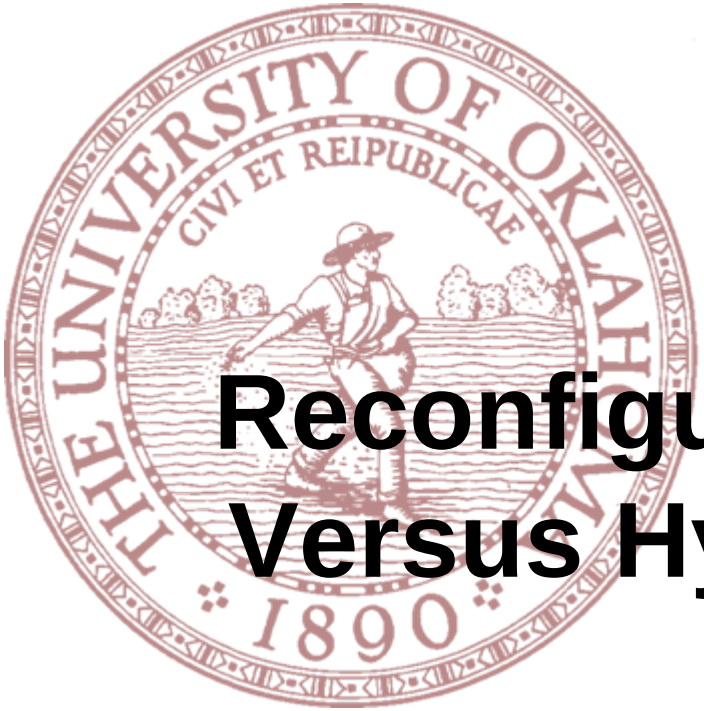


Reconfigurable Versus Fixed Versus Hybrid Architectures

John K. Antonio

Oklahoma Supercomputing Symposium 2008

Norman, Oklahoma
October 6, 2008

Computer Science, University of Oklahoma

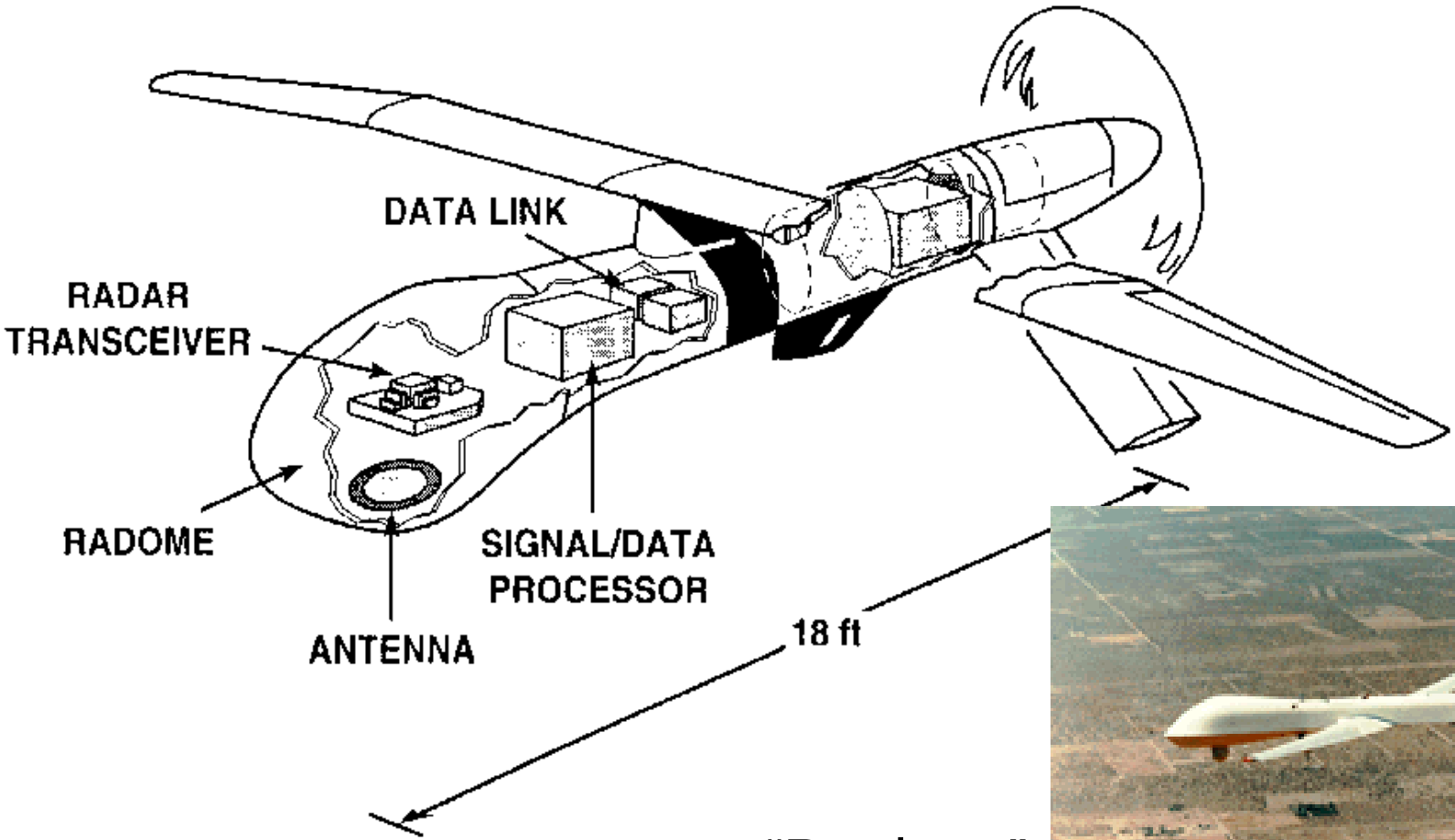
Overview

- The (past) world of reconfigurable computing
- The (past) world of multi-core
- The (emerging) world of reconfigurable multi-core architectures
- Illustrative analysis
- Conclusions

Drivers for reconfigurable computing

- Near performance of custom ASIC
- Near cost of commodity processor
- More flexible than custom ASIC
- “Programming” tools improving steadily
- Often used in embedded applications having high computational throughput requirements and strict SWAP constraints

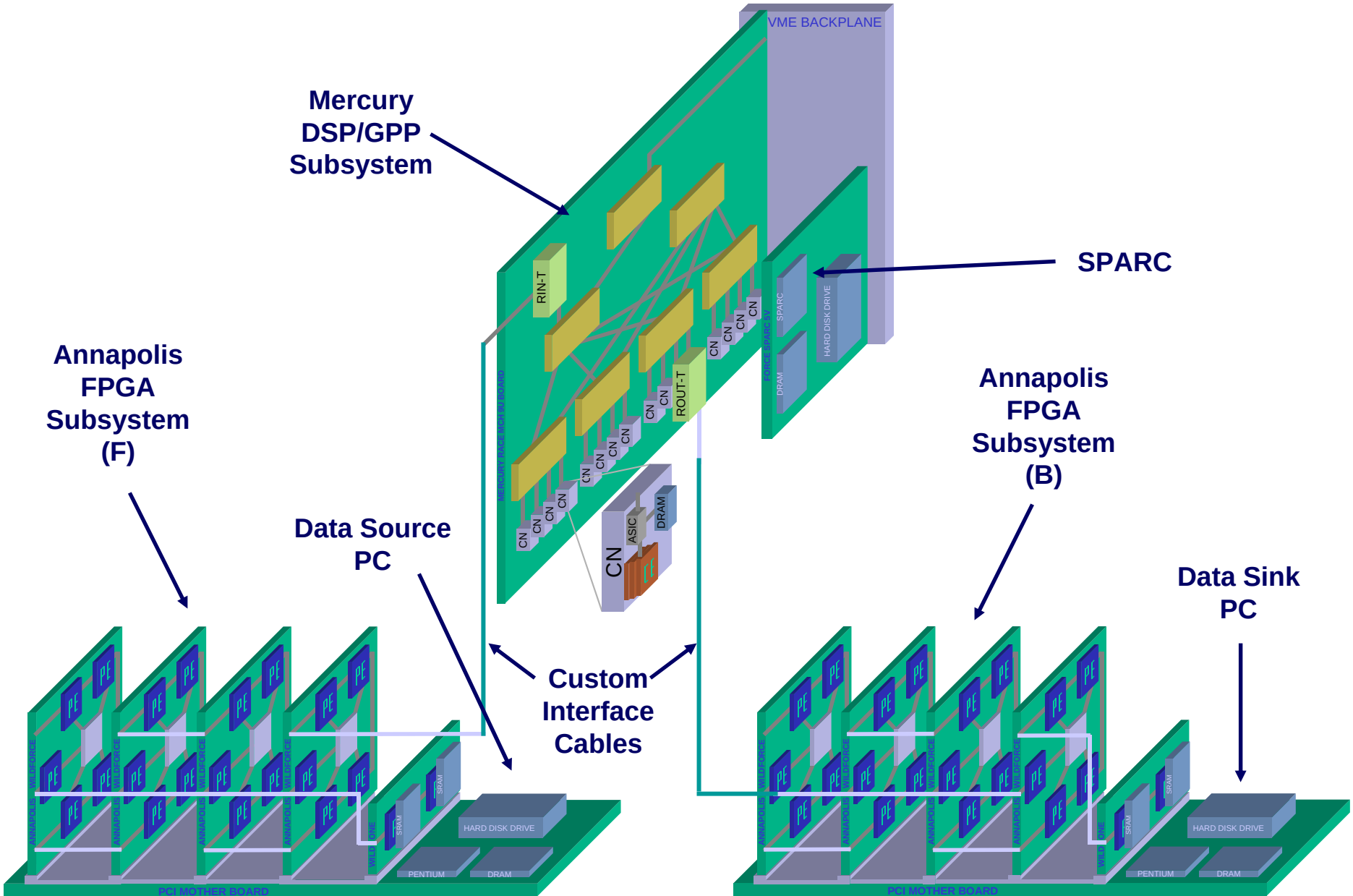
SAR processing on a UAV



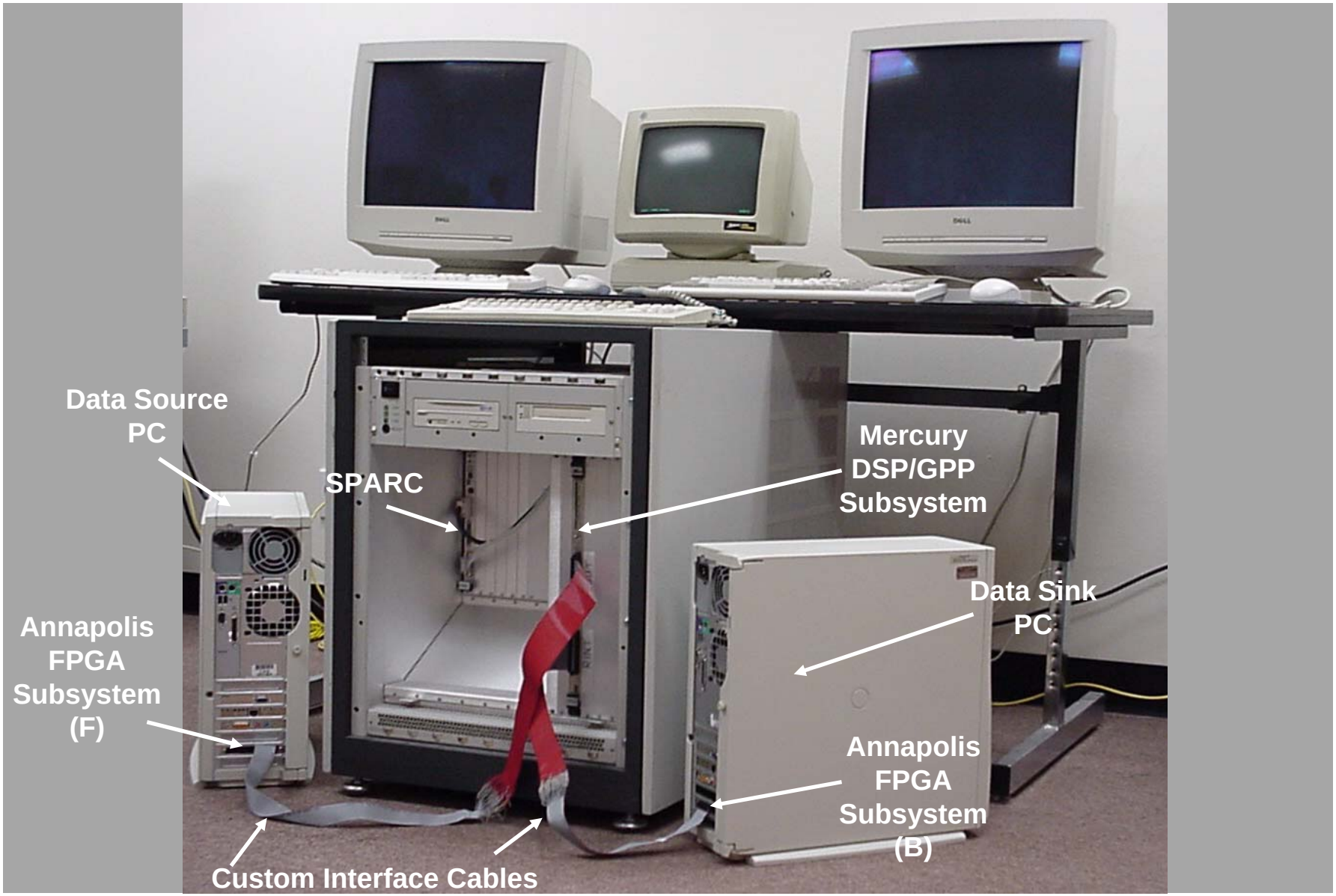
“Predator”

Jeffrey T. Muehring, “Optimal Configuration of a Parallel Embedded System for Synthetic Aperture Radar Processing,” MS Thesis, Texas Tech University, Dec. 1997.

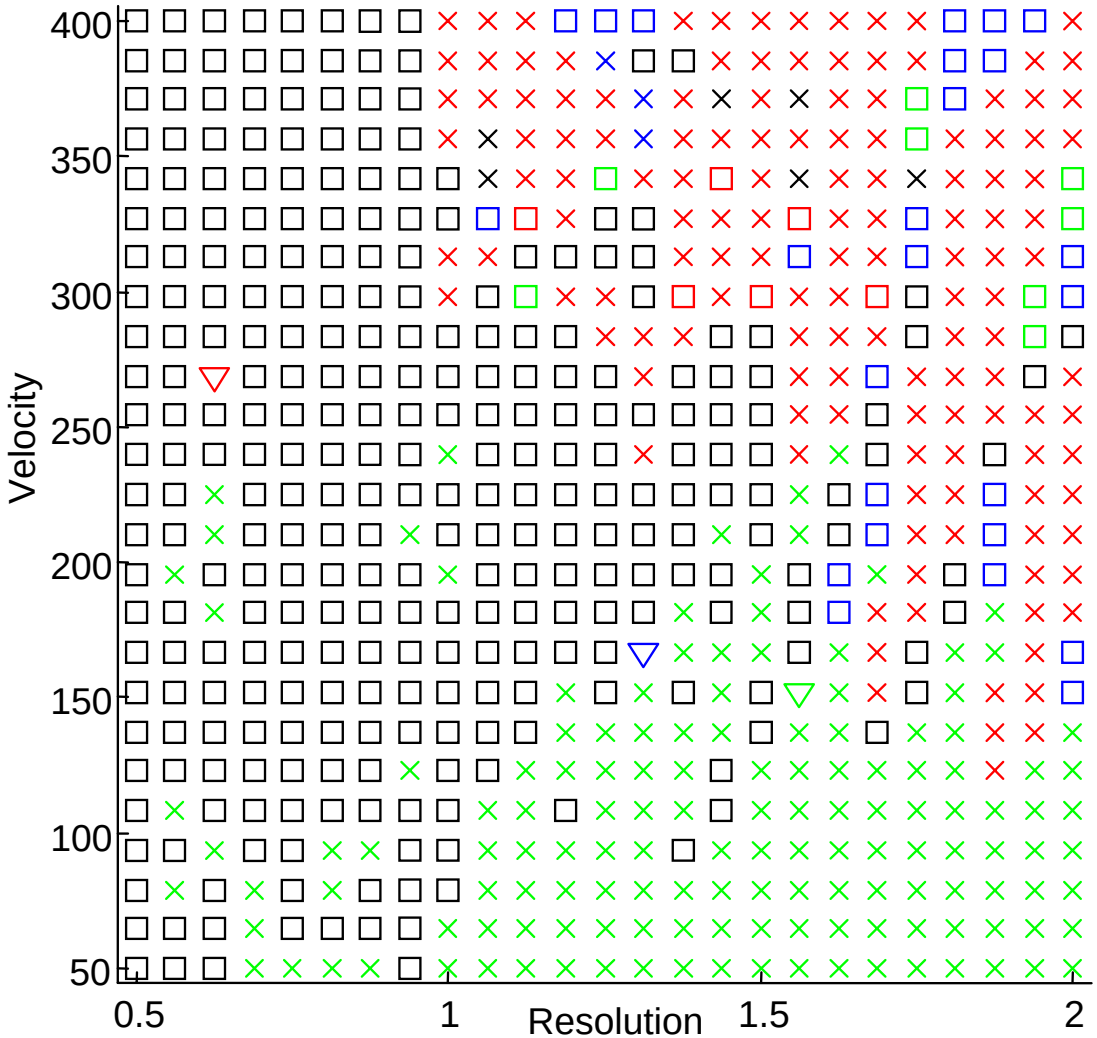
A prototype hybrid system



A prototype hybrid system



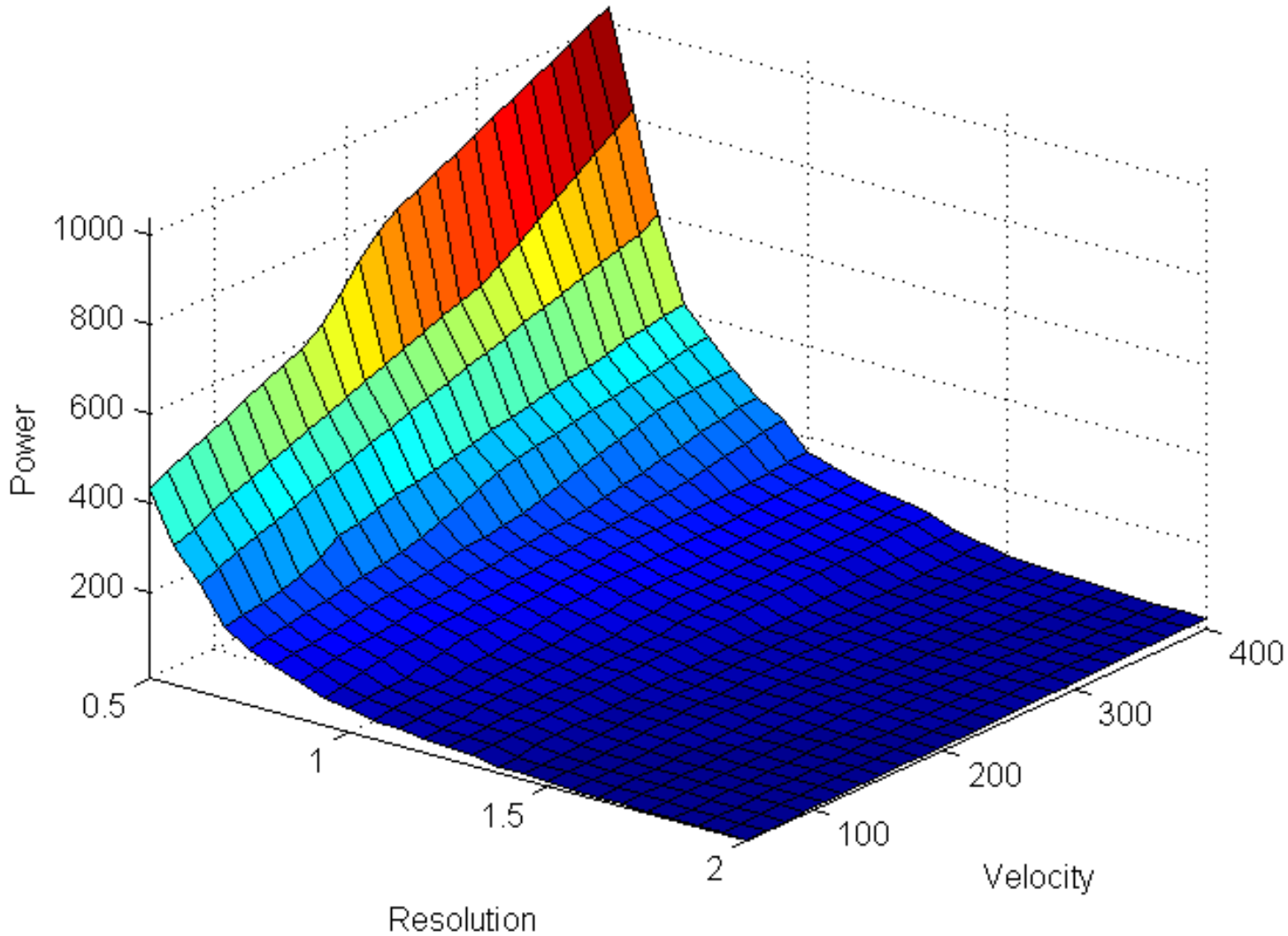
Minimum Power Configurations



	X_T	X_r	X_a	Y_T	Y_r	Y_a
×	1	1	2			
×	2	1	1			
×	1	1	2	1	2	1
×	1	1	2	2	0	1
□	1	1	2	2	0	2
□	1	1	2	2	1	1
□	1	2	1	2	0	2
□	1	3	0	2	0	2
▽	1	3	0	2	1	1
▽	2	0	2	2	1	1
▽	2	1	1	2	2	0

Jeffrey T. Muehring, “Optimal Configuration of a Parallel Embedded System for Synthetic Aperture Radar Processing,” MS Thesis, Texas Tech University, Dec. 1997.

Minimum Power



Jeffrey T. Muehring, “Optimal Configuration of a Parallel Embedded System for Synthetic Aperture Radar Processing,” MS Thesis, Texas Tech University, Dec. 1997.

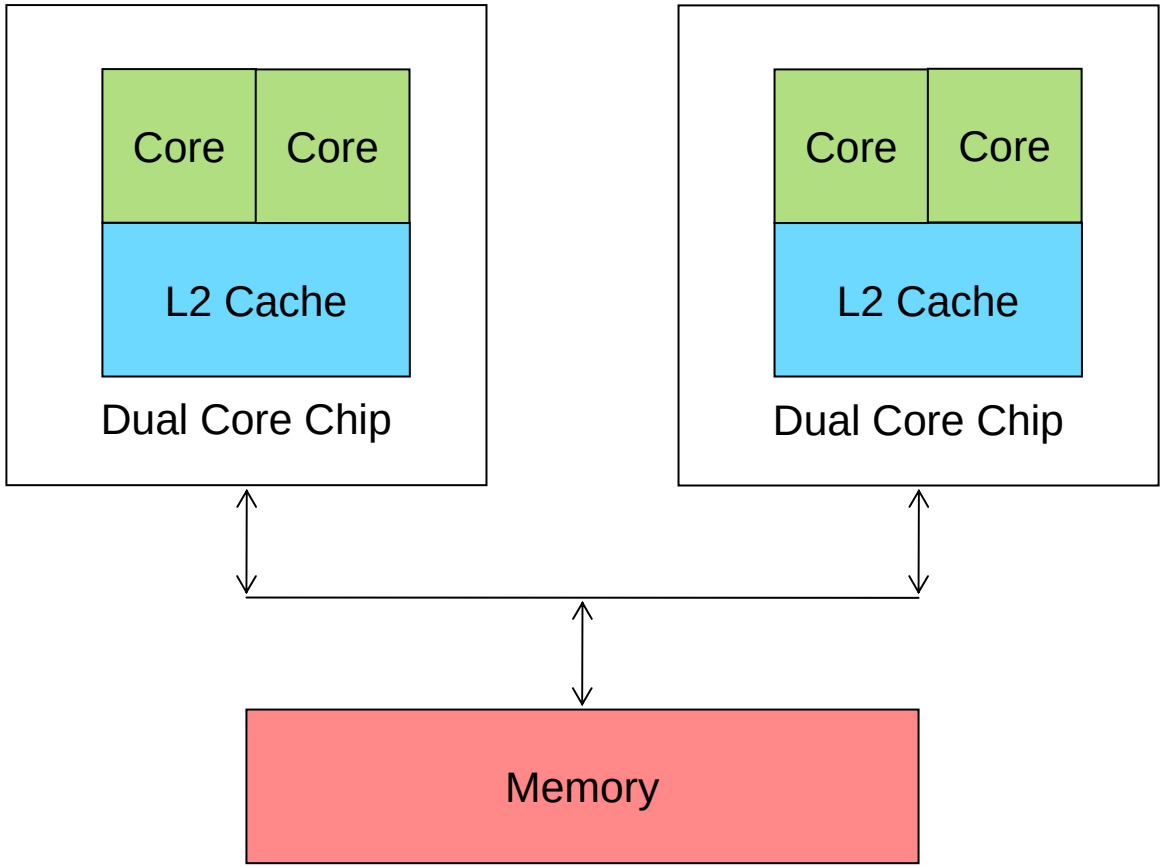
Overview

- The (past) world of reconfigurable computing
- The (past) world of multi-core
- The (emerging) world of reconfigurable multi-core architectures
- Illustrative analysis
- Conclusions

Drivers for multi-core technology path

- Single-core path leading to increased cost, heat, and power consumption
- Single-core path widens the pocessor/memory speed gap
- Multi-core path transparent to many application domain developers
- Multi-core path can improve performance of threaded software

Typical multi-core architecture*



*L. Chai, Q. Gao, D.K. Panda, “Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System,” *Seventh Int'l Symposium on Cluster Computing and the Grid (CCGrid)*, Rio de Janeiro - Brazil, May 2007.

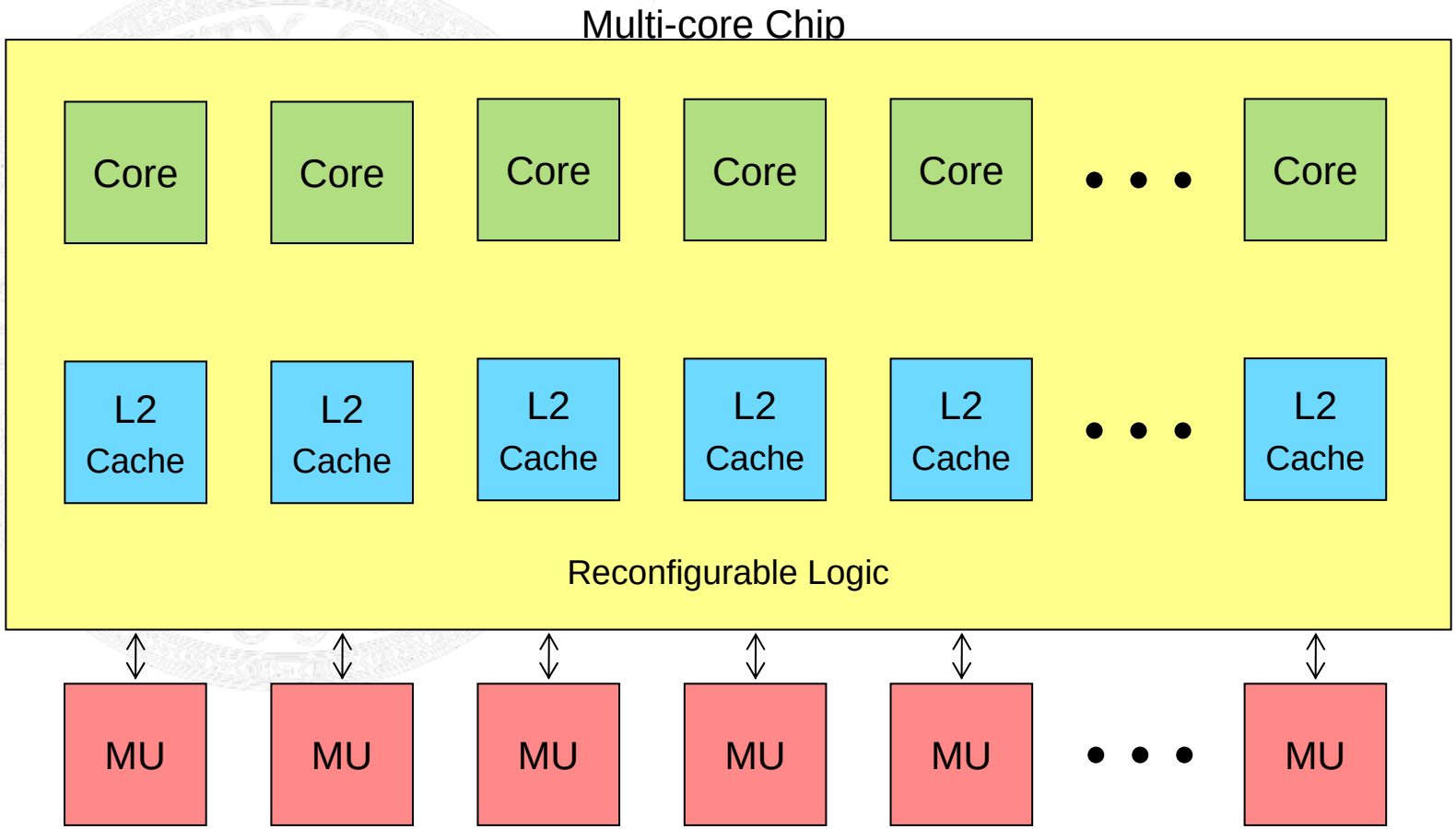
Overview

- The (past) world of reconfigurable computing
- The (past) world of multi-core
- The (emerging) world of reconfigurable multi-core architectures
- Illustrative analysis
- Conclusions

Emerging drivers and requirements for multi-core architectures

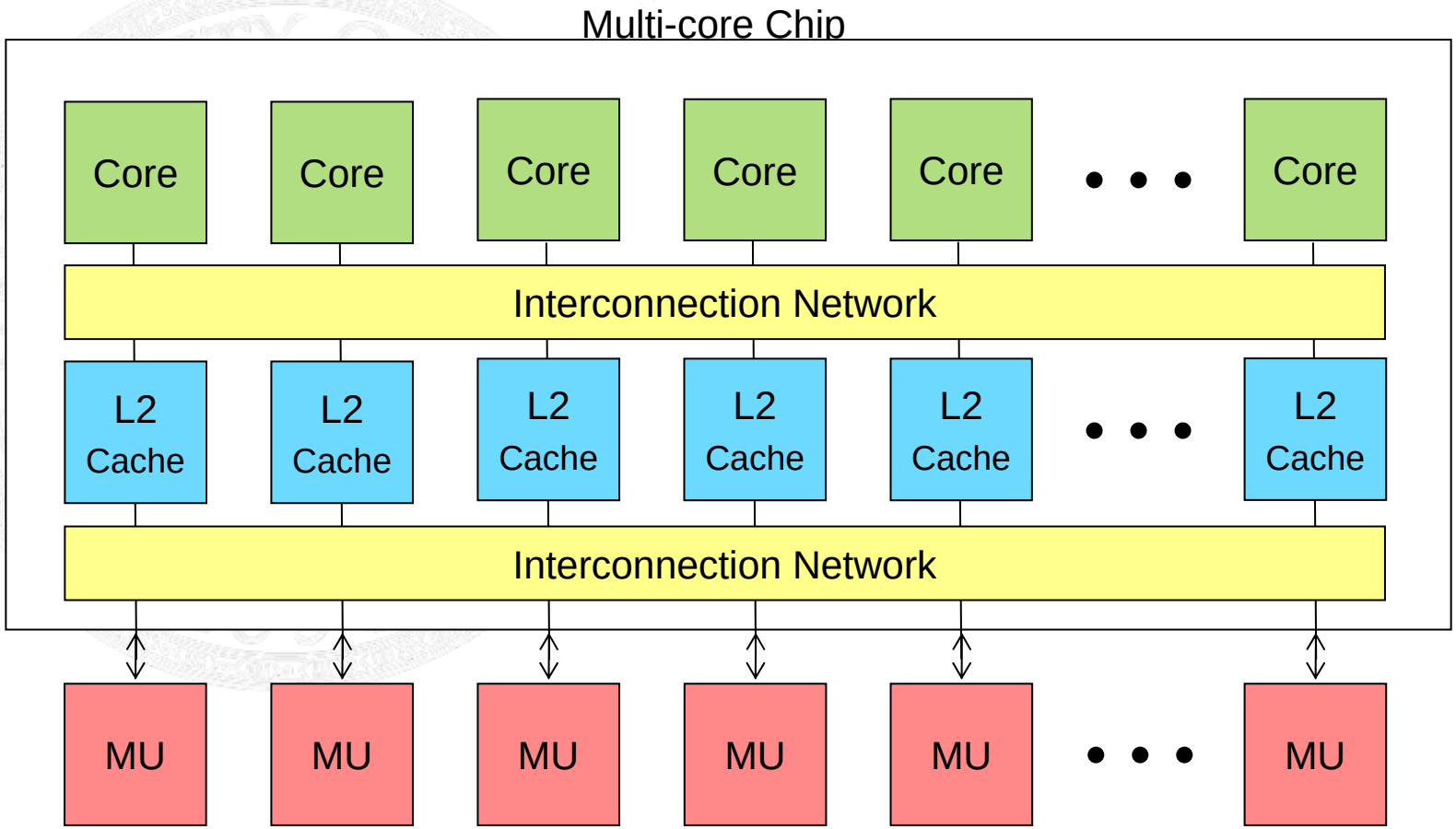
- Scale to support massively data parallel (SPMD) applications
- Match coupling among cores with application granularity
- Power is a major challenge for large data centers and supercomputing facilities

Hybrid architectural framework

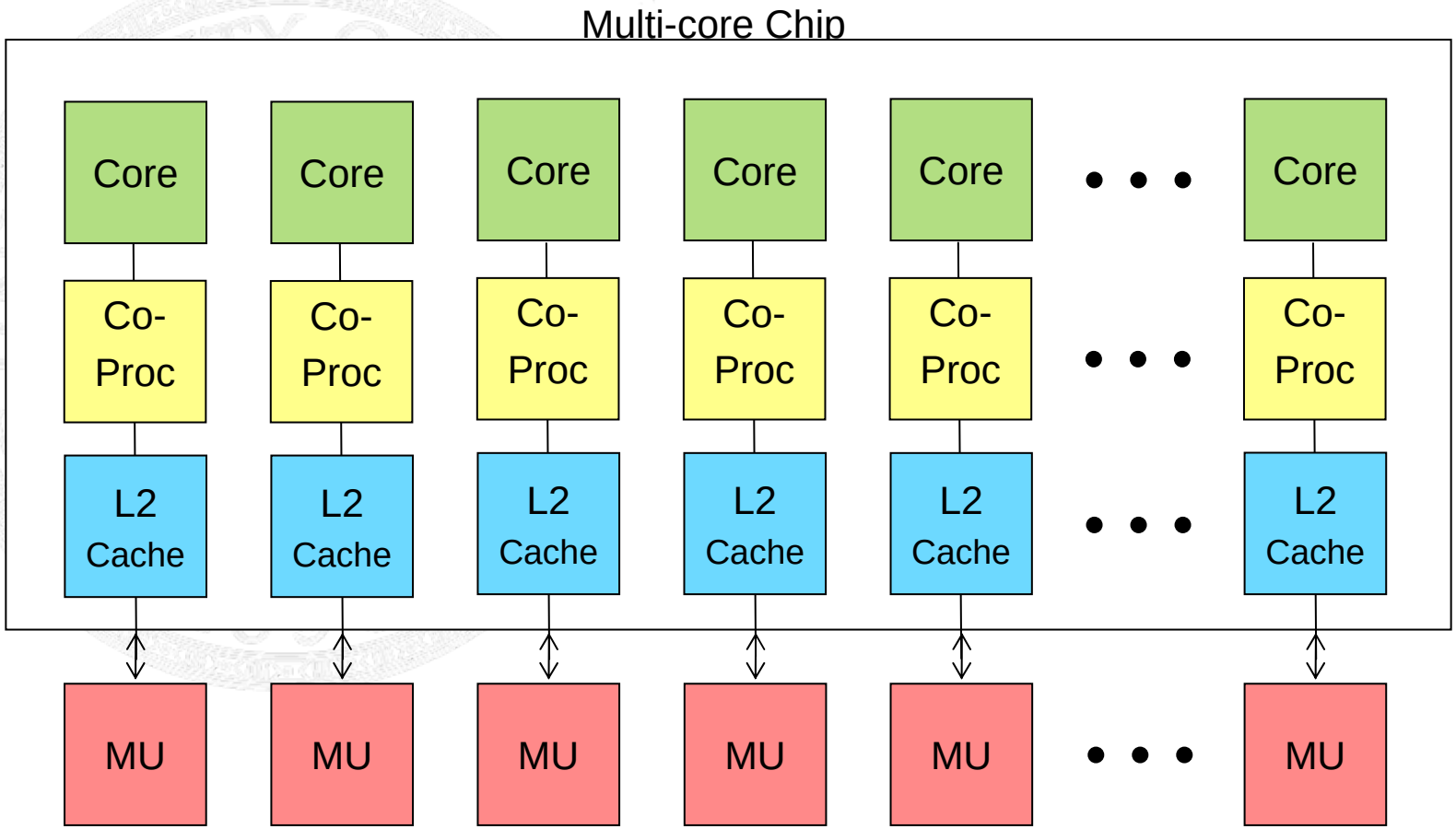


Reconfigurable logic

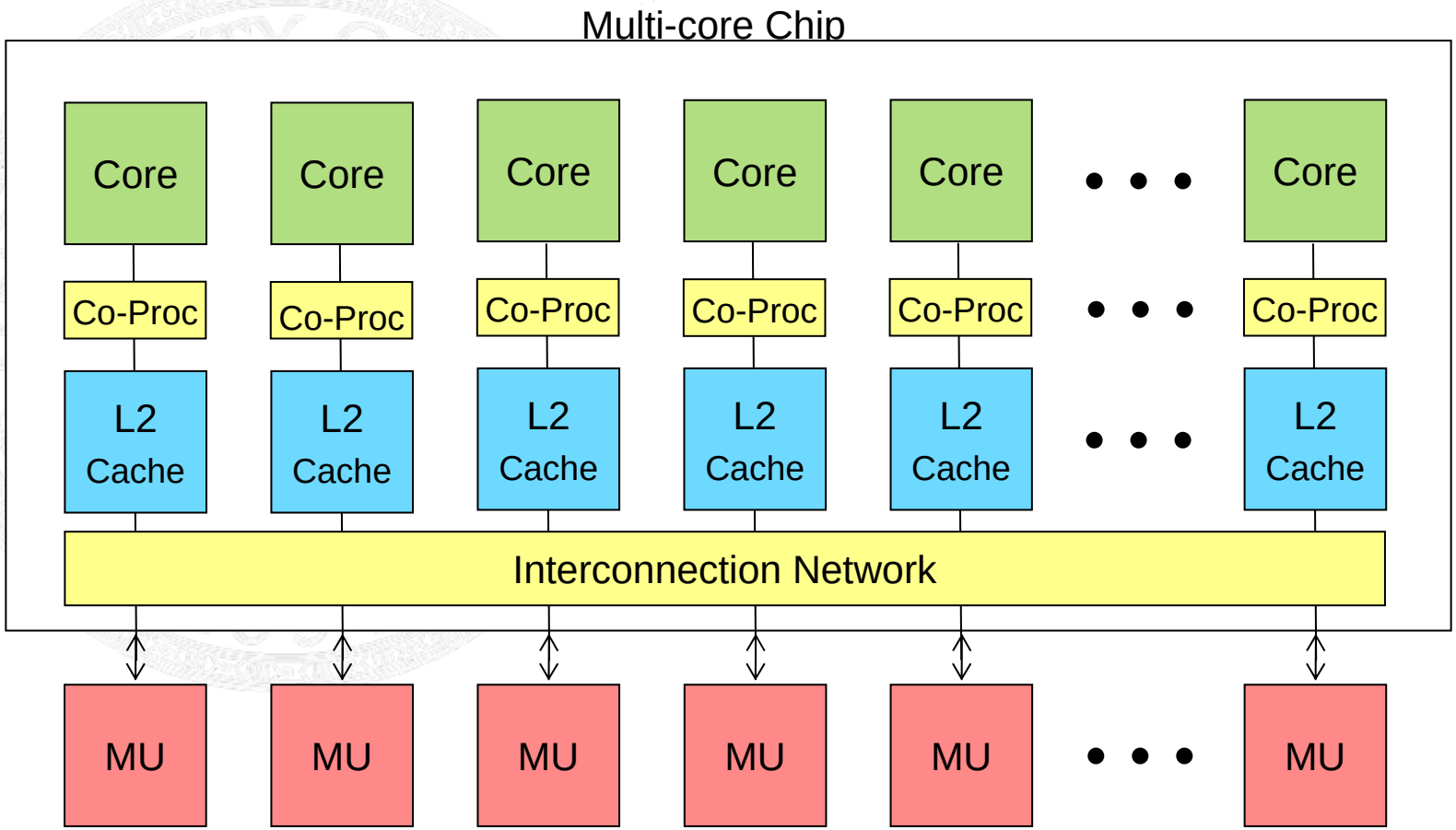
Shared everything configuration



Shared nothing configuration



Hybrid configuration

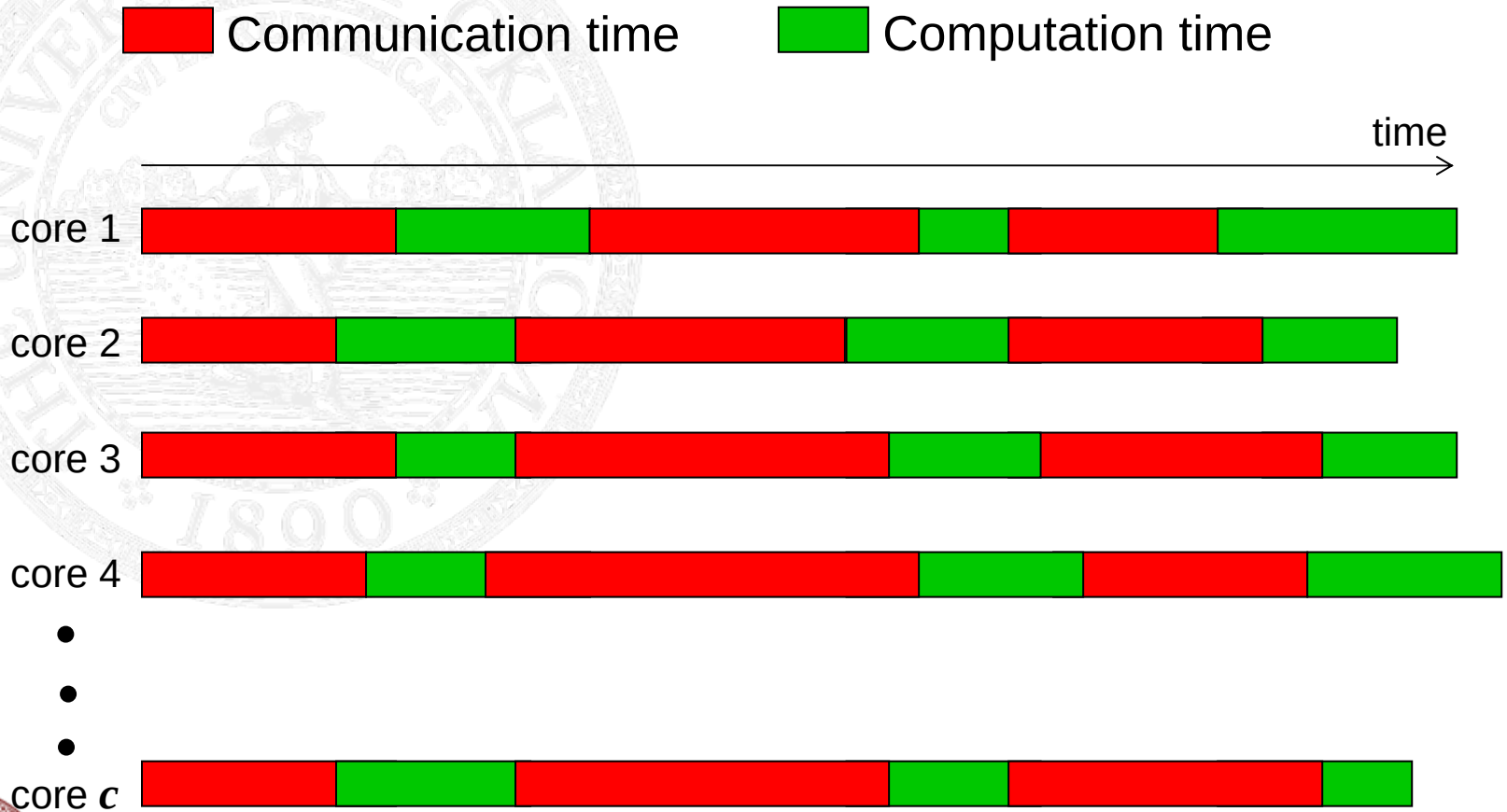


Features of hybrid architecture

- Match core coupling and core processing capacity with application granularity
  - Fixed multiprocessor architecture not well matched with all application granularities
  - Proposed reconfigurable multi-core architecture can be configured to match core coupling with application granularity

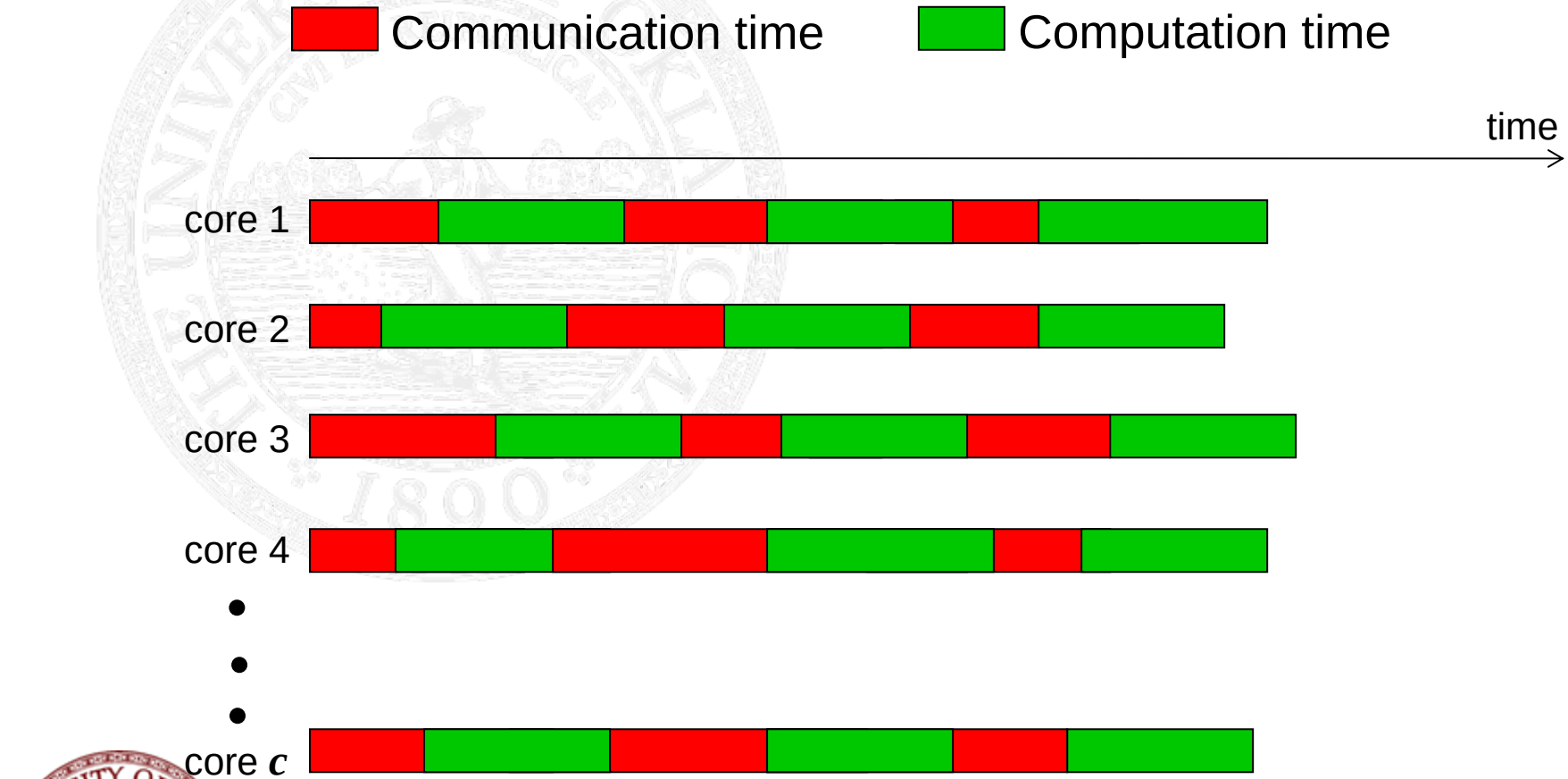
Mismatched SPMD execution

Core coupling too loose relative to application granularity



Matched SPMD execution

Core coupling tightened to match application granularity



Overview

- The (past) world of reconfigurable computing
- The (past) world of multi-core
- The (emerging) world of reconfigurable multi-core architectures
- Illustrative analysis
- Conclusions

Illustrative Analysis

- Notation
  - Number of cores: c
  - Problem size: n
  - Sequential time complexity: $T_S(n)$
  - Parallel time complexity:

$$T_P(c,n) = K \times f(c,n) + L \times g(c,n)$$

  - Computational complexity: $f(c,n)$
  - Communication complexity: $g(c,n)$
  - Core coupling ratio: K / L

Example

Sequential Time: $T_S(n) = n$

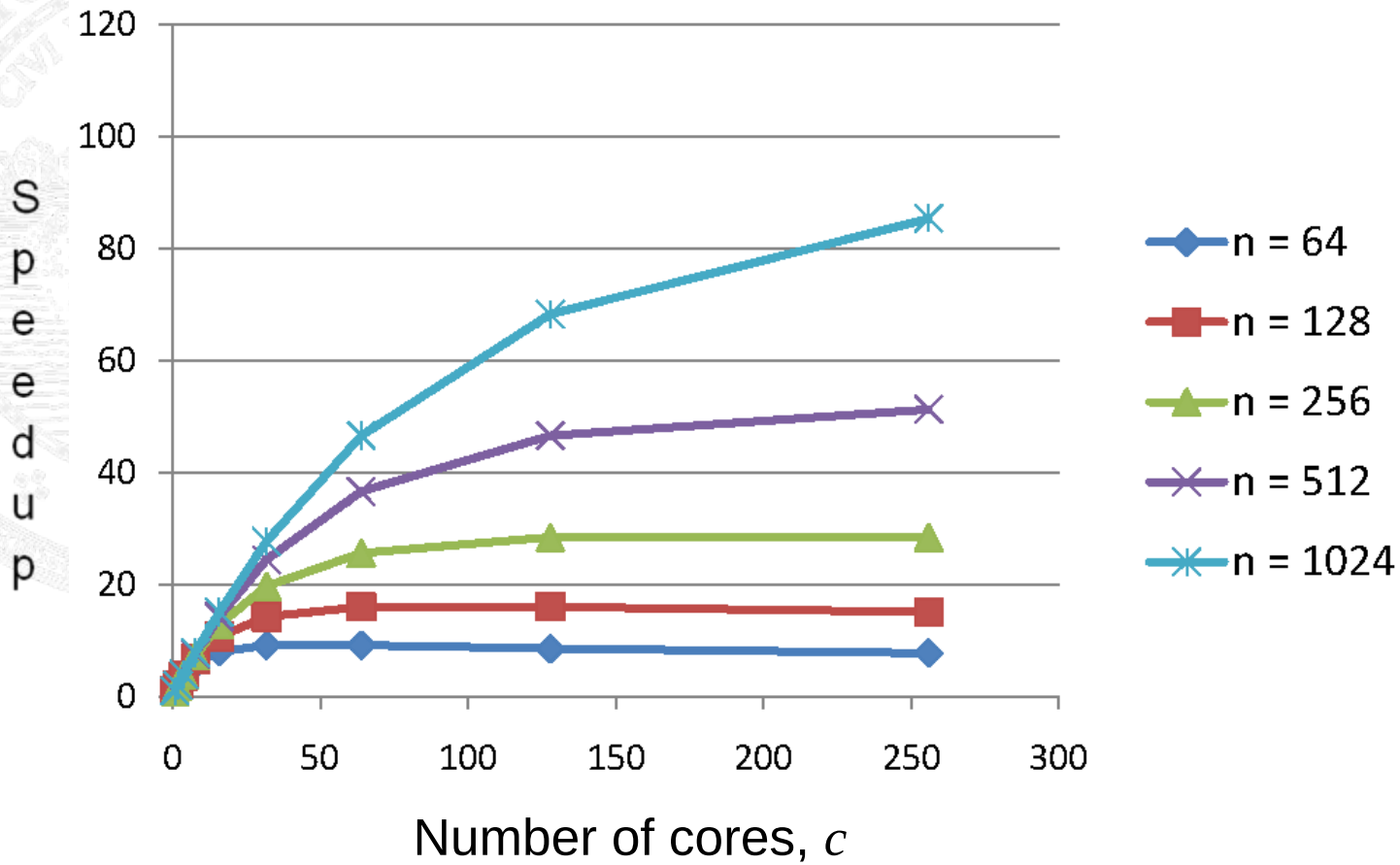
Parallel Time: $T_P(c, n) = K \times (n/c) + L \times \log c$

Speedup: $$S = \frac{n}{K \times (n/c) + L \times \log c}$$

The value of K: related to core processing capacity

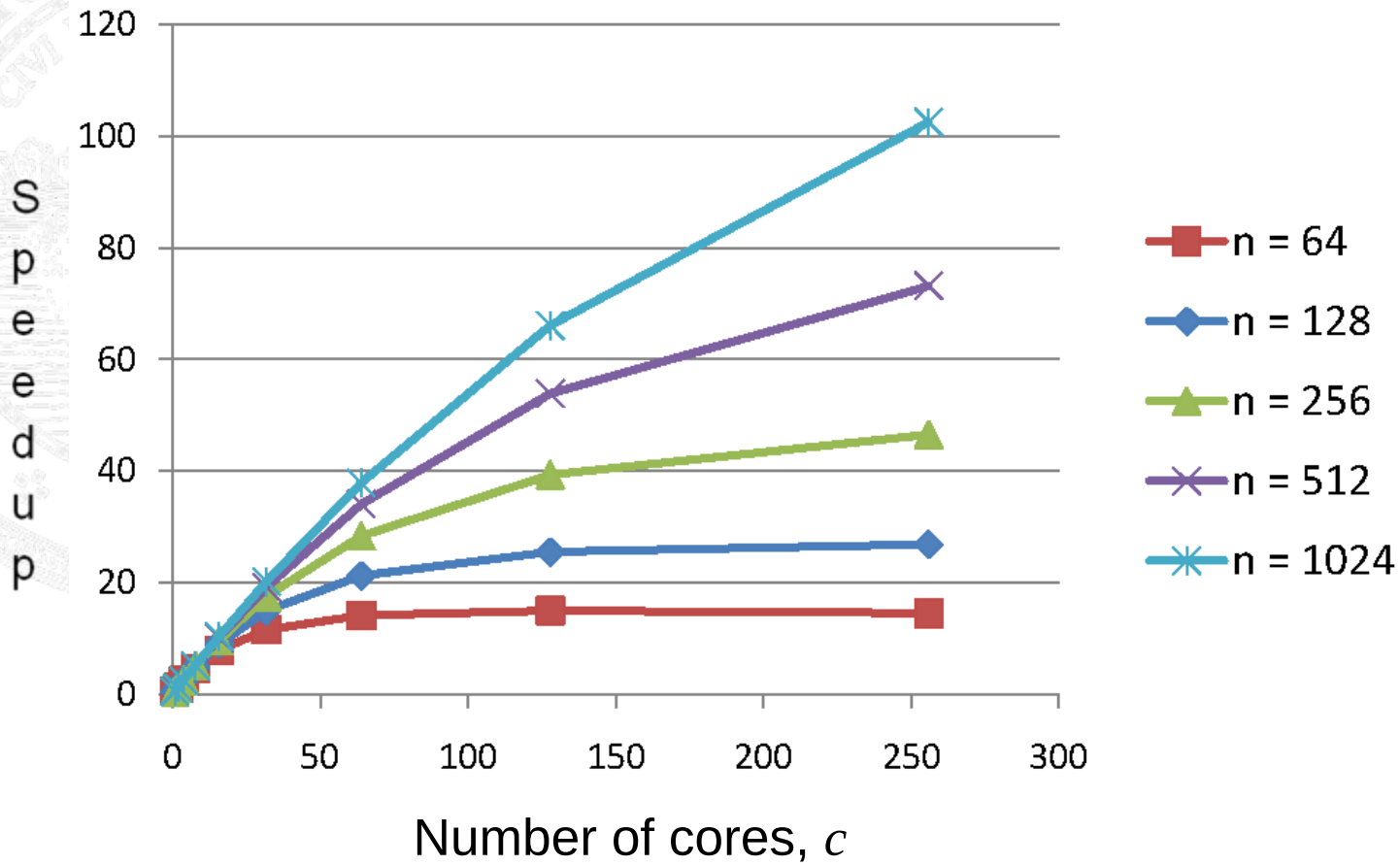
The value of L: related to interconnection among cores

$K = 1.0, L = 1.0$


120
100
80
60
40
20
0
Speedup
0
50
100
150
200
250
300
Number of cores, c
n = 64
n = 128
n = 256
n = 512
n = 1024

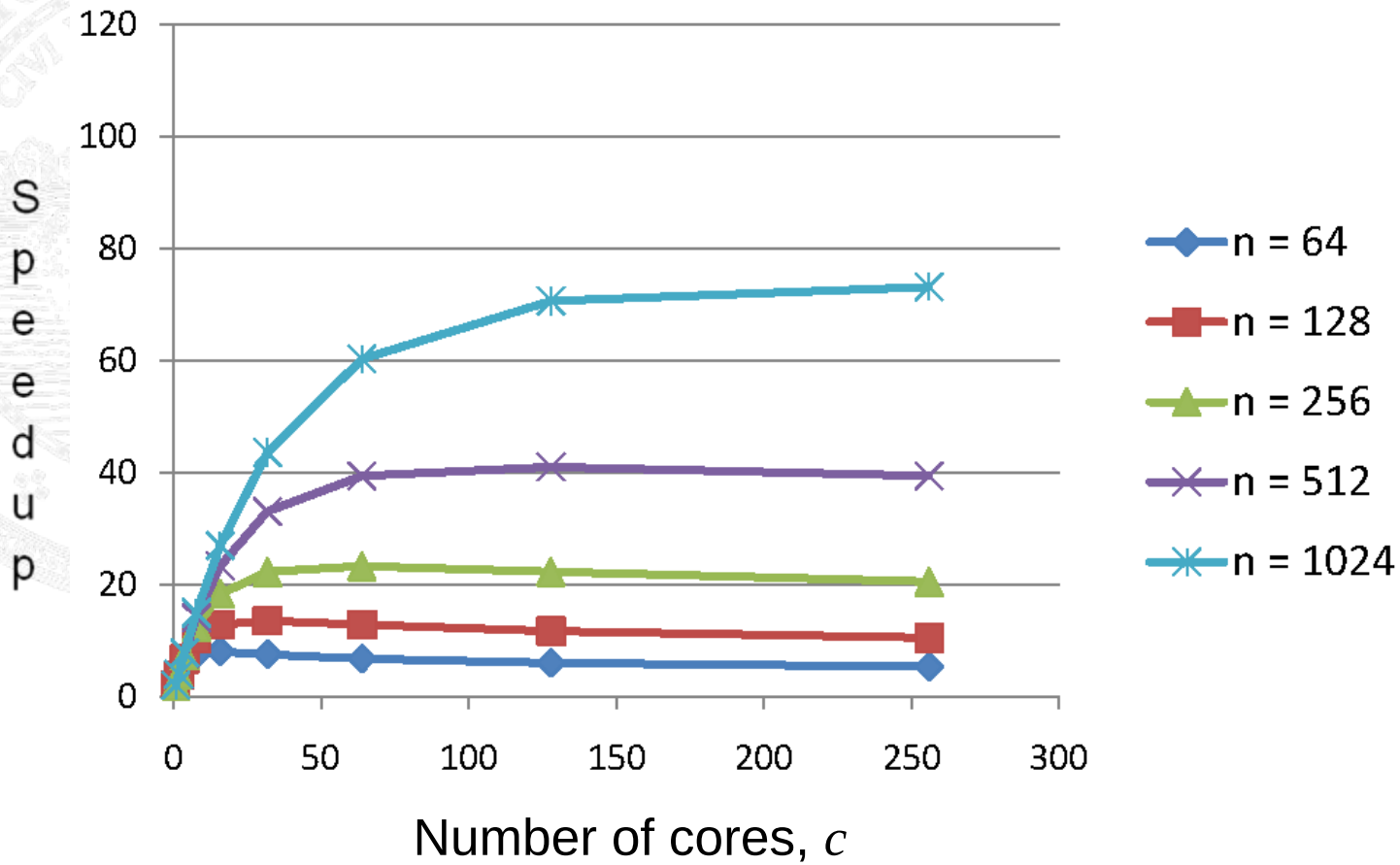

THE UNIVERSITY OF OKLAHOMA
CIVI ET REIPUBLICAE
1890

$K = 1.5, L = 0.5$


120
100
80
60
40
20
0
Speedup
0
50
100
150
200
250
300
Number of cores, c
n = 64
n = 128
n = 256
n = 512
n = 1024

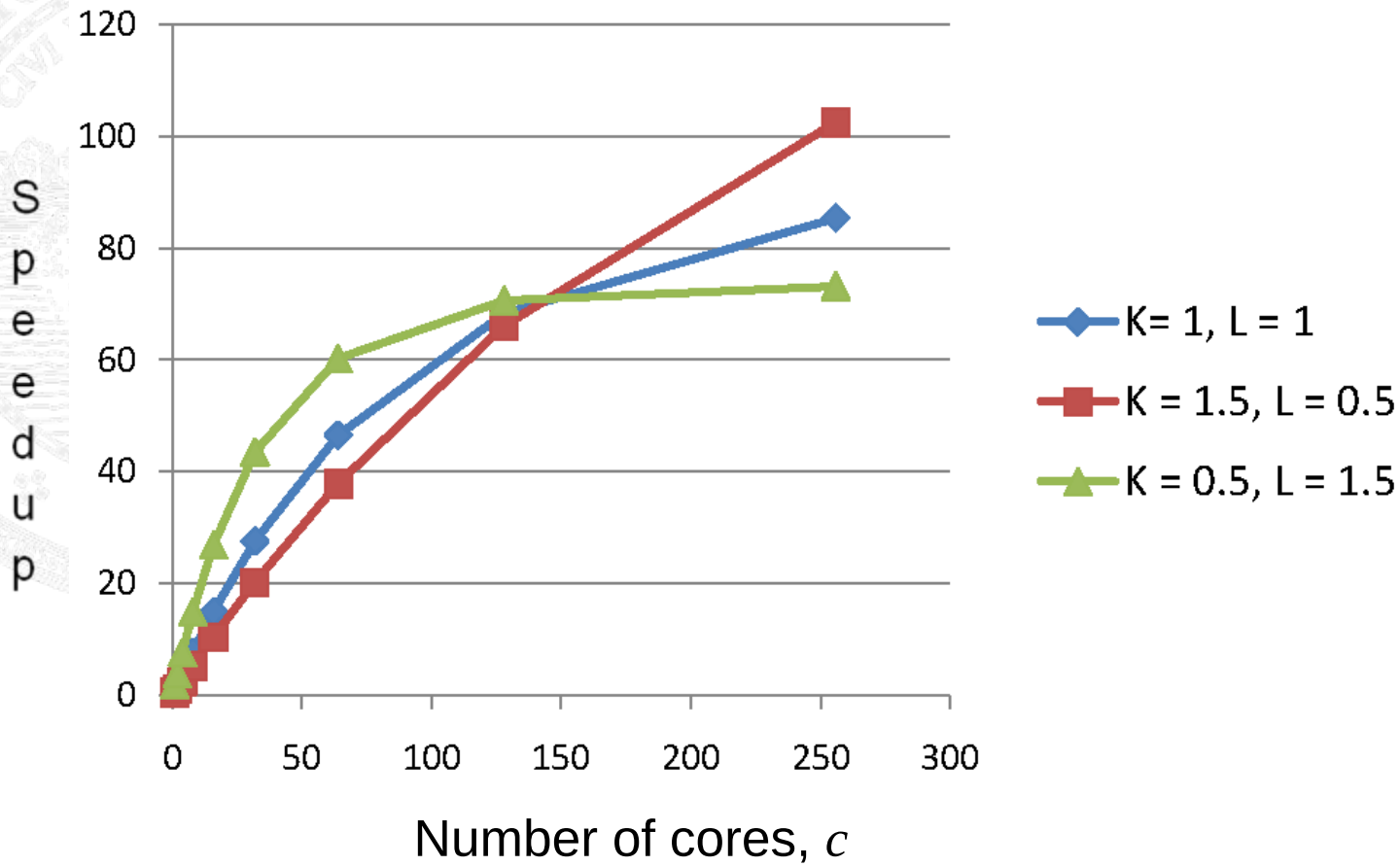

THE UNIVERSITY OF OKLAHOMA
CIVI ET REIPUBLICAE
1890

$K = 0.5, L = 1.5$


120
100
80
60
40
20
0
Speedup
0
50
100
150
200
250
300
Number of cores, c
n = 64
n = 128
n = 256
n = 512
n = 1024


THE UNIVERSITY OF OKLAHOMA
CIVI ET REIPUBLICAE
1890

$n = 1024$


120
100
80
60
40
20
0
Speedup
0
50
100
150
200
250
300
Number of cores, c
K= 1, L = 1
K = 1.5, L = 0.5
K = 0.5, L = 1.5


THE UNIVERSITY OF OKLAHOMA
CIVI ET REIPUBLICAE
1890

Overview

- The (past) world of reconfigurable computing
- The (past) world of multi-core
- The (emerging) world of reconfigurable multi-core architectures
- Illustrative analysis
- Conclusions

Conclusions

- Current multi-core approaches may not scale to support massive parallelism
- Hybrid reconfigurable multi-core approach enables trades between core coupling and core processing capacity
- More research needed in reconfigurable micro-architecture to support hybrid architectures