item</td>
<td>http/removeItem</td>
<td>Payment</td>
<td>http/purchase</td>
<td>Login</td>
<td>http/login</td>
</tr>
<tr>
<td>Update Cart Quantities</td>
<td>http/updateCart</td>
<td>List Orders</td>
<td>http/viewOrders</td>
<td>View Item</td>
<td>http/viewItem</td>
</tr>
<tr>
<td>Edit Account</td>
<td>http/editAccount</td>
<td>User Profile</td>
<td>http/userProfile</td>
<td>Edit Account Form</td>
<td>http/editAccountForm</td>
</tr>
</tbody>
</table>

iBatis JPetStore

- Focused on services of "typical user sessions"
- 9 services / 13 request types (labeled by †)
Markov4JMeter Profile for JPetStore (cont’d)

Application Model

Figure: Session layer of application model and protocol states of 2 application states.

Test Plan

Markov4JMeter Profile for JPetStore (cont’d)

Experiment Configuration

Platform Workload Intensity Metric (PWI)

Figure: Graphs visualizing active traces history and PWI.

Response Time Analysis

1. Analyzed impact of PWI on response time statistics
   - Minimum, maximum, mean, variance, and standard deviation
   - Mode
   1. quartile, median, 3. quartile
   - Skewness, and
   - Outlier ratio.

2. Distribution fitting
Results

Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

Experiment Report

Automatic generation of
- Plots for each experiment run and operation
- PWI vs. response time statistics for each operation

Results (1/2)

- Workload intensity impacts (most) response time statistics
  - Maximum very sensitive
  - Mean more sensitive than median
  - Upper quartiles more sensitive than lower quartiles
  - Increasing IQR
  - Minimum largely unaffected
  - Observed no correlation with outlier ratio

Results (2/2)

- Distributions right-shifted, long-tailed, right-skewed
- Most monotonically increasing curves show characteristic "performance knees" by Jain (1991) [Jai91]
- Identified 4 distribution shapes
  - Bimodal with 2 major clusters
  - Bimodal with minor and major cluster
  - Multimodal becoming unimodal
  - Unimodal
- Indication for need of probabilistic workload
- In large parts, 3-parameter log-normal distribution fits left sides of unimodal data samples

Distribution Fitting with 3-parameter Log-normal Distr.

- In large parts, 3-parameter log-normal distribution fits left sides of unimodal data samples
- In most cases, tails of response time samples shorter than those of estimated distribution

Bimodal Distribution Shapes
Bimodal Distribution Shapes

Anomaly Detection in Software Timing Behavior

- Anomaly considered response time exceeding a given threshold $\tau$
- Execution of operation $o$ is tuple $(o, st, rt)$
- Anomaly detector (AD) must decide for execution whether or not it is an anomaly (based on historical data)
- Quality of AD: Error rate with type I/II errors

Plain Anomaly Detector (PAD) (1/2)

- PAD classifies an execution as anomalous iff its response time exceeds operation-specific threshold $\tau$
  \[
  \text{PAD}(e) := \begin{cases} 
  1, & rt > \tau \\
  0, & \text{else}
  \end{cases}
  \]

Plain Anomaly Detector (PAD) (2/2)

- Example 2: PAD with Varying Workload Intensity
- Example 1: PAD with Constant Workload Intensity

Error rate is 0 for $\tau \in [106.4, 144.9]$
Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

Results

Workload-Intensity-sensitive Anomaly Det. (WISAD)

WISAD explicitly considers varying workload intensity by including:
1. Platform workload intensity (PWI) during time of execution
2. Workload intensity normalization factor

Example 3: WISAD with Varying Workload Intensity

Error rate 0 for threshold values between 106–118

Conclusions

1. Probabilistic Workload Driver
   - Methodology for probabilistic workload modeling based on Markov chains
   - Design resulted in Markov4JMeter [vH07] (GPL-licensed)

2. Case Study with Response Time Analysis
   - Evaluated Markov4JMeter approach
   - Executed large number of experiments with varying workload intensity
   - Analyzed workload intensity vs. response time statistics

3. Workload-sensitive Anomaly Detection Prototype
   - AD prototype which considers varying workload intensity
   - Evaluation with "real" data is work in progress [RvHGH07]
Workload-sensitive Timing Behavior Anomaly Detection in Large Software Systems

Structure

1 Motivation
2 Foundations
3 Hypothesis & Goals
4 Results
5 Conclusions
6 Related Work

Related Work

- **Workload Generation**
  - Workload characterization of system in productional use, e.g. Artill et al. [AKR01], Menascé et al. [MAR07], [MA03]
  - Customer Behavior Model Graph (CBMG) by Menascé et al. [MAFM99]
  - Extended Finite State Machine (EFSM) by Shams et al. [SKF06]
  - Freely available and commercial workload generators, e.g. Mercury LoadRunner [Men07], OpenSTA [Ope05], Siege [Ful06]

- **Response Time Analysis**
  - Time analysis of ERP systems by Mielke [Mie02]
  - Timing Behavior Anomaly Detection
  - Agarwal et al. [AAG+04]

Related Work

- **Bonus Scenes**

Box-and-Whisker Plot

- **Visualizes**
  - Quartiles
  - Interquartile range (IQR)
  - Normal and extreme outliers

- **Figure: Description of a box-and-whisker plot [MR06]**
Density Estimation

Density Estimation:
- Goal: Estimate underlying density function \( f \)
- Parametric (based on parametric distribution family)
- Non-parametric (e.g. kernel density estimation [Sil66])

\[ \hat{f}(x) = \frac{1}{n h} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) \]

**Figure**: Kernel density estimations of a data sample using a normal kernel and window sizes 2 and 20.

Application Model

Workload Configuration Data Model

- Contains all information to generate valid sessions
- 2-layered hierarchical state machine

**Session Layer**
- Non-det. finite state machine
- Application transitions can be labeled with guards and actions
- Transitions represent valid sequence of service calls in session

**Protocol Layer**
- Contains required protocol details for session generation
- Det. state machine for each application state
- Again: Transitions can be labeled with guards and actions

High-level Design

**Architecture and iterative execution model**
**Session model composition**

Instrumentation of JPetStore

**Table**: Identified monitoring points and coverage of request types.

<table>
<thead>
<tr>
<th>Request Type</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>service.CatalogService.getProductListByCategory(String)</td>
<td>1</td>
</tr>
<tr>
<td>service.CatalogService.getItemListByProduct(String)</td>
<td>1</td>
</tr>
<tr>
<td>service.CatalogService.getItem(String)</td>
<td>1</td>
</tr>
<tr>
<td>service.AccountService.getAccount(String, String)</td>
<td>1</td>
</tr>
<tr>
<td>presentation.OrderBean.newOrder()</td>
<td>1</td>
</tr>
<tr>
<td>presentation.CatalogBean.viewItem()</td>
<td>1</td>
</tr>
<tr>
<td>presentation.CatalogBean.viewCategory()</td>
<td>1</td>
</tr>
<tr>
<td>presentation.CartBean.addItemToCart()</td>
<td>1</td>
</tr>
<tr>
<td>presentation.AccountBean.signon()</td>
<td>1</td>
</tr>
<tr>
<td>persistence.sqlmapdao.OrderSqlMapDao.insertOrder(Order)</td>
<td>1</td>
</tr>
<tr>
<td>persistence.sqlmapdao.ItemSqlMapDao.getItem(String)</td>
<td>1</td>
</tr>
<tr>
<td>persistence.sqlmapdao.AccountSqlMapDao.getAccount(String, String)</td>
<td>1</td>
</tr>
<tr>
<td>struts.action.ActionServlet.doPost(HttpServletRequest, HttpServletResponse)</td>
<td>1</td>
</tr>
<tr>
<td>struts.action.ActionServlet.doGet(HttpServletRequest, HttpServletResponse)</td>
<td>1</td>
</tr>
</tbody>
</table>
Constructive Definition of PWI

1. Trace history \( H \subseteq \mathbb{N}^t \) with tuples of trace start and stop times
2. Event history \( E \subseteq \mathbb{N} \times \{-1, 1\} \)
3. Active traces history \( A \subseteq \mathbb{N}^2 \)
   \( A := \{(t, k) \in \mathbb{N}^2 \mid \exists a \in \{-1, 1\}, (t, a) \in E \wedge k = \sum_{a \in \{-1, 1\}} b\} \)
4. Step function \( \text{activeTraces}_k : N \rightarrow N \)
   \( \text{activeTraces}_k(t) = \left\lfloor k, \exists t' \in N : t' = \max\{r \in \{r' \in \mathcal{R} \mid t \leq t', (r, k) \in A\} \}
   \text{ else.} \)
5. Platform workload intensity \( \text{pw}_A(t) : N \rightarrow \mathcal{R}^+ \)
   \( \text{pw}_A(t) = \sum_{i=1}^{t} \text{activeTraces}_i(t - i) \)
**Anomaly Detection in Software Timing Behavior**

- **Anomaly** considered response time exceeding a given mean value by $\alpha$ percent in a period $\beta$
- Execution of operation $o$ is tuple $\langle o, st, rt \rangle$
- **Anomaly detector** (AD) must decide for all executions in set of executions $\mathcal{X}$ whether or not it is an anomaly
- It knows set of observations $\mathcal{X}$ (History) assumed to contain no anomalies
- AD decides by comparing $\mathcal{Y}$ with $\mathcal{X}$
- Quality of AD: Error rate with type I/II errors

**Example 1: PAD with Constant Workload Intensity**

Synthetic workload scenario with
- Single operation
- **Constant workload intensity**

- Error rate is 0 for $\tau \in [106.4, 144.9]$
- Assuming $\bar{P}_{o} = 100, \delta \in [1.06, 1.44]$

**Example 2: PAD with Varying Workload Intensity**

Synthetic workload scenario with
- Single operation
- **Increasing workload intensity**

- Minimum error rate is 8% ($\tau > 176$)
- But then: No anomaly detected

**Workload-Intensity-sensitive Anomaly Det. (WISAD)**

- Explicitly considers varying workload intensity by including
  - Function $pwf : \bar{N} \times \bar{N} \rightarrow \mathbb{R}$
  
  \[ pwf(e) = \frac{1}{n} \sum_{t=1}^{n} \text{activeTraces}_{t}(t) \]

  - Function $wnf_{o} : \mathbb{R} \rightarrow \mathbb{R} ; w \rightarrow wnfo_{o}(w)$
    
    For a given workload intensity $w$, $wnfo_{o}(w)$ is a workload intensity normalization factor for the response time threshold that applies to a workload intensity of 1.

- Given history $\mathcal{X}$, set of executions $\mathcal{Y}$, $e = \langle o, st, rt \rangle \in \mathcal{Y}$, historical sample mean $\bar{P}_{o,1}$ of $o$ at $w$ 1

  \[ WISAD(e) = \begin{cases} 1, & rt > \bar{P}_{o,1} + wnfo_{o}(pwf(e)) \times \delta \\ 0, & \text{else} \end{cases} \]

**Example 3: WISAD with Varying Workload Intensity**

Synthetic varying workload scenario from Example 3
- Values of $pwf$ follow the equation $1 + \frac{\tau}{rto}$

  \[ wnfo_{o}(w) = \frac{w}{wto} \]  

  and $\bar{P}_{o,1} = 100$

- Error rate 0 for threshold values between 106–118