CaT
Content-aware Tracing and Analysis for Distributed Systems

Tânia Esteves, Francisco Neves, Rui Oliveira and João Paulo

INESC TEC & University of Minho

in 22nd International Middleware Conference (Middleware '21)

Universidade do Minho

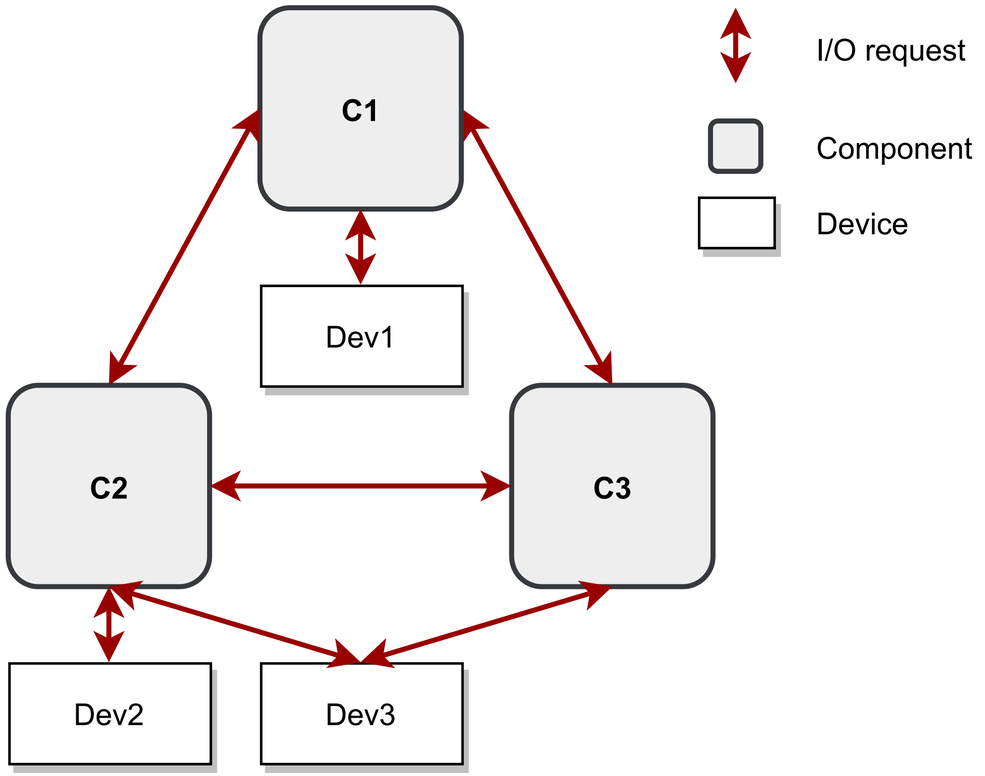
Acknowledgments: This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.

Tracing Distributed Systems

- Developing, configuring, and managing distributed systems are difficult, costly, and challenging tasks
- Tracing and analysis frameworks provide valuable insights into how the system's state evolves over time
- Key for performance analysis, diagnosing anomalies, correctness and security

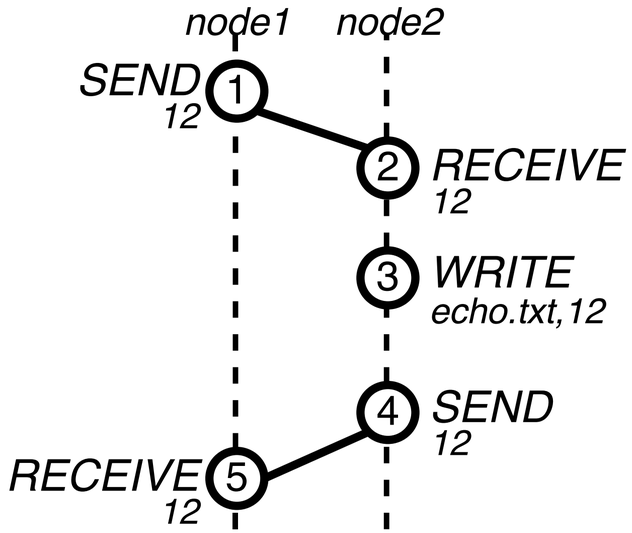
Challenges And Problems

- Performance and storage overhead
- Transparency
- Accuracy
- Causality
- Automation and visualization



State Of The Art

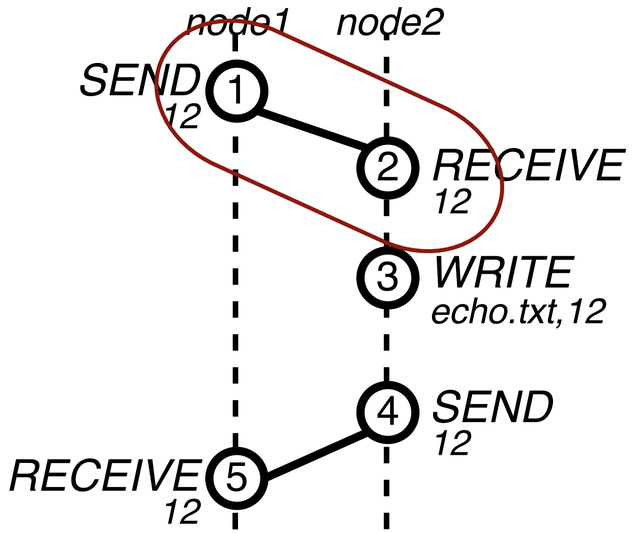
- Current tools either take an intrusive approach or only take into account the requests' context.



Context-based tracing

State Of The Art

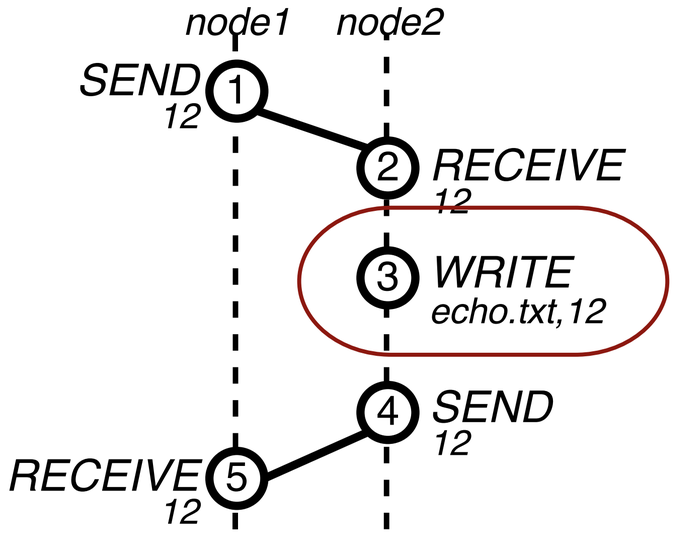
- Current tools either take an intrusive approach or only take into account the requests' context.



Context-based tracing

State Of The Art

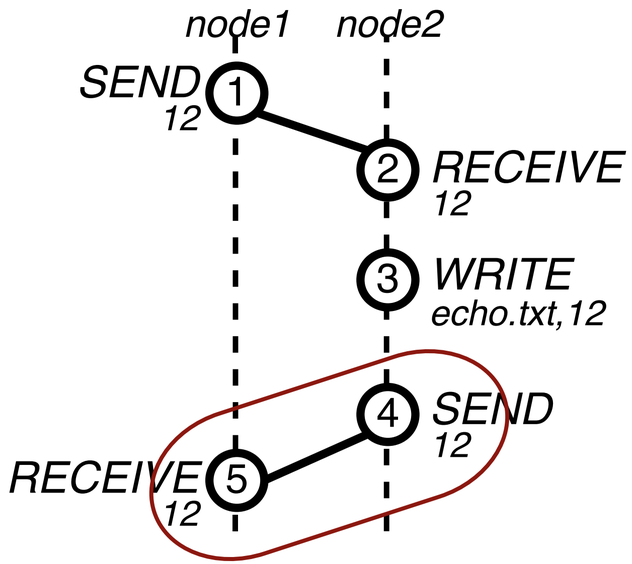
- Current tools either take an intrusive approach or only take into account the requests' context.



Context-based tracing

State Of The Art

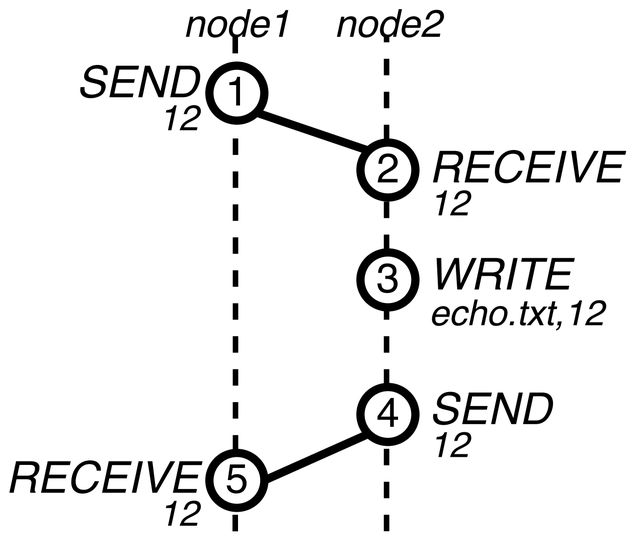
- Current tools either take an intrusive approach or only take into account the requests' context.



Context-based tracing

State Of The Art

- Current tools either take an intrusive approach or only take into account the requests' context.

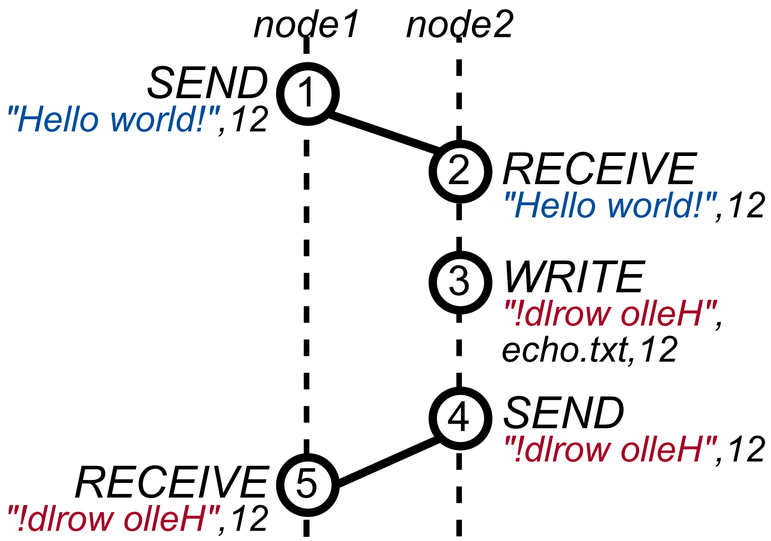


What if the content is actually different?

Context-based tracing

Key Insights

- To capture and analyze both the context and content of requests.

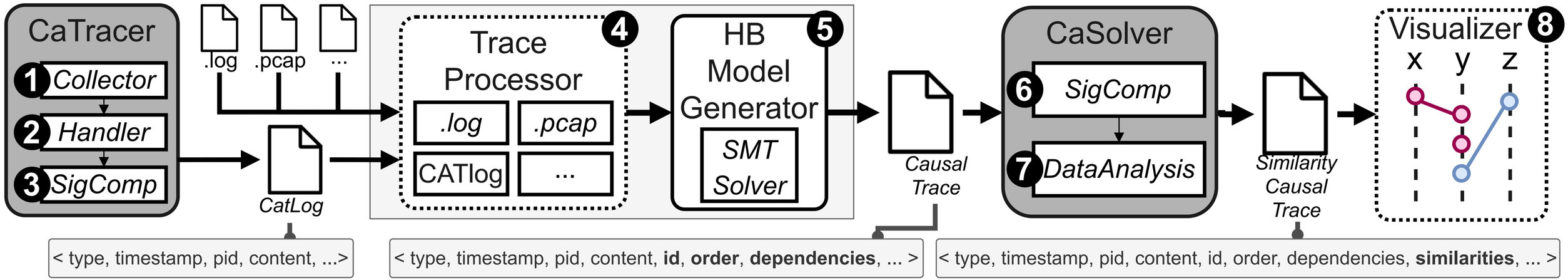


Content-based tracing

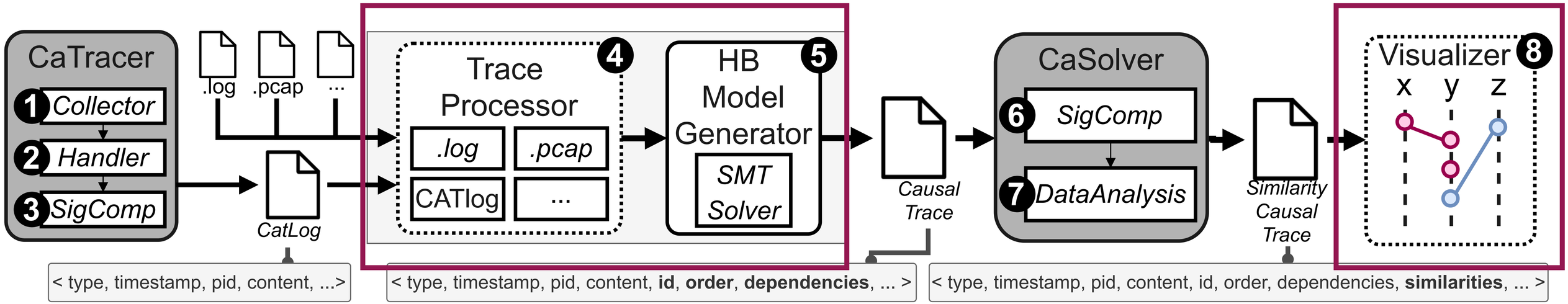
Contributions

- **Content-aware tracing**: Captures and analyses the context and content of applications' I/O requests
- **Non-intrusive tracing**: Uses kernel-level tracing tools (Strace and eBPF) to capture I/O requests
- **Open-source prototype**: A fully integrated pipeline to capture, analyze and visualize the context and content of I/O requests
- **Evaluation**: A detailed evaluation using two real Big Data applications: TensorFlow and Apache Hadoop

CAT Architecture

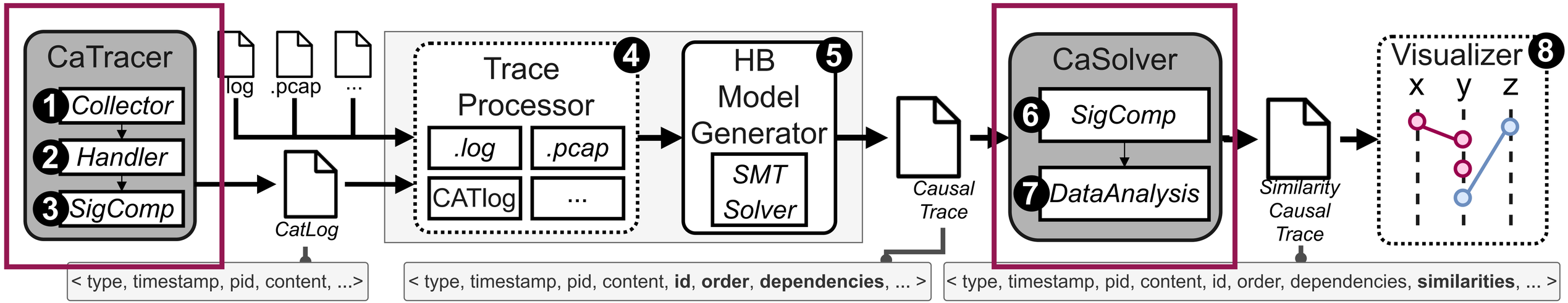


CAT Architecture

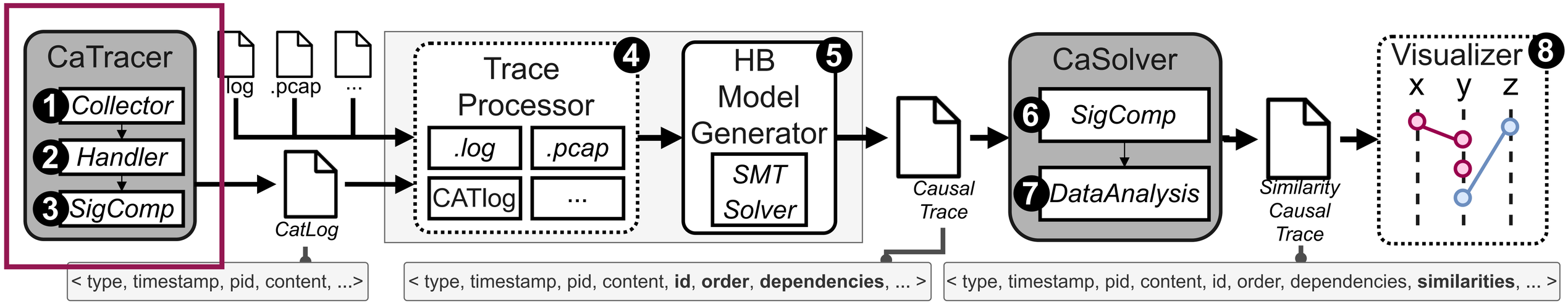


F. Neves, N. Machado and J. Pereira, "**Falcon: A Practical Log-Based Analysis Tool for Distributed Systems**," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)

CAT Architecture



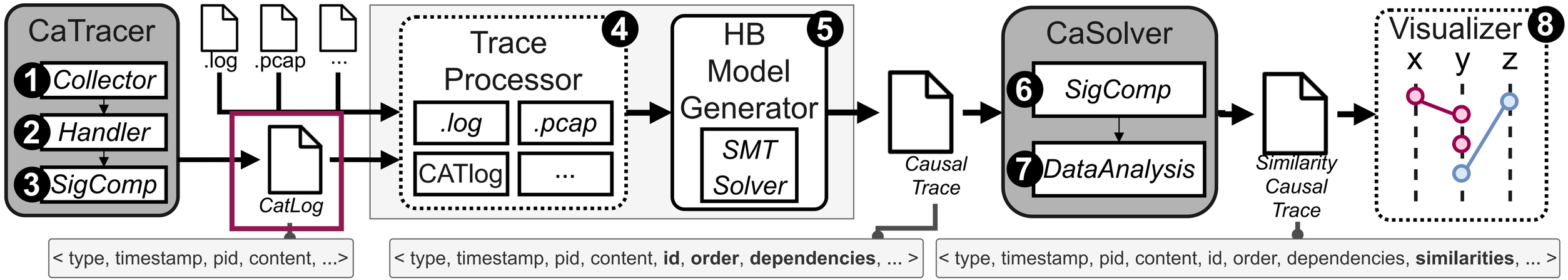
CAT Architecture



CaTracer: collects information about I/O requests

I/O request => event

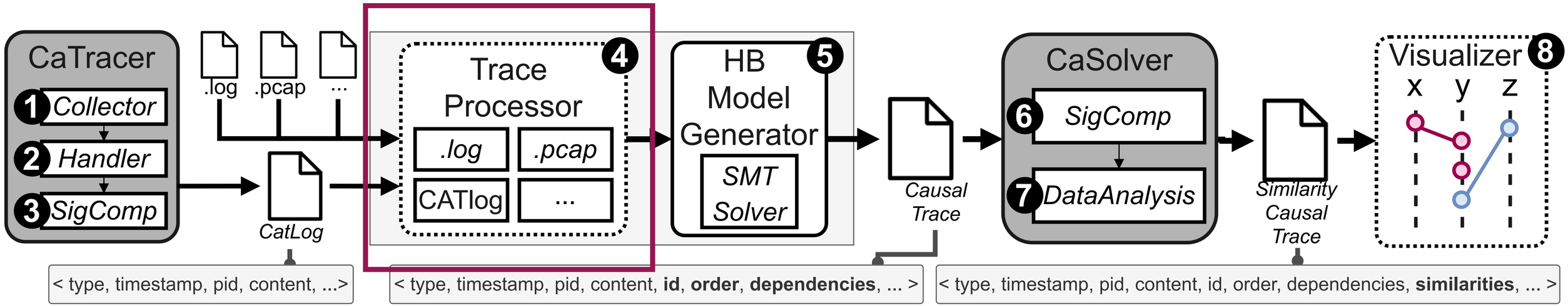
CAT Architecture



CaTracer: collects information about I/O requests

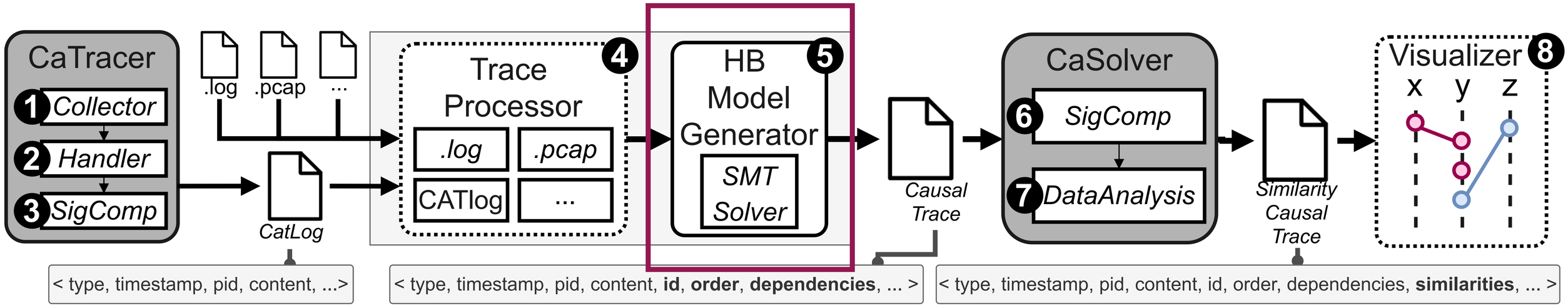
I/O request => event

CAT Architecture



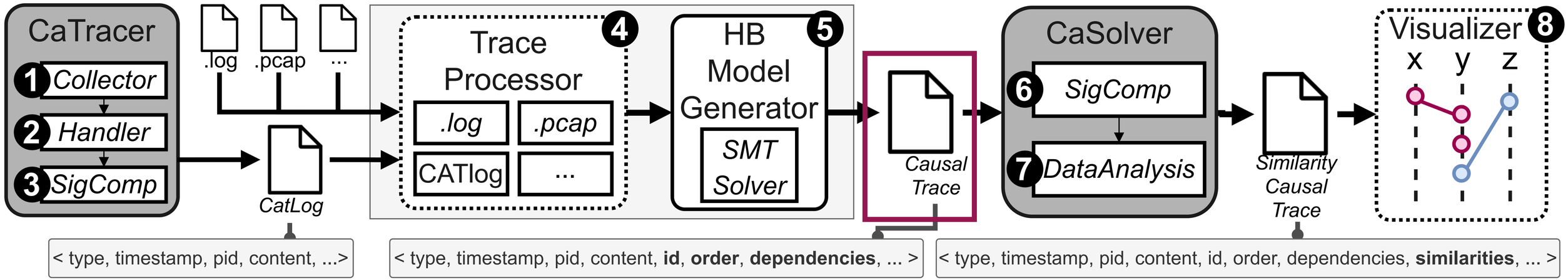
Trace Processor: parses and organizes the events into different data structures.

CAT Architecture

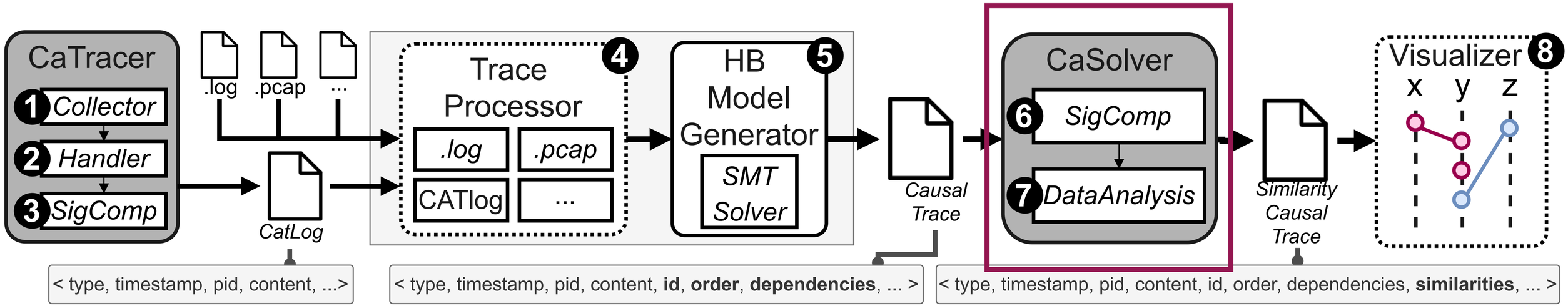


HB Model Generator: infers the causality between events.

CAT Architecture

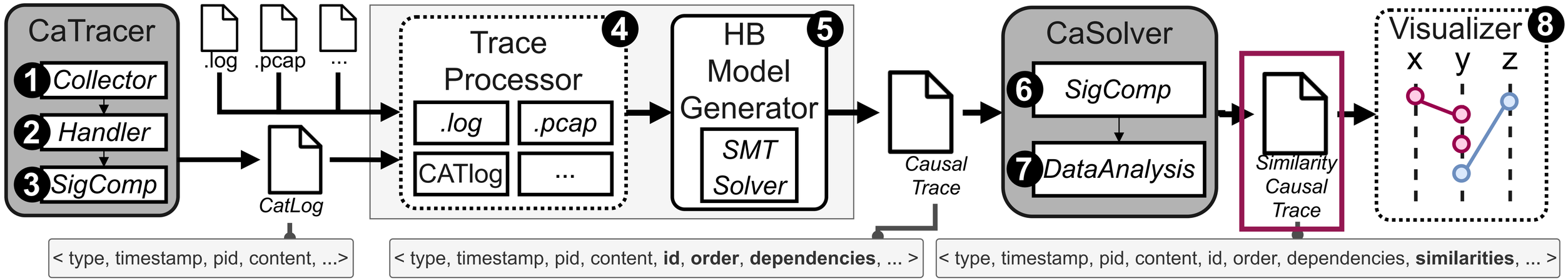


CAT Architecture

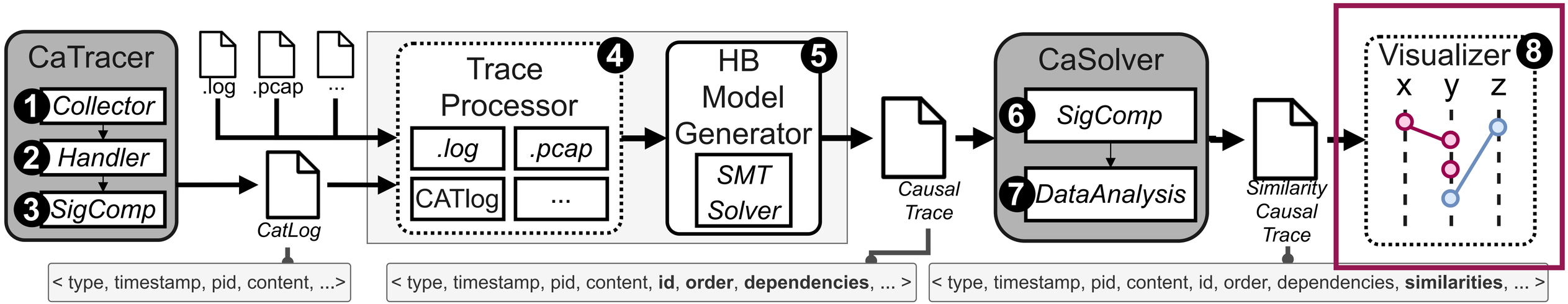


CaSolver: finds events with a high probability of operating over the same data flow.

CAT Architecture



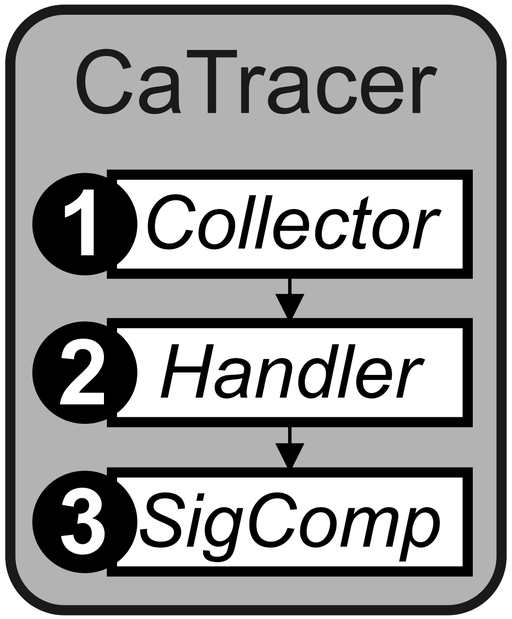
CAT Architecture



Visualizer: builds a space-time diagram representing the targeted system execution, the events causal relationships and their data flows.

CaTracer

- Three main submodules:
  - **Collector**: captures applications I/O requests
  - **Handler**: parses, organizes and saves the requests
  - **SigComp**: compute hash sums of requests' content
- Two implementations:
  - **CatStrace** - strace-based tracer
  - **CatBpf** - eBPF-based tracer



CASOLVER

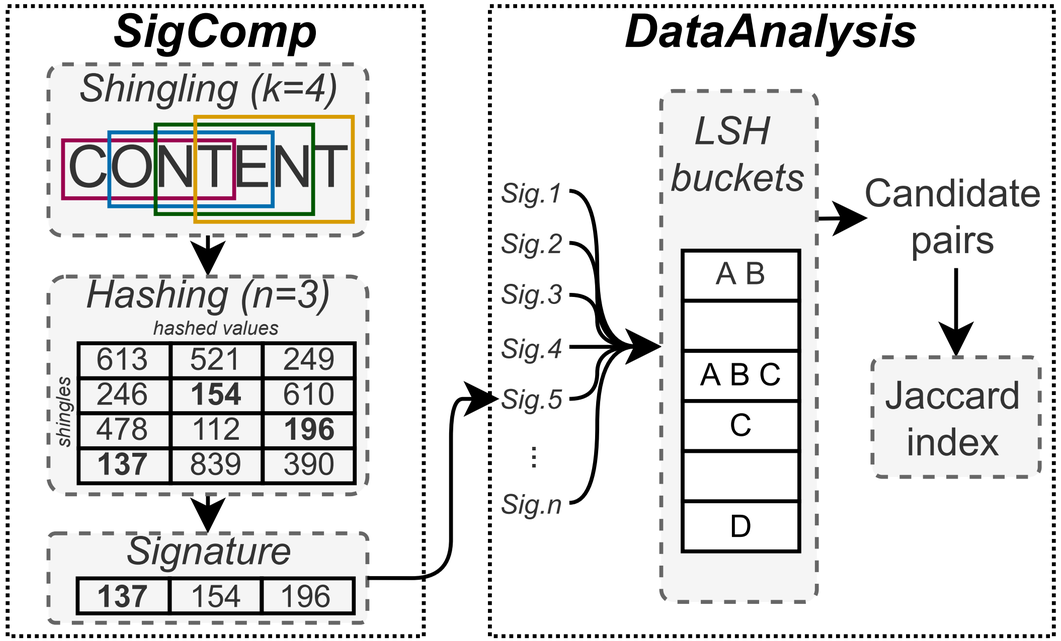
SigComp submodule:

- Resorts to the *min-wise hashing* (MinHash) algorithm to summarize the events content into a small set of signatures

DataAnalysis submodule:

- Resorts to the *Locality-sensitive hashing* (LSH) mechanism to find candidate pairs referring to similar content

- *Jaccard index* is used to computed the similarity between the candidate pairs



Visualizer

Color-based representation for data dependencies:

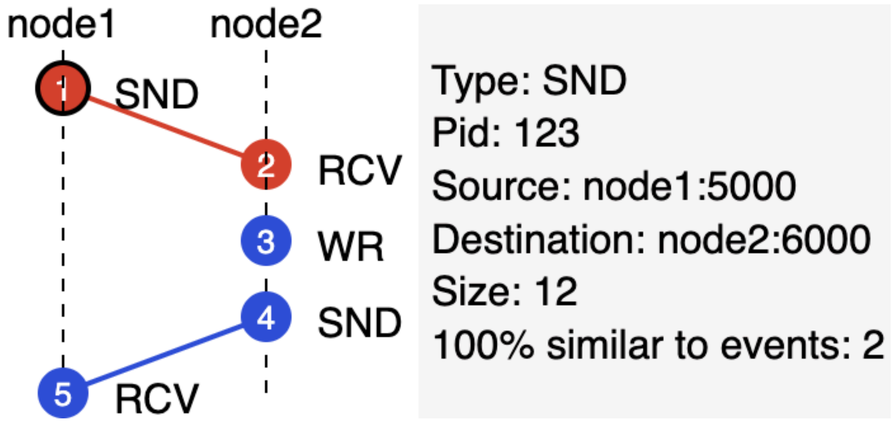
- Events with similar content are depicted with the same color
- Events with unique content are depicted with the black color

Additional information:

- By selecting a specific event or relationship it is possible to consult additional information about it

Storage-based representation:

- An horizontal representation for storage related events



Evaluation

Content-aware tracers evaluation:

- What is the performance impact, resource usage, storage overhead, and accuracy of each CaTracer?

CAT Framework in Action:

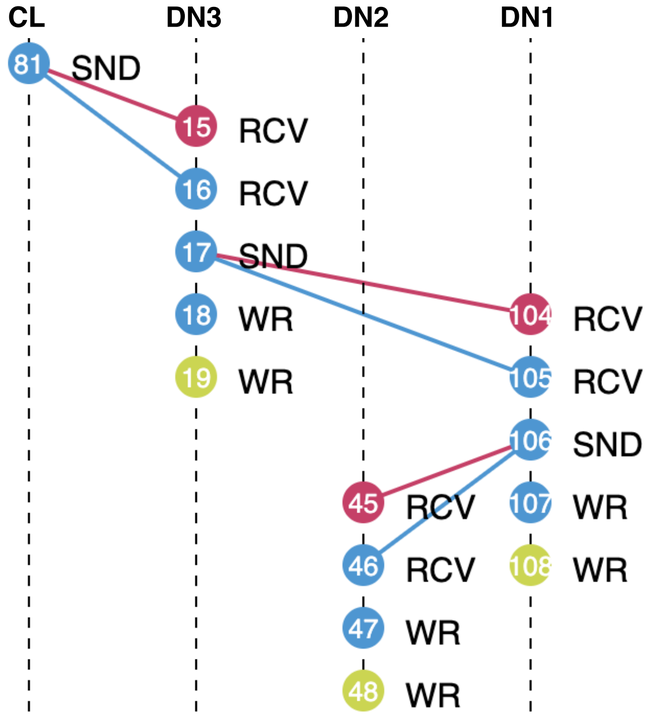
- What novel insights can CAT's content-aware approach provide?

Content-aware tracers evaluation

	CatBpf	CatStrace
Performance and Storage overhead		
	• Minimal	• Significant performance overhead • Can easily generate a file with significant size
Accuracy		
	• Can lose information • Captures only 4 KiB of requests content	• Captures all the requests • Captures 256 KiB of requests content
Resource Usage		
	• Considerable usage of CPU and RAM	• Lower resource usage **(!)**

CAT Framework In Action

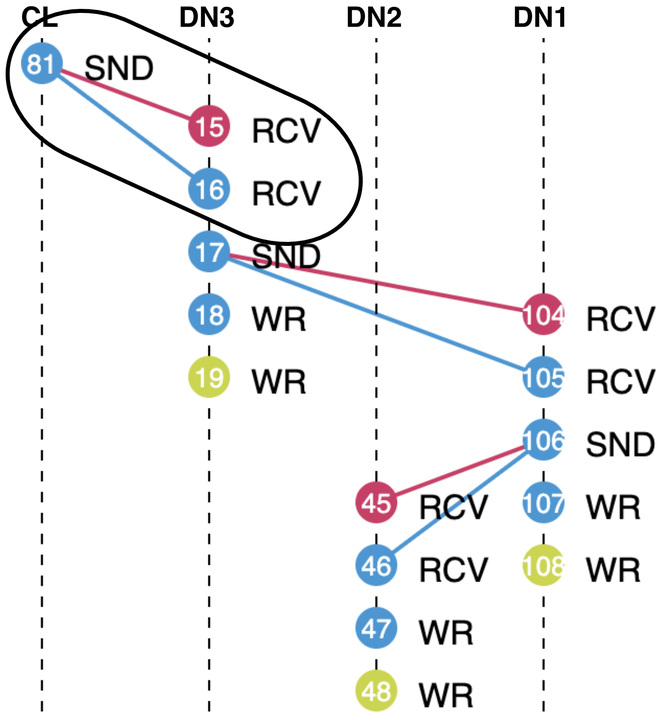
Storage and replication of a file in HDFS



a) Normal execution

CAT Framework In Action

Storage and replication of a file in HDFS

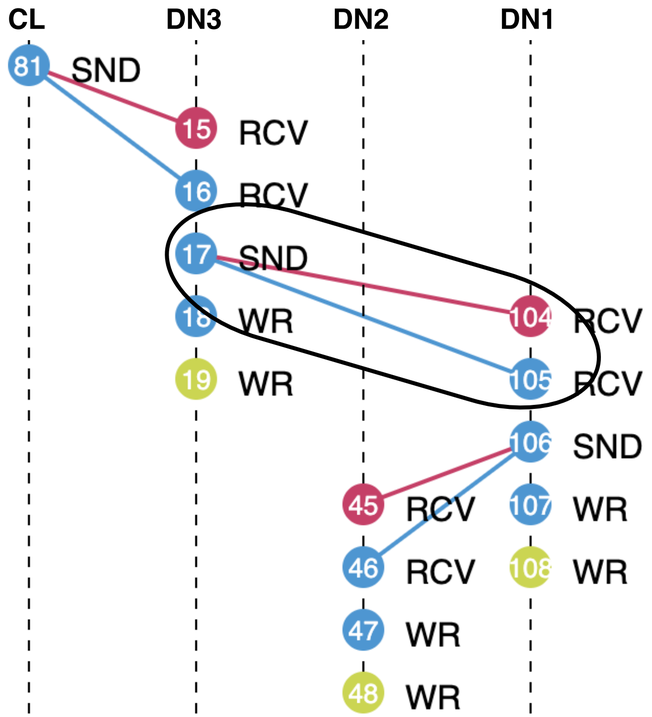


a) Normal execution

Client sent the file to DN3 (81)

CAT Framework In Action

Storage and replication of a file in HDFS

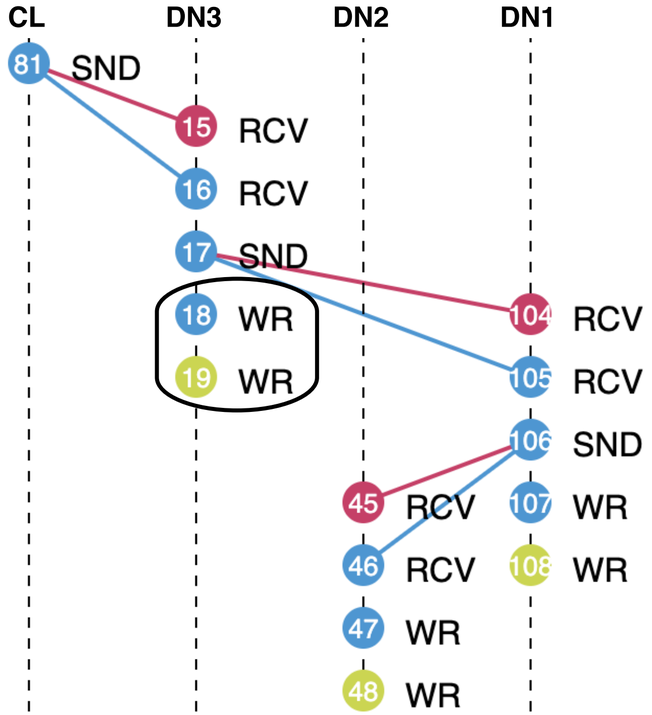


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)

CAT Framework In Action

Storage and replication of a file in HDFS

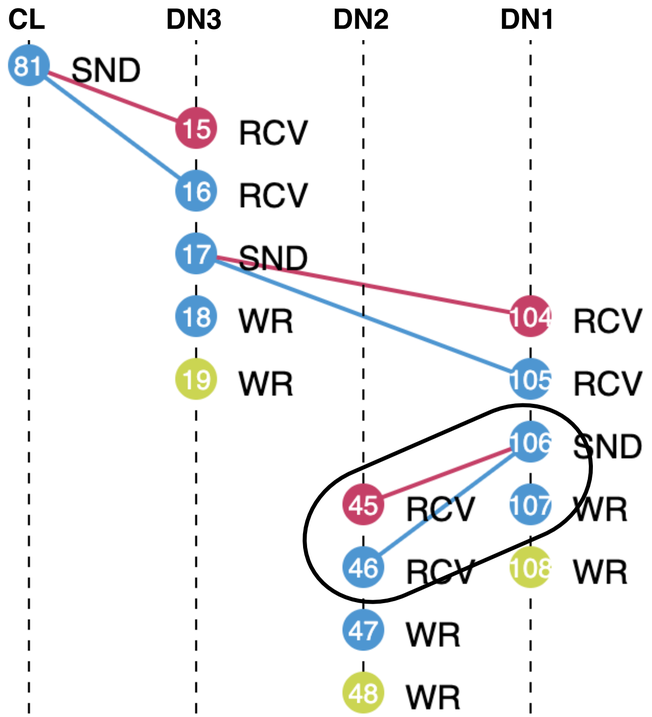


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)

CAT Framework In Action

Storage and replication of a file in HDFS

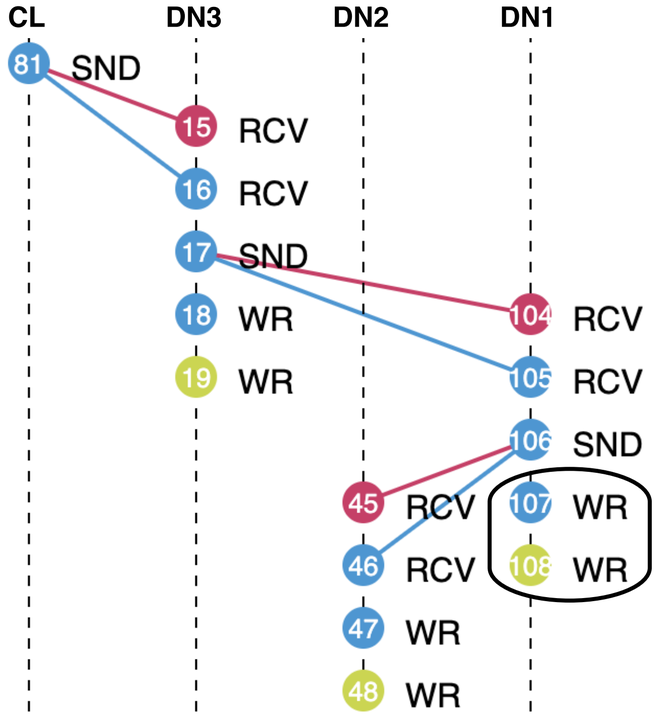


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)

CAT Framework In Action

Storage and replication of a file in HDFS

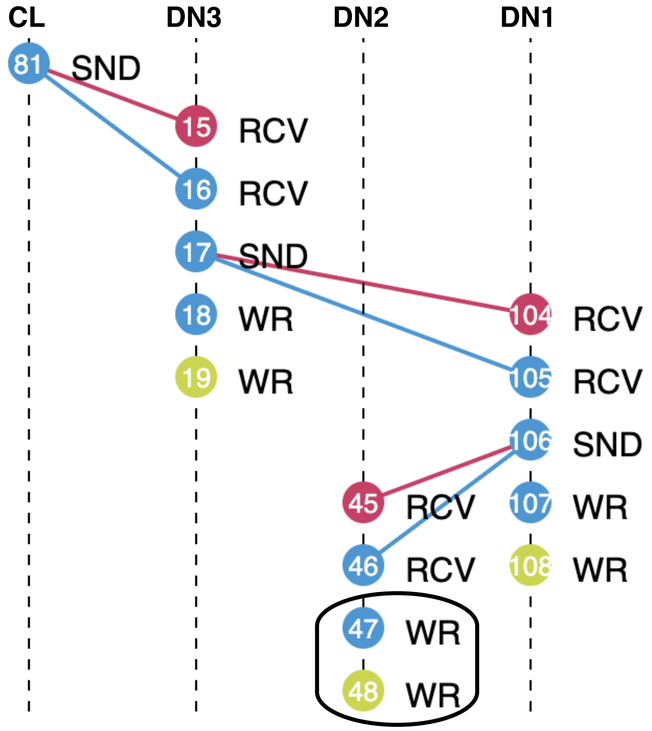


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)

CAT Framework In Action

Storage and replication of a file in HDFS

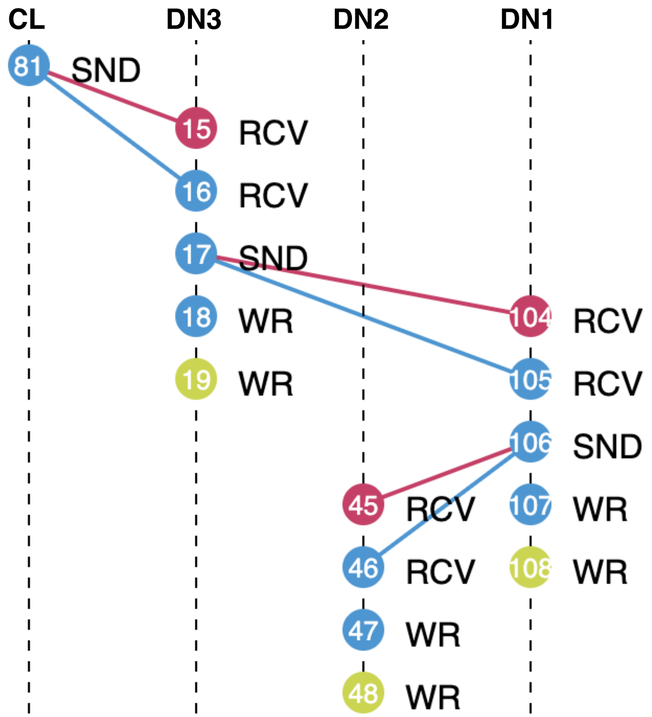


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

CAT Framework In Action

Storage and replication of a file in HDFS

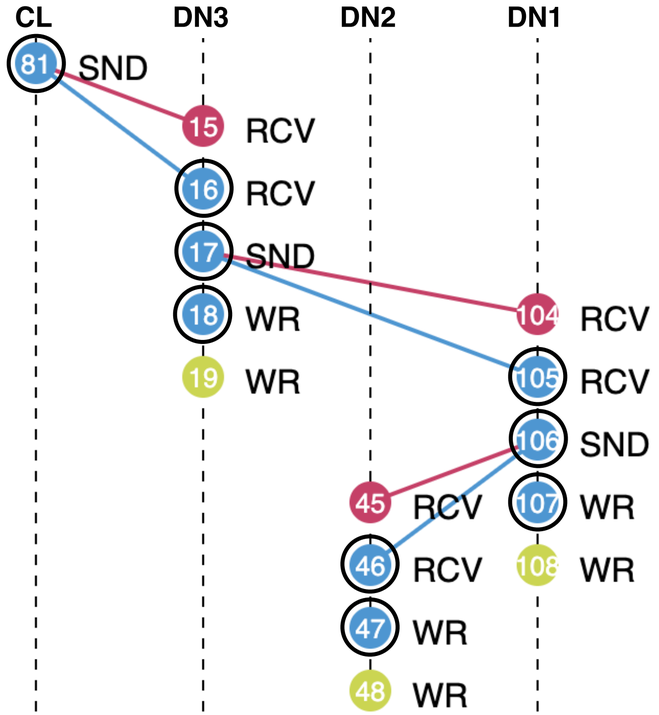


a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

CAT Framework In Action

Storage and replication of a file in HDFS

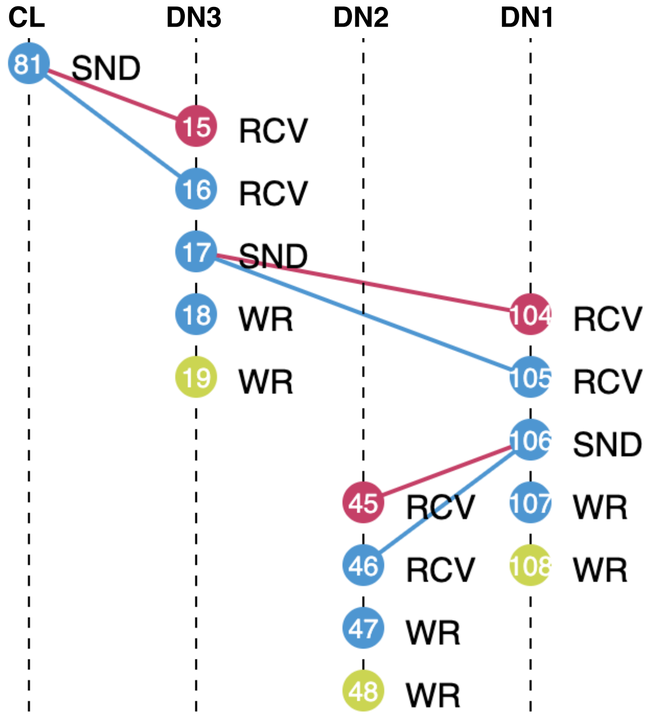


a) Normal execution

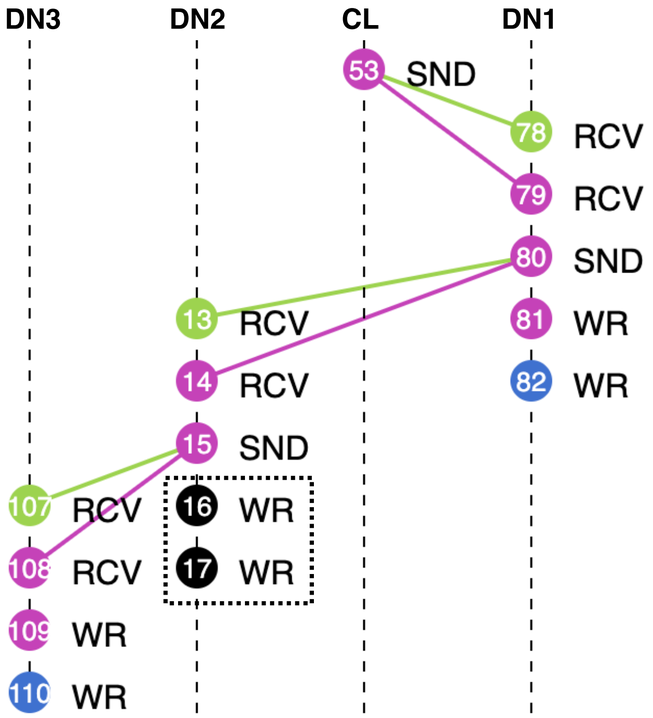
Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

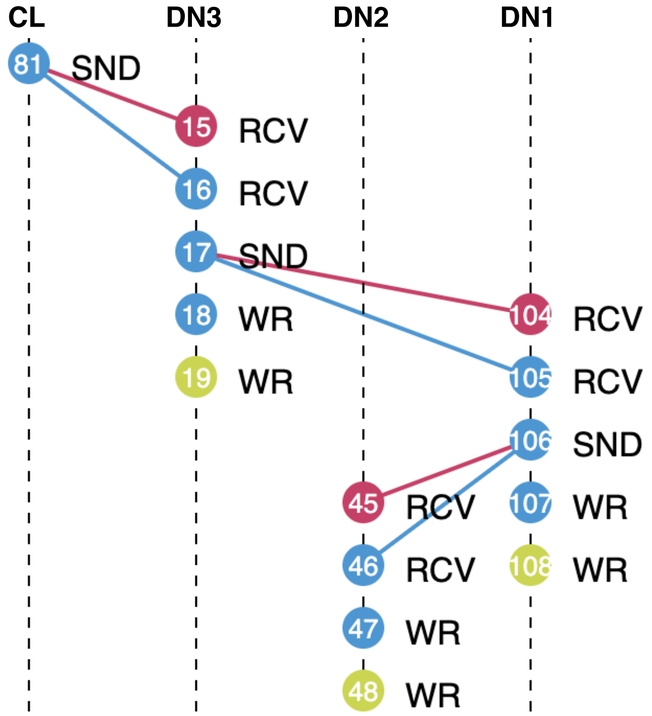


b) Storage corruption

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

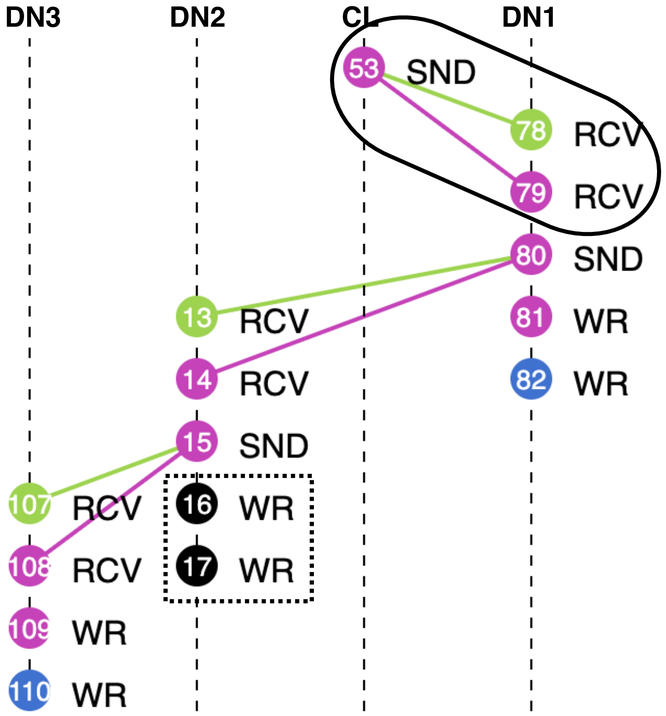
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

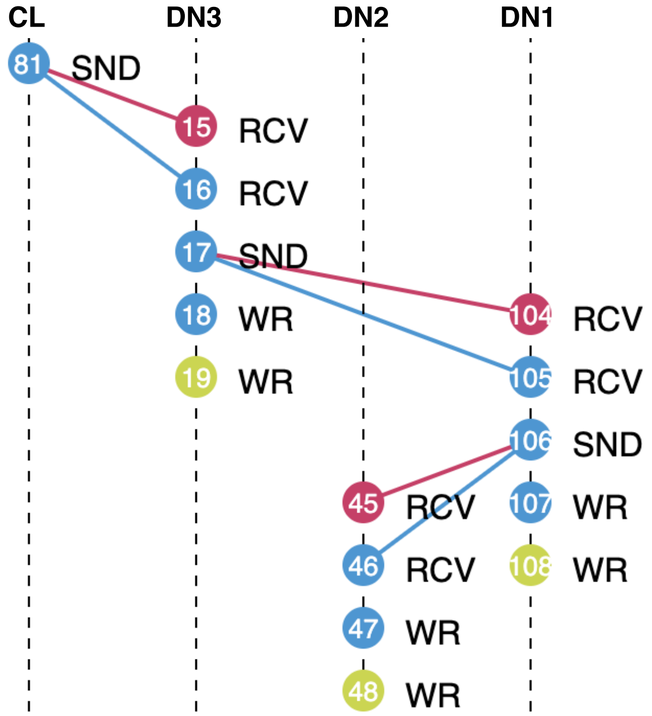


b) Storage corruption

Client sent the file to DN1 (53)

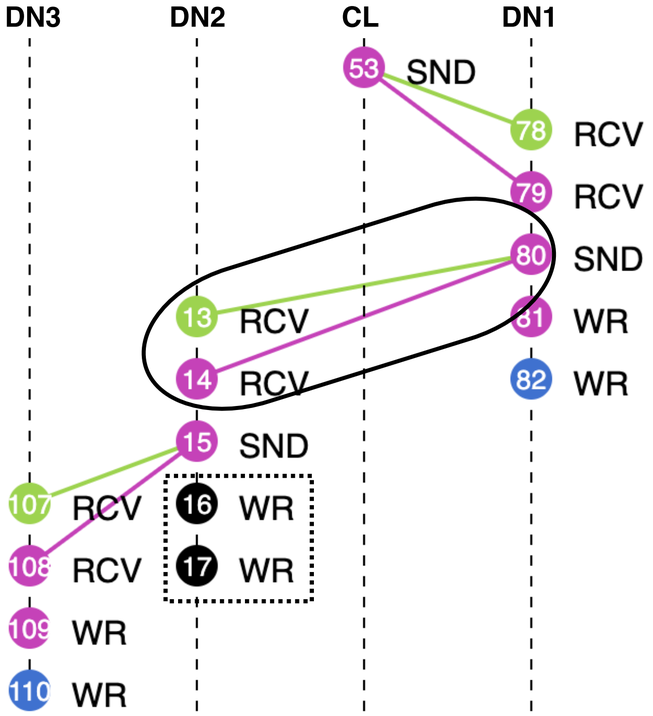
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

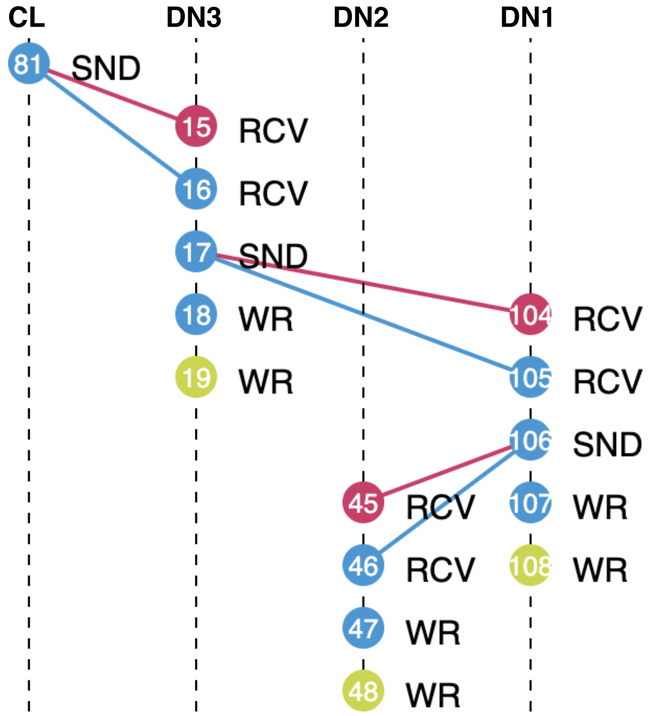


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)

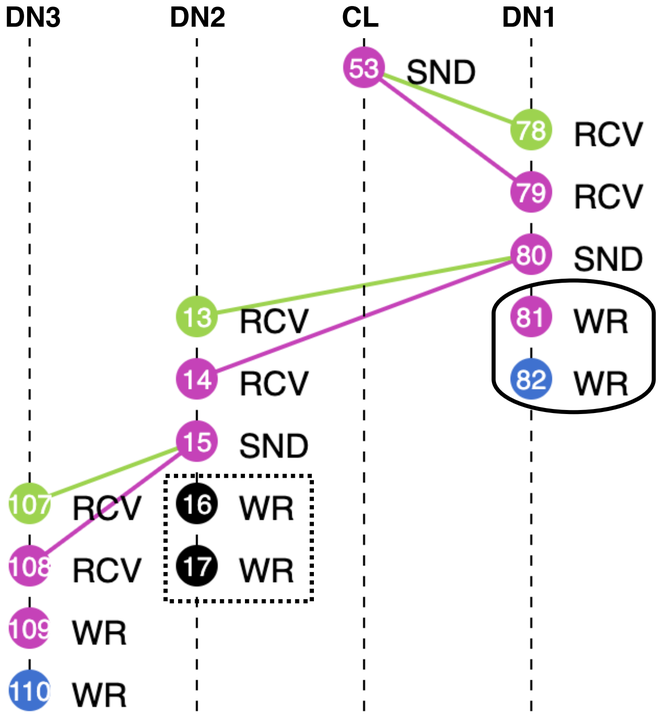
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

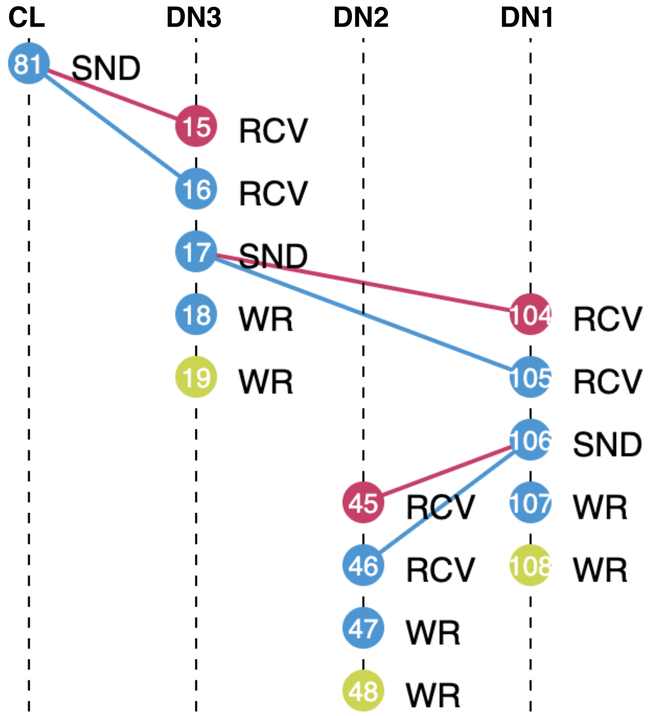


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)

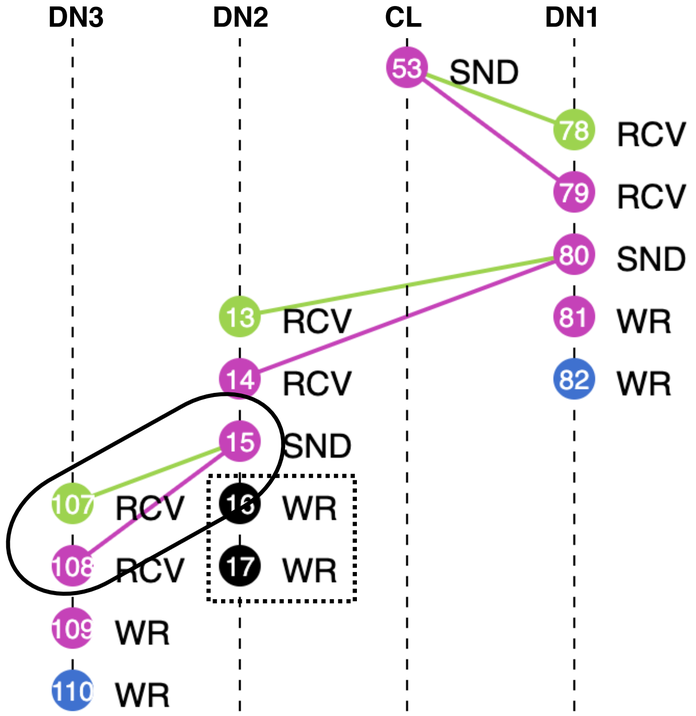
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

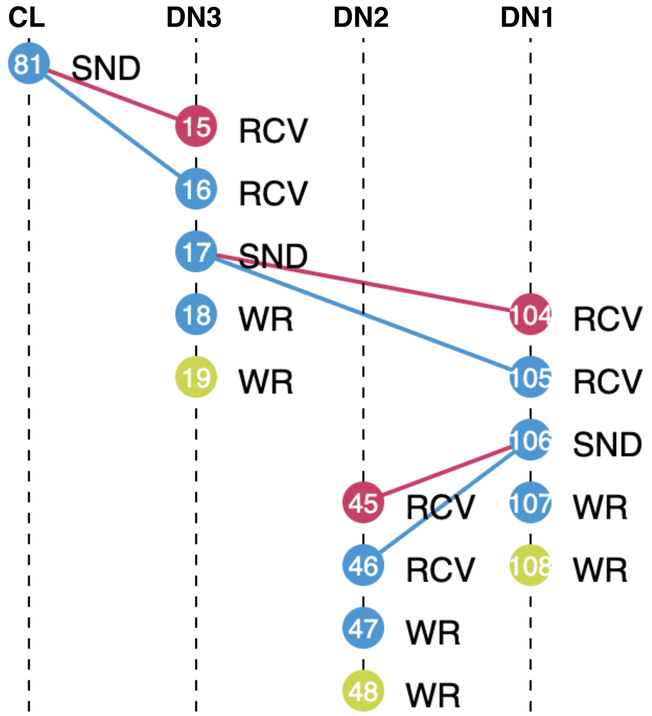


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)

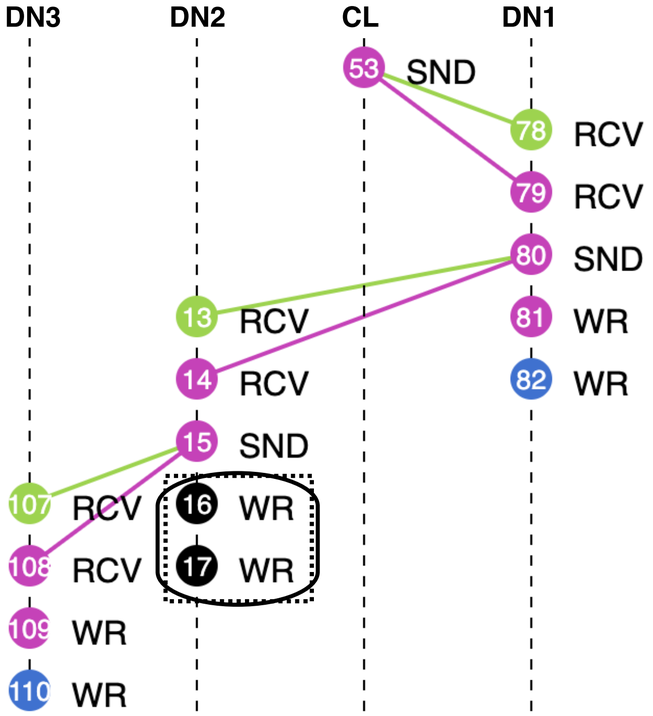
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

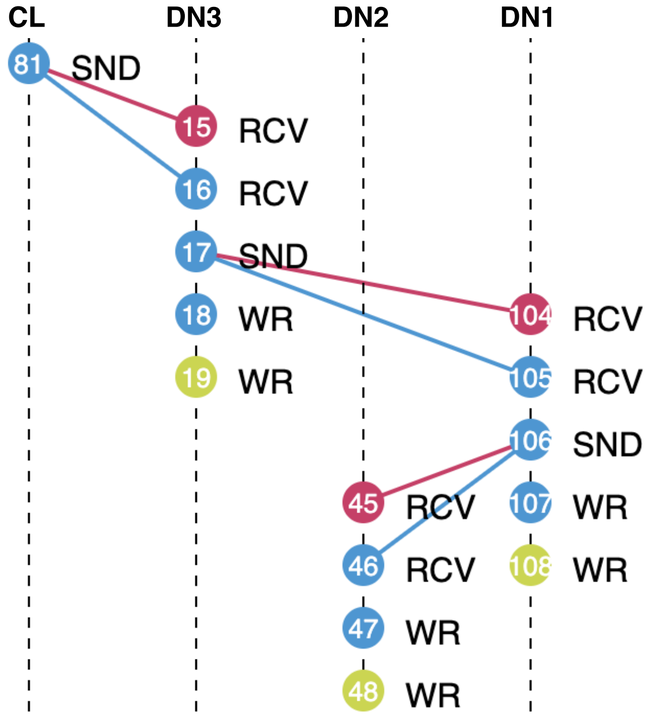


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)

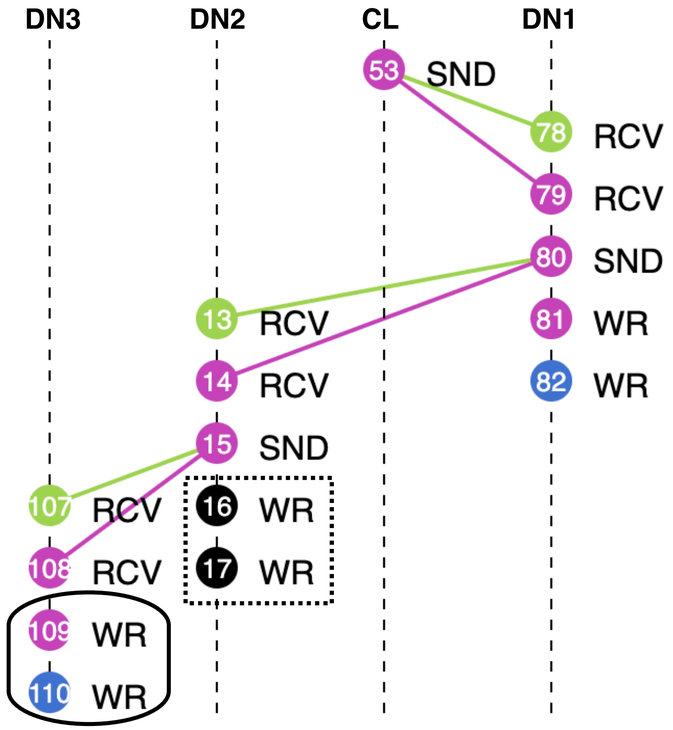
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

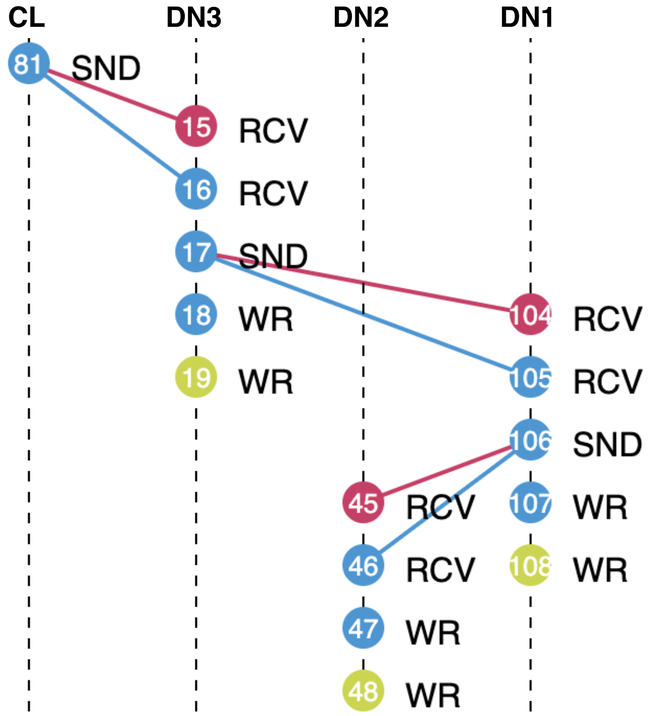


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

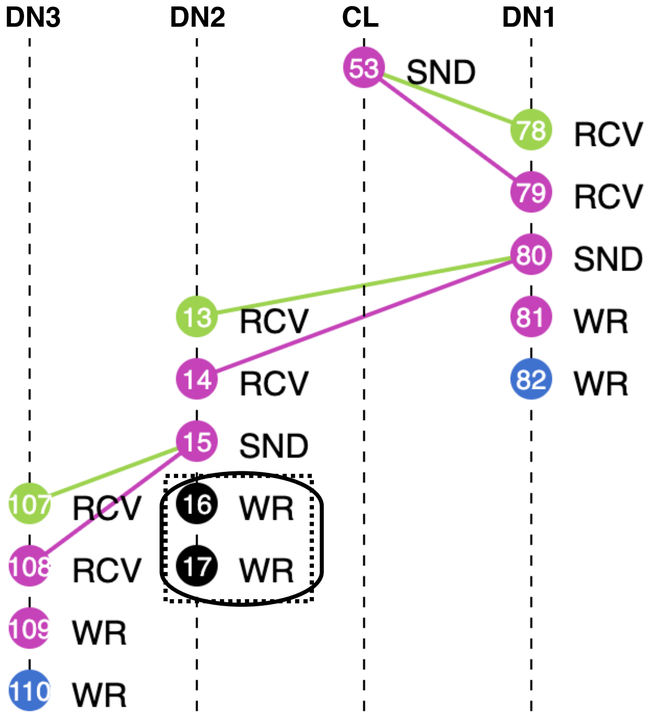
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

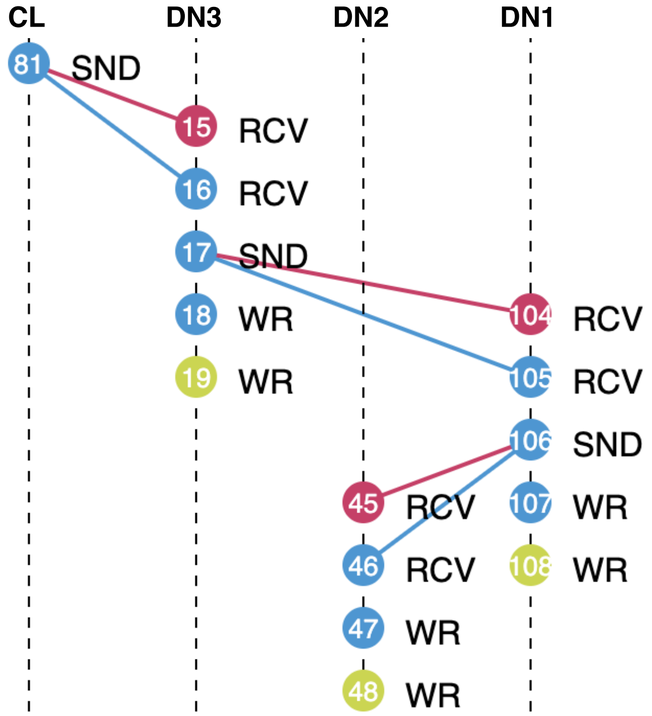


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

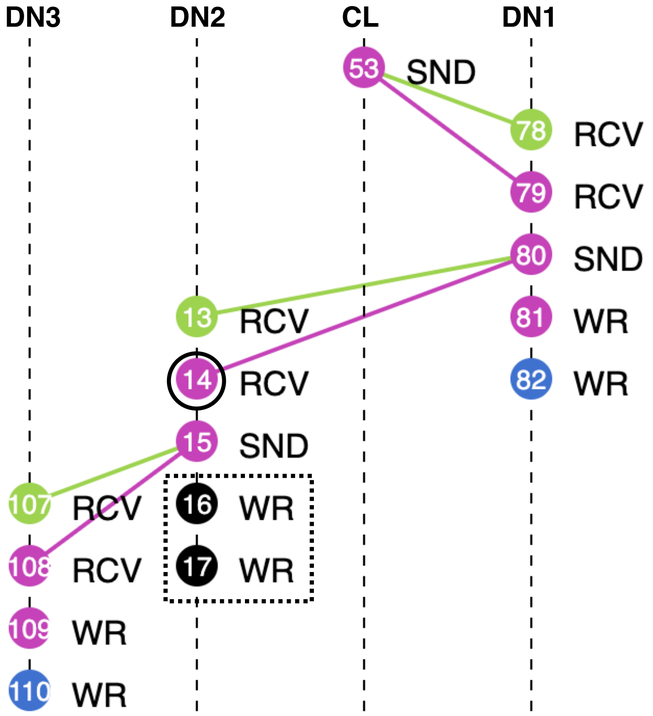
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

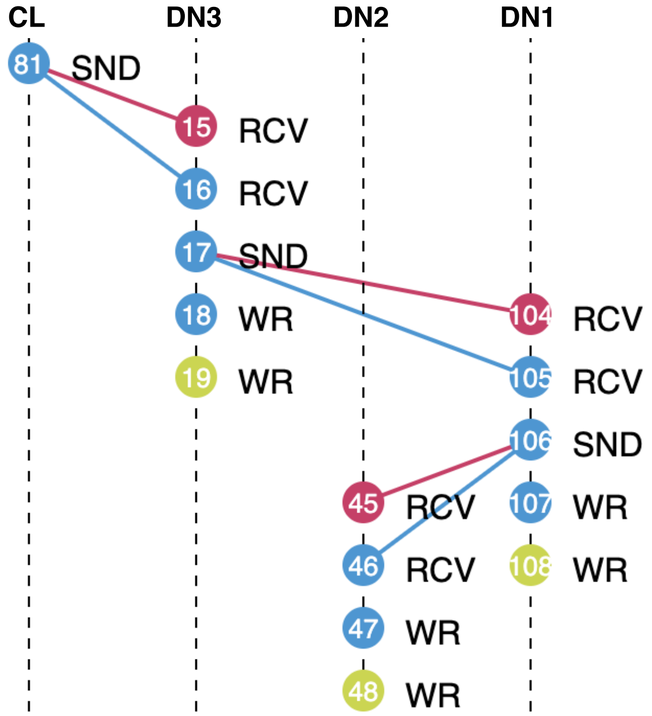


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

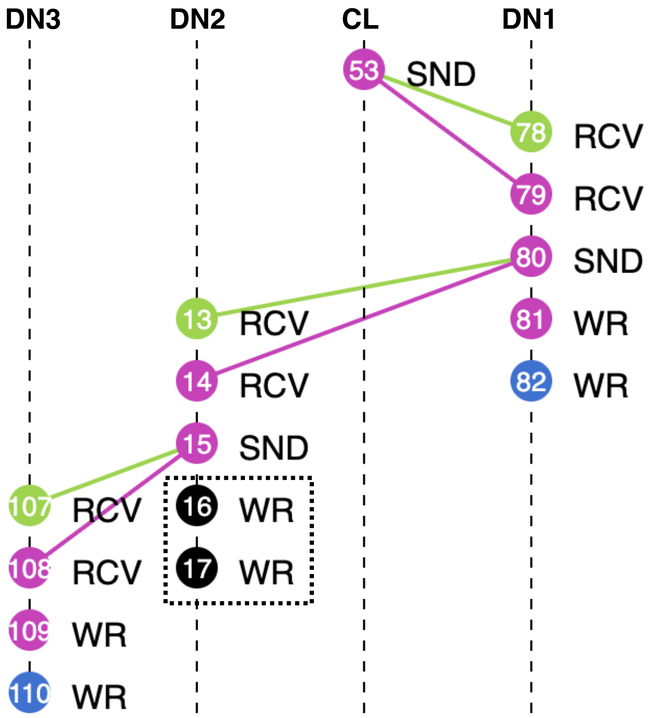
CAT Framework In Action

Storage and replication of a file in HDFS



a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)

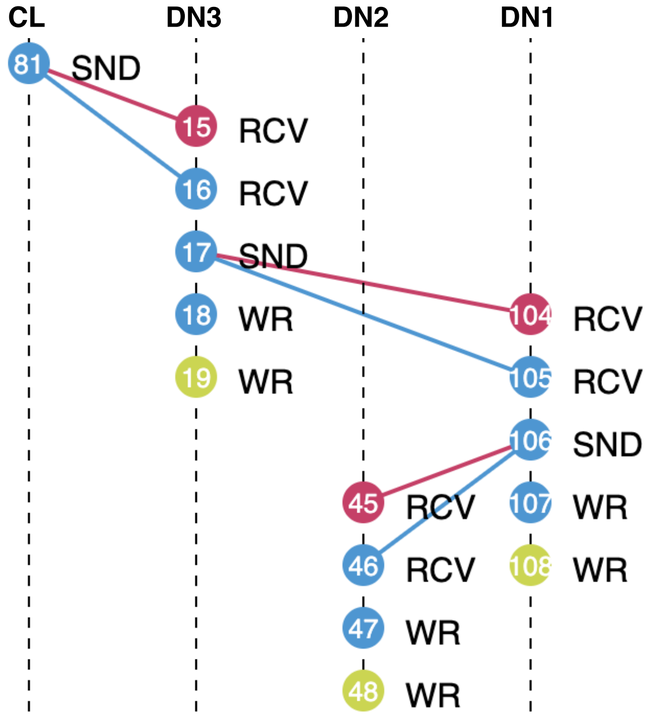


b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

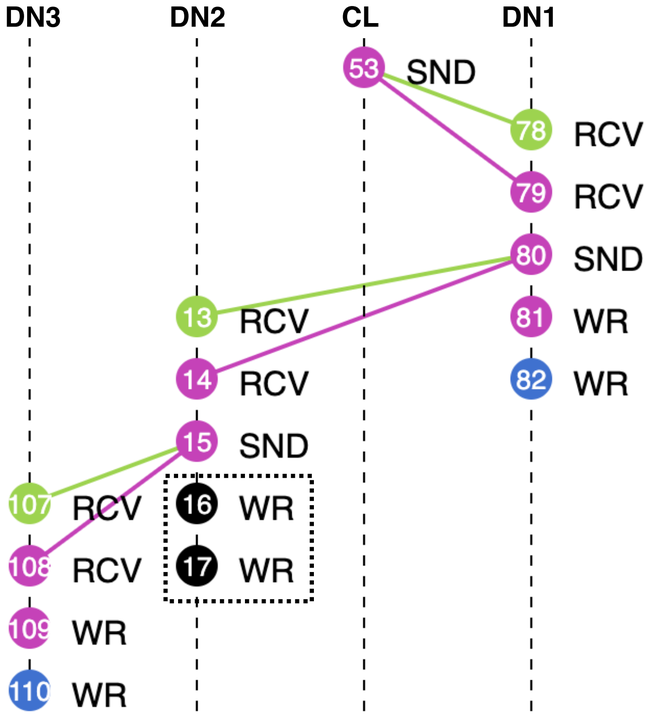
CaT Framework In Action

Storage and replication of a file in HDFS



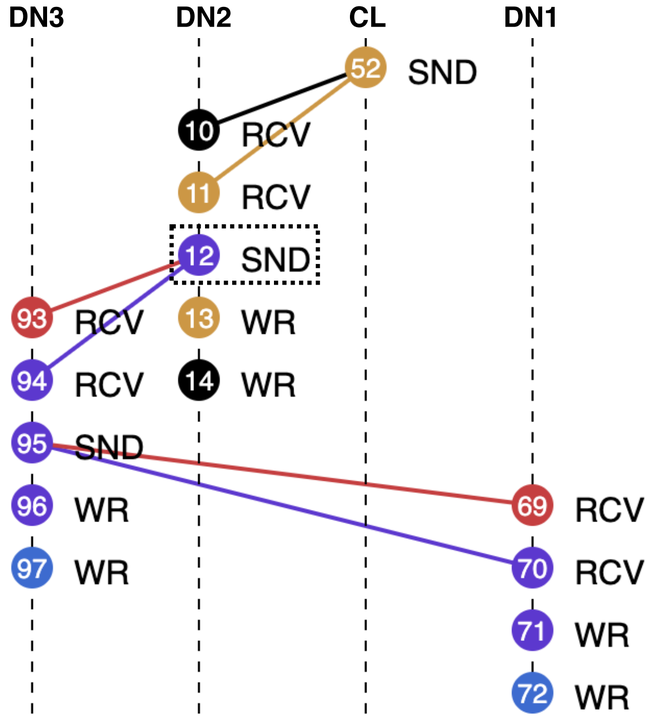
a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)



b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

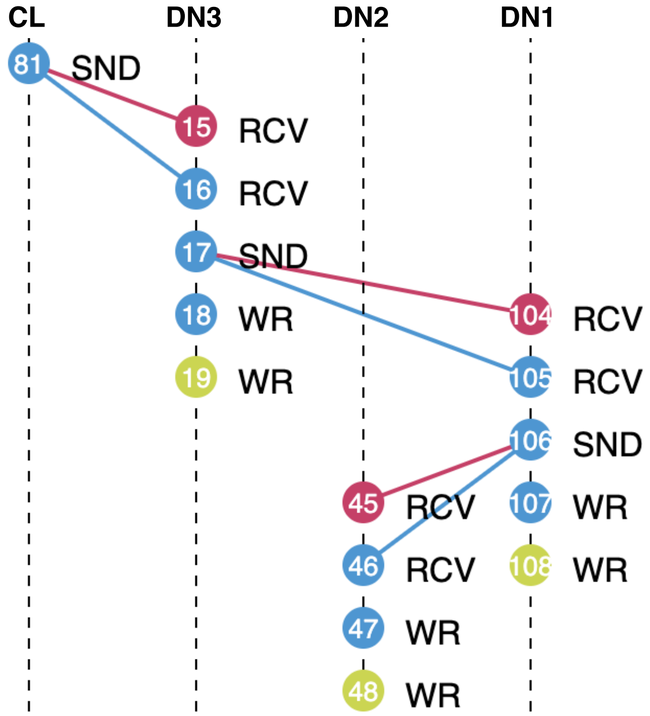


c) Network corruption

Client sent the file to DN2 (52)
DN2 sent it to DN3 (12) and persisted it in disk (13 & 14)
DN3 sent it to DN1 (95) and persisted it in disk (96 & 97)
DN1 persisted it in disk (71 & 72)

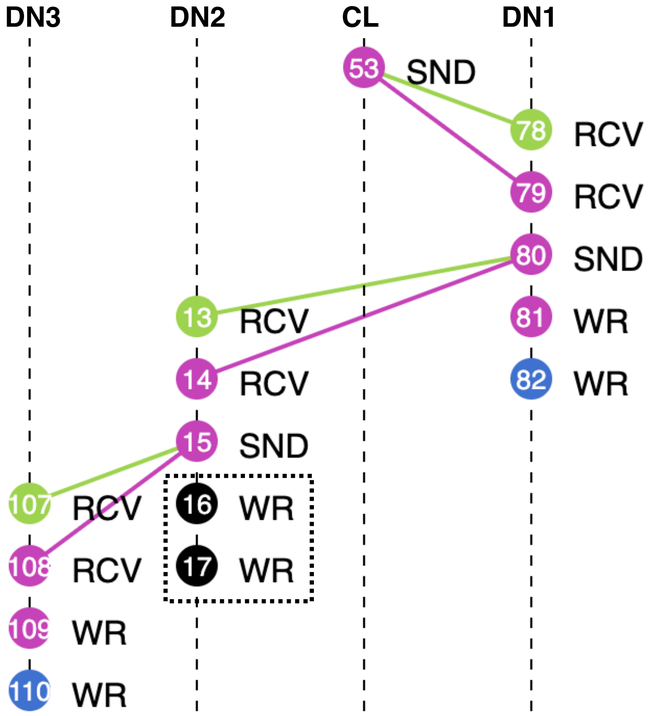
CaT Framework In Action

Storage and replication of a file in HDFS



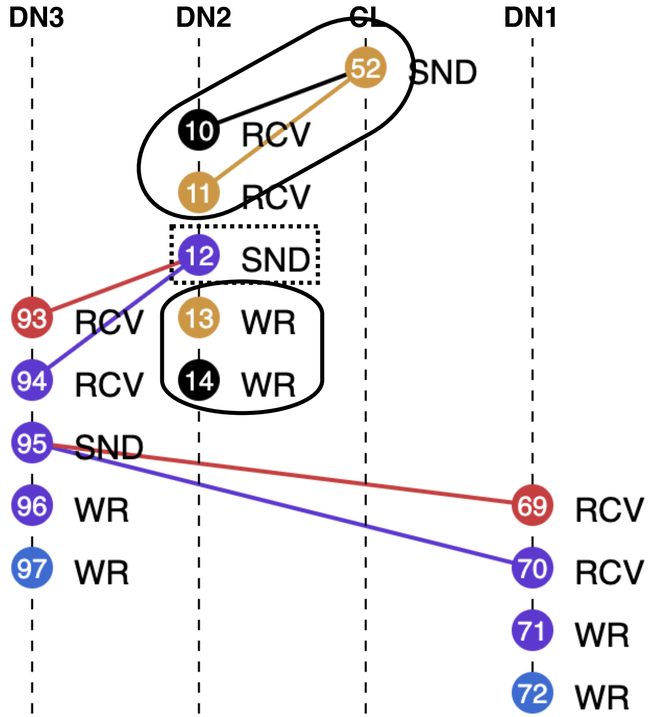
a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)



b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

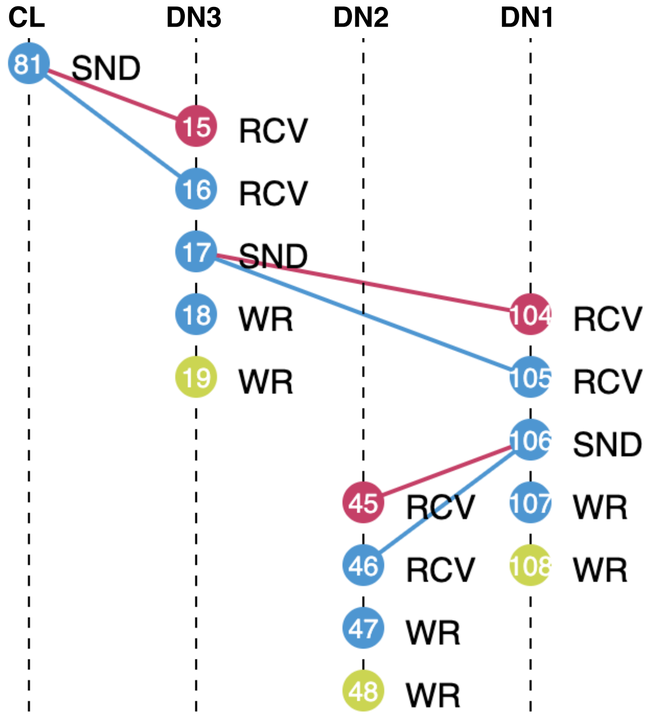


c) Network corruption

Client sent the file to DN2 (52)
DN2 sent it to DN3 (12) and persisted it in disk (13 & 14)
DN3 sent it to DN1 (95) and persisted it in disk (96 & 97)
DN1 persisted it in disk (71 & 72)

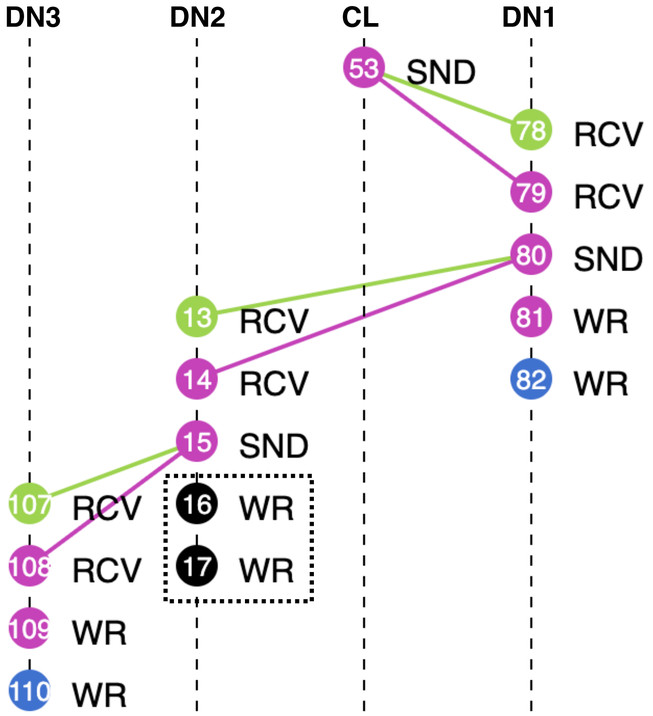
CaT Framework In Action

Storage and replication of a file in HDFS



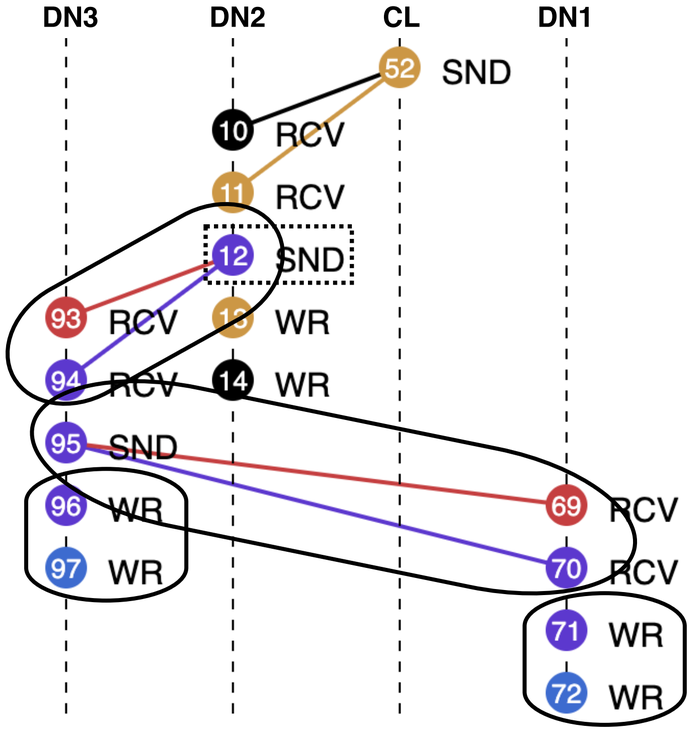
a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)



b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)

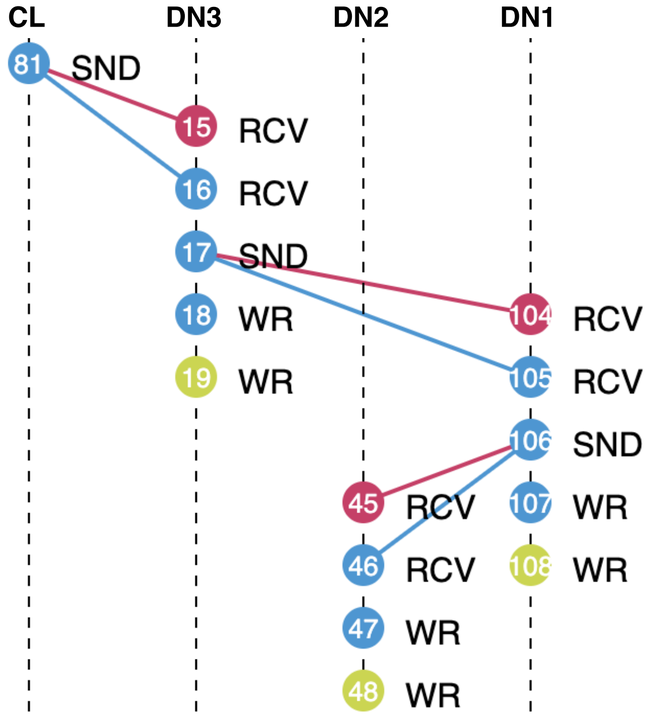


c) Network corruption

Client sent the file to DN2 (52)
DN2 sent it to DN3 (12) and persisted it in disk (13 & 14)
DN3 sent it to DN1 (95) and persisted it in disk (96 & 97)
DN1 persisted it in disk (71 & 72)

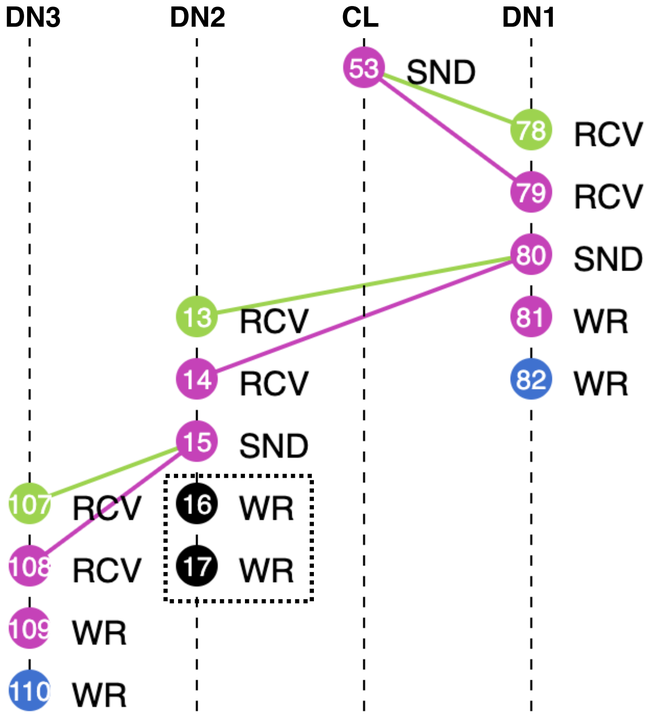
CaT Framework In Action

Storage and replication of a file in HDFS



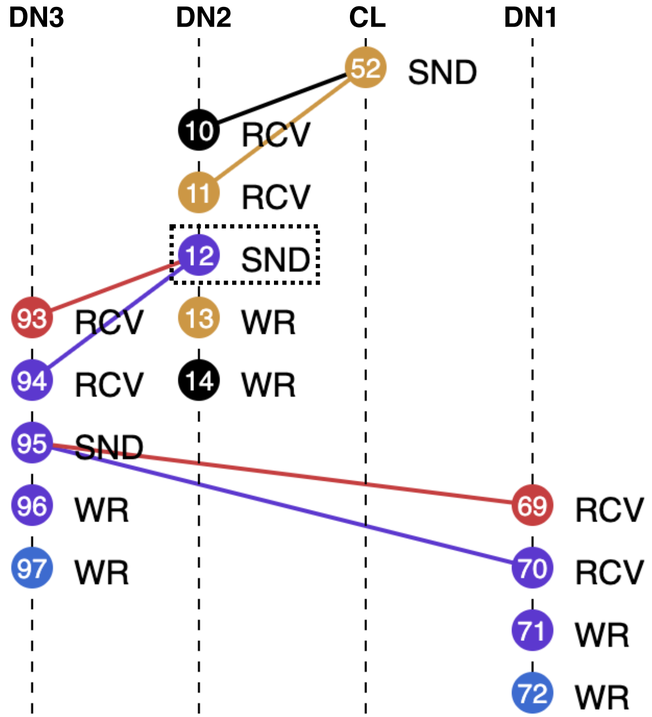
a) Normal execution

Client sent the file to DN3 (81)
DN3 sent it to DN1 (17) and persisted it in disk (18 & 19)
DN1 sent it to DN2 (106) and persisted it in disk (107 & 108)
DN2 persisted it in disk (47 & 48)



b) Storage corruption

Client sent the file to DN1 (53)
DN1 sent it to DN2 (80) and persisted it in disk (81 & 82)
DN2 sent it to DN3 (15) and persisted it in disk (16 & 17)
DN3 persisted it in disk (109 & 110)



c) Network corruption

Client sent the file to DN2 (52)
DN2 sent it to DN3 (12) and persisted it in disk (13 & 14)
DN3 sent it to DN1 (95) and persisted it in disk (96 & 97)
DN1 persisted it in disk (71 & 72)

Conclusion

- A novel framework for collecting and analyzing I/O requests of distributed systems
  - Open-source prototype: https://github.com/dsrhaslab/cat
- Content-aware tracing and analysis strategy that correlates the context and content of requests to better understand the data flow of systems
- Depending on the target workload, it is possible to capture most of the I/O requests while incurring negligible performance overhead
- CAT's content-aware approach can improve the analysis of distributed systems by pinpointing their data flows and I/O access patterns

CaT
Content-aware Tracing and Analysis for Distributed Systems

in 22nd International Middleware Conference (Middleware '21)

CaT's prototype: https://github.com/dsrhaslab/cat

CaT's documentation: https://github.com/dsrhaslab/cat/wiki

Universidade do Minho

Acknowledgments: This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.