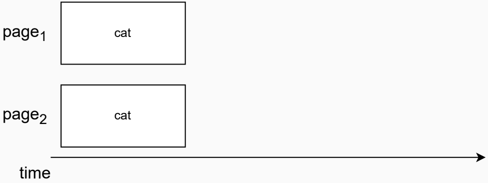
As if Time Had Stopped – Checking Memory Dumps for Quasi-Instantaneous Consistency

Jenny Ottmann, Üsame Cengiz, Frank Breitinger, Felix Freiling

Chair of IT Security Infrastructures
Friedrich-Alexander-Universität Erlangen-Nürnberg

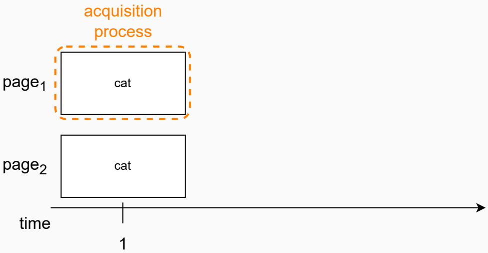
Inconsistencies in Memory Dumps



[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

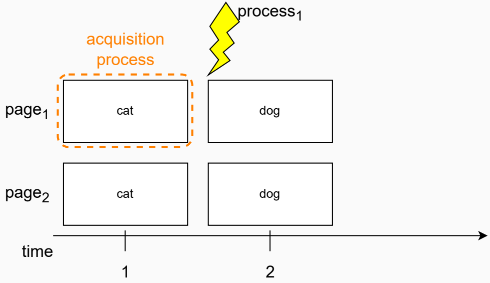
Inconsistencies in Memory Dumps



[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

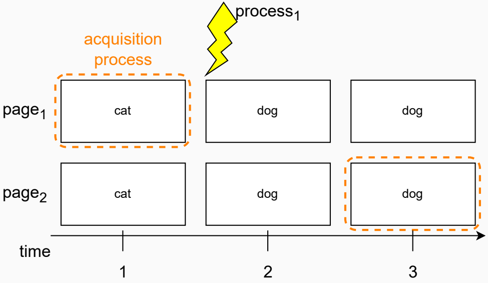
Inconsistencies in Memory Dumps



[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

Inconsistencies in Memory Dumps



[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

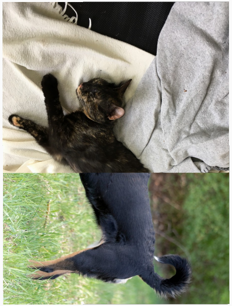
[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

Inconsistencies in Memory Dumps

[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

Inconsistencies in Memory Dumps



- Inhibit the analysis[1]

[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

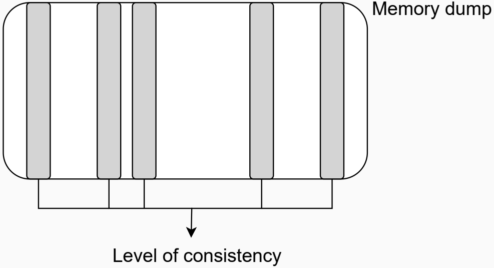
Inconsistencies in Memory Dumps

- Inhibit the analysis[1]
- Not that easy to measure[2]

[1] Case and Richard III 2017; Pagani, Fedorov, and Balzarotti 2019

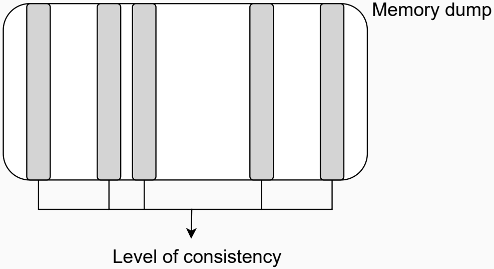
[2] Vömel and Stüttgen 2013; Gruhn and Freiling 2016

Consistency Indicators



[3] Pagani, Fedorov, and Balzarotti 2019

Consistency Indicators

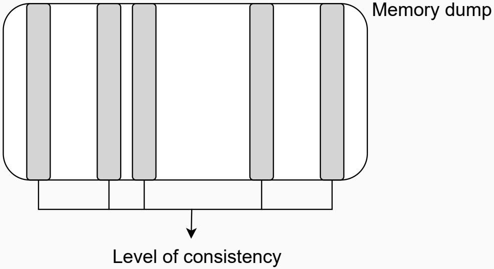


- Already existent

  E.g., VMA count [3], process list

[3] Pagani, Fedorov, and Balzarotti 2019

Consistency Indicators



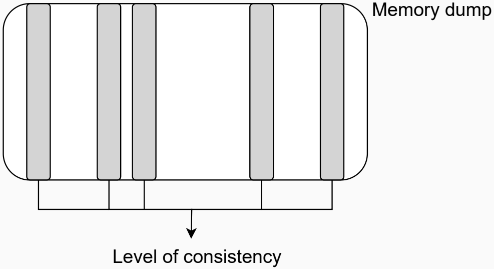
- Already existent

  E.g., VMA count [3], process list

- Deliberately placed

[3] Pagani, Fedorov, and Balzarotti 2019

Consistency Indicators



- Already existent

  E.g., VMA count [3], process list

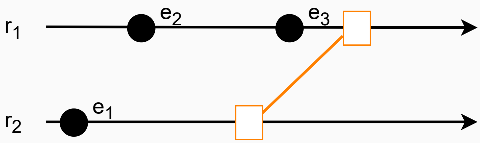
- Deliberately placed

  →Observe quasi-instantaneous consistency

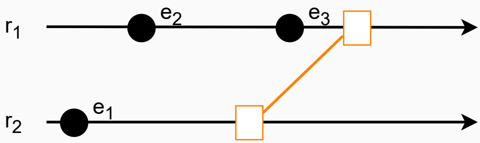
[3] Pagani, Fedorov, and Balzarotti 2019

Model

Space-Time Diagrams

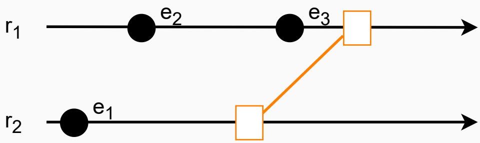


Space-Time Diagrams



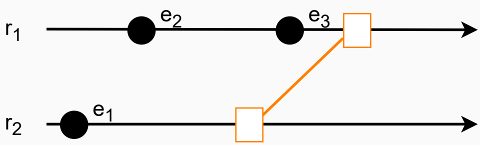
- Set of n memory regions: $R = \{r_1, \ldots, r_n\}$

Space-Time Diagrams



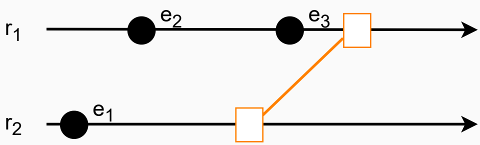
- Set of n memory regions: $R = \{r_1, \ldots, r_n\}$
- Memory: $m : R \times T \rightarrow V$

Space-Time Diagrams



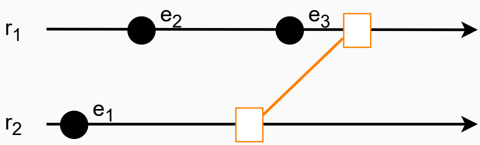
- Set of n memory regions: $R = \{r_1, \ldots, r_n\}$
- Memory: $m : R \times T \to V$
- Only events change contents,

Space-Time Diagrams



- Set of n memory regions: $R = \{r_1, \ldots, r_n\}$
- Memory: $m : R \times T \rightarrow V$
- Only events change contents, therefore, T is defined as the set of natural numbers $\mathbb{N}$

Space-Time Diagrams



- Set of n memory regions: $R = \{r_1, \ldots, r_n\}$
- Memory: $m : R \times T \to V$
- Only events change contents, therefore, T is defined as the set of natural numbers $\mathbb{N}$
- Snapshot: $s : R \to V \times T$

Quasi-Instantaneous Consistency

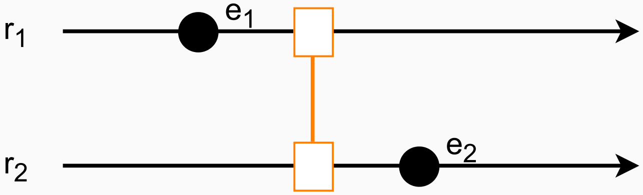
Ideal Case of Memory Acquisition

Instantaneous Consistency[4]

[4] Ottmann, Breitinger, and Freiling 2022.

Ideal Case of Memory Acquisition

Instantaneous Consistency[4]



[4] Ottmann, Breitinger, and Freiling 2022.

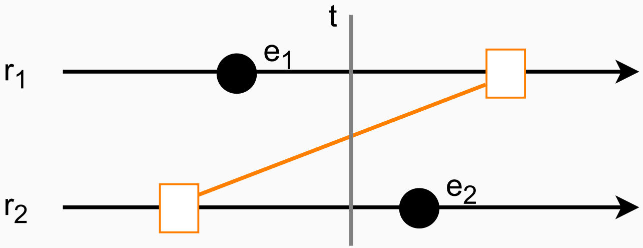
Approximation

Quasi-Instantaneous Consistency[5]

[5] Ottmann, Breitinger, and Freiling 2022.

Approximation

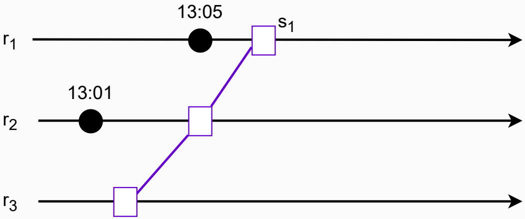
Quasi-Instantaneous Consistency[5]



[5] Ottmann, Breitinger, and Freiling 2022.

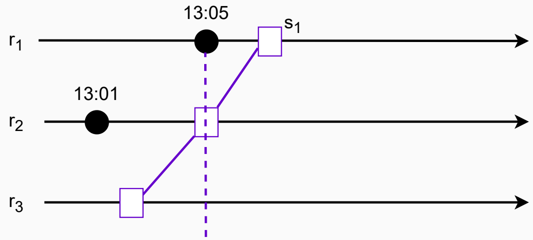
Observing Quasi-Instantaneous Consistency

Idea



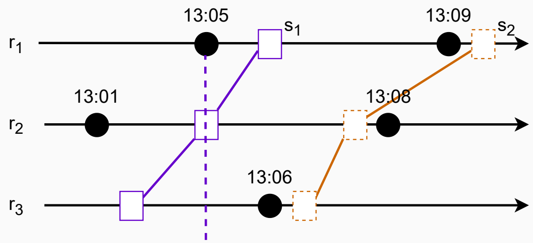
Known states	-	13:05
	13:01	13:01
	-	-
t:	1	2

Idea



Known states	-	13:05
	13:01	13:01
	-	-
t:	1	2

Idea



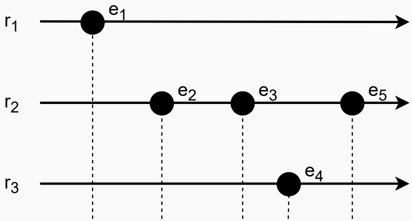
Known states	-	13:05	13:05	13:05	13:09
	13:01	13:01	13:01	13:08	13:08
	-	-	13:06	13:06	13:06
t:	1	2	3	4	5

Observation Elements

Local counters & global counter array

Observation Elements

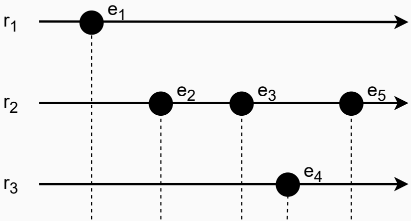
Local counters & global counter array



G	0	**1**	1	1	1	1
	0	0	**2**	**3**	3	**5**
	0	0	0	0	**4**	4
t:	0	1	2	3	4	5

Observation Elements

Local counters & global counter array



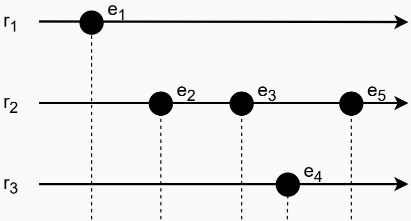
G					
0	**1**	1	1	1	1
0	0	**2**	**3**	3	**5**
0	0	0	0	**4**	4
t: 0	1	2	3	4	5

Prerequisites

- Detection of events

Observation Elements

Local counters & global counter array



G	0	**1**	1	1	1	1
	0	0	**2**	**3**	3	**5**
	0	0	0	0	**4**	4
t:	0	1	2	3	4	5

Prerequisites

- Detection of events
- Ability to save counters locally
- Ability to save counters in global counter array

Formal Proof

Aim

We want to show that:

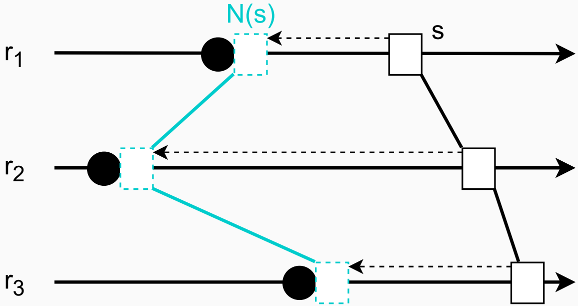
- Local counters & global counter array suffice to check quasi-instantaneous consistency

Normalized Snapshot $N(s)$

Only events change contents

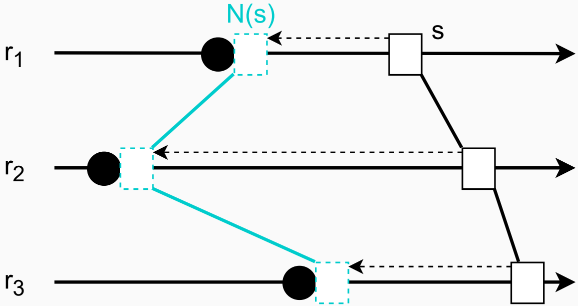
Normalized Snapshot $N(s)$

Only events change contents



Normalized Snapshot $N(s)$

Only events change contents

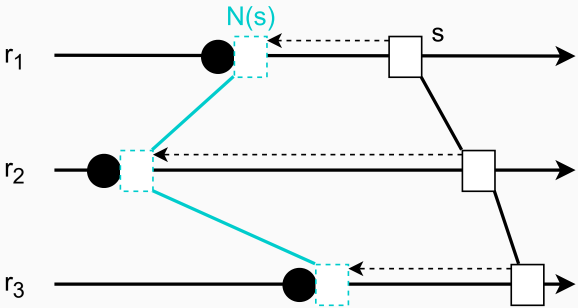


Times

- s: $r_1.t = r_2.t = r_3.t = 3$

Normalized Snapshot $N(s)$

Only events change contents



Times

- s: $r_1.t = r_2.t = r_3.t = 3$
- $N(s)$:

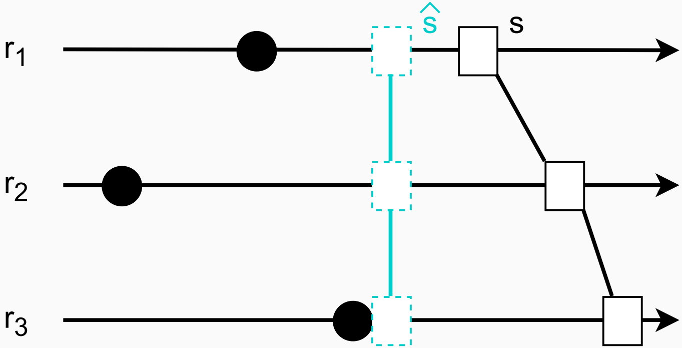
$$r_1.t = 2$$
$$r_2.t = 1$$
$$r_3.t = 3$$

Associated Instantaneous Snapshot $\hat{s}$

A snapshot is quasi-instantaneous if a hypothetical instantaneous snapshot with the same values exists

Associated Instantaneous Snapshot $\hat{s}$

A snapshot is quasi-instantaneous if a hypothetical instantaneous snapshot with the same values exists

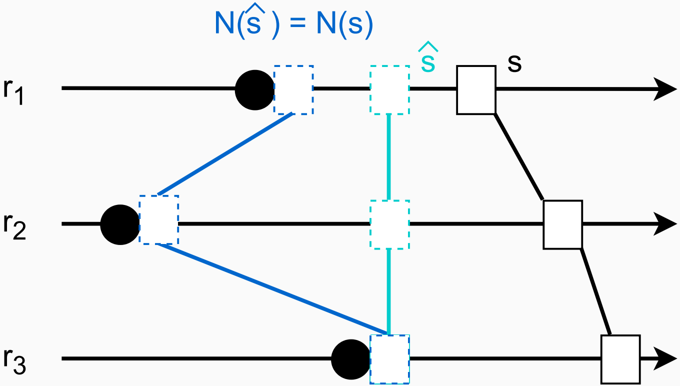


Determining Quasi-Instantaneous Consistency

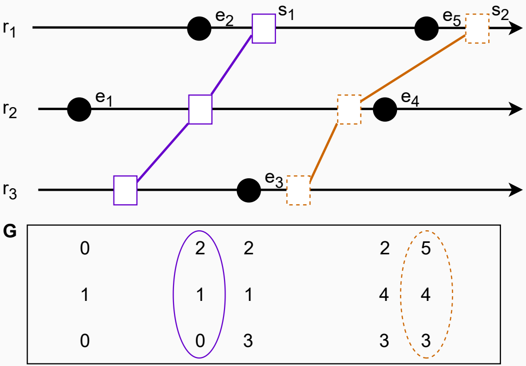
We can use $\hat{s}$ to determine if the snapshot is quasi-instantaneous

Determining Quasi-Instantaneous Consistency

We can use $\hat{s}$ to determine if the snapshot is quasi-instantaneous

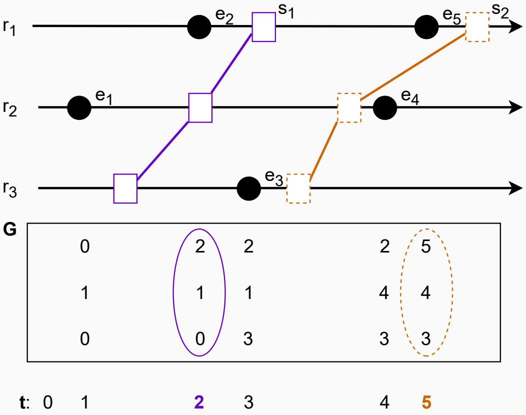


Example



t:	0	1	2	3	4	5
G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3

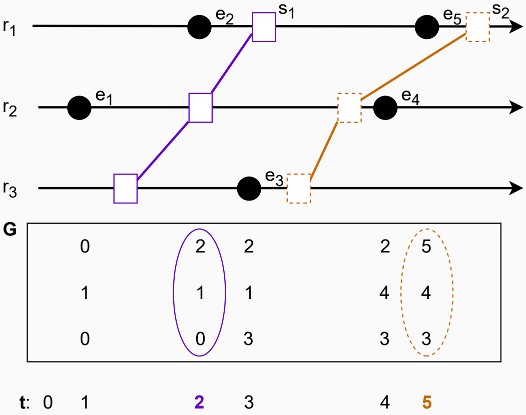
Example



G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3
t:	0	1	**2**	3	4	**5**

S_1

Example

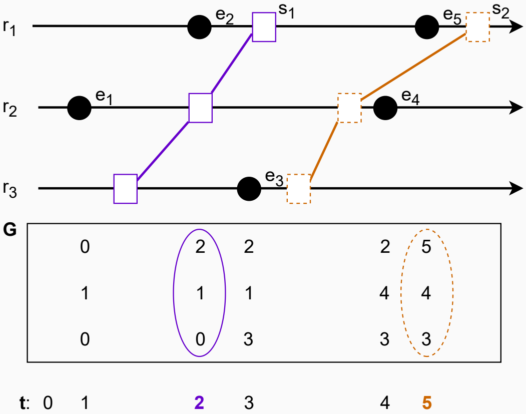


t:	0	1	2	3	4	5
G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3

s_1

- $T_{N(s_1)} = (2, 1, 0)$

Example

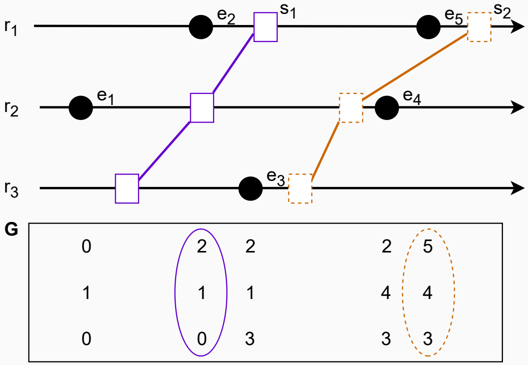


t:	0	1	2	3	4	5
G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3

s_1

- $T_{N(s_1)} = (2, 1, 0)$
- $T_{N(\hat{s}_1)} = (2, 1, 0)$

Example



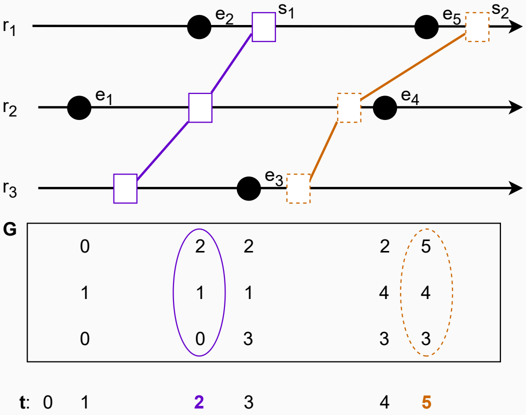
t:	0	1	**2**	3	4	**5**
G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3

s_1

- $T_{N(s_1)} = (2, 1, 0)$
- $T_{N(\hat{s}_1)} = (2, 1, 0)$

s_2

Example



t:	0	1	2	3	4	5
G		0	2	2	2	5
		1	1	1	4	4
		0	0	3	3	3

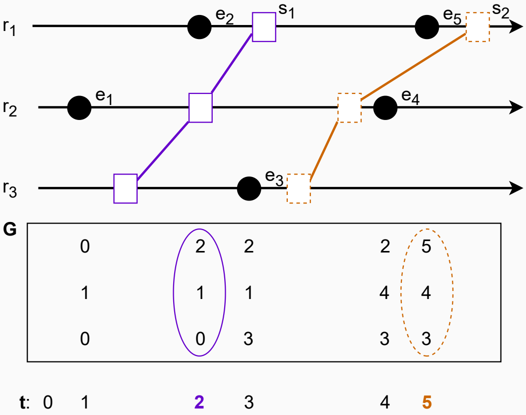
s_1

- $T_{N(s_1)} = (2, 1, 0)$
- $T_{N(\hat{s}_1)} = (2, 1, 0)$

s_2

- $T_{N(s_2)} = (5, 1, 3)$

Example



G					
	0	2	2	2	5
	1	1	1	4	4
	0	0	3	3	3

t: 0 1 **2** 3 4 **5**

s_1

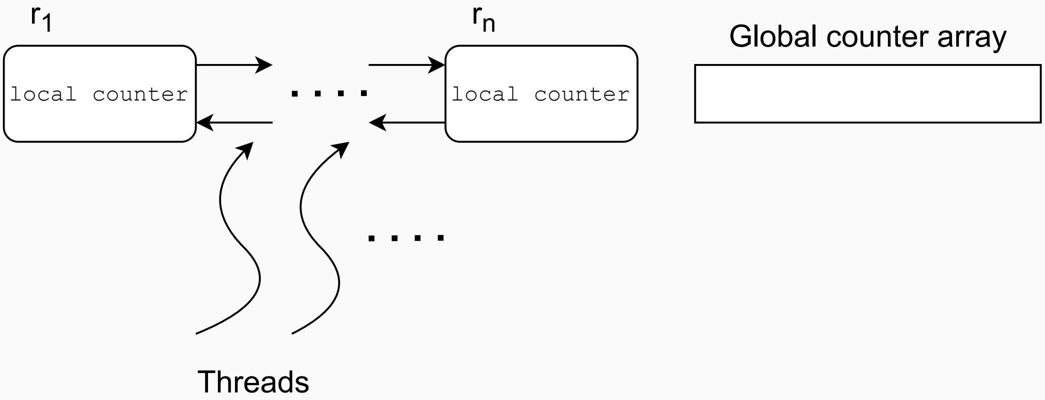
- $T_{N(s_1)} = (2, 1, 0)$
- $T_{N(\hat{s}_1)} = (2, 1, 0)$

s_2

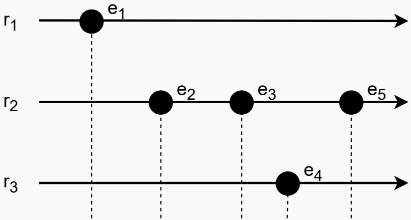
- $T_{N(s_2)} = (5, 1, 3)$
- $T_{N(\hat{s}_2)} = (5, 4, 3)$

Practical Evaluation

Pivot Program

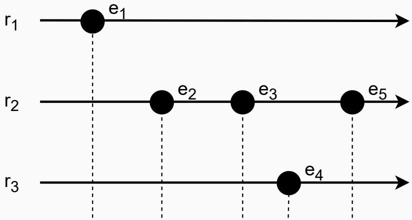


Global Counter Array Simplification

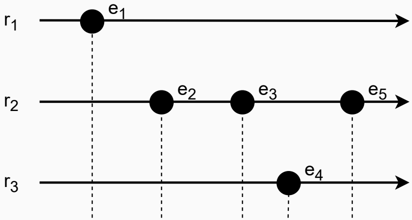


G	0	**1**	1	1	1	1
	0	0	**2**	**3**	3	**5**
	0	0	0	0	**4**	4
t:	0	1	2	3	4	5

Global Counter Array Simplification



G	0	**1**	1	1	1	1
	0	0	**2**	**3**	3	**5**
	0	0	0	0	**4**	4
t:	0	1	2	3	4	5



G	0	1	2	2	3	2
t:	0	1	2	3	4	5

Experiment Setup

Environment

- VM
- Ubuntu 18.04
- 4 GB RAM

[6] `https://github.com/504ensicsLabs/LiME`

[7] `https://github.com/volatilityfoundation/volatility`

[8] Pagani, Fedorov, and Balzarotti 2019

Experiment Setup

Environment

- VM
- Ubuntu 18.04
- 4 GB RAM

Tools

- LiME [6]
- Volatility [7]

[6] https://github.com/504ensicsLabs/LiME

[7] https://github.com/volatilityfoundation/volatility

[8] Pagani, Fedorov, and Balzarotti 2019

Experiment Setup

Environment

- VM
- Ubuntu 18.04
- 4 GB RAM

Tools

- LiME [6]
- Volatility [7]

Consistency indicators

- Local counters & global counter array in pivot program

[6] https://github.com/504ensicsLabs/LiME

[7] https://github.com/volatilityfoundation/volatility

[8] Pagani, Fedorov, and Balzarotti 2019

Experiment Setup

Environment

- VM
- Ubuntu 18.04
- 4 GB RAM

Tools

- LiME [6]
- Volatility [7]

Consistency indicators

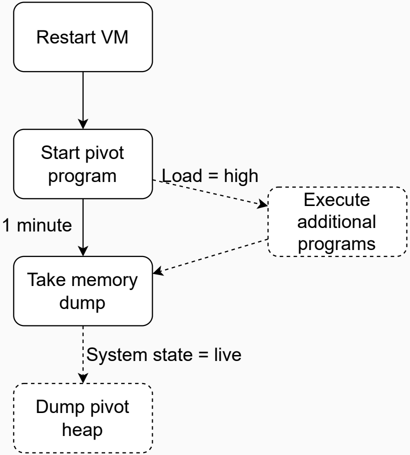
- Local counters & global counter array in pivot program
- VMA count comparison[8]

[6] https://github.com/504ensicsLabs/LiME

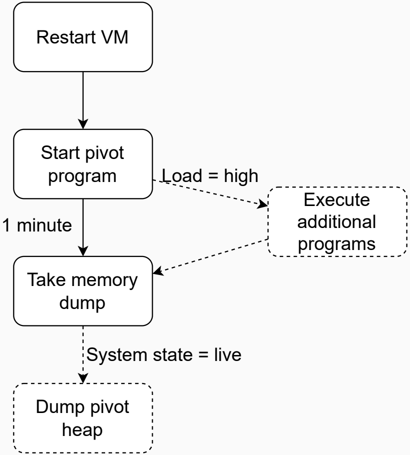
[7] https://github.com/volatilityfoundation/volatility

[8] Pagani, Fedorov, and Balzarotti 2019

Procedure


Restart VM
Start pivot program
Load = high
Execute additional programs
1 minute
Take memory dump
System state = live
Dump pivot heap

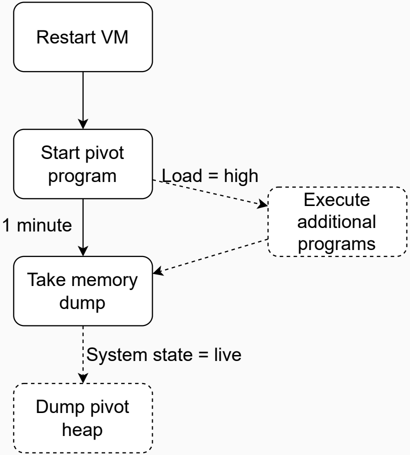
Procedure



Load

- Low: One thread

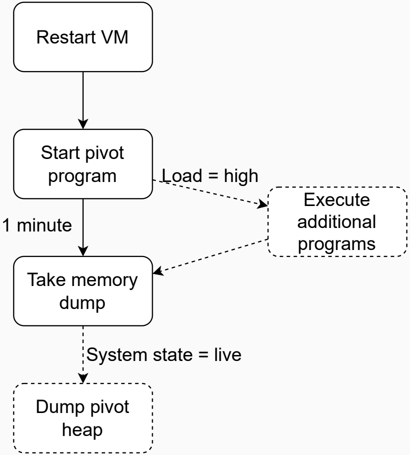
Procedure



Load

- Low: One thread
- High: Eight threads

Procedure



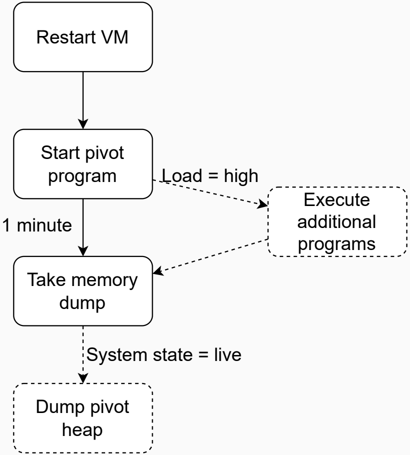
Load

- Low: One thread
- High: Eight threads

System states

- Frozen

Procedure



Load

- Low: One thread
- High: Eight threads

System states

- Frozen
- Live

Results

Sytem State	Inconsistency type	Activity	Min	Max	Average	Affected dumps
Live	Quasi-instantaneous	Low	0	3	0.8	5/10
		High	0	37	13.8	7/10
	VMA	Low	0	1	0.1	1/10
		High	3	7	4.9	9/9

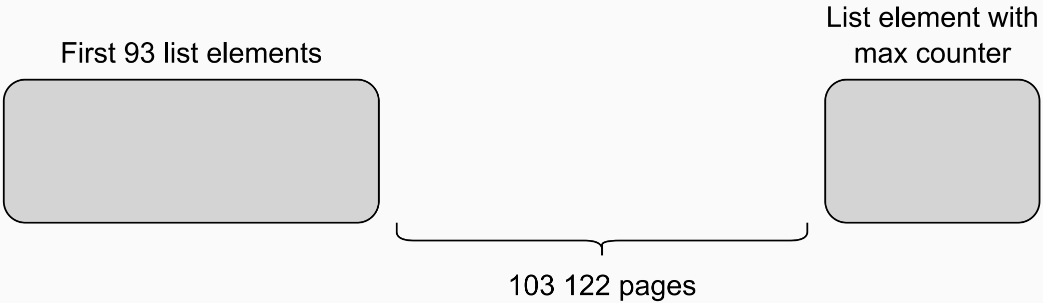
Discussion

Inconsistencies and Fragmentation

#	Inconsistencies	Range (in pages)	Distances <= 10 pages	Max distance
1	37	224 575	61	103 122
2	30	423 245	47	79 613
3	21	141 591	20	54 774
4	17	150 635	33	53 319
5	16	267 028	44	82 596
6	15	79 296	85	71 215
7	2	99 921	81	55 761
8	0	82 526	76	62 653
9	0	12 132	75	3 170
10	0	4 431	97	2 665

Taking a Closer Look

Dump no. 1



Pivot Program

Benefits

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies
- Observe the influence of fragmentation

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies
- Observe the influence of fragmentation
- Size of observed range flexible

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies
- Observe the influence of fragmentation
- Size of observed range flexible

Possible adjustments

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies
- Observe the influence of fragmentation
- Size of observed range flexible

Possible adjustments

- Influence fragmentation

Pivot Program

Benefits

- Exact quantification & localization of inconsistencies
- Observe the influence of fragmentation
- Size of observed range flexible

Possible adjustments

- Influence fragmentation
- Influence position in physical address space

Conclusion

Summary

- Observation method works
  - Theoretical proof
  - Practical case study

Summary

- Observation method works
    - Theoretical proof
    - Practical case study

- Most memory dumps are not quasi-instantaneous

Summary

- Observation method works
  - Theoretical proof
  - Practical case study

- Most memory dumps are not quasi-instantaneous
- Benefits of deliberately placed consistency indicators

Future Work

- Influence position of pivot program

Future Work

- Influence position of pivot program
- Observe quasi-instantaneous consistency at a higher level

Future Work

- Influence position of pivot program
- Observe quasi-instantaneous consistency at a higher level

Future Work

- Influence position of pivot program
- Observe quasi-instantaneous consistency at a higher level

- Extensive tool evaluations

Future Work

- Influence position of pivot program
- Observe quasi-instantaneous consistency at a higher level

- Extensive tool evaluations
- Search for additional consistency indicators

Future Work

- Influence position of pivot program
- Observe quasi-instantaneous consistency at a higher level

- Extensive tool evaluations
- Search for additional consistency indicators

Thank you for your attention!

References i

References

- Case, Andrew and Golden G Richard III (2017). “Memory forensics: The path forward”. In: *Digital Investigation* 20, pp. 23–33.
- Gruhn, Michael and Felix C Freiling (2016). “Evaluating atomicity, and integrity of correct memory acquisition methods”. In: *Digital Investigation* 16, S1–S10.
- Lempereur, Brett, Madjid Merabti, and Qi Shi (2012). “Pypette: A Platform for the Evaluation of Live Digital Forensics”. In: *Int. Journal of Digital Crime and Forensics* 4.4, pp. 31–46.

References ii

- Ottmann, Jenny, Frank Breitinger, and Felix Freiling (2022). “Defining Atomicity (and Integrity) for Snapshots of Storage in Forensic Computing”. In: *Proceedings of the Digital Forensics Research Conference Europe (DFRWS EU)*. Oxford.
- Pagani, Fabio, Oleksii Fedorov, and Davide Balzarotti (2019). “Introducing the temporal dimension to memory forensics”. In: *ACM Transactions on Privacy and Security (TOPS)* 22.2, pp. 1–21.
- Vömel, Stefan and Johannes Stüttgen (2013). “An evaluation platform for forensic memory acquisition software”. In: *Digital Investigation* 10, S30–S40.