www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Generating and Exploiting Deep Learning Variants to Increase Heterogeneous Resource Utilization in the NVIDIA Xavier

Roger Pujol[†¶], Hamid Tabani[†], Leonidas Kosmidis[†], Enrico Mezzetti[†], Jaume Abella[†], Francisco J. Cazorla[†]

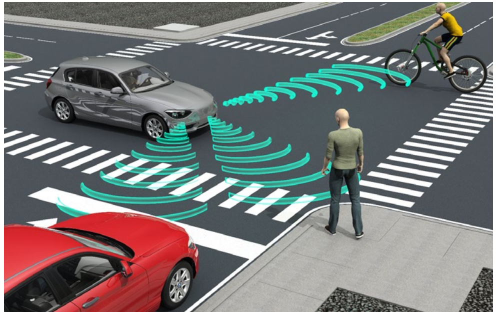
† Barcelona Supercomputing Center
Centro Nacional de Supercomputación

¶ UNIVERSITAT POLITÈCNICA DE CATALUNYA

31st Euromicro Conference on Real-Time Systems (ECRTS'19)
July 12, 2019

Deep-Learning (DL) based algorithms

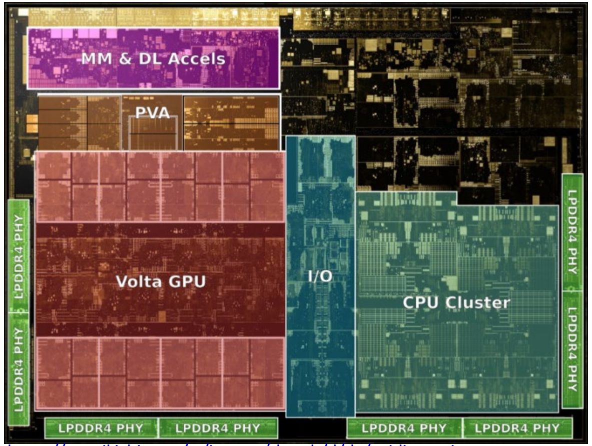
- **Heavily used in critical systems in areas like robotics & autonomous driving (AD)**
  - Vision (object detection and tracking)
  - Trajectory Prediction
  - ...
- **Benefits**
  - Higher-accuracy than traditional algorithms
  - Some problems only solvable with DL approaches
- **Challenges**
  - Unprecedented performance demands in critical systems
  - Tens of tera operations per second!



https://www.rdmag.com/article/2018/01/rise-autonomous-vehicles-planning-deployment-not-just-development

GPUs

- **GPUs are at the forefront of the computing solutions for DL**
  - They are already under evaluation by OEMs/TIER1
- **Modern GPUs**
  - Offer a powerful set of accelerating computing elements (CEs)
  - Offer massive and flexible computation capacity
- **NVIDIA AGX Xavier SoC**
  - CPU
  - GPU Regular cores (GPUrc)
  - GPU Tensor Cores (GPUtc)
  - NVIDIAL Deep-Learning Accelerator (NVDLA)



https://en.wikichip.org/w/images/thumb/d/da/nvidia_xavier_die_shot_%28annotated%29.png/900px-nvidia_xavier_die_shot_%28annotated%29.png

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Modern GPU – Modern DL Libraries Mismatch

- **Modern SoC offer a variety of CEs**
  - CPUs, GPUrc, GPUtc, …
- **DL libraries**
  - Used in many modules → several instances run in parallel
  - Mostly exploit a single CE (GPUrc)
- **Huge loss of performance capacity and flexibility!**

- **Our view**
  - The **ability** to run DL-based variants, each using different CEs …
  - **Improves** timing and throughput
  - **Pays off the extra effort** required to implement those different variants

Our work

- **Analysis**
  - Active DNN instances during execution of Apollo AD software
- **Develop DNN Variants**
  - Running DNN variants on different CEs of NVIDIA Xavier (CPU, GPU, DLA)
- **Timing Characterization**
  - In-depth analysis of the different variants of DL libraries
- **Scheduling multiple DNN instances**
  - Modelling a multicore cyclic executive scheduler as a LP problem

Outline

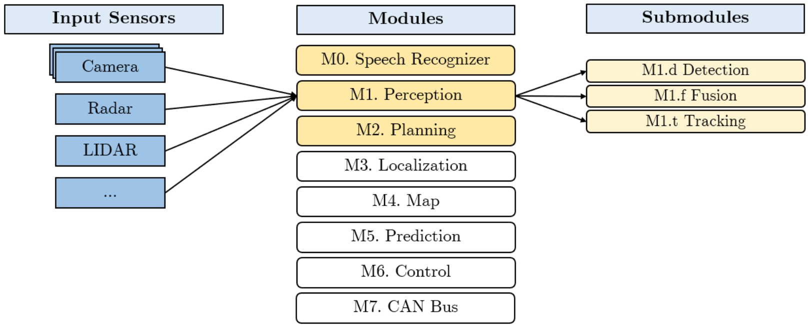
- Motivation
- Background: Apollo and Xavier
- Analysis on the number of active DNN instances
- Timing Characterization
- Scheduling multiple DNN instances
- Conclusion

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Introduction to the Apollo AD framework

Apollo \1

- **One of the most sophisticated open-source projects implementing an entire AD software stack**

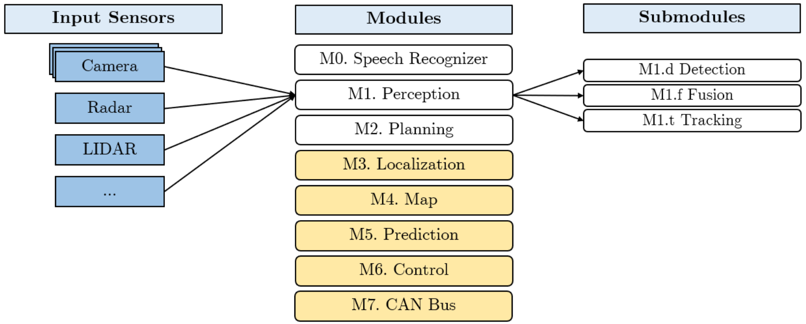


- **M0. Speech recognizer** processes the voice-based commands and transmit them to the control unit
- **M1. Perception** identifies the surrounding area around the autonomous car
  - **M1 .d** The **detection** submodule is in charge of detecting obstacles and objects from different sensors
  - **M1.f fusion** takes the results of all detected objects from different sensors and combines them by a sensor fusion algorithm
  - **M1.t Tracker** follows the detected objects and matches them with the previously detected objects
- **M2. The Planning** plans the spatio-temporal trajectory for the vehicle to take

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Apollo \2

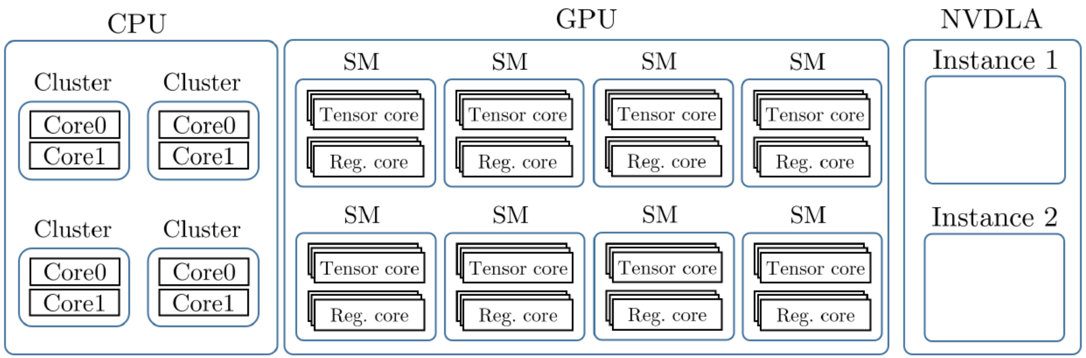
- **One of the most sophisticated open-source projects implementing an entire AD software stack**



- **M3. Localization** leverages information received from different input sensors to estimate vehicle position
- **M4. The Map** provides ad-hoc structured information regarding the roads
- **M5. Prediction** anticipates the future motion trajectories of perceived obstacles/objects
- **M6. Control** generates control commands such as accelerating/braking and steering
- **M7. CAN Bus** passes all the control commands to the vehicle hardware

CEs in the Jetson Xavier

1. **CPU cores**
   - 8x Carmel ARMv8.2 processors
   - 4 clusters, each with 2 cores
2. **GPU regular cores**
   - 512 regular cores, 8 SMs
   - Volta Architecture
3. **GPU tensor cores**
   - 64 Tensor cores, 8 cores per SM
   - To accelerate large matrix operations
4. **NVDLA**
   - NVIDIA Deep Learning Accelerators
   - Specialized for deep learning acceleration



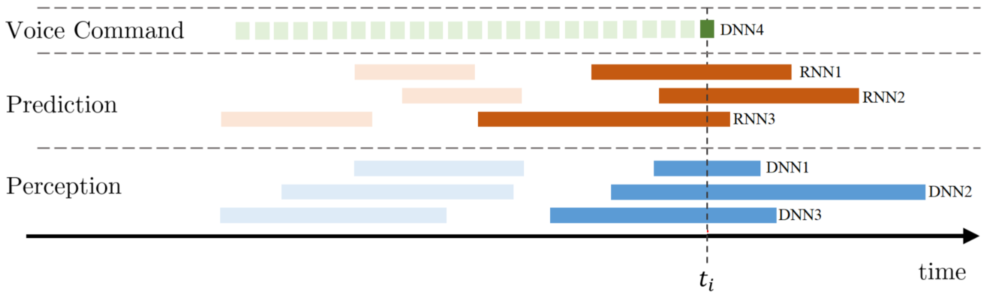
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Analysis on the number of active DNN instances

DNN/RNN active instances

- **We used real traces from an AD car running Apollo**
- **Observations:**
  - DNN/RNN have different durations and periods
  - In Apollo, up to 7 DNN/RNN instances are run concurrently
    - Small DNNs, such as the speech DNN, has small duration and short periods
    - Other DNN/RNNs have longer durations and longer periods

	Deep Learning Software	Description
M0. Speech Recognition	Voice Command and Control	A DNN-based accurate speech recognition application to process speech commands
M1. Perception	Camera Object Detection	A DNN-based algorithm to identify objects and traffic signals from camera sensors
	LiDAR object Detection	A DNN-based algorithm to identify objects from LiDAR sensors
	Object Tracker	A DNN-based algorithm to track identified objects in consecutive frames
M5. Prediction	Lane sequences (RNN1)	A RNN for lane sequences
	Obstacle status (RNN2)	A RNN for obstacle status
	RNN using RNN1 and RNN3	A RNN to compute the probability of each lane sequence based on RNN1 and RNN2



DNN/RNN active instances: Projection

- **More input sensors**
  - Increase in the number of sensors toward fully AD (level 5)
  - Today, AD cars employ several heterogenous sensors
- **More sophisticated algorithms**
  - To increase the accuracy, larger and more complex DNNs/RNNs are designed
  - DNNs are using more and more layers to improve the accuracy
- **More functionalities**
  - In-cabin features such as gesture control, driver-monitoring systems, etc.
- **Conclusion**
  - All the aforementioned items will be translated into **more computation power** and **more deep learning instances**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Generating DNN Variants

Specialized per-CE Libraries

- **We used optimized libraries to implement the software for each particular CE.**

1. **CPU**
   – We used OpenMP for all the functions running on the CPU cores
2. **GPU Regular Cores**
   – The baseline GPU implementation uses regular cores to run the kernels
3. **GPU Tensor Cores**
   – We have adapted the GPU code to exploit tensor cores
4. **NVDLA**

Example: GPU Tensor Core Implementation

- **Set the Math mode**
- **Some preconditions**
  - Multiples of 4

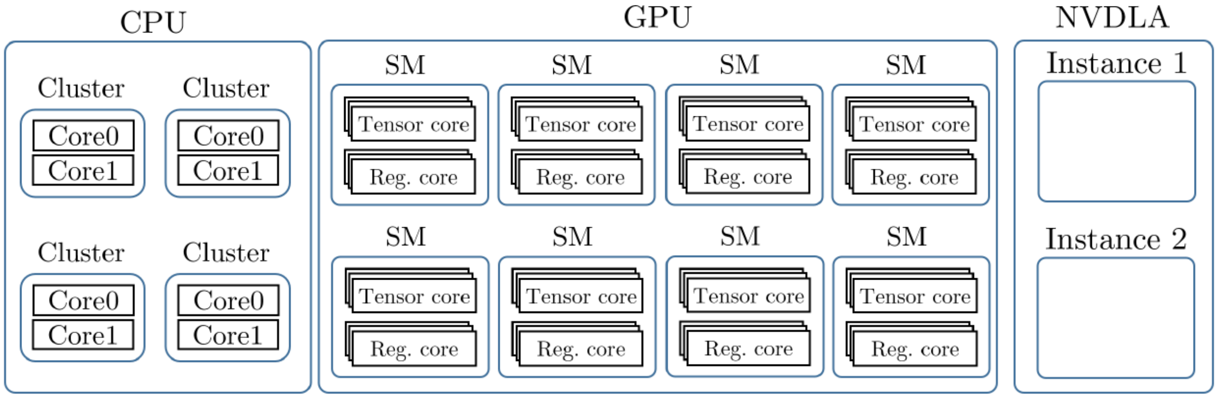
```
void GTCSgemmNN(int M, int N, int K, float ALPHA, float const *A,
int lda, float const *B, int ldb, float BETA, float *C, int ldc)
{
    static int init[16] = {0};              // Vector for initialized handles
    static cublasHandle_t handle[16];       // Vector of actual handles
    int i;
    cudaGetDevice(&i);                      // Get current device
    if(!init[i]) {                          // If not initialized
        cublasCreate(&handle[i]);           // Creates the handle
        init[i] = 1;
    }

    // Set math mode to enable Tensor cores
    cublasSetMathMode(handle[i], CUBLAS_TENSOR_OP_MATH);
    cudaError_t status = cublasSgemm(handle[i],
            CUBLAS_OP_N, CUBLAS_OP_N,       // Select the non-transpose matrices
            N, M, K,                        // Sizes of the matrices
            &ALPHA,
            B, ldb,                         // B and it's leading size
            A, lda,                         // A and it's leading size
            &BETA,
            C, ldc);                        // C and it's leading size
    if (status != cudaSuccess)              // Check if there is any error
        printf("CUDA Error: %s\n",cudaGetErrorString(status));
}
```

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Timing Characterization

The Experiments

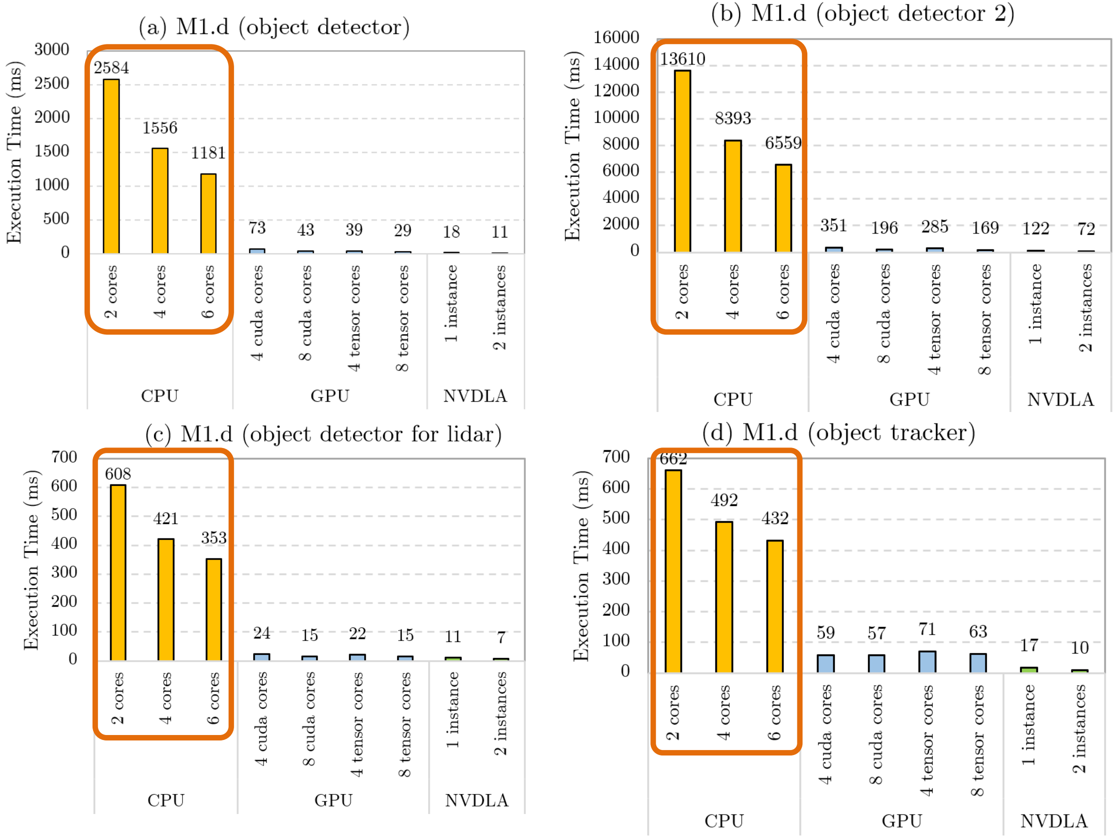
- **Exec time for each DNN/RNN variant for different CEs**
  - CPU Cores (2, 4, 6 cores), GPU RC (4, 8 SMs), GPU TC (4, 8 SMs), NVDLA (1,2)
  - 2 cores are always reserved for managing OS tasks and GPU/NVDLA tasks



CPU	GPU	NVDLA
1 cluster	4 SMs	1 instance
2 clusters	8 SMs	2 instances
3 clusters		

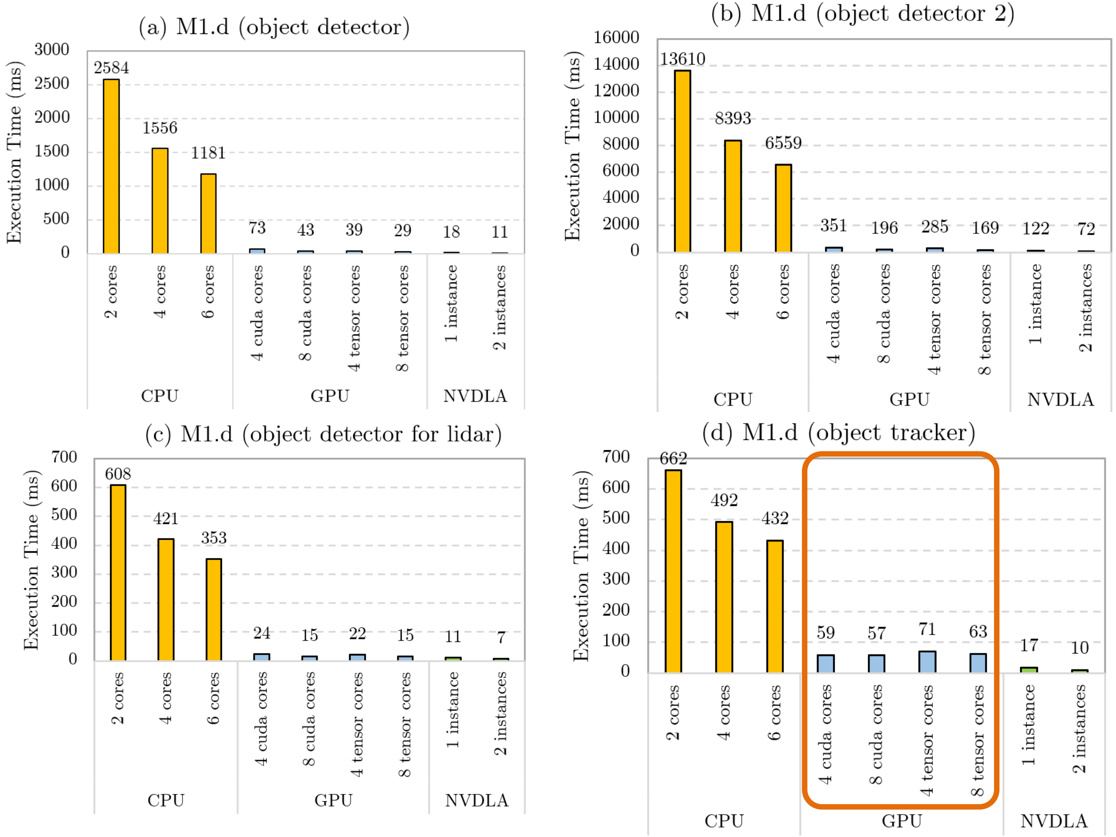
Timing Results for Different Apollo Neural Networks /1

- **Performance improves by increasing the number of CPU cores**



Timing Results for Different Apollo Neural Networks /1

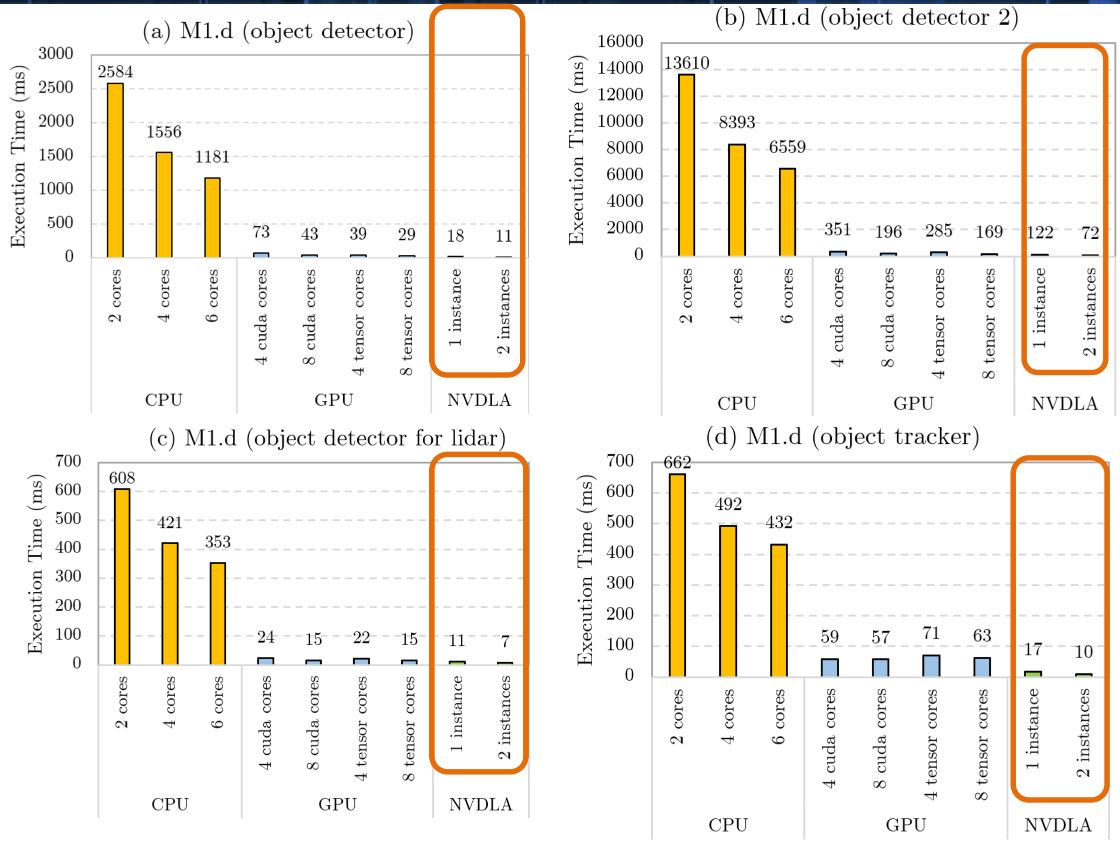
- Performance improves by increasing the number of CPU cores
- **Tensor cores are <u>NOT</u> always providing better performance**



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Timing Results for Different Apollo Neural Networks /1

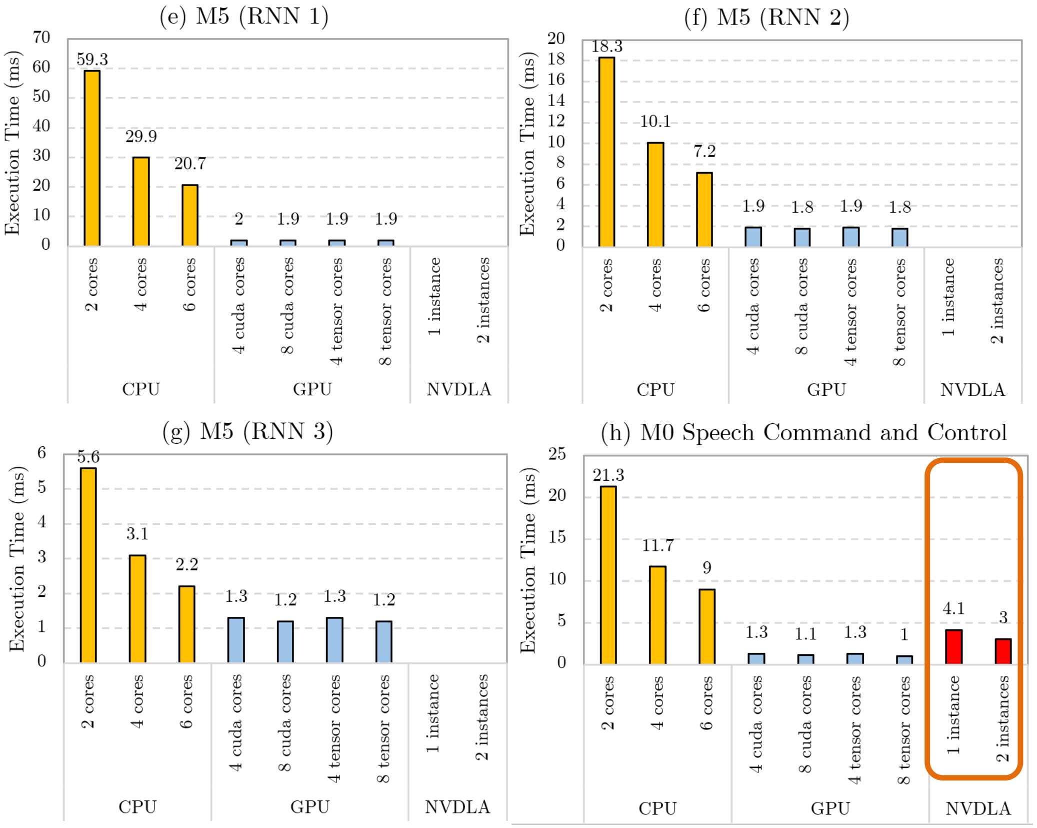
- Performance improves by increasing the number of CPU cores
- Tensor cores are <u>NOT</u> always providing better performance
- **NVDLA provides the best performance for these Apollo modules**



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Timing Results for Different Apollo Neural Networks /2

- **NVDLA provides worse performance in comparison to GPU for small modules such as Speech (due to initialization overhead)**



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Scheduling multiple DNN instances

Exploiting Diversity to Increase Schedulability

- **Platforms supporting diverse CE configurations**
  - Applications timing behaviour largely varies on CE they are mapped to
  - Overall mapping strategy is fundamental for schedulability
- **Multiple DNN instances supporting AD functions in Apollo**
  - Instances can be modelled as (relatively independent) *recurrent applications*
    - Frame rate depending on the frequency at which inputs need to be elaborated
  - Periodic task set to be scheduled on a set of *unrelated* processors
    - System supporting k CE configurations $\mathcal{CE} \coloneqq \{ce_1, \cdots, ce_k\}$
    - $\tau_i \coloneqq (p_i, d_i = p_i, C_i)$
    - $C_i = \{c_{i,1}, \cdots, c_{i,k}\}$ $c_{i,j}$ denoting execution time bound of τ_i on configuration $ce_j \in \mathcal{CE}$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Modeling Schedulability of Multiple RNN/DNN with LP

- **Cyclic-executive static scheduling**
  - Still a preferred solution in several embedded real-time domains
  - DNN/RNN modelled as a set of recurrent activities
  - Mapping strategy that allows all DNNs to complete within their frame
- **Linear Programming model**
  - 0/1 optimization (minimization) problem for the total system utilization
  - <u>Failing to find a solution means the taskset is not schedulable</u>
  - Other optimization criteria may be enforced (with weights)

LP Formulation

- **Instantiation to the Xavier SoC**

$$\mathcal{CE}_{Xavier} \coloneqq \{CPU, GPU^{RC}, GPU^{RC-comb}, GPU^{TC}, GPU^{TC-comb}, GPU^{RC+TC}, NVDLA, NVDLA^{comb}\}$$

- **Boolean decision variables**
  - $|\Gamma| \times |\mathcal{CE}|$ $\mathrm{B}[\tau_i \in \Gamma][ce_j \in \mathcal{CE}]$ representing whether τ_i is mapped to ce_j
- **Objective function**
  - $min \sum_{\tau_i \in \Gamma, ce_j \in \mathcal{CE}} B[\tau_i][ce_j] \times U[\tau_i][ce_j]$
- **Constraints**
  - $|\Gamma|$ constraints to ensure tasks are only mapped to one ce
  - $|\Gamma|$ constraints to ensure tasks will meet their deadlines
  - $|\mathcal{CE}|$ constraints to avoid >100% utilization on each ce
  - Constraints also handle inter-correlations between ces
    - Number of constraints depends on supported TLP

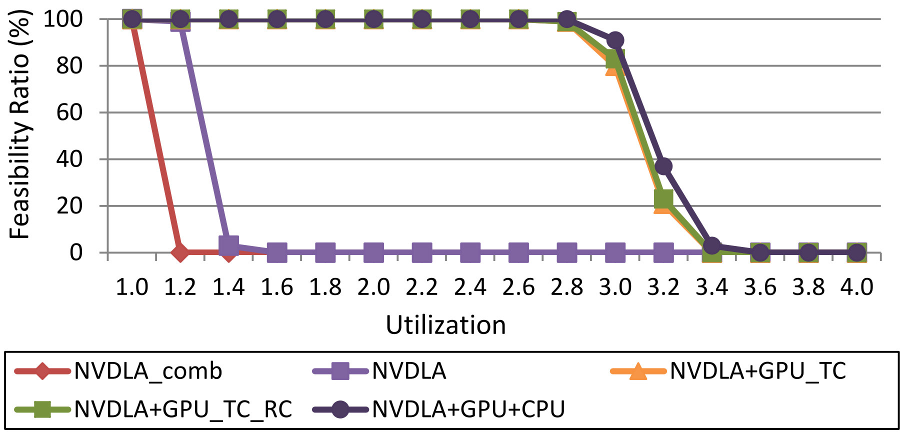
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Experimental objectives and setup

- **Show diverse DNN/RNN implementations allow flexible use of CE**
  - Confirm how this can be leveraged to sustain the schedulability of systems otherwise not schedulable
    - Evaluate increase in ratio of schedulable tasksets
- **Scenario-based evaluation supporting different and flexible use of CE**
  - $\left\langle NVDLA \middle| GPU^{\mathrm{TC}} \middle| GPU^{RC} | CPU \right\rangle, \left\langle GPU^{\mathrm{TC}} \middle| GPU^{RC} | CPU \right\rangle, \langle GPU^{RC} | CPU \rangle$
- **Synthetic task sets generation**
  - For each CE scenario we generated 16,000 synthetic task sets under different overall utilization thresholds
    - Generated by randomly selecting several instances of the diverse DNN/RNN types
    - Utilizations derived from the RNN/DNN timing characterization

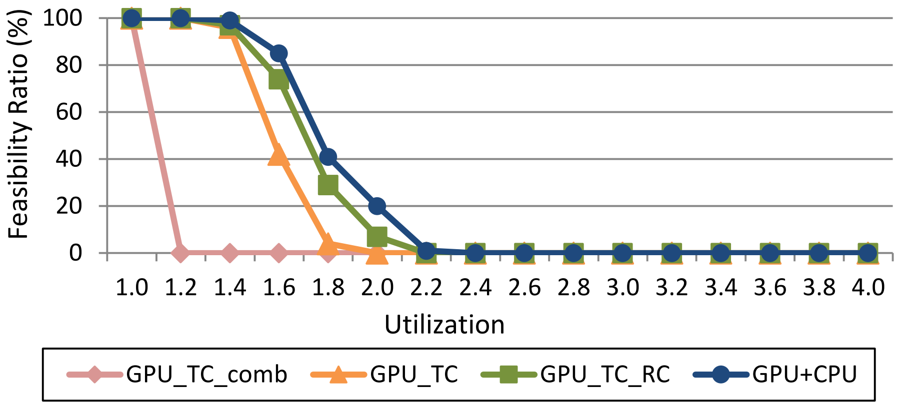
Improved Schedulability with Diverse DNN Variants

- $\langle \mathbf{NVDLA} | \mathbf{GPU}^{\mathbf{TC}} | \mathbf{GPU}^{\mathbf{RC}} | \mathbf{CPU} \rangle$



- **Flexible NVDLA provides better performance**
  - Using 2 NVDLA instances as a cluster does not exploit full parallelism
- **Enabling GPU largely improves over NVDLA alone**
  - DNN/RNNs can be successfully offloaded onto GPU
  - Some DNN are not taking benefit of NVDLA
- **Averaging number of instances in between 12 and 49**

- $\langle \mathbf{GPU}^{\mathbf{TC}} | \mathbf{GPU}^{\mathbf{RC}} | \mathbf{CPU} \rangle$



- **Flexible Tensor Cores improves significantly**
  - Using 8 SMs provides small relative improvement over using just 4
- **GPU Regular cores can still improve over Tensor**
  - Tensor cores are over-specialized
  - Sometimes counter-productive
- **CPUs bring relatively marginal improvement**
- **Averaging number of instances in between 10 and 14**

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Conclusions

Conclusions

- **AD system requires multiple DNN/RNN instances**
  - Supported by powerful accelerating CE in modern platforms
- **CEs can be exploited to meet performance requirements**
  - DNN/RNN need to be tailored to run on multiple CEs
  - Flexible use of CEs allows to successfully support the execution of more instances
- **Supporting different DNN/RNN variants is an enabler for exploiting the diversity of modern accelerators**
- **In this work**
  - Focused on Jetson AGX Xavier as representative AD platform
  - Implemented different variants of the Apollo DNN/RNN to execute on multiple CE
  - Implemented a LP model of a static scheduler to show how tailoring AD functions to different CEs allows to successfully sustain otherwise non-schedulable workloads

Acknowledgements

- This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grant TIN2015-65316-P, the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 772773), and the HiPEAC Network of Excellence. MINECO partially supported Jaume Abella under Ramon y Cajal postdoctoral fellowship (RYC-2013-14717), Enrico Mezzetti under Juan de la Cierva-Incorporación postdoctoral fellowship (IJCI-2016-27396), and Leonidas Kosmidis under Juan de la Cierva-Formación postdoctoral fellowship (FJCI-2017-34095).

www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Generating and Exploiting Deep Learning Variants to Increase Heterogeneous Resource Utilization in the NVIDIA Xavier

Roger Pujol[†¶], Hamid Tabani[†], Leonidas Kosmidis[†],
Enrico Mezzetti[†], Jaume Abella[†], Francisco J. Cazorla[†]

† Barcelona Supercomputing Center
Centro Nacional de Supercomputación

¶ UNIVERSITAT POLITÈCNICA DE CATALUNYA

31st Euromicro Conference on Real-Time Systems (ECRTS'19)
July 12, 2019