e-Business Threshold Management for Proactive Problem Determination
Anca Sailer and Gautam Kar, IBM T. J. Watson Research Center
Soila Pertet and Priya Narasimhan, Carnegie Mellon University

Motivation

- Traditionally, problem determination is related to the state of components at the system level (e.g., CPU, memory)
- If problems manifest at the transaction level, the management service has no knowledge on the cause of the problem

Autonomic Problem Determination (IBM)

We build on work done by Sailer et al that pinpoints the root cause of performance degradation in multi-tier distributed systems

Experimental Observations

- The c-SLO does not need to be recomputed if the system operates within the linear part of the response time curve
- Precision of problem determination in instrumented code = 100%
- Precision of non-instrumented code ≈ 50% -100%

Proactive Fault-Tolerance (CMU)

Determine if we can exploit performance monitoring to identify pre-fault symptoms and initiate proactive (rather than reactive) actions for faster fault-recovery

Pertet and Narasimhan developed a proactive fault-recovery strategy for distributed middleware systems in the presence of resource-exhaustion faults.

Problem Determination Steps

1. Monitor transaction-related resource dependencies and resource behavior performance models
2. Use monitored data to decompose transaction-SLOs (T-SLOs) into component-SLOs (c-SLO)
3. Localize the root-cause of problem based on the degree to which components violate their constructed c-SLO

A. Decompose Transaction-SLO into Component-SLO

NO SLO VIOLATIONS
1. Read response time RT for component C
2. Lookup graph to create Tc set [T1, T2] of transactions that depend on C
3. \[ T_{load}(C) = \sum_{i} \frac{RT_{i}}{compsdependent_{i}} \]
4. \[ PC = \frac{T_{load}(C)}{RT_{bad}} \]
5. \[ c\text{-SLO} = T_{SLO} \ast \text{avg}(PC) \]

B. Detect c-SLO Violations and Localize Root Cause

SLO VIOLATION
1. If \( T_{Local} > c\text{-SLO} \) then \( \text{bad}_{avg}(C) = \text{bad}_{avg}(C) + (1 - \alpha) \text{Local} \)
2. If \( \text{bad}_{N}(C) = b_{avg}(C) / c\text{-SLO} \)
3. Sort nodes by severity value
4. When problem is diagnosed \( \text{Reset bad}_{N}(C) = \text{bad}_{avg}(C) = 0 \)

New Collaborative Approach

Research Question: What is the minimal level of system monitoring required to proactively diagnose performance problems in transaction-oriented distributed applications?

- How do we detect SLO violations based on metrics other than response time? (e.g., resource usage, throughput violations)
- How much monitoring and instrumentation is required to effectively detect SLO violations of various kinds?
- Should we monitor everything or get partial data and extrapolate values through statistical inference or machine learning?
- What granularity of monitoring is most appropriate to detect SLO violations for different metrics?
- Does looking at more metrics provide higher accuracy in problem determination, and potentially more focused recovery?
- Can system monitoring help us identify patterns of abnormal behavior to enable proactive fault-tolerance?