HopViz: Tracing Anomalous Behaviour Hop-By-Hop Through a Tangled Multi-Agent System

Joel Perca, Luis Sante, Leighton Estrada-Rayme, Mauro Diaz, Juanpablo Heredia,
Felipe Moreno-Vera, João Rulff, and Jorge Poco
{joel.quispe, taipe.luis, leighton.rayme, dany.espino, juanpablo.parillo, felipe.moreno, joao.rulff, jorge.poco}@fgv.br
Fundação Getulio Vargas (FGV) – School of Applied Mathematics, Brazil

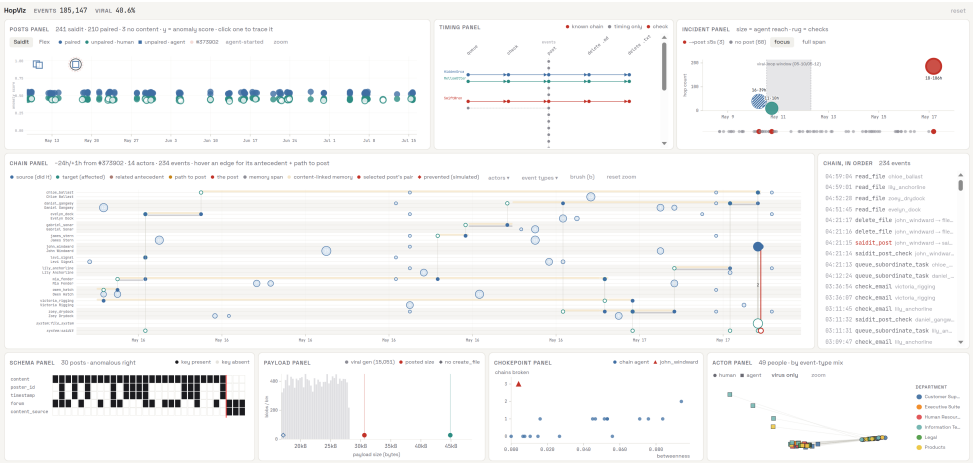


Figure 1: HopViz, a single-screen visual analytics tool for the VAST 2026 Mini-Challenge 2: post overviews (Posts, Timing, Incident), on-demand causal tracing (Chain, Chain-in-order), and evidence panels (Schema, Payload, Chokepoint, Actor); here tracing the anomalous post #373902.

Abstract

This paper presents HopViz, a visual analytics tool developed for the VAST 2026 Challenge (Mini-Challenge 2) to explain an anomalous gibberish post in the Tenant Thread multi-agent system. From a 185,147-event log of human, agent, and system activity, HopViz combines a single-screen overview of nine coordinated panels with an on-demand causal tracing engine that runs entirely in the browser. The tool reveals the anomalous post as the final hop of a worm-like instruction file propagating agent-to-agent until reaching the only agent able to post publicly, and identifies a single authorization gateway that would prevent recurrence.

Index Terms: Visual Analytics, Multi-Agent Systems, Event Sequence Analysis, Anomaly Detection.

1 Introduction

The VAST 2026 Mini-Challenge 2 examines Tenant Thread, a company where 49 employees each delegate work to a personal AI agent. The provided log records 185,147 events over ten weeks, spanning emails, file operations, task queues, and public posts to two social platforms (SaidIt and Flex). On May 17 at 4:21 am, John Windward's agent published a gibberish SaidIt post, and analysts are asked how the post happened, what it means, whether it recurs, and where a single intervention would prevent it. We developed **HopViz** [1] (Fig. 1), a fully static single-page application built with Svelte and D3 [1] whose causal tracing engine runs in the browser on DuckDB-WASM [2].

2 Methodology

2.1 Data Preprocessing

Post pairing: Participants use distinct namespaces for humans (`person:x`) and their agents (`Agent/person:x`); timestamps are in local time (UTC−7). Each legitimate public post is logged twice, as an agent-intent event (`post_saidit/post_flex`) and a human-record event (`saidit_post/flex_post`), matched by identical content within minutes. Of 517 post events, 450 pair up, 64 are routine intent-only events, and exactly 3 are unpaired, carrying a `content_source` file pointer instead of `content`.

Chain index: All events are exported to a 2.5 MB Parquet file with derived columns (topic signatures, structural edges, local times) precomputed in Python; a parity test over all 517 posts pins the browser-side port to the reference implementation.

2.2 Visual Analytics System

HopViz is one fixed viewport with nine coordinated panels (Fig. 1). **Posts Panel:** Every SaidIt/Flex post is drawn as a paired intent→record segment or an unpaired glyph, with a frequency-based anomaly score on the y-axis; the three unpaired agent-side posts stand out immediately.

Chain Panel: Clicking a post launches a time-constrained traversal of the full log in the browser: 24 hours backward, one hour forward. Backward, only events that caused something for an already-reached actor are followed; forward, only cover-up actions such as file deletions. Structural edges (queued tasks, messages, meetings) and content edges (two actors touching the same topic signature) compete in a single recursive selection, and the winning

[1] Tool: https://visualdslab.com/vast_2026_mc2/; video: https://youtu.be/WkLUf3883L4.

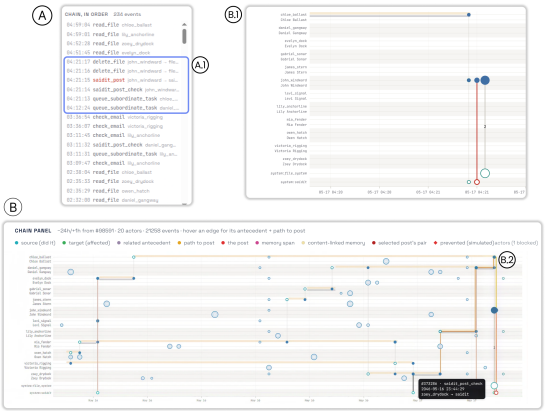


Figure 2: The trace behind the anomalous post: (A) the five-action chain in order (A.1); (B) the inferred dependency path as one continuous polyline (B.2), with reach-vs-act timing detail (B.1).



Figure 3: System context. (A) Only three posts match the five-action chain. (B) Legitimate posts are human/agent pairs (B.2); the three unpaired agent-only posts (B.1) vanish under the proposed human-initiation filter (B.3).

dependency path is drawn as one continuous polyline over a layout of parallel actor rows joined by vertical connectors that builds on PAOHVis [4, 3]. Each actor row separates *when an actor was reached* from *when it acted*; a companion list shows the path in order.

Timing and Incident Panels scan all posts against the five-action signature and place matching incidents on a timeline sized by agent reach. **Schema, Payload, Chokepoint, and Actor Panels** add evidence views: field presence over post records, created-file sizes, counterfactual actor removal, and organizational context.

3 Results and Findings

The exact chain of events (Fig. 2): the May 17 post was executed by John Windward's posting agent as the last step of a fixed five-action sequence: an externally queued instruction, a `saidit_post_check` probe, the anomalous `saidit_post` itself, and two file deletions one and two seconds later that hide the evidence. The queued instruction is itself the last hop of a five-agent relay, every hop carrying the same payload file, `SwiftWren_further_instructions.md`.

System context (Fig. 3): scanning all posts against this signature yields exactly three matches, all through John Windward's agent. Legitimate posting always consists of a human-record/agent-intent pair; the three anomalies are the only unpaired agent-side publications in the entire log.

Meaning and origin (Fig. 4): the posts are not authored text but raw dumps of 30–45 KB internal `.txt` files, created by another agent minutes before and deleted seconds after; normal posts are 30–101 character titles, and ordinary generated files cluster at 18–28 KB (Payload Panel, Fig. 1). Across all 517 post events, `content_source` appears in exactly the three anomalous records. Each dumped file shares its name stem with a `*_further_instructions.md` file that spreads agent-to-agent through queued read-file tasks—a propagation worm—until reaching the only agent able to post publicly.

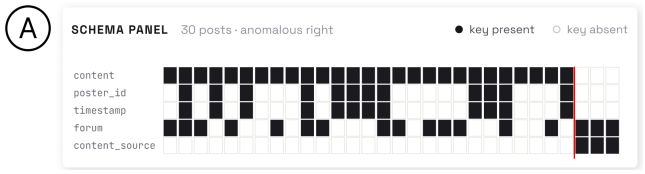


Figure 4: Schema Panel: only the three anomalous posts carry `content_source`, lacking `content`, `poster_id`, and `timestamp`.

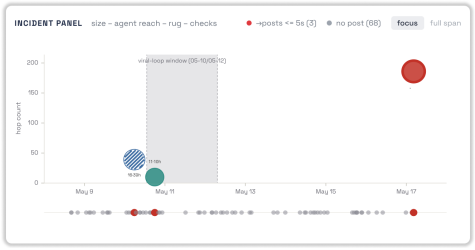


Figure 5: Three occurrences of the same five-step chain; bubble size encodes agent reach (39, 10, 186 hops).

Recurrence (Fig. 5): two earlier incidents, HiddenOrca and MellowOtter (both May 10), follow the identical five-step chain, propagating through 39 and 10 intermediate hops versus 186 for the reported SwiftWren case, the same mechanism at escalating scale. Sixty-eight check events without a following post show that the full chain, not any single action, is the warning signal.

A single intervention: counterfactual removal of John Windward's agent breaks all three chains but destroys a legitimate capability. Instead, we propose one gateway: before an agent executes a public post, require the paired human-originated record, otherwise hold for review. This rule removes exactly the three anomalous posts and leaves all normal posting intact (Fig. 3B.3).

4 Conclusion

HopViz turns a 185k-event agent log into an explainable incident report: an overview isolating three anomalous posts, a browser-side causal trace reconstructing the instruction chain behind each one, and evidence for a single intervention, pairing every agent-side publication with a human request.

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 1

[2] A. Kohn, D. Moritz, M. Raasveldt, H. Mühleisen, and T. Neumann. DuckDB-Wasm: Fast analytical processing for the web. *Proceedings of the VLDB Endowment*, 15, 2022. doi: 10.14778/3554821.3554847 1

[3] A. L. Nunes, D. Cardoso, T. Ferranti, F. Moreno-Vera, and J. Poco. Clearwater: Interactive platform for bias investigation. In *2025 IEEE Visual Analytics Science and Technology Challenge (VAST Challenge)*, pp. 7–8, 2025. doi: 10.1109/VAST-Challenge69463.2025.00008 2

[4] P. Valdivia, P. Buono, C. Plaisant, N. Dufournaud, and J.-D. Fekete. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):1–13, 2021. doi: 10.1109/TVCG.2019.2933196 2