

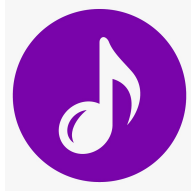


# Holistic Runtime Performance and Security-aware Monitoring in Public Cloud Environment

**Devki Nandan Jha**, Graham Lenton, James Asker, David Blundell, David Wallom



# Motivation



Applications



Cloud providers

# Motivation

---

## 1. System Performance



## 2. System Security



# Run-time monitoring

---

User Applications

Operating System

Hardware

# Run-time monitoring

---

Network-level monitoring

User Applications

Operating System

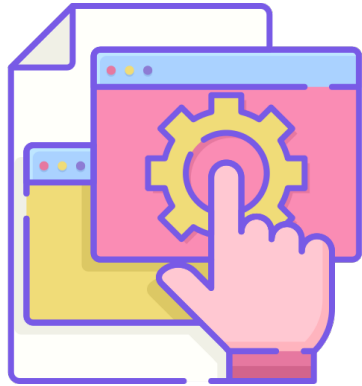
Hardware

User-level monitoring

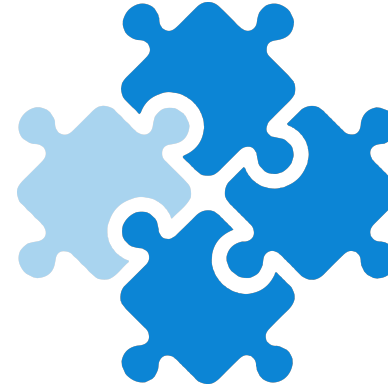
Kernel-level monitoring

# Run-time Monitoring Challenges

---



1. Monitoring features



2. Context Matching



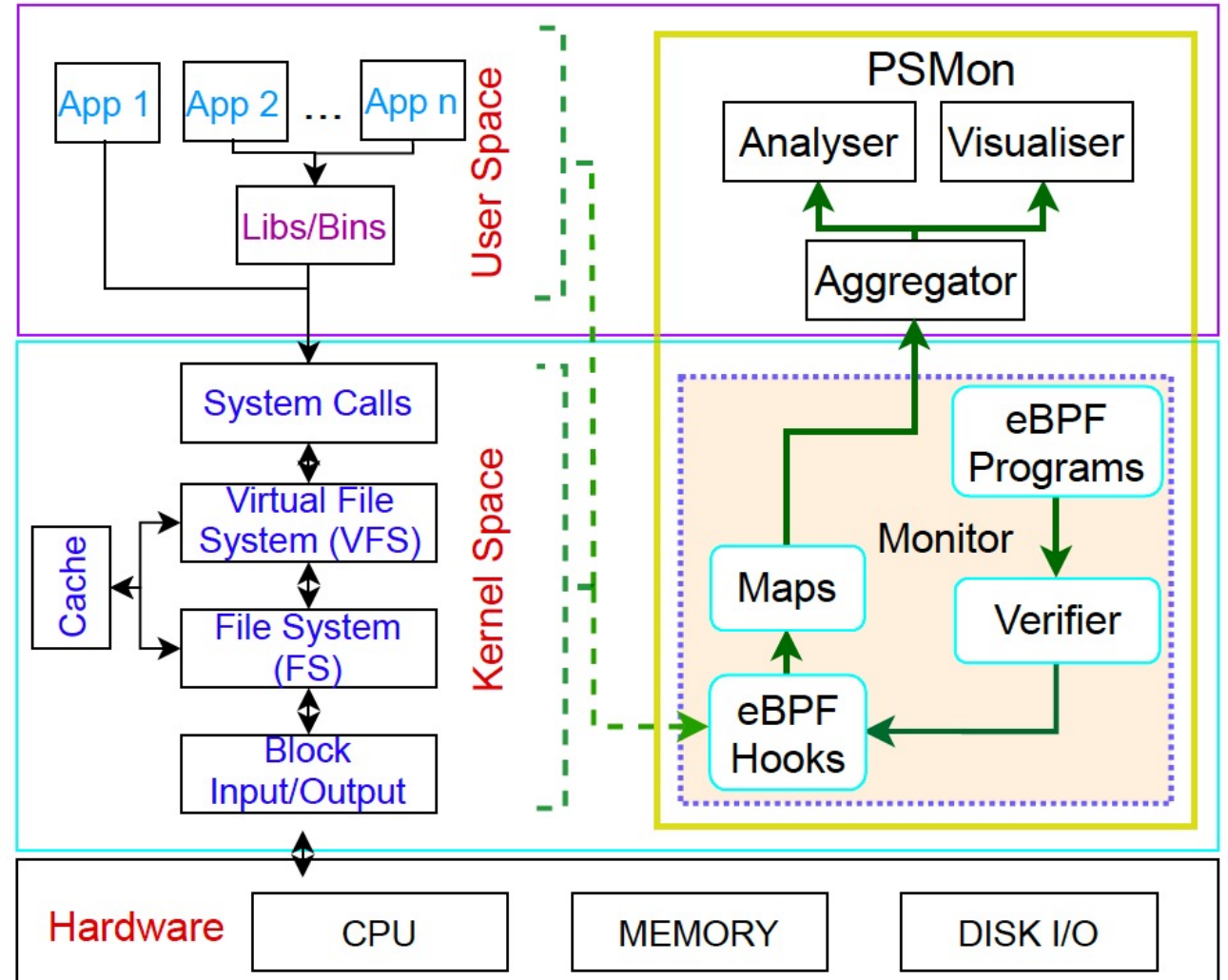
3. Monitoring Overhead



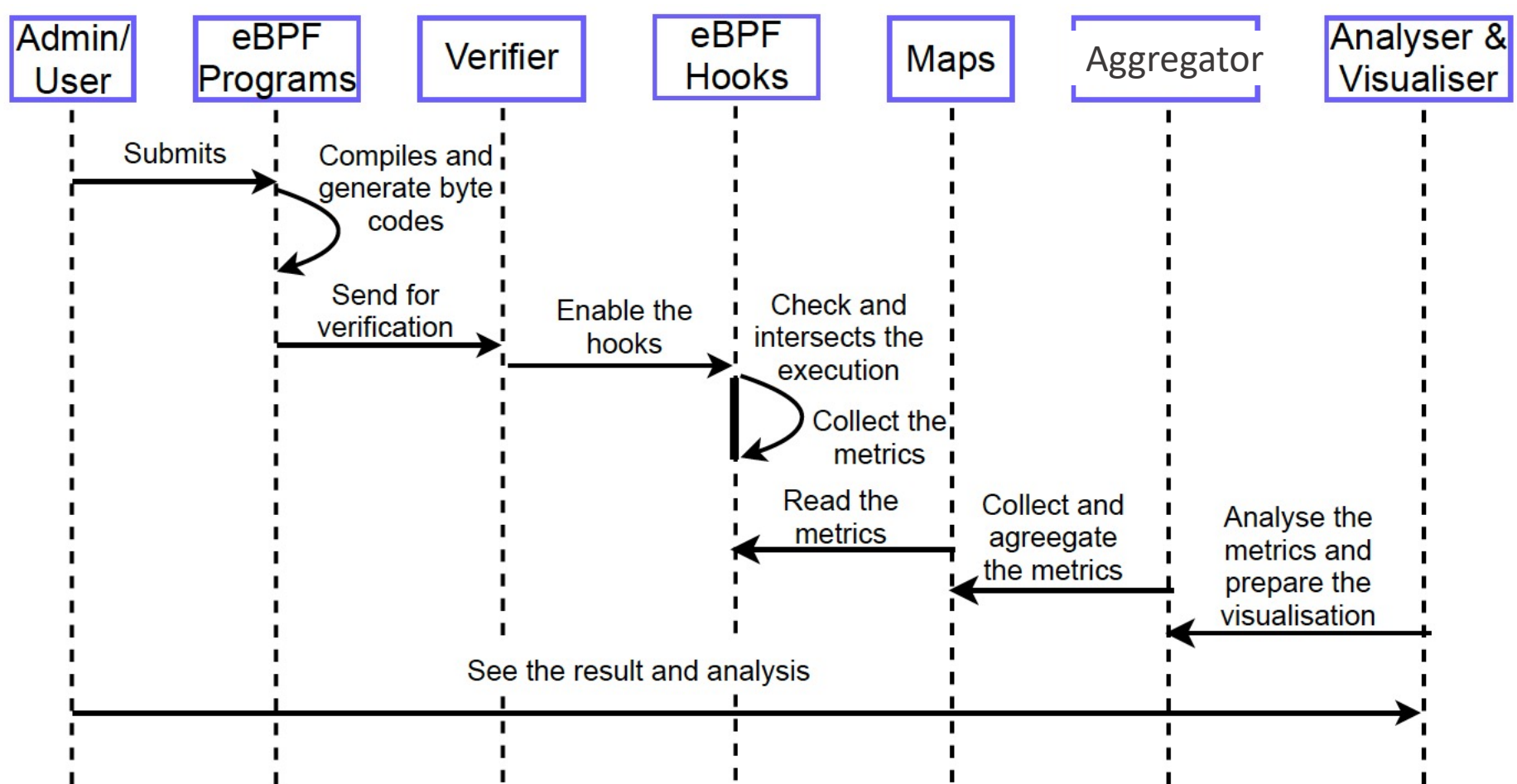
4. System Reliability

# PerSecMon framework

- System Architecture:
  - Monitor
  - Aggregator
  - Analyser
  - Visualiser



# PerSecMon framework





# PerSecMon framework

- Probes:
  - User process
  - Memory
  - System calls
  - Virtual file system
  - Block Input/Output

No.	Probe	Type	Description
1	<i>gc__start</i>	User Process probes	Start of garbage collection
2	<i>function__entry</i>		Entry of a method
3	<i>getaddrinfo/gethostbyname, do_entry</i>		Start of a process call
4	<i>getaddrinfo/gethostbyname, do_return</i>		Completion of a process call
5	<i>malloc, calloc, realloc</i>	Memory tracepoints	Allocation of memory
6	<i>memalign, posix_memalign</i>		Allocate aligned memory
7	<i>free</i>		Memory release
8	<i>fname_open</i>	Kernel probes (System calls)	Name of open files
9	<i>fname_openat</i>		Time of files opened at
10	<i>vfs_read</i>	Kernel probes (VFS)	Data reading operation
11	<i>vfs_write</i>		Data writing operation
12	<i>vfs_open</i>		File opening operation
13	<i>blk_account_io_start</i>	Kernel probes (Block I/O)	Start of a block I/O
14	<i>blk_account_io_done</i>		Completion of a block I/O
15	<i>blk_mq_start_request</i>		Start of a queuing process
16	<i>function</i>	Kernel Probes (Stack)	Kernel function

# System Evaluation

---

- Environment:
  - Ubuntu 21.04 server
- Benchmarks:
  - Fibonacci Sequence
  - Bonnie++
  - Netperf

# System Evaluation

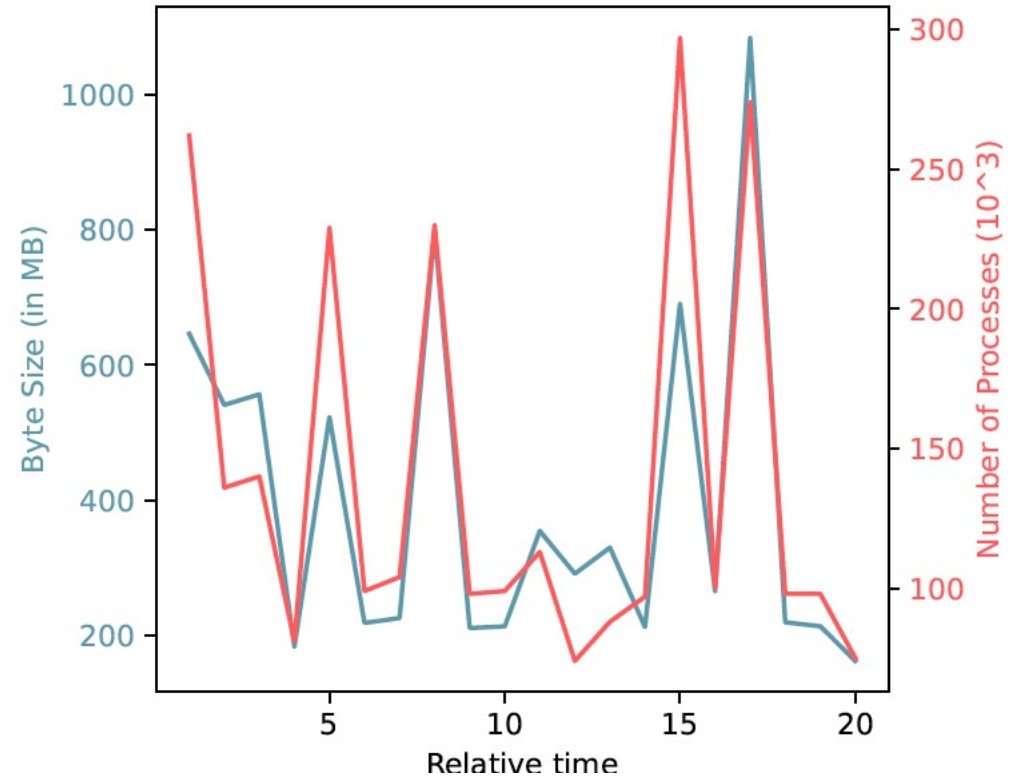
- Results:
  - User process [Fibonacci sequence (Python)]

```
{
  "_index": "ustat",
  "_type": "_doc",
  "_id": "GACvv30BGFTwtdeLXrMJ",
  "_score": 1,
  "_source": {
    timestamp → "ts": "2021-12-15T19:42:03.070793",
    execution command → "command": "python ",
    garbage collector → "gc": 0,
    no. of threads → "thread": 0,
    ← process id "pid": 119428,
    ← exceptions "excp": 0,
  }
}
```

Information about one process in JSON format

# System Evaluation

- Results:
  - Memory leaks recognition

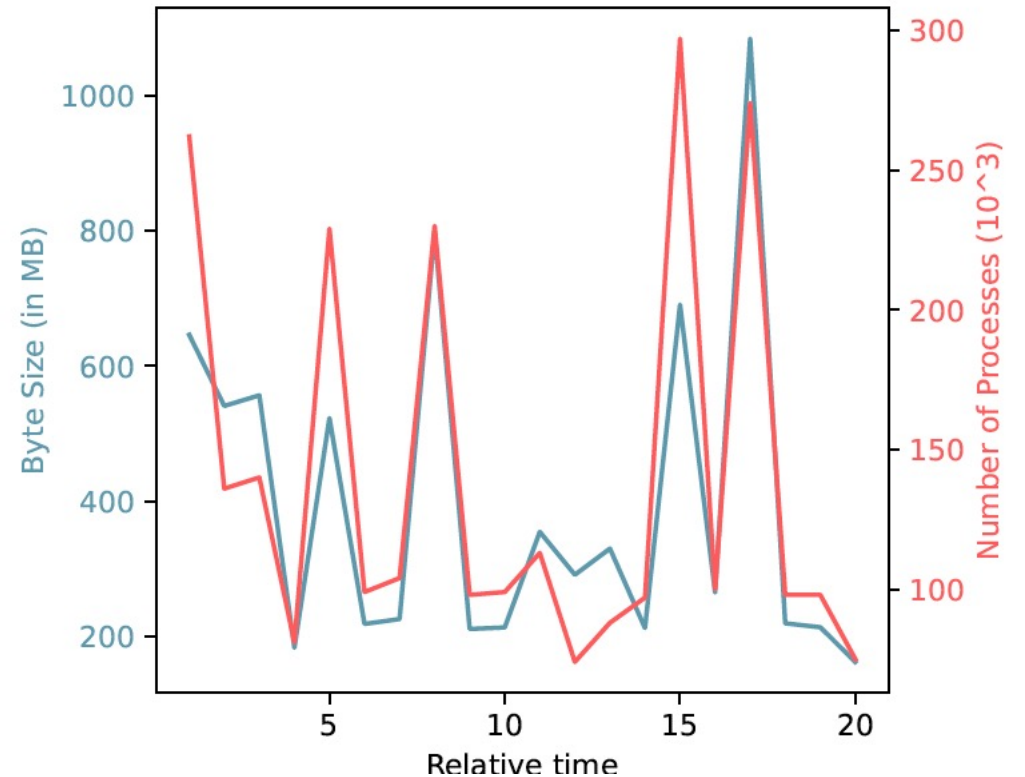


# System Evaluation

- Results:
  - Memory leaks recognition

```
184410112 bytes in 45022 allocations from stack
__alloc_pages_nodemask+0x236 [kernel]
__alloc_pages_nodemask+0x236 [kernel]
alloc_pages_vma+0x87 [kernel]
do_anonymous_page+0xec [kernel]
handle_pte_fault+0x229 [kernel]
__handle_mm_fault+0x59a [kernel]
handle_mm_fault+0xd7 [kernel]
do_user_addr_fault+0x1a0 [kernel]
exc_page_fault+0x69 [kernel]
asm_exc_page_fault+0x1e [kernel]
```

Result showing the processes of the stack with most outstanding memory allocation



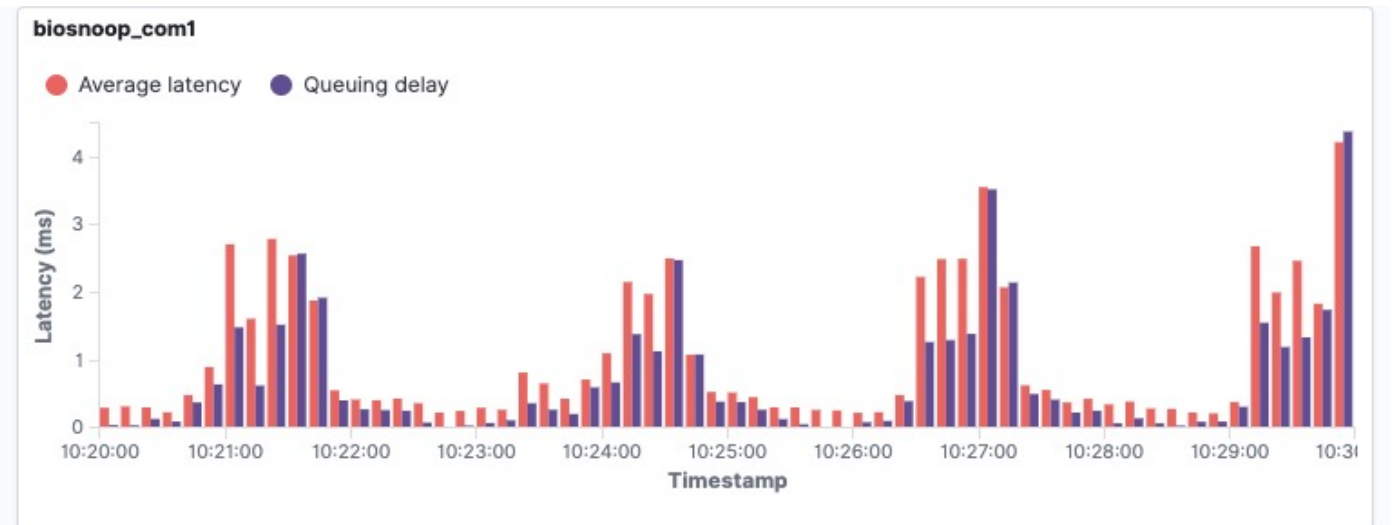
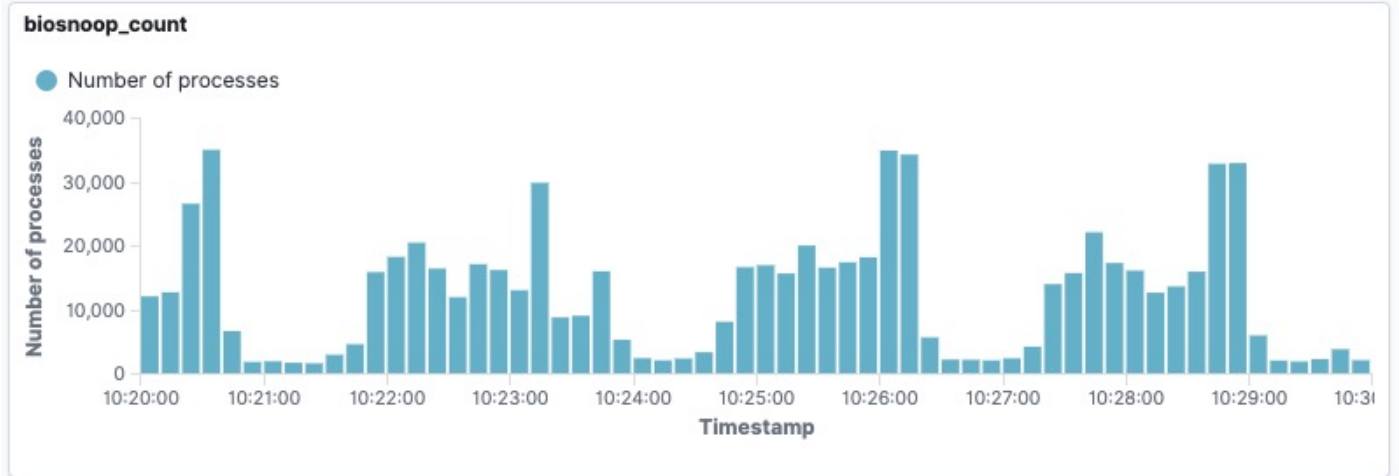
# System Evaluation

- Results:
  - Network latency
  - Shows the live latency of all the processes accessing the network host

TIME	PID	COMM	LATms	HOST
18:55:43	119252	wget	1.48	github.com
18:55:44	119252	wget	0.78	codeload.github.com
18:55:44	119253	curl	0.33	localhost
18:55:44	119255	curl	1.40	google.com
18:55:44	119257	ping	1.04	fb.com
18:56:23	119258	ping	0.34	localhost:9200
18:56:23	119259	netserver	0.03	:::0
18:56:23	119259	netserver	0.01	0.0.0.0
18:56:23	119260	netperf	0.28	localhost
18:56:23	119260	netperf	0.06	0.0.0.0
18:56:23	119260	netperf	0.09	localhost
18:56:23	119260	netperf	0.02	0.0.0.0
18:56:23	119261	netserver	0.26	0.0.0.0

# System Evaluation

- Results:
  - Block I/O
  - comparative view of the number of processes and the the latency



# Conclusion

---

- Run-time monitoring is still challenging
- Designed and implemented a solution, PerSecMon
- Captured metrics are analysed and visualised easily
- Can be easily adapted for Mobile cloud environment



---

# Questions?

Contact:

Devki Nandan Jha

Oxford e-Research centre, University of Oxford

100 Percent IT Ltd. (Cyberhive)

mail: [devki.jha@eng.ox.ac.uk](mailto:devki.jha@eng.ox.ac.uk)

web: <https://dnjha.github.io>

