Real-Time Encrypted Traffic Classification with P4-DPDK

Amith GSPN, Ali Mazloum, Samia Choueiri, Sergio Elizalde, Elie F. Kfoury, Jorge Crichigno

University of South Carolina, USA

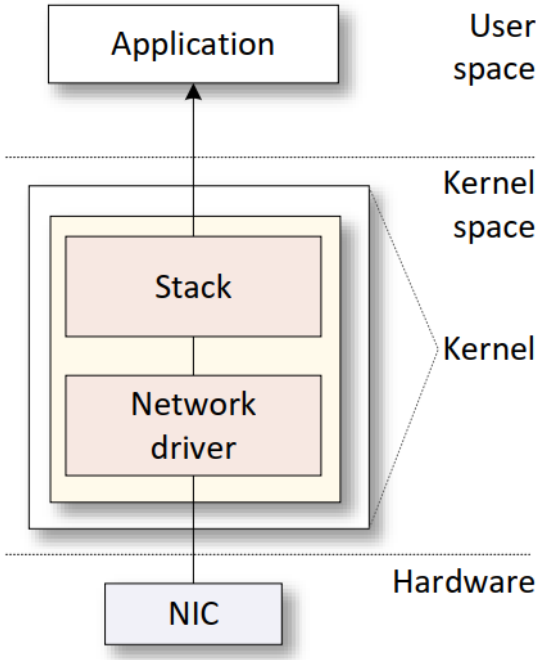
THE LINUX FOUNDATION

Agenda

- Background on packet processing acceleration
- Introduction to Data Plane Development Kit (DPDK) and P4-DPDK
- Proposed system
- Evaluation
- Conclusion

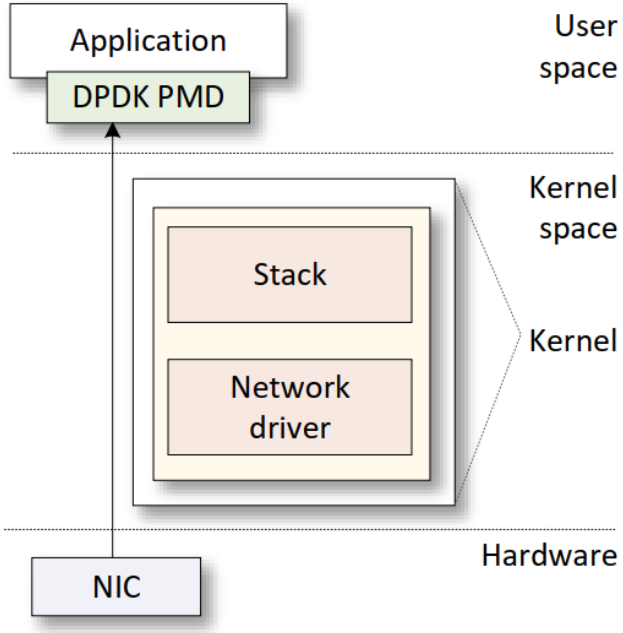
Standard Packet Processing

- The Network Interface Card (NIC) driver pre-allocates kernel memory buffers where the packets are stored.
- The NIC driver pre-allocate the transmit (TX) and receive (RX) ring buffer in the memory.
- The ring buffers store the packet buffer pointer and its length.
- The NIC copies the packet to the location using Direct Memory Access (DMA).
- The NIC triggers an interrupt.



Kernel-Bypass Packet Processing Using DPDK

- Bypassing the kernel is a solution to avoid kernel overheads and accelerate packet processing.
- DPDK is a set of optimized libraries for processing packets in the user space while bypassing the kernel.
- DPDK uses Poll Mode Drivers (PMD), which constantly poll the NICs for new packets to avoid overheads resulting from interrupts.



Using P4 Programming for DPDK

DPDK

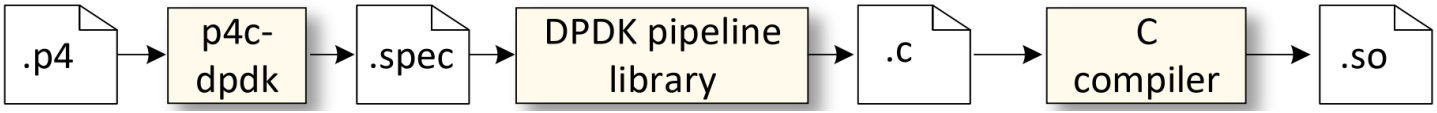
- DPDK may be complex to program. It requires knowledge of DPDK libraries
- DPDK requires many lines of code to implement a packet processing pipeline
- Adding features and modifying a large may become complex

P4-DPDK

- Combining P4 with DPDK enables programmers to write packet processing apps and obtain high performance
- P4-DPDK optimizes the size of the code compiled into a DPDK programmable pipeline
- P4 quickly adapts and deploys new network functions to the data plane

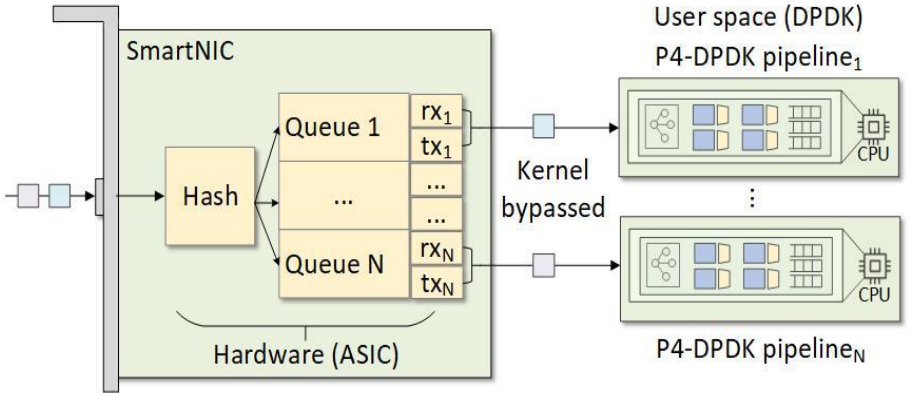
The P4-DPDK Workflow

- The system is implemented in P4 using the Portable NIC Architecture (PNA).
- The P4 code consists of all packet processing functions, including customized code.
- The p4c-dpdk compiler accepts the P4 code as input and generates a representation file (.spec file) that aligns with the DPDK software switch pipeline.
- The DPDK pipeline runs on the CPU core of the host, bypassing the kernel.
- The C code is compiled, and a shared object (.so) is generated.



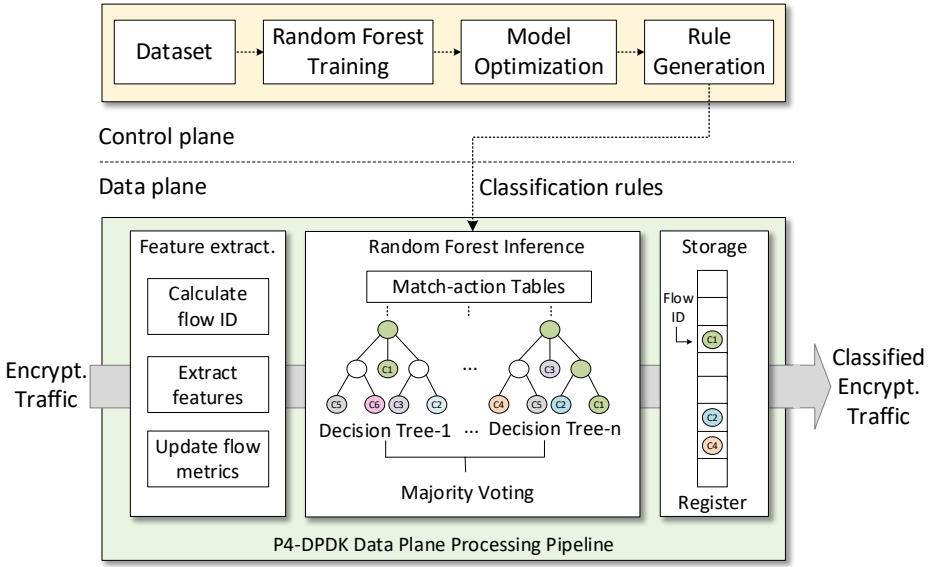
Core Affinity and Multicore Processing

- When a NIC receives a packet, it typically goes through a single CPU core for pipeline processing; this pipeline is often the application's bottleneck.
- Core affinity allows binding packet processing to a specific CPU core.
- The NIC can distribute packets across multiple CPU cores using the Receive Side Scaling (RSS).
- RSS uses hashing for load balancing.



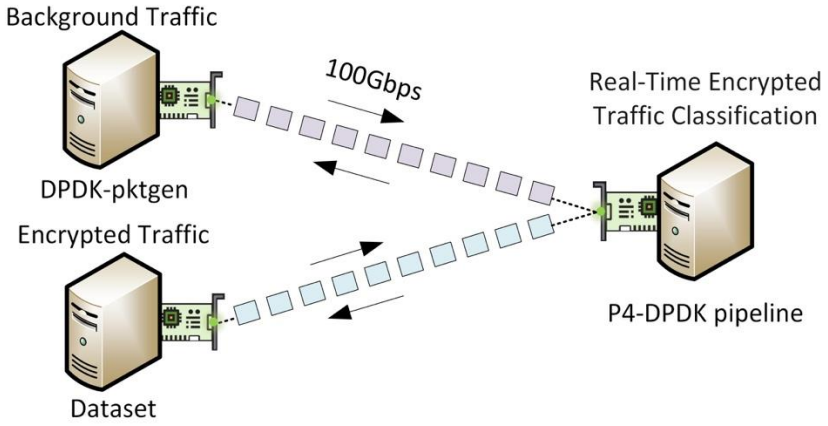
Proposed System

- The control plane is responsible for training the model offline.
- The system uses Random Forest to classify Encrypted Traffic.
- The control plane converts trained models into classification rules for deployment in the data plane.
- The data plane is responsible for extracting the features, implementing the trained model, and storing the results.

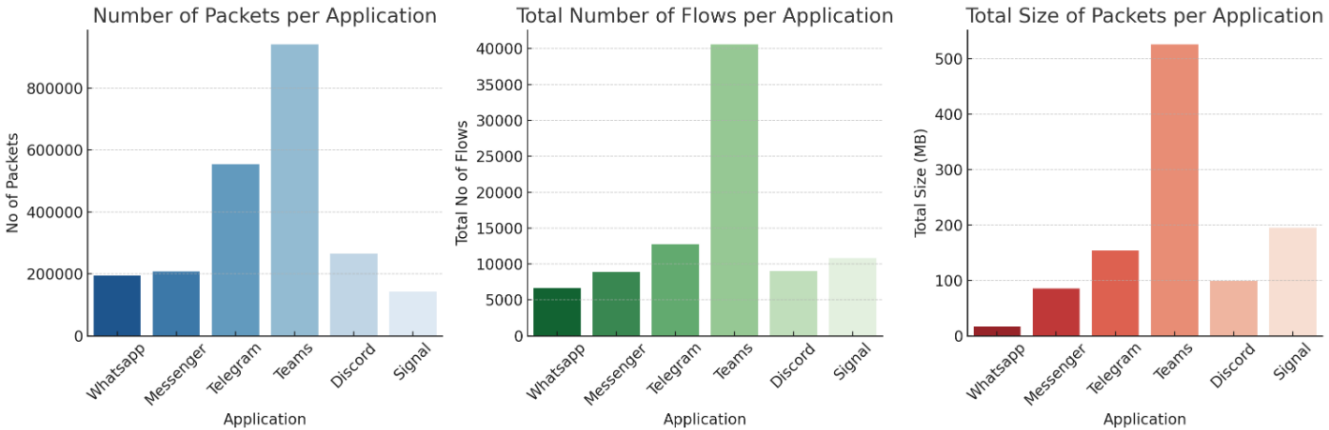


Experiment Topology

- The topology consists of two nodes, equipped with an NVIDIA ConnectX-6 NIC (100Gbps). Nodes are deployed on FABRIC[1].
- DPDK-pktgen[2] and Encrypted traffic belong to a single node.
- DPDK-pktgen[2] is utilized for generating background traffic, while encrypted traffic is produced by replaying packets from the dataset.



Dataset Used for Experiment

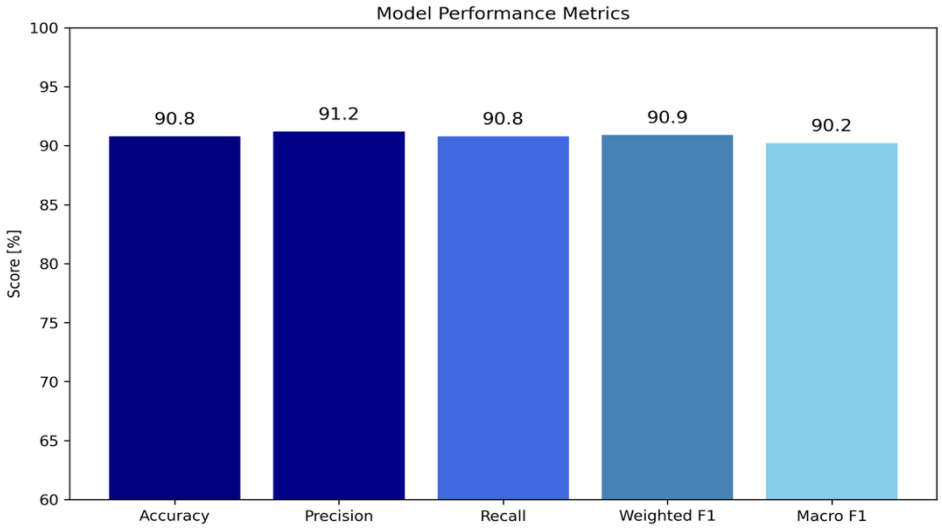


- The dataset used for this project is NIMS IMA[1]
- The number of samples per class clearly shows that the data is not well-balanced.

1. Zolboo Erdenebaatar, Riyad Alshammari, Nur Zincir-Heywood, Marwa Elsayed, Biswajit Nandy, Nabil Seddigh, January 23, 2023, "Encrypted Mobile Instant Messaging Traffic Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/aer2-kq52.

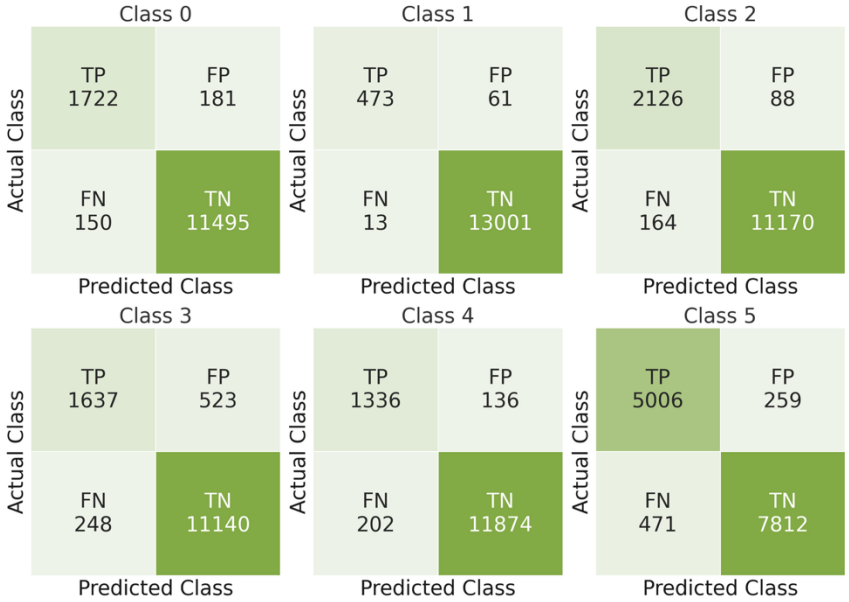
Performance Results of the Model

- Accuracy, precision, and recall are closely aligned, ensuring balance.
- The performance is consistent across different class distributions.
- High precision indicates fewer false positives in classification.
- Strong recall minimizes false negatives for better detection.
- Weighted F1 score confirms consistent performance across classes.
- Macro F1 ensures fairness in minority class predictions.



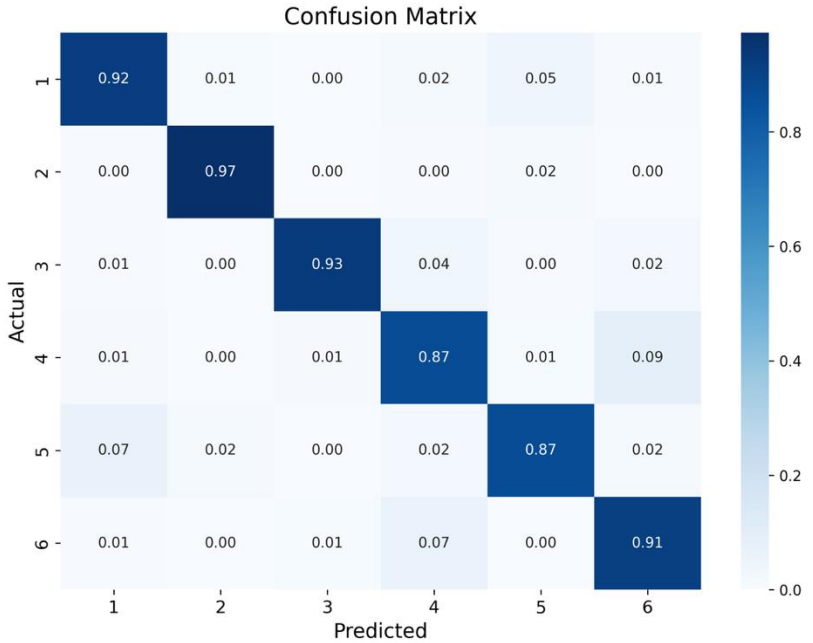
Evaluation Results on Individual Classes

- Model performance varies slightly across different class distributions.
- Balanced confusion matrices suggest no major overfitting issues.
- True negatives confirm correct rejection of non-matching classes.
- True positives are high, indicating strong classification accuracy.



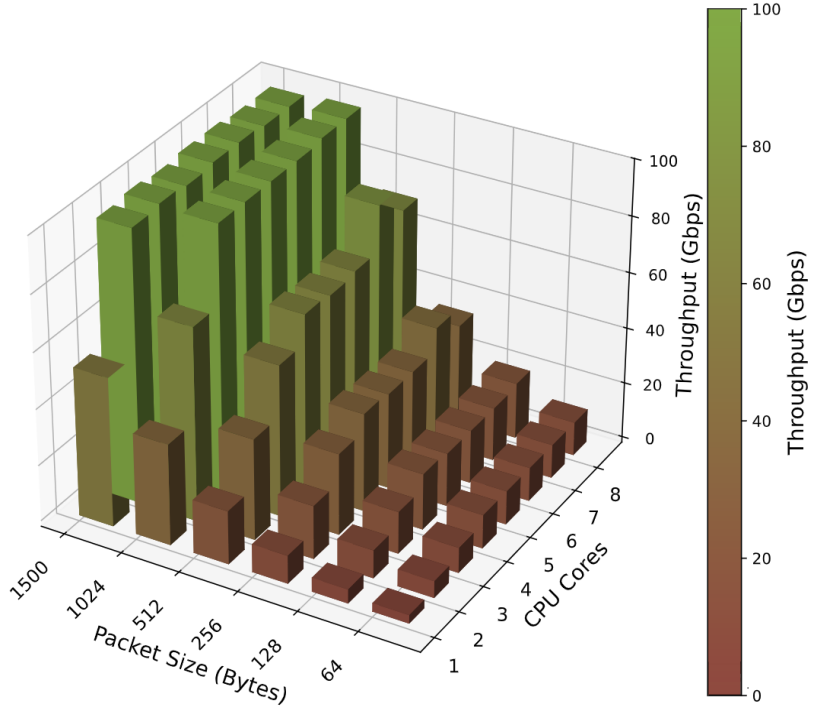
Accuracy Results Across the Classes

- The model's high accuracy and balanced performance make it reliable for practical applications.
- The confusion matrix suggests the model is not overfitting and maintains accuracy across different inputs.
- Class 2 - (WhatsApp) has the best precision.
- Misclassifications are minimal, ensuring the model doesn't frequently mistake one class for another.

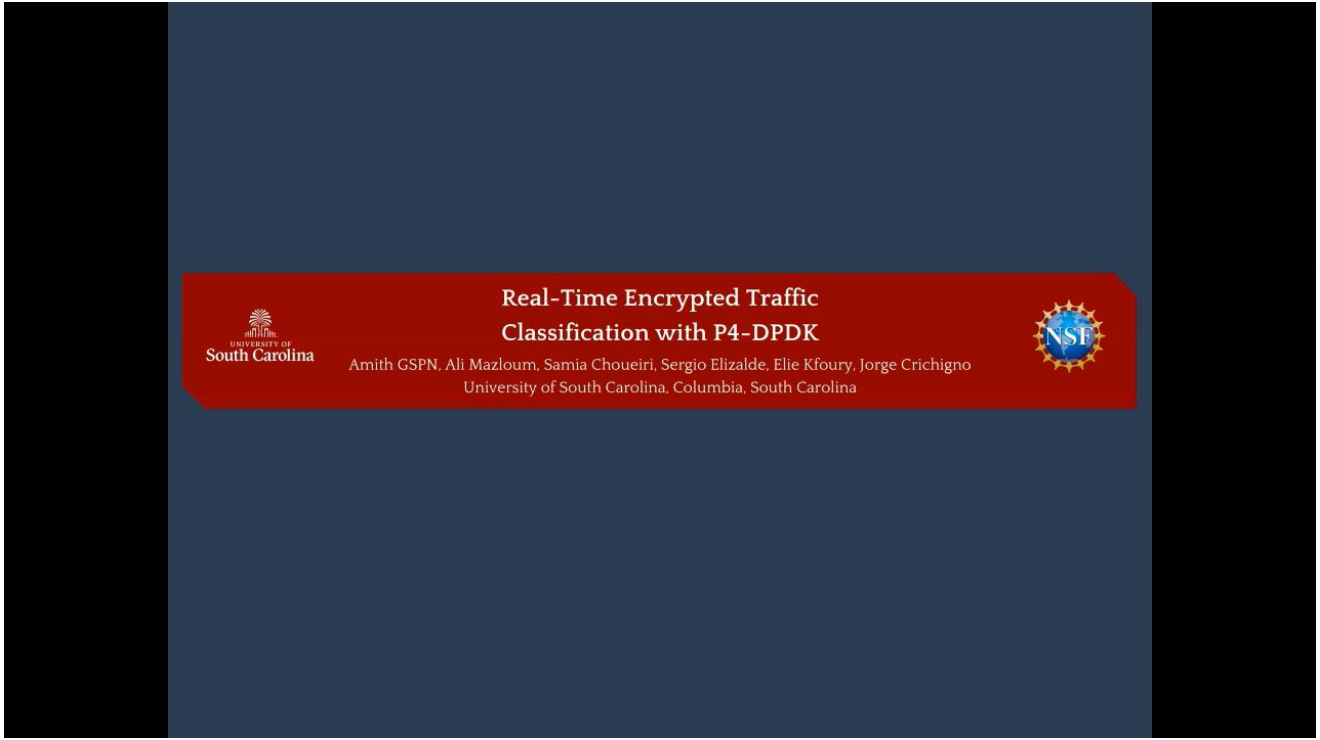


Throughput Test Results

- Throughput Increases with an increase in the CPU Cores.
- Packet sizes of 1024-1500 bytes show significantly better throughput than smaller packets.
- Despite increasing CPU cores, smaller packets (64-128 bytes) exhibit limited throughput improvements.



Demo


University of South Carolina
Real-Time Encrypted Traffic Classification with P4-DPDK
Amith GSPN, Ali Mazloum, Samia Choueiri, Sergio Elizalde, Elie Kfoury, Jorge Crichigno
University of South Carolina, Columbia, South Carolina
NSF

Conclusion

- This paper presents a packet-processing scheme for end devices.
- The scheme enables programmers to write applications using a high-level language (P4) and obtain high performance (DPDK).
- Using a Real-Time Encrypted Traffic Classification as an example, the paper showcases features to help balance compromising resources:
  - P4-DPDK enables high-speed encrypted traffic classification by leveraging programmable data plane processing, ensuring minimal latency.
  - Increasing throughput by using multiple CPU cores
- Future work could explore improving the classification accuracy, experimenting with datasets featuring a much larger number of classes.
- Develop real-time anomaly detection for encrypted traffic patterns.

Acknowledgement

- This work was supported by the U.S. National Science Foundation (NSF), under awards 2346726 and 2403360. The authors would also like to acknowledge the FABRIC team.

For additional information, please refer to
https://research.cec.sc.edu/cyberinfra/

Email: amithgspn@sc.edu, {elizalds,
Choueiri, amazloum, ekfoury}@email.sc.edu, jcrichigno@cec.sc.edu

UNIVERSITY OF
SOUTH CAROLINA

THE LINUX FOUNDATION