FAuST: Striking a Bargain between Forensic Auditing's Security and Throughput

Muhammad Adil Inam, Akul Goyal, **Jason Liu**, Jaron Mink, Noor Michael, Sneha Gaur, Adam Bates, Wajih Ul Hassan

Annual Computer Security Applications Conference (ACSAC) 2022
December 9, 2022

Logs are Critical but Expensive

- High-profile APT attacks last for years [1]

[1] https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/
[2] https://www.logsign.com/blog/how-long-should-security-logs-be-kept/
[3] https://www.auditboard.com/blog/security-log-retention-best-practices/
[4] Shiqing Ma et al., USENIX ATC '18
[5] Kyu Hyung Lee et al., CCS '13
[6] https://www.techtarget.com/searchsecurity/feature/Splunk-Enterprise-Security-Product-overview

Logs are Critical but Expensive

- High-profile APT attacks last for years [1]

[1] https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/
[2] https://www.logsign.com/blog/how-long-should-security-logs-be-kept/
[3] https://www.auditboard.com/blog/security-log-retention-best-practices/
[4] Shiqing Ma et al., USENIX ATC '18
[5] Kyu Hyung Lee et al., CCS '13
[6] https://www.techtarget.com/searchsecurity/feature/Splunk-Enterprise-Security-Product-overview

Logs are Critical but Expensive

- High-profile APT attacks last for years [1]
- Most organizations store logs for a few months [2,3]

[1] https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/
[2] https://www.logsign.com/blog/how-long-should-security-logs-be-kept/
[3] https://www.auditboard.com/blog/security-log-retention-best-practices/
[4] Shiqing Ma et al., USENIX ATC '18
[5] Kyu Hyung Lee et al., CCS '13
[6] https://www.techtarget.com/searchsecurity/feature/Splunk-Enterprise-Security-Product-overview

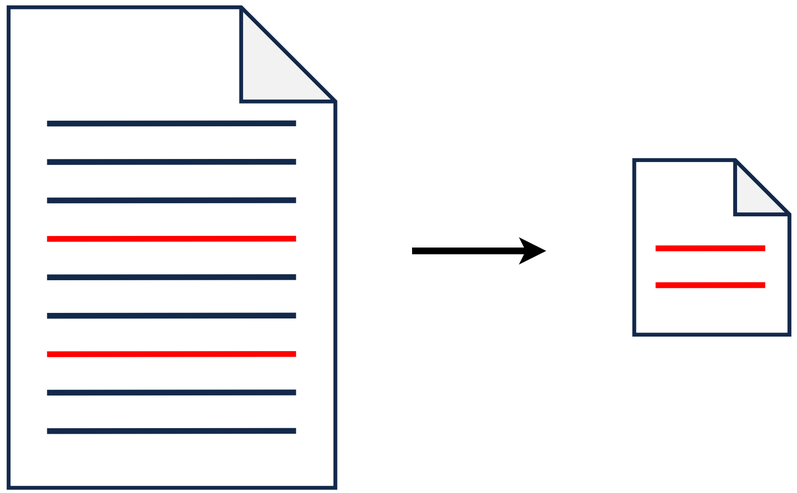
Logs are Critical but Expensive

- High-profile APT attacks last for years [1]
- Most organizations store logs for a few months [2,3]
- Each machine can generate 400-1200 GB per year [4,5]
- Log analysis often costs at least $1500 per GB [6]

[1] https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/
[2] https://www.logsign.com/blog/how-long-should-security-logs-be-kept/
[3] https://www.auditboard.com/blog/security-log-retention-best-practices/
[4] Shiqing Ma et al., USENIX ATC '18
[5] Kyu Hyung Lee et al., CCS '13
[6] https://www.techtarget.com/searchsecurity/feature/Splunk-Enterprise-Security-Product-overview

Keeping only Important Logs

- Data compression only gets so far [1]
  - Investigation tools need to be able to search for key events
- Researchers instead leverage data provenance to reduce logs



[1] https://www.elastic.co/blog/filebeat-modiles-access-logs-and-elasticsearch-storage-requirements

Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log  Provenance Graph

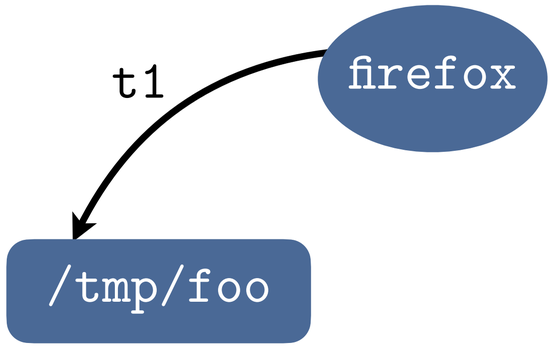
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
```

Provenance Graph



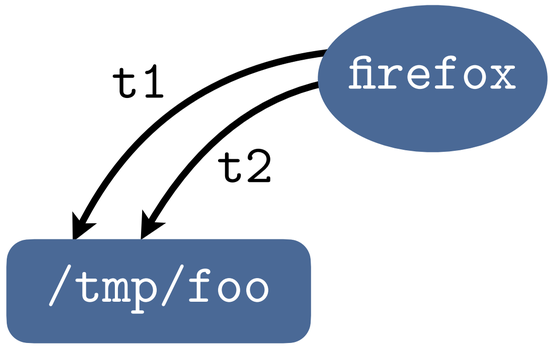
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
```

Provenance Graph



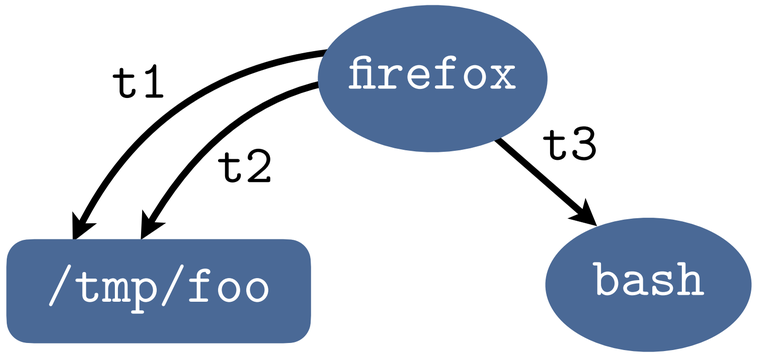
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
```

Provenance Graph



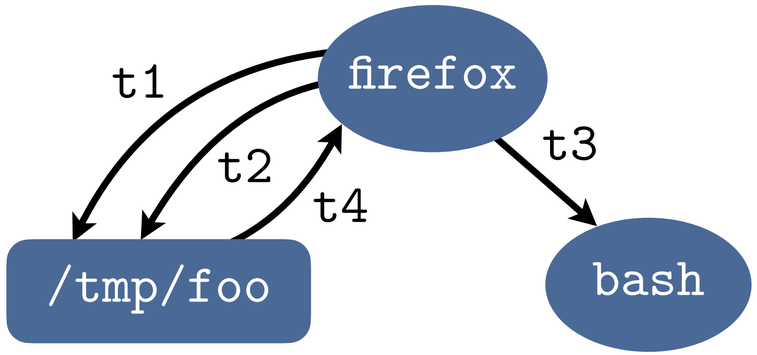
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
```

Provenance Graph



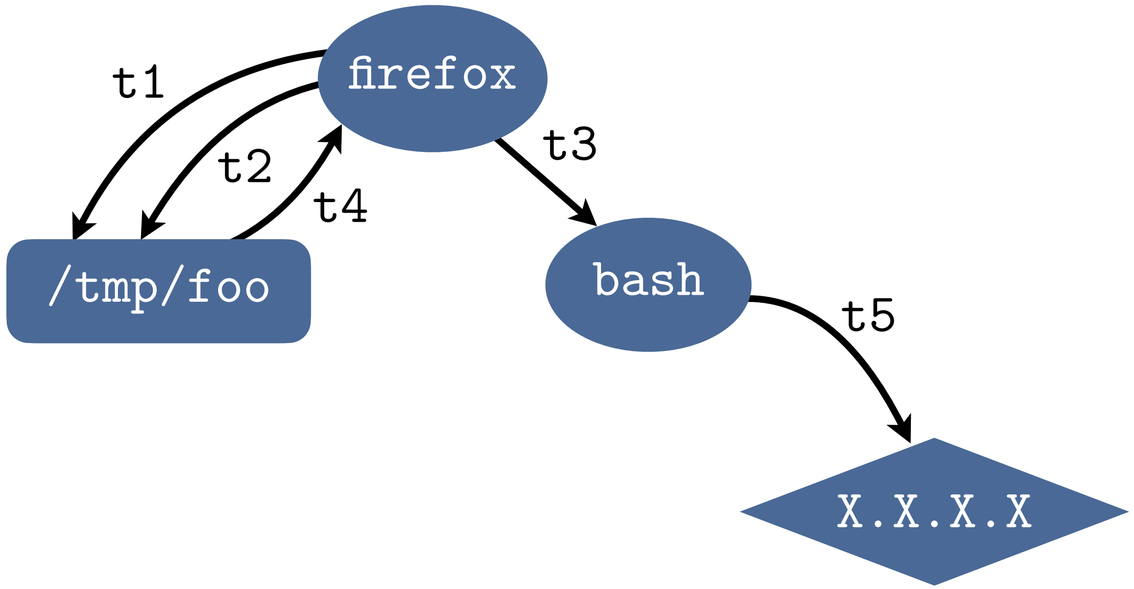
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
```

Provenance Graph



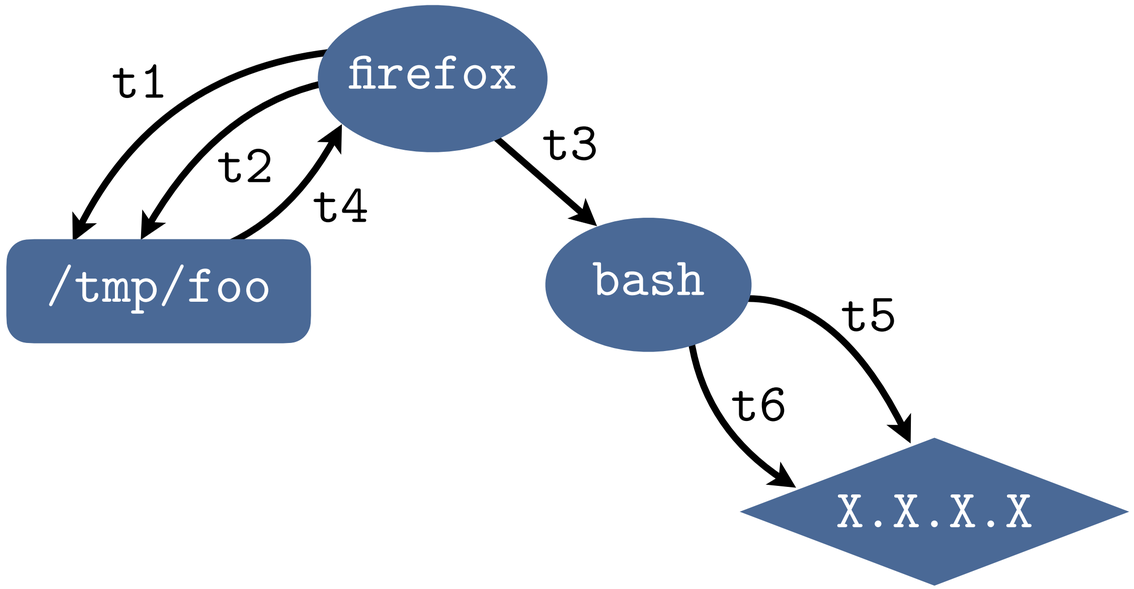
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
```

Provenance Graph



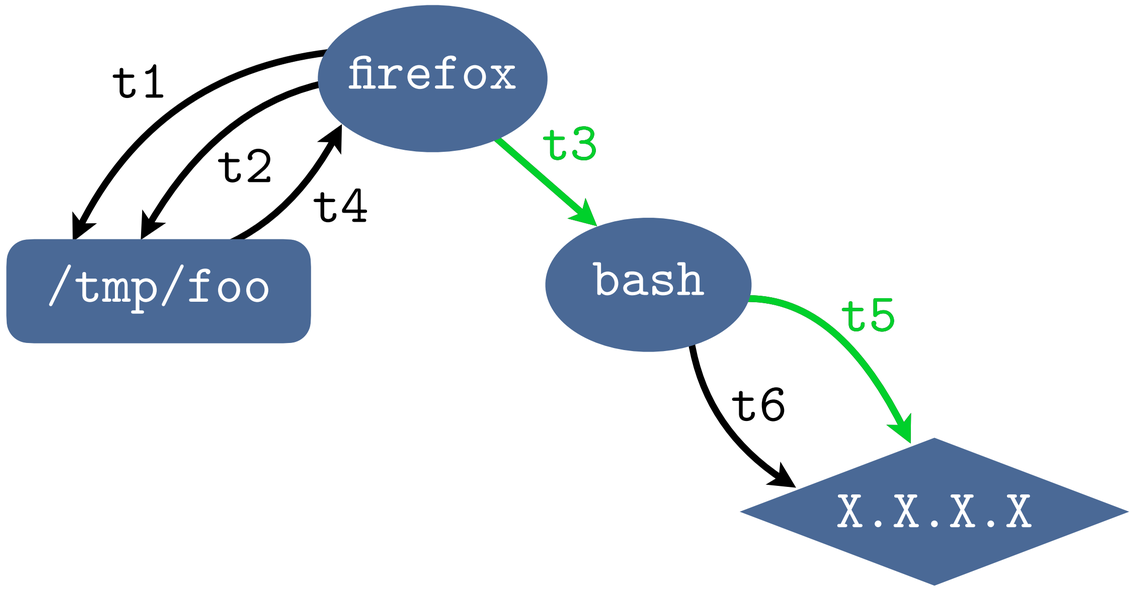
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
```

Provenance Graph



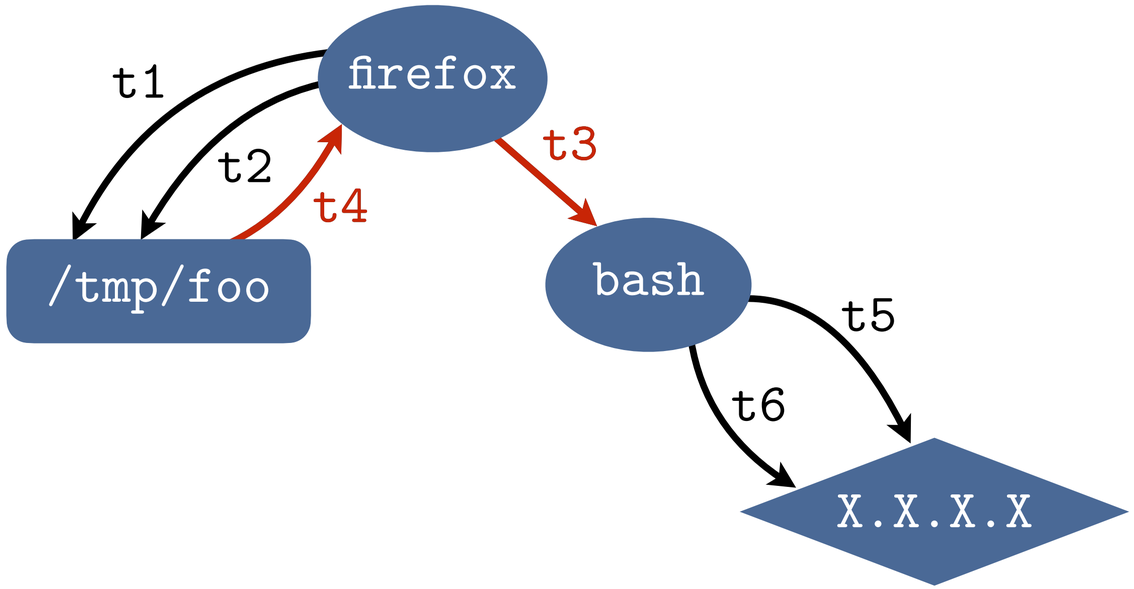
Background: Data Provenance

- Provenance graphs allow us to reason about causality relationships

Audit Log

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
```

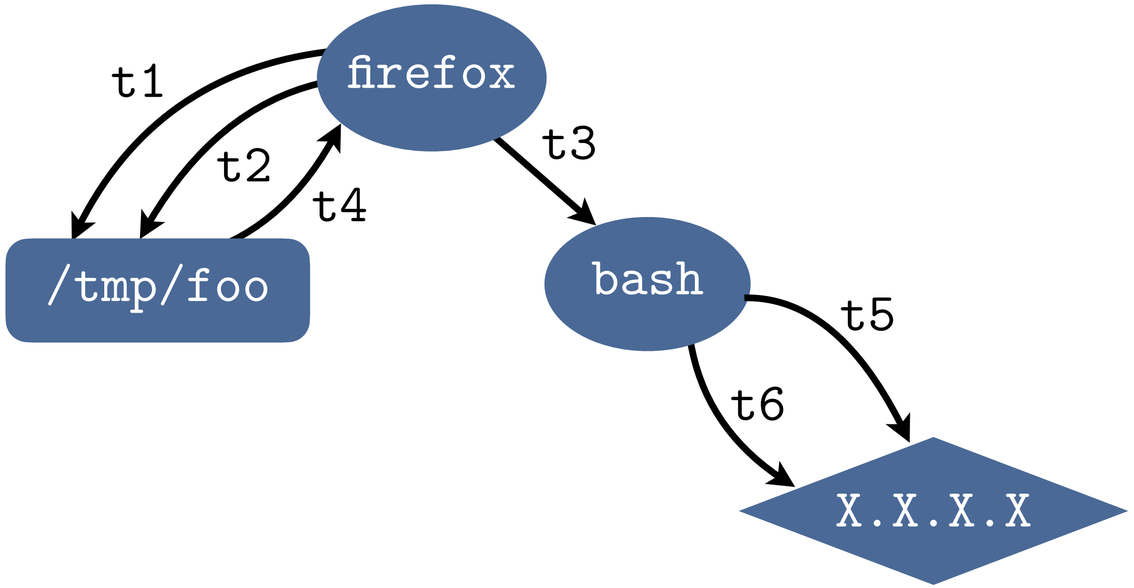
Provenance Graph



Reduction Techniques

- LogGC: remove temporary file I/O that cannot affect other parts of the graph [1]
  - Claimed ~93-97% reduction
- CPR: remove parallel edges that do not add any new causal information [2]
  - Claimed ~56% reduction, and that it can be combined with LogGC

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
```

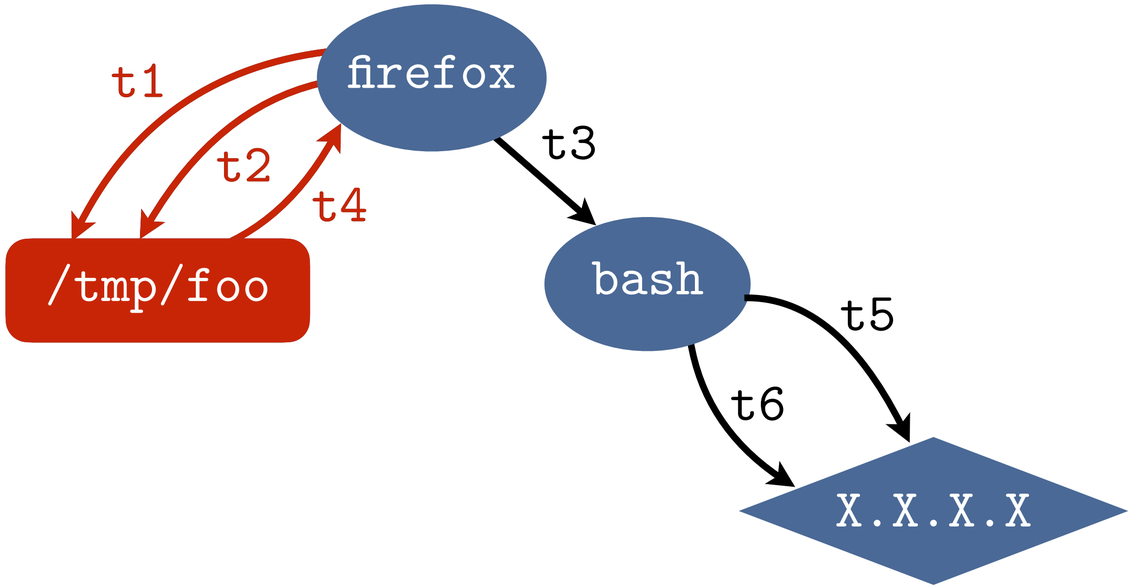


[1] Kyu Hyung Lee et al., CCS '13
[2] Zhang Xu et al., CCS '16

Reduction Techniques

- LogGC: remove temporary file I/O that cannot affect other parts of the graph [1]
  - Claimed ~93-97% reduction
- CPR: remove parallel edges that do not add any new causal information [2]
  - Claimed ~56% reduction, and that it can be combined with LogGC

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
t7, close, firefox, /tmp/foo
t8, unlink, firefox, /tmp/foo
```

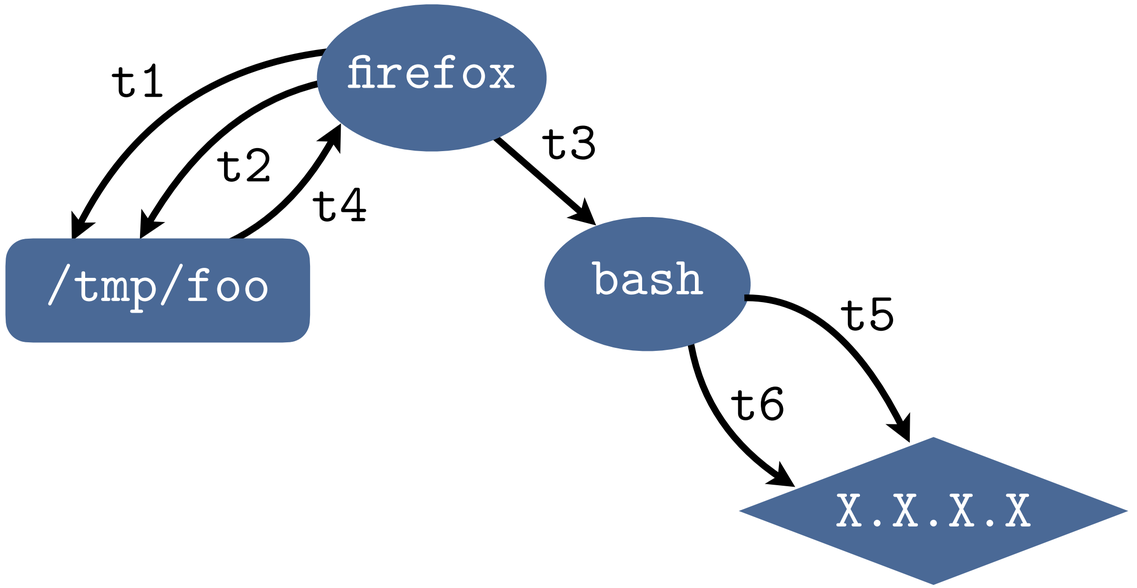


[1] Kyu Hyung Lee et al., CCS '13
[2] Zhang Xu et al., CCS '16

Reduction Techniques

- LogGC: remove temporary file I/O that cannot affect other parts of the graph [1]
  - Claimed ~93-97% reduction
- CPR: remove parallel edges that do not add any new causal information [2]
  - Claimed ~56% reduction, and that it can be combined with LogGC

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
t7, close, firefox, /tmp/foo
t8, unlink, firefox, /tmp/foo
```

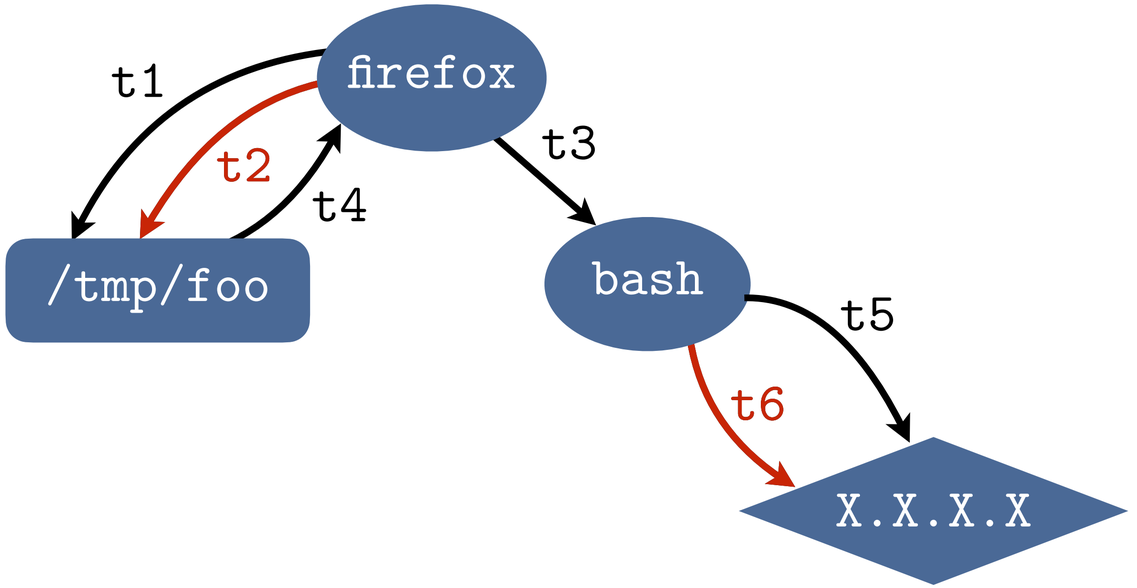


[1] Kyu Hyung Lee et al., CCS '13
[2] Zhang Xu et al., CCS '16

Reduction Techniques

- LogGC: remove temporary file I/O that cannot affect other parts of the graph [1]
  - Claimed ~93-97% reduction
- CPR: remove parallel edges that do not add any new causal information [2]
  - Claimed ~56% reduction, and that it can be combined with LogGC

```
t1, open, firefox, /tmp/foo
t2, write, firefox, /tmp/foo
t3, clone, firefox, bash
t4, read, firefox, /tmp/foo
t5, socket, bash, X.X.X.X
t6, write, bash, X.X.X.X
t7, close, firefox, /tmp/foo
t8, unlink, firefox, /tmp/foo
```



[1] Kyu Hyung Lee et al., CCS '13
[2] Zhang Xu et al., CCS '16

Comparing Reduction Techniques

- To what degree are different techniques actually orthogonal?

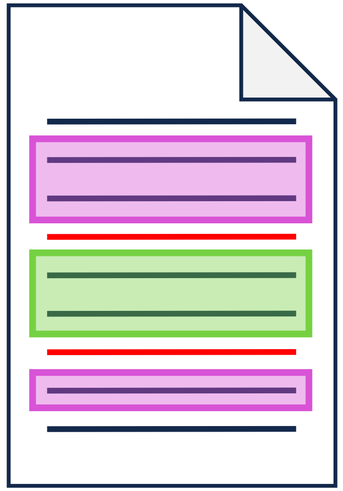
- Particular log datasets may be better suited to certain techniques
  - Is ~95% reduction for LogGC and ~56% for CPR a fair comparison?

Comparing Reduction Techniques

- To what degree are different techniques actually orthogonal?

LogGC

CPR



Low Redundancy

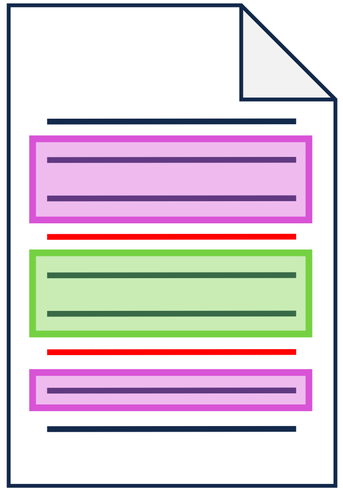
- Particular log datasets may be better suited to certain techniques
  - Is ~95% reduction for LogGC and ~56% for CPR a fair comparison?

Comparing Reduction Techniques

- To what degree are different techniques actually orthogonal?

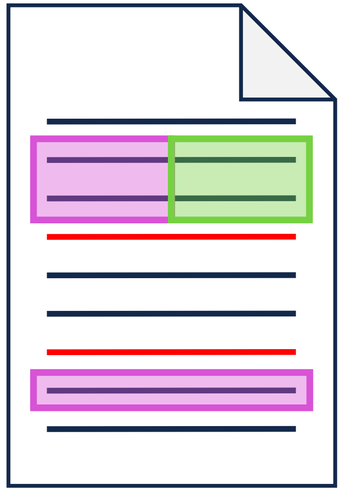
LogGC

CPR



Low Redundancy

vs.



High Redundancy

- Particular log datasets may be better suited to certain techniques
  - Is ~95% reduction for LogGC and ~56% for CPR a fair comparison?

FAuST:
Transparent & Modular Reduction

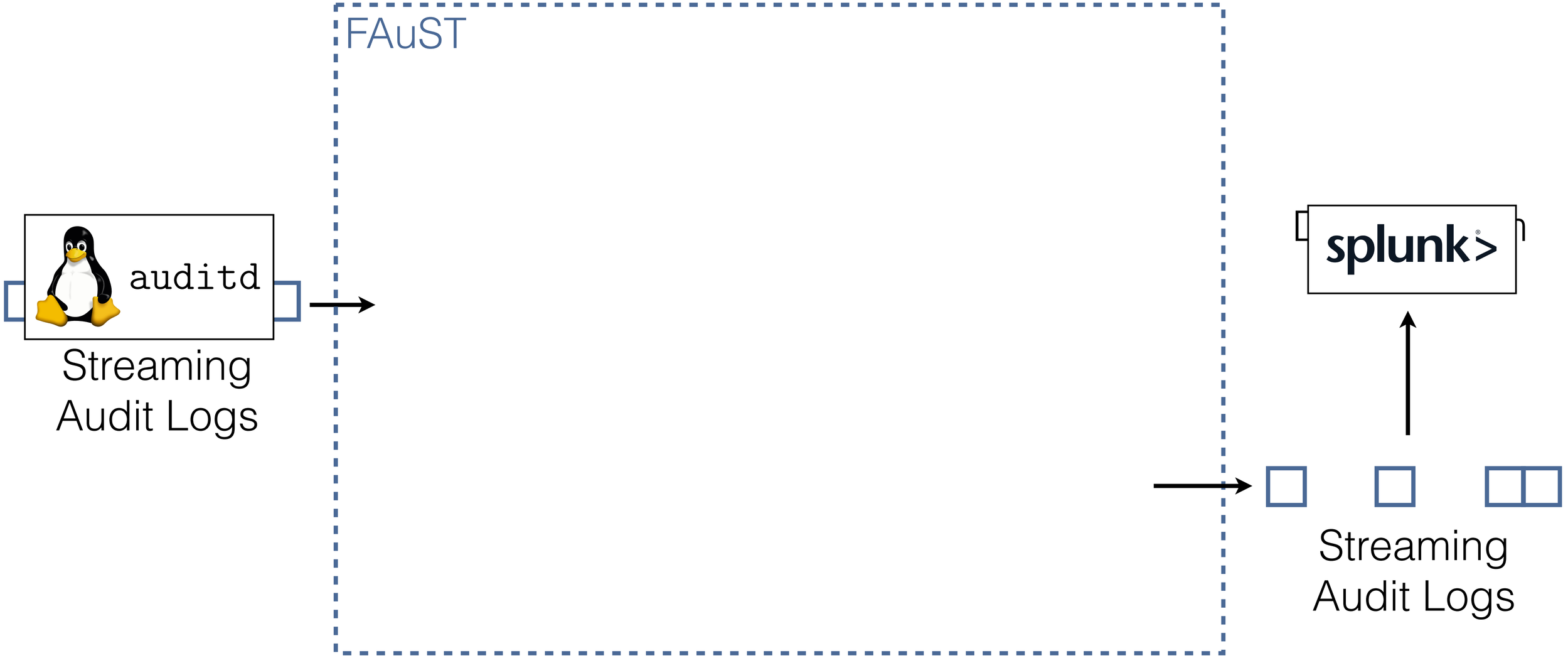
- Implement reduction techniques in an extensible modular framework
- Combine multiple reduction techniques simultaneously
- Evaluate and compare reduction performance and throughput for any combination

FAuST Design Overview

FAuST Design Overview

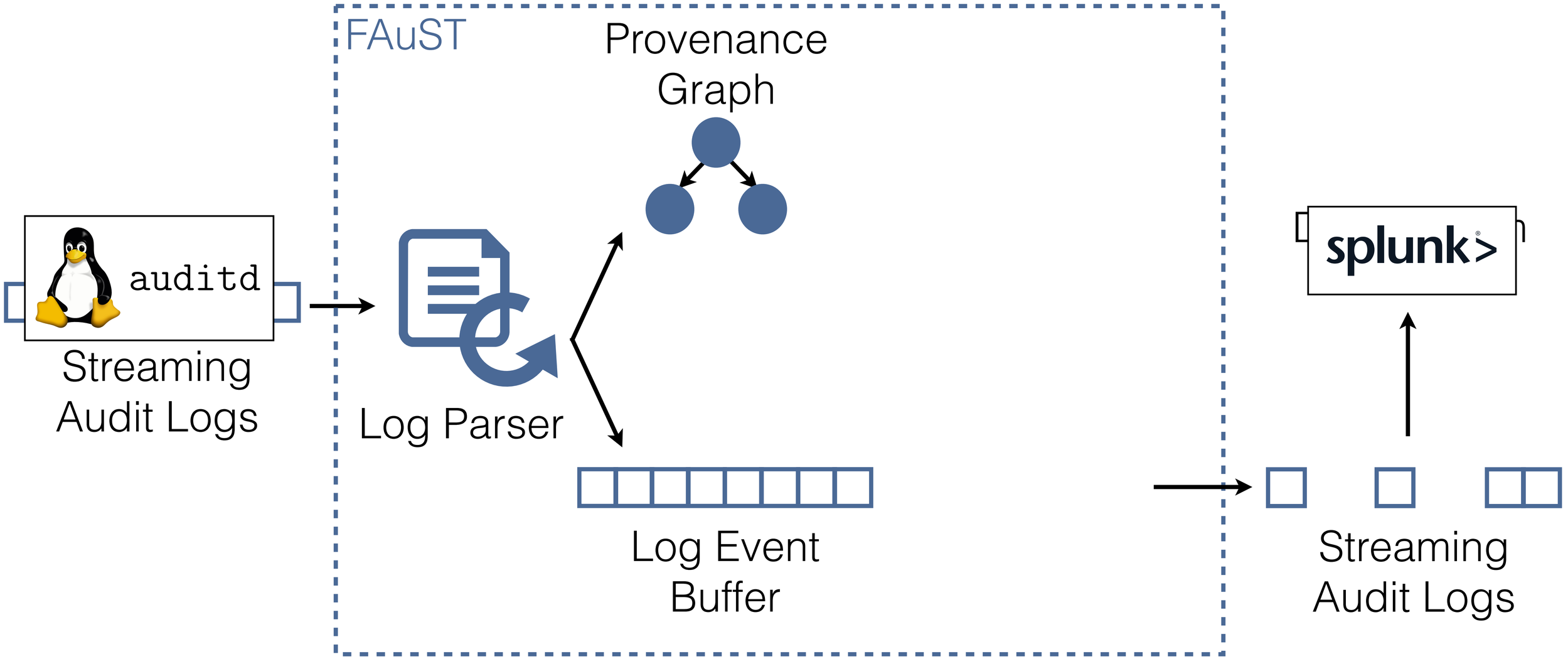
FAuST Design Overview

FAuST Design Overview

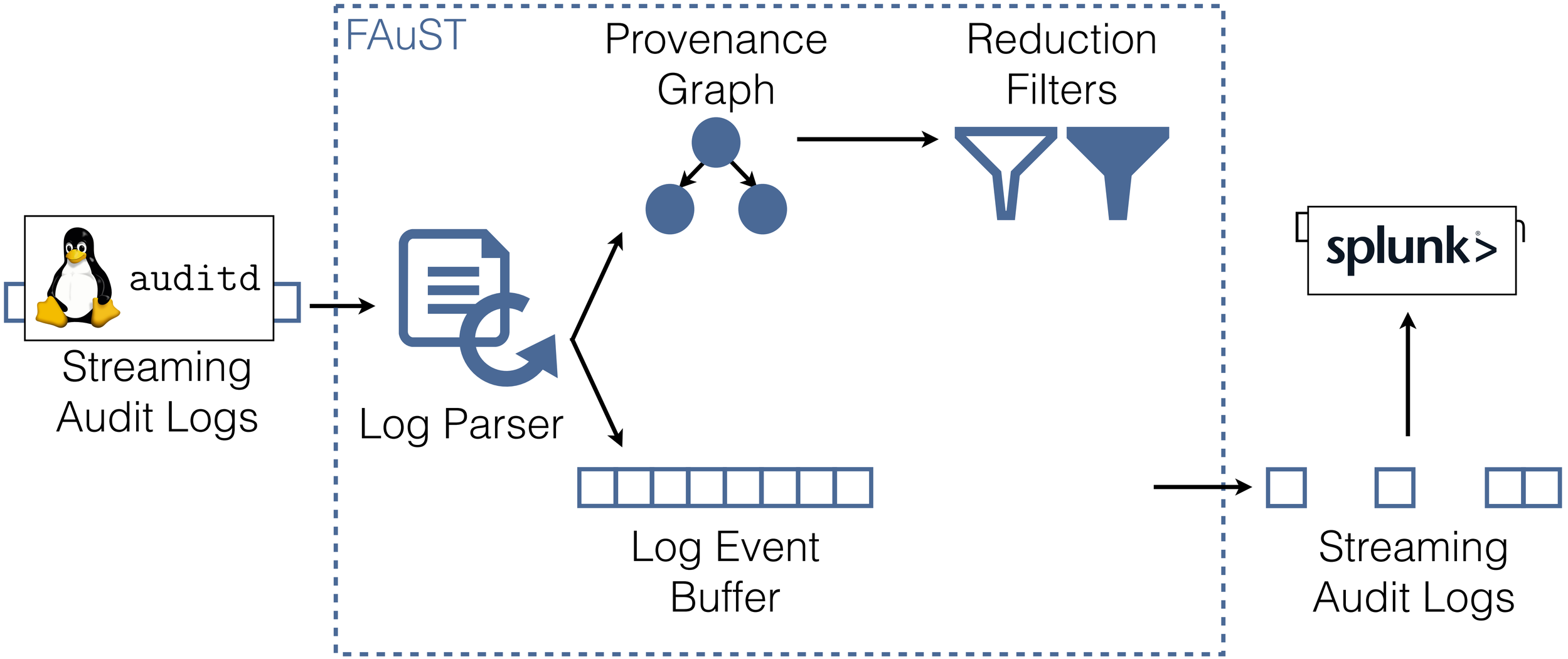


FAuST Design Overview

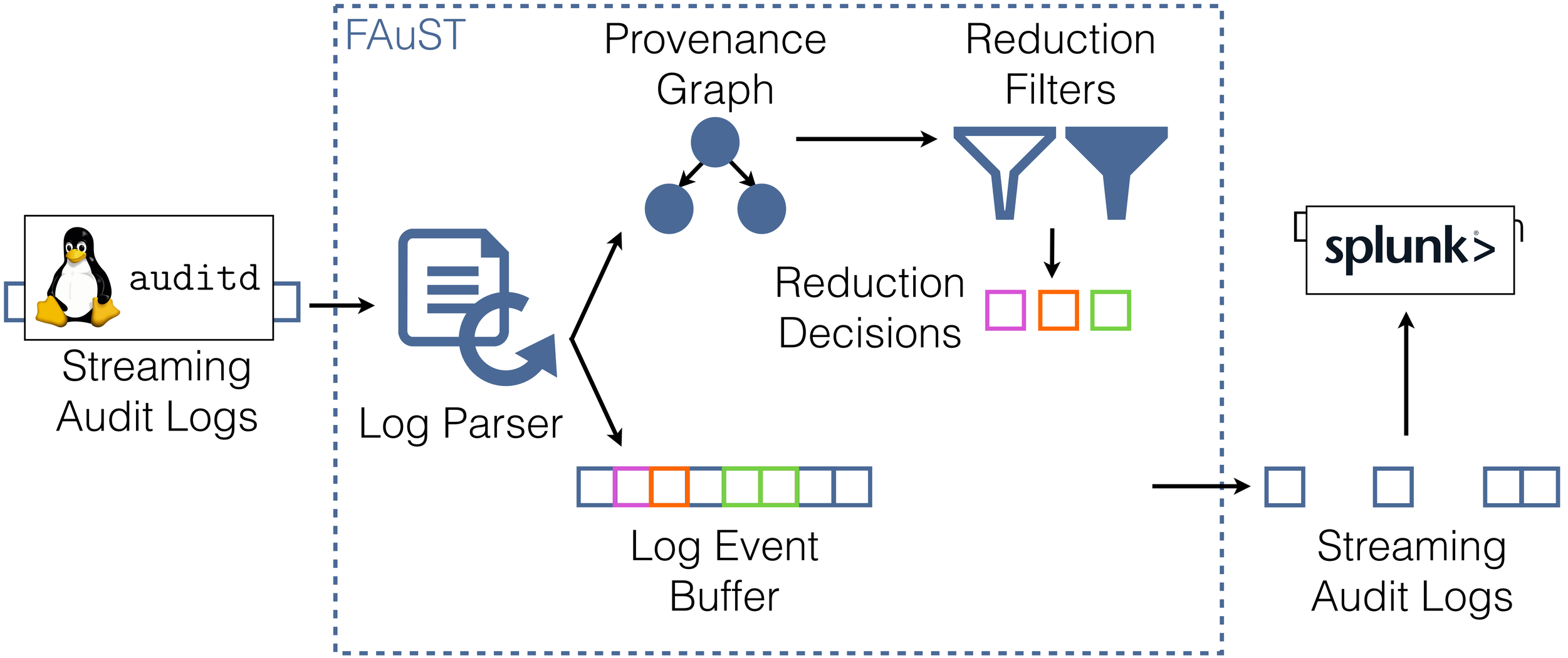
FAuST Design Overview



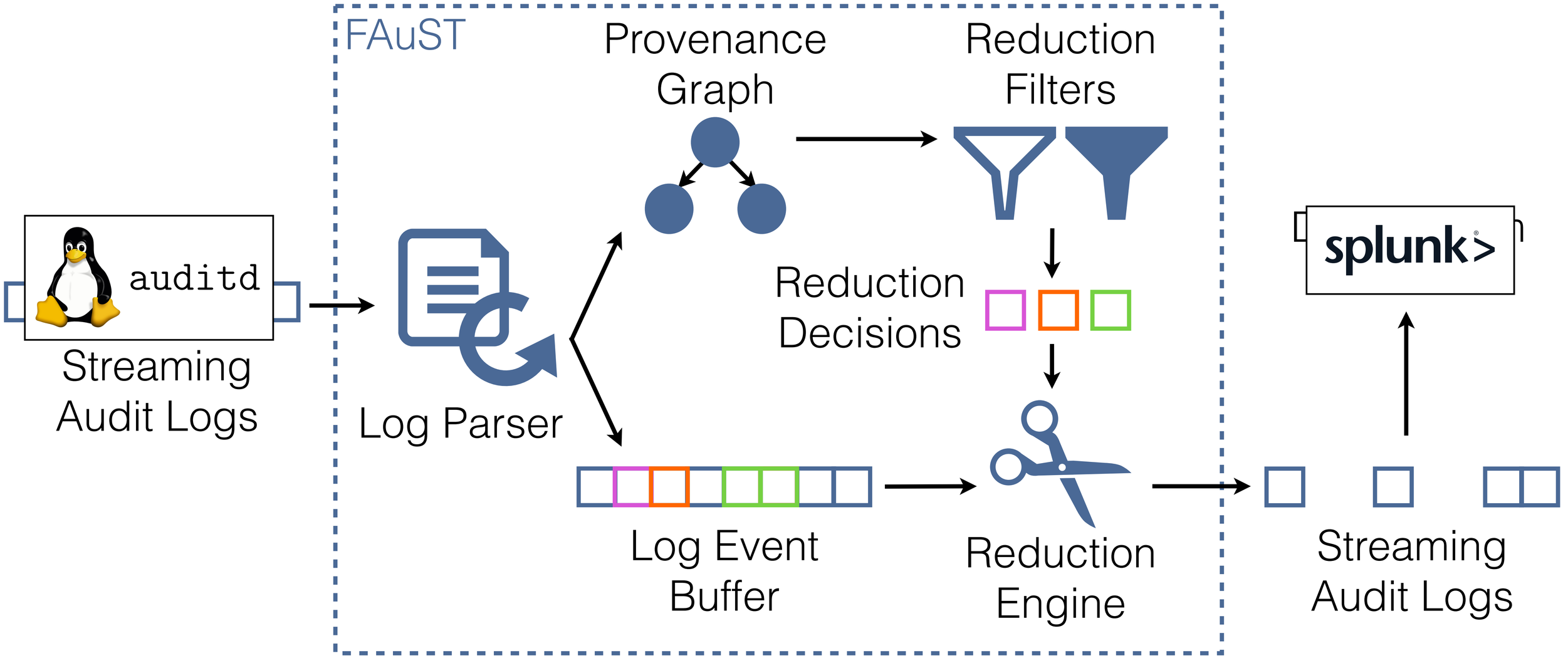
FAuST Design Overview



FAuST Design Overview

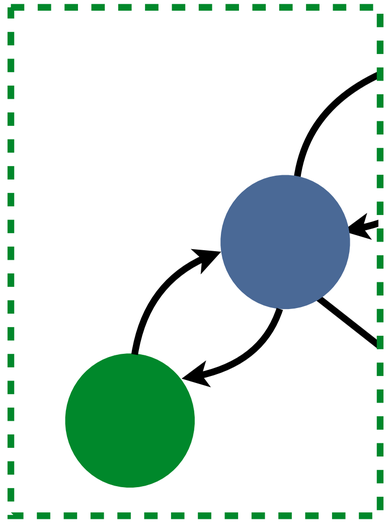


FAuST Design Overview



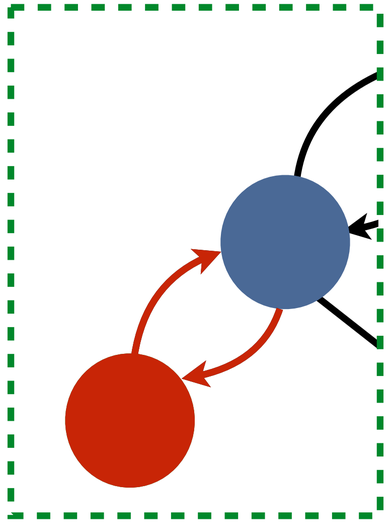
Local and Batch Filters

- Local techniques: analyze subgraphs in response to certain events
- Global techniques: analyze entire graph in offline setting



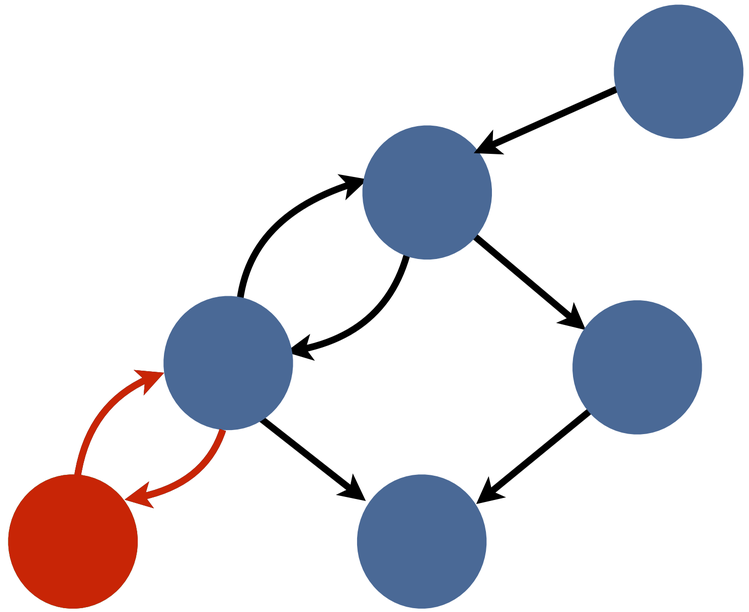
Local and Batch Filters

- Local techniques: analyze subgraphs in response to certain events
- Global techniques: analyze entire graph in offline setting



Local and Batch Filters

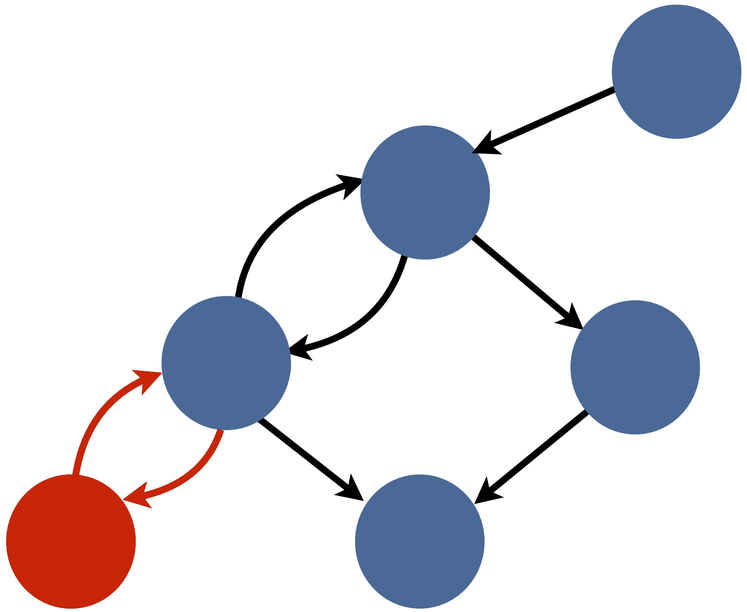
- Local techniques: analyze subgraphs in response to certain events
- Global techniques: analyze entire graph in offline setting



Local and Batch Filters

- Local techniques: analyze subgraphs in response to certain events
- Global techniques: analyze entire graph in offline setting

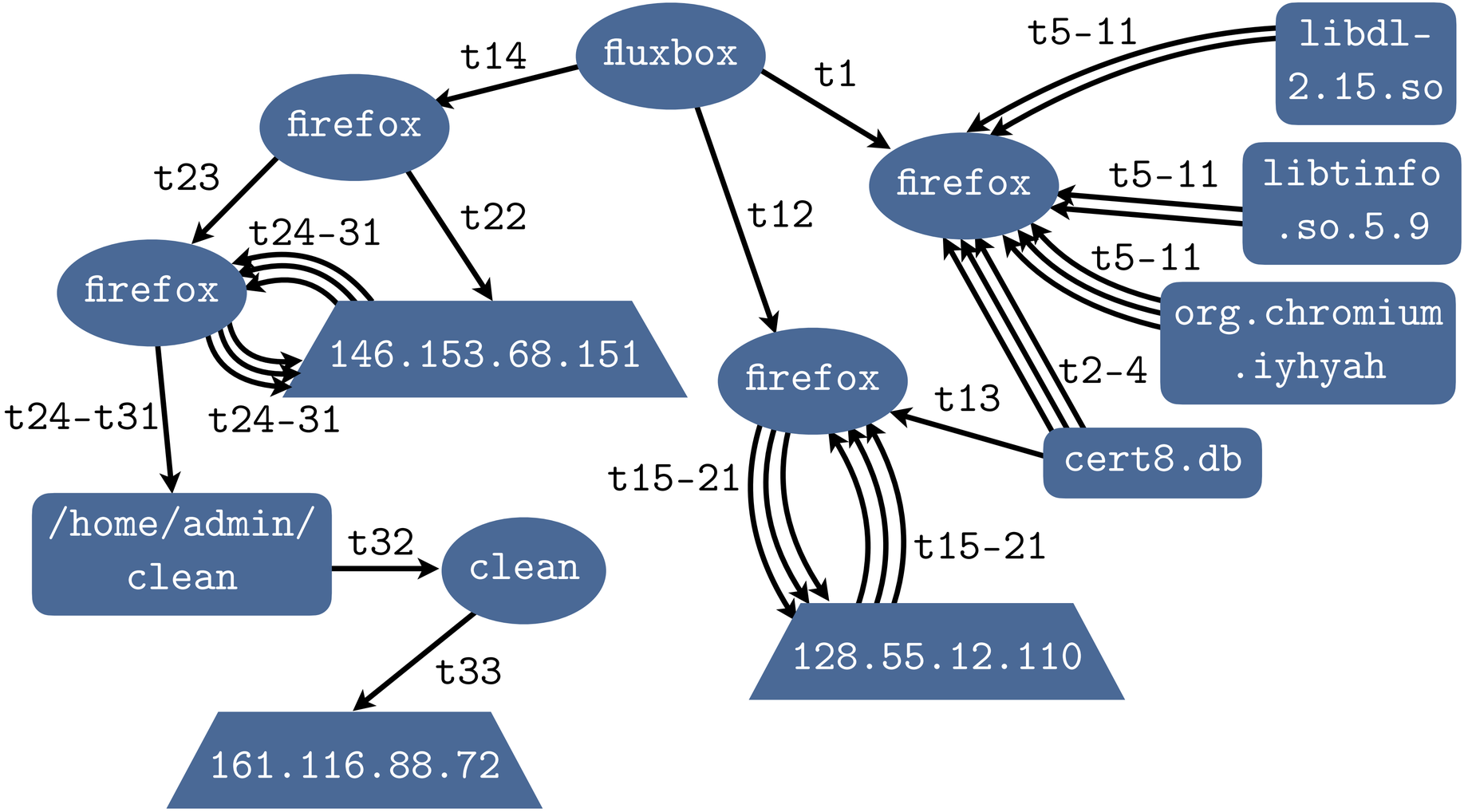
Local	Batch
LogGC Kyu Hyung Lee et al. CCS '13	NodeMerge Yutao Tang et al. CCS '18
CPR and PCAR Zhang Xu et al. CCS '16	F- and S-DPR Md Nahid Hossain et al. USENIX Security '18
	Winnower Wajih Ul Hassan et al. NDSS '18
	LogApprox Noor Michael et al. ACSAC '20



Reduction Filters: CPR

CPR

```
t1, clone, ...
t2, read, ...
...
t4, read, ...
t5, read, ...
...
t11, read, ...
t12, clone, ...
t13, read, ...
t14, clone, ...
t15, send, ...
...
t21, recv, ...
t22, connect, ...
t23, clone, ...
t24, send, ...
...
t30, recv, ...
t31, write, ...
t32, exec, ...
t33, connect, ...
```

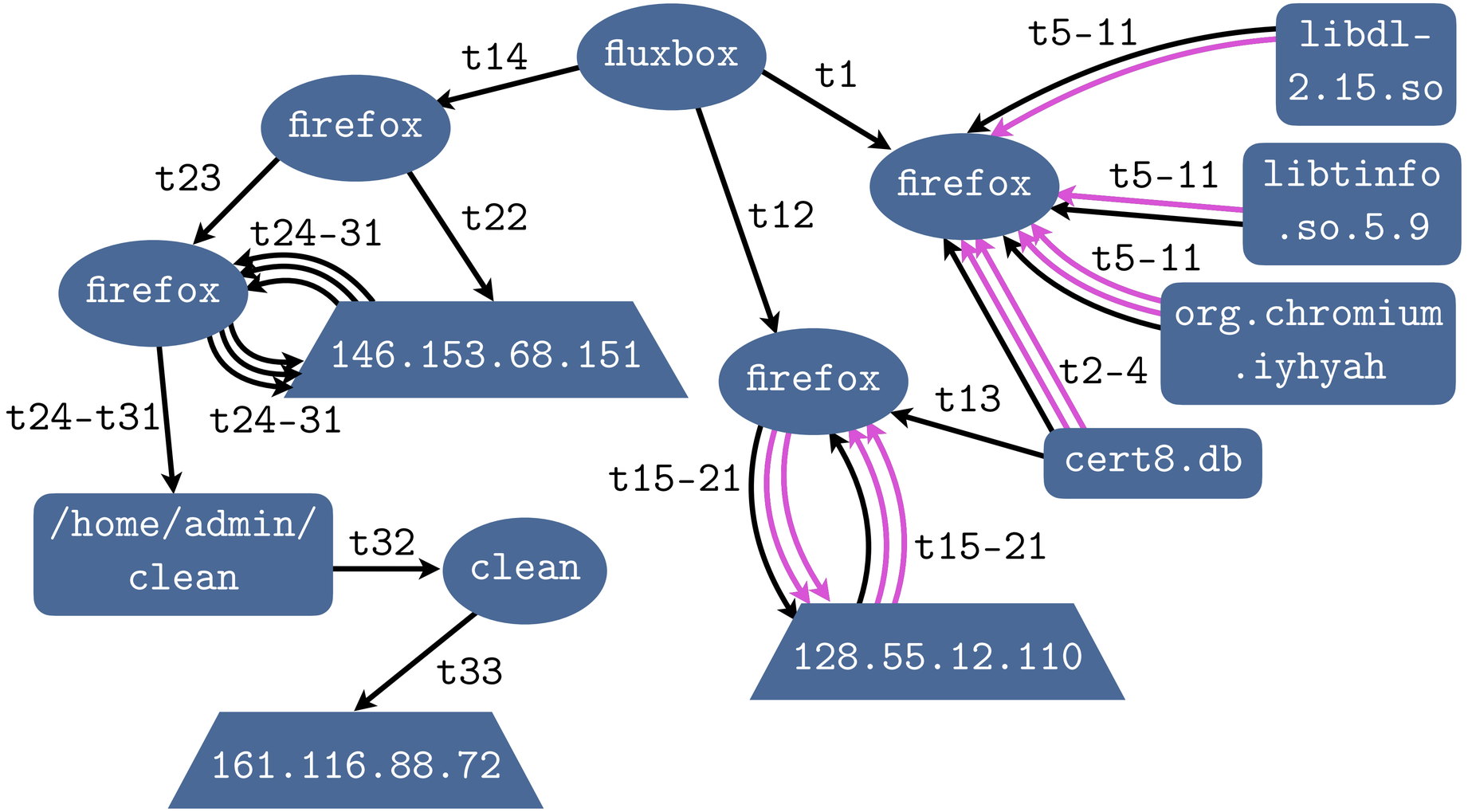


Zhang Xu et al., CCS '16

Reduction Filters: CPR

CPR

```
  t1, clone, ...
  t2, read, ...
* ...
* t4, read, ...
  t5, read, ...
* ...
* t11, read, ...
  t12, clone, ...
  t13, read, ...
  t14, clone, ...
  t15, send, ...
* ...
* t21, recv, ...
  t22, connect, ...
  t23, clone, ...
  t24, send, ...
  ...
  t30, recv, ...
  t31, write, ...
  t32, exec, ...
  t33, connect, ...
```

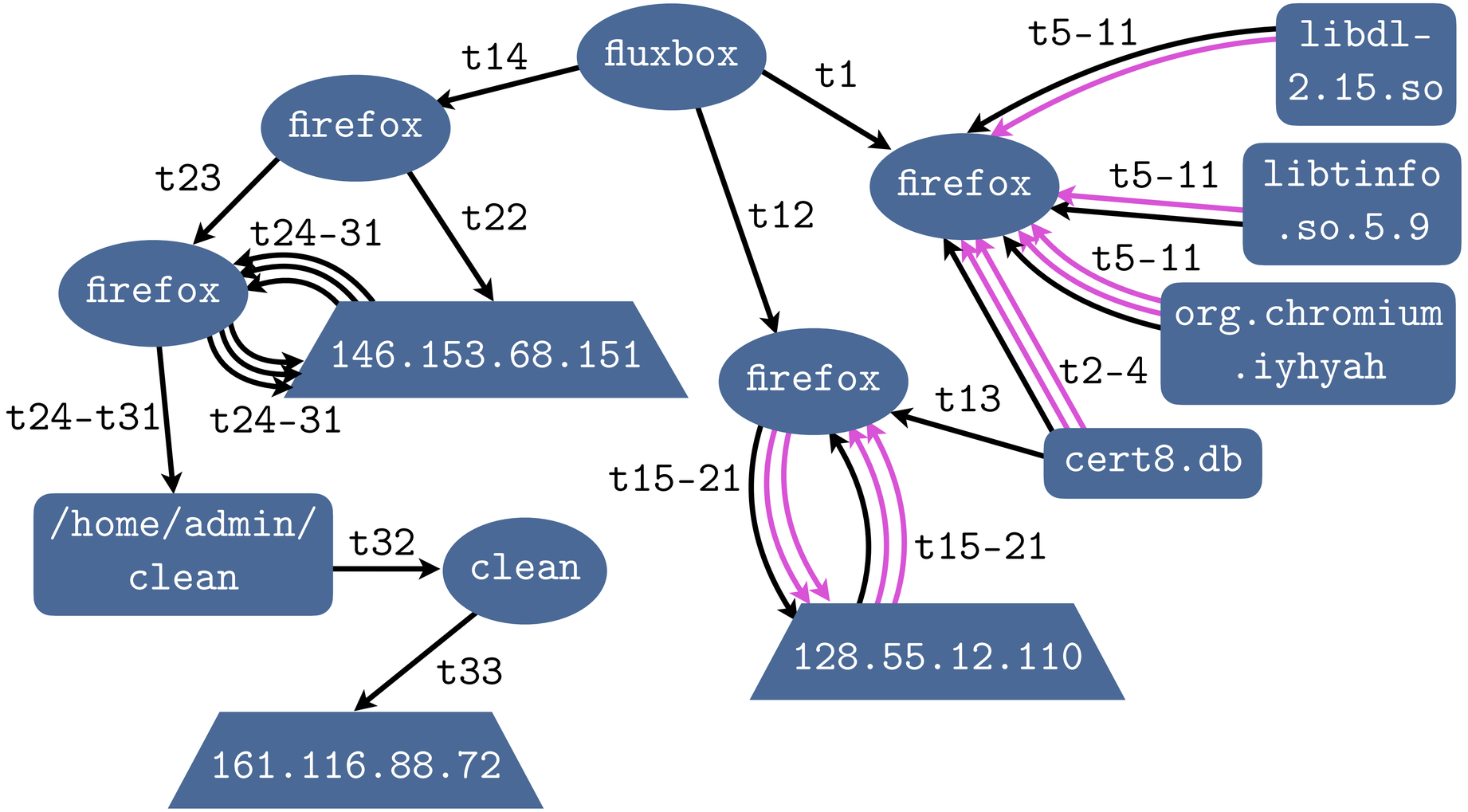


Zhang Xu et al., CCS '16

Reduction Filters: NodeMerge

CPR
NodeMerge

```
  t1, clone, ...
  t2, read, ...
* ...
* t4, read, ...
  t5, read, ...
* ...
* t11, read, ...
  t12, clone, ...
  t13, read, ...
  t14, clone, ...
  t15, send, ...
* ...
* t21, recv, ...
  t22, connect, ...
  t23, clone, ...
  t24, send, ...
  ...
  t30, recv, ...
  t31, write, ...
  t32, exec, ...
  t33, connect, ...
```

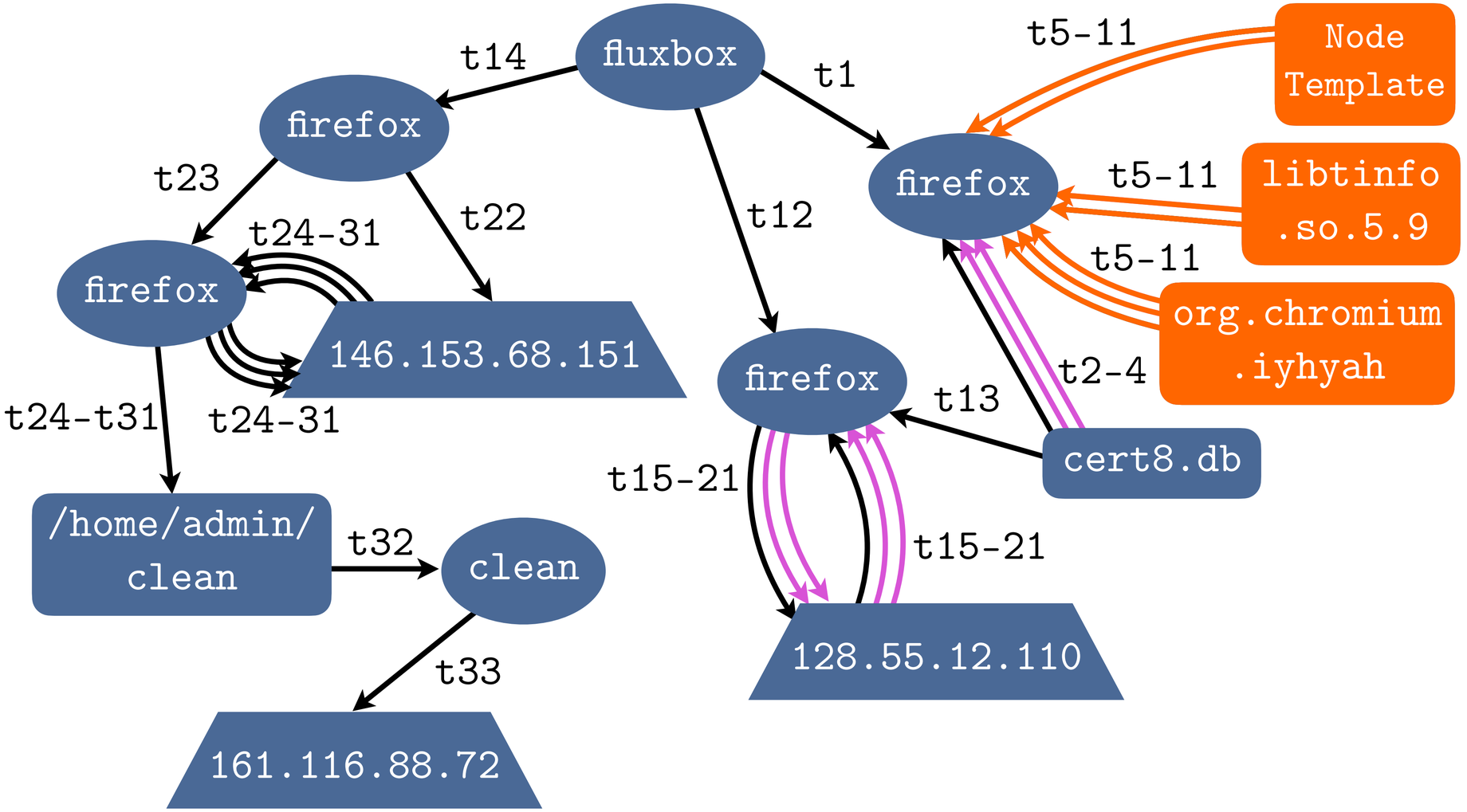


Yutao Tang et al., CCS '18

Reduction Filters: NodeMerge

CPR
NodeMerge

```
    t1, clone, ...
    t2, read, ...
  * ...
  * t4, read, ...
*   t5, read, ...
* * ...
* * t11, read, ...
    t12, clone, ...
    t13, read, ...
    t14, clone, ...
    t15, send, ...
  * ...
  * t21, recv, ...
    t22, connect, ...
    t23, clone, ...
    t24, send, ...
    ...
    t30, recv, ...
    t31, write, ...
    t32, exec, ...
    t33, connect, ...
```

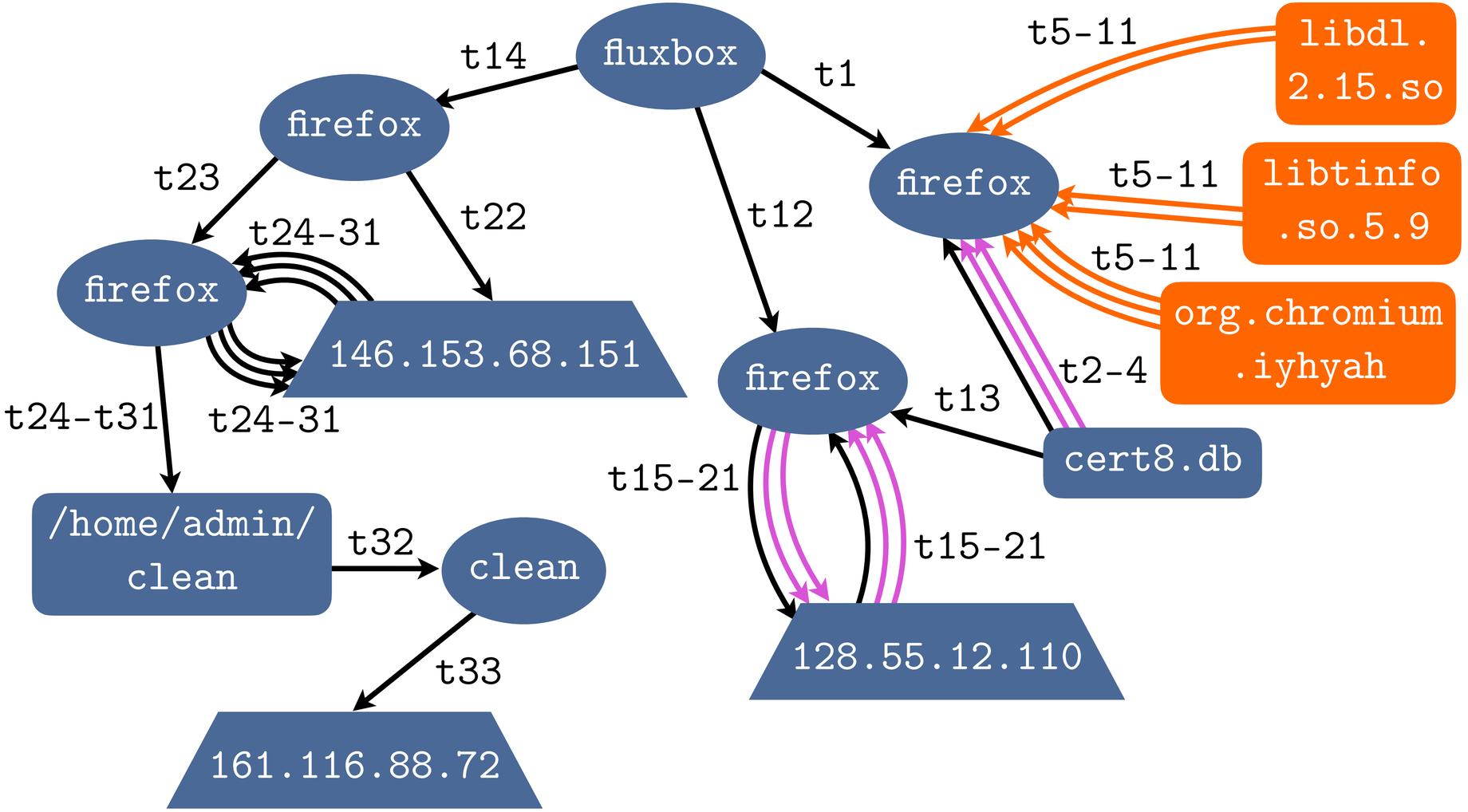


Yutao Tang et al., CCS '18

Reduction Filters: S-DPR

CPR
NodeMerge
S-DPR

```
    t1, clone, ...
    t2, read, ...
  * ...
  * t4, read, ...
*   t5, read, ...
* * ...
* * t11, read, ...
    t12, clone, ...
    t13, read, ...
    t14, clone, ...
    t15, send, ...
  * ...
  * t21, recv, ...
    t22, connect, ...
    t23, clone, ...
    t24, send, ...
    ...
    t30, recv, ...
    t31, write, ...
    t32, exec, ...
    t33, connect, ...
```

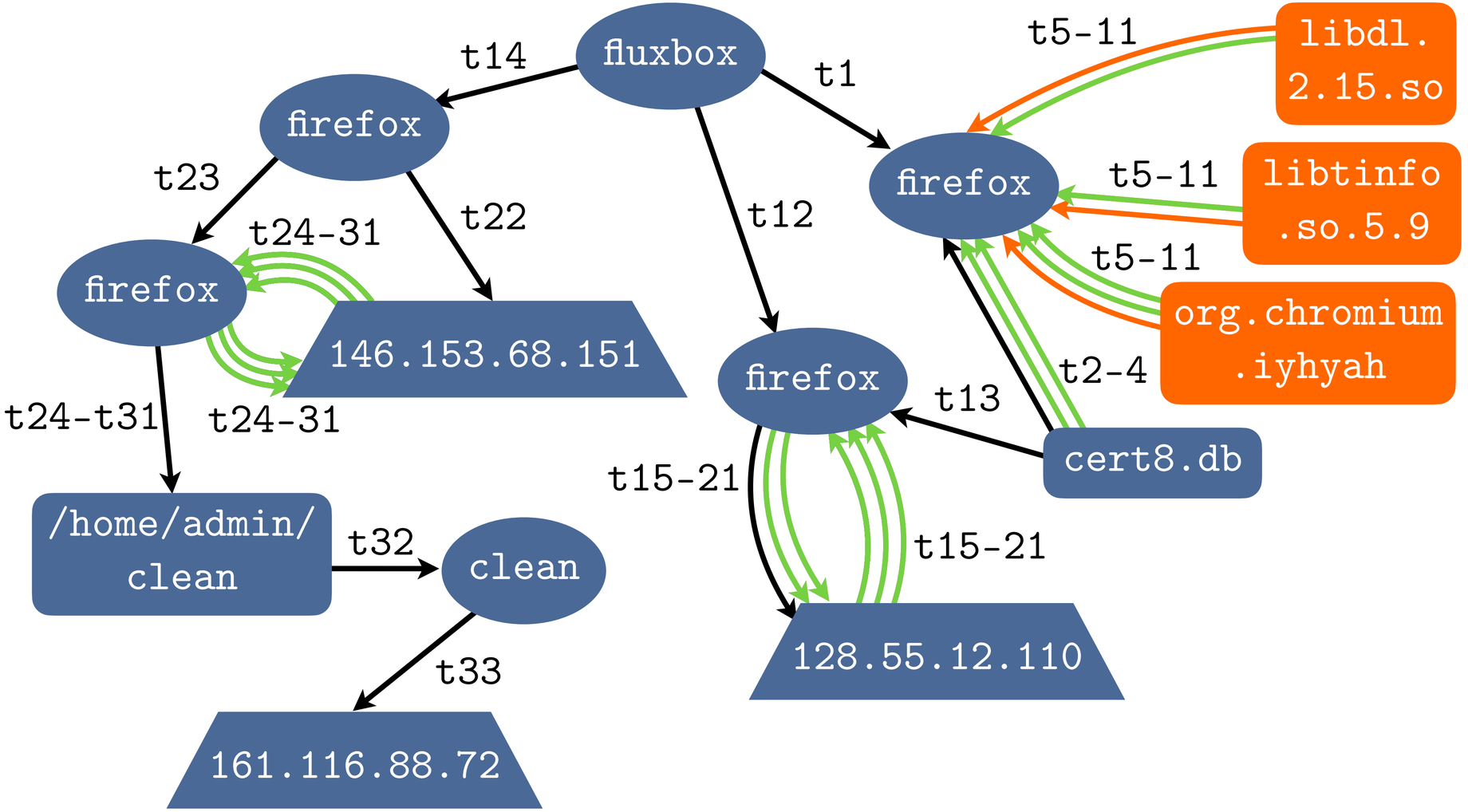


Md Nahid Hossain et al., USENIX Security '18

Reduction Filters: S-DPR

CPR
NodeMerge
S-DPR

```
        t1, clone, ...
        t2, read, ...
*   *   ...
*   *   t4, read, ...
  *     t5, read, ...
***     ...
***     t11, read, ...
        t12, clone, ...
        t13, read, ...
        t14, clone, ...
        t15, send, ...
*   *   ...
*   *   t21, recv, ...
        t22, connect, ...
        t23, clone, ...
*       t24, send, ...
*       ...
*       t30, recv, ...
        t31, write, ...
        t32, exec, ...
        t33, connect, ...
```

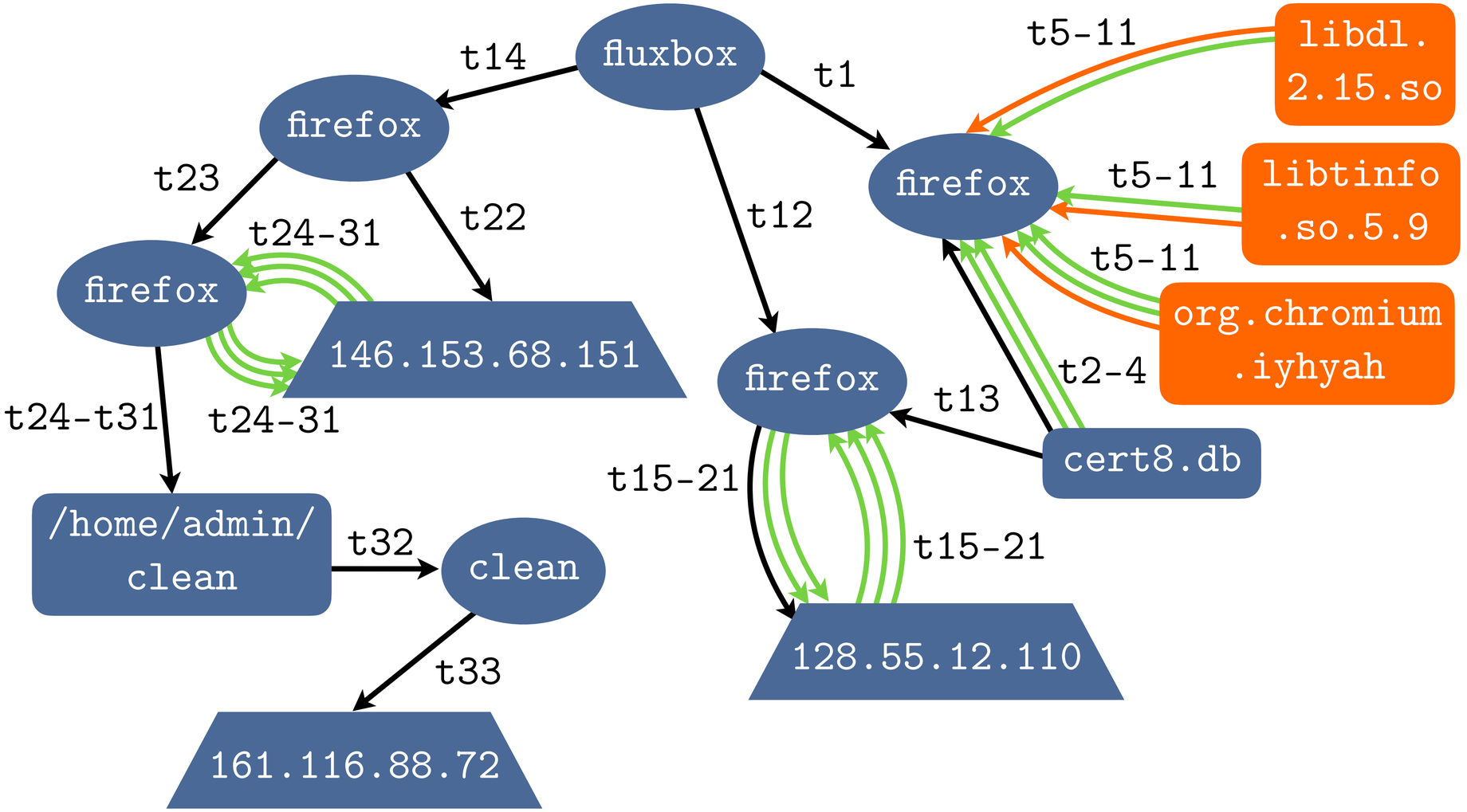


Md Nahid Hossain et al., USENIX Security '18

Final Reduction

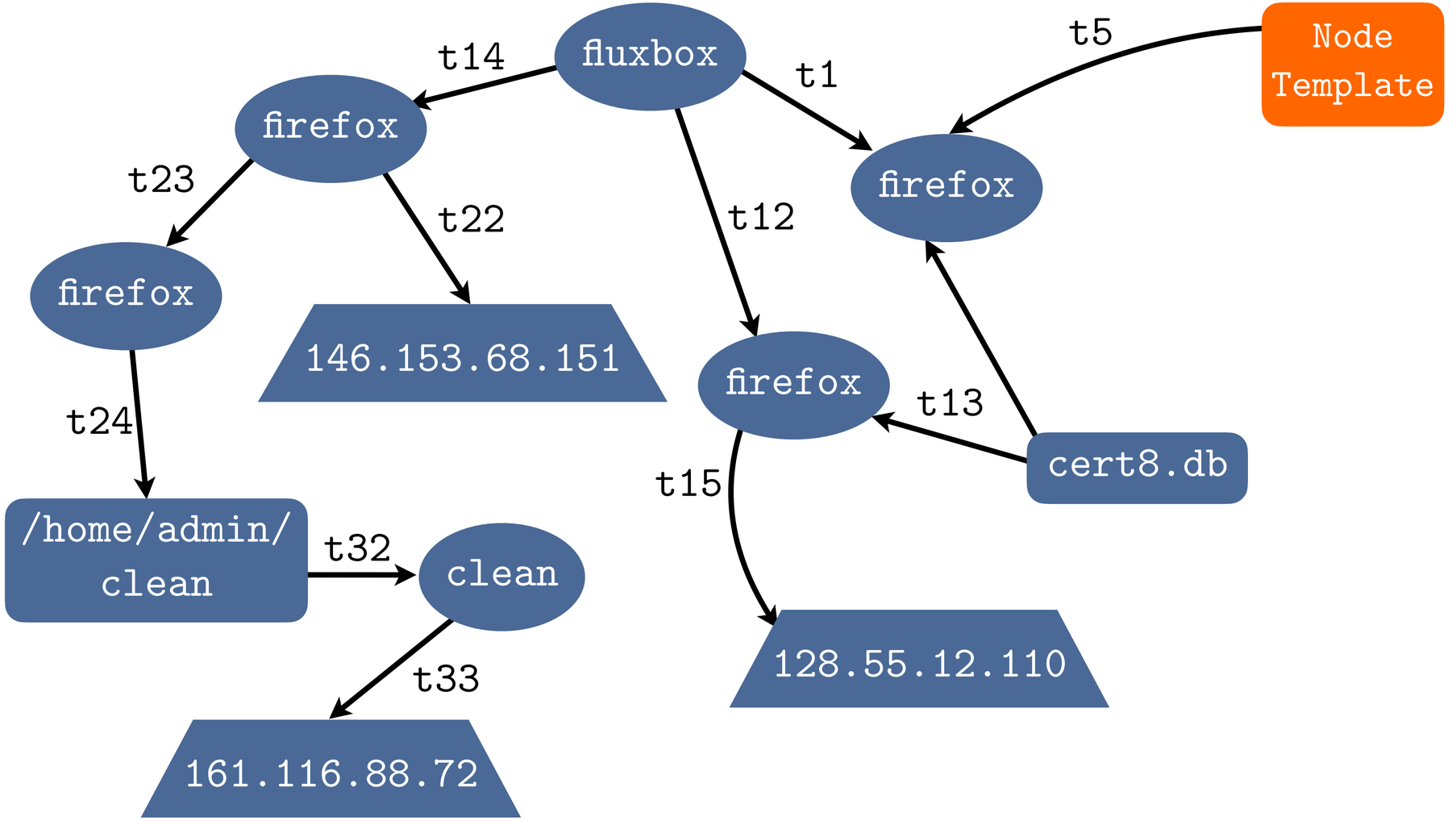
CPR
NodeMerge
S-DPR

```
        t1, clone, ...
        t2, read, ...
*   *   ...
*   *   t4, read, ...
  *     t5, read, ...
***     ...
***     t11, read, ...
        t12, clone, ...
        t13, read, ...
        t14, clone, ...
        t15, send, ...
*   *   ...
*   *   t21, recv, ...
        t22, connect, ...
        t23, clone, ...
*       t24, send, ...
*       ...
*       t30, recv, ...
        t31, write, ...
        t32, exec, ...
        t33, connect, ...
```

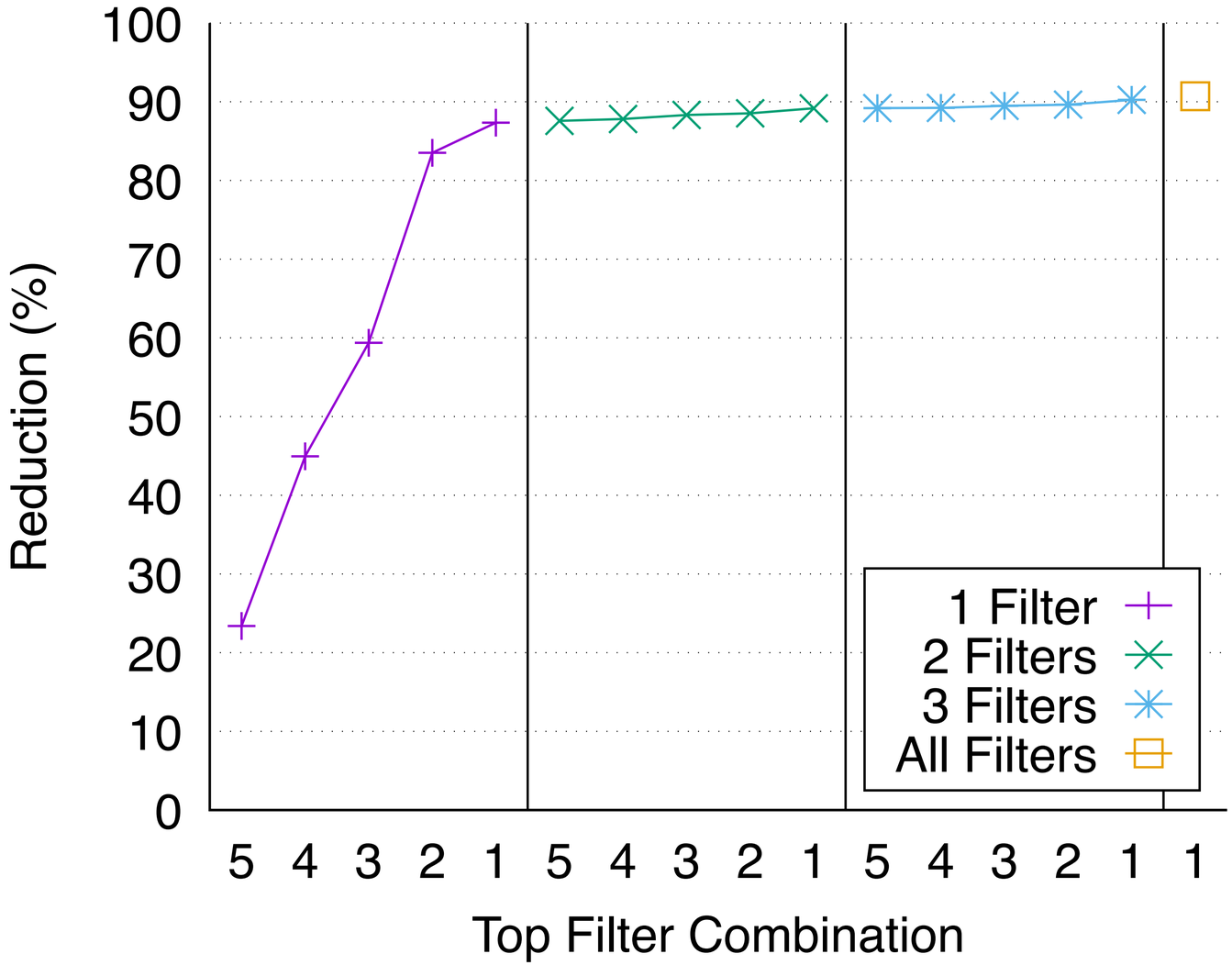


Final Reduction

```
t1, clone, ...
t2, read, ...
t5, read, ...
t12, clone, ...
t13, read, ...
t14, clone, ...
t15, send, ...
t22, connect, ...
t23, clone, ...
t31, write, ...
t32, exec, ...
t33, connect, ...
```

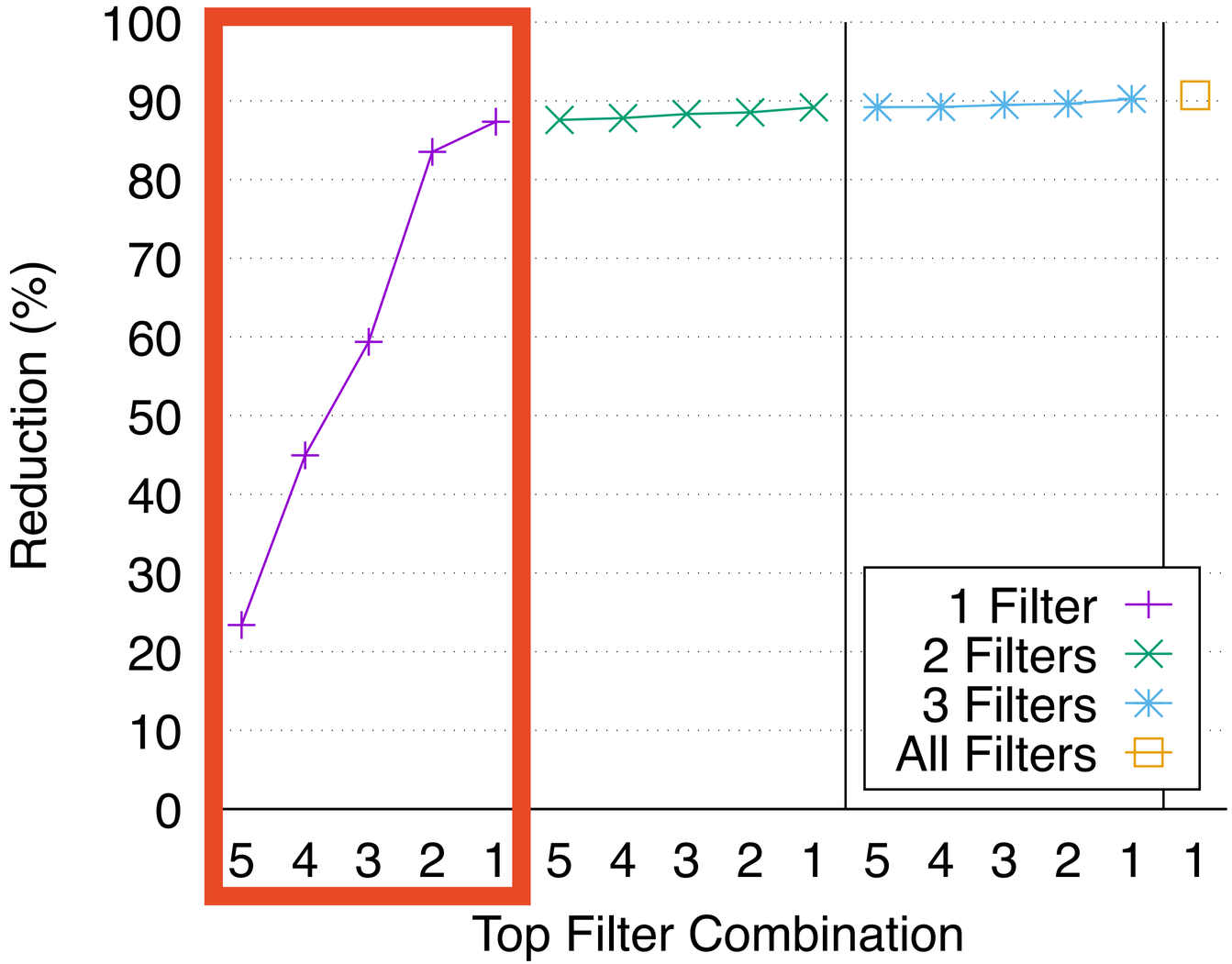


Reduction Evaluation



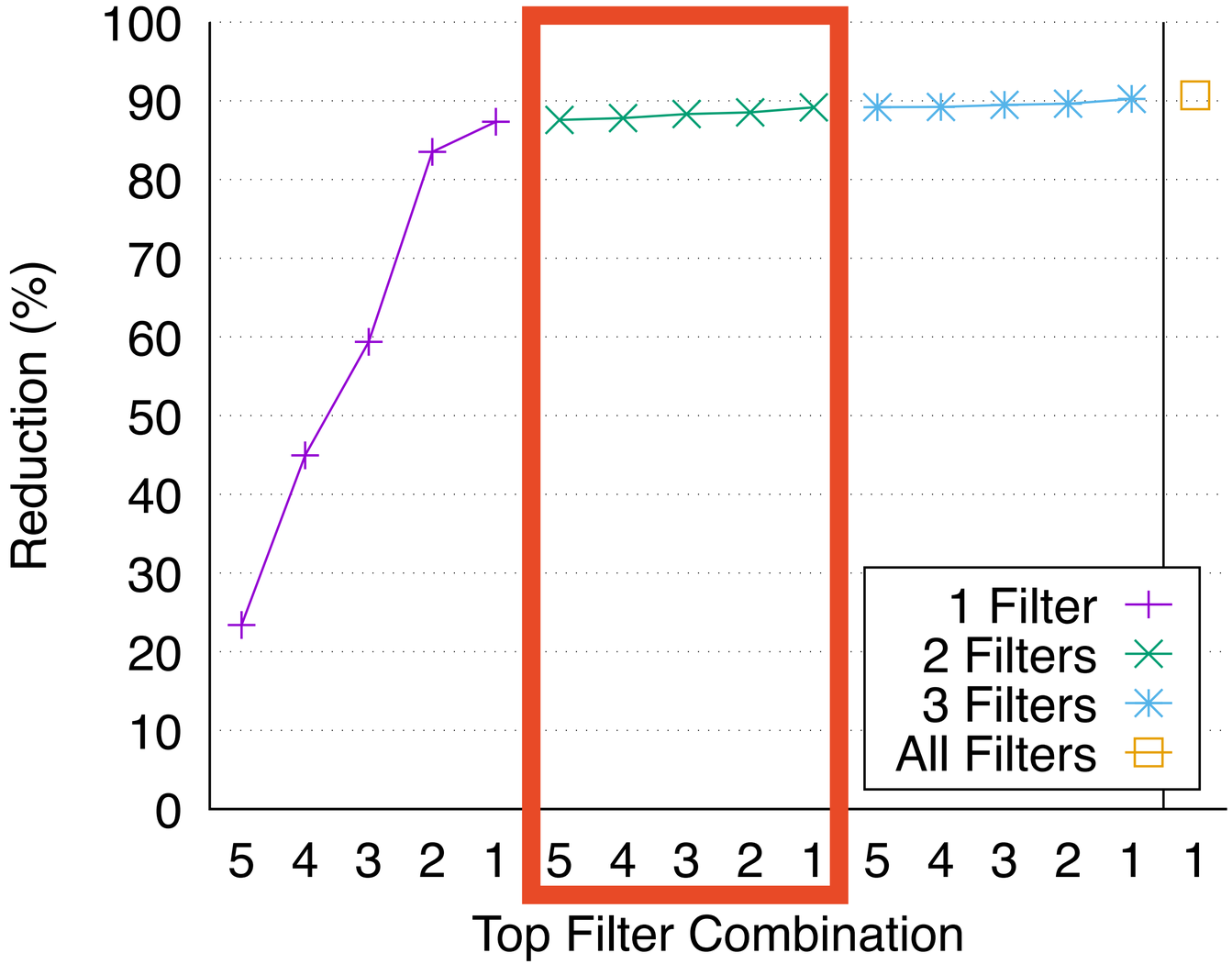
- Diminishing returns for increasing number of filters
  - 2-3 filters is generally a decent tradeoff of reduction to performance

Reduction Evaluation



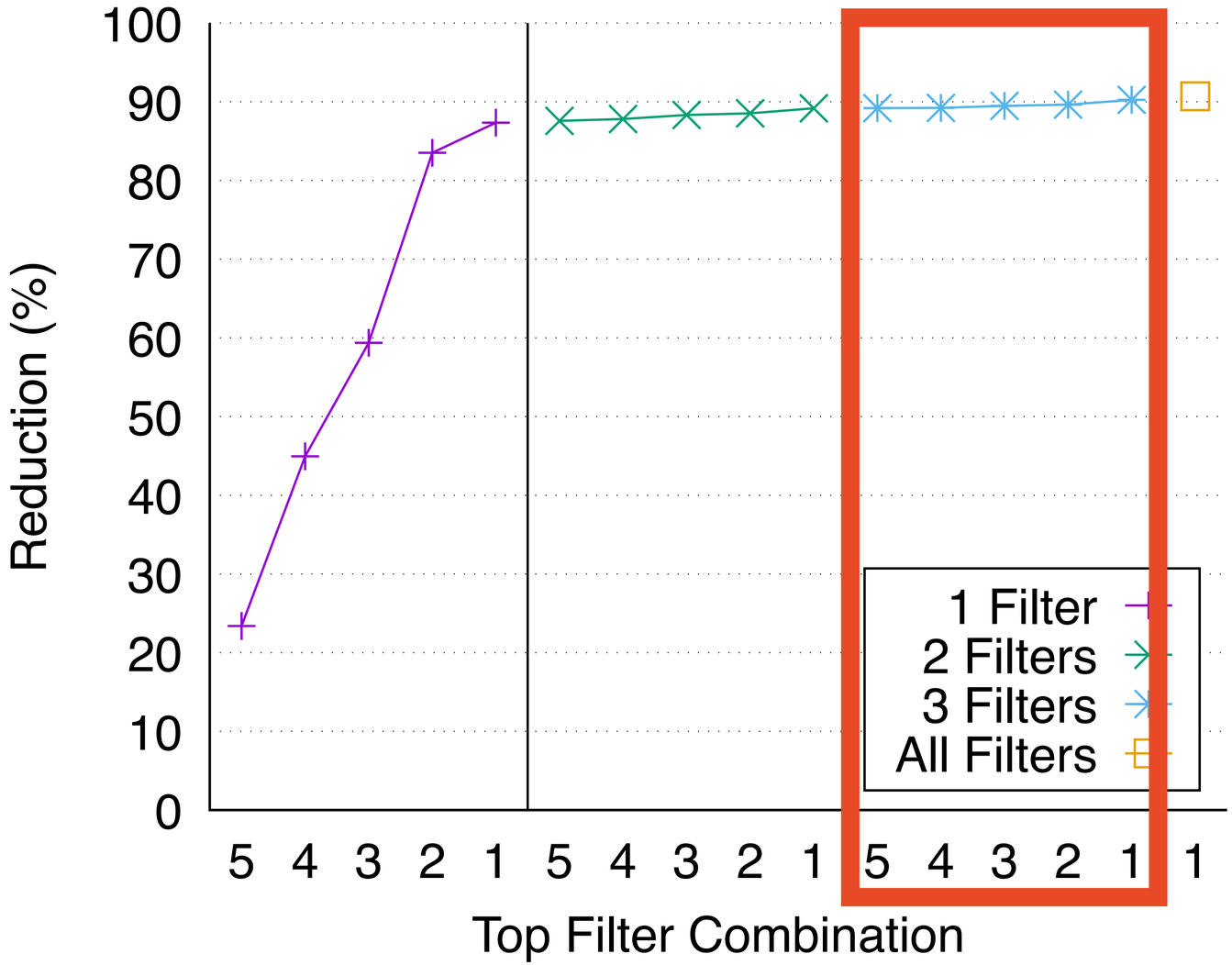
- Diminishing returns for increasing number of filters
  - 2-3 filters is generally a decent tradeoff of reduction to performance

Reduction Evaluation



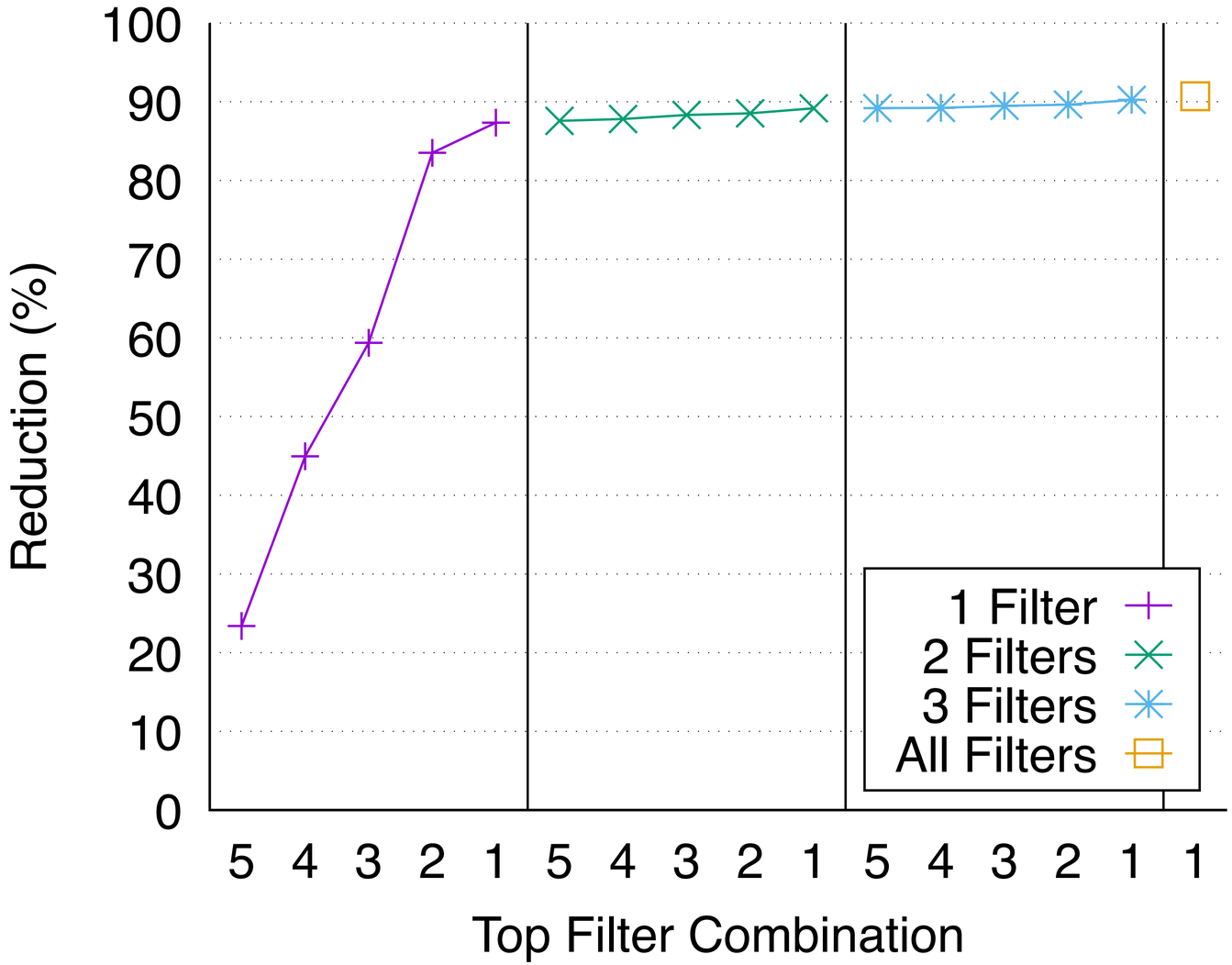
- Diminishing returns for increasing number of filters
  - 2-3 filters is generally a decent tradeoff of reduction to performance

Reduction Evaluation



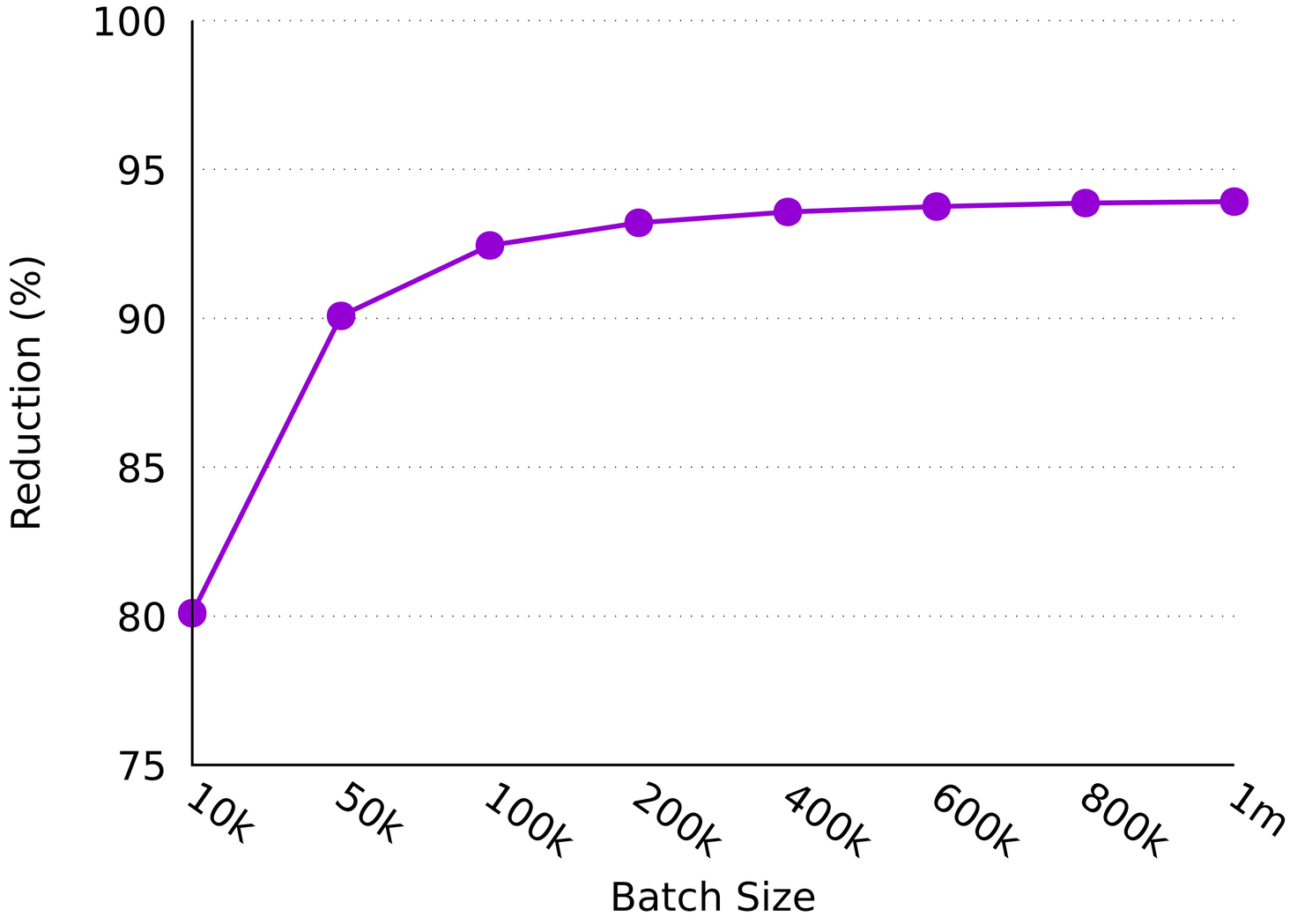
- Diminishing returns for increasing number of filters
  - 2-3 filters is generally a decent tradeoff of reduction to performance

Reduction Evaluation



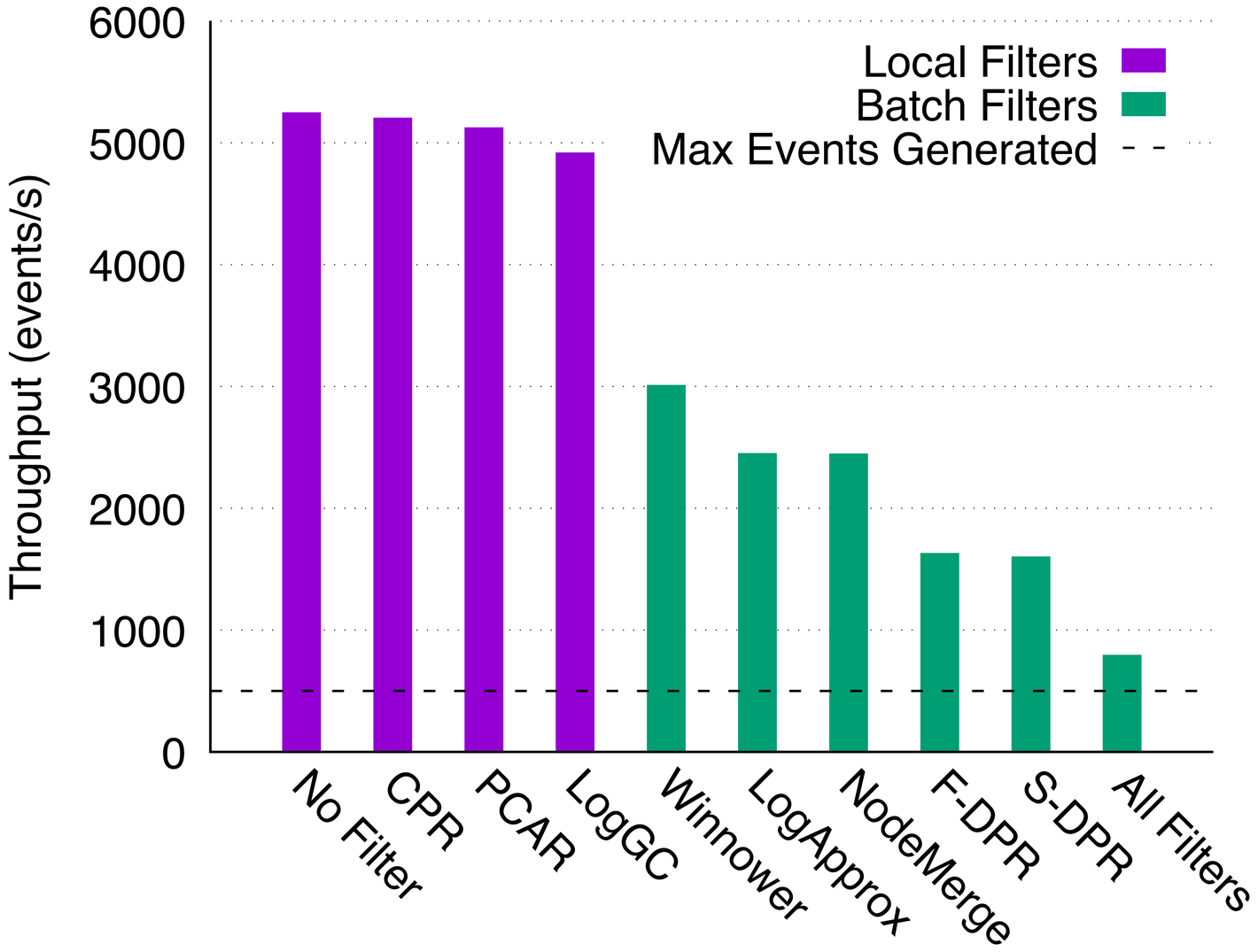
- Diminishing returns for increasing number of filters
  - 2-3 filters is generally a decent tradeoff of reduction to performance

Performance Evaluation



Reduction performance largely levels off by 100k logs per batch

Performance Evaluation



Local filters have much higher throughput than batch filters

Conclusion

- FAuST: easily implement and evaluate log reduction techniques
- Available open-source at https://bitbucket.org/sts-lab/faust
  - Transparent log reduction tool for any log analysis project or workflow
  - Easy baseline comparison with 8 existing techniques for new reductions
  - We use FAuST to enable our SoK on log reduction techniques [1]

Muhammad Adil Inam <mainam2@illinois.edu>
Jason Liu <jdliu2@illinois.edu>

[1] Muhammad Adil Inam et al., IEEE SP'23