TOSHIBA

Leading Innovation

An Evaluation of an Energy Efficient Many-Core SoC with Parallelized Face Detection

Hiroyuki Usui, Jun Tanabe, Toru Sano, Hui Xu, and Takashi Miyamori

Toshiba Corporation, Kawasaki, Japan

Executive Summary

- **Future architecture will have many cores**
- **A key challenge : How to efficiently use them?**
- **We evaluated techniques to accelerate one type of important application (face detection)**
- **Performance scales up to 64 cores**
- **Energy efficiency is 20x better than desktop CPU**

Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Two Key Trends in Embedded Systems

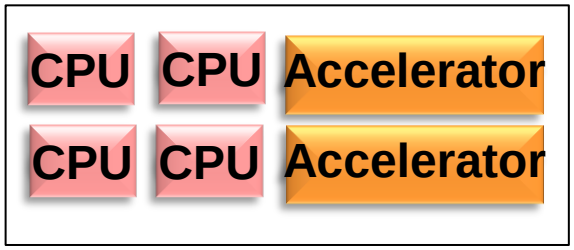
- **Trend 1 : New applications (e.g. image recognition) need more computing power while keeping low power**
- **Trend 2 : New architecture can enable much more parallelism than before**

Two Key Trends in Embedded Systems

- **Trend 1 : New applications (e.g. image recognition) need more computing power while keeping low power**
- **Trend 2 : New architecture can enable much more parallelism than before**

  Now : 500GOPS

Heterogeneous **Multi-Core**



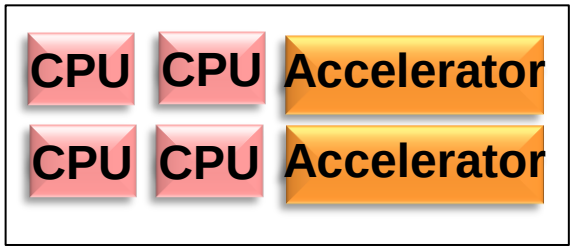
Visconti™2 [ISSCC'12]

Two Key Trends in Embedded Systems

- **Trend 1 : New applications (e.g. image recognition) need more computing power while keeping low power**
- **Trend 2 : New architecture can enable much more parallelism than before**

Now : 500GOPS

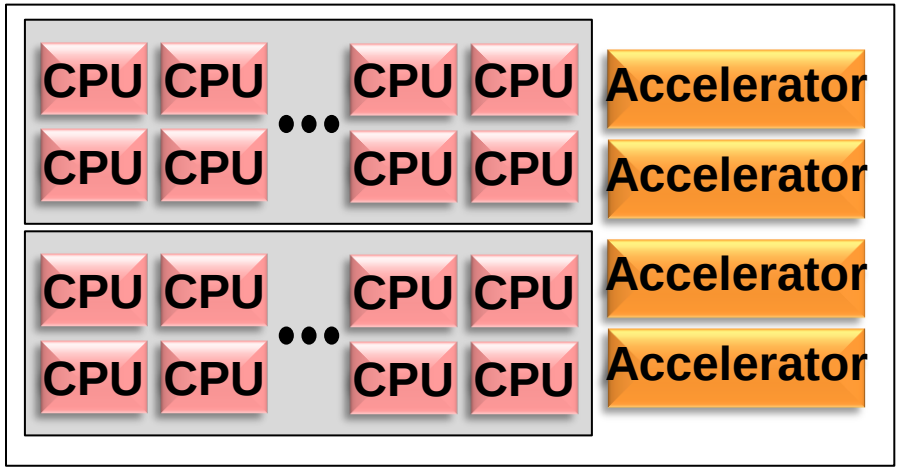
Heterogeneous **Multi-Core**

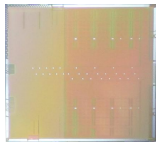


Visconti™2 [ISSCC'12]

Future : More than 1TOPS

Heterogeneous **Many-Core**





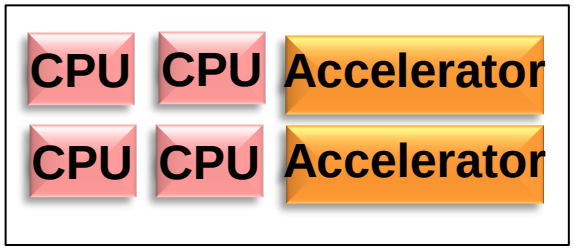
Toshiba Many-Core [VLSI Sympo. '12]

Two Key Trends in Embedded Systems

- **Trend 1 : New applications (e.g. image recognition) need more computing power while keeping low power**
- **Trend 2 : New architecture can enable much more parallelism than before**

Now : 500GOPS

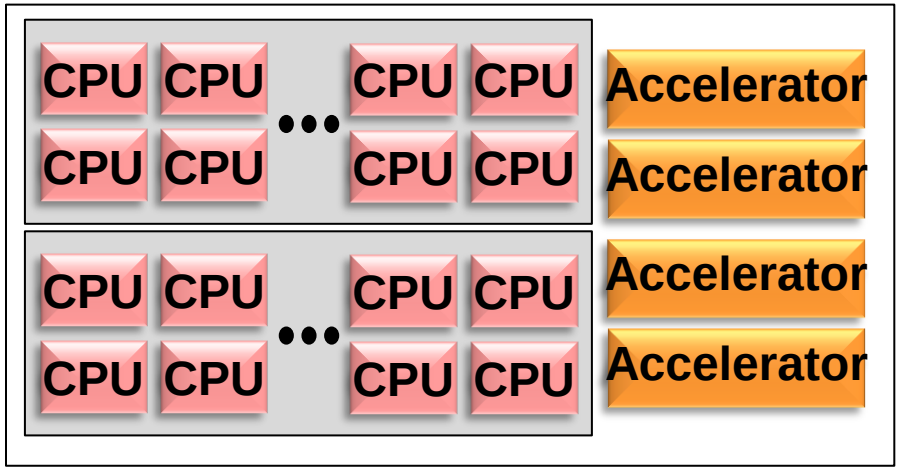
Heterogeneous **Multi-Core**

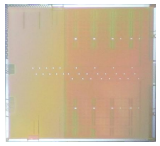


Visconti™2 [ISSCC'12]

Future : More than 1TOPS

Heterogeneous **Many-Core**

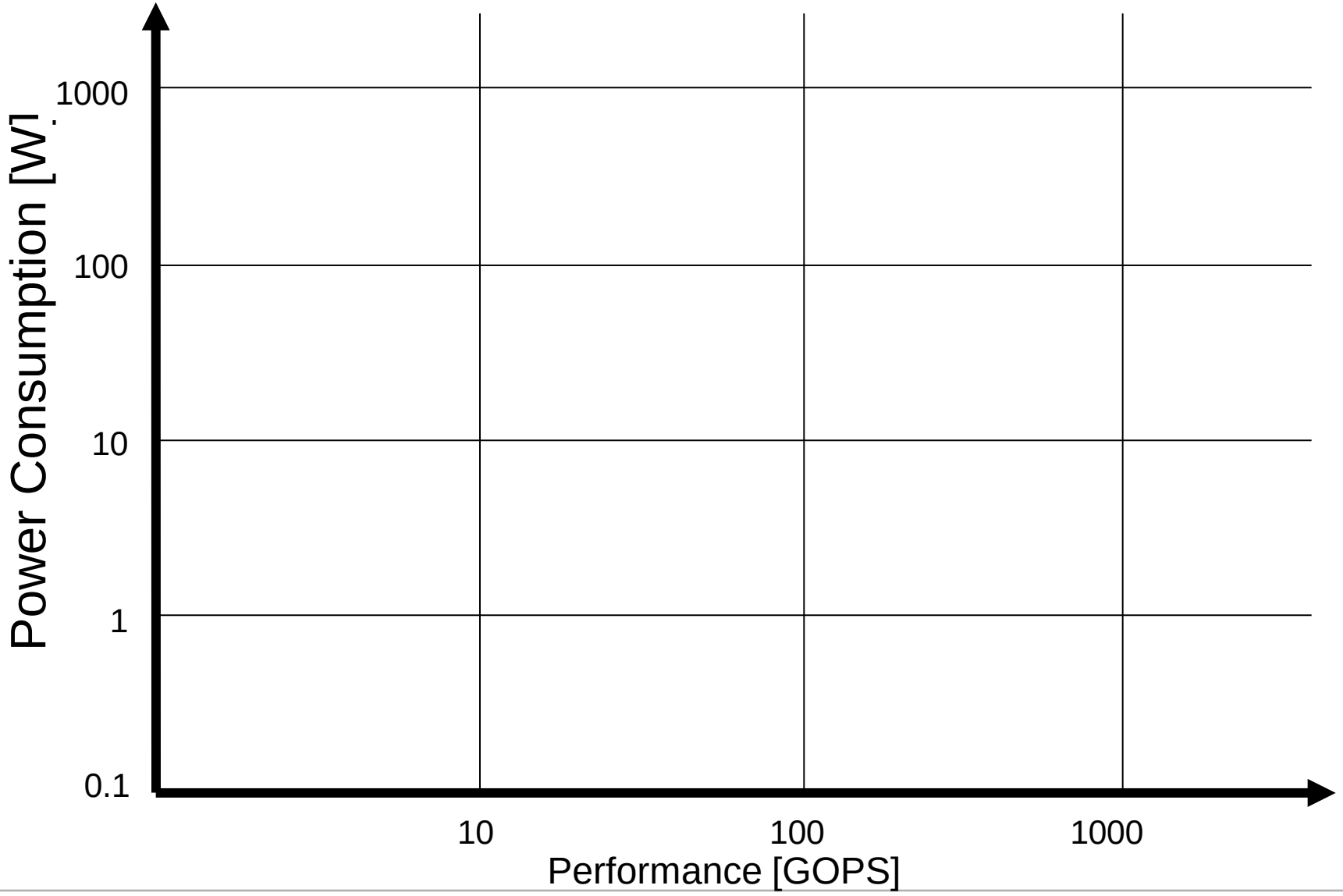




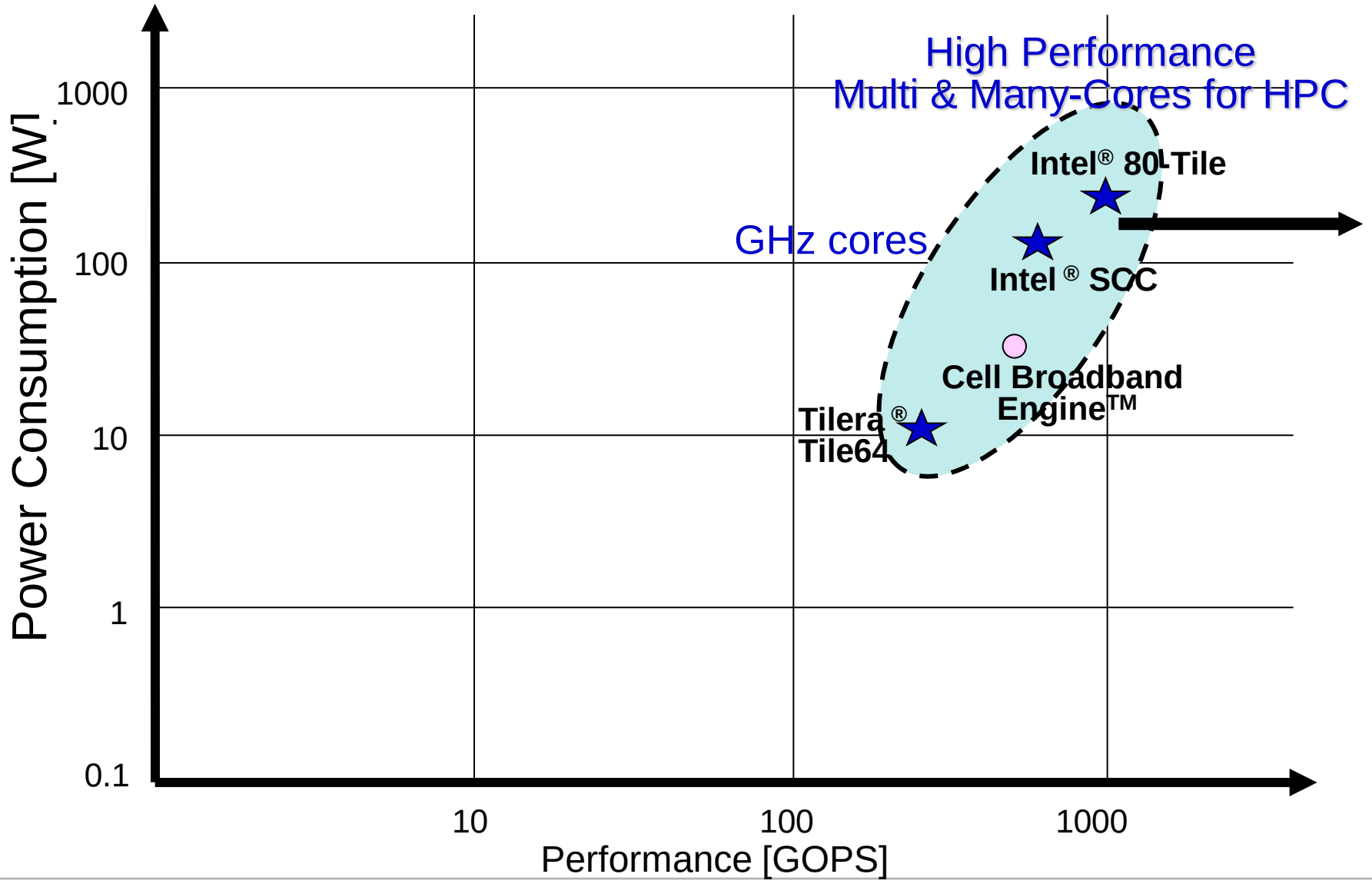
Toshiba Many-Core [VLSI Sympo. '12]

Result : A need for efficient and scalable application performance on many-core

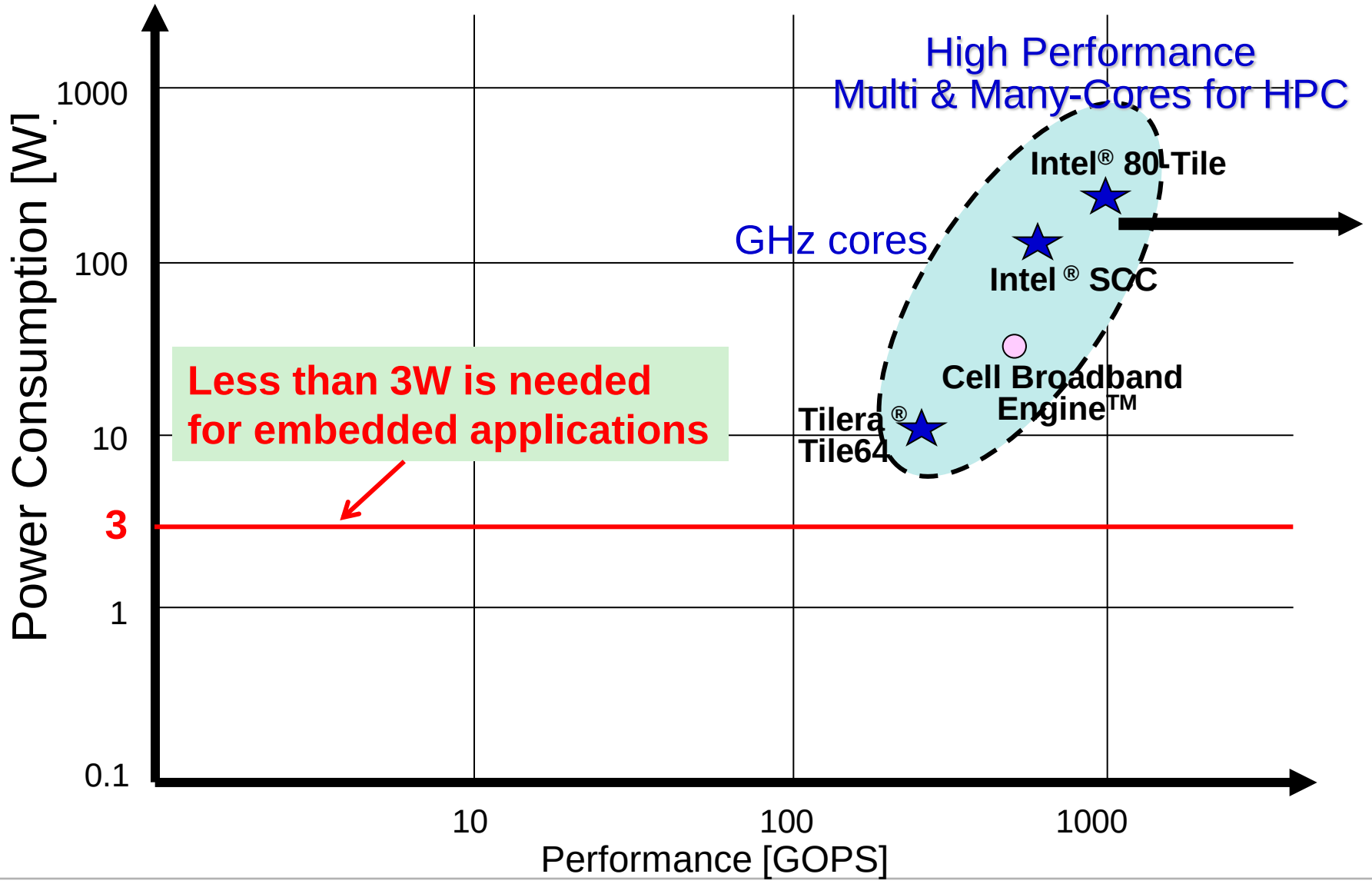
Power and Performance Target of our Many-Core



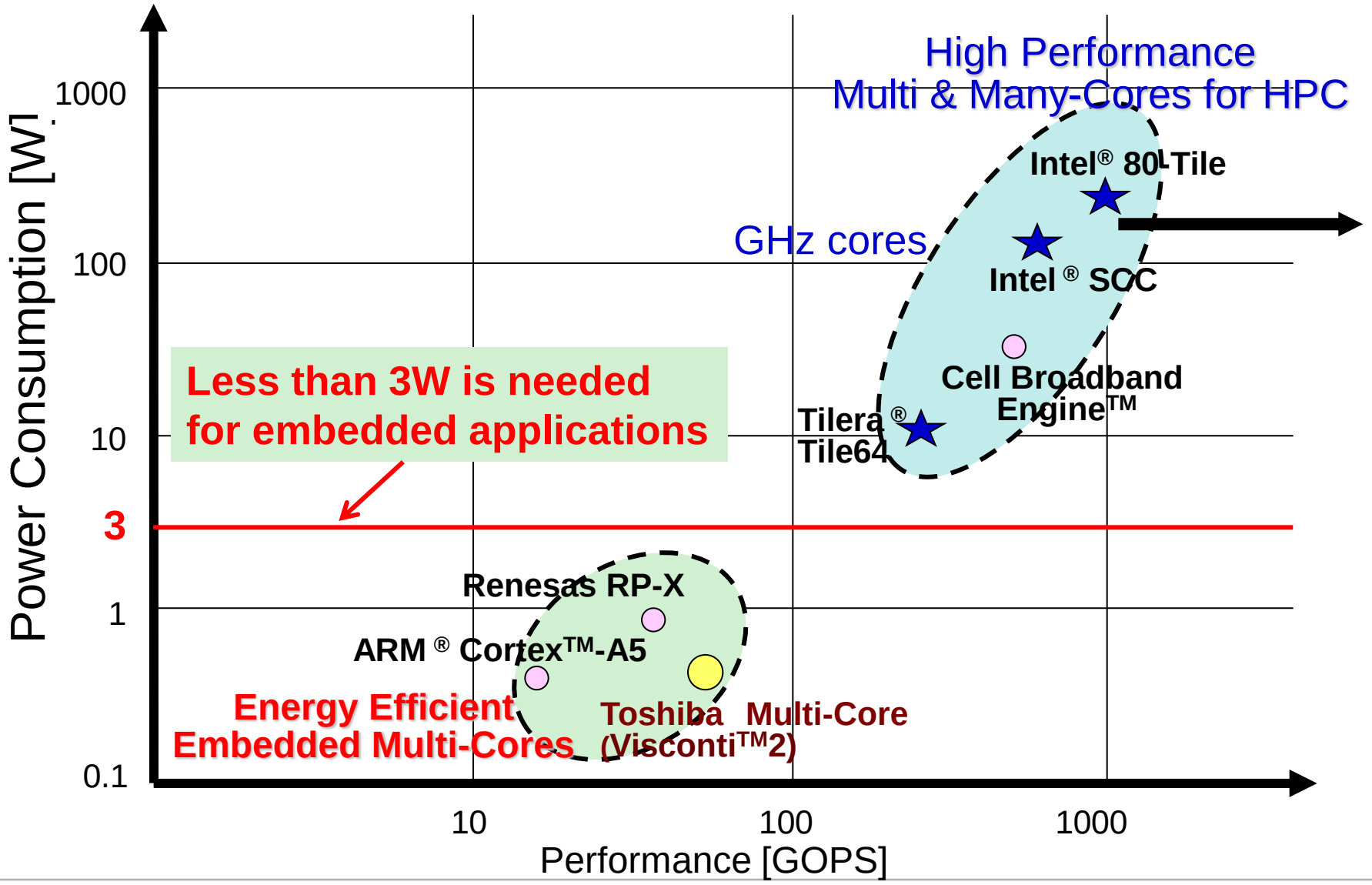
Power and Performance Target of our Many-Core



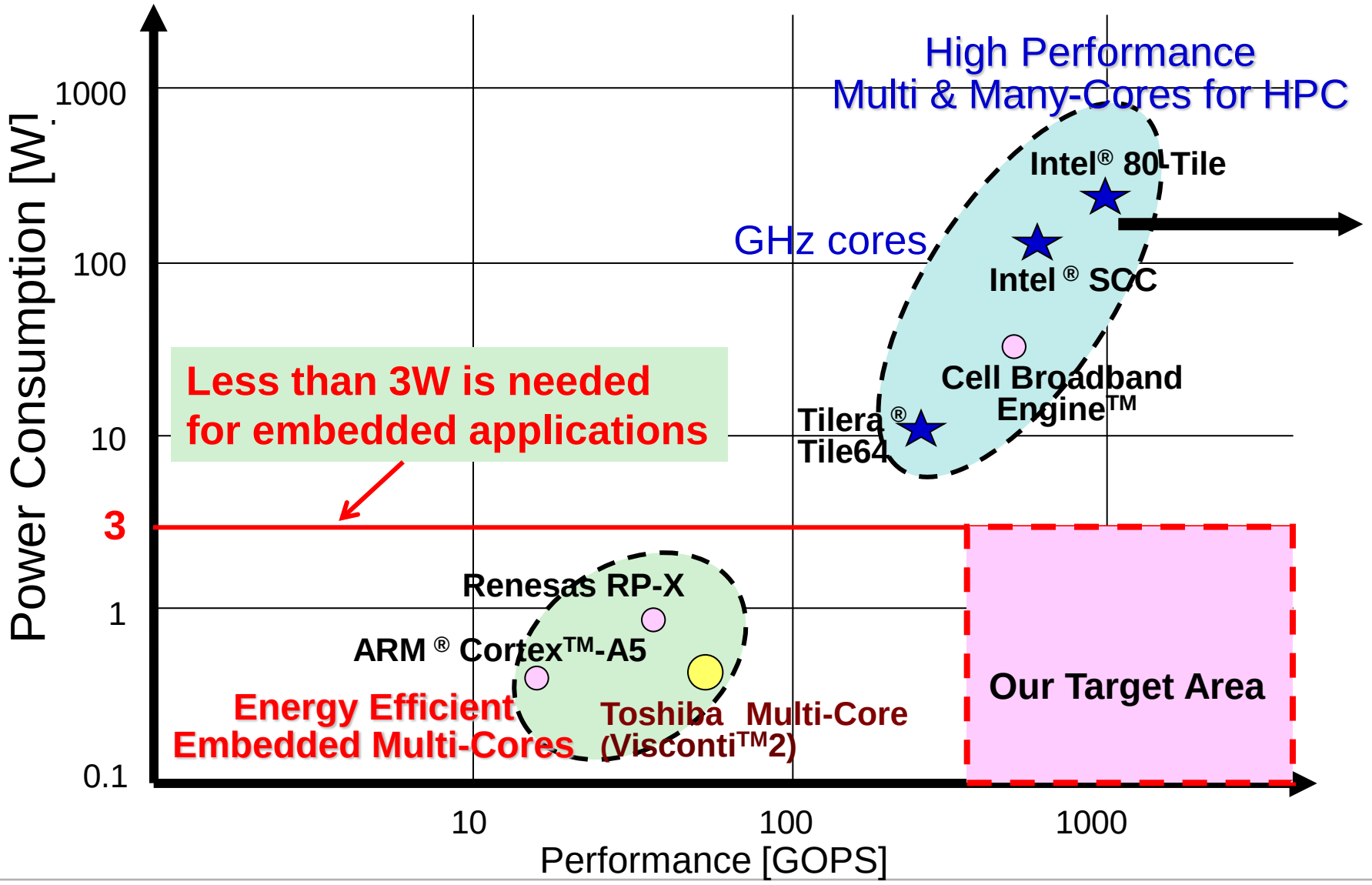
Power and Performance Target of our Many-Core



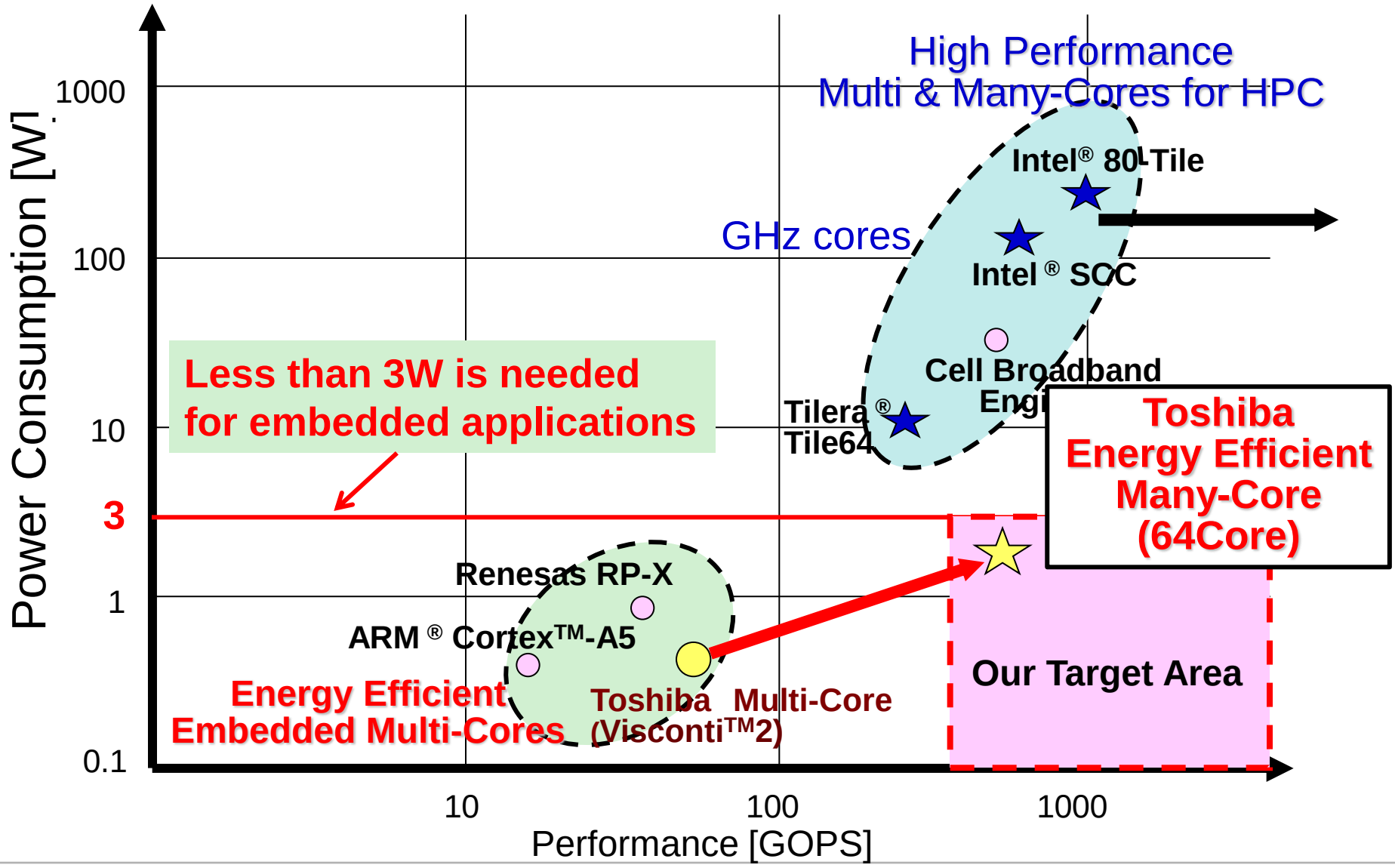
Power and Performance Target of our Many-Core



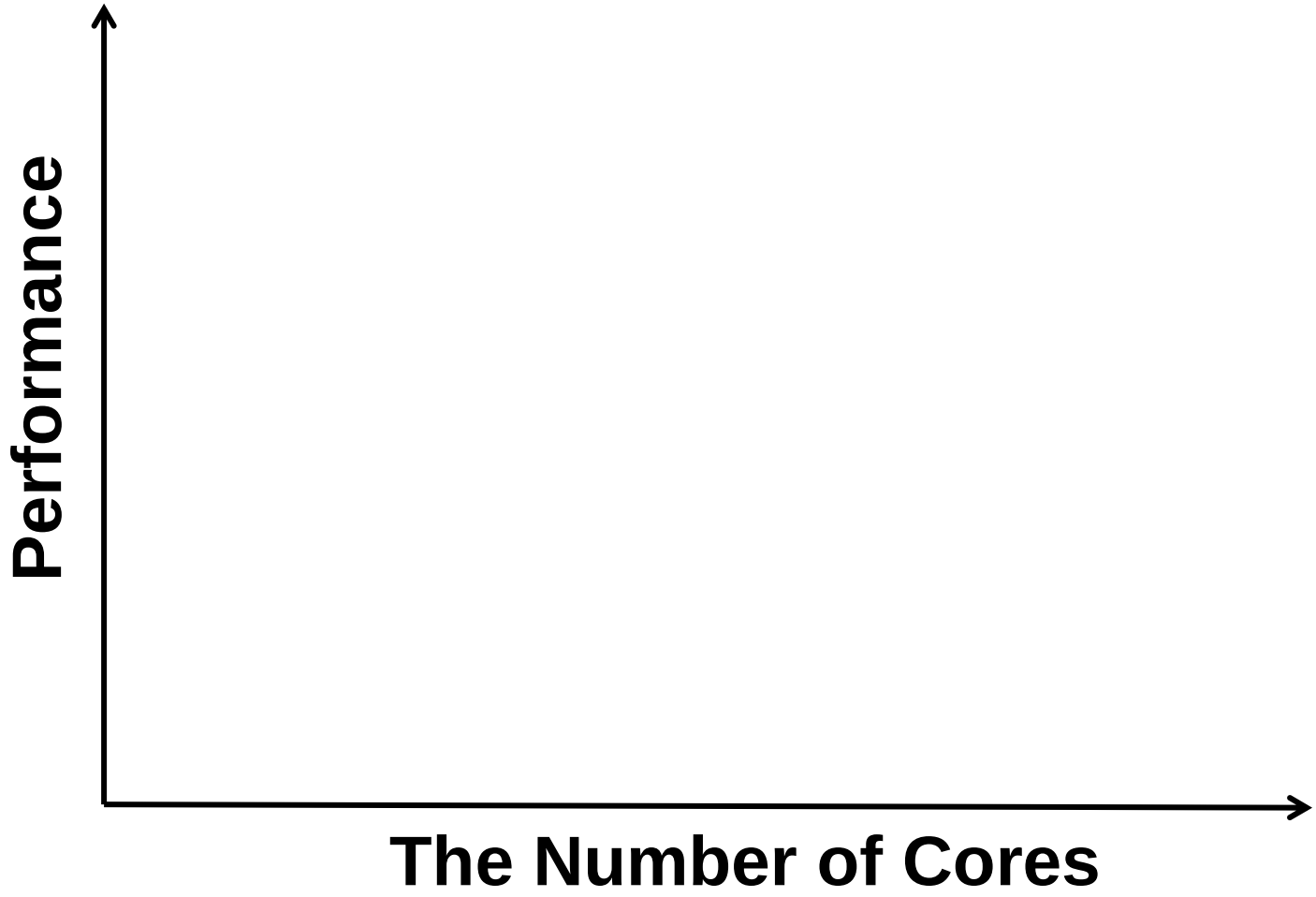
Power and Performance Target of our Many-Core



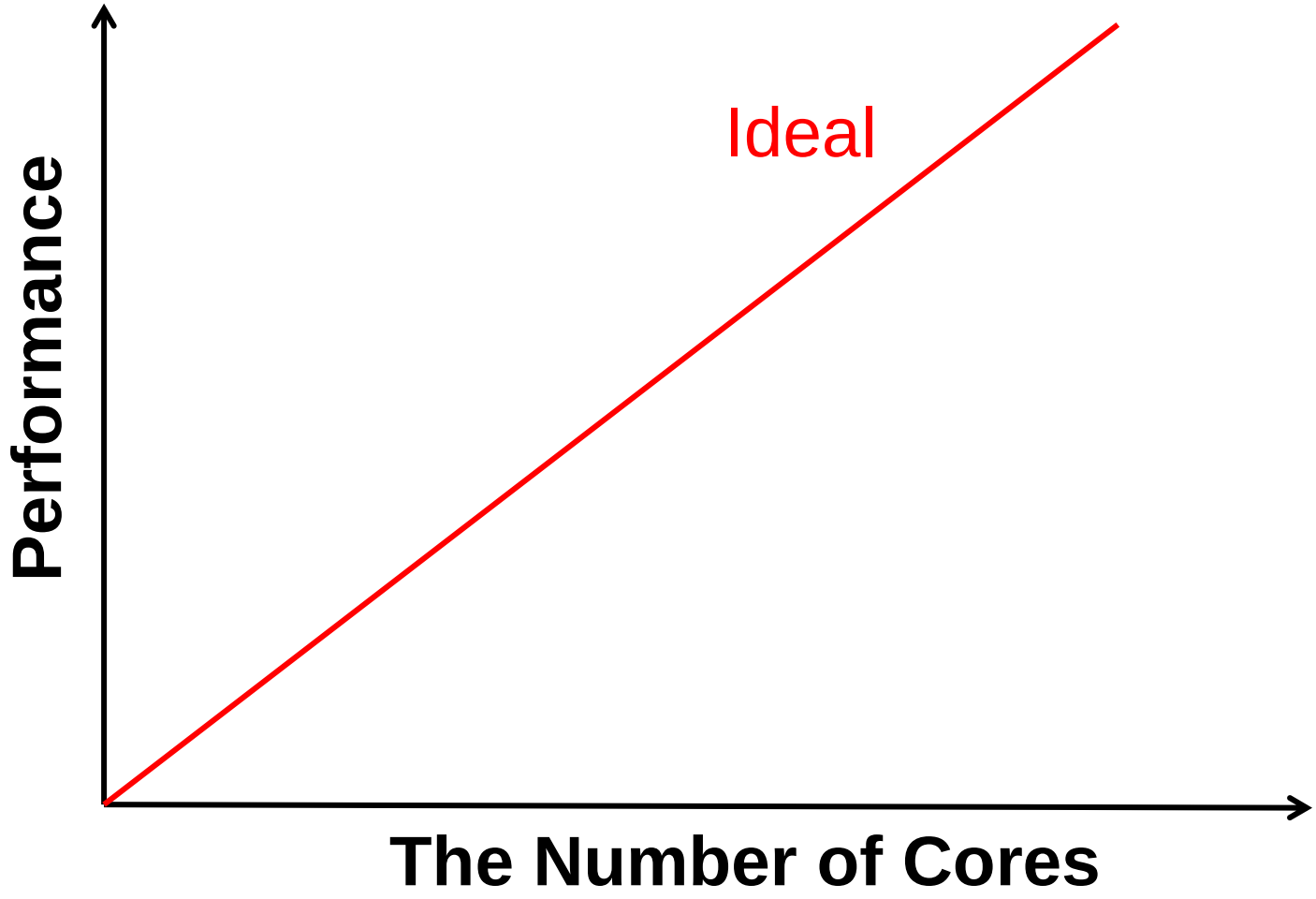
Power and Performance Target of our Many-Core



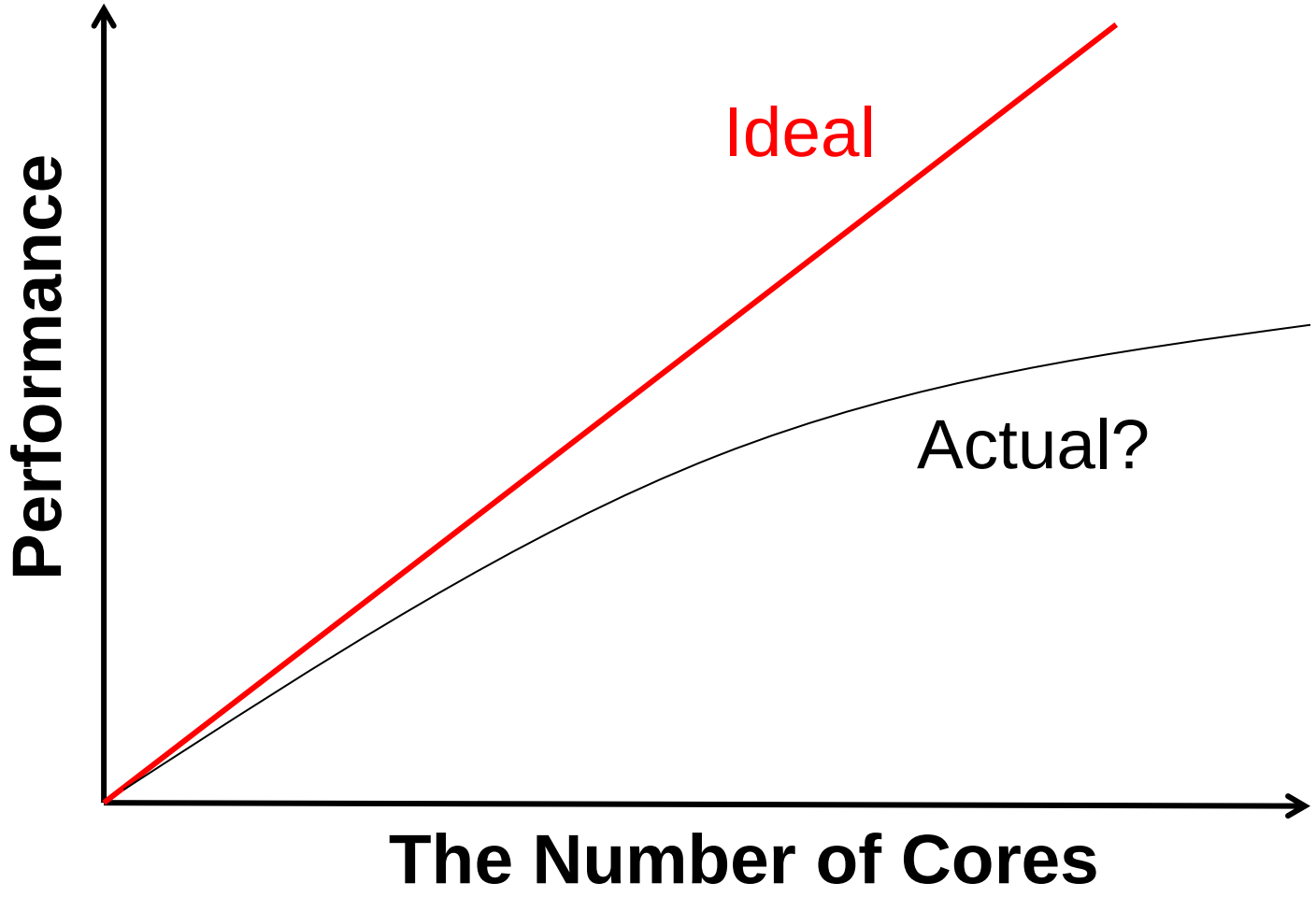
Many-Core Scalability



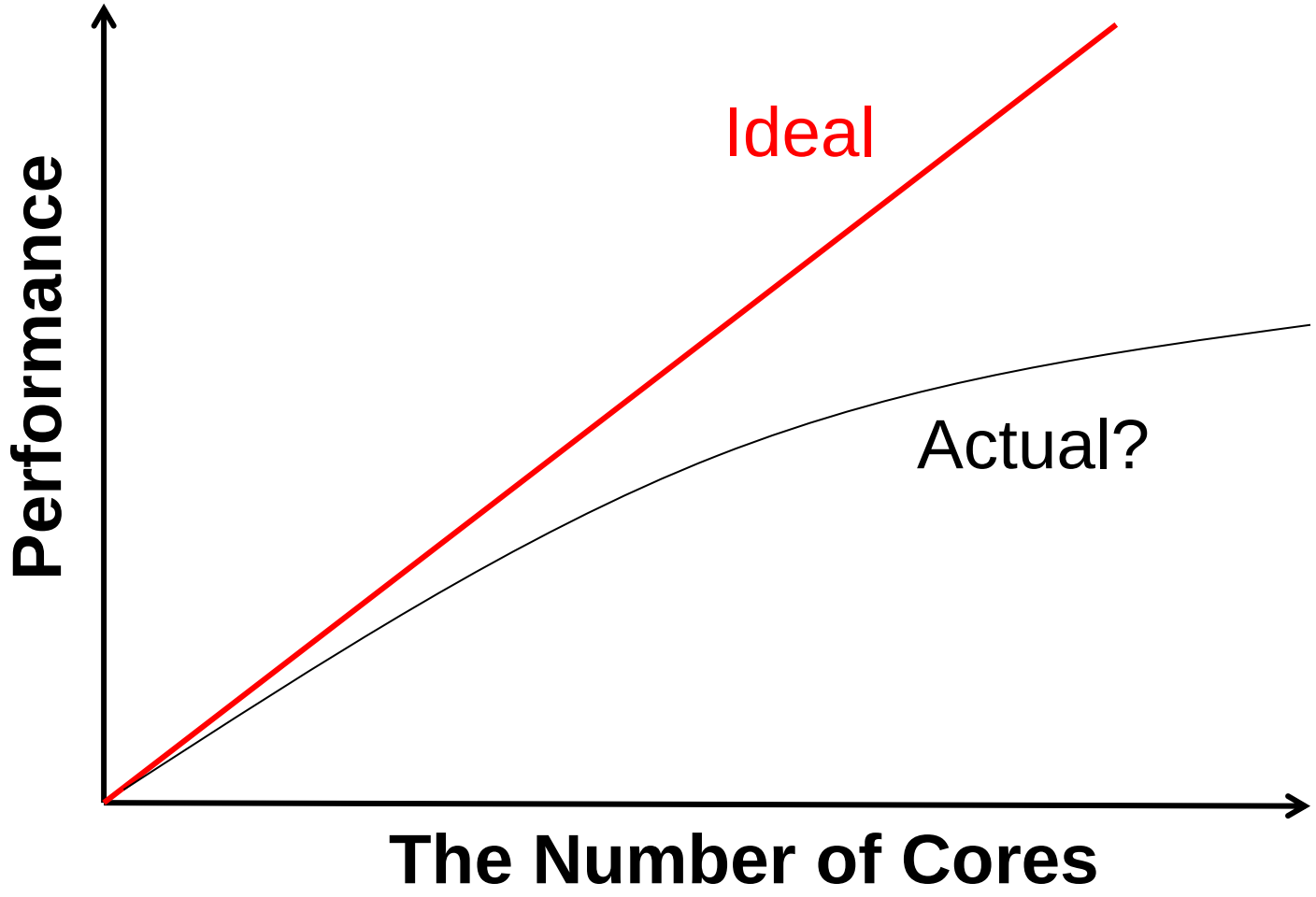
Many-Core Scalability



Many-Core Scalability



Many-Core Scalability



Can we achieve good performance scaling-up on face detection?

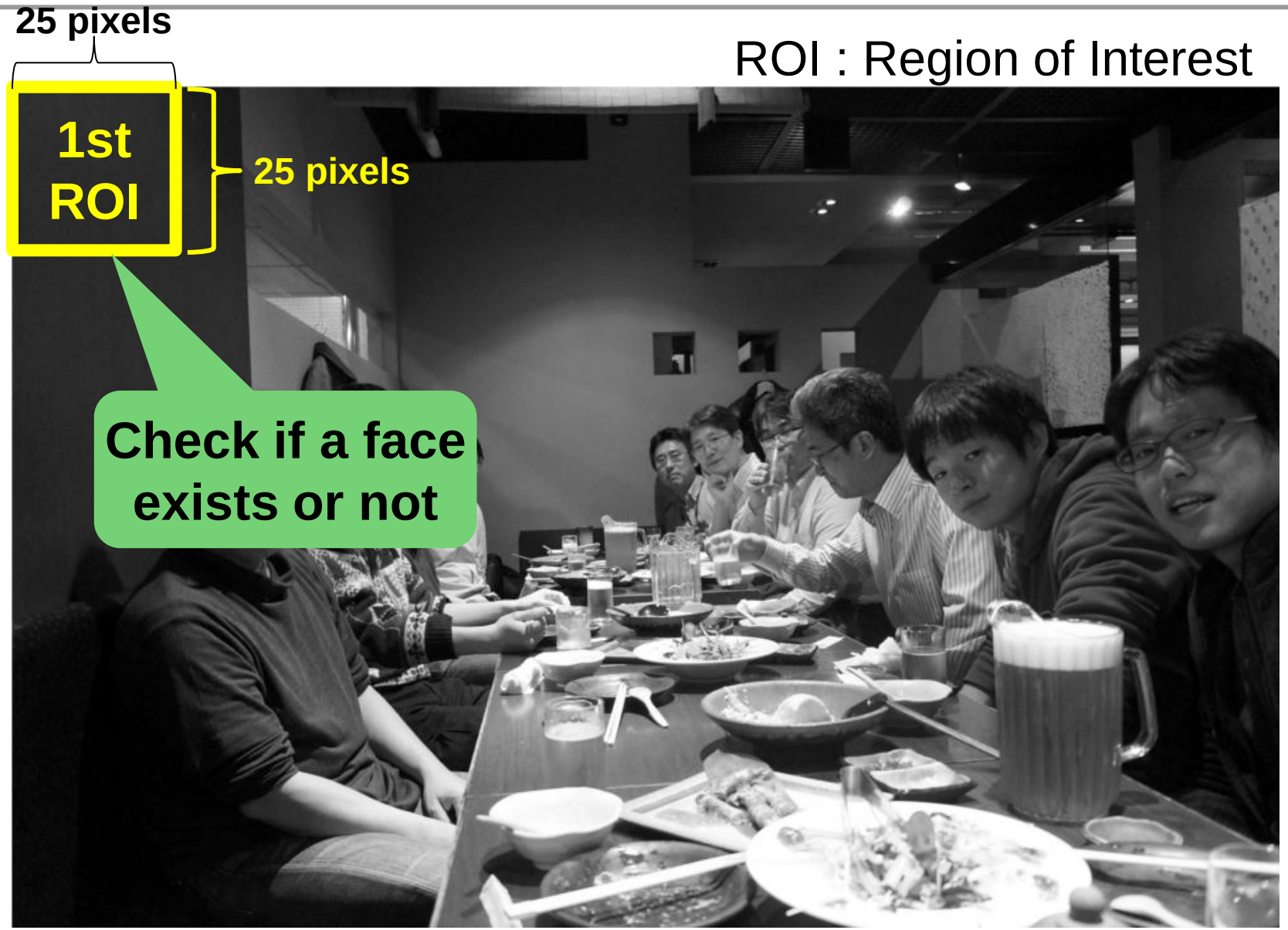
Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Face Detection

Face Detection

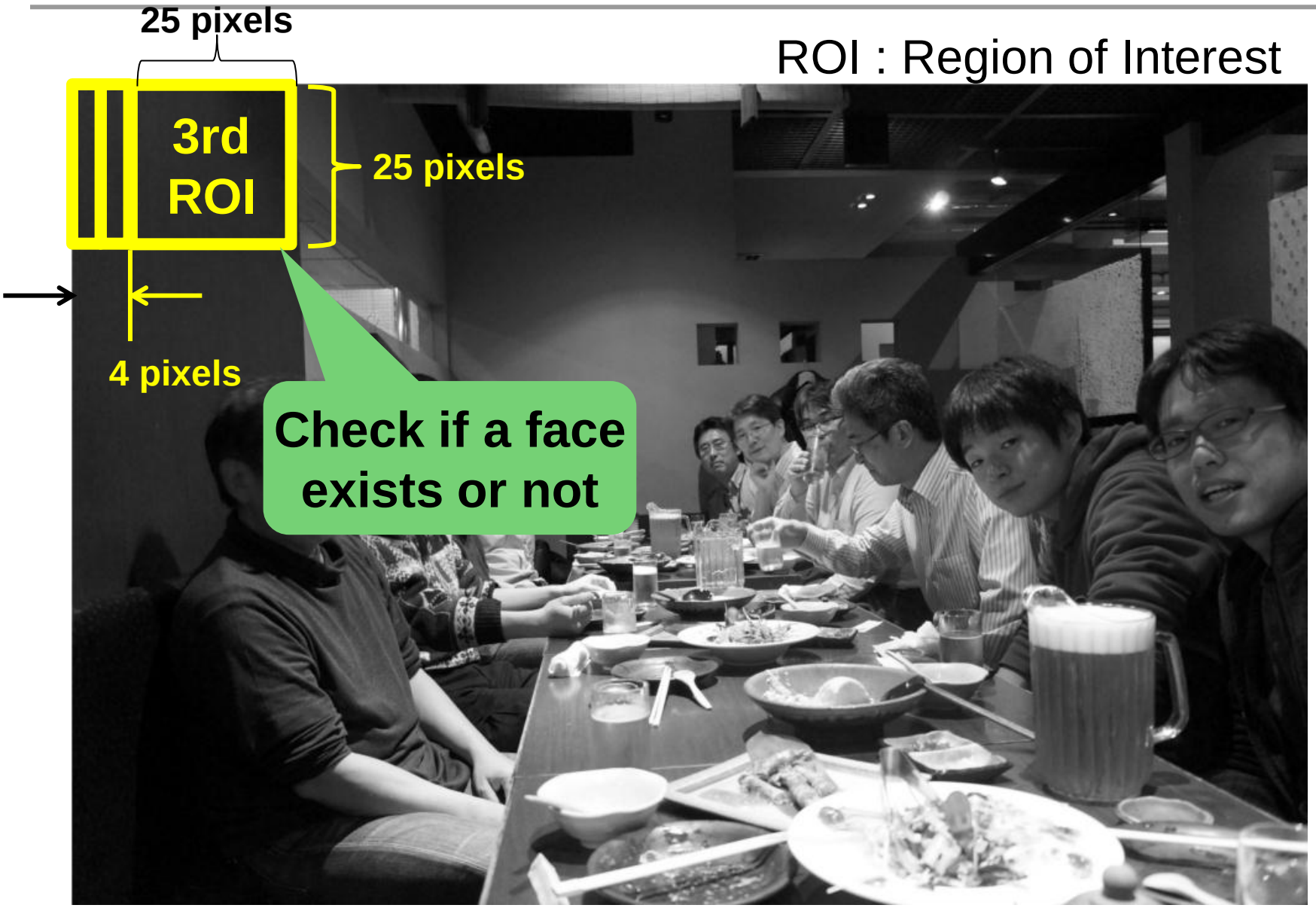
ROI : Region of Interest



Face Detection

ROI : Region of Interest

Face Detection



ROI : Region of Interest

Face Detection

ROI : Region of Interest

Face Detection

ROI : Region of Interest

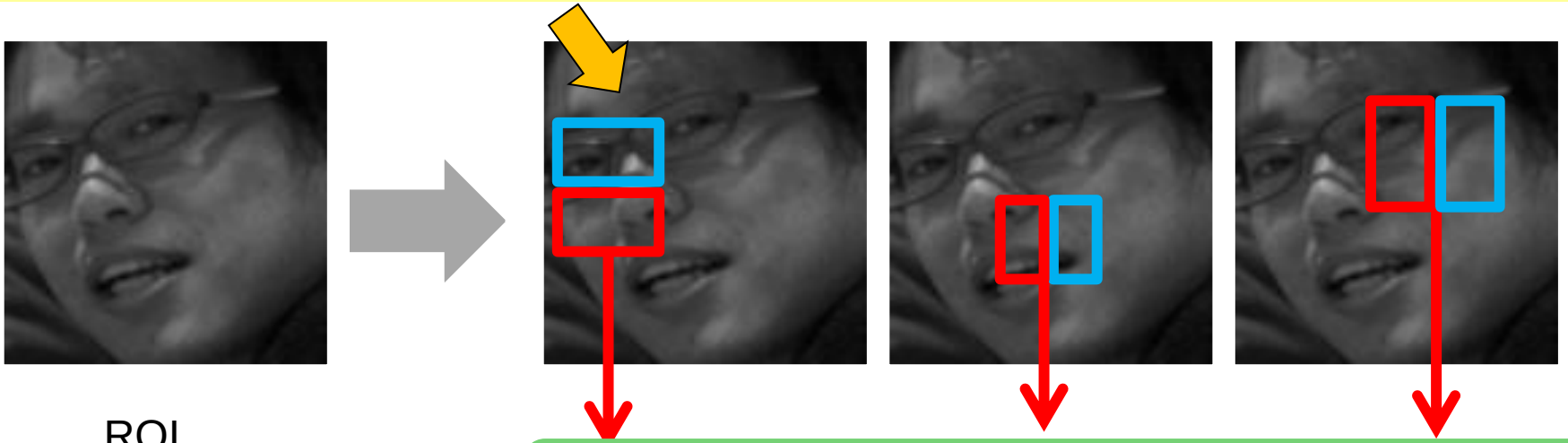
Face Detection

ROI : Region of Interest

Joint Haar-Like Features [ICCV ‘05]

- **Extension to widely-used Viola and John’s Method [CVPR ‘01] (using Haar-like features)**

Haar-like feature : Difference of image intensities between blue and red rectangles.



ROI
(Region Of Interest)

Compared to each threshold
If greater than : 1, otherwise : 0

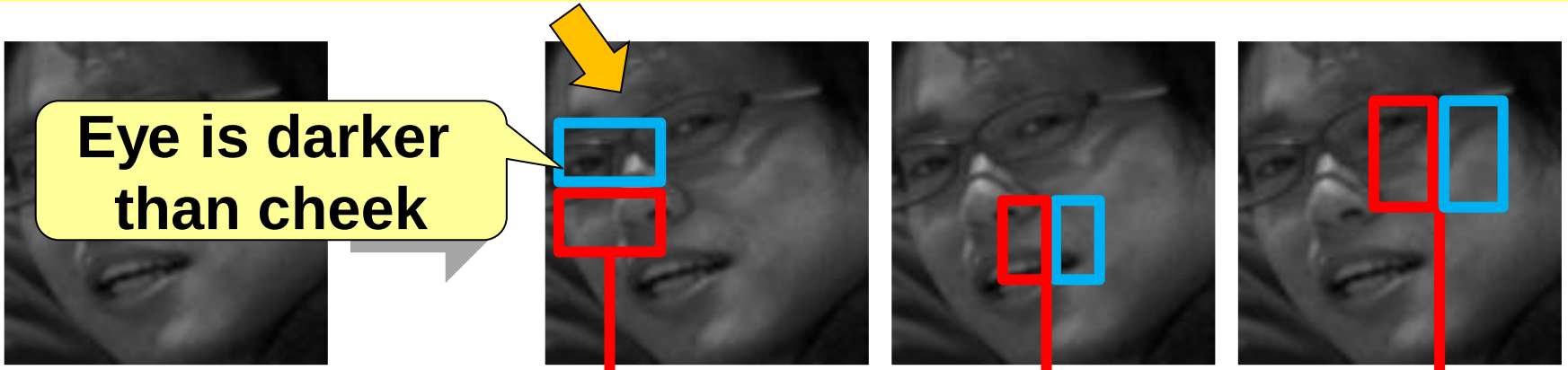
Joint Haar-Like Feature

1 **1** **0**

Joint Haar-Like Features [ICCV ‘05]

- **Extension to widely-used Viola and John’s Method [CVPR ‘01] (using Haar-like features)**

Haar-like feature : Difference of image intensities between blue and red rectangles.



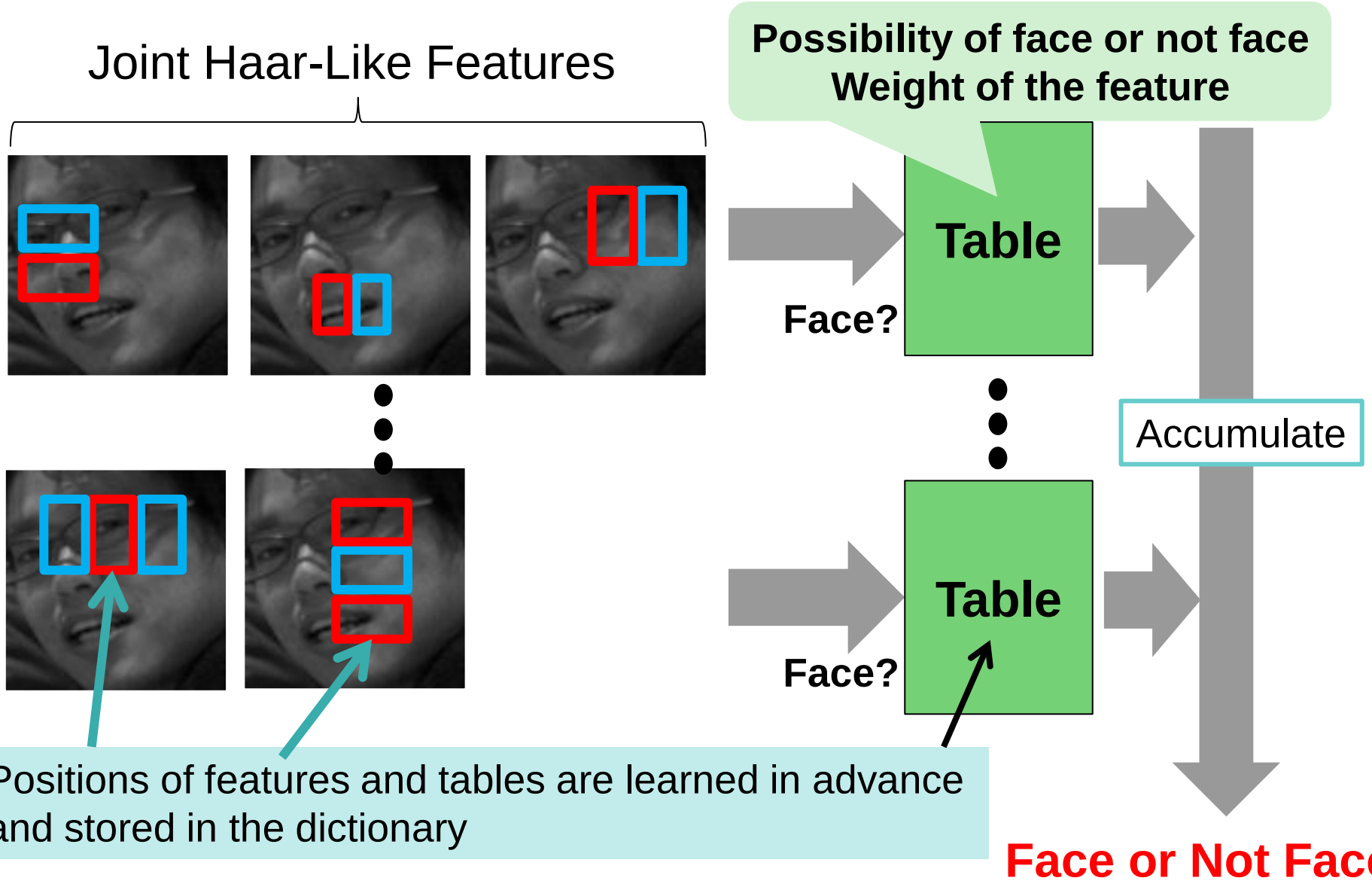
ROI
(Region Of Interest)

Compared to each threshold
If greater than : 1, otherwise : 0

Joint Haar-Like Feature

1 **1** **0**

Classifier using Joint Haar-Like Features



Characteristics of Face Detection

- **Face detection for each ROI can be executed in parallel**
- **There are a lot of ROIs in an image**
  - 3M ROIs when image size is 4000x3200

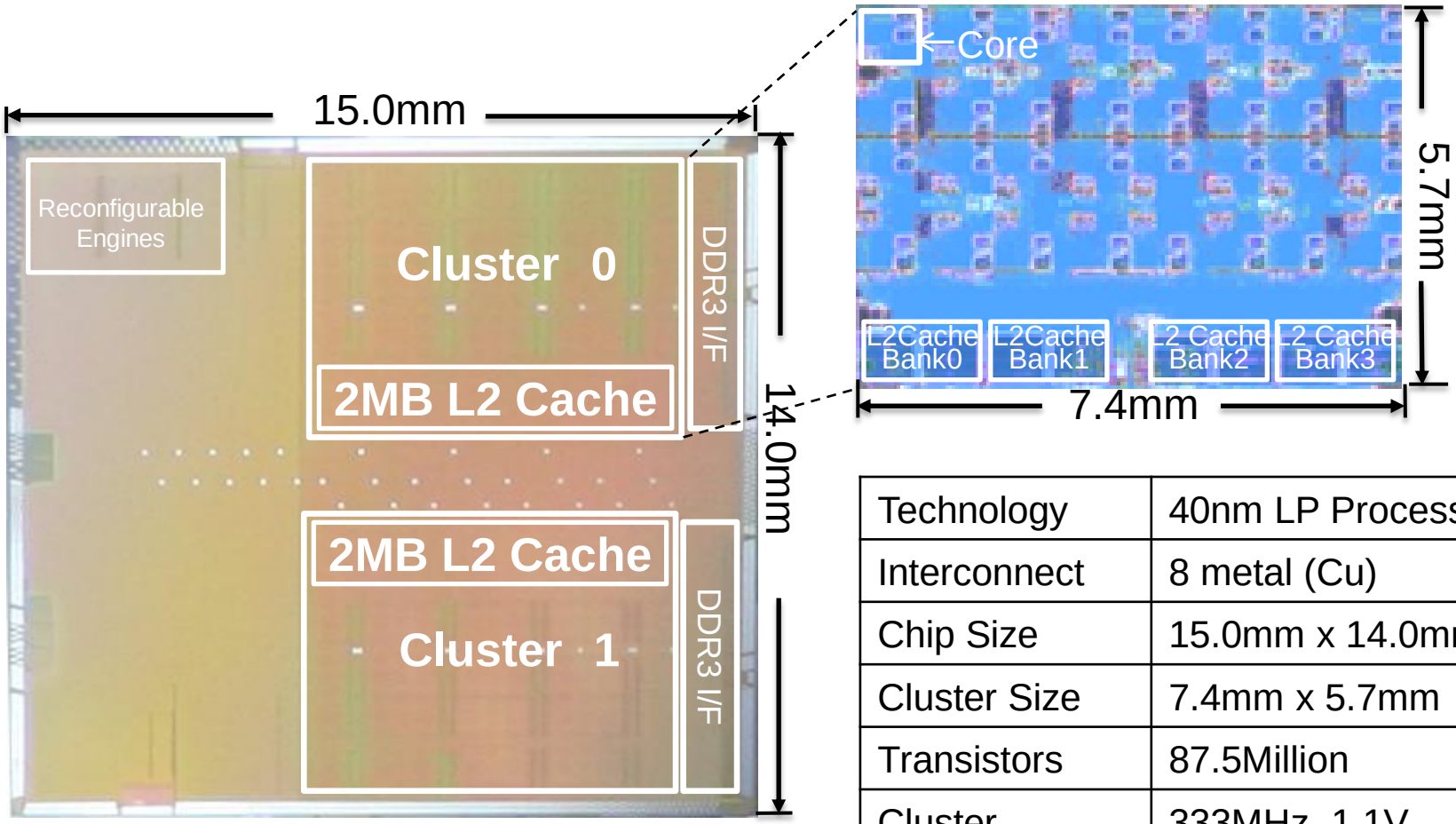
- **A lot of coarse grain thread parallelism based on ROIs**
  - Overhead of thread scheduling can be minimized

Many-core is good for face detection !

Outline

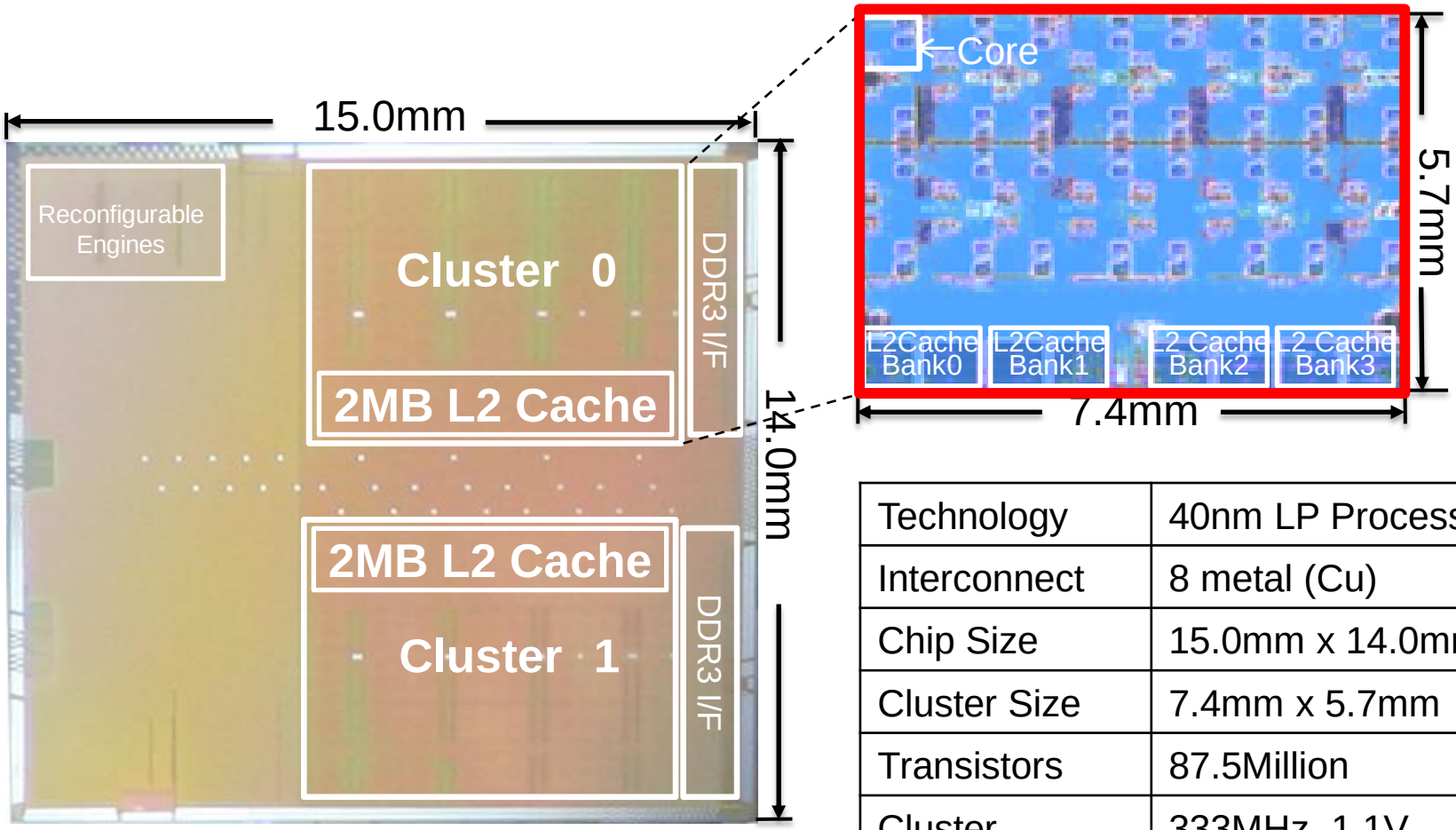
- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Chip Micrograph and Features



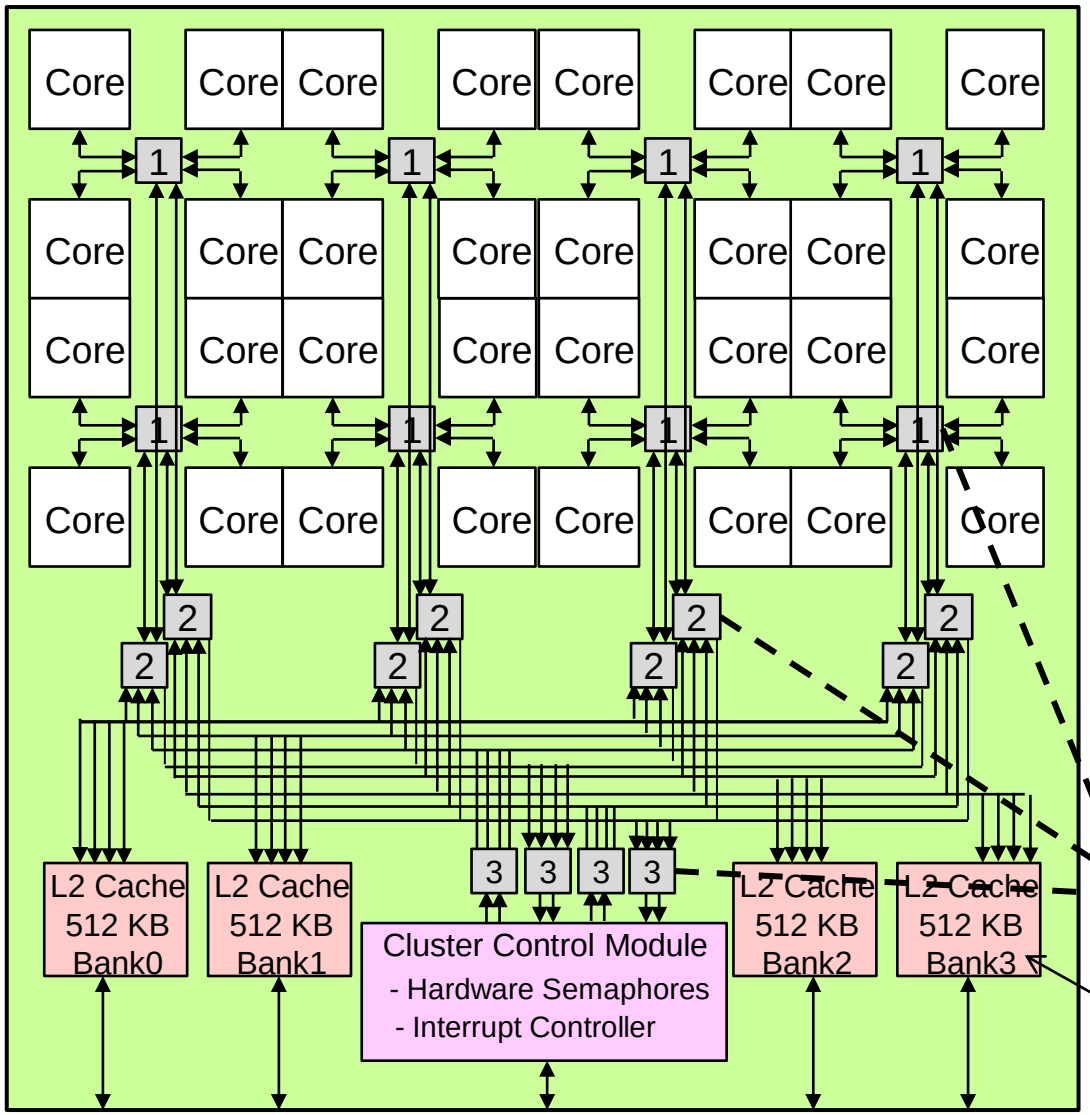
Technology	40nm LP Process
Interconnect	8 metal (Cu)
Chip Size	15.0mm x 14.0mm
Cluster Size	7.4mm x 5.7mm
Transistors	87.5Million
Cluster Frequency	333MHz, 1.1V
Package	1369-pin FCBGA

Chip Micrograph and Features

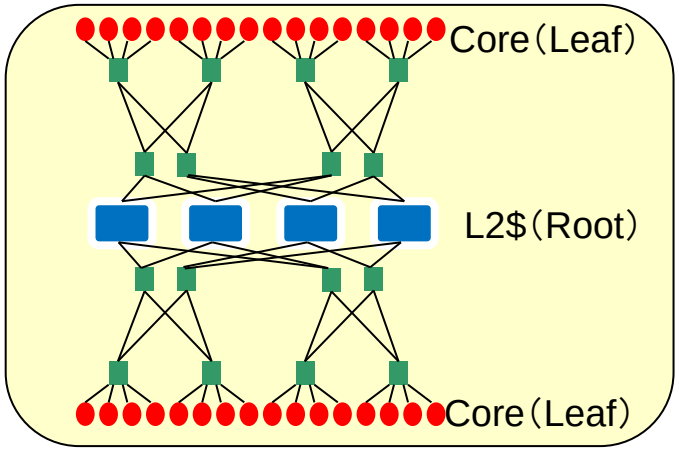


Technology	40nm LP Process
Interconnect	8 metal (Cu)
Chip Size	15.0mm x 14.0mm
Cluster Size	7.4mm x 5.7mm
Transistors	87.5Million
Cluster Frequency	333MHz, 1.1V
Package	1369-pin FCBGA

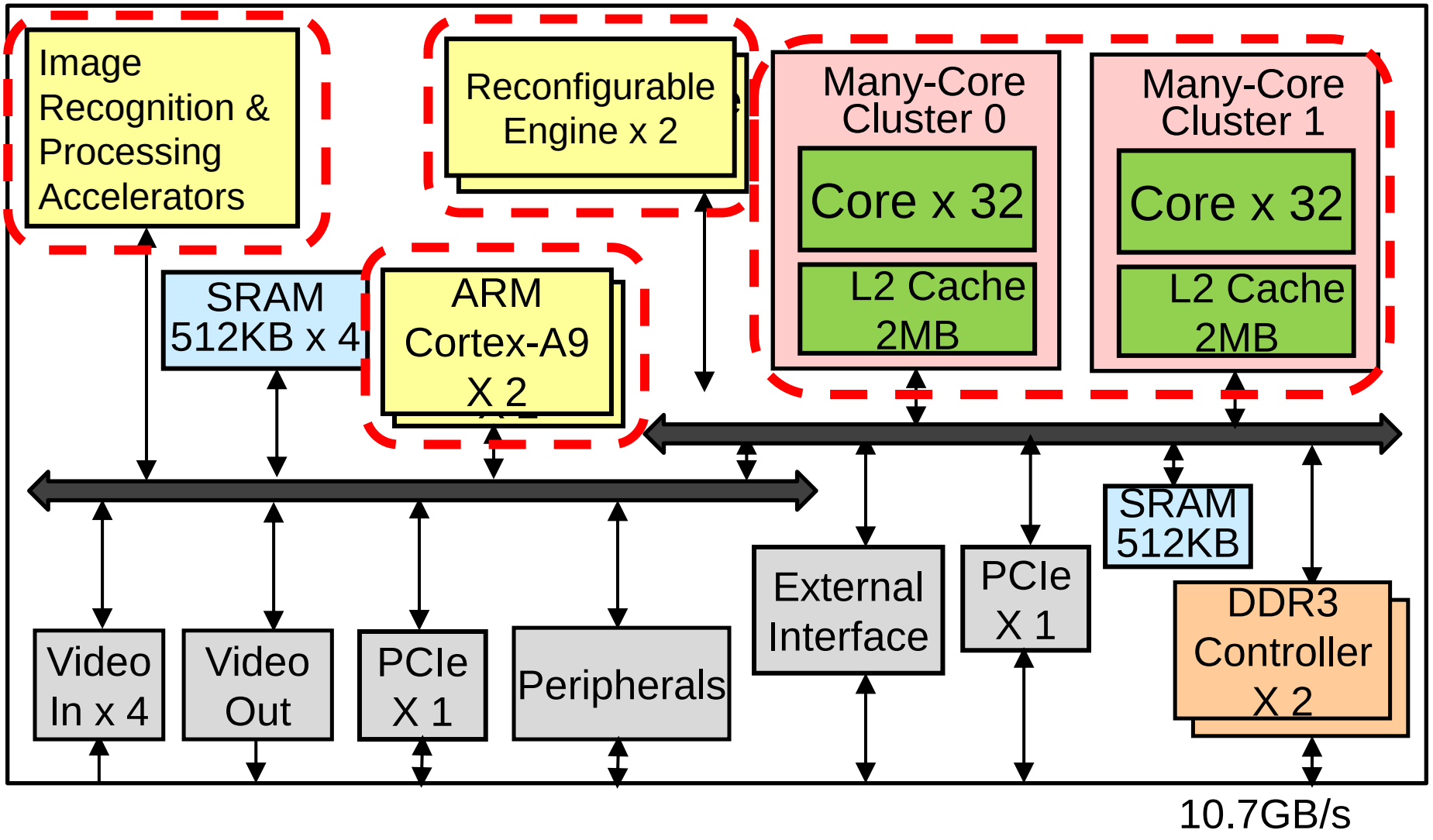
Structure of Many-Core Cluster



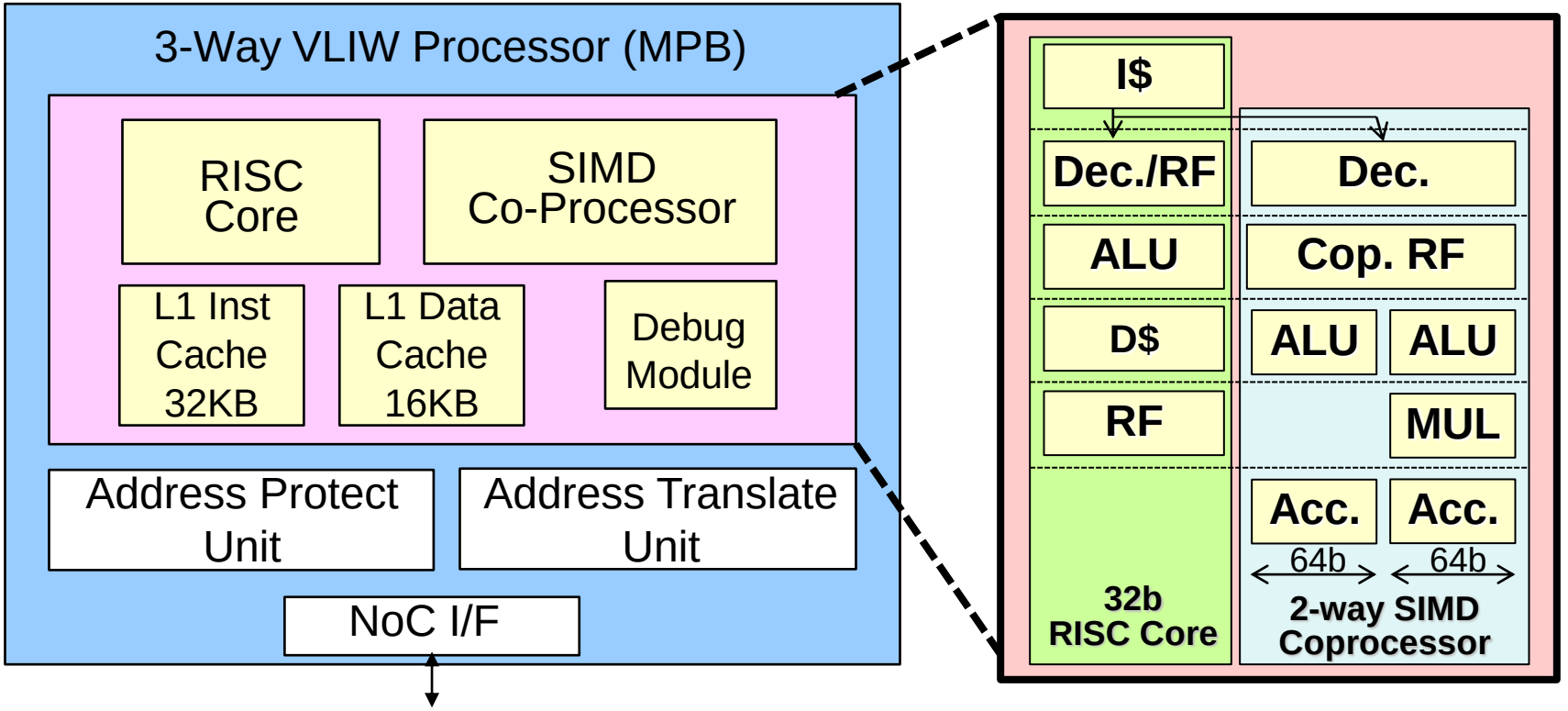
- **Tree-based NoC**
  - Leaf nodes: Core
  - Root nodes: L2 cache banks



Many-Core SoC Architecture



Core : Media Processing Block (MPB)



- **3-Way VLIW Processor**
- **L1 Instruction Cache: 32KB**
- **L1 Data Cache: 16KB**
- **333 MHz**

Exploits multi-grain parallelism

- **Thread level by many cores**
- **Instruction level by VLIW architecture**
- **Data level by SIMD instructions**

Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Issues in implementing parallelized face detection

- **High coarse-grain parallelism: Good for Parallelization**
  - There are enough ROIs to exploit by many cores
- **Imbalanced workload: Bad for Processor Utilization**
  - The workload of an ROI where a face exists is higher than that of an ROI without a face

Implementation of parallelized face-detection

- Minimize the number of threads in order to reduce synchronization cost
  - Allocate one thread to one core
- Find a good thread partitioning with balancing workload of threads
- Reduce data bandwidth (L1$-L2$ and L2$-DDR3)

Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

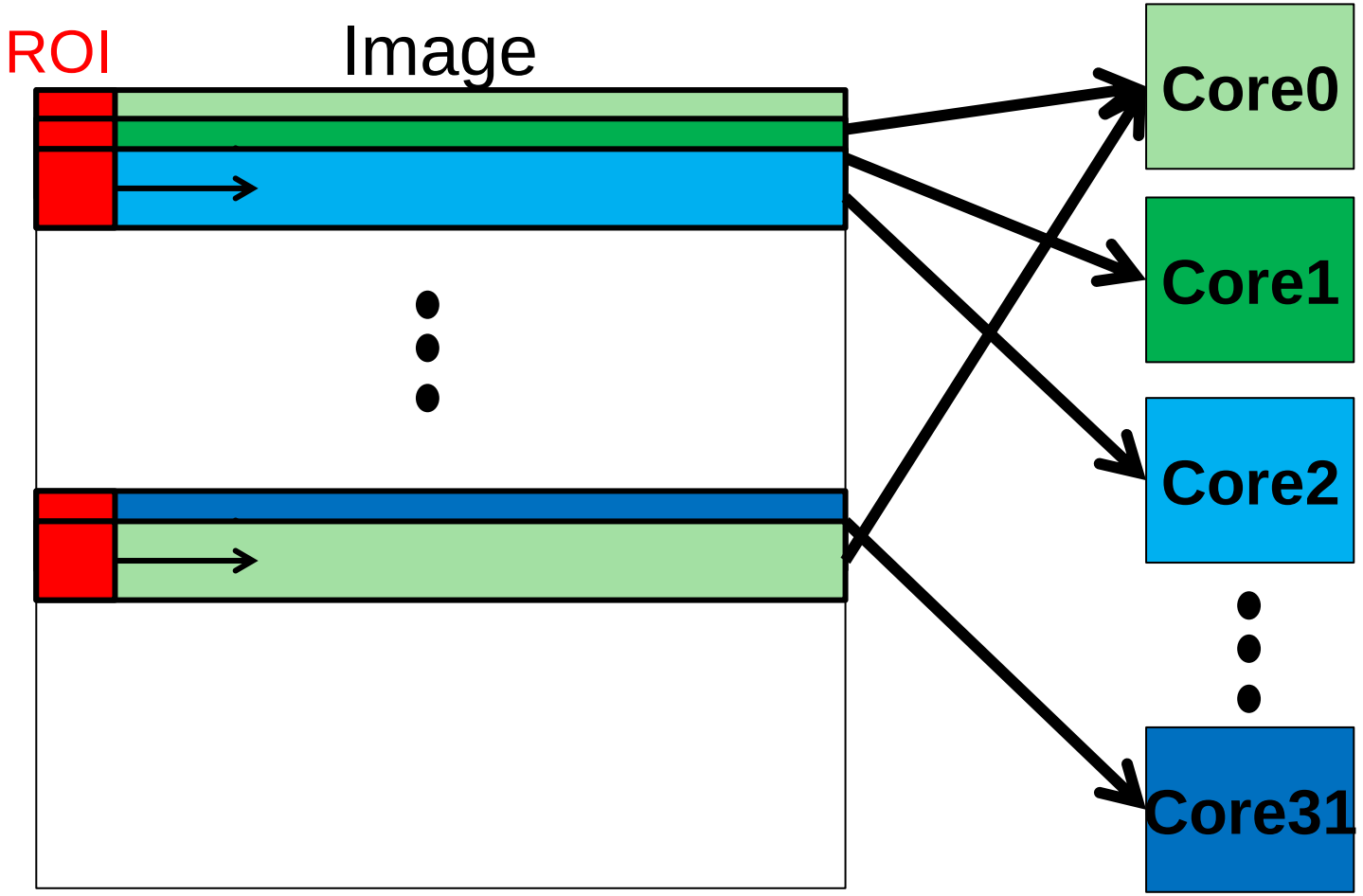
Implementation on the Single Cluster

- **We implemented the face detection with two methods to allocate image to cores**
  - Allocating Cyclically
  - Splitting Equally

(1) Allocating Cyclically

This way allocates lines to each core cyclically

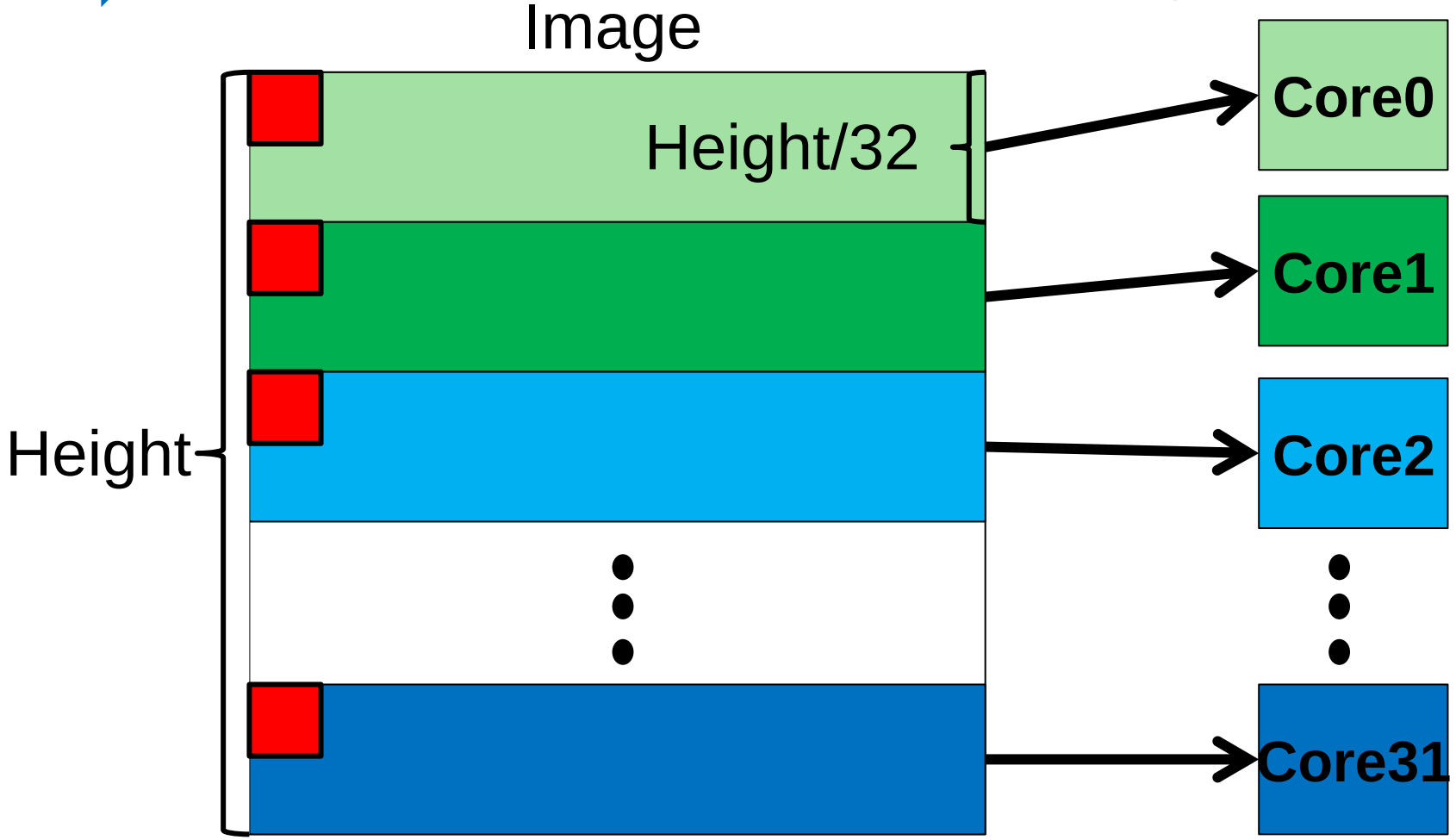
➡ Effective in balancing workload



(2) Splitting Equally

This way divides the image evenly

Effective to reduce data size read by each core

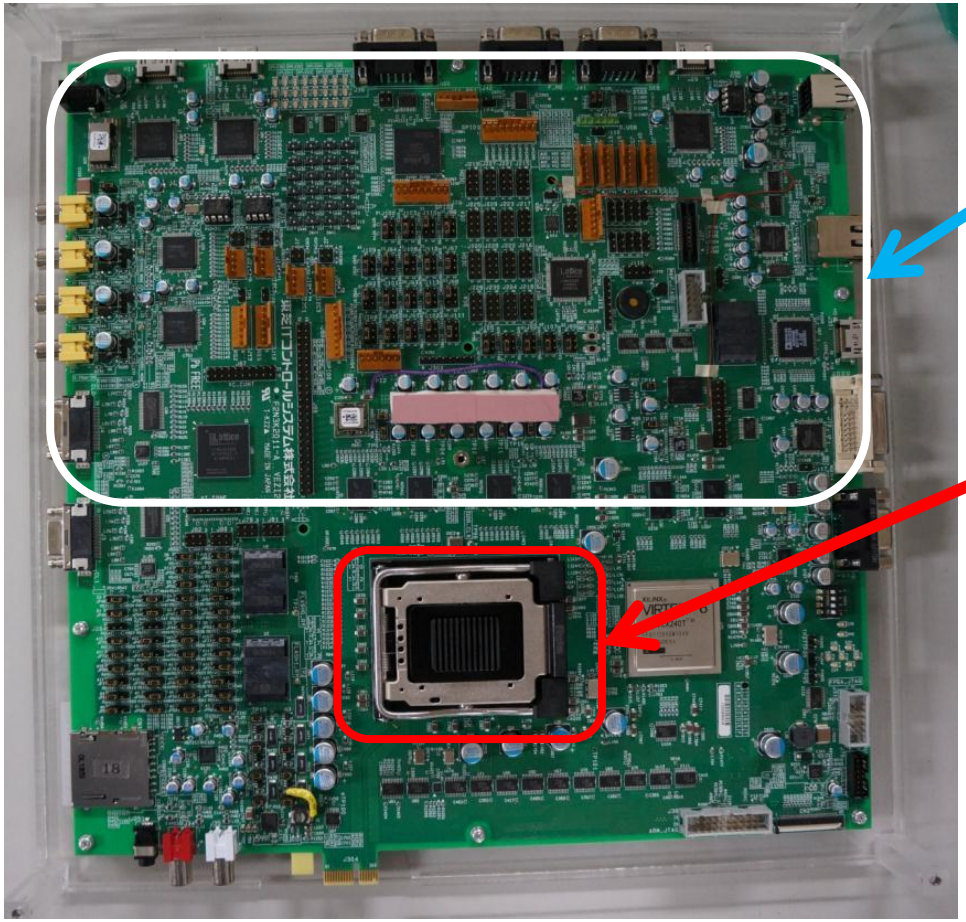


Images for Evaluation

- **High Resolution Images (5.76-12.7Mp) including many faces**

No.	Resolution	Number of Faces
0	4000x1440	30
1	3000x4082	37
2	4083x3062	78
3	4094x3107	148
4	3568x2568	9
5	3568x2568	10

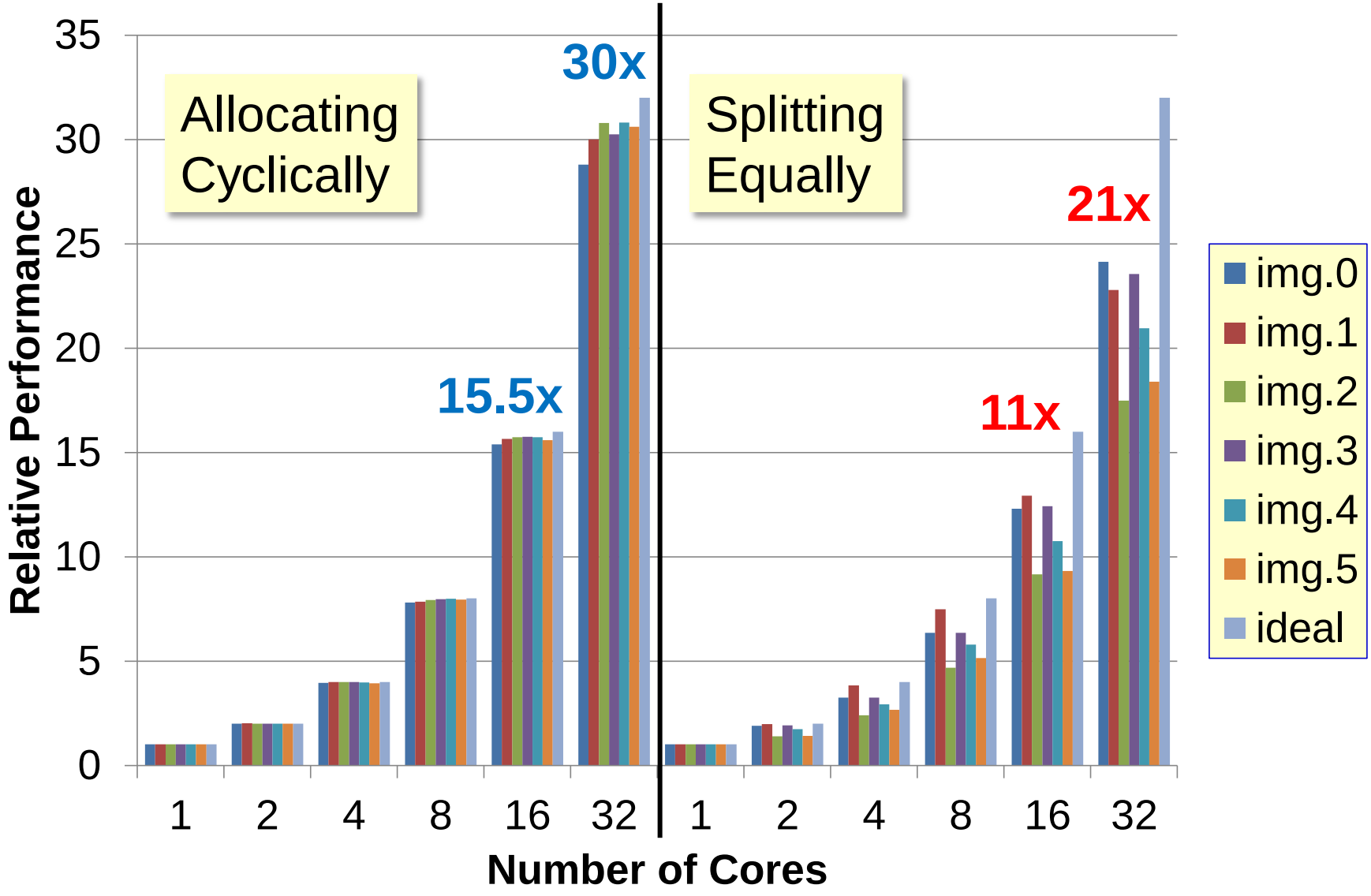
Evaluation Board



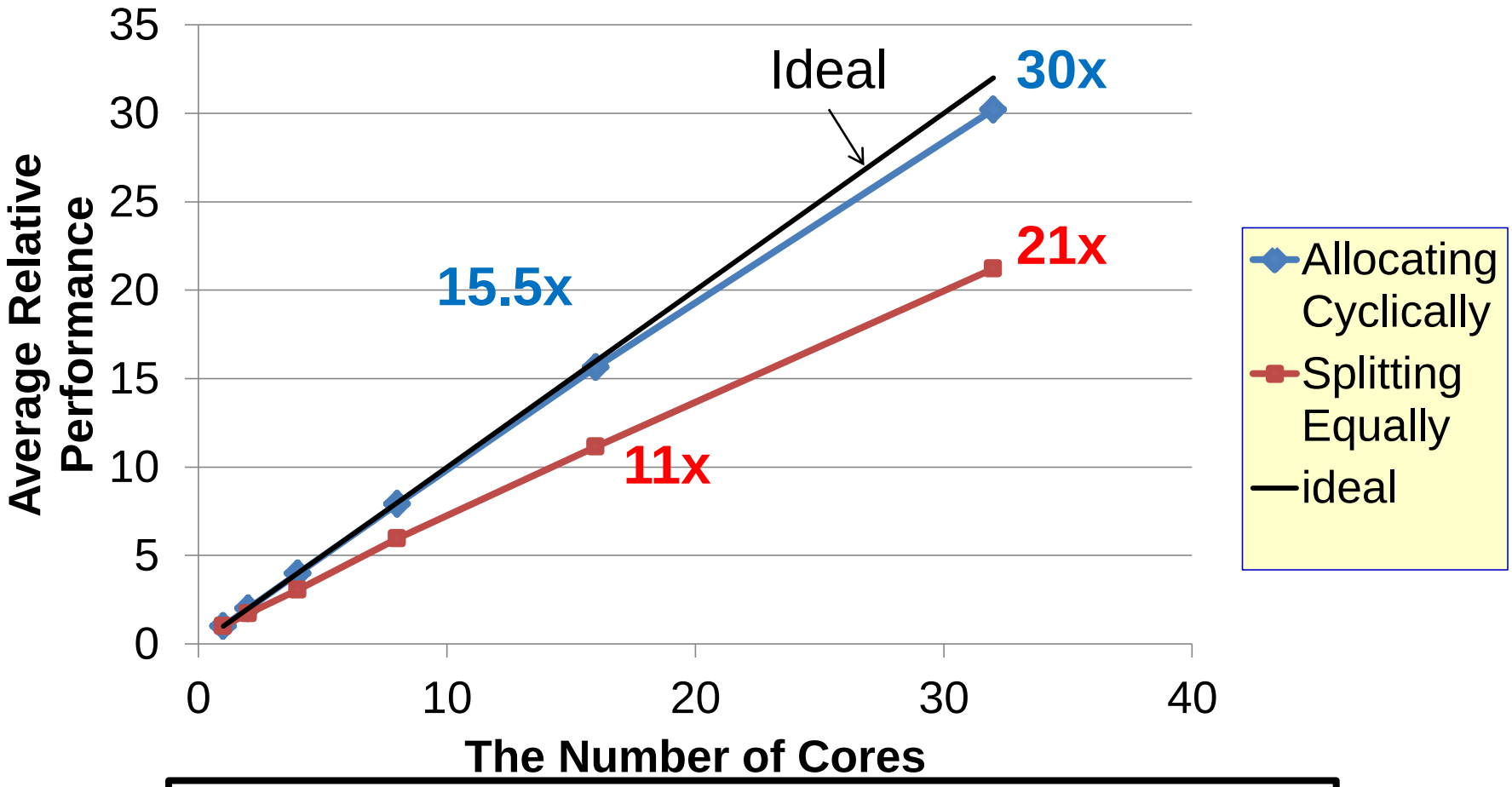
I/O and switches for evaluation

Many-Core SoC (Fan-less Cooling)

Relative Performance on Single Cluster

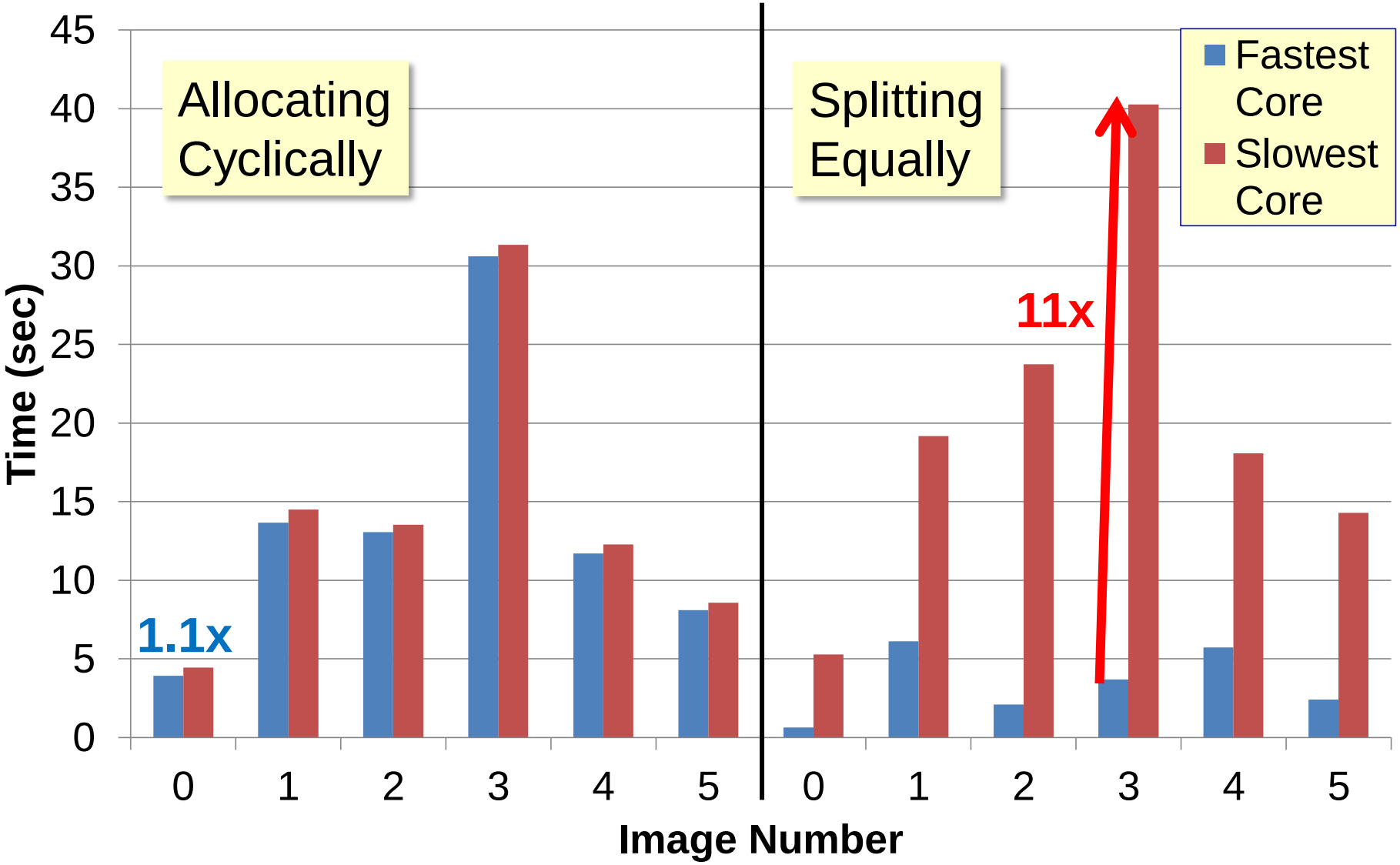


Average Relative Performance on Single Cluster



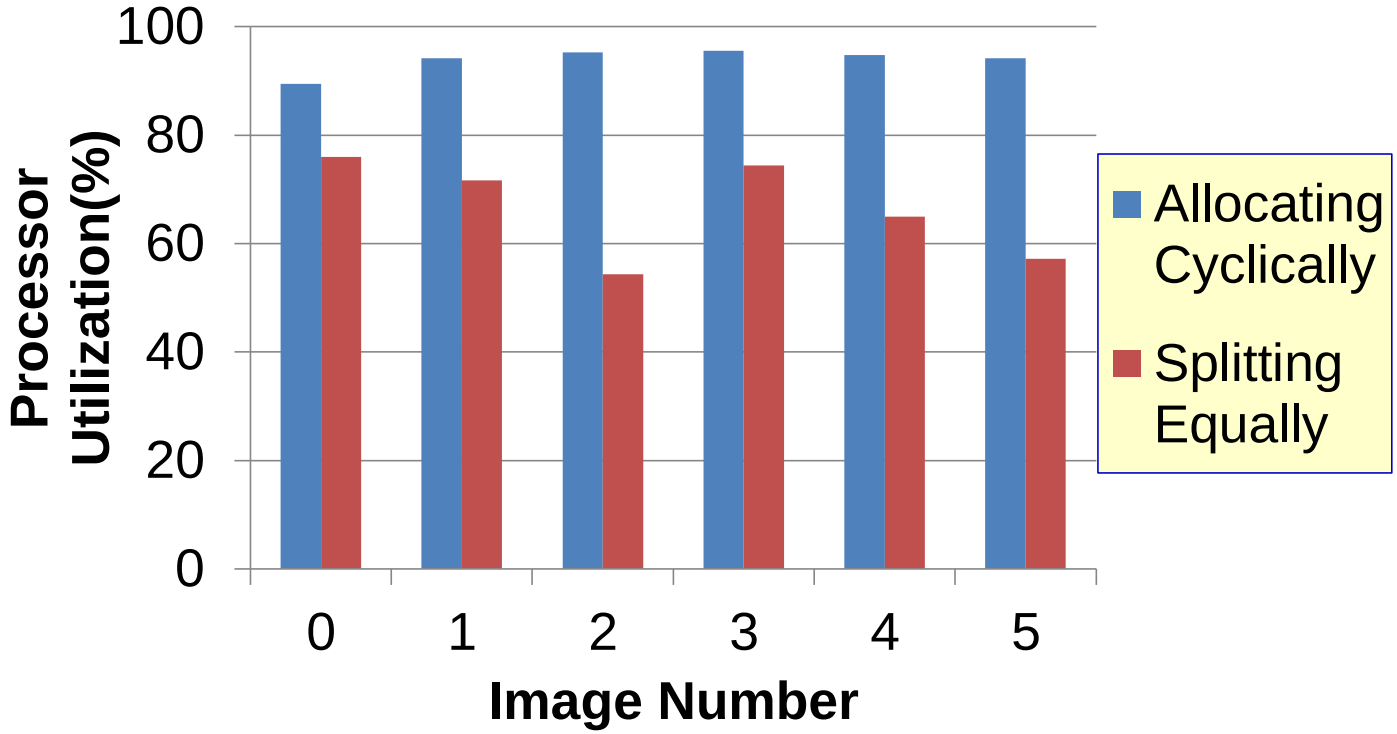
With Allocating Cyclically, performance scales up to 32 cores

Execution Time of the Fastest and Slowest Cores



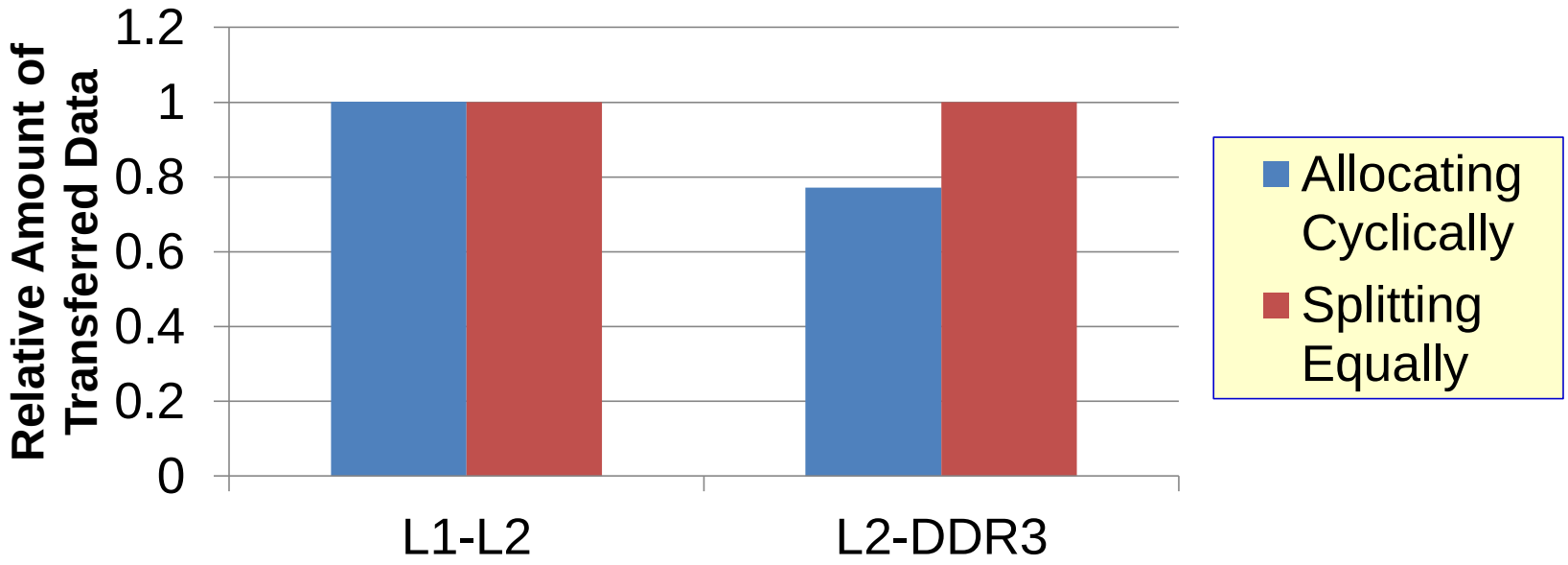
Processor Utilization

- **Allocating Cyclically : 90 ~ 95%**
- **Splitting Equally : 55 ~ 75%**



Low processor utilization deteriorates the performance of Splitting Equally

Bandwidth of L2 Cache and DDR3



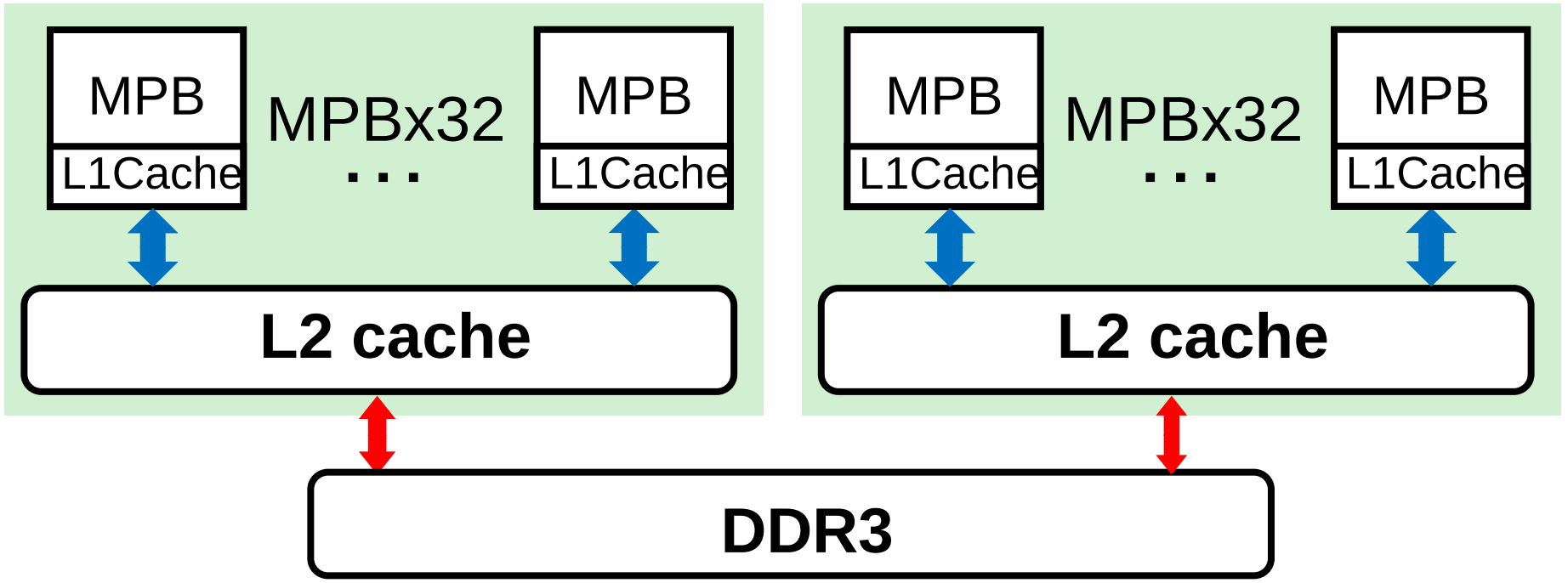
- **L1-L2 bandwidth is nearly the same**
  - L1 cache is not enough to store ROI line
- **About L2-DDR3, Allocating Cyclically is better**
  - All cores access the small area at the same

Outline

- **Introduction**
- **Face Detection using Joint Haar-Like Features**
- **Architecture of Energy Efficient Many-Core SoC**
- **Issues in Implementing Parallelized Face Detection**
- **Implementation and Evaluation of Parallelized Face Detection**
  - On the Single Cluster
  - On the Dual Cluster
- **Conclusion**

Implementation on Dual Cluster

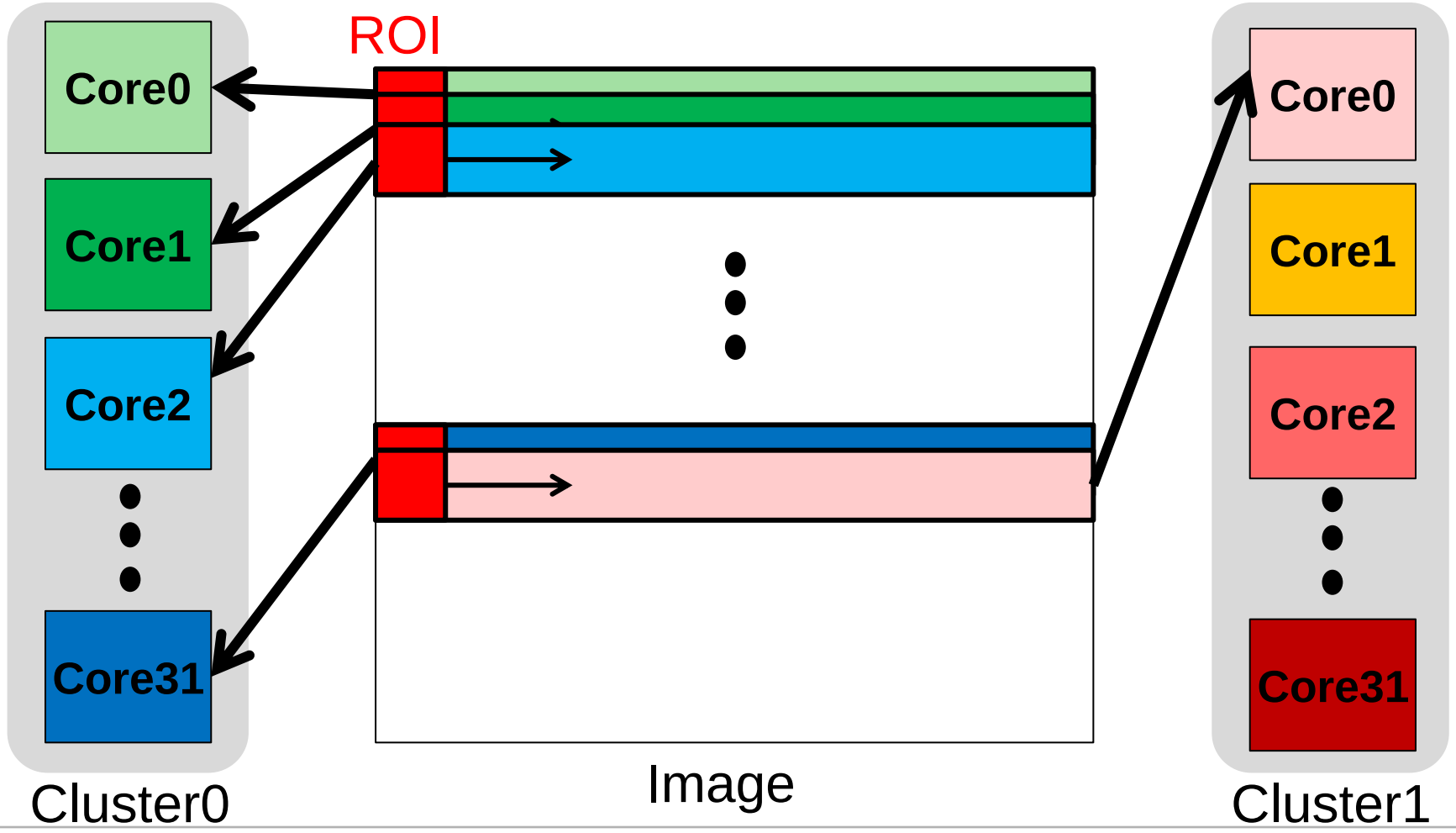
- **Each cluster has its own L2 cache and shares DDR3**
  - Because bandwidth is narrower than L1 and L2 cache, reducing bandwidth between L2 cache and DDR3 is important
- **We implemented the two ways**
  - Allocating Cyclically
  - Bisection



(1) Allocating Cyclically

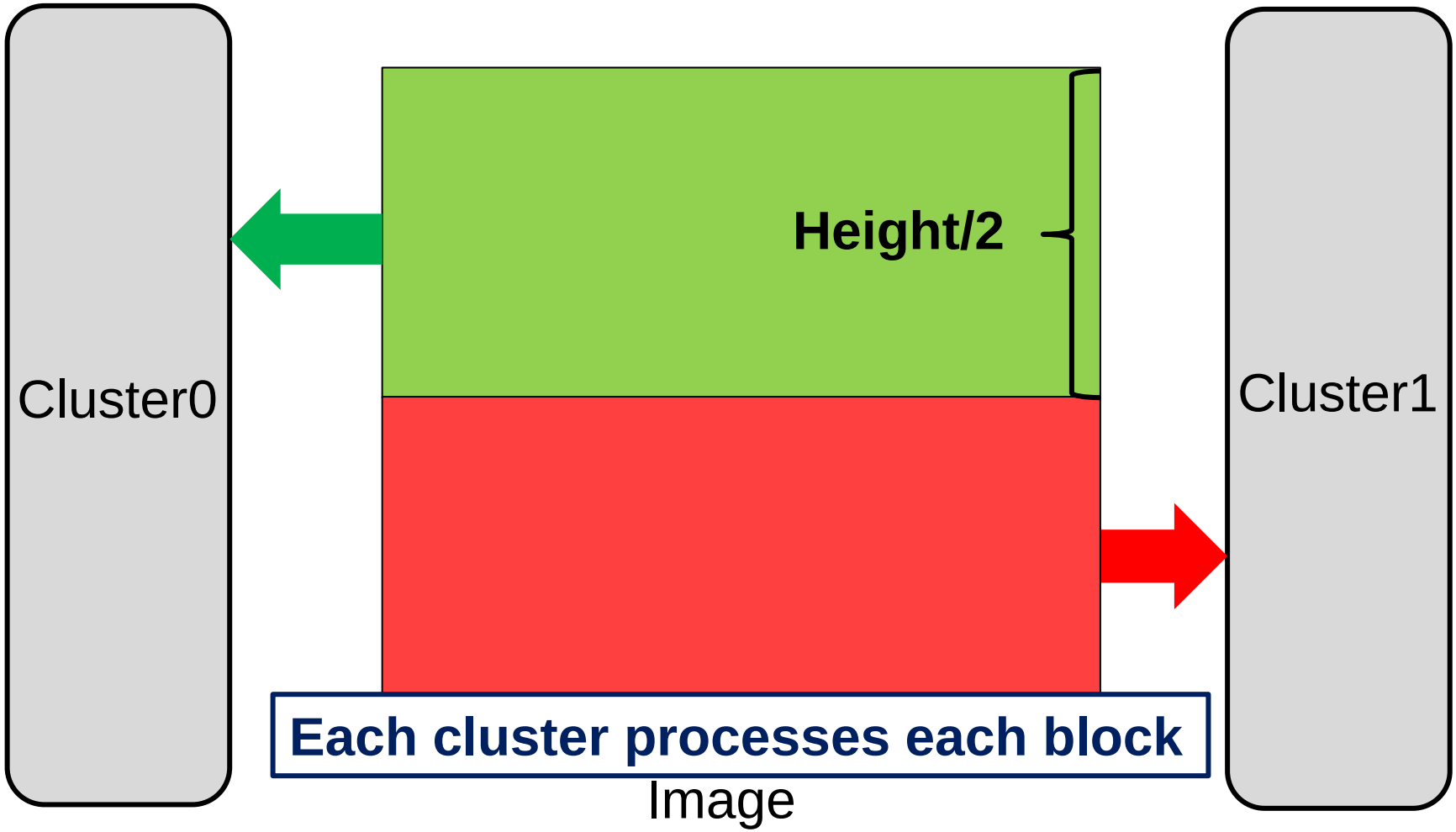
This way is the same as that of a single cluster

➡ Effective in balancing workload



(2) Bisection

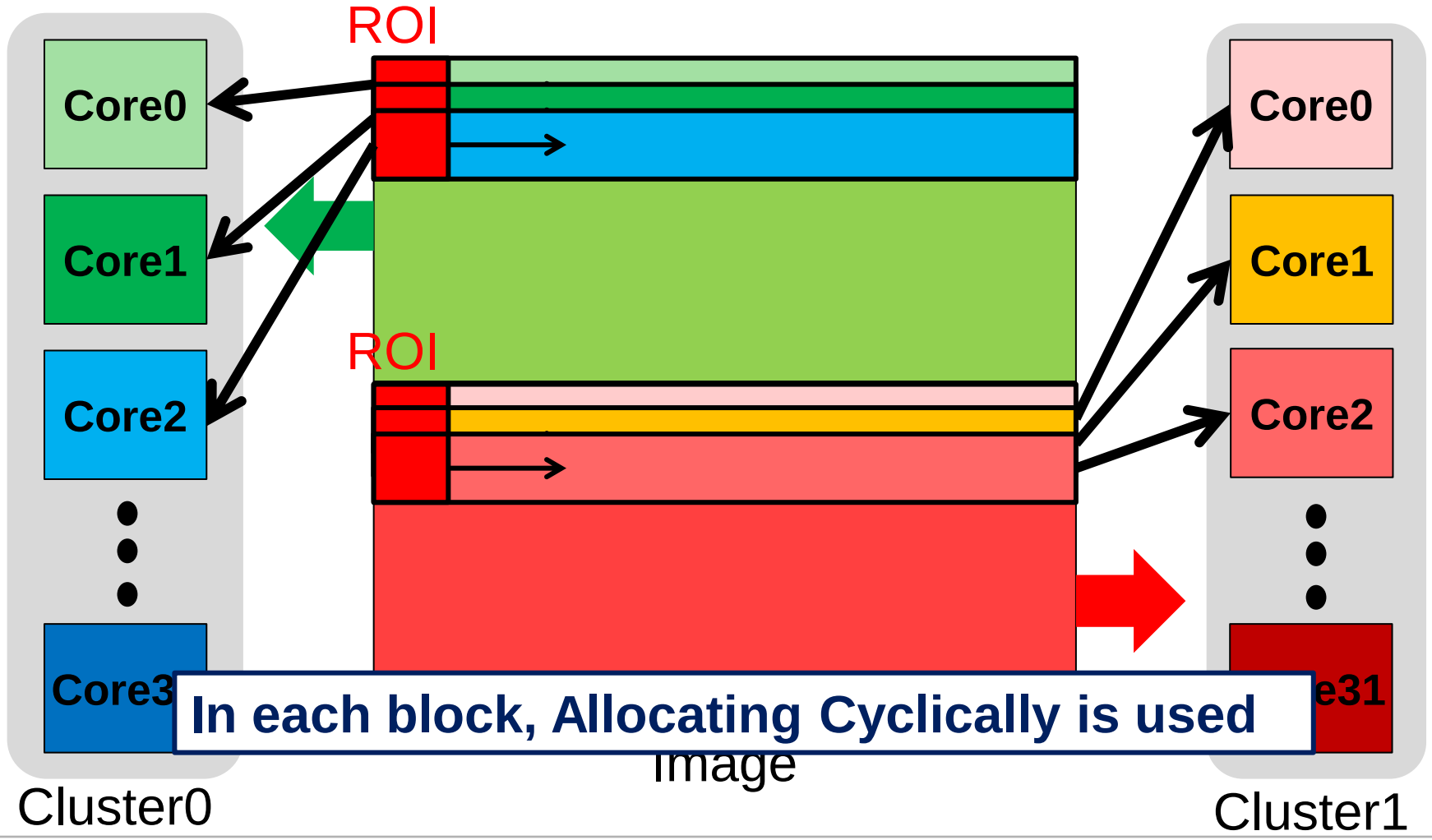
This way divides the image into two blocks



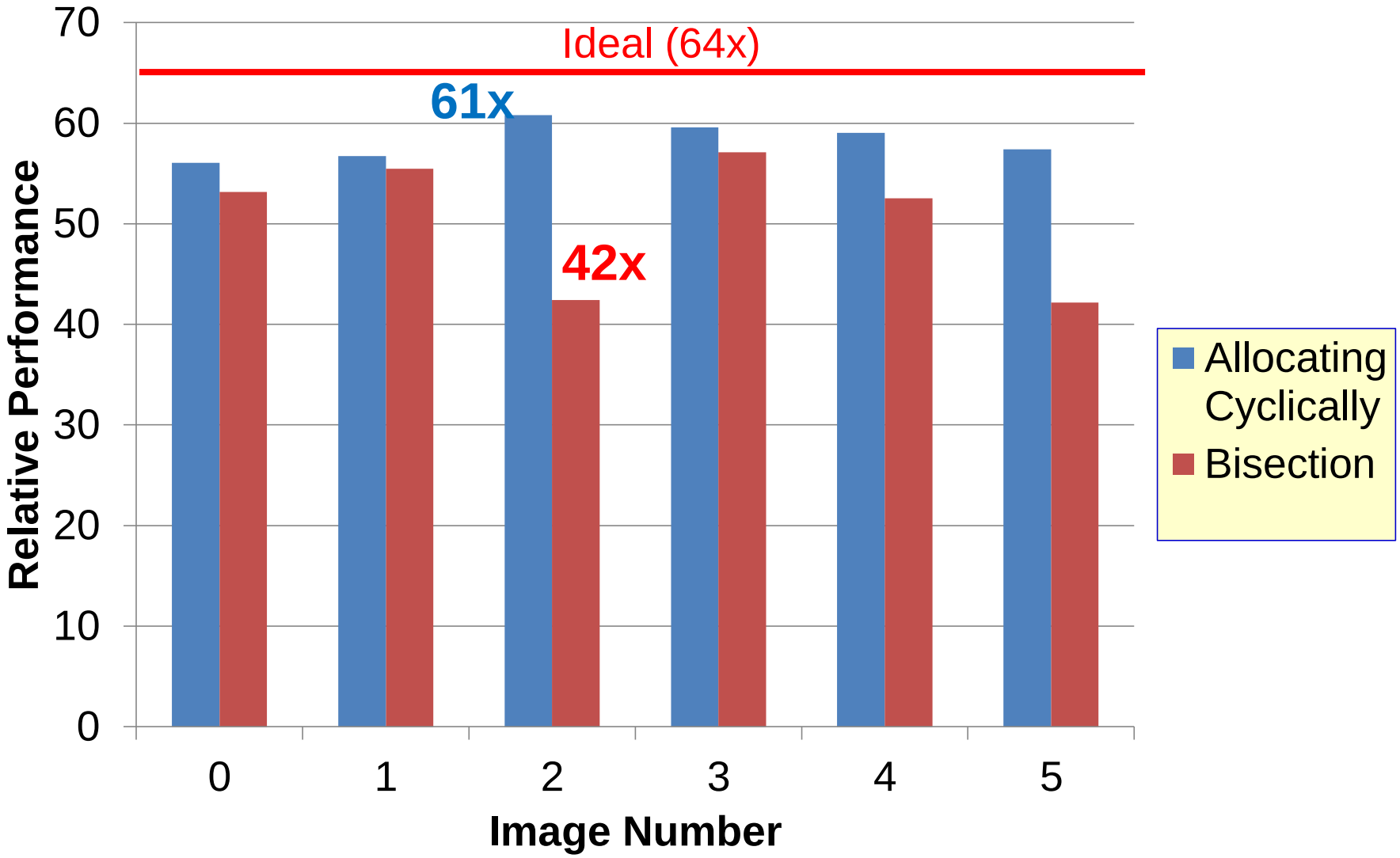
(2) Bisection

This way divides the image into two blocks

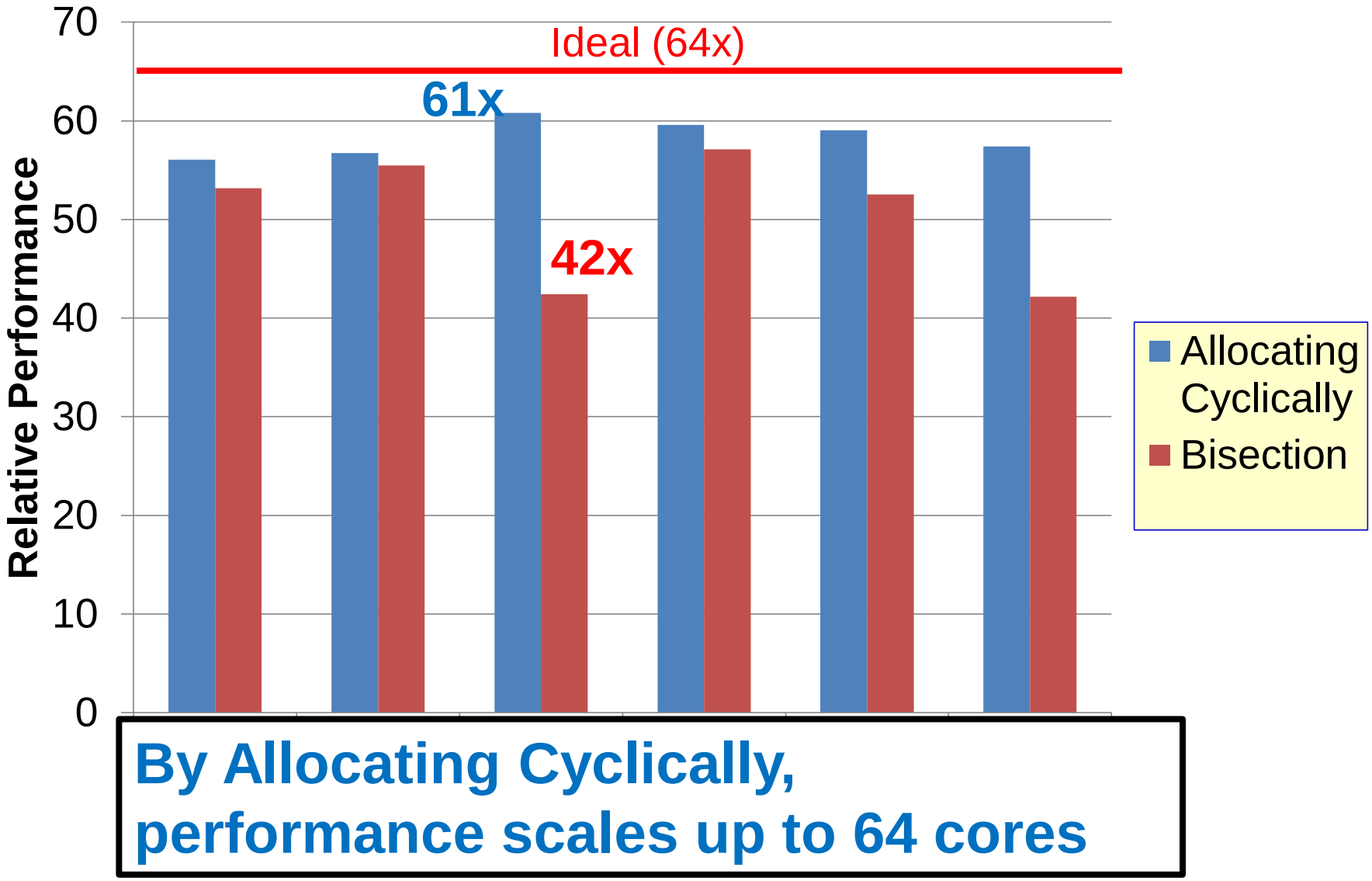
➡ Effective to reduce data size read by each cluster



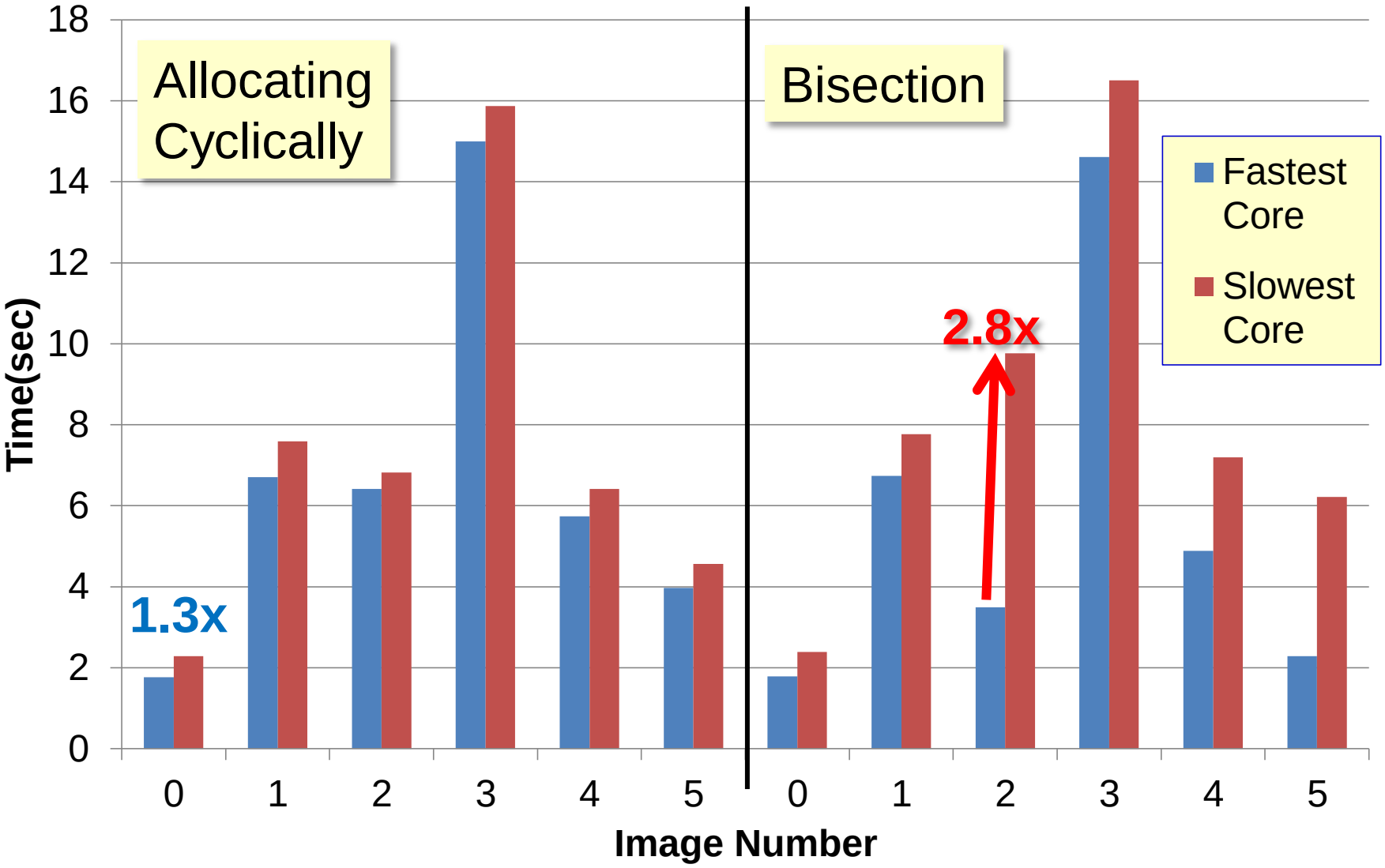
Performance of Dual Cluster (64 Cores)



Performance of Dual Cluster (64 Cores)

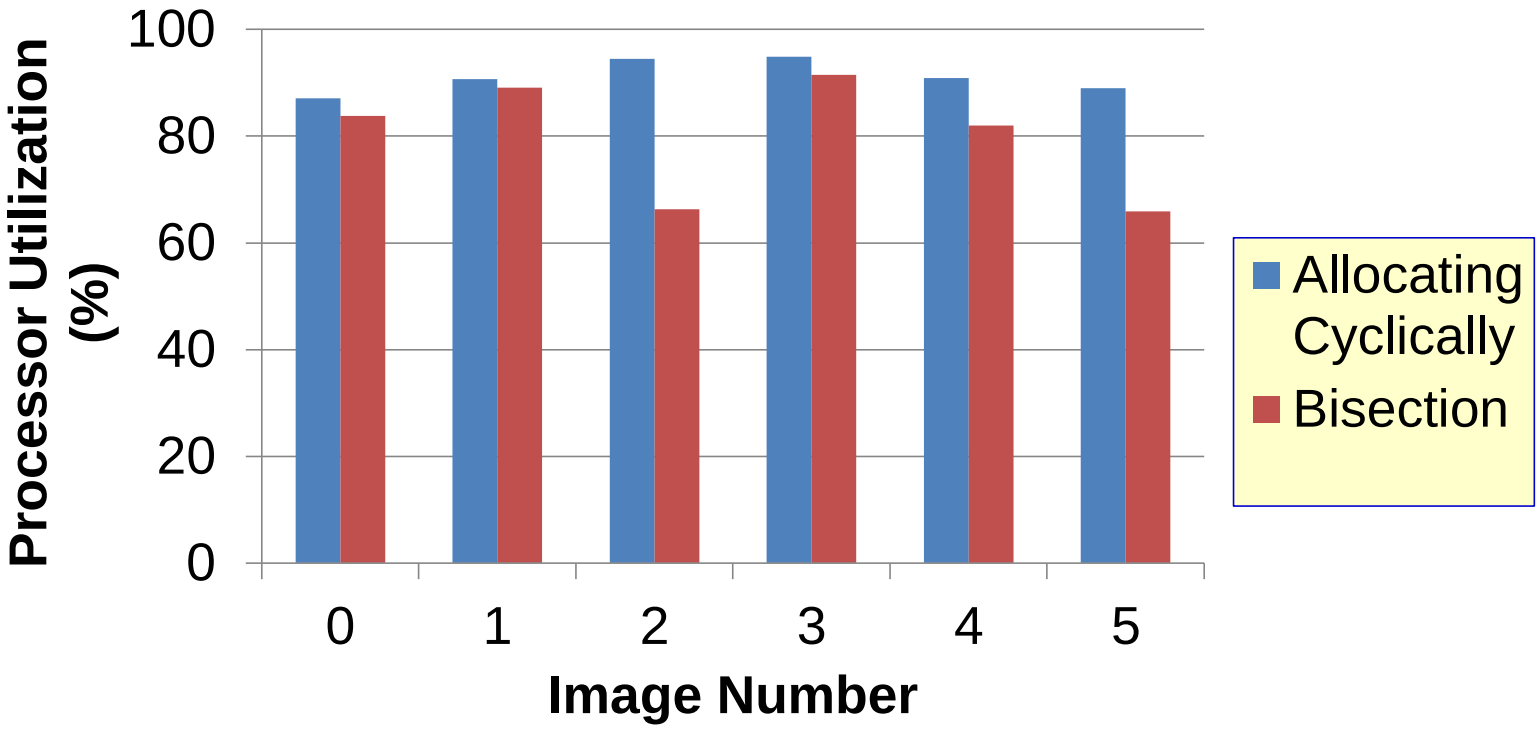


Execution Time of the Fastest and Slowest Cores



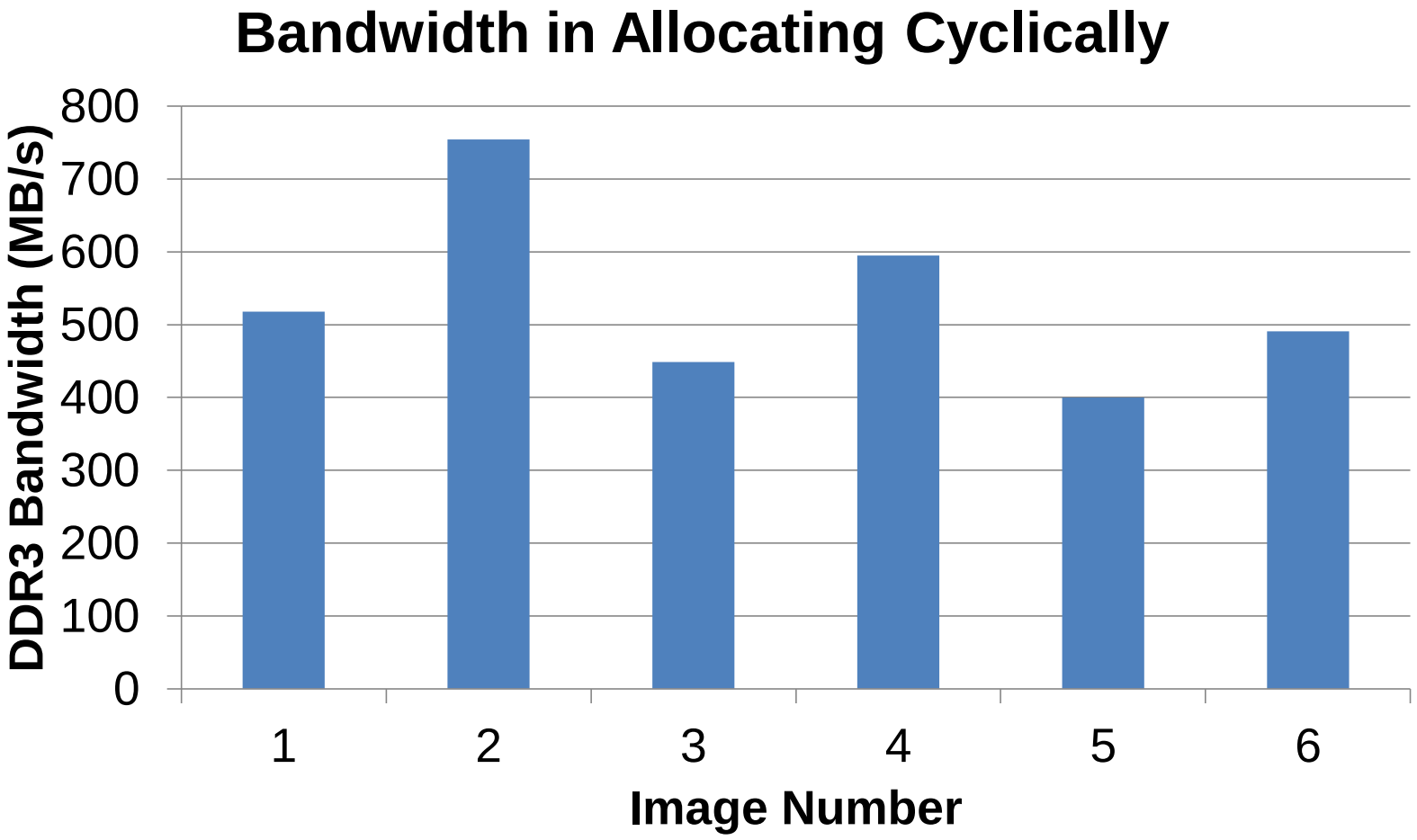
Processor Utilization

- **Allocating Cyclically : 87 ~ 95%**
- **Bisection : 66 ~ 91%**



Low processor utilization deteriorates the performance of Bisection

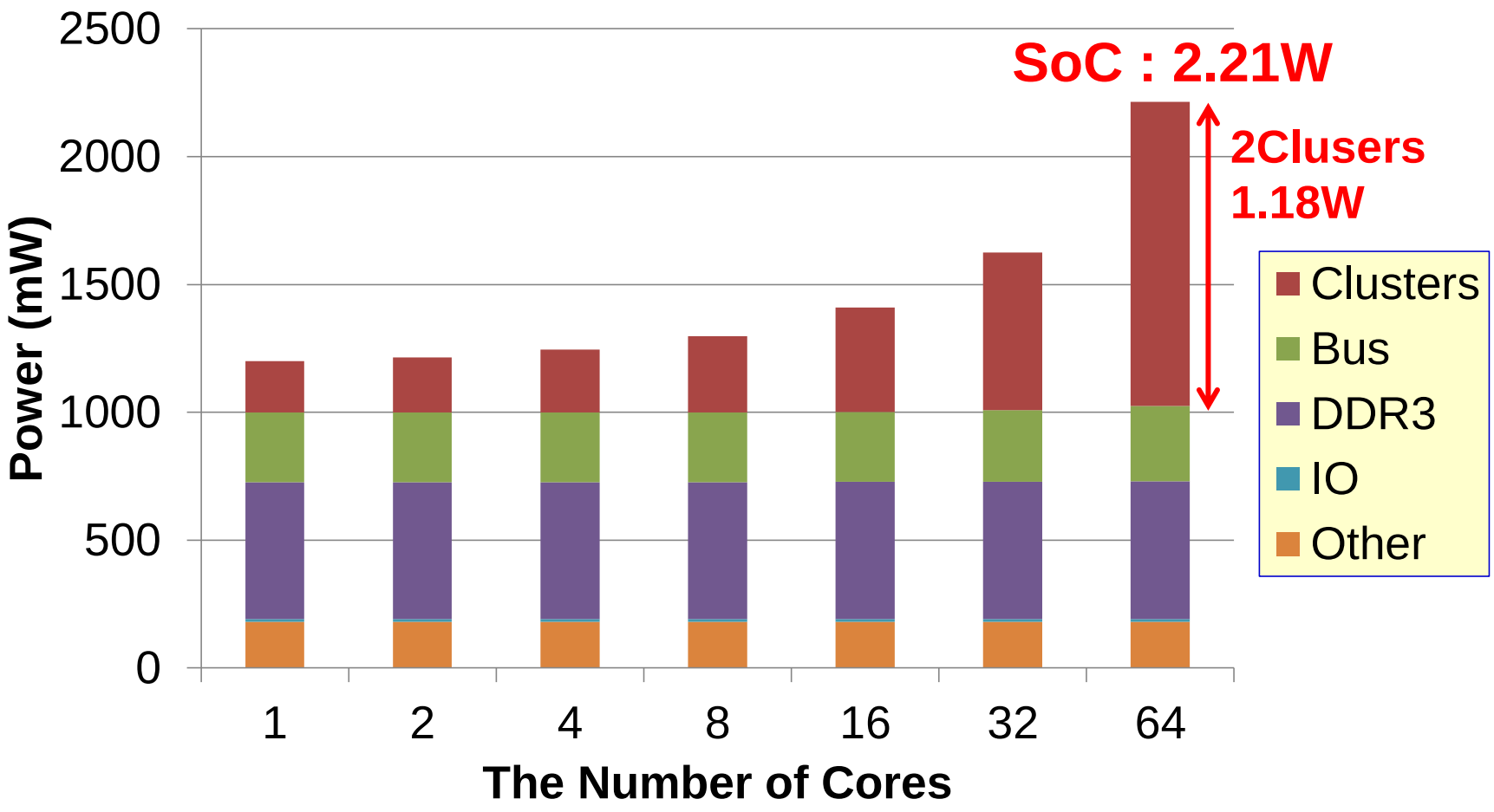
DDR3 Bandwidth



Utilized bandwidth is 750MB/s (only 7% of maximum (10.7GB/s))
Memory bandwidth is not bottleneck even when two clusters operate.

Power Consumption

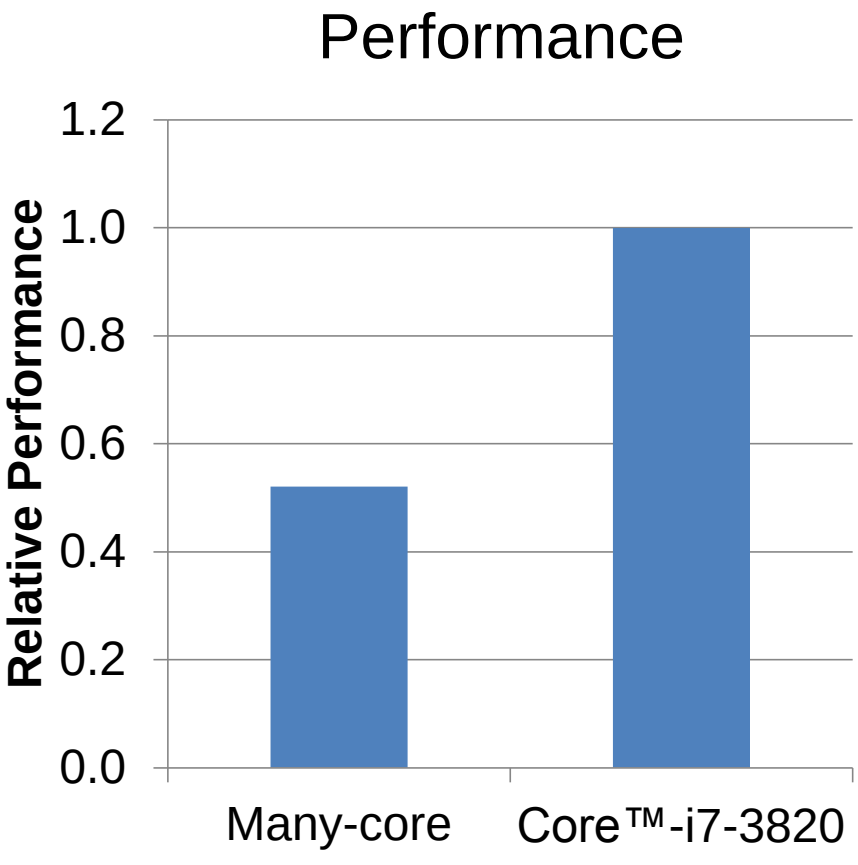
Our many-core SoC achieves less than 3W

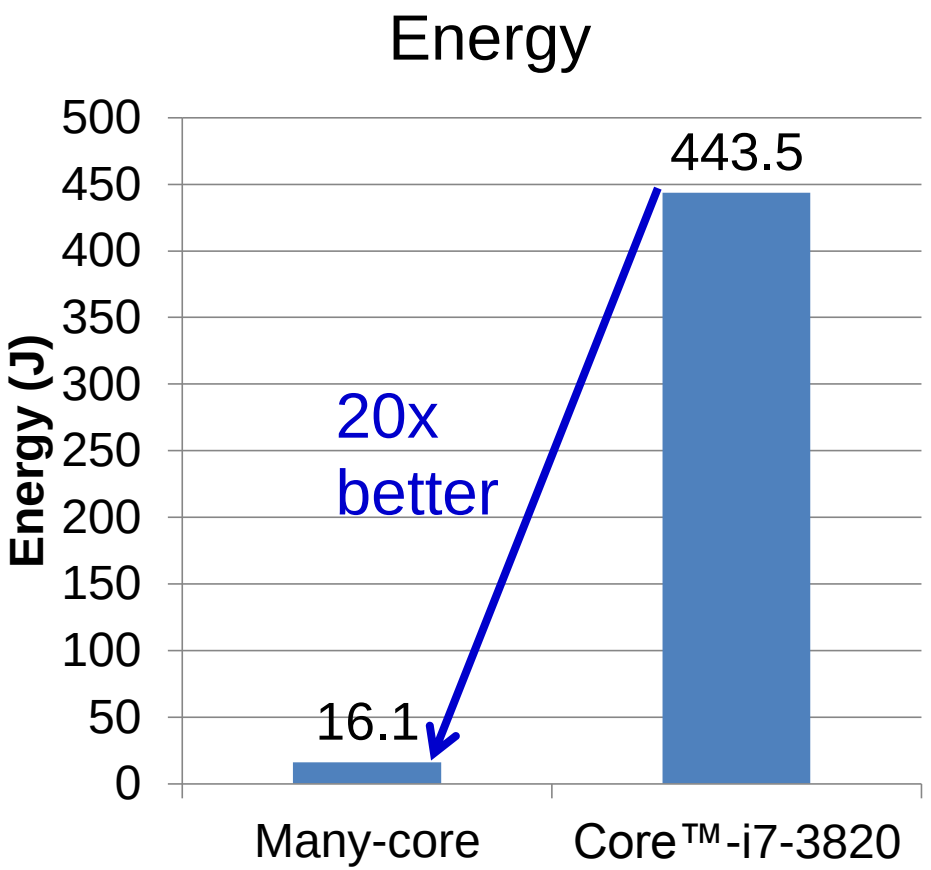


Typical Process, Room Temperature, using Allocating Cyclically

Comparison with Desk-Top CPU

- **Compared with Desk-Top CPU**
  (Core™-i7-3820: 3.6GHz, 4 Cores, 8 Threads)





TDP of Core™-i7-3820 (130W) is used for calculating energy

Conclusion

- **Future architecture will have many cores**
  - A key challenge : How to efficiently use them?
- **We evaluated the many-core SoC with parallelized face detection**
  - Many-core is suited for the face detection because it exploits ROI based coarse-grained parallelism efficiently
    - **Scale up by 30x (32 cores) to 60x (64 cores)**
    - **Balancing workload is important**
- **Power consumption is only 2.21W under actual workload : enables fan-less cooling**
  - Our many-core SoC is remarkably energy efficient in image recognition applications
    - **20x better than the desk-top CPU**

Thank you!