International Workshop on *Design Principles for Next Generation Embedded Computing Systems*

Amit Kumar Singh

School of Computer Science and Electronic Engineering

University of Essex

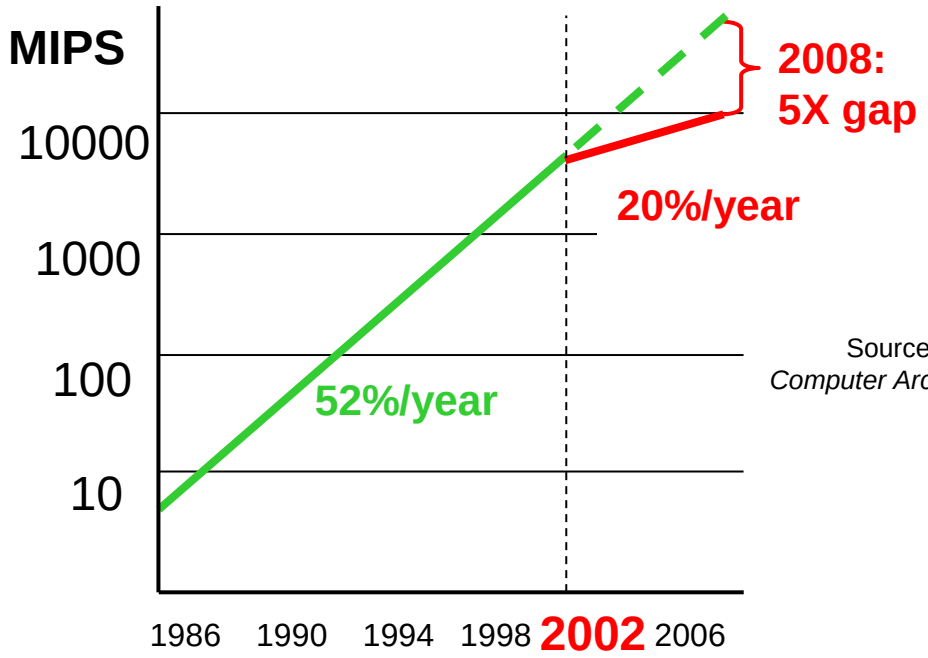
United Kingdom

W: http://aksingh.co.uk/ E: a.k.singh@essex.ac.uk

Multi-core Based Next Generation Embedded Systems

Multi-core Systems Revolution

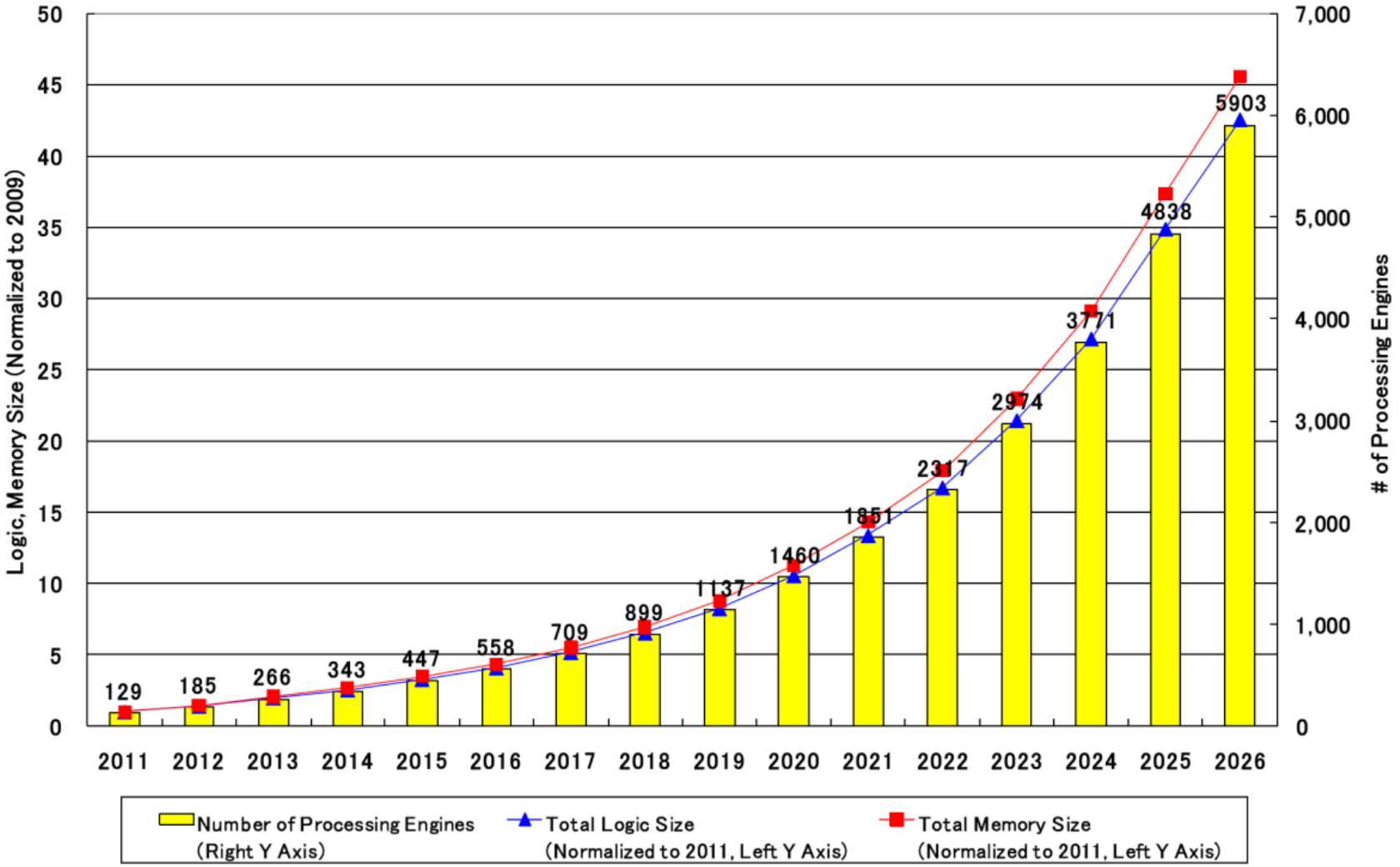
- **Single Core Performance:**
  - **Steady until 2002**
  - **Performance has fallen off Moore's Law**
    - **Maximum operational frequency has hit the roof**



Source: Hennessy & Patterson, *Computer Architecture: A Quantitative Approach*, 4th ed.

- **Parallel processing is the only choice**

Evolution in number of cores

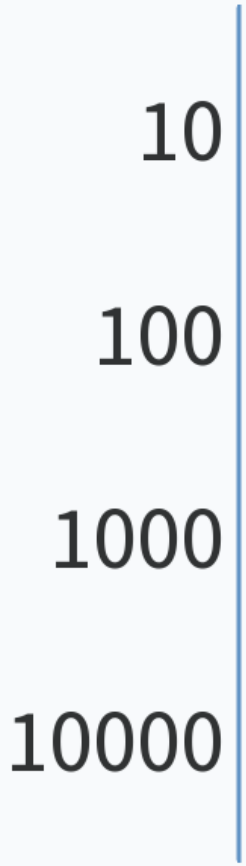


Source: ITRS

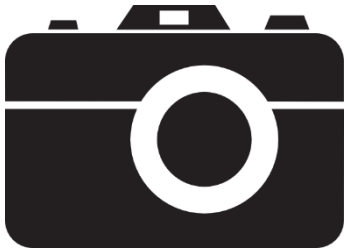
When poll is active, respond at **pollev.com/amitsingh510**

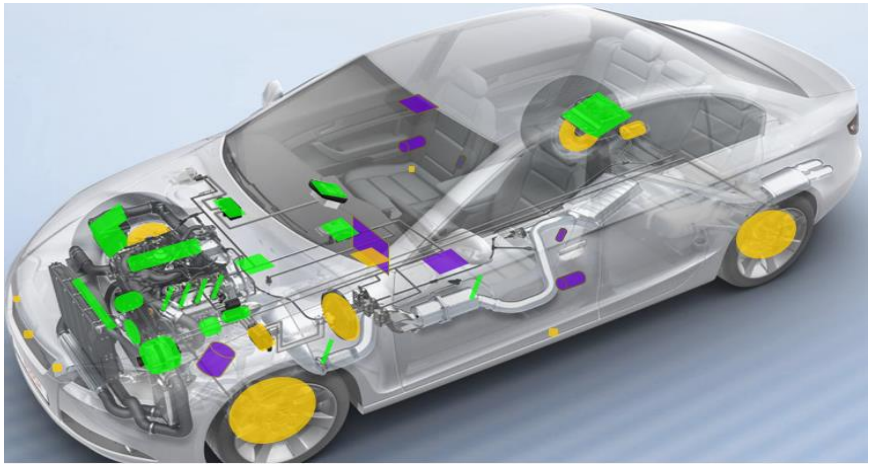
Text **AMITSINGH510** to **22333** once to join

How many cores in a modern chip?



Multi-core based Systems





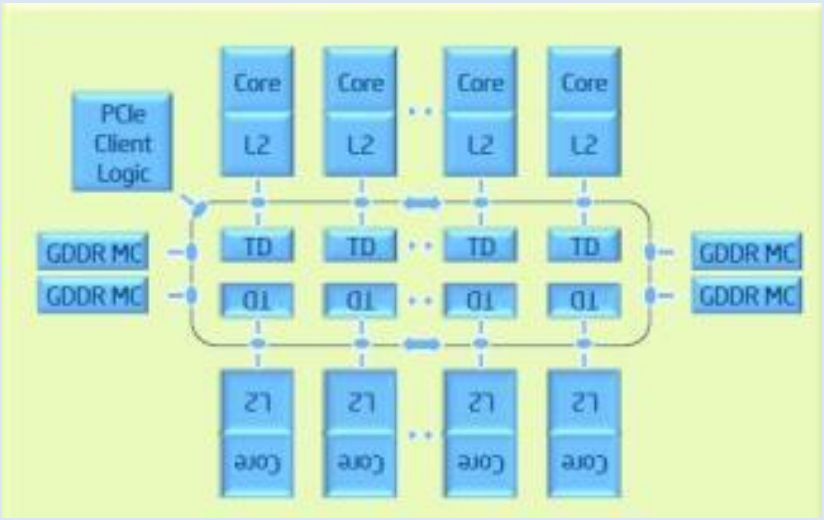
Multi-core wearable devices

Product (Announced)	SoC	CPU (#core)	Freq (MHz)	Memory	Typical CPU Power (mW)
Google Glass (Apr, 2012)	TI OMAP4430	ARM Cortex-A9 (dual-core)	1000	2GB RAM 16GB Flash	350
Vuzix M100 (Jan, 2013)	TI OMAP4460	ARM Cortex-A9 (dual-core)	1200	1GB RAM 4GB Flash	400
Qualcomm toq (Oct, 2013)	ST STM32	ARM Cortex-M3 (single-core)	200	16MB SRAM 2GB Flash	10
Optinvent ORA-1 (Aug, 2014)	Not available	ARM Cortex (dual-core)	1200	4GB Flash	Not available
Sony Smartwatch 3 (Sep, 2014)	Qualcomm Snapdragon 400	ARM Cortex-A7 (quad-core)	1200	512MB RAM 4GB Flash	450
LG G watch R (Sep, 2014)	Qualcomm Snapdragon 400	ARM Cortex-A7 (quad-core)	1200	512MB RAM 4GB Flash	450
Samsung Gear S2 3G (Aug, 2015)	Not available	ARM Cortex-A7 (dual-core)	1000	512MB RAM 4GB Flash	Not available
Motorola Moto 360 2ed generation (Sep, 2015)	Qualcomm Snapdragon 400	ARM Cortex-A7 (quad-core)	1200	512MB RAM 4GB Flash	450

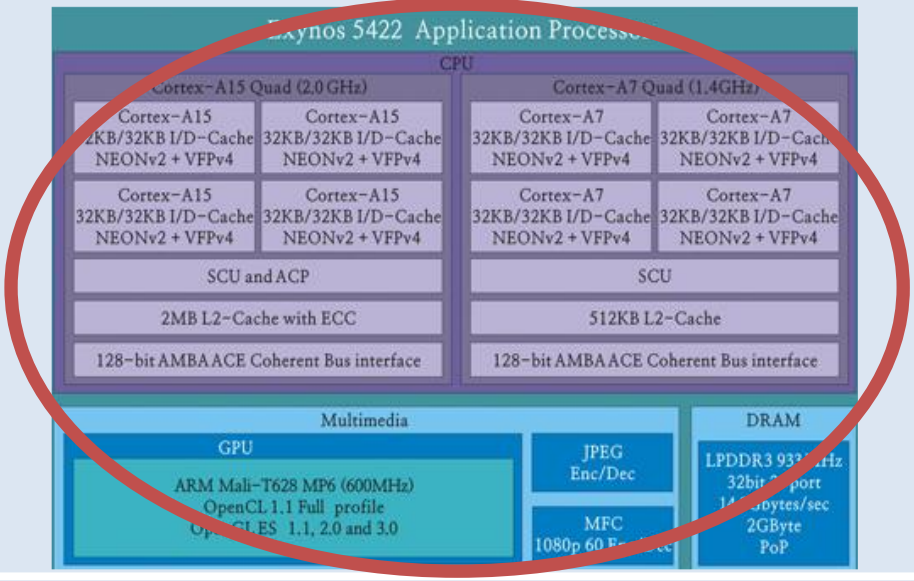
Source: Tan et. al. "LOCUS: Low-Power Customizable Many-Core Architecture for Wearables"

Multi-core Platform Examples

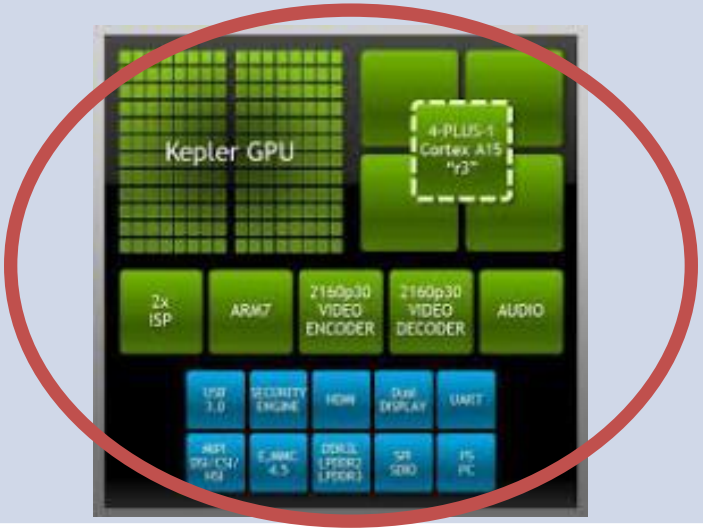
Intel Xeon Phi - **Homogeneous 61 Cores**



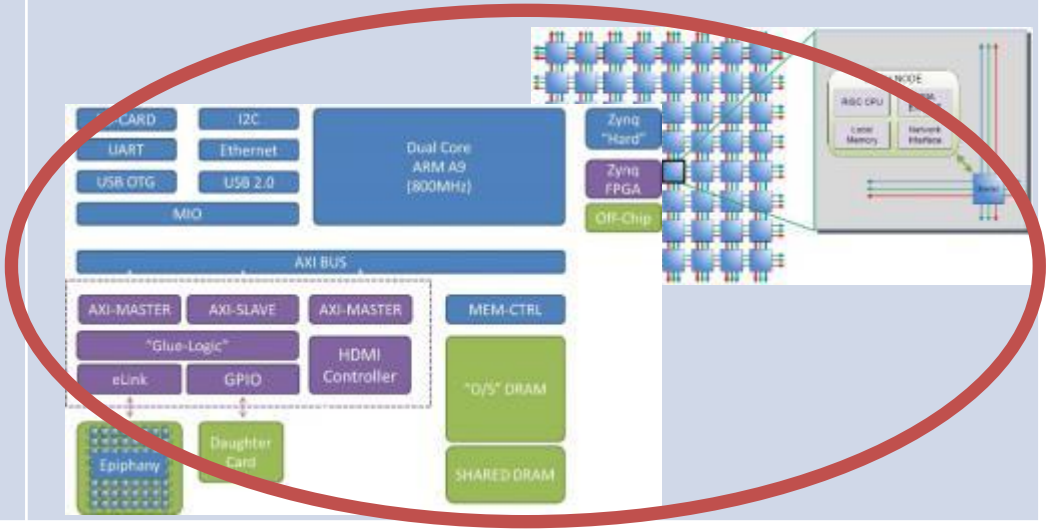
ODROID XU3 – **8 core big.LITTLE CPU + 6 cores GPU**



Nvidia Jetson TK1 - **Quad core CPU + 192 cores GPU**

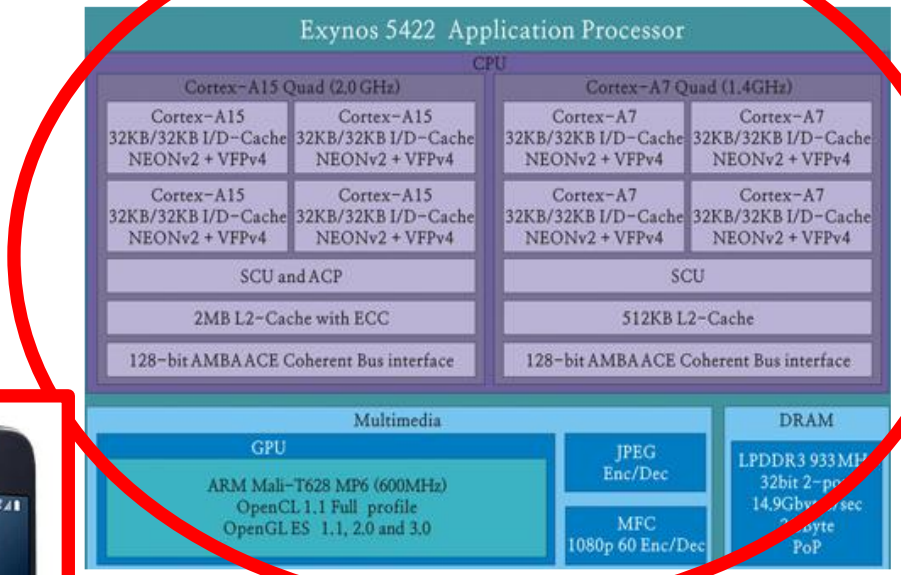


Parallella - **Dual core CPU + FPGA + 16 cores NoC**

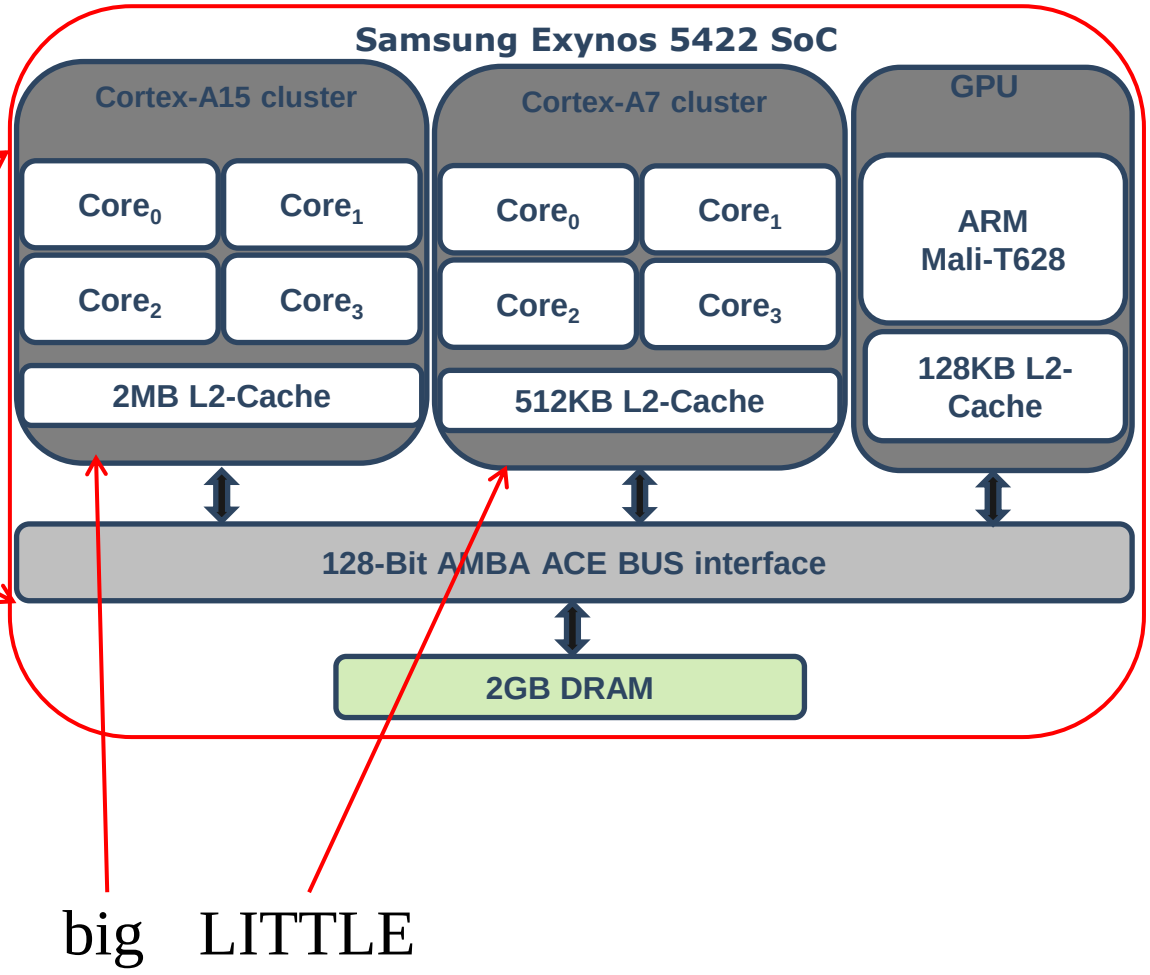


Heterogeneous Multi-core Usage

ODROID XU3 – **8 core big.LITTLE CPU + 6 core GPU**



Exynos 5422 SoC

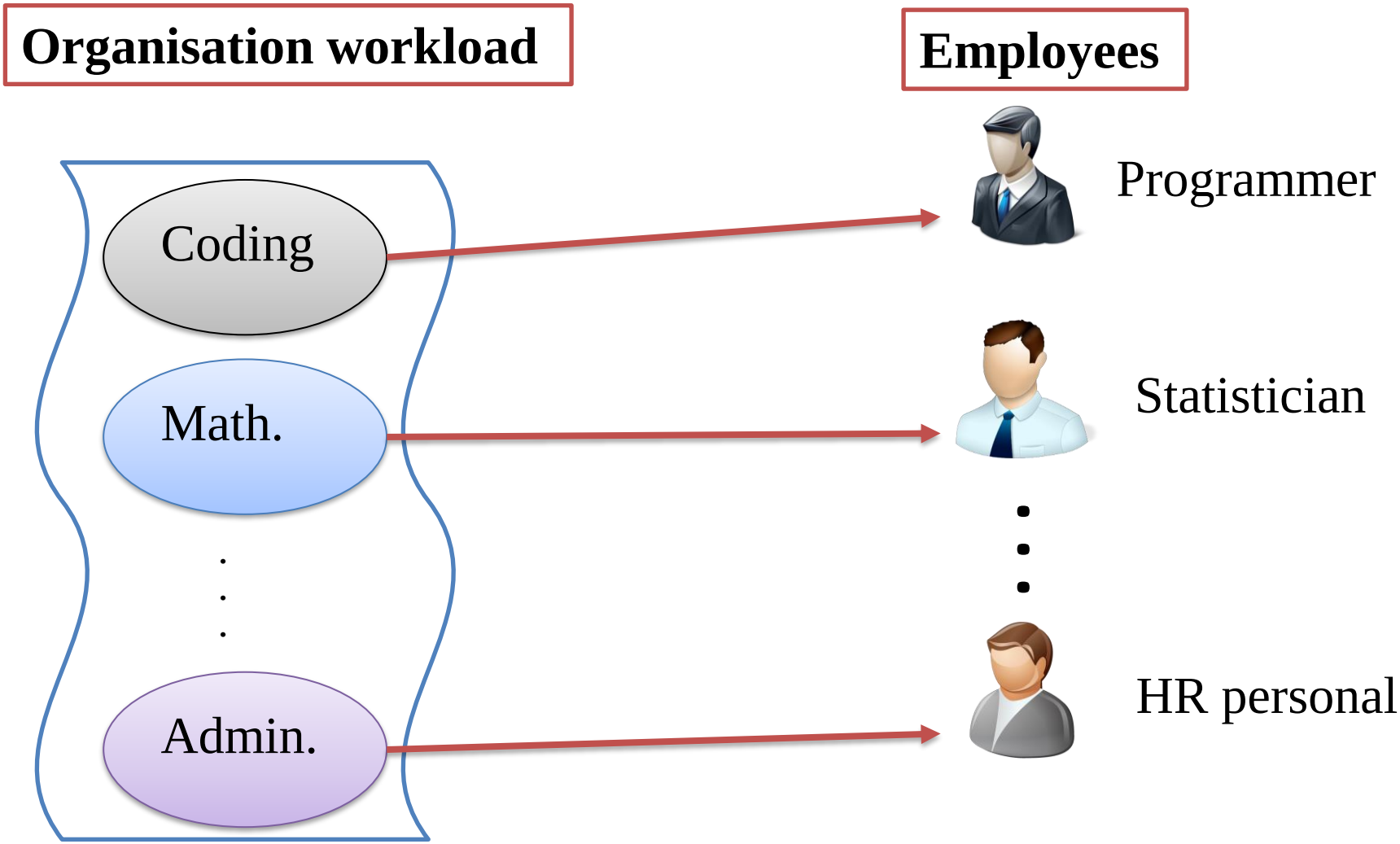


What if there is no heterogeneity in terms of processing capability of cores?

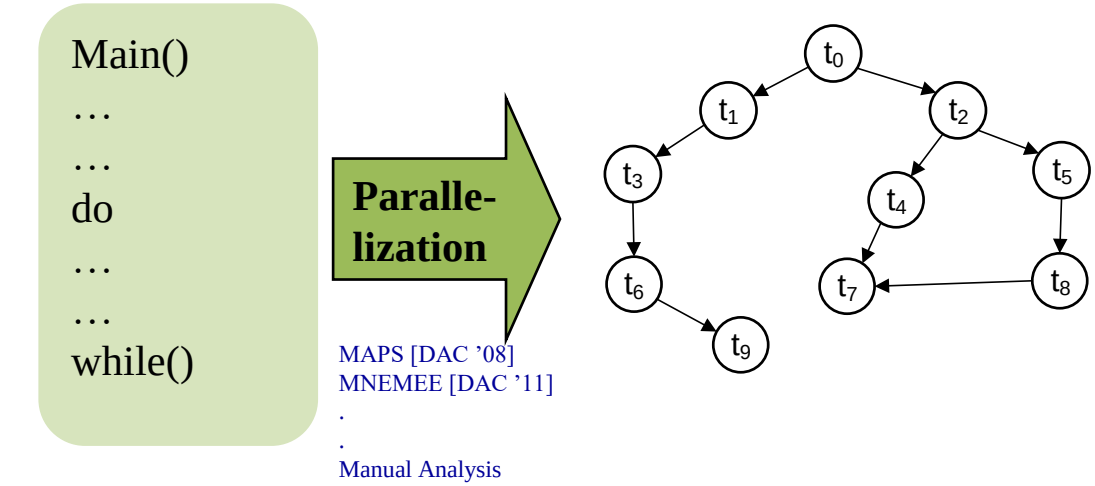
There will be no impact

It can impact performance

Heterogeneity Exploitation



Applications Execution on Multi-cores

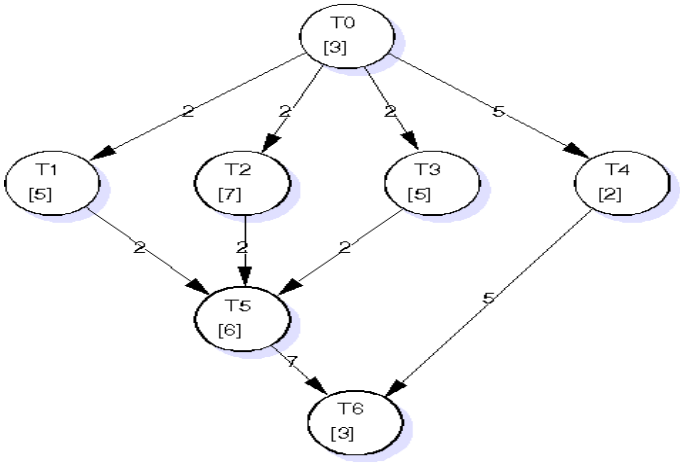


Sequential Application

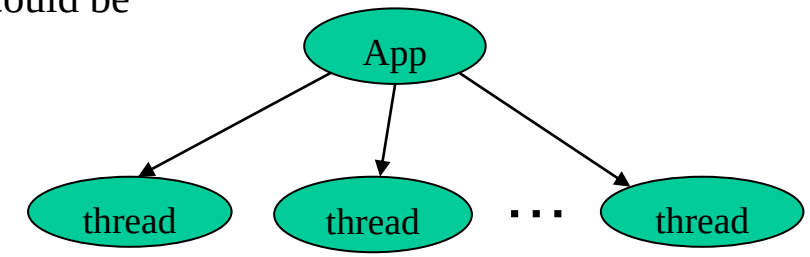
Application Task Graph

Applications Execution on Multi-cores

- Applications Representation
  - Task parallelism



  - Thread parallelism
    - A task/application could be multi-threaded



Respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join, then **A, B, C, or D**

If an application cannot have parallel representation, how to best improve its performance?

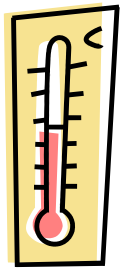
Performance cannot be improved	**A**
By assigning to a suitable processing element	**B**
By designing a dedicated circuit for it	**C**
By running on a GPU	**D**

Important metrics for Embedded Systems

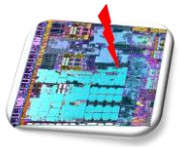
- **Performance**

- **Energy**

- **Temperature**



- **Reliability**

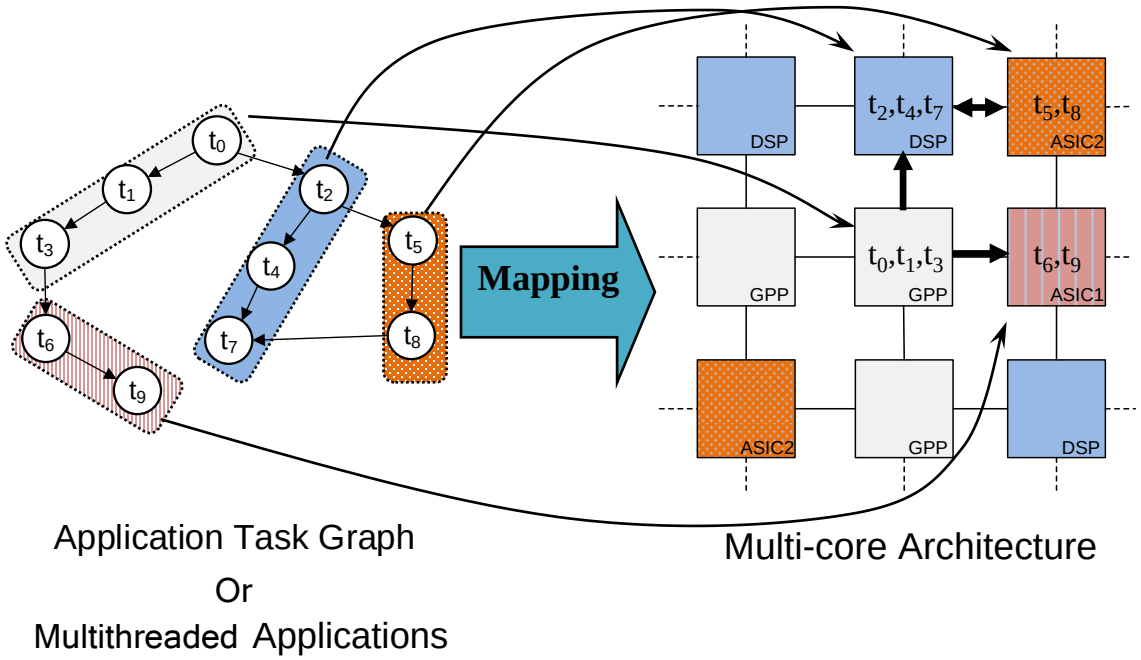


- **Security**

Can you think of any other metric?

Optimisation Knobs/Controls

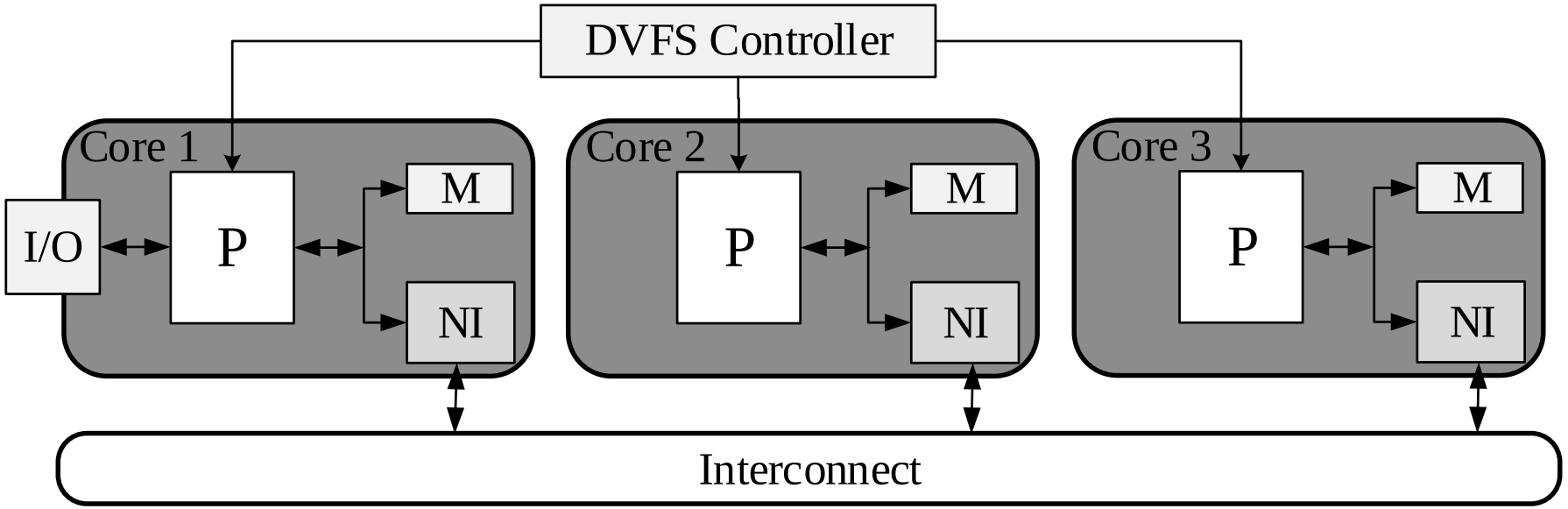
Applications Mapping on Multi-cores



- Mapping process defines assignment and ordering of the tasks and their communications onto the platform resources in view of some optimization criteria such as energy consumption and compute performance.
- Solving where, when, why (objective) problem.

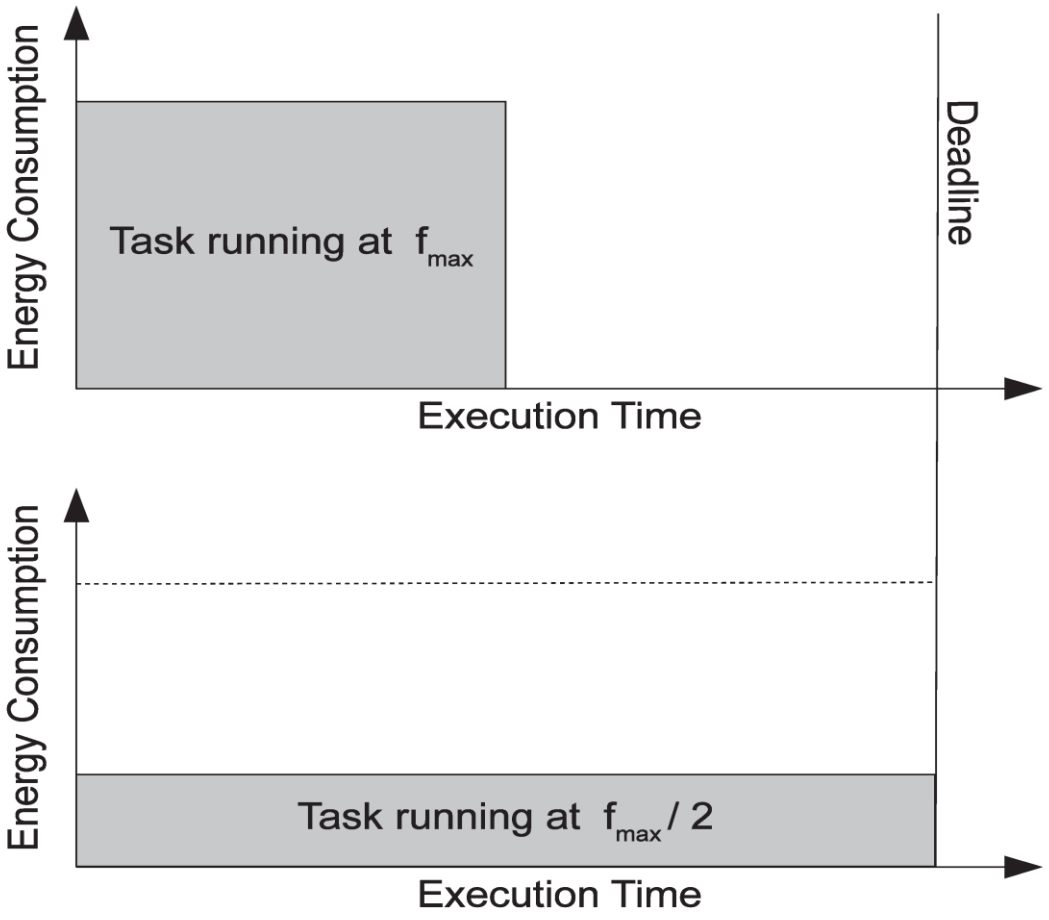
DVFS for Optimisation

- **Dynamic Voltage and Frequency Scaling (DVFS)**
  - **Supported in many advanced processor**
    - **Marvell StrongARM, Intel XScale, Transmeta Crusoe, ARM1176, many other modern processors**



P: Processor; M: Memory; NI: Network Interface

DVFS for Energy Savings



Dynamic power management (DPM)

- Shut downs processing elements (PEs) when inactive
  - Greedy: Go to sleep as soon as processing is finished
  - Timeout: Stay on expecting a new request. After time *t* of idleness go to sleep

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

Which approach you think is better?

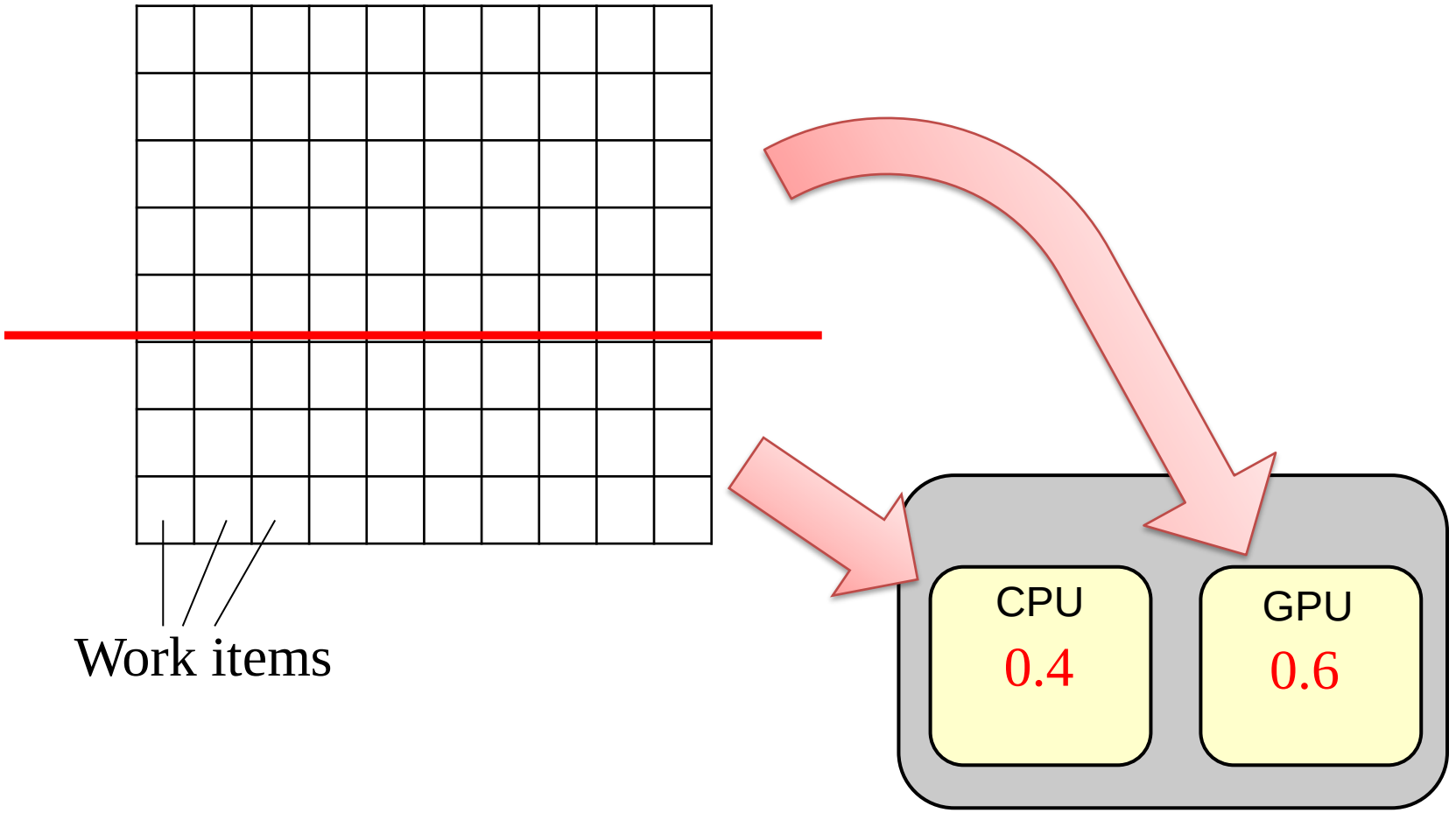
Greedy

Timeout

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Application Partitioning for Optimisation

Partitioning for Single (data-parallel) Application

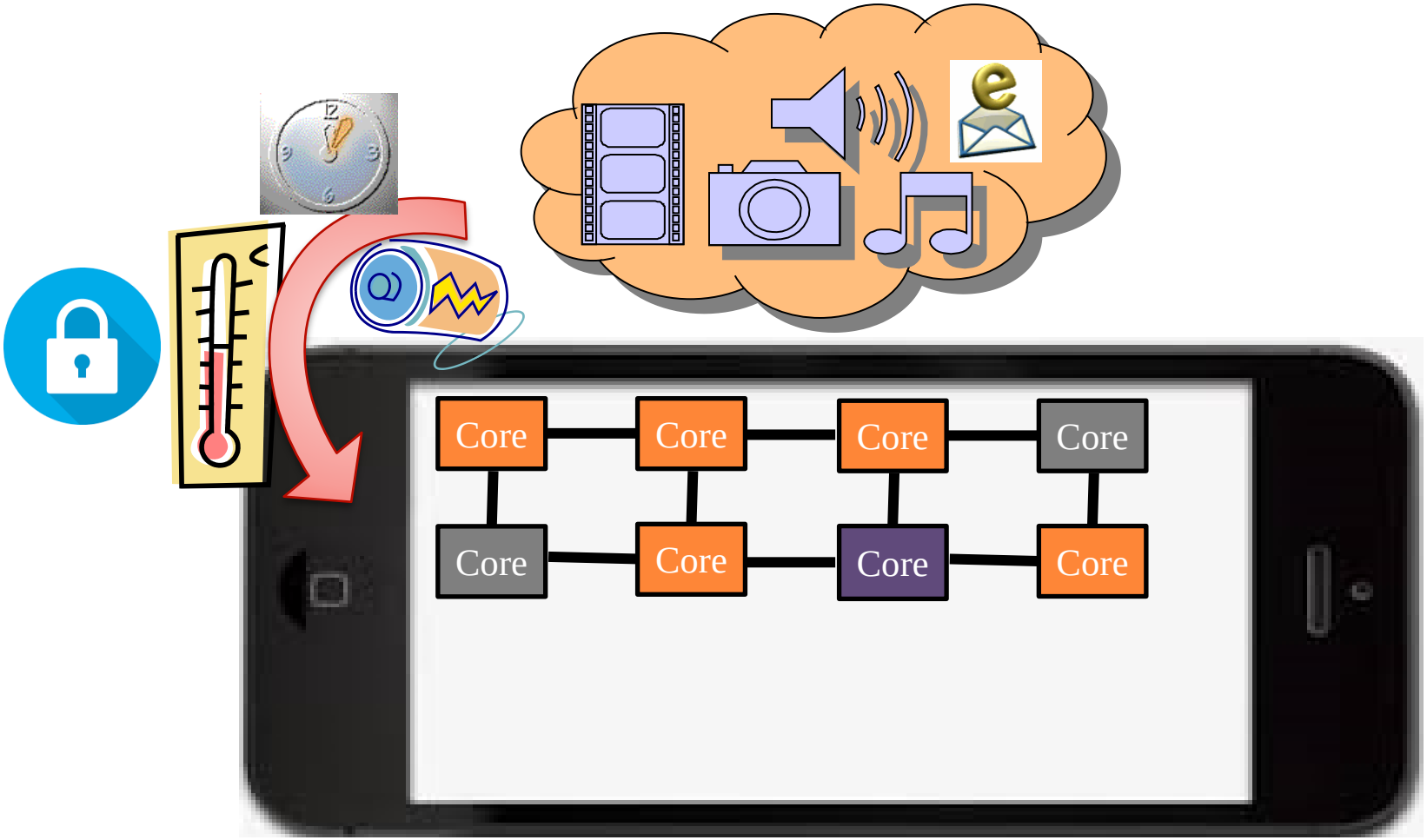


OpenCL provides this opportunity

Any other ways to optimise any metrics (e.g. performance, energy, temperature, reliability and security)?

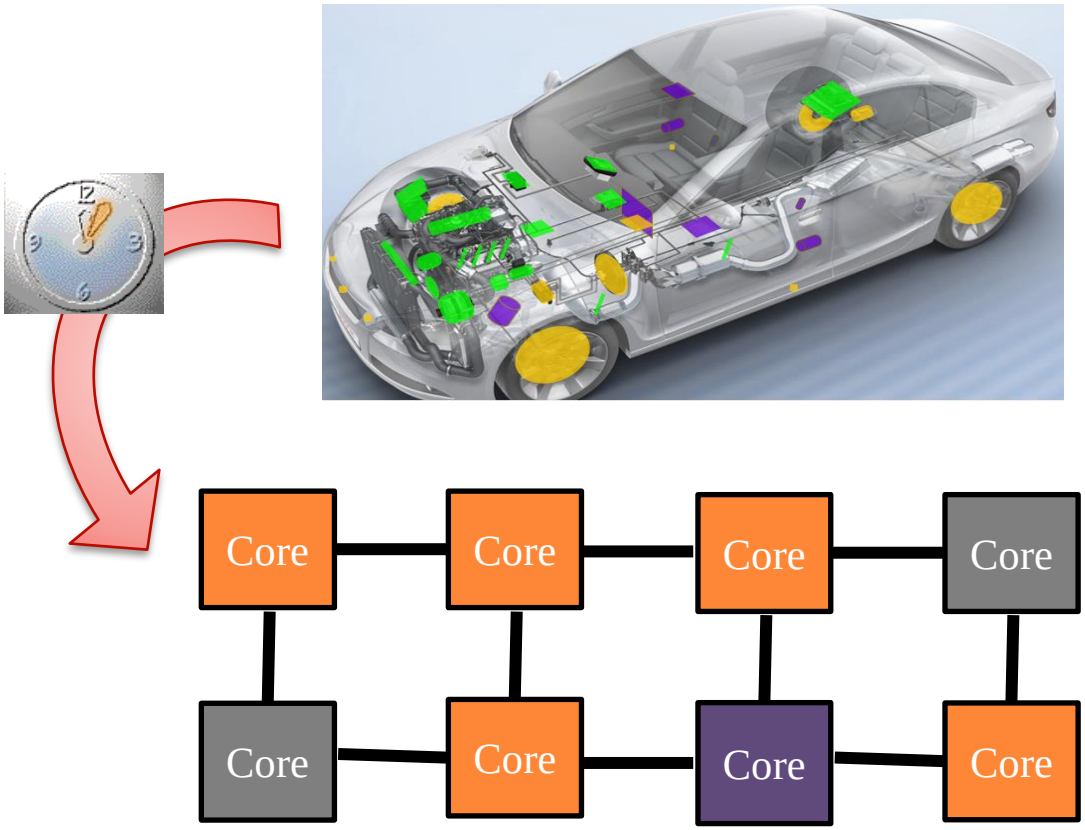
Example Application Domains for Optimisations with Earlier Principles

Multimedia Applications



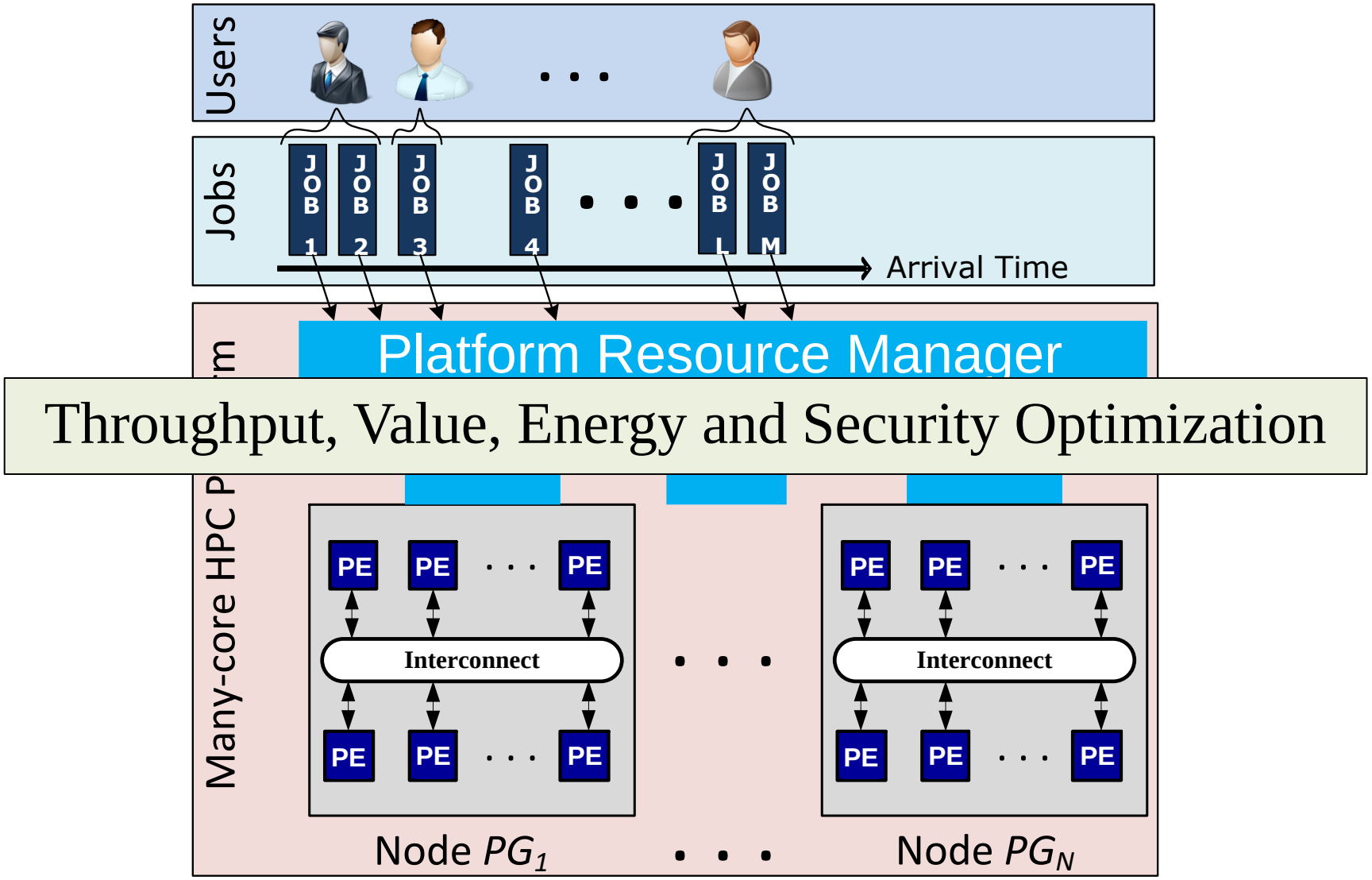
- Load and adapt the application tasks on system resources at run-time

Automotive Applications



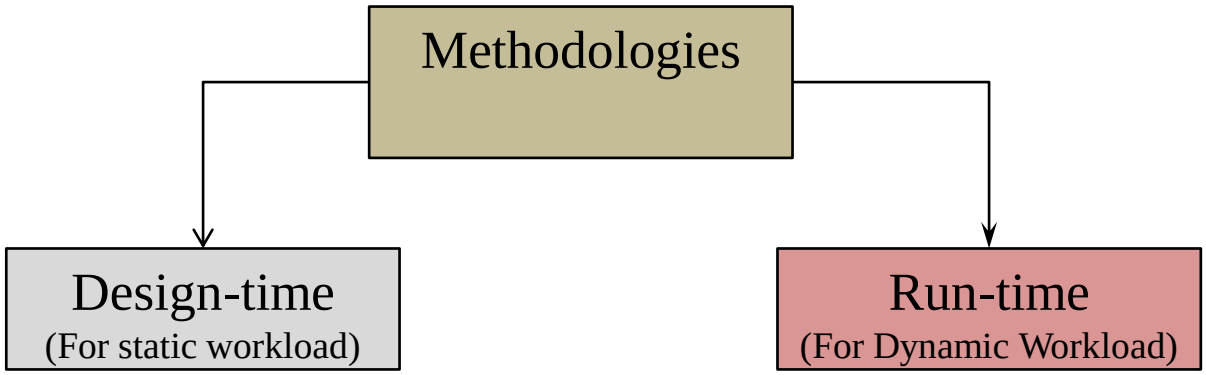
- Load active tasks during various modes without any deadline violation

Server and Cloud Computing Applications



Any other application domains you can think of Optimisations with Earlier Principles?

Optimisation methodologies



When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

Can we have any other option apart from design-time and run-time optimisations?

Yes

No

Design-time Optimisations

Design-time Optimisations

- The optimization is performed with global and thorough view of system resources
- Normally, a better quality of result is achieved than the run-time optimisations
- A huge literature falls under design-time optimisation category

Design-time Optimisation Principles

- Different well established search approaches have been extensively used
  - Simulated Annealing (SA)
  - Tabu Search
  - Integer Linear Programming (ILP)
  - Genetic Algorithm (GA)
  - …
  - Pruning strategies have been incorporated to prune the search space in order to reduce the computational costs.

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

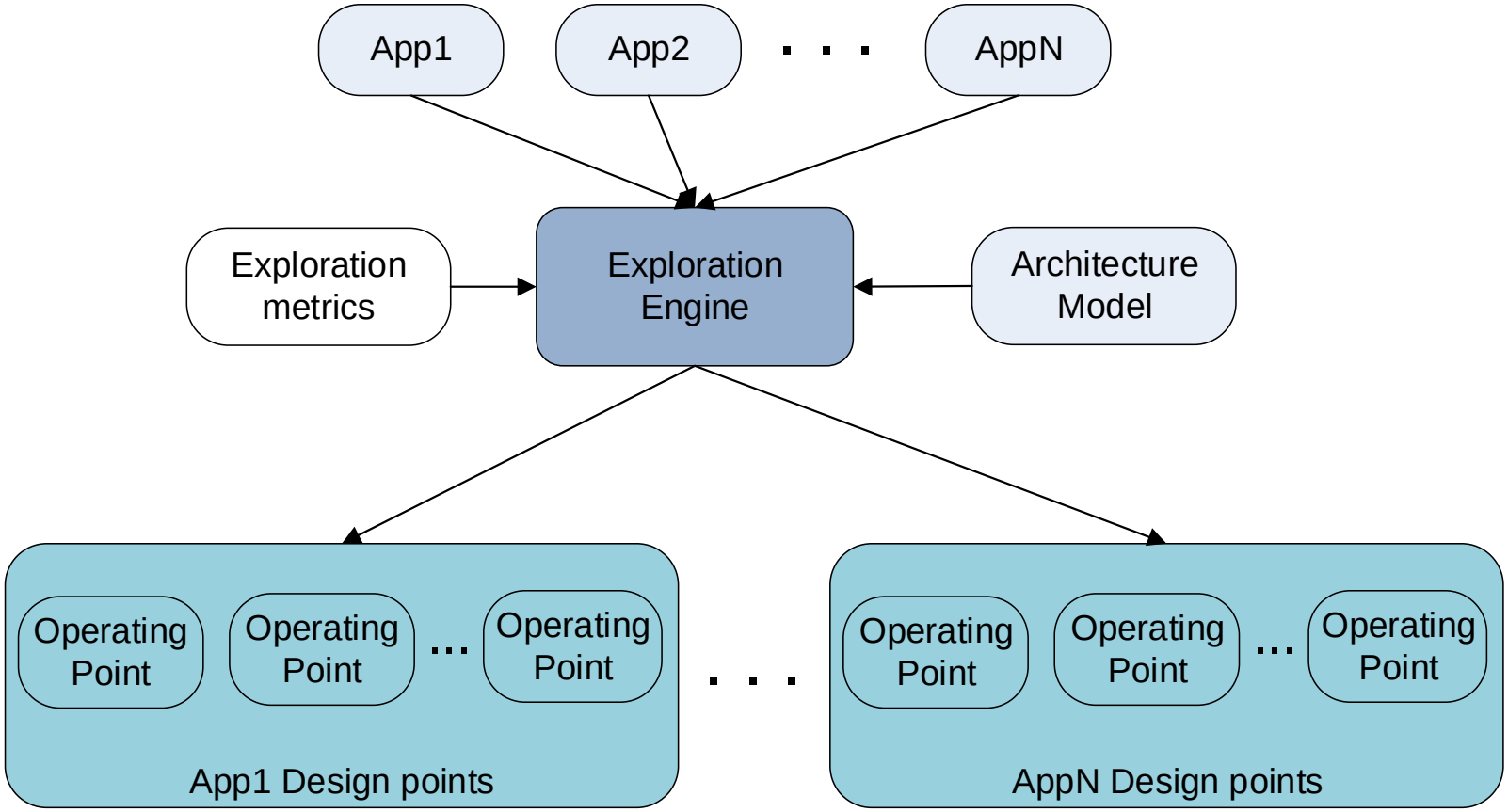
Do we have any disadvantage(s) to prune the search (design) space?

No

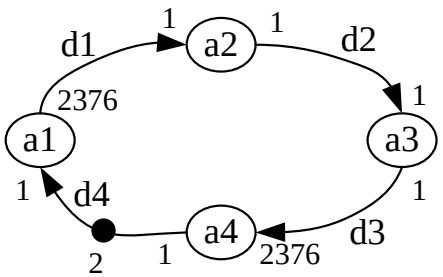
Yes

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Design-time Exhaustive DSE



Pruning-based DSE

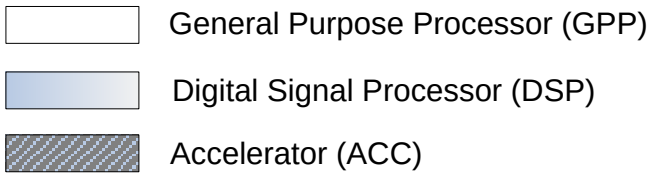


H.263 decoder

DSE

DSE Extension to Heterogeneous Tiles

- **DSE for an application modeled with 3 tasks (actors): a1, a2 and a3**

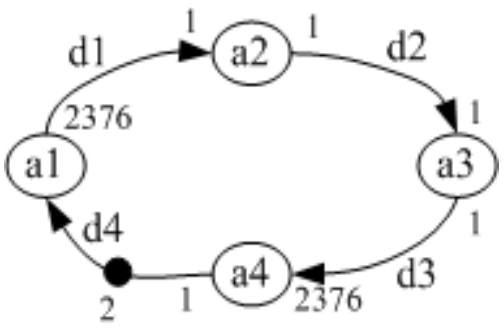


Pruning Advantage

Number of Tasks	Number of Mappings			
	Homogeneous tiles		Heterogeneous: 2 types of tiles	
	Exhaustive	Pruning	Exhaustive	Pruning
1	1	1	2	2
2	2	2	6	6
3	5	5	22	15
4	15	11	94	31
5	52	21	454	56
6	203	36	2,430	92
7	877	57	14,214	141
8	4,140	85	89,918	205
9	21,147	121	610,182	286
10	115,975	166	4,412,798	386
14	190,899,322	456	20,732,504,062	1,016

- Exhaustive evaluation takes more than 24 hours beyond 10 tasks
- Pruning based design exploration process accelerates exploration significantly

Communication-aware DSE



H.263 decoder

CADSE
(homogeneous)

Core0	Core1	Core2	Core3
a_4	a_3	a_2	a_1
a_4	a_3	a_2a_1	
a_4	a_3a_2	a_1	
a_4	$a_3a_2a_1$		
a_4a_3	a_2	a_1	
a_4a_3	a_2a_1		
a_4a_1	a_3	a_2	
a_4a_1	a_3a_2		
$a_4a_3a_2$	a_1		
$a_4a_3a_1$	a_2		
$a_4a_2a_1$	a_3		
$a_4a_3a_2a_1$			

Only Connected tasks are mapped on the same core

Facts about each design point evaluation

- The simulation time to evaluate the design points forms the real bottleneck in the DSE

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

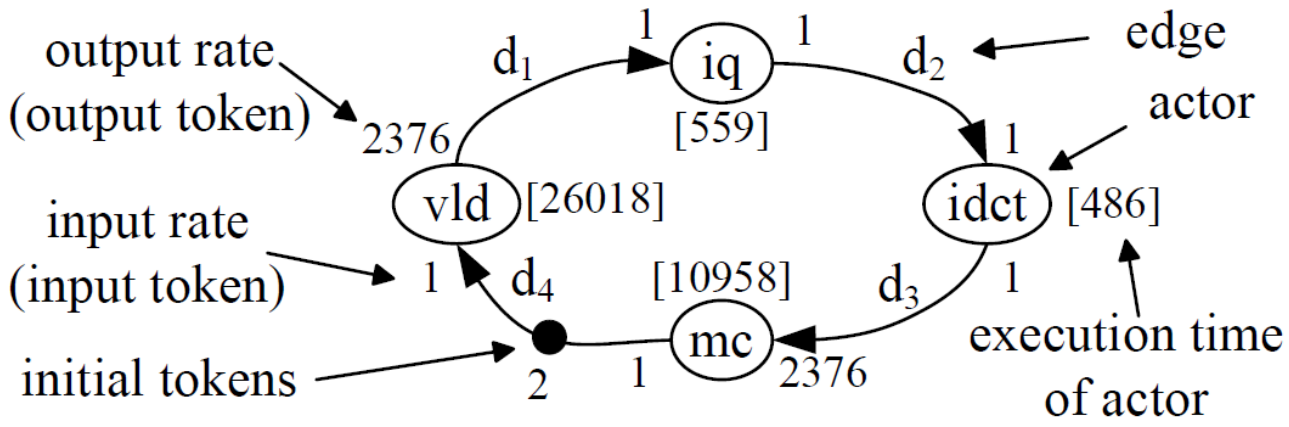
What can be done to accelerate the DSE process?

Analytical estimations can be considered.

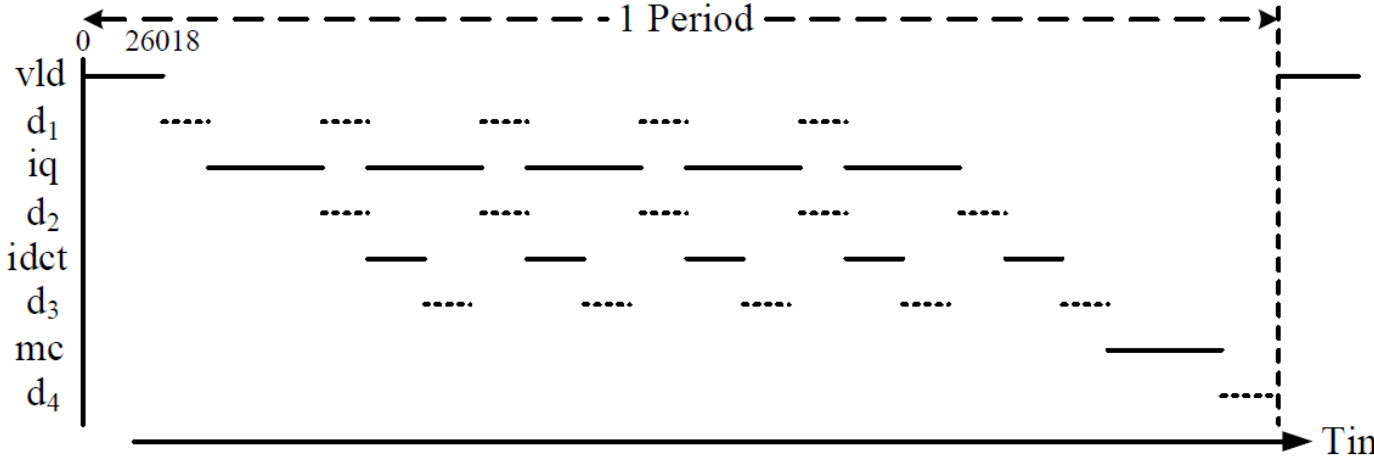
A combined simulation and estimation can be considered.

Accelerating Pruning-based DSE process

Trace-based Analysis and Simulation

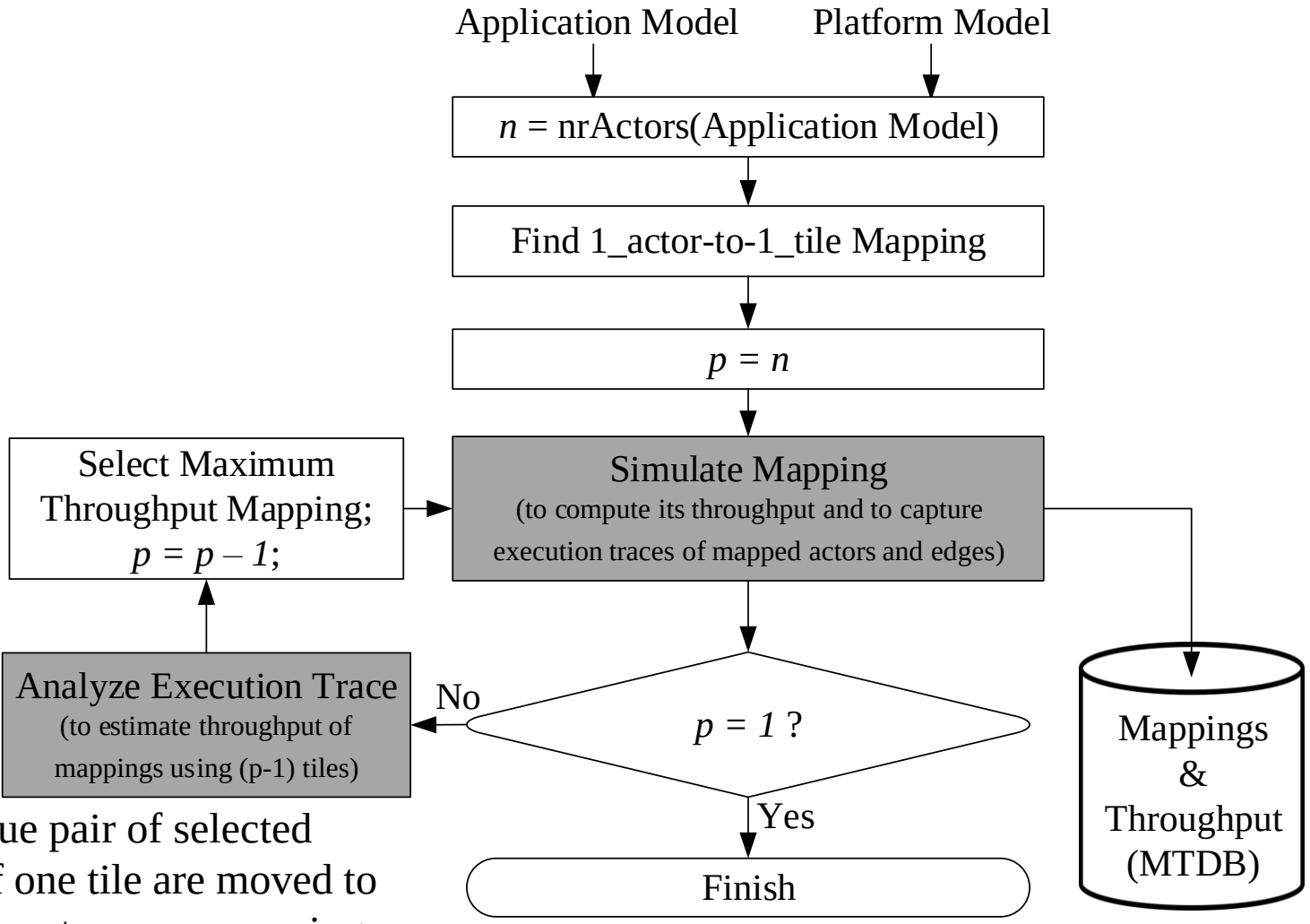


Execution Trace: (For easier realization, shown for one period while considering rates as 5 in places of 2376)



Accelerating Pruning-based DSE process

Trace-based Analysis and Simulation



For each unique pair of selected tiles, actors of one tile are moved to another to generate a new mapping that uses (p - 1) tiles

Estimation method

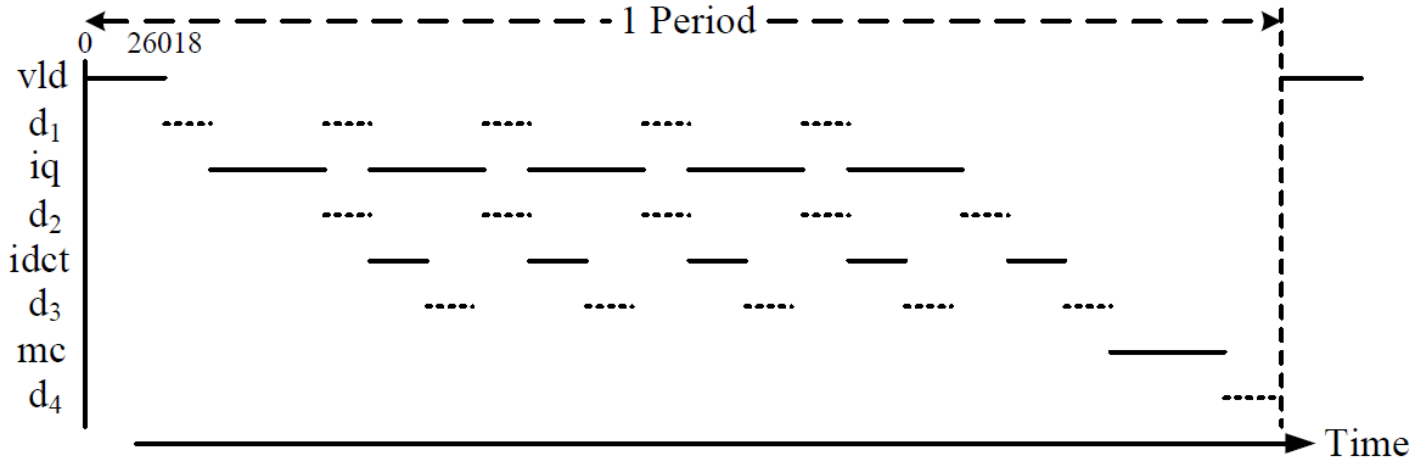
- Period (1/Throughput) of the mapping using (p - 1) tiles is estimated by utilizing period of the mapping using p tiles as follows:

$$period_{\beta} = period_{\alpha} + gain_{\alpha,\beta} + loss_{\alpha,\beta}$$

- *gain* and *loss*: are the increase and decrease in the period of the mapping using p tiles when the new mapping is generated by moving actors from one tile to another.

Example Demonstration

- **Period increase:** When parallel executing actors mapped on selected pair of tiles are forced to execute sequentially by mapping the actors on the same tile.



- **Period decrease:** When execution of the edge(s) between the selected pair of tiles is not in parallel with other actors and edges.

References

- Singh, Amit Kumar, Akash Kumar, and Thambipillai Srikanthan. "Accelerating throughput-aware runtime mapping for heterogeneous mpsocs." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 18, no. 1 (2013): 1-29.

- Singh, Amit Kumar, Muhammad Shafique, Akash Kumar, and Jörg Henkel. "Mapping on multi/many-core systems: Survey of current and emerging trends." In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1-10. IEEE, 2013.

- Singh, Amit Kumar, Anup Das, and Akash Kumar. "RAPIDITAS: RAPId design-space-exploration incorporating trace-based analysis and simulation." In *2013 Euromicro Conference on Digital System Design*, pp. 836-843. IEEE, 2013.

- …….

When poll is active, respond at **pollev.com/amitsingh510**

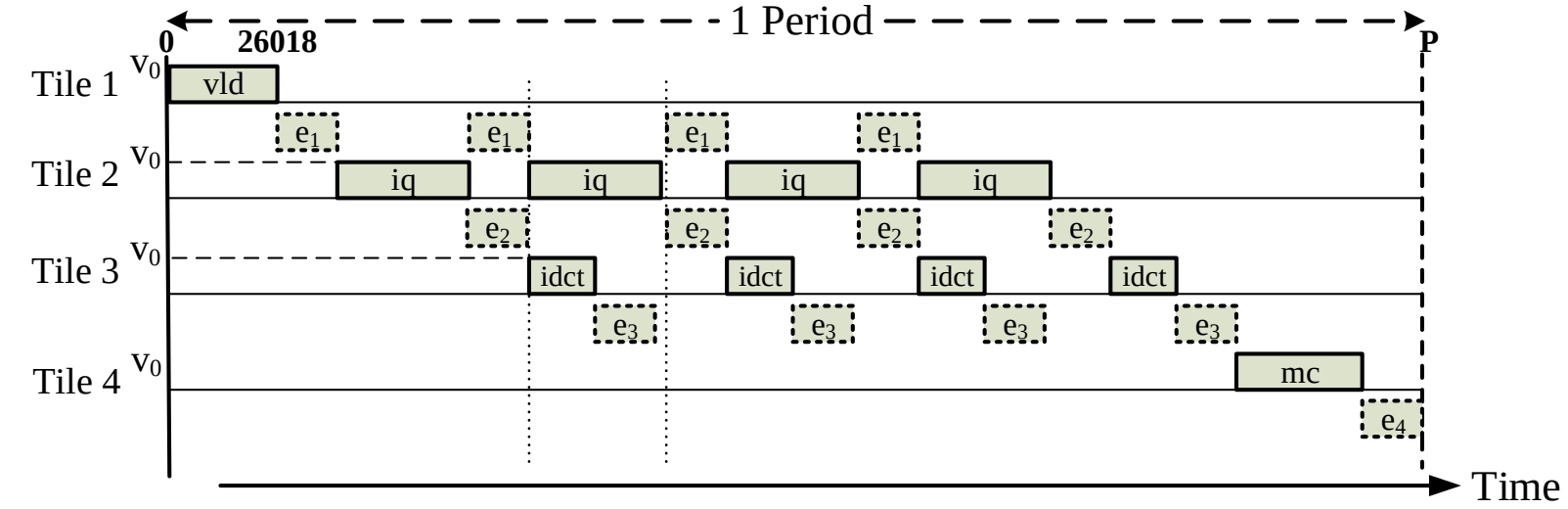
Text **AMITSINGH510** to **22333** once to join

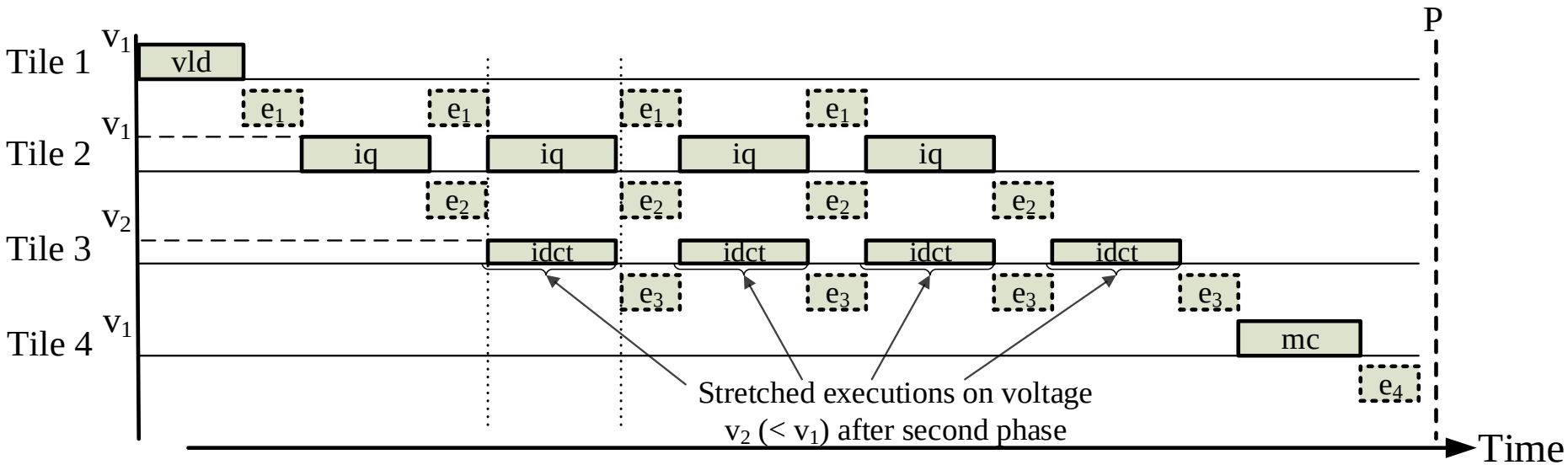
Above design-time DSE approaches take only number of cores into account. What if each core supports DVFS?

Design points will remain the same.

Design points will increase exponentially.

Example DSE with DVFS





Genetic Algorithm (GA) with an Example
(considering both Mapping and DVFS)

->
Next topic

Questions?