Large Scale Multiprocessing: NoCs to Supercomputers

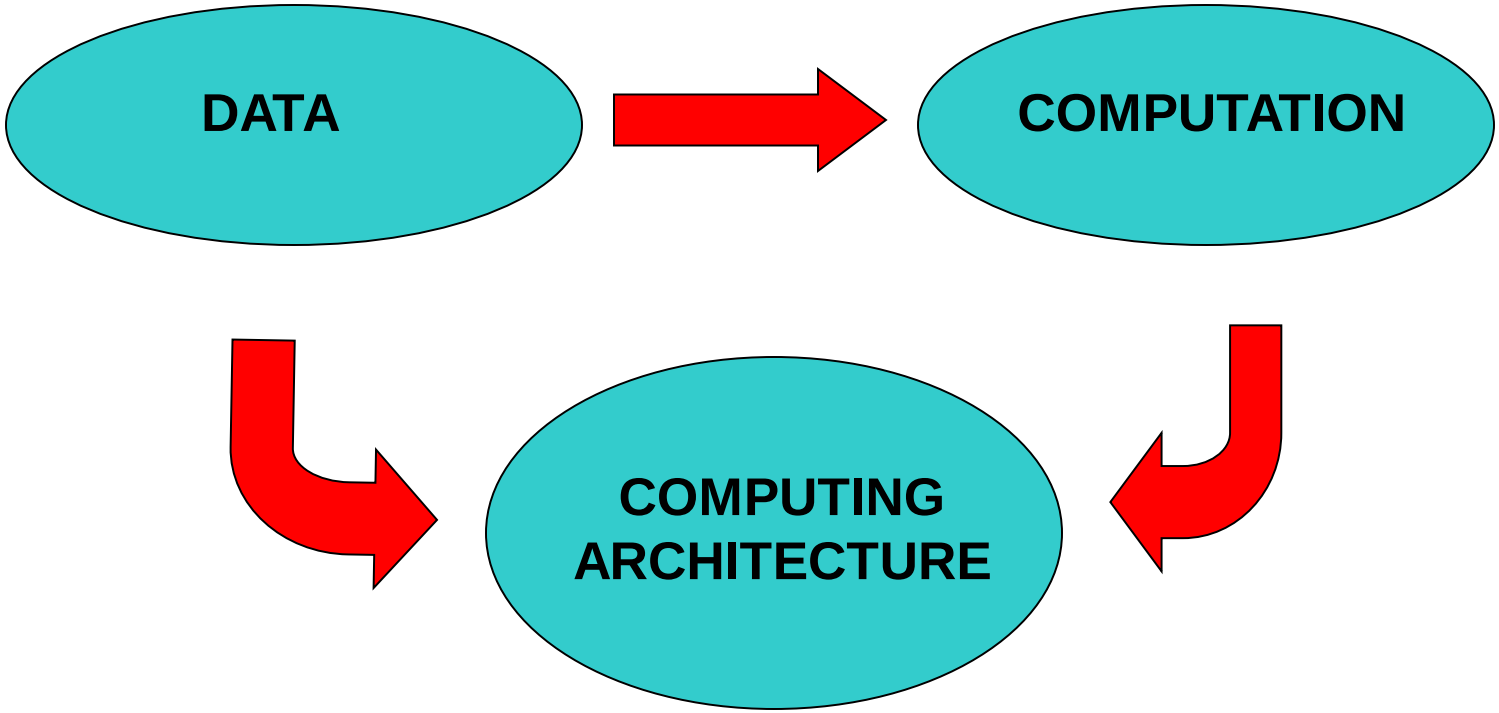
Souradip Sarkar
Researcher
Intel Exascience labs and Ghent University

2012

Outline

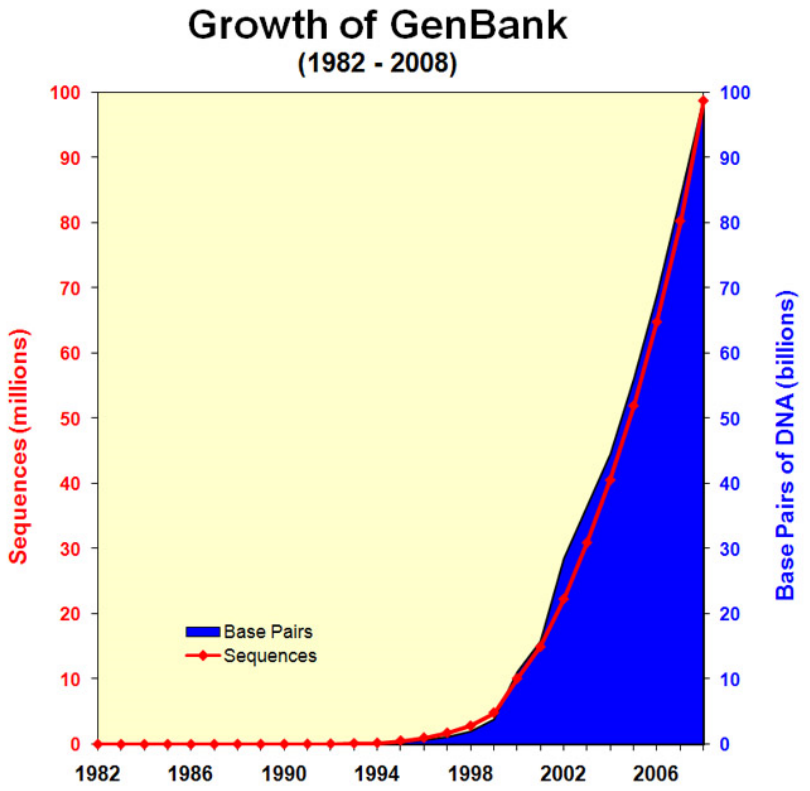
- Motivation
- Introduction
  - Multi-cores, NoCs, Applications
- Biocomputing applications
  - Sequence Alignment
  - Phylogenetic Reconstruction
- Conclusions
- Power aware multi-core simulation

Motivation



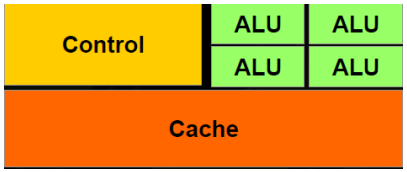
Why Multi-cores Chips?

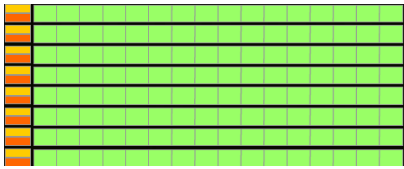
- Need for massive computation power
- Scientific Applications
  - Computational Chemistry and Biology
  - Weather prediction
  - Astrophysics
  - Forensics
  - Document and text processing
- Consumer electronics
  - Graphics, audio and video processing



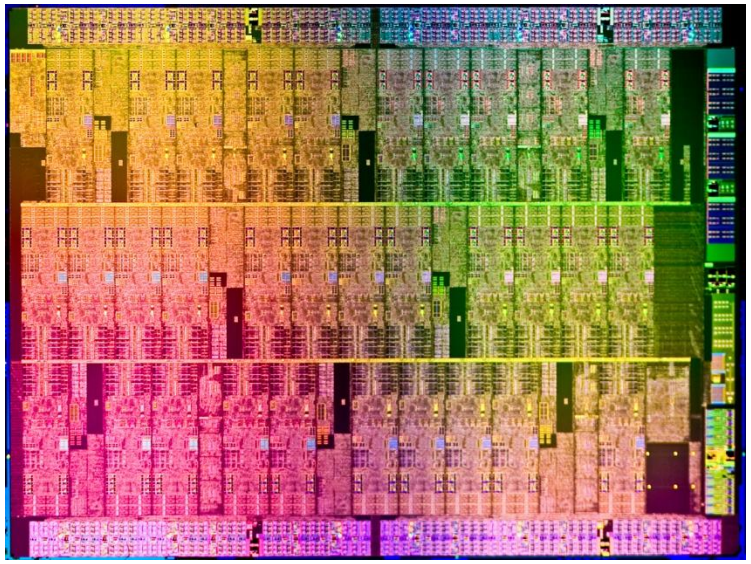
Massive Scale of Computing through Multi-Core Chips

- Keep up with the demands on computational power
  - Scaling of clock frequency => not happening
- Increasing number of cores=>parallelism
  - General purpose dual-core and quad core processors from Intel and AMD
  - MIC Architecture
  - Custom system on Chips





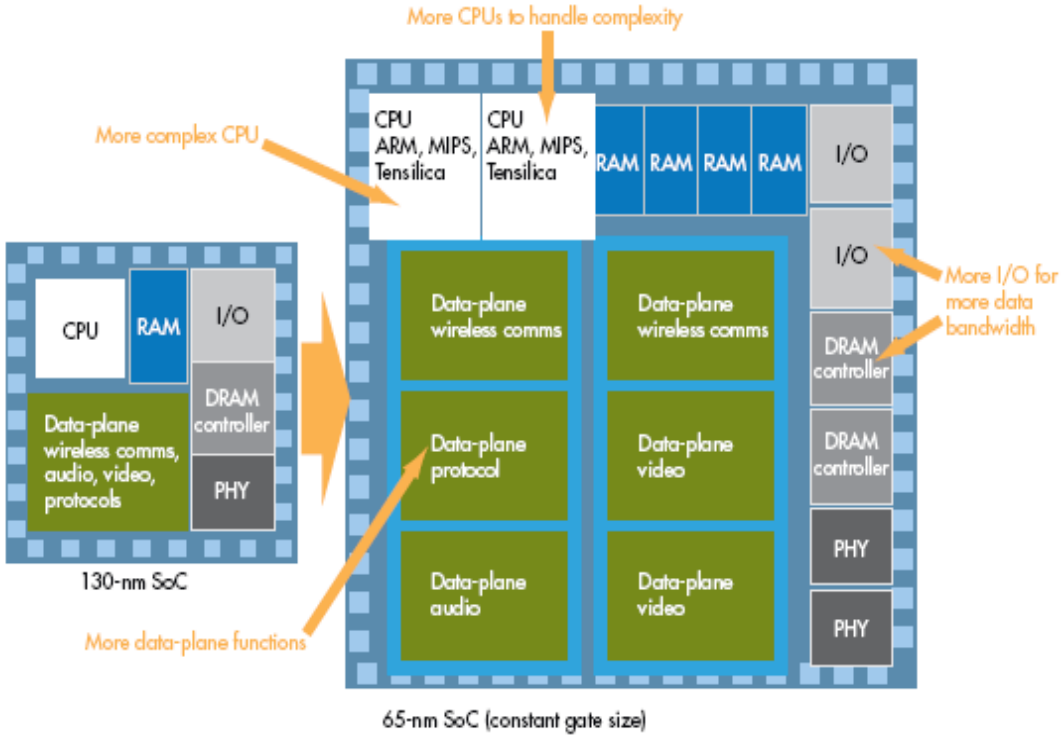
CPU GPU

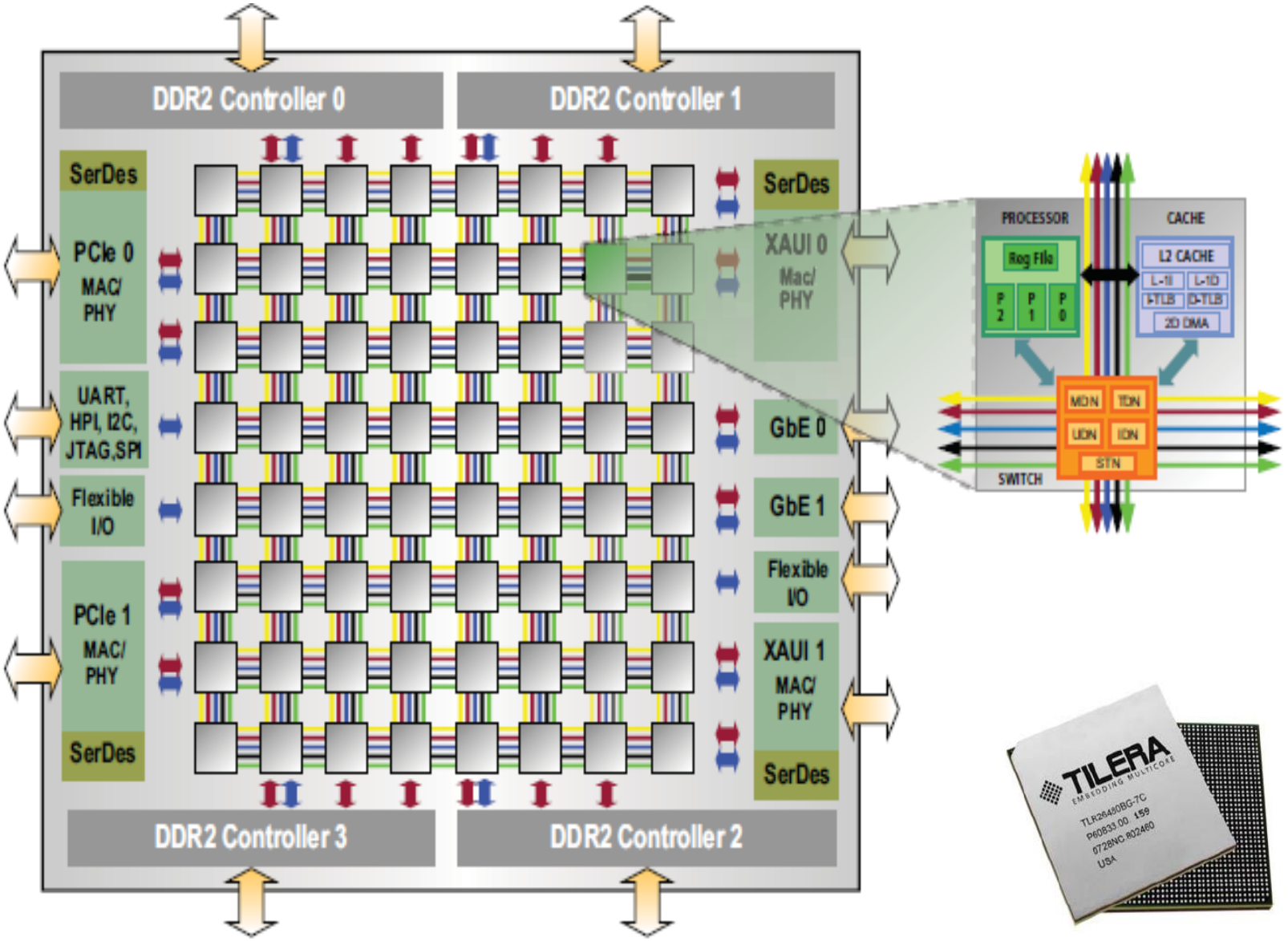


Courtesy: Nvidia & Intel

Network-on-Chip

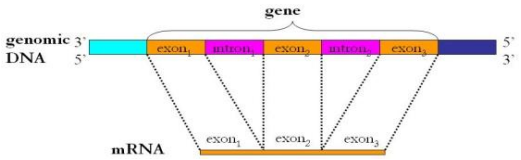
- Driven by
  - Increased levels of integration
  - Complexity of large SoCs
  - Need for platform-based design methodologies





Courtesy Tilera Corp.

Examples of Bio-Computing Applications

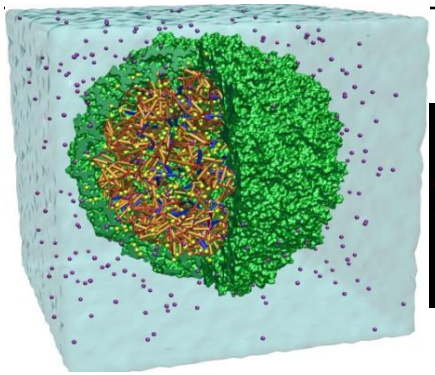


atgcatatcatag-agta

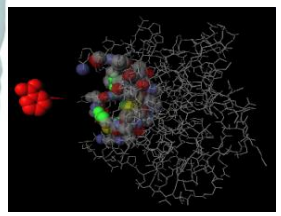
atgcgtatc- tagcagta

Sequence alignment

Tree of Life (Phylogenetics)



Molecular Dynamics



Molecular Docking

Algorithms:	Combinatorial Optimization	Simulation-based
Software-level Parallelism (coarse-grained)		
CPU	FPGA	GPU	CBE	General purpose Multicores	**Custom NoCs**

Hardware-level parallelism (fine-grained) w/ or w/o co-processor mode

Figure sources:

Tree of life Web Project: http://www.tolweb.org/tree/
Theoretical and computational Biophysics Group, UIUC, http://www.ks.uiuc.edu/Research/STMV/
Molecular docking http://wwwcs.uni-paderborn.de/~lst/HotDock/

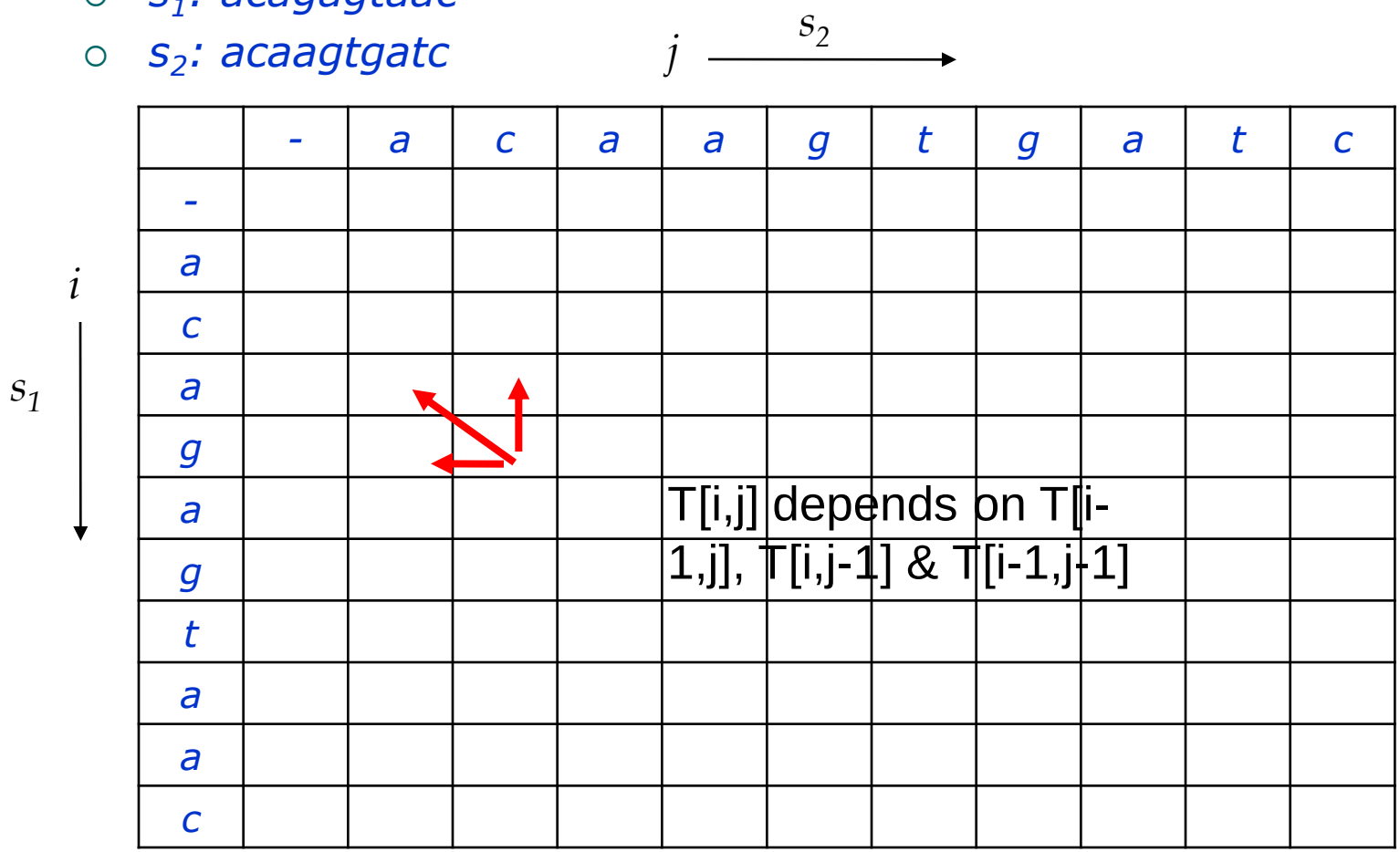
NoC-based Hardware Accelerators

- Custom Core & interconnect infrastructure.
- Offers freedom to choose communication architecture most apt for the application.
- Other existing hardware solutions more generic => fail to exploit the custom features pertinent to a particular application.
- Evaluate performance of NoC-based accelerators in comparison to other hardware accelerators.

Application – Sequence Alignment

Global Alignment: Dynamic Programming Table

- s_1: acagagtaac
- s_2: acaagtgatc



	-	a	c	a	a	g	t	g	a	t	c
-											
a											
c											
a											
g											
a											
g											
t											
a											
a											
c											

Global Alignment: Dynamic Programming Table

- s_1: acagagtaac
- s_2: acaagtgatc

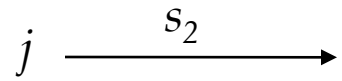
j → s_2

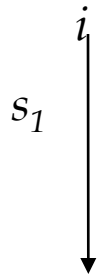
i ↓ s_1

	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1										
c	-2										
a	-3										
g	-4										
a	-5										
g	-6										
t	-7										
a	-8										
a	-9										
c	-10										

Global Alignment: Dynamic Programming Table

- s_1: acagagtaac
- s_2: acaagtgatc





	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
c	-2										
a	-3										
g	-4										
a	-5										
g	-6										
t	-7										
a	-8										
a	-9										
c	-10										

Global Alignment: Dynamic Programming Table

- s_1: *acagagtaac*
- s_2: *acaagtgatc*

$j \rightarrow s_2$

$i \downarrow s_1$

	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
c	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
a	-3										
g	-4										
a	-5										
g	-6										
t	-7										
a	-8										
a	-9										
c	-10										

Global Alignment: Dynamic Programming Table

- s_1: acagagtaac
- s_2: acaagtgatc

j → s_2

i ↓ s_1

	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
c	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
a	-3	-1	1	3	2	1	0	-1	-2	-3	-4
g	-4	-2	0	2	2	3	2	1	0	-1	-2
a	-5	-3	-1	1	3	2	2	1	2	1	0
g	-6	-2	-2	0	2	4	3	3	2	1	0
t	-7	-3	-3	-1	1	3	5	4	3	3	2
a	-8	-4	-4	-2	0	2	4	3	5	4	3
a	-9	-5	-5	-3	-1	1	3	3	4	4	3
c	-10	-6	-4	-4	-2	0	2	2	3	3	**5**

Optimal score

Global Alignment: Dynamic Programming Table

- s_1: acagagtaac
- s_2: acaagtgatc

j → s_2

i ↓ s_1

	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
c	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
a	-3	-1	1	3	2	1	0	-1	-2	-3	-4
g	-4	-2	0	2	2	3	2	1	0	-1	-2
a	-5	-3	-1	1	3	2	2	1	2	1	0
g	-6	-2	-2	0	2	4	3	3	2	1	0
t	-7	-3	-3	-1	1	3	5	4	3	3	2
a	-8	-4	-4	-2	0	2	4	3	5	4	3
a	-9	-5	-5	-3	-1	1	3	3	4	4	3
c	-10	-6	-4	-4	-2	0	2	2	3	3	**5**

Optimal score

Global Alignment: Dynamic Programming Table

- s_1: *acagagtaac*
- s_2: *acaagtgatc*

j → s_2

i ↓ s_1

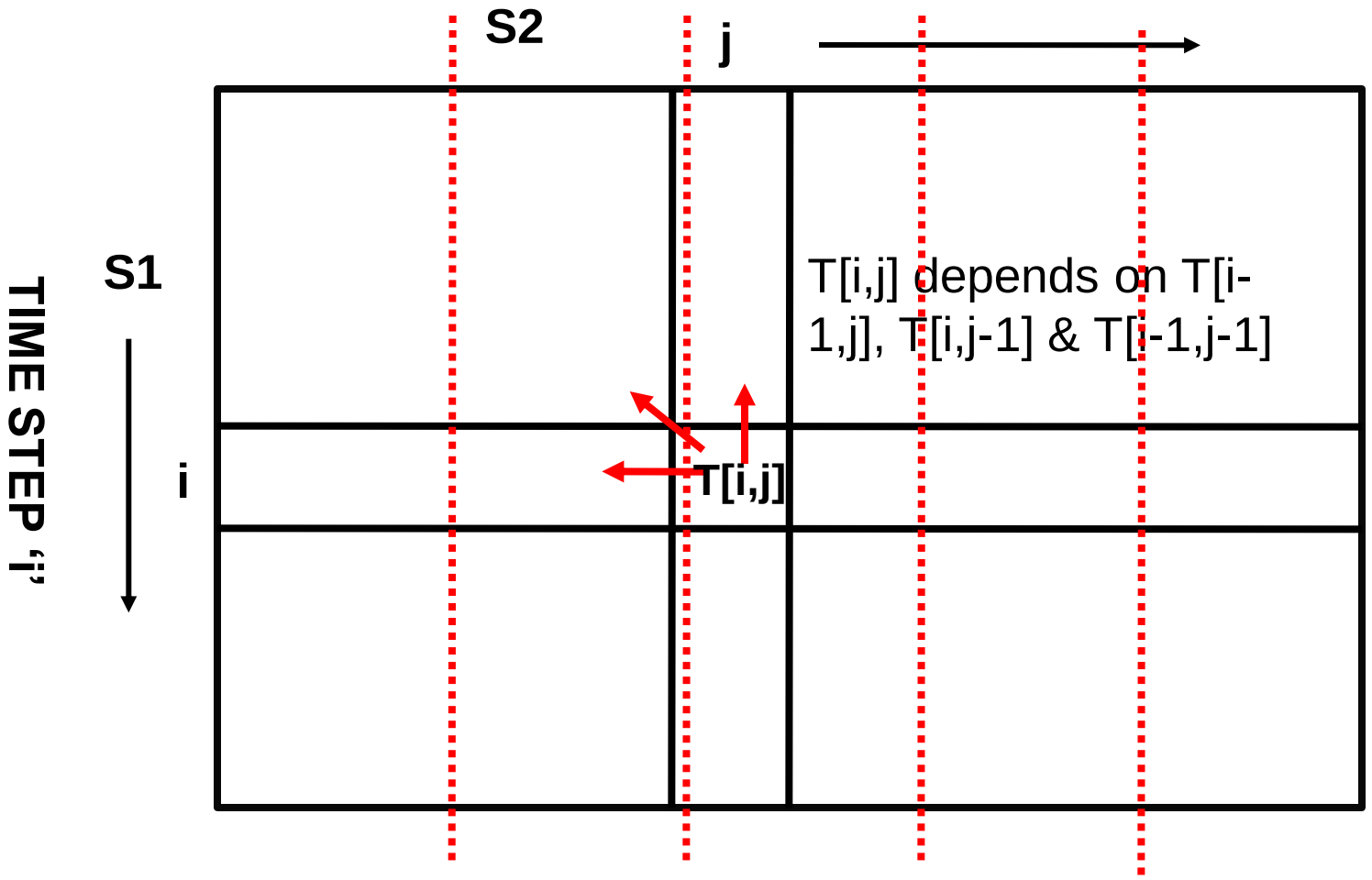
	-	a	c	a	a	g	t	g	a	t	c
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
a	-1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
c	-2	0	2	1	0	-1	-2	-3	-4	-5	-6
a	-3	-1	1	3	2	1	0	-1	-2	-3	-4
g	-4	-2	0	2	2	3	2	1	0	-1	-2
a	-5	-3	-1	1	3	2	2	1	2	1	0
g	-6	-2	-2	0	2	4	3	3	2	1	0
t	-7	-3	-3	-1	1	3	5	4	3	3	2
a	-8	-4	-4	-2	0	2	4	3	5	4	3
a	-9	-5	-5	-3	-1	1	3	3	4	4	3
c	-10	-6	-4	-4	-2	0	2	2	3	3	**5**

Optimal score

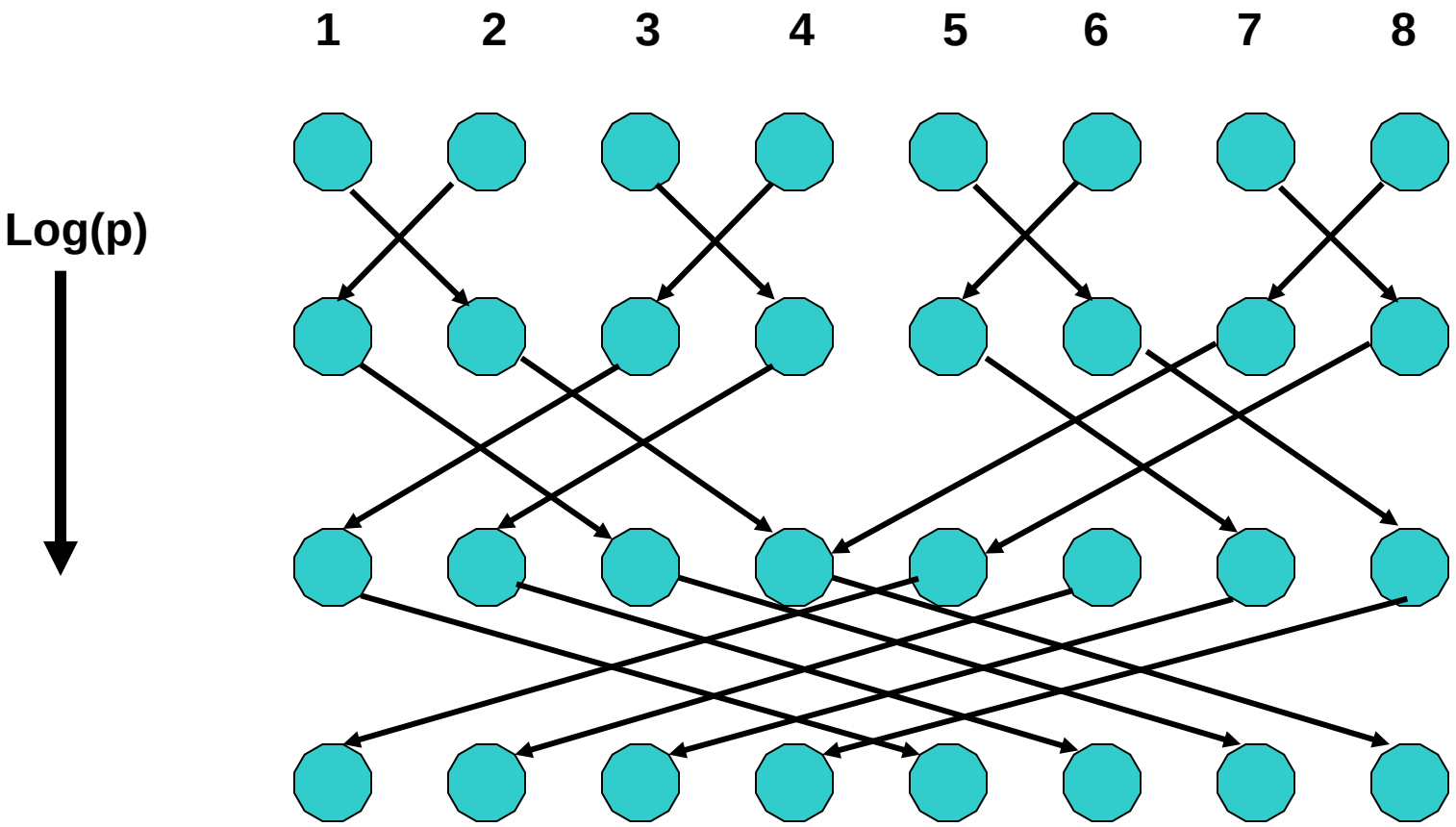
Related Algorithms

- Several course-grain parallel algorithms => varying degrees of computational complexities and ease of implementation.
- **Aluru's Parallel Prefix based approach** – O((m*n)/p) time and O(m+n/p) space.
- **Huang's Antidiagonal based approach** – $O((m+n)^2/p)$ time & O((m+n)/p) space.

Parallel Prefix Approach

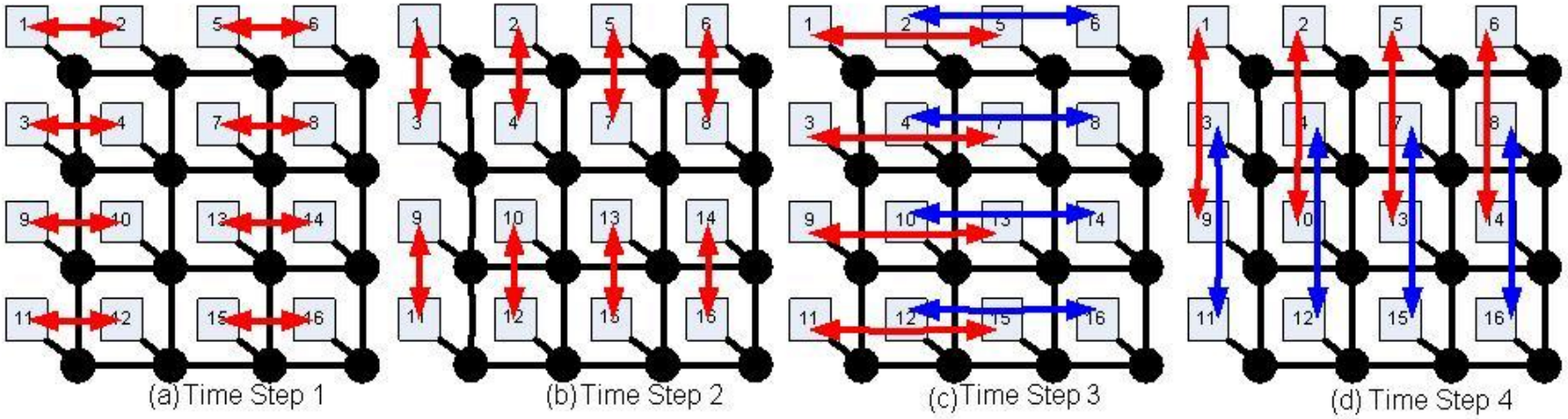


Parallel Prefix Approach - Communication

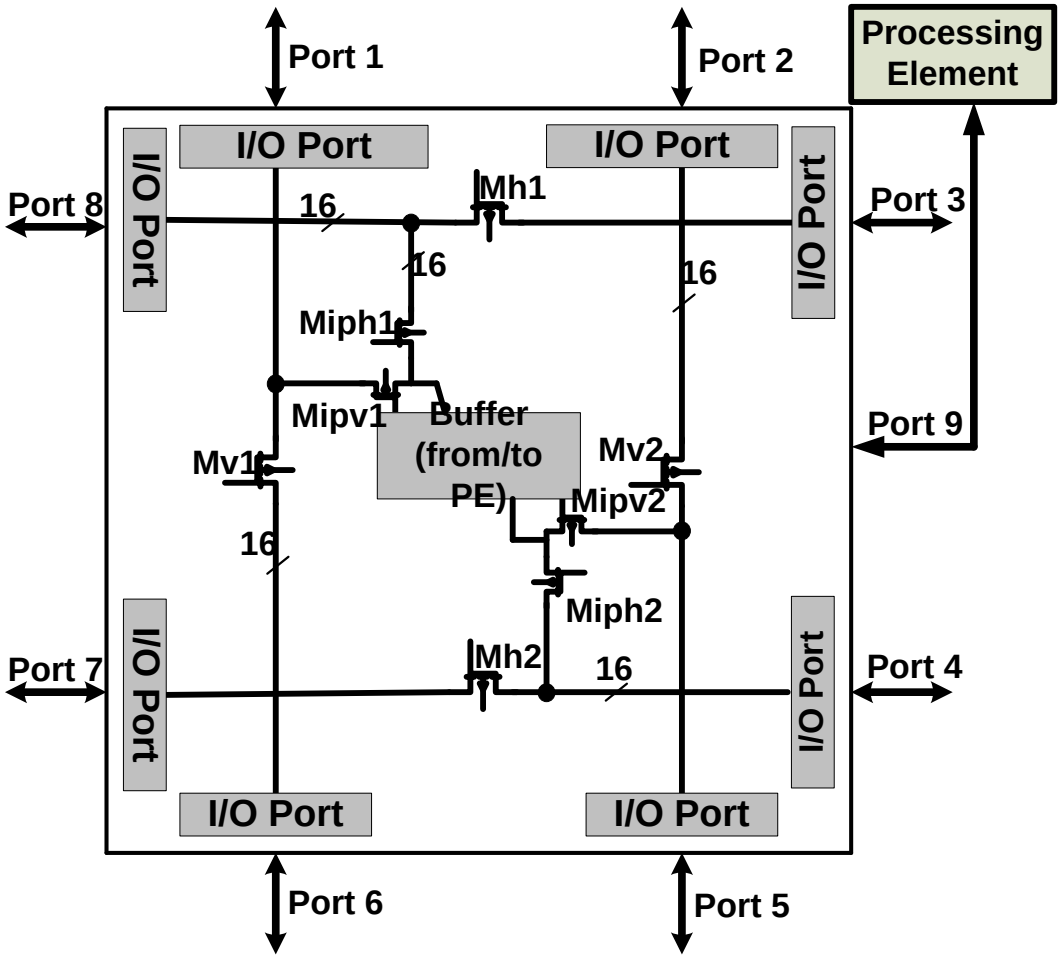


Communication Pattern

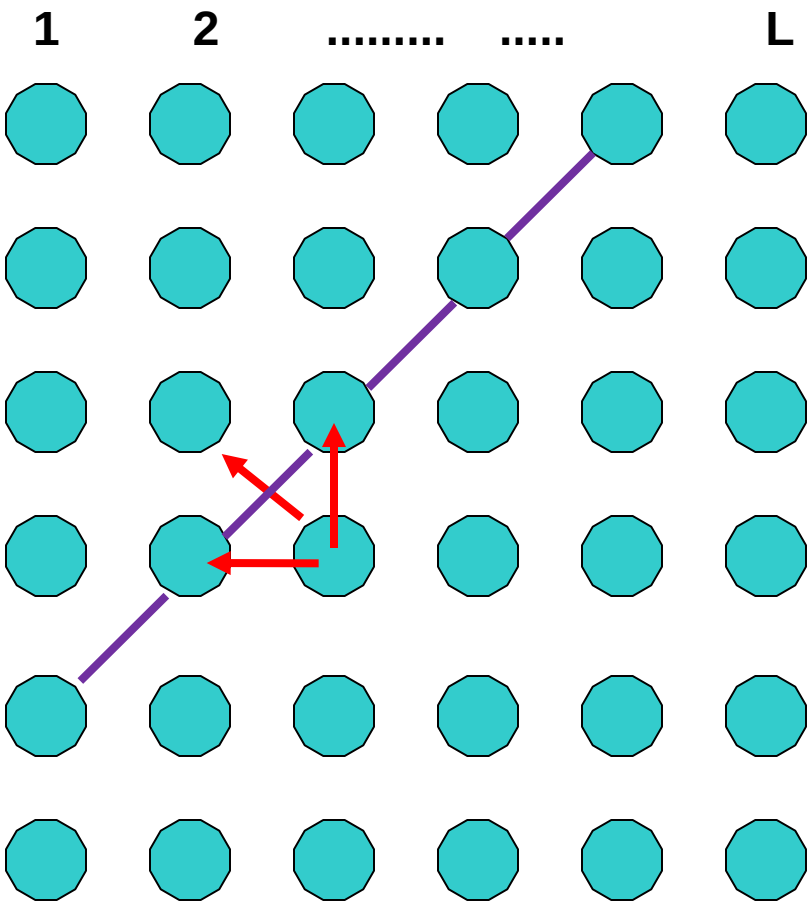
- Hypercubic communication mapped to 2 D Mesh.
- Integer communication => Circuit switching.
- Use of **Bypass strategy** for achieving communication in log(N) time steps.



NoC Switch Architecture for PSA

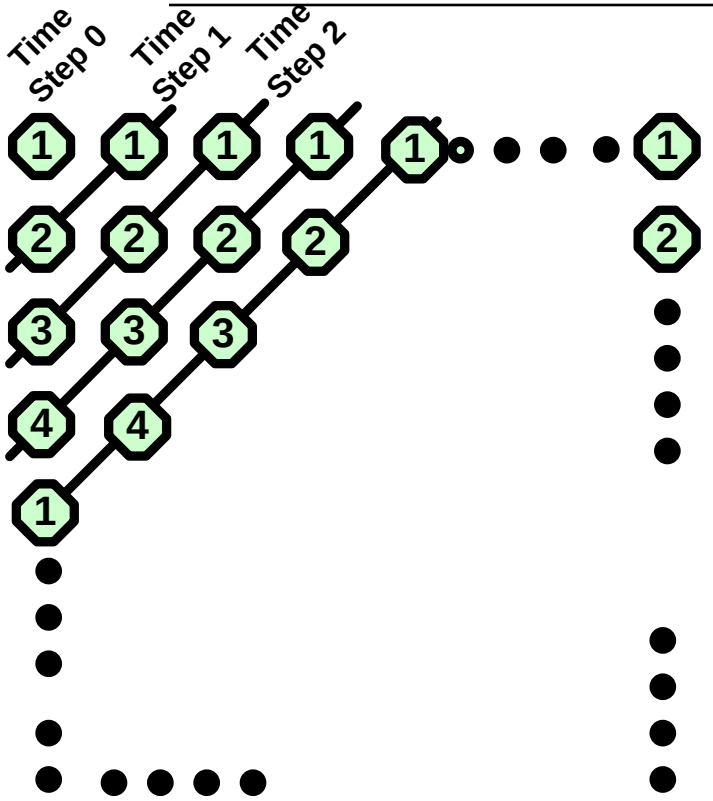


Anti-Diagonal Approach

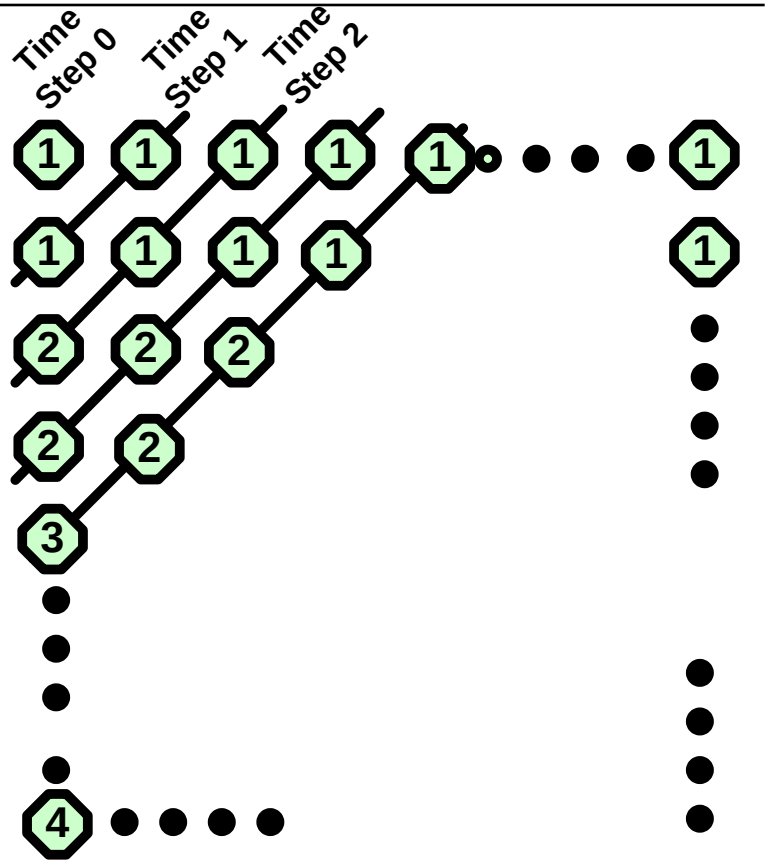


Cases:
1) L <= p
2) L > p

Anti-Diagonal Approach



Strategy 1



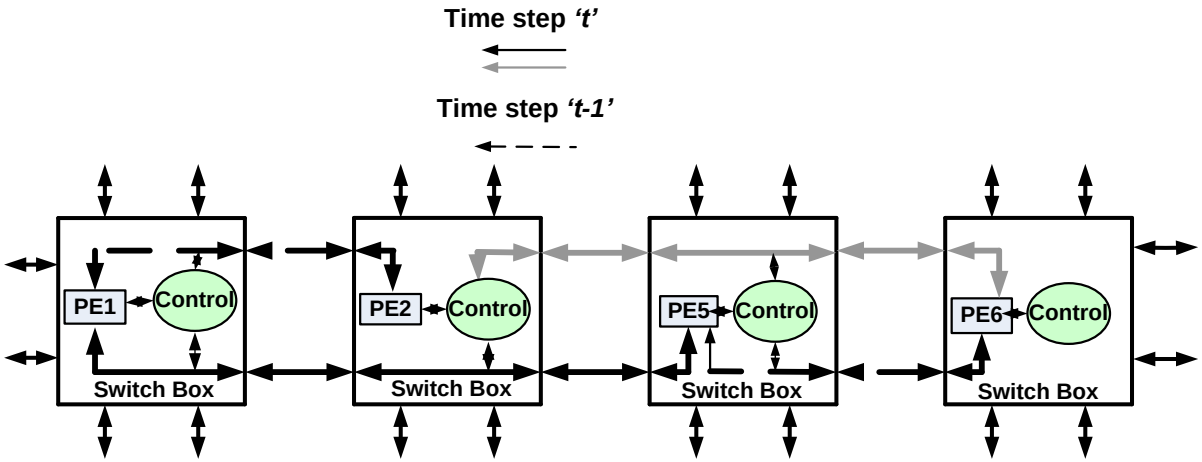
Strategy 2

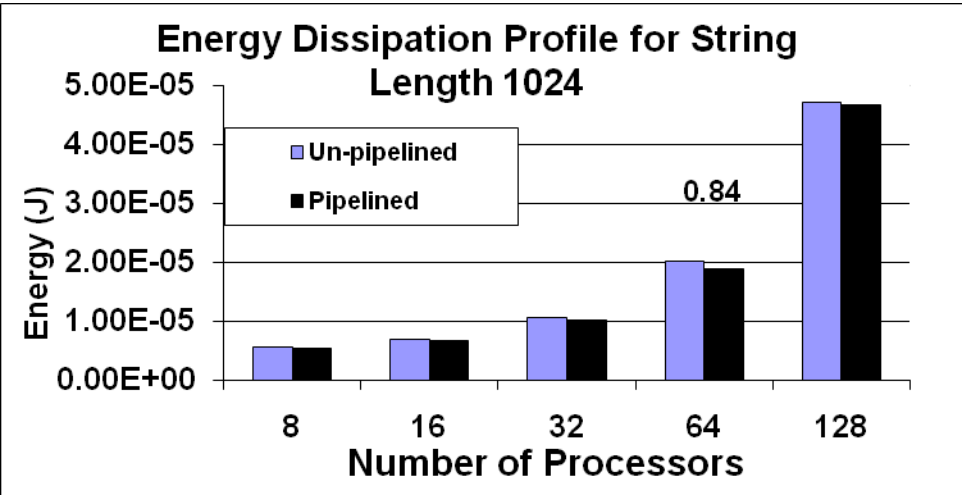
Experimental Results

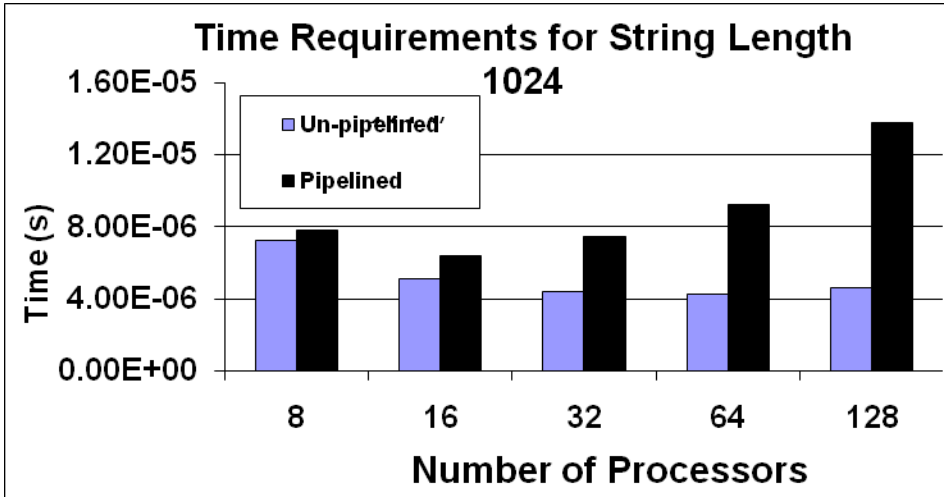
- **Input Data:** Arbitrary DNA sequences of length 1024 each.
- **Experimental Setup:** NoC implementations for PP and AD algorithms.
  - PEs and switches of the NoC implemented by synthesizing VHDL RTL using Synopsys Design Compiler and 90nm libraries .
  - The switches designed using Cadence Spectra.
  - To reduce delay => instead of building a multi-hop pipelined communication link, a direct (Bypass) strategy adopted.
  - Communication Schemes: Pipelined and Un-pipelined

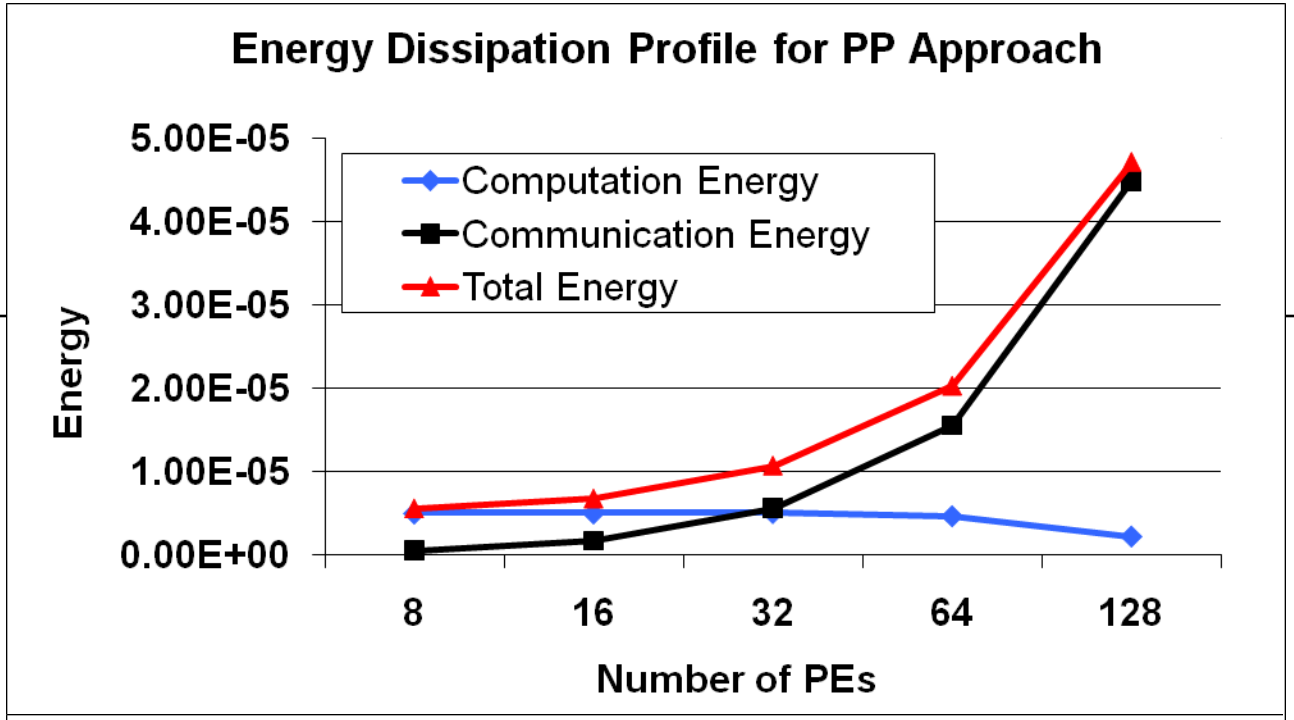
Experimental Results

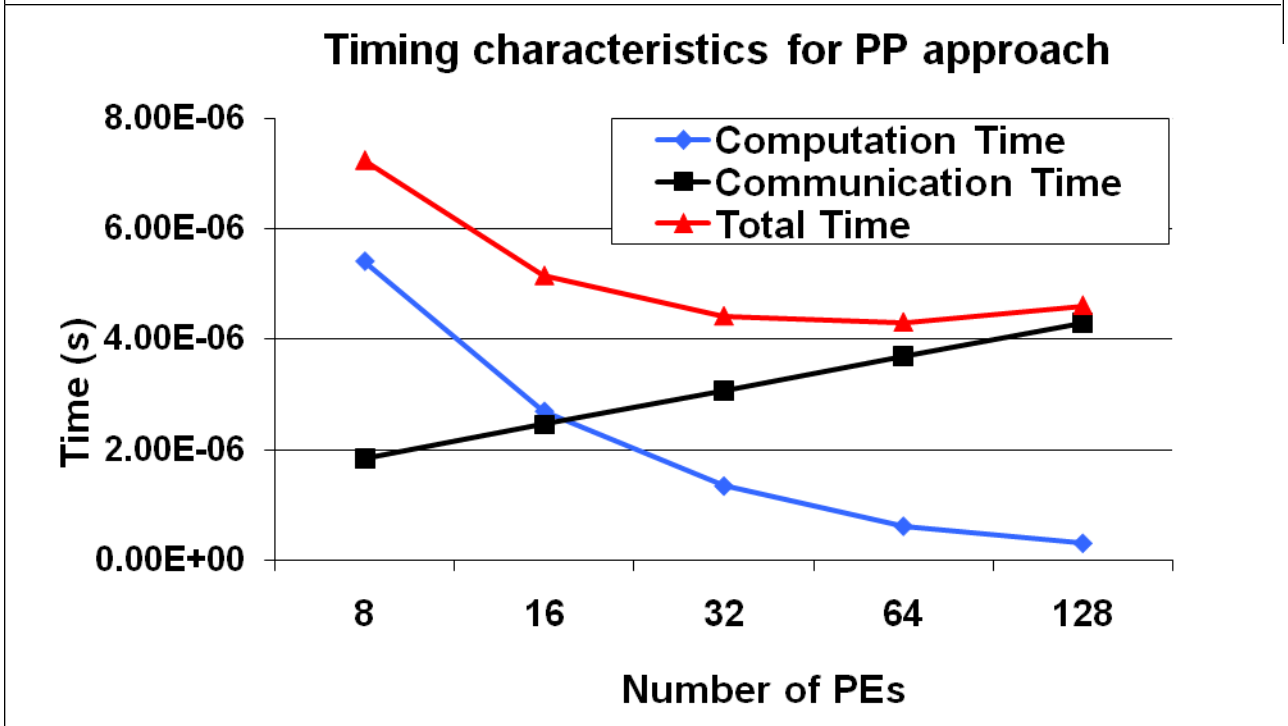
- Implementation
  - Pipelined
  - Un-pipelined



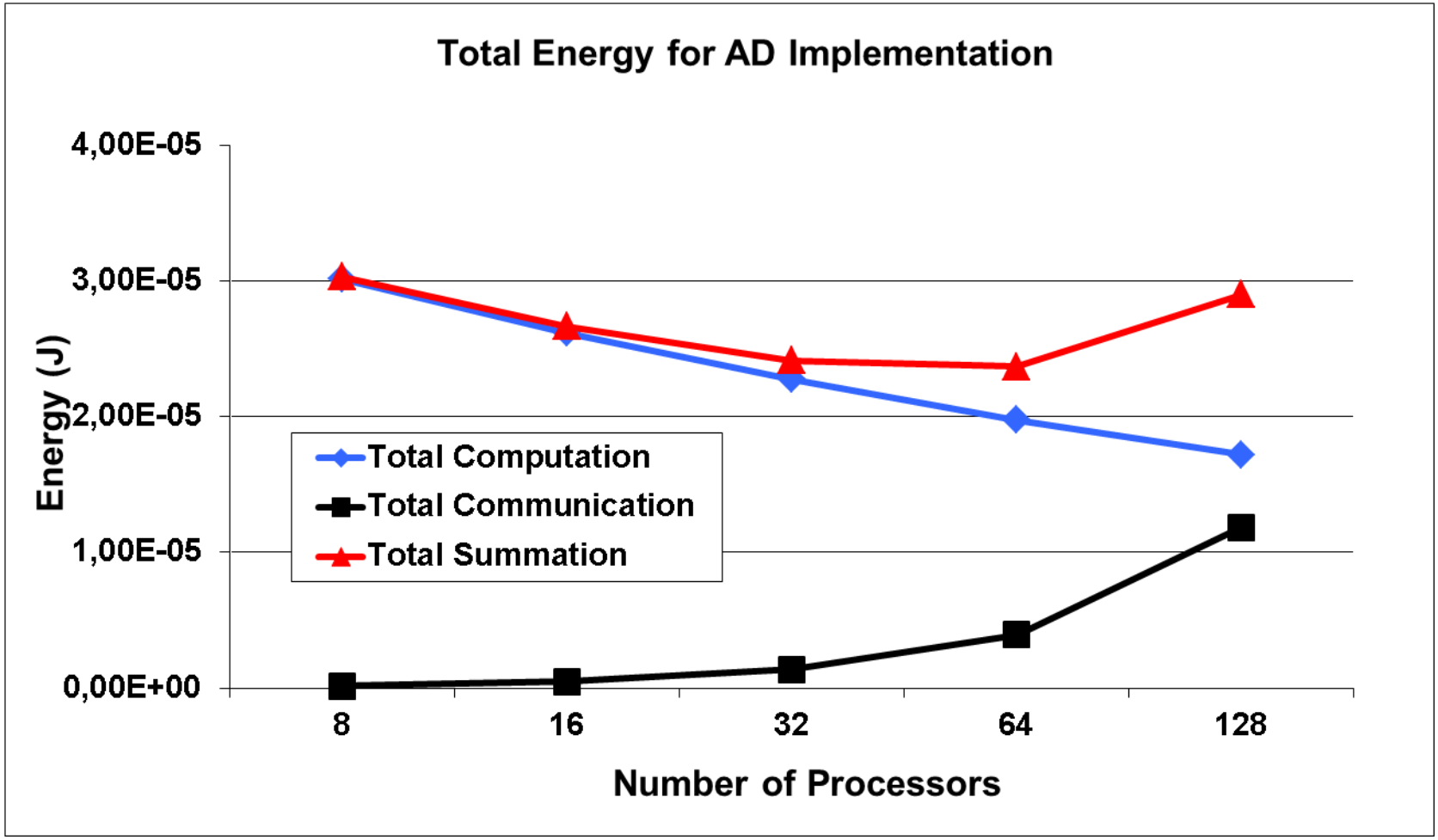




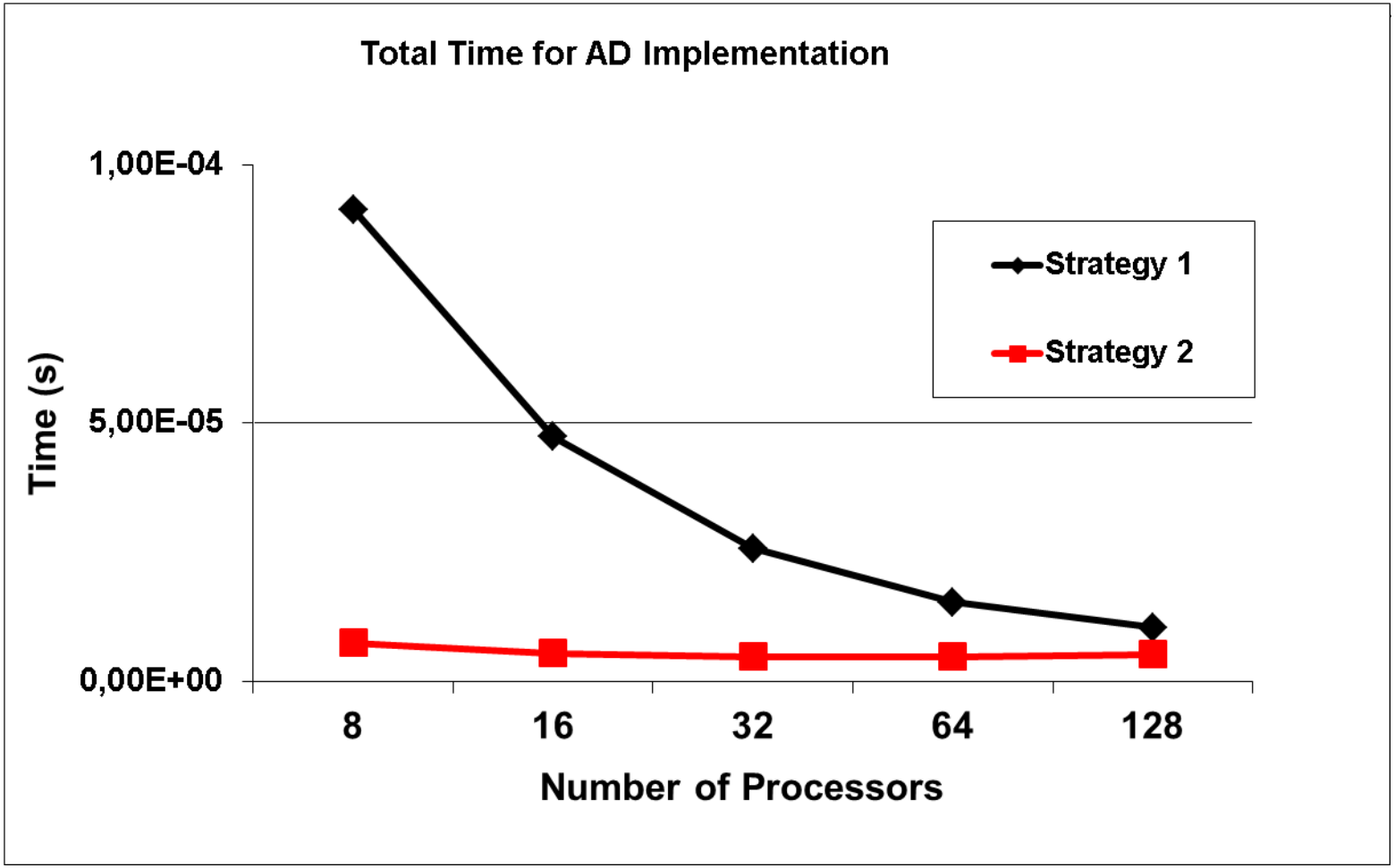

Energy Dissipation Profile for PP Approach
5.00E-05
4.00E-05
3.00E-05
2.00E-05
1.00E-05
0.00E+00
Energy
Computation Energy
Communication Energy
Total Energy
8
16
32
64
128
Number of PEs


Timing characteristics for PP approach
8.00E-06
6.00E-06
4.00E-06
2.00E-06
0.00E+00
Time (s)
Computation Time
Communication Time
Total Time
8
16
32
64
128
Number of PEs

Energy characteristics for AD



Timing Characteristics



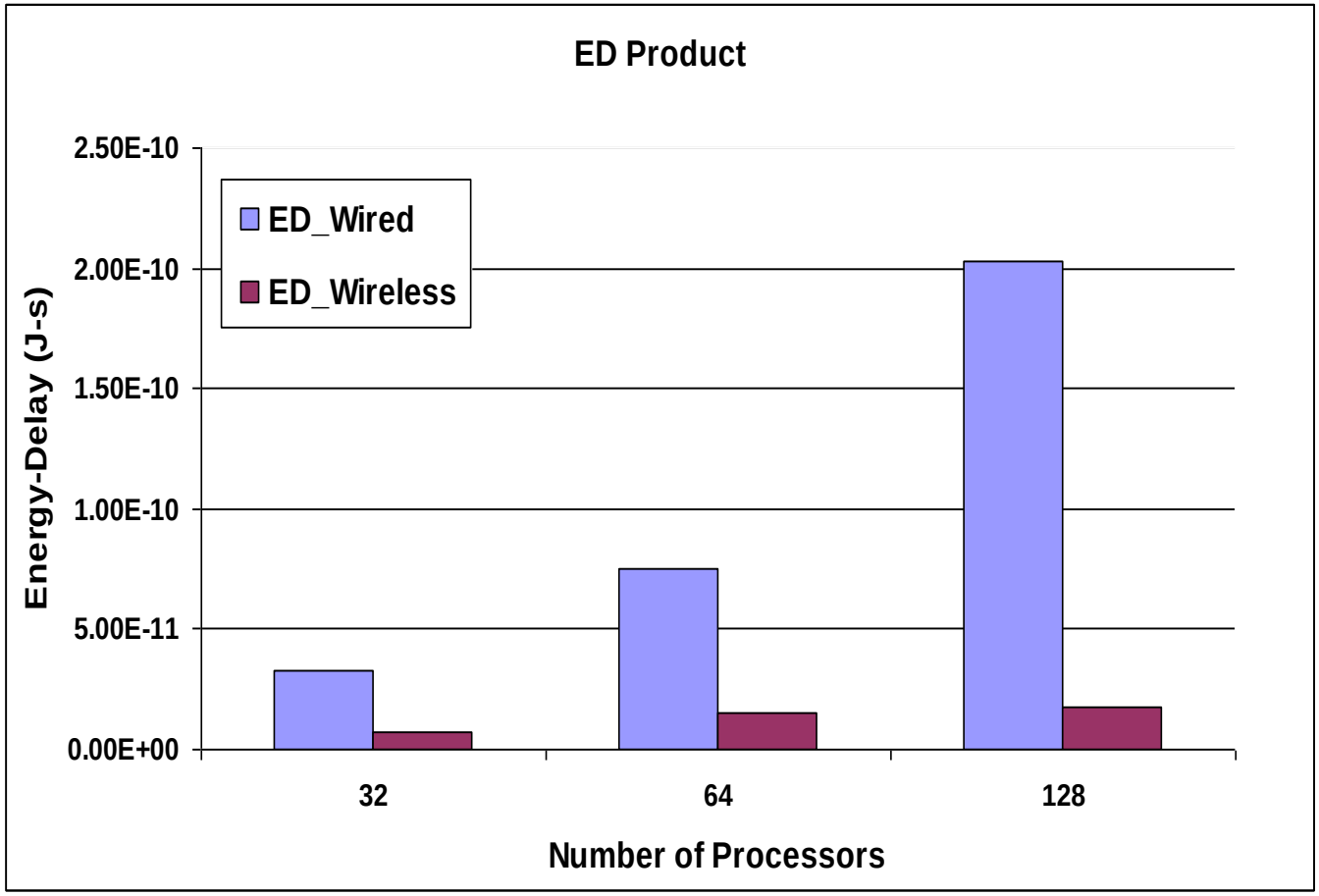
Speedup of Various Accelerators over our serial implementation

Serial implementation on a 2.3GHz Xeon CPU

	Intel 2.3GHz Xeon CPU	GPU	CBE	CBE	FPGA	OUR NoC IMPLEMENTATION	
						PP	AD
Time (ms)	**100**	**1.43**	**0.65**	**17.5**	**1**	**0.00439**	**0.00478**
Speedup over serial implementation	**1**	**69.93**	**153.85**	**5.7**	**100**	**22779.04**	**20920.5**

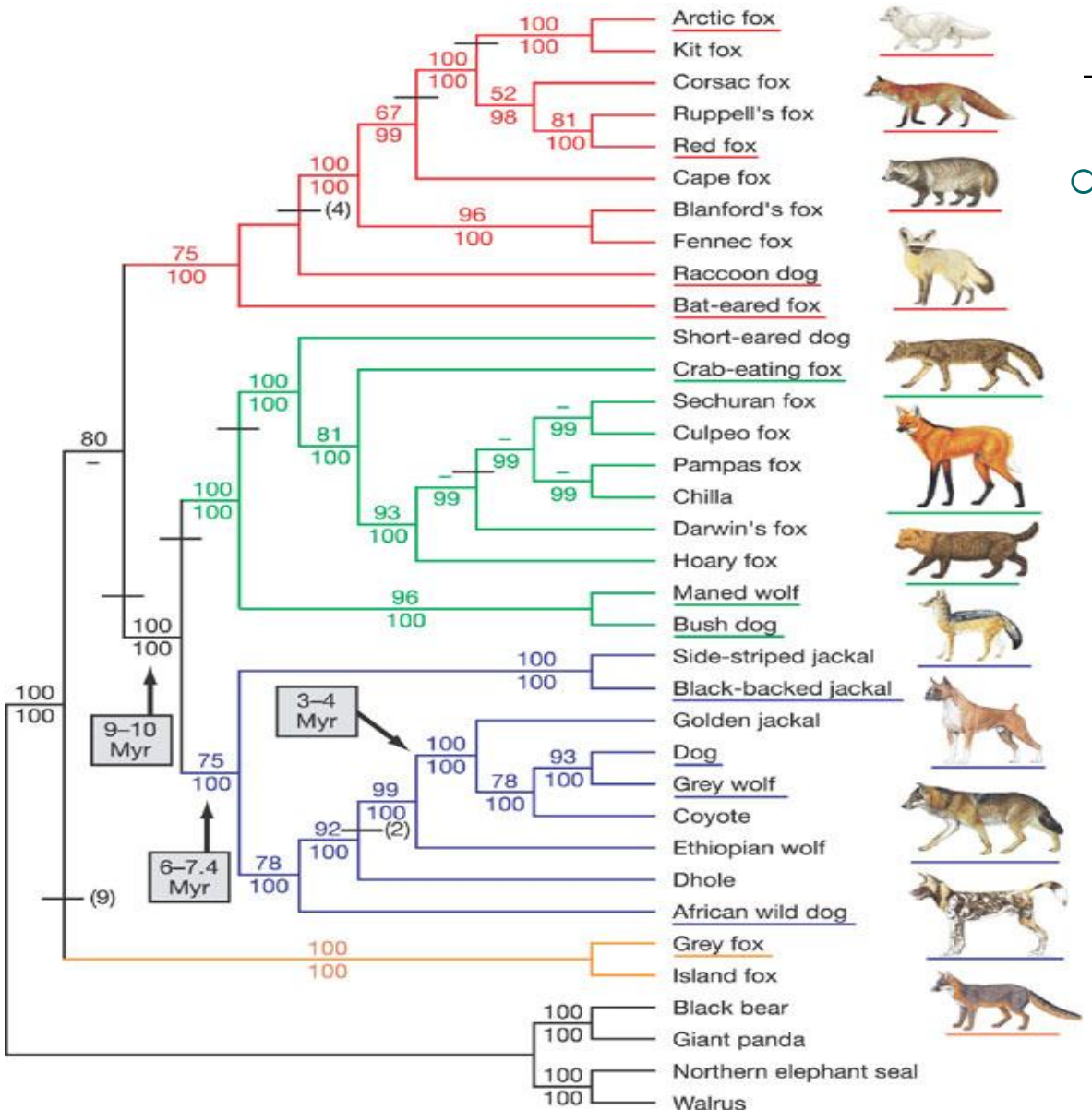
2010 **Souradip Sarkar**, Gaurav Ramesh Kulkarni, Partha Pratim Pande**,** Ananth Kalyanaraman, "NetworkonChip Hardware Accelerators for Biological Sequence Alignment", *IEEE Transactions on Computers***,** 59(1): 29-41**.**

Energy Delay Product using wireless network infrastructure



Application – Phylogenetic Reconstruction

Phylogenetic Tree – An Example

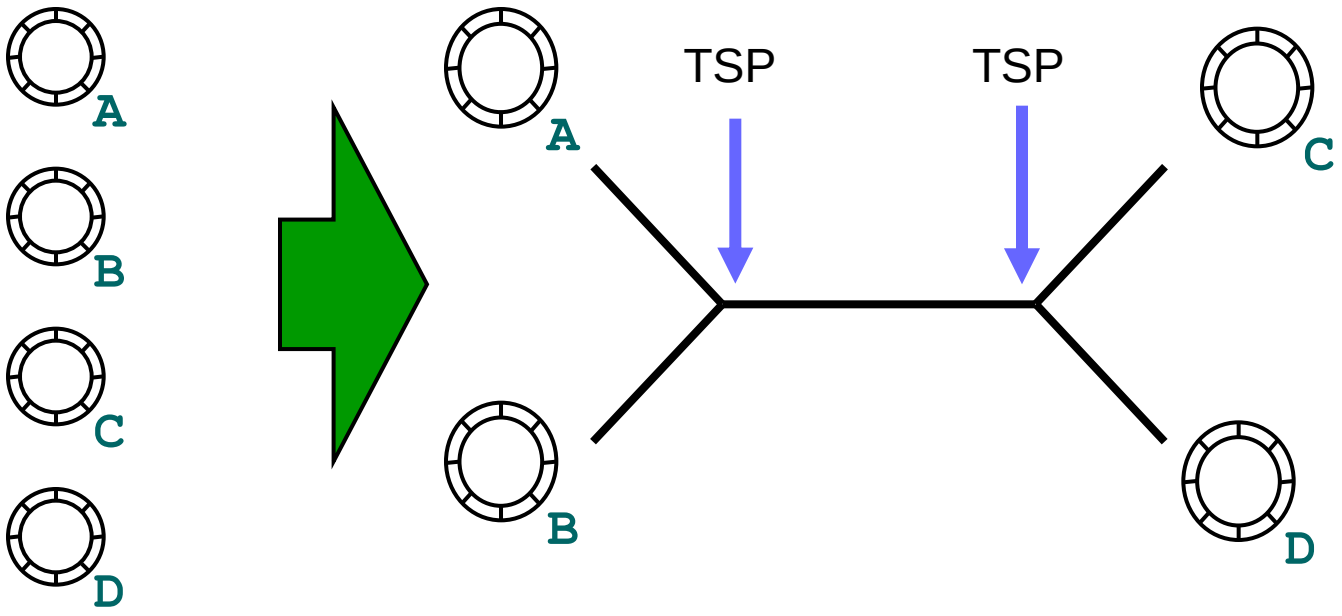


- Important for:
  - Biomedical research
  - Drug design
  - Protein structure prediction

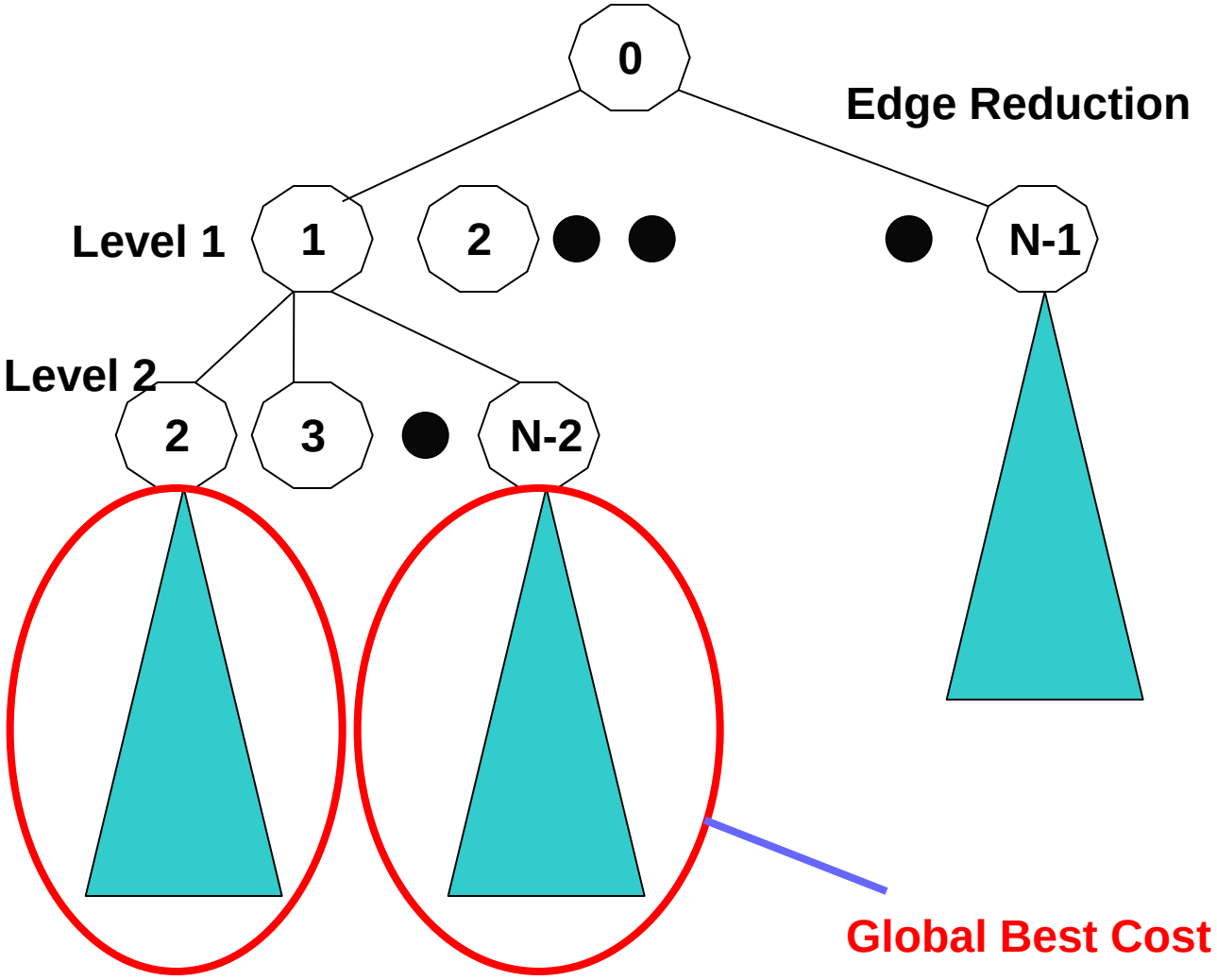
K. L. Toh et al, "Genome sequence, comparative analysis and haplotype structure of the domestic dog", Nature 438, 803-819 (8 December 2005)

Maximum Parsimony

- Inferring the phylogeny of a set of *N species* => reconstructing a phylogenetic tree (on distance or probability measures)



Traveling Salesman Problem



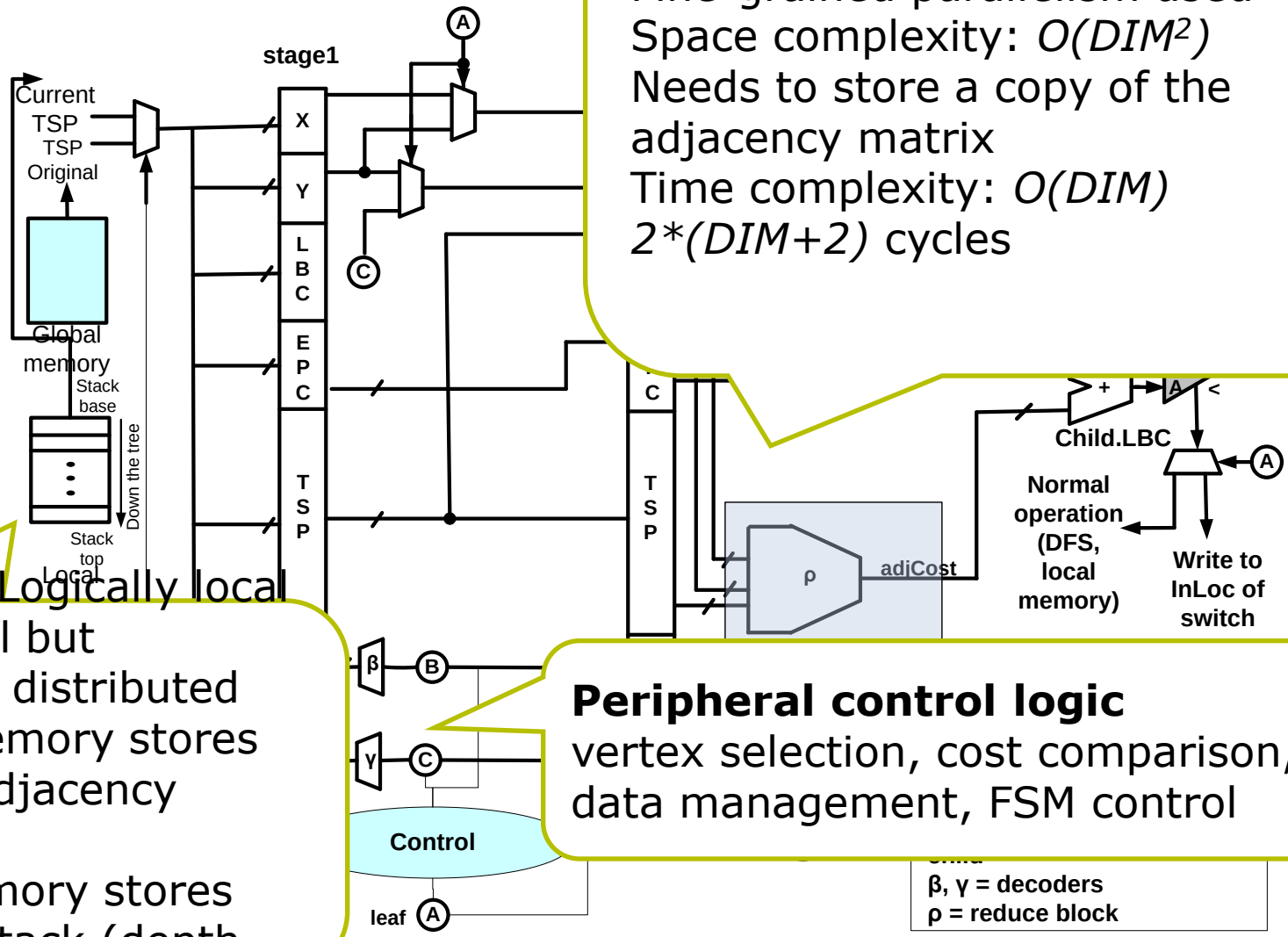
Reduction operation

- ***R[i,k] = R[k,j] = R[j,1] =*** ∞ *for all 1≤k≤DIM*

	1	2	j=3	4	5	6
1	**0**	**3**	**1**	**3**	**3**	**3**
i=2	**3**	**0**	**3**	**3**	**2**	**3**
3	**1**	**3**	**0**	**3**	**1**	**2**
4	**3**	**3**	**2**	**0**	**2**	**3**
5	**1**	**1**	**2**	**3**	**0**	**3**
6	**3**	**3**	**3**	**3**	**2**	**0**

	1	2	j=3	4	5	6
1	**0**	**3**	∞	**3**	**3**	**3**
i=2	∞	∞	∞	∞	∞	∞
3	∞	**3**	∞	**3**	**1**	**2**
4	**3**	**3**	∞	**0**	**2**	**3**
5	**1**	**1**	∞	**3**	**0**	**3**
6	**3**	**3**	∞	**3**	**2**	**0**

PE Architecture

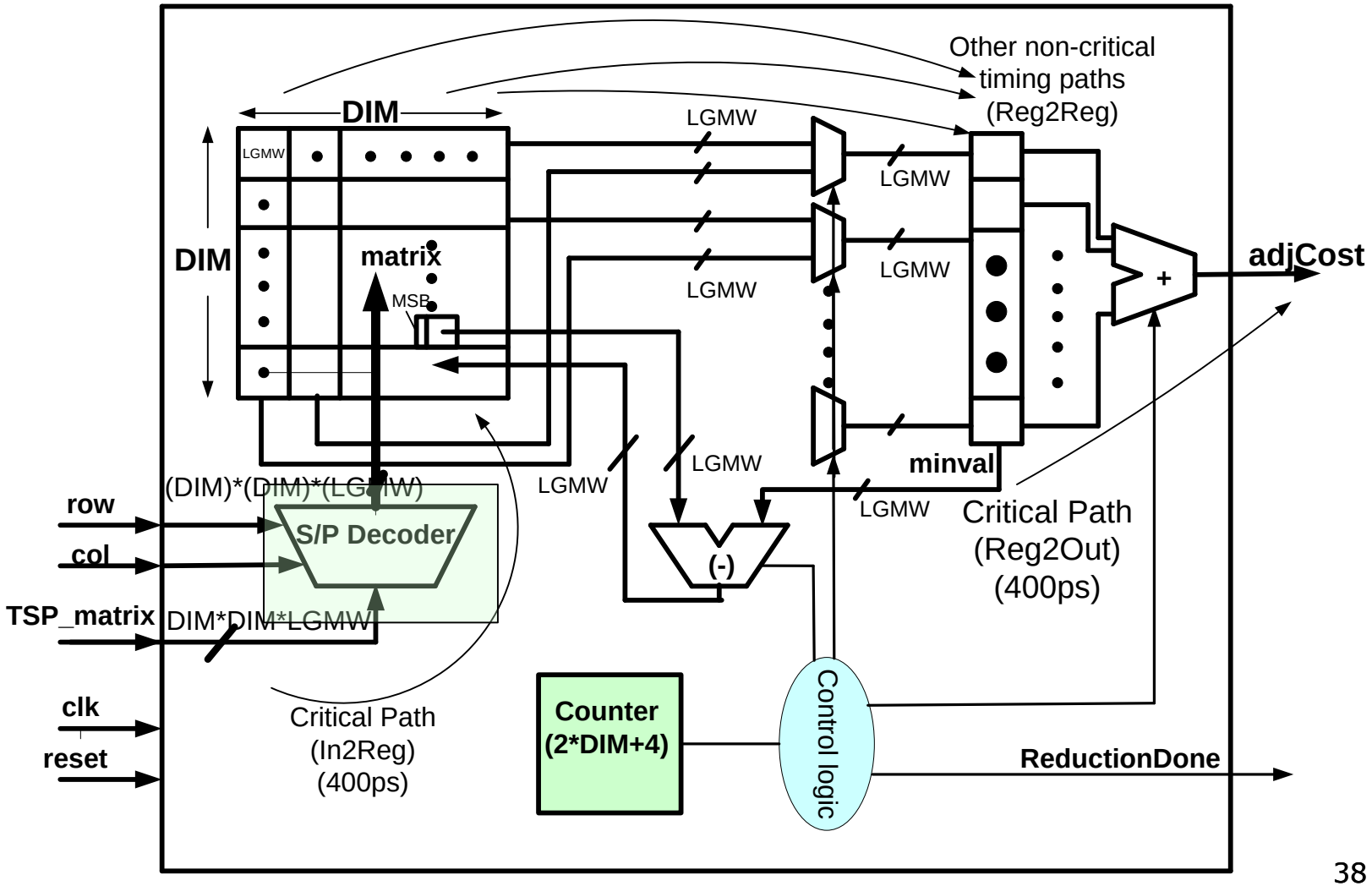


Reduction block
Fine-grained parallelism used
Space complexity: *O(DIM²)*
Needs to store a copy of the adjacency matrix
Time complexity: *O(DIM)*
2(DIM+2)* cycles

Memory Logically local and global but physically distributed
Global memory stores subtree adjacency matrix.
Local memory stores the DFS stack (depth =

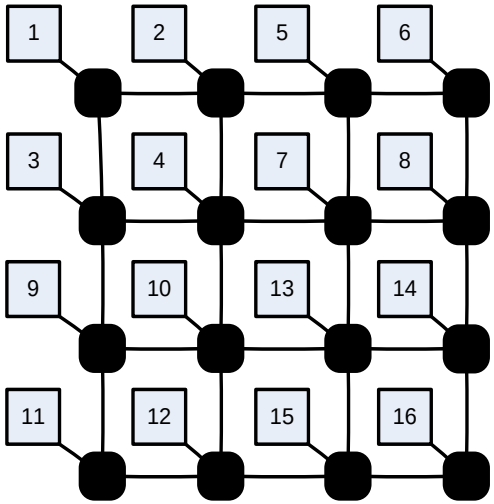
Peripheral control logic
vertex selection, cost comparison, data management, FSM control

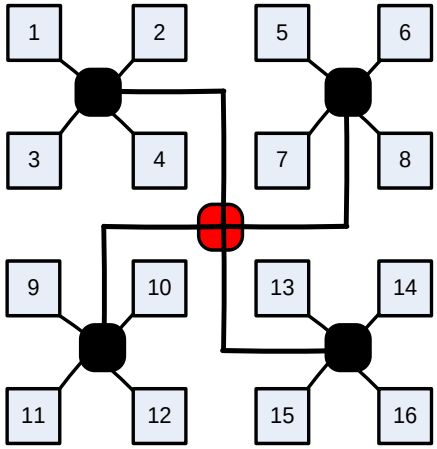
Reduce Block

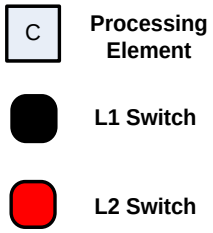


LGMW = $\log_2 MW$

Network Topologies





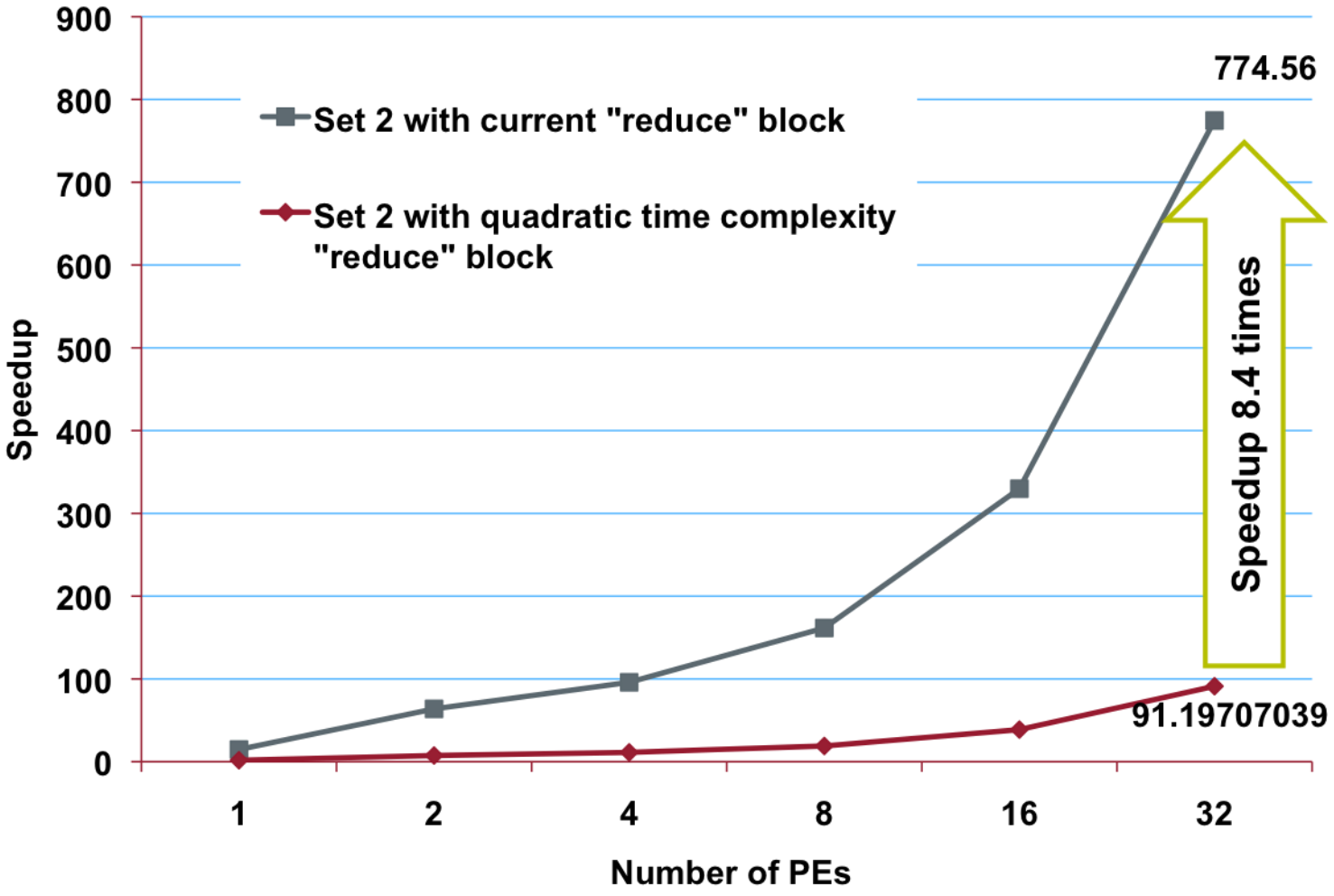


- Explored two different network architectures – the mesh and the quad-tree.
- Diameter of the network determines worst-case communication latency.

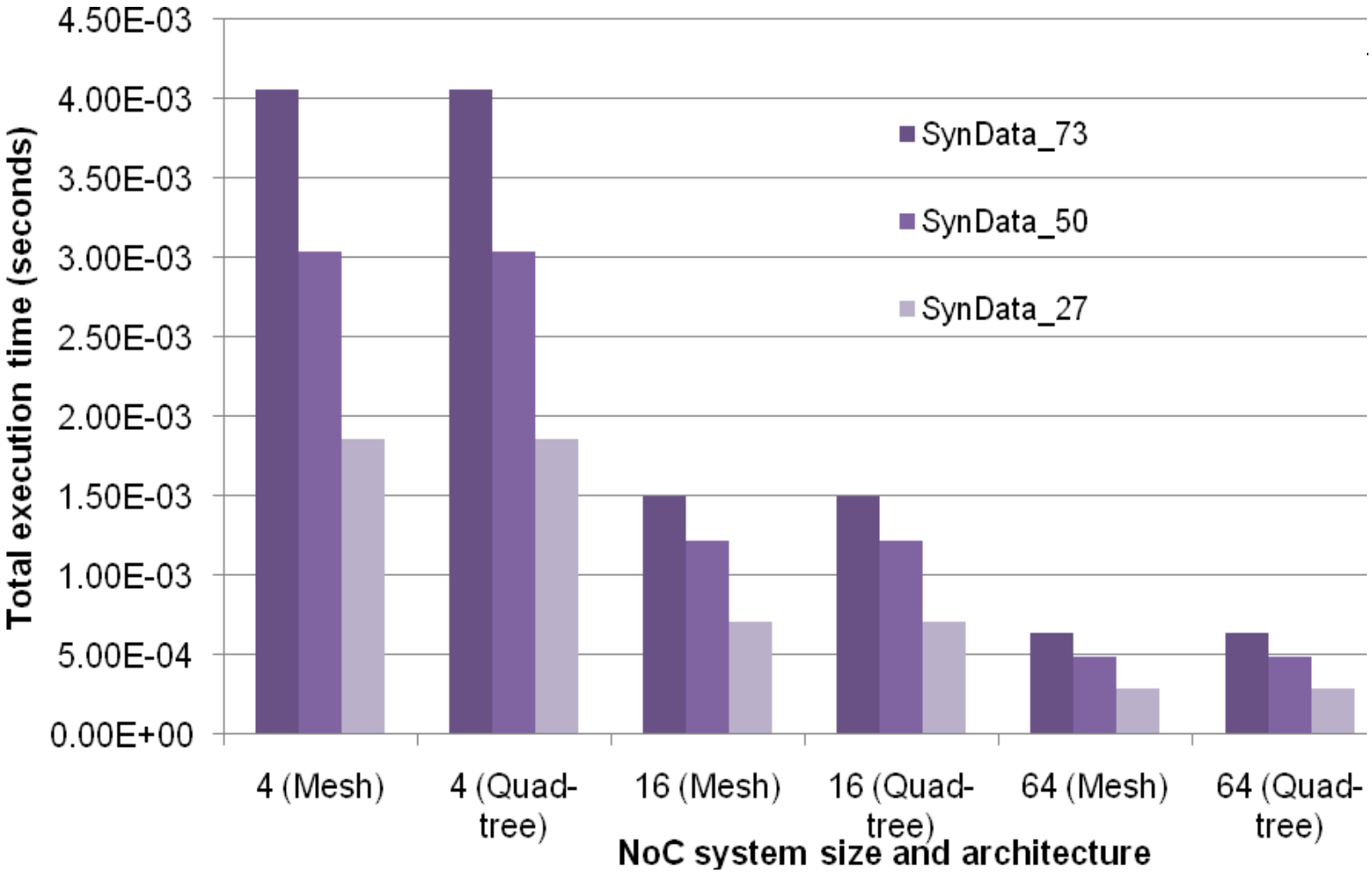
Worst Case Write Latency

N	Mesh	Quad-tree
4	6	6
8	9	10
16	12	10
64	14	12
256	30	14
1024	62	16

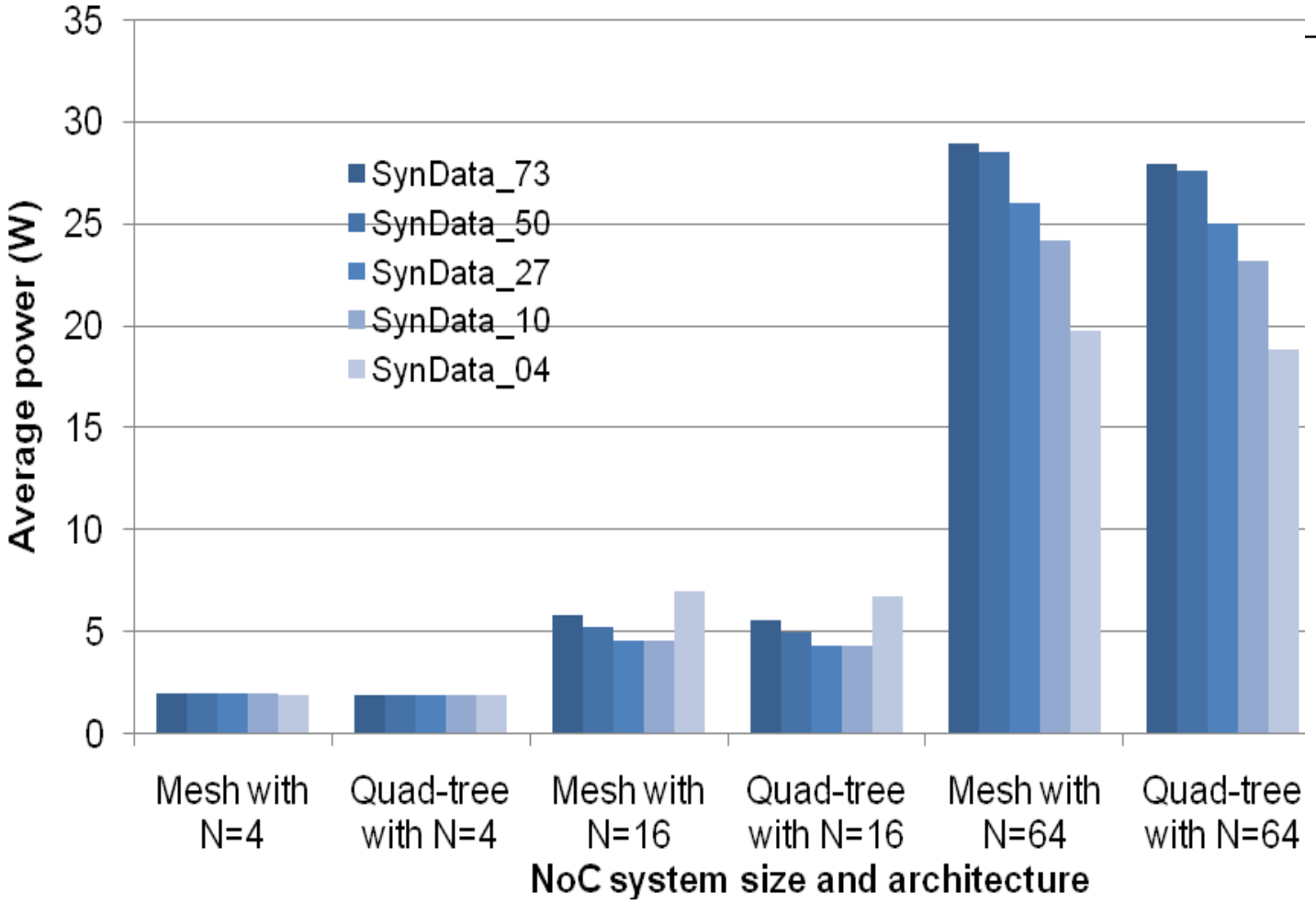
Advantage of linear time matrix reduction



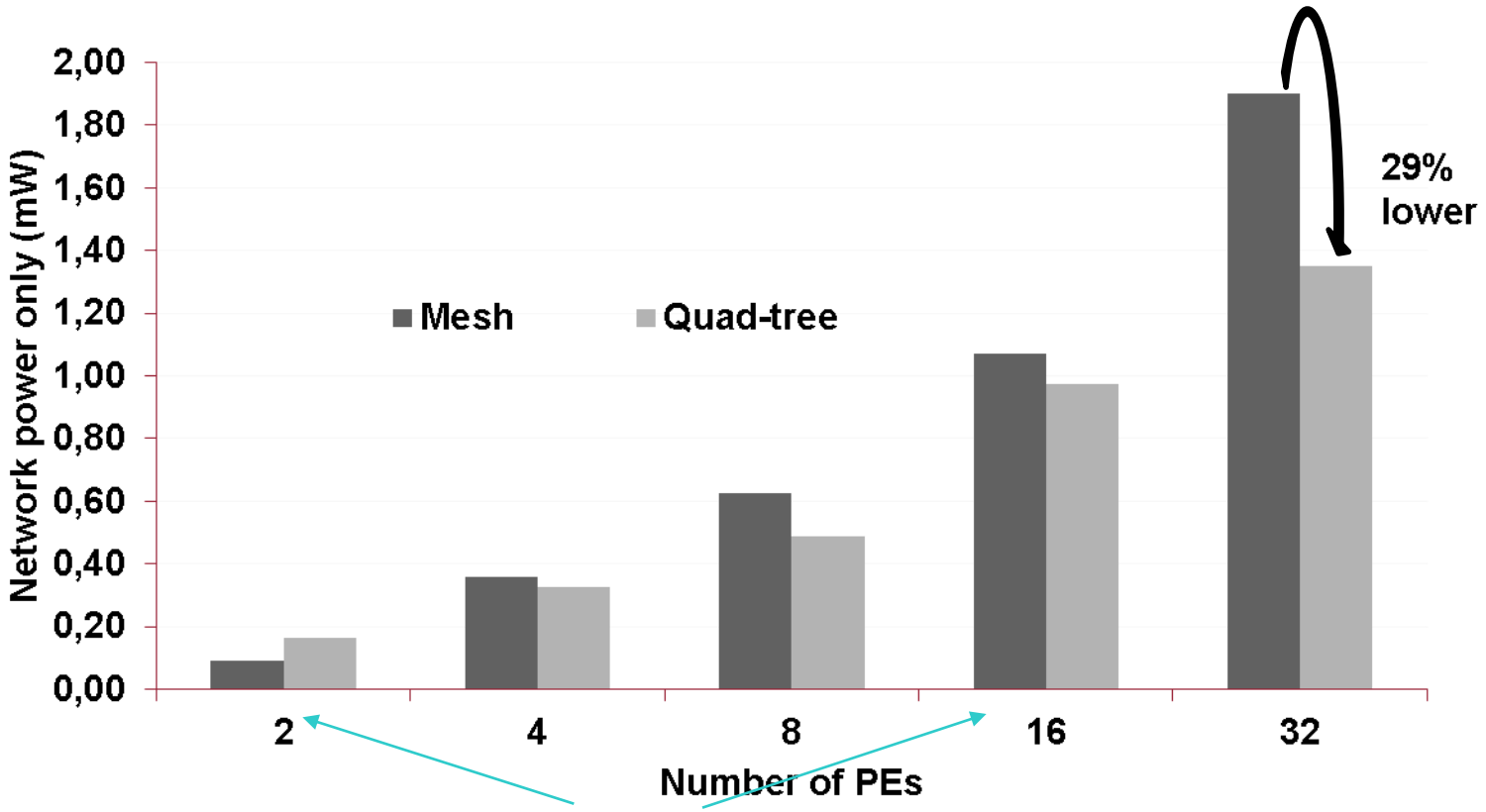
Execution Time



Power consumption for 16-vertex graph



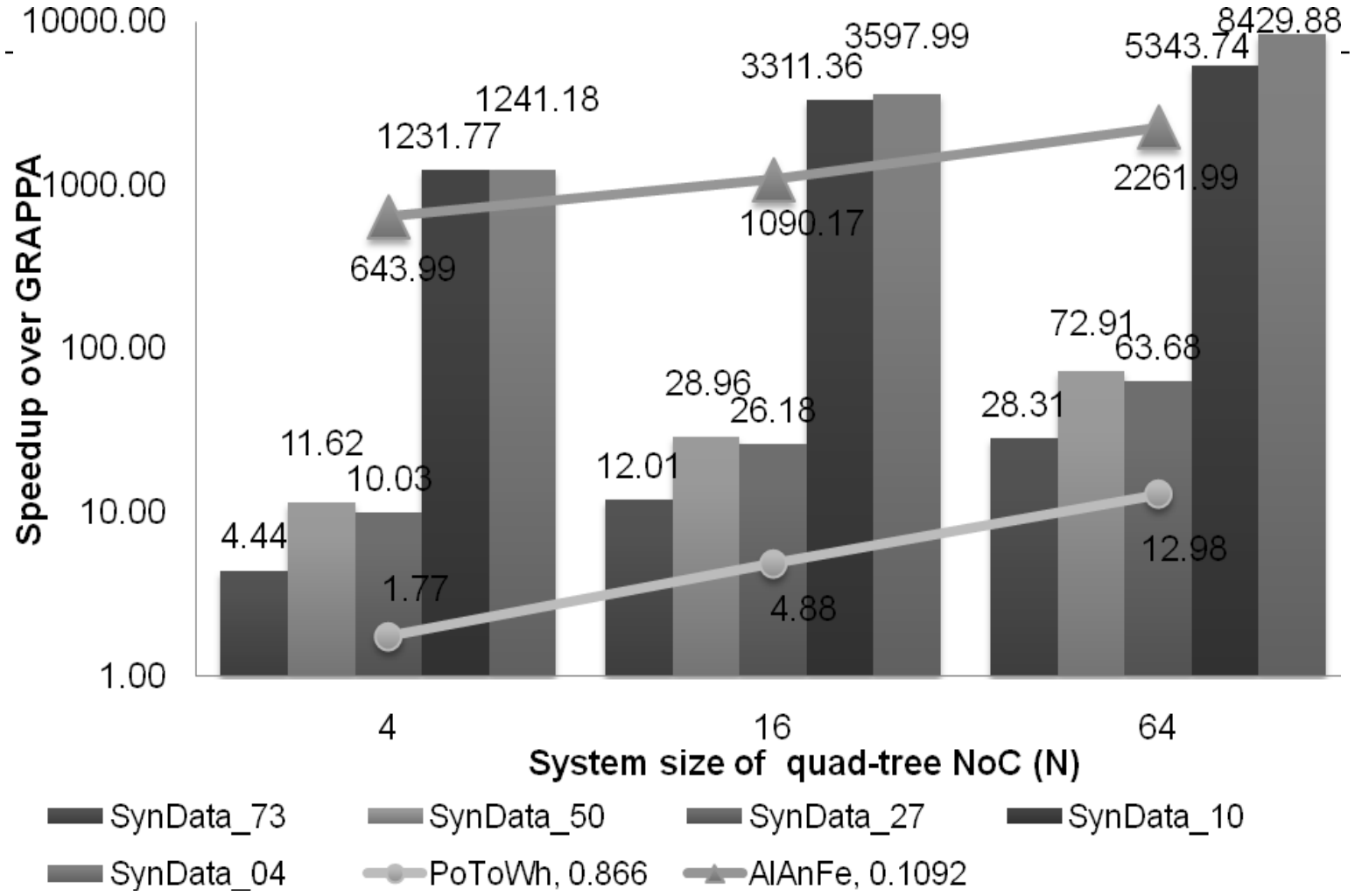
Interconnect power consumption (16-vertex graph)



Complete tree

43

Speedup


10000.00
1000.00
100.00
10.00
1.00
Speedup over GRAPPA
4.44
11.62
10.03
1231.77
1241.18
643.99
1.77
12.01
28.96
26.18
3311.36
3597.99
1090.17
4.88
28.31
72.91
63.68
5343.74
8429.88
2261.99
12.98
4
16
64
System size of quad-tree NoC (N)
SynData_73
SynData_50
SynData_27
SynData_10
SynData_04
PoToWh, 0.866
AlAnFe, 0.1092

Conclusions

- To the best of our knowledge, this is the first NoC-based approach to tackle these biocomputing problems.
- SA:
  - Designed and characterized NoC architectures for PP and AD.
  - Evaluate performance of NoC architectures with different types of long-range links. Bypass links ->Low power and high-speed intra-chip interconnects (wireless links).
- MP:
  - Designed a multi-threaded software for MP.
  - Designed NoC Architecture and currently comparing with real Phylogenetic data and other hardware solutions.
  - Compare the timing performance of our serial and multithreaded code with that of the existing platform GRAPPA.
  - Benchmarking against real phylogenetic data.

Hardware/Software Co-Optimization (Current)

Computer systems are increasingly power/energy-constrained

Workload-optimized system design is important paradigm

- Widely employed for smartphones, tablets, game machines, data centers, supercomputers, etc.

Hardware/software co-optimization allows for even higher levels of performance and power/energy-efficiency

- widely adopted for embedded system design, yet to be explored for high-performance processor design

Challenging simulation requirements

- Efficient architecture exploration
- Simulate entire workload and simulate different versions of the workload

Sniper/McPAT – a fast and accurate simulation environment for hardware/software co-optimization in early design stages

Fast and Accurate Simulation is Needed

- Simulation use cases
  - Architecture exploration
  - Pre-silicon software optimization
- Current practice: cycle-accurate simulation
  - Too slow for exploring multi/many-core design space and software
- Key questions
  - Can we raise the level of abstraction?
  - What is the right level of abstraction?
  - When to use these abstraction models?

Power Predictions with McPAT

- McPAT: “Multi-Core Power, Area and Timing”

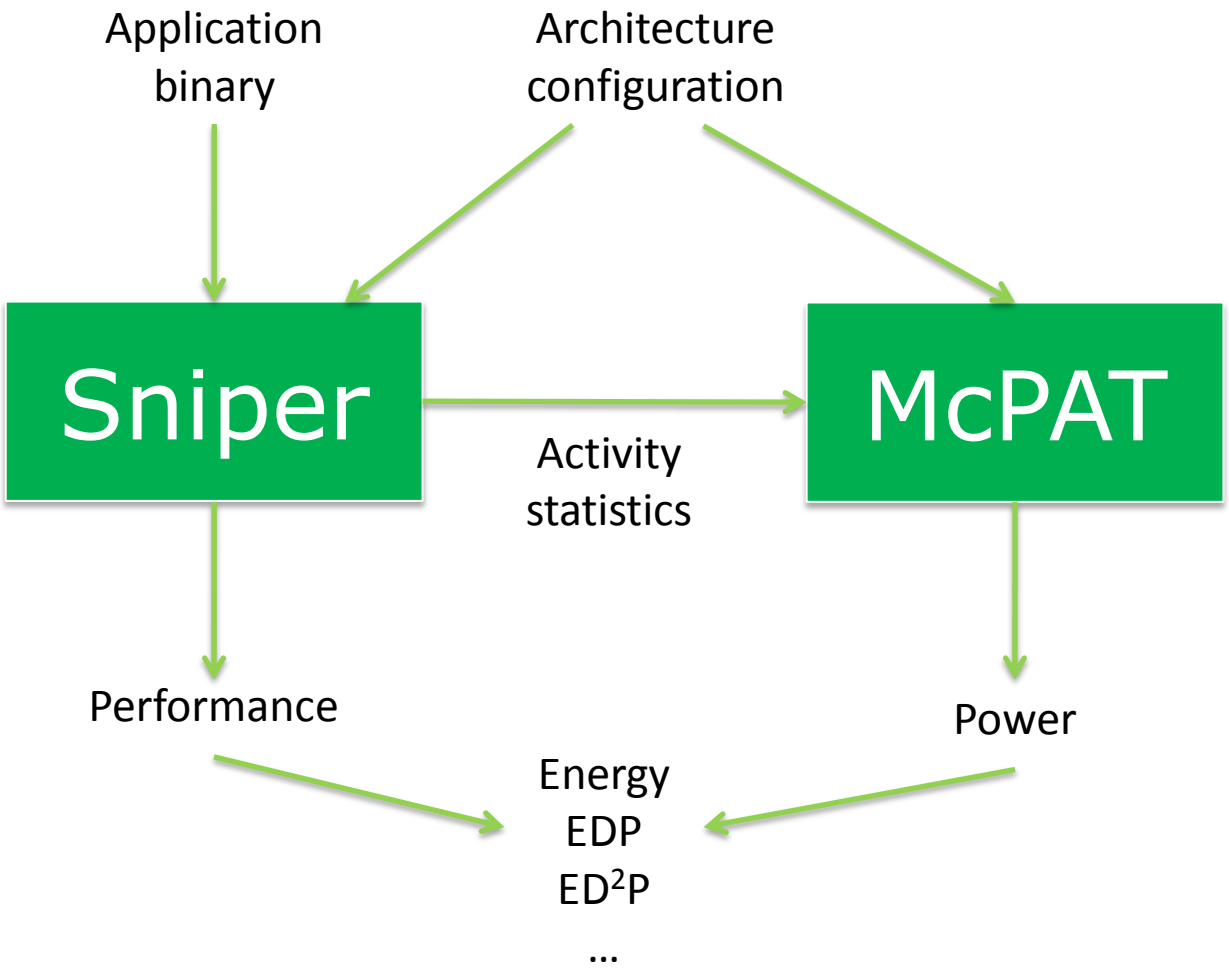
  [Li et al., MICRO 2009]

- Integrated power, area and timing estimates for multi-core processors

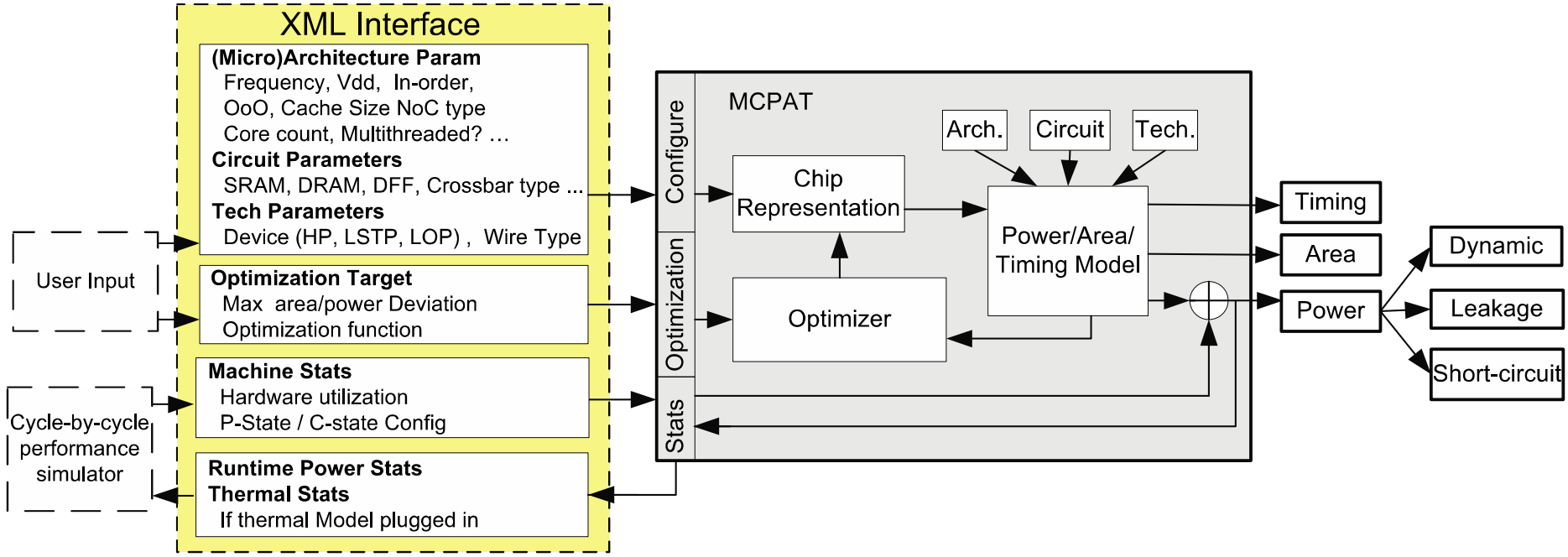
  using CACTI (caches) and ORION (NoC) models

- Input: architecture description, activity factors, event counts
- Output: per-component power (static, dynamic), area estimates

Sniper/McPAT Simulation Framework



McPAT framework block diagram



Sniper/McPAT Simulation Framework

High abstraction modeling

- Some McPAT inputs are not readily available given Sniper's high abstraction level, e.g., # ROB reads
- Solution:
  - Simple estimation (e.g., 2 ROB reads per instruction) proved sufficient
  - Duty cycle of ALUs: use instruction mix and ALU throughput (e.g., each FMUL instruction uses FP ALU for one cycle, divide by total cycles)

Memory power model

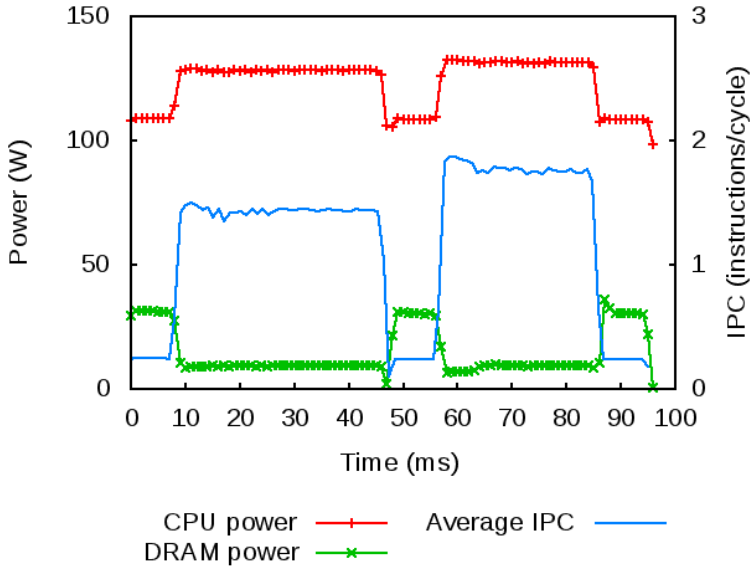
- McPAT lacks a memory power model
- Solution: Micron DDR3 DRAM model

Sniper/McPAT Simulation Framework

Results:

Sniper: performance estimates, CPI stacks

McPAT: per-component power & area estimates



Simulation speed:

Sniper: unaffected

McPAT: fixed, O(1 minute)

Power model is basically free!

Unlike detailed cycle-accurate performance/power simulation, see for example Wattch w/ 30% overhead

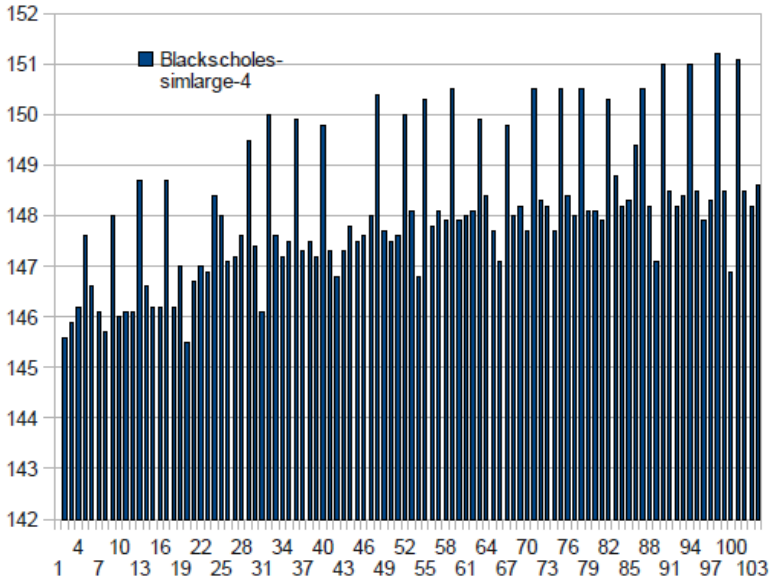
Validation: Methodology

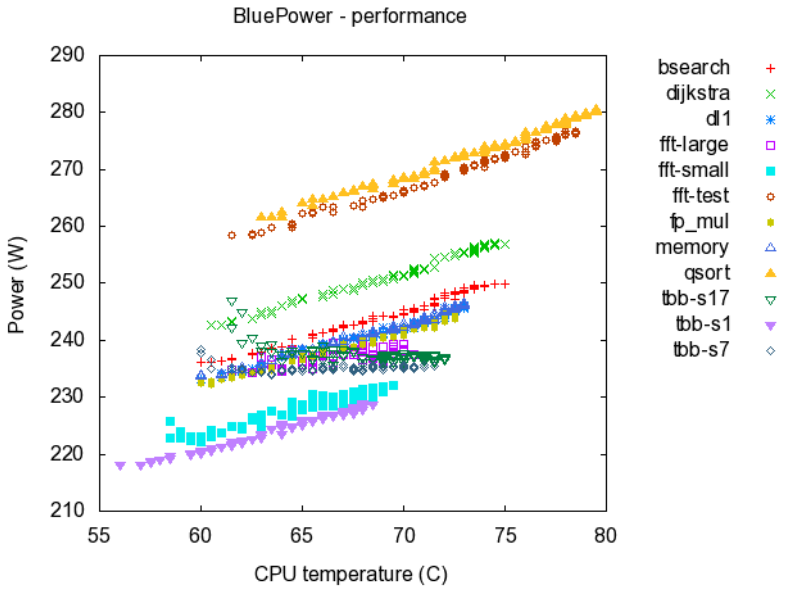
One HW power measurement per second

- Select benchmarks with long parallel region
- Subtract server idle power
- Apply temperature correction (static power)
- Compute average dynamic power when thermal equilibrium is reached

McPAT + DRAM power model

- Compute CPU + DRAM dynamic power of workload's parallel region
- Validate against HW measurments





Relevant Publications

Journals

1) 2010 **Souradip Sarkar**, Gaurav Ramesh Kulkarni, Partha Pratim Pande**,** Ananth Kalyanaraman, "Network on Chip Hardware Accelerators for Biological Sequence Alignment", *IEEE Transactions on Computers***,** 59(1): 29-41**.**

2) 2012 Turbo Majumder, **Souradip Sarkar**, Partha Pratim Pande, Ananth Kalyanaraman, “NoCBased Hardware Accelerator for Breakpoint Phylogeny”, *IEEE Transactions on Computers ,* 61(6): 857-869 (2012).

Conferences

1) 2010 Turbo Majumder, **Souradip Sarkar**, Ananth Kalyanaraman, Partha Pratim Pande, “An Optimized NoC Architecture for Accelerating TSP Kernels in Breakpoint Median Problem” *Proceedings of 21st IEEE International Conference on Application Specific System Architectures and Processors,* ASAP: 8996.

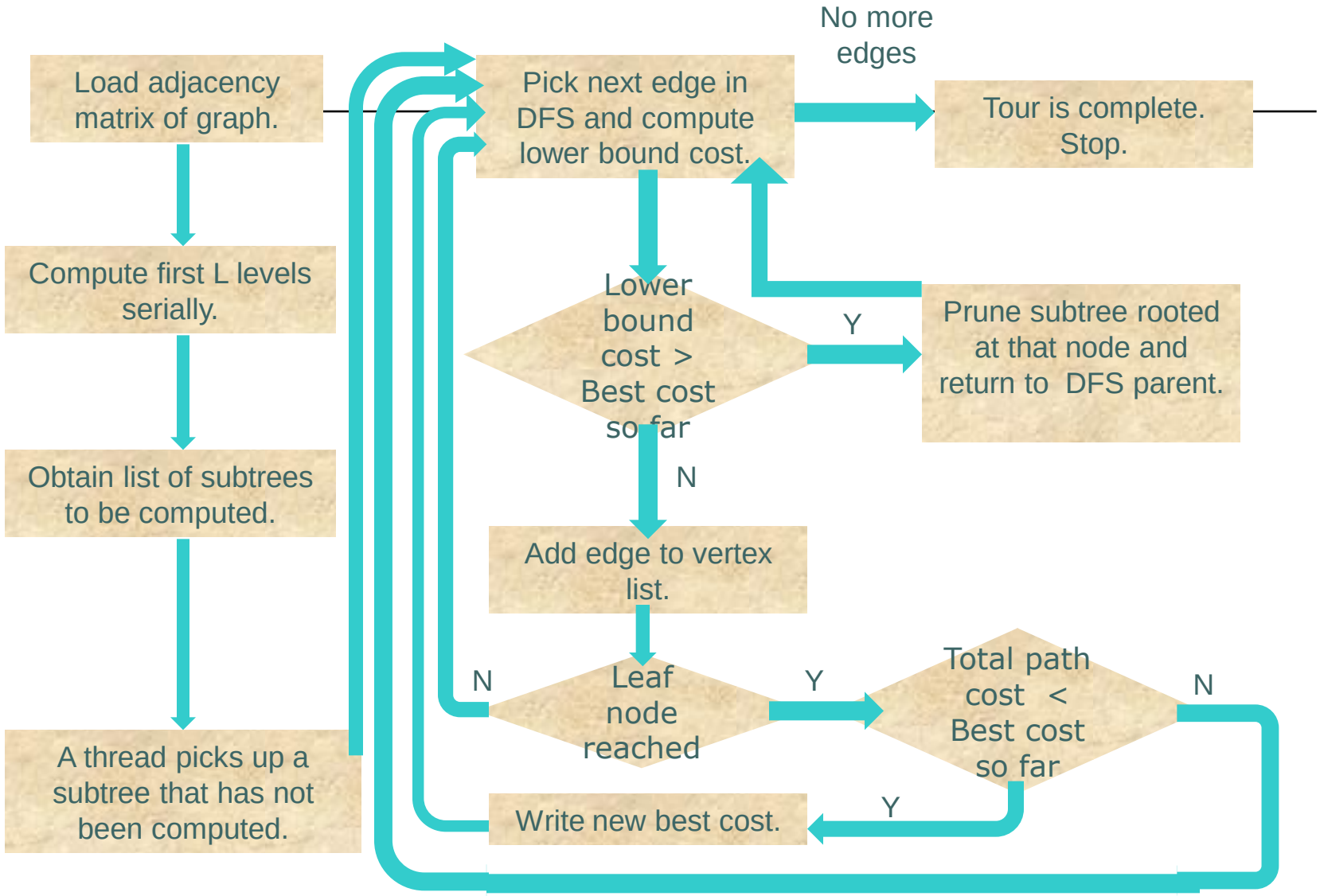
2) 2010 **Souradip Sarkar**, Turbo Majumder, Ananth Kalyanaraman, Partha Pratim Pande, “Hardware accelerators for Biocomputing: A Survey” *Proceedings of 2010 IEEE International Symposium on Circuits and Systems,* ISCAS: 37893792.

3) Wim Heirman, **Souradip Sarkar**, Trevor E. Carlson, Ibrahim Hur, Lieven Eeckhout: Power-aware multi-core simulation for early design stage hardware/software co-optimization. PACT 2012: 3-12.

Thank You

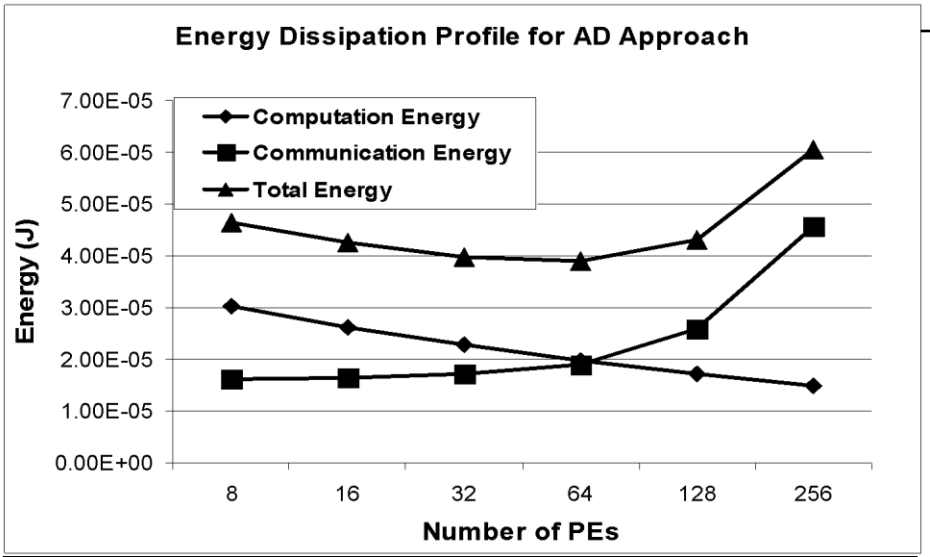
- Questions
- Comments

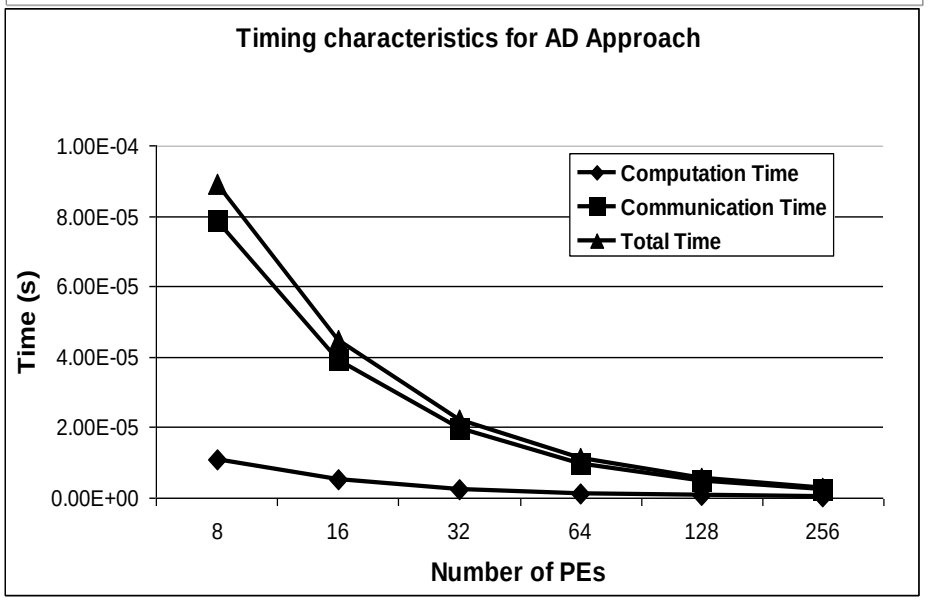
Algorithm



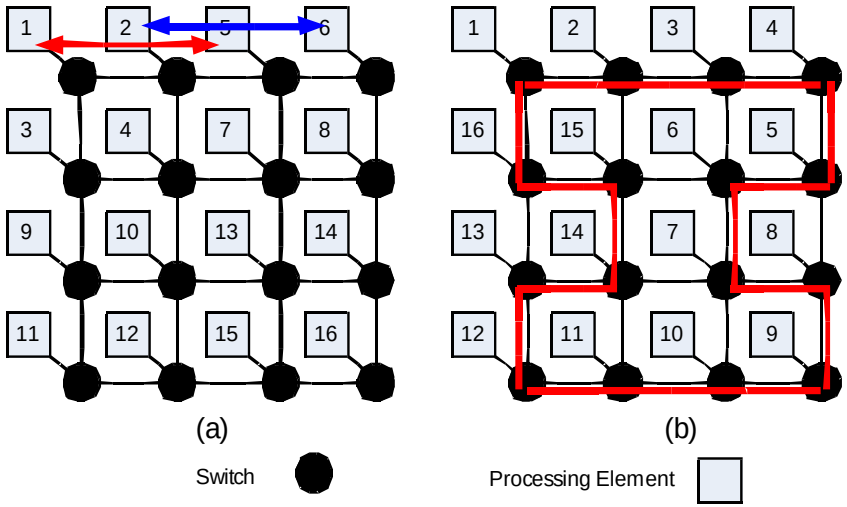
E. Horowitz and S. Sahni, “Branch-and-bound” in Fundamentals of computer algorithms, Potomac, MD: Computer Science Press, 1984, pp. 370-421.

Energy and Timing characteristics for AD





Anti-diadonal Setup



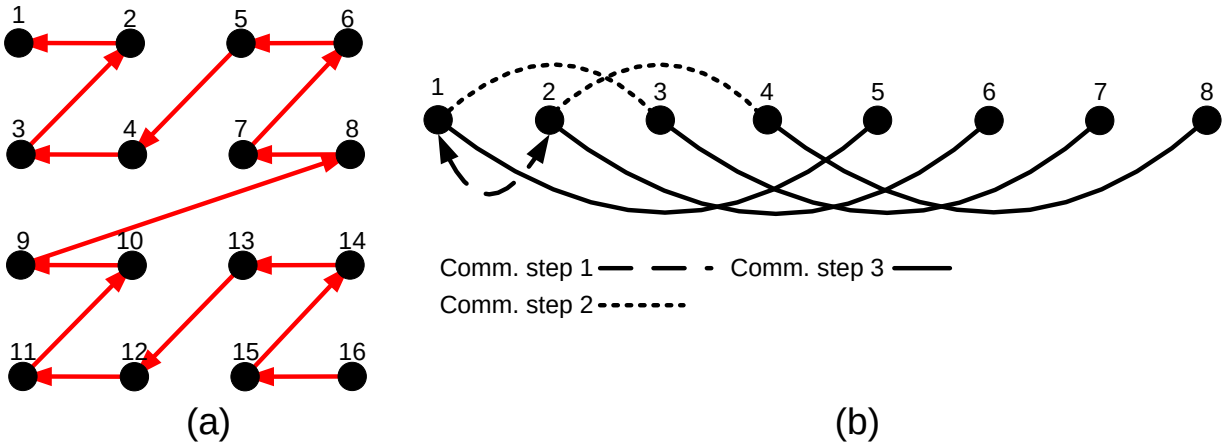
Speedup over existing Hardware Accelerators

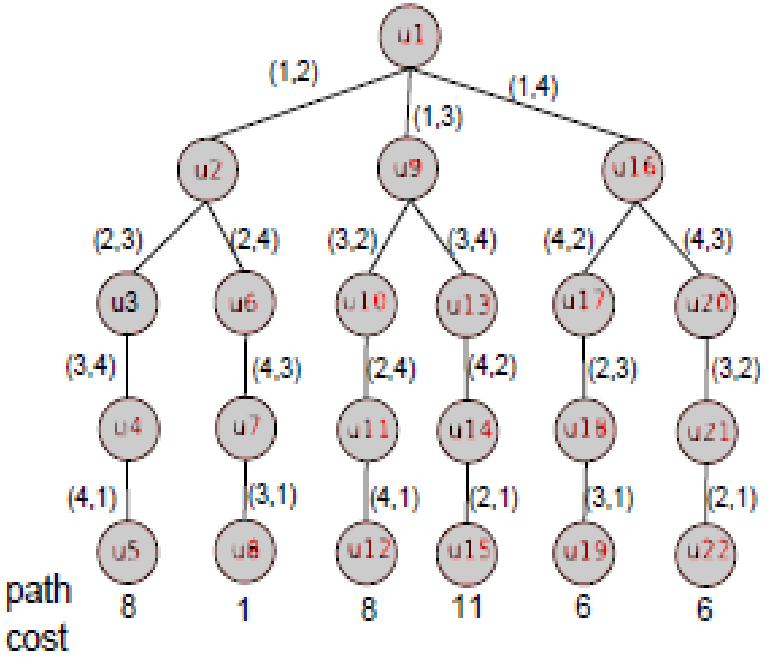
Other Accelerators		FPGA	CBE	CBE	GPU
Our Implementation	PP	227.79	148	3986.3	325.74
	AD	94.87	61.67	1660.3	135.67`

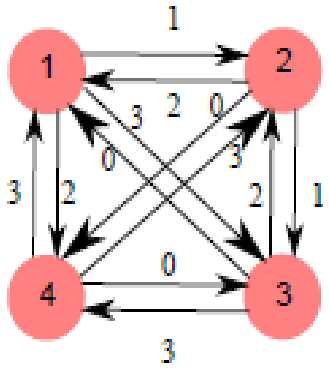
Performance on real phylogenentic data

	N = 4			N = 64		
	Avg. reductions per PE	Std. deviation of reductions per PE	Max reductions per PE	Avg. reductions per PE	Std. deviation of reductions per PE	Max reductions per PE
PoToWh	15672.75	1847.57	17430	1942.73	194.73	2342
AlAnFe	69222	8558.69	79516	19496.84	1700.02	22614

Communication pattern for the Backward pass for both PP and AD




u1
(1,2)
(1,3)
(1,4)
u2
u9
u16
(2,3)
(2,4)
(3,2)
(3,4)
(4,2)
(4,3)
u3
u6
u10
u13
u17
u20
(3,4)
(4,3)
(2,4)
(4,2)
(2,3)
(3,2)
u4
u7
u11
u14
u18
u21
(4,1)
(3,1)
(4,1)
(2,1)
(3,1)
(2,1)
u5
u8
u12
u15
u19
u22
path cost
8
1
8
11
6
6


1
2
4
3
1
3
2
0
3
0
3
2
2
1
0
3