Vessels: Efficient and Scalable Deep Learning Prediction on Trusted Processors

Kyungtae Kim*, Chung Hwan Kim°, Junghwan John Rhee¶, Xiao Yu§, Haifeng Chen§, Dave (Jing) Tian*, Byoungyoung Lee˥

*Purdue University, °University of Texas at Dallas,
¶University of Central Oklahoma,
§ NEC Labs America, ˥Seoul National University

PURDUE UNIVERSITY UT DALLAS UCO NEC Laboratories America Relentless passion for innovation SEOUL NATIONAL UNIVERSITY

Deep Learning Systems in the Cloud

- Deep Learning (DL) systems are widely used
  - Face recognition, intelligent personal assistants, object detection, etc
- Cloud platforms are popular for running DL services
  - Cost reduction, scalability, flexibility
  - MLaaS competition: AWS, Google Cloud, MS Azure, etc

Data Breaches and Untrusted Environment in the cloud

- Sophisticated data breaches in the cloud

- Emergence of cyber attacks stealing/ manipulating ML data
    - Model inversion attacks
    - Neural net. Trojan'ing

- Untrusted computing environment
    - Cloud provider and tenants
    - Compromised VM/container instances

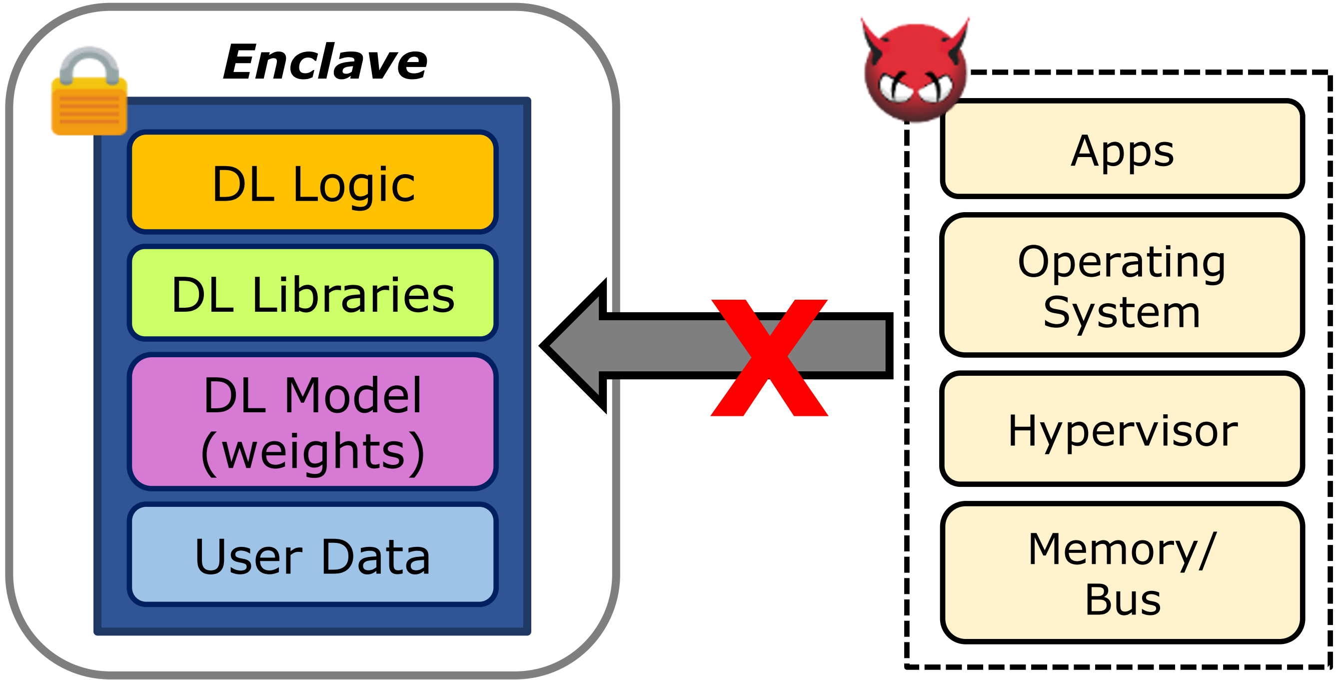

Top 5 Cloud Security related Data Breaches!

Posted By **Naveen Goud**

The Year 2017 has so far witnessed some data slip-ups from the worlds top cloud storage providers and the details are as follows-

Protecting Deep Learning using Intel SGX

- Intel Software Guard Extensions (SGX)
  - *Enclave*: a hardware-protected memory region
  - Memory protection against privileged software (e.g., Hypervisor, OS)
  - Can protect ML program, model, and data from attacks
  - Availability in the cloud: IBM Cloud Computing Shield, MS Azure Confidential Computing



Limitation of SGX

- Runtime overhead remains to be a problem
  - EPC (Enclave Page Cache): 128 MB (~92MB after metadata for all enclaves)
  - DL with SGX: Large memory, 4-23x prediction time in Linux
  - *Frequent EPC page swapping* leads to significant performance degradation
  - EPC thrashing: *EPC memory is shared by all enclaves*

Models	# Layers	Peak mem size (bytes)	Execution time (sec)	
			Non-SGX	SGX
AlexNet	13	274 M	1.03	21.56
ResNext152	204	566 M	6.88	29.23
DenseNet201	304	376 M	2.54	18.82
InceptionV3	145	337 M	8.34	38.63
VGG16	24	1,121 M	7.43	117.79

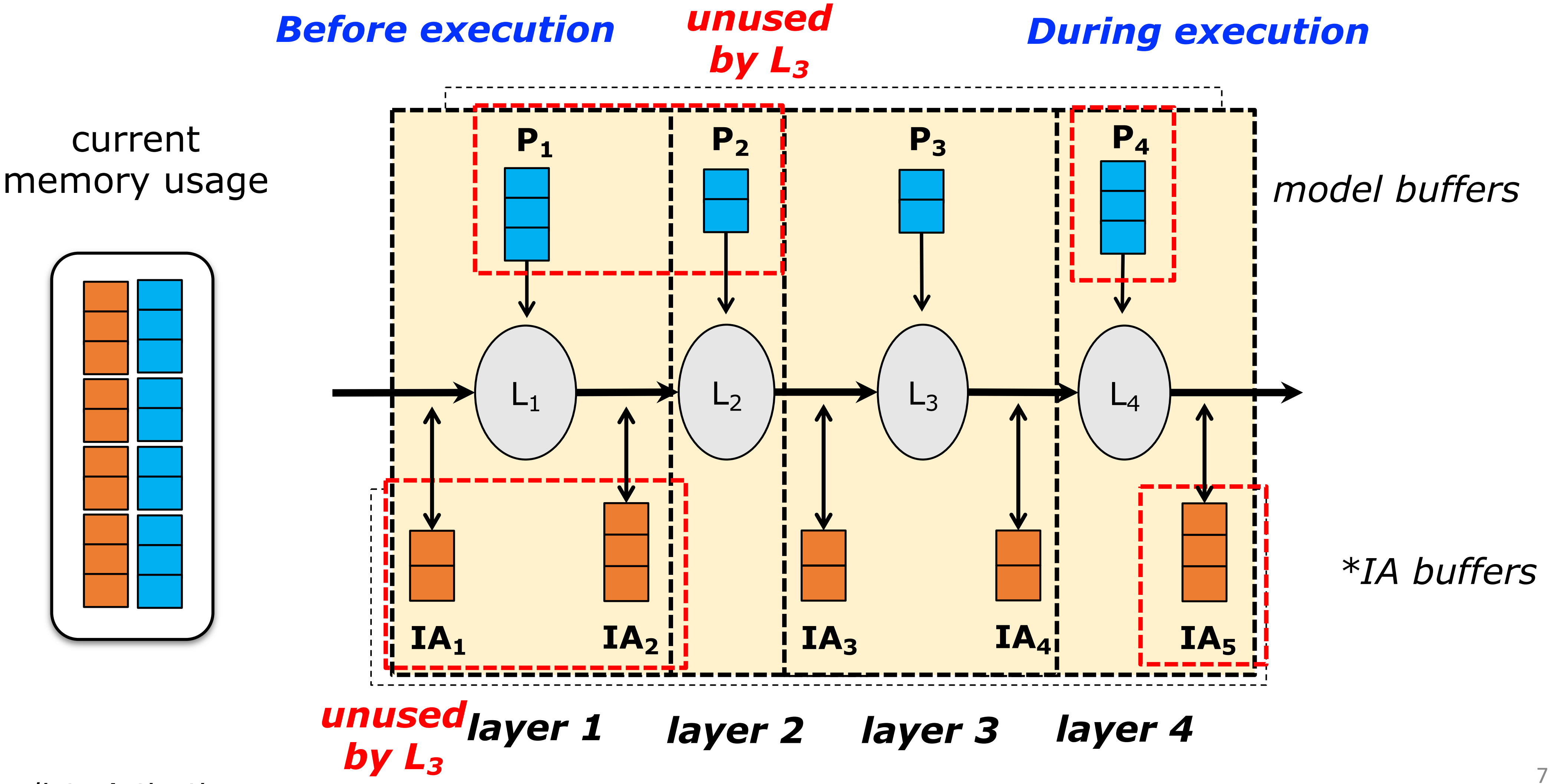
(>128M)

Vessels: Efficient and Scalable DL with SGX

- Goal: *Minimize memory footprint of DL*
  - Reduce the memory size of DL prediction enclave
  - Efficiency and scalability close to non-SGX prediction
  - Target : CPU-only SGX computation, DL prediction system
  - No accuracy loss (vs. compression and pruning)

- Our approach
  - 1. Profile memory usage of DL: Redundancy discovered
  - 2. DL framework with SGX: Optimize based on profiling
    - *Memory usage planning*
    - *On-demand parameter loading*
    - *EPC-aware scheduling*

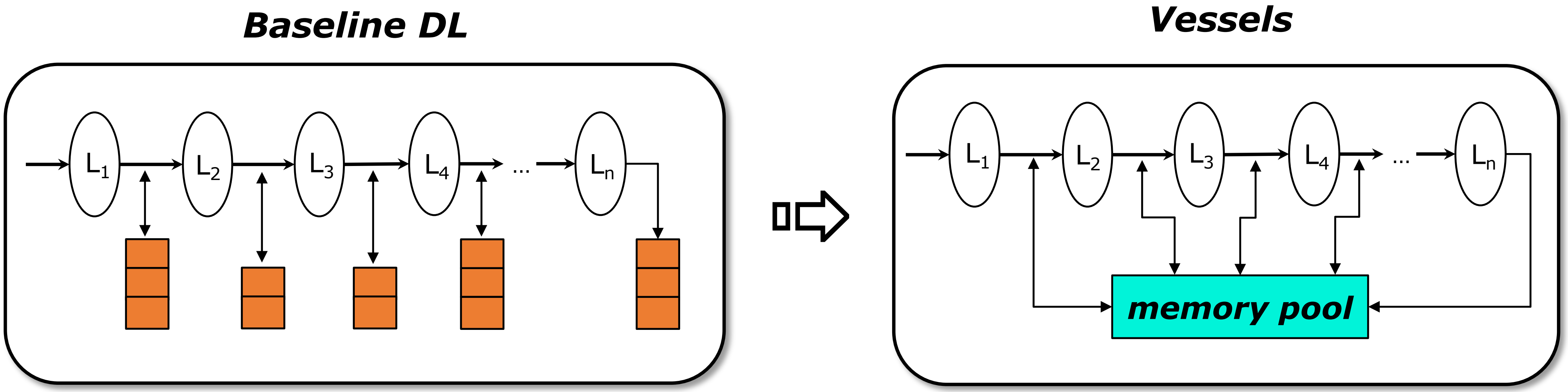
Memory Usage Profiling of DL Prediction



*IA = Intermediate Activations

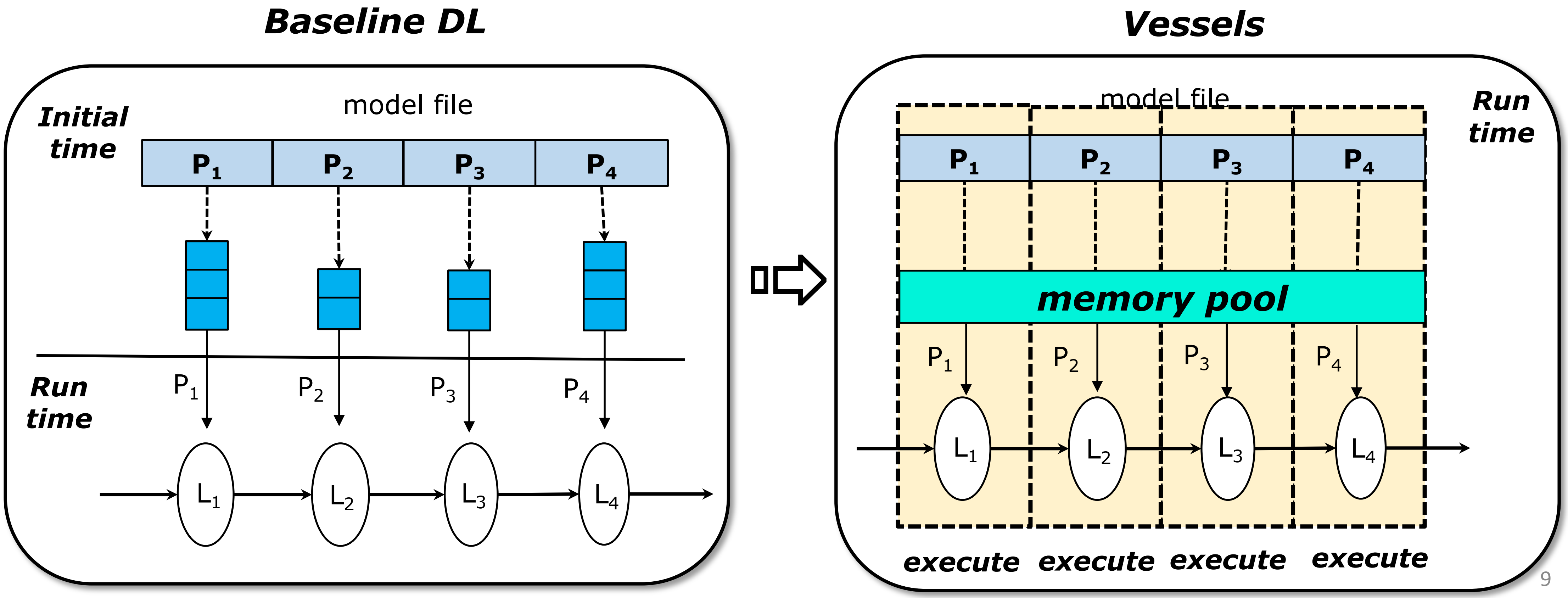
Memory Usage Planning

- *Optimized memory pool*
  - One single memory buffer shared by all layers
  - Recycled for high memory reusability
  - Reduces page swapping significantly (minimal changes to working set)



On-demand Parameter Loading

- Model params are *partially loaded* in an on-demand fashion
  - Model parameters are assigned to a specific layer
  - Identify the *in-file* location of the model parameters for each layer



EPC-aware Prediction Scheduling

- Concurrent enclaves with multiple cores?
  - A production DL system: receives a large number of prediction requests
  - EPC is not scalable

- Enclave scheduling with an *EPC-commit upper bound*
  - Create a new enclave *only if* the memory usage *will not* exceed the bound
  - Estimate the EPC usage *before* launching it
  - Avoid EPC thrashing
  - Requests are added into a FIFO queue if it has to wait

Evaluation

- Implementation
  - Darknet DL framework
  - Docker-based
- Environmental setup:
  - 9 pre-trained DL models (AlexNet, VGG16, etc)
  - Prediction dataset from ImageNet
- Experiments
  - *Single prediction*
  - *Concurrent predictions*

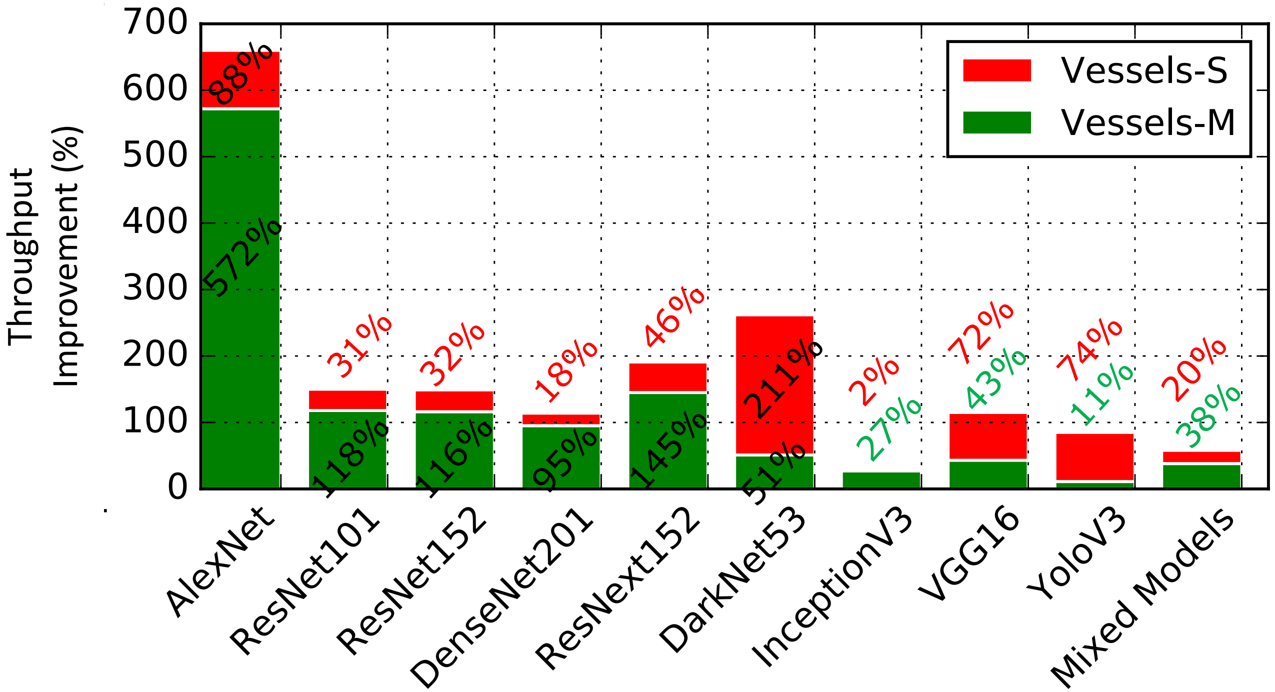
Evaluation – Single Prediction

- Compared to the baseline SGX system

Models	Peak EPC Reduction	Secure Paging Reduction	Latency Improvement
AlexNet	89.5%	100%	94.01%
ResNet101	88.1%	100%	69.81%
ResNet152	91.2%	100%	66.90%
DenseNet201	88.8%	100%	63.02%
ResNext152	89.6%	100%	69.47%
Darknet53	80.6%	100%	69.37%
InceptionV3	85.4%	100%	57.44%
VGG16	86.1%	50.53%	57.44%
YoloV3	73.2%	17.92%	18.89%

Evaluation – Concurrent Predictions

- Based on # of processed requests for 100 minutes compared to baseline
  - Vessels-S: with EPC-aware scheduling, Vessels-M: EPC usage optimization only
  - Improvement : 131% for Vessels-M, 195% for Vessels-S (on average)



Conclusion

- Systematic study on EPC usage of current DL prediction systems and Discovery of the inefficiency
- Vessels: Efficient and scalable DL prediction with full SGX protection
- 90% EPC footprint reduction per enclave and 195% higher throughput with concurrent enclaves
- No functionality or accuracy loss

Thank you