

Infering information in case of lost events in a trace

Marie Martin, Michel Dagenais

École Polytechnique de Montréal DORSAL

Agenda

Introduction

2 Framework

Event inference Content inference

3 Results

Methodology View Benchmark

4 Conclusion



Introduction



Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais

Context

Lost events during tracing \rightarrow Discard & overwrite modes



A trace with missing events

Parallel analysis of traces

 \rightarrow "missing" past events

Find incoherent events ;

- 2 Show inconsistency and uncertainty ;
- **3** Infer information about missing events.



- Find incoherent events ;
- Ø Show inconsistency and uncertainty ;
- **3** Infer information about missing events.

- Find incoherent events ;
- Ø Show inconsistency and uncertainty ;
- **8** Infer information about missing events.



Using Finite-State Machines (FSM): consistency & state certainty check

 \rightarrow when reading the trace, update the state of the FSM with each event



Detection of an inconsistent state

- 1 Find incoherent events ;
- 2 Show inconsistency and uncertainty ;
- **③** Infer information about missing events.

Framework



Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais

Example – intuitive reasoning



A trace with an inconsistent state - as displayed in the Control Flow View

		17:24:26.536	17:24:26.538	17:	24:26.540	17	7:24:26.542	17:	24:26.544	17:24:26.5	46	17:
Sansfil-Securise-Etudiants-Lassor												
lttng-consumerd swapper/0												F
gnome-terminal- swapper/1 rcu_sched swapper/3				j						Incoherer	ł	
	_			1							П	1

A trace with an inconsistent state – as displayed in the Coherence View

Event inference

Objective

Find states between the first certain state after the incoherence and the last known consistent state

Basis

 $\mathsf{FSM} \leftrightarrow \mathsf{Graph}$

 $\begin{array}{l} \mathsf{State} \leftrightarrow \mathsf{Node} \\ \mathsf{Transition} \leftrightarrow \mathsf{Arc} \end{array}$

 \Rightarrow Find a path between a starting state and a target state

Event inference

Algorithm

Dijkstra's shortest path: computes the shortest path in a directed graph

Iterates over states, updating their distance from the start using weights on arcs (see following example)

Weights: statistics on transitions

 \rightarrow for a given scenario, the most likely transition is the one that has been taken most frequently



A trace with an inconsistent state - as displayed in the Control Flow View



The simplified FSM modeling a process





Set of possible transitions, following from the consistency check \rightarrow Select the "best" (most likely) transition to get the starting point of the algorithm



	usermode	syscall	wait for cpu	wait blocked
dist	∞	∞	∞	0
prev	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED



	usermode	syscall	wait for cpu	wait blocked
dist	∞	∞	∞	0
prev	UNDEFINED	UNDEFINED	UNDEFINED	UNDEFINED



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	∞	∞	0
prev	wait blocked	UNDEFINED	UNDEFINED	UNDEFINED



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	1	∞	0
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	1	∞	0
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

unvisited = { usermode, syscall, wait for cpu }



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	1	∞	0
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

unvisited = { usermode, syscall, wait for cpu } current = min_dist(unvisited) = usermode



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	1	1	0
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

unvisited = { usermode, syscall, wait for cpu }



	usermode	syscall	wait for cpu	wait blocked
dist	0.0135	1	1	0
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

$$\begin{array}{l} {\sf unvisited} = \{ \mbox{ usermode, syscall, wait for cpu } \} \\ {\sf current} = {\sf target} \Rightarrow {\sf STOP} \end{array}$$



	usermode	syscall	wait for cpu	wait blocked
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais



	usermode	syscall	wait for cpu	wait blocked
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

path = (usermode \rightarrow wait blocked \rightarrow usermode)



	usermode	syscall	wait for cpu	wait blocked
prev	wait blocked	wait blocked	UNDEFINED	UNDEFINED

path = (usermode \rightarrow wait blocked \rightarrow usermode)

inferred events = (sched_switch)

Progress Report Meeting / May 2018 – Marie Martin, Michel Dagenais

Content inference

Basis

If a transition occurs, then its conditions are verified.

We can make assumptions based on the conditions, in order to infer information about the content of the event.

```
<test id="prev_state_0">
	<if>
	<condition>
		<field name="prev_state" />
		<stateValue type="long" value="0" />
		</condition>
		</if>
</test>
```

XML condition

Progress Report Meeting / May 2018 – Marie Martin, Michel Dagenais

Multi-valued inferred event

Sometimes, several values are possible for the same event field.



XML condition

We can try to discriminate between values using information from the state system.

Otherwise, we leave it to the user to select the most appropriate value (see following views).

Results



Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais

Methodology

- Definition of the FSM in XML (from the state provider definition)
- 2 Deletion of chosen events from a 'real-world' trace
- 8 Execution of the analysis in Trace Compass
- Selection of the Global Coherence View to show inferred events

Inference view

	17:24:26.536	17:24:26.538	17:24:26.540	17:24:26.542	17:24:26.544	17:24:26.546	17
V Sansfil-Securise-Etudiants-Lassor							
ktng-consumerd							
gnome-terminal-						-	
swapper/1			-	_			
swapper/3							

Control Flow View of the original trace



Inference selection

🕼 Histogram 🔳 F	ropertie	s 🛄 e	ookmarks 🗖 Coher	ence View 🛅 G	lobal Inference Vi	ew 23	Ø.	5 🕸 🗉 🖞	53	R & & -	😚 🤑 🔍 🔍 🕅	🝷 🗂 اې ما ته م	~
			17:24:26.510	17:24:26.520	17:24:26.530	17:24:26.540	17:24:26.550	17:24:26	560	17:24:26.570	17:24:26.580	17:24:26.590	
▼ trace100-109.2		- E							6				*
lttng-consumerd	16185	17:2		_							-	_	
swapper/0	0	17:2		Inferer	nce resolution						11		A CONTRACTOR OF
gnome-terminal-	5078	17:2		-							-		
swapper/1	0	17:2	تبصحين فعازه									یں۔ جی بر این	
rcu_sched	8	17:2		Inferred	events					-		(· · · ·)	÷ · · · ·
swapper/3	0	17:2									-	1.00	
rcuos/2	29	17:2		17:2	4:26.543 921 899 s	iched_switch	next_tid						
rcuos/3	37	17:2											
kworker/3:2	8254	17:2											
Xorg	4285	17:2									(1.01.01
cinnamon	4971	17:2											and the second se
kworker/u16:3	13068	17:2											
i915/signal:0	396	17:2											
rngd	999	17:2											
kworker/0:4	20850	17:2-							OK				
kworker/3:1H	468	17:2											
java	551	17:2-								_			
Timer	8842	17:2		1					_				
firefox	7739	17:2											
netdata	1741	17:2-											
java	32741	17:2											
kworker/u17:0	869	17:2-											
InputThread	4381	17:2											
kworker/1:2	5897	17:2											
itting-sessiond	18609	17:2											
migration/2	20	17:2											
× KWORKET/2:2	nferred e	17:21-				Infer	red		* * * *				
l '													•

Inference selection

		runtime-EclipseApplication	Te	ace100-109Eclipse Platform
Edit Navigate Pro	ect Ru	n File Window Help Windo	V Inference resolution	
• 🗟 💁 •	e •	- \$ + \$ \$ + \$ \$ + \$	4	Quick Access
👍 Histogram 🔳 R	ropertie	is 💷 Bookmarks 🗖 Coheren	e V Inferred events	
		17:24:26.510 17	245 17:24:26 542 921 999 rehad owitch	570 17:24:26.580 17:24:26.590
▼ trace100-109.2			17.24.20.343 921 099 Stile0_SWICH • NOXE_10	
lttng-consumerd	16185	17:2	869	
swapper/0	0	17:2	4701	
gnome-terminal-	5078	17:2	10	
swapper/1	0	17:2	16502	a second a s
rcu_sched	8	17:2	687	
swapper/3	0	17:2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	4726	in the second
rcuos/2	29	17:2	-	
rcuos/3	37	17:2		
kworker/3:2	8254	17:2	16494	
Xorg	4285	17:2	16486	
cinnamon	4971	17:2	5032	
kworker/u16:3	13068	17:2	4733	
i915/signal:0	396	17:2	7739	
rngd	999	17:2	4700	
kworker/0:4	20850	17:2	-	
kworker/3:1H	468	17:2	10400	
java	551	17:2	16501	
Timer	8842	17:2	4725	
firefox	7739	17:2	5014	
netdata	1741	17:2	22282	
java	32741	17:2	4783	
kworker/u17:0	869	17:2	16402	
InputThread	4381	17:2	16455	
kworker/1:2	5897	17:2	4/50	
lttng-sessiond	18609	17:2	4775	
migration/2	25	17:2	16485	
kworker/2:2	22282	17:2		
×	Inferred	event		
				ок
017-08-30 17:24:26.507	041343			

Inferred event details

👔 Histogram 🗇 Properties 💷 Bookmarks 🗇 Coherence View 🗇 Global Inference View 😆 🧉 🖉 🔛 🖉 😴 🖄 🤴 🤟 🗞 💘 🗮 🕀 😪 💥 🔶 🖋 🖉 🖉 🖉												
			17:24:26.510	17:24:26.520	17:24:26.530	17:24:26.540	17:24:26.550	17:24:26.560	17:24:26.570	17:24:26.580	17:24:26.590	
▼ trace100-109.2												*
lttng-consumerd	16185	17:2			-		-	_				
swapper/0	0	17:2		-				-	10.000			100 C 100 C
gnome-terminal-	5078	17:2		1							100 C	
swapper/1	0	17:2	أبصحواضي				1	a second second	_		یند-خده-قب	
rcu_sched	8	17:2	1 /								r · · ·	· · · ·
swapper/3	0	17:2-0	ماردند بالأرجاب	1.0	1.00		1	-		-	1.00	
rcuos/2	29	17:2										
rcuos/3	37	17:2	1 1/	-	-							
kworker/3:2	8254	17:2	1 1/									
Xorg	4285	17:2			[17:24:26.543	921 899] sched_sv	vitch { prev_prio=	0,				10000
cinnamon	4971	17:2			next_tid=869,	prev_tid=5078, p	rev_state=1, next	_prio=0,				let a
kworker/u16:3	13068	17:2			prev_comm=	gnome-terminal-,	next_comm=kwc	orker/u17:0 }	-			
i915/signal:0	396	17:2		-								_
rngd	999	17:2							-			_
kworker/0:4	20850	17:2				Valie	ier					_
kworker/3:1H	468	17:2	-						-			_
java	551	17:2										_
Timer	8842	17:2		1								_
firefox	7739	17:2		1					-			
netdata	1741	17:2										
java	32741	17:2										_
kworker/u17:0	869	17:2										
InputThread	4381	17:2										
Kworker/1:2	5897	1/2										_
Ittng-sessiond	18609	17:2						1				
migration/2	25	1/2						1 14				
worker/2:2	22282	17:2				Inferred						
	eu e	10				internet						,

Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais

Benchmark

Average on 15 runs of the analysis module

 \rightarrow Force check for every event

	mytrace1	mytrace2	many-threads	trace2	mytrace3
size	164K	26M	8.1M	14M	86M
nb. events	2188	40902	240644	595641	2689393
no check (s)	1.24	3.31	9.56	15.43	41.7
check + infer (s)	1.26	3.53	12.18	19.4	93.6
nb. inferred events	1	3	29433	769	1501
overhead (%)	1.6	6.6	27.4	25.7	124.5

Analysis execution with event inference

(CPU AMD A10-8700P, 4 cores, 16G RAM)

Conclusion



Limits

Current cost of XML analysis is high for big traces

XML FSM is user-defined

Content inference lacks semantics

The most likely path may not be the path that has really been taken, especially as losing events may occur in an unusual case.

Future work

Continuous work on improving the algorithms and the view \rightarrow scalability

Machine-learning for more accurate probabilities

Automatic computation of the XML FSM, given some traces of the system

Conclusion

Improve Trace Compass by helping the user be aware of trace (un)consistency

Proof-of-concept that we can retrieve lost/inaccessible information in a trace, using state machines

A step towards the parallel analysis of traces

Any questions?

marie.martin@polymtl.ca github : MMartin5/events-analysis

Progress Report Meeting / May 2018 - Marie Martin, Michel Dagenais

Architecture

Inference algorithm

 \rightarrow On-demand, after the analysis \rightarrow Incoherences are collected from the previous analysis \rightarrow Creates inferred event(s) for

each incoherence

Inference trace

 \rightarrow "Artificial" trace with inferred events \rightarrow Only used by the Global Coherence View

