Extending the Amalthea model to introduce hardware heterogeneity

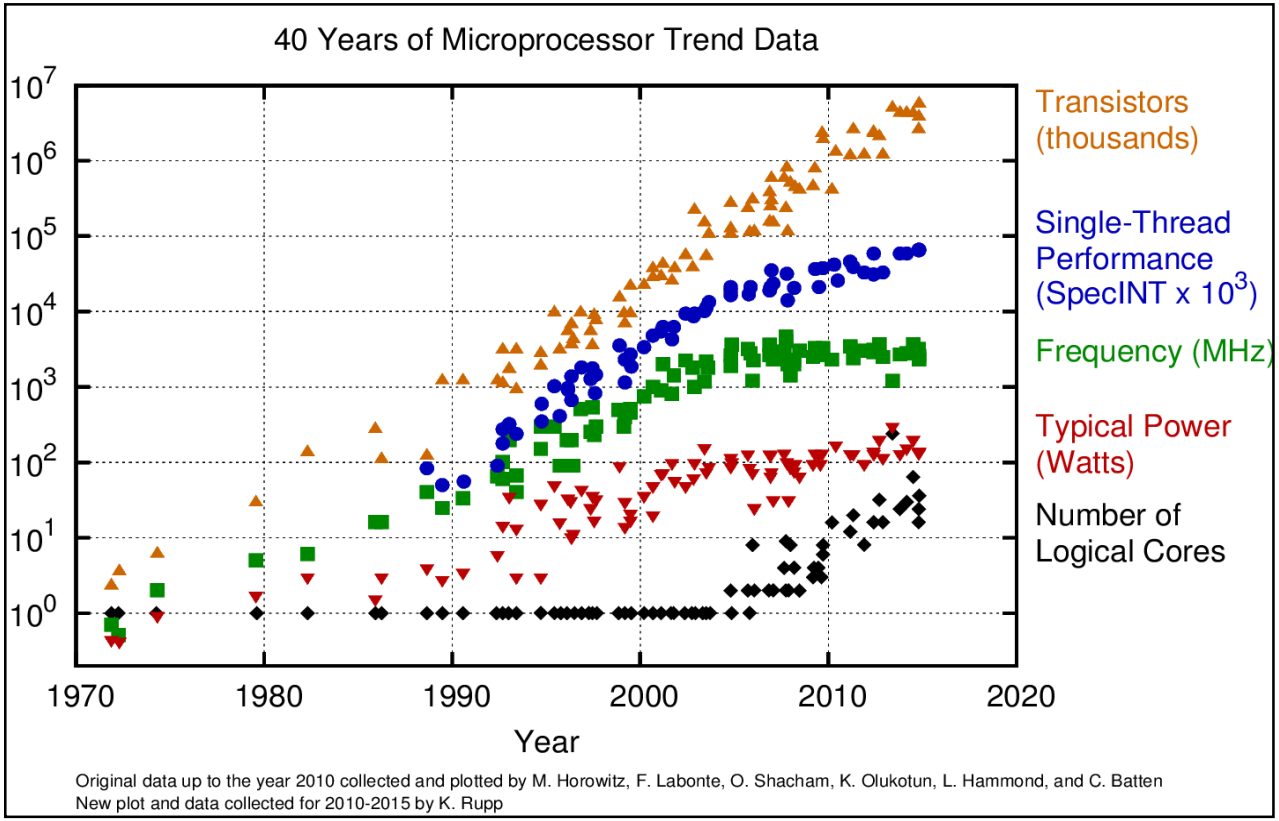
Paolo Burgio
paolo.burgio@unimore.it

HERCULES

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement: 688860

Next-generation, SWaP-efficient systems…

- Dennard's scaling can't keep the pace of computation power demand
- Also for embedded systems
- The rise of many cores!



…also in automotive!

hindustantimes

india world sports cities fifa world cup opinion entertainment lifestyle education photos videos board results 2018 •••

HOME > TECH

Acer launches new NVIDIA Tesla GPUs-powered servers in India

The server can host up to eight NVIDIA Tesla V100 32GB SXM2 GPU accelerators, wherein every GPU pair includes one Periphe Component Interconnect (PCIe) slot for high-speed interconnect.

TECH Updated: Jun 26, 2018 11:25 IST

Indo Asian News Service
Indo Asian News Service

just auto Search just-auto Go

NEWS & ANALYSIS IN-DEPTH FEATURES DATABANK COMPANIES SECTORS REGIONS RESEARCH STORE QUBE ABOUT NEWSLETTER SIGN-UP

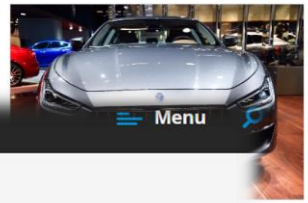
just-auto home > News & insights > News

FRANKFURT - Maserati switches to EPS to enable new ADAS functions

By Graeme Roberts | 13 September 2017 Font size

Maserati has switched its Ghibli, Quattroporte and Levante models to electric power steering (EPS) for the 2018 model year, enabling new driver assistance systems (ADAS).

"EPS has replaced hydraulic power steering on both sedans, as well as on the model retaining Maserati



Most popular news

1 Redesigned Ford Foc wagons
2 BMW stresses comm
3 UK car manufacturin
4 Toyota Motor Europe than boss

DIGITAL TRENDS Menu

CARS

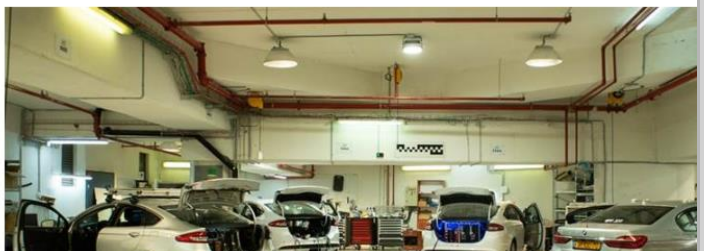
Volkswagen, Bosch, and Nvidia team up to bring autonomous cars one step closer

By Ronan Glon — P

GLOBES
ISRAEL'S BUSINESS ARENA

Front

Mobileye to test autonomous cars in Fiat hometown Turin



REUTERS World Business Markets Politics TV

Detained In Myanmar | Energy & Environment | Brexit | North Korea | Charged: The Future of Autos | Future of Money | Breaki

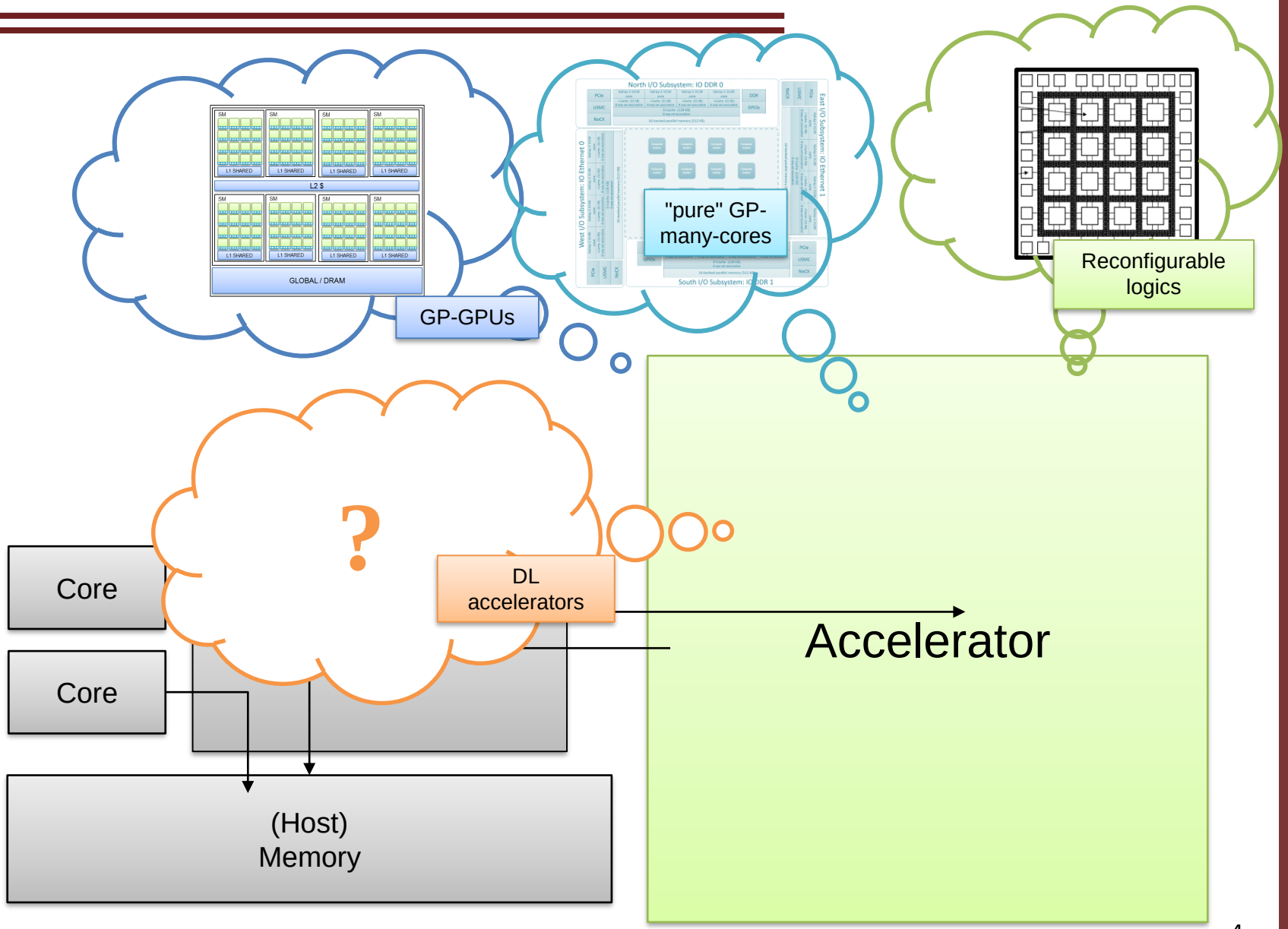
BUSINESS NEWS JUNE 27, 2017 / 7:53 AM / A YEAR AGO

Volvo and Autoliv team up with Nvidia for self-driving cars

Reuters Staff 2 MIN READ

STOCKHOLM (Reuters) - Volvo Cars and Swedish car safety supplier Autoliv (ALV.N) have signed a deal with U.S. firm Nvidia Corp (NVDA.O), best known for its graphics technology in computer

Many-core accelerators



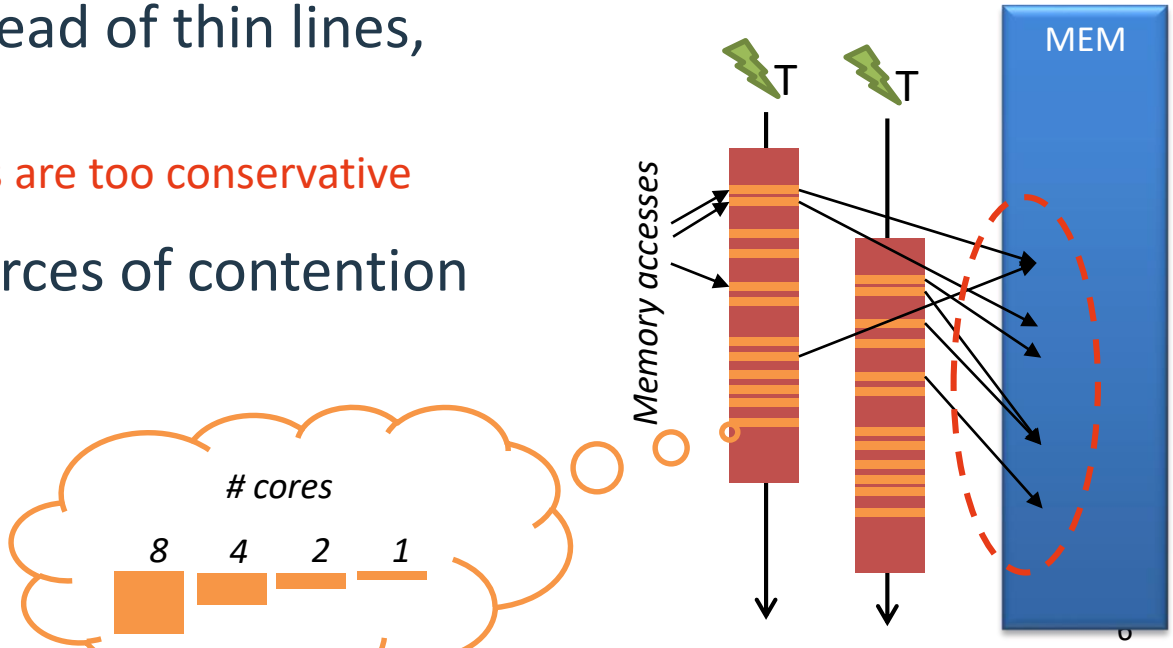
Problems

(..or opportunities?)

Issue #1 - The importance of memory

Beyond traditional predictability

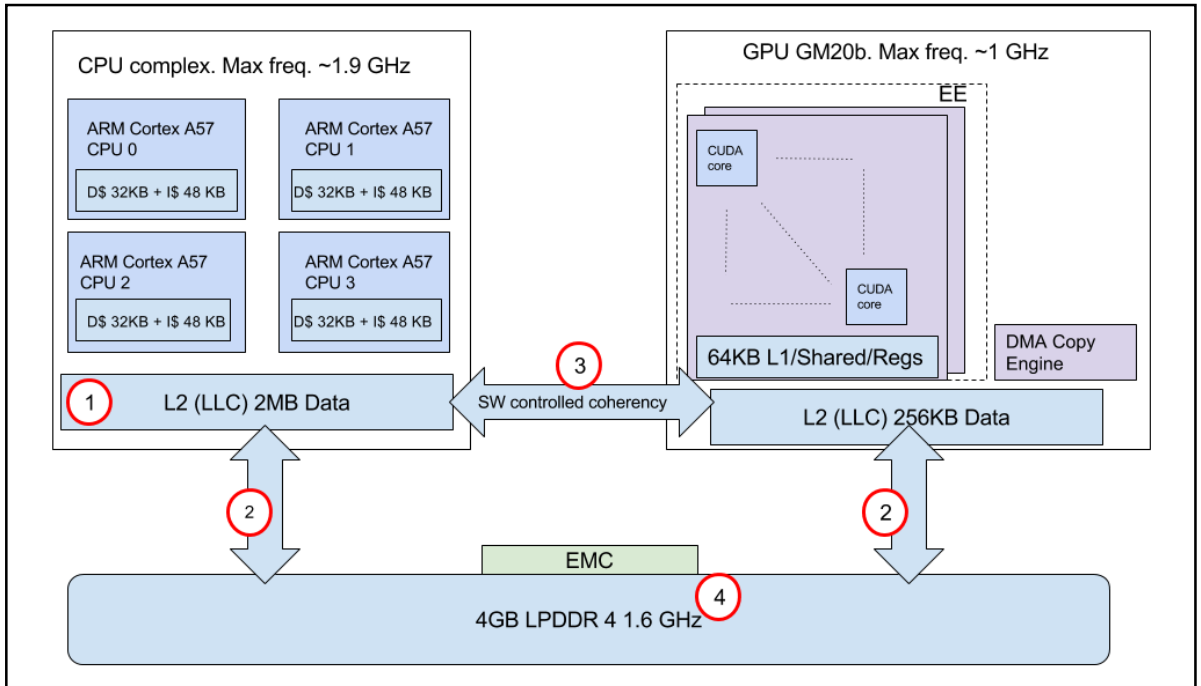
- More degrees of "freedom"
  - Shared resources (e.g., memory, SSDs, IOs, caches..)
  - The complexty of analysis grows exponentially w/number of cores
- Mem accesses: instead of thin lines, thick bars
  - Traditional techniques are too conservative
- Heterogeneous sources of contention



Example: NVIDIA Tegra X2

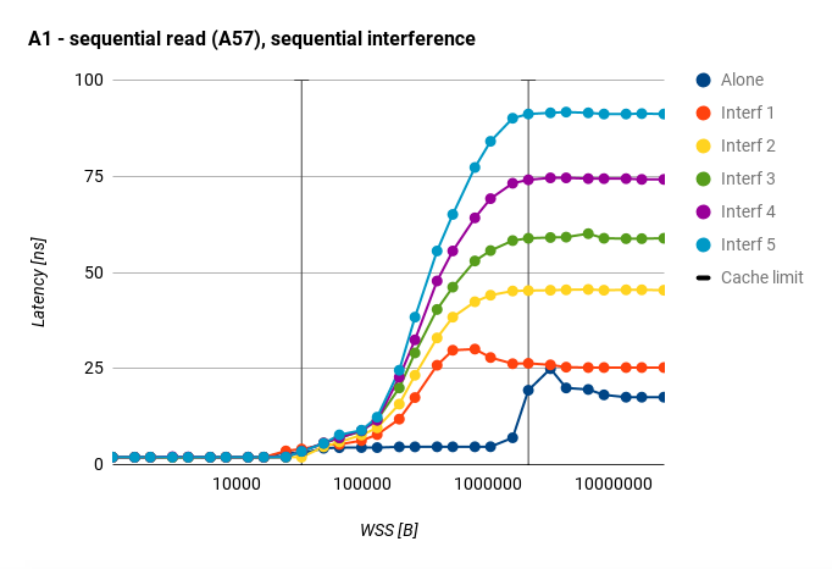
- Shared memory between CPU/GPU complex
  - "Unified Virtual Memory"
  - Unlike traditional "discrete" GPU systems

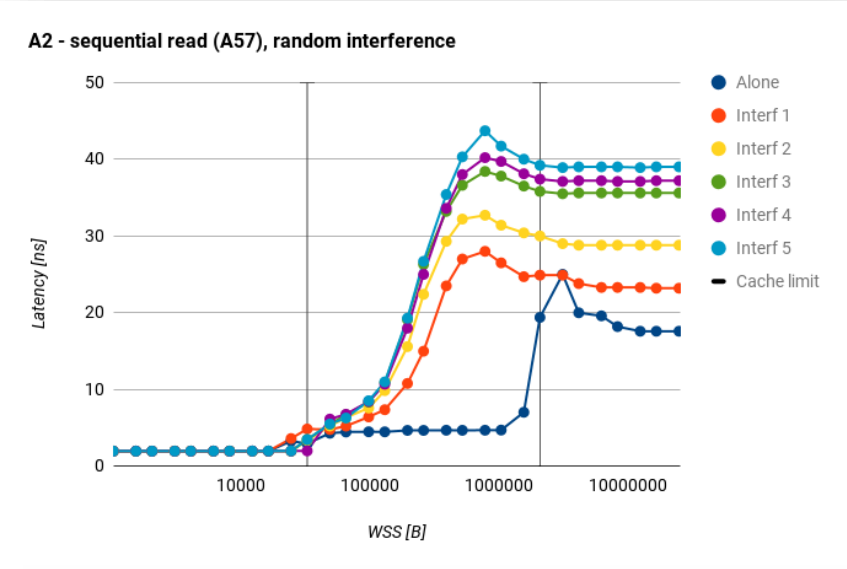
1 Notable contention points in memory hierarchy

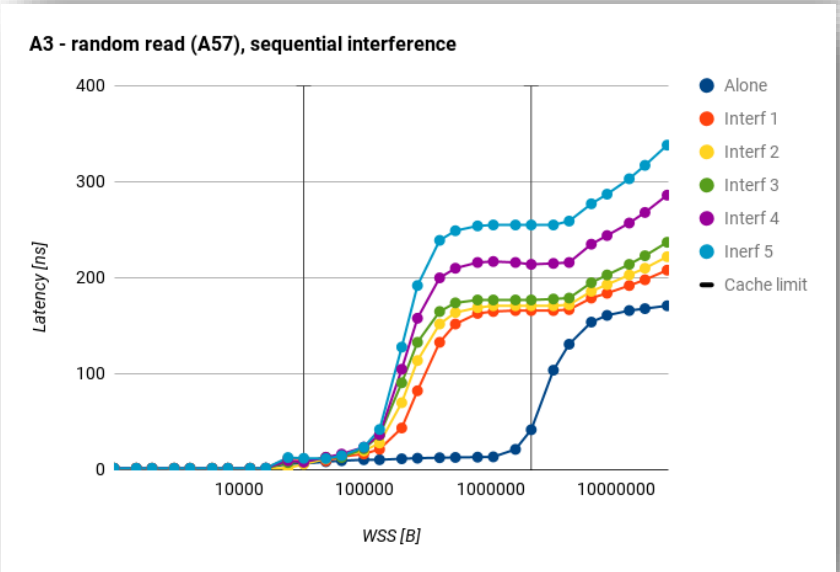


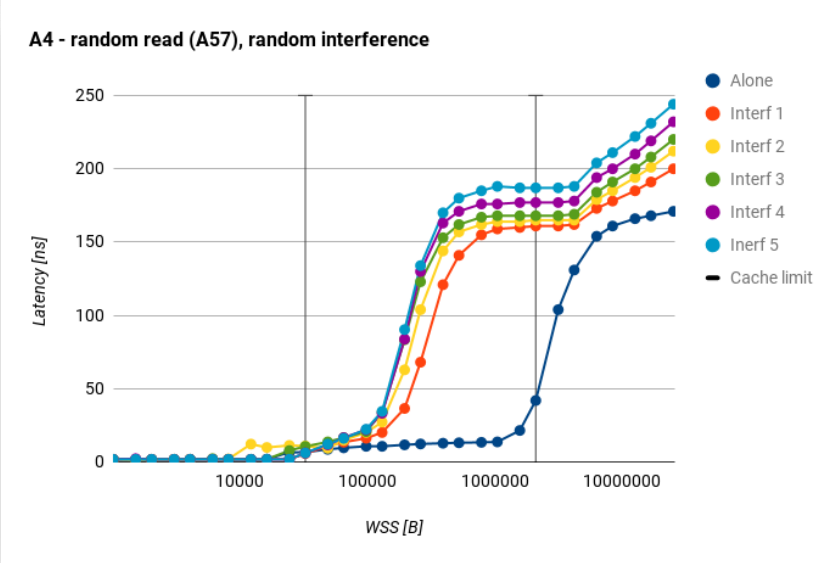
Tegra X2 – A57 - Test 'A'

2017 paper
@ ETFA



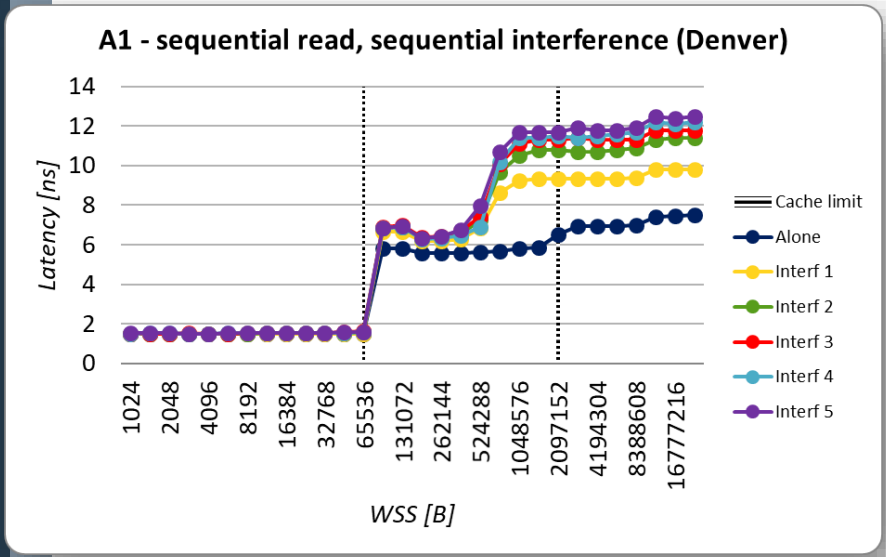


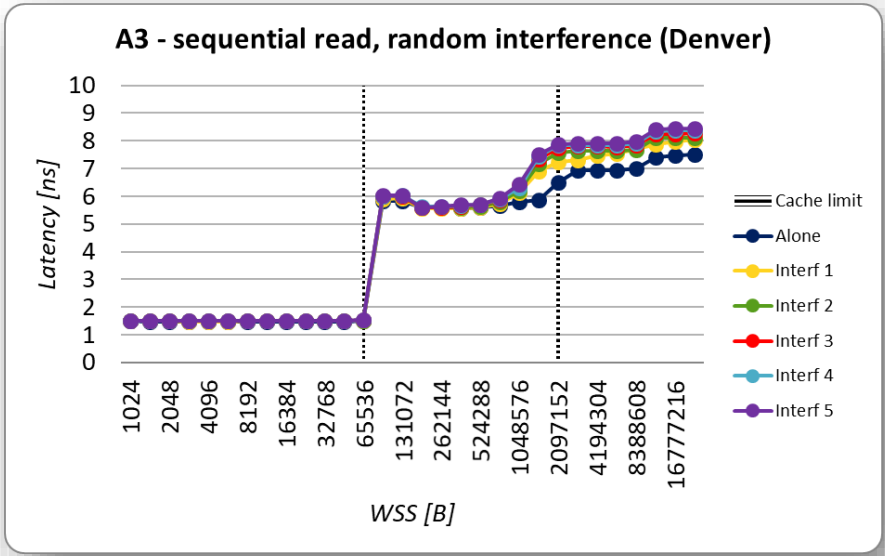


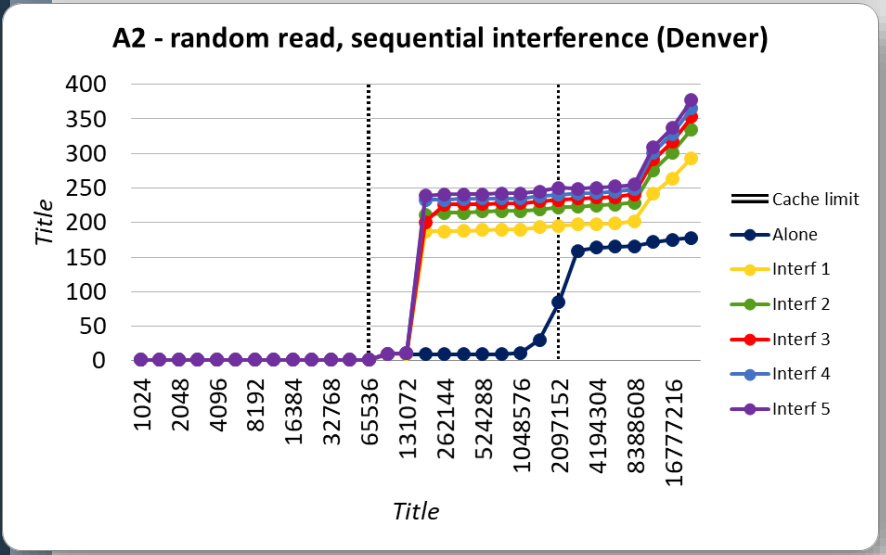


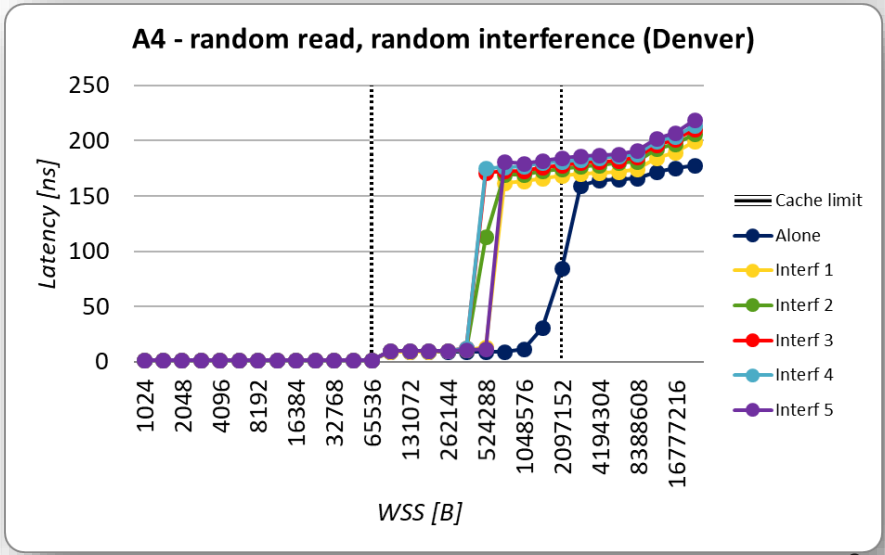
Tegra X2 – Denver - Test 'A'

2017 paper
@ ETFA



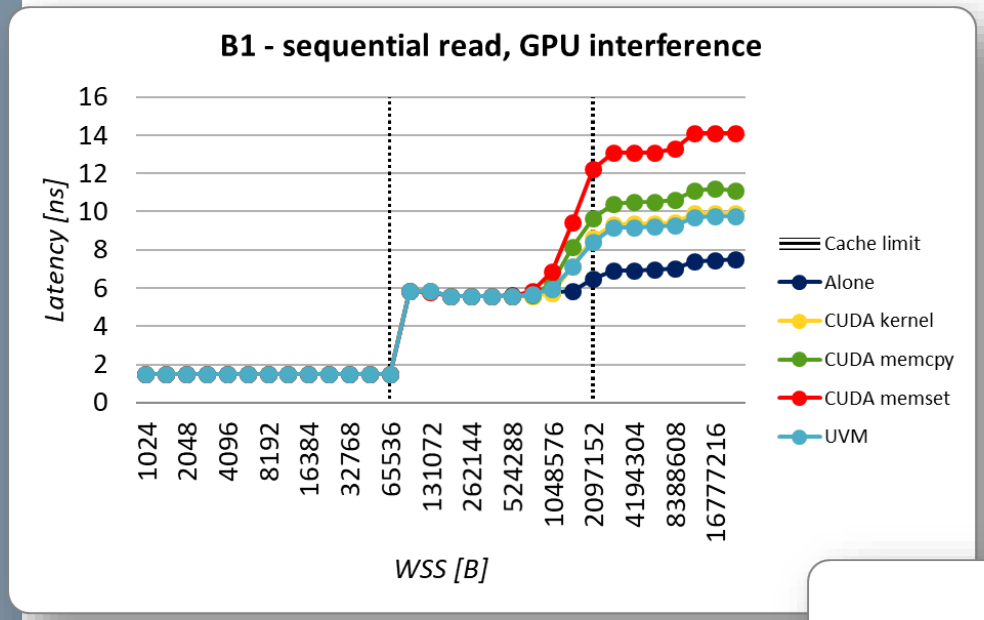


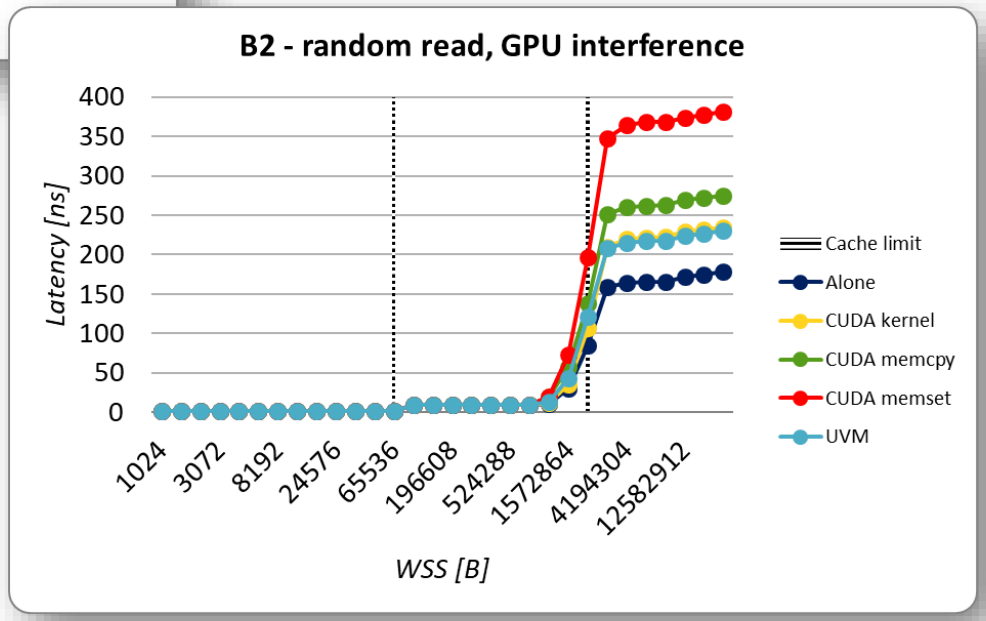




Tegra X2 - Test 'B'

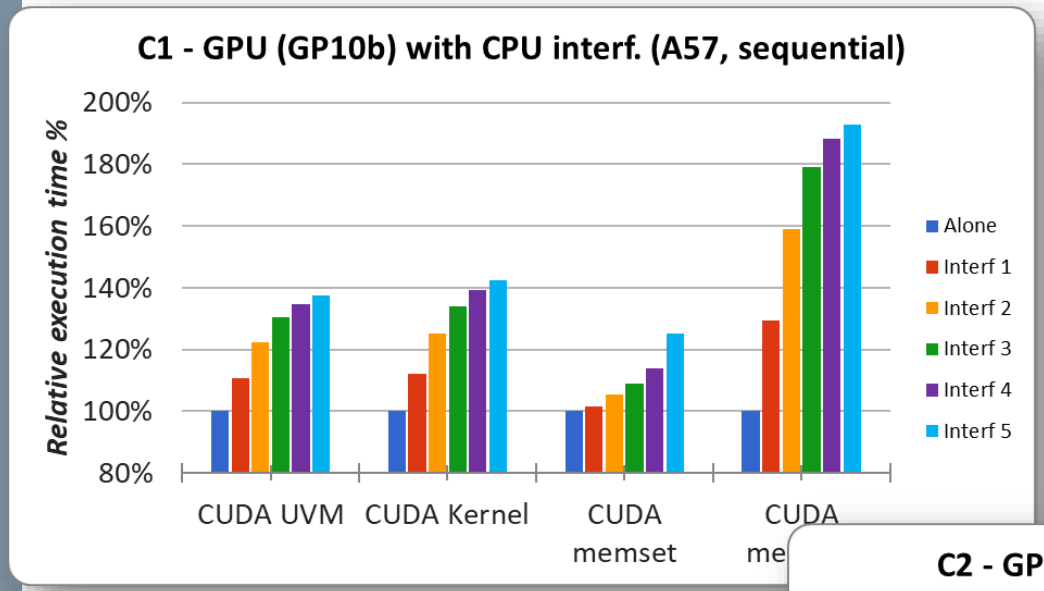
2017 paper
@ ETFA

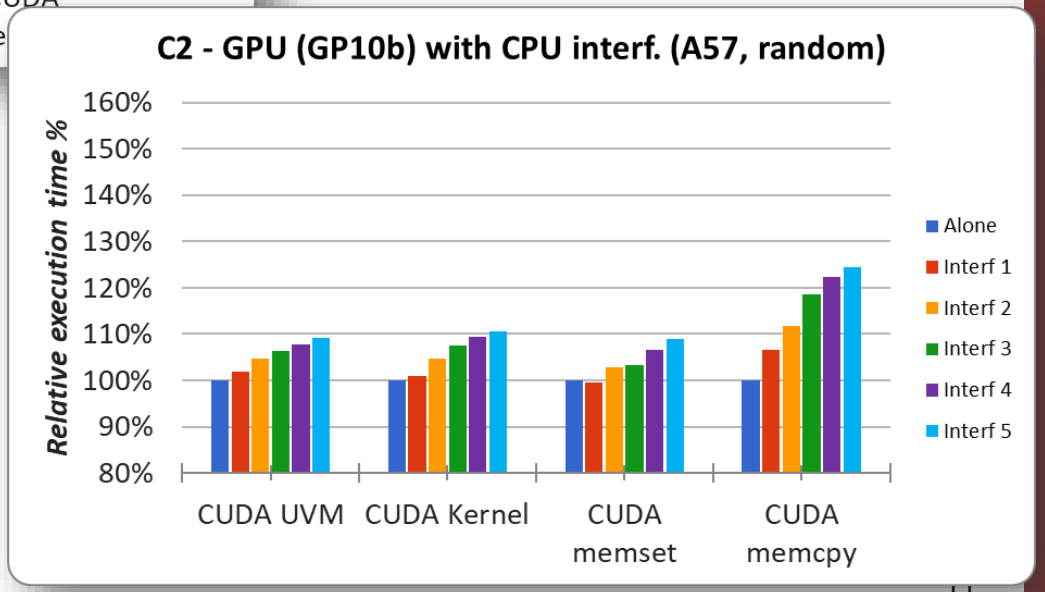




Tegra X2 – A57 - Test 'C'

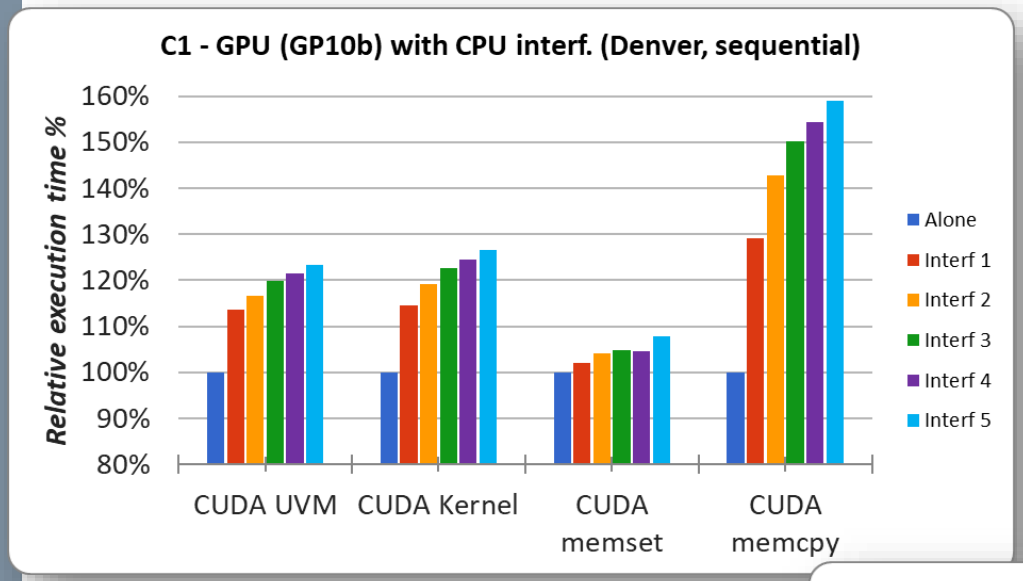
2017 paper
@ ETFA

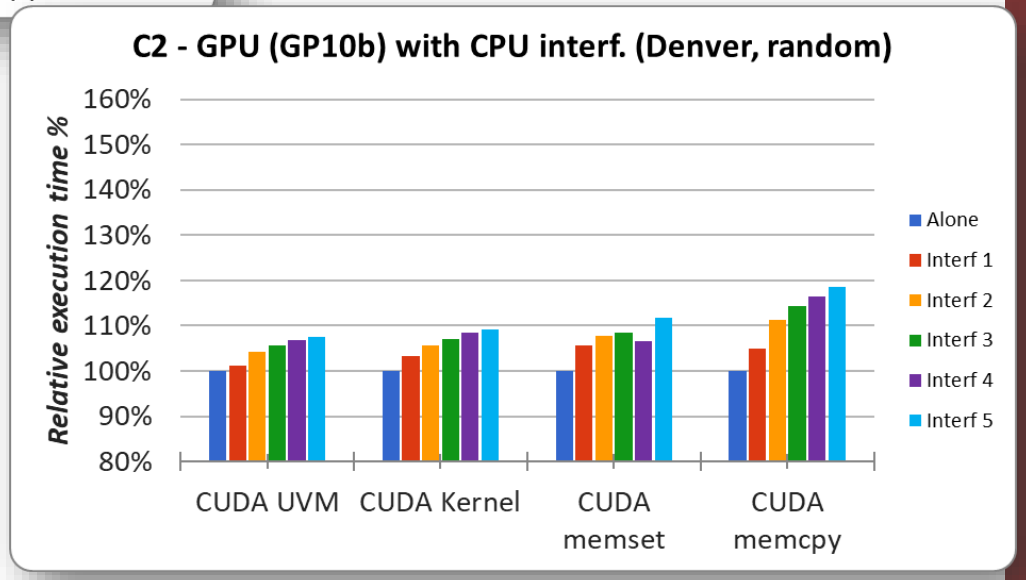




Tegra X2 – Denver - Test 'C'

2017 paper
@ ETFA





(Issue #2) - Offload-based execution models

What parallel programmers do

- Exploit maximum parallelism
- Modern applications are complex! (multi-level, irregular parallelism)
- Explicit shared-memory programming on NUMA hierarchies

An issue….

(Issue #2) - Offload-based execution models

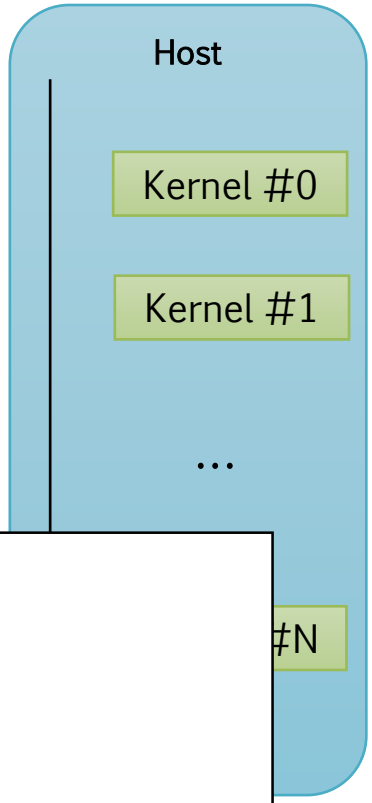
What parallel programmers do

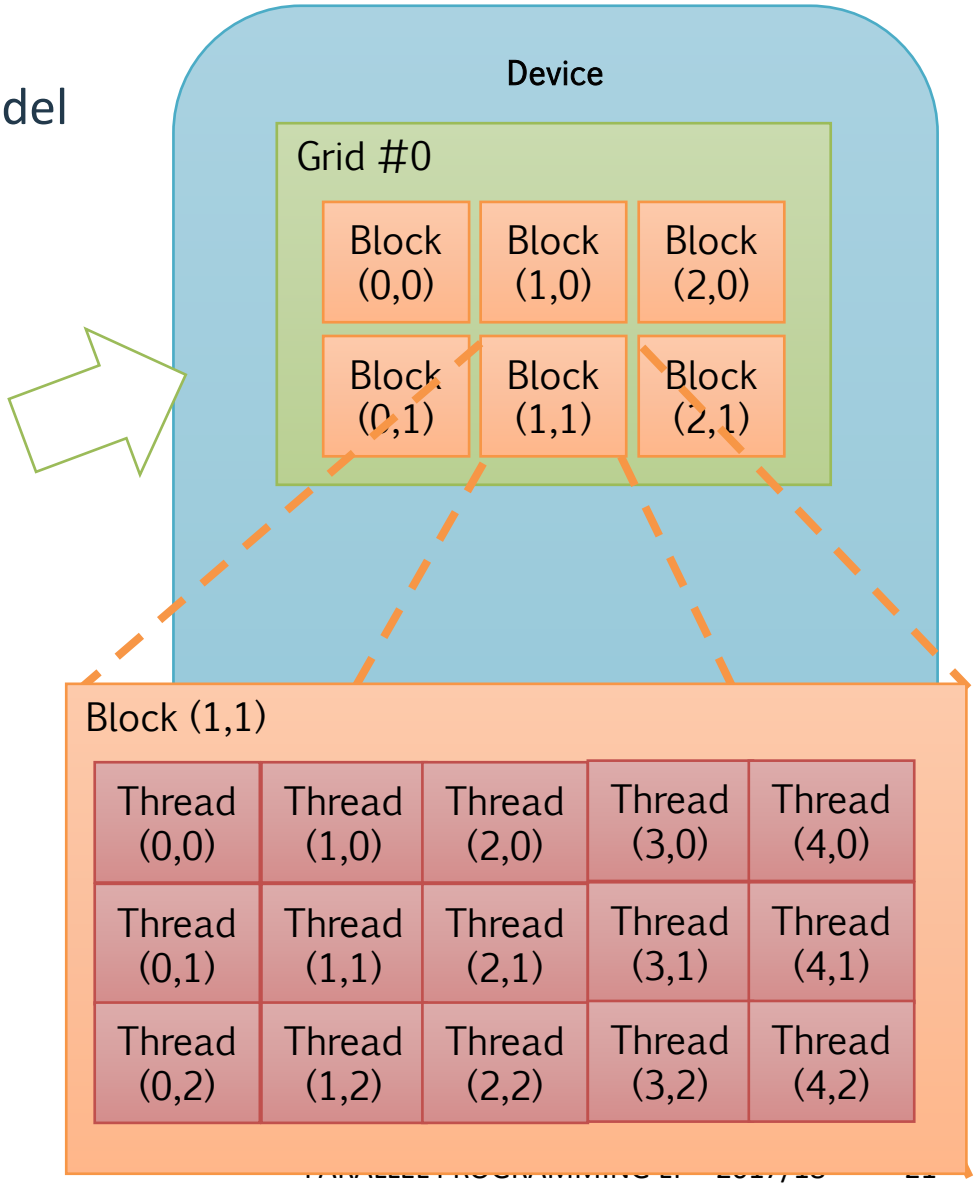
- Exploit maximum parallelism
- Modern applications are complex! (multi-level, irregular parallelism)
- Explicit shared-memory programming on NUMA hierarchies

..or the solution!?!

1) Offload-based execution models: CUDA

- Esposed in the programming model
- Based on the concepts of
  - Grid(s)
  - Block(s)
  - Thread(s)





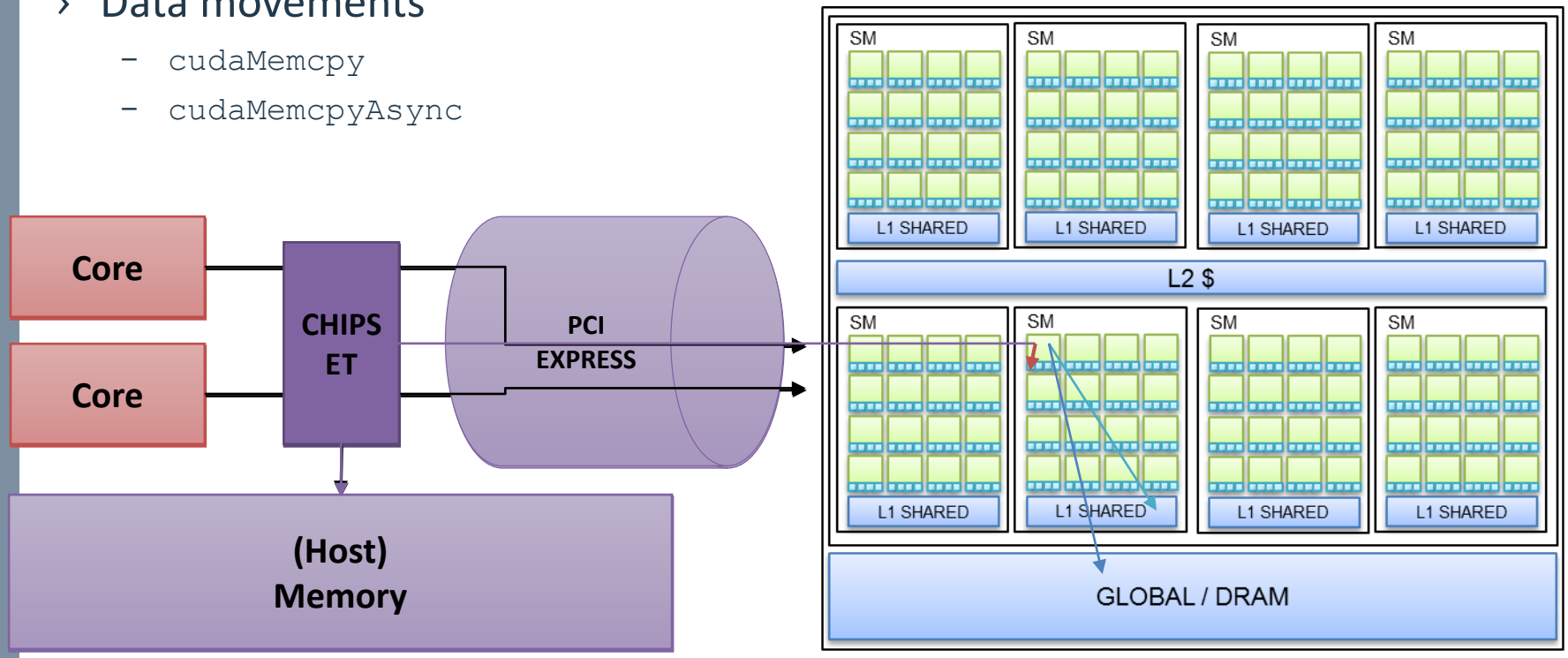

```
dim3 grid_size;
grid_size.x = 3;
grid_size.y = 2;

dim3 blk_size;
blk_size.x = 5;
blk_size.y = 3;

myKernel<<<grid_size,blk_size>>>();
```

2) Exploit NUMA hierarchy in CUDA

- Runtime must be aware of all
- Memory allocations
  - `cudaHostAlloc` ➔ Host mem
  - `cudaMalloc` ➔ Global mem
  - `__shared__` keyword ➔ Shared mem
- Data movements
  - `cudaMemcpy`
  - `cudaMemcpyAsync`



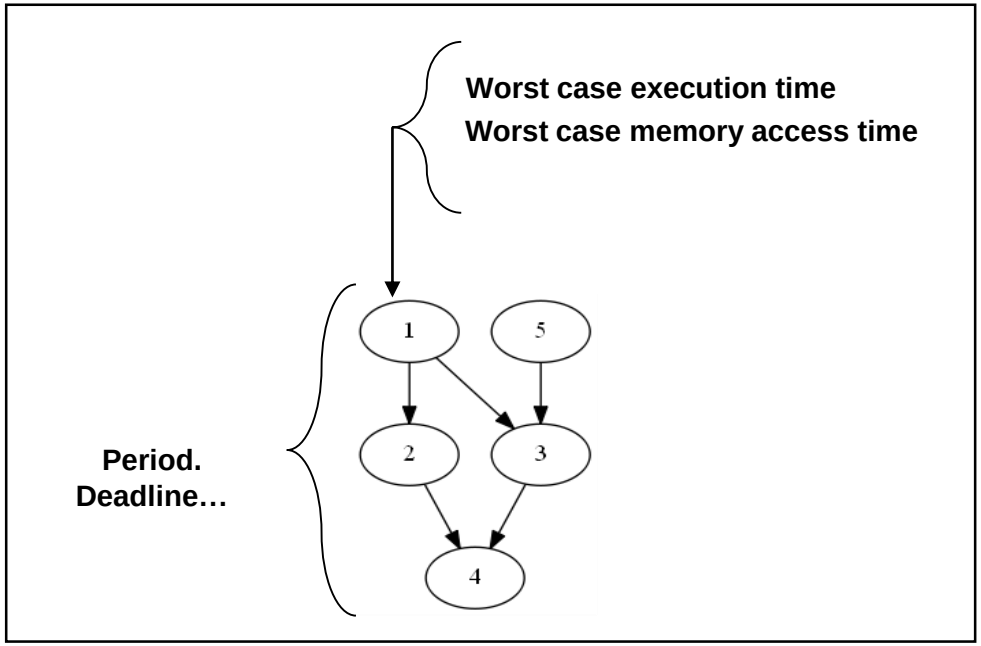
Most heterogeneous programming models are memory-centric programming models

2015 paper @ RTEST

Meanwhile,
in the RT community…

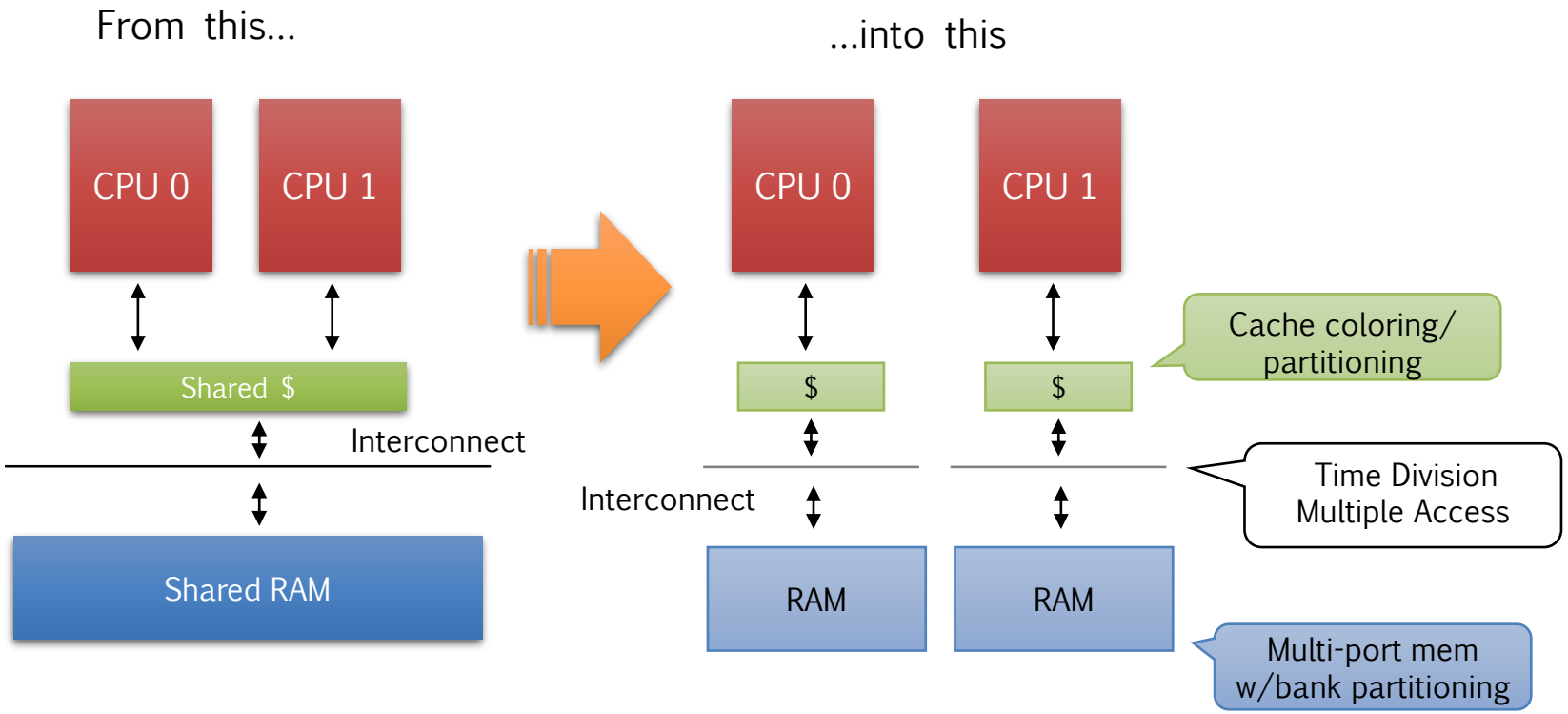
Graphs for RT systems – RT-DAGs

- Each task is specified by a directed acyclic graph (RT-DAG), where nodes represent task parts, and edges represent precedence constraints.
- Each node (runnable) $\tau_{i,j}$ has an associated:
  - worst-case computation time $\mathbf{C_{i,j}}$,
  - worst-case memory access size $\mathbf{M_{i,j}}$,
- Each task $\boldsymbol{\tau_{i,j}}$ is characterized by a period $\mathbf{T_i}$ and a relative deadline $\mathbf{D_i} \leq \mathbf{T_i}$.....

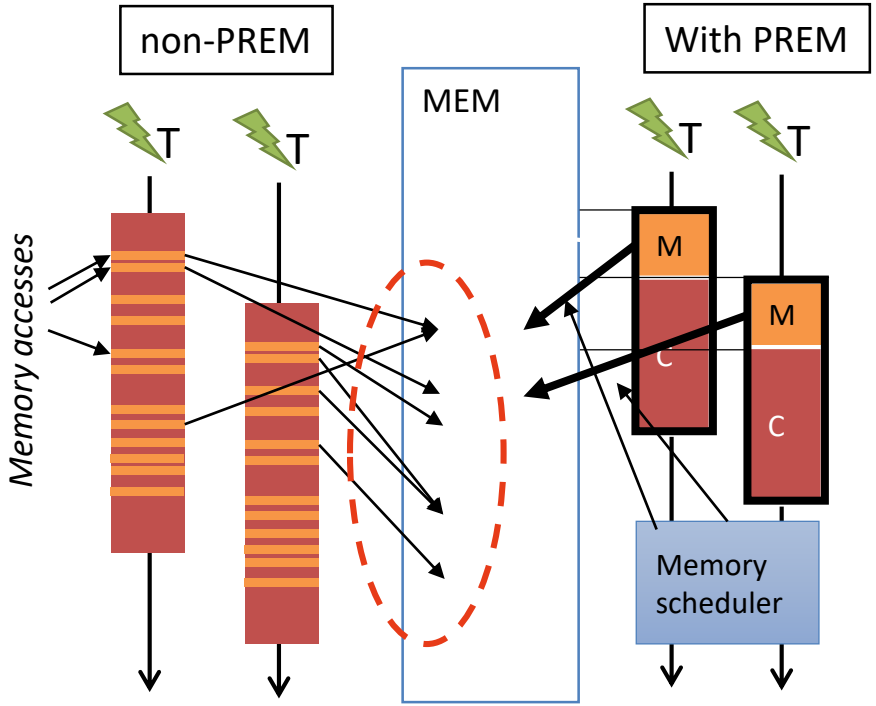


Single-Core Equivalence

› A set of techniques to turn the view of the system that software has..



PREM - PRedictable Execution Models



- Group memory access at the beginning of every software task
- Co-schedule memory accesses and tasks-to-cores
- Greatly reduces the complexity of the scheduling problem

...and increases performance

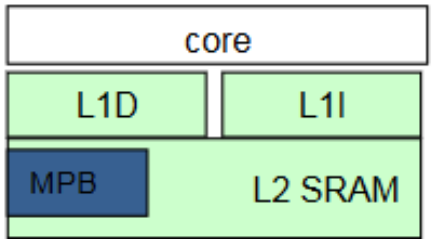
Up to 4x predictable performance on a many-core platform

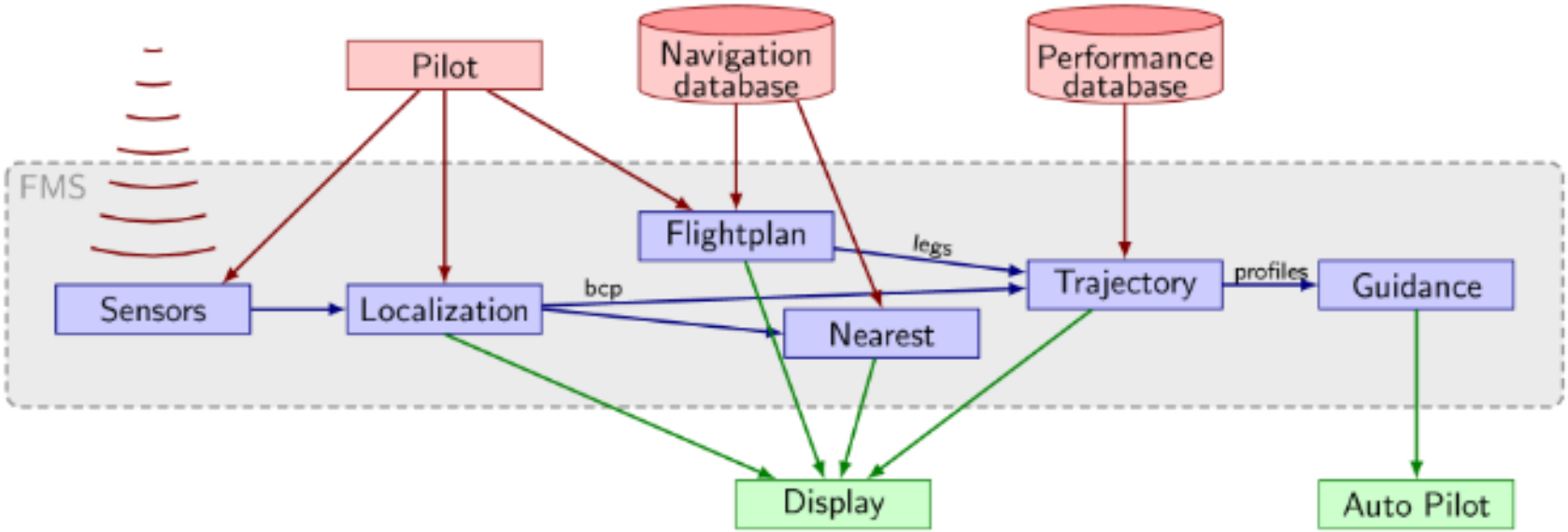
2015 paper @ RTEST

# Cores/threads	1	2	4	8
No-PREM – Worst (Analytical)	0.026	0.047	0.088	0.170
PREM – Worst (Analytical)	0.010	0.014	0.022	0.038
Speedup	2.6×	3.4×	4.0×	4.5×

AER Model

- Explicit memory management
  - Exploit local SPM memories
  - Reserved storage (MPB)
- Mapping + scheduling
- From the avionic domain



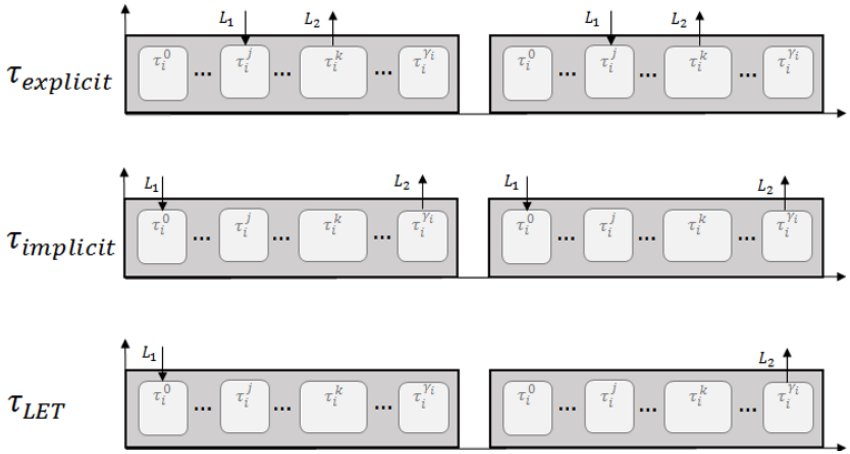


The explicit, the implicit and the LET

Explicit communication, a task access shared variables at any point during its execution

Implicit communication, tasks accessing shared labels should work on task-local copies instead of the original labels.

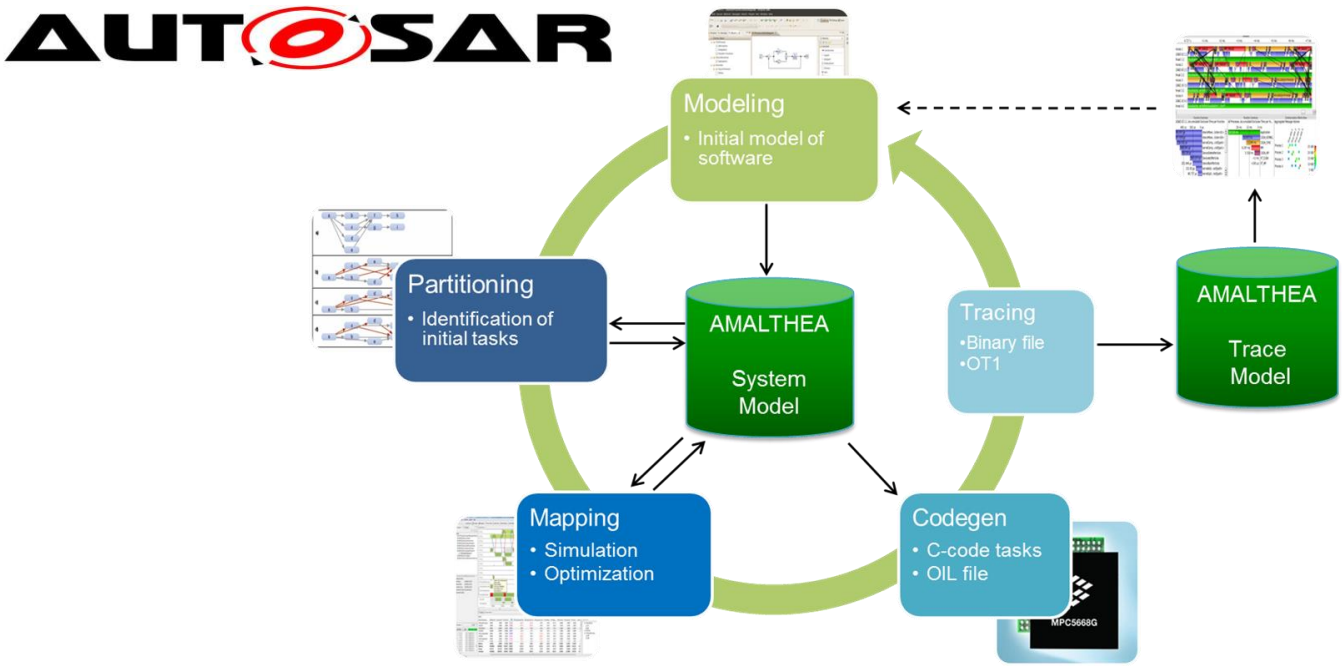
Logical Execution Time (LET), inputs and outputs are updated logically at the beginning and at the end of the so called communication interval



Industrial challenge

AMALTHEA is an XML-based open source document format for modeling embedded multi-/many-core systems, supporting the AUTOSAR standard.

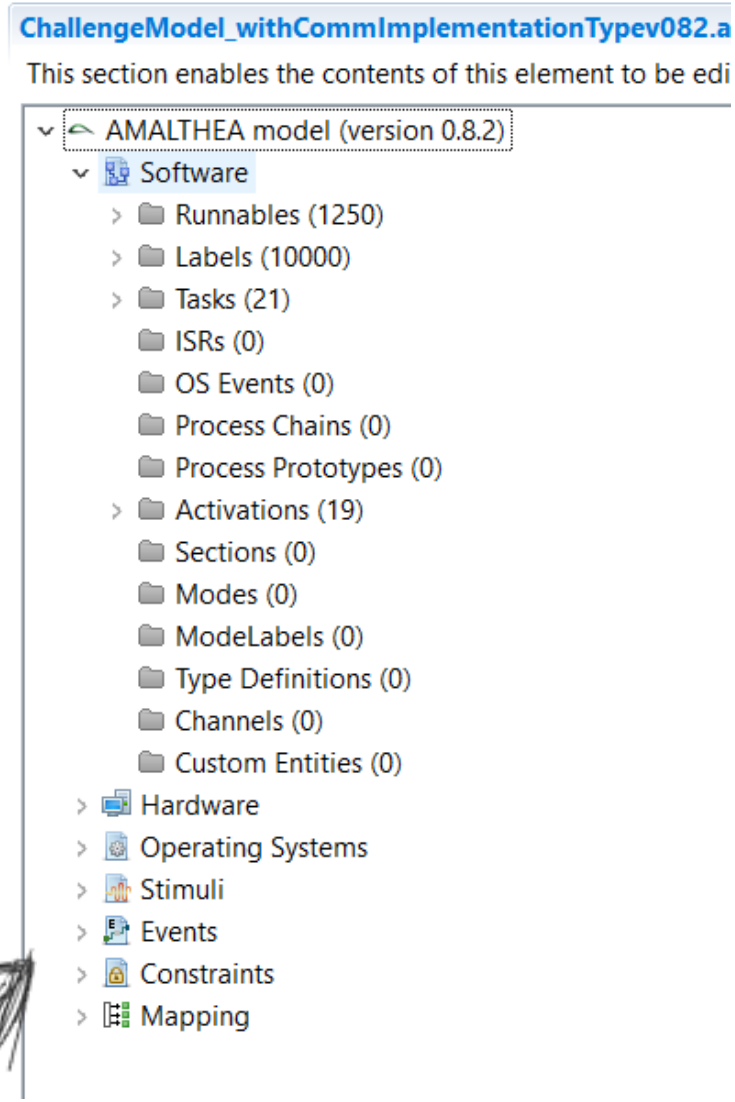
› The Amalthea platform allows users to distribute data and tasks to the target hardware platforms, with the focus on optimization of timing and scheduling.



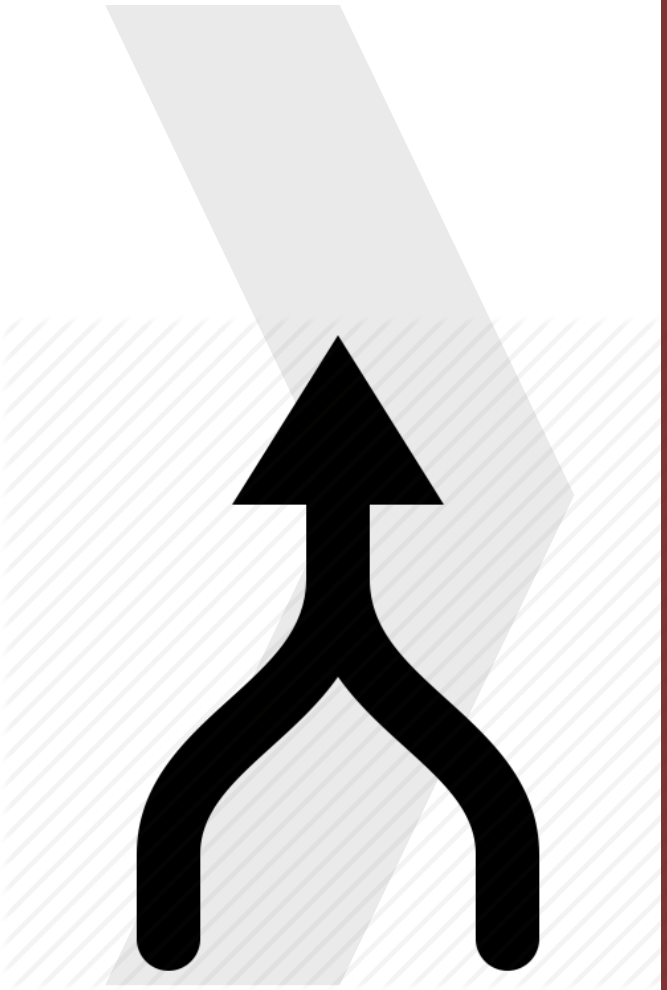
Amalthea – SW modeling

- Tasks & runnables
- Labels
- Effect chains
- …

@See 2017 Waters challenge



Put them all together!



Convergence

2017 paper @ FDL

2017 paper @ WATERS

Amalthea goes along pretty well with RT-DAGs and mem-aware models

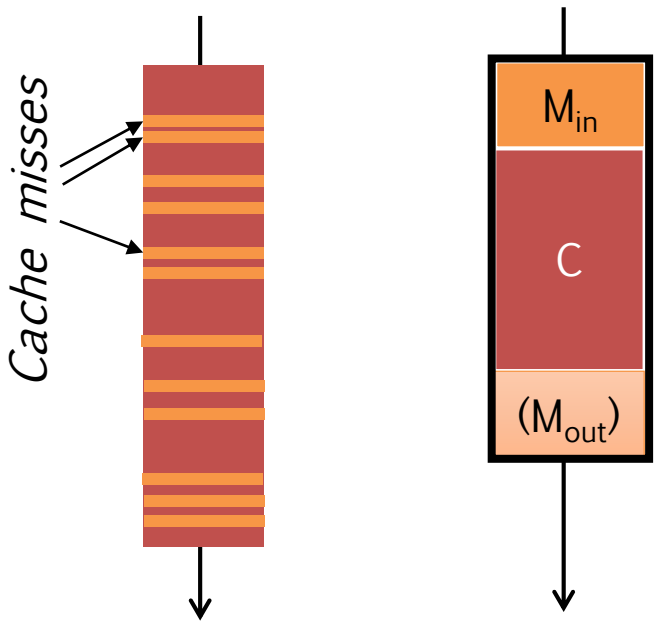
- Map parallel tasks/RT-DAGs to massively concurrent execution engines/accelerators
- LABELS/PREM's mem phases naturally open to memory-centric (co)scheduling
- Conditional DAGs ?

Offload-based, Real-Time, execution models

Only host

Raw

Memory-aware (LET, PREM, AER...)



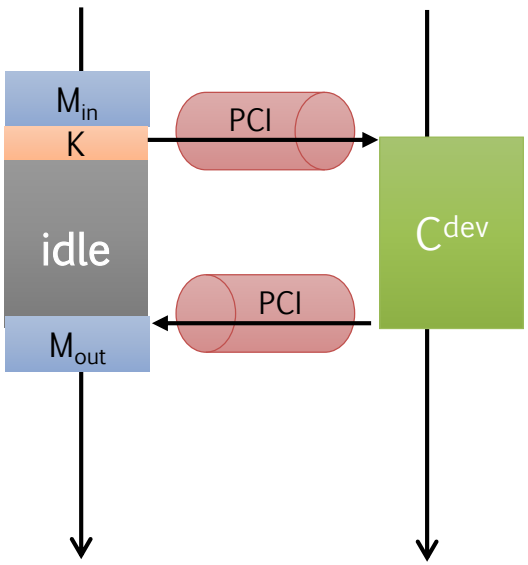
$$\tau_i = \{P_i, D_i, C_i, (M_{in}, M_{out})\}$$

- Non PREM vs PREM
- M_{out} can be moved to subsequent M_{in}s

Host + accel.

CPU

GP-GPU



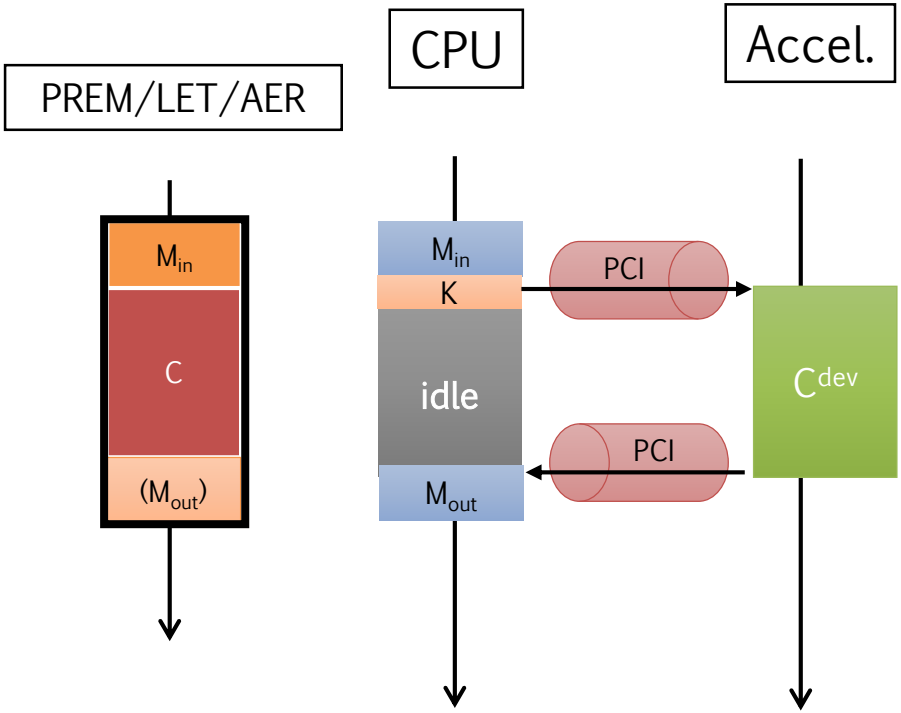
$$\tau_i = \{P_i, D_i, C_i, C_i^{dev}, M_{in}^{dev}, M_{out}^{dev}\}$$

- Explicit Mem transfer to/from accel.
- Constant offload overhead K

…challenges

Challenge #1

*S*cheduling on heterogeneous µcore accelerators



$$\tau_i = \{P_i, D_i, C_i, C_i^{dev}, M_{in}^{dev}, M_{out}^{dev}\}$$

1. Schedule data transfers through I/O port
2. Schedule tasks on GPU vs. leave them on CPU
3. Exploit CPU idle time (Async offload execution)

Challenge #2

Example: from **discrete** GP-GPU…

› Comm via PCI-E

..to integrated GP-GPUs (iGPUs)

› Embedded system (automotive)

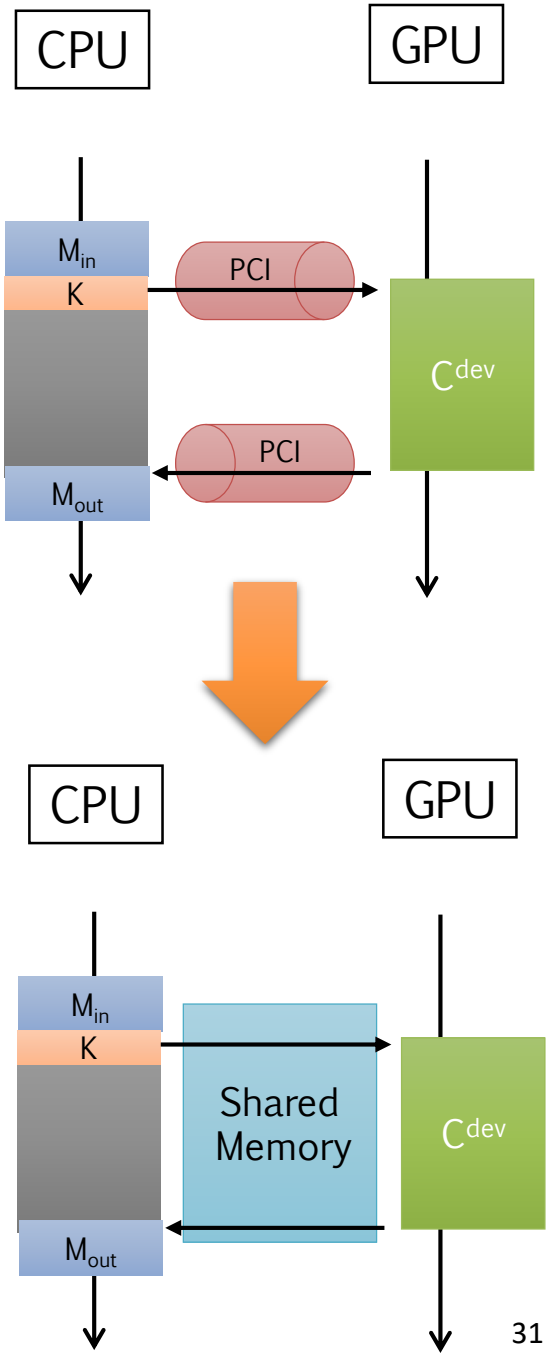
› Comm via shared mem banks

› Nvidia's Unified Virtual Mem (UVM)

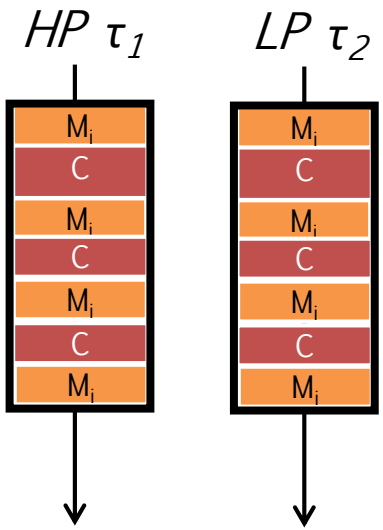
From I/O problem, to memory problem

› Bound CPU-GPU interference

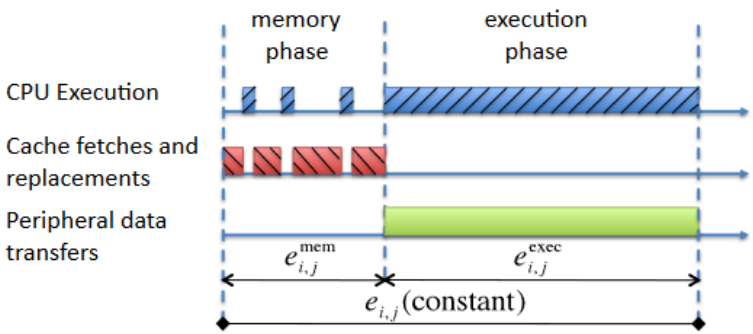
› Remembed ETFA's paper?



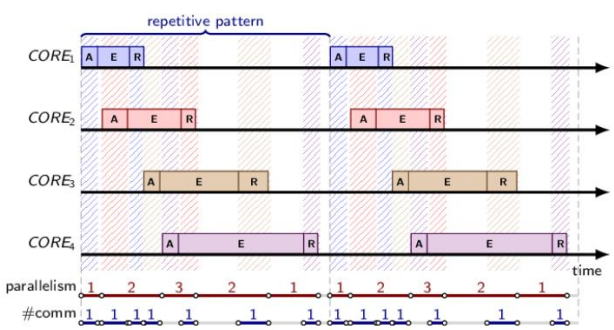
Challenge #2: the story so far.. (CPU vs. CPU)



<u>How to schedule memory accesses?</u>



Predictable Execution Model: Concept and Implementation
Rodolfo Pellizzoni, et al.

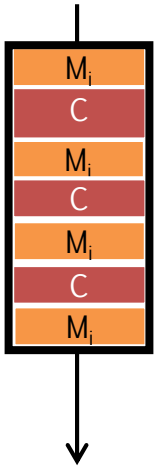


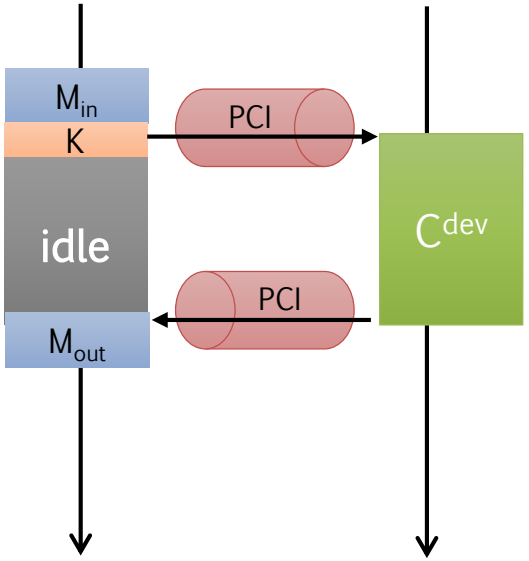
Deterministic platform software for hard real-time systems using multi-core COTS
Sylvain Girbal, et al.

...to accelerator and beyond (CPU vs. GPU)

$HP\ \tau_1 \longrightarrow LP\ \tau_2$

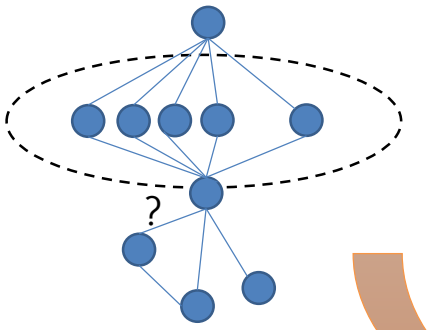
$LP\ \tau_2 \longleftarrow HP\ \tau_1$

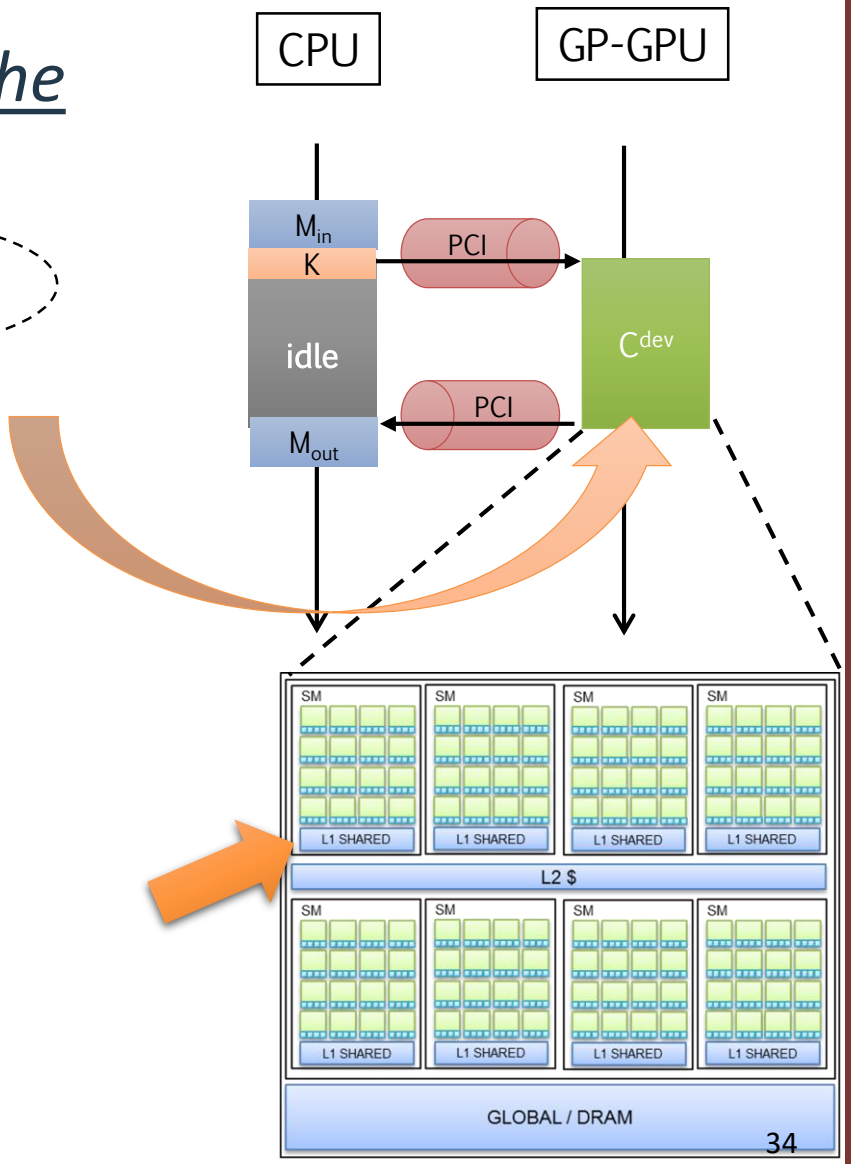




Challenge #3 – "for real men"

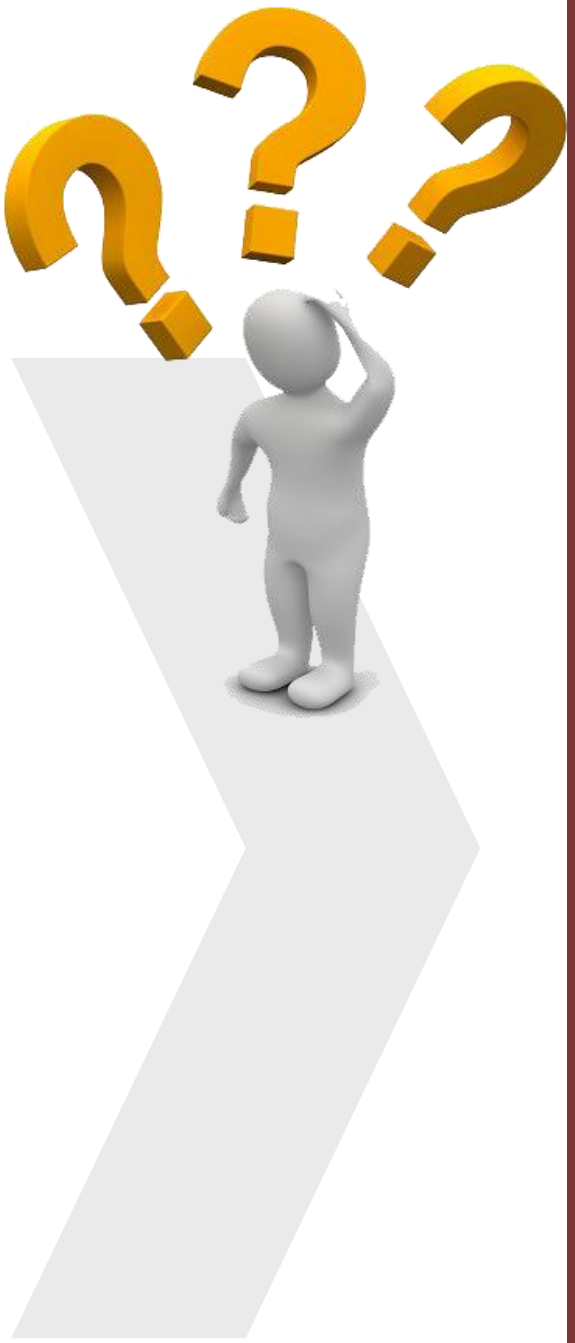
Schedule parallel DAGs onto the accelerator





- Include app-specific DAG models in AMALTHEA
  - E.g., DNN
- Exploit clustered nature of accelerators (e.g., GPU SMs)
- Enforce spatial and timing mem. isolation

What's missing?



Amalthea – HW

- Cores
- On-chip IC
- Memory
  - Local vs. Global
  - Access latency
- Heterogeneous PUs
  - Accelerators

@See Waters challenge 2018

AMALTHEA Contents Tree

ChallengeModel_withCommImplementationTypev082.amxmi Contents

This section enables the contents of this element to be edited.

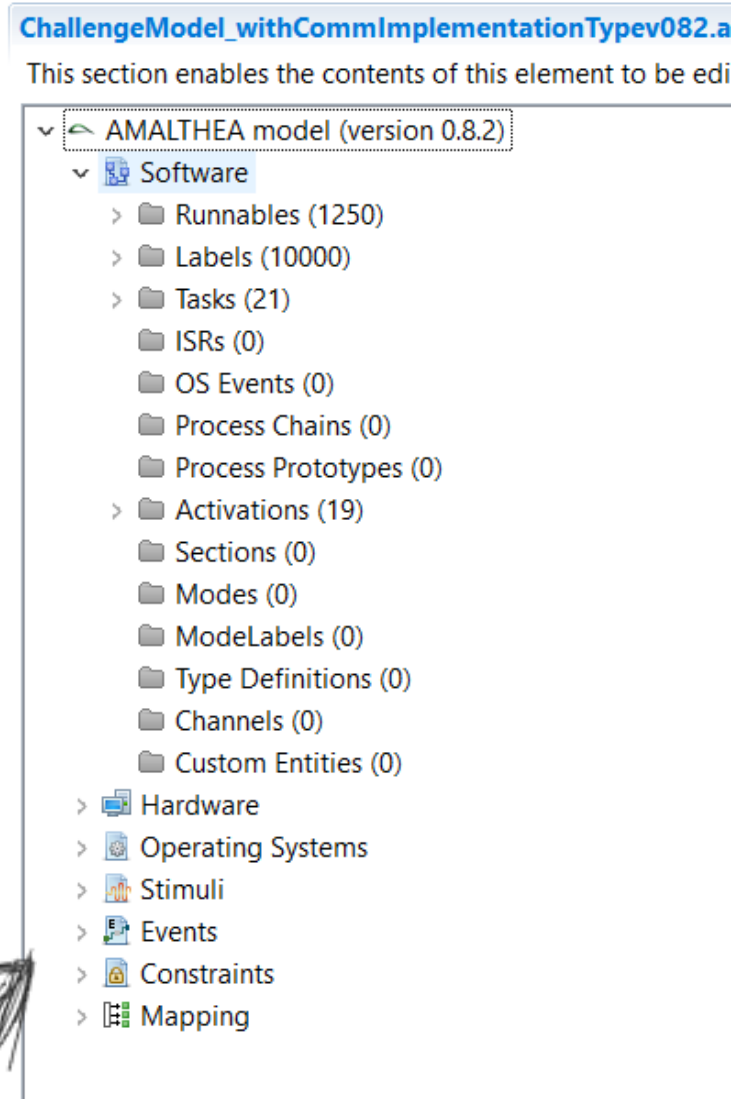
- AMALTHEA model (version 0.8.2)
  - Software
  - Hardware
    - Generic Core
    - GenericLocalRAM
    - GenericRAM
    - GenericCrossbarSwitch
    - GenericLocalBus
    - AccessPath (Latency) CORE0_to_LRAM0 : CORE0 --> LRAM0
    - AccessPath (Latency) CORE0_to_LRAM1 : CORE0 --> LRAM1
    - AccessPath (Latency) CORE0_to_LRAM2 : CORE0 --> LRAM2
    - AccessPath (Latency) CORE0_to_LRAM3 : CORE0 --> LRAM3
    - AccessPath (Latency) CORE1_to_LRAM0 : CORE1 --> LRAM0
    - AccessPath (Latency) CORE1_to_LRAM1 : CORE1 --> LRAM1
    - AccessPath (Latency) CORE1_to_LRAM2 : CORE1 --> LRAM2
    - AccessPath (Latency) CORE1_to_LRAM3 : CORE1 --> LRAM3
    - AccessPath (Latency) CORE2_to_LRAM0 : CORE2 --> LRAM0
    - AccessPath (Latency) CORE2_to_LRAM1 : CORE2 --> LRAM1
    - AccessPath (Latency) CORE2_to_LRAM2 : CORE2 --> LRAM2
    - AccessPath (Latency) CORE2_to_LRAM3 : CORE2 --> LRAM3
    - AccessPath (Latency) CORE3_to_LRAM0 : CORE3 --> LRAM0
    - AccessPath (Latency) CORE3_to_LRAM1 : CORE3 --> LRAM1
    - AccessPath (Latency) CORE3_to_LRAM2 : CORE3 --> LRAM2
    - AccessPath (Latency) CORE3_to_LRAM3 : CORE3 --> LRAM3
    - AccessPath (Latency) CORE0_to_GRAM : CORE0 --> GRAM
    - AccessPath (Latency) CORE1_to_GRAM : CORE1 --> GRAM
    - AccessPath (Latency) CORE2_to_GRAM : CORE2 --> GRAM
    - AccessPath (Latency) CORE3_to_GRAM : CORE3 --> GRAM
    - HW System System
      - ECU
      - GenericPLL
  - Operating Systems

Amalthea – SW modeling

- Tasks & runnables
- Labels
- Effect chains

- Multi-device model

@See 2017/18 Waters challenge

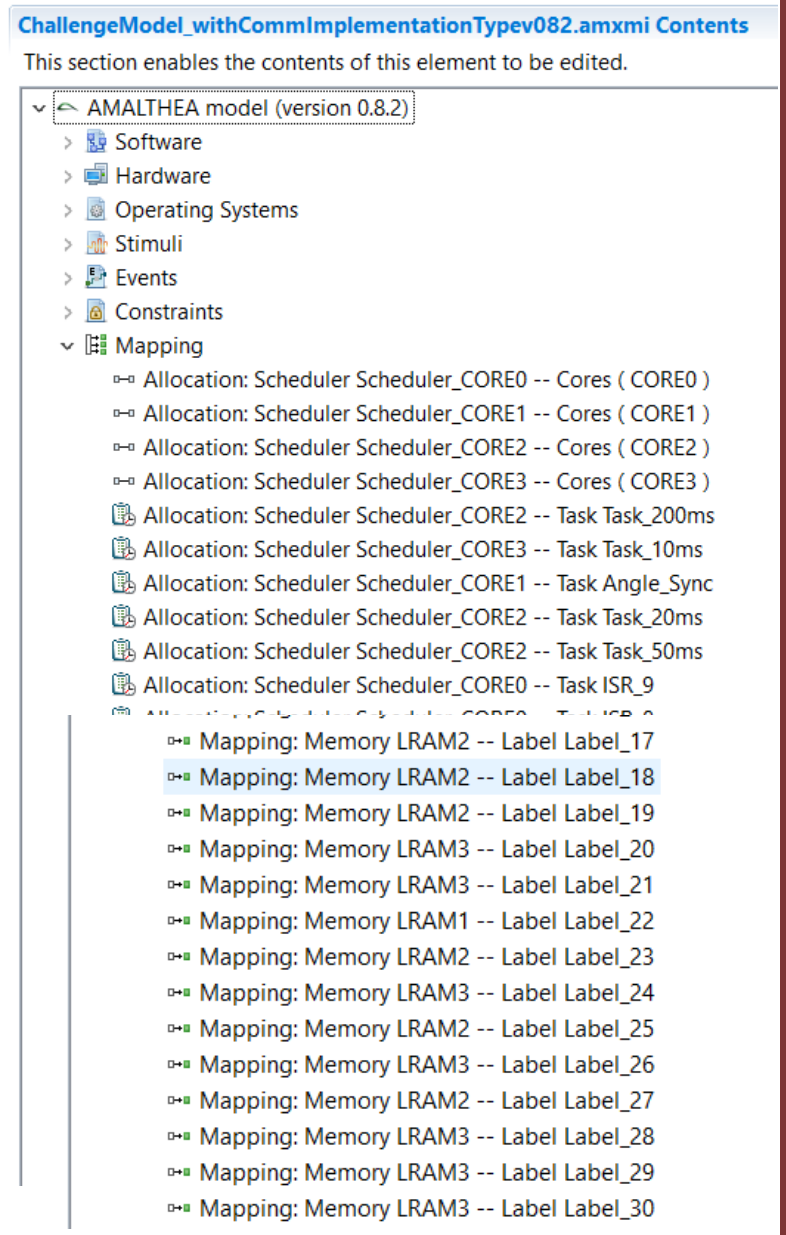


SW-HW mapping

Map

- Tasks onto Cores
- Labels onto memory blocks

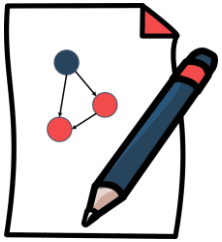
- Parallel tasks onto Het. PUs
  - Runnable-level?
- Labels onto memory transfers



Heterogeneous HW model

- Computing unit made of (clusters of) many-cores
- Hierarchical memory with **non-global** access space
  - Chances for unified virtual memory
- Explicit (runtime-compler driven) memory transfers

HiPeRT Generator Tool - HGT



- Starting from Amalthea description of a **RT** application
- Generates **ready-to-use**, timing accurate synthetic code that **correctly** mimics it
- On given target architecture