Correlating Events with Time Series for Incident Diagnosis

Chen Luo[1] , Jian-Guang Lou [2], Qingwei Lin[2] , Qiang Fu[2] , Rui Ding [2], Dongmei Zhang [2] , Zhe Wang[1]

[1]Jilin University, China, [2]Microsoft Research Asia

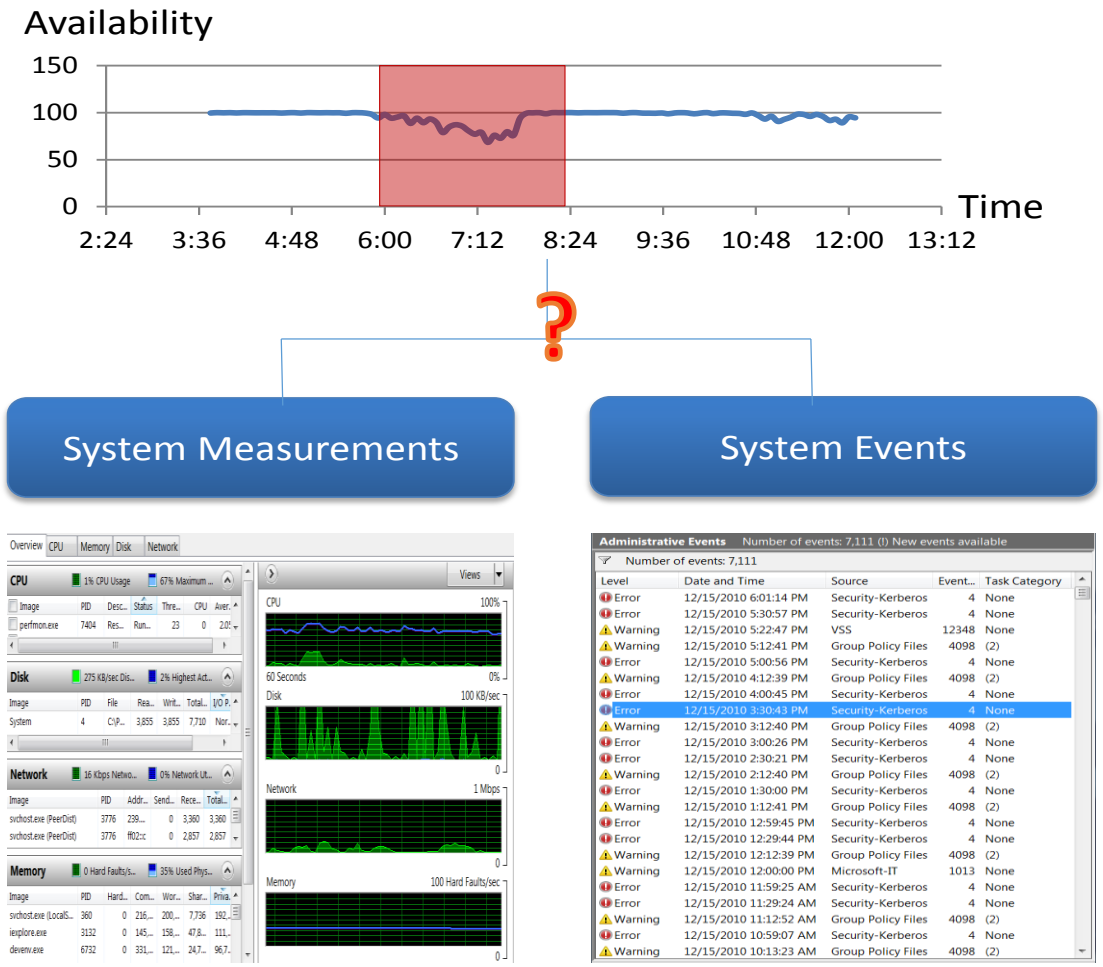
Aug. 2014

Background

- Online services – the wave of IT industry
- Incident diagnosis is critical for services
- Incident diagnosis depends on data analysis

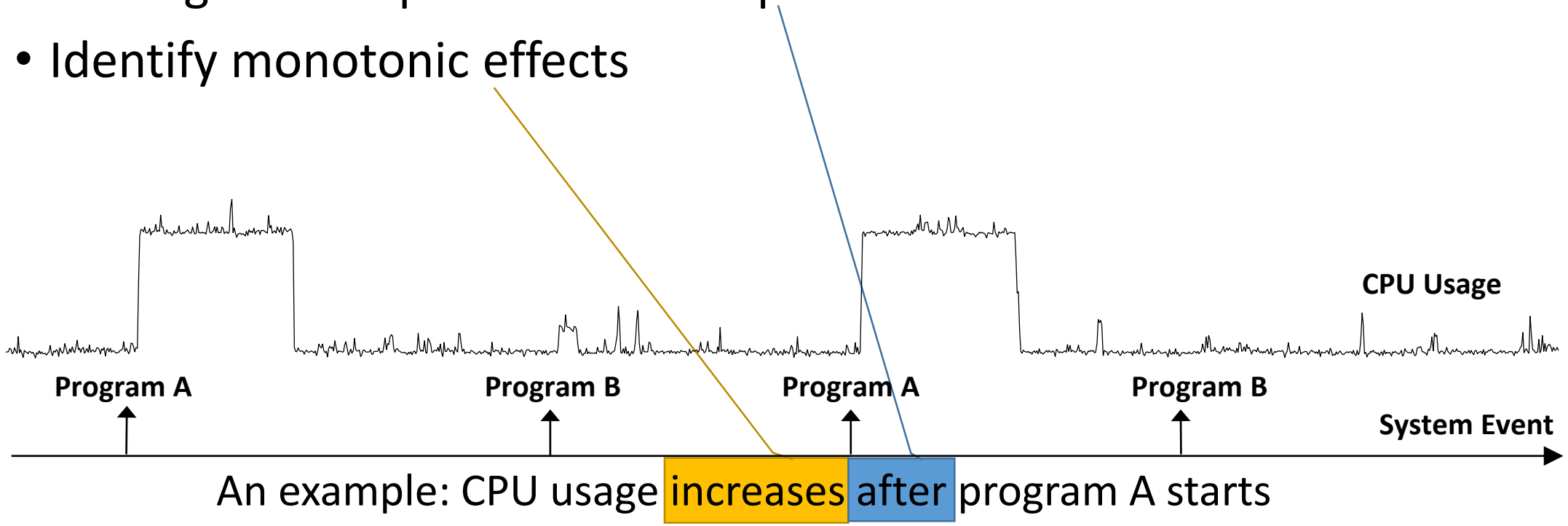
Correlation Analysis for Incident Diagnosis

- Correlation analysis is a major tool for incident diagnosis
- Why correlation?
  - Correlation often provides hints for causation
  - Engineers start their diagnosis through hunting metrics correlated to KPIs (e.g. availability, latency)

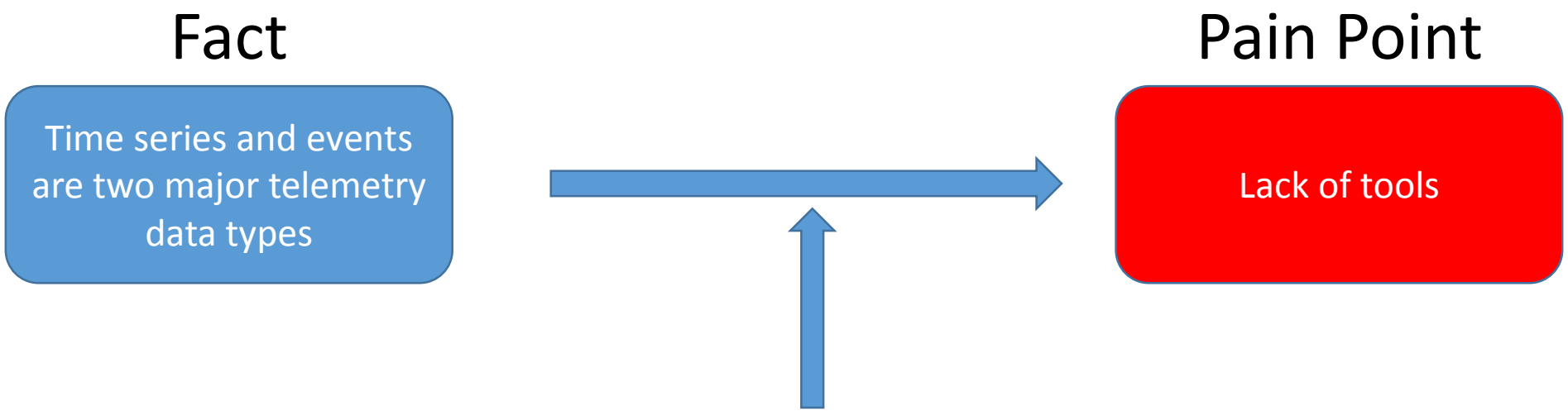


Practical Requirements in Incident Diagnosis

- Handling heterogeneous data, e.g., time series and events
- Detecting the existence of correlation
- Finding out temporal relationships
- Identify monotonic effects



Pain Points



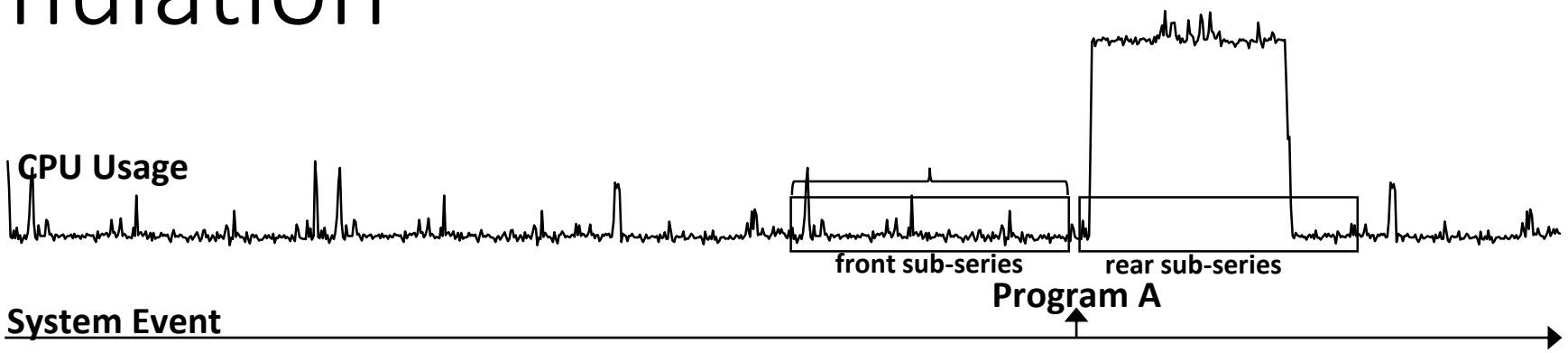
Difficult to apply existing tools (e.g., Pearson, J-Measure)

- Cannot handle heterogeneous data
- Only consider point-to-point corresponding relationship
- Cannot model co-occurrence/value-trend together
- Do not meet all requirements in incident diagnosis

Our Approach

- Basic idea
  - Modeling co-occurrence and value-trend
  - Formulating the analysis as a two-sample problem
- Implementation
  - Resolving the problem with a Nearest Neighbor Method
  - Automatic selecting parameters

Formulation



Intuitions behind	Formulation as a two-sample problem
Correlation	Events have corresponding value changes in time series
Temporal Relationships	Value changes occur before or after events
Monotonic Effects	Value always increases or decreases as events happen

Corresponding sample (front or rear sub-series) is different from the base sample (normal sub-series)

Corresponding sample has significant value difference from the base sample

The Overall Algorithm

Algorithm 1: The Overall Algorithm

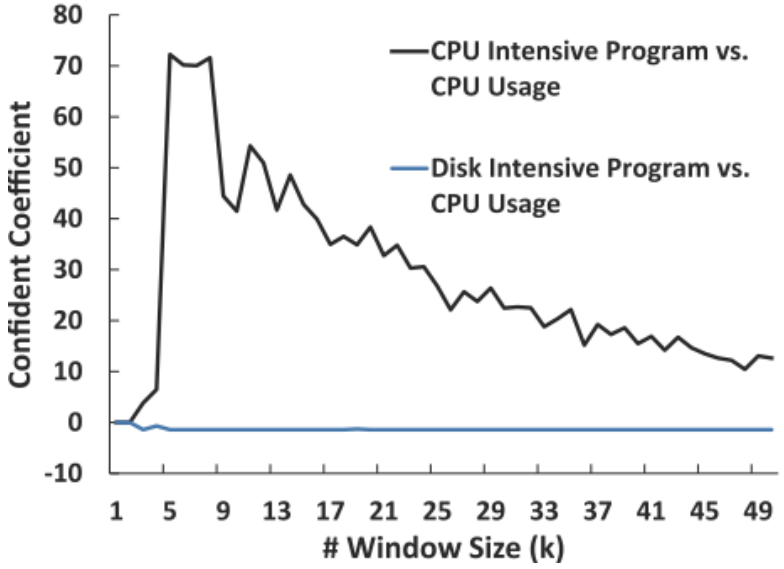
Input: Event $E = (e_1, e_2, ..., e_n)$, and Time Series $S = (s_1, s_2, ..., s_m)$, and the sub-series length k.
Output: The correlation flag C, the direction D, and the effect type T

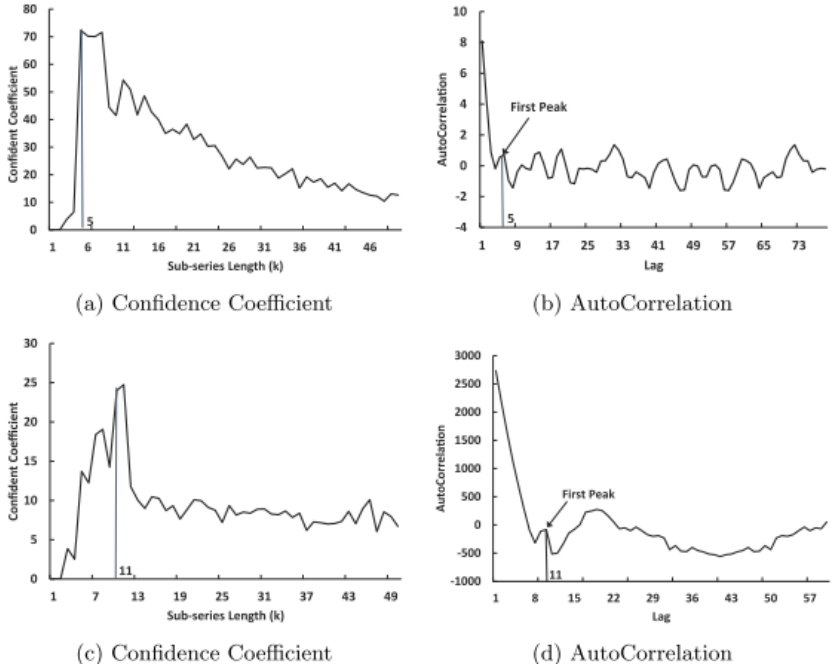
Initialize Γ^{front} and Γ^{rear};
Initialize Θ;
Initialize $R = false$, $D = NULL$, $T = NULL$;
Normalize each $\ell_k^{front}(S, e_i)$ and $\ell_k^{rear}(S, e_i)$.;
Test Γ^{front} and Θ using Nearest Neighbors Method. The result is denoted as D_f.;
Test Γ^{rear} and Θ using Nearest Neighbors Method. The result is denoted as D_r.;
if $(D_r == true \&\& D_f == false)$ **then**
    $R = true$.;
    Calculate t_{score} using Equation (8).;
    if $(t_{score} > \alpha)$ **then**
        $T = E \bar{\rightarrow} S$.;
    else if $(t_{score} < -\alpha)$ **then**
        $T = E \stackrel{+}{\rightarrow} S$.;
else if $(D_r == false \&\& D_f == true)$ || $(D_r == true \&\& D_f == true)$ **then**
    $R = true$.;
    Calculate t_{score} using Equation (8).;
    if $(t_{score} > \alpha)$ **then**
        $T = S \bar{\rightarrow} E$.;
    else if $(t_{score} < -\alpha)$ **then**
        $T = S \stackrel{+}{\rightarrow} E$.;
Out put R, D and T;
Algorithm End.

- Nearest Neighbors based algorithm
  - The proportion of pairs from the same sample among all pairs in a neighborhood follows a normal distribution if two samples have a similar distribution.
  - Otherwise, it should be not
- Test results answer 3 requirements based on rules

Automatic Parameter Tuning

- Number of neighbors is set following suggestions of [28]
- Window size selection
  - Confident coefficient increases at first, and then decreases
  - The first peak of ACF is a good choice





Evaluation on Controlled Environment

- Data source: data observed in a controlled environment
  - Events – starting events of 3 programs
  - Time series – usage data of CPU/memory/disk

Name	Type	Description
CPU Intensive program	Event	A multi-thread process, which will let the CPU Usage achieve nearly 90%
Memory Intensive program	Event	A process that will apply for a nearly 2G memory space
Disk Intensive program	Event	A copy files process which can sharply increase disk transfer rate
Query Alert	Event	When SQL query delay exceed the maximum limit, an alert occurrence.
CPU Usage	Time Series	record the CPU usage every second
Memory Usage	Time Series	record the Memory usage every second
Disk Transfer Rate	Time Series	record the Disk Transfer Rate every second

- Result

Name	CPU	Memory	Disk
CPU Intensive Program	$\overset{+}{\rightarrow}$	-	-
Memory Intensive Program	$\overset{+}{\rightarrow}$	$\overset{+}{\rightarrow}$	-
Disk Intensive Program	-	-	$\overset{+}{\rightarrow}$
Query Alert	$\overset{+}{\rightarrow}$	$\overset{+}{\rightarrow}$	-

Evaluation on Real Data (1)

- Baseline Algorithms
  - 1. Pearson Correlation (considering event sequence as a 0/1 time series)
  - 2. J-Measure (transforming time series to event sequence)
- Evaluation Method $F - Measure$

$$F_1 = \frac{2 * True\ Positive}{2 * True\ Positive + False\ Negative + False\ Positive}$$

- Dataset
  - 1. System Monitoring Dataset (Timer Job and Performance Counter)
  - 2. Custom Support Dataset (HTTP Status Code and Custom Call)

Evaluation on Real Data (2)

Data Set	Methods	Existence	Temporal Order	Effect Type
		F_1 Score	F_1 Score	F_1 Score
System Monitoring Data	Correlation Mining (L1)	0.7916	0.8020	0.8016
	Correlation Mining (L2)	**0.8205**	0.7612	**0.8780**
	Correlation Mining (DTW)	0.7962	**0.8021**	0.8210
	Pearson Correlation	0.6974	N/A [2]	0.6732
	J-Measure	0.6148	N/A	N/A
Custom Support Data	Correlation Mining (L1)	0.7915	0.7659	0.7204
	Correlation Mining (L2)	0.8423	0.7870	0.8334
	Correlation Mining (DTW)	**0.8631**	**0.8205**	**0.8532**
	Pearson Correlation	0.6030	N/A	0.6501
	J-Measure	0.7398	N/A	N/A

Our approach performs much better than the baseline algorithms.

Summary

- Motivated by requirements in incident diagnosis, we investigated the problem of correlation mining between time series data and event data.

- We formulate the correlation problem as a two-sample problem, and propose a novel framework to resolve the problem.

- The experiment on simulated data and real data from a Microsoft service showed the effectiveness of our method.

Note: it has been implemented as a building block of a diagnosis toolset.

Questions