Automated Analysis of the Results of Performance Regression Tests

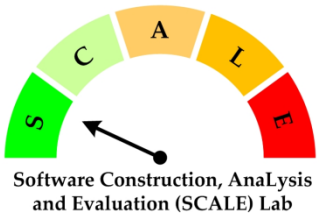
SPEC DevOps Performance WG

Friday, 05/29/2015

Zhen Ming (Jack) Jiang

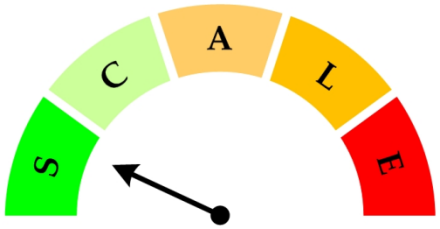
Software Construction, AnaLysis and Evaluation (SCALE) Lab

York University, Canada



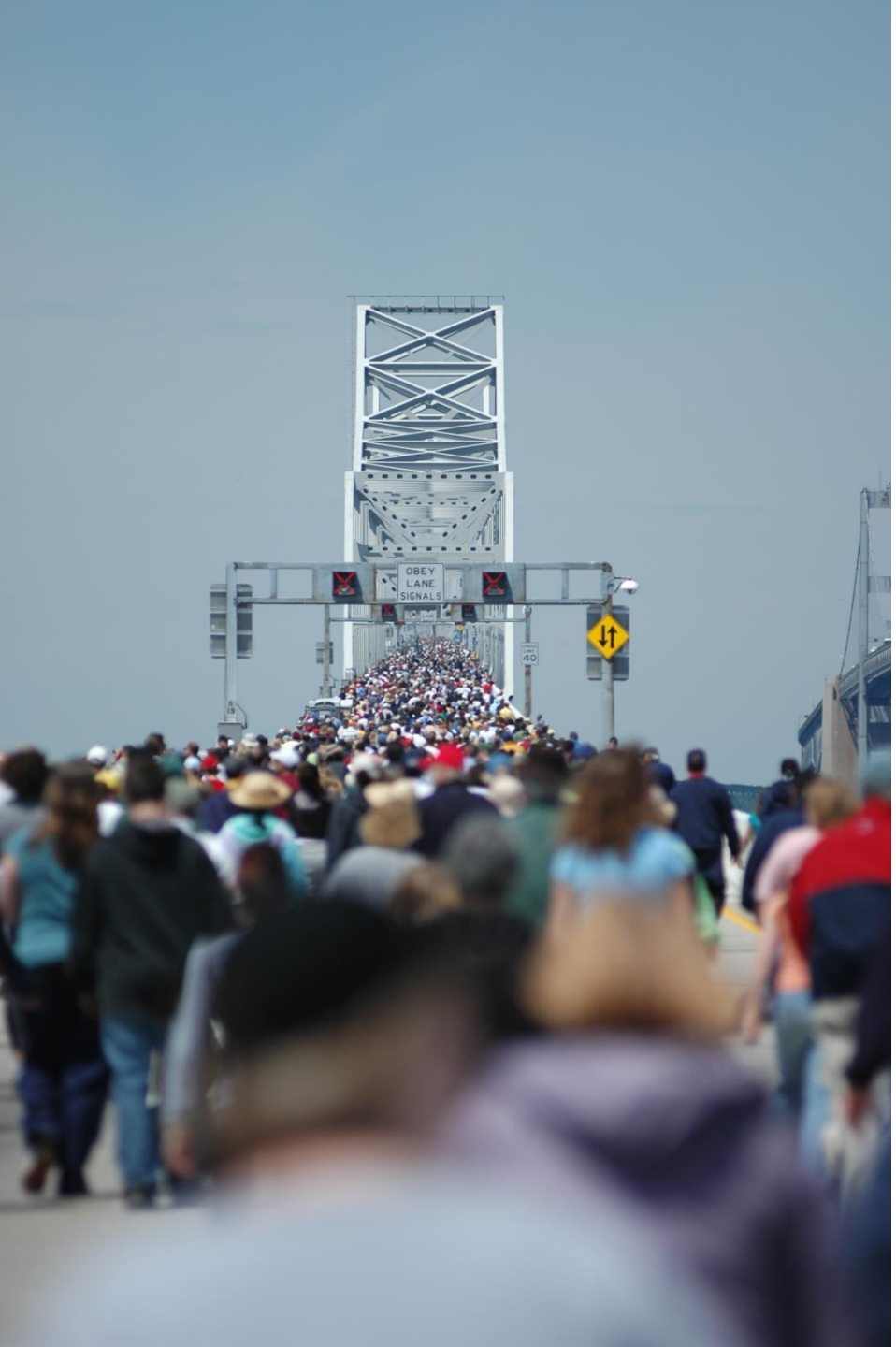
Zhen Ming (Jack) Jiang

- Assistant Professor, Department of Electrical Engineering and Computer Science, Lassonde School of Engineering, York University. Toronto, ON, Canada.
- Head of Software Construction, AnaLytics and Evaluation (SCALE) Lab
- Co-founder and co-organizer for the International Workshop on Large-Scale Testing (LT)
- Research Interests:
  - Software Performance Engineering
  - Mining Software Engineering Data



Software Construction, AnaLysis and Evaluation (SCALE) Lab

LT 2012 – present



Most field problems for large scale systems are *rarely functional* instead they are *load-related*

at&t

mobile me

BlackBerry

amazon web services

Microsoft Office 365

PayPal

Software Testing

Unit Testing

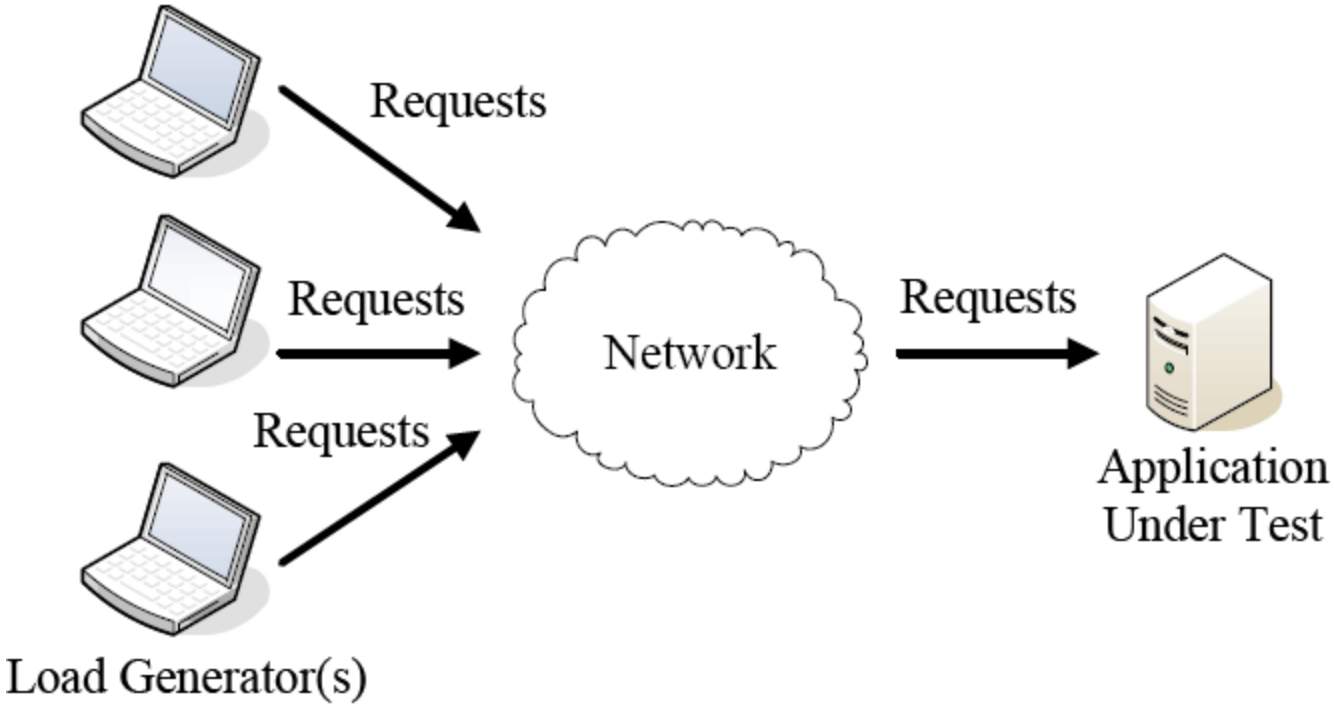
System Integration Testing

(Functional) Regression Testing

Performance Regression Testing

Non-Functional Testing

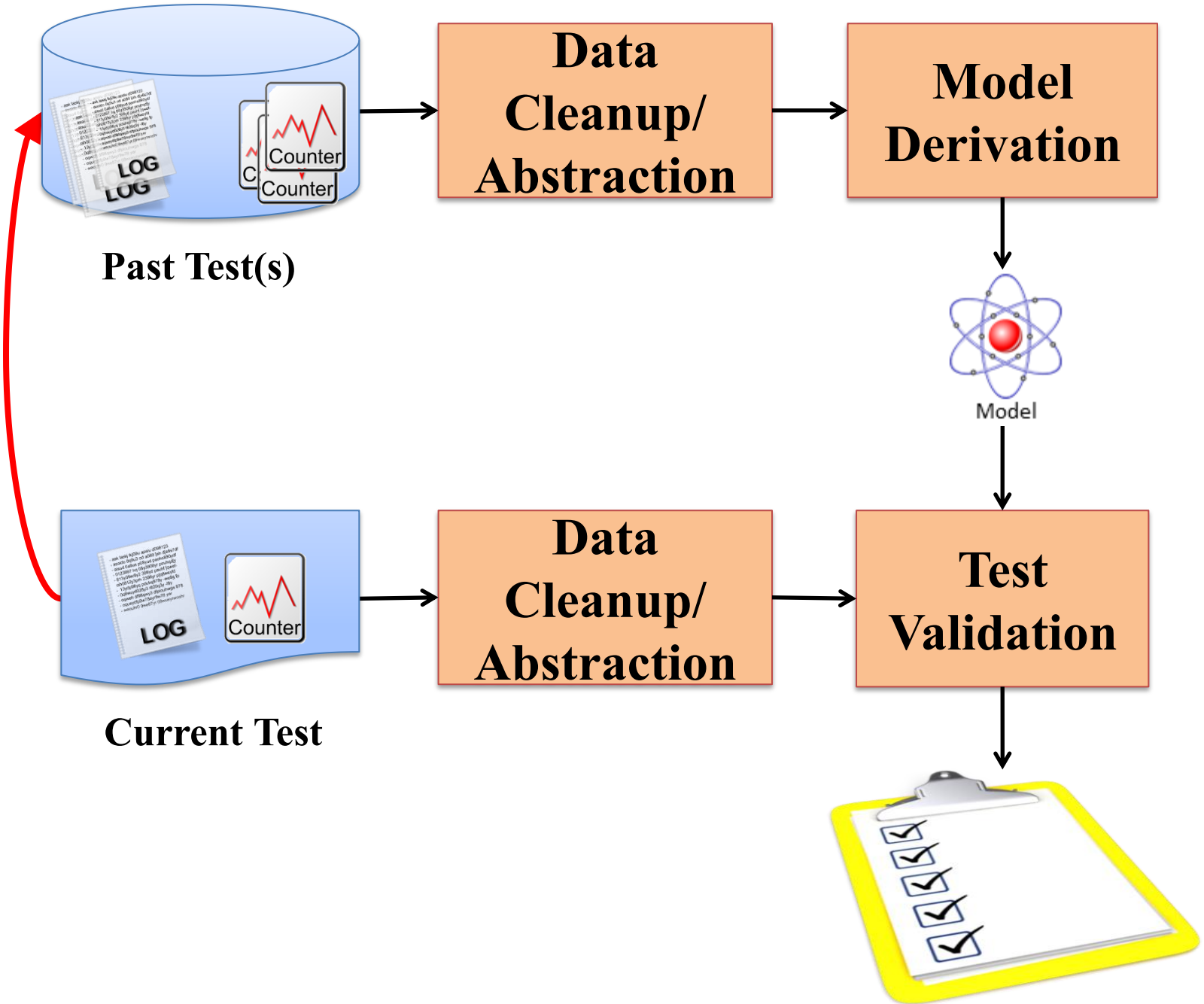
Performance Regression Testing



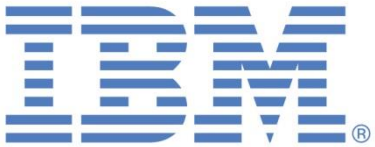
Mimics multiple users repeatedly performing the same tasks

Take hours or even days

Is the system ready for release?


LOG
LOG
Counter
Counter
Past Test(s)
Data Cleanup/ Abstraction
Model Derivation
Model
LOG
Counter
Current Test
Data Cleanup/ Abstraction
Test Validation

Studied Systems



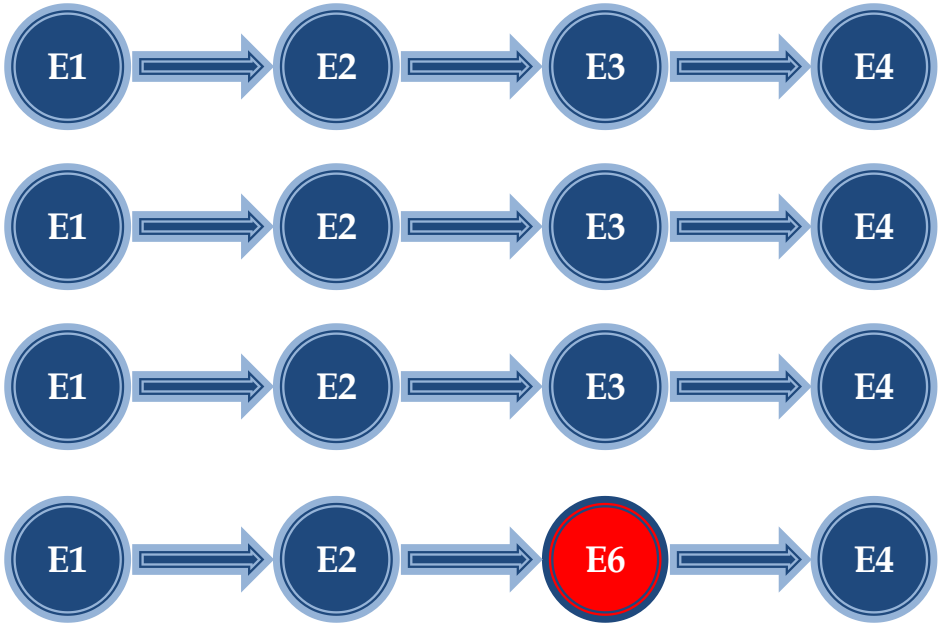
Execution Logs

#	Log Lines
1	time=1, thread=1, session=1, receiving new user registration request
2	time=1, thread=1, session=1, inserting user information to the database
3	time=1, thread=2, session=2, user=Jack, browse catalog=novels
4	time=1, thread=2, session=2, user=Jack, sending search queries to the database
5	time=3, thread=1, session=1, user=Tom, registration completed, sending confirmation email to the user
6	time=3, thread=2, session=2, database connection error: session timeout
7	time=4, thread=1, session=1, fail to send the confirmation email, number of retry = 1
8	time=6, thread=2, session=2, user=Jack, successfully retrieved data from the database
9	time=7, thread=2, system health check
10	time=8, thread=1, session=1, registration email sent successfully to user=Tom
11	time=9, thread=2, session=3, user=Tom, browse catalog=travel
12	time=10, thread=2, session=3, user=Tom, sending search queries to the database
13	time=10, thread=3, session=4, user=Jim, updating user profile
14	time=11, thread=3, session=4, user=Jim, database error: deadlock

Automated Functional Analysis

- **(E2, E3) are always together:**
  - (acquire_lock, release_lock)
  - (open_inbox, close_inbox)

- **If we see (E2, E6) this might be a problem**



Dell DVD Store

#	Z-Stat	Kinds	Min	Max	Total	Event
						SessionID=19420, Entering purchase for simple quantity queries

Freq	Sample	Details (Sort by Freq)
87,528 (98%)	ds2logs.txt 688 ds2logs.txt 689	E_{13} --> SessionID=19420, Entering purchase for simple quantity queries E_{14} --> SessionID=19420, Initial purchase, update cart
1,436 (<1%)	ds2logs.txt 2,484 ds2logs.txt 2,488	E_{13} --> SessionID=16242, Entering purchase for simple quantity queries E_{13} --> SessionID=16242, Entering purchase for simple quantity queries
358 (<1%)	ds2logs.txt 10,020 ds2logs.	E_{13} --> SessionID=13496, Entering purchase for simple quantity queries ore commit

E_{19}	34.73	2	317	16,273		
E_{22}	20.65	2	1	3,857		

99.99% reduction in viewed log lines

Automated Performance Analysis

- **Each load test has periods of peak/light load**
  - Fluctuations of response time

Test # 1

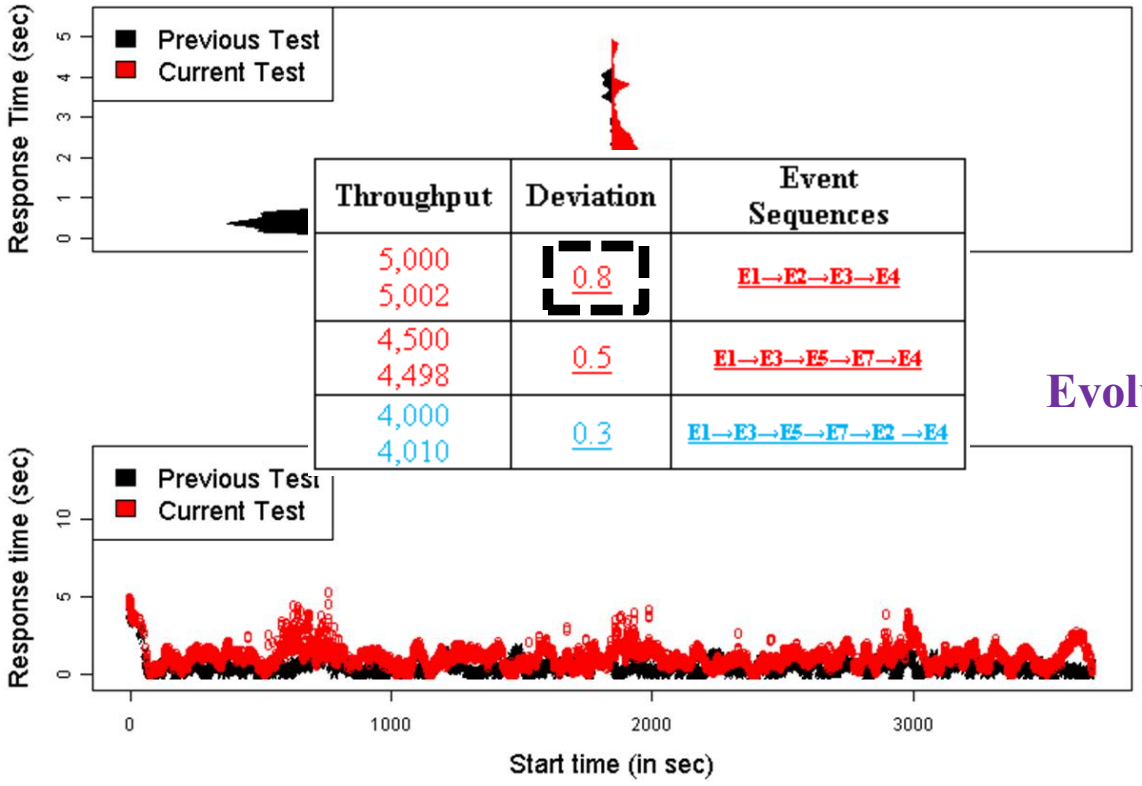
Test # 2

Test # 3

- **Same workload applied across different tests**
  - Similar response time distributions

Performance Analysis Report - Visual Comparison

Throughput	Deviation	Event Sequences
5,000 5,002	0.8	E1→E2→E3→E4
4,500 4,498	0.5	E1→E3→E5→E7→E4
4,000 4,010	0.3	E1→E3→E5→E7→E2→E4

Throughput	Deviation	Event Sequences
5,000 5,002	0.8 Distribution Response Time Distribution  Evolution	E1→E2→ E3→E4
4,500 4,498	0.5	E1→E3→E5→ E7→E4
4,000 4,010	0.3	E1→E3→E5→ E7→E2→E4

Throughput	Deviation	Event Sequences
5,000 5,002	0.8	E1→E2→E3→E4
4,500 4,498	0.5	E1→E3→E5→E7→E4
4,000 4,010	0.3	E1→E3→E5→E7→E2→E4

Throughput	Deviation	Event Sequences
5,000 5,002	0.8	E1→E2→E3→E4
4,500 4,498	0.5	E1→E3→E5→E7→E4
4,000 4,010	0.3	E1→E3→E5→E7→E2→E4

Performance Analysis Report - Step-wise Performance Diagnosis

Throughput	Deviation	Event Sequences
5,000 5,002	0.8	E1→E2→E3→E4
4,500 4,498		
4,000 4,010		

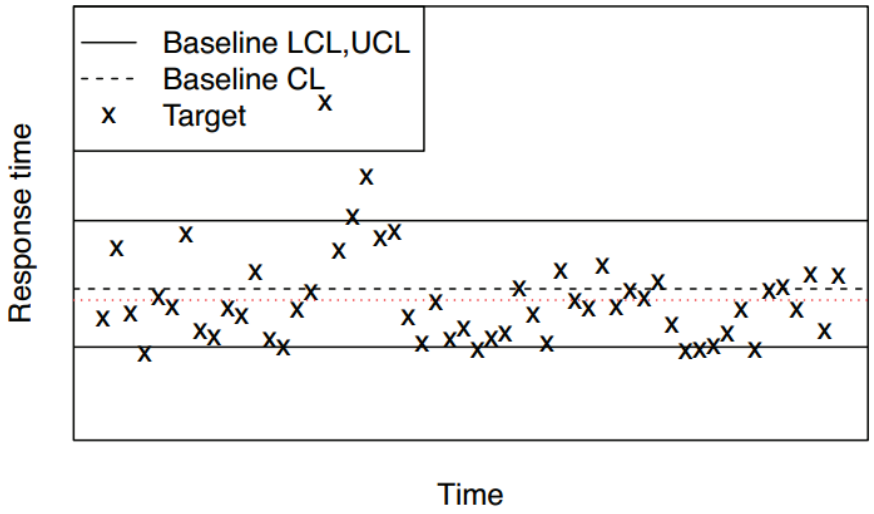
Average	Deviation	Event Sequences — Stepwise
		E1: sessionId=2, time=1, user=Tom, login successfully
1.0 1.1	0.1	**E2:** sessionId=2, time=1, user=Tom, browse catalog, catalog=bird
0.5 1.8	0.9	**E3:** sessionId=2, time=2, user=Tom, add item, item=parrot, quantity=1
0.5 0.5	0.1	**E4:** sessionId=2, time=3, user=Tom, user checkout successfully

We discovered a MySQL performance bug

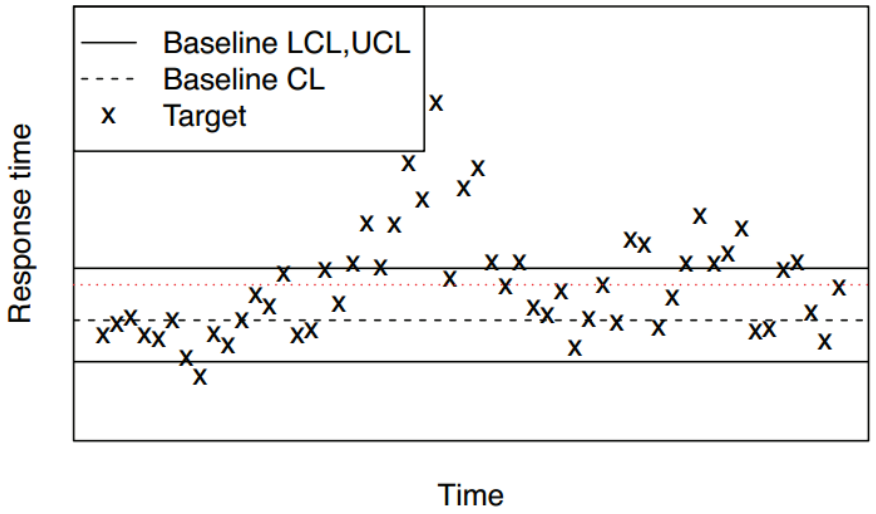
Performance Counters

	A	B	C	D	E
1	Time	Disk Reads/sec	Disk Writes/sec	Page Faults/sec	Memory
2	2/29/08 16:58	0.049986394	0.000723659	0.003876542	3534848
3	2/29/08 17:01	0	0	0	3534848
4	2/29/08 17:04	0.060612225	0.027551011	0.016530607	3534848
5	2/29/08 17:07	0	0	0	3534848
6	2/29/08 17:10	0	0	0	3534848
7	2/29/08 17:13	0.060733302	0.027606046	0.016563628	3534848
8	2/29/08 17:16	0	0	0	3534848
9	2/29/08 17:19	0.060727442	0.027603383	0.01656203	3534848
10	2/29/08 17:22	0	0	0	3534848
11	2/29/08 17:25	0	0	0	3534848
12	2/29/08 17:28	0	0	0	3534848
13	2/29/08 17:31	0	0	0	3534848
14	2/29/08 17:34	0.121368621	0.055167555	0.038617289	3534848
15	2/29/08 17:37	0	0	0	3534848
16	2/29/08 17:40	0	0	0	3534848
17	2/29/08 17:43	0	0	0	3534848
18	2/29/08 17:46	0	0	0	3534848
19	2/29/08 17:49	0	0	0	3534848
20	2/29/08 17:52	0	0	0	3534848
21	2/29/08 17:55	0.121392912	0.055178596	0.033107158	3534848
22	2/29/08 17:58	0.060592703	0.027542138	0.02203371	3534848

Deriving Performance Ranges Using Control Charts



Normal Operations



Suspicious Test

- **Derive control charts from the past good tests**
- **Flag new tests as anomalous if there are many violations in the control charts**

[Nguyen et al., ICPE 2012]

Deriving Frequent Itemset from Past Test(s)

Time	DB read/sec	Throughput	Request Queue Size
10:00	Medium	Medium	Low
10:03	Medium	Medium	Low
10:06	Low	Medium	Medium
10:09	Medium	Medium	Low
10:12	Medium	Medium	Low
10:15	Medium	Medium	Low

Deriving Frequent Itemset from Past Test(s)

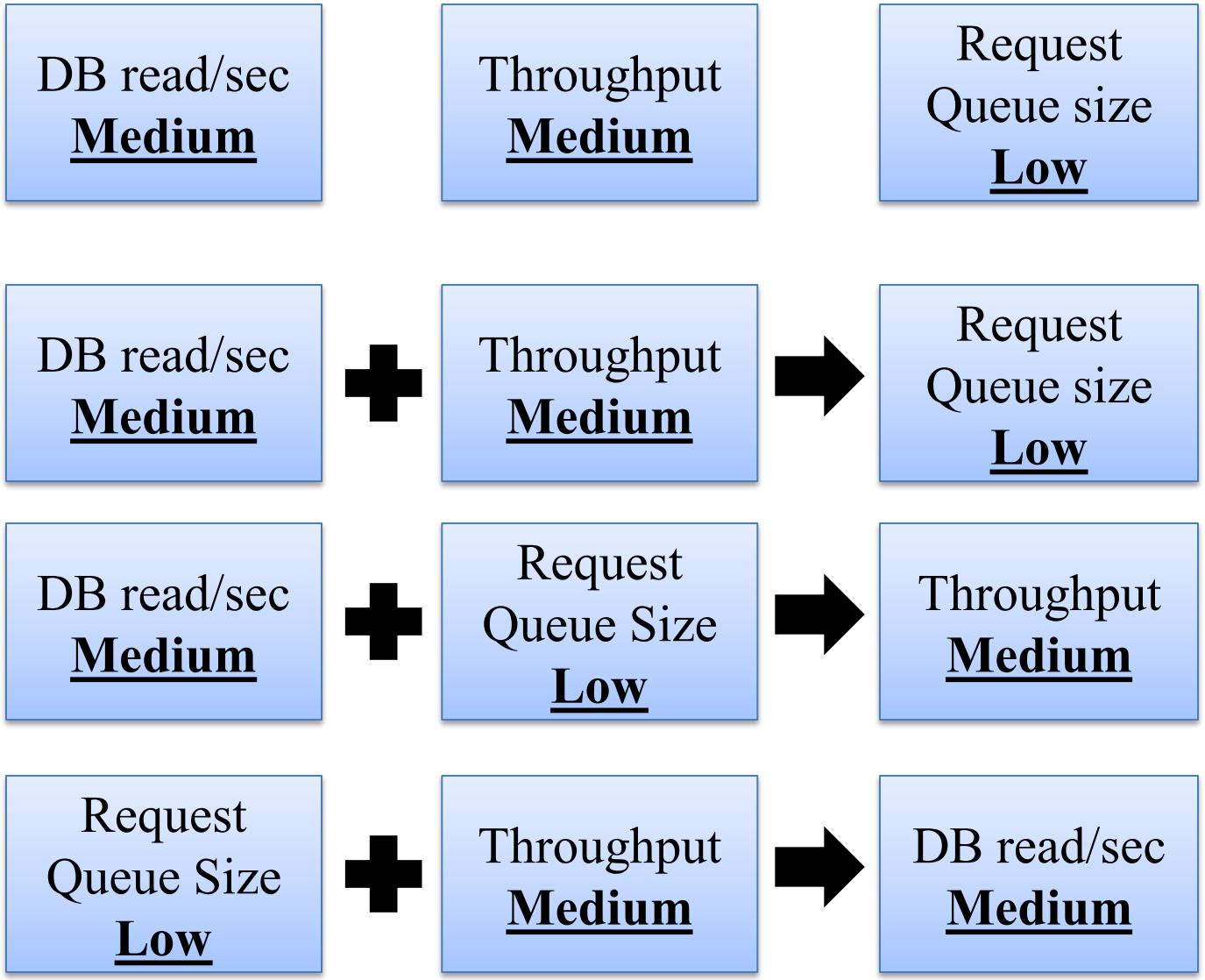
Time	DB read/sec	Throughput	Request Queue Size
10:00	Medium	Medium	Low
10:03	Medium	Medium	Low
10:06	Low	Medium	Medium
10:09	Medium	Medium	Low
10:12	Medium	Medium	Low
10:15	Medium	Medium	Low

DB read/sec **Medium**

Throughput **Medium**

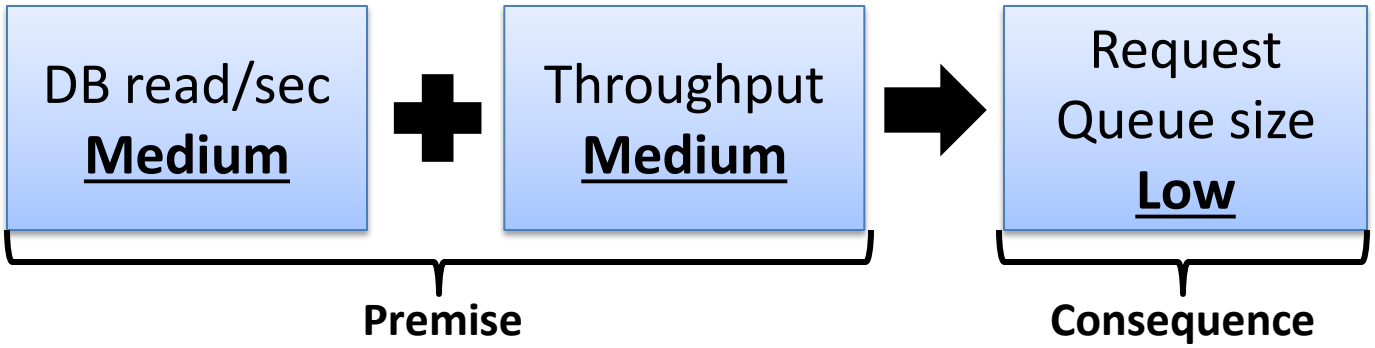
Request Queue size **Low**

Deriving Frequent Itemset from Past Test(s)



Automated Derivation of Performance Rules

- We have derived from the past test(s) a set of performance rules:



- Flags tests where the rule does not hold

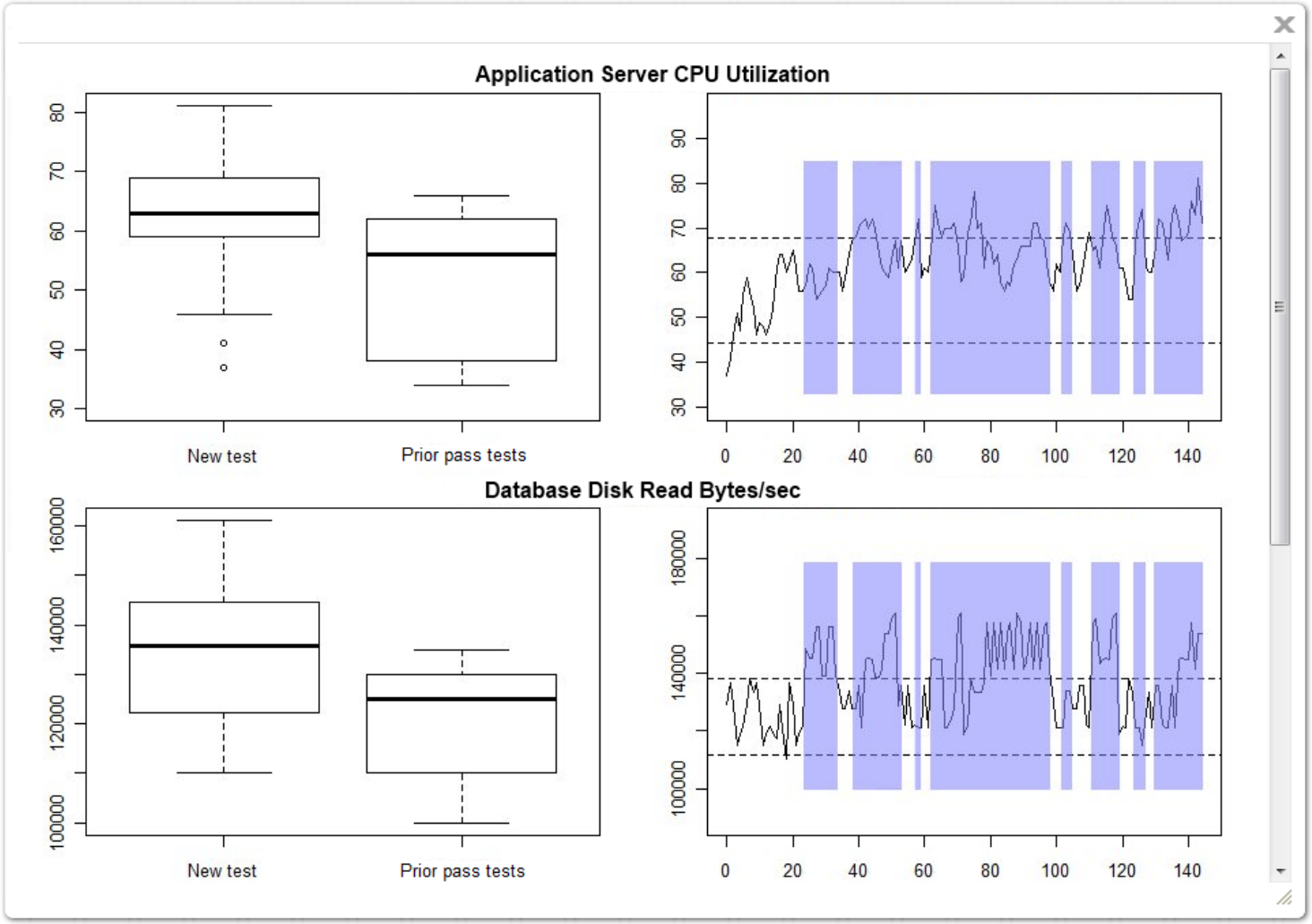
Severity	Performance Regression	Symptoms
0.66	Application Server CPU Utilization	Show Rules
0.60	Application Server Memory Utilization	Show Rules
0.58	Database Disk Read Bytes/sec	Show Rules

Counter Analysis Report

Severity	Performance Regression	Symptoms
0.66	Application Server CPU Utilization	Hide Rules Graph | Conf. Change | Expected Correlation  | 0.79 | Database Logical Disk Reads/sec=Mid Database Memory Page Writes/sec=Mid Database CPU Utilization=Mid Database Memory Page Reads/sec=Mid Application Server CPU Utilization=Mid(High) Application Server Memory Utilization=Mid(High)  | 0.65 | Database Logical Disk Reads/sec=Mid Database Memory Page Reads/sec=Mid Application Server CPU Utilization=Mid(High) Application Server Memory Utilization=Mid(High) Database Disk Read Bytes/sec=Mid(High)
0.60	Application Server Memory Utilization	Show Rules
0.58	Database Disk Read Bytes/sec	Show Rules

Severity	Performance Regression	Symptoms
0.66	Application Server CPU Utilization	Show Rules
0.60	Application Server Memory Utilization	Show Rules
0.58	Database Disk Read Bytes/sec	Show Rules

Counter Analysis Report

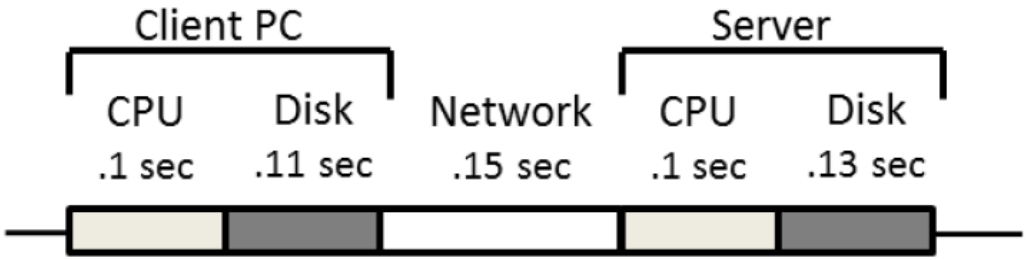


Severity	
0.66	App
0.60	App
0.58	Data

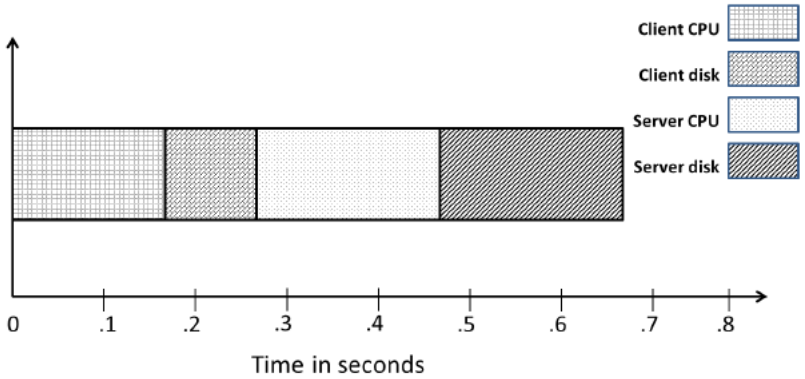
High)
Mid(High)
High)
Mid(High)
)

Deriving Transaction Profiles Using Queuing Theory

- Every transaction visits various resources (e.g., CPU and disk)



- The complete series of service demands for one transaction is known as the Transaction Profile (TP)



TP for transaction type A

TP should be stable across different tests.

Otherwise, there is an anomaly

[Ghaith et al., CSMR 2013, STVR 2015]

Correlating Logs and Counters

#	Log Lines
1	time=1, thread=1, session=1, receiving new user registration request
2	time=1, thread=1, session=1, inserting user information to the database
3	time=1, thread=2, session=2, user=Jack, browse catalog=novels
4	time=1, thread=2, session=2, user=Jack, sending search queries to the database
5	time=3, thread=1, session=1, user=Tom, registration completed, sending confirmation email to the user
6	time=3, thread=2, session=2, database connection error: session timeout
7	time=4, thread=1, session=1, fail to send the confirmation email, number of retry = 1
8	time=6, thread=2, session=2, user=Jack, successfully retrieved data from the database
9	time=7, thread=2, system health check
10	time=8, thread=1, session=1, registration email sent successfully to user=Tom
11	time=9, thread=2, session=3, user=Tom, browse catalog=travel
12	time=10, thread=2, session=3, user=Tom, sending search queries to the database
13	time=10, thread=3, session=4, user=Jim, updating user profile
14	time=11, thread=3, session=4, user=Jim, database error: deadlock

	A	B	C	D	E
1	Time	Disk Reads/sec	Disk Writes/sec	Page Faults/sec	Memory
2	2/29/08 16:58	0.049986394	0.000723659	0.003876542	3534848
3	2/29/08 17:01	0	0	0	3534848
4	2/29/08 17:04	0.060612225	0.027551011	0.016530607	3534848
5	2/29/08 17:07	0	0	0	3534848
6	2/29/08 17:10	0	0	0	3534848
7	2/29/08 17:13	0.060733302	0.027606046	0.016563628	3534848
8	2/29/08 17:16	0	0	0	3534848
9	2/29/08 17:19	0.060727442	0.027603383	0.01656203	3534848
10	2/29/08 17:22	0	0	0	3534848
11	2/29/08 17:25	0	0	0	3534848
12	2/29/08 17:28	0	0	0	3534848
13	2/29/08 17:31	0	0	0	3534848
14	2/29/08 17:34	0.121368621	0.055167555	0.038617289	3534848
15	2/29/08 17:37	0	0	0	3534848
16	2/29/08 17:40	0	0	0	3534848
17	2/29/08 17:43	0	0	0	3534848
18	2/29/08 17:46	0	0	0	3534848
19	2/29/08 17:49	0	0	0	3534848
20	2/29/08 17:52	0	0	0	3534848
21	2/29/08 17:55	0.121392912	0.055178596	0.033107158	3534848
22	2/29/08 17:58	0.060592703	0.027542138	0.02203371	3534848

Combining Logs and Counters

00:01, USER starts a conversation with USER

00:01, USER says MSG to USER

00:02, USER says MSG to USER

00:11, USER says MSG to USER

00:12, USER says MSG to USER

00:18, USER ends a conversation with USER

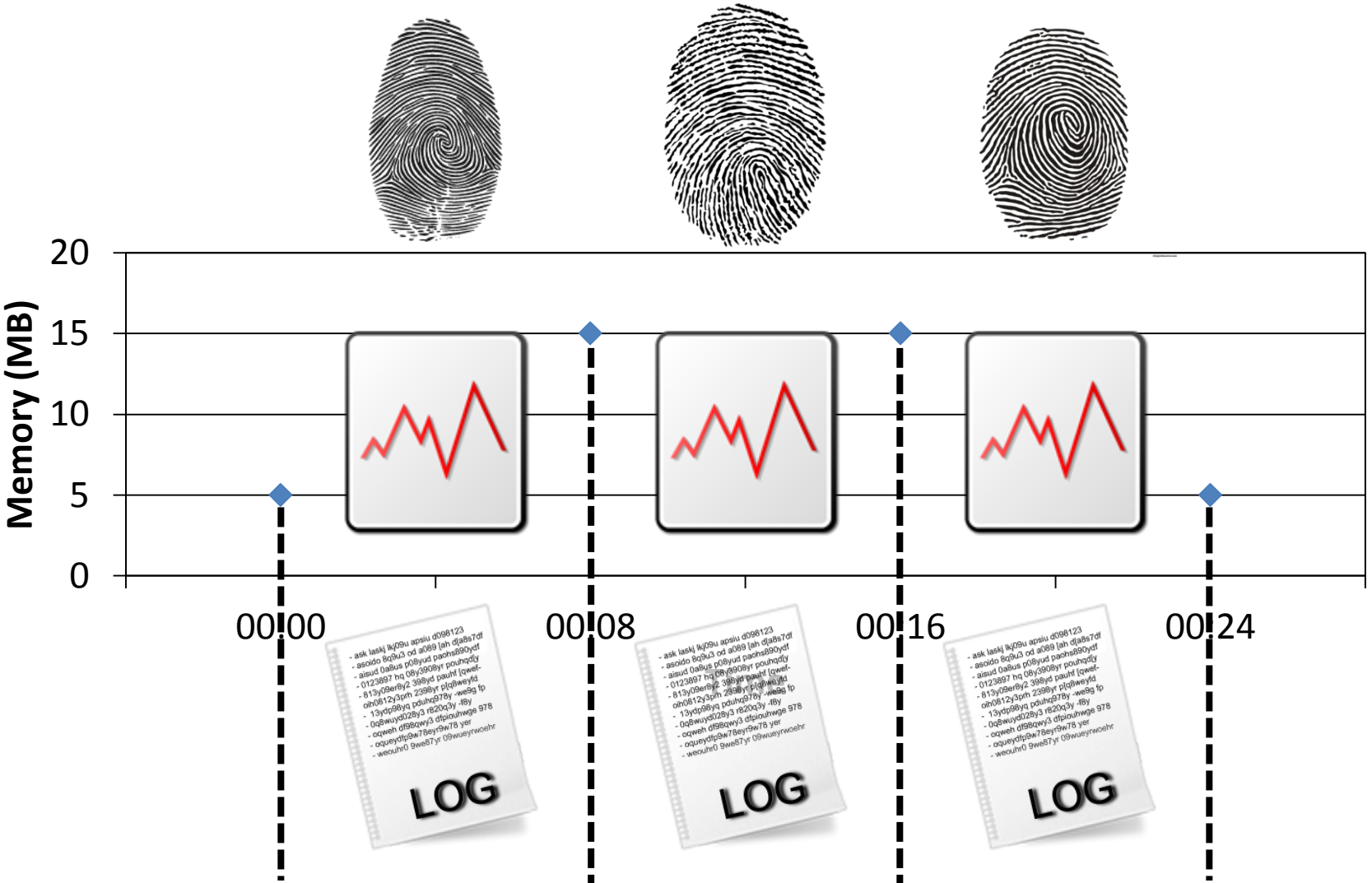
00:00, 5MB

00:08, 15MB

00:16, 15MB

00:24, 5MB

Diagnosing Memory-Related Problems



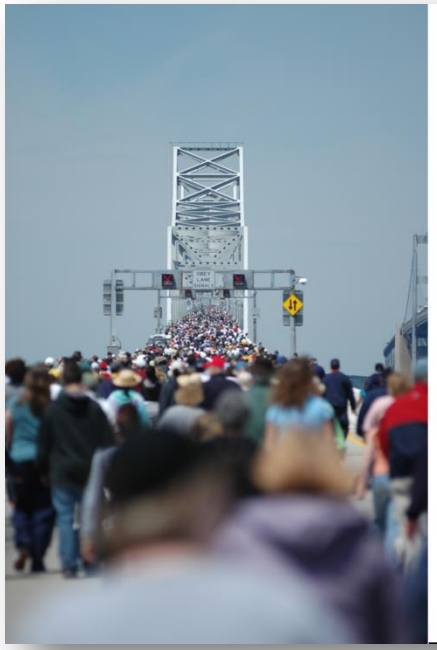
Creating A Research Community

LT: International Workshop on Large-Scale Testing

Large-Scale Testing includes all different objectives and strategies of testing large-scale software systems using *load*. Examples of large-scale testing include live upgrade testing, load testing, high availability testing, operational profile testing, performance testing, reliability testing, stability testing and stress testing.

Workshop: http://lt2016.eecs.yorku.ca/

Contact: zmjiang@cse.yorku.ca



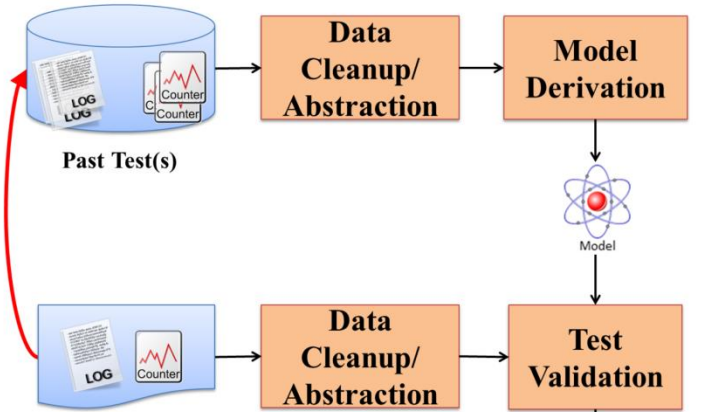
Most field problems for large scale systems are *rarely functional* instead they are *load-related*

at&t mobile me BlackBerry

amazon web services

Office 365 PayPal

Is the system ready for release?



Creating A Research Community

LT: International Workshop on Large-Scale Testing

LT 2016
Large-Scale Testing