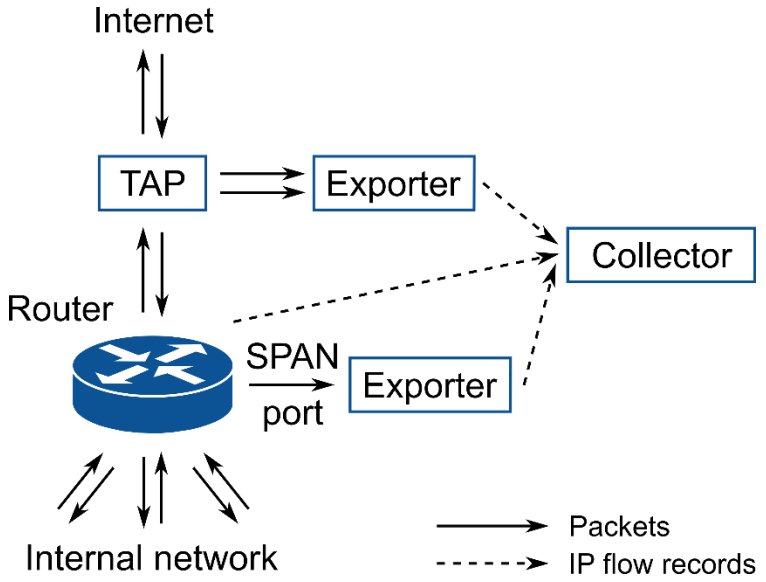
MUNI
C4E

Stream-Based IP Flow Analysis

Milan Cermak and Pavel Celeda

IP flow monitoring

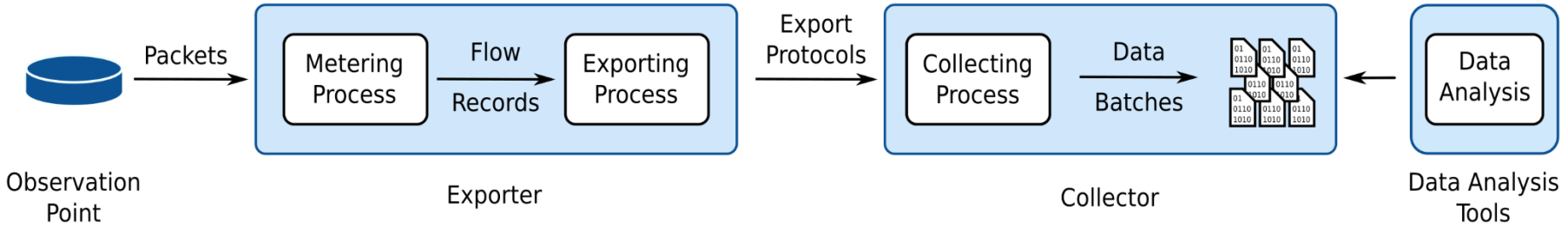
- Aggregation of packets into **n-tuples that have a set of common properties** (flow keys).
- IP flow records can be extended with **application data** information (HTTP, DNS, VoIP, …).
- Widely used for accounting, attack and anomaly detection, data retention, and network diagnostics in **high-speed and large-scale networks**.




```
Flow start  Duration  Proto  Src IP Addr:Port      Dst IP Addr:Port   Flags  Packets  Bytes
09:41:21.933   0.000  UDP   192.168.1.54:26011 ->      8.8.8.8:53     ......        1     58
09:41:21.967   0.000  UDP        8.8.8.8:53    -> 192.168.1.54:26011  ......        1    353
09:41:22.107   0.192  TCP   192.168.1.54:52172 -> 77.75.74.172:443    .AP.SF       13   1347
09:41:22.160   0.199  TCP   77.75.74.172:443   -> 192.168.1.54:52172  .AP.SF       27  29697
```


MUNI C4E

Monitoring workflow



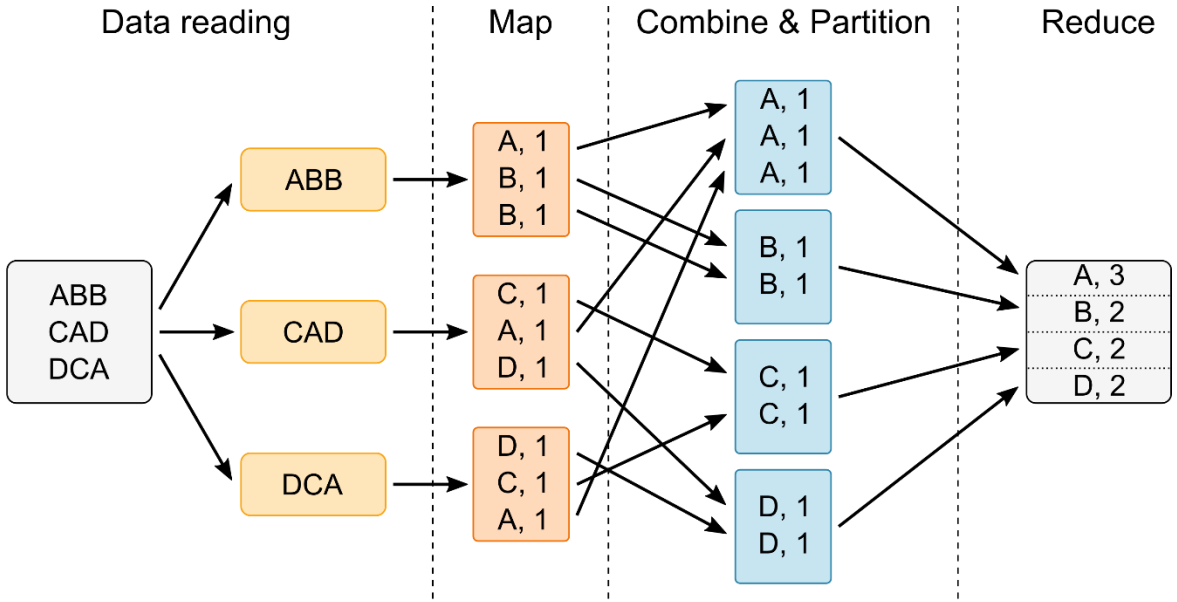
Issues and challenges

- **Delays may occur in every component** of the IP flow monitoring workflow, whereas in the worst case, the harmful event can be detected even in the order of minutes.
- Volume and velocity of **IP flow data continually increase** (currently, we observed an average of 12 000 IP flows/s generation rate in the MU and 110 000 IP flows/s rate in the CESNET network).
- Analytical methods are typically based on overall network state, whereas an **in-depth overview** is provided only in a limited way.

MUNI C4E

Distributed data stream processing

- IP flow processing has become a **Big Data problem**, as volume and velocity continuously increase.
- **MapReduce programming model** addresses batch data processing issues and enabled the development of new distributed architectures of data processing and storage systems.
- Novel analytical systems **ensure parallelization**, data storage, load balancing, and fault tolerance.



MUNI C4E

Research goals

1) **Evaluate the suitability of distributed stream processing** for analysis of IP flow data in high-speed and large-scale networks.

2) **Define an improved IP flow collection and analysis workflow** based on distributed stream processing of IP flow data.

3) **Propose advanced IP flow analysis methods** for high-speed and large-scale network monitoring and threat detection.

Benchmark of distributed stream processing systems

RG 1: Evaluate the suitability of distributed stream processing for analysis of IP flow data in high-speed and large-scale networks.

MUNI C4E

Comparison of stream processing systems

Each system is suited to a **different type of data and analysis** purpose and **wasn't designed for IP flow data analysis**, which means processing a large volume of small structured messages.

	Novel distributed stream processing system		
	Samza	**Storm**	**Spark**
Data source	Consumer	Spout	Receiver
Cluster manager	YARN, Mesos	YARN, Mesos, Zookeeper	Standalone, YARN, Mesos
Parallelism	Stream partitions based	Configured in Topology	Configured in SparkContext
Message processing	Sequential	Sequential	Small batches
Data sharing between nodes	Database, User implemented communication	Database, User implemented communication	Proprietary – SparkContext, Tachyon
Programming language	Java, Scala	Java, Clojure, Scala, any other using JSON API	Java, Scala, Python
Time window	Proprietary	User definition of Spout	Proprietary
Count window	Separate Job	User definition of Bolt	Accumulator

MUNI C4E

Performance benchmark for IP flow Analysis

Selected analysis operations

- Identity, Filter, Count, Aggregation, Top N, and SYN DoS

Dataset

- Sample from a **CAIDA dataset** transformed into IP flows records in **JSON format**.
- The average size of the one message of the dataset basis is **270 Bytes**.


```
{"date_first_seen":"2015-07-18T18:07:33.475+01:00", "date_last_seen":"2015-07-18T18:07:33.475+01:00",
"duration":0.000, "src_ip_addr":"86.135.210.175", "dst_ip_addr":"31.157.1.1", "src_port":54700,
"dst_port":80, "protocol":6, "flags":".A....", "tos":0, "packets":1, "bytes":56}
```

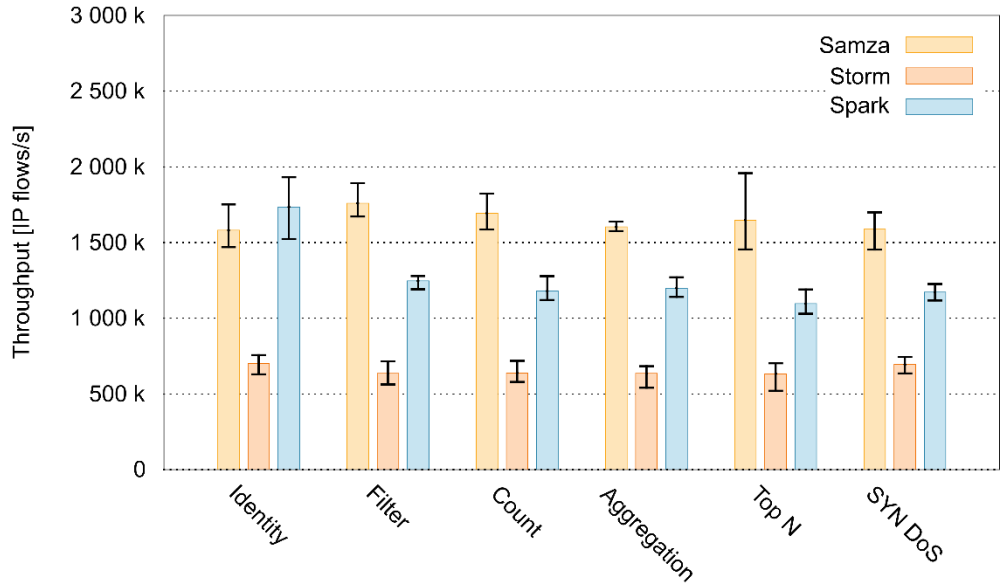

Benchmark architecture

- Kafka messaging system used to provide IP flow data and store analysis results.
- Selected distributed stream processing systems deployed on four types of environment:
  - **One node** – 32 vCPUs and 128 GB RAM, 16 vCPUs and 64 GB RAM
  - **Four-node cluster** – 4x: 8 vCPUs and 32 GB RAM, 4x: 4vCPUs and 16 GB RAM

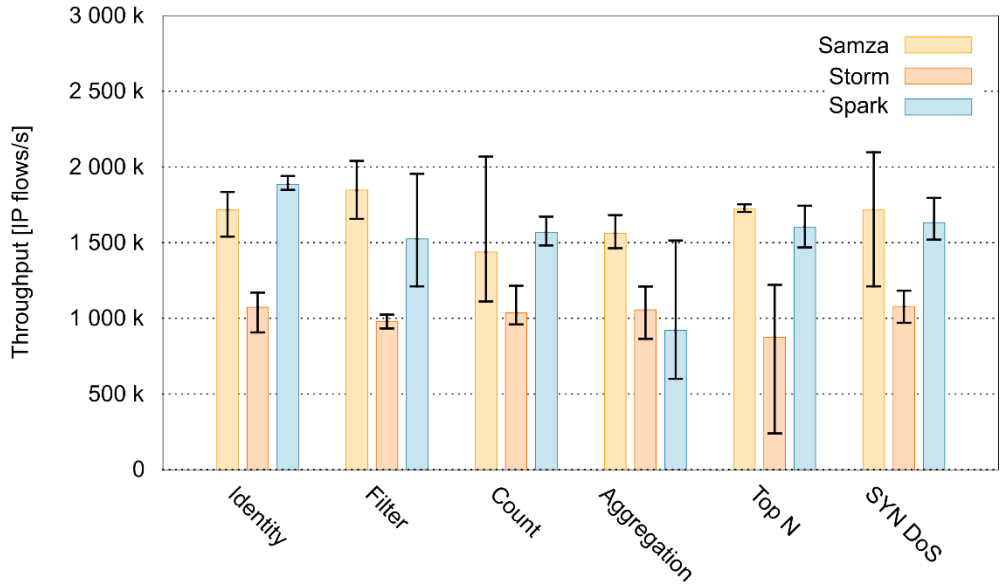
MUNI C4E

Benchmark results

Cluster – 4x: 4 vCPUs and 16 GB memory



Cluster – 4x: 8 vCPUs and 32 GB memory



- Each distributed stream processing system can **process at least 500 k IP flows/second** using 16 or 32 processor cores.
- **Samza's strict requirement for a number of data partitions** corresponding to available processor cores may be an issue in the real-world deployment.

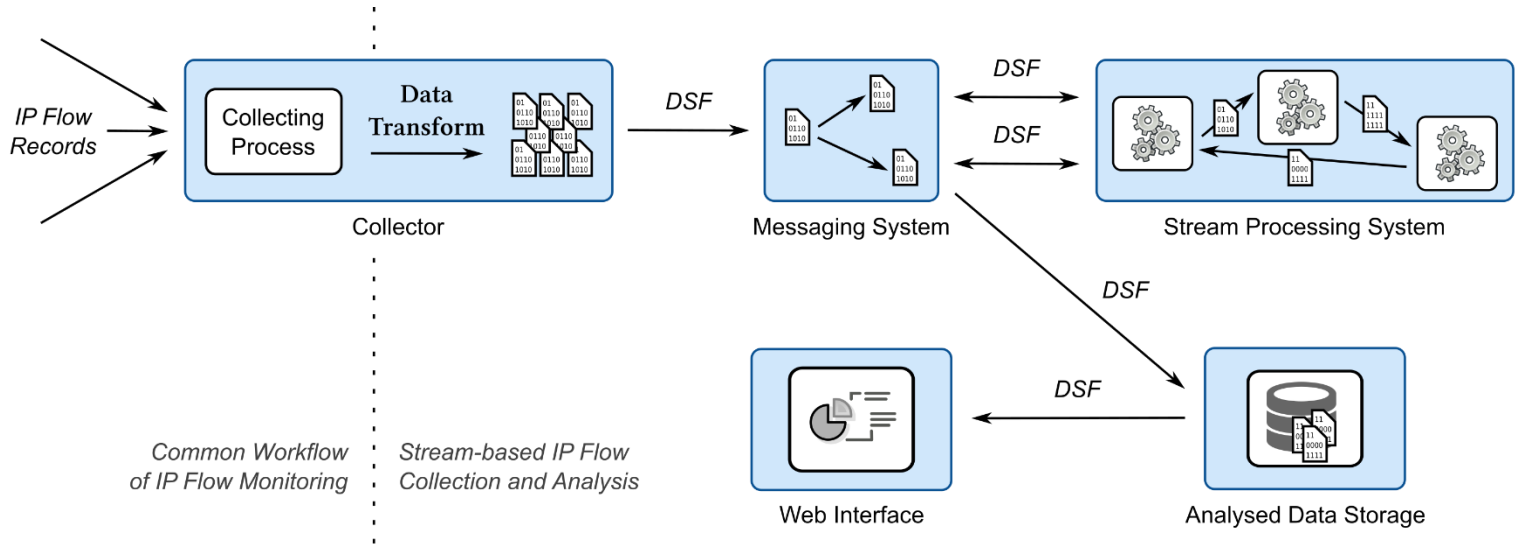
MUNI C4E

Stream-based IP flow analysis

RG 2: Define an improved IP flow collection and analysis workflow based on distributed stream processing of IP flow data.

MUNI C4E

Workflow design



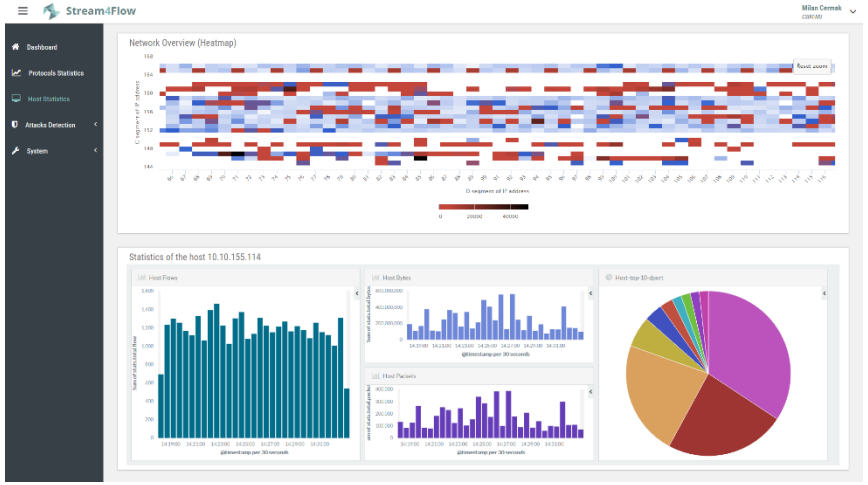
- The **collector may be omitted** from the workflow if the IP flow exporter is able to provide IP flow in the selected data serialization format.
- Messaging system allows to **utilize any available distributed stream processing system** and, at the same time, export analysis results to data storage or forward them to another data processing system.
- **JSON data serialization format** is a common format used by distributed stream processing systems.

MUNI C4E

Stream4Flow analysis framework

An open-source framework for near real-time analysis of IP flows based on world-leading technologies for stream data processing, network traffic monitoring, and visualization.

https://github.com/CSIRT-MU/Stream4Flow/



Framework architecture

- **IPFIXcol collector** – a receiver of IP flows from a majority of Netflow/IPFIX exporters able to transform them into the **JSON format**
- **Kafka messaging system** – scalable and fast **distribution of IP flow records** within the framework
- **Apache Spark** – the core of the framework with **custom applications** for near real-time IP flow analysis
- **Elastic Stack** – **storage of analysis results** providing basic visualizations using the Kibana framework
- **Additional web interface** – customized **visualization** of analysis results

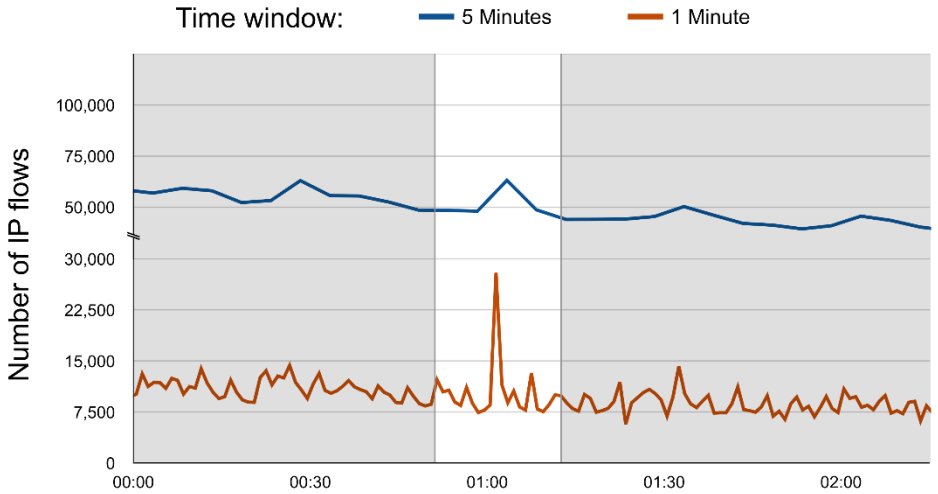
MUNI C4E

Framework evaluation

Experimentally deployed within Masaryk University campus network with **24 000 active IP addresses** and **12 000 IP flows/s** observed on average.

Near real-time intrusion detection

- IP flows are analyzed **on the fly** as soon as the system receives them.
- Analysis can be done over **short time windows** to reveal information lost by aggregation.
- We were able to **transform the most used methods** by utilizing the sliding time windows functionality.



In-depth cyber situational awareness

- Utilization of the MapReduce programming model for **creating the micro view** on network traffic by computing statistics for all host's IP addresses in the network.

MUNI C4E

Network monitoring and threat detection

RG 3: Propose advanced IP flow analysis methods for high-speed and large-scale network monitoring and threat detection.

MUNI C4E

Detection of DNS traffic anomalies

IP flows extended by information from a DNS traffic

- Useful **DNS packet fields** – queried domain name, queried record type, response return code, response value corresponding to the query record type, and time to live of the response.
- Extended DNS IP flows can be **immediately exported** to the collector to speed up analysis and save flow cache storage.

DNS traffic monitoring

- Change in DNS traffic statistics may reveal **misconfigured hosts** and **indicate a cybersecurity risk**.
- The most important **TOP N statistics** – DNS record types, DNS response codes, queried domains, queried external DNS resolvers, hosts with the most queries or responses.

Anomaly and intrusion detection

- **Malware domains query** – near real-time correlation of observed domains with a predefined blacklist.
- **External DNS resolver usage** – a combination of observed IP flows to verify access to a queried domain.
- **Open DNS resolvers** – response analysis and identification of non-local domain queries.

Near real-time patterns detection

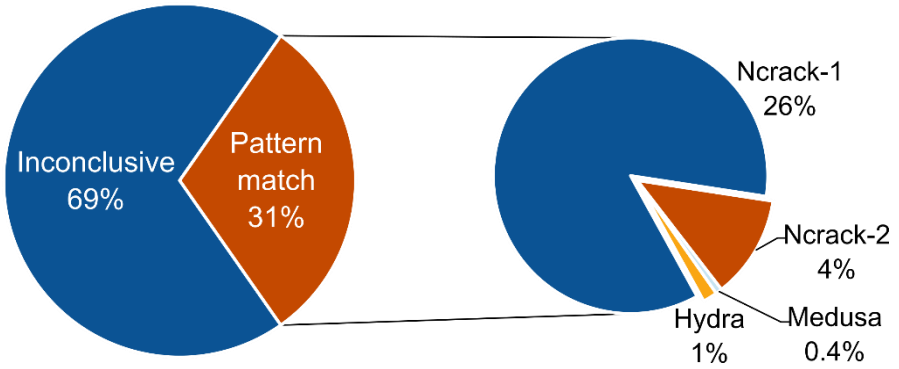
Utilization of stream-based IP flow analysis and MapReduce programming model for computation of traffic **aggregations for each pair of connection peers** or each observed IP address.

Detection approach

1) Extraction of the **selected features** from bi-flow or one-direction IP flows.
2) Continuous comparison of extracted vectors with defined patterns and **computation of their distance**.
3) Aggregation of computed distances and traffic **classification based on the degree of certainty**.

Patterns definition

- Testbed environment or analysis of real-world traffic and filtering of desired anomalies and attacks.
- Statistical operations (e.g., average, median), machine learning, or clustering methods.



MUNI C4E

Summary

MUNI C4E

Main objectives

Research how the IP flow collection and analysis can be improved by distributed stream data processing and investigate its capabilities for advance network monitoring and threat detection.

Achieved contributions

- ✓ Proposed a novel **benchmark of distributed stream processing systems** allowing their evaluation concerning IP flow monitoring requirements.
- ✓ Introduced **a stream-based IP flow analysis workflow** for processing network traffic data in high-speed and large-scale networks in near real time.
- ✓ Prepared an open-source framework **Stream4Flow** based on the workflow.
- ✓ Defined **new IP flow analysis methods** utilizing stream-based processing.

Thank you for your attention!

The thesis with all details is available at https://is.muni.cz/th/tgxb6/.

MUNI C4E

MUNI
C4E

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

C4E.CZ